FREE EBOOK

oliveboard

Tamil Nadu Textbook

Class 12 - Tamil Medium

Computer Science

For all TN Exams

தமிழ்நாடு அரசு

மேல்நிலை இரண்டாம் ஆண்டு

கணினி அறிவியல்

தமிழ்நாடு அரசு விலையில்லாப் பாடநூல் வழங்கும் திட்டத்தின்கீழ் வெளியிடப்பட்டது

பள்ளிக் கல்வித்துறை

தீண்டாமை மனித நேயமற்ற செயலும் பெருங்குற்றமும் ஆகும்

தமிழ்நாடு அரசு

முதல்பதிப்பு - *2019*
திருத்திய பதிப்பு - *2020*

(புதிய பாடத்திட்டத்தின்கீழ்
வெளியிடப்பட்ட நூல்)

விற்பனைக்கு அன்று

பாடநூல் உருவாக்கமும் தொகுப்பும்

மாநிலக் கல்வியியல் ஆராய்ச்சி
மற்றும் பயிற்சி நிறுவனம்

© SCERT 2019

நூல் அச்சாக்கம்

தமிழ்நாடு பாடநூல் மற்றும்
கல்வியியல் பணிகள் கழகம்

www.textbooksonline.tn.nic.in

Index.indd 2 12-02-2020 16:52:56

முகப்புரை

கணினி மற்றும் கணினி தொழில்நுட்பத்தின் மிகப்பெரிய தாக்கம் நவீன சமுதாயத்தை / மனிதவளர்ச்சியோடு வடிவமைப்பதில் உள்ளது. மனித நாகரிமானது, கணினியின் வளர்ச்சியுடன் சேர்ந்த இந்த **"இணையயுகத்தில்"** மிக உயர்ந்த உச்சத்தை அடைந்துள்ளது.

எளிமையான தொடரியல் மற்றும் சக்தி வாய்ந்த நினைவக மேலாண்மை கொண்ட உயர்நிலை மொழியான பைத்தான் புதிய நிரலர்களால் எளிதாக கற்க உதவுகிறது. அனைத்து பயன்பாடுகளில் செயல்பட பைத்தானை பயில்வது சிறந்த தேர்வாகும்.

நீங்கள் XI வகுப்பில் C++ மொழியை நன்கு கற்றுள்ளதால் தற்போது பைத்தான் மொழியை பயில்வது கடினமாக இருக்காது. இந்த புத்தகத்திலுள்ள எடுத்துக்காட்டுகள் மிக எளிமையாக வடிவமைக்கப்பட்டுள்ளதால் மாணவர்கள் பைத்தான் மொழியை கற்ற முன் அனுபவமோ அல்லது தேர்ந்த அறிவோ தேவையில்லை.

இந்த பாடப்புத்தகத்தை எவ்வாறு பயன்படுத்துவது?

- இந்த பாடப்புத்தகத்தை படிக்க கணினி அறிவியல் பற்றிய முன்னறிவு தேவையில்லை.
- ஒவ்வொரு அலகும், ஆசிரியர் மற்றும் மாணவர்கள் செய்து பழகக்கூடிய எளிய செயல்முறைகள் மற்றும் செயல்பாடுகளைக் கொண்டுள்ளது.
- தொழில்நுட்ப கலைச்சொற்களை எளிதில் புரிந்து கொள்வதற்காக, "சொற்களஞ்சியம்" இணைக்கப்பட்டுள்ளது.
- "உங்களுக்குத் தெரியுமா?" பெட்டி செய்தி, கற்பவருக்கு பாடத் தொடர்பான கூடுதல் தகவல்களை தருகிறது.
- மென்பொருள் பயன்பாடுகளைப் பயன்படுத்தி, செய்முறை பயிற்சிகளைத் தீர்ப்பதற்கான "பயிற்சி பட்டறை" அறிமுகம் செய்யப்பட்டுள்ளது.
- QR குறியீடு, பாடத் தொடர்பான கூடுதல் தகவல்களை மின்கற்றல் மூலம் பெறும் வகையில் இணைக்கப்பட்டுள்ளது.

QR குறியீட்டை எவ்வாறு இணைப்பது?

QR GUIDE

o உங்கள் ஸ்மார்ட் கைபேசியில், QR ஸ்கேனர் பயன்பாட்டை கூகுள் பிளே ஸ்டோரிலிருந்தோ அல்லது ஆப்பிள் ஆப் ஸ்டோரிலிருந்தோ பதிவிறக்கம் செய்து, நிறுவிக்கொள்ளவும்.

o QR ஸ்கேனர் பயன்பாட்டை திறந்து கொள்ளவும்.

o ஸ்மார்ட் கைபேசியை பாடப்புத்தகத்திலுள்ள QR குறியீட்டுக்கு அருகில் கொண்டு செல்லவும்.

o ஸ்மார்ட் கைபேசி, QR குறியீட்டை படித்தவுடன், குறியீட்டில் உள்ள இணைய முகவரி உங்கள் ஸ்மார்ட் கைபேசியில் இணைக்கப்பட்டு, பாடத் தொடர்பான கூடுதலான தகவல்களை அறிந்து கொள்ளலாம்.

Index.indd 3 10-02-2020 19:49:11

CAREER GUIDANCE AFTER 12TH

COMPUTER SCIENCE

Computer Knowledge Hub

- Centre for Development of Advanced Computing https://www.cdac.in/
- Indian Computing Olympiad https://www.iarcs.org.in/inoi
- International Olympiad of Informatics https://ioinformatics.org
- Microsoft certification exams https://www.microsoft.com/en-in/learning/exam-list.aspx
- National Cyber Olympiad http://www.sofworld.org/nco
- National Institute of Electronics & Information Technology http://www.nielit.gov.in/
- National Programme on Technology Enhanced Learning https://nptel.ac.in/

After Completing +2

Arts & Science BSc Courses | Animation & Multimedia
Computer Science | Geography
Journalism / Mass Media Communication | Library Science
Maths / Physics / Chemistry / Statistics | Psychology/ Sociology
Social Work | Visual Communication | B.C.A.

Technical Diploma

Diploma in Engineering

Professional Degree & Entrance Exams

- Hotel & Catering Institute http://www.ihmchennai.org/ihmchennai/
- AIEEE- All India Engineering Entrance Exam
- B.E/B.Tech/ B.Arch (JEE, AIEEE in IITs and NITs)
- Fashion Technology & Design http://nift.ac.in
- GATE-Graduate Aptitude Test in Engineering www gate.iitm.ac.in
- Indian Navy – 10+2 BTech Entry Scheme
- Institute of Chartered Accountants of India https://icai.org
- Institute of Company Secretary https://www.icsi.edu
- Institute of Cost Accountants of India http://icmai.in
- Institute of Banking Personal Selection IBPS https://ibps.in
- JEE-Joint Entrance Examination https://jeemain.nic.in/
- JEST- Joint Entrance Screening Test https://www.jest.org.in
- Law 5-Year Integrated Course http://tndalu.ac.in
- National Defence Academy https://www.nda.nic.in
- NET- National Eligibility Test (CSIR and UGC) http://cbsenet.nic.in
- TamilNadu Dr. Ambedkar Law University http://tndalu.ac.in
- Technical Entry Scheme – Army/Navy/Airforce
- TIFR GS - Tata Institute of Fundamental Research Graduate School http://univ.tifr.res.in
- Visual Arts Degree http://www.tndipr.gov.in/television-traininginstitute.aspx

PG in Computer Science

Artificial Intelligence
Information Management and Data Analytics
Algorithms
Applied Computer Science
Big Data Analytics
Bio Informatics
Cloud Computing
Compiler Design & Construction
Computer and Network Security
Computer Networks
Computer Security
Computer Simulation
Cyber Security
Data Analytics
Data Mining
Database Management Systems
Fundamentals of Algorithms
Graphics
Machine Learning
Mobile and Web Computing
Mobile Device Programming
Modern Programming Practices
Modern Web Applications
Operating Systems
Parallel Programming
Robotics
Software Engineering
Software Engineering
Software Testing
Systems Analysis and Design
Web Application Architecture
Web Application Programming

Scholarships for graduate and post graduate courses

- DST – INSPIRE Fellowships (for Ph.D)
- DST – INSPIRE Scholarships (for UG and PG)
- In addition various fellowships for SC/ST/PWD,
- Indira Gandhi Fellowship for Single girl child (for UG and PG)
- Moulana Azad Fellowship for minorities (for Ph.D)
- UGC National Fellowship (for Ph.D)
- International Olympiad: for getting stipend for Higher Education in Science and Mathematics
- OBC etc are available.
- Visit website of University Grants Commission

Index.indd 4 10-02-2020 19:49:11

CAREER GUIDANCE

After PG courses

MPhil –Computer Science | PhD – Computer Science

Following are the latest topics for PhD in computer science:

BioInformatics and Computational Biology
Capturing and Visualizing Persona Through Faces
Comprehensive analysis of RNA sequencing experiments
Designing and Evaluating Information Gathering Robots
Digital Pathology: Diagnostic Errors, Viewing Behavior and Image Characteristics
Embedded Systems | Game Theory | Graph Theory
Graphics and Visualization | Human Computer Interaction – HCI
Improving Fault Tolerance and Performance of Data Center Networks
Increasing Access to Computer Science for Blind Students
In-situ Semantic 3D Modeling
Intelligent Crowdsourcing for Natural Language Learning and Other AI Applications
Learning Robust Tractable Models for Vision
Manipulators And Manipulation In High Dimensional Spaces
New Algorithmic Tools for Distributed Similarity Search
Reproducible measurements of web security and privacy
The Security and Privacy of Web and Mobile Advertising
Towards More Practical Reinforcement Learning, with Applications to Educational Games

Topics for Research after PG in Computer Studies

Architectures, Compiler Optimization, and Embedded Systems.
Bioinformatics and Computational Biology | Cloud Computing
Data Mining, Databases, and Geographical Information Systems.
Graphics and Visualization | High Performance Computing.
Human Computer Interaction | Internet of Things
Natural Language Processing | Networks, Distributed Systems, and Security.

Competitive Exams for Govt. Jobs

Airforce Common Admission Test – AFCAT | Army Education Officer Entry- AEC
Combined Defence Services –CDS | Defence Service Staff College, Nilgiris
Indian Defence Services | Indian Military Academy
Judge Advocate General Department-JAG | NCC Entry | Railway Board Examination
SSC NAVY (Pilot/Observer) | SSC Tech Entry – Officers Training School
Staff Selection Examination | Tamil Nadu Public Service Commission
Teacher Recruitment Board | Technical Graduate Course –TGC
Territorial Army | Union Public Service Examination | Women Special Entry Scheme

COMPUTER SCIENCE

Computer Related Jobs

Applications Software Developer | Big Data Analysts
Cloud Computing Programmer | College/ University Faculty
Computer Programmer | Computer Teacher
Computer Vocational Instructor | Computer Information Research Scientist
Computer Information Systems Manager | Computer Network Architect
Computer Support Specialist | Computer System Analysts
Data Mining Specialist | Database Administrator
Information Security Analysts | Market Research Analysts
Network & Computer System Administrator | Research Assistant
Systems Software Developer | User Interface Designer | Web Developer

Research Institutions in various areas of science

- Bhaba Atomic Research centre (BARC) Mumbai www.barc.gov.in
- BITS Pilani, https://www.bits-pilani.ac.in
- Central Universities www.ugc.ac.in
- Chennai Mathematical Institute (CMI) Chennai www.cmi.ac.in
- Delhi University, Delhi www.du.ac.in
- Hyderabad central university, Hyderabad www.uohyd.ac.in
- IISER Educational Institutions www.iiseradmission.in
- Indian Institute of Technology in various places (IIT's) www.iitm.ac.in
- Institute of Mathematical Sciences (IMSc) Chennai www.imsc.res.in
- Jawaharlal Nehru University (JNU) www.jnu.ac.in
- Mumbai University, Mumbai www.mu.ac.in
- National Institute of Science Education and Research (NISER), www.niser.ac.in
- National Institute of Technology (NITs) www.nitt.edu
- SavithiribaiPhule Pune university, Pune www.unipune.ac.in
- State Universities https://www.ugc.ac.in
- Tata Institute of Fundamental Research (TIFR) Mumbai www.tifr.res.in

Index.indd 5 10-02-2020 19:49:11

பொருளடக்கம்

கணினி அறிவியல்–இரண்டாம் ஆண்டு

அலகு எண்	இயல் எண்	பாடத்தலைப்புகள்	பக்க எண்	மாதம்
அலகு–I சிக்கலைத் தீர்க்கும் யுக்திகள்	1	செயற்கூறு	1	ஜூன்
	2	தரவு அருவமாக்கம்	12	ஜூன்
	3	வரையெல்லை	22	ஜூன்
	4	நெறிமுறையின் யுக்திகள்	32	ஜூன்
அலகு–II பைத்தான் மையக் கருத்துருக்கள்	5	பைத்தான் அறிமுகம்–மாறிகள் மற்றும் செயற்குறிகள்	49	ஜூலை
	6	கட்டுப்பாட்டுக் கட்டமைப்புகள்	71	ஜூலை
	7	பைத்தான் செயற்கூறுகள்	93	ஜூலை
	8	சரங்கள் மற்றும் சரங்களைக் கையாளுதல்	118	ஆகஸ்ட்
அலகு–III கூறுநிலை மற்றும் பொருள்நோக்கு நிரலாக்கம்	9	List, Tuples, Set மற்றும் Dictionary தொகுப்பு தரவினங்கள்	136	ஆகஸ்ட்
	10	பைத்தான் இனக்குழுக்கள் மற்றும் பொருள்கள்	174	ஆகஸ்ட்
அலகு–IV தரவுத்தள கருத்துருக்கள் மற்றும் MySQL	11	தரவுதள கருத்துருக்கள்	192	அக்டோபர்
	12	வினவல் அமைப்பு மொழி	208	அக்டோபர்
	13	பைத்தான் மற்றும் CSV கோப்புகள்	236	அக்டோபர்
அலகு–V பைத்தானை MySQL மற்றும் C++ உடன் ஒருங்கிணைத்தல்	14	பைத்தானில் C++ நிரல்களை இறக்கம் செய்தல்	271	அக்டோபர்
	15	SQL மூலம் தரவுகளைக் கையாளுதல்	295	நவம்பர்
	16	தரவுக் காட்சிப்படுத்துதல்: PYPLOT பயன்படுத்தி – கோட்டு வரைபடம், வட்ட வரைபடம் மற்றும் பட்டை வரைபடம் உருவாக்குதல்	323	நவம்பர்
		சொற்களஞ்சியம்	338	
இணைப்பு–1		பைத்தான் செயற்கூறுகளின் பட்டியல்	342	
இணைப்பு–2		MySQL ஐ நிறுவிக் கொண்டு, ஒரு தரவுத்தளத்தை உருவாக்குதல்	346	
இணைப்பு–3		விண்டோஸில் MinGW–w64ஐ நிறுவுவதற்கான படிநிலைகள்	347	
இணைப்பு–4		Pipஐ பதிவிறக்கம் செய்து நிறுவுதல்	352	

3CGXJI
மின்னூல்

NR1JQZ
மதிப்பீடு

B237_12_COMP_SCI_TM
இணைய வளங்கள்

Index.indd 6 10-02-2020 19:49:11

அலகு I

பாடம் 1

செயற்கூறு

YXSVC8

கற்றலின் நோக்கங்கள்

இந்தப் பாடப்பகுதியைக் கற்றபின் மாணவர்கள் அறிந்துக் கொள்வது,

- செயற்கூறு வரையறை
- அளபுருக்கள் மற்றும் செயலுருபுக்கள்
- இடைமுகம் மற்றும் செயல்படுத்துதல்
- pure செயற்கூறுகள்
- பக்க விளைவுகள் ஆகியவற்றைப் பற்றி அறிந்து கொள்ளுதல்.

1.1 அறிமுகம்

ஒரு செயலைச் செய்து முடிப்பதற்கான நேரமே, நிரல் நெறிமுறையை எழுதுவதற்கும் மற்றும் மதிப்பீடு செய்வதற்கும் மிக முக்கியமான அடிப்படை ஆகும். நிரல் நெறிமுறையின் அர்த்தமுள்ள ஒப்பீட்டைப் பெறுவதற்கு (கணக்கீட்டு நேரத்தின்கால அளவானது) நிரலாக்க மொழி, நிரல் பெயர்ப்பி மற்றும் பயன்படுத்தும் கணினி ஆகியவற்றைச் சார்ந்திருக்கக்கூடாது. நிரலாக்க மொழியின் கூற்றுகளைப் பயன்படுத்தி நிரல் நெறிமுறைகள் வெளிப்படுத்தப்படுகின்றன என்பதை நீங்கள் அறிவீர்கள். மொத்தமான கூற்றுகள், பலமுறை மீண்டும் மீண்டும் செய்யப்பட வேண்டும் எனில், அந்த செயலைச் செய்து முடிப்பதற்காக துணை நிரல்கள் (Subroutines) பயன்படுகின்றன.

துணை நிரல்கள் கணினி மொழிகளின் அடிப்படைக் கட்டுமான தொகுதியாக விளங்குகின்றன. துணை நிரல்கள் என்பன ஒரு குறிப்பிட்ட செயலை மீண்டும் மீண்டும் செய்யப் பயன்படும் சிறிய நிரல் தொகுதியாகும். நிரலாக்க மொழிகளில் இத்துணை நிரல்கள் செயற்கூறுகள் (Functions) என்று அழைக்கப்படுகின்றன.

1.2 நிரலாக்க மொழியில் செயற்கூறுகள்

செயற்கூறு என்பது குறிமுறையின் ஒரு அலகு ஆகும். இது பெரும்பாலும் ஒரு பெரிய குறிமுறை கட்டமைப்பில் வரையறுக்கப்படும். குறிப்பாக, குறிமுறையின் தொகுப்பைக் கொண்டிருக்கும். செயற்கூறானது பல வகை உள்ளீடுகளான மாறிகள் மற்றும் கோவைகளின் மீது செயல்பட்டு நிலையான வெளியீட்டைத் தருகிறது.

1.2.1 செயற்கூறு வரையறை

a:=(24) என்ற உதாரணத்தைக் கவனிக்கவும். a:=(24) என்பது கோவையைக் கொண்டுள்ளது ஆனால் (24) என்பது கோவையல்ல. மாறாக, இது ஒரு செயற்கூறு வரையறை ஆகும். வரையறைகள், மதிப்புகளைப் பெயருடன் பிணைக்கின்றன. இங்கு, 24 என்ற மதிப்பு 'a' என்ற பெயருடன் பிணைக்கின்றது. வரையறைகள் கோவைகள் அல்ல, அதே நேரத்தில் கோவைகளை வரையறை எனக் கருதக் கூடாது. வரையறைகள் தனித்தன்மையான தொடரியல் தொகுதிகளைக் கொண்டதாகும். வரையறைகள் உள்ளமைவாக உள்ள கோவைகளைக் கொண்டதாகவோ அல்லது நேர்மாறாகவும் இருக்கலாம்.

1.2.2 அளபுருக்கள் மற்றும் செயலுருபுக்கள்

அளபுருக்கள் என்பது செயற்கூறு வரையறையில் உள்ள மாறிகள் ஆகும்.

செயலுருபுக்கள் என்பது செயற்கூறு வரையறைக்கு அனுப்பப்படும் மதிப்புகள் ஆகும்.

1.2.2.1 தரவு வகை இல்லா அளபுருக்கள்

செயற்கூறு வரையறைக்கு ஓர் எடுத்துக்காட்டைக் காணலாம்:

```
(requires: b>=0 )
(returns: a to the power of b)
let rec pow a b:=
      if b=0 then 1
      else a * pow b (a-1)
```

மேலேயுள்ள செயற்கூறு வரையறையில் 'b' என்ற மாறி அளபுரு ஆகும். மாறி 'b'க்கு அனுப்பப்படும் மதிப்பானது செயலுருபு ஆகும். செயற்கூறின் முன் நிபந்தனை (requires) மற்றும் பின் நிபந்தனை (return) கொடுக்கப்பட்டுள்ளது. நாம் எந்த தரவினத்தையும் குறிப்பிடவில்லை என்பதைக் நினைவில் கொள்க. சில மொழிகளின் நிரல் பெயர்ப்பி இவ்வகை சிக்கல்களை நிரல் நெறிமுறைப்படி சரி செய்கிறது. ஆனால் சில நிரல் பெயர்ப்பிக்கு தரவு வகையைக் குறிப்பிடுவது கட்டாயமாகும்.

மேலே உள்ள செயற்கூறு வரையறையில், if கோவை, then கிளைக்கு மதிப்பு 1 யைத் திருப்பி அனுப்பினால், தரவு வகை (data type) விதிப்படி if கோவை முழுவதுமே 'int' தரவு வகைக் கொண்டிருக்கும். if கோவையின் தரவு வகை int ஆக இருப்பதால் செயற்கூறின் திருப்பி அனுப்பும் மதிப்பும் int ஆக இருக்கும். 'b' யின் மதிப்பு சுழியத்தோடு = செயற்குறியுடன் ஒப்பீடு செய்யப்படுகிறது. அதனால் 'b' யின் தரவுவகையும 'int' ஆகும். * செயற்குறியுடன் 'a' யின் மதிப்பு மற்றொரு கோவையோடு பெருக்குத்தொகையைக் கணக்கிடுவதால் ,'a'-யின் வகையும் int ஆகும்.

1.2.2.2 தரவு வகையுடன் கூடிய அளபுரு

சில காரணங்களுக்காக, இப்பொழுது நாம் அதே செயற்கூறு வரையறை தரவு வகையுடன் எழுதலாம் :

```
(requires: b>  0 )
(returns: a to the power of b )
let rec pow (a: int) (b: int) : int :=
      if b=0 then 1
      else a * pow b (a-1)
```

'a' மற்றும் 'b' தரவு வகை குறிப்பு (type annotations) எழுதும் போது, அடைப்புக்குறிக்குள் ( ) அவசியமானது ஆகும். பொதுவாக, இந்த குறிப்புகளை நாம் விட்டுவிடலாம், ஏனெனில், நிரல்பெயர்ப்பி இவற்றை அனுமானிப்பது மிகவும் எளிது. முன்பெல்லாம் நாம் வெளிப்படையாகவே தரவு வகைகளை எழுதுவோம். எந்த வித அர்த்தமும் இல்லாத தரவு வகை பிழைச் செய்தியைப் பெறும் போது, இது மிகவும் பயனுள்ளதாகும். தரவு வகைக்கு வெளிப்படையாக தரவுவகை குறிப்பு எழுதுவது பிழைச் செய்தியைத் திருத்தம் செய்வதற்கு பயனுள்ளதாக இருக்கும். செயற்கூறை வரையறுப்பதாற்கான தொடரியல், கணித பயன்பாட்டோடு நெருக்கம் கொண்டது. வரையறை let என்ற சிறப்புச் சொல்லோடு அறமுகப்படுத்தப்படுகிறது. அதனைத் தொடர்ந்து செயற்கூறின் பெயர் மற்றும் அதனுடைய செயலுருபுகள் பிறகு செயலுருபுவைக் கணக்கீடு செய்யும் வாய்ப்பாடு= குறிக்குப் பிறகு எழுத வேண்டும். தற்சுழற்சி செயற்கூற்றை வரையறுக்க விரும்பினால் let-க்குப் பதிலாக let rec என்று பயன்படுத்த வேண்டும்.

செயற்கூறு வரையறையின் தொடரியல்:

```
let rec fn  a1 a2 ... an := k
```

இங்கு 'fn' என்பது ஒரு மாறி ஆகும். இது செயற்கூறின் பெயரைக் குறிக்கும். குறிப்பெயராகும். 'a1' முதல் 'an' வரை உள்ள மாறிகள் அளபுருக்களைக் குறிக்கும் குறிப்பெயராகும். 'fn' என்பது தற்சுழற்சி செயற்கூறினால் rec என்ற சிறப்புச் சொல் தேவை, இல்லையெனில் அதை விட்டு விடலாம்.

Chapter 1.indd 2 10-02-2020 19:54:30

குறிப்பு

தன்னைத் தானே அழைத்துக் கொள்ளும் செயற்கூறு வரையறைக்கு தற்சுழற்சி செயற்கூறு என்று பெயர்.

எடுத்துக்காட்டாக, உள்ளிடப்பட்ட எண் ஒற்றைப்படை எண்ணா அல்லது இரட்டைப்படை எண்ணா என சோதிக்கும் எடுத்துக்காட்டைக் காண்போம்.

```
(requires: x>= 0)
        let rec even x :=
        x=0 || odd (x-1)
        return 'even'
(requires: x>= 0)
        let odd x :=
        x<>0 && even (x-1)
        return 'odd'
```

செயற்கூறு வகைகளின் தொடரியல்

$x \rightarrow y$

$x1 \rightarrow x2 \rightarrow y$

$x1 \rightarrow \ldots \rightarrow xn \rightarrow y$

'x' மற்றும் 'y' மாறிகள் செயற்கூறு வகைகளைக் குறிக்கும். x → y என்ற செயற்கூறு வகையானவை 'x' வகையின் உள்ளீட்டைப் பெற்று 'y' வகையின் வெளியீட்டைத் திருப்பி அனுப்பும். அதே சமயம் x1 → x2 → y என்ற செயற்கூறு வகையானது இரண்டு உள்ளீட்டுகளைப் பெற்று, முதல் உள்ளீட்டின் வகை 'x1' ஆகும், இரண்டாவது உள்ளீட்டின் வகை 'x2' ஆகும். திருப்பி அனுப்பும் வெளியீட்டின் வகை 'y' ஆகும். x1 → … →xn → y யில் n அளபுருக்களின் உள்ளீட்டு வகை x ஆகும். வெளியீட்டின் வகை 'y' ஆகும்.

குறிப்பு

அனைத்து செயற்கூறுகளும் static வரையறையாகும். dynamic செயற்கூறு வரையறைகளே கிடையாது.

1.3 இடைமுகம் VS செயல்படுத்துதல்

ஒரு பொருள் (Object) செய்யக்கூடிய செயல்களின் தொகுப்பு இடைமுகம் ஆகும். எடுத்துக்காட்டாக, மின் விளக்கின் சுவிட்ச்சை அழுத்தும் போது மின் விளக்கு ஒளிர்கிறது. அது எவ்வாறு ஒளிர்கிறது என்பது கவலையில்லை. பொருள் நோக்கி நிரலாக்க மொழியில், இடைமுகம் என்பது அனைத்து செயற்கூறுகளின் விளக்கங்கள் (descriptions) ஆகும். ஒரு புதிய இடைமுகமாக இருப்பதற்கு இனக்குழு கண்டிப்பாக இவற்றைக் கொண்டிருக்க வேண்டும். நமது எடுத்துக்காட்டில், மின் விளக்கை போல் செயல்படும் எதுவும் turn_on() மற்றும் turn_off() என்ற செயற்கூறு வரையறையைக் கொண்டிருக்கும். X,Y,Z செயற்கூறுகளைக் கட்டாயமாக கொண்டிருக்கும் இனக்குழுவாகிய TYPE T (இடைமுகம் எதுவாக இருந்தாலும்) யின் பண்புகளை வலுக்கட்டாயமாக செயல்படுத்துவதே இடைமுகத்தின் நோக்கம் ஆகும்.

இனக்குழு அறிவிப்பானது வெளிப்புற இடைமுகத்தோடு அதன் உள்ளமை நிலை அந்த இடைமுகத்தைச் செயல்படுத்தும் செயல்பாட்டுடன் பண்புகளை உடைய குறிமுறை இணைக்கிறது.

பொருள் என்பது இனக்குழுவால் உருவாக்கப்பட்ட சான்றுரு ஆகும்.

வெளி உலகிற்கு பொருளின் காண்புநிலைப் பாங்கை இடைமுகம் வரையறுக்கிறது.

இடைமுகம் மற்றும் செயல்படுத்துதல் இரண்டுக்கும் இடையே உள்ள வேறுபாடு என்னவெனில்,

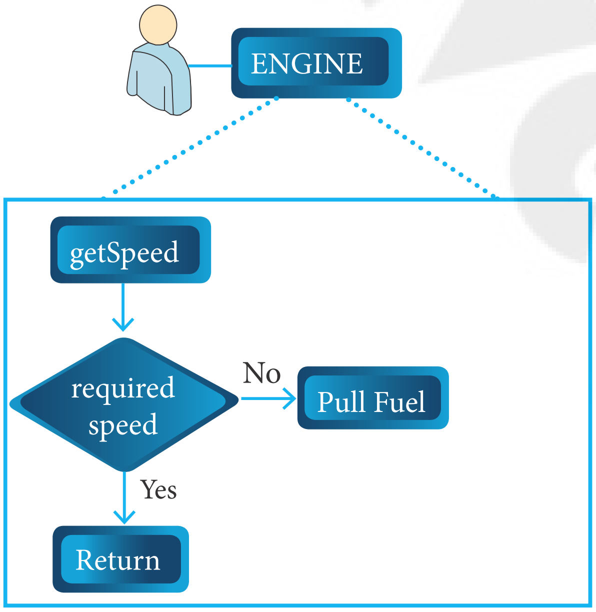
இடைமுகம்	செயல்படுத்துதல்
ஒரு பொருள் செய்யக்கூடிய நடவடிக்கையை வரையறுக்கிறது, ஆனால் அவற்றை உண்மையில் செய்யக் கூடியது இல்லை.	இடைமுகத்தில் வரையறுக்கப்பட்டுள்ள கட்டளைகளை நிறைவேற்றுகிறது.

பொருள் நோக்கு நிரலாக்க மொழியில் இனக்குழு என்பது இடைமுகம் மற்றும் பொருள் எவ்வாறு செயல்படுத்தப்பட்டு நிறை வேற்றப்படுகிறது என்பதை செயல்படுத்துதல் ஆகும்.

1.3.1 இடைமுகத்தின் பண்புகள்

- ஒரு பொருளை முறையாக உருவாக்கி வழங்குவதற்கும் அதனை செயல்படுத்துவதற்கும் தேவையான இடைமுகத்தை இனக்குழு வார்ப்புரு குறிப்பிடுகிறது.
- செயற்கூறுகளைப் பொருளுக்கு அனுப்புவதன் மூலம் பொருளின் பண்புகளையும் பண்புக்கூறுகளையும் கட்டுப்படுத்த முடிகிறது.

காரின் வேகத்தை அதிகரிக்கும் எடுத்துக்காட்டை எடுத்துக் கொள்வோம்.



காரை ஓட்டும் நபர் அந்த காரின் உட்புற செயல்பாடுகள் பற்றி அறிந்திருக்க வேண்டிய அவசியமில்லை. காரின் வேகத்தை அதிகப்படுத்த, அவர் காரின் துரிதப்படுத்தியை (accelerator) அழுத்தி விரும்பிய பண்பை பெறுவார். இங்கு துரிதப்படுத்தி என்பது கார் ஓட்டுநருக்கும் (அழைக்கும் பொருள்) இயந்திரத்துக்கும் (அழைக்கப்படும் பொருள்) இடையேயான இடைமுகம் ஆகும். இதில், அழைக்கும் செயற்கூறு இவ்வாறு இருக்க வேண்டும் speed(70): இது ஒரு இடைமுகமாகும்.

உட்புறமாக, காரின் இயந்திரம் அனைத்து வேலைகளையும் செய்கிறது. எரிபொருள், காற்று, அழுத்தம் மற்றும் மின்சாரம் ஆகியவை இங்கு ஒன்றாக சேர்ந்து, ஆற்றலை உருவாக்கி, வாகனத்தை நகர்த்துகிறது. இந்த அனைத்து நடவடிக்கைகளும் கார் ஓட்டுநரிடம் இருந்து பிரித்து வைக்கப்பட்டுள்ளது. அவர் வேகமாக காரை செலுத்துவதில் ஆர்வமாக இருப்பார். இதனால், இடைமுகத்தை செயல்படுத்துதலில் இருந்து பிரித்து வைக்கப்படுகிறது.

ஒரு எளிமையான உதாரணத்தைக் காணலாம். கொடுக்கப்பட்டுள்ள 3 செயலுருபுகளில் குறைந்த மதிப்பைக் காணும் செயற்கூற்றைச் செயல்படுத்துவதைக் கவனி.

```
let min 3 x y z :=
        if x < y then
                if x < z then x else z
        else
                if y < z then y else z
```

1.4 Pure செயற்கூறுகள்

ஒரே மாதிரியான அளபுருக்களை அனுப்பும் போது, சரியான விடையைத் தரும் செயற்கூறு *pure* செயற்கூறுகள் ஆகும். எடுத்துக்காட்டாக, கணித செயற்கூறு *sin(0)*-ன் விடை எப்பொழுதுமே *0* ஆகும். இதன் அர்த்தம் என்னவென்றால், அதே அளபுருக்களைக் கொண்டு செயற்கூறினை ஒவ்வொரு முறையும் அழைக்கும் போது, அதே சரியான விடையை எப்பொழுதும் பெறலாம். மாறியின் பண்பை மாற்றக் கூடிய எந்த விதமான வெளிப்புற மாறியும் இல்லாமல் இருந்தால் அந்த செயற்கூறு *pure* செயற்கூறாகும்.

ஒரு எடுத்துக்காட்டை காண்க

```
let square x
        return: x * x
```

மேலேயுள்ள square செயற்கூறு pure செயற்கூறு ஆகும். ஏனென்றால் ஒரே மாதிரியான உள்ளீட்டிற்கு வேறுவித்தியாசமான வெளியீட்டைத் தராது.

Chapter 1.indd 4 10-02-2020 19:54:30

கோட்பாடு சார்ந்த நன்மைகளை pure செயற்கூறுகள் கொண்டுள்ளன. இதன் ஒரு நன்மை என்னவென்றால், pure செயற்கூறாக இருக்கும் பொழுது, அதே அளபுருக்களுடன் செயற்கூறுவை பல தடவைகள் அழைக்கும்போது, உண்மையில் பெயர்ப்பிக்கு செயற்கூறுவை மீண்டும் ஒரு தடவை அழைக்கும் தேவை மட்டுமே ஏற்படுகிறது.

```
let i: = 0;
    if i <strlen (s) then
  -- Do something which doesn't affect s
    ++i
```

இது இயக்கப்படும் போது, ஒவ்வொரு முறையும் strlen (s) அழைக்கப்படுகிறது. strlen க்கு ஒட்டு மொத்தமாக ‘s’ தற்சுழற்ச்சி செய்ய தேவைப்படுகிறது. strlen என்பது pure செயற்கூறு என்றும், ‘s’யை மடக்கினுள் புதுப்பிக்காமலும் இருந்தால், strlen க்கு தேவைக்கு அதிகமான அழைப்பை நீக்கிவிட்டு, மடக்கை ஒரே ஒரு முறை செயல்படுததும். இதிலிருந்து நாம் தெரிந்து கொள்வது யாதெனில், strlen என்பது pure செயற்கூறாகும். ஏனென்றால், செயற்கூறு அளபுருவாக ஒரே ஒரு மாறியை எடுத்துக் கொண்டு அதனுடைய நீளத்தை கணக்கிடுகிறது. இந்த செயற்கூறு வெளி நினைவகத்தில் இருந்து உள்ளீட்டை எடுத்துக் கொள்கிறது. ஆனால் மதிப்புகளை மாற்றுவதில்லை திருப்பி அனுப்பும் மதிப்புகள் வெளி நினைவகத்தில் இருந்து பெறப்பட்டதாகும்.

குறிப்பு

pure செயற்கூற்றை மதிப்பீடு செய்யும் போது, எந்தவொரு பக்கவிளைவும் ஏற்படாது.

1.4.1 Impure செயற்கூறுகள்

செயற்கூறுக்கு அளபுருக்களை அனுப்பாதபோதும், செயற்கூறின் உள்ளே உள்ள மாறியானது பக்க விளைவுகளை ஏற்படுத்தும். இந்த வகையான செயற்கூறை impure செயற்கூறு என்பர். ஒரு செயற்கூறு அந்த வரையறை தொகுதியின் வெளியே உள்ள மாறிகள் அல்லது செயற்கூறுகளைச் சார்ந்து இருந்து ஒவ்வொரு முறை அழைக்கும் பொழுதும் செயற்கூறு ஒரே மாதிரியாக இயக்கப்படும் என கூற இயலாது. எடுத்துக்காட்டாக, random() என்கிற கணித செயற்கூறுஒரேமாதிரியான அழைப்புக்கூற்றுக்கு வெவ்வேறு விதமான வெளியீடுகளைக் கொடுக்கும்.

```
let Random number
let a := random()
   if a > 10 then
      return: a
   else
      return: 10
```

இங்கு, Random என்பது impure செயற்கூறு ஆகும். ஏனெனில் இதனை அழைக்கும் பொழுது என்ன விடை கிடைக்கும் என நிச்சயமாக கூற இயலாது.

1.4.2 பக்க விளைவுகள் (Impure செயற்கூறுகள்)

செயற்கூறு கவனிக்கத்தக்க வெளியுலக தொடர்பில் இருக்கும் போது பக்க விளைவுகள் ஏற்படும் என்பதை அறிவீர்கள். நம்முடைய நோக்கம் Pure செயற்கூறுவை உருவாக்குவதாக இருப்பினும் அது சில சமயங்களில் impure செயற்கூறுவாக மாறி விடுகிறது. பக்க விளைவு என்பது கேடு விளைவிக்கும் செயல் அல்ல, என்பதை நினைவில் கொள்க. சில சமயங்களில் அவை பயனுள்ளதாகும்.

செயற்கூறுவிற்கு வெளியில் மாறியை மாற்றுதல்

செயற்கூறுவின் வெளியே மாறியை மாற்றம் செய்வது என்பது பக்கவிளைவுகளில் ஒன்றாகும்.

எடுத்துக்காட்டு

```
let y: = 0
(int) inc (int) x
 y: = y + x;
return (y)
```

மேலே கூறப்பட்ட எடுத்துக்காட்டில், y-ன் மதிப்பு செயற்கூறு வரையறையின் உள்ளே மாறுவதால் விடையானது ஒவ்வொரு முறையும் மாறும். inc() செயற்கூறுவின் பக்க விளைவு என்னவென்றால் வெளிப்புற மாறியான 'y' ன் மதிப்பை மாற்றுவதாகும். சில பக்க விளைவுகளை அடையாளம் காண்பது எளிதானதாகும் மற்றும் சில யோசிக்க தகுந்ததாக இருக்கும். நமது Impure செயற்கூறு எந்த அளபுருக்களையும் எடுத்துக் கொள்ளாது, எந்த மதிப்பையும் திருப்பி அனுப்பாது என்பது ஒரு நல்ல அறிகுறியாகும்.

கொடுக்கப்பட்ட அனைத்து எடுத்துக்காட்டுகள் மற்றும் விளக்கங்களில் இருந்து நாம் pure மற்றும் impure செயற்கூறுகளின் வேறுபாட்டை பின்வருமாறு காணலாம்.

Pure செயற்கூறு	Impure செயற்கூறு
Pure செயற்கூறுவின் திருப்பி அனுப்பும் மதிப்பு முற்றிலும் அளபுருக்களை பொறுத்தே அமையும். அதனால், pure செயற்கூறினை அதே அளபுருக்களைக் கொண்டு அழைத்தால் எப்பொழுதும் அதே திருப்பி அனுப்பும் மதிப்பே கிடைக்கும்	Impure செயற்கூறுவின் திருப்பி அனுப்பும் மதிப்பு முற்றிலும் அளபுருக்களை பொறுத்து அமையாது. அதனால், impure செயற்கூறினை அதே அளபுருக்களைக் கொண்டு அழைத்தால் வெவ்வேறான திருப்பி அனுப்பும் மதிப்பு கிடைக்கும்.
இது எந்த பக்க விளைவுகளையும் கொண்டிருக்காது.	
இந்த செயற்கூறு அளபுருக்களை மாற்றம் செய்யாது.	இந்த செயற்கூறு அளபுருக்களை மாற்றம் செய்யும்.

இரண்டு நேர்ம முழு எண்களின் மீப்பெரு வகுஎண்ணை கண்டுபிடிக்கும் pure செயற்கூறினை எடுத்துக்காட்டாக காணலாம்.

```
let rec gcd a b :=
    if b <> 0 then gcd b (a mod b) else return a;
```

வெளியீடு

```
    gcd 13 27;
    - : int = 1
    gcd 20536 7826;
    - : int = 2
```

மேலே கொடுக்கப்பட்டுள்ள 'gcd' என்பது செயற்கூறுவின் பெயர் ஆகும். மாறி 'b' ன் மதிப்பு 0 ஆகும் வரை செயற்கூறு தன்னைத் தானே அழைத்துக் கொள்ளும் b மற்றும் (a mod b) ஆகிய இரண்டு அளபுருக்களை gcd செயற்கூறுவினுள் 'a' மற்றும் 'b' க்கு அனுப்புவதை நினைவில் கொள்க.

14.3 செயற்கூறுவை பயன்படுத்தி குரோமிலேண்டில் பச்சோந்திகள் என்ற சிக்கல் தீர்த்தல்

பதினொன்றாம் வகுப்பில் படித்த குரோமிலெண்டில் பச்சோந்திகள் என்ற பகுதியை நினைவில் கூர்க. இரண்டு வகையான பச்சோந்திகள் சமமான எண்ணாக இருந்தால், இந்த இரண்டு வகையும் சேர்ந்து மூன்றாவது வகையின் நிறத்தை மாற்றுவதற்கான நிரல் நெறிமுறையை உருவாக்கவும். முடிவில் அனைத்தும் ஒரே நிறத்தை காண்பிக்க வேண்டும். ஒவ்வொரு வகை பச்சோந்தியின் எண்ணிக்கையையும் a,b,c மாறிகளாக எடுத்துக் கொள்வோம். அதனுடைய தொடக்க மதிப்பு முறையே A, B மற்றும் C . a = b என்பது உள்ளீட்டு பண்பு ஆகும். இதன் உள்ளீட்டு வெளியீட்டுக்கான தொடர்பு a = b = 0 மற்றும் c = A+B+C ஆகும். இதனுடைய நிரல் நெறிமுறைக்கு monochromatize எனப் பெயரிடலாம். இதை monochromatize (a , b , c) என குறிப்பிடலாம்.

Chapter 1.indd 6 10-02-2020 19:54:31

-- உள்ளீடுகள்	:	a = A, b = B, c = C, a = b
-- வெளியீடுகள்	:	a = b = 0, c = A+B+C

ஒவ்வொரு சுழற்சி நிலையிலும் 2 வகையான ஒரே எண்ணைக் கொண்ட பச்சோந்திகள் சந்தித்து மூன்றாவது வகைக்கு அதன் வண்ணத்தை மாற்றும். எடுத்துக்காட்டாக, A, B, C = 4, 4, 6, எனில் சந்திப்பின் வரிசையானது பின்வருமாறு,

iteration	a	b	c
0	4	4	6
1	3	3	8
2	2	2	10
3	1	1	12
4	0	0	14

ஒவ்வொரு சந்திப்பிலும், a மற்றும் b-ன் மதிப்பு ஒன்று குறைகிறது மற்றும் c-ன் மதிப்பு 2அதிகரிக்கிறது. இதற்கான தீர்வு சுழற்சி நெறிமுறையில் வெளிப்படுத்துகிறது.

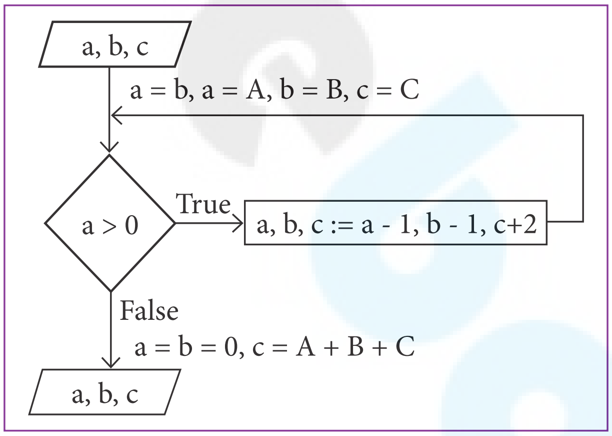
monochromatize (a, b, c)

-- உள்ளீடுகள்	:	a = A, b = B, c = C, a = b
-- வெளியீடுகள்	:	a = b = 0, c = A+B+C

while a > 0

a, b, c := a -1, b -1, c+2

இந்த நெறிமுறை பாய்வுப்படமாக கீழே காண்பிக்கப்பட்டுள்ளது.



இப்பொழுது நிரல் நெறிமுறையை செயற்கூறுவைப் பயன்படுத்தி எழுதலாம்.

```
let rec monochromatize a  b  c :=
        if a > 0 then
a, b, c := a-1, b-1, c+2
        else
          a:=0, b:=0, c:= a + b + c
        return c
```

நினைவில் கொள்க

- நெறிமுறைகள் நிரலாக்க மொழியின் கூற்றுகளைப் பயன்படுத்தி வெளிப்படுத்தப்படுகின்றன.
- ஒரு குறிப்பிட்ட செயலைச் செய்வதற்காக மீண்டும் மீண்டும் பயன்படுத்தப்படும் குறிமுறையின் சிறிய பகுதியே துணை நிரலாக்கமாகும்.
- செயற்கூறு என்பது குறிமுறையின் ஒரு அலகு ஆகும். இது பெரும்பாலும் ஒரு பெரிய குறிமுறை கட்டமைப்பில் வரையறுக்கப்படும்.
- செயற்கூறானது பல வகை உள்ளீடுகளைக் கொண்டு செயல்பட்டு நிலையான வெளியீட்டைத் தருகிறது.
- வரையறைகள் தனித்தன்மையான தொடரியல் தொகுதிகளைக் கொண்டதாகும்.
- அளபுருக்கள் என்பது செயற்கூறு வரையறையில் உள்ள மாறிகள் ஆகும். செயலுருபுக்கள் என்பது செயற்கூறு வரையறைக்கு அனுப்பப்படும் மதிப்புகள் ஆகும்.
- தரவு வகைக்கு குறிப்பு எழுதும் போது, செயற்கூறு வரையறையில் () அடைப்புக்குறிகள் அவசியமானதாகும்.
- ஒரு பொருள் செய்யக்கூடிய செயல்பாடுகளின் தொகுப்பு இடைமுகம் ஆகும். இது நடவடிக்கையை வரையறுக்கிறது. ஆனால், உண்மையில் அவற்றை செய்யக்கூடியது இல்லை.
- செயல்படுத்துதல் என்பது இடைமுகத்தில் வரையறுக்கப்பட்டுள்ள கட்டளைகளை நிறைவேற்றுகிறது.
- ஒரே மாதிரியான அதே அளபுருக்களை அனுப்பும் போது, சரியான விடையைத் தரும் செயற்கூறு pure செயற்கூறு எனப்படும்.
- செயற்கூறுக்கு அளபுருக்களை அனுப்பாத போதும், செயற்கூறின் உள்ளே உள்ள மாறியானது பக்க விளைவுகளை ஏற்படுத்தும். இந்த வகைச் செயற்கூறு impure செயற்கூறு எனப்படும்.

செய்முறைப் பயிற்சி

1. மூன்று எண்களில் சிறிய எண்ணை கண்டுபிடிப்பதற்கான செயற்கூறு வரையரையை கொண்ட நெறிமுறையை எழுதுக.
2. n வரையுள்ள எண்களின் கூட்டுத்தொகையைக் கணக்கிடும் தற்சுழற்சி செயற்கூறு வரையறையைக் கொண்ட நெறிமுறையை எழுதுக.

Chapter 1.indd 8 10-02-2020 19:54:31

DJ995R

மதிப்பீடு

பகுதி–அ

சரியான விடையை தேர்ந்தெடுத்து எழுதுக **(1 மதிப்பெண்)**

1. ஒரு குறிப்பிட்ட செயலைச் செய்வதற்காக பயன்படுத்தப்படும் குறிமுறையின் சிறிய பகுதியே

   (அ) துணை நிரல்கள் (ஆ) கோப்புகள்

   (இ) Pseudo குறிமுறை (ஈ) தொகுதிகள்

2. பின்வரும் எந்த அலகு ஒரு பெரிய குறிமுறை கட்டமைப்பில் வரையறுக்கப்பட்டுள்ளது?

   (அ) துணை நிரல்கள் (ஆ) செயற்கூறு

   (இ) கோப்புகள் (ஈ) தொகுதிகள்

3. பின்வரும் எது தனித்தன்மையான தொடரியல் தொகுதிகளைக் கொண்டதாகும்?

   (அ) துணை நிரல்கள் (ஆ) செயற்கூறு

   (இ) வரையறை (ஈ) தொகுதிகள்

4. செயற்கூறு வரையறையில் உள்ள மாறிகள் எவ்வாறு அழைக்கப்படுகிறது?

   (அ) துணை நிரல்கள் (ஆ) செயற்கூறு

   (இ) அளப்புருக்கள் (ஈ) செயலுருபு

5. செயற்கூறு வரையறைக்கு அனுப்பப்படும் மதிப்புகள் எவ்வாறு அழைக்கப்படுகிறது?

   (அ) செயலுருபுகள் (ஆ) துணை நிரல்கள்

   (இ) செயற்கூறு (ஈ) செயற்கூறு

6. தரவு வகை குறிப்பு எழுதும்போது, எது கட்டாயமாகிறது?

   (அ) { } (ஆ) ( )

   (இ) [ ] (ஈ) < >

7. பின்வரும் எது ஒரு பொருள் செய்ய வேண்டியதை தீர்மானிக்கிறது?

   (அ) இயக்க அமைப்பு (ஆ) நிரல் பெயர்ப்பி

   (இ) இடைமுகம் (ஈ) தொகுப்பான்

8. பின்வரும் எது இடைமுகத்தில் வரையறுக்கப்பட்ட கட்டளைகளை நிறைவேற்றுகிறது?

   (அ) இயக்க அமைப்பு (ஆ) நிரல் பெயர்ப்பி

   (இ) செயல்படுத்துதல் (ஈ) தொகுப்பான்

9. ஒரே மாதிரியான அதே அளபுருக்களை செயற்கூறுவிற்கு அனுப்பினால் சரியான விடையைத் தரும் செயற்கூறு எவ்வாறு அழைக்கப்படும்?

(அ) Impure செயற்கூறு (ஆ) Partial செயற்கூறு

(இ) Dynamic செயற்கூறு (ஈ) Pure செயற்கூறு

10. அளபுருக்களை அனுப்பும் போது பக்க விளைவுகளை ஏற்படுத்தும் செயற்கூறு எவ்வாறு அழைக்கப்படும்?

(அ) impure செயற்கூறு (ஆ) Partial செயற்கூறு

(இ) Dynamic செயற்கூறு (ஈ) Pure செயற்கூறு

பகுதி–ஆ

அனைத்து வினாக்களுக்கும் விடையளி **(2 மதிப்பெண்)**

1. துணைநிரல் என்றால் என்ன?
2. நிரலாக்க மொழியைப் பொறுத்து செயற்கூறுவை வரையறுக்கவும்.
3. X:=(78) இதன் மூலம் அறிவது என்ன?
4. இடைமுகத்தையும், செயல்படுத்துதலையும் வேறுபடுத்துக.
5. பின்வருவனவற்றுள் எது சாதாரண செயற்கூறு வரையறை மற்றும் எது தற்சுழற்சி செயற்கூறு வரையறை

i) let rec sum x y:

return x + y

ii) let disp :

print 'welcome'

iii) let rec sum num:

if (num!=0) then return num + sum (num-1)

else

return num

பகுதி–இ

அனைத்து வினாக்களுக்கும் விடையளி **(3 மதிப்பெண்)**

1. இடைமுகத்தின் பண்புக்கூறுகள் யாவை?
2. strlen ஏன் pure செயற்கூறு என்று அழைக்கப்படுகிறது?
3. impure செயற்கூறுவின் பக்க விளைவுகள் யாவை? எடுத்துக்காட்டுடன் விளக்குக?
4. pure மற்றும் impure செயற்கூற்றை வேறுபடுத்துக.
5. ஒரு செயற்கூறிக்கு வெளியே ஒரு மாறியை மாற்றினால் என்ன விளைவுகள் ஏற்படும்? ஒரு எடுத்துக்காட்டு தருக.

Chapter 1.indd 10 10-02-2020 19:54:31

பகுதி ஈ

அனைத்து வினாக்களுக்கும் விடையளி **(5 மதிப்பெண்)**

1. செயலுருபுகள் என்றால் என்ன?

   (அ) தரவுவகை இல்லாத அளபுருக்கள்

   (ஆ) தரவு வகையுடன் கூடிய அளபுருக்கள் விவரி?

2. பின்வரும் நிரலில்

```
let rec gcd a b :=
      if b <> 0 then gcd b (a mod b) else return a
```

   அ) செயற்கூறுவின் பெயர்

   ஆ) தற்சுழற்சி செயற்கூறு கூற்று

   இ) அளபுருக்கள் கொண்ட மாறியின் பெயர்

   ஈ) செயற்கூறுவை தற்சுழற்சிக்கு அழைக்கும் கூற்று

   உ) தற்சுழற்சியை முடிவுக்கு கொண்டுவரும் கூற்று ஆகியவற்றை எழுதுக.

3. pure மற்றும் impure செயற்கூறுவை எடுத்துக்காட்டுடன் விளக்குக.

4. இடைமுகம் மற்றும் செயல்படுத்துதலை எடுத்துக்காட்டுடன் விளக்குக.

REFERENCES

1. *Data Structures and Algorithms in Python By Michael T.Goodrich, Roberto Tamassia and Michael H. Goldwasser.*
2. *Data Structure and Algorithmic Thinking in Python By Narasimha Karumanchi*
3. *https://www.python.org*

அலகு I

பாடம் 2

தரவு அருவமாக்கம்

S5PLYA

கற்றலின் நோக்கங்கள்

இந்தப் பாடத்தைக் கற்றபின், மாணவர்கள் கற்றுக்கொள்வது,

- அருவமாக்க தரவு கட்டமைப்பை புரிந்து கொள்ளுதல்.
- அருவமாக்க தரவு வகை
- கான்கீரிட் மற்றும் அருவமாக்கம் தரவு வகை செயல்படுத்தலில் உள்ள வேற்றுமைகள்.
- இணை (Pairs)-ன் பயன்பாடு
- கட்டுருவில் தரவு அருவமாக்கம்.

2.1 தரவு அருவமாக்கம் அறிமுகம்

தரவு அருவமாக்கம் கணினி அறிவியலில் சக்தி வாய்ந்த கருத்துருவாகும். இது நிரலர்களை, நிரல் குறியீட்டை ஒரு பொருளாக கருத வழிச் செய்கிறது. உதாரணமாக வாகனம், பென்சில் மக்கள் போன்றவற்றை போல் நிரலையும் ஒரு பொருளாகக் கருதலாம். நிரலர்கள் நிரல் எவ்வாறு செயல்படுத்தப்படுகின்றது என்பதைப் பற்றி கவலைப்பட வேண்டிய அவசியம் இல்லை, இது அவர்கள் அப்பொருள் என்ன செய்கின்றது என்று அறிந்திருந்தால் போதும்.

குறிப்பாக பல நிரலர்கள் ஈடுபடும் செயல்திட்டத்தை (Project) செயல்படுத்தும் போது இது முக்கியமானதாகும். இங்கு செயல் திட்டம் என்பது நிரலாக்கத்தை குறிக்கிறது. தரவு அருவமாக்கத்தில் குழு உறுப்பினர்கள், குறிமுறையை புரிந்து கொள்ள ஒவ்வொரு வரியாக வாசிக்க தேவையில்லை, அவர்கள் அது ஒரு பணியினை செய்கிறது என்று கருதி கொண்டால் போதும்.

அருவமாக்கம் நிரலுக்கு கூறுநிலைமையை வழங்குகிறது. (ஒரு நிரல் பல தொகுதிகளாக பிரிக்கப்படுவது கூறு நிலையாகும்) இனக்குழுக்கள் போன்றவை அருவமாக்க தரவு வகையின் ("Abstract Data Types,"(ADT)) உருவமைப்பாகும்.

2.2 தரவு அருவமாக்கின் வகைகள்

அருவமாக்க தரவு வகை (Abstract Data type (ADT)) என்பது பொருள்களுக்கான வகை (அல்லது இனக்குழு) ஆகும், இதன் செயல் மதிப்பின் தொகுப்பு மற்றும் செயல்பாடுகளின் தொகுப்பால் வரையறுக்கப்படுகிறது.

ADT யின் வரையறுப்பு என்ன வகையான செயல்பாடுகள் செய்யப்பட வேண்டும் என்று குறிப்பிடுகின்றது, அவை எப்படி செயல்பட வேண்டும் என்று குறிப்பிடப்படுவதில்லை. இவை தரவுகள் நினைவகத்தில் எவ்வாறு சேமிக்கப்படுகின்றது என்றோ அல்லது எந்த நெறிமுறைகள் செய்யப்பட வேண்டும் என்ற வழிமுறையோ குறிப்பிடப்படுவதில்லை. சுயமாக இவை செயல்படுத்தப்படுவதில்லை எனவே, இதை அருவமாக்கம் என்று அழைக்கிறோம். விவரங்களை மறைத்து அவசியமானவற்றை மட்டும் வழங்கும் செயல்முறையை அருவமாக்கம் என்கிறோம்.

இவ்வகை வரையறைகள், ADTs எவ்வாறு உருவமைக்கப்படுகின்றன என்றோ அல்லது எவ்வாறு செயல்பாடுகள் கையாளப்படுகிறதோ என்று குறிப்பிடப்படுவதில்லை என்பதை நீங்கள் காணலாம். ADT யை பல்வேறு வகையில் செயல்படுத்தலாம், உதாரணமாக List ADT ஒற்றை இணைக்கப்பட்ட பட்டியல் அல்லது இரட்டை இணைக்கப்பட்ட பட்டியல் கொண்டு

Chapter 2.indd 12
30-01-2020 19:08:01

செயல்படுத்தப்படலாம். இதுபோன்றே அடுக்கு (Stack) ADT மற்றும் வரிசை ADT, List பயன்படுத்தி செயல்படுத்தப்படலாம்.

தரவு அருவமாக்கம் நாம் இந்த உலகை பற்றி நினைப்பதை பிரிதிப்பலிக்கிறது. உதாரணமாக, ஒருவர் மகிழுந்தை ஓட்டி செல்வதற்கு, அதன் பொறி இயந்திரம் எவ்வாறு கட்டமைக்கப்பட்டது என்றோ டயர்கள் எவ்வகையான பொருட்கள் கொண்டு தயாரிக்கப்பட்டுள்ளது என்றோ தெரிந்து வைத்திறுக்க வேண்டியதில்லை மகிழுந்து எவ்வாறு இயக்கப்பட வேண்டும் என்ற பயிற்சி இருந்தால் போதும், தரவு அருவமாக்கினை செயல்படுத்த, ஆக்கிகள் (Constructor) மற்றும் செலக்டர் (Selectors) என்ற இரண்டு செயற்கூறுகள் உருவாக்கப்பட வேண்டும்.

2.3 ஆக்கிகள் மற்றும் செலக்டர்கள் (constructors and selectors)

ஆக்கி செயற்கூறுகள் அருவமாக்கம் தரவு வகையை கட்டமைக்க பயன்படுகிறது. செலக்டர் செயற்கூறுகள் தகவல்களை தரவு வகையிலிருந்து பெறுவதற்கு பயன்படுகிறது.

எடுத்துக்காட்டாக, *city* என்று ஒரு அருவமாக்க தரவு வகை உள்ளது என வைத்துக் கொள். *City* என்ற பொருள் நகரத்தின் பெயர், அட்சரேகை மற்றும் தீர்க்கரேகை பற்றிய தகவல்களை சேமித்திருக்கும் *city* என்ற பொருளை உருவாக்க பின்வரும் செயற்கூற்றினை பயன்படுத்தலாம்.

city = makecity (name, lat, lon)

city பொருளின் தகவல்களை பெறுவதற்கு பின்வரும் செயற்கூறுகளை பயன்படுத்தலாம்.

- getname(city)
- getlat(city)
- getlon(city)

பின்வரும் போலி குறிமுறை, இரு நகரங்களுக்கு இடையேயான தொலைதுரத்தை கணக்கிடும்:

```
distance(city1, city2):
    lt1, lg1 := getlat(city1), getlon(city1)
    lt2, lg2 := getlat(city2), getlon(city2)
      return ((lt1 - lt2)**2 + (lg1 - lg2)**2))^1/2
```

மேலே காணும் குறிமுறையில், distance(),getlat() மற்றும் getlon() ஆகியவை செயற்கூறுகள் ஆகும். lt என்பது அட்ச ரேசை மற்றும் lg என்பது தீர்க்கரேகையும் குறிக்கிறது. longitude. := என்பதை “assigned as” அல்லது “becomes” என்று வாசிக்க வேண்டும்.

lt1, lg1 := getlat(city1), getlon(city1)

இதனை lt1 என்பது getlat(city1)ன் மதிப்பாகிறது மற்றும் lg1 என்பது getlont(city1) ன் மதிப்பாகிறது. என்று வாசிக்க வேண்டும்.

இந்த செயற்கூறுகள் எவ்வாறு செயல்படுத்தப்படுகிறது என்பதை தெரிந்து கொள்ள வேண்டியதில்லை. இதை வேறு ஒருவர் நமக்காக வரையறுத்துள்ளார் என்று கருதிக் கொள்ள வேண்டும். பயனர் செயற்கூறுகள் எப்படி செயல்படுத்தப்படுகின்றது என்பது தெரிந்து வைத்திருக்க வேண்டியதில்லை எனினும் வேறு ஒருவரால் இச்செயற்கூறுகள் வரையறுக்கப்பட்டிருக்க வேண்டும்.

மேலே காணும் குறிமுறையில், ஆக்கி மற்றும் செலக்டர்களை அடையாளம் காண்போம். கண்ஸ்டரக்டரஸ் செயற்கூறுகள் அருவமாக்க தரவு வகையை கட்டமைக்கப் பயன்படுகிறது என்பதை நாம் அறிவோம். மேலே காணும் போலிக் குறிமுறையில், cityயின் பொருளை உருவாக்கும் செயற்கூறு, ஆக்கி ஆகும்.

city = makecity(name, lat, lon)

இங்கு makecity(name, lat, lon) என்ற ஆக்கி city எனும் பொருளை உருவாக்குகிறது.

Chapter 2.indd 13 30-01-2020 19:08:01

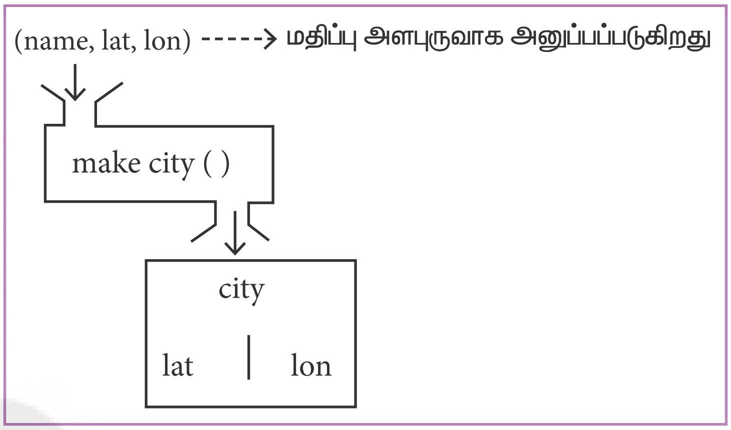
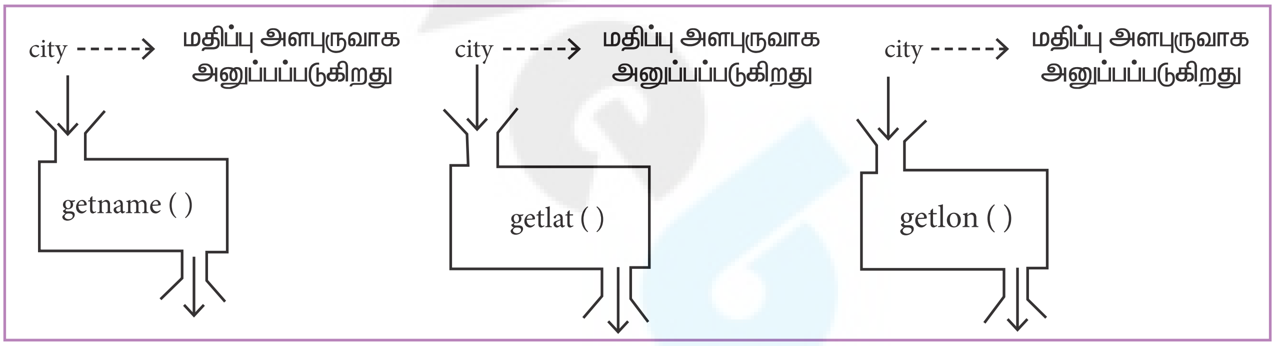


Fig 1 constructor

செலக்டார் செயற்கூறுகள் தகவல்களை தரவு வகையிலிருந்து பெறுவதற்கு பயன்படுகிறது. மேலே காணும் குறிமுறையில்,

- getname(city)
- getlat(city)
- getlon(city)

என்பவை city எனும் பொருளிலிருந்து தகவல்களை பெற்றுத் தரும் செலக்டர் செயற்கூறுகளாகும்.



படம் 2 செலக்டார்கள்

ஆக்கி மற்றும் செலக்டர்களை அடையாளம் காணும் மேலும் ஒரு எடுத்துக்காட்டைக் காணலாம். --என்ற குறியீட்டை குறிப்புரைகளாக என (Comments) என வாசிக்கவும்.

```
- - constructor
makepoint(x, y):
      return x, y
- - selector
 xcoord(point):
      return point[0]
- -selector
ycoord(point):
      return point[1]
```


Chapter 2.indd 14 30-01-2020 19:08:01

> **குறிப்பு**
>
> தரவு அருவமாக்கம் என்பது அருவமாக்க தரவு இனம் (Abstract Data Type (ADT)), அறிவிக்கப்பயன்படுகிறது. இது ஆக்கி மற்றும் செலக்டாரஸின் தொகுப்பாகும். ஆக்கிகள், பல்வேறு தகவல் துணுக்குகளை கொண்டு பொருள் உருவாக்கும், செலக்டார்ஸ் ஒவ்வொரு சிறு தகவல்களை பொருளிலிருந்து பெற உதவுகிறது.

2.4 விகிதமுறு எண்களைக் கொண்டு அருவமாக்க தரவு வகையை உருவமைப்பு செய்தல்

தரவு அருவமாக்கத்தின் அடிப்படை நோக்கம், நிரலினை அருவமாக்க தரவில் இயக்கும்படி கட்டமைப்பதாகும். அதாவது, நிரலில் தரவுகளை பயன்படுத்திக் கொள்ளும் போது அதைப்பற்றி சில ஊகங்களை செய்து கொள்ள வாய்ப்புள்ளது. அதை வேளையில் காண்கீரிட் தரவு உருவமைக்கும் போது நிரலின் தனிப்பகுதியாக வரையரை செய்ய வேண்டும்.

> **குறிப்பு**
>
> கான்கீரிட் தரவு உருவமைப்பில், அனைத்து செயற்கூறுகளின் வரையறையும் தெரிந்திருக்க வேண்டும்.

அனைத்து நிரல்களும் இரு பகுதிகளை கொண்டிருக்கும். அருவமாக்க தரவின் மீது செயல்படும் பகுதி மற்றும் அருவமாக்க தரவு வகையை செயல்படுத்தும் சில செயற்கூறுகளின் தொகுப்பை கொண்டு கான்கீரிட் உருவமைப்பை வரையறுக்கும் பகுதி என இரு பிரிவுகளை கொண்டிருக்கும். இந்த நுட்பத்தை புரிந்து கொள்வதற்கு, விகித முறு எண்களை கையாளும் சில செயற்கூறுகளுடன் வடிவமைக்கப்பட்ட எடுத்துக்காட்டை காணலாம்.

எடுத்துக்காட்டு:

விகிதமுறு எண்கள் முழு எண்களின் விகிதமாகும். அவை மெய் எண்களின் முக்கிய துணை வர்க்கமாக அமைகிறது. 8/3 அல்லது 19/23 போன்ற விகித முறு எண்களை பின்வருமாறு எழுதலாம்.

< தொகுதி >/< பகுதி >

தொகுதி மற்றும் பகுதி ஆகிய இரண்டும் முழு எண்ணின் இடம் மதிப்புகளாகும். இந்த இரண்டு பகுதிகளும் விகிதமுறு எண்களை சரியாக கனிப்பதற்கு பயன்படுகிறது. முழு எண்களை வகுக்கும் போது முழு எண்ணின் துல்லியம் இழந்து தோராயமான மிதவை எண் கிடைக்கிறது.

8/3 =2.6666666666666665

எனினும், சரியான முறையில் விகித முறு எண்களை குறிப்பதற்கு தொகுதி, பகுதி ஆகிய இரண்டையும் இணைந்து கொடுக்க வேண்டும்.

செயற்கூறு கருத்துக்களைக் கொண்டு, நிரலின் சில பகுதிகளை செயல்படுத்துவதற்கு முன்னதாக நாம் ஒரு நிரலின் செயல்திறனை மேம்படுத்திக் கொள்ள முடியும். நாம், ஒரு விகிதமுறு என்னை, தொகுதி மற்றும் பகுதி கொண்டு அமைக்க முடியும் என்ற கருதிக்கொள்வோம். அதே போன்று, கொடுக்கப்பட்ட விகிதமுறு எண்ணிற்கு நாம் தொகுதி மற்றும் பகுதியை தேர்ந்தெடுக்க முடியும். மேலும் கண்ஸ்டரக்டர் மற்றும் செலக்டர்ஸ் கொண்டும் நிறைவேற்றலாம்.

இதில் விஸ்புல் திங்கிங் என்ற சக்தி வாய்ந்த செயல் யுக்தியை பயன்படுத்தி நிரல் வடிவமைக்கப்படுகின்றது. நாம் இதுவரை, ஒரு விகிதமுறு எண் எவ்வாறு அமைக்க வேண்டும் என்றோ அல்லது ஒரு கண்ஸ்டரக்டர் மற்றும் செலக்டர் எவ்வாறு செயல்படுத்தப்படுகின்றது என்று நாம் பார்க்கவில்லை.

> **குறிப்பு**
>
> விஸ்புல் திங்கிங் என்பது யதார்த்தத் திற்கு பதிலாக நம்பிக்கையின் அடிப்படையில் விருப்பதற்கு ஏற்ப முடிவெடுப்பதாகும்.

எடுத்துக்காட்டு:– விகிதமுறு எண்ணின் ADT

- - constructor

- - constructs a rational number with numerator n, denominator d

rational(n, d)

- - selector

numer(x) → returns the numerator of rational number x

denom(y) → returns the denominator of rational number y

இப்பொழுது, விகிதமுறு எண்களின் செயல்பாடுகள் செலக்டார் செயற்கூறுகள் நியூமர் மற்றும் மீனாம் கொண்டும் மற்றும் கண்ஸ்டரக்டர் செயற்கூறு ரேஸனல் கொண்டும் வரையறுக்கப்படுகிறது. ஆனால் இச்செயற்கூறுகள் வரையறுக்கப்படவில்லை. நமது தேவை என்னவென்றால், தொகுதி மற்றும் பகுதியை இணைத்து கூட்டு மதிப்பை உருவாக்கும் வழிமுறையாகும்.

மேலே காணும் கண்ஸ்டரக்டர்ஸ் மற்றும் செலக்டர்ஸ் கொண்டு விகிதமுறு எண்களை உருவமைக்கும் போலிக் குறிமுறை

x,y:=8,3

rational(*x*,y)

numer(x)/denum(y)

- - output : 2.6666666666666665

2.5 Lists,Tuples

தரவு அருவமாக்கினை ஸ்திரமுடன் செயல்படுத்த, பைதான் போன்ற மொழிகள் பேர்ஸ் எனும் கூட்டு அமைப்பை வழங்குகிறது. இவை லிஸ்ட் மற்றும் டூபுல்ஸ் ஆகியவற்றால் உருவாக்கப்படுகிறது. பேர்ஸ்சை, லிஸ்ட் கொண்டு எளிதாக செயல்படுத்தலாம்.

2.5.1 List

List அமைப்பு கோவைகளை சதுர அடைப்புக்குறிக்குள் காற்புள்ளியில் பிரிக்கப்பட்டிருக்கும். இவ்கோவைகளை *List literal* என்று அழைக்கப்படுகிறது. *List* பல மதிப்புகளை சேமிக்கும் இம்மதிப்புகள் எவ்வகையாகவும் இருக்கலாம் அல்லது மற்றொரு லிஸ்டாகவும் இருக்கலாம்.

எடுத்துக்காட்டு List is [10, 20].

list-ன் உறுப்புகளை இரு வழியில் அணுகலாம். முதல் வழி, அதிகம் பயன்படுத்தப்படும் பன்மடங்கு மதிப்பிருத்தல். இம்முறையில், list-ன் உறுப்புகள் பிரிக்கப்பட்டு, அனைத்து உறுப்புகளும் வேறு பெயர்களுடன் இணைக்கப்படுகிறது.

lst := [10, 20]

x, y := lst

மேலே காணும் எடுத்துக்காட்டில், xயின் மதிப்பு 10 என்றும், Y யின் மதிப்பு 20 என மதிப்பிருத்தப்படும்.

இரண்டாம் முறையில் லிஸ்டின் உறுப்புகள், உறுப்புகள் தேர்வு மூலம் அணுகப்படுகிறது. செயற்குறிகள் சதுர அடைப்புக்குறிக்குள் கொடுக்கப்பட்டிருக்கும். லிஸ்ட் லிட்டரல்ஸ் போல் அல்லாது கோவையில் ஒரு சதுர அடைப்புக்குறியை தொடர்ந்து வரும் மற்றொரு சதுர அடைப்புக்குறி லிஸ்டின் மதிப்புகளாக எடுத்துக்கொள்ளப்படுவதில்லை. மற்றாக முந்தய கோவையின் உறுப்பை மதிப்பாக தேர்வு செய்யும்.

lst[0]

10

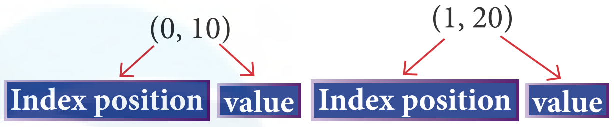
lst[1]

20

மேலே காணும் எடுத்துக்காட்டினை கணித முறையில் set போன்று அமைக்கலாம்.

Chapter 2.indd 16 30-01-2020 19:08:01

lst[(0, 10), (1, 20)] - where



இரு மதிப்புகளை ஒன்றாக இணைக்கும் முறையை *Pairs* என்று அழைக்கிறோம். *List* அதிகமாக பயன்படுத்தப்படும் முறையாகும். எனவே, *List*, *Pairs* என்று அழைக்கப்படுகிறது.

2.5.1.1 List கொண்டு விகிதமுறு எண்களை உருவமைப்பு.

போலிக் குறிமுறையில், தொகுதி, பகுதி ஆகிய இரு முழு எண்களின் Pairs கொண்டு விகிதமுறு எண்களை உருவமைக்கலாம்.

```
rational(n, d):
    return [n, d]
numer(x):
    return x[0]
denom(x):
    return x[1]
```

2.5.2 Tuple

Pairs என்பது இரு தரவு பகுதிகளை சேமிக்கும் கலவை தரவு என்பதை நினைவில் கொள்ள வேண்டும். இதுவரை இருவேறு முறைகளில் pairs தரவு வகை உருவமைப்பை பார்த்தோம். முதல் வழி, List கொண்டு உருவமைத்தல் மற்றும் இரண்டாம் வழி pairs கொண்டு வடிவமைத்தல் ஆகும்.

tuple என்பது அடைப்புக்குறிக்குள் கொடுக்கப்பட்டுள்ள தொடர் மதிப்புகளை காற்புள்ளியில் பிரிக்கப்பட்டிருக்கும். tuple என்பது List போன்றதாகும். இவை இரண்டிற்கும் உள்ள வேறுபாடு என்னவென்றால் tuple-ல் கொடுக்கப்பட்ட உறுப்புகளை மாற்ற முடியாது, ஆனால் List-ல் இடம்பெற்றுள்ள உறுப்புகளை மாற்றலாம்.

எடுத்துக்காட்டு colour= ('red', 'blue', 'Green')

2.5.2.1 Tuple-சை Pair உருவமைத்தல்

```
nums := (1, 2)
nums[0]
1
nums[1]
2
```

சதுர அடைப்புக்குறி குறியீடு, பேர்ஸில் சேமிக்கப்பட்டுள்ள தரவுகளை அணுகுவதற்கு பயன்படுத்துகிறது என்பதை கவனிக்கவும். தரவுகள் பூஜ்யம் சுட்டு வரிசைப்பட்டுள்ளதால், முதல் உறுப்பு nums[0] என்றும் இரண்டாம் உறுப்பு nums[1] என்று அணுகப்படும்.

2.6 ஸ்டரக்சர்லில் தரவு அருவமாக்கம்

List தரவு அருவமாக்கம் மூலம் நினைவக செல்கள் தொகுப்பிற்கு பெயரிட உதவுகின்றது என்பதை நாம் முன்னதாக அறிந்துள்ளோம். உதாரணமாக மாஸ்டர் மைண்ட் விளையாட்டில், ஆட்டக்காரர்கள் யூகிக்கும் நான்கு நிறங்களின் பட்டியலை கண்காணிக்க வேண்டும். நான்கு வெவ்வேறு மாறிகளுக்கு (color1, color2, color3, மற்றும் color4) மாற்றாக 'Predict', என்ற ஒற்றை மாறியை பயன்படுத்தலாம்.

எடுத்துக்காட்டு, Predict =['red', 'blue', 'green', 'green']

List பல விருப்பப்படி பொருளின் பல்வேறு பகுதிகளை பெயரிட அனுமதிப்பதில்லை. Predict முறையில் நீங்கள் பகுதிகளை பெயரிட தேவையில்லை. அட்டவணையைக் கொண்டு ஒவ்வொரு நிரத்தினையும் பெறலாம்.

ஆனால் மிகவும் சிக்கலான விஷயத்தில் ஒரு நபர் எடுத்துக்கொண்டால், அதில் பல உருப்படி பொருளில், அனைத்து உருப்புகளும் பெயரிடப்பட்டிருக்கும். முதல் பெயர், கடைசி பெயர், அடையாளம், மின்னஞ்சல் ஒரு நபரை லிஸ்ட் கொண்டு குறிப்பிடலாம்:

Chapter 2.indd 17 30-01-2020 19:08:01

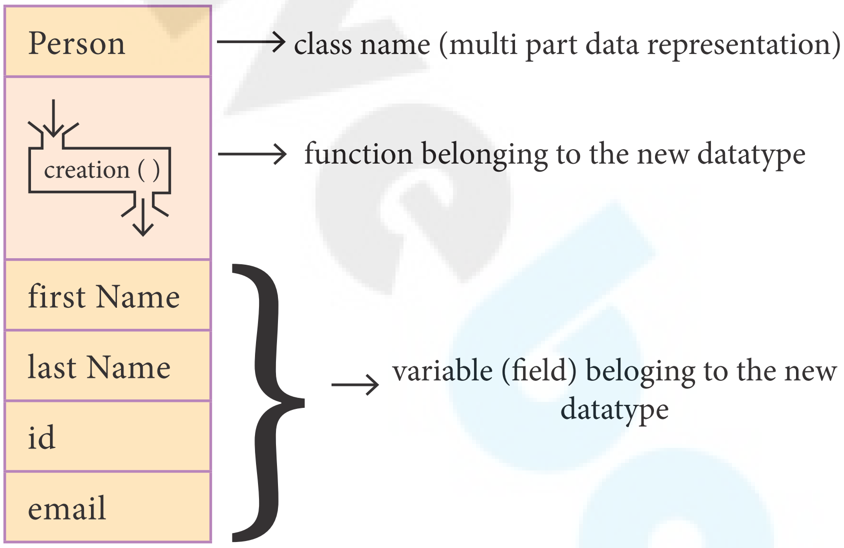
person=['Padmashri', 'Baskar', '994-222-1234', 'compsci@gmail.com']

ஆனால் இது போன்ற குறியீடு ஒவ்வொரு உருப்படியும் எதை குறிக்கின்றது என்று வெளிப்படையாக குறிப்பதில்லை.

இச்சிக்கலை list பயன்படுத்துவதற்கு மாற்றாக structure அமைப்பை (OOP மொழியில் இது இனக்குழு அமைப்பு என்று அழைக்கப்படுகிறது) கொண்டு பல உருப்படி பொருளில் அனைத்து உருப்புகளும் பெயரிடப்படும் (பெயர் கொடுக்கப்பட்டு பின்வரம் போலிக்குறிமுறையை காணவும்:

```
class Person:
        creation( )
        firstName := " "
        lastName := " "
        id := " "
        email := " "
```

Person எனும் புதிய தரவு வகையை படவடிவில் உருவமைக்கலாம்.



பின்வருபவனவற்றை main() கொண்டிருக்கும்	
p1:=Person()	பொருள் உருவாக்கும் கூற்று.
firstName := " Padmashri "	-- firstName எனும் புலத்திற்கு Padmashri என மதிப்பிருத்தப்படுகிறது.
lastName :="Baskar"	lastName எனும் புலத்திற்கு Baskar என மதிப்பிருத்தப்படுகிறது.
id :="994-222-1234"	id எனும் புலத்திற்கு 994-222-1234 என மதிப்பிருத்தப்படுகிறது.
email="compsci@gmail.com"	email எனும் புலத்திற்கு compsci@gmail.com என மதிப்பிருத்தப்படுகிறது.
- - output of firstName : Padmashri	

Chapter 2.indd 18 30-01-2020 19:08:01

இனக்குழு கட்டமைப்பு ஒரு நபரை குறிக்கும் பல பகுதி பொருளுக்கு வடிவம் வரையறுக்கிறது. இந்த வரையறுப்பு புதிய தரவு வகையை சேர்க்கிறது. இதில் நபர் என்பது வகையாகும். ஒரு முறை வரையறுத்தலின், நாம் புதிய மாறிகளை உருவாக்கலாம். இந்த எடுத்துக்காட்டில் Person என்பது இனக்குழு அல்லது வகையாகும. p1 என்பது தனிப்பட்ட குக்கீ குறிக்கின்றது. குக்கீ கட்டர் கொண்டு பல குக்கீகளை தயாரிக்கலாம். அதுபோன்று இவ்இனக்குழுவைக் கொண்டு பல பொருள்களை உருவாக்கலாம்.

இதுவரை, இனக்குழு தொடர்புடைய தரவுகளைக் கொண்டு தரவு அருவமாக்கத்தை எவ்வாறு வரையறுக்கிறது என்று பார்த்தோம். ஓர் இனக்குழு என்பது தரவுகளை மட்டுமில்லாது செயற்கூற்றினையும் தன்னுடன் கொண்டிருக்கும். இவ்வகை செயற்கூறுகள், இனக்குழுவிற்கு கீழ்படந்தவை, ஏனெனில் இவை இனக்குழுவின் தரவுகளை வைத்து செயல்படுகிறது. எடுத்துக்காட்டு Person பொருளின் தரவுகளை மாற்றம் செய்தல் அல்லது ஆய்வு செய்தல் எனவே, இனக்குழு என்பது தரவு மற்றும் அத்தரவின் மீது செயலாற்றும் செயற்கூறுகளை கொண்ட தொகுப்பாகும். முன்னதாக பார்த்த எடுத்துக்காட்டு மற்றும் விளக்கத்தின் மூலம் அருவமாக்கம் என்பது சிக்கலான தரவுகளை எளிய முறையில் கையாள உதவுகின்றது என்பதை உணரலாம்.

Therefore we can define ***a class as bundled data and the functions that work on that data***. From All the above example and explanation one can conclude the beauty of data abstraction is that we can treat complex data in a very simple way.

நினைவில் கொள்க

- ஒரு பொருளின் இயக்கம் அருவமாக்கம் தரவு வகையின் (ADT) அல்லது இனக்குழு தொடர் மதிப்பு மற்றும் தொடர் செயல்பாடுகள் மூலம் வரையறுக்கப்படுகிறது.
- ADT வரையறை, எத்தகைய செயல்பாடுகள் செயப்பட வேண்டும் என்பதை குறிக்கும், ஆனால் இச்செயல்பாடுகள் எவ்வாறு செயல்படுத்தப்படும் என்று குறிப்பிடப்படுவதில்லை.
- ADT தரவுகள் நினைவகத்தில் எவ்வாறு சேமிக்கப்படுகிறது என்றோ செயல்பாடுகளை நிறைவேற்றும் வழிமுறைகளை குறிப்பிடுவதில்லை.
- கண்ஸ்டரக்டர் செயற்கூறுகள் அருவமாக்கம் தரவு வகையை உருவாக்குகின்றது.
- செலக்டர்ஸ் செயற்கூறுகள் தரவு வகையிலிருந்து தகவல்களை மீட்டெடுக்க பயன்படுகிறது.
- கான்கீரிட் தரவு வகை (CDT's) அல்லது ஸ்டக்சர்ஸ் எளிய கருத்தினை செயல்படுத்த உதவுகிறது.
- அருவமாக்கம் தரவு வகை (ADT's) ஒரு கருத்து எவ்வாறு செயல்படுத்தப்படுகின்றது என்பதை தவிர்த்து, அக்கருத்து பற்றிய உயர் நிலை பார்வை (மற்றும் பயன்பாடு) வழங்குகிறது.
- காண்கீரிட் தரவு வகையின் உருவமைப்பு அறியப்பட்டது, ஆனால் அருவமாக்கம் தரவு வகையின் உருவமைப்பு அறியப்படாது.
- Pairs எனும் கலவை அமைப்பு, List அல்லது Tuples கொண்டு உருவாக்கப்படுகிறது.
- List கோவை, சதுர அடைப்புக்குறிக்குள் காற்புள்ளியால் பிரிக்கப்பட்டு உருவமைக்கப்படும்.
- List-ல் உள்ள உருப்புகளை இருவழியில் அணுகலாம். முதல் வழிமுறை, பல்வேறு மதிப்பிருத்தல் முறை மற்றும் இரண்டாம் வழிமுறை உறுப்பு தேர்ந்தெடுப்பு செயற்கூறி ஆகும்.
- இரு மதிப்புகளை ஒன்றாக பிணைப்பது பேர்ஸ் என்று கருதப்படுகிறது.
- பல் உருப்பு பொருளின் பல்வேறு பகுதிகளை பெயரிட List அனுமதிப்பதில்லை.

Chapter 2.indd 19 30-01-2020 19:08:02

GXXBD2

மதிப்பீடு

பகுதி–அ

சரியான விடையை தேர்ந்தெடுத்து எழுதுக (1 மதிப்பெண்)

1. பின்வரும் எந்த செயற்கூறு அருவமாக்கம் தரவு வகையை உருவமைக்கப் பயன்படுகிறது ?

   (அ) Constructors (ஆ) Destructors (இ) recursive (ஈ)Nested

2. பின்வரும் எந்த செயற்கூறு தரவு வகையில் இருந்து தகவல்களை மீட்டெடுக்கும்?

   (அ) Constructors (ஆ) Selectors (இ) recursive (ஈ)Nested

3. வரிசைப்படுத்தப்பட்ட உருப்புகளை மாற்றக்கூடிய தரவு கட்டமைப்பு

   (அ) Built in (ஆ) List (இ) Tuple (ஈ) Derived data

4. மாற்ற செய்ய முடியாத பொருளின் தொடர் வரிசை

   (அ) Built in (ஆ) List (இ) Tuple (ஈ) Derived data

5. உருவமைப்பு அறியப்பட்ட தரவு வகை எவ்வாறு அழைக்கப்படுகிறது.

   (அ) Built in datatype (ஆ) Derived datatype

   (இ) Concrete datatype (ஈ) Abstract datatype

6. உருவமைப்பு அறியப்படாத தரவு வகை எவ்வாறு அழைக்கப்படுகிறது

   (அ) Built in datatype (ஆ) Derived datatype

   (இ) Concrete datatype (ஈ) Abstract datatype

7. பின்வருவனவற்றில் எது கலவை அமைப்பு?

   (அ) Pair (ஆ) Triplet (இ) single (ஈ) quadrat

8. இரு மதிப்புகளை ஒன்றாக பிணைப்பு எந்த வகை கருதப்படுகிறது.

   (அ) Pair (ஆ) Triplet (இ) single (ஈ) quadrat

9. பின்வருவனவற்றில் எது பல் உருப்பு பொருளின் பல்வேறு பகுதிகளை பெயரிட அனுமதிக்கிறது?

   (அ) Tuples (ஆ) Lists (இ) Classes (ஈ) quadrats

10. பின்வருவனவற்றில் எது கோவைகளை சதுர அடைப்புக்குறிக்குள் வைத்து உருவமைகிறது ?

    (அ) Tuples (ஆ) Lists (இ) Classes (ஈ) quadrats

பகுதி–ஆ

அனைத்து வினாக்களுக்கும் விடையளி (2 மதிப்பெண்)

1. தரவு அருவமாக்கம் வகை என்றால் என்ன?
2. ஆக்கிகள் மற்றும் செலக்டர்கள் வேறுபாடு தருக.

3. Pair என்றால் என்ன? எடுத்துக்காட்டு தருக..

4. List என்றால் என்ன? எடுத்துக்காட்டு தருக

5. Tuple என்றால் என்ன? எடுத்துக்காட்டு தருக.

பகுதி–இ

அனைத்து வினாக்களுக்கும் விடையளி **(3 மதிப்பெண்)**

1. கான்கிரிட் தரவு வகை மற்றும் அருவமாக்கம் தரவு வகை வேறுபடுத்துக.
2. நிரல் வடிவமைப்பில் பின்பற்றப்படும் யுக்தி எது? யுக்தியை வரையறுக்க.
3. பின்வருவனவற்றில் எது constructors and selectors என்று அடையாளம் காணவும்?

   (a) N1=number() (b) accetnum(n1) (c) displaynum(n1)

   (d) eval(a/b) (e) x,y= makeslope (m), makeslope(n)

   (f) display()
4. list உள்ள உருப்புகளை அணுகும் பல்வேறு வழிமுறைகள் யாவை? எடுத்துக்காட்டு தருக.
5. பின்வருவனவற்றில் எது List, Tuple மற்றும் இனக்குழு (class) என அடையாளம் காண்க.

   (a) arr [1, 2, 34] (b) arr (1, 2, 34) (c) student [rno, name, mark]

   (d) day= (‘sun’, ‘mon’, ‘tue’, ‘wed’) (e) x= [2, 5, 6.5, [5, 6], 8.2]

   (f) employee [eno, ename, esal, eaddress]

பகுதி–ஈ

அனைத்து வினாக்களுக்கும் விடையளி **(5 மதிப்பெண்)**

1. தரவு அருவமாக்கம் எவ்வாறு செயல்படுத்துவாய்? எடுத்துக்காட்டுடன் விளக்குக.
2. List என்றால் என்ன? ஏன் List, Pairs என்று அழைக்கப்படுகிறது. எடுத்துக்காட்டுடன் விவரி.
3. பல் உருப்பு பொருளை எவ்வாறு அணுகுவாய் எடுத்துக்காட்டுடன் விளக்கு.

Reference Books

1. *Data structure and algorithmic thinking with python by narasimha karumanchi*
2. *sign and analysis of algorithms by s sridhar*
3. *Data Structures and Algorithms in Python by Goodrich, Tamassia & Goldwasser*
4. *https://www.tutorialspoint.com*

அலகு I

பாடம் 3

வரையெல்லை

ZDBQDJ

கற்றலின் நோக்கங்கள்

இந்தப் பாடப்பகுதியைக் கற்றபின் மாணவர்கள் அறிந்துக் கொள்வது,

- வரையெல்லையை பற்றி புரிந்து கொள்ளல்
- LEGB விதியை நடைமுறைப்படுத்துதல்
- தொகுதிகள் (Modules) பற்றி புரிந்து கொள்ளல்
- நிரலாக்க மொழியில் அணுகுதலின் கட்டுப்பாட்டை செயலாக்கம் பற்றிஅறிதல்.

3.1 அறிமுகம்

வரையெல்லை என்பது மாறிகள், அளபுருக்கள் மற்றும் செயற்கூறுகளின் அணுகியல்பை நிரலின் ஒரு பகுதியில் இருந்து மற்றொரு பகுதிக்கு குறிப்பதாகும். அதாவது, நிரலின் எந்தப் பகுதியை அணுக அல்லது பயன்படுத்த முடியும் என்பதைக் குறிக்கிறது. பொதுவாக, நிரலில் வரையறுக்கப்பட்ட ஒவ்வொரு மாறியும் முழுதளாவிய வரையெல்லையைக் கொண்டுள்ளன. ஒரு முறை வரையறுக்கப்பட்டால், நிரலின் ஒவ்வாரு பகுதியும் அந்த மாறியை அணுக முடியும். ஆனால், ஒரே ஒரு வரையறைக்குள் மாறிகளின் வரையெல்லை உட்படுத்துவது சிறந்த வழிமுறை ஆகும். இதில் எதிர்பாராத விதமாக செயற்கூறுக்கு உள்ளே உள்ள மாறிகளில் ஏற்படும் மாற்றங்கள் செயற்கூறுவுக்கு வெளியே எந்த மாற்றத்தையும் ஏற்படுத்தாது.

3.2 மாறியின் வரையெல்லை

நிரலாக்க மொழியில் உள்ள மாறிகளின் வரையெல்லையை அறிந்து கொள்வதற்கு மாறிளைப் பற்றி தெரிந்து கொள்ளுதல் அவசியம். முக்கியமாக, இவைகள் நினைவகத்தில் உள்ள ஒரு பொருளின் முகவரியாகும். (குறிப்புகள் (அ) சுட்டுகள்) ஒரு பொருளுக்கு ஒரு மாறியை : = குறியுடன் இருத்தும்போது மாறியும் பொருளும் ஒன்றாகப் பிணைக்கப்படுகிறது. ஒன்றிக்கு மேற்பட்ட மாறிகளுக்கு மதிப்புகள் இருக்க முடியும் (Map).

குறிப்பு

மாறியின் பெயரை ஒரு பொருளுடன் பிணைக்கும் செயல்முறையே மேப்பிங் (Mapping) எனப்படும். : = என்ற குறியீடு செயல்குறி நிரலாக்க மொழியில் மாறி மற்றும் பொருளை = (சமக்குறி) (Map) இருக்கிறது.

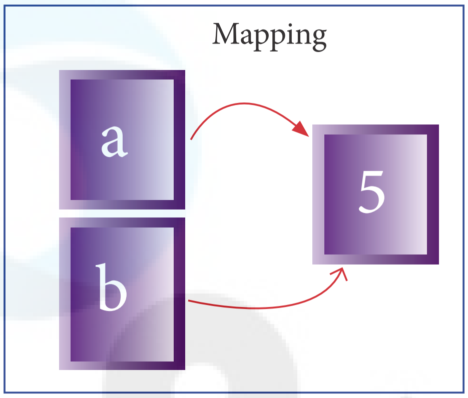
நிரலாக்க மொழி இதைப்போன்ற அனைத்து மேப்பிங்களையும் namespaces உடன் கண்காணிக்கிறது. namespaces என்பது மாறியின் பெயரை பொருளுடன் மேப்பிங் செய்வதற்கான கொள்கலனாகும். இவைகளை Dictionary, சொற்களைக் கொண்ட List மற்றும் அதற்கான அர்த்தங்களாகவும் எடுத்துக் கொள்ளலாம். Dictionary-ல் வார்த்தைகள் அதன் அர்த்தங்களுடனும், பெயர்கள் பொருள்களுடனும் (name:=object) மேப் செய்யப்பட்டுள்ளன. இது நீங்கள் தேர்வு செய்த பெயரால் பொருளை அணுக அனுமதிக்கிறது. பின்வரும் எடுத்துக்காட்டில் a என்பது 5 என்ற முழு எண்ணுடன் முதலில் மேப் செய்யப்படுகிறது. இங்கு a என்பது மாறியின் பெயர். முழு எண் மதிப்பு 5 என்பது பொருளாகும். பின்னர், a என்பது b-ல் இருத்தப்படுகிறது. அதாவது, b என்பது a-ல் உள்ள அதே முழு எண் மதிப்பாகிறது.

மேப்பிங்

Chapter 3.indd 22 30-01-2020 19:11:47

1. a:=5

2. b:=a

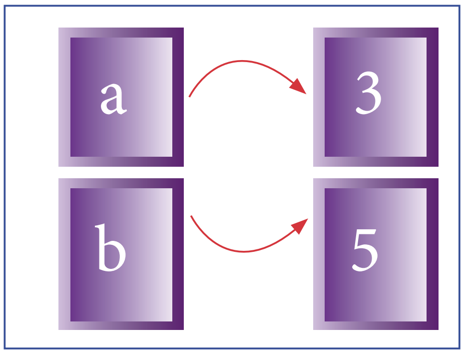


மேலே கொடுக்கப்பட்டுள்ள கூற்றுகளில் சில மாற்றத்தை கீழே வருமாறு காண்போம்.

1. a:=5
2. b:=a
3. a:=3

அதன் பிறகு a-ன் மதிப்பை 3 என மாற்றினால் தொடக்க நிரலர் b-ன் மதிப்பும் 3 என மாறும் என்று கருதுவார். ஆனால், b-ன் மதிப்பு தற்பொழுதும் 5 என்ற முழு எண் மதிப்பாகவே இருக்கும். a-ன் மதிப்பு மட்டுமே 3 என்ற முழு எண் மதிப்பாக மாற்றம் பெற்றிருக்கும்.

a-வின் மதிப்பை மாற்றிய பிறகு மேப்பிங்



மாறியின் வரையெல்லை என்பது அது குறிமுறையில் எங்கு புலப்படுகிறதோ அல்லது காணப்படுகிறதோ அந்தப் பகுதியாகும். அதை அணுகுவதற்கு எந்த முன்னொட்டையும் பயன்படுத்த வேண்டிய அவசியமில்லை. ஒரு எடுத்துக்காட்டு,

1. **Disp**():
2.   a:=7

a-வின் மதிப்பை குறிமுறைக்கு வெளியே காண்பிக்க முற்படும்போது நிரல் **"name 'a' is not defined"** என்ற பிழைச் செய்தியை வெளியிடும். ஏனெனில், மாறியின் வாழ்நாள் குறிமுறையின் இறுதிவரை மட்டுமே ஆகும். **ஒரு மாறி குறிமுறையில் பயன்படும் நேரமே அதனுடைய வாழ்நாள் 'Life time'** எனப்படும்.

3.3 LEGB விதிமுறை

வரையெல்லை என்பது சரியான மதிப்பை பெறுவதற்காக மாறிகள் எந்த வரிசையில் பொருளுடன் Map செய்யப்பட வேண்டும் என்பதை வரையறுக்கிறது. கீழே கொடுக்கப்பட்ட எளிய எடுத்துக்காட்டைக் காண்போம்.

1. x:= 'outer x variable'
2. display():
3.   x:= 'inner x variable'
4.   print x
5. display()

மேலே உள்ள கூற்றுகளை இயக்கும்போது கூற்று (4) மற்றும் (5) பின்வரும் விடையைக் காண்பிக்கிறது.

வெளியீடு

outer x variable

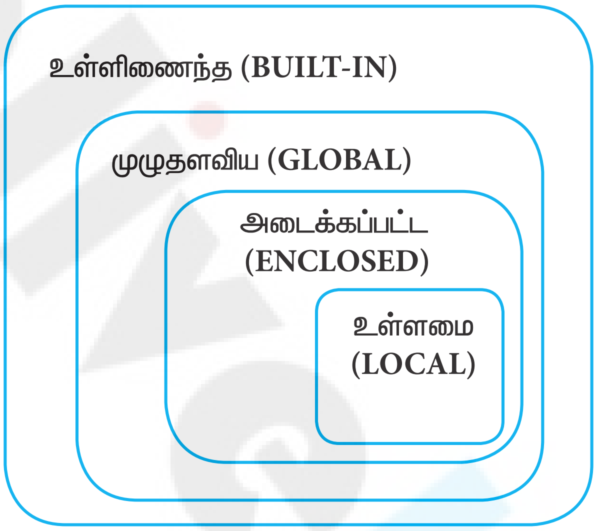
inner x variable

மேலே உள்ள கூற்றுகள் வெவ்வேறு வெளியீடுகளைத் தருகிறது. ஏனெனில், மாறி *x* என்பது வெவ்வேறு வரையெல்லைகளில் உள்ளது. ஒன்று display ( ) என்ற செயற்கூறுவுக்கு உள்ளேயும், மற்றொன்று அதன் மேல் கூற்றிலும் உள்ளது. **'Outer *x* Variable'** என்ற மதிப்பு, *x* என்பது செயற்கூறுவின் வரையறைக்கு வெளியில் அணுகப்படும்போது வெளியிடப்படுகிறது. ஆனால் display ( ) செயற்கூறு இயக்கப்படும்போது, **"inner *x* Variable"** என்ற மதிப்பு அச்சிடப்படுகிறது. இது செயற்கூறு வரையறைக்குள் உள்ள *x*-ன் மதிப்பாகும். மேலே கொடுக்கப்பட்டுள்ள எடுத்துக்காட்டில், ஒரு மாறி எந்த வரையெல்லையில் எடுத்துக் கொள்ளப்பட வேண்டும் என்பதை தீர்மானிக்க ஒரு விதிமுறை பின்பற்றப்படுவதை கணித்துக் கொள்ள முடிகிறது.

Chapter 3.indd 23
30-01-2020 19:11:47

LEGB விதி வரையெல்லை தேடப்பட வேண்டிய (Scope resolution) வரிசையை தீர்மானிக்கப் பயன்படுகிறது. வரையெல்லைகள் பின்வருமாறு படிநிலை முறையில் பட்டியலிடப்பட்டுள்ளன. (பெரியதிலிருந்து சிறியது)

உள்ளமை Local(**L**)	செயற்கூறு / இனக்குழுவிற்கு உள்ளே வரையறுக்கப்பட்டவை
இணைக்கப்பட்ட Enclosed(**E**)	பின்னலான செயற்கூறுகளுக்குள் வரையறுக்கப்பட்டவை
முழுதளவிய Global(**G**)	மேல்நிலையில் வரையறுக்கப்பட்டவை
உள்ளிணைந்த Built-in (**B**)	உள்ளிணைந்த செயற்கூறுகளில் (கூறுகள்) உள்ள முன்னரே வரையறுக்கப்பட்ட பெயர்களாகும்.



3.4 மாறியின் வரையெல்லை வகைகள்

4 வகையான வரையெல்லைகள் உள்ளன. அவற்றை ஒன்றன்பின் ஒன்றாகக் காண்போம்.

3.4.1 உள்ளமை வரையெல்லை (Local Scope)

உள்ளமை வரையெல்லை, நடப்பு செயற்கூறில் வரையறுக்கப்பட்ட மாறிகளைக் குறிக்கும். செயற்கூறு, எப்பொழுதும் மாறியின் பெயரை முதலில் அதன் உள்ளமை வரையெல்லையில் பார்வையிடும் அந்த வரையெல்லையில் இல்லையென்றால் மட்டுமே வெளி வரையெல்லையில் சோதிக்கும். இந்த எடுத்துக்காட்டை காண்போம்.

	முழுநிரல்	நிரலின் வெளியீடு
1. **Disp():** 2. a:=7 3. print a 4. Disp()	Disp( ): a:=7 print a Disp ( )	7

மேலே உள்ள குறிமுறையை இயக்கும்போது, மாறி a என்பது 7 என்ற மதிப்பை வெளியிடுகிறது. ஏனெனில், இது உள்ளமை வரையெல்லையில் வரையறுக்கப்பட்டு, அங்கேயே அச்சிடப்படுகிறது.

3.4.2 முழுதளாவிய வரையெல்லை

நிரலின் அனைத்து செயற்கூறுகளுக்கும் வெளியே அறிவிக்கப்பட்ட மாறிகள் முழுதளாவிய மாறிகள் எனப்படும். அதாவது, முழுதளாவிய மாறிகளை நிரலின் அனைத்து செயற்கூறுகளுக்கு உட்புறமும், வெளிப்புறமும் அணுக முடியும். பின்வரும் எடுத்துக்காட்டைக் காண்போம்.

1. a:=10 2. **Disp()**: 3. a:=7 4. print a 5. Disp() 6. print a	முழுநிரல் a:=10 Disp( ) a:=7 print a Disp ( ): print a	நிரலின் வெளியீடு 7 10

மேலே உள்ள குறிமுறையில் disp ( ) என்ற செயற்கூறு அழைக்கப்படும்போது, அதனுள் வரையறுக்கப்பட்டிருக்கும் மாறி **a**, 7 என்ற மதிப்பை வெளியிடும், பின்னர் இது 10 என்ற மதிப்பை வெளியிடும் ஏனெனில் **a** என்ற மாறி முழுதளாவிய வரையெல்லையில் வரையறுக்கப்பட்டுள்ளது.

3.4.3. அடைக்கப்பட்ட வரையெல்லை

அனைத்து நிரலாக்க மொழிகளும் பின்னலான செயற்கூறுகளைக் கொடுக்க அனுமதிக்கின்றன. ஒரு செயற்கூறின் (வழிமுறை) உள்ளே மற்றொரு செயற்கூறு அடைக்கப்பட்டிருந்தால் அது **பின்னலான செயற்கூறு** எனப்படும். **மற்றொரு செயற்கூறு வரையறையை, தன்னுள் கொண்ட ஒரு வெளி செயற்கூறினுள் ஒரு மாறி அறிவிக்கப்பட்டால், உள்செயற்கூறானது, வெளி செயற்கூறினுள் உள்ள மாறிகளை அணுக முடியும். இதுவே, அடைக்கப்பட்ட வரையெல்லை எனப்படும்.**

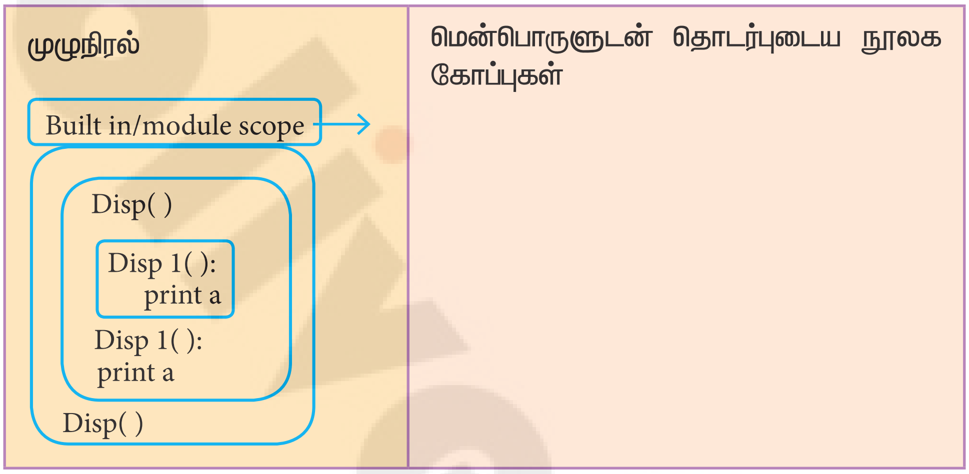
நிரல்பெயர்ப்பி அல்லது தொகுப்பான் ஒரு நிரலில் மாறியை தேடும்பொழுது அது முதலில் உள்ளமை வரையெல்லையில் தேடும். பின்னர் அடைக்கப்பட்ட வரையெல்லையில் தேடும். பின்வரும் எடுத்துக்காட்டை காண்போம்.

1. Disp(): 2. a:=10 3. Disp1(): 4. print a 5. Disp1() 6. print a 7. Disp()	முழுநிரல் Disp( ) a:=10 Disp1( ): print a Disp1( ): print a Disp( )	நிரலின் வெளியீடு 10 10

மேலே குறிப்பிட்ட எடுத்துக்காட்டில் Disp1 ( ) என்ற செயற்கூறு Disp ( ) என்ற செயற்கூறினுள் வரையறுக்கப்பட்டுள்ளது. 'a' என்ற மாறி Disp ( ) என்ற செயற்கூறினுள் வரையறுக்கப்பட்டிருந்தாலும் Disp 1 ( ) என்ற செயற்கூறும் அதைப் பயன்படுத்த முடியும். ஏனெனில் இந்த செயற்கூறு Disp ( ) என்பதன் உறுப்பாகும்.

3.4.4. உள்ளிணைந்த வரையெல்லை

இறுதியாக, நாம் விரிந்த வரையெல்லை பற்றி விவாதிப்போம். நிரல்பெயர்ப்பி அல்லது தொகுப்பாணை தொடங்கும் பொழுது உள்ளிணைந்த வரையெல்லையனானது நிரல் வரையெல்லையில் ஏற்கனவே கொடுக்கப்பட்ட அனைத்து பெயர்களையும் கொண்டிருக்கும். நிரலாக்க மொழியின் நூலக செயற்கூறினுள் வரையறுக்கப்பட்ட மாறி அல்லது தொகுதி உள்ளிணைந்த வரையெல்லையைக் கொண்டுள்ளது. இவைகள், நூலக கோப்புகள் நிரலில் செயல்பட தொடங்கியவுடன் இறக்கப்படும்.



பொதுவாக செயற்கூறுகள் அல்லது தொகுதி மட்டுமே மென்பொருளுடன் தொகுப்பாக வருகிறது. எனவே, இவைகள் உள்ளிணைந்த வரையெல்லையின் கீழ் வருகின்றன.

3.5 தொகுதி (Module)

நிரலின் ஒரு பகுதியே தொகுதியாகும். நிரல்கள் ஒன்று அல்லது அதற்கு மேற்பட்ட தனித்து உருவாக்கப்பட்ட தொகுதிகளால் அமைக்கபடுகிறது. ஒரு தொகுதி தொடர்புடைய பல கூற்றுகளை கொண்டுள்ளது. தனித்த நிலையில் தொகுதிகள் சரியாக வேலை செய்கிறது மற்றும் பிற தொகுதிகளோடு ஒருங்கிணைக்கப் படுகிறது. ஒரு நிரலை எளிதாக்கவும், பிழை திருத்தவும் அது தொகுதிகளாக பிரிக்கப்பட்டு வெளியீடு பெறப்படுகிறது. ஒரு கணிப்பொறி நிரலை பல துணை நிரல்களாக பிரிக்கும் செயல்முறையே தொகுதி நிரலாக்கம் எனப்படும். தொகுதி நிரலாக்கத்திற்கு எடுத்துக்காட்டுகள், செய்லமுறைகள், துணை நிரல்கள் மற்றும் செயற்கூறுகள்.

3.5.1 தொகுதியின் பண்புயியல்புகள்

தொகுதி பண்பியல்கள் பின்வருமாறு:

1. தொகுதிகள் தரவு, தகவல் மற்றும் தருக்க செயலாக்கத்தைக் கொண்டுள்ளன.
2. தொகுதிகள் தனியாக தொகுக்கப்பட்டு நூலகத்தில் சேமிக்கப்படும்.
3. தொகுதிகள் நிரலில் சேர்க்க முடியும்.
4. ஒரு பெயரையும், சில அளபுருக்களையும் பயன்படுத்தி தொகுதி பிரிவுகள் செயல்படுத்தப்படுகின்றன.
5. ஒரு தொகுதியின் பிரிவுகள் மற்ற தொகுதிகளால் பயன்படுத்தப்படுகின்றன.

3.5.2 தொகுதி நிரலாக்கத்தின் பயன்கள்

- குறைந்த வரிகளைக் கொண்ட குறிமுறையை எழுதினால் போதுமானது.
- மறுபயனாக்கத்திற்கும் பலமுறை குறிமுறை தட்டச்சு செய்வதை தவிர்ப்பதற்கு, ஒற்றை செயல்முறையை உருவாக்க வேண்டும்.
- நிரல்கள் மிக எளிதாக வடிவமைக்கப் படுகின்றன. ஏனெனில், முழு குறிமுறையும்

Chapter 3.indd 26
30-01-2020 19:11:47

சிறிய பகுதிகளாக பிரிக்கப்பட்டு சிறிய குழுவினரால் கையாளப்படுகிறது.

- தொகுதி நிரலாக்கம் பல நிரலர்களை ஒரே பயன்பாட்டில் வேலை செய்ய அனுமதிக்கிறது.
- பல கோப்புகளில் இந்த குறிமுறை சேமிக்கப்படுகிறது.
- குறிமுறை சிறியதாக, எளியதாக, புரிந்து கொள்ளும் வகையில் உள்ளது.
- துணை நிரல்களாக அல்லது செயல்கூறுகளாக இருப்பதால் பிழைகளை எளிதாக கண்டு பிடிக்க இயலும்.
- ஒரே குறிமுறை பல பயன்பாடுகளில் பயன்படுத்தப்படலாம்.
- மாறிகளின் வரையெல்லையை எளிதில் கட்டுப்படுத்த முடியும்.

3.5.3 அணுகல் கட்டுப்பாடு

அணுகல் கட்டுப்பாடு என்பது கணினி சூழலில் உள்ள வளங்களை யாரெல்லாம் பார்வையிட மற்றும் பயன்படுத்த முடியும் என்பதை வரைமுறைப்படுத்தும் ஒரு பாதுகாப்பு தொழில்நுட்பமாகும். இது பாதுகாப்பின் ஒரு அடிப்படைக் கருதாகும். பொருளுக்கான ஆபத்தைக் குறைக்கிறது. அதாவது அணுகல் கட்டுப்பாடு என்பது தரவை அணுகுவதற்கான குறிப்பிடப்பட்ட கட்டுப்பாடாகும். பொருள் நோக்கு நிரலாக்க மொழியில் இது அணுகியல்பு வரையறுப்புகள் மூலம் செயல்படுத்தப்படுகிறது. C++ மற்றும் Java போன்ற பொருள் நோக்கு மொழிகள் private, protected, public என்னும் சிறப்புச் சொற்களைப் பயன்படுத்தி இனக்குழு உறுப்புகளின் அணுகலைக் கட்டுப்படுத்துகிறது. private உறுப்புகளை இனக்குழுவிற்கு வெளியே இருந்து அணுக முடியாது. இனக்குழுவிற்கு உள்ளே மட்டும்தான் கையாள முடியும்.

public உறுப்புகளை இனக்குழுவிற்கு வெளியே இருந்தும் அணுக முடியும். public வழிமுறையை செயலாக்க அந்த இனக்குழுவின் பொருள் தேவைப்படுகிறது. private தரவுகள் மற்றும் public வழி முறைகளுக்கான இந்த ஏற்பாடு உறை பொதியாக்க கொள்கையை உறுதிப்படுத்துகிறது.

protected உறுப்புகள் அந்த இனக்குழு மற்றும் அதன் துணை இனக்குழுக்களால் அணுகப்படலாம். இதை அணுகுவதற்கு வேறு எந்த செயல்முறையும் அனுமதிக்கப்படாது. இது (Child) இனக்குழுவின் மரபுரிமை பெற்ற Parent இனக்குழுவின் குறிப்பிட்ட வளங்களை செயல்படுத்துகிறது. மாறி அல்லது வழிமுறையின் அணுகலை திறம்பட கட்டுப்படுத்துவதற்கான எந்த விதமான வழிமுறையையும் பைத்தான் கொண்டிருக்கவில்லை.

பைத்தான் ஒரு மாறி அல்லது வழிமுறையின் பெயருக்கு முன்னே ஒற்றை மற்றும் இரட்டை அடிக்கோடிடும் வழக்கத்தைப் பரிந்துரைக்கிறது. இதனால் Private மற்றும் Protected அணுகியல்பு வரையறுப்பிகள் சில பண்புகளைப் பின்பற்றுகின்றன.

பைத்தானில் தானமைவாக இனக்குழுவின் அனைத்து உறுப்புகளும் public உறுப்புகளாகவும், C++ மற்றும் Java-வில் தானமைவாக private உறுப்புகளாகவும் உள்ளன. பைத்தானில் அனைத்து உறுப்புகளையும் இனக்குழுவிற்கு வெளியில் இருந்து அணுகமுடியும். ஆனால், C++ மற்றும் Java-வில் இவ்வாறு அணுக முடியாது.

Chapter 3.indd 27 30-01-2020 19:11:47

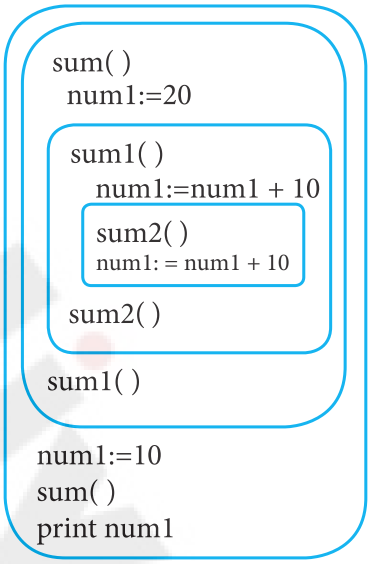
நினைவில் கொள்க

- வரையெல்லை என்பது மாறிகள், அளபுருக்கள் மற்றும் செயற்கூறுகளின் அணுகியல்பை நிரலின் ஒரு பகுதியில் இருந்து மற்றொரு பகுதிக்கு குறிப்பதாகும்.
- மாறியின் பெயரை ஒரு பொருளுடன் பிணைக்கும் செயல்முறையே மேப்பிங் (Mapping) எனப்படும். = செயல்குறி நிரலாக்க மொழியில் மாறி மற்றும் பொருளை மேப் செய்கிறது.
- namespaces என்பது மாறியின் பெயரை பொருளுடன் மேப்பிங் செய்வதற்கான இடம்.
- மாறியின் வரையெல்லை என்பது அது குறிமுறையில் எங்கு புலப்படுகிறதோ அல்லது காணப்படுகிறதோ அந்தப் பகுதியாகும்.
- LEGB விதி வரையெல்லை தேடப்பட வேண்டிய வரிசையை தீர்மானிக்கப் பயன்படுகிறது.
- உள்ளமை வரையெல்லை நடப்பு செயற்கூறில் வரையறுக்கப்பட்ட மாறிகளைக் குறிக்கும்.
- நிரலின் அனைத்து செயற்கூறுகளுக்கும் வெளியே அறிவிக்கப்பட்ட மாறிகள் முழுதளாவிய மாறிகள் எனப்படும்.
- ஒரு செயற்கூறின் உள்ளே மற்றொரு செயற்கூறு அடைக்கப்பட்டிருந்தால் அது பின்னலான செயற்கூறு எனப்படும்.
- மற்றொரு செயற்கூறு வரையறையை, தன்னுள் கொண்ட ஒரு வெளி செயற்கூறினுள் ஒரு மாறி அறிவிக்கப்பட்டால் உள்செயற்கூறானது, வெளி செயற்கூறினுள் உள்ள மாறிகளை அணுக முடியும். இதுவே, இணைக்கப்பட்ட வரையெல்லை (Enclosed Scope) எனப்படும்.
- நிரல் பெயர்ப்பி அல்லது தொகுப்பானை தொடங்கம் பொழுது உள்ளிணைந்த வரையெல்லையானது நிரல் வரையெல்லையில் ஏற்கனவே கொடுக்கப்பட்ட அனைத்து பெயர்களையும் கொண்டிருக்கும்.
- நிரலின் ஒரு பகுதியே தொகுதியாகும். நிரல்கள் ஒன்று அல்லது அதற்கு மேற்பட்ட தனித்து உருவாக்கப்பட்ட தொகுதிகளால் அமைக்கப்படுகிறது.
- ஒரு கணிப்பொறி நிரலை பல துணை நிரல்களாக பிரிக்கும் செயல்முறையே தொகுதி நிரலாக்கம் எனப்படும்.
- அணுகல் கட்டுப்பாடு என்பது கணினி சூழலில் உள்ள வளங்களை யாரெல்லாம் பார்வையிட மற்றும் பயன்படுத்த முடியும் என்பதை வரைமுறைப்படுத்தும் ஒரு பாதுகாப்பு தொழில் நுட்பமாகும். இது பாதுகாப்பின் ஒரு அடிப்படைக் கருத்தாகும். பொருளுக்கான ஆபத்தை குறைக்கிறது.
- public உறுப்புகளை இனக்குழுவிற்கு வெளியே இருந்தும் அணுக முடியும்.
- protected உறுப்புகள் அந்த இனக்குழு மற்றும் அதன் துணை இனக்குழுக்களால் அணுகப்படலாம்.
- private உறுப்புகளை இனக்குழுவிற்கு வெளியே இருந்து அணுக முடியாது. இனக்குழுவிற்கு உள்ளே மட்டும்தான் கையாள முடியும்.
- பைத்தான் ஒரு மாறி அல்லது வழிமுறையின் பெயருக்கு முன்னே ஒற்றை மற்றும் இரட்டை அடிக்கோடிடும் வழக்கத்தைப் பரிந்துரைக்கிறது. இதனால் private மற்றும் protected அணுகியல்பு வரையறுப்பிகள் சில பண்புகளைப் பின்பற்றுகின்றன.
- C++ மற்றும் Java போன்ற பொருள் நோக்கு மொழிகள் private, public, protected என்னும் சிறப்புச் சொற்களைப் பயன்படுத்தி இனக்குழு உறுப்புகளின் அணுகலைக் கட்டுப்படுத்துகிறது.
- பைத்தானில் தானமைவாக இனக்குழுவின் அனைத்து உறுப்புகளும் public உறுப்புகளாகவும், C++ மற்றும் Javaவில் தானமைவாக private உறுப்புகளாகவும் உள்ளன.

Chapter 3.indd 28 30-01-2020 19:11:47

செய்முறைப் பயிற்சி

1. கீழே கொடுக்கப்பட்டுள்ள படத்திற்கு ஏற்ற போலி குறிமுறையை எழுதுக.



KRVCK1

மதிப்பீடு

பகுதி–அ

சரியான விடையை தேர்ந்தெடுத்து எழுதுக **(1 மதிப்பெண்)**

1. பின்வருவனவற்றுள் எது நிரலின் ஒரு பகுதியின் அணுகியல்பை மற்றொரு பகுதிக்கு குறிப்பதாகும்?

   (அ) வரையெல்லை (ஆ) நினைவகம் (இ) முகவரி (ஈ) அணுகுமுறை

2. மாறியின் பெயரை ஒரு பொருளுடன் பிணைக்கும் செயல்முறையை என்னவென்று அழைக்கப்படும்?

   (அ) வரையெல்லை (ஆ) மேப்பிங் (இ) பின் பிணைத்தல் (ஈ) முன் பிணைத்தல்

3. பின்வருவனவற்றுள் எது நிரலாக்க மொழியில் மாறியையும் பொருளையும் மேப் செய்யப் பயன்படுகிறது?

   (அ) :: (ஆ) := (இ) = (ஈ) ==

4. எது மாறியின் பெயரை பொருளுடன் மேப்பிங் செய்வதற்கான இடம் ஆகும்.

   (அ) வரையெல்லை (ஆ) மேப்பிங் (இ) பிணைத்தல் (ஈ) Namespaces

5. எந்த வரையெல்லை நடப்பு செயற்கூறில் வரையறுக்கப்படும் மாறிகளைக் குறிக்கும்?

   (அ) உள்ளமை வரையெல்லை (ஆ) முழுதளாவிய வரையெல்லை

   (இ) தொகுதி வரையெல்லை (ஈ) செயற்கூறு வரையெல்லை

6. ஒரு கணிப்பொறி நிரலை பல துணை நிரல்களாக பிரிக்கும் செயல்முறையே என்னவென்று அழைக்கப்படும்.

(அ) செயல்முறை நிரலாக்கம் (ஆ) தொகுதி நிரலாக்கம்

(இ) நிகழ்வு இயக்க நிரலாக்கம் (ஈ) பொருள் நோக்கு நிரலாக்கம்

7. எது கணினி சூழலில் உள்ள வளங்களை யார் பார்வையிட மற்றும் பயன்படுத்த முடியும் என்பதை வரைமுறைப்படுத்தும் ஒரு பாதுகாப்பு தொழில்நுட்பமாகும்.

(அ) கடவுச் சொல் (ஆ) அங்கீகாரம்

(இ) அணுகல் கட்டுப்பாடு (ஈ) சான்றிதழ்

8. எந்த இனக்குழுவின் உறுப்புகளை இனக்குழுவின் உள்ளே மட்டும்தான் கையாள முடியும்.

(அ) public உறுப்புகள் (ஆ)protected உறுப்புகள்

(இ) pecured உறுப்புகள் (ஈ) private உறுப்புகள்

9. எந்த உறுப்புகளை இனக்குழுவிற்கு வெளியே இருந்தும் அணுக முடியும்?

(அ) public உறுப்புகள் (ஆ) protected உறுப்புகள்

(இ) pecured உறுப்புகள் (ஈ) private உறுப்புகள்

10. எது வரையறுக்கப்பட்ட இனக்குழு மற்றும் அதன் துணை இனக்குழுக்களால் அணுகப்படும் உறுப்புகள் ஆகும்.

(அ) public உறுப்புகள் (ஆ) protected உறுப்புகள்

(இ) pecured உறுப்புகள் (ஈ) private உறுப்புகள்

பகுதி–ஆ

அனைத்து வினாக்களுக்கும் விடையளி **(2 மதிப்பெண்)**

1. வரையெல்லை என்றால் என்ன?
2. மாறிகளுக்கு எதற்காக வரையெல்லை பயன்படுத்தப்பட வேண்டும்? காரணம் கூறுக?
3. மேப்பிங் என்றால் என்ன?
4. Namespaces சிறுகுறிப்பு வரைக?
5. private மற்றும் protected அணுகியல்புகளை பைத்தான் எவ்வாறு குறிப்பிடுகிறது.

பகுதி–இ

அனைத்து வினாக்களுக்கும் விடையளி **(3 மதிப்பெண்)**

1. உள்ளமை வரையெல்லையை எடுத்துக்காட்டுடன் விவரி?
2. முழுதளாவிய வரையெல்லையை எடுத்துக்காட்டுடன் விவரி?
3. அடைக்கப்பட்ட வரையெல்லையை எடுத்துக்காட்டுடன் விவரி?
4. அணுகல் கட்டுப்பாடு எதற்குத் தேவைப்படுகிறது?

5. பின்வரும் போலிக் (Pseudo) குறிமுறையில் மாறிகளின் வரையெல்லையைக் கண்டறிந்து வெளியீட்டை எழுதுக.

```
output
color:= Red
mycolor():
  b:=Blue
  myfavcolor():
      g:=Green
      printcolor, b, g
  myfavcolor()
  printcolor, b
mycolor()
print color
```

பகுதி ஈ

அனைத்து வினாக்களுக்கும் விடையளி **(5 மதிப்பெண்)**

1 மாறியின் வரையெல்லைகளின் வகைகளை விளக்குக (அல்லது) LEGB விதியை எடுத்துக்காட்டுடன் விளக்குக?

2. தொகுதிகளின் ஐந்து பண்பியல்புகளை எழுதுக?

3. தொகுதி நிரலாக்கத்தின் பயன்களை எழுதுக?

REFERENCES

1. *Data Structures and Algorithms in Python By Michael T.Goodrich, Roberto Tamassia and Michael H. Goldwasser.*

2. *Data Structure and Algorithmic Thinking in Python By Narasimha Karumanchi*

3. *https://www.python.org*

Chapter 3.indd 31 30-01-2020 19:11:47

அலகு I

பாடம் 4

நெறிமுறையின் யுக்திகள்

VYMN69

கற்றலின் நோக்கங்கள்

இந்தப் பாடப்பகுதியைக் கற்றபின் மாணவர்கள் அறிந்துக் கொள்வது,

- நெறிமுறையின் அடிப்படை மற்றும் தொழில்நுட்ப தொலைநோக்குப் பற்றி அறிதல்.
- நெறிமுறையின் செயல்திறன், நேரம் மற்றும் இடசிக்கல் ஆகியவற்றைப் பற்றிப் புரிந்து கொள்ளுதல்.
- தேட மற்றும் வரிசைப்படுத்துவதற்கான நெறிமுறைகளை உருவாக்குதல் மற்றும் ஆய்வு செய்தல்.
- நெறிமுறை அணுகுமுறை மூலம் மாறும் நிரலாக்கத்தைக் கற்றல்.

4.1 நிரல் நெறிமுறையின் யுக்திகள் –ஓர் அறிமுகம்

ஒரு குறிப்பிட்ட செயலை நிறைவேற்றுவதற்கான வரையறுக்கப்பட்ட கட்டளைகளின் தொகுப்பு நெறிமுறையாகும். கொடுக்கப்பட்ட சிக்கலைத் தீர்க்கும் படிநிலை உடைய செய்முறை ஆகும். நெறிமுறையை எந்தவொரு பொருத்தமான நிரலாக்க மொழியிலும் செயல்படுத்த முடியும்.

நிரல் நெறிமுறையில் உள்ளீடு, வெளியீடு இருக்க வேண்டும். மேலும் பின்வரும் பண்புகளையும் பூர்த்தி செய்ய வேண்டும். வரையறுக்கும் தன்மை, உண்மை தன்மை மற்றும் செயல்திறத் தன்மை ஆகியவை தரவு கட்டமைப்புகள் மூலம் தரவு நிர்வகிக்கப்படுகிறது மற்றும் திறமையாக கையாளப்படுகிறது.

இத்தகைய தரவு கட்டமைப்புகளிலிருந்து தரவைச் சேமித்து, கையாளவும் மற்றும் மீட்டெடுக்கவும் நெறிமுறைகளை உருவாக்கலாம். தரவு கட்டமைப்புக்கு எடுத்துக்காட்டு: அணிகள், கட்டுருக்கள், List, tuples, dictionary இன்னும் பிற நிரல் நெறிமுறையைப் பயன்படுத்தி திறமையாக குறிக்கக்கூடிய சில முக்கிய செயல்கள் பின்வருமாறு

தேடல்	வரிசைமுறைத் தேடல் மற்றும் இருமத்தேடல் மூலம் தரவு கட்டமைப்பில் உள்ள ஒரு உருப்படியைத் தேட.
வரிசையாக்கம்	ஒரு குறிப்பிட்ட முறையில் வழிமுறைகளைப் பயன்படுத்தி உருப்படிகளை வரிசைபடுத்த முறைகளாக குமிழி வரிசையாக்கம் செருகும் வரிசையாக்கம், தேர்ந்தெடுப்பு வரிசைபாக்கம் போன்றவை
செருகுதல்	தரவு கட்டமைப்பில் ஒரு உருப்படியைச் (களை) செருகுவதற்கு.
மேம்படுத்துதல்	தரவு கட்டமைப்பில் தற்பொழுது உள்ள ஒரு உருப்படியை மேம்படுத்த.
நீக்குதல்	தரவு கட்டமைப்பில் தற்பொழுதுள்ள ஒரு உருபடி (களை) நீக்க.

ஒரு நெறிமுறையை வரையறுக்கும் முறையை நெறிமுறை யுக்தி என்கிறோம். கொடுக்கப்பட்ட n-ன் தொடர் பெருக்கலைக் காணும் உதாரணத்தை எடுத்துக் கொள்வோம். தொடர்பெருக்கலைக்கணக்கிடும்செயற்கூறினை வரையறுப்பதன் மூலம் இதைச் செய்ய முடியும். பிறகு, தேவைப்படும் சுழற்சியின் எண்ணிக்கை

Chapter 4.indd 32 30-01-2020 19:13:38

வரும் வரை, அந்த செயற்கூறை தற்சுழற்சி செயற்கூறாக அழைக்க வேண்டும்.

உங்களுக்குத் தெரியுமா?

நெறிமுறை என்ற சொல், கணித பாடப்புத்தகத்தை எழுதிய பெர்சியன் எழுத்தாளர் Abu Jafar Mohammed ibn Musa al Khowarizmi (c. 825 AD(CE)), இருந்து பெறப்பட்டது. நெறிமுறை என்ற சொல்லுக்கு ஒரு சிக்கலைத் தீர்க்கும் வழி என்பதைக் குறிப்பதாகும்.

4.1.1 நெறிமுறையின் பண்புகள்

பின்வரும் பண்புகளை நெறிமுறைகள் கொண்டிருக்க வேண்டும்.

உள்ளீடு	பூஜ்ஜியம் அல்லது அதிக எண்ணிக்கையில் வழங்கப்பட வேண்டும்.
வெளியீடு	குறைந்தபட்சம் ஒன்றாவது உருவாக்கப்பட வேண்டும்.
எல்லையுற்றது	வரையறுக்கப்பட்ட எண்ணிக்கையிலான படிநிலைகளில் நெறிமுறை நிறுத்தப்பட வேண்டும்.
வரையறுத்தல்	அனைத்து செயல்பாடுகளும் நன்றாக வரையறுக்கப்பட் வேண்டும். எடுத்துக்காட்டாக, பூஜ்ஜியம் மூலம் வகுத்தல் அல்லது எதிர்ம எண்ணுக்கு வர்க்க மூலத்தை கணக்கிடுதல் ஆகிய செயல்பாடுகள் ஏற்றுக் கொள்ள முடியாதது ஆகும்.
செயல்தன்மை	ஒவ்வொரு கட்டளைகளும் திறம்பட நடத்தப்பட வேண்டும்.
உண்மைத் தன்மை	நெறிமுறைகள் பிழை இல்லாததாக இருக்க வேண்டும்.
எளிமை	செயல்படுத்துவதற்கு மிக எளிதாக இருக்க வேண்டும்.
குழப்பமற்றது	நெறிமுறையானது தெளிவாகவும் குழப்பமற்றதாகவும் இருத்தல் வேண்டும்.அதன் ஒவ்வொரு படிநிலைகளும் மற்றும் அதனுடைய உள்ளீடுகள்/ வெளியீடுகள் தெளிவானதாக இருக்க வேண்டும் மற்றும் ஒரே ஒரு பொருளுக்கு வழிவகுக்க வேண்டும்.
செயலாக்கம்	கிடைக்கும் வளங்களை வைத்து செயலாக்க வல்லது.
அடக்கமானது	நெறிமுறை பொதுவானதாக இருக்க வேண்டும். அனைத்து வகையான உள்ளீடுகளையும் கையாள்வதற்கு எந்த நிரலாக்க மொழியையும் மற்றும் இயக்க அமைப்பையும் சாராமல் இருக்க வேண்டும்.
சார்பற்றது	நெறிமுறையானது படிநிலை வழிமுறைகளைக் கொண்டிருக்க வேண்டும். அது எந்த நிரலாக்க குறிமுறையை சாராமல் இருக்க வேண்டும்.

4.1.2 நெறிமுறையை எழுதுதல்

நெறிமுறை கணினிக்கு மட்டுமே ஒட்டப்பட்டதல்ல, பொதுவானது. நிகழ் உலக பல வகையான செயல்பாடுகளுக்கும் இது பயன்படும். தெரிந்தோ, தெரியாமலோ நாம் அன்றாட வாழ்வில் பல நெறிமுறைகளை பின்பற்றுகிறோம். அதாவது, புத்தகப்பையில் புத்தகங்களை அடுக்குவது, ஒரு இடத்தை அடைவதற்காக மிகக் குறைந்த தூரம் உள்ள பாதையைக் கண்டறிவது, அன்றாட செயல்களைத் திட்டமிடுதல், தேர்வுக்குத் தயாராகுதல் மற்றும் பல, அனைத்து நிரலாக்க மொழிகளும் அடிப்படை குறிமுறை கட்டமைப்பை பகிர்கின்றன என்பதை நாம் அறிவோம். அதாவது, நிபந்தனைகள், சுழற்சிகள் ஆகியவற்றைப் பயன்படுத்தி நெறிமுறைகளை எழுதலாம்.

முறையான நெறிமுறை பின்வரும் படம் 4.1 ல் காண்பிக்கப்பட்டுள்ளது. பின்வரும் படம் 4.1 ல் காண்பிக்கப்பட்டுள்ளது.

படம் *4.1* பொதுவான நிரல்நெறிமுறை

எடுத்துக்காட்டு

காபி தயாரிக்கும் முறையை எடுத்துக்காட்டாக நோக்குக. காபியை தயாரிப்பதற்கு பின்வரும் பொருள்கள் நமக்கு தேவைப்படுகின்றன. தண்ணீர், பால் காபித்தூள், சர்க்கரை. இந்த பொருள்களானது நெறிமுறையின் உள்ளீடுகளாகும். ஒரு டம்ளர் காபி தயாரித்தல் என்பது செயல் (process) ஆகும். இதன் வெளியீடு காபி ஆகும்.

காபி தயாரித்தலின் செயல்முறை (procedure) பின்வருமாறு:

1. ஒரு கிண்ணத்தில் காபித்தூளை எடுத்துக் கொள்ளவும்
2. தண்ணீரை சூடாக்கி காபித்தூள் உள்ள கிண்ணத்தில் ஊற்றவும்
3. அதை வடி கட்டவும்
4. பாலை சூடாக்கவும்
5. சூடாக்கப்பட்ட பாலுடன் சர்க்கரை மற்றும் வடிகட்டிய காபியை ஊற்றி கலக்கவும்
6. காபியை டம்ளரில் ஊற்றி வழங்கவும்

இந்த வகையான செயமுறையை நெறிமுறை மூலம் உருவாக்கலாம். எனவே, நெறிமுறையானது, ஒரு செயலை செய்து முடிக்கத் தேவையான படிப்படியான வழிமுறைகளை கொண்டிருக்கிறது. நிரலருக்கு நிரலை உருவாக்க இது உதவுகிறது.

பயிற்சி: ஒரு எண்ணின் வர்க்கம் (square) கண்டுபிடித்து அதன் விடையை காண்பிப்பதற்கான நிரல் நெறிமுறையை எழுதுக.

நிரல் நெறிமுறையை பின்வருமாறு காணலாம்.

படிநிலை 1 – செயலை தொடங்கவும் (start the process)

படிநிலை 2 – x-ன் மதிப்பை உள்ளிடவும் (get the input x)

படிநிலை 3 – x- ன் வர்கத்தை கணக்கிட, உள்ளிட்ட மதிப்பை பெருக்கவும் அதாவது, (calculate the square by multiplying the input value ie., square ← x* x)

படிநிலை 4 – வெளியீட்டை அறிவிக்கவும் (display the result square)

படிநிலை 5 – நிறுத்து (stop)

நெறிமுறையானது கொடுக்கப்பட்ட சிக்கலுக்கு தீர்வுகாண வடிவமைக்கப்படுகிறது. ஒரு சிக்கலானது ஒன்றிற்கு மேற்பட்ட வழிகளில் தீர்வு காணப்படலாம். ஒரு சிக்கலுக்கு பல நெறிமுறைகளை உருவாக்கலாம். ஆனால், மிகவும் பொருத்தமான ஒன்றை செயல்பாட்டிற்கு எடுத்துக் கொள்ள வேண்டும்.

உங்களுக்குத் தெரியுமா?

தகுதி வாய்ந்த உள்ளீட்டிற்கு எதிர்பார்த்த வெளியீட்டை கொடுக்கும் நெறிமுறையே "**நிரல் நெறிமுறை தீர்வு** " எனப்படும்.

4.1.3. நெறிமுறைக்கும், நிரலுக்கும் உள்ள வேறுபாடு

நெறிமுறை	நிரல்
சிக்கலை தர்க்கரீதியாக தீர்க்க உதவுகிறது மற்றும் நிரலுடன் வேறுபட்டிருக்கலாம்.	நிரல் என்பது நிரலாக்கமொழியில் நெறிமுறையின் வெளிப்பாடாகும்.

Chapter 4.indd 34 30-01-2020 19:13:39

செயல்படுத்தப்படும் முறை மற்றும் வடிவமைப்பு நுட்பம் ஆகியவற்றை பொறுத்து இதனை வகைப்படுத்தலாம்.	கட்டமைப்பு நிரல் அணுகுமுறை அல்லது பொருள் நோக்கு நிரலாக்க அணுகுமுறையை பயன்படுத்தி இதனை செயல்படுத்த முடியும்.
நெறிமுறைக்கு குறிப்பிட்ட விதிமுறைகள் கிடையாது ஆனால் சில வழிகாட்டுதல்களை பின்பற்ற வேண்டும்.	தேர்ந்தெடுகக்பபட்ட மொழியில் குறிப்பிட்ட கட்டளை அமைப்புகளைக் கொண்டு நிரலை எழுதலாம்.
போலி குறிமுறையை ஒத்திருக்கும் இதை எந்த கணிப்பொறி மொழியில் வேண்டுமானாலும் செயல்படுத்தலாம்	நிரலாக்க மொழிக்கு மிகவும் குறிப்பிடத் தாக்கதாக நிரல் இருக்க வேண்டும்.

அட்டவணை *4.1* நெறிமுறை *Vs.*நிரல்

4.1.4. நெறிமுறை பகுப்பாய்வு

கணினி வளங்கள் வரம்புக்குட்பட்டது. நேரம்மற்றும்இடச்சிக்கல்களைபயன்படுத்துவதன் மூலம் நிரல் நெறிமுறையின் செயல்திறன் வரையறுக்கப்படுகிறது.

சம்பந்தப்பட்ட பல்வேறு செயல்பாடுகளுடைய இயக்க நேரத்தோடு நெறிமுறை பகுப்பாய்வு தொடர்புடையது. ஒவ்வொரு செயல்பாட்டிற்காக செயல்படுத்தப்படும் கணினி கட்டளைகளின் எண்ணிக்கை மூலம் செயல்பாட்டின் இயங்கு நேரம் வரையறுக்கப்படுகிறது.

நெறிமுறை பகுப்பாய்வு மற்றும் செயல்திறன் மதிப்பீடுகளை இரண்டு கட்டங்களாக பிரிக்கலாம்.

1. **முன்னறிப்பு (Priori Estimates) மதிப்பீடுகள்:** இது நெறிமுறையின் கோட்பாட்டு செயல்திறன் பகுப்பாய்வு ஆகும். வெளிப்புற காரணிகளை ஏற்றுக் கொண்டு நெறிமுறையின் செயல் திறன் அளக்கப்படுகிறது.

2. **பின்னிய (Posteriori testing) சோதனை:** இது செயல்திறன் அளவிடுதல் ஆகும். இந்த பகுப்பாய்வில் உண்மையான புள்ளி விவரங்களான இயங்கும் நேரம் மற்றும் நிரல் நெறிமுறை நிறைவேற்ற தேவைப்படும் இடம் ஆகியவை சேகரிக்கப்படுகிறது.

? உங்களுக்குத் தெரியுமா?

ஒரு நெறிமுறையின் பல்வேறு உள்ளீட்டு அளவுக்கான நேர மற்றும் இட சிக்கல்களின் மதிப்பிடுவது நெறிமுறை பகுப்பாய்வு ஆகும்.

4.2 நெறிமுறையின் சிக்கல்

எடுத்துக்காட்டாக, A நெறிமுறை, n உள்ளீட்டு தரவின் அளவாக, இருப்பின் A ன் செயல்திறனை இரண்டு முக்கிய காரணிகளான (time) நேரம் மற்றும் இடம் ஆகியவை தீர்மானிக்கிறது.

நேரம் காரணி - நெறிமுறைக்கு பொருத்தக் கூடிய முக்கிய செயல்பாடுகளின் எண்ணிக்கையை எண்ணுவதன் மூலம் நேரம் அளவிடப்படுகிறது, வரிசையாக்கம் நெறி முறையிலுள்ள பொருத்தங்களின் எண்ணிக்கை

இடகாரணி (space) - நெறிமுறைக்கு தேவைப்படும் மிக அதிகபட்ச நினைவக இடத்தை கணக்கிடுவதன் மூலம் இது அளவிடப்படுகிறது. நெறிமுறை f(n)-ன் சிக்கலானது, அது n அளவிலான உள்ளீட்டு தரவை எடுத்துக்கொண்டு இயங்கும் நேரம் மற்றும் நினைவகத்தில் அதற்கு தேவைப்படும் இட ஒதுக்கீடு பொறுத்தது.

4.2.1 நேரசிக்கல் (Time complexcity)

ஒரு நெறிமுறை செயலை செய்து முடிக்க எண்ணிக்கையே நெறிமுறையின் நேரசிக்கல் எனப்படும்.

4.2.2. இடசிக்கல் (space)

ஒரு நெறிமுறையின் செயல்பாடு முடியும்வரை அதற்கு தேவைப்படும் நினைவக இடமே இடச்சிக்கல் எனப்படும். நெறிமுறைக்கு தேவைப்படும் இடம் பின்வரும் இரு கூறுகளின் கூட்டுத் தொகையாகும்.

நிலையான பகுதி இது நெறிமுறைக்கு தேவையான தரவு மற்றும் மாறிகளை சேமிக்க பயன்படும் கூட்டு இடத்தை வரையறுக்கும். எடுத்துக்காட்டு நிரல் நெறிமுறையில் பயன்படுத்தப்படும் மாறிகள் மற்றும் மாறிலிகள்

மாறும் பகுதி சிக்கலின் அளவு மற்றும் சுழற்சிக்கு தேவைபடும் அனைத்து மாறிகளின் கூட்ட இடத்தின் அளவை பொறுத்து 'n' வரையறுக்கப்படும். எடுத்துக்காட்டாக: என்ற மதிப்பின் தொடர் பெருக்களை தற்சுழற்சி மூலம் கண்டறிதல்.

4.3 நெறிமுறையின் செயல்திறன்

கணினி வளங்கள் எல்லைக்குட்பட்டதால் திறம்பட அதனை பயன்படுத்த வேண்டும். நெறிமுறையின் செயல்திறன் என்பது நெறிமுறை பயன்படுத்தும் கணக்கீட்டு (computational) வளங்களின் எண்ணிக்கையை வரையறுக்கும். நெறிமுறையின், பகுப்பாய்வு அதனுடைய வளங்களின் பயன்பாட்டைத் பொருத்து தீர்மானிக்கப்படுகிறது. பல்வேறு வளங்களின் பயன்பாட்டின் அடிப்படையில் நெறிமுறையின் செயல்திறனை அளவிட வேண்டும்.

நெறிமுறையின் அதிகபட்ச செயல்திறனுக்காக நாம் வளங்களின் பயன்பாட்டைக் குறைக்க விரும்புகிறேம். எனினும், வெவ்வேறு வளங்களான நேரம் மற்றும் இடச்சிக்கல் ஆகியவற்றை நேரடியாக ஒப்பிட முடியாது. இருப்பினும், இவை, இரண்டும் நெறிமுறையின் செயல்திறனுக்கு கருதப்பட வேண்டும்.

4.3.1 செயல்திறனைத் தீர்மானிக்கம் வழிமுறை (Method for determining Efficiency)

நெறிமுறையின் செயல்திறன், நேரத்தையும், நினைவக இருப்பிடத்தையும் எவ்வாறு அது திறமையாகப் பயன்படுத்துகிறது என்பதைத் பொறுத்ததாகும்.

நெறிமுறையின் நேர செயல்திறனை அளவிடுவதற்கு பல்வேறு காரணிகள் உள்ளன. எடுத்துக்காட்டாக ஒரு நிரலாக்க மொழியின்அடிப்படையில் நிரலை எழுதி, அதனைச் செயல்படுத்தி, அது இயக்குவதற்கு எடுத்துக் கொள்ளும் நேரத்தை அளவிடவும். இங்கு, அளவிடப்படும் இயங்கு நேரம் பின்வரும் காரணிகளைப் பொறுத்ததாகும்:

- இயந்திரத்தின் வேகம்
- நிரல்பெயர்ப்பி மற்றும் பிற கணினி மென்பொருள் கருவிகள்
- இயக்க அமைப்பு
- பயன்படுத்தப்பட்ட நிரலாக்க மொழி
- தரவு தொகுதி.

ஒரு நெறிமுறை கொடுக்கப்பட்ட ஒரு சிக்கலை எவ்வாறு திறனுடன் தீர்க்கிறது என்பதை கண்டறிய அதன் இயக்க நேரம் எவ்வாறு அந்நெறிமுறையின் இயல்பை பாதிக்கிறது என்பதை கண்டறிய வேண்டும். எனவே, நெறிமுறையின் இயல்பில் ஒரு நிரலின் செயல்திறனை தீர்மானிக்கும் அடிப்படை விதிகளை வகுக்கவேண்டும்.

> **? உங்களுக்குத் தெரியுமா?**
>
> நெறிமுறையை வடிவமைக்கும் வழிமுறையே நெறிமுறை யுக்தி ஆகும்.

4.3.2 இடம் நேரம் பரிமாற்றம்

இடம் நேரம் பரிமாற்றம் அல்லது நேரம் நினைவகம் பரிமாற்றம் என்பது குறைந்த நேரத்தில் அதிகமான சேமிக்கும் இடத்தைப் பயன்படுத்தி நெறிமுறையை தீர்க்கும் வழியாகும்.

Chapter 4.indd 36 30-01-2020 19:13:39

கொடுக்கப்பட்ட நிரலாக்கத்தின் சிக்கலைத் தீர்ப்பதற்கு பல விதமான நெறிமுறைகளைப் பயன்படுத்தலாம். அதில் சில நெறிமுறைகள் மிகவும் நேர செயல்திறன் உடையதாகவும் மற்றும் சில நெறிமுறைகள் மிகவும் இட செயல்திறன் உடையதாகவும் இருக்கும்.

நேரம் / இடம் பரிமாற்றம் என்பது மெதுவான நிரல் இயங்கு நேரத்தில் நினைவகப் பயன்பாட்டைக் குறைக்கும் நிலைமையை அல்லது அதிகமான நினைவக பயன்பாட்டில் இயங்கு நேரத்தைக் குறைக்கும் நிலைமையைக் குறிப்பதாகும்.

? உங்களுக்குத் தெரியுமா?

நினைவகத்தில் தேவைப்படும் குறைந்த இடத்தையும் மற்றும் அதன் கட்டளைகளை இயக்குவதற்கு குறைந்த நேரத்தையும் எடுத்துக்கொண்டு வெளியீட்டினை வெளியிடும் நெறிமுறையே கொடுக்கப்பட்ட சிக்கலைத் தீர்க்கும் சிறந்த நெறிமுறையாகும்.

4.3.3 Asymptotic குறியீடுகள்

இந்த Asymptotic குறியீடுகள் நேரம் மற்றும் இடச்சிக்கலைகளைப்பற்றிய அர்த்தமுள்ள கூற்றுகளைப் பயன்படுத்தும் ஒரு மொழியாகும். பின்வரும் மூன்று Asymptotic குறியீடுகள் நெறிமுறையில் நேரச் சிக்கலைக் குறிக்கும் மிகவும் பயன்படுகிறது.

4.3.3.1 Big O

நெறிமுறையின் மிக மோசமான நிலையை விவரிக்க Big O பெரும்பாலும் பயன்படுத்தப்படுகிறது.

4.3.3.2 Big Ω

Big Omega, Big O-வின் தலைகீழ் ஆகும். Big O-asymptotic (மோசமான நிலையில்) செயற்கூறின் உச்ச வரம்பையும், Big Omega அதன் கீழ்வரையை குறிக்கும் (சிறந்த நிலையில்).

4.3.3.3 Big Θ

நெறிமுறையானது கீழ் எல்லை = மேல் எல்லை என்னும் சிக்கலைக் கொண்டிருந்தால், உதாரணத்திற்கு O(n log n) மற்றும் Ω(n log n), ஆகிய சிக்கல்களைக் கொண்டுள்ளது என வைத்துக் கொள்வோம். உண்மையில் அதனுடைய சிக்கல் Θ(n log n), என்பது ஆகும். இதனுடைய அர்த்தம் என்னவென்றால் நெறிமுறையின் இயங்கு நேரம் மிகச் சிறந்த நிலை மற்றும் மிக மோசமான நிலை ஆகிய இரண்டு நிலையிலுமே எப்பொழுதும் n log n ஆக இருக்கும்.

4.3.4 சிறந்த, மோசமான மற்றும் சராசரி நிலைகளின் செயல்திறன்

எடுத்துக்காட்டாக, ஒரு அணியில் n-மதிப்புகள் உள்ளதாக வைத்துக் கொள்வோம். இந்த பட்டியலில் இருந்து ஒரு குறிப்பிட்ட உறுப்பைத் தேட வேண்டும். நெறிமுறையானது n-உறுப்புகளைக் கொண்ட பட்டியலில் இருந்து தேட வேண்டிய பட்டியலில் உள்ள ஒவ்வொரு உறுப்புகளோடு வரிசையாக ஒப்பீடு செய்து உருப்புகளை பட்டியலில் ஒவ்வொன்றாக அதன் முதன்மை உறுப்பை கொண்டு ஒப்பிட வேண்டும்.

இந்த எடுத்துக்காட்டில், பட்டியலில் உள்ள முதல் உறுப்பு தேடப்படும் உறுப்போடு பொருத்தமாக இருந்தால் அது தான் மிகச் சிறந்த நிலையாகும். இந்த நிலையின் செயல்திறனை O(1) என்று வெளிப்படுத்தலாம். ஏனெனில், ஒரே ஒரு ஒப்பீடுதான் தேவைப்படுகிறது.

அதைப்போல, இந்த எடுத்துக்காட்டில், பட்டியல் முழுவதற்கும் சென்று தேடினால், அந்த உறுப்பு பட்டியலின் இறுதியில் கண்டுபிடிக்கப்பட்டால் அல்லது பட்டியலில் இல்லாமல் இருந்தால் அதுதான் மோசமான நிலையாகும். இந்த நிலையின் செயல்திறனை O(n) என்று வெளிப்படுத்தலாம். ஏனென்றால், n ஒப்பீடுகள் செய்யப்படுகின்றன.

அதே எடுத்துக்காட்டுடன் தொடர்ந்தால் சராசரி ஒப்பீடுகளின் எண்ணிக்கையைக் கண்டுபிடிப்பதன் மூலம் சராசரி நிலையின் செயல்திறனைப் பெறலாம். இங்கு,

குறைந்தபட்ச ஒப்பீடுகளின் எண்ணிக்கை = 1

அதிகப்பட்ச ஒப்பீடுகளின் எண்ணிக்கை = n

கண்டுபிடிக்கப்படாத உறுப்பின் அதிகபட்ச ஒப்பீடுகளின் எண்ணிக்கை = n

எனவே, சராசரி ஒப்பீடுகளின் எண்ணிக்கை = (n + 1)/2

ஆகையால், சராசரி நிலையின் செயல்திறனை O(n) என்று வெளிப்படுத்தலாம்.

4.4 தேடல் முறைகளுக்கான நெறிமுறை

4.4.1 வரிசைமுறைத் தேடல்

வரிசைமுறைத் தேடல் அல்லது தொடர் தேடல் பட்டியலில் விரும்பும் உறுப்பைக் கண்டுபிடிக்கும் வரை அல்லது பட்டியல் முடியும் வரை வரிசையிலுள்ள ஒவ்வொரு உறுப்புகளையும் சரிபார்த்து, குறிப்பிட்ட மதிப்பைப் பட்டியலில் கண்டுபிடிக்கும் வழிமுறையாகும். பட்டியலை வரிசைப்படுத்த வேண்டிய தேவை இல்லை.

போலி குறிமுறை

1. for மடக்கினைப் பயன்படுத்தி அணியில் பயணித்தல்.
2. ஒவ்வொரு சுழற்சியிலும், இலக்கு மதிப்பை தற்போதைய மதிப்புடன் ஒப்பிடவும்.
   - o மதிப்புகள் பொருத்தமாக இருந்தால் அணியின் தற்போதைய சுட்டெண்ணைத் திருப்பி அனுப்பும்.
   - o மதிப்புகள் பொருந்தாவிட்டால் அணியில் அடுத்துள்ள உறுப்புக்கு சென்று விடும்.
3. பொருத்தம் எதுவும் இல்லையென்றால், -1 மதிப்பைத் திருப்பி அனுப்பும்.

கீழே கொடுக்கப்பட்டுள்ள அணியில் 25 என்ற எண்ணைத் தேடுவதற்கு, வரிசைமுறைத் தேடலானது படிப்படியாக தொடர் வரிசையில் கொடுக்கப்பட்ட அணியின் முதல் உறுப்பிலிருந்து தேடலைத் தொடங்கும். தேடப்படும் உறுப்பு கண்டுப்பிடிக்கப்பட்டால் அந்த சுட்டெண் திரும்ப அனுப்பப்படும். இல்லையெனில், அணியின் இறுதி சுட்டெண் வரை தேடல் தொடரும். இந்த எடுத்துக்காட்டில், எண் 25 சுட்டெண் 3-ல் காணப்படுகிறது.

அணியின் சுட்டு	0	1	2	3	4
மதிப்புகள்	10	12	20	25	30

எடுத்துக்காட்டு 1:

Input: values[] = {5, 34, 65, 12, 77, 35}
target = 77
Output: 4

எடுத்துக்காட்டு 2:

Input: values[] = {101, 392, 1, 54, 32, 22, 90, 93}
target = 200
Output: -1 (not found)

4.4.3. இருமத் தேடல்

இருமத் தேடலை பாதி இடைவெளித் தேடல் நெறிமுறை என்றும் அழைக்கலாம். வரிசைப்படுத்தப்பட்ட அணிக்குள் இலக்கு மதிப்பின் இருப்பிடத்தைக் கண்டுபிடிக்கிறது. பிரித்து-கைப்பற்றுதல் நெறிமுறையைப் போல் இருமத் தேடலைச் செய்து மடக்கை நேரத்தில் நிறைவேற்றப்படும்.

Chapter 4.indd 38 30-01-2020 19:13:39

இருமத் தேடலுக்கான போலி குறிமுறை

1. மைய உறுப்பிலிருந்து தொடங்கவும் :

   o இலக்கு மதிப்பும் அணியின் மைய உறுப்பும் நிகர் எனில் (i.e. அதாவது, மைய இலக்கு = உறுப்புகளின் எண்ணிக்கை /2) மைய உறுப்பின் சுட்டெண்ணைத் திருப்பி அனுப்பும்.

   o நிகரில்லை என்றால், மைய உறுப்பை மதிப்போடு ஒப்பிடவும்,

   o மைய சுட்டெண்ணிலுள்ள எண் இலக்கு மதிப்பை விட பெரியது எனில், மைய சுட்டெண்ணிற்கு வலப்புறம் உள்ள உறுப்புகளைத் தேர்ந்தெடுத்து படிநிலை -1லிருந்து தொடங்கவும்.

   o மைய சுட்டெண்ணிலுள்ள எண் இலக்கு மதிப்பை விட சிறியது எனில் மைய சுட்டெண்ணிற்கு இடப்புறம் உள்ள உறுப்புகளைத் தேர்ந்தெடுத்து படிநிலை -1 லிருந்து தொடங்கவும்.

2. பொருத்தமான தேடல் கண்டுபிடிக்கப்பட்டால், பொருந்திய உறுப்பின் சுட்டெண்ணைத் திருப்பி அனுப்பும்.

3. பொருத்தம் இல்லையெனில், -1 என்ற மதிப்பைத் திருப்பி அனுப்பும் அல்லது தேடல் நிறைவேற்றப்படவில்லை என்ற தகவலை அறிவிக்கவும்.

இருமத் தேடல் இயங்கும் கோட்பாடுகள்

இருமத்தேடலில் பயன்படும் அணி வரிசையாக்கம் செய்யப்பட்ட அணியாகயிருக்க வேண்டும். இருமத் தேடலைப் பயன்படுத்தி மதிப்பு 60-ன் இருப்பிடத்தைத் தேடுவதாக எடுத்துக்கொள்வோம்.

10	20	30	40	50	60	70	80	90	99
0	1	2	3	4	5	6	7	8	9

முதலில் நாம் அணியின் மைய உறுப்பை mid = low + (high - low) / 2 என்ற வாய்ப்பாட்டைப் பயன்படுத்தித் தீர்மானிக்க வேண்டும்.

இங்கு, 0 + (9 - 0) / 2 = 4 (4.5 யின் முழு மதிப்பு எடுத்துக்கொள்ளவும்). அதனால் அணியின் மையம் 4 ஆகும்.

10	20	30	40	50	60	70	80	90	99
0	1	2	3	4	5	6	7	8	9

இப்பொழுது நாம் 4-ம் சுட்டெண் இருப்பிடத்தில் சேமிக்கப்பட்ட மதிப்போடு தேடப்படும் மதிப்பை (அதாவது, 60) ஒப்பிடு செய்வோம். 4-ம் சுட்டெண் இருப்பிடத்தில் உள்ள மதிப்பான 50 என்பது, இது தேடப்படும் மதிப்பு கிடையாது. தேடப்படும் மதிப்பானது 50-விட அதிகமாக இருப்பதால்

10	20	30	40	50	60	70	80	90	99
0	1	2	3	4	5	6	7	8	9

low மதிப்பை m + 1 என மாற்றி புதிய mid மதிப்பை மறுபடியும் கண்டுபிடிக்க வேண்டும்.

low = mid + 1

mid = low + (high - low) / 2

இப்பொழுது நமது mid மதிப்பு 7 ஆகும். நாம் இருப்பிடம் 7-ல் சேமிக்கப்பட்ட மதிப்பை இலக்கு மதிப்போடு (அதாவது 60) ஒப்பிடுவோம்.

10	20	30	40	50	60	70	80	90	99
0	1	2	3	4	5	6	7	8	9

இருப்பிடம் 7ல் சேமிக்கப்பட்ட மதிப்பு தேடப்படும் மதிப்பு கிடையாது. மாறாக, நாம் தேடுவதை விட அதிகமான மதிப்பாக இருக்கிறது. எனவே, தேடப்படும் மதிப்பு இந்த இருப்பிடத்தை விட குறைவான பகுதியில் இருக்க வேண்டும்.

10	20	30	40	50	60	70	80	90	99
0	1	2	3	4	5	6	7	8	9

எனவே, நாம் mid மதிப்பைத் திரும்பவும் கணக்கீடு செய்ய வேண்டும்.

high = mid -1

mid = low + (high - low)/2

தற்பொழுது midமதிப்பு 5 ஆகும்.

↓

10	20	30	40	50	60	70	80	90	99
0	1	2	3	4	5	6	7	8	9

நாம் இருப்பிடம் 5-ல் உள்ள சேமிக்கப்பட்ட மதிப்பை இலக்கு மதிப்போடு ஒப்பீடு செய்வோம். இது ஒரு சரியான பொருத்தமாகும்.

10	20	30	40	50	60	70	80	90	99
0	1	2	3	4	5	6	7	8	9

இலக்கு மதிப்பு 60, இருப்பிடம் 5-ல் சேமிக்கப்பட்டுள்ளது என்ற நாம் முடிவு செய்கிறோம். எடுத்துக்காட்டாக, இலக்கு மதிப்பு 95 எனில், இந்த செயல்முறை -1 என்ற மதிப்பைத் திருப்பி அனுப்பும்.

4.5 வரிசையாக்க முறைகள்

4.5.1 குமிழி வரிசையாக்க நெறிமுறை (Bubble Sort Algorithm)

குமிழி வரிசையாக்கம் ஒரு எளிமையான வரிசையாக்க நெறிமுறை ஆகும். வரிசைப் படுத்தப்பட்ட பட்டியலின் படிநிலைகளை மீண்டும் மீண்டும் செய்து, ஒவ்வொரு ஜோடி அருகிலுள்ள உருப்படிகளை ஒப்பீடு செய்து, வரிசையாக்கம் செய்யப்படாத வரிசை எனில் அவற்றை இடமாற்றம் செய்யும். இடமாற்றம் தேவைப்படும் வரை அவை மீண்டும் மீண்டும் பட்டியலிடப்படும். இது பட்டியல் வரிசையாக்கம் செய்யப்பட்டுள்ளது என்பதைக் குறிக்கும். இந்த ஒப்பீட்டு வரிசையாக்கம் நெறிமுறையில் பட்டியலின் மேல் பகுதியில் குமிழியைப் போல் சிறிய உறுப்புகளை அமைக்கும் முறையினால் இதற்கு இந்த பெயரிடப்பட்டது. இந்த நெறிமுறை எளிமையானதாக இருந்த போதிலும், இது மிகவும் மெதுவானது மற்றும் செருகும் வரிசையாக்கத்தோடு (insertion sort) ஒப்பீடு செய்யும் போது இது சாத்தியமற்றதாகும்.

n உறுப்புகளை கொண்ட அணியை கருதிக்கொள்ளவும் இடமாற்ற செயல்முறை (swap function) மதிப்புகளை இடமாற்றம் செய்யும்

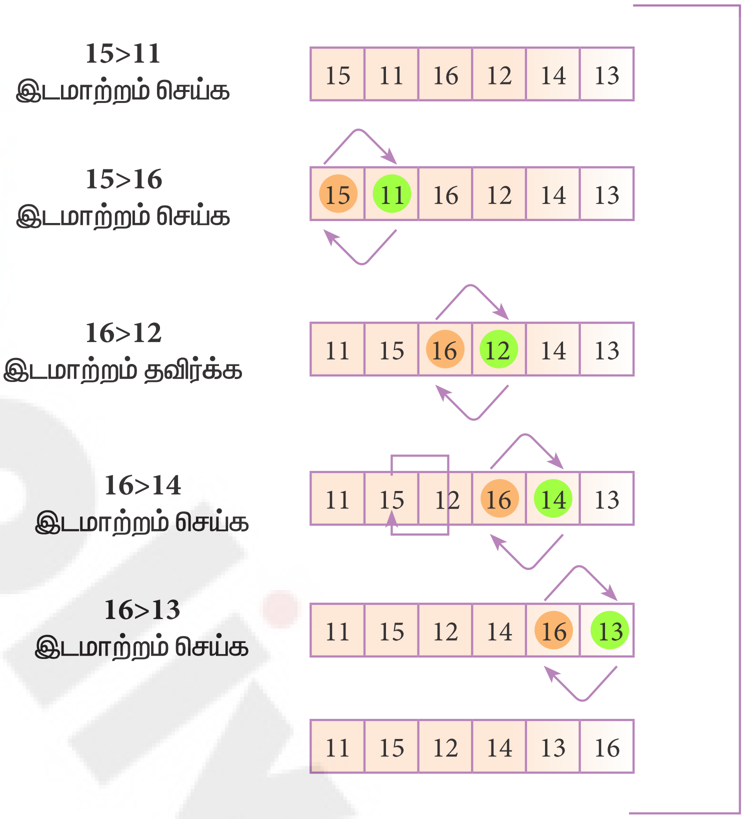
போலி குறிமுறை

1. முதல் உறுப்புடன் (சுட்டெண் = 0), அணியின் தற்போதைய உறுப்போடு அடுத்த உறுப்பை ஒப்பீடு செய்யவும்.
2. தற்போதைய உறுப்பு அடுத்த உறுப்பை விட அதிகம் எனில், அவற்றை இடமாற்றம் செய்யவும்.
3. தற்போதைய உறுப்பு அடுத்த உறுப்பை விட சிறியது எனில், அடுத்த உறுப்பிற்கு செல்லவும் மீண்டும் படிநிலை -1லிருந்து தொடங்கவும்.

{15, 11, 16, 12, 14, 13} மதிப்புகளோடு கூடிய அணியை எடுத்துக் கொள்வோம்.

கீழே குமிழி வரிசையாக்கம் கொடுக்கப்பட்ட அணியை எவ்வாறு வரிசையாக்கம் செய்கிறது என்பதற்கான விளக்கப்படம் கீழே கொடுக்கப்பட்டுள்ளது.

Chapter 4.indd 40 30-01-2020 19:13:42



எனவே, மேலே குறிப்பிடப்பட்டுள்ளதை முதல் சுழற்சி படமாகும். இதேபோல், எல்லா சுழற்சி செய்யப்படும். இறுதி சுழற்சிக்கு பிறகு வரிசையாக்கம் செய்யப்பட்ட அணியை கொடுக்கும். அந்த அணி இவ்வாறு இருக்கும்.

அதைப்போலவே, இரண்டாவது சுழற்சிக்குப்பிறகு 15 என்ற மதிப்பு இரண்டாவது இறுதி சுட்டெண்ணில் இருத்தி வைக்கப்படும். இப்படியாக பிற மதிப்புகளுக்கும் செய்யப்படும்.

11	12	13	14	15	16

4.5.2 தேர்ந்தெடுப்பு வரிசையாக்கம் (Selection Sort)

பட்டியலில் ஒவ்வொரு முறையும் நுழையும் போது ஒரே ஒரு இடமாற்றம் மட்டுமே இருப்பதால், இது குமிழி வரிசையாக்கத்தை விட மேம்பட்டதாகும். தேர்ந்தெடுப்பு வரிசையாக்க-மானது கருத்துருவின் படி மிகவும் எளிமையான வரிசையாக்க நெறிமுறையாகும். இந்த நெறிமுறை, முதலில் மிகச்சிறிய உறுப்பை அணியில் கண்டுப்பிடித்து அதனை முதல் இருப்பிடத்திலுள்ள உறுப்பில் இடமாற்றம் செய்யும். இதைப்போன்றே அணியிலுள்ள அனைத்து உறுப்புகளையும் வரிசைப்படுத்தப்படும் வரை இடமாற்றமானது நடைபெறும்.

அடுத்த மிகச்சிறிய உறுப்பைத் தேர்ந்தெடுத்து, அதனை சரியான இடத்தில் இடமாற்றம் செய்வதை மீண்டும் மீண்டும் இந்த நெறிமுறை செய்வதால் இதனை தேர்ந்தெடுப்பு வரிசையாக்கம் என அழைக்கப்படுகிறது.

போலி குறிமுறை

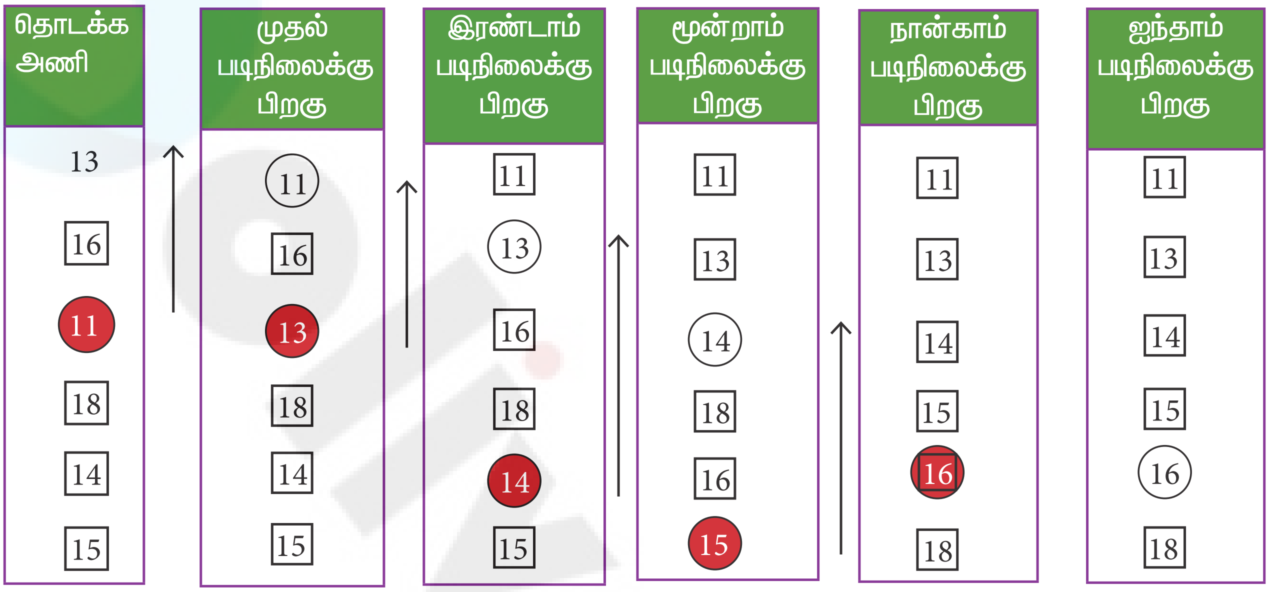
1. முதல் உறுப்பில் தொடங்கி அணியில் உள்ள மிகச் சிறிய உறுப்பைத் தேடி, முதல் இடத்தில் உள்ள உறுப்போடு இடமாற்றம் செய்ய வேண்டும். (அணியின் சுட்டு எண் 0-ல்)
2. பிறகு இரண்டாவது இடத்திற்கு சென்று, துணை அணியில் உள்ள மிகச் சிறிய உறுப்பை சுட்டெண் 1-லிருந்து இறுதி சுட்டெண் வரை தேட வேண்டும்.
3. கொடுக்கப்பட்ட அணியில் இரண்டாவது இடத்தில் படிநிலை 2-ல் கண்டறிந்த உறுப்பை இடமாற்றம் செய்க இடமாற்றம் அல்லது இது துணை அணியின் முதல் இடத்தில் இருக்கும்.

Chapter 4.indd 41 30-01-2020 19:13:43

4. அணி வரிசையாக்கம் செய்யப்படும் வரை, இதை மீண்டும் மீண்டும் செய்தல் வேண்டும்.

{13, 16, 11, 18, 14, 15} மதிப்புகளைக் கொண்ட ஒரு அணியை எடுத்துக்கொள்வோம்.

தேர்ந்தெடுப்பு வரிசையாக்கம் கொடுக்கப்பட்ட அணியை எவ்வாறு வரிசையாக்கம் செய்கிறது என்பதற்கான விளக்கப்படம் கீழே கொடுக்கப்பட்டுள்ளது.



முதல் சுற்றில், மிகச் சிறிய எண் 11 ஆகும். எனவே, அது முதல் இடத்தில் இருத்தி வைக்கப்படுகிறது.

முதல் உறுப்பைத் தவிர்த்து, மீதமுள்ள உறுப்புகளில் இருந்து அடுத்த மிகச் சிறிய உறுப்பைத் தேட வேண்டும். 13 தான் அடுத்த மிகச்சிறிய எண் ஆதலால் அதை இரண்டாம் இடத்தில் இருத்தி வைக்கப்படுகிறது. பிறகு 11 மற்றும் 13-யைத் தவிர்த்து (ஏனெனில் அவை சரியான இடத்தில் உள்ளன) மீதமுள்ள உறுப்புகளிலிருந்து அடுத்த மிகச்சிறிய எண்ணைத் மூன்றாம் இடத்தில் இருத்தி வைக்கப்படுகிறது. இவ்வாறு அணியானது வரிசைப்படுத்தப்படும் வரை மீண்டும் மீண்டும் இச்செயல் நடைபெறும்.

4.5.3 செருகும் வரிசையாக்கம் (Inserction sort)

எளிமையான வரிசையாக்க நெறி முறையான இது நெறிமுறையின் முடிவில் இறுதியாக வரிசையாக்கம் செய்யப்பட்ட அணியினை அமைக்கும். இது அணியின் கீழ்பகுதியில் வரிசையாக்கம் செய்யப்பட்ட துணைப் பட்டியலை எப்பொழுதும் அமைத்துகொள்ளும்.

போலிக் குறிமுறை

படி நிலை 1 – முதல் உறுப்பாக இருந்தால், அது ஏற்கனவே வரிசையாக்கம் செய்யப்பட்டது.

படி நிலை 2 – அடுத்த உறுப்பினைத் தேர்வு செய்ய வேண்டும்.

படிநிலை 3 – வரிசைப்படுத்தப்பட்ட துணைப்பட்டியலுள்ள உறுப்புகளோடு ஒப்பீடு செய்ய வேண்டும்.

படிநிலை 4 – வரிசைப்படுத்தப்பட்ட துணைப்பட்டியலிலுள்ள அனைத்து உறுப்புகளும் வரிசைப்படுத்தப்பட வேண்டிய மதிப்பை விட பெரிய மதிப்பாக இருந்தால் அதனை இடமாற்றம் செய்ய வேண்டும்.

படி நிலை 5 – மதிப்பைச் செருகுதல் வேண்டும்.

படி நிலை 6 - பட்டியல் வரிசைபடுத்தப்படும் வரை இச்செயலை மீண்டும் மீண்டும் செய்ய வேண்டும்.

Chapter 4.indd 42 30-01-2020 19:13:43

44	16	83	07	67	21	34	45	10	Assume 44 is a sorted list of 1 item
16	44	83	07	67	21	34	45	10	inserted 16
16	44	83	07	67	21	34	45	10	inserted 83
07	16	44	83	67	21	34	45	10	inserted 07
07	16	44	67	83	21	34	45	10	inserted 67
07	16	21	44	67	83	34	45	10	inserted 21
07	16	21	34	44	67	83	45	10	inserted 34
07	16	21	34	44	45	67	83	10	inserted 45
07	10	16	21	34	44	45	67	83	inserted 10

மேலே கொடுக்கப்பட்டுள்ள பட்டியலின் மதிப்புகளைச் செருகும் வரிசையாக்கம் நெறிமுறை மூலம் வரிசையாக்கம் செய்யப்பட்டது.

07	10	16	21	34	44	45	67	83

4.6. இயங்கு நிரலாக்கம்

இயங்கு நிரலாக்கம் என்பது ஒரு சிக்கலுக்கு தீர்வுகான வரிசையான முடிவுகளின் மூலம் செயல்படுத்தப்படும் நெறிமுறை வடிவ முறையாகும்.

இயங்கு நிரலாக்க அணுகுமுறை கொடுக்கப்பட்ட சிக்கலை சிறிய சிக்கல்களாகப் பிரிப்பதில் பிரித்து கைப்பற்றுதல் முறை சிக்கலை சிறு-சிறு சிக்கலாக பிரித்து செயல்படுத்துவதாகும்.

இயங்கு நிரலாக்கத்தை எங்கு சிக்கல்கள் உள்ளதோ அங்கு பயன்படுத்தலாம். சிக்கல்களை ஒரே மாதிரியான துணை சிக்கல்களாக பிரிப்பதனால், அதன் மூலம் கிடைக்கும் தீர்வை மீண்டும் பயன்படுத்தலாம். பெரும்பாலும், இந்த நெறிமுறைகள் உகந்த தீர்வாகப் பயன்படுத்தப்படுகிறது. கையில் உள்ள துணை சிக்கல் தீர்ப்பதற்கு முன் இந்த செயல்முறையானது ஏற்கனவே தீர்வு காணப்பட்ட துணை சிக்கல்களின் முடிவுகளை ஆராய முயற்சிக்கும். மிகச் சிறந்த தீர்வை அடைவதற்கு துணை சிக்கல்களின் தீர்வுகளை ஒன்றிணைத்தல் வேண்டும்.

இயங்கு நிரலாக்கத்தின் படிநிலைகள்

- சிக்கல்களை சிறிய ஒன்றோடு ஒன்றிணைந்த துணை சிக்கல்களாகப் பிரிக்க வேண்டும்.
- சிறிய துணை சிக்கல்களின் உகந்த தீர்வைப் பயன்படுத்தி, சிக்கலின் உகந்த தீர்வை அடைய வேண்டும்.
- இயங்கு நிரலாக்கம் நினைவிருத்தலை (Memoization) பயன்படுத்துகிறது.

குறிப்பு

நினைவிருத்தல் என்பது ஒரு உகந்த யுக்தி ஆகும். ஒரே மாதிரியான உள்ளீடு மீண்டும் கொடுக்கப்படும் போது, மிக உயர்ந்த செயற்கூறு அழைப்புகளின் முடிவுகளை சேமித்து தற்காலிக சேமிப்பு முடிவுகளை திருப்பி அனுப்பி, கணினி நிரல்களை வேகப்படுத்துவதற்கு முதன்மையாகப் பயன்படுகிறது.

14.6.1 பைபோனாசி வரிசை (Fibonacci Series) – ஓர் எடுத்துக்காட்டு

பைபோனாசி வரிசையானது முந்தைய இரண்டு எண்களை கூட்டி அடுத்தடுத்த எண்களை உருவாக்கும். பைபோனாசி வரிசை Fib0 மற்றும் Fib1 ஆகிய இரண்டு எண்களுடன் தொடங்கும். Fib0 மற்றும் Fib1 தொடக்க மதிப்பு 0,1 எடுத்துக்கொள்வோம்.

பின்வரும் நிபந்தனைகளை பைபோனாசி வரிசை நிறைவேற்றும்:

$Fib_n = Fib_{n-1} + Fib_{n-2}$

n-ன் மதிப்பு 8 ஆக உள்ளபோது பைனேனொசி வரிசை இவ்வாறு தோன்றும்.

Fib_8 = 0 1 1 2 3 5 8 13

4.6.3 பைபோனாசி சுழற்சி நெறிமுறை (Fibonacci) - இயங்கு நிரலாக்க முறையில் இயக்கு நிரலாக்க முறையில்

முதலில் நாம், பைபோனாசி வரிசைக்கு சுழற்சி நெறிமுறையை வரையறுக்க முயற்சி செய்யலாம்.

f0=0, f1 =1 என தொடக்க மதிப்பிருத்தல் வேண்டும்

படிநிலை-1: Print the initial values of Fibonacci f0 and f1

படிநிலை-2: fib ← f0 + f1 என மதிப்பிருத்தல் வேண்டும்

படிநிலை-3: மதிப்பிருத்தல் f0← f1, f1← fib

படிநிலை-4: பைபோனாசியின் அடுத்த மதிப்பை fib காண்பிக்கவும்

படிநிலை-5: குறிப்பிட்ட வரிசை உருவாகும் படிநிலை-2 வரை திரும்பச் செய்தல்

உள்ளீடு n = 10

10 இலக்க வரை பைபோனாசி நெறிமுறை கீழே கொடுக்கப்பட்டுள்ளது:

பைபோனாசி வரிசை : 0 1 1 2 3 5 8 13 21 34 55

நினைவில் கொள்க

- நெறிமுறை என்பது குறிப்பிட்ட செயலை முடிப்பதற்கான வரையறுக்கப்பட்ட கட்டளைகளின் தொகுதியாகும்.
- நெறிமுறையாளர் என்பவர் நெறிமுறை எழுதுவதில் திறமையானவர் எல்லையுற்றது (finiteness) – நெறிமுறை வரையறுக்கப்பட்ட படிநிலைகளின் எண்ணிக்கைக்குப் பிறகு நிறுத்தப்பட்ட வேண்டும் என்பதை வரையறுக்கிறது.
- நிரல் என்பது நிரலாக்க மொழியில் நெறிமுறையை வெளிப்படுத்துதல் ஆகும்.
- நெறிமுறை ஆய்வு (algorithm analysis) என்பது நெறிமுறையின் நேர மதிப்பீடு மற்றும் இட சிக்கல்களை மாறுபட்ட உள்ளீட்டு அளவுகளுக்கு வரையறுப்பதாகும்.
- இடச்சிக்கல் என்பது நெறிமுறைக்குத் தேவைப்படும் இடத்தின் அளவீடு ஆகும்.
- மேல் எல்லை என்பது நெறிமுறையின் மிகவும் மோசமான தீர்வு ஆகும்.
- கீழ்எல்லை (lower bound) என்பது கொடுக்கப்பட்டுள்ள சிக்கலுக்கான கோட்பாட்டளவில் மிகச் சிறந்த தீர்வு ஆகும்.
- நெறிமுறை எடுத்துக்கொள்ளும் படிநிலைகளின் எண்ணிக்கையின் மூலம் நெறிமுறையின் நேரச்சிக்கல் அளவிடப்படுகிறது.

நினைவில் கொள்க

- நெறிமுறையின் செயல்திறன் என்பது நெறிமுறையின் ஒரு பண்பாகும். நெறிமுறை பயன்படுத்தும் கணக்கீட்டு வளங்களின் எண்ணிக்கையோடு தொடர்புடையது.
- நெறிமுறையுக்தி என்பது நெறிமுறையை வடிவமைக்கும் முறையாகும்.
- நினைவிட நேரம் அல்லது நேரம் நினைவிட பரிமாற்றம் என்பது சிக்கலைத் தீர்க்கும் அல்லது குறைந்த நேரத்தில் அதிக சேமிப்பு இடத்தை அல்லது நினைவிடத்தை கணக்கீடு செய்யும் முறை ஆகும்.
- Asymptotic குறியீடுகளானது நேரம் மற்றும் இடச்சிக்கலுக்கான அர்த்தமுள்ள கூற்றுகளைப் பயன்படுத்தும் மொழியாகும்.
- வரிசைமுறை தேடல் அல்லது தொடர் தேடல் என்பது பட்டியலில் ஒரு குறிப்பிட்ட மதிப்பை கண்டுபிடித்து, வரிசையிலுள்ள ஒவ்வொரு உறப்பையும் சோதித்து, வேண்டிய உறுப்பைக் கண்டுப்பிடிக்கும் வரை தேடும் முறையாகும்.
- இரும தேடல் நெறிமுறையானது பிரித்து கைப்பற்றுதல் தேடலைப் போல இலக்கு மதிப்பின் இருப்பிடத்தைக் கண்டுபிடிக்கிறது.
- குமிழி வரிசையாக்கமானது எளிய வரிசையாக்க நெறிமுறையாகும். வரிசைப்படுத்தப்படாத வரிசையாக இருத்தல் மீண்டும் மீண்டும் அடுத்தடுத்த உறுப்புகளை இடமாற்றம் செய்யும்.
- தேர்ந்தெடுப்பு வரிசையாக்கம் குமிழி வரிசையாக்கத்தைவிட மேம்பட்டதாகும். பட்டியலில் ஒவ்வொரு முறை நுழையும் போதும் ஒரே ஒரு இடமாற்றம் நடைபெறும்.
- ஒன்றிணைத்தல் வரிசையாக்கம் முதலில் அணியிலுள்ள மதிப்புகளை சரிபாதியாக பிரித்து அனைத்தையும் வரிசையாக்கப்பட்ட அணியாக ஒன்றிணைக்கிறது.
- இயங்கு நிரலாக்கமானது சிக்கலைத் தீர்த்து சிறிய துணை சிக்கல்களாக பிரித்து அதன்பிறகு மீண்டும் அதை பயன்படத்துவதற்கு பயன்படுகிறது.

மதிப்பீடு

AK41YS

பகுதி–அ

சரியான விடையைத் தேர்ந்தெடுக்கவும்: **(1 மதிப்பெண்)**

1. எந்த சொல் பெர்ஷிய கணிதமேதை அபு ஜா:ஃபர் முகமது இபின்-ஜமுசா அல் கௌவாரிஸ்மி பெயரில் இருந்து வந்தது?

   (அ) Flowchart (ஆ) Flow (இ) Algorithm (ஈ) Syntax

2. பின்வரும் வரிசையாக்க நெறிமுறையில், எந்த நெறிமுறைக்கு குறைந்த எண்ணிக்கையிலான இடமாற்றம் தேவைப்படும்?

   (அ) குமிழி (ஆ) விரைவு (இ) ஒன்றிணைந்த (ஈ) தேர்ந்தெடுப்பு

3. நெறிமுறையின் செயல்திறனை அளவிட இரண்டு முக்கிய அளவீடுகள் எவை?

   (அ) செயலி மற்றும் நினைவகம் (ஆ) சிக்கல் மற்றும் கொள்ளளவு

   (இ) நேரம் மற்றும் இடம் (ஈ) தரவு மற்றும் இடம்

4. வரிசைமுறை தேடல் நெறிமுறையின் சிக்கல்தன்மை எது?

(அ) O(n) (ஆ) O(log n) (இ) O(n2) (ஈ) O(n log n)

5. பின்வரும் வரிசையாக்க நெறிமுறையில் எது மிகவும் குறைவான மோசமான சிக்கல் தன்மையை உடையது?

(அ) குமிழி (ஆ) விரைவு (இ) ஒன்றிணைந்த (ஈ) தேர்ந்தெடுப்பு

6. பின்வருவனவற்றுள் எது நிலையான வரிசையாக்க நெறிமுறை அல்ல?

(அ) செருகுதல் (ஆ) தேர்ந்தெடுப்பு (இ) குமிழி (ஈ) ஒன்றிணைந்த

7. குமிழி வரிசையாக்கத்தின் மிகச் சிறந்த நிலையில் அதன் நேர சிக்கல்தன்மை

(அ) θ (n) (ஆ) θ (nlogn) (இ) θ (n2) (ஈ) θ (n(logn) 2)

8. Θ என்ற குறியீடு asymptotic மதிப்பீட்டில் எதைக் குறிக்கிறது?

(அ) அடிப்படை நிலை (ஆ) மிதமான நிலை (இ) மோசமான நிலை (ஈ) NULL நிலை

9. ஒரு சிக்கல் துணைச் சிக்கல்களாக பிரித்து அதனை பல முறை பயன்படுத்தினால், அந்த சிக்கல் எந்த பண்பை பெறும்?

(அ) ஒன்றோடு ஒன்றிணைந்த துணைச்சிக்கல் (ஆ) உகந்த துணை கட்டமைப்பு

(இ) நினைவிருத்தல் (ஈ) பொறாமை

10. இயங்கு நிரலாக்கத்தில், ஏற்கனவே கணக்கீடு செய்த மதிப்புகளை சேமிக்கு யுக்தியை இவ்வாறு அழைக்கலாம்.

(அ) மதிப்பை சேமிக்கும் பண்பு (ஆ) மதிப்பை சேகரிக்கும் பண்பு

(இ) நினைவிருத்தல் (ஈ) படமிடல்

பகுதி–ஆ

அனைத்து வினாக்களுக்கும் விடையளி **(2 மதிப்பெண்)**

1. நெறிமுறை என்றால் என்ன?
2. போலிக் குறிமுறை வரையறை.
3. நெறிமுறையாளர் என்பவர் யார்?
4. வரிசையாக்கம் என்றால் என்ன?
5. தேடல் என்றால் என்ன? அதன் வகைகளை எழுதுக.

Chapter 4.indd 46 30-01-2020 19:13:43

பகுதி–இ

அனைத்து வினாக்களுக்கும் விடையளி (3 மதிப்பெண்)

1. நெறிமுறையின் பண்பியல்புகளைப் பட்டியலிடுக.
2. சிக்கல்தன்மை மற்றும் வகைகளைப் பற்றி விவாதிக்க.
3. இடம் மற்றும் இடசிக்கல்களின் மீது தாக்கத்தை ஏற்படுத்தும் காரணிகள் யாவை?
4. Asymptotic குறியீடு-குறிப்பு வரைக.
5. இயங்கு நிரலாக்கத்தைப் பற்றி நீவிர் அறிவின யாவை?

பகுதி ஈ

அனைத்து வினாக்களுக்கும் விடையளி (5 மதிப்பெண்)

1. நெறிமுறையின் பண்பியல்புகளை விவரி.
2. வரிசைமுறை தேடல் நெறிமுறையை விவாதிக்கவும்.
3. இருமத் தேடல் என்றால் என்ன? எடுத்துக்காட்டுடன் விளக்குக.
4. குமிழி வரிசையாக்க நெறிமுறையை எடுத்துக்காட்டுடன் விளக்குக.
5. இயங்கு நிரலாக்கத்தின் கருத்துருவை பொருத்தமான எடுத்துக்காட்டுடன் விளக்கவும்.

Reference Books

1. *Fundamentals Computer Algorithms, Ellis Horowitz, Sartaj Sahni, Sanguthevar, Rajasekaran, Second Edition, University press (India) Limited, 2013.*
2. *Design and Analysis of Algorithms, S. Sridhar, Oxford University Press, 2015*

Web References

www.wickipedia.org

மாணவர்கள் செயல்பாடு

1. உனது வகுப்பு மாணவர்களின் தரநிலையை காலாண்டுத் தேர்வின் மதிப்பெண்களின் அடிப்படையில் அனைத்து நிபந்தனைகளையும் நிறைவேற்றும் வகையில் நெறிமுறையை உருவாக்கவும்.
2. உனது வகுப்பின் மாணவர்களின் உயரத்தை மாணவனின் உயரம் மற்றும் மாணவியின் உயரம் என்று பிரிக்கப்பட்ட பின் அவற்றை ஏறுவரிசையில் வரிசையாக்கம் செய்யப்பட்ட ஒரே பட்டியலாக ஒன்றிணைக்கவும்.

Chapter 4.indd 47 30-01-2020 19:13:43

பன்னிரெண்டாம் வகுப்பு பாடத்திட்டத்தில் பைத்தான் கற்றலின் என்ன?

ஆசிரியர்கள் மற்றும் மாணவர்கள் மனதில் ஏற்பட்டுள்ள வினா இதுவே.

ஒரு புதியவர் கூட பைத்தான் நிரலை மிக எளிதாக படித்துப் புரிந்து கொள்ளும் வகையில் இந்தபுத்தகம் வடிவமைக்கப்பட்டுள்ளது.

நற்சான்றுகள்

- கூகுள் இன்க் நிறுவனத்தின் Search quality துறையின் இயக்குனராகிய "பீட்டர்நார்விக்" என்பவர் பைத்தானைப் பற்றி பின்வருமாறு சான்றளிக்கிறார். "கூகுள் அமைப்பின் ஆரம்பம் முதலே அதன் உருவாக்கம் மற்றும் வளர்ச்சியில் பைத்தான் பெரும் பங்கு வகிக்கிறது. இன்று பல கூகுள் பொறியாளர்கள் பைத்தானைப் பயன்படுத்துகிறார்கள். மேலும், பைத்தான் மொழியில், தகுதி வாய்ந்த பல பொறியாளர்களை எதிர்ப்பார்க்கிறோம்".
- Youtube நிறுவனத்தின் மென்பொருள் வடிவமைப்பாளர், "குயாங் டோ" பைத்தானைப் பற்றி பின்வருமாறு சான்றளிக்கிறார். "எங்களது வலைத்தளத்தில் பைத்தான் மிக வேகமாகச் செயல்படுகிறது. மிகக் குறைவான நேரத்தில், குறைந்த நிரலர்களால் அதிக சிறப்பம்சங்கள் தருவதற்கு பைத்தான் அனுமதிக்கிறது".

சமீப வருடங்களில் மிதமிஞ்சிய மற்றும் பிரபலமான வளர்ச்சியைப் பைத்தான் அடைந்துள்ளது. இன்று, ஒவ்வொரு நிரலருக்கும் பைத்தானைப் பற்றி தெரிந்து கொள்ளுதல் மிகவும் நன்மை அளிக்கிறது. ஏனெனில், அனைத்து மென்பொருள் தீர்வுகளிலும் பைத்தான் மிக முக்கிய பங்கு வகிக்கிறது. பிற மொழிகளை ஒப்பிடுகையில் பைத்தான் அனைவரின் கவனத்தையும் ஈர்க்கக்கூடிய வகையில் வளர்ச்சியடைந்துள்ளது.

நியூயார்க்கில் உள்ள Institute of Electrical & Electronics Engineers என்ற நிறுவனத்தால் வெளியிடப்படுகிறது. IEEE Spectrum என்ற இதழ் பைத்தானை 2018ல் தொடர்ந்து இரண்டாவது முறையாக மிக உயர்ந்த மொழியாக தரவரிசைப்படுத்தியுள்ளது.

மேலே கொடுக்கப்பட்டுள்ள புள்ளி விவரப்படி பைத்தான் 100 சதவீதத்தையும், C++ 99.7 சதவீதத்தையும் பெற்றுள்ளது.

Google. Instagram, Pinterest, Yahoo, Disney, IBM, Nokia மற்றும் பல நிறுவனத்தின் தொழில் நுட்பவியலாளர்களால் பைத்தான் பிரபலமாக பயன்படுத்தப்படுகிறது.

1. Python	100.0
2. C++	99.7
3. Java	97.5
4. C	96.7
5. C#	89.4
6. PHP	84.9
7. R	82.8
8. JavaScript	82.6
9. Go	76.4
10. Assembly	74.1

பின்வரும் காரணங்களால் பைத்தான் பல வணிகங்களில் பயன்படுத்தப்பட அறிவுறுத்தப்படுகிறது.

- எளிய தொடரியல்களையும் வாசிப்புத்தன்மையும் கொண்டுள்ளது.
- பொருள்நோக்கு நிரலாக்கத்துடன் (OOPS) கூடிய உயர்நிலை குறியீட்டு மொழியைக் கொண்டுள்ளது.
- மகத்தான செயற்கூறுகள், தொகுதிகள், நூலகக் கூறுகள், கட்டமைப்புகள் மற்றும் கருவிகள் ஆகியவற்றுக்கு மிக பிரபலமானதாகும்.
- உள்ளிணைந்த செயற்கூறுகள் அறிவியல் கணக்கீடுகளை ஆதரிக்கிறது.

அனைத்து துறைகளிலும் கணினியின் வருகையால் நாம் வேலை செய்யும் முறையில் குறிப்பிடத்தக்க மாற்றங்கள் ஏற்பட்டுள்ளது. ஒவ்வொரு நிறுவனத்திலும் கணினிமயமாக்கல் என்பது உற்பத்தியை பெருக்கியுள்ளது மற்றும் முடிவெடுக்கும் திறனை துரிதப்படுத்தியுள்ளது. கணிப்பொறியாளர்கள், மருத்துவர்கள், பட்டைய கணக்காளர்கள் மற்றும் வீட்டு நிர்வாகிகள் போன்ற பல தனிநபர்களின் வாழ்க்கை முறையிலும் கடுமையான மாற்றத்தை ஏற்படுத்தியுள்ளது. இந்த மாற்றங்களை ஏற்றுக் கொண்டு ஒளிமயமான எதிர்காலத்தை நோக்கிப் பயணிப்போம்.

python interleaf page.indd 48 10-02-2020 19:58:07

அலகு II

பாடம் 5

பைத்தான் அறிமுகம்–மாறிகள் மற்றும் செயற்குறிகள்

DYS473

கற்றலின் நோக்கங்கள்

இந்தப் பாடப்பகுதியைக் கற்றபின் மாணவர்கள் அறிந்துக் கொள்வது:

- வரைகலை பயனாளர் இடைமுகம் (GUI) மற்றும் ஒருங்கிணைந்த உருவாக்க சூழலில் பைத்தான் நிரல்களை உருவாக்குதல்.
- ஊடாடும் மற்றும் ஸ்கிரிப்ட் முறைமையில் நிரல் உருவாக்குதல்.
- மாறிகளை உருவாக்குதல் மற்றும் அவற்றிக்கு மதிப்பிருத்தல்.
- பைத்தானில் உள்ள பல்வேறு தரவு வகைகளின் கருத்துகள் மற்றும் அவற்றின் பயன்பாடுகளை பற்றி அறிந்து கொள்ளுதல்.
- பல்வேறு வகையான செயற்குறிகளின் பயன்பாடுகளை புரிந்துக் கொள்ளுதல் (கணித, ஒப்பீட்டு மற்றும் தருக்க)
- பைத்தான் கோவைகள் மற்றும் கூற்றுகளை உருவாக்குதல்.

5.1 அறிமுகம்

பைத்தான் என்பது பொது பயன்பாடு நிரலாக்க மொழி ஆகும். இதை நெதர்லாந்தின் கணிதவியல் மற்றும் கணினி அறிவியல் தேசிய ஆராய்ச்சி நிறுவனம் என்று அழைக்கப்படும் CWI (Centrum Wiskunde & Informatica)-யை சேர்ந்த கைடோ வான் ரோசும் என்பவரால் உருவாக்கப்பட்டது. இம்மொழி 1991ஆம் ஆண்டு வெளியிடப்பட்டது. பைத்தான் அதன் பெயரை, எழுபதுகளில் பிரபலமான பிபிசி நகைச்சுவை தொடரான "மோன்டி பைத்தான்'ஸ் ஃப்ளையிங் சர்க்கஸ்" இருந்து பெற்றது. பைத்தான் கட்டமைக்கப்பட்ட நிரலாக்க மொழி மற்றும் பொருள் நிரலாக்க மொழியை ஆதரிக்கிறது.

5.2 பைத்தானின் சிறப்பம்சங்கள்

✓ இது ஒரு பொது பயன்பாட்டு மொழியாகும். இதை அறிவியல் மற்றும் அறிவியல் அல்லாத நிரலாக்கத்திற்கு பயன்படுத்தலாம்.

Chapter 5.indd 49 30-01-2020 19:20:39

- ✓ இது இயக்க முறையை சாராத நிரலாக்க மொழியாகும்.
- ✓ பைத்தான் நிரல் எளிதாக புரிந்து கொள்ள இயலும்.

பைத்தான் 3.x பதிப்பில் IDLE (Integrated Development Learning Environment) நிரல் குறிமுறையை உருவாக்கவும், இயக்கவும் பயன்படுகிறது. இதை www.python.org என்ற இணைய முகவரியில் இருந்து பதிவிறக்கம் செய்து கொள்ளலாம்.

5.3 பைத்தான் நிரலாக்கம்

பைத்தான் நிரலை, ஊடாடும் முறைமை அல்லது ஸ்கிரிப்ட் முறைமைகளில் எழுதலாம். ஊடாடும் முறைமை பைதான் தூண்டுகுறியில் (>>>) நிரலை எழுத உதவும். ஸ்கிரிப்ட் முறைமையில் நிரல் குறிமுறை தனி கோப்பாக, **.py** என்ற நீட்டிப்புடன் சேமிக்கப்பட்டு இயக்கப்படும். ஸ்கிரிப்ட் முறைமை பைதான் மூல நிரலை உருவாக்கவும், பதிப்பாய்வு செய்யவும் பயன்படுகிறது.

5.3.1 ஊடாடும் முறைமை நிரலாக்கம்

ஊடாடும் முறைமையில் பைத்தான் குறிமுறையை நேரடியாக தூண்டுகுறியில் உள்ளீடப்பட்டவுடன், மொழிப்பெயர்ப்பி தீர்வுகளை உடனடியாக காட்டும். ஊடாடும் முறைமை ஒரு எளிய முறை **கால்குலேட்டரை** போல் பயன்படுத்தலாம்.

5.3.1.1 பைத்தான் IDLEயை தொடங்குதல்

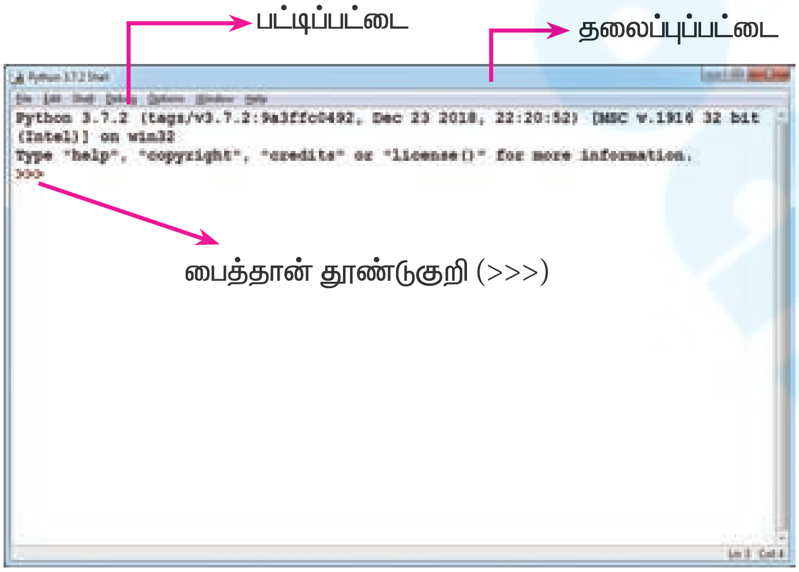
பின்வரும் கட்டளையை கொண்டு விண்டோஸ் இயக்க அமைப்பில் பைத்தான் IDLEயை துவக்கலாம்.

Start → All Programs → Python 3.x → IDLE (Python 3.x)

(அல்லது)

முகப்பு திரையில் என்ற குறும்படம் இருப்பின் அதை கிளிக் செய்து தொடங்கலாம்.

இப்போது, படம் 5.1 போன்று பைத்தான் IDLE திரை தோன்றும்.



படம் 5.1 – பைத்தான் IDLE சாளரம்

Chapter 5.indd 50 30-01-2020 19:20:39

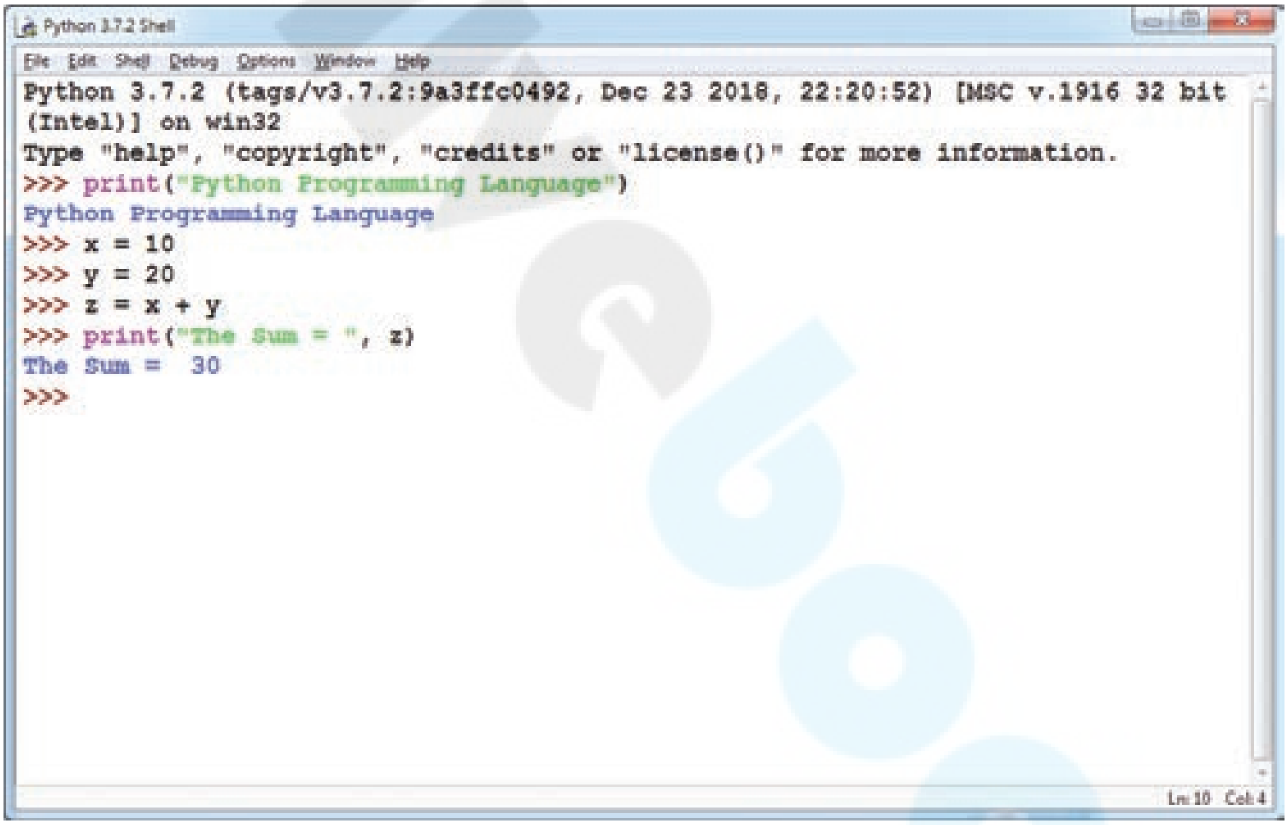
>>> என்ற தூண்டுகுறி பைதான் மொழிப்பெயர்ப்பி கட்டளைப் பெற்றுக் கொள்ள தயார் நிலையில் இருப்பதை குறிக்கிறது. அதாவது IDLE திரையில் தோன்றும் தூண்டுகுறி ஊடாடும் முறைமையில் இருப்பதை குறிக்கிறது. இப்பொழுது ஒரு எளிய முறை கால்குலேட்டரில், கணக்கீடு கோவையை செய்து பார்க்கலாம்.

எடுத்துக்காட்டு 1:

```
>>> 5 + 10
15
>>> 5 + 50 *10
505
>>> 5 ** 2
25
```

எடுத்துக்காட்டு 2:

```
>>>print ("Python Programming Language")
>>>x=10
>>>y=20
>>>z=x + y
>>>print ("The Sum", z)
```



படம் 5.2 – பைத்தான் ஊடாடும் சாளரம்

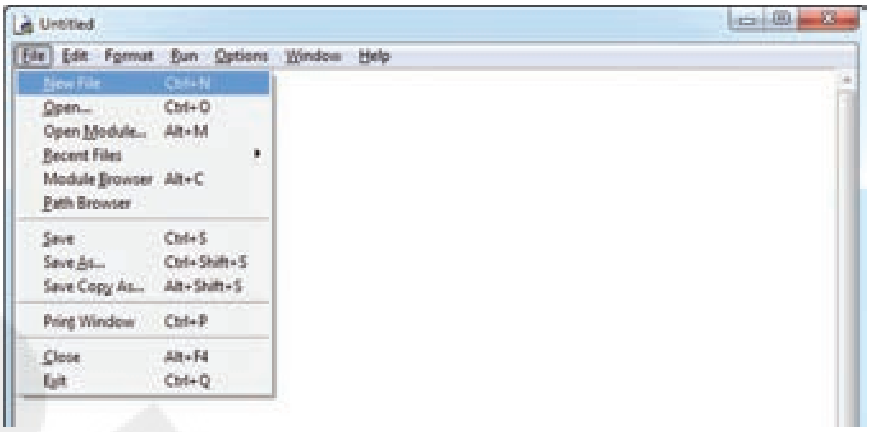
5.3.2 பைத்தான் ஸ்கிரிப்ட் நிரலாக்கம்

அடிப்படையாக, ஒரு ஸ்கிரிப்ட் என்பது பைத்தான் கட்டளைகளை கொண்ட ஒரு உரை ஆவணத்தைக் குறிக்கிறது. பைத்தான் ஸ்கிரிப்ட் குறிமுறையை மறுபயனாக்கம் செய்துக் கொள்ளலாம். ஒரு முறை ஸ்கிரிப்ட் கோப்பை எழுதிவிட்டால் பல முறை இயக்கிக் கொள்ளலாம் மீண்டும் அந்த ஸ்கிரிப்டை உருவாக்க தேவையில்லை. ஸ்கிரிப்ட் பதிப்பாய்வு செய்ய கூடியவை.

5.3.2.1 பைத்தான் ஸ்கிரிப்ட் எழுதுதல்

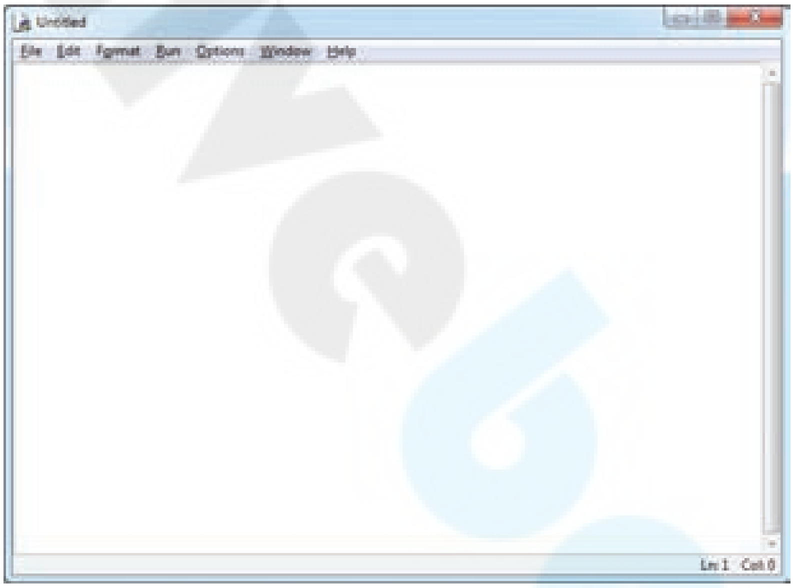
1. பைத்தான் shell சாளரத்தில் File → New File அல்லது Ctrl + N என்பதை அழுத்தவும்.

P6JESB



படம் *5.3* – புதிய கோப்பு உருவாக்குதல்

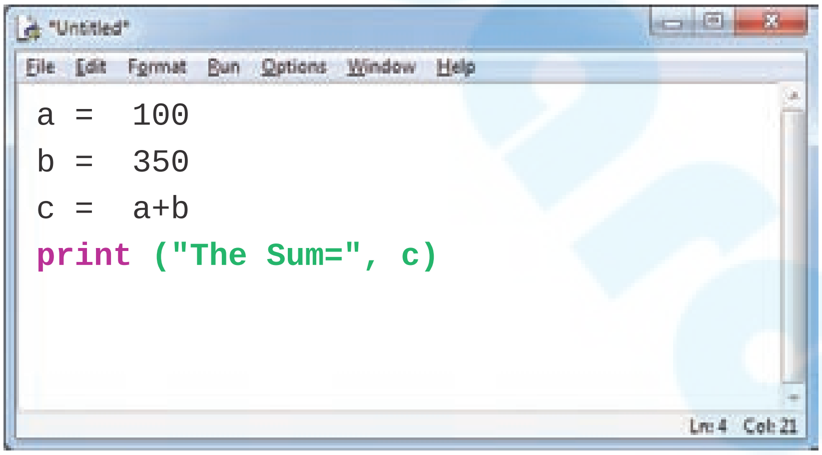
2. படம் 5.3(அ) வில் காணப்படுவதைப்போல, பெயரிடப்படாத Untitled வெற்று ஸ்கிரிப்ட் Text editor தோன்றும்.



படம் *5.3 (அ)* பெயரிடப்படாத Untitled வெற்று ஸ்கிரிப்ட் editor

3. பின்வரும் கட்டளைகளை ஸ்கிரிப்ட் editorல் உள்ளிடவும்.

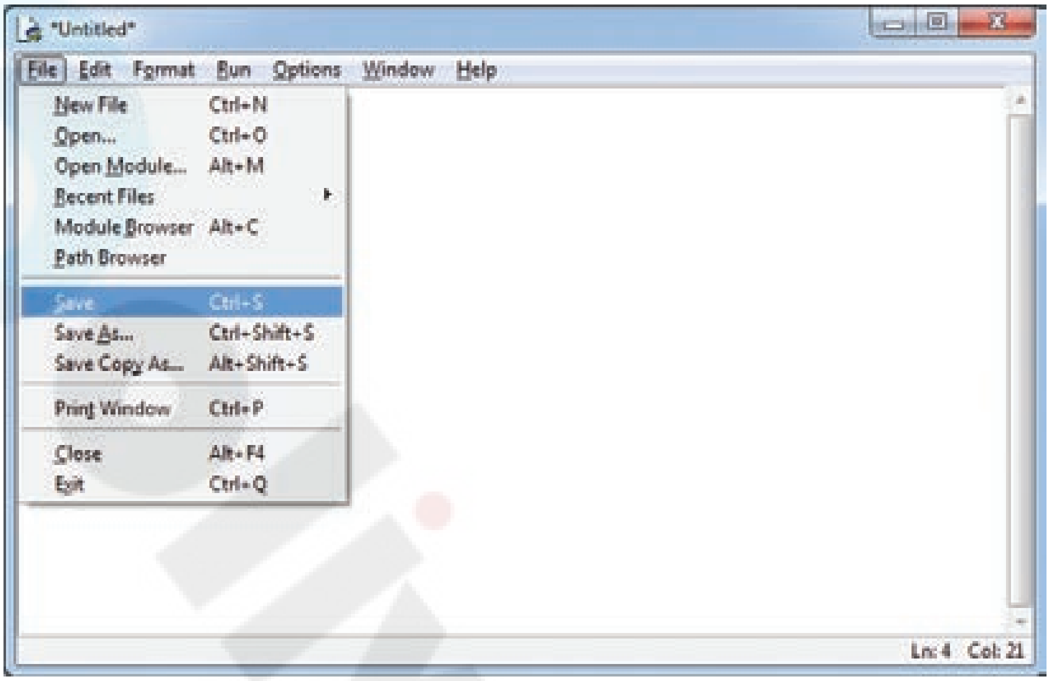
```
a =100

b = 350

c = a+b

print ("The Sum=", c)
```



படம் *5.4* – பைத்தான் மாதிரி குறிமுறை

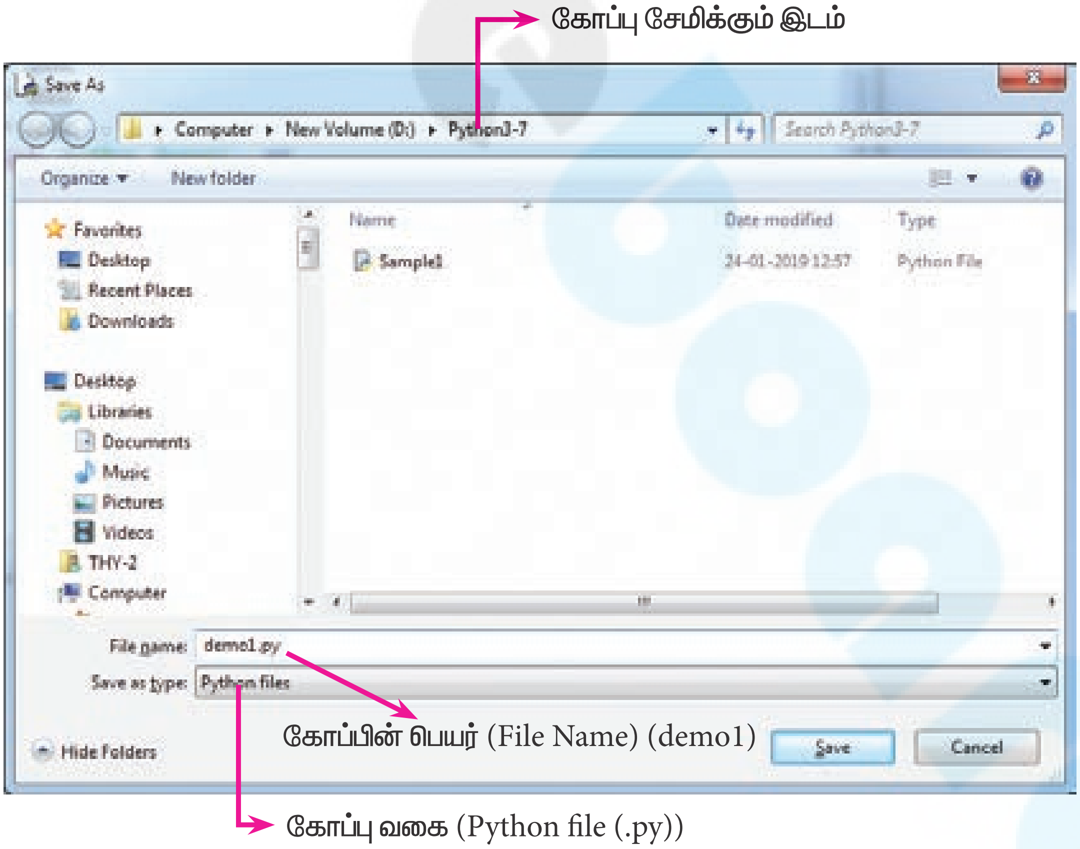
5.3.2.2. பைத்தான் ஸ்கிரிப்ட்யை சேமித்தல்

(1) **File → Save** கட்டளையை தேர்ந்தெடுக்கவும் அல்லது **Ctrl + S** அழுத்தவும்.



படம் *5.5* – முதன்முறை ஒரு கோப்பை சேமித்தல்

(2) இப்பொழுது, படம் 5.6 காணப்படும் **Save As** உரையாடல் பெட்டி திரை தோன்றும்.



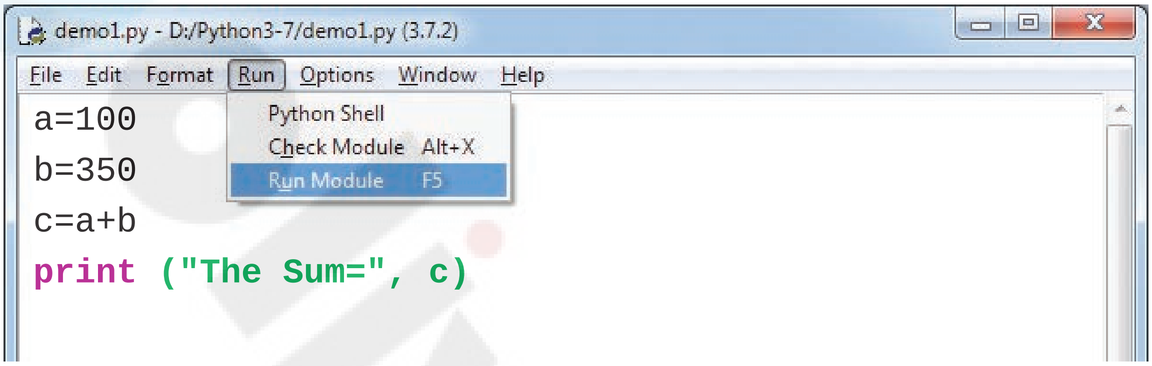
படம் *5.6 – Save As* உரையாடல் பெட்டி

(3) **Save As** உரையாடல் பெட்டியில், கோப்பு சேமிக்கப்பட வேண்டிய இடத்தை தேர்ந்தெடுக்க வேண்டும். பின்னர் கோப்பின் பெயரை **File Name** பெட்டியில் உள்ளிட வேண்டும். முன்னியல்பாக பைதான் கோப்புகள் **.py** என்ற நீட்டிப்புடன் சேமிக்கப்படுகிறது. எனவே, பைத்தான் ஸ்கிரிப்ட் **editor**-ல் கோப்புகளுக்கு பயனர் நீட்டிப்பை குறிப்பிட வேண்டிய அவசியமில்லை.

(4) இறுதியாக **Save** பொத்தானை தேர்ந்தெடுத்து, பைத்தான் ஸ்கிரிப்ட்டை கோப்பை சேமிக்கலாம்.

5.3.2.3 Executing பைத்தான் ஸ்கிரிப்ட்டை இயக்க

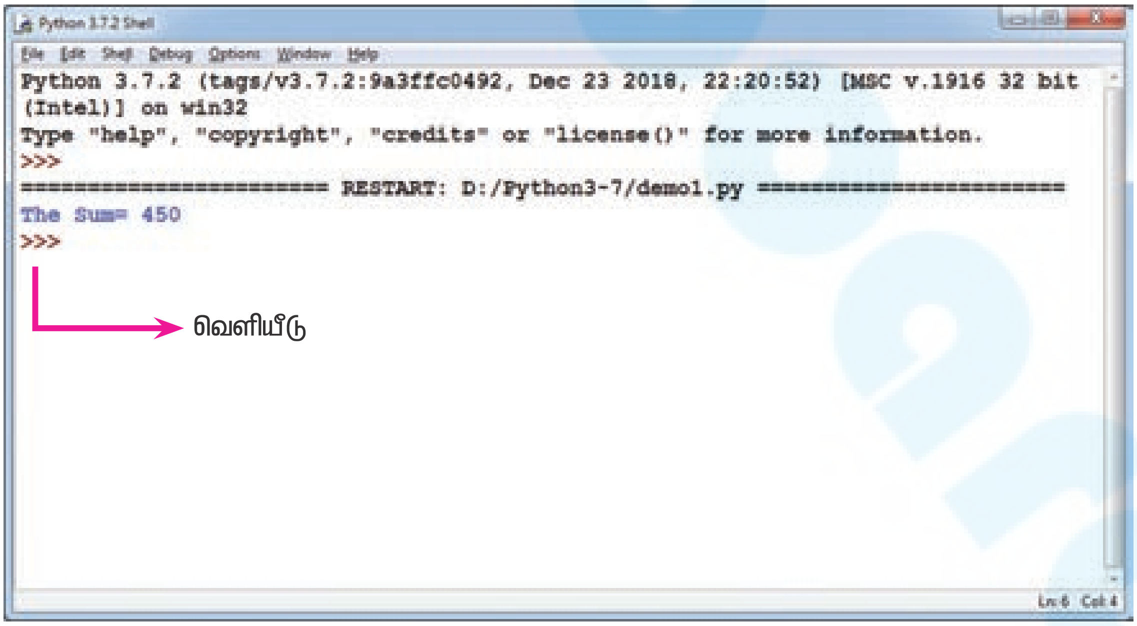
(1) **Run → Run Module** அல்லது **F5** யை தேர்ந்தெடுக்கவும்.



படம் 5.7 – பைத்தான் ஸ்கிரிப்ட் இயக்குதல்

(2) குறிமுறையில் பிழைகள் இருப்பின் அவை சிகப்பு நிறத்தில் **IDLE** திரையில் காண்பிக்கப்பட்டு, பைத்தான் பிழைக்கான காரணத்தை விளக்கும். பிழைகளை திருத்துவதற்கு, ஸ்கிரிப்ட் editorக்கு சென்று, பிழைகளைத் திருத்தி, கோப்பை **Ctrl + S** அல்லது **File → Save** கட்டளைக் கொண்டு சேமித்தபின் மீண்டும் இயக்க வேண்டும்.

(3) பிழைகள் இல்லாத பைத்தான் குறிமுறையை இயக்கியவுடன் அதன் வெளியீடு பைத்தான் **IDLE** திரை 5.8 –ல் உள்ளது போன்று தோன்றும்.



படம் 5.8 –பைத்தான் ஸ்கிரிப்ட் வெளியீடு திரை

Chapter 5.indd 54 30-01-2020 19:20:39

5.4 உள்ளீடு மற்றும் வெளியீடு செயற்கூறுகள்

ஒரு நிரல் பயனர் விரும்பிய பணியை நிறைவேற்றுவதற்கு பயனருடன் தொடர்பு கொள்ள வேண்டும். இதற்கு உள்ளீடு - வெளியீடு செயற்கூறுகள் பயன்படுகிறது. **input()** செயற்கூறு, ஓர் நிரலை இயக்கும் பொழுது தரவுகளை உள்ளீடு செய்யவும், **print()** என்ற வெளியீடு செயற்கூறு நிரலின் தீர்வுகளை திரையில் காண்பிக்க உதவுகிறது.

5.4.1 Print() செயற்கூறு

பைத்தானில், **print()** செயற்கூறு நிரலை இயக்கும் பொழுது தரவுகளை வெளியிட பயன்படுகிறது.

print() செயற்கூறின் தொடரியல் கீழே கொடுக்கப்பட்டுள்ளது

```
print (“string to be displayed as output ” )
print (variable )
print (“String to be displayed as output ”, variable)
print (“String1 ”, variable, “String 2”, variable, “String 3” ……)
```

எடுத்துக்காட்டு:

```
>>> print (“Welcome to Python Programming”)
        Welcome to Python Programming
>>> x = 5
>>> y = 6
>>> z = x + y
>>> print (z)
        11
>>> print (“The sum = ”, z)
        The sum = 11
>>> print (“The sum of ”, x, “ and ”, y, “ is ”, z)
        The sum of 5 and 6 is 11
```

வெளியீடு:

```
        Welcome to Python programming
        11
        The sum = 11
        The sum of 5 and 6 is 11
```

print() செயற்கூறு தீர்வுகளை திரையில் காண்பிக்கும் முன் கோவையை மதிப்பீடு செய்யும். **print()** செயற்கூறினுள் கொடுக்கப்படும் முழு கூற்றினையும் திரையில் காட்டும். காற்புள்ளி(,) கொண்டு **print()** செயற்கூறினுள் ஒன்றுக்கு மேற்பட்ட உருப்புகளை பிரிக்கலாம்.

Chapter 5.indd 55 30-01-2020 19:20:39

5.4.2 input()

பைத்தானில், input() செயற்கூறு நிரலை இயக்கும் பொழுது தரவுகளை உள்ளீடாக பெற்றுக் கொள்ளப் பயன்படுகிறது. input() செயற்கூறின் தொடரியல்,

```
Variable = input (“prompt string”)
```

இந்த **தொடரியலில்** உள்ள **“prompt string”** பயனர் கொடுக்கப்பட வேண்டிய உள்ளீடு எதுவென்பதை உணர்த்தும் கூற்று அல்லது செய்தியை கொண்டிருக்கும்.

input() கட்டளை அமைப்பில் உள்ள **“prompt string”** திரையில் தோன்றும், பயனர் இதை படித்து உள்ளீடு சாதனத்திலிருந்து எதிர்பார்க்கப்படும் தரவுகளை கொடுக்கலாம். **input()** செயற்கூறு, விசைப்பலகையின் மூலம் தட்டச்சு செய்தவற்றை மாறியில் சேமித்து வைக்கும். **“prompt string”** கொடுக்கப்படவில்லையெனில் திரையில் எந்த தகவலும் தோன்றாது. இதனால் பயனர் எவற்றை உள்ளீடு செய்ய வேண்டும் என்பதை தெரிந்து கொள்ள முடியாது.

எடுத்துக்காட்டு 1: “prompt string” கொண்டுள்ள input() செயற்கூறு

```
>>> city=input (“Enter Your City: ”)
        Enter Your City: Madurai
>>> print (“I am from “, city)
        I am from Madurai
```

எடுத்துக்காட்டு 2: “prompt string” இடம்பெறாத input( ) செயற்கூறு

```
>>> city=input()
        Rajarajan
>>> print (I am from", city)
        I am from Rajarajan
```

மேலே உள்ள எடுத்துக்காட்டு 2ல், input( ) செயற்கூற்றில் **“prompt string”** இடம்பெறவில்லை. எனவே, எவற்றை உள்ளீடாக தர வேண்டும் என்பது பயனர் புரிந்து கொள்ள இயலாது. நிரலுக்கு தொடர்பில்லாத தரவுகளை உள்ளிட்டால், தவறான தீர்வை வெளியிடும். எனவே பயனர் நிரலை உரிய முறையில் இயக்க, **“prompt string”** input() செயற்கூறில் பயன்படுத்த வேண்டும்.

input( ) செயற்கூறு அனைத்து தரவுகளையும் சரங்கள் அல்லது குறியுருவாக ஏற்றுக் கொள்ளும், எண்களாக அல்ல. பயனர் எண்களை மதிப்புகளாக உள்ளீடு செய்யும் போது, அவை வெளிப்படையாக எண்வகை தரவுகளாக மாற்றப்பட வேண்டும். **int()** செயற்கூறு சரவகை தரவை எண்வகை தரவாக வெளிப்படையாக மாற்ற உதவுகிறது. இது போன்ற மற்ற செயற்கூறுகளை பின்வரும் பாடங்களில் படிக்கலாம்.

எடுத்துக்காட்டு 3:

```
x = int (input("Enter Number 1: "))
y = int (input("Enter Number 2: "))
print ("The sum = ", x+y)
```

வெளியீடு:

```
Enter Number 1: 34
Enter Number 2: 56
The sum = 90
```

எடுத்துக்காட்டு 4: மேலே காணும் நிரலுக்கு ஒரு மாற்று முறை

```
x,y=int (input("Enter Number 1 :")),int(input("Enter Number 2:"))
print ("X = ",x," Y = ",y)
```

வெளியீடு:

```
Enter Number 1 :30
Enter Number 2:50
X =  30  Y =  50
```

5.5 பைத்தான் குறிப்புரை

பைத்தானில் குறிப்புரை ஹாஷ் (#) குறியுடன் தொடங்கும். # குறியுடன் தொடங்கும் வரிகள் குறிப்புரையாக கருதப்பட்டு, பைத்தான் மொழிப்பெயர்ப்பி இவற்றை குறிமுறையின் பகுதியாக கருதாது. குறிப்புரைகள், ஒரு வரி அல்லது பல வரியாக இருக்கலாம். பல வரி குறிப்புரை கீழே காண்பதைப் போல் இரட்டை ''' குறிக்குள் ஒரு தொகுதியாக கொடுக்கப்பட வேண்டும்.

```
# It is Single line Comment
'''It is multiline comment
which contains more than one line '''
```

5.6 உள்தள்ளல்

பைத்தான், இடைவெளி மற்றும் தத்தல்களை கொண்டு நிரலின் தொகுப்புகளை குறிக்கிறது. ஆனால், C, C++, java போன்ற மொழிகள் நெளிவு { } அடைப்புக்குறி, நிரலில் இனக்குழு, செயற்கூறுகள், மடக்கு, தேர்ந்தெடுப்பு தொகுப்புகளை குறிக்கப் பயன்படுகிறது. நிரலில் தொகுப்புகளை குறிப்பிட வேண்டிய வெற்று இடைவெளி எண்ணிக்கை உள்தள்ளல் ஒரு குறிப்பிட்ட அளவில் இருக்காது (வெற்று இடைவெளி மற்றும் தத்தல்கள்). ஆனால் ஓர் தொகுப்புகள் இடம் பெறும் கூற்றின் அனைத்திற்கும் ஒரே அளவிலான இடைவெளி இருக்க வேண்டும்.

5.7 வில்லைகள்

பைத்தான், நிரலில் இடம்பெறும் வரிகளை அடிப்படை சொற்களாக பிரிக்கிறது. இந்த கூறுகள் வில்லைகள் எனப்படும் பொதுவாக வில்லைகளின் வகையான,

1) குறிப்பெயர்கள்
2) சிறப்புச் சொற்கள்
3) செயற்குறிகள்
4) வரம்புக்குறி மற்றும்
5) நிலைஉரு

வில்லைகளை பிரிப்பதற்கு வெற்று இடைவெளி கண்டிப்பாக இருத்தல் வேண்டும்.

5.7.1. குறிப்பெயர்கள்

மாறி, செயற்கூறு, இனக்குழு, தொகுதி அல்லது பொருளின் பெயர்களை குறிப்பெயர் என்று அழைக்கப்படுகிறது.

- குறிப்பெயர்கள் எழுத்துக்கள் (A..Z or a..z) அல்லது அடிக்கீறு (_) கொண்டு தொடங்க வேண்டும்.
- குறிப்பெயர்கள் எண்கள் (0 .. 9) கொண்டிருக்கலாம்.
- பைத்தான் குறிப்பெயர்கள் எழுத்து வடிவுணர்வு கொண்டது. அதாவது ஆங்கில பெரியயெழுத்துகள் (uppercase) மற்றும் ஆங்கில சிறியயெழுத்துகள் (lowercase) வெவ்வேறாக கருதிக் கொள்ளும்.
- குறிப்பெயர்கள் பைத்தான் சிறப்புச் சொற்களாக இருக்க கூடாது.
- பைத்தானில் %,$, @ etc., போன்ற நிறுத்தற்குறிகள் குறிப்பெயர்களில் இடம் பெறக்கூடாது.

சரியான குறிப்பெயர்களின் எடுத்துக்காட்டு:

Sum, total_marks, regno, num1

தவறான குறிப்பெயர்களின் எடுத்துக்காட்டு :

12Name, name$, total-mark, continue

5.7.2. சிறப்புச் சொற்கள்

நிரலின் அமைப்பை அடையாளம் காண பைத்தான் மொழிப்பெயர்ப்பி சிறப்புச் சொற்களை பயன்படுத்துகிறது. சிறப்பு சொற்கள் மொழிபெயர்ப்பியில் குறிப்பிட்ட பொருள் கொண்டுள்ளதால் இவற்றை பிற பயன்பாட்டிற்கு பயன்படுத்தக்கூடாது.

அட்டவணை 5.1 பைத்தான் சிறப்புச் சொற்கள்

False	class	finally	is	return
none	continue	for	lambda	try
True	def	from	nonlocal	while

and	del	global	not	with
as	elif	if	or	yield
assert	else	import	pass	None
break	except	in	raise	

5.7.3 செயற்குறிகள்

கணிப்பொறி நிரலாக்க மொழியில் செயற்குறிகள் என்ற சிறப்பு குறியீடுகள் கணிப்பீடுகள், நிபந்தனை சோதிப்பு போன்ற செயல்பாடுகளை செய்ய பயன்படுகிறது. செயற்குறிகள் என்பவை குறியீடுகள் இவை செயலேற்பிகளில் உள்ள மதிப்புகளில் செயல்படும். கணித, ஒப்பீடு, தருக்க, மதிப்பிருத்தல் போன்று பல வகை செயற்குறிகள் உள்ளன. மதிப்புகள் மற்றும் மாறிகள் செயற்குறியுடன் பயன்படுத்தும் போது செயலேற்பிகள் என்று அழைக்கப்படுகிறது.

5.7.3.1 கணித செயற்குறிகள்

கணித செயற்குறிகள், இரு செயலேற்பிகள் ஏற்றுக் கொண்டு அதன் மீது கணித செயல்பாடுகளை செய்யும். அவை எளிய முறை கணித செயல்பாடுகளாக பயன்படுகிறது. பெரும்பான்மையாக கணிப்பொறி மொழிகள் இதுபோன்ற செயற்குறிகளை கொண்டிருக்கும், அவை சமன்பாடுகளை பயன்படுத்தி தொடர் கணிப்பீடுகளை செய்யப் பயன்படுகிறது.

பைத்தான் பின்வரும் கணித செயற்குறிகளை கொண்டுள்ளது

செயற்குறி–செயல்பாடு	எடுத்துக்காட்டு	தீர்வு
a=100 , b=10 என மதிப்பாக கொண்டு பின்வரும் கோவையை மதிப்பிடுக		
+ (கூட்டல்)	>>> a + b	110
- (கழித்தல்)	>>>a – b	90
* (பெறுக்கல்)	>>> a*b	1000
/ (வகுத்தல்)	>>> a / b	10.0
% (வகுமீது)	>>> a % 30	10
** (அடுக்கு)	>>> a ** 2	10000
// (முழு எண் வகுத்தி)	>>> a//30 (Integer Division)	3

Chapter 5.indd 59 30-01-2020 19:20:40

குறிமுறை 5.1 கணித செயற்குறியை செயல் விளக்கம்:

```
#Demo Program to test Arithmetic Operators
        a=100
        b=10
        print ("The Sum       = ",a+b)
        print ("The Difference = ",a-b)
        print ("The Product    = ",a*b)
        print ("The Quotient   = ",a/b)
        print ("The Remainder  = ",a%30)
        print ("The Exponent   = ",a**2)
        print ("The Floor Division =",a//30)
#Program End
```

வெளியீடு:

```
        The Sum              = 110
        The Difference       = 90
        The Product          = 1000
        The Quotient         = 10.0
        The Remainder        = 10
        The Exponent         = 10000
        The Floor Division   = 3
```

5.7.3.2 தொடர்புடைய அல்லது ஒப்பீடு செயற்குறிகள்

தொடர்புடைய செயற்குறிகள் ஒப்பீடு செயற்குறிகள் என்றும் அழைக்கப்படுகிறது. இது இரு செயலேற்பிகளுக்கு இடையேயான உறைமுறையை சோதித்தரிய உதவும். சோதனை சரியாக இருந்தால் விடை சரி என்றும், தவறெனில் தவறு என்ற விடையை தரும்.

பைத்தான் பின்வரும் ஒப்பீடு செயற்குறிகளை கொண்டுள்ளது

செயற்குறிகள்	எடுத்துக்காட்டு	தீர்வு
a = 100, b = 35 என மதிப்பாக கொண்டு பின்வரும் கோவையை மதிப்பிடுக		
== (நிகர்)	>>> a==b	தவறு
> (விடப் பெரியது)	>>> a > b	சரி
< (விடச் சிறியது)	>>> a < b	தவறு
>= (விடப் பெரியது அல்லது நிகர்)	>>> a >= b	சரி
<= (விடச் சிறியது அல்லது நிகர்)	>>> a <= b	தவறு
!= (நிகர் அல்லாத)	>>> a != b	சரி

Chapter 5.indd 60 30-01-2020 19:20:40

குறிமுறை 5.2 ஒப்பீடு செயற்குறி சோதித்தல்:

```
#Demo Program to test Relational Operators
        a=int (input("Enter a Value for A:"))
        b=int (input("Enter a Value for B:"))
        print ("A = ",a," and B = ",b)
        print ("The a==b = ",a==b)
        print ("The a > b = ",a>b)
        print ("The a < b = ",a<b)
        print ("The a >= b = ",a>=b)
        print ("The a <= b = ",a<=0)
        print ("The a != b = ",a!=b)
#Program End
```

வெளியீடு:

```
        Enter a Value for A:35
        Enter a Value for B:56
        A =  35  and B  = 56
        The a==b                = False
        The a > b               = False
        The a < b               = True
        The a >= b              = False
        The a <= b              = False
        The a != b              = True
```

5.7.3.3 தருக்க செயற்குறிகள் உள்ளன

பைத்தானில், தருக்க செயற்குறிகள் கொடுக்கப்பட்ட ஒப்பீடு கோவையின் மீது தருக்க செயல்பாடுகளை மேற்கொள்ள பயன்படுகிறது. **and, or** மற்றும் **not** ஆகிய மூன்று தருக்க செயற்குறிகள் உள்ளன.

செயற்குறிகள்	எடுத்துக்காட்டு	முடிவு
a மதிப்பு 97, b மதிப்பு 35 எனக் கொண்டு பின்வரும் தருக்க கோவையை மதிப்பீடு செய்யவும்.		
or	>>> a>b or a==b	சரி
and	>>> a>b and a==b	தவறு
not	>>> not a>b	தவறு (அதாவது) சரி கிடையாது.

குறிமுறை 5.3 தருக்க செயற்குறிகளை சோதித்தல்:

எடுத்துக்காட்டு குறிமுறை	எடுத்துக்காட்டு முடிவு

#Demo Program to test Logical Operators
a=int (input("Enter a Value for A:"))
b=int (input("Enter a Value for B:"))
print ("A = ",a, " and b = ",b)
print ("The a > b or a == b = ",a>b or a==b)
print ("The a > b and a == b = ",a>b and a==b)
print ("The not a > b = ",not a>b)
#Program End
	Enter a Value for A:50
	Enter a Value for B:40
	A = 50 and b = 40
	The a > b or a == b = True
	The a > b and a == b = False
	The not a > b = False

5.7.3.4 மதிப்பிருத்து செயற்குறிகள்

பைத்தானில், = என்பது ஒரு மதிப்பிருத்தல் செயற்குறியாகும். இது மாறிகளுக்கு மதிப்பிருத்த பயன்படுகிறது. a = 5 மற்றும் b = 10 என்று கூற்று a யின் மதிப்பு 5 என்றும், b-ன் மதிப்பு 10 என்று மதிப்பிருந்தும். இவ்விரு மதிப்பிருத்தல் கூற்றினை வேறுவிதமாக a,b=5,10 என்றும் கொடுக்கலாம், இதுவும் வலது பக்கம் இருக்கு 5 மற்றும் 10 என்ற மதிப்பினை a,b என்ற மாறிகளில் முறையே மதிப்பிருத்தும். பைத்தான், +=, -=, *=, /=, %=, **= மற்றும் //= போன்ற கூட்டு செயற்குறிகளையும் ஏற்கும்.

செயற்கூறு	விளக்கம்	எடுத்துக்காட்டு
x=10 எனில்		
=	வலப்பக்கமுள்ள செயலேர்பியை இடப்பக்கமுள்ள மாறிக்கு இருத்தும்	>>> x=10 >>> b=”Computer”
+=	செயலேர்பிகளை கூட்டி வரும் விடையை இடப்பக்கமுள்ள செயலேற்பியில் இருத்தும் அதாவது, x=30	>>> x+=20 # x=x+20
-=	செயலேர்பிகளை கழித்து வரும் விடையை இடப்பக்கமுள்ள செயலேற்பியில் இருத்தும் x=25	>>> x-=5 # x=x-5
*=	செயலேர்பிகளை கழித்து வரும் விடையை இடப்பக்கமுள்ள செயலேற்பியில் இருத்தும் எனவே x=50	>>> x*=5 # x=x*5
/=	செயலேர்பிகளை வகுத்து வரும் விடையை இடப்பக்கமுள்ள செயலேற்பியில் இருத்தும் எனவே, x=5	>>> x/=2 # x=x/2
%=	இரு செயலேர்பிகள் மீது செயல்பட்டு, வகுமீதியை (மீதம்) கணக்கிட்டு விடையை இடப்பக்கமுள்ள செயலேற்பியில் இருத்தும் அதாவது x=1.	>>> x%=3 # x=x%3
**=	செயலேர்பியின் அடுக்குக்குறியை செயற்படுத்தி இடப்பக்கமுள்ள செயலேற்பியில் இருத்தும், அதாவது x=100	>>> x**=2 # x=x**2
//=	செயலேற்பியின் மீது முழு எண் வகுத்தல் மதிப்பிட்டு இடப்பக்க செயலேற்பியில் இருத்தும், x=3	>>> x//=3

Chapter 5.indd 62 30-01-2020 19:20:40

குறிமுறை 5.4 மதிப்பிருத்தல் செயற்குறிகளை சோதித்தல்:

நிரல் குறிமுறை	வெளியீடு

#Demo Program to test Assignment Operators
x=int (input("Type a Value for X : "))
print ("X = ",x)
print ("The x is   =",x)
x+=20
print ("The x += 20 is =",x)
x-=5
print ("The x -= 5 is = ",x)
x*=5
print ("The x *= 5 is = ",x)
x/=2
print ("The x /= 2 is = ",x)
x%=3
print ("The x %= 3 is = ",x)
x**=2
print ("The x **= 2 is = ",x)
x//=3
print ("The x //= 3 is = ",x)
#Program End
	Type a Value for X : 10
	X = 10
	The x is = 10
	The x += 20 is = 30
	The x -= 5 is = 25
	The x *= 5 is = 125
	The x /= 2 is = 62.5
	The x %= 3 is = 2.5
	The x **= 2 is = 6.25
	The x //= 3 is = 2.0

5.7.3.5 நிபந்தனை செயற்குறி

மும்ம செயற்குறி நிபந்தனை செயற்குறி என்றும் அழைக்கப்படுகிறது. இது சமன்பாடுகளின் நிபந்தனையை சரி அல்லது தவறா என்று சோதித்து செயல்படுத்தும், மும்ம செயற்குறி பல வரி if..else கூற்று போல் அல்லாது நிபந்தனைகளை ஒற்றை வரியில் சோதிக்க அனுமதிக்கிறது.

நிபந்தனைச் செயற்குறியின் தொடரியல்,

Variable Name = [on_true] if [Test expression] else [on_false]

எடுத்துக்காட்டு :

```
min= 50 if 49<50 else 70 // min = 50
min= 50 if 49>50 else 70 // min = 70
```

குறிமுறை 5.5 நிபந்தனை செயற்குறி(மும்ம செயற்குறி) சோதித்தல்:

```
# Program to demonstrate conditional operator
        a, b = 30, 20
# Copy value of a in min if a < b else copy b
        min = a if a < b else b
        print ("The Minimum of A and B is ",min)
# End of the Program
```

வெளியீடு:

```
        The Minimum of A and B is  20
```

5.7.4 வரம்புக்குறிகள்

பைத்தான், குறியீடு அல்லது குறியீடுகளின் தொகுப்பை, கோவை, பட்டியல், அகராதி மற்றும் சரங்களில் பயன்படுத்துகிறது. பின்வருவன பைத்தான் வரம்புக்குறிகளாகும்.

(	)	[	]	{	}
,	:	.	‘	=	;
+=	-=	*=	/=	//=	%=
&=	|=	^=	>>=	<<=	**=

5.7.5 நிலைஉருக்கள்

நிலைஉருக்கள் என்பது மாறிகள் அல்லது மாறிலிகளுக்கு வழங்கப்படும் மூல தரவாகும். பைத்தானில் பல்வேறு வகையான நிலைஉருகள் உள்ளன.

1) எண்கள்

2) சரம்

3) பூலியன்

5.7.5.1 எண் நிலைஉருக்கள்:

எண் நிலைஉருக்கள் எண்களை கொண்டிருக்கும். இவற்றை மாற்ற முடியாது. இவை முழு எண், மிதப்பு மற்றும் சிக்கலான என்று 3 வகையாக எண் நிலைஉருக்களாக பிரிக்கப்படுகிறது.

Chapter 5.indd 64 30-01-2020 19:20:40

நிரல் குறிமுறை 5.5 எண் இலக்கங்களை நிரூபித்தல்

```
# Program to demonstrate Numeric Literals
        a = 0b1010                  #Binary Literals
        b = 100                     #Decimal Literal
        c = 0o310                   #Octal Literal
        d = 0x12c                   #Hexadecimal Literal
        print ("Integer Literals :",a,b,c,d)
        #Float Literal
        float_1 = 10.5
        float_2 = 1.5e2
        print ("Float Literals :",float_1,float_2)
        #Complex Literal
        x = 1 + 3.14 j
        print ("Complex Literals :", x)
        Print ("x = ", x , "Imaginary part of x = ", x.imag, "Real part of x = ", x.real)
#End of the Program
```

வெளியீடு:

```
        Integer Literals : 10 100 200 300
        Float Literals : 10.5 150.0
        Complex Literals :
        x = (1.3.14) Imaginary part of x = 3.14 Real part of x = 1.0
```

5.7.5.2 சர நிலையுருக்கள்

பைத்தானில், சர நிலையுருக்கள், குறியுருக்களின் தொடர், மேற்கோள் குறிக்குள் கொண்டிருக்கும். பைத்தானில், சரங்களை ஒற்றை, இரட்டை மற்றும் மூன்று மேற்கோள் குறிகளில் அடைக்கலாம். குறியுரு மதிப்பிருக்கள் ஒற்றை அல்லது இரட்டை மேற்குறியை கொண்டிருக்கும். பல வரி சர நிலையுருவை மூன்று மேற்கோள் ''' ''' குறிக்குள் கொடுக்கப்பட வேண்டும்.

குறிமுறை 5.7 சர நிலையுரு செயல் விளக்கம்:

```
# Demo Program to test String Literals
        strings = "This is Python"
        char = "C"
        multiline_str = '''This is a multiline string with more than one line code.'''
        print (strings)
        print (char)
        print (multiline_str)
# End of the Program
```

வெளியீடு:

```
        This is Python
        C
        This is a multiline string with more than one line code.
```

5.7.5.3 பூலியன் நிலையுருக்கள்

பூலியன் நிலையுருக்கள் சரி அல்லது தவறு ஆகிய இரு மதிப்புகளை கொண்டிருக்கும்.

குறிமுறை 5.8 பூலியன் மதிப்புரு சோதித்தல்:

```
# Demo Program to test String Literals
        boolean_1 = True
        boolean_2 = False
        print ("Demo Program for Boolean Literals")
        print ("Boolean Value1 :",boolean_1)
        print ("Boolean Value2 :",boolean_2)
# End of the Program
```

வெளியீடு:

```
        Demo Program for Boolean Literals
        Boolean Value1 : True
        Boolean Value2 : False
```

5.7.5.4 விடுபடு தொடர்

பைத்தானில், பின்சாய்வுகோடு ("\") ஆகிய சிறப்பு குறியீட்டை **"விடுபடு"** குறியுரு என்று அழைக்கிறோம். இது சில வெற்று இடைவெளியை குறிப்பிட பயன்படுகிறது. "\t" என்பது தத்தல், "\n" என்பது புதியவரி மற்றும் "\r" என்பது நகர்த்தி திரும்பலை (carriage return) குறிக்கிறது. எடுத்துக்காட்டு "It's raining" என்ற செய்தியை பின்வரும் பைத்தான் கட்டளை வெளியிடும்,

>>> print ("It\'s raining")

It's raining

பைத்தான் பின்வரும் விடுபடு தொடர்களை கொண்டுள்ளது.

விடுபடு தொடர் குறியுரு	விளக்கம்	எடுத்துக்காட்டு	வெளியீடு
\\	பின்சாய்வுக்கோடு	>>> print("\\test")	\test
\’	ஒற்றை மேற்கோள்	>>> print("Doesn\'t")	Doesn't
\”	இரட்டை மேற்கோள்	>>> print("\"Python\"")	"Python"
\n	புதிய வரி	print("Python","\n","Lang..")	Python Lang..
\t	தத்தல்	print("Python","\t","Lang..")	Python Lang..

5.8 பைத்தான் தரவு வகைகள்

பைத்தானில் அனைத்து தரவு மதிப்புகளும் பொருளாக கருதப்படுகிறது. ஒவ்வொரு பொருள் அல்லது மதிப்புகள் பல்வேறு வகைப்படும். பைத்தானில், உள்ளினைந்த அல்லது அடிப்படை தரவு வகைகள் எண்கள், சரங்கள், பூலியன், டூபுல்ஸ், லிஸ்ட் மற்றும் டிக்ஸ்னரிகள் ஆகும்.

Chapter 5.indd 66 30-01-2020 19:20:40

5.8.1 எண் தரவு வகை

பைத்தானில் உள்ளிணைந்த எண் பொருள் முழு எண், மிதப்பு புள்ளி எண்கள் மற்றும் சிக்கல் எண்களை கொண்டுள்ளது.

முழு எண் தரவு பதின்ம எண் (Decimal), எண்ணிலை (octal) எண் அல்லது பதினாறு நிலை எண்ணாக (Hexadecimal) இருக்கலாம். எண்ம எண்யை குறிப்பதற்கு 0 (பூஜ்ஜியம் மற்றும் சிறிய எழுத்தில் o பயன்படுகிறது) (பதினாறு நிலை எண்ணை குறிப்பதற்கு 0X அல்லது 0x (பூஜ்ஜியம் மற்றும் பெரிய அல்லது சிறிய எழுத்துக்களில் X பயன்படுகிறது) மற்றும் L (பெரிய எழுத்து) பெரிய முழு எண்ணை குறிக்க பயன்படுகிறது.

எடுத்துக்காட்டு:

```
102, 4567, 567            # Decimal integers
0o102, 0o876, 0o432       # Octal integers
0X102, 0X876, 0X432       # Hexadecimal integers
34L, 523L                 # Long decimal integers
```

மிதப்பு புள்ளி தரவு, தசம புள்ளியை கொண்ட தசம எண்களாக குறிப்பிடப்படுகின்றன. எக்ஸ்போனன்ட் தரவு வகை, தசம எண் பகுதி, தசம புள்ளி, எக்ஸ்போனன்ட் பகுதியை தொடர்ந்து ஒன்று அல்லது பல இலக்கங்களை கொண்டிருக்கும்.

எடுத்துக்காட்டு:

```
123.34, 456.23, 156.23    # Floating point data
12.E04, 24.e04            # Exponent data
```

சிக்கல் எண்கள் இரு மிதப்பு புள்ளி மதிப்புகளை ரியல் மற்றும் இம ஜனரி பகுதிகளாக கொண்டிருக்கும்.

5.8.2 பூலியன் தரவு வகை

பூலியன் தரவுகள், சரி அல்லது தவறு என இரு மதிப்புகளில் ஒன்றை கொண்டிருக்கும்.

எடுத்துக்காட்டு:

```
Bool_var1=True
Bool_var2=False
```

5.8.3 சர தரவு வகை

சர தரவு வகை ஒற்றை அல்லது இரட்டை அல்லது மூன்று மேற்கோள் குறிக்குள் கொடுக்கப்பட வேண்டும்.

எடுத்துக்காட்டு:

```
Char_data = 'A'
String_data= "Computer Science"
Multiline_data= """String data can be enclosed with single quote or double quote or triple quote."""
```

நினைவில் கொள்க

- கைடோ வான் ரோசும் என்பவரால் உருவாக்கப்பட்ட ஒரு எளிய முறை நிரலாக்க மொழி பைத்தான்.
- பைத்தான் ஷெல், ஊடாடும் முறை (Interactive mode) மற்றும் ஸ்கிரிப்ட் முறை என்று இரு வழிகளில் செயல்படுத்தப்படுகிறது.
- பைத்தானில் வெற்று இடைவெளியைக் (இடைவெளி மற்றும் தத்தல்) கொண்டு நிரலின் தொகுதிகள் பிரிக்கப்படுகிறது.
- வில்லைகள், வெற்று இடைவெளியால் பிரிக்கப்பட வேண்டும்.
- பயனர் எதிர்பார்க்கும் பணியினை நிரல் செயல்படுத்த உள்ளீடு மற்றும் வெளியீடு வசதி மூலம் பயனர் நிரலுடன் தொடர்பு கொள்ளலாம்.
- பைத்தான் ஒவ்வொரு தருக்க வரியையும் வில்லையாக பிரிக்கும்.
- சிறப்புச் சொற்கள் எனப்படும் முக்கிய சொற்கள் மூலம் பைத்தான் மொழிப்பெயர்ப்பி நிரலின் அமைப்பை அறிந்து கொள்கிறது.

மதிப்பீடு

பகுதி–அ

சரியான விடையைத் தேர்ந்தெடுக்கவும்: (1 மதிப்பெண்)

1. பைத்தானை உருவாக்கியவர் யார்?

அ) ரிட்ஸீ | ஆ) கைடோ வான் ரோஷும்

இ) பில் கேட்ஸ் | ஈ) சுந்தர் பிச்சை

2. இவற்றுள் எந்த தூண்டு குறி நிரல் பெயர்ப்பி கட்டளைகளை ஏற்று கொள்ள தயார் நிலையில் இருப்பதை குறிக்கிறது?

அ) >>> | ஆ) <<<

இ) # | ஈ) <<

3. பின்வரும் எந்த சாவி சேர்மானம் ஓர் புதிய பைத்தான் நிரலை உருவாக்கப்பயன்படுகிறது.

அ) Ctrl+C | ஆ) Ctrl+F

இ) Ctrl+B | ஈ) Ctrl+N

4. பின்வரும் எந்த குறியுரு பைத்தான் நிரலில் குறிப்புகளை உள்ளீடு செய்ய பயன்படுகிறது?

அ) # | ஆ) &

இ) @ | ஈ) $

Chapter 5.indd 68 30-01-2020 19:20:40

5. எந்த குறி ஒன்றுக்கு மேற்பட்ட உறுப்புகளை ஒற்றை வரியில் அச்சிடும்

அ) அரைப்புள்ளி   ஆ) டாலர்

இ) காற்புள்ளி   ஈ) முக்காற்புள்ளி

6. பின்வரும் எது வில்லைகள் கிடையாது?

அ) நிரல்பெயர்ப்பி   ஆ) குறிப்பெயர்கள்

இ) சிறப்புச் சொற்கள்   ஈ) செயற்குறிகள்

7. பின்வருவனவற்றில் எது பைதான் சிறப்புச் சொல் கிடையாது?

அ) break   ஆ) while

இ) continue   ஈ) operators

8. எந்த செயற்குறியை ஒப்பீடு செயற்குறி என்று அழைக்கப்படுகிறது?

அ) கணக்கீடு   ஆ) தொடர்புடைய

இ) தருக்க   ஈ) மதிப்பிருத்தல்

9. பின்வருவனவற்றில் எது தருக்க செயற்குறி கிடையாது?

அ) and   ஆ) or

இ) not   ஈ) like

10. எந்த செயற்குறி நிபந்தனை செயற்குறி என்று அழைக்கப்படுகிறது?

அ) மும்ம செயற்குறி   ஆ) தொடர்புடைய

இ) தருக்க   ஈ) மதிப்பிருத்தல்

பகுதி–ஆ

Very short answers (2 மதிப்பெண்)

11. பைத்தான் நிரலினை சோதிக்க எத்தனை வகை முறைமைகள் உள்ளன?
12. வில்லைகள் பற்றி சிறு குறிப்பு வரைக.
13. பைத்தானில் உள்ள பல்வேறு செயற்குறிகள் யாவை?
14. குறிப்பெயர்கள் என்றால் என்ன? குறிப்பெயர்கள் வகைகள் யாவை?
15. எக்ஸ்போனன்ட் தரவு பற்றி குறிப்பு வரைக.

பகுதி–இ

Short answers (3 மதிப்பெண்)

16. கணித செயற்குறிகள் பற்றி குறிப்பு வரைக. எடுத்துக்காட்டு தருக.
17. பைத்தானில் மதிப்பிருத்தல் செயற்குறிகள் என்றால் என்ன?
18. மும்ம செயற்குறியை எடுத்துக்காட்டுடன் எழுதுக.

19. விடுபடு வரிசைப்பற்றி குறிப்பு எழுதி எடுத்துக்காட்டு தருக.

20. சரநிலையுரு என்றால் என்ன?

பகுதி–ஈ

Essay answers (5 மதிப்பெண்)

21. செயல்முறை ஸ்கிரிப்ட் முறைமை நிரலாக்கம் பற்றி எழுதுக.

22. input() மற்றும் print() செயற்கூறுகள் பற்றி விளக்கு.

23. பைத்தானில் உள்ள வில்லைகள் பற்றி எழுதுக.

Chapter 5.indd 70 30-01-2020 19:20:40

அலகு II

பாடம் 6

கட்டுப்பாட்டு கட்டமைப்புகள்

HSQ5E2

கற்றலின் நோக்கங்கள்

இந்தப் பாடப்பகுதியைக் கற்றபின் மாணவர்கள் அறிந்துக் கொள்வது:

- பைத்தான் மொழியில் பல்வேறு பாய்வுக்கட்டுப்பாடுகளைப் பற்றிய அறிவைப் பெறுதல்.
- நிபந்தனை அமைப்பை பயன்படுத்தி நிரலின் பாய்வு செயல் திறனை எவ்வாறு மேம்படுத்துவது என்பதை கட்டளை அமைப்பு மூலம் அறிதல்.
- பன்முறை செயல் அமைப்பை அல்லது மடக்கை பயன்படுத்தி நிரல் பகுதியை குறிப்பிட்ட எண்ணிக்கை வரை அல்லது நிபந்தனை நிறைவேற்றப்படும் வரை திரும்ப செய்யும் குறிமுறையை உருவாக்குதல்.

6.1 அறிமுகம்

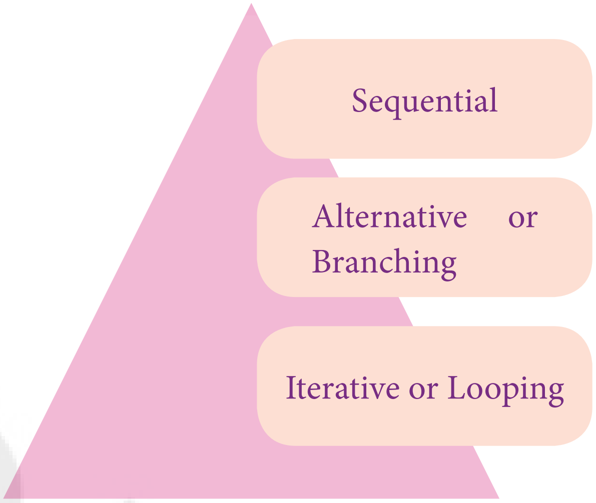
நிரல்கள் கூற்றுகளின் தொகுதியை கொண்டிருக்கும். இந்த கூற்றுகளே விடைகளைக் கொடுக்கும் இயக்கப் பகுதிகளாகும். பொதுவாக, கூற்றுகள் வரிசையாக, அதாவது கூற்றுகள் ஒன்றன் பின் ஒன்றாக நிறைவேற்றப்படும். ஏதேனும் சில சமயங்களில் நிகழ் நேர நிரல்களை இயக்கும் போது நிரலின் ஒரு பகுதியை அல்லது கூற்றுகளின் தொகுதியை நிறைவேற்றாமல் விட்டு விட்டு நிபந்தனையின் அடிப்படையில் நிரலின் மற்றொரு பகுதியை இயக்க நேரிடும். இதற்கு **"மாற்று"** அல்லது **"கிளைப்பிரிப்பு"** என்று பெயர். மேலும், கூற்றுகளில் உள்ள ஒரு தொகுதியை பல தடவை நிறைவேற்றவேண்டி இருக்கும். இதற்கு **பன்முறைசெயல்** அல்லது **மடக்கு** என்று பெயர். இந்தப் பாடப் பகுதியில் பைத்தானில் பல்வேறு கட்டுப்பாட்டு கட்டமைப்புகள் அதன் கட்டளை அமைப்புகள் மற்றும் அவற்றைப் பயன்படுத்தி நிரல்கள் உருவாக்குதல் மற்றும் கற்றல் ஆகியவற்றைக் காண்போம்.

6.2 கட்டுப்பாட்டு கட்டமைப்புகள்

கட்டுப்பாட்டு நிரலின் ஒரு பகுதியில் இருந்து இன்னொரு பகுதிக்கு தாவுவதற்கு காரணமான நிரல் கூற்றுகள் **கட்டுப்பாட்டு கட்டமைப்பு** அல்லது **கட்டுப்பாட்டு கூற்றுகள்** எனப்படும். ஏற்கனவே, C++ல் கற்றதைப் போல் இந்த கட்டுப்பாட்டுக் கூற்றுகள் கலப்புக் கூற்றுகளாகும். செயல் முறையின் நிலையைப் பொறுத்து, செயல்முறையின் கட்டுப்பாட்டு பாய்வை அல்லது நிரலின் கட்டுப்பாட்டு பாய்வை மாற்றும்.

பைத்தானில் மூன்று முக்கிய வகையான கட்டுப்பாட்டு அமைப்புகள் உள்ளன.

Chapter 6.indd 71 30-01-2020 19:24:37



6.2.1 வரிசை முறை கூற்றுகள் (Sequential Statements)

ஒன்றன் பின் ஒன்றாக நிறைவேற்றப்படும் கூற்றுகளின் வரிசையைக் கொண்டது வரிசை முறை கூற்று ஆகும். உன்னுடைய பெயர், முகவரி மற்றும் தொலைபேசி எண்ணை அச்சிடும் குறிமுறை, வரிசைமுறை கூற்றுக்கு எடுத்துக்காட்டு ஆகும்.

எடுத்துக்காட்டு 6.1

```
# உன்னுடைய பெயர் மற்றும் முகவரியை அச்சிடும் நிரல்- வரிசைமுறை கூற்றுக்கு எடுத்துக்காட்டு
print ("Hello! This is Shyam")
print ("43, Second Lane, North Car Street, TN")
```

வெளியீடு :

```
Hello! This is Shyam
43, Second Lane, North Car Street, TN
```

6.2.2 மாற்று அல்லது கிளைப் பிரிப்புகூற்று (Alternative or Branching Statement)

நம்முடைய அன்றாட வாழ்வில் நாம் பல்வேறு முடிவுகளை எடுக்க வேண்டியுள்ளது. நமது நோக்கத்தை அடைய மாற்றுப் பாதையைத் தேர்ந்தெடுக்கிறோம். தினசரி நாம் செல்லும் சாலை தடை செய்யப்பட்டிருந்தால் மாற்றுப் பாதையை பயன்படுத்தி, நாம் சென்றடைய வேண்டிய இடத்திற்கு சென்று சேர்ந்திடுவோம். இந்த முடிவெடுத்தலை நாம் மாற்று அல்லது கிளைப் பிரிப்பு கூற்று மூலம் கற்க இருக்கின்றோம். கொடுக்கப்பட்ட எண் நேர்மறை எண் அல்லது எதிர்மறை எண்ணா அல்லது ஒற்றைப்படை எண்ணா அல்லது இரட்டைப்படை எண்ணா என்பதை கண்டறிவதற்கு மாற்று அல்லது கிளைப் பிரிப்பு கூற்று பயன்படுகிறது.

பைத்தானில் கீழ்க்கண்ட மாற்று (அல்லது) கிளைப்பிரிப்பு கூற்று வகைகள் உள்ளன:

- Simple if கூற்று
- if..else கூற்று
- if..elif கூற்று

6.2.2.1 Simple if கூற்று

Simple if கூற்று என்பது அனைத்து தீர்மானிப்பு கூற்றுகளிலும் மிக எளிதான கூற்றாகும். நிபந்தனையானது ஒப்பீட்டு கோவையாகவோ அல்லது தருக்க சேவையாகவோ இருத்தல் வேண்டும்.

Chapter 6.indd 72 30-01-2020 19:24:37

பொதுவடிவம்:

```
if <condition>:
        statements-block1
```

மேலே உள்ள கூற்றில் நிபந்தனை சரி என்றால் **statements-block1** நிறைவேற்றப்படும்.

எடுத்துக்காட்டு 6.2:

```
# வயதை சரி பார்த்து வாக்களிக்க தகுதியா என அச்சிடும் நிரல்
x=int (input("Enter your age :"))
if x>=18:
        print ("You are eligible for voting")
```

வெளியீடு1:

```
Enter your age :34
You are eligible for voting
```

வெளியீடு 2:

```
Enter your age :16
>>>
```

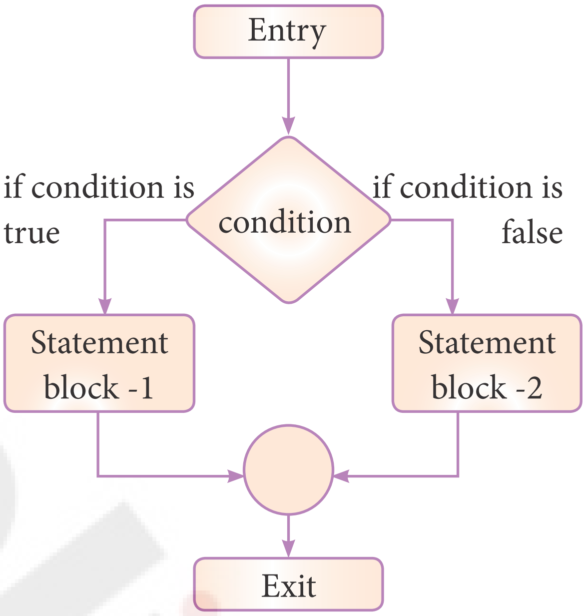
இரண்டாவது வெளியீட்டில் வெளியீடு அச்சிடப்படவில்லை, பைத்தானில் **தூண்டு குறி** மட்டுமே காண்பிக்கிறது. ஏனென்றால், நிபந்தனை தவறாகும் போது நிரலில் என்ன செய்ய வேண்டும் என்பது கொடுக்கப்படவில்லை.

6.2.2.2 if..else கூற்று

if .. else கூற்றானது சரி தொகுதி மற்றும் தவறு தொகுதி இரண்டையுமே சரிபார்ப்பதற்கான கட்டுப்பாட்டை வழங்குகிறது. **'if..else'** கூற்றின் தொடரியல்.

தொடரியல்:

```
if <condition>:
    statements-block 1
else:
    statements-block 2
```



படம் 6.1 *if..else* கூற்றின் செயல்பாடு

if..else கூற்று இரு மாற்று வழிகளை வழங்குகிறது. நிபந்தனையானது எந்த தொகுதியை நிறைவேற்றப்பட வேண்டும் என்பதைத் தீர்மானிக்கிறது.

எடுத்துக்காட்டு 6.3:

#உள்ளிடப்பட்ட எண் ஒற்றைப்படை அல்லது இரட்டைப்படை எண்ணா என்பதைக் கண்டறியும் நிரல்

```
a = int(input("Enter any number :"))
if a%2==0:
        print (a, " is an even number")
else:
        print (a, " is an odd number")
```

வெளியீடு 1:

```
Enter any number :56
56  is an even number
```

வெளியீடு 2:

```
Enter any number :67
67  is an odd number
```

மேற்கண்ட நிரலுக்கு மாற்று முறையில் வேறு மாதிரியாக பைத்தானில் எழுத முடியும். முழுமையான **if..else** கூற்று பின்வருமாறு எழுதலாம்:

தொடரியல்:

variable = variable1 if condition else variable 2

Chapter 6.indd 74 30-01-2020 19:24:37

குறிப்பு:

if ல் குறிப்பிடப்பட்டுள்ள நிபந்தனை பரிசோதிக்கப்படும் நிபந்தனை சரி எனில் மாறி1-ன் (variable1) மதிப்பு, இடது பக்கத்திலுள்ள மாறியில் இருத்தப்படும். இல்லையெனில், மாறி2-ன் (variable2) மதிப்பு இருத்தப்படும்.

எடுத்துக்காட்டு 6.4: # உள்ளிடப்பட்ட எண் ஒற்றைப்படை அல்லது இரட்டைப்படை எண்ணா என்பதைக் கண்டறியும் நிரல் (மாற்று முறையைப் பயன்படுத்தி)

```
a = int (input("Enter any number :"))
x="even" if a%2==0 else "odd"
print (a, " is ",x)
```

வெளியீடு 1:

```
Enter any number :3
3  is  odd
```

வெளியீடு 2:

```
Enter any number :22
22  is  even
```

6.2.2.3 பின்னலான if..elif...else கூற்று:

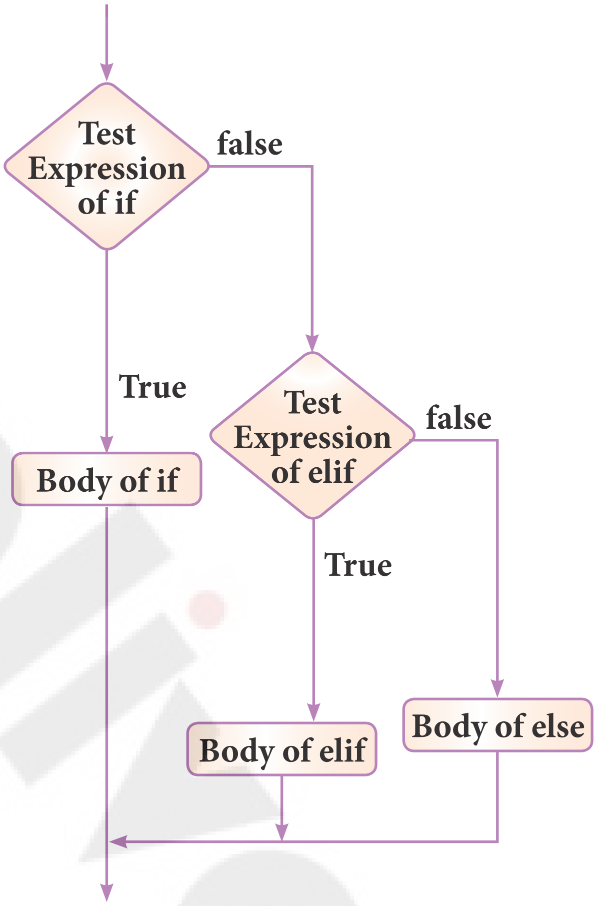
if கூற்றுகளைத் தொடர் கூற்றுகளாக அமைக்க விரும்பும் போது 'else' பகுதிக்குப் பதிலாக 'elif' பகுதி பயன்படுத்தலாம்.

தொடரியல் :

```
if <condition-1>:
    statements-block 1
elif <condition-2>:
    statements-block 2
else:
    statements-block n
```

மேற்கண்ட if..elif..else பொதுவடிவத்தில், condition-1 பரிசோதிக்கப்படும். condition -1 சரி எனில் statements-block 1 செயல்பாட்டுத் condition-1 நிறைவேற்றப்படும். இல்லையெனில் கட்டுப்பாடு, condition-2யைப் பரிசோதிக்கும். condition-2 சரி எனில் (statements-block-2) நிறைவேற்றப்படும். இல்லையெனில் else பகுதியில் உள்ள (statements-block n) நிறைவேற்றப்படும்.

Chapter 6.indd 75 30-01-2020 19:24:37



படம் *6.2 if..elif..else* கூற்றின் செயல்பாடு

'elif' பகுதி **if..else-if..else** கூற்றுகளை இணைக்கு ஒன்றிணைத்து ஒரே **if..elif…else**. 'elif' ன் விரிவாக்கம் 'else if'. 'if' கூற்றில் பயன்படுத்தப்படும் 'elif' பகுதிக்கு வரம்பு இல்லை, ஆனால் else பகுதி பயன்படுத்தும்போது கூற்றின் இறுதியில் பயன்படுத்த வேண்டும்.

குறிப்பு

C++ல் படித்த பின்னலான **if** கூற்றுக்கு நிகரானது பைத்தானில் உள்ள **if..elif…else** கூற்று.

Chapter 6.indd 76 30-01-2020 19:24:37

எடுத்துக்காட்டு 6.5:

கீழே கொடுக்கப்பட்ட நிபந்தனையின் படி மாணவர்கள் பெற்ற தர வரிசையை அச்சிட வேண்டிய நிரல்

சராசரி	தரம்
>=80 and above	A
>=70 and <80	B
>=60 and <70	C
>=50 and <60	D
Otherwise	E

```
#பின்னலான if கூற்றின் பயன்பாட்டை விளக்கும் நிரல்
m1=int (input("Enter mark in first subject : "))
m2=int (input("Enter mark in second subject : "))
avg= (m1+m2)/2
if avg>=80:
        print ("Grade : A")
elif avg>=70 and avg<80:
        print ("Grade : B")
elif avg>=60 and avg<70:
        print ("Grade : C")
elif avg>=50 and avg<60:
        print ("Grade : D")
else:
        print ("Grade : E")
```

வெளியீடு 1:

```
Enter mark in first subject : 34
Enter mark in second subject : 78
Grade : D
```

வெளியீடு 2 :

```
Enter mark in first subject : 67
Enter mark in second subject : 73
Grade : B
```

குறிப்பு

கொடுக்கப்பட்ட எடுத்துக்காட்டில் if-பகுதிக்கும் நான்கு இடைவெளிகள் உள் தள்ளப்பட்டுள்ளது. இது பைத்தானில் பொதுவாக உள் தள்ளப்பட வேண்டிய அளவாகும். பிற நிரலாக்க மொழிகளில் உள் தள்ளல் என்பது நிரலை அழகாக காண்பிப்பதற்காக பயன்படுகிறது. ஆனால் பைத்தானில் எந்த தொகுதியினுடைய கூற்றுகள் என்று குறிப்பதற்காக உள் தள்ளல் அவசியம் தேவைப்படுகிறது.

எடுத்துக்காட்டு 6.5.a:

```
# if கூற்றில் 'in' மற்றும் 'not in' பயன்பாட்டை விளக்கும் நிரல்
ch=input ("Enter a character :")
# to check if the letter is vowel
if ch in ('a', 'A', 'e', 'E', 'i', 'I', 'o', 'O', 'u', 'U'):
        print (ch,' is a vowel')
# to check if the letter typed is not 'a' or 'b' or 'c'
if ch not in ('a', 'b', 'c'):
        print (ch,' the letter is not a/b/c')
```

வெளியீடு1:

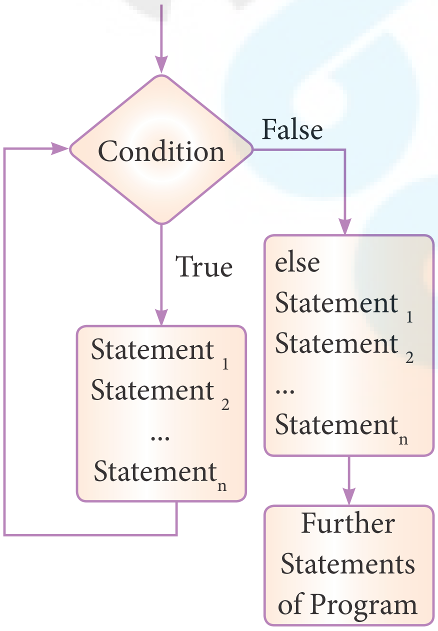
```
Enter a character :e
e is a vowel
```

வெளியீடு 2:

```
Enter a character :x
x the letter is not a/b/c
```

6.2.3. பன்முறைச் செயல் அல்லது மடக்கு அமைப்பு

பன்முறைச் செயல் அல்லது மடக்கு என்பது பயனர் விரும்பும் குறிமுறைத் தொகுதியை குறிப்பிட்ட எண்ணிக்கை வரை அல்லது நிபந்தனை நிறைவேற்றப்படும் வரை இயக்குவதாகும். ஒரு கூற்று அல்லது பல கூற்றுகளை பல முறை நிறைவேற்ற கூற்று அனுமதிக்கிறது.



WE6I7K

படம் 6.3 மடக்கு அமைப்பு எவ்வாறு இயங்குகிறது என்பதை விளக்கும் படம்

பைத்தான் இரண்டு வகையான மடக்கு அமைப்புகளை வழங்குகிறது:

- while மடக்கு
- for மடக்கு

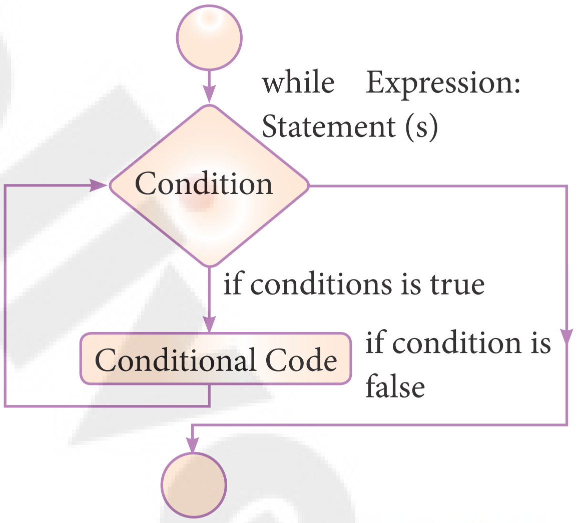
Chapter 6.indd 78 30-01-2020 19:24:37

6.2.3.1 while மடக்கு

பைத்தானில் **while** மடக்கு பின்வரும் தொடரியல் கொண்டிருக்கும்.

தொடரியல்:

while <condition>:
statements block 1
[else:
statements block2]



படம் *6.4* மடக்கின் செயல்பாடு

while மடக்கில், நிபந்தனையானது ஏதாவது ஒரு தகுதியான பூலியன் கோவை ஆகும். இது சரி அல்லது தவறு என்ற மதிப்பை தரும். இதன் **else பகுதி கட்டாயப் பகுதி அல்ல, நிபந்தனை சரி என்று இருக்கும் வரை, செயல்பாட்டுத் தொகுதி 1, (statements block 1) நிறைவேற்றப்படும்**. **else** பகுதி **எழுதப்பட்டிருந்தால் நிபந்தனை தவறு** எனில் **else பகுதி நிறைவேற்றப்படும்**. **while** மடக்கானது நுழைவு சோதிப்பு மடக்கு என்பதை நினைவில்கொள்க. மடக்கின் உள்ளே நுழையும் போது நிபந்தனை தவறு என்றால் மடக்கில் உள்ள கூற்றுகள் ஒரு முறை கூட நிறைவேற்றப்படாது.

எடுத்துக்காட்டு 6.6: while மடக்கை பயன்படுத்தி 10லிருந்து 15 வரை அனைத்து எண்களையும் அச்சிடுவதை விளக்கும் நிரல்.

```
i=10                    # intializing part of the control variable
while (i<=15):          # test condition
    print (i,end='\t')  # statements - block 1
    i=i+1               # Updation of the control variable
```

வெளியீடு:

```
10    11    12    13    14    15
```

கட்டுப்பாட்டு மாறியான in் தொடக்க மதிப்பு 10 ஆகும். i<=15 என்ற சோதனை பரிசோதிக்கப்பட்டு i-ன் மதிப்பு சரியாக இருந்தால் **i** ன் மதிப்பு அச்சிடப்படும். பிறகு, i ன் மதிப்பு i=i+1 என புதுப்பிக்கப்படும் (இதை i+=1 என குறுக்கு வழி மதிப்பிருந்து செயற்குறியை பயன்படுத்தியும் எழுதலாம். i=16 என ஆகும் போது நிபந்தனைதவறாகும், அப்பொழுது மடக்கானது நிறுத்தப்படும்.

குறிப்பு

print கூற்று **end** மற்றும் **sep** அளபுருக்களை கொண்டிருக்கும். **end** அளபுருவை விடுபடு வரிசையை கொடுக்க விரும்பும்போது ('**\t**'தத்தல், '**\n**' புதிய வரி மற்றும் பல) பயன்படுத்தலாம். **sep** அளபுருசிறப்பு குறியீடுகளான, (காற்புள்ளி) ;(அரைப்புள்ளி) போன்ற மதிப்புகளை பிரிக்க பயன்படும் **print**() வடிவூட்டல் விருப்பங்களை இதற்கு முன் பாடத்தில் படித்ததை நினைவு கூறுக.

பின்வரும் எடுத்துக்காட்டு **while** மடக்கில் **else** பயன்பாட்டை காட்டுகிறது.

எடுத்துக்காட்டு 6.7: # else பகுதியுடன் கூடிய while மடக்கை விளக்கும் நிரல்

```
i=10                                      # intializing part of the control variable
while (i<=15):                            # test condition
        print (i,end='\t')                # statements - block 1
        i=i+1                             # Updation of the control variable
else:
      print ("\nValue of i when the loop exit ",i)
```

வெளியீடு: 1

```
10    11    12    13    14    15
Value of i when the loop exit  16
```

6.2.3.2 for மடக்கு

for மடக்கு சுலபமாக பயன்படுத்தக் கூடிய ஓர் மடக்காகும். இது ஒரு நுழைவு சோதிப்பு மடக்காகும். நிபந்தனை முதலிலேயே சோதிக்கப்பட்டு சரி எனில் மடக்கின் உடற்பகுதி செயல்பாட்டுத் தொகுதியை நிறைவேற்றப்படும். இல்லையெனில் மடக்கு நிறைவேறாமல் வெளியேறும்.

தொடரியல்:

```
for counter_variable in sequence:
        statements-block 1
[else:                          # optional block
        statements-block 2]
```

தொடரியலில் குறிப்பிட்டுள்ள தொடரியல் மாறியானது (counter_variable), C++-ல் உள்ள for மடக்கில் பயன்படுத்தும் கட்டுப்பாட்டு மாறியை ஒத்ததாகும். வரிசை என்பது தொடக்க, இறுதி மற்றும் மிகுப்பு மதிப்புகளைக் குறிப்பதாகும். பொதுவாக, பைத்தானில் for மடக்கில் வரிசையில் உள்ள தொடக்க, இறுதி, மதிப்புகளைக் குறிப்பதற்காக range( ) செயற்கூறு பயன்படுகிறது. range() செயற்கூறு start முதல் stop-1 வரையிலான மதிப்பு பட்டியலை உருவாக்குகிறது.

Chapter 6.indd 80 30-01-2020 19:24:38

range ()ன் தொடரியல்:

range(start,stop,[step])

இதில்,

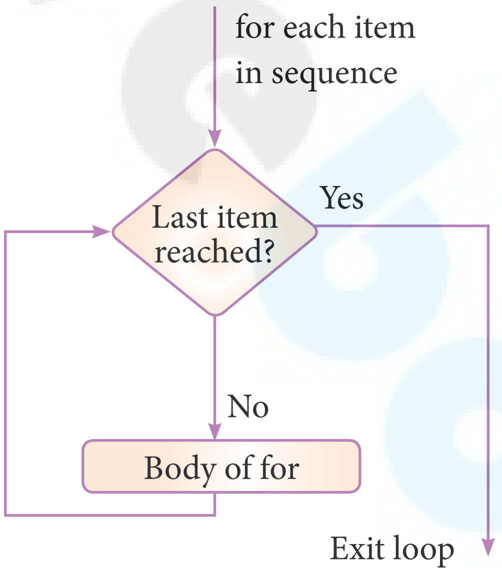
start – தொடக்க மதிப்பைக் குறிக்கும்

stop – இறுதி மதிப்பைக் குறிக்கும்

step – மிகுப்பு மதிப்பைக் குறிக்கும். இது விருப்பப் பகுதியாகும்.

எடுத்துக்காட்டு 6.8: range() எடுத்துக்காட்டுகள் :

range (1,30,1)	*range*-ன் மதிப்புகள் *1* முதல் தொடங்கி *29* வரை
range (2,30,2)	*range()* மதிப்புகள் *2* முதல் *28* வரை
range (30,3,-3)	*range()* மதிப்புகள் *30* முதல் *6* வரை
range (20)	*20* என்ற மதிப்பை இறுதி மதிப்பாக எடுத்துக்கொண்டு (அல்லதுமேல்வரம்பு) *0* முதல் *19* வரையிலான மதிப்புகளைக் கொடுக்கும் (*range()* எப்பொழுதும் *stop* மதிப்பு-*1* வரை மதிப்பைக் கொடுக்கும் என்பதை நினைவில்கொள்க).



படம் *6.5 for* மடக்கின் செயல்பாடு

எடுத்துக்காட்டு 6.9: #for மடக்கு பயன்படுத்தி ஒற்றை இலக்க இரட்டைப் படை எண்ணை அச்சிட விளக்கும் நிரல்

```
for i in range (2,10,2):
        print (i, end=' ')
```

வெளியீடு:

2 4 6 8

Chapter 6.indd 81 30-01-2020 19:24:38

பின்வரும் நிரல் **for** மடக்கின் **else** பகுதியை விளக்கும்

எடுத்துக்காட்டு 6.10 : #for மடக்கின் else பகுதியைப் பயன்படுத்தி, ஒற்றை இலக்க இரட்டைப் படை எண்ணை அச்சிட விளக்கும் நிரல்

```
for i in range(2,10,2):
        print (i,end=' ')
else:
        print ("\nEnd of the loop")
```

வெளியீடு:

```
        2 4 6 8
        End of the loop
```

குறிப்பு

பைத்தானில், மடக்கு மற்றும் பிற கட்டுப்பாட்டு கூற்றுகளில் உள் தள்ளல் மிக முக்கியமானதாகும். C, C++ போன்ற மொழிகளில் நெறிவு அடைப்புக் குறியைப்{} பயன்படுத்தி தொகுதிகளை உருவாக்குவதைப் போல, பைத்தான் உள்தள்ளல் மூலம் தொகுதிகளையும் துணை தொகுதிகளையும் உருவாக்குகிறது.

range()யைப் பயன்படுத்தி, 1 முதல் 100 வரையிலான எண்களின் கூட்டுத் தொகையைக் கண்டுபிடிப்பதை விளக்கும் மற்றொரு நிரல் கீழே கொடுக்கப்பட்டுள்ளது.

எடுத்துக்காட்டு 6.11: # 1 முதல் 100 வரையிலான எண்களின் கூட்டுத் தொகையை கணக்கிடும் நிரல்

```
n = 100
sum = 0
for counter in range(1,n+1):
        sum = sum + counter
print("Sum of 1 until %d: %d" % (n,sum))
```

மேலேயுள்ள குறியீட்டில், n-ன் தொடக்க மதிப்பு 100, sum-ன் மதிப்பு தொடரியல் மாறியின் மதிப்புடன் கூட்டி (சேர்த்து) sumல் சேமிக்கப்படுகிறது. for மடக்குச் செயலானது ஒன்றிலிருந்து தொடங்கி இறுதி வரம்பு-1 மதிப்பு வரை நிறைவேற்றப்படும் (அதாவது nன் மதிப்பு 100 ஆகும். அதனால் இந்த மடக்கு, 1 முதல் 99 மதிப்பு வரை மட்டுமே நிறைவேற்றப்படும். இறுதி வரம்பை n+1 என்று அமைப்பதனால் மடக்கின் இறுதி வரம்பு 100 என்ற எண் வரை செயல்படுத்த முடியும்.)

வெளியீடு:

```
        Sum of 1 until 100: 5050
```

குறிப்பு

சரம், பட்டியல், அகராதி முதலியனவற்றில் இருந்தும் **range()** செயற்கூறு மதிப்புகளை எடுத்துக்கொள்ளும். இவற்றைப் பற்றி அடுத்து வரும் படங்களில் படிக்கலாம்.

பின்வரும் எடுத்துக்காட்டு, range() செயற்கூறில் சரத்தின் பயன்பாட்டை விளக்குவதாகும்.

எடுத்துக்காட்டு 6.12: #for மடக்கில் range() செயற்கூறில் சரத்தின் பயன்பாட்டை விளக்கும் நிரல்.

```
for word in 'Computer':
    print (word,end=' ')
else:
    print ("\nEnd of the loop")
```

வெளியீடு

```
Computer
End of the loop
```

6.2.3.3 பின்னலான மடக்கு அமைப்பு:

ஒரு மடக்கின் உள்ளே மற்றொரு மடக்கு இடம் பெற்றிருந்தால் அது பின்னலான மடக்கு அமைப்பாகும். **while** மடக்கின் உள்ளே மற்றொரு **while** மடக்கு, **for மடக்கின்** உள்ளே மற்றொரு **for** மடக்கு, **for** மடக்கினுள்ளே **while** மற்றும் **while** மடக்கினுள்ளே **for** மடக்கு என இது போன்ற பின்னலான மடக்குகளை உருவாக்கலாம்.

பின்வரும் எடுத்துக்காட்டானது, **for** மடக்கைப் பயன்படுத்தி பின்வரும் அமைப்பில் எண்களை அச்சிடுவதை விளக்குவதாகும்.

```
1
1   2
1   2   3
1   2   3   4
1   2   3   4   5
```

எடுத்துக்காட்டு 6.13: # பின்னலான மடக்கு –for மடக்கின் உள்ளே while மடக்கை விளக்கும் நிரல்

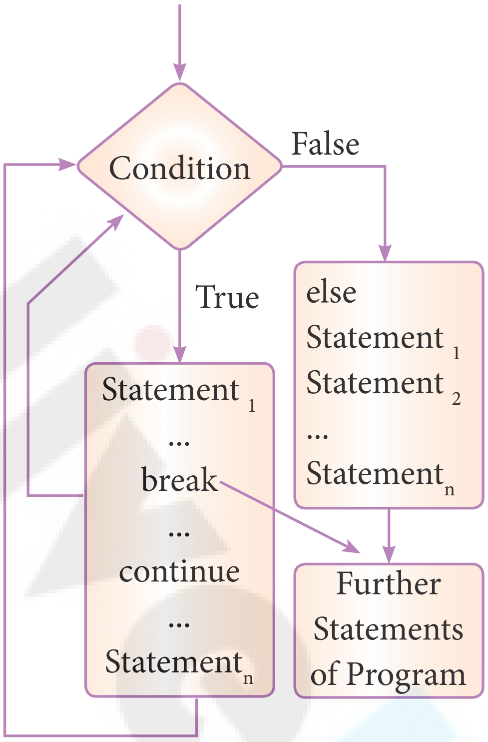
```
i=1
while (i<=6):
    for j in range (1,i):
        print (j,end='\t')
    print (end='\n')
    i +=1
```

வெளியீடு:

```
1
1   2
1   2   3
1   2   3   4
1   2   3   4   5
```

6.2.4 பைத்தானில் JUMP கூற்றுகள்:

பைத்தானிலுள்ள **JUMP** கூற்று, கட்டுப்பாட்டை எந்தவொரு நிபந்தனையுமின்றி, நிரலின் ஒரு பகுதியில் இருந்து மற்றொரு பகுதிக்கு இடமாற்றம் செய்ய பயன்படுகிறது. பைதானில் **Jump** கூற்றை பயன்படுத்த மூன்று சிறப்புச் சொற்கள் உள்ளன. அவை: **break, continue, pass**. பின்வரும் பாய்வுப்படம் **break** மற்றும் **continue** பயன்பாட்டை விளக்கும்.



படம் 6.6 மடக்கு அமைப்பில் *break* மற்றும் *continue* கூற்றுகளின் பயன்பாடு

6.2.4.1 break கூற்று

break கூற்றானது அதை உள்ளடக்கிய மடக்கை விட்டு வெளியேறச் செய்கிறது. நிரலின் கட்டுப்பாடானது, மடக்கின் உடற்பகுதியைத் தொடர்ந்து உடனடியாக இருக்கும் கூற்றுக்கு பாய்கிறது.

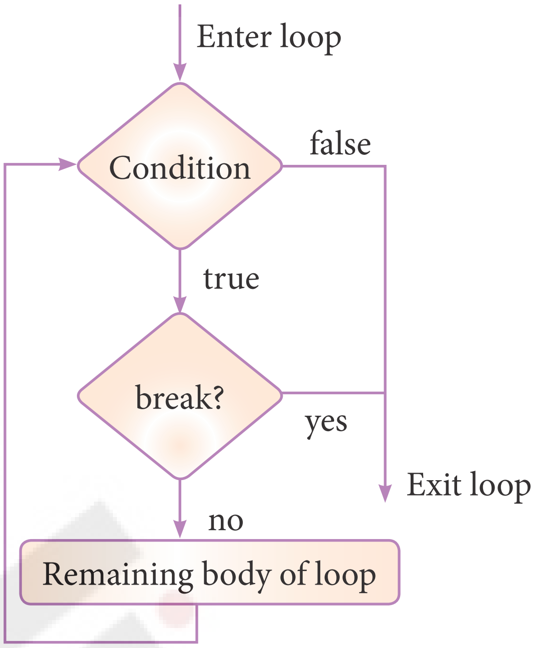
நிபந்தனையானது தவறு என்று பரிசோதிக்கும் வரை **while** அல்லது **for** மடக்கு செயல்படுத்தப்படும். ஆனால், **break** கூற்றைப் பயன்படுத்தி கட்டுப்பாட்டை மடக்கை விட்டு நிறுத்தி வெளியேறச் செய்யமுடியும். **break** கூற்று நிறைவேற்றப்படும் போது, நிரலின் கட்டுப் பாட்டுபாய்வு, மடக்கை விட்டு வெளியேறி, மடக்கு அமைப்பிற்கு **இறுதிக்கு அடுத்து** உள்ள குறி முறை பகுதியில் இருந்து நிறைவேற்றத் தொடங்கும்.

break கூற்று பின்னலான மடக்கின் உள்ளே இருந்தால், **break** கூற்று மிகவும் உள்ளே உள்ள மடக்கை விட்டு வெளியேறும்.

தொடரியல்:

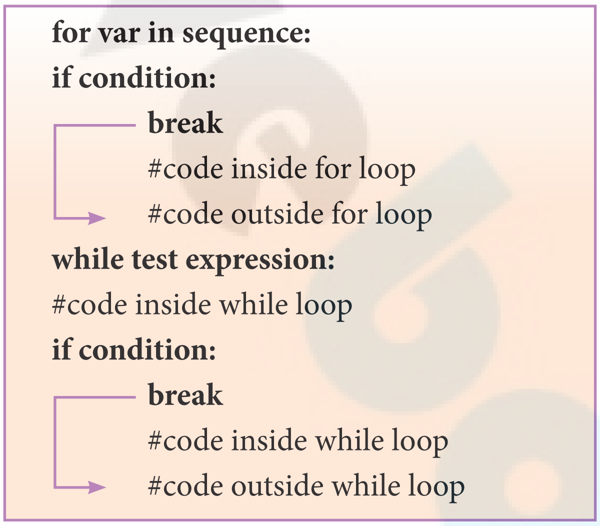
break

Chapter 6.indd 84 30-01-2020 19:24:38



படம் *6.7 break* கூற்றின்செயல்பாடு

for மடக்கு மற்றும் while மடக்கினுள் break கூற்றின் செயல்பாடு கீழே கொடுக்கப்பட்டுள்ளது.



எடுத்துக்காட்டு 6.14: #for மடக்கின் உள்ளே break கூற்றின் பயன்பாட்டை விளக்குகிறது

```
for word in "Jump Statement":
    if word = = "e":
        break
print (word, end='')
```

வெளியீடு:

Jump Stat

மேற்கண்ட நிரல், கொடுக்கப்பட்ட **"Jump statement"** என்ற சரத்தை மடக்குச் செயலாக மீண்டும் மீண்டும் இயக்குகிறது. கொடுக்கப்பட்ட சர வரிசையில் **'e'** என்ற எழுத்து வரும் வரை ஒவ்வொரு எழுத்தாக பரிசோதிக்கப்படும். அந்த எழுத்து வரும்போது கட்டுப்பாடானது மடக்குக்கு வெளியே செல்லும் அல்லது மடக்கை நிறுத்திவிடும். வெளியீட்டில் காட்டியுள்ளவாறு, கொடுக்கப்பட்ட சரத்தில் **'e'** என்ற எழுத்து வரும் வரை ஒவ்வொரு எழுத்தாக அச்சிடப்பட்டு மடக்கு நிறுத்தப்படும்.

இதில் முக்கியமாக கவனிக்க வேண்டியது என்னவென்றால், மடக்கானது **'break'** கூற்றுடன் முடிந்தால் **'else'** பகுதி நிறைவேற்றப்படாது. இதை விளக்குவதற்கு **else** பகுதியுடன் மேலே குறிப்பிட்ட நிரலை மேம்படுத்தி அதன் வெளியீடு என்னவென்று காண்போம்:

எடுத்துக்காட்டு 6.15: # for மடக்கின் உள்ளே break கூற்றின் பயன்பாட்டை else பகுதியுடன் விளக்கும் நிரல்

```
for word in "Jump Statement":
    if word = = "e":
        break
        print (word, end='')
else:
        print ("End of the loop")
print ("\n End of the program")
```

வெளியீடு:

```
Jump Stat
End of the program
```

break கூற்று, மடக்கின் 'else' பகுதியைத் தவிர்த்து கட்டுப்பாட்டை மடக்கு தொகுதிக்குப் பிறகுள்ள உள்ளே அடுத்த கூற்றுக்கு கொண்டு செல்கிறது.

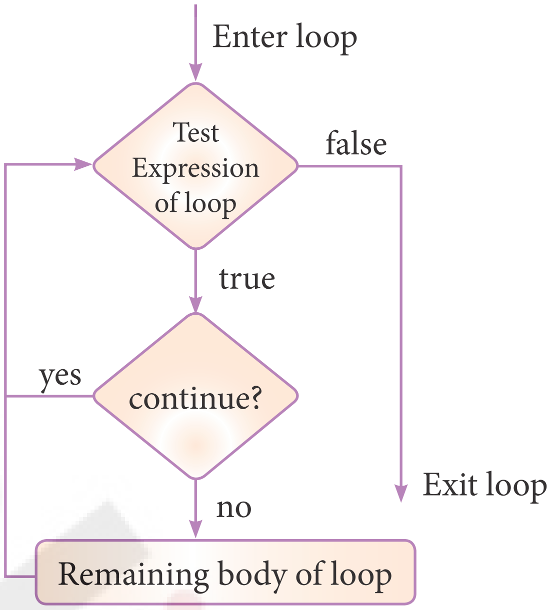
6.2.4.2 continue கூற்று

continue கூற்றானது break கூற்றைப் போல் இல்லாமல், மடக்கின் மீதமுள்ள குறிமுறையைத் தவிர்த்து அடுத்த மடக்கு செயலை ஆரம்பிக்கும்.

தொடரியல்:

continue

Chapter 6.indd 86 30-01-2020 19:24:38



படம் 6.8 *continue* கூற்றின் செயல்பாடு

for மற்றும் while மடக்கினுள் **continue** கூற்றின் செயல்பாடானது கீழே கொடுக்கப்பட்டுள்ளது.

```
for var in sequence:
        # code inside for loop
        if condition:
                continue
        #code inside for loop
#code outside for loop
while test expression:
        #code inside while loop
        if condition:
                continue
        #code inside while loop
#code outside while loop
```

எடுத்துக்காட்டு 6.16: # for மடக்கினுள் continue கூற்றின் பயன்பாட்டை விளக்கும் நிரல்

```
for word in "Jump Statement":
    if word = = "e":
        continue
    print (word, end='')
print ("\n End of the program")
```

வெளியீடு:

```
Jump Statmnt
End of the program
```


Chapter 6.indd 87 30-01-2020 19:24:38

மேலே கொடுக்கப்பட்ட நிரலும் '**break**' கூற்றுக்கு எழுதப்பட்ட நிரலும் ஒன்றே, ஆனால் '**break**'க்கு பதிலாக இங்கு '**continue**' கூற்று பயன்படுத்தப்பட்டுள்ளது. '**e**' என்ற எழுத்தைத் தவிர மற்ற அனைத்து எழுத்துக்களும் அச்சிடப்படுகிறது.

6.2.4.3 pass கூற்று

pass கூற்று ஒரு **null** கூற்றாகும். நிரல் பெயர்ப்பி **pass** கூற்றை நிறைவேற்றும் பொழுது, அதே கூற்றை முழுவதுமாக புறக்கணித்துவிடும். **pass** கூற்றை இயக்கும் பொழுது எந்த செயல்பாடும் நடைபெறாது.

'**if**' கூற்றிலும், மடக்கினுள்ளும் எந்த கட்டளையையும் அல்லது கூற்றையும் நிறைவேற்ற வேண்டாம் என நினைக்கும் போது **pass** கூற்று இயக்கப்படுகிறது.

தொடரியல்:

pass

எடுத்துக்காட்டு 6.17:# pass கூற்றின் பயன்பாட்டை விளக்கும் நிரல்

```
a=int (input("Enter any number :"))
if (a==0):
    pass
else:
    print ("non zero value is accepted")
```

மேற்கண்ட நிரலை இயக்கும் போது உள்ளீட்டின் மதிப்பு 0 எனில் ஒரு செயலும் நடைபெறாது. மற்ற பிற உள்ளீட்டு மதிப்புகளுக்கு பின்வரும் வெளியீடு கிடைக்கும்:

வெளியீடு:

```
Enter any number :3
non zero value is accepted
```

குறிப்பு

pass கூற்று பொதுவாக இட ஒதுக்கீட்டிற்காக பயன்படுகிறது. மடக்கு அல்லது செயற்கூற்றைதற் பொழுது செயல்படுத்தாமல் பிறகு செயல்படுத்த விரும்பும் போது, வெற்று உடற்பகுதியாக உருவாக்க முடியாது. ஏனெனில், நிரல்பெயர்ப்பி பிழை செய்தியைத் தரும். இதை தவிர்க்க '**pass**' கூற்று எதையும் நிறைவேற்றாத உடற்பகுதியில் அமைக்கப்படுகிறது.

எடுத்துக்காட்டு 6.18: # for மடக்கினுள் 'pass' கூற்றை விளக்கும் நிரல்

```
for val in "Computer":
        pass
print ("End of the loop, loop structure will be built in future")
```

வெளியீடு:

```
        End of the loop, loop structure will be built in future.
```


Chapter 6.indd 88 30-01-2020 19:24:38

நினைவில் கொள்க:

- நிரல் என்பது வரிசையாக நிறை வேற்றப்படும் கூற்றுகளைக் கொண்டதாகும். அந்த வரிசையை மாற்றுவதற்கு கட்டுப்பாட்டுக் கூற்றுகள் பயன்படுகிறது.
- கட்டுப்பாட்டு நிரலின், ஒரு பகுதியில் இருந்து இன்னொரு பகுதிக்கு தாவுவதற்கு காரணமான நிரல் கூற்றுகள் கட்டுப்பாட்டு கட்டமைப்பு (அல்லது) கட்டுப்பாட்டு கூற்றுகள் எனப்படும்.
- மூன்று வகையான கட்டுப்பாட்டுப் பாய்வுகள் உள்ளன
  - o வரிசை முறை
  - o மாற்று அல்லது கிளை பிரிப்பு
  - o பன்முகச் செயல் அல்லது மடக்குச் செயல்
- பைத்தானில் பல்வேறு வகையான if அமைப்புகளை பயன்படுத்தி கிளைப்பிரிப்பு சாத்தியமாகிறது.
- உள் தள்ளல் பைத்தானில் மிக முக்கிய பங்கு வகிக்கிறது. நெளிவு அடைப்புக் குறியை ({}) பயன்படுத்தாமல் கூற்றுகளை ஒரு தொகுதிக்குள் குறிப்பிட உள்தள்ளல் பயன்படுகிறது.
- பைத்தானில் தவறான உள் தள்ளல்களுக்கு நிரல் பெயர்ப்பிபிழை செய்திகளைக் காண்பிக்கும்.
- print() செயற்கூறு விடுபடு வரிசையைப் பயன்படுத்தி பயனர் விரும்பும் வெளியீட்டை வடிவமைக்க ஆதரவளிக்கிறது.
- range() செயற்கூறு for மடக்கில் வரம்பிற்குட்பட்ட மதிப்புகளை வழங்குகிறது.
- பைத்தானில் break, continue மற்றும் pass ஆகிய கூற்றுகள் jump கூற்றுகளாக செயல்படுகிறது.
- பைத்தானில் pass கூற்று ஒரு null கூற்றாகும். இது பொதுவாக இட ஒதுக்கீட்டிற்காக பயன்படுகிறது.

செய்முறைப் பயிற்சி

1. கொடுக்கப்பட்ட எழுத்து 'உயிரெழுத்தா இல்லையா' என கண்டறியும் நிரலை எழுதுக.
2. if..else..elif கூற்றைப் பயன்படுத்தி கொடுக்கப்பட்ட மூன்று எண்களில் மிகச் சிறிய எண்ணை கண்டறியும் நிரலை எழுதுக.
3. கொடுக்கப்பட்ட எண் நேர்மறை எண்ணா அல்லது எதிர்மறை எண்ணா அல்லது பூஜ்யமா எனக் கண்டறியும் நிரலை எழுதுக.
4. கொடுக்கப்பட்ட வருடம் லீப் வருடமா இல்லையா எனக் கண்டறியும் நிரலை எழுதுக.

Chapter 6.indd 89 30-01-2020 19:24:38

செய்முறைப் பயிற்சி

1. பைபோனாசி தொடரை (0 1 1 2 3 4 5......n) வரை வெளியிடுவதற்கான நிரலை எழுதுக.
2. n வரையிலான இயல் எண்களின் கூட்டுத் தொகையை வெளியிடுவதற்கான நிரலை எழுதுக.
3. கொடுக்கப்பட்ட எண் பாலின் ரோமா இல்லையா என்பதை சரிபார்க்கும் நிரலை எழுதுக.
4. பின்வரும் அமைப்பை அச்சிடுவதற்கான நிரலை எழுதுக.

```
* * * * *
* * * *
* * *
* *
*
```

மதிப்பீடு

SZHFZA

பகுதி I

I சரியான விடையைத் தேர்ந்தெடுத்து எழுதுக **(1 மதிப்பெண்)**

1. பைத்தானில் எத்தனை முக்கியமான கட்டுப்பாட்டு கட்டமைப்புகள் உள்ளன?
   அ) 3 ஆ) 4
   இ) 5 ஈ) 6
2. elif என்பதன் விரிவாக்கம்
   அ) nested if ஆ) if..else
   இ) else if ஈ) if..elif
3. பைத்தான் நிரலில் எது முக்கிய பங்கு வகிக்கிறது?
   அ) கூற்றுகள் ஆ) கட்டுப்பாடு
   இ) அமைப்பு ஈ) உள்தள்ளல்
4. எந்த கூற்று பொதுவாக இட ஒதுக்கீட்டிற்காகப் பயன்படுகிறது?
   அ) continue ஆ) break
   இ) pass ஈ) goto
5. if கூற்றின் நிபந்தனை பின்வரும் எந்த வடிவில் இருக்க வேண்டும்
   அ) கணித அல்லது ஒப்பீட்டுக் கோவைகள்
   ஆ) கணித அல்லது தருக்கக் கோவைகள்
   இ) ஒப்பீட்டு அல்லது தருக்கக் கோவைகள்
   ஈ) கணித கோவைகள்

Chapter 6.indd 90 30-01-2020 19:24:38

6. எது மிகவும் சுலபமான மடக்கு எது?

அ) do..while　　ஆ) while

இ) for　　ஈ) if..elif

7. பின்வரும் குறிமுறையின் வெளியீடு என்ன?

```
i=1
while True:
        if i%3 ==0:
                break
        print(i,end='')
        i +=1
```

அ) 12　　ஆ) 123

இ) 1234　　ஈ) 124

8. பின்வரும் குறிமுறையின் வெளியீடு என்ன?

```
T=1
while T:
        print(True)
        break
```

அ) தவறு　　ஆ) சரி

இ) 0　　ஈ) வெளியீடு இல்லை

9. பின்வருவனவற்றில் எது jump கூற்று கிடையாது ?

அ) for　　ஆ) pass

இ) continue　　ஈ) break

10. எந்த நிறுத்தற்குறி பின்வரும் அடிக்கோடிட்ட இடத்தில் இடம் பெற வேண்டும்?

```
if <condition>_
   statements-block 1
else:
   statements-block 2
```

அ) ;　　ஆ) :

இ) ::　　ஈ) !

பகுதி–ஆ

II பின்வரும் வினாக்களுக்கு விடையளி　　(2 மதிப்பெண்)

11. பைத்தானில் உள்ள கட்டுப்பாட்டு கட்டமைப்புகளைப் பட்டியலிடுக.

12. break கூற்றைப் பற்றி குறிப்பு வரைக.

13. if..else கூற்றின் பொது வடிவத்தை எழுதுக?

Chapter 6.indd 91　　30-01-2020 19:24:38

14. கட்டுப்பாட்டு கட்டமைப்பு என்றால் என்ன?

15. range() செயற்கூறு குறிப்பு வரைக.

பகுதி–இ

III பின்வரும் வினாக்களுக்கு விடையளி (3 மதிப்பெண்)

16. பின்வரும் வெளியீட்டைப் பெற நிரலை எழுதுக

A

A B

A B C

A B C D

A B C D E

17. if..else அமைப்பைப் பற்றி குறிப்பு வரைக.

18. if..else..elif கூற்றைப் பயன்படுத்தி கொடுக்கப்பட்ட மூன்று எண்களில் பெரிய எண்ணைக் கண்டுபிடிப்பதற்கான பொருத்தமான நிரலை எழுதுக.

19. while மடக்கின் பொதுவடிவம் யாது?

20. break மற்றும் continue கூற்றுகளின் வேறுபாடு யாது?

பகுதி–ஈ

IV பின்வரும் வினாக்களுக்கு விடையளி (5 மதிப்பெண்)

21. for மடக்கைப் பற்றி விரிவான விடையளிக்கவும்.

22. if..else..elif கூற்றை எடுத்துக்காட்டுடன் விளக்குக.

23. அனைத்து மூன்று இலக்க ஒற்றைப்படை எண்களை வெளியிடுவதற்கான நிரலை எழுதுக.

24. கொடுக்கப்பட்ட எண்ணின் பெருக்கல் வாய்ப்பாட்டை வெளியிடும் நிரலை எழுதுக.

Chapter 6.indd 92 30-01-2020 19:24:38

அலகு II

பாடம் 7

பைத்தான் செயற்கூறுகள்

7KXTST

கற்றலின் நோக்கங்கள்

இந்த பாடத்தை கற்றபிறகு, மாணவர்கள்:

- செயற்கூறுகளின் கருத்து மற்றும் அதன் வகைகளைப் பற்றி அறிந்து கொள்ள முடியும்.
- பயனர் வரையறுக்கும் மற்றும் உள்ளிணைந்த செயற்கூறுகளுக்கு இடையே உள்ள வேறுபாட்டை அறிந்து கொள்ள முடியும்.
- ஒரு செயற்கூறினை எவ்வாறு அழைப்பது என்பதை தெரிந்து கொள்ள முடியும்.
- செயற்கூறின் அளபுருக்களைப் புரிந்து கொள்ள முடியும்.
- பெயரில்லாத செயற்கூறுவை அறிந்து கொள்ள முடியும்.
- கணித மற்றும் சரங்களின் செயற்கூறுகளைப் பற்றி அறிந்து கொள்ள முடியும்.

7.1 அறிமுகம்

ஒரு குறிப்பிட்ட செயலினை செய்வதற்காக வடிவமைக்கப்பட்டு பெயரிடப்பட்ட குறிமுறையின் தொகுதியே செயற்கூறு எனப்படும். செயற்கூறில் வரையறுத்த குறிப்பிட்ட செயலினை செய்ய விரும்பினால் அதற்கு பொருத்தமான செயற்கூறின் பெயரை கொண்டு அழைக்க வேண்டும். நிரலில், ஒரே செயலை பல தடவை மீண்டும் மீண்டும் செய்ய நேரிட்டால் அதற்கான குறிமுறையை மீண்டும் எழுதத் தேவையில்லை. அதற்கான செயற்கூறுவை அழைத்தால் மட்டும் போதும், அந்த அழைப்பு பைத்தானின் செயற்கூற்றின் உள்ளேயுள்ள குறிமுறையை இயக்க செய்யும். செயற்கூறுவின் பயன்பாடு ஒரு நிரலை, எழுத, படிக்க, சோதிக்க மற்றும் பிழை திருத்தும் வேலைகளை எளிதாக்குகிறது.

செயற்கூறுவின் முதன்மை நன்மைகள்

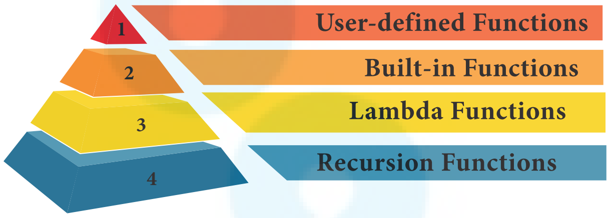
- குறிமுறையை மீண்டும் எழுதுவதை தவிர்த்து குறிமுறையின் மறு பயனாக்கத்திற்கு உதவுகிறது.
- நமது பயன்பாட்டிற்குச் சிறந்த கூறுநிலையை வழங்குகிறது.

குறிப்பு

செயற்கூறுகள் என்பது குறிப்பிட்ட செயலை செய்வதற்கான தொடர்புடைய கூற்றுகளின் தொகுதி ஆகும்.

7.1.1 செயற்கூறுகளின் வகைகள்

அடிப்படையாக செயற்கூறுகளை பின்வருமாறு வகைப்படுத்தலாம்:



படம் – 7.1 – பைத்தான் செயற்கூறுவின்வகைகள்

செயற்கூறுகள்	விளக்கம்
பயனர் வரையறுக்கும் செயற்கூறுகள்	பயனர்கள் தாங்களாகவே வரையறுக்கும் செயற்கூறுகள்.

உள்ளிணைந்த செயற்கூறுகள்	பைத்தானில் உள்ளடக்கப்பட்ட செயற்கூறுகள்
லாம்டா செயற்கூறுகள்	பெயரில்லாத செயற்கூறுகள்
தற்சுழற்சி செயற்கூறுகள்	தற்சுழற்சியாக தன்னைத் தானே அழைத்துக் கொள்ளும் செயற்கூறுகள்

அட்டவணை – 7.1 – பைத்தானின் செயற்கூறுகள் மற்றும் அவற்றின் விளக்கமும்

7.2 செயற்கூறுவை வரையறுத்தல்

ஒரு செயல்பாட்டை உருவாக்கி அதை பயன்படுத்துவதற்கு செயற்கூறினை வரையறுக்க வேண்டும். பைத்தான் மொழியில் பல உள்ளிணைந்த செயற்கூறுகள் உள்ளன. (எடுத்துக்காட்டாக, print() செயற்கூறு), இருப்பினும் பயனர் அவருக்குத் தேவையான செயற்கூறினையும் வரையறுக்க முடியும். செயற்கூறினை வரையறுக்கும் போது நினைவில் கொள்ள வேண்டியவை.

- செயற்கூறு தொகுதி def என்ற சிறப்புச் சொல்லுடன் தொடங்கி செயற்கூறுவின் பெயர் மற்றும் () அடைப்புக்குறியுடன் முடிய வேண்டும்.
- ஏதேனும் உள்ளீட்டு செயலுருப்புகள் அல்லது அளபுருக்கள் இருப்பின் அவற்றை செயற்கூற்றை வரையறுக்கும் போதே ( ) என்ற அடைப்புக்குறிக்குள் கொடுக்க வேண்டும்.
- குறிமுறை தொகுதியானது எப்பொழுதும் முக்காற்புள்ளிக்கு பிறகு உள்தள்ளி வர வேண்டும்.
- “return [கோவை]” கூற்று செயற்கூறுவை முடித்து வைக்கும். விருப்பப்பட்டால் கோவையின் மதிப்பை, அழைக்கும் கூற்றுக்கு திருப்பி அனுப்பும். செயலுருபுகள் இல்லாத **return, return None**-க்கு நிகரானது.

குறிப்பு

பைத்தான் சிறப்புச் சொற்கள் செயற்கூறுவின் பெயர்களாக பயன்படுத்தக் கூடாது.

7.2.1 பயனர் வரையறுக்கும் செயற்கூறுவின் தொடரியல்

```
def <function_name ([parameter1, parameter2…] )> :
        <Block of Statements>
        return <expression / None>
```

குறிப்பு

மேலே உள்ள பொது வடிவத்தில் சதுர அடைப்புக் குறிக்குள் [] உள்ளவை அனைத்தும் விருப்பத் தேர்வாகும்

தொகுதி:

தொகுதி என்பது ஒன்று அல்லது ஒன்றிற்கு மேற்பட்ட வரிகளையுடைய குறிமுறையைக் குழுவாக ஒன்றிணைத்து, அதனை நிறைவேற்றும் போது ஒரு பெரிய வரிசையின் கூற்றுகளாக எடுத்துக் கொள்கிறது. பைத்தானில், தொகுதியில் உள்ள கூற்றுகளை உள்தள்ளல் மூலம் எழுதப்படுகிறது. பொதுவாக, ஒரு வரி உள்தள்ளப் படும்போது (4 இடைவெளி) தொகுதியானது தொடங்கும். தொகுதியின் உள்ளே உள்ள அனைத்து கூற்றுகளையும் ஒரே மாதிரியாக உள்தள்ள வேண்டும்.

பின்னலான தொகுதி

ஒரு தொகுதி மற்றொரு தொகுதியைக் கொண்டிருந்தால் அது பின்னலான தொகுதியாகும். முதல் தொகுதியின் கூற்றுகளை ஒற்றை தத்தல் இடைவெளி அளவிற்கு உள்தள்ளினால், இரண்டாவது தொகுதியின் கூற்றுகளை இரண்டு தத்தல் இடைவெளி அளவிற்கு உள்தள்ள வேண்டும்.

செயற்கூற்றை வரையறைக்கும் எடுத்துக்காட்டு,

Chapter 7.indd 94 30-01-2020 19:30:58

```
def Do_Something( ):
value =1          #Assignment கூற்று
return value      #Return கூற்று
```

இப்பொழுது செயற்கூறின் செயல்பாட்டை சோதிப்பின் மூலம் அது நிரலில் எவ்வாறு வேலை செய்கிறது என்பதை காணலாம். பின்வரும் எடுத்துக்காட்டில் ஒரு சாதாரண செயற்கூறு சரத்தை வெளியிடுவதைக் காணலாம்.

எடுத்துக்காட்டு

```
def hello():
        print ("hello - Python")
        return
```

7.2.2 பயனர் வரையறுக்கும் செயற்கூறுகளின் நன்மைகள்

1. செயற்கூறுகள் ஒரு நிரலை சிறு தொகுதியாக பிரிக்க உதவுகிறது. இது குறிமுறையை எளிதாக கையாள உதவுகிறது.

2. குறிமுறையின் மறுபயனாக்கத்திற்கு உதவுகிறது. ஒவ்வொரு முறையும் கூற்றுகளின் வரிசைகளை நிறைவேற்றும் போது, நாம் அந்த செயற்கூற்றினை அழைத்தால் போதுமானது.

3. செயற்கூற்றின் செயல்பாடுகளை மாற்றம் செய்வது எளிதாகிறது. வெவ்வேறு நிரலர்கள் வெவ்வேறு செயற்கூற்றில் வேலை செய்ய முடியும்.

7.3 செயற்கூறினை அழைத்தல்

எடுத்துக்காட்டு 7.2.1 ல் hello() செயற்கூறினை அழைக்க இந்த குறிமுறை பயன்படும். "hello()" செயற்கூறினை அழைக்கும் பொழுது நிரல் பின்வரும் சரத்தை வெளியீடுகிறது.

வெளியீடு

hello – Python

மாற்றாக, "hello()" செயற்கூறினை print() செயற்கூற்றிலிருந்து அழைப்பதை பின்வரும் எடுத்துக்காட்டில் காணலாம்.

எடுத்துக்காட்டு:

```
def hello():
        print ("hello - Python")
        return
print (hello())
```

return கூற்றில் எந்த அளபுருவும் இல்லை எனில் வெளியீட்டின் இறுதிக் கூற்றாக, "None" வெளியிடப்படும்.

மேலே உள்ள செயற்கூறு பின்வரும் வெளியீட்டைக் கொடுக்கும்.

வெளியீடு

hello – Python
None

7.4 செயற்கூறினுள் அளபுருக்களை அனுப்புதல்

அளபுருக்கள் அல்லது செயலுருப்புகளை செயற்கூறினுக்கு அனுப்பலாம்.

def function_name (parameter(s) separated by comma):

அளபுருக்களின் பயன்பாட்டினை செயற்கூறு வரையறையில் பார்க்கலாம். அடைப்புக் குறிக்குள் கொடுக்கப்படும் அளபுருக்கள் அந்த செயற்கூறினுள்ளேயே பயன்படுத்தப்படும். அனைத்து விதமான தரவுகளையும் செயற்கூறினுள் அனுப்ப முடியும். அளபுருக்களை செயற்கூறினுக்கு அனுப்பும் செயற்கூறு வரையறை எடுத்துக்காட்டு நிரலில் கொடுக்கப்பட்டுள்ளது.

எடுத்துக்காட்டு :

```
# assume w = 3 and h = 5
    def area(w,h):
        return w * h
    print (area (3,5))
```

மேலே உள்ள குறிமுறையில் அகலம் மற்றும் உயரத்தின் மதிப்புகள் முறையே w மற்றும் h அளபுருக்களில் இருத்தப்படுகிறது. “area” என்ற செயற்கூறினை உருவாக்குவதற்கு இந்த அளபுருக்கள் பயன்படுகின்றன. மேலே குறிப்பிட்ட செயற்கூறினை அழைக்கும் போது வெளியீடாக அகலம் மற்றும் உயரத்தின் பெருக்கல் மதிப்பினை திருப்பி அனுப்புகிறது. w மற்றும் h அளபுருக்களுக்கு முறையே 3 மற்றும் 5 மதிப்புகளை அனுப்பினால் வெளியீடாக 15 என்ற மதிப்பதை திருப்பி அனுப்பும்.

? உங்களுக்குத் தெரியுமா?

நாம் அடிக்கடி அளபுருக்கள் மற்றும் செயலுருபுக்கள் என்ற வார்த்தையை மாற்றி மாற்றி உபயோகப்படுத்துகிறோம். எனினும், இரண்டிற்கும் இடையே ஒரு சிறிய வித்தியாசம் உள்ளது. அளபுருக்கள் என்பவை செயற்கூறு வரையறையில் பயன்படுத்தப்படும் மாறிகள். ஆனால் செயலுருப்புகள் என்பது செயற்கூறின் அளபுருக்களுக்கு அனுப்பப்படும் மதிப்புகள் ஆகும்.

7.5 செயற்கூறு செயலுருபுகள் (Function Arguments)

செயலுருபுகள் செயற்கூறினை அழைக்க பயன்படுகிறது. செயலுருபுகள் நான்கு வகைப்படும் தேவைப்படும் செயலுருபுகள் (Required arguments), சிறப்பு சொல் செயலுருபுகள் (Keyword arguments), தானமைவு செயலுருபுகள் (Default arguments), மாறும் நீள செயலுருபுகள் (variable length arguments).

Chapter 7.indd 96 30-01-2020 19:30:58

செயற்கூறு செயலுருபுக்கள்
(Function Arguments)

1. தேவைப்படும் செயலுருபுக்கள் (Required arguments)
2. சிறப்புச் சொல் செயலுருபுக்கள் (Keyword arguments)
3. தானமைவு செயலுருபுக்கள் (Default arguments)
4. மாறும்-நீள செயலுருபுக்கள் (Variable-length arguments)

7.5.1 தேவைப்படும் செயலுருபுகள்

செயலுருபுகளை சரியான இடவரிசையில் செயற்கூற்றுக்கு அனுப்புவதே **தேவைப்படும் செயலுருபுக்கள்** எனப்படும். இங்கு, செயற்கூறு அழைப்புக் கூற்றில் உள்ள செயலுருபுகளின் எண்ணிக்கை, செயற்கூறு வரையறையோடு சரியாக பொருந்த வேண்டும். தொடரியல் பிழையை தவிர்த்து, தேவையான வெளியீட்டை பெறுவதற்கு குறைந்தது ஒரு செயலுருபாவது தேவை.

எடுத்துக்காட்டு

```
def printstring(str):
        print ("Example - Required  arguments ")
        print (str)
        return
# Now you can call printstring() function
printstring()
```

மேலே உள்ள குறிமுறை இயக்கப்படும் போது இது பின்வரும் பிழையை கொடுக்கிறது.

```
Traceback (most recent call last):
File "Req-arg.py", line 10, in <module>
printstring()
    TypeError: printstring() missing 1 required positional argument: 'str'
```

மேலே உள்ள குறிமுறையில் print string( )க்கு பதிலாக printstrin(“Welcome”) பயன்படுத்தினால் இதன் வெளியீடு,

வெளியீடு

Example - Required arguments
Welcome

7.5.2 சிறப்புச் சொல் செயலுருபுகள்

அளபுருக்களின் பெயரை அடையாளம் கண்ட பின்பு சிறப்புச் சொல் செயலுருபானது செயற்கூறினை அழைக்கிறது. சிறப்புச் சொல் செயலுருபின் மதிப்பு அளபுரு பெயருடன் பொருந்த வேண்டும். எனவே, முறையற்ற வரிசையிலும் செயலுருபுகளைக் கொடுக்கலாம்.

எடுத்துக்காட்டு:

```
def printdata (name):
        print ("Example-1 Keyword arguments")
        print ("Name :",name)
        return
# Now you can call printdata() function
printdata(name = "Gshan")
```

மேற்கண்ட குறிமுறை இயக்கப்படும் போது பின்வரும் வெளியீடு கிடைக்கும்.

வெளியீடு

```
Example-1 Keyword arguments
Name :Gshan
```

எடுத்துக்காட்டு:

```
def printdata (name):
        print ("Example-2 Keyword arguments")
        print ("Name :", name)
        return
# Now you can call printdata() function
printdata (name1 = "Gshan")
```

மேற்கண்ட குறிமுறை இயக்கப்படும் போது பின்வரும் வெளியீடு கிடைக்கும்.

TypeError: printdata() got an unexpected keyword argument 'name1'

எடுத்துக்காட்டு:

```
def printdata (name, age):
        print ("Example-3 Keyword arguments")
        print ("Name :",name)
        print ("Age :",age)
        return
# Now you can call printdata() function
printdata (age=25, name="Gshan")
```

மேற்கண்ட குறிமுறை இயக்கப்படும் போது பின்வரும் வெளியீடு கிடைக்கும்.

Chapter 7.indd 98 30-01-2020 19:30:58

வெளியீடு

Example-3 Keyword arguments
Name : Gshan
Age : 25

குறிப்பு

மேலே உள்ள நிரலில் அளபுருக்களின் வரிசைமுறை மாறி உள்ளது.

7.5.3 தானமைவு செயலுருபுகள்

பைத்தானில், செயற்கூறை அழைக்கும் போது எந்த மதிப்பும் கொடுக்கப்படவில்லை எனில், செயலுருபானது தானாகவே மதிப்பை எடுத்துக் கொள்ளும். இதுவே **தானமைவு செயலுருபு** ஆகும்.

பின்வரும் எடுத்துக்காட்டில் செயலுருபுவை அனுப்பாதபோது தானமைவு Salary-யை அச்சிடுவதைக் காணலாம்.

எடுத்துக்காட்டு:

```
def printinfo( name, salary = 3500):
        print ("Name: ", name)
        print ("Salary: ", salary)
        return
printinfo("Mani")
```

மேற்கண்ட குறிமுறை இயக்கப்படும் போது பின்வரும் வெளியீடு கிடைக்கும்.

வெளியீடு

Name: Mani
Salary: 3500

மேற்கண்ட குறிமுறையை printinfo("Ram", 2000)என மாற்றும் போது பின்வரும் வெளியீடு கிடைக்கும்.

வெளியீடு

Name: Ram
Salary: 2000

மேலே உள்ள குறிமுறையில் 2000 என்ற மதிப்பை salary என்ற செயலுருபுக்கு அனப்பினால் ஏற்கனவே உள்ள தானமைவு மதிப்பு நிராகரிக்கப்படும்.

7.5.4 மாறும் நீள செயலுருபுகள்

சில சமயங்களில், ஏற்கனவே குறிப்பிடப்பட்ட செயலுருபுகளை விட அதிகமான செயலுருபுகளை அனுப்ப வேண்டிய தேவை இருக்கும். மீண்டும் செயற்கூறினை மறுவரையறை செய்வது கடினமான செயல்

அதற்கு பதிலாக மாறும் நீள செயலுருபுகளை பயன்படுத்தலாம். செயற்கூறு வரையறையில் இது குறிப்பிடப்படுவதில்லை. மாறாக இந்த செயலுருபுகளை குறிப்பிடுவதற்கு * குறியீடு பயன்படுகிறது. மூன்றுக்கு அதிகமான செயலுருபுகளை sum( ) செயற்கூறுவிற்கு அனுப்பும் பொழுது என்ன நிகழும் என பார்ப்போம்.

எடுத்துக்காட்டு:

```
def sum(x,y,z):
        print("sum of three nos :",x+y+z)
sum(5,10,15,20,25)
```

மேலே உள்ள குறிமுறையை இயக்கும் போது பின்வரும் வெளியீடு கிடைக்கும்.

TypeError: sum() takes 3 positional arguments but 5 were given

7.5.4.1 மாறும் நீள செயலுருபின் பொதுவடிவம்

```
def function_name(*args):
        function_body
        return_statement
```

எடுத்துக்காட்டு:

```
def printnos (*nos):
        for n in nos:
        print(n)
        return
# now invoking the printnos() function
print ('Printing two values')
printnos (1,2)
print ('Printing three values')
printnos (10,20,30)
```

வெளியீடு

```
Printing two values
        1
        2
Printing three values
        10
        20
        30
```

மதிப்பீடு செய்க

மேலே உள்ள நிரலில், printnos என்ற செயற்கூறின் பெயரை printnames என்ற அனைத்து இடங்களிலும் மாற்றி ஏற்ற தரவுகளை கொடுக்கவும். எடுத்துக்காட்டு, printnos(10,20,30) என்பதை printnames('mala', 'kala', 'bala') என மாற்றி கொடுத்து வெளியீட்டைக் காண். மாறி நீள செயலுருபுகளில், செயலுருபுகளை இரண்டு வழிகளில் அனுப்பலாம்.

In Variable Length arguments we can pass the arguments using two methods.

1. சிறப்புச் சொற்களற்ற மாறி செயலுருபுகள்
2. சிறப்புச் சொல் மாறி செயலுருபுகள்

Chapter 7.indd 100 30-01-2020 19:30:59

சிறப்புச் சொற்களற்ற மாறி செயலுருபுகளை tuples என்று அழைக்கிறோம். இதைப்பற்றி விரிவாக பின்வரும் பாடங்களில் காணலாம்.

குறிப்பு

சிறப்புச் சொல் மாறி செயலுருபு இந்த புத்தகத்திற்கு அப்பாற்பட்டது.

உங்களுக்குத் தெரியுமா?

பைத்தானில் print() செயற்கூறு என்பது மாறும் நீள செயலுருபுக்கு ஒரு எடுத்துக்காட்டு ஆகும்.

7.6 பெயரில்லாத செயற்கூறுகள் (Anonymous Function)

பெயரில்லாத செயற்கூறுகள் என்றால் என்ன?

பைத்தானில், பெயரில்லாமல் வரையறுக்கப்படும் செயற்கூறுவுக்கு **பெயரில்லாத செயற்கூறு** என்று பெயர். மற்ற சாதாரண செயற்கூறுகள் **def** என்ற சிறப்பு சொல்லுடன் வரையறுக்கப்படுகிறது. பைத்தானில் பெயரில்லாத செயற்கூறுகள் **லாம்டா** சிறப்புச் சொல்லுடன் வரையறுக்கப்படுகிறது. எனவே, பெயரில்லா செயற்கூறுகளை லாம்டா **செயற்கூறுகள்** என்றும் அழைக்கலாம்.

லாம்டா அல்லது பெயரில்லா செயற்கூறின் பயன்கள்

- லாம்டா செயற்கூறு பெரும்பாலும் சிறிய மற்றும் ஒரு முறை பெயரில்லாத செயற்கூறை உருவாக்க பயன்படுகிறது.
- filter( ), map() மற்றும் reduce() போன்ற செயற்கூறுகளுடன் சேர்த்து லாம்டா செயற்கூறுகளை பயன்படுத்தலாம்.

குறிப்பு

filter(), map() மற்றும் reduce() செயற்கூறுகள் இந்த புத்தகத்திற்கு அப்பாற்பட்டது.

உங்களுக்குத் தெரியுமா?

லாம்டா செயற்கூறு எண்ணற்ற செயலுருபுகளை எடுத்துக் கொண்டு திருப்பி அனுப்பும் ஒரே ஒரு மதிப்பை கோவை வடிவில் கொடுக்கும். லாம்டா செயற்கூற்றால் முழுதளாவிய மாறிகள் மற்றும் அளபுரு பட்டியலில் உள்ள மாறிகளை மட்டுமே அணுக முடியும்.

7.6.1 பெயரில்லா செயற்கூறின் பொது வடிவம்

lambda [argument(s)] :expression

எடுத்துக்காட்டு:

```
sum = lambda arg1, arg2: arg1 + arg2
print ('The Sum is :', sum(30,40))
print ('The Sum is :', sum(-30,40))
```

வெளியீடு

```
The Sum is : 70
The Sum is : 10
```

மேற்கண்ட லாம்டா செயற்கூறு arg 1 செயலுருபுவை arg 2 செயலுருபுடன் கூட்டி விடையை sum மாறியில் சேமிக்கும். **print ( )** செயற்கூறை பயன்படுத்தி வெளியீட்டை வெளியிடும்.

7.7 return கூற்று

- return கூற்று செயற்கூறினை முடித்து வைத்து அழைப்புக் கூற்றுக்கு மதிப்பை திருப்பி அனுப்பும். பொதுவாக செயற்கூறின் நோக்கம் உள்ளீடைப் பெற்று ஏதேனும் ஒரு மதிப்பை திருப்பி அனுப்புவதாகும்.
- ஒரு செயற்கூறு மதிப்பை அழைப்புக்கூற்றுக்கு திருப்பி அனுப்ப தயாராக இருக்கும் போது return கூற்று பயன்படுத்தப்படுகிறது. எனவே, இயக்க நேரத்தில் பல return கூற்றுகள் இருந்தாலும் ஒரே ஒரு return கூற்று மட்டுமே இயக்கப்படும்.
- செயற்கூறில் ஒன்றுக்கு மேற்பட்ட return கூற்றுகள் இருந்தாலும் ஒரே ஒரு return கூற்று மட்டுமே இயக்க நேரத்தில் இயக்கப்படும்.

7.7.1 return கூற்றின் பொது வடிவம்

return [கோவைகளின் பட்டியல்]

இந்த கூற்றில் உள்ள கோவைகள் மதிப்பீடு செய்யப்பட்டு, மதிப்பைத் திருப்பி அனுப்புகிறது. செயற்கூறினுள் உள்ள கூற்றில் கோவைகள் இல்லாமலோ அல்லது return கூற்று இல்லாமலோ இருந்தால் செயற்கூறு “None” பொருளைத் தருபிப் அனுப்பும்.

Chapter 7.indd 102 30-01-2020 19:30:59

எடுத்துக்காட்டு :

```
# return statment
def usr_abs (n):
        if n>=0:
                return n
        else:
                return –n
# Now invoking the function
x=int (input(“Enter a number :”)
print (usr_abs (x))
```

வெளியீடு 1:

```
        Enter a number : 25
                25
```

வெளியீடு 2:

```
        Enter a number : -25
                25
```

7.8 மாறிகளின் வரையெல்லை

இது நிரலின் அணுகக்கூடிய பகுதியைக் குறிப்பதாகும். அதாவது, எந்த பகுதியில் மாறியைப் பயன்படுத்துகிறோமோ அதைக் குறிக்கிறது. வரையெல்லையானது நடப்பு மாறித் தொகுதிகள் மற்றும் அதன் மதிப்புகளைக் கொண்டிருக்கும். இரண்டு வகையான வரையெல்லைகளைப் பார்க்கலாம். உள்ளமை வரையெல்லை மற்றும் குளோபல் வரையெல்லை.

7.8.1 உள்ளமை வரையெல்லை (Local Scope)

ஒரு செயற்கூறுவின் உடற்பகுதியின் உள்ளே அல்லது உள்ளமை வரையெல்லையில் மாறியை அறிவிப்பது உள்ளமை மாறி எனப்படும்.

உள்ளமை மாறியின் விதிமுறைகள்

- உள்ளமை மாறியின் வரையெல்லை அது வரையறுக்கப்பட்டுள்ள தொகுதிக்குள் மட்டுமே பயன்படுத்த முடியும்.
- செயற்கூறினுள் மாறி உருவாக்கப்படும் போது அது உள்ளமைவாக அமையும்.
- செயற்கூறு இயக்கப்படும் போது மட்டுமே உள்ளமை மாறிகள் உருவாக்கப்படும்.

எடுத்துக்காட்டு : உள்ளமை மாறியை உருவாக்குதல்

```
def loc():
        y=0 # local scope
        print(y)
loc()
```

வெளியீடு

```
        0
```

எடுத்துக்காட்டு : வரையெல்லைக்கு வெளியே உள்ளமை மாறிகளை அணுகுதல்

```
def loc():
        y = "local"
loc()
print(y)
```

மேலே உள்ள குறிமுறையை இயக்கும் போது வெளியீட்டில் பின்வரும் பிழை தோன்றும்.

குளோபல் வரையெல்லையில் உள்ளமை மாறியான ‘ y’ யை அணுக முற்பட்டதால் மேற்கண்ட பிழை தோன்றுகிறது.

NameError: name 'y' is not defined

7.8.2 குளோபல் வரையெல்லை (Global Scope)

குளோபல் வரையெல்லை உடைய மாறியை நிரலில் எங்கு வேண்டுமானாலும் அணுக முடியும். எந்த ஒரு செயற்கூறு வரையெல்லைக்கு வெளியேயும் மாறியை வரையறுத்து உருவாக்க முடியும்.

குளோபல் வரையெல்லை சிறப்புச் சொல்லின் விதிமுறைகள்

- செயற்கூறுக்கு வெளியே மாறியை அறிவிக்கும் போது அது தானமைவாக குளோபல் ஆகும். ‘global’ என்ற சிறப்புச் சொல்லை பயன்படுத்த வேண்டியதில்லை.
- செயற்கூறினுள் முழுதளாவிய மாறியை படிக்க மற்றும் எழுத ‘global’ சிறப்புச்சொல் பயன்படுத்த வேண்டும்.
- செயற்கூறுவிற்கு வெளியே ‘global’ என்ற சிறப்புச் சொல் எந்த விளைவையும் ஏற்படுத்தாது.

'Global' சிறப்புச் சொல்லின் பயன்பாடு

எடுத்துக்காட்டு : செயற்கூறுவின் உள்ளிருந்து குளோபல் மாறியை அணுகுதல்

```
c = 1            # global variable
def add():
        print(c)
add()
```

வெளியீடு

```
        1
```


Chapter 7.indd 104   30-01-2020 19:30:59

எடுத்துக்காட்டு : செயற்கூறுவின் உள்ளிருந்து முழுதளாவிய மாறியை மாற்றுதல்

மேற்கண்ட நிரலை இயக்கும் போது வெளியீடு பின்வரும் பிழையை காண்பிக்கும்.

```
c = 1          # global variable
def add():
    c = c + 2 # increment c by 2
    print(c)
add()
```

வெளியீடு

Unbound Local Error: local variable 'c' referenced before assignment

குறிப்பு

'global' என்ற சிறப்பு சொல்லைப் பயன்படுத்தாமல் குளோபல் மாறியை செயற்கூறின் உள்ளே மாற்றம் செய்ய முடியாது. ஆனால் அணுக மட்டுமே முடியும்.

எடுத்துக்காட்டு : செயற்கூறின் உள்ளிருந்து 'global' சிறப்புச் சொல்லைப் பயன்படுத்தி குளோபல் மாறியை மாற்றுதல்

```
x = 0                   # global variable
def add():
    global x
    x = x + 5           # increment by 2
    print ("Inside add() function  x value is :", x)
add()
print ("In main x value is :", x)
```

மேற்கண்ட நிரலை இயக்கும் போது பின்வரும் வெளியீடு தோன்றும்.

வெளியீடு

```
Inside add() function  x value is : 5
In main x value is : 5
```

மேலே உள்ள நிரலில், x என்பது குளோபல் மாறி ஆகும். add() செயற்கூறினுள், 'global' என்ற சிறப்புச் சொல் x மாறிக்கு பயன்படுத்தப்பட்டு அதனுடன் 5 என்ற மதிப்பு கூட்டப்படுகிறது. இப்பொழுது நாம் மாற்றம் செய்யப்பட்ட குளோபல் மாறி xயை செயற்கூறின் வெளியே காணலாம். அதாவது, x ன் மதிப்பு 5 ஆகும்.

7.8.3 குளோபல் மற்றும் உள்ளமை மாறிகள்

இங்கு, நாம் ஒரே குறிமுறையில் குளோபல் மற்றும் உள்ளமை மாறிகளை எவ்வாறு பயன்படுத்துவது என்பதைக் காணலாம்.

Chapter 7.indd 105 30-01-2020 19:30:59

எடுத்துக்காட்டு : ஒரே குறிமுறையில் குளோபல் மற்றும் உள்ளமை மாறிகளின் பயன்பாடு

```
x=8                 # x is a global variable
def loc():
        global x
        y = "local"
        x = x * 2
        print(x)
        print(y)
loc()
```

நிரலை இயக்கும் போது வெளியீடு பின்வருமாறு தோன்றும்

வெளியீடு

```
16
local
```

மேற்கண்ட நிரலில் x என்பது குளோபல் மாறியாகவும், y என்பது உள்ளமை மாறியாகவும் loc() செயற்கூறில அறிவிக்கப்பட்டுள்ளது. Loc() செயற்கூறினை அழைத்த பிறகு, xன் மதிப்பு 16 என மாறிவிடும். ஏனெனில் x=x*2 என பயன்படுத்தி உள்ளோம். அதன்பிறகு, உள்ளமை மாறி yயின் மதிப்பு அச்சிடப்பட்டுள்ளது.

எடுத்துக்காட்டு : ஒரே பெயரைக் கொண்ட குளோபல் மற்றும் உள்ளமை மாறிகள்

```
x = 5
def loc():
        x = 10
        print ("local x:", x)
loc()
print ("global x:", x)
```

வெளியீடு

```
local x: 10
global x: 5
```

மேற்கண்ட நிரலில், ஒரே பெயரைக் கொண்ட x மாறி குளோபல் மற்றும் உள்ளமை மாறியாக பயன்படுத்தப்பட்டுள்ளது. இரண்டு வரையெல்லையிலும் மாறியை அறிவித்துள்ளதால், அச்சிடும் போது வெவ்வேறான விடைக் கிடைக்கிறது. அதாவது, loc() செயற்கூறின் உள்ளே உள்ள x மாறி உள்ளமை வரையெல்லையாகவும் செயற்கூறின் வெளியே உள்ள *x* மாறி குளோபல் வரையெல்லையாகவும் இருக்கும்.

வெளியீடு :- local *x*: 10, is called local scope of variable.

வெளியீடு:- global *x*: 5, is called global scope of variable.

Chapter 7.indd 106 30-01-2020 19:30:59

7.9 உள்ளிணைந்த நூலகத்தை பயன்படுத்தும் செயற்கூறுகள்

7.9.1 உள்ளிணைந்த மற்றும் கணித செயற்கூறுகள்

செயற்கூறு	விளக்கம்	பொதுவடிவம்	எடுத்துக்காட்டு
abs ( )	எண்ணின் முழு எண்ணை திருப்பி அனுப்பும். செயலுருபானவது முழு எண்ணாகவோ அல்லது தசம எண்ணாகவோ இருக்கலாம்.	abs (x)	x=20 y=-23.2 print('x = ', abs(x)) print('y = ', abs(y)) **வெளியீடு** x = 20 y = 23.2
ord ( )	கொடுக்கப்பட்ட யுனிக்கோடு எழுத்திற்கு ASCII மதிப்பை திருப்பி அனுப்பும். இது chr() செயற்கூறின் தலைகீழாகும்.	ord (c)	c= 'a' d= 'A' print ('c = ',ord (c)) print ('A = ',ord (d)) **வெளியீடு** c = 97 A = 65
chr ( )	கொடுக்கப்பட்ட ASCII மதிப்பிற்கு யுனிக்கோடு எழுத்தை திருப்பி அனுப்பும். இது ord() செயற்கூறின் தலைகீழாகும்.	chr (i)	c=65 d=43 print (chr (c)) prin t(chr (d)) **வெளியீடு** A +
bin ( )	கொடுக்கப்பட்ட முழு எண்ணிற்கு நிகரான இரும எண்ணை "0b" யை முன்னொட்டமாக கொண்டு திருப்பி அனுப்பும். **குறிப்பு:** இதற்கு பதிலாக format () செயற்கூறுவை பயன்படுத்தலாம்.	bin (i)	x=15 y=101 print ('15 in binary : ',bin (x)) print ('101 in binary : ',bin (y)) **வெளியீடு** 15 in binary : 0b1111 101 in binary : 0b1100101

type ( )	கொடுக்கப்பட்ட பொருளின் தரவின வகையைத் திருப்பி அனுப்பும். **குறிப்பு:** ஒரு அளபுரு பொருளுடன் இது பயன்படுத்தப்படுகிறது.	type (object)	x= 15.2 y= 'a' s= True print (type (x)) print (type (y)) print (type (s)) **வெளியீடு** <class 'float'> <class 'str'> <class 'bool'>
id ( )	கொடுக்கப்பட்ட பொருளின் நினைவக முகவரியைத் திருப்பி அனுப்பும். **குறிப்பு:** x, y நினைவக முகவரி ஒவ்வொரு கணிப்பொறிக்கும் வேறுபடும்.	id (object)	x=15 y='a' print ('address of x is :',id (x)) print ('address of y is :',id (y)) **வெளியீடு** address of x is : 1357486752 address of y is : 13480736
min ( )	கொடுக்கப்பட்ட பட்டியலில் இருந்து மிகச்சிறிய மதிப்பைத் திருப்பி அனுப்பும்	min (list)	MyList = [21,76,98,23] print ('Minimum of MyList :', min(MyList)) **வெளியீடு** Minimum of MyList : 21
max ( )	கொடுக்கப்பட்ட பட்டியலில் இருந்து மிகப்பெரிய மதிப்பைத் திருப்பி அனுப்பும்	max (list)	MyList = [21,76,98,23] print ('Maximum of MyList :', max(MyList)) **வெளியீடு** Maximum of MyList : 98
sum ( )	கொடுக்கப்பட்ட பட்டியலில் உள்ள மதிப்புகளின் கூட்டுத் தொகையை திருப்பி அனுப்பும்	sum (list)	MyList = [21,76,98,23] print ('Sum of MyList :', sum(MyList)) **வெளியீடு** Sum of MyList : 218

Chapter 7.indd 108 30-01-2020 19:30:59

format ( )	கொடுக்கப்பட்ட எண்ணை வேறு எண்முறைக்கு மாற்றி திருப்பி அனுப்புகிறது. 1. இருமநிலை வடிவம். வெளியீட்டை இருமநிலை வடிவில் கொடுக்கும் 2. 2. எண்ணிலை வடிவம். வெளியீட்டை எண்ணியை வடிவில் கொடுக்கும் 8. 3. நிலையான புள்ளி வடிவம். வெளியீட்டை நிலையான புள்ளி எண் வடிவில் கொடுக்கும் தானமைவு தசம துல்லியம் 6 ஆகும்.	format (value [, format_spec])	x= 14 y= 25 print ('x value in binary :',format(x,'b')) print ('y value in octal :',format(y,'o')) print('y value in Fixed-point no ',format(y,'f')) **வெளியீடு** x value in binary : 1110 y value in octal : 31 y value in Fixed-point no : 25.000000
round ( )	கொடுக்கப்பட்ட எண்ணிற்கு அருகே உள்ள முழு எண்ணாக மாற்றி திருப்பி அனுப்பும். 1. முதல் செயலுருபு கொடுக்கப்பட்ட எண்ணைக் குறிக்கும் 2. இரண்டாவது செயலுருபு முழு எண்ணிற்கு பிறகு தசம புள்ளிகளின் எண்ணிக்கையைக் குறிக்கும்.	round (number [,ndigits])	x= 17.9 y= 22.2 z= -18.3 print ('x value is rounded to', round (x)) print ('y value is rounded to', round (y)) print ('z value is rounded to', round (z)) **வெளியீடு:1** x value is rounded to 18 y value is rounded to 22 z value is rounded to -18 n1=17.89 print (round (n1,0)) print (round (n1,1)) print (round (n1,2)) **வெளியீடு:2** 18.0 17.9 17.89

pow ( )	கொடுக்கப்பட்ட எண்ணின் ab அடுக்கு பெருக்கத்தை திருப்பி அனுப்பும். (a**b) ன் அடுக்கு b.	pow (a,b)	a= 5 b= 2 c= 3.0 print (pow (a,b)) print (pow (a,c)) print (pow (a+b,3)) **வெளியீடு:** 25 125.0 343

கணித செயற்கூறுகள்

குறிப்பு

அனைத்து கணித செயற்கூறுகளுக்கும் தேவையான முக்கியமான math கூறுகள்

செயற்கூறு	விளக்கம்	பொது வடிவம்	எடுத்துக்காட்டு
floor ( )	x-யை விடக்குறைவான அல்லது x-க்கு நிகரான பெரிய முழு எண்ணைத் திருப்பி அனுப்பும்.	math.floor (x)	import math x=26.7 y=-26.7 z=-23.2 print (math.floor (x)) print (math.floor (y)) print (math.floor (z)) **வெளியீடு:** 26 -27 -24
ceil ( )	x-யை விட பெரிய அல்லது xக்கு நிகரான சிறிய முழு எண்ணைத் திருப்பி அனுப்பும்.	math.ceil (x)	import math x= 26.7 y= -26.7 z= -23.2 print (math.ceil (x)) print (math.ceil (y)) print (math.ceil (z)) **வெளியீடு:** 27 -26 -23

sqrt ( )	*x*- ன் வர்க்கமூலத்தை திருப்பி அனுப்பும். குறிப்பு: *x*-ன் மதிப்பு பூஜ்ஜியத்தை விட பெரிய மதிப்பாக இருத்தல் வேண்டும்.	sqrt (x)	import math a= 30 b= 49 c= 25.5 print (math.sqrt (a)) print (math.sqrt (b)) print (math.sqrt (c)) **வெளியீடு:** 5.477225575051661 7.0 5.049752469181039

7.9.2 செயற்கூறில் தொகுப்பு (Composition)

செயற்கூறுகளின் தொகுப்பு என்றால் என்ன?

செயற்கூறுகளின் தொகுப்பு என்றால் என்ன? செயற்கூறு திருப்பி அனுப்பும் மற்றொரு செயற்கூறிற்கு செயலுருபாக, பின்னலான அமைப்பில் பயன்படுத்தினால் அதற்கு **தொகுப்பு (composition)** என்று பெயர். எடுத்துக்காட்டாக, பயனரிடமிருந்து எண் மதிப்பை அல்லது கோவையை உள்ளீடாகப் பெற விரும்பினால், input() செயற்கூறு மூலம் பயனரிடமிருந்து சரத்தை உள்ளீடாகப் பெற்று eval() செயற்கூறு மூலம் அதன் மதிப்பை மதிப்பீடு செய்ய வேண்டும்.

எடுத்துக்காட்டு :

```
# இந்த நிரல் தொகுப்பை விளக்குகிறது
        >>> n1 = eval (input ("Enter a number: "))
Enter a number: 234
        >>> n1
        234
>>> n2 = eval (input ("Enter an arithmetic expression: "))
        Enter an arithmetic expression: 12.0+13.0 * 2
>>> n2
        38.0
```

7.10 பைத்தான் –தற்சுழற்சி செயற்கூறுகள்

ஒரு செயற்கூறு தன்னைத் தானே அழைத்தால் அதை தற்சுழற்சி என்றழைக்கப்படும். மடக்கினை போன்ற தற்சுழற்சியும் செயல்படும் ஆனால் சில சமயங்களில் மடக்கினை பயன்படுத்துவதற்கு பதில் தற்சுழற்சியை பயன்படுத்தலாம். எந்தவொரு மடக்கினையும் தற்சுழற்சியாக மாற்றமுடியும்.

தற்சுழற்சி செயற்கூறு தன்னைத்தானே அழைக்கும். ஒரு நிபந்தனையில் நிறுத்தப்படாவிட்டால், ஒரு செயல்முறை காலவரையின்றி செயல்படும். இந்த செயற்பாட்டை காலவரையின்றி செயல்படும் தற்சுழற்சி எனப்படும். தற்சுழற்சி செயற்கூறு கொடுக்கப்படும் நிபந்தனையை அடிப்படை நிபந்தனை எனப்படும். ஒவ்வொரு தற்சுழற்சி செயற்கூற்றிக்கும் அடிப்படை நிபந்தனை கொடுக்கப்பட வேண்டும் இல்லையென்றால் மடக்கு காலவரையின்றி இயங்கும்.

தற்சுழற்சி செயற்கூறு எவ்வாறு செயல்படும்

1. தற்சுழற்சி செயற்கூறு வெளிபுர குறிமுறையிலிருந்து அழைக்கப்படும்.

2. அடிப்படை நிபந்தனை நிறைவேற்றப்பட்டால் நிரலானது ஏற்ற வெளியீடு கொடுத்து வெளியேறும்.

3. இல்லையெனில், செயற்கூறானது தேவையான செயற்பாட்டை இயக்கும் மேலும் தற்சுழற்சி முறையில் தன்னைத் தானே அழைத்துக் கொள்ளும்.

தற்சுழற்சி பயன்படுத்தி காரணிபடுத்துதலுக்கு எடுத்துக்காட்டு.

எடுத்துக்காட்டு :

```
def fact(n):
        if n == 0:
                return 1
        else:
                return n * fact (n-1)
print (fact (0))
print (fact (5))
```

வெளியீடு

```
        1
        120
```

உங்களுக்குத் தெரியுமா?

print (fact (2000)) கூற்று Runtime பிழைச் செய்தியைக் கொடுக்கும். அதிகபட்ச மறு நிகழ்வின் சுழற்சியை ஒப்பிடுகையில் இது அதிகம். இது எதனால் எனில், தானமைவாக 1000 அழைப்புகளுக்குப் பிறகு பைத்தான் தன்னைத்தானே அழைப்பதை நிறுத்திவிடும். வரம்புகளை (limit) sys.setrecursionlimit (limit_value) பயன்படுத்தி மாற்றியமைக்க அனுமதிக்கிறது.

எடுத்துக்காட்டு

```
import sys
sys.setrecursionlimit(3000)
def fact(n):
        if n == 0:
        return 1
         else:
        return n * fact(n-1)
print(fact (2000))
```

நினைவில் கொள்க

- செயற்கூறுகள் ஒரு குறிப்பிட்ட செயலினை செய்வதற்காக வடிவமைக்கப்பட்டு பெயரிடப்பட்ட குறிமுறையின் தொகுதியே செயற்கூறு எனப்படும்.
- பயனர் வரையறைத்த, உள்ளிணைந்த, லாம்டா மற்றும் தற்சுழற்சிக்கு செயற்கூறுகளின் வகைகள் ஆகும்.
- செயற்கூறு தொகுதி "def" என்ற சிறப்புச் சொல்லுடன் தொடங்கி செயற்கூறுவின் பெயர் மற்றும் ( ) அடைப்புக்குறியுடன் முடிய வேண்டும்.
- செயலுறுப்புகளில் இல்லாத return, None - ற்க்கு நிகரானது. return கூற்றொரு விருப்பத்தேர்வுயில்லை
- பைத்தானில், ஒரு தொகுதிக்குள் உள்ள கூற்றுகள் உள்தள்ளிருக்க வேண்டும்.
- ஒரு தொகுதிக்குள் வேறொரு தொகுதியிருப்பின் அதை பின்னலான தொகுதி எனப்படும்.
- செயற்கூறை அழைப்பதற்கு செயலுருபுகள் பயன்படுகின்றன. 4 வகையான செயலுருப்புக்கள் செயற்கூறில் பயன்படுத்தாம்: தேவையான செயலுருபுக்கள், சிறப்பு செயல் செயலுருப்புகள், தானமைவு செயலுருபுக்கள் மற்றும் மாறும்-நீள செயலுருப்புகள்.
- தேவையான செயலுருபுகள் செயற்கூறில் சரியான இடத்திற்கு அனுப்பப்பட வேண்டிய செயலுருபுகளாகும்.
- அளபுருக்களின் பெயரை அடையாளம் கண்ட பின்பு சிறப்புச் சொல் செயலுருபானது செயற்கூறினை அழைக்கிறது.
- பைத்தானில், செயற்கூறை அழைக்கும் போது எந்த மதிப்பும் கொடுக்கப்படவில்லை எனில் செயலுருபானது தானாகவே மதிப்பை எடுத்துக் கொள்ளும்.
- * குறியீடு மாறும்-நீள செயலுருபுகளை வரையறைக்க பயன்படும்.
- பெயரில்லாமல் வரையறுக்கப்படும் செயற்கூறுவுக்கு பெயரில்லாத செயற்கூறாகும்.
- மாறிகளின் வரையெல்லை, நிரலின் அணுகக் கூடிய பகுதியைக் குறிப்பதாகும்.
- செயற்கூறு திருப்பி பின்னலான அமைப்பில் பயன்படுத்தினால் அதற்கு தொகுப்பு. என்று பெயர்.
- ஒரு செயற்கூறு தன்னைத்தானே அழைத்தால் அதை தற்சுழற்சி என்றழைக்கப்படும். மடக்கினை போன்ற தற்சுழற்சியும் செயல்படும். ஆனால் சில சமயங்களில் மடக்கினை பன்படுத்துவதற்கு பதில் தற்சுழற்சியை பயன்படுத்தலாம்.

Chapter 7.indd 113 30-01-2020 19:30:59

செய்முறைப் பயிற்சி:1

எடுத்துக்காட்டு 7.5.3-ல் உள்ள நிரலில் பின்வரும் குறிமுறையை முயற்சித்துப் பார்க்கவும்.

வ.எண்	குறிமுறை	வெளியீடு
1	printinfo(“3500”)	
2	printinfo(“3500”,”Sri”)	
3	printinfo(name=”balu”)	
4	printinfo(“Jose”,1234)	
5	printinfo(“ ”,salary=1234)	

செய்முறைப் பயிற்சி :2

பின்வரும் செயற்கூறுகளை மதிப்பீடு செய்து அதன் வெளியீட்டை எழுதுக.

வ.எண்	செயற்கூறு	வெளியீடு
1	1) abs(-25+12.0)) 2) abs(-3.2)	
2	1) ord('2') 2) ord('$')	
3	type('s')	
4	bin(16)	
5	1) chr(13) 2) print(chr(13))	
6	1) round(18.2,1) 2) round(18.2,0) 3) round(0.5100,3) 4) round(0.5120,3)	
7	1) format(66, 'c') 2) format(10, 'x') 3) format(10, 'X') 4) format(0b110, 'd') 5) format(0xa, 'd')	

Chapter 7.indd 114 30-01-2020 19:30:59

8	1) pow(2,-3) 2) pow(2,3.0) 3) pow(2,0) 4) pow((1+2),2) 5) pow(-3,2) 6) pow(2*2,2)	

செய்முறைப் பயிற்சி:3

பின்வரும் செயற்கூறுகளை மதிப்பீடு செய்து அதன் வெளியீட்டை எழுதுக

வ.எண்	செயற்கூறு	வெளியீடு
1	eval('25*2-5*4')	
2	math.sqrt(abs(-81))	
3	math.ceil(3.5+4.6)	
4	math.floor(3.5+4.6)	

மதிப்பீடு

பகுதி–அ

சரியான விடையை தேர்ந்தெடுத்து எழுதுக **(1 மதிப்பெண்)**

1. ஒரு குறிப்பிட்ட செயலைச் செய்வதற்காக வடிவமைக்கப்பட்டு, பெயரிடப்பட்ட குறிமுறையின் தொகுதி

   (அ) மடக்கு (ஆ) கிளைப்பிரிப்பு

   (இ) செயற்கூறு (ஈ) தொகுதி

2. தன்னைத்தானே அழைத்துக் கொள்ளும் செயற்கூறை இவ்வாறு அழைப்பர்.

   (அ) உள்ளிணைந்த (ஆ) தற்சுழற்சி

   (இ) லாம்டா (ஈ) return கூற்று

ZSUKEV

3. எந்த செயற்கூறை பெயரில்லா செயற்கூறு என்று அழைக்கப்படுகிறது?

   (அ) லாம்டா (ஆ) தற்சுழற்சி

   (இ) செயற்கூறு (ஈ) வரையறை

4. செயற்கூறு தொகுதியை எந்த சிறப்புச்சொல் தொடங்கிவைக்கிறது?

(அ) define (ஆ) for

(இ) finally (ஈ) def

5. எந்த சிறப்புச்சொல் செயற்கூறு தொகுதியை முடித்து வைக்கிறது?

(அ) define (ஆ) return

(இ) finally (ஈ) def

6. செயற்கூறு வரையிறையில் பின்வரும் எந்த குறியீடு பயன்படுத்தப்படுகிறது?

(அ) ; (அரைப் புள்ளி) (ஆ) . (புள்ளி)

(இ) : (முக்காற் புள்ளி) (ஈ) $ (டாலர்)

7. செயற்கூறுக்கு எந்த செயலுருபு சரியான இட வரிசையில் செயலுருபுகளை அனுப்பும்?

(அ) தேவைப்படும் (ஆ) சிறப்புச்சொல்

(இ) தானமைவு (ஈ) மாறிநீளம்

8. பின்வரும் கூற்றுகளைப் படித்து, சரியான கூற்றுகளைத் தேர்ந்து எடுக்கவும்.

(I) பைத்தானில், செயற்கூறை வரையறுக்கும் போது குறிப்பிட்ட தரவு வகைகளைக் குறிப்பிடத் தேவையில்லை

(II) பைத்தான் சிறப்புச் சொற்களைச் செயற்கூறின் பெயராகப் பயன்படுத்தலாம்.

(அ) I சரி மற்றும் II தவறு (ஆ) இரண்டுமே சரி

(இ) I தவறு மற்றும் II சரி (ஈ) இரண்டுமே தவறு

9. கொடுக்கப்பட்ட கூற்றை வெற்றிகரமாக நிறை வேற்றுவதற்கு, பின்வருவனவற்றுள் சரியான ஒன்றைத் தேர்ந்தெடு

if _____ : print(x, " is a leap year")

(அ) x%2=0 (ஆ) x%4==0

(இ) x/4=0 (ஈ) x%4=0

10. testpython() செயற்கூறைவரையறுக்க பின்வரும் எந்த சிறப்புச் சொல்பயன்படுகிறது?

(அ) define (ஆ) pass

(இ) def (ஈ) while

பகுதி–ஆ

அனைத்து வினாக்களுக்கும் விடையளி (2 மதிப்பெண்)

1. செயற்கூறு என்றால் என்ன?
2. செயற்கூறின் வகைகளை எழுதுக.
3. செயற்கூறுவின் முக்கிய நன்மைகள் யாவை?
4. மாறியின் வரையெல்லை என்றால் என்ன? அதன் வகைகளைக் குறிப்பிடுக.

5. குளோபல் வரையெல்லை-வரையறு

6. தன்னைத் தானே அழைக்கும் செயற்கூறில் அடிப்படை நிபந்தனை என்றால் என்ன?

7. தன்னைத்தானே அழைக்கும் செயற்கூறுக்கு வரம்பை எவ்வாறு அமைக்க வேண்டும்? எடுத்துக்காட்டு தருக.

பகுதி–இ

அனைத்து வினாக்களுக்கும் விடையளி **(3 மதிப்பெண்)**

8. உள்ளமை மாறிகளுக்கான விதிமுறைகளை எழுதுக.

9. பைத்தானிலுள்ள முழுதளாவி சிறப்புச் சொல்லுக்கான அடிப்படை விதிமுறைகளை எழுதுக.

10. செயற்கூறினுள் முழுதளாவி மாறியை மாற்றம் செய்தால் என்ன நிகழும்?

11. ceil() மற்றும் floor() செயற்கூறுகளை வேறுபடுத்துக.

12. கொடுக்கப்பட்ட வருடம் லீப் வருடமா இல்லையா என்பதனைச் சோதிக்கும் பைத்தான் நிரலை எழுதுக.

13. செயற்கூறில் தொகுப்பு என்பது என்ன?

14. தற்சுழற்சி எவ்வாறு செயல்படுகிறது?

15. செயற்கூறினை வரையறுக்கும் போது குறிப்பிடப்பட வேண்டிய குறிப்புகள் யாவை?

பகுதி–ஈ

அனைத்து வினாக்களுக்கும் விடையளி **(5 மதிப்பெண்)**

16. செயற்கூறின் வகைகளை எடுத்துக்காட்டுடன் விவரி

17. மாறியின் வரையெல்லைகளை எடுத்துக்காட்டுடன் விளக்குக.

18. பின்வரும் உள்ளிணைந்த செயற்கூறுகளை விளக்குக.

    (a) id()

    (b) chr()

    (c) round()

    (d) type()

    (e) pow()

19. இரண்டு எண்களின் LCM கண்டுப்பிடிப்பதற்கான பைத்தான் நிரலை எழுதுக.

20. தற்சுழற்சி செயற்கூறு பற்றி எடுத்துக்காட்டுன் விளக்குக.

Reference Books

1. *Python Tutorial book from tutorialspoint.com*

2. *Python Programming: A modular approach by Pearson – Sheetal, Taneja*

3. *Fundamentals of Python –First Programs by Kenneth A. Lambert*

அலகு II

பாடம் 8

சரங்கள் மற்றும் சரங்களை கையாளுதல்

AZLW14

கற்றலின் நோக்கங்கள்

இப்பாடத்தைக் கற்றறிந்த பிறகு, மாணவர்கள் பின்வரும் செயல்பாடுகளை செய்ய இயலும்.

- உரையை செயல்படுத்தும் வழிமுறைகள் பற்றி தெரிந்து கொள்ளுதல்.
- பைத்தானில் உள்ள சரங்களை கையாளும் செயற்கூறுகள் பற்றி புரிந்து கொள்ளுதல்.
- சரங்களை வடிவூட்டல் செய்யும் முறைகள் பற்றி அறிதல்.
- சரங்களை பிரிக்கும் முறை பற்றி தெரிந்து கொள்ளுதல்.
- நடப்புலகில் சரங்களின் பயன்பாடு பற்றி தெரிந்து கொள்ளுதல்.

8.1 அறிமுகம்

குறியுருக்களின் அணியை கையாளுவதற்கான ஒரு தரவு இனம் சரம் (String) ஆகும். சரங்கள் என்பன, ஒற்றை, இரட்டை, அல்லது மூன்று மேற்கோள் குறிகளுக்குள் கொடுக்கப்படும், எழுத்து, எண்கள் அல்லது சிறப்பு குறியீடுகளின் சேர்க்கையை கொண்டிருக்கும் ஒருங்கமைவு (Unicode) குறியுருக்களின் வரிசை ஆகும்.

எடுத்துக்காட்டு

'Welcome to learning Python'

"Welcome to learning Python"

''' "Welcome to learning Python" '''

பைத்தானில் சர தரவுவகையானது மாற்ற இயலாதது. அதாவது சரத்தரவு வகையை வரையறுத்த பின்பு இயங்கு நேரத்தில் அதை மாற்றியமைக்க முடியாது.

8.2 சரம் உருவாக்குதல்

நாம் முன்பே கற்றது போல பைத்தானில் சரமானது ஒற்றை மேற்கோள் குறிகளுக்குள்ளோ அல்லது இரட்டை மேற்கோள் குறிகளுக்குள்ளோ அல்லது மூன்று மேற்கோள் குறிகளையோ கொண்டு உருவாக்கப்படும். ஒற்றை மேற்கோள் குறிகளானது மற்றுமொரு ஒற்றை மேற்கோள் குறிகளை கொண்ட குறியுருவை கொண்டிருக்க முடியாது. ஏனெனில் நிரல் பெயர்ப்பியானது சரத்தின் தொடக்கம் மற்றும் முடிவை அறிய இயலாது. இந்த சிக்கலை தீர்க்க இரட்டை மேற்கோள் குறிகளை பயன்படுத்தலாம். சரமானது இரட்டை மேற்கோள் குறிகளையும் உறுப்பாக பெற்றிருந்தால் அது மூன்று மேற்கோள் குறிகளுக்குள்

Chapter 8.indd 118 10-02-2020 20:01:04

கொடுக்கப்படல் வேண்டும். மூன்று மேற்கோள் குறிகளுக்குள் கொடுக்கப்படும் சரமானது பல வரி சரங்களை உருவாக்க அனுமதிக்கும்.

எடுத்துக்காட்டு

#ஒற்றை மேற்கோள் குறிகளுக்குள் வரையறுக்கப்பட்டுள்ள சரம்

```
>>> print ('Greater Chennai Corporation')
        Greater Chennai Corporation
```

ஒற்றை மேற்கோள் குறிகளுக்குள், ஒற்றை மேற்கோள் குறியுடன் வரையறுக்கப்பட்ட சரம்.

```
>>> print ('Greater Chennai Corporation's student')
        SyntaxError: invalid syntax
```

இரட்டை மேற்கோள் குறிகளுக்குள் வரையறுக்கப்பட்ட சரம்

```
>>>print ("Computer Science")
        Computer Science
```

மூன்று மேற்கோள் குறிகளுக்குள், இரட்டை மேற்கோள் குறியுடன் வரையறுக்கப்பட்ட சரம்

```
>>> print (''' "Computer Science" ''')
        "Computer Science"
```

மூன்று மேற்கோள் குறிகளுக்குள், ஒற்றை மற்றும் இரட்டை மேற்கோள் குறியுடன் வரையறுக்கப்பட்ட பலவரி சரம்.

```
>>> print (''' "Strings are immutable in 'Python',
              which means you can't make any changes
              once you declared" ''')

        "Strings are immutable in 'Python',
        which means you can't make any changes once you declared"
```

8.3 சரத்தில் உள்ள குறியுருக்களை அணுகுதல்

சரம் ஒருமுறை வரையறுக்கப்பட்டு விட்டால், பைத்தான் சரத்தில் உள்ள ஒவ்வொரு எழுத்துருவிற்கும் ஒரு சுட்டெண்ணை (Index) ஒதுக்கீடு செய்யும். சரத்தின் உறுப்புகளை அணுகுவதற்கும், கையாளுடுவதற்கும் பயன்படும் சுட்டெண் கீழ்ஒட்டு (Subscript) எனவும் அழைக்கப்படுகிறது. கீழ் ஒட்டானது நேர்மறை அல்லது எதிர்மறை முழு எண்ணாக இருக்கலாம்.

நேர்மறை கீழ் ஒட்டு 0 என்பது சரத்தின் முதல் குறியுருவிற்கும் n-1 கீழ் ஒட்டு சரத்தின் இறுதி குறியுருவிற்கும் ஒதுக்கப்படும். n என்பது சரத்தில் உள்ள குறியுருக்களின் எண்ணிக்கை ஆகும். எதிர்மறை கீழ் ஒட்டு சரத்தின் இறுதி குறியுருவிலிருந்து துவங்கி முதல் குறியுரு வரை பின்னோக்கு வரிசையில் -1 முதல் ஒதுக்கப்படும்.

எடுத்துக்காட்டு

சரம்	S	C	H	O	O	L
நேர்மறை கீழ்ஒட்டு	0	1		3	4	5
எதிர்மறை கீழ்ஒட்டு	-6	-5	-4	-3	-2	-1

எடுத்துக்காட்டு 8.1 கொடுக்கப்பட்ட சரத்தில் உள்ள குறியுருக்களை அதன் நேர்மறை கீழ்ஒட்டை கொண்டு அணுகும் நிரல்.

```
str1 = input ("Enter a string: ")
index=0
for i in str1:
        print ("Subscript[",index,"] : ", i)
        index + = 1
```

வெளியீடு

```
        Enter a string: welcome
        Subscript [ 0 ] :  w
        Subscript [ 1 ] :  e
        Subscript [ 2 ] :  l
        Subscript [ 3 ] :  c
        Subscript [ 4 ] :  o
        Subscript [ 5 ] :  m
        Subscript [ 6 ] :  e
```

எடுத்துக்காட்டு 8.2 கொடுக்கப்பட்ட சரத்தில் உள்ள குறியுருக்களை அதன் எதிர்மறை கீழ்ஒட்டை கொண்டு அணுகும் நிரல்

```
str1 = input ("Enter a string: ")
index=-1
while index >= -(len(str1)):
        print ("Subscript[",index,"] : " + str1[index])
        index += -1
```

வெளியீடு

```
        Enter a string: welcome
        Subscript [ -1 ] : e
        Subscript [ -2 ] : m
        Subscript [ -3 ] : o
        Subscript [ -4 ] : c
        Subscript [ -5 ] : l
        Subscript [ -6 ] : e
        Subscript [ -7 ] : w
```

சரத்தை திருத்துதல் மற்றும் நீக்குதல்

நாம் ஏற்கனவே கற்றது போல் பைத்தானில் சரங்களை மாற்றியமைக்க முடியாது. அதாவது சரத்தை ஒருமுறை வரையறுத்த பின்பு அதை திருத்துதல் அல்லது நீக்குதல் போன்ற செயல்பாடுகள் அனுமதிக்கப்பட மாட்டாது. சரத்தை திருத்த விரும்பினால் புதிய சர மதிப்பானது ஏற்கனவே உள்ள சர மாறிக்கு ஒதுக்கப்படும்.

எடுத்துக்காட்டு

```
>>> str1="How are you"
>>> str1[0]="A"
Traceback (most recent call last):
        File "<pyshell#1>", line 1, in <module>
        str1[0]="A"
    TypeError: 'str' object does not support item assignment
```

மேற்கண்ட எடுத்துக்காட்டு நிரலில் strl என்ற சரமாறிக்கு “How are you” என்ற சரமானது முதல் கூற்றின் மூலம் ஒதுக்கப்பட்டு உள்ளது. அடுத்த கூற்றில், சரத்தின் முதல் குறியுருவை 'A' என்ற குறியுருவை கொண்டு மாற்ற முயற்சிக்கப்பட்டுள்ளது. ஆனால் பைத்தான் இம்மாற்றத்தை அனுமதிக்காது. எனவே, TYPE ERROR என்ற பிழை செய்தியை காட்டும்.

இச்சிக்கலை தீர்க்க ஏற்கனவே உருவாக்கப்பட்டுள்ள சரமாறிக்கு புதிய சர மதிப்பினை வரையறுக்க முடியும். பைத்தான் ஏற்கனவே உள்ள சரத்திற்கு பதிலாக புதிய சரத்தினை மாற்றிவிடும்.

எடுத்துக்காட்டு

```
>>> str1="How are you"
>>> print (str1)
    How are you
>>> str1="How about you"
>>> print (str1)
    How about you
```

பொதுவாக, பைத்தான் தனது சரங்களின் மீது எந்த வித மாற்றத்தையும் செய்ய அனுதிக்காது. ஆனால், replace() என்ற செயற்கூறு மூலம் ஏற்கனவே உள்ள சரத்தில் ஒரு குறிப்பிட்ட குறியுரு உள்ள இடங்களில் எல்லாம் வேறு ஒரு குறியுருவை மாற்றிட முடியும்.

replace() செயற்கூறுக்கான தொடரியல்

replace(“char1”, “char2”)

replace() செயற்கூறு char1 குறியுரு வரும் இடங்களில் எல்லாம் char 2 குறியுருவைக் கொண்டு மாற்றிவிடும்.

எடுத்துக்காட்டு

```
>>> str1="How are you"
>>> print (str1)
    How are you
>>> print (str1.replace("o", "e"))
    Hew are yeu
```

சரத்தை திருத்துவது போல், பைத்தான் சரத்தில் உள்ள ஒரு குறிப்பிட்ட குறியுருவை நீக்கவும் அனுமதிக்காது. ஆனால் del- கட்டளை பயன்படுத்தி ஒரு முழு சர மாறியையும் நீக்க முடியும்.

எடுத்துக்காட்டு 1: கீழ்க்காணும் குறிமுறை சரத்தில் உள்ள குறிப்பிட்ட குறியுருவை நீக்குகிறது.

```
>>> str1="How are you"
>>> del str1[2]
Traceback (most recent call last):
        File "<pyshell#7>", line 1, in <module>
                del str1[2]
        TypeError: 'str' object doesn't support item deletion
```

எடுத்துக்காட்டு 2: கீழ்க்காணும் குறிமுறை ஒரு சரமாறியை நீக்குகிறது:

```
>>> str1="How about you"
>>> print (str1)
        How about you
>>> del str1
>>> print (str1)
Traceback (most recent call last):
        File "<pyshell#14>", line 1, in <module>
                print (str1)
        NameError: name 'str1' is not defined
```

8.4 சர செயற்குறிகள்

பைத்தான் பின்வரும் செயற்குறிகளை வழங்குகிறது. இச்செயற்குறிகள் சரங்களை கையாள உதவுகிறது.

8.4.1 இணைப்பு (Concatenation +)

இரண்டு அல்லது அதற்கு மேற்பட்ட சரங்களை இணைக்கும் செயல்பாடு சேர்த்தல்/இணைத்தல் எனப்படும். கூட்டல் செயற்குறியானது சரங்களை பைத்தானில் இணைத்துக் கொள்ளப் பயன்படுகிறது.

எடுத்துக்காட்டு

```
>>> "welcome" + "Python"
        'welcomePython'
```

8.4.2 சேர்த்தல் (Append +=)

ஏற்கனவே உள்ள சரத்தின் இறுதியில் மேலும் புதிய சரம் அல்லது சரங்களை சேர்க்கும் செயல் சேர்த்தல் எனப்படும்.

+= செயற்குறி ஏற்கனவே உள்ள சரத்தின் இறுதியில் புதிய சரத்தினை சேர்க்க பயன்படுகிறது.

எடுத்துக்காட்டு

```
>>> str1="Welcome to "
>>> str1+="Learn Python"
>>> print (str1)
        Welcome to Learn Python
```

8.4.3 பலமுறை (Repeating (*))

பெருக்கல் செயற்குறி கொடுக்கப்பட்ட சரத்தினை பல தடவைகள் வெளிப்படுத்த பயன்படுகிறது.

எடுத்துக்காட்டு

```
>>> str1="Welcome "
>>> print (str1*4)
        Welcome Welcome Welcome Welcome
```

8.5 சரத்தை துண்டாக்குதல் (அல்லது) சரத்தை பகுதியாக பிரித்தல்

மூலச்சரத்தில் உள்ள ஒரு துணைச்சரம் (சரத்தின் ஒரு பகுதி) துண்டு (Slice) எனப்படும். மூலச் சரத்திலிருந்து [ ] என்ற செயற்குறி மற்றும் சுட்டு அல்லது கீழ்ஒட்டு மதிப்புகளைக் கொண்டு துணைச் சரம் உருவாக்கப்படும். இதனால் [ ] செயற்குறி துண்டு அல்லது பிரித்தல் செயற்குறி எனப்படும். துண்டு அல்லது பிரித்தல் செயற்குறியை கொண்டு ஒன்று அல்லது அதற்கு மேற்பட்ட துணைச் சரங்களை / (துண்டுகளாக) மூலச் சரத்திலிருந்து பிரிக்க முடியும்.

பிரிப்பதற்கான தொடரியல்

str[start:end]

Start என்பது துவக்க சுட்டு மதிப்பு ஆகும். மேலும் end என்பது சரத்தில் உள்ள இறுதி குறியுருவின் சுட்டு மதிப்பு ஆகும். பைத்தான் குறிப்பிடப்பட்டுள்ள இறுதி சுட்டு எண் மதிப்பிலிருந்து ஒரு மதிப்பை குறைத்து எடுத்துக் கொள்ளும்.

எடுத்துக்காட்டாக, முதல் 4 குறியுருக்களை மட்டும் சரத்திலிருந்து பிரிக்க விரும்பினால் சுட்டு எண் மதிப்பை 0வில் இருந்து 5 என குறிப்பிட வேண்டும். ஏனெனில் பைத்தான் இறுதி மதிப்பிலிருந்து ஒரு மதிப்பை குறைத்து மட்டுமே கருத்தில் கொள்ளும்.(இறுதி மதிப்பு -1)

எடுத்துக்காட்டு 1: சரத்தில் இருந்து ஒரு குறியுருவை மட்டும் பிரித்தல்

```
>>> str1="THIRUKKURAL"
>>> print (str1[0])
    T
```

எடுத்துக்காட்டு 2: சுட்டு எண் மதிப்பு 0 முதல் 4 வரை உள்ள துணைச் சரத்தை பிரித்தல்

Chapter 8.indd 123 10-02-2020 20:01:05

```
>>> print (str1[0:5])
```

THIRU

எடுத்துக்காட்டு 3: துவக்க சுட்டு எண் மதிப்பு குறிப்பிடப்படாமல் 0 முதல் 4 வரை சுட்டெண்ணைக் கொண்ட துணைச் சரத்தை பிரித்தல்

```
>>> print (str1[:5])
```

THIRU

எடுத்துக்காட்டு 4: இறுதி சுட்டு எண் மதிப்பு குறிப்பிடப்படாமல் 6 முதல் 10 வரை சுட்டெண்களாகக் கொண்ட துணைச் சரத்தை பிரித்தல்

```
>>> print (str1[6:])
```

KURAL

எடுத்துக்காட்டு 8.3 for மடக்கை பயன்படுத்தி துணைச் சரத்தை பிரிக்க துண்டாக்கும் நிரல்

```
str1="COMPUTER"
index=0
for i in str1:
        print (str1[:index+1])
        index+=1
```

வெளியீடு

```
C
CO
COM
COMP
COMPU
COMPUT
COMPUTE
COMPUTER
```

8.6 மூன்றாம் அளபுரு (Stride) வைப் பயன்படுத்தி சரத்தை பிரித்தல்

சரத்தினை பிரிக்கும் / துண்டாக்கும் செயல்பாட்டில் மூன்றாவது அளபுருவையும் குறிப்பிட முடியும். சரத்தில் குறிக்கப்பட்டுள்ள முதல் குறியுரு அணுகப்பட்டத்திலிருந்து எத்தனை எண்ணிக்கையிலான குறியுருக்கள் முன்னோக்கி செல்ல வேண்டும் என்பதை குறிக்க மூன்றாம் அளபுரு பயன்படுகிறது. மூன்றாம் அளபுருவின் கொடாநிலை மதிப்பு 1 ஆகும்.

Chapter 8.indd 124 10-02-2020 20:01:05

எடுத்துக்காட்டு

```
>>> str1 = "Welcome to learn Python"
>>> print (str1[10:16])
        learn
>>> print (str1[10:16:4])
        r
>>> print (str1[10:16:2])
        er
>>> print (str1[::3])
Wceoenyo
```

குறிப்பு : பைத்தான் இறுதி சுட்டெண் மதிப்பிலிருந்து ஒரு மதிப்பை குறைத்து எடுத்துக் கொள்ளும் என்பதை நினைவில் கொள்க.

மூன்றாவது அளபுருவில் எதிர்மறை மதிப்பையும் பயன்படுத்த முடியும். எதிர்மறை மதிப்பை குறிப்பிட்டால் சரமானது பின்னோக்கு வரிசையில் அச்சிடப்படும்.

எடுத்துக்காட்டு

```
>>> str1 = "Welcome to learn Python"
>>> print(str1[::-2])
        nhy re teolW
```

8.7 சரவடிவமைப்பு செயற்குறிகள்

பைத்தானில் உள்ள மிக முக்கியமான சிறப்பம்சங்களில் சரவடிவமைப்பு செயற்குறியும் ஒன்று. சரங்களை வடிவமைக்கவும், மாறியில் சேமிக்கப்பட்டுள்ள சரங்களின் பகுதியினை மாற்றியமைக்கவும் சரவடிவமைப்பு செயற்குறியான % பயன்படுகிறது.

தொடரியல்

("String to be display with %val1 and %val2" %(val1, val2))

எடுத்துக்காட்டு

```
name = "Rajarajan"
mark = 98
print ("Name: %s and Marks: %d" %(name,mark))
```

வெளியீடு

```
Name: Rajarajan and Marks: 98
```


Chapter 8.indd 125 10-02-2020 20:01:05

8.8 வடிவமைப்பு குறியுருக்கள்

வடிவமைப்பு குறியுரு	பயன்பாடு
%c	குறியுரு
%d (அல்லது) %i	குறியிட்ட முழு எண்
%s	சரம்
%u	குறியில்லா முழு எண்
%o	எண்ம முழு எண்
%x அல்லது %X	பதினறும எண் (சிறிய எழுத்து x, a-f வரையும், பெரிய எழுத்து X, A-F வரையும் குறிக்கும்)
%e அல்லது %E	அடுக்கெண்
%f	மதிப்பு புள்ளி எண்கள்
%g அல்லது %G	மிதப்பு புள்ளி வடிவிலான குறு எண் அல்லது அடுக்கு எண் குறியீட்டு முறை

8.9 பைத்தான் விடுபடு வரிசை

விடுபடு வரிசை பின் சாய்வுக் கோட்டை கொண்டு துவங்கும். ஒற்றை மேற்கோள் குறியுடன் சரத்தை குறிப்பிடும் போது சரத்தில் உள்ள அனைத்து ஒற்றை மேற்கோள் குறிகளும் விடுபடும்.

எடுத்துக்காட்டு

ஒற்றை மேற்கோள் குறியுடன் சரத்தை வெளியிடும் மூன்று மேற்கோள் குறிகளுக்குள் குறிப்பிடப்பட்டிருக்கும் சரம்

```
>>> print ('''They said, "What's there?"''')
        They said, "What's there?"
```

இரட்டை மேற்கோள் குறியுடன் தோன்றும் சரம் ஒற்றை மேற்கோள் குறிக்குள் விடுபடு வரிசையுடன்

```
>>> print ('They said, "What\'s there?"')
        They said, "What's there?"
```

இரட்டை மேற்கோள் குறியுடன் தோன்றும் சரம் இரட்டை மேற்கோள் குறிக்குள் விடுபடு வரிசையுடன்

```
>>> print ("They said, \"What's there?\"")
        He said, "What's there?"
```

பைத்தான் விடுபடுவரிசைகள்

விடுபடு வரிசை	விளக்கம்
\newline	பின்சாய்வு மற்றும் புதிய வரி தவிர்த்தல்
\\	பின்சாய்வு
\'	ஒற்றை மேற்கோள் குறி
\"	இரட்டை மேற்கோள் குறி

Chapter 8.indd 126 10-02-2020 20:01:05

\a	ASCII மணி ஒலிப்பு
\b	ASCII பின்னிட வெளி
\f	ASCII படிவ செலுத்தி
\n	ASCII வரி செலுத்தி
\r	ASCII நகர்த்தி திருப்பி
\t	ASCII கிடைமட்ட தத்தல்
\v	ASCII செங்குத்து தத்தல்
\ooo	எண்ம எண்ணுடன் கூடிய குறியுரு
\xHH	பதினறும எண்ணுடன் கூடிய குறியுரு

8.10 format() செயற்கூறு

format() செயற்கூறானது சரங்களை வடிவமைக்கப் பயன்படும் முக்கிய செயற்கூறாகும். நெளிவு அடைப்புக்குறி { } இட நிரப்பியாக அல்லது புலத்தின் பிரதியீடாக பயன்படுகிறது. இது format() செயற்கூறில் பயன்படுத்தப்படும்.

எடுத்துக்காட்டு

```
num1=int (input("Number 1: "))
num2=int (input("Number 2: "))
print ("The sum of { } and { } is { }".format(num1, num2,(num1+num2)))
```

வெளியீடு

```
Number 1: 34
Number 2: 54
The sum of 34 and 54 is 88
```

8.11 உள்ளிணைந்த சர செயற்கூறுகள்

சரங்களை கையாளுவதற்கு பைத்தானானது பின்வரும் உள்ளிணைந்த செயற்கூறுகளை ஆதரிக்கிறது.

தொடரியல்	விளக்கம்	எடுத்துக்காட்டு

len(str)	சரத்தின் நீளத்தை (குறியுருக்களின் எண்ணிக்கையை) திரும் தரும்	>>> A="Corporation"
		>>> print(len(A))
		11

capitalize( )	சரத்தின் முதல் குறியுருவை பெரிய எழுத்தாக மாற்ற பயன்படுகிறது.	>>> city="chennai"
		>>> print(city.capitalize())
		Chennai

Chapter 8.indd 127 10-02-2020 20:01:05

தொடரியல்	விளக்கம்	எடுத்துக்காட்டு

center(width, fillchar)	அறிவிக்கப்பட்டுள்ள சரத்தின் மொத்த நீளத்தில் கொடுக்கப்பட்ட சரத்தினை மையத்தில் வெளிப்படும் மேலும் குறியுருக்களை பெற்றிக்காத நெடுவரிசைகள் fillchar குறியுருவை கொண்டு நிரப்பப்படும்.	>>> str1="Welcome"
		>>> print(str1.center(15,'*'))
		****Welcome****

find(sub[, start[, end]])	இச்செயற்கூறானது கொடுக்கப்பட்டுள்ள சரத்தில் குறிப்பிட்ட துணைச்சரத்தினை தேட பயன்படுகிறது. துணைச் சரம் முதலில் தோன்றும் இடத்தின் சுட்டெண்ணை திரும்ப தரும். தேடப்படும் துணைச் சரம் இல்லையெனில் -1 என்ற மதிப்பை திரும்ப தரும்.	>>>str1='mammals' >>>str1.find('ma') ***0*** ***start*** அளபுரு தவிர்க்கப்பட்டால் சரத்தின் தொடக்கத்திலிருந்து செயற்கூறு தேடலை தொடங்கும். >>>str1.find('ma',2) 3 >>>str1.find('ma',2,4) -1 கொடுக்கப்பட்ட சரத்தில் சுட்டெண் ***2*** முதல் ***4-1*** வரை தேடப்படும் துணை நிரல் கண்டுப்பிடிக்கப்படவில்லை எனவே, ***-1*** என வெளியீடு தோன்றும். >>>str1.find('ma',2,5) 3
isalnum( )	ஒரு சரம் எழுத்துகள் மற்றும் எண்களை மட்டும் கொண்டிருந்தால் மெய் என்ற மதிப்பை திருப்பும். சரம் சிறப்புக் குறியீடுகளான_, @, #, *, போன்றவற்றை கொண்டிருந்தால் பொய் என்ற மதிப்பை திருப்பும்.	>>>str1='Save Earth' >>>str1.isalnum() ***False*** கொடுக்கப்பட்ட சரத்தில் இடைவெளி ஒரு சிறப்புக் குறியுரு ஆகும். எனவே இந்த செயற்கூறு பொய் என திருப்பும். >>>'Save1Earth'.isalnum() ***True***
isalpha( )	ஒரு சரம் எழுத்துக்களை மட்டும் கொண்டிருந்தால் மெய் என திருப்பும். இல்லாவிடில் பொய் என திருப்பும்.	>>>'Click123'.isalpha() ***False*** >>>'python'.isalpha( ) ***True***
isdigit( )	ஒரு சரம் எண்களை மட்டும் கொண்டிருந்தால் மெய் என திருப்பும். இல்லாவிடில் பொய் என திருப்பும்.	>>> str1='Save Earth' >>>print(str1.isdigit( )) ***False***
lower( )	கொடுக்கப்பட்ட சரத்தின் அனைத்து எழுத்துக்களையும் சிறிய எழுத்துக்களாக திருப்பும்.	>>>str1='SAVE EARTH' >>>print(str1.lower()) ***save earth***

தொடரியல்	விளக்கம்	எடுத்துக்காட்டு
islower( )	கொடுக்கப்பட்ட சரம் சிறிய எழுத்துக்களில் இருந்தால் மெய் என திருப்பும்.	>>> str1='welcome' >>>print (str1.islower( )) *True*
isupper( )	கொடுக்கப்பட்ட சரம் பெரிய எழுத்துக்களில் இருந்தால் மெய் என திருப்பும்.	>>> str1='welcome' >>>print (str1.isupper( )) *False*
upper( )	கொடுக்கப்பட்ட சரத்தின் அனைத்து எழுத்துக்களையும் பெரிய எழுத்துக்களாக திருப்பும்.	>>> str1='welcome' >>>print (str.upper( )) *WELCOME*

title( )	கொடுக்கப்பட்ட சரத்தின் வார்த்தைகளின் முதல் எழுத்தை மட்டும் பெரிய எழுத்துக்களாக திருப்பும்.	>>> str1='education department'
		>>> print(str1.title())
		Education Department

swapcase( )	சரத்தில் உள்ள ஒரு எழுத்து பெரிய எழுத்தாக இருந்தால் அது சிறிய எழுத்தாகவும் நேர் மாறாகவும் திருப்பும்.	>>> str1="tAmiL NaDu"
		>>> print(str1.swapcase())
		TaMIl nAdU

count(str, beg, end)	ஒரு சரத்தில் கொடுக்கப்பட்ட பரப்பிற்குள் உள்ள துணை சரங்களின் (substring) எண்ணிக்கையை திருப்பும். ஒரு துணை சரம் என்பது ஒற்றை குறியுருவாக கூட இருக்கலாம். பரப்பு அளபுருக்களை கொடுக்க வேண்டியது கட்டாயமில்லை கொடுக்காவிட்டால் பைத்தான் முழு சரத்தையும் தேடும். தேடல் எழுத்துணர்வு மிக்கது (case sensitive) என்பதை நினைவில் கொள்க	>>> str1="Raja Raja Chozhan"
		>>> print(str1.count('Raja'))
		2
		>>> print(str1.count('r'))
		0
		>>> prin***t(str1.count('R'))***
		2
		>>> print(str1.count('a'))
		5
		>>> print(str1.count('a',0,5))
		2
		>>> print(str1.count('a',11))
		1

ord(char)	கொடுக்கப்பட்ட குறியுருவின் ASCII மதிப்பை திருப்பும்	>>> ch = 'A'
		>>> print(ord(ch))
		65
		>>> print(***ord('B'))***
		66

chr(ASCII)	கொடுக்கப்பட்ட ASCII மதிப்புக்கு நிகரான குறியுருவை திருப்பும்	>>> ch=97
		>>> print(chr(ch))
		a
		>>> print(chr(87))
		W

உறுப்பு செயற்குறிகள்

கொடுக்கப்பட்டுள்ள சரம் மற்றொரு சரத்தில் இடம்பெற்றுள்ளதா என்பதை கண்டறிய in மற்றும் not in செயற்குறிகள் பயன்படுகின்றன. எனவே இச்செயற்குறிகள் உறுப்பு செயற்குறிகள் என அழைக்கப்படுகின்றன.

எடுத்துக்காட்டு

```
str1=input ("Enter a string: ")
str2="chennai"
if str2 in str1:
        print ("Found")
else:
        print ("Not Found")
```

வெளியீடு : 1

```
Enter a string: Chennai G HSS, Saidapet
Found
```

வெளியீடு : 2

```
Enter a string: Govt G HSS, Ashok Nagar
Not Found
```

சரங்களை பயன்படுத்தி நிரல்

எடுத்துக்காட்டு 8.4: கொடுக்கப்பட்ட சரம் பாலின்ட்ரோமா இல்லையா என்பதை சோதிக்கும் நிரல்

```
str1 = input ("Enter a string: ")
str2 = ' '
index=-1
for i in str1:
        str2 += str1[index]
        index -= 1
print ("The given string = { } \n The Reversed string = { }".format(str1, str2))
if (str1==str2):
        print ("Hence, the given string is Palindrome")
else:
        print ("Hence, the given is not a palindrome")
```

வெளியீடு : 1

```
Enter a string: malayalam
The given string = malayalam
The Reversed string = malayalam
Hence, the given string is Palindrome
```

வெளியீடு : 2

```
Enter a string: welcome
The given string = welcome
The Reversed string = emoclew
Hence, the given string is not a palindrome
```

எடுத்துக்காட்டு 8.5: பின்வரும் வடிவத்தை அச்சிடும் நிரல்

```
*
* *
* * *
* * * *
* * * * *
str1=' * '
i=1
while i<=5:
        print (str1*i)
        i+=1
```

வெளியீடு

```
*
* *
* * *
* * * *
* * * * *
```

எடுத்துக்காட்டு 8.6: கொடுக்கப்பட சரத்தில் உள்ள உயிர் எழுத்துக்கள் மற்றும் மெய் எழுத்துக்களின் எண்ணிக்கையை கண்டறியும் நிரல்

```
str1=input ("Enter a string: ")
str2="aAeEiIoOuU"
v,c=0,0
for i in str1:
        if i in str2:
                v+=1
        else:
                c+=1
print ("The given string contains { } vowels and { } consonants".format(v,c))
```

வெளியீடு

```
Enter a string: Tamilnadu School Education
The given string contains 11 vowels and 15 consonants
```

எடுத்துக்காட்டு 8.7: ஏபோரியன் தொடரை உருவாக்கும் நிரல் (ஏபோரியன் தொடர் அகர வரிசைப்படி பட்டியலை உருவாக்கும்)

```
str1="ABCDEFGH"
str2="ate"
for i in str1:
        print ((i+str2),end='\t')
```

வெளியீடு

```
Aate    Bate    Cate    Date    Eate    Fate    Gate    Hate
```

எடுத்துக்காட்டு 8.8: பயனரிடமிருந்து பெறப்படும் சரத்தில் உள்ள உயிர் எழுத்துக்களை நீக்கிவிட்டு அதே சரத்தை வெளிப்படுத்தும் நிரல்

```
def rem_vowels(s):
        temp_str=''
        for i in s:
                if i in "aAeEiIoOuU":
                        pass
                else:
                        temp_str+=i
        print ("The string without vowels: ", temp_str)
str1= input ("Enter a String: ")
rem_vowels (str1)
```

வெளியீடு

```
Enter a String: Mathematical fundations of Computer Science
The string without vowels:  Mthmtcl fndtns f Cmptr Scnc
```

எடுத்துக்காட்டு 8.9: கொடுக்கப்பட்ட சரத்தில் ஒரு குறிப்பிட்ட குறியுரு எத்தனை முறை இடம் பெறுகிறது என்பதை கணக்கிடும் நிரல்.

```
def count(s, c):
        c1=0
        for i in s:
                if i == c:
                        c1+=1
        return c1
str1=input ("Enter a String: ")
ch=input ("Enter a character to be searched: ")
cnt=count (str1, ch)
print ("The given character {} is occurs {} times in the given string".format(ch,cnt))
```

வெளியீடு

```
Enter a String: Software Engineering
Enter a character to be searched: e
The given character e is occurs 3 times in the given string
```

நினைவில் கொள்க

- சரம் என்பது பைத்தானில் உள்ள ஒரு தரவு வகையாகும்.
- சர தரவுவகை மாறா தன்மையற்றது. அதாவது சரத்தை ஒரு முறை வரையறுத்த பின்பு இயக்க நேரத்தில் அதை நம்மால் மாற்ற இயலாது.
- சரங்களை மூன்று மேற்கோள் குறிகளுக்குள் வரையறுக்கலாம். மேலும் இது பல வரி சரங்களை உருவாக்க உதவுகிறது.
- சரத்தில் உள்ள ஒவ்வொரு குறியுருவிற்கு பைத்தான் சுட்டு மதிப்புகளை ஒதுக்கீடு செய்கிறது.
- கீழ் ஒட்டானது நேர்மறை அல்லது எதிர்மறை முழு எண்ணாக இருக்கலாம்
- மூலச்சரத்தில் உள்ள ஒரு துணைச்சரம் (சரத்தின் ஒரு பகுதி) துண்டு (Slice) எனப்படும்.

Chapter 8.indd 132 10-02-2020 20:01:05

நினைவில் கொள்க

- சரத்தினை பிரிக்கும் / துண்டாக்கும் செயல்பாட்டில் மூன்றாவது அளபுருவையும் குறிப்பிட முடியும்.
- விடுபடு வரிசை பின் சாய்வுக் கோட்டை கொண்டு துவங்கும்.
- format() செயற்கூறானது சரங்களை வடிவமைக்கப் பயன்படும் முக்கிய செயற்கூறாகும்.
- கொடுக்கப்பட்டுள்ள சரம் மற்றொரு சரத்தில் இடம்பெற்றுள்ளதா என்பதை கண்டறிய in மற்றும் not in செயற்குறிகள் பயன்படுகின்றன

செய்முறைப் பயிற்சி

1. சரத்தின் நீளத்தை கண்டறிய பைத்தான் நிரல் எழுதுக.
2. கொடுக்கப்பட்டுள்ள சரத்தில் ஒவ்வொரு வார்த்தையும் எத்தனை முறை தோன்றுகிறது என்பதை கண்டறிய நிரல் எழுதுக.
3. கொடுக்கப்பட்டுள்ள சரத்தின் ஒவ்வொரு வரியிலும் முன்னொட்டாக உரையை சேர்க்க நிரல் எழுதுக.
4. குறிப்பிட்ட அகலத்தின் வலது புறத்தில் '*' குறியீட்டுடன் முழு எண்ணை அச்சிட நிரல் எழுதுக.
5. கொடுக்கப்பட்ட சரத்தினை தலைகீழாக எழுத நிரல் எழுதுக. (எ.கா) "Wel"= "lew"
6. கொடுக்கப்பட்ட சரத்தில் ஒரு குறிப்பிட்ட குறியுரு தோன்றும் அனைத்து இடங்களிலும் அதை நீக்க நிரல் எழுதுக.
7. ஒரு சரத்தினை இன்னொரு சரத்துடன் சேர்க்க "+ =" செயற்குறியை பயன்படுத்தாமல் நிரல் எழுதுக.
8. இரண்டு சரங்களை இடமாற்றம் செய்ய நிரல் எழுதுக.
9. ஒரு சரத்தை மற்றொரு சரத்தை கொண்டு மாற்றிட replace() செயற்கூறை பயன்படுத்தாமல் நிரல் எழுதுக.
10. கொடுக்கப்பட்டுள்ள சரத்தில் குறியுருக்களின் எண்ணிக்கை, வார்த்தைகளின் எண்ணிக்கை மற்றும் வரிகளின் எண்ணிக்கையை கணக்கிட நிரல் எழுதுக.

மதிப்பீடு

L7E7LC

பகுதி–அ

சரியான விடையை தேர்ந்தெடுத்து எழுதுக **(1 மதிப்பெண்)**

1. பின்வருவனவற்றுள் எது கீழ்க்கண்ட பைத்தான் நிரலுக்கான வெளியீடாகும்?

```
str1="TamilNadu"
print(str1[::-1])
```

(அ) Tamilnadu (ஆ) Tmlau

(இ) udanlimaT (ஈ) udaNlimaT

Chapter 8.indd 133 10-02-2020 20:01:05

2. பின்வரும் குறியுருக்கான வெளியீடு யாது?

str1 = "Chennai Schools"

str1[7] = "-"

(அ) Chennai-Schools (ஆ) Chenna-School

(இ) Type error (ஈ) Chennai

3. பின்வருவனவற்றுள் எது சரங்களை இணைக்க பயன்படும் செயற்குறியாகும்?

(அ) + (ஆ) & (இ) * (ஈ) =

4. மூன்று மேற்கோள் குறிகளுக்குள் தரப்படும் சரமானது பின்வருபவனவற்றுள் எதை உருவாக்க அனுமதிக்கும்:

(அ) ஒரு வரி சரம் (ஆ) பல வரி சரங்கள்

(இ) இரு வரி சரம் (ஈ) ஒன்றுக்கு மேற்பட்ட சரங்கள்

5. பைத்தானில் சரங்களானது:

(அ) மாற்றக்கூடியது (ஆ) மாறாத்தன்மையுடையது

(இ) பரஸ்பரதன்மையற்றது (ஈ) நெகிழ்வானது

6. பின்வருவனவற்றுள் எது சரத்தினை துண்டாக்கும் (Slicing) செயற்குறியாகும்?

(அ) { } (ஆ) [ ] (இ) < > (ஈ) ( )

7. stride என்பது பின்வருபவனவற்றுள் எதை குறிக்கும்?

(அ) slide செயல்பாட்டின் கீழ்ஒட்டு மதிப்பாகும்.

(ஆ) slice செயற்பாட்டின் முதல் அளபுருவாகும்

(இ) slice செயற்பாட்டின் இரண்டாவது அளபுருவாகும்

(ஈ) slice செயற்பாட்டின் மூன்றாவது அளபுருவாகும்.

8. பின்வரும் வடிவமைப்பு குறியுருக்களுள் அடுக்கு குறியீட்டில் அச்சிட உதவும் மேல் எழுத்து எது?

(அ) %e (ஆ) %E (இ) %g (ஈ) %n

9. பின்வருபவனவற்றுள் எந்தக் குறியீடு format( ) செயற்கூறுடன் பயன்படும் பதிலீடு குறியீடாகும்?

(அ) { } (ஆ) < > (இ) ++ (ஈ) ^^

10. சரத்தின் கீழ்ஒட்டானானது:

(அ) நேர்மறை எண்கள் (ஆ) எதிர்மறை எண்கள்

(இ) (அ) மற்றும் (ஆ) (ஈ) (அ) அல்லது (ஆ)

பகுதி–ஆ

அனைத்து வினாக்களுக்கும் விடையளி **(2 மதிப்பெண்)**

1. சரம் என்றால் என்ன?
2. பைத்தானில் சரங்களை மாற்றம் செய்ய முடியுமா?
3. பைத்தானில் சரத்தை எவ்வாறு நீக்குவாய்?

4. பின்வரும் பைத்தான் குறிமுறையின் வெளியீடு யாது?

```
str1 = "School"
print(str1*3)
```

5. சரத்தை துண்டாக்குதல்/பிரித்தல் என்றால் என்ன?

பகுதி–இ

அனைத்து வினாக்களுக்கும் விடையளி **(3 மதிப்பெண்)**

1. கொடுக்கப்பட்ட வடிவத்தை அச்சிடும் பைத்தான் நிரலை எழுதுக?

```
C O M P U T E R
C O M P U T E
C O M P U T
C O M P U
C O M P
C O M
C O
C
```

2. பின்வருபவனவற்றை பற்றி தகுந்த எடுத்துக்காட்டுடன் குறிப்பு வரைக

   (அ) capitalize( ) (ஆ) swapcase( )

3. கீழே கொடுக்கப்பட்டுள்ள பைத்தான் நிரலின் வெளியீடு யாது?

```
str1 = "welcome"
str2 = "to school"
str3=str1[:2]+str2[len(str2)-2:]
print(str3)
```

4. format( ) செயற்கூறின் பயன் யாது? எடுத்துக்காட்டு தருக?
5. பைத்தானில் count( ) செயற்கூறு பற்றி குறிப்பு வரைக.

பகுதி ஈ

அனைத்து வினாக்களுக்கும் விடையளி **(5 மதிப்பெண்)**

1. பைத்தானில் பயன்படும் சர செயற்குறிகளை தகுந்த எடுத்துக்காட்டுடன் விளக்குக

Reference Books

1. *https://docs.python.org/3/tutorial/index.html*
2. *https://www.techbeamers.com/python-tutorial-step-by-step/#tutorial-list*
3. *Python programming using problem solving approach – Reema Thareja – Oxford University press.*
4. *Python Crash Course – Eric Matthes – No starch press, San Francisco.*

அலகு III

பாடம் 9

(List, Tuples, Set மற்றும் Dictionary) தொகுப்பு தரவினங்கள்

ETJX83

கற்றலின் நோக்கங்கள்

இந்த பாடத்தை கற்ற பிறகு மாணவர்கள் அறிந்து கொள்வது,

- List, Tuples, Set மற்றும் Dictionary போன்ற பல வகையான தொகுப்பு தரவினங்களின் அடிப்படை கருத்துருக்களை புரிந்து கொள்ளுதல்.
- பல்வேறு செயற்கூறுகளைப் பயன்படுத்தி List, Tuples, Set மற்றும் Dictionary - ல் வேலை செய்தல்.
- List, Tuples, Set மற்றும் Dictionary யைப் பயன்படுத்தி பைத்தான் நிரல்களை எழுதுதல்.
- List, Tuples, Set மற்றும் Dictionary - களுக்கு இடையேயான உறவுகளை புரிந்து கொள்ளுதல்.

9.1 List ஓர் அறிமுகம்

பைத்தான் நிரலாக்க மொழியில் நான்கு வகையான தொகுப்பு தரவினங்கள் உள்ளன. அவை List, Tuples, Set மற்றும் Dictionary ஆகும். பைத்தானில் உள்ள List சரத்தைப் போன்றே "வரிசைமுறை தரவினம்" ஆகும். இது சதுர அடைப்புக் குறிக்குள்[ ] அடைக்கப்பட்ட மதிப்புகளின் வரிசைப்படுத்தப்பட்ட தொகுப்பாகும். List-ல் உள்ள ஒவ்வொரு மதிப்பும் உறுப்பு (element) என்றழைக்கப்படுகிறது. இது எண்கள், எழுத்துகள், சரநிலையுருக்கள் மற்றும் பின்னலான List போன்ற எந்த வகையாகவும் இருக்கலாம். உறுப்புகள் மாறும் தன்மையுடையன. அதாவது உறுப்புகளை மாற்றியமைக்கவும், சேர்க்கவும் அல்லது நீக்கவும் முடியும். ஒவ்வொரு உறுப்பும் List-ல் ஒரு குறிப்பிட்ட இடத்தில் அமைந்திருக்கும். ஒரு உறுப்பின் இருப்பிட குறியீட்டெண் அல்லது சுட்டெண் சுழியம் என்ற தொடக்க எண்ணால் குறிப்பிடப்படுகிறது. இந்த சுட்டெண் குறிப்பிட்ட உறுப்பை கண்டறிவதற்கும், அதனை கையாளுவதற்கும் பயன்படுகிறது. இதனால் List என்பது பதினொன்றாம் வகுப்பில் நீங்கள் பயின்ற அணிகளை ஒத்தது என்பதை நினைவில் கொள்க.

9.1.1 பைத்தானில் List–ஐ உருவாக்குதல்:

பைத்தானில், List சதுர அடைப்புக்குறியைப் பயன்படுத்தி எளிமையாக உருவாக்கப்படுகின்றன. List ன் உறுப்புகளை கட்டயமாக சதுர அடைப்புக்குறிக்குள் குறிப்பிட வேண்டும். பின்வரும் தொடரியல் List உருவாக்கும் விதத்தை விளக்குகிறது.

தொடரியல்:

Variable = [element-1, element-2, element-3 …… element-n]

Chapter 9.indd 136 30-01-2020 19:37:06

எடுத்துக்காட்டு:

Marks = [10, 23, 41, 75]
Fruits = ["Apple", "Orange", "Mango", "Banana"]
MyList = [ ]

மேலே கொடுக்கப்பட்டுள்ள எடுத்துக்காட்டில், Marks என்ற List நான்கு முழு எண் (integer) உறுப்புகளையும், இரண்டாவது List, Fruits-ல் நான்கு சரநிலையுரு உறுப்புகளையும் கொண்டுள்ளது. மூன்றாவது List ஒரு வெற்று List ஆகும். List-ன் உறுப்புகள் ஒரே தரவினமாக இருக்க வேண்டிய அவசியமில்லை. பின்வரும் List பல வகை உறுப்புகளைக் கொண்டுள்ளது.

MyList = ["Welcome", 3.14, 10, [2, 4, 6]]

மேலே உள்ள எடுத்துக்காட்டில் , **My**List என்ற List மற்றொரு List-டை உறுப்பாகக் கொண்டுள்ளது. இந்த வகை List "பின்னலான List (**Nested** List)" என்று அழைக்கப்படுகிறது.

பின்னலான List என்பது மற்றொரு List-ஐ ஒரு உறுப்பாக கொண்டுள்ள List ஆகும்.

9.1.2 List உறுப்புகளை அணுகுதல்

பைத்தான், List-ன் ஒவ்வொரு உறுப்புக்கும் சுழியத்திலிருந்து தொடங்குகின்ற தானமைவு சுட்டெண் மதிப்பை இருத்துகிறது. சுட்டெண் மதிப்பு, பட்டியலின் ஒரு உறுப்பை அணுகுவதற்கு பயன்படுகிறது. பைத்தானில், சுட்டெண் மதிப்பு நேர்மறை அல்லது எதிர்மறை முழு எண் மதிப்பாக இருக்கலாம்.

எடுத்துக்காட்டு:

Marks = [10, 23, 41, 75]

Marks	10	23	41	75
நேர்மறை சுட்டெண் (Positive)	0	1	2	3
எதிர்மறை சுட்டெண் (Negative)	-4	-3	-2	-1

நேர்மறை சுட்டெண் மதிப்பு List-ன் தொடக்கத்திலிருந்தும் எதிர்மறை சுட்டெண் மதிப்பு இறுதியிலிருந்து பின் வரிசையில் (அதாவது பின்னோக்கு வரிசை) உறுப்புகளை கணக்கிடுகிறது.

List-லிருந்து ஒரு உறுப்பை அணுகுவதற்கு, சதுர அடைப்புக் குறிக்குள் கொடுக்கப்பட்ட உறுப்பின் சுட்டெண்ணை பின்னொட்டாகக் கொண்ட List-ன் பெயரை எழுதவும்.

தொடரியல்

List_Variable = [E1, E2, E3 …… En]

print (List_Variable[index of a element])

Chapter 9.indd 137 30-01-2020 19:37:06

எடுத்துக்காட்டு (ஒற்றை உறுப்பை அணுகுதல்):

```
>>> Marks = [10, 23, 41, 75]
>>> print (Marks[0])
10
```

மேலே உள்ள எடுத்துக்காட்டில், சுழியம் என்ற சுட்டெண்ணின் மதிப்பு 10 ஆக உள்ளதால், print கட்டளை 10 என்ற வெளியீட்டை அச்சிடுகிறது.

எடுத்துக்காட்டு (உறுப்புகளை பின்னோக்கு வரிசையில் அணுகுதல்):

```
>>> Marks = [10, 23, 41, 75]
>>> print (Marks[-1])
75
```

குறிப்பு

எதிர்மறை சுட்டெண், உறுப்புகளை பின்னோக்கு வரிசையில் அணுகுவதற்கு பயன்படுகிறது.

9.1.2.1 List ன் அனைத்து உறுப்புகளையும் அணுகுதல்

List-ன் அனைத்து உறுப்புகளையும் அணுகுவதற்கு மடக்குகள் பயன்படுகின்றன. மடக்கின் தொடக்க மதிப்பு சுழியமாக இருக்க வேண்டும். List-ன் தொடக்க சுட்டெண் மதிப்பு சுழியமாகும்.

எடுத்துக்காட்டு

```
Marks = [10, 23, 41, 75]
i = 0
while i < 4:
        print (Marks[i])
        i = i + 1
```

வெளியீடு

```
10
23
41
75
```

மேலே உள்ள எடுத்துக்காட்டில் marks என்ற List 10,23,41,75 என்ற நான்கு முழு எண் உறுப்புகளைக் கொண்டுள்ளது. ஒவ்வொரு உறுப்புகளும், சுழியத்திலிருந்து தொடங்குகின்ற சுட்டெண் மதிப்பைப் பெற்றுள்ளது. உறுப்புகளின் சுட்டெண் மதிப்பு முறையே 0,1,2,3 என்பதாகும். இங்கு while மடக்கு List-ன் அனைத்து உறுப்புகளையும் அச்சிடுவதற்கு பயன்படுகிறது. மடக்கின் தொடக்க மதிப்பு சுழியமாகவும், நிபந்தனை சோதிப்பு i < 4 ஆகவும் உள்ளது. நிபந்தனை சோதிப்பு சரி எனில், மடக்கு செயல்படுத்தப்பட்டு தொடர்புடைய வெளியீடு அச்சிடப்படும்.

Chapter 9.indd 138 30-01-2020 19:37:06

முதல் சுழற்சியின் போது, i ன் மதிப்பு சுழியமாகும். கொடுக்கப்பட்ட நிபந்தனை மதிப்பு மெய், எனவே தொடர்ந்து வரும் **print (Marks[i])** செயல்படுத்தப்பட்டு, Marks[0] என்ற உறுப்பின் 10 என்ற மதிப்பு அச்சிடப்படும்.

அடுத்த கூற்று **i = i + 1, i** ன் மதிப்பை சுழியத்திலிருந்து, 1-என மிகுக்கிறது. இப்போது, பாய்வுக் கட்டுப்பாடு, நிபந்தனை சோதிப்பை சரிபார்க்க While கூற்றுக்கு மாற்றப்படுகிறது. கொடுக்கப்பட்ட, while மடக்கின் நிபந்தனை சோதிப்பு பொய் என வரும் வரை **Marks** லிஸ்ட்ன் அனைத்து உறுப்புகளையும் அச்சிடும் செயல் மீண்டும் மீண்டும் செயல்படுத்தப்படும். மீதமுள்ள உறுப்புகளை அச்சிட மீண்டும் செய்யப்படுகிறது.

பின்வரும் அட்டவணை, மடக்கு செயல்படும் விதம் மற்றும் அச்சிடப்பட வேண்டிய மதிப்பைக் காண்பிக்கிறது.

சுழற்சி	i	while i < 4	print (Marks[i])	i = i + 1
1	0	0 < 4 True	Marks [0] = 10	0 + 1 = 1
2	1	1 < 4 True	Marks [1] = 23	1 + 1 = 2
3	2	2 < 4 True	Marks [2] = 41	2 + 1 = 3
4	3	3 < 4 True	Marks [3] = 75	3 + 1 = 4
5	4	4 < 4 False	--	--

9.1.2.2 பின்னோக்கு சுட்டு (Reverse Indexing)

பைத்தான், List உறுப்புகளுக்கு பின்னோக்கு அல்லது எதிர்மறை, சுட்டெண்களை வழங்குகிறது. இதனால் பைத்தான், சுட்டெண்களை எதிர் வரிசையில் பட்டியலிடுகிறது. பைத்தான், List–ன் கடைசி உறுப்பிற்கு -1 முந்தைய உறுப்பிற்கு -2 என்ற சுட்டெண் மதிப்புகளையும் இருத்துகிறது. இதுவே பின்னோக்கு சுட்டு என அழைக்கப்படுகிறது.

எடுத்துக்காட்டு

```
Marks = [10, 23, 41, 75]
i = -1
while i >= -4:
        print (Marks[i])
        i = i + -1
```

வெளியீடு

```
75
41
23
10
```


Chapter 9.indd 139 30-01-2020 19:37:06

பின்வரும் அட்டவணை மேலே உள்ள பைத்தான் குறிமுறை வேலை செய்யும் விதத்தை காண்பிக்கிறது.

சுழற்சி	i	while i >= -4	print (Marks[i])	i = i + -1
1	-1	-1 >= -4 True	Marks[-1] = 75	-1 + (-1) = -2
2	-2	-2 >= -4 True	Marks[-2] = 41	-2 + (-1) = -3
3	-3	-3 >= -4 True	Marks[-3] = 23	-3 + (-1) = -4
4	-4	-4 >= -4 True	Marks[-4] = 10	-4 + (-1) = -5
5	-5	-5 >= -4 False	--	--

9.1.3 List –ன் நீளம்

பைத்தானில் உள்ள len( ) செயற்கூறு, List-ன் நீளத்தை கண்டறிய பயன்படுகிறது. (அதாவது, List-ல் உள்ள உறுப்புகளின் எண்ணிக்கை). பொதுவாக, len( ) செயற்கூறு List-ல் அனைத்து உறுப்புகளையும் அணுகுவதற்கான மடக்கின் உச்ச வரம்பை நிர்ணயிக்க பயன்படுகிறது. ஒரு List-ல் மற்றொரு List-ஐ ஒரு உறுப்பாக கொண்டிருந்தால், len( ) செயற்கூறு உட்புற List-ஐ ஒரு உறுப்பாகவே எடுத்துக்கொள்ளும்.

எடுத்துக்காட்டு :ஒற்றை உறுப்பை அணுகுதல்

```
>>> MySubject = ["Tamil", "English", "Comp. Science", "Maths"]
>>> len(MySubject)
    4
```

எடுத்துக்காட்டு : மடக்கை பயன்படுத்தி ஒரு List–ல் உள்ள உறுப்புகளை அச்சிடும் நிரல்

```
MySubject = ["Tamil", "English", "Comp. Science", "Maths"]
i = 0
while i < len(MySubject):
        print (MySubject[i])
        i = i + 1
```

வெளியீடு

```
Tamil
English
Comp. Science
Maths
```

9.1.4 for மடக்கை பயன்படுத்தி உறுப்புகளை அணுகுதல்

பைத்தானில் உள்ள for மடக்கு, List-ல் உள்ள அனைத்து உறுப்புகளையும் ஒவ்வொன்றாக அணுகுவதற்கு பயன்படுகிறது. இது, C++ போன்ற பிற நிரலாக்க மொழிகளில் உள்ள for மடக்கை போன்றது.

தொடரியல்:

for *index_var* ***in*** List:
    print *(index_var)*

Chapter 9.indd 140 30-01-2020 19:37:06

இங்கு, index_var என்பது List-ல் உள்ள ஒவ்வொரு உறுப்பின் சுட்டு மதிப்பைக் குறிக்கிறது. பைத்தான் ஆங்கில மொழியில் படிப்பது போன்றே, “for” கூற்றை கையாள்கிறது: “For (every) element in (the List of) List and print (the name of the) List items”

எடுத்துக்காட்டு

```
Marks=[23, 45, 67, 78, 98]
for x in Marks:
        print( x )
```

வெளியீடு

```
23
45
67
78
98
```

மேலே உள்ள எடுத்துக்காட்டில், Marks List ஐந்து உறுப்புகளைக் கொண்டுள்ளது; ஒவ்வொரு உறுப்பிற்கும் 0 முதல் 4 வரை சுட்டெண் இருத்தப்படுகிறது. பைத்தான், ஆங்கில மொழியில் வாசிப்பதை போன்றே for மடக்கு மற்றும் **print** கூற்றுகளை படிக்கிறது: “For (every) element (represented as x) in (the List of) Marks and print (the values of the) elements”.

9.1.5 List உறுப்புகளை மாற்றம் செய்தல்

பைத்தானில், List-கள் மாறும் தன்மையுடையவை, அதாவது List உறுப்புகள் மாற்றப்படலாம். List உறுப்பு அல்லது தொடர் உறுப்புகளை எளிய மதிப்பிருத்து செயற்குறியை (=) பயன்படுத்தி மாற்றலாம்.

தொடரியல்:

List_Variable [index of an element] = மாற்றப்படும் மதிப்பு

List_Variable [index from : index to] = மாற்றப்படும் மதிப்பு

இங்கு, index from என்பது தொடரின் தொடக்க சுட்டெண் ஆகும். index to என்பது தொடரின் உச்ச வரம்பாகும். கொடுக்கப்படும் உச்ச மதிப்பு தொடரின் உச்ச மதிப்பாக சேர்க்கப்பட்டாது. எடுத்துக்காட்டாக, தொடர் [0:5] என்று நீங்கள் பொருத்தினால், பைத்தான் 0 முதல் 4 என்ற உறுப்பு சுட்டெண்களை மட்டுமே எடுத்துக் கொள்ளும். எனவே, நீங்கள் 1 முதல் 4 தொடர் உறுப்புகளை மாற்றம் செய்ய விரும்பினால், கட்டாயமாக, [1:5] என்று குறிப்பிட வேண்டும்.

Chapter 9.indd 141 30-01-2020 19:37:06

எடுத்துக்காட்டு 9.1: ஒற்றை மதிப்பை மாற்றுவதற்கான பைத்தான் நிரல்

```
MyList = [2, 4, 5, 8, 10]
print ("MyList elements before update... ")
for x in MyList:
    print (x)
MyList[2] = 6
print ("MyList elements after updation... ")
for y in MyList:
    print (y)
```

வெளியீடு

```
MyList elements before update...
        2
        4
        5
        8
        10
MyList elements after updation...
        2
        4
        6
        8
        10
```

எடுத்துக்காட்டு 9.2: தொடர் மதிப்புகளை மாற்றுவதற்கான பைத்தான் நிரல்

```
MyList = [1, 3, 5, 7, 9]
print ("List Odd numbers... ")
for x in MyList
    print (x)
MyList [0:5] = 2,4,6,8,10
print ("List Even numbers... ")
for y in MyList:
    print (y)
```

வெளியீடு

```
List Odd numbers...
        1
        3
        5
        7
        9
List Even numbers...
        2
        4
        6
        8
        10
```


Chapter 9.indd 142 30-01-2020 19:37:06

9.1.6 List–ல் உறுப்பகளை சேர்த்தல்

பைத்தானில், **append** ( ) செயற்கூறு ஒரு உறுப்பையும் **extend**( ) செயற்கூறு ஒன்றுக்கும் மேற்பட்ட உறுப்புகளையும் ஏற்கனவே உள்ள List-ல் சேர்க்க பயன்படுகின்றது.

தொடரியல்:

List.*append (element to be added)*
List..*extend ([elements to be added])*

extend( ) செயற்கூறில், பல உறுப்புகளை சதுர அடைப்புக் குறிக்குள் செயற்கூறின் செயலுருபுகளைப் போலவே குறிப்பிட வேண்டும்.

எடுத்துக்காட்டு

```
>>> MyList=[34, 45, 48]
>>> MyList.append(90)
>>> print(MyList)
        [34, 45, 48, 90]
```

மேலே உள்ள எடுத்துக்காட்டில், My List என்பது மூன்று உறுப்புகளுடன் உருவாக்கப்பட்டுள்ளது. >>> MyList.append(90) என்ற கூற்றின் மூலம் 90 என்ற கூடுதல் மதிப்பு ஏற்கனவே உள்ள List-ல் கடைசி உறுப்பாக சேர்க்கப்படுகிறது. Print கூற்று MyList என்ற லிஸ்ட்-ல் உள்ள அனைத்து உறுப்புகளையும் காண்பிக்கிறது.

எடுத்துக்காட்டு

```
>>> MyList.extend ([71, 32, 29])
>>> print(MyList)
        [34, 45, 48, 90, 71, 32, 29]
```

மேலே உள்ள குறிமுறையில், extend( ) செயற்கூறு பல உறுப்புகளை சேர்ப்பதற்கு பயன்படுகிறது. Print கூற்று, கூடுதல் உறுப்புகள் சேர்க்கப்பட்ட பிறகு List-ல் உள்ள அனைத்து உறுப்புகளையும் காண்பிக்கிறது.

9.1.7 List உறுப்புகளை செருகுதல்

நீங்கள் ஏற்கனவே படித்தபடி, பைத்தானில் உள்ள append( ) ஒரு உறுப்பை சேர்க்கப் பயன்படுகிறது. ஆனால், இது List-ன் இறுதியில் உறுப்பை சேர்க்கிறது. நீங்கள் உங்களுக்கு விருப்பமான இடத்தில் ஒரு உறுப்பை சேர்க்க விரும்பினால், insert ( ) செயற்கூறை பயன்படுத்த வேண்டும். insert ( ) செயற்கூறு, List-ன் எந்தவொரு இடத்திலும் ஒரு உறுப்பை சேர்க்கப் பயன்படுகிறது.

தொடரியல்:

List.*insert (position index, element)*

எடுத்துக்காட்டு

```
>>> MyList=[34,98,47,'Kannan', 'Gowrisankar', 'Lenin', 'Sreenivasan' ]
>>> print(MyList)
        [34, 98, 47, 'Kannan', 'Gowrisankar', 'Lenin', 'Sreenivasan']
>>> MyList.insert(3, 'Ramakrishnan')
>>> print(MyList)
[34, 98, 47, 'Ramakrishnan', 'Kannan', 'Gowrisankar', 'Lenin', 'Sreenivasan']
```

மேலே உள்ள எடுத்துக்காட்டில், insert( ) செயற்கூறு, 'Ramakrishnan' என்ற புதிய உறுப்பை மூன்றாவது சுட்டிடெண்ணில் அதாவது நான்காவது இடத்தில் சேர்க்கிறது. புதிய உறுப்பை ஏற்கனவே உள்ள உறுப்புகளின் இடையில் ஒரு குறிப்பிட்ட இடத்தில் சேர்க்கும் போது, ஏற்கனவே உள்ள உறுப்புகள் வலது பக்கமாக ஒரு இடம் நகர்த்தப்படும்.

9.1.8 Listலிருந்து உறுப்புகளை நீக்குதல்

List-ல் இருந்து ஒரு உறுப்பை நீக்குவதற்கு இரண்டு வழிகள் உள்ளன. அவை del கூற்று மற்றும் remove( ) செயற்கூறு ஆகும். del கூற்று தெரிந்த உறுப்புகளை நீக்குவதற்கு பயன்படுகிறது. ஆனால், remove( ) செயற்கூறு சுட்டெண் தெரியாத உறுப்புகளை List-லிருந்து நீக்குவதற்கு பயன்படுகிறது. del கூற்று முழு List-ஐ நீக்குவதற்கும் பயன்படுகிறது.

தொடரியல்:

del List *[index of an element]*
\# ஒரு குறிப்பிட்ட உறுப்பை நீக்க
del List *[index from : index to]*
\# ஒன்றுக்கும் மேற்பட்ட உறுப்புகளை நீக்க
del List
\# ஒரு List-யை நீக்க

எடுத்துக்காட்டு

```
>>> MySubjects = ['Tamil', 'Hindi', 'Telugu', 'Maths']
>>> print (MySubjects)
        ['Tamil', 'Hindi', 'Telugu', 'Maths']
>>> del MySubjects[1]
>>> print (MySubjects)
        ['Tamil', 'Telugu', 'Maths']
```

மேலே உள்ள எடுத்துக்காட்டில், **MySubjects** என்ற List நான்கு உறுப்புகளுடன் உருவாக்கப்பட்டுள்ளது. Print கூற்று List-ன் அனைத்து உறுப்புகளையும் காண்பிக்கிறது. >>> del MySubjects[1] கூற்று, சுட்டெண் 1 என்ற உறுப்பை நீக்குகிறது. அதன்பின் வருகின்ற print List-ன் மீதமுள்ள உறுப்புகளை காண்பிக்கிறது.

எடுத்துக்காட்டு

```
>>> del MySubjects[1:3]
>>> print(MySubjects)
        ['Tamil']
```

மேலே உள்ள குறிமுறையில், >>> **del MySubjects[1:3]** என்பது இரண்டு மற்றும் மூன்றாவது உறுப்புகளை List-லிருந்து நீக்குகிறது. சதுர அடைப்புக்குறிக்குள் குறிப்பிடப்பட்டுள்ள சுட்டெண்ணின் உச்ச மதிப்பு -1 என எடுத்துக்கொள்ளும்.

Chapter 9.indd 144 30-01-2020 19:37:06

எடுத்துக்காட்டு

```
>>> del MySubjects
>>> print(MySubjects)
Traceback (most recent call last):
File "<pyshell#9>", line 1, in <module>
print(MySubjects)
NameError: name 'MySubjects' is not defined
```

இங்கு, >>> del MySubjects, என்பது MySubjects என்ற List-ஐ முழுவதுமாக நீக்குகிறது. இப்போது உறுப்புகளை அச்சிட முயன்றால், பைத்தான் லிஸ்ட் is not defined என்ற பிழையை காண்பிக்கும். அதாவது, my Subjects என்ற List முழுவதுமாக நீக்கப்பட்டுவிட்டது என்று பொருள்.

ஏற்கனவே குறிப்பிட்டபடி, remove( ) செயற்கூறு சுட்டெண் தெரியாத ஒன்று அல்லது அதற்கு மேற்பட்ட உறுப்புகளை நீக்குவதற்கு பயன்படுகிறது. remove( ) செயற்கூறு மட்டுமல்லாமல், pop( ) செயற்கூறும் கொடுக்கப்பட்ட சுட்டெண்களை பயன்படுத்தி, ஒரு உறுப்பை நீக்க பயன்படுகிறது. pop( ) செயற்கூறு சுட்டெண் கொடுக்கப்படாத போது List-ன் கடைசி உறுப்பை நீக்கி அதை காண்பிக்கிறது.

clear( ) செயற்கூறு List-ன் அனைத்து உறுப்புகளையும் நீக்க பயன்படுகிறது. இது உறுப்புகளை மட்டுமே நீக்கி, List-ஐ தொடர்ந்து வைத்திருக்கிறது. Del கூற்று முழு லிஸ்ட்-ஐ நீக்குகிறது என்பதை நினைவில் கொள்க.

தொடரியல்:

List.*remove(element)*   # ஒரு குறிப்பிட்ட உறுப்பை நீக்க

List.*pop(index of an element)*

List.*clear( )*

எடுத்துக்காட்டு

```
>>> MyList=[12,89,34,'Kannan', 'Gowrisankar', 'Lenin']
>>> print(MyList)
        [12, 89, 34, 'Kannan', 'Gowrisankar', 'Lenin']
>>> MyList.remove(89)
>>> print(MyList)
        [12, 34, 'Kannan', 'Gowrisankar', 'Lenin']
```

மேலே உள்ள எடுத்துக்காட்டில், MyList மூன்று முழு எண்கள் மற்றும் மூன்று சரநிலையுரு உறுப்புகளுடன் உருவாக்கப்பட்டுள்ளது. அதன்பின் உள்ள print கூற்று List-ல் உள்ள அனைத்து உறுப்புகளையும் காண்பிக்கிறது. >>> MyList.remove(89) கூற்று, List-ல் இருந்து 89 என்ற உறுப்பை நீக்குகிறது. பின்வரும் print கூற்று மீதமுள்ள உறுப்புகளை காண்பிக்கிறது.

எடுத்துக்காட்டு

```
>>> MyList.pop(1)
        34
>>> print(MyList)
        [12, 'Kannan', 'Gowrisankar', 'Lenin']
```

மேலே உள்ள குறிமுறையில், pop( ) செயற்கூறு குறிப்பிட்ட உறுப்பை அதன் சுட்டெண்ணை பயன்படுத்தி நீக்குவதற்கு பயன்படுகிறது. உறுப்பு நீக்கப்பட்டவுடன் pop( ) செயற்கூறு நீக்கப்பட்ட உறுப்பை காண்பிக்கிறது. pop( ) செயற்கூறு லிஸ்ட்ல் இருந்து ஒரு உறுப்பை மட்டும் நீக்கப்பயன்படுகிறது. Del கூற்று பல உறுப்புகளை நீக்குகிறது என்பதை நினைவில் கொள்க.

எடுத்துக்காட்டு

```
>>> MyList.clear( )
>>> print(MyList)
        [ ]
```

மேலே உள்ள குறிமுறையில், clear( ) செயற்கூறு உறுப்புகளை மட்டும் நீக்கி List-ஐ தொடர்ந்து வைத்திருக்கப் பயன்படுகிறது. உறுப்புகள் நீக்கப்பட்ட List-ஐ அச்சிடுவதற்கு நீங்கள் முயன்றால், எந்தவொரு உறுப்பும் இல்லாத காலியான சதுர அடைப்புக் குறி தோன்றும். அதாவது லிஸ்ட் காலியாக இருக்கும்.

9.1.9 List மற்றும் range ( ) செயற்கூறுகள்

range( ) என்பது பைத்தானில் தொடர் மதிப்புகளை உருவாக்கப் பயன்படும் செயற்கூறாகும். range () செயற்கூறை பயன்படுத்தி நீங்கள் தொடர் மதிப்புகளுடன் List-ஐ உருவாக்கலாம். range( ) செயற்கூறு மூன்று செயலுருபுகளைக் கொண்டுள்ளது.

range ( ) செயற்கூறின் தொடரியல்:

range (start value, end value, step value)

இங்கு,

- **start value** – தொடரின் தொடக்க மதிப்பு. சுழியம் தானமைவு தொடக்க மதிப்பாகும்.
- **end value** – தொடரின் உச்ச வரம்பு. பைத்தான் இறுதி மதிப்பை உச்ச வரம்பு-1 என எடுத்துக் கொள்கிறது.
- **step value** – இது ஒரு விருப்ப செயலுருபு (கொடுக்க வேண்டியது கட்டாயமில்லை) இது வெவ்வேறு இடைவெளிகளில் மதிப்புகளை உருவாக்கப் பயன்படுகிறது.

Chapter 9.indd 146 30-01-2020 19:37:06

எடுத்துக்காட்டு 9.3: 10 வரை உள்ள முழு எண்களை உருவாக்குதல்:

```
for x in range (1, 11):
        print(x)
```

வெளியீடு

```
1
2
3
4
5
6
7
8
9
10
```

எடுத்துக்காட்டு 9.4: முதல் 10 இரட்டைப்படை எண்களை உருவாக்குதல்

```
for x in range (2, 11, 2):
        print(x)
```

வெளியீடு

```
2
4
6
8
10
```

9.1.9.1 தொடர் மதிப்புகளுடன் லிஸ்ட்–ஐ உருவாக்குதல்

range( ) செயற்கூறை பயன்படுத்தி நீங்கள் தொடர் மதிப்புகளுடன் கூடிய லிஸ்ட்-ஐ உருவாக்கலாம். range( ) செயற்கூறின் விடையை லிஸ்ட் ஆக மாற்றுவதற்கு, List( ) செயற்கூறு பயன்படுகிறது. List( ) செயற்கூறு range( ) ன் விடையை லிஸ்ட் ஆக உருவாக்குகிறது.

தொடரியல்:

List_*Varibale* = List (*range* ( ))

குறிப்பு

list()செயற்கூறு பைத்தானில் லிஸ்ட் உருவாக்கப்பயன்படுகிறது.

Chapter 9.indd 147 30-01-2020 19:37:06

எடுத்துக்காட்டு

```
>>> Even_List = List(range(2,11,2))
>>> print(Even_List)
[2, 4, 6, 8, 10]
```

மேலே உள்ள குறிமுறையில், List( )செயற்கூறு range( ) ன் விடையை Even_List என்ற List உறுப்புகளாக எடுத்துக் கொள்கிறது. எனவே, Even_List என்ற List முதல் ஐந்து இரட்டைப்படை எண்களை உறுப்புகளாக பெற்றிருக்கும்.

இதுபோன்று, எந்த தொடர் எண்களையும் range( ) பயன்படுத்தி உருவாக்கலாம். பின்வரும் எடுத்துக்காட்டு முதல் 10 இயல் எண்களின் 2ன் அடுக்கங்களுடன் கூடிய List-ஐ உருவாக்குதலை விளக்குகிறது.

எடுத்துக்காட்டு 9.5 முதல் 10 இயல் எண்களின் 2-ன் அடுக்குகளை உருவாக்குதல்

```
squares = [ ]
for x in range(1,11):
        s = x ** 2
        squares.append(s)
print (squares)
```

மேலே உள்ள நிரலில், "squares" என்ற பெயருடைய ஒரு வெற்று லிஸ்ட் உருவாக்கப்பட்டுள்ளது. பின்னர், for மடக்கு 1 முதல் 10 இயல் எண்களை range( ) செயற்கூறை பயன்படுத்தி உருவாக்குகிறது. மடக்கின் உள்ளே, **xன்** தற்போதைய மதிப்பு, 2-ன் அடுக்கேற்றத்தால் அதிகப்படுத்தப்பட்ட பின்னர் s என்ற மாறியில் சேமிக்கப்படுகிறது. இறுதியாக, நிரல் பின்வரும் மதிப்புகளை விடையாகக் காண்பிக்கிறது.

வெளியீடு

[1, 4, 9, 16, 25, 36, 49, 64, 81, 100]

9.1.10 List சுருக்கம் (List comprehensions)

List சுருக்கம் என்பது, கொடுக்கப்பட்ட நிபந்தனைகளுக்கு உட்பட்டு உருவாகும் தொடர் மதிப்புகளை ஏற்கும் List-யை உருவாக்கும் எளிய வழிமுறையாகும்.

தொடரியல்:

List = [*expression for variable in range*]

எடுத்துக்காட்டு 9.6: List சுருக்கத்தைப் பயன்படுத்தி முதல் 10 இயல் எண்களின் 2–ன் அடுக்குகளை உருவாக்குதல்.

```
>>> squares = [ x ** 2 for x in range(1,11) ]
>>> print (squares)
```

வெளியீடு

[1, 4, 9, 16, 25, 36, 49, 64, 81, 100]

Chapter 9.indd 148 30-01-2020 19:37:06

மேலே உள்ள எடுத்துக்காட்டில், x ** 2 என்ற கோவை மடக்குச் செயலின் ஒவ்வொரு சுழற்சியிலும் மதிப்பீடு செய்யப்படுகிறது. இது தொடர் மதிப்புகளை உருவாக்குவதற்கான குறுக்கு வழிமுறையாகும்.

9.1.11 பிற முக்கியமான List செயற்கூறுகள்

செயற்கூறு	விளக்கம்	தொடரியல்	எடுத்துக்காட்டு
copy ( )	List-ன் நகலை தரும்.	List.copy( )	MyList=[12, 12, 36] x = MyList.copy() print(x) வெளியீடு [12, 12, 36]
count ( )	List-ல் உள்ள ஒரே மாதிரியான உறுப்புகளின் எண்ணிக்கையை தரும்.	List.count(value)	MyList=[36 ,12 ,12] x = MyList.count(12) print(x) வெளியீடு 2
index ( )	முதலில் வரும் உறுப்பின் சுட்டெண் மதிப்பை தருகிறது.	List.index(element)	MyList=[36 ,12 ,12] x = MyList.index(12) print(x) வெளியீடு 1
reverse ( )	List-ல் உள்ள உறுப்புகளின் வரிசையை மறுபக்கமாக (தலைகீழாக) திருப்புகிறது,	List.reverse( )	MyList=[36 ,23 ,12] MyList.reverse() print(MyList) வெளியீடு [12 ,23 ,36]
sort ( )	List-ல் உள்ள உறுப்புகளை வரிசையாக்கம் செய்கிறது.	List.sort(reverse=True|False, key=myFunc)	
இரண்டு செயலுருபுகளும் கட்டாயமில்லாதவை • reverse ஐ True என பொருத்தினால் இறங்கு வரிசையில் லிஸ்ட் வரிசையாக்கமாகும். • ஏறுவரிசை தானமைவு வரிசையாகமாகும். • Key=myFunc; "myFunc" – வரிசையாக்க வரண்முறையைக் குறிப்பிடும் பயனர் வரையறுத்த செயற்கூறின் பெயர். குறிப்பு: sort( ) மூல List-யை பாதிக்கும்.		MyList=['Thilothamma', 'Tharani', 'Anitha', 'SaiSree', 'Lavanya'] MyList.sort( ) print(MyList) MyList.sort(reverse=True) print(MyList) வெளியீடு ['Anitha', 'Lavanya', 'SaiSree', 'Tharani', 'Thilothamma'] ['Thilothamma', 'Tharani', 'SaiSree', 'Lavanya', 'Anitha']	

Chapter 9.indd 149 30-01-2020 19:37:06

max( )	ஒரு List-ன் மதிப்புகளில் உச்ச மதிப்பை தரும்.	max(List)	MyList=[21,76,98,23] print(max(MyList)) வெளியீடு 98
min( )	ஒரு List-ல் உள்ள மதிப்புகளில், மிகக் குறைந்த மதிப்பைத் தரும்.	min(List)	MyList=[21,76,98,23] print(min(MyList)) வெளியீடு 21
sum( )	ஒரு List-டிலுள்ள மதிப்புகளின் கூட்டுத் தொகையை தரும்.	sum(List)	MyList=[21,76,98,23] print(sum(MyList)) வெளியீடு 218

9.1.12 List பயன்படுத்தப்பட்ட நிரல்கள்

நிரல் 1: 1 முதல் 20 வரையான எண்களில் 4-ல் வகுபடும் எண்களை பெறும் List ஒன்றை உருவாக்கும் நிரல் ஒன்றை எழுதுக.

```
divBy4=[ ]
for i in range(21):
        if (i%4==0):
                divBy4.append(i)
print(divBy4)
```

வெளியீடு

```
[0, 4, 8, 12, 16, 20]
```

நிரல் 2: BRICS ன் உறுப்பு நாடுகளின் List–ஐ வரையறை செய்து, உள்ளீடு செய்யப்படும் நாடு BRICS ன் உறுப்பா (அல்லது) இல்லையா என்பதை பரிசோதிக்கும் நிரலை எழுதுக.

```
country=["India", "Russia", "Srilanka", "China", "Brazil"]
is_member = input("Enter the name of the country: ")
if is_member in country:
        print(is_member, " is the member of BRICS")
else:
        print(is_member, " is not a member of BRICS")
```

வெளியீடு

```
Enter the name of the country: India
India  is the member of BRICS
```

வெளியீடு

```
Enter the name of the country: Japan
Japan  is not a member of BRICS
```


Chapter 9.indd 150 30-01-2020 19:37:06

நிரல் 3: ஆறு பாடங்களின் மதிப்பெண்களை உள்ளீடாகப் பெற்று, ஒவ்வொரு பாடத்திலும் பெற்ற மதிப்பெண்களை அச்சிட்டு மதிப்பெண்களின் கூட்டுத்தொகையை காண்பிக்கும் பைத்தான் நிரலை எழுதுக.

```
marks=[]
subjects=['Tamil', 'English', 'Physics', 'Chemistry', 'Comp. Science', 'Maths']
for i in range(6):
        m=int(input("Enter Mark = "))
        marks.append(m)
for j in range(len(marks)):
        print("{ }. { } Mark = { } ".format(j1+,subjects[j],marks[j]))
print("Total Marks = ", sum(marks))
```

வெளியீடு

```
Enter Mark = 45
Enter Mark = 98
Enter Mark = 76
Enter Mark = 28
Enter Mark = 46
Enter Mark = 15
  1. Tamil Mark = 45
  2. English Mark = 98
  3. Physics Mark = 76
  4. Chemistry Mark = 28
  5. Comp. Science Mark = 46
  6. Maths Mark = 15
  Total Marks =  308
```

நிரல் 4: 5 பொருள்களின் விலையை List–ல் பெற்று, அனைத்து விலைகளின் கூட்டுத்தொகை, பெருக்கற்தொகை மற்றும் சராசரியைக் கண்டறியும் பைத்தான் நிரலை எழுதுக.

```
items=[]
prod=1
for i in range(5):
        print ("Enter price for item { } : ".format(i+1))
        p=int(input())
        items.append(p)
for j in range(len(items)):
        print("Price for item { } = Rs. { }".format(j+1,items[j]))
        prod = prod * items[j]
print("Sum of all prices = Rs.", sum(items))
print("Product of all prices = Rs.", prod)
print("Average of all prices = Rs.",sum(items)/len(items))
```


Chapter 9.indd 151 30-01-2020 19:37:07

வெளியீடு

```
Enter price for item  1 :
5
Enter price for item  2 :
10
Enter price for item  3 :
15
Enter price for item  4 :
20
Enter price for item  5 :
25
Price for item  1 = Rs. 5
Price for item  2 = Rs. 10
Price for item  3 = Rs. 15
Price for item  4 = Rs. 20
Price for item  5 = Rs. 25
Sum of all prices = Rs. 75
Product of all prices = Rs. 375000
Average of all prices = Rs. 15.0
```

நிரல் 5: ஆண்டுக்கு 1 இலட்சத்திற்கு மேல் ஊதியம் பெறும் பணியாளர்களின் எண்ணிக்கையை கணக்கிடும் பைத்தான் நிரல் n எண்ணிக்கையிலான பணியாளர்களின் மாத ஊதியத்தை உள்ளீடாக பெற வேண்டும்.

```
count=0
n=int(input("Enter no. of employees: "))
print("No. of Employees",n)
salary=[]
for i in range(n):
        print("Enter Monthly Salary of Employee { } Rs.: ".format(i+1))
        s=int(input())
        salary.append(s)

for j in range(len(salary)):
        annual_salary = salary[j] * 12
        print ("Annual Salary of Employee { } is:Rs. { }".format(j+1,annual_salary))
        if annual_salary >= 100000:
                count = count + 1
print("{ } Employees out of { } employees are earning more than Rs. 1 Lakh per annum".
format(count, n))
```


Chapter 9.indd 152 30-01-2020 19:37:07

வெளியீடு

```
Enter no. of employees: 5
No. of Employees 5
Enter Monthly Salary of Employee 1 Rs.:
3000
Enter Monthly Salary of Employee 2 Rs.:
9500
Enter Monthly Salary of Employee 3 Rs.:
12500
Enter Monthly Salary of Employee 4 Rs.:
5750
Enter Monthly Salary of Employee 5 Rs.:
8000
Annual Salary of Employee 1 is:Rs. 36000
Annual Salary of Employee 2 is:Rs. 114000
Annual Salary of Employee 3 is:Rs. 150000
Annual Salary of Employee 4 is:Rs. 69000
Annual Salary of Employee 5 is:Rs. 96000
2 Employees out of 5 employees are earning more than Rs. 1 Lakh per annum
```

நிரல் 6: 1 முதல் 10 வரையுள்ள தொடர் எண்களை ஏற்கும் List ஒன்றை உருவாக்குவதற்கான நிரலை எழுதுக. பிறகு அனைத்து இரட்டைப்படை எண்களையும் நீக்கவிட்டு இறுதிப்பட்டியலை அச்சிடுக.

```
Num = []
for x in range(1,11):
    Num.append(x)
print("The List of numbers from 1 to 10 = ", Num)

for index, i in enumerate(Num):
    if(i%2==0):
        del Num[index]
print("The List after deleting even numbers = ", Num)
```

வெளியீடு

```
The List of numbers from 1 to 10 =  [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
The List after deleting even numbers =  [1, 3, 5, 7, 9]
```

நிரல் 7: பைபோனாசி வரிசையை உருவாக்கி அதை List –ல் சேமிப்பதற்கான நிரலை எழுதுக. பிறகு அனைத்து மதிப்புகளின் கூட்டுத்தொகையைக் கண்டறிக.

```
a=-1
b=1
n=int(input("Enter no. of terms: "))
i=0
sum=0
Fibo=[]
while i<n:
        s = a + b
        Fibo.append(s)
        sum+=s
        a = b
        b = s
        i+=1
print("Fibonacci series upto "+ str(n) +" terms is : " + str(Fibo))
print("The sum of Fibonacci series: ",sum)
```

வெளியீடு

```
        Enter no. of terms: 10
        Fibonacci series upto 10 terms is : [0, 1, 1, 2, 3, 5, 8, 13, 21, 34]
        The sum of Fibonacci series:  88
```

9.2 Tuples

Tuples அறிமுகம்:

Tuples காற்புள்ளியால் பிரிக்கப்பட்ட பல மதிப்புகளை வளைவு அடைப்புக் குறிக்குள் கொண்ட தரவினமாகும். டப்புல்ஸ் List-க்கு இணையானதாகும். ஆனால் List-ல் மதிப்புகளை மாற்ற முடியும். Tuples -ல் மாற்ற முடியாது.

உங்களுக்குத் தெரியுமா?

Tuples என்ற வார்த்தை லத்தீன் மொழியிலிருந்து தோற்றுவிக்கப்பட்டது. இந்த லத்தீன் வார்த்தை எண் தொடர்களைக் குறிக்கிறது:

single(1), double(2), triple(3), quadruple(4), quinTuples(5), sexTuples(6), sepTuples(7), ocTuples(8), ..., n-Tuples, ...,

9.2.1 List–க்கு மேலான Tuples–ன் நன்மைகள்

1. List-ன் உறுப்புகளை மாற்றலாம் ஆனால் Tuples-ன் உறுப்புகளை மாற்ற முடியாது. இதுவே List மற்றும் Tuples-க்கு இடையே உள்ள முக்கியமான வேறுபாடு ஆகும்.

2. List-ன் உறுப்புகள் சதுர அடைப்புக் குறிக்குள் அடைக்கப்பட்டிருக்கும். ஆனால், Tuple -ன் உறுப்புகள் வளைவு குறிக்குள் அடைக்கப்பட்டிருக்கும்

3. Tuples-ன் மடக்குச் செயல் List-ஐ காட்டிலும் விரைவானது.

9.2.2 Tuples உருவாக்குதல்

List-ஐ போன்றே Tuples உருவாக்கப்படும். List-ல் உறுப்புகள் சதுர அடைப்புக்குறிக்குள் அடைக்கப்பட்டு வரையறை செய்யப்படுகின்றன. ஆனால் டப்புள்ஸ்ல் அவைகள் வளைந்த அடைப்புக் குறிக்குள் அடைக்கப்பட்டிருக்கலாம். Tuples-ன் உறுப்புகளை வளைந்த அடைப்புக்குறி இல்லாமலும் வரையறுக்க முடியும். எவ்வாறு பயன்படுத்தினாலும் Tuples-ன் வேலை செய்யும் விதம் ஒன்றாகவே இருக்கும்.

தொடரியல்:

```
# வெற்று Tuples
Tuples_Name = ( )
# n எண்ணிக்கை உறுப்புகளுடன் Tuples
Tuples_Name = (E1, E2, E2 ....... En)
# அடைப்புக்குறி இல்லாத Tuple உறுப்புகள்
Tuples_Name = E1, E2, E3 ..... En
```

எடுத்துக்காட்டு

```
>>> MyTup1 = (23, 56, 89, 'A', 'E', 'I', "Tamil")
>>> print(MyTup1)
        (23, 56, 89, 'A', 'E', 'I', 'Tamil')

>>> MyTup2 = 23, 56, 89, 'A', 'E', 'I', "Tamil"
>>> print (MyTup2)
        (23, 56, 89, 'A', 'E', 'I', 'Tamil')
```

9.2.2.1 செயற்கூறை பயன்படுத்தி Tuples உருவாக்குதல்

Tuples ( ) செயற்கூறு லிஸ்ட்டிலிருந்து Tuples- ஐ உருவாக்கவும் பயன்படுகிறது. லிஸ்ட்-ல் இருந்து Tuples-ஐ உருவாக்கும் போது, உறுப்புகள் சதுர அடைப்புக்குறிக்குள் அடைக்கப்பட்டிருக்க வேண்டும்.

தொடரியல்:

```
Tuples_Name = Tuples ( [List elements] )
```

எடுத்துக்காட்டு

```
>>> MyTup3 = tuple( [23, 45, 90] )
>>> print(MyTup3)
        (23, 45, 90)
>>> type (MyTup3)
        <class 'Tuples'>
```


Chapter 9.indd 155 30-01-2020 19:37:07

குறிப்பு

பைத்தானில் ஒரு பொருளின் தரவினத்தை அறிய type() செயற்கூறு பயன்படுகிறது.

9.2.2.2 ஒற்றை உறுப்பு கொண்ட Tuples உருவாக்குதல்

Tuples-ஐ ஒற்றை உறுப்புடன் உருவாக்கும் போது, உறுப்பின் இறுதியில் காற்புள்ளியை சேர்க்க வேண்டும். காற்புள்ளி இல்லையெனில், பைத்தான் அந்த உறுப்பை ஒரு சாதாரண தரவினம் போல எடுத்துக் கொள்ளும், Tuples ஆக கருதாது. ஒரு உறுப்புடன் Tuples- ஐ உருவாக்குவது "சிங்கிள்டன் (singleton)" Tuples என அழைக்கப்படுகிறது.

எடுத்துக்காட்டு

```
>>> MyTup4 = (10)
>>> type(MyTup4)
        <class 'int'>
>>> MyTup5 = (10,)
>>> type(MyTup5)
        <class 'Tuples'>
```

9.2.3 Tuples–ன் மதிப்புகளை அணுகுதல்

List-ஐ போலவே, Tuples உறுப்புகளும் சுழியத்தில் தொடங்கும் சுட்டெண்களை பெற்றிருக்கும். Tuples-ன் உறுப்புகளை சுட்டெண்ணை பயன்படுத்தி எளிமையாக அணுகலாம்.

எடுத்துக்காட்டு

```
>>> Tup1 = (12, 78, 91, "Tamil", "Telugu", 3.14, 69.48)

# Tuples-ன் அனைத்து உறுப்புகளையும் அணுக.
>>> print(Tup1)
        (12, 78, 91, 'Tamil', 'Telugu', 3.14, 69.48)

#சுட்டெண்ணை பயன்படுத்தி தேர்வு செய்யப்பட்ட உறுப்புகளை அணுகுதல்.
>>> print(Tup1[2:5])
        (91, 'Tamil', 'Telugu')

#முதல் உறுப்பிலிருந்து குறிப்பிடப்பட்ட சுட்டெண் வரை அணுகுதல்.
>>> print(Tup1[:5])
        (12, 78, 91, 'Tamil', 'Telugu')

# குறிப்பிடப்பட்ட உறுப்பிலிருந்து இறுதி உறுப்பு வரை அணுகுதல்.
>>> print(Tup1[4:])
        ('Telugu', 3.14, 69.48)

# முதல் உறுப்பிலிருந்து இறுதி உறுப்பு வரை அணுகுதல்
>>> print(Tup1[:])
        (12, 78, 91, 'Tamil', 'Telugu', 3.14, 69.48)
```


Chapter 9.indd 156 30-01-2020 19:37:07

9.2.4 Tuples–ஐ புதுப்பித்தல் மற்றும் நீக்குதல்:

நீங்கள் அறிந்தபடி, Tuples-ஐ மாற்ற முடியாது என அறிவோம். அதாவது Tuples-ன் உறுப்புகளை மாற்றம் (புதுப்பிக்க) செய்ய முடியாது . Tuples-ன் மதிப்புகளை மாற்றுவதற்கு பதிலாக, இரண்டு Tuples- ஐ இணைக்கவும் அல்லது முழு Tuples-ஐ நீக்கவும் முடியும்.

எடுத்துக்காட்டு

```
# இரண்டு Tuples-களை இணைக்கும் நிரல்
Tup1 = (2,4,6,8,10)
Tup2 = (1,3,5,7,9)
Tup3 = Tup1 + Tup2
print(Tup3)
```

வெளியீடு

```
(2, 4, 6, 8, 10, 1, 3, 5, 7, 9)
```

முழு Tuples-ஐ நீக்குவதற்கு, del கட்டளை பயன்படுகிறது.

தொடரியல்:

del Tuples_*name*

எடுத்துக்காட்டு

```
Tup1 = (2,4,6,8,10)
print("The elements of Tup1 is ", Tup1)
del Tup1
print (Tup1)
```

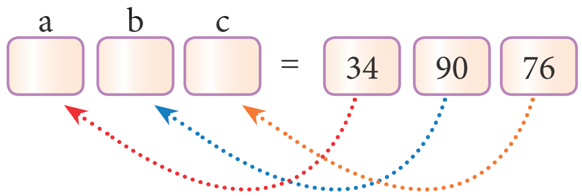
வெளியீடு

```
The elements of Tup1 is  (2, 4, 6, 8, 10)
Traceback (most recent call last):
File "D:/Python/Tuple Examp 1.py", line 4, in <module>
print (Tup1)
NameError: name 'Tup1' is not defined
```

குறிப்பு: மேலே உள்ள குறிமுறையில் print கூற்று உறுப்புகளை அச்சிடுகிறது. பின்னர், del கூற்று முழு Tuples-ஐ நீக்குகிறது. நீக்கப்பட்ட Tuples-ஐ நீங்கள் அச்சிட முயன்றால், பைத்தான் பிழையைக் காண்பிக்கும்.

9.2.5 Tuples-க்கு மதிப்பிருத்துதல்

Tuples மதிப்பிருத்தல் என்பது பைத்தானில் ஆற்றல் மிக்க சிறப்பியல்பாகும். இது, மதிப்பிருத்து செயற்குறியின் இடது பக்கத்தில் உள்ள மாறிகளில் மதிப்பிருத்த செயற்குறியின் வலது பக்கத்தில் உள்ள மதிப்புகளை இருத்த அனுமதிக்கிறது. ஒவ்வொரு மதிப்பும் அதற்குரிய மாறியில் இருத்தப்படுகிறது.



எடுத்துக்காட்டு

```
>>> (a, b, c) = (34, 90, 76)
>>> print(a,b,c)
        34 90 76
# மதிப்பிருத்துவதற்கு முன் கோவைகள் மதிப்பீடு செய்யப்படுகின்றன
>>> (x, y, z, p) = (2**2, 5/3+4, 15%2, 34>65)
>>> print(x,y,z,p)
        4 5.666666666666667 1 False
```

குறிப்பு: நீங்கள், Tuples- ல் மதிப்புகளை இருத்தும் போது, மதிப்பிருத்து செயற்குறியின் இரு பக்கங்களிலும் உள்ள மதிப்புகளின் எண்ணிக்கை சமமாக உள்ளதா என்பதை உறுதி செய்து கொள்ள வேண்டும். இல்லையெனில், பைத்தானில் பிழை உருவாக்கப்படும்.

9.2.6 Tuples–ல் ஒன்றுக்கும் மேற்பட்ட மதிப்புகளை திருப்புதல்

ஒரு செயற்கூறு ஒரு நேரத்தில் ஒரு மதிப்பை மட்டுமே திருப்பி அனுப்ப முடியும். ஆனால் பைத்தான், செயற்கூறிலிருந்து ஒன்றுக்கு மேற்பட்ட மதிப்புகளை திருப்பி அனுப்புகிறது. பைத்தான் பல மதிப்புகளை தொகுத்து, அவைகளை ஒன்றாக திருப்பி அனுப்புகிறது.

எடுத்துக்காட்டு 9.7: List அதிகபட்ச மற்றும் குறைந்தபட்ச மதிப்புகளை திருப்பி அனுப்பும் நிரல்

```
def Min_Max(n):
    a = max(n)
    b = min(n)
    return(a, b)
Num = (12, 65, 84, 1, 18, 85, 99)
(Max_Num, Min_Num) = Min_Max(Num)
print("Maximum value = ", Max_Num)
print("Minimum value = ", Min_Num)
```

வெளியீடு

```
    Maximum value =  99
    Minimum value =  1
```


Chapter 9.indd 158 30-01-2020 19:37:07

9.2.7 பின்னலான Tuples

பைத்தானில், ஒரு Tuples-ஐ மற்றொரு Tuples-க்குள் வரையறை செய்வதை பின்னலான Tuples என்கிறோம். பின்னலான Tuples-ல் ஒவ்வொரு Tuples-ல் ஒரு உறுப்பாக கருதப்படுகிறது. for மடக்கு பின்னலான Tuples-ன் அனைத்து உறுப்புகளை அணுகுவதற்கு பயன்படுகிறது.

எடுத்துக்காட்டு

```
Toppers = (("Vinodini", "XII-F", 98.7), ("Soundarya", "XII-H", 97.5),
           ("Tharani", "XII-F", 95.3), ("Saisri", "XII-G", 93.8))
for i in Toppers:
      print(i)
```

வெளியீடு

```
('Vinodini', 'XII-F', 98.7)
('Soundarya', 'XII-H', 97.5)
('Tharani', 'XII-F', 95.3)
('Saisri', 'XII-G', 93.8)
```

குறிப்பு

லிஸ்ட்ல் பயன்படுத்திய அனைத்து செயற்கூறுகளும் Tuples-க்கும் பொருந்தும்

9.2.8 Tuples-ஸ் பயன்படுத்தப்பட்ட நிரல்கள்

நிரல் 1: Tuples-ஸ் மதிப்பிருத்தலை பயன்படுத்தி இரண்டு மதிப்புகளை இடமாற்றுவதற்கான நிரலை எழுதுக.

```
a = int(input("Enter value of A: "))
b = int(input("Enter value of B: "))
print("Value of A = ", a, "\n Value of B = ", b)
(a, b) = (b, a)
print("Value of A = ", a, "\n Value of B = ", b)
```

வெளியீடு

```
Enter value of A: 54
Enter value of B: 38
Value of A =  54
 Value of B =  38
Value of A =  38
 Value of B =  54
```


Chapter 9.indd 159 30-01-2020 19:37:07

நிரல் 2: செயற்கூறைப் பயன்படுத்தி, வட்டத்தின் ஆரத்தை செயற்கூறுக்கு அளபுருவாக அனுப்பி, வட்டத்தின் பரப்பு மற்றும் சுற்றளவை திருப்பி அனுப்புவதற்கான நிரலை எழுதுக.

```
pi = 3.14
def Circle(r):
        return (pi*r*r, 2*pi*r)

radius = float(input("Enter the Radius: "))
(area, circum) = Circle(radius)
print ("Area of the circle = ", area)
print ("Circumference of the circle = ", circum)
```

வெளியீடு

```
Enter the Radius: 5
Area of the circle =  78.5
Circumference of the circle =  31.400000000000002
```

நிரல் 3: நேர்மறை மற்றும் எதிர்மறை எண்களின் List–ஐ கொண்ட நிரலை எழுதுக. லிஸ்ட்ல் இருந்து நேர்ம எண்களை மட்டும் பெறுகின்ற புதிய Tuples உருவாக்குக.

```
Numbers = (5, -8, 6, 8, -4, 3, 1)
Positive = ( )
for i in Numbers:
        if i > 0:
                Positive += (i, )
print("Positive Numbers: ", Positive)
```

வெளியீடு

```
Positive Numbers:  (5, 6, 8, 3, 1)
```

9.3 SET

அறிமுகம்

பைத்தானில், Set என்பது தரவின தொகுப்பின் மற்றொரு வகையாகும். Set என்பது மாறக்கூடிய மற்றும் நகல்கள் இல்லாத வரிசைப்படுத்தப்படாத உறுப்புகளின் தொகுப்பாகும். அதாவது, Set-ல் உள்ள உறுப்புகள் மீண்டும் இடம்பெற முடியாது. இந்த சிறப்பியல்பு உறுப்பு சோதனையை சேர்க்கவும் மற்றும் நகல் உறுப்புகளை நீக்கவும் பயன்படுகிறது.

9.3.1 Set உருவாக்குதல்

நெளிவு அடைப்புக்குறிக்குள் காற்புள்ளியால் பிரிக்கப்பட்ட அனைத்து உறுப்புகளையும் இருத்துவதன் மூலம் Set உருவாக்கப்படுகிறது. Set() செயற்கூறு, பைதானில் Set-ஐ உருவாக்கப் பயன்படுகிறது.

தொடரியல்:

Set_Variable = {E1, E2, E3 …….. En}

Chapter 9.indd 160 30-01-2020 19:37:07

TF1B1L

எடுத்துக்காட்டு

```
>>> S1={1,2,3,'A',3.14}
>>> print(S1)
        {1, 2, 3, 3.14, 'A'}

>>> S2={1,2,2,'A',3.14}
>>> print(S2)
        {1, 2, 'A', 3.14}
```

மேலே கொடுக்கப்பட்ட எடுத்துக்காட்டில், Set S1 என்பது நகல்கள் இல்லாத பல்வேறு வகை உறுப்புகளுடன் உருவாக்கப்பட்டுள்ளது. Set S2 என்பது நகல் மதிப்புகளுடன் உருவாக்கப்பட்டுள்ளது. ஆனால் பைத்தான், நகல் மதிப்புகளிலிருந்து ஒரேயொரு உறுப்பை மட்டுமே எடுத்துக் கொள்ளும். அதாவது, பைத்தான் நகல் மதிப்புகளை நீக்கிவிடும் ஏனெனில் பைத்தானில் Set நகல் உறுப்புகளை பெற்றிருக்காது.

குறிப்பு

Set-ல் இருந்து உறுப்புகளை அச்சிடும் பொழுது, பைத்தான் மதிப்புகளை வெவ்வேறு வரிசையில் காண்பிக்கும்.

9.3.2 List அல்லது Tuple ஐ பயன்படுத்தி Set உருவாக்குதல்

ஒரு List அல்லது Tuple ஐ Set() செயற்கூறைப் பயன்படுத்தி Set ஆக மாற்ற முடியும். இது மிகவும் எளிய முறையாகும். List அல்லது Tuple ஐ உருவாக்கிய பின் அதன் மாறியை Set() செயற்கூறினுள் அளபுருவாக கொடுக்க வேண்டும்.

எடுத்துக்காட்டு

```
MyList=[2,4,6,8,10]
MySet=set(MyList)
print(MySet)
```

வெளியீடு

```
        {2, 4, 6, 8, 10}
```

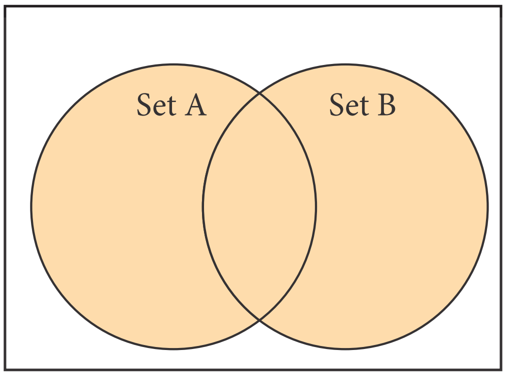
9.3.3 Set செயற்பாடுகள்

கணிதத்தில் கற்ற Set செயற்பாடுகள் ஆகிய ஒட்டு (Union), வெட்டு (intersection)வேறுபாடு (difference) சமச்சீரான வேறுபாடு (Symmetric difference), போன்றவற்றை பைத்தானிலும் பயன்படுத்தலாம்.

9.3.3.1 ஒட்டு (Union):

இது இரண்டு அல்லது அதற்கு மேற்பட்ட Set-களின் அனைத்து உறுப்புகளையும் உள்ளடக்கும்.

Chapter 9.indd 161 30-01-2020 19:37:07



பைத்தானில் | என்ற செயற்குறி இரண்டு Set களின் ஒட்டை உருவாக்கப்பயன்படுகிறது. **Union** செயற்கூறும் பைத்தானில் இரண்டு Set களை இணைக்கப் பயன்படுகிறது.

எடுத்துக்காட்டு: ஒட்டு (Union) செயற்குறியைப் பயன்படுத்தி இரண்டு Set களை இணைப்பதற்கான நிரல்

```
set_A={2,4,6,8}
set_B={'A', 'B', 'C', 'D'}
U_set=set_A|set_B
print(U_set)
```

வெளியீடு

```
{2, 4, 6, 8, 'A', 'D', 'C', 'B'}
```

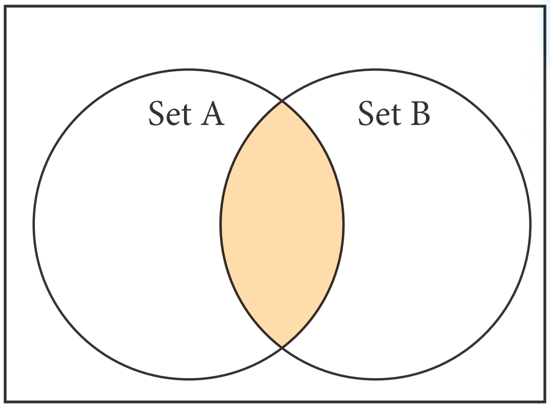
எடுத்துக்காட்டு: ஒட்டு செயற்கூறுவைப் பயன்படுத்தி இரண்டு Setகளை இணைப்பதற்கான நிரல்

```
set_A={2,4,6,8}
set_B={'A', 'B', 'C', 'D'}
set_U=set_A.union(set_B)
print(set_U)
```

வெளியீடு

```
{'D', 2, 4, 6, 8, 'B', 'C', 'A'}
```

9.3.3.2 வெட்டு (intersection) இது இரண்டு Setகளின் பொதுவான உறுப்புகளை உள்ளடக்கியது



Chapter 9.indd 162 30-01-2020 19:37:07

பைத்தானில் & செயற்குறி இரண்டு Set களை வெட்டுவதற்கு பயன்படுகிறது. வெட்டு (intersection()) செயற்கூறும் பைத்தானில் இரண்டு Set களை வெட்டுவதற்கு பயன்படுகிறது.

எடுத்துக்காட்டு: வெட்டு intersection()செயற்குறியைப் பயன்படுத்தி இரண்டு Setகளை வெட்டுவதற்கான நிரல்

```
set_A={'A', 2, 4, 'D'}
set_B={'A', 'B', 'C', 'D'}
print(set_A & set_B)
```

வெளியீடு

```
{'A', 'D'}
```

எடுத்துக்காட்டு: வெட்டு intersection()செயற்குறியைப் பயன்படுத்தி இரண்டு Setகளை வெட்டுவதற்கான நிரல்

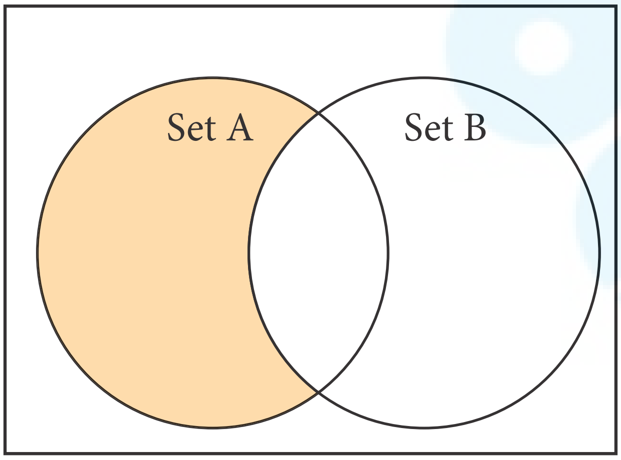
```
set_A={'A', 2, 4, 'D'}
set_B={'A', 'B', 'C', 'D'}
print(set_A.intersection(set_B))
```

வெளியீடு

```
{'A', 'D'}
```

9.3.3.3 வேறுபாடு Difference

இது முதல் Set(A) ல் உள்ள அனைத்து உறுப்புகளையும் உள்ளடக்கியது. இது இரண்டாவது Set-ஐ தவிர்க்கிறது.



பைத்தானில் –(minus) செயற்குறி Set செயற்பாட்டின் வேறுபாட்டைக் கண்டறிய பயன்படுகிறது. difference() செயற்கூறும் வேறுபாட்டு செயற்பாட்டிற்காக பயன்படுகிறது.

எடுத்துக்காட்டு: செயற்குறியைப் பயன்படுத்தி இரண்டு Setகளின் வேற்றுமைக்கான நிரல்

```
set_A={'A', 2, 4, 'D'}
set_B={'A', 'B', 'C', 'D'}
print(set_A - set_B)
```

வெளியீடு

```
{2, 4}
```

எடுத்துக்காட்டு: difference செயற்கூறுவைப் பயன்படுத்தி இரண்டு Setகளின் வேற்றுமைக்கான நிரல்

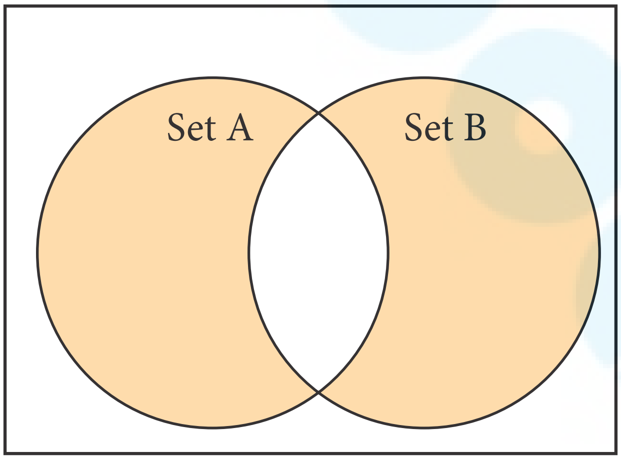
```
set_A={'A', 2, 4, 'D'}
set_B={'A', 'B', 'C', 'D'}
print(set_A.difference(set_B))
```

வெளியீடு

```
{2, 4}
```

9.3.3.4 சமச்சீரான வேறுபாடு (Symmetric difference)

இது இரண்டு Set-ல் உள்ள பொதுவான உறுப்புகளை மட்டும் தவிர்த்து மற்ற அனைத்து உறுப்புகளையும் உள்ளடக்கியது.



Caret(^) செயற்குறி பைத்தானில் சமச்சீரான வேறுபாட்டை கண்டறிய பயன்படுகிறது. Symmetric difference() செயற்கூறும் அதே செயலை செய்ய பயன்படுகிறது.

Chapter 9.indd 164 30-01-2020 19:37:07

எடுத்துக்காட்டு: Caret(^) செயற்குறியைப் பயன்படுத்தி சமச்சீரான வேறுபாட்டை கண்டறியும் நிரல்.

```
set_A={'A', 2, 4, 'D'}
set_B={'A', 'B', 'C', 'D'}
print(set_A ^ set_B)
```

வெளியீடு

```
{2, 4, 'B', 'C'}
```

எடுத்துக்காட்டு: Symmetric difference செயற்கூறைப் பயன்படுத்தி சமச்சீரான வேறுபாட்டை கண்டறியும் நிரல்

```
set_A={'A', 2, 4, 'D'}
set_B={'A', 'B', 'C', 'D'}
print(set_A.symmetric_difference(set_B))
```

வெளியீடு

```
{2, 4, 'B', 'C'}
```

9.3.4 setகளைப் பயன்படுத்தி நிரல்

நிரல் 1: பகா எண்களைக் கொண்ட ஒரு Set, இரட்டைப்படை எண்களை கொண்ட மற்றொரு Set உருவாக்குவதற்கான நிரல் (ஒட்டு, வெட்டு, வேறுபாடு, சமச்சீரான வேறுபாடு போன்ற செயற்பாடுகளுக்கான விடையை நிரூபிக்கவும்).

எடுத்துக்காட்டு

```
even=set([x*2 for x in range(1,11)])
primes=set()
for i in range(2,20):
        j=2
        f=0
        while j<=i/2:
                if i%j==0:
                        f=1
                j+=1
        if f==0:
                primes.add(i)
print("Even Numbers: ", even)
print("Prime Numbers: ", primes)
print("Union: ", even.union(primes))
print("Intersection: ", even.intersection(primes))
print("Difference: ", even.difference(primes))
print("Symmetric Difference: ", even.symmetric_difference(primes))
```

வெளியீடு

Even Numbers: {2, 4, 6, 8, 10, 12, 14, 16, 18, 20}
Prime Numbers: {2, 3, 5, 7, 11, 13, 17, 19}
Union: {2, 3, 4, 5, 6, 7, 8, 10, 11, 12, 13, 14, 16, 17, 18, 19, 20}
Intersection: {2, 4}
Difference: {6, 8, 10, 12, 14, 16, 18, 20}
Symmetric Difference: {3, 5, 6, 7, 8, 10, 11, 12, 13, 14, 16, 17, 18, 19, 20}

9.4 Dictionaries

அறிமுகம்

பைத்தானில், Dictionary என்பது பல்வேறு வகையான உறுப்புகளின் தொகுப்பாகும். பிற தரவினங்களாகிய List அல்லது Tuples போன்று இல்லாமல், Dictionary வகை உறுப்புகளுடன் அதற்கான திறவு கோலையும் (இணைப்புப் பெயர்) சேமிக்கிறது. பைத்தான் Dictionary உள்ள திறவு கோல்கள் முக்காற்புள்ளியாலும் (:) உறுப்புகள் காற்புள்ளியாலும் (,) பிரிக்கப்பட வேண்டும். திறவுகோல் மற்றும் மதிப்புகள் நெளிவு அடைப்புக்குறிக்குள் { } வரையறுக்கப்படும்.

Dictionaryஐ வரையறுப்பதற்கான தொடரியல்:

```
Dictionary_Name = { Key_1: Value_1,
                    Key_2:Value_2,
                    ……..
                    Key_n:Value_n
                  }
```

Dictionary உள்ள திறவுகோல் தனித்தன்மை வாய்ந்த ஒரே வகையான (Case Sensitive) மற்றும் பைத்தானில் எந்தவொரு தகுதிவாய்ந்த தரவினமாகவும் இருக்கலாம்.

9.3.1 Dictionaryஐ உருவாக்குதல்

காலியான வெற்று Dictionary

Dict1 = { }

திறவுகோலைக் கொண்ட Dictionary

Dict_Stud = { 'RollNo': 1234, 'Name':'Murali', 'Class':'XII', 'Marks':451}

9.3.2 Dictionary சுருக்கம்

பைத்தானில், சுருக்கம் என்பது Dictionary ஐ உருவாக்குவதற்கான மற்றொரு வழியாகும். இந்த வகை Dictionary ஐ உருவாக்குவதற்கான தொடரியல் பின்வருமாறு கொடுக்கப்பட்டுள்ளது.

Chapter 9.indd 166 30-01-2020 19:37:07

தொடரியல்:

Dict = { expression for variable in sequence [if condition] }

if நிபந்தனை கட்டாயமல்ல. If நிபந்தனையை குறிப்பிட்டால், நிபந்தனை பூர்த்தி செய்யும் மதிப்புகள் மட்டுமே கோவையைப் பயன்படுத்தி மதிப்பீடு செய்யப்படுகின்றன.

எடுத்துக்காட்டு

*Dict = { x : 2 * x for x in range(1,10)}*

மேலே உள்ள குறிமுறையின் வெளியீடு

{1: 2, 2: 4, 3: 6, 4: 8, 5: 10, 6: 12, 7: 14, 8: 16, 9: 18}

9.3.3 Dictionary உறுப்புகளை அணுகுதல், சேர்த்தல், மாற்றியிடுதல் மற்றும் நீக்குதல்

Dictionary ன் அனைத்து உறுப்புகளையும் அணுகுதல் List மற்றும் Tuples-ஐ போன்றே உள்ளது. எளிய print செயற்கூறு அனைத்து உறுப்புகளையும் அணுகப் பயன்படுகிறது. நீங்கள், ஒரு குறிப்பிட்ட உறுப்பை அணுக விரும்பினால், திறவுகோலுடன் சதுர அடைப்புக்குறி பயன்படுத்தப்பட வேண்டும்.

எடுத்துக்காட்டு 9.8: Dictionary சேமிக்கப்பட்ட அனைத்து மதிப்புகளையும் அணுகுவதற்கான நிரல்.

```
MyDict = { 'Reg_No': '1221',
                'Name' : 'Tamilselvi',
                'School' : 'CGHSS',
                'Address' : 'Rotler St., Chennai 112' }
print(MyDict)
print("Register Number: ", MyDict['Reg_No'])
print("Name of the Student: ", MyDict['Name'])
print("School: ", MyDict['School'])
print("Address: ", MyDict['Address'])
```

வெளியீடு

```
{'Reg_No': '1221', 'Name': 'Tamilselvi', 'School': 'CGHSS', 'Address': 'Rotler St., Chennai 112'}
Register Number:  1221
Name of the Student:  Tamilselvi
School:  CGHSS
Address:  Rotler St., Chennai 112
```

குறிப்பு: முதல் print கூற்று Dictionary–ல் உள்ள அனைத்த மதிப்புகளையும் அச்சிடுகிறது. மற்ற கூற்றுகள், சதுர அடைப்புக்குறிக்குள் உள்ள குறிப்பிட்ட மதிப்புகளை மட்டுமே அச்சிடுகின்றன.

ஏற்கனவே உள்ள Dictionary-ல் மதிப்பை அதன் திறவுகோலுடன் இருத்துவதன் வழியாக பல மதிப்புகளை சேர்க்கலாம். பின்வரும் தொடரியல் Dictionary பல உறுப்புகள் சேர்த்தலை புரிந்து கொள்ள பயன்படுகிறது.

dictionary_name [key] = value/element

எடுத்துக்காட்டு 9.9: Dictionary புதிய மதிப்பை சேர்ப்பதற்கான நிரல்

```
MyDict = { 'Reg_No': '1221',
           'Name' : 'Tamilselvi',
           'School' : 'CGHSS', 'Address' : '
           Rotler St., Chennai 112'}
print(MyDict)
print("Register Number: ", MyDict['Reg_No'])
print("Name of the Student: ", MyDict['Name'])
MyDict['Class'] = 'XII - A'              # Adding new value
print("Class: ", MyDict['Class'])        # Printing newly added value
print("School: ", MyDict['School'])
print("Address: ", MyDict['Address'])
```

Dictionary உள்ள மதிப்பை மாற்றியிடுதல், உறுப்புகளை சேர்ப்பது போன்றதே ஆகும். நீங்கள் திறவுகோலுக்கு ஒரு மதிப்பை இருத்தும் போது, அது பழைய மதிப்பின் மேல் எழுதப்படுகிறது.

பைத்தான் Dictionary, del சிறப்புச் சொல் ஒரு குறிப்பிட்ட உறுப்பை நீக்குவதற்கு பயன்படுகிறது. clear( ) செயற்கூறு Dictionary-ன் அனைத்து உறுப்புகளையும் நீக்குவதற்கு பயன்படுகிறது. Dictionary யை நீக்குவதற்கு del சிறப்புச் சொல்லுடன் Dictionary ன் பெயரையும் பயன்படுத்த வேண்டும்.

தொடரியல்:

```
# குறிப்பிட்ட உறுப்பை நீக்க.
del Dictionary_name[key]

# அனைத்து உறுப்புகளையும் நீக்க
Dictionary_name.clear( )

# முழு Dictionary நீக்க
del Dictionary_name
```

எடுத்துக்காட்டு 9.10: Dictionary ல் இருந்து உறுப்புகளை நீக்குதல் மற்றும் இறுதியாக டிக்ஷ்னரிஐ நீக்குவதற்கான நிரல்.

```
Dict = {'Roll No' : 12001, 'SName' : 'Meena', 'Mark1' : 98, 'Marl2' : 86}
print("Dictionary elements before deletion: \n", Dict)
del Dict['Mark1']                 # குறிப்பிட்ட உறுப்பை நீக்குதல்
print("Dictionary elements after deletion of a element: \n", Dict)
Dict.clear()                      # அனைத்து உறுப்புகளையும் நீக்குதல்
print("Dictionary after deletion of all elements: \n", Dict)
del Dict
print(Dict)                       # முழு Dictionary ஐ நீக்குதல்
```


Chapter 9.indd 168 30-01-2020 19:37:07

வெளியீடு

```
Dictionary elements before deletion:
        {'Roll No': 12001, 'SName': 'Meena', 'Mark1': 98, 'Marl2': 86}
Dictionary elements after deletion of a element:
        {'Roll No': 12001, 'SName': 'Meena', 'Marl2': 86}
Dictionary after deletion of all elements:
        { }
Traceback (most recent call last):
        File "E:/Python/Dict_Test_02.py", line 8, in <module>
        print(Dict)
NameError: name 'Dict' is not defined
```

9.3.4 List மற்றும் Dictionary இடையேயான வேறுபாடு

(1) List என்பது வரிசைப்படுத்திய உறுப்புகளின் தொகுப்பாகும். ஆனால், Dictionary ஒரு உறுப்பை (திறவுகோல்) மற்றொரு உறுப்புடன் (மதிப்பு) பொருத்தப் பயன்படும் தரவு அமைப்பாகும்.

(2) List-ன் சுட்டெண்கள் குறிப்பிட்ட உறுப்பை அணுகுவதற்குப் பயன்படுகின்றன. ஆனால், Dictionary-ல் திறவுகோல் சுட்டெண்ணைக் குறிக்கிறது. ஒரு சரத்தின் எண்ணாகவும், திறவுகோல் இருக்கலாம் என்பதை நினைவில் கொள்க.

(3) List-ன் மதிப்பை பார்த்துக் கொள்ளப்பயன்படுகிறது. Dictionary ஒரு மதிப்பை எடுத்துக் கொண்டு மற்றொரு மதிப்பை பார்த்துக் கொள்ள பயன்படுகிறது.

நினைவில் கொள்க:

- பைத்தான் நிரலாக்க மொழியில் நான்கு வகையான தரவினங்கள் உள்ளன. அவை List, Tuples, Set மற்றும் Dictionary ஆகும்.
- List "வரிசைமுறை தரவினம்" என்றும் அழைக்கப்படுகிறது. List-ல் உள்ள ஒவ்வொரு மதிப்பும் உறுப்பு என்றழைக்கப்படுகிறது.
- List உள்ள உறுப்புகள் சதுர அடைப்புக்குறிக்குள் அடைக்கப்பட்டிருக்க வேண்டும்.
- ஒவ்வொரு உறுப்பும் பூஜ்ஜியத்திலிருந்து தொடங்குகின்ற தனித்தன்மை வாய்ந்த சுட்டெண்ணை கொண்டுள்ளது.
- பைத்தான் நேர்மறை மற்றும் எதிர்மறை எண்களை சுட்டெண்ணாக அனுமதிக்கிறது.
- மடக்குகள், List இருந்து அனைத்து உறுப்புகளையும் அணுகப் பயன்படுகின்றன.
- "for" மடக்கு அனைத்து உறுப்புகளையும் ஒன்றன்பின் ஒன்றாக அணுகுவதற்கு பொருத்தமான மடக்காகும்.
- append( ), extend( ) மற்றும் insert( ) செயற்கூறுகள் List அதிக உறுப்புகளை சேர்க்கப் பயன்படுகின்றன.

Chapter 9.indd 169 30-01-2020 19:37:07

நினைவில் கொள்க:

- del, remove( ) மற்றும் pop( ) List இருந்து உறுப்புகளை நீக்கப்பயன்படுகின்றன.
- Tuples காற்புள்ளியால் பிரிக்கப்பட்ட பல மதிப்புகளை வளைவு அடைப்புக்குறிக்குள் கொண்டுள்ளது.
- Tuples-ன் மடக்குச் செயல் List-ஐ காட்டிலும் விரைவானது.
- Tuples( ) செயற்கூறு List-ல் இருந்து Tuples-ஐ உருவாக்கவும் பயன்படுகிறது.
- ஒரு உறுப்புடன் Tuples-ஐ உருவாக்குவது "ஒற்றை" Tuples என அழைக்கப்படுகிறது.
- Dictionary என்பது பல்வேறு வகையான உறுப்புகளின் தொகுப்பாகும்.

செய்முறைப் பயிற்சி

1. List-ல் இருந்து நகர்த்துவதற்கான நிரலை எழுதுக.
2. Tuples-ல் இருந்து மிகப் பெரிய மதிப்பை அச்சிடும் நிரலை எழுதுக.
3. While மடக்கைப் பயன்படுத்தி Tuples-ல் உள்ள அனைத்து எண்களின் கூட்டுத்தொகை கண்டறியும் நிரலை எழுதுக.
4. List உள்ள அனைத்து இரட்டைப்படை எண்களின் கூட்டுத்தொகை கண்டறியும் நிரலை எழுதுக.
5. மடக்கைப் பயன்படுத்தி List-ஐ தலைகீழாக மாற்றுவதற்கான நிரலை எழுதுக.
6. List ஒரு மதிப்பை குறிப்பிட்ட இடத்தில் செருகுவதற்கான நிரலை எழுதுக.
7. 3 அல்லது 6 ஆல் வகுபடக்கூடிய, 1 முதல் 50 எண்களின் List-ஐ உருவாக்கும் நிரலை எழுதுக.
8. 1 முதல் 20 தொடர் எண்களைக் கொண்ட List-ஐ உருவாக்குவதற்கான நிரலை எழுதுக. பின்னர் 3 ஆல் வகுபடக்கூடிய அனைத்து எண்களையும் லிஸ்ட்ல் இருந்து நீக்குக.
9. List ஒரு மதிப்பு எத்தனை முறை தோன்றுகிறது என்பதை கணக்கிடும் நிரலை எழுதுக. மடக்கை பயன்படுத்தவும்.
10. Dictionary உள்ள அதிகபட்ச மற்றும் குறைந்தபட்ச மதிப்பை அச்சிடும் நிரலை எழுதுக.

மதிப்பீடு

Q1C8TB

பகுதி–அ

சரியான விடையை தேர்ந்தெடுத்து எழுதுக **(1 மதிப்பெண்)**

1. தரவினத் தொகுதியின் தொடர்பில்லாத ஒன்றைத் தேர்வு செய்க

   (a) List (b) Tuple (c) Dictionary (d) Loop

2. Let list1=[2,4,6,8,10], எனில் print(List1[-2]) ன் விடை

   (a) 10 (b) 8 (c) 4 (d) 6

3. பின்வரும் எந்த செயற்கூறு List-ல் உள்ள உறுப்புகளின் எண்ணிக்கையைக் கணக்கிட பயன்படுகிறது?

   (a) count() (b) find() (c)len() (d) index()

4. If List=[10,20,30,40,50] எனில் List[2]=35 ன் விடை

   (a) [35,10,20,30,40,50] (b) [10,20,30,40,50,35]

   (c) [10,20,35,40,50] (d) [10,35,30,40,50]

5. If List=[17,23,41,10] எனில் List.append(32) ன் விடை

   (a) [32,17,23,41,10] (b) [17,23,41,10,32]

   (c) [10,17,23,32,41] (d) [41,32,23,17,10]

6. பின்வரும் எந்த பைத்தான் செயற்கூறு ஏற்கனவே உள்ள List-ல் ஒன்றுக்கும் மேற்பட்ட உறுப்புகளை சேர்க்கப் பயன்படுகிறது?

   (a) append() (b) append_more() (c)extend() (d) more()

7. பின்வரும் பைத்தான் குறிமுறையின் விடை என்ன?

   S=[x**2 for x in range(5)]

   print(S)

   (a) [0,1,2,4,5] (b) [0,1,4,9,16] (c) [0,1,4,9,16,25] (d) [1,4,9,16,25]

8. பைத்தானில் type() செயற்கூறின் பயன் என்ன?

   (a) Tuple உருவாக்க

   (b) Tuple உள்ள உறுப்புகளின் வகையைக் கண்டறிய

   (c) பைத்தான் பொருளின் தரவினத்தை கண்டறிய

   (d) பட்டியலை உருவாக்க

9. பின்வரும் எந்த கூற்று சரியானது அல்ல?

(a) List மாற்றம் செய்யலாம்

(b) Tuples மாற்றம் செய்ய முடியாது.

(c) Append() செயற்கூறு, ஒரு உறுப்பை சேர்க்கப் பயன்படுகிறது.

(d) Extend() செயற்கூறு லிஸ்ட்ல் உறுப்புகளை சேர்க்க Tuples-ல் பயன்படுகிறது

10. SetA={3,6,9}, setB={1,3,9}. எனில், பின்வரும் நிரலின் வெளியீடு என்ன?

```
print(setA|setB)
```

(a) {3,6,9,1,3,9} (b) {3,9} (c) {1} (d) {1,3,6,9}

11. பின்வரும் எந்த set செயல்பாடு, இரண்டு set-களுக்கும் போதுவான உறுப்புகள் நீங்கலாக மற்ற அனைத்து உறுப்புகளையும் உள்ளடக்கியது?

(a) சமச்சீரான வேறுபாடு (b) வேறுபாடு

(c) வெட்டு (d) ஒட்டு

12. பைத்தான், Dictionary-ல் திறவுகோல்கள் எதனால் குறிப்பிடப்படுகின்றன

(a) = (b) ; (c)+ (d) :

பகுதி–ஆ

அனைத்து வினாக்களுக்கும் விடையளி **(2 மதிப்பெண்)**

1. பைத்தானில் List என்றால் என்ன?
2. List உறுப்புகளை பின்னோக்கு வரிசையில் தலைகீழாக எவ்வாறு அணுகுவாய்
3. பின்வரும் பைத்தான் குறிமுறையில் x ன் மதிப்பு என்ன?

```
List1=[2,4,6[1,3,5]]
x=len(List1)
```

4. List-ன் del மற்றும் remove( ) செயற்கூறின் வேறுபாடுகள் யாவை?
5. ஒரு Tuples n எண்ணிக்கை உறுப்புகளுடன் உருவாக்குவதற்கான தொடரியலை எழுதுக.
6. பைத்தானில் set என்றால என்ன?

பகுதி–இ

அனைத்து வினாக்களுக்கும் விடையளி **(3 மதிப்பெண்)**

1. List -ஐ விட மேலான Tuples-ன் நன்மைகளை எழுதுக
2. Sort( ) பற்றி சிறுகுறிப்பு எழுதுக.
3. பின்வரும் குறிமுறையின் வெளியீடு என்ன?

```
list = [2**x for x in range(5)]
print(list)
```

4. del மற்றும் clear( ) செயற்கூறுகளுக்கு இடையேயான வேறுபாடுகளை எடுத்துக்காட்டுடன் விளக்குக.
5. பைத்தானின் set செயல்பாடுகளை பட்டியலிடுக.
6. List மற்றும் Dictionary இடையேயான வேறுபாடுகள் யாவை?

பகுதி–ஈ

அனைத்து வினாக்களுக்கும் விடையளி **(5 மதிப்பெண்)**

1. List-ல் ஒரு உறுப்பை சேர்ப்பதற்கான பல்வேறு வழிகள் யாவை? பொருத்தமான எடுத்துக்காட்டுடன் விளக்குக.
2. range( )ன் நோக்கம் என்ன? எடுத்துக்காட்டுடன் விளக்குக.
3. பின்னலான Tuple என்றால் என்ன? எடுத்துக்காட்டுடன் விளக்குக.
4. பைத்தானிலுள்ள பல்வேறு set செயல்பாடுகளை பொருத்தமான எடுத்துக்காட்டுகளுடன் விளக்குக.

References

1. https://docs.python.org/3/tutorial/index.html
2. https://www.techbeamers.com/python-tutorial-step-by-step/#tutorial-list
3. Python programming using problem solving approach – Reema Thareja – Oxford University press.
4. Python Crash Course – Eric Matthes – No starch press, San Francisco.

அலகு III

பாடம் 10

பைத்தான் இனக்குழுக்கள் மற்றும் பொருள்கள்

4LSLLU

கற்றலின் நோக்கங்கள்

இந்த பாடப்பகுதியைக் கற்றபின், மாணவர்கள்

- பொருள் நோக்கு நிரலாக்கத்தின் அடிப்படை கருத்துருக்களாகிய இனக்குழு, பொருள் ஆக்கி மற்றும் அழிப்பி ஆகியவற்றைப் புரிந்து கொள்ளுதல்.
- பைத்தானில் இனக்குழுக்களையும், பொருள்களையும் உருவாக்குவதைப் பற்றிய அறிவைப் பெறுதல்.
- ஆக்கிகளுடன் கூடிய இனக்குழுக்களை உருவாக்குதல்.
- இனக்குழுவைப் பயன்படுத்தி **பைத்தானில்** சிக்கலான நிரல்களை எழுதுதல்.

10.1 அறிமுகம்

பைத்தான் ஒரு பொருள் நோக்கு நிரலாக்க மொழி (Object Oriented Programming Language) ஆகும். இனக்குழுக்களும் (Class), பொருள்களும் (Object) பொருள் நோக்கு நிரலாக்கத்தின் முக்கிய அம்சமாகும். இதன் இனக்குழு மற்றும் பொருள்களின் கோட்பாடுகள், C++ கோட்பாடுகளை ஒத்திருக்கும். ஆனால் C++ உடன் ஒப்பிடும் போது, பைத்தானில் இனக்குழு மற்றும் பொருள்களை உருவாக்குவதும், செயல்படுத்துவதும் மிக எளிது.

பைத்தானில் மிக முக்கிய கட்டமைப்பு கூறுகளாகத் திகழ்பவை இனக்குழுவாகும். பொருள் என்பது தரவுகளின் மீது செயல்படும் செயற்கூறு மற்றும் தரவுகளின் தொகுப்பாகும். இனக்குழு என்பது பொருளின் வார்ப்புரு (template) ஆகும்.

பொருள்நோக்கு நிரலாக்கத்தின்படி பொருள்கள் இனக்குழுவின் சான்றுரு (instance) (அல்லது) இனக்குழு மாறி (class variables) என்று அழைக்கப்படும். பைத்தானில் அனைத்துமே பொருள்களாகும். எடுத்துக்காட்டாக, நிரலில் பயன்படும் அனைத்து முழு எண் மாறிகளும் “int” இனக்குழுவின் பொருளாகும். அதே போல அனைத்து சரங்களின் மாறிகளும் string இனக்குழுவின் பொருளாகும்.

10.2 இனக்குழுவை வரையறுத்தல்

பைத்தானில், இனக்குழுவை வரையறுக்க “class” என்னும் சிறப்புச் சொல் பயன்படுத்தப்படுகிறது. ஒவ்வொரு இனக்குழுவும் தனித்த பெயருடன் முக்காற்புள்ளி (:) யைக் கொண்டு முடியும்.

Chapter 10.indd 174 30-01-2020 19:40:01

தொடரியல்

```
class class_name:
    statement_1
    statement_2
    …………..
    …………..
    statement_n
```

இனக்குழு வரையறுப்பில், கூற்று (statement) என்பது மாறி அறிவிப்பாகவோ, தேர்ந்தெடுப்பு கூற்றாகவோ, மடக்காகவோ அல்லது செயற்கூறு வரையறையாகவோ இருக்கலாம். இனக்குழுக்குள்ளே வரையறுக்கப்படும் மாறிகள் இனக்குழு மாறிகள் (Class variables) என்றும் செயற்கூறுகள் வழிமுறைகள் (methods) என்றும் அழைக்கப்படும். இனக்குழு மாறிகள் மற்றும் வழிமுறைகள் சேர்ந்து இனக்குழுவின் உறுப்புகள் (members) எனப்படும். இனக்குழுவின் உறுப்புகளை இனக்குழுவின் பொருள்கள் அல்லது சான்றுருக்கள் மூலமாகவே அணுகுதல் வேண்டும். ஒரு இனக்குழுவை பைத்தான் நிரலில் எங்கு வேண்டுமானாலும் வரையறுக்கலாம்.

எடுத்துக்காட்டு: இனக்குழுவை வரையறுக்கும் நிரல்

```
class Sample:
    x, y = 10, 20        # class variables
```

மேலே குறிப்பிட்டுள்ள இனக்குழு வரையறையில், “sample” என்பது இனக்குழுவின் பெயர். இதில் x மற்றும் y என இரு மாறிகள் உள்ளன. இதன் தொடக்க மதிப்பு முறையே 10 மற்றும் 20 ஆகும். இனக்குழுவின் உள்ளே வரையறுக்கப்பட்டுள்ள மதிப்புகளை அணுகுவதற்கு இனக்குழுவின் பொருள் அல்லது சான்றுரு தேவை.

10.3 பொருள்களை உருவாக்குதல்

இனக்குழு உருவாக்கப்பட்ட பின் அந்த இனக்குழுவின் பொருள் அல்லது சான்றுரு உருவாக்கப்பட வேண்டும். பொருளை உருவாக்கும் இந்த முறைக்கு “சான்றுருவாக்கல்” (instantiation) என்று பெயர்.

தொடரியல்

```
Object_name = class_name( )
```

WTPD8W

குறிப்பு- இனக்குழு சான்றுருவாக்கல், செயற்கூறு குறியீட்டைப் பயன்படுத்துகிறது. அதாவது, இனக்குழு பெயரானது அடைப்புக்குறியைக் கொண்டிருக்கும்.

10.4 இனக்குழு உறுப்புகளை அணுகுதல்

இனக்குழு உறுப்புகளை (அதாவது இனக்குழு மாறி அல்லது வழிமுறை (செயற்கூறு)) புள்ளி செயற்குறி மூலம் அணுக முடியும்.

தொடரியல்

```
Object_name . class_member
```


Chapter 10.indd 175 30-01-2020 19:40:01

எடுத்துக்காட்டு 10.1 : இனக்குழுவை வரையறுத்து அதன் உறுப்பு மாறிகளை அணுகும் நிரல்

```
class Sample:
#class variables
        x, y = 10, 20
S=Sample( )              # class instantiation
print("Value of x = ", S.x)
print("Value of y = ", S.y)
print("Value of x and y = ", S.x+S.y)
```

வெளியீடு

```
Value of x =  10
Value of y =  20
Value of x and y =  30
```

மேலே உள்ள நிரலில் இனக்குழுவின் பெயர் sample இனக்குழுவிற்குள்ளே, x மற்றும் y மாறிகள் முறையே 10 மற்றும் 20 என்ற தொடக்க மதிப்புகளை இருத்துகிறது. இந்த இரண்டு மாறிகளும் இனக்குழு மாறிகள் அல்லது இனக்குழுவின் உறுப்பு மாறிகள் (member variable) என்றும் அழைக்கப்படுகிறது. இந்த இனக்குழு சான்றுருவாக்கல் முறையில் இனக்குழுவின் உறுப்புகளை அணுகுவதற்கு S என்ற பொருள் உருவாக்கப்பட்டுள்ளது. முதல் இரண்டு print கூற்றுகள் இனக்குழுவின் மாறிகளான x மற்றும் y ன் மதிப்புகளை அச்சிடுகிறது. கடைசி print கூற்று இரண்டு இனக்குழு மாறிகளான x மற்றும் y மதிப்புகளின் கூட்டுத் தொகையை அச்சிடுகிறது.

10.5 இனக்குழு வழிமுறைகள்

பைத்தானில் இனக்குழு செயற்கூறு அல்லது வழிமுறை சாதாரண செயற்கூறுவைப்போல் ஒத்திருந்தாலும், ஒரு சிறிய வேறுபாடு உள்ளது. இனக்குழு வழிமுறையின் முதல் அளபுரு கண்டிப்பாக self என இருக்க வேண்டும். வழிமுறைகளை அழைக்கும் போது இந்த அளபுருவுக்கு மதிப்பை அனுப்ப தேவையில்லை. பைத்தான் தானாகவே இதற்கு மதிப்பை வழங்கும். வழிமுறையில் அளபுருக்களை எடுத்துக் கொள்ளாத போதும், முதல் அளபுருவாக self வரையறுக்கப்பட வேண்டும். வழிமுறை ஒரே ஒரு அளபுருவை ஏற்க வரையறுக்கப்பட்டால் அது இரண்டு அளபுருவாக எடுத்துக் கொள்ளும். அதாவது, self மற்றும் வரையறுக்கப்பட்ட அளபுரு.

இனக்குழுவிற்குள் மாறிகளை அறிவிக்கும் பொழுது, வழிமுறைகள் முன்னொட்டாக இனக்குழுவின் பெயர் மற்றும் புள்ளி(.) செயற்குறியைக் கொண்டு இருக்க வேண்டும்.

குறிப்பு

இனக்குழுவின் உள்ளே வரையறுக்கப்படும் கூற்றுகள் முறையாக உள்தள்ளப்பட்டிருக்க வேண்டும்.

Chapter 10.indd 176 30-01-2020 19:40:01

எடுத்துக்காட்டு 10.2 : இனக்குழுவைப் பயன்படுத்தி மொத்தம் மற்றும் சராசரி மதிப்பெண்ணைக் கணக்கிடும் நிரல்.

```
class Student:
    mark1, mark2, mark3 = 45, 91, 71          #class variable
    def process(self):                        #class method
        sum = Student.mark1 + Student.mark2 + Student.mark3
        avg = sum/3
        print("Total Marks = ", sum)
        print("Average Marks = ", avg)
        return
S=Student()
S.process()
```

மேலே உள்ள நிரலில் இனக்குழு வரையறுக்கப்பட்டபின் S என்ற பொருள் உருவாக்கப்பட்டுள்ளது. S.process( ) என்ற கூற்று தேவையான வெளியீட்டைப் பெறுவதற்காக செயற்கூறுவை அழைக்கிறது.

mark1, mark2 மற்றும் mark3 ஆகிய மூன்று மாறிகளின் மதிப்பு முறையே 45, 91, 71 என அறிவிக்கப்பட்டுள்ளது. process() என்னும் வழிமுறை self அளபுருவுடன் வரையறுக்கப்பட்டுள்ளது. அதாவது, எந்த மதிப்பையும் அனுப்பப்போவதில்லை. process வழிமுறை முதலில் இனக்குழு மாறிகளின் மதிப்பை கூட்டுகிறது. Sum என்ற மாறியில் விடையைச் சேமிக்கிறது. சராசரியைக் கணக்கிட்டு காண்பிக்கிறது.

மேலே உள்ள நிரலின் படி, பின்வரும் வெளியீடு தோன்றும்.

வெளியீடு

```
Total Marks =  207
Average Marks =  69.0
```

எடுத்துக்காட்டு 10.3: இனக்குழுவைப் பயன்படுத்தி கொடுக்கப்பட்ட எண் ஒற்றைப்படை அல்லது இரட்டைப்படை எண்ணா என்று சோதித்து அச்சிடும் நிரல்

```
class Odd_Even:
    even = 0                    #class varaiable
def check(self, num):
    if num%2==0:
        print(num," is Even number")
    else:
        print(num," is Odd number")
n=Odd_Even()
x = int(input("Enter a value: "))
n.check(x)
```

இந்த நிரலை இயக்கும் போது பைத்தான், பயனர் கொடுக்கும் மதிப்பை உள்ளீடாகப் பெற்று, அதை வழிமுறைக்கு அனுப்பி, பொருள் மூலம் சரிபார்க்கிறது.

வெளியீடு 1

```
Enter a value: 4
4  is Even number
```

வெளியீடு 2

```
Enter a value: 5
5  is Odd number
```


Chapter 10.indd 177 30-01-2020 19:40:01

10.6 பைத்தானில் ஆக்கிகள் மற்றும் அழிப்பிகள்

ஓர் இனக்குழுவின் சான்றுரு பயன்பாட்டிற்கு வரும்பொழுது ஆக்கி என்னும் சிறப்புச் செயற்கூறு தானாகவே இயக்கப்படுகிறது. பைத்தானில், “init” என்னும் சிறப்பு செயற்கூறு ஆக்கியாக செயல்படுகிறது. இது இரட்டை அடிக்கீறில் (Under Score) தொடங்கி இரட்டை அடிக்கீறலுடன் முடிய வேண்டும்.

இந்த செயற்கூறு சாதாரண செயற்கூறுவைப் போல செயல்படும். ஆனால் பொருள் உருவாக்கப்பட்டவுடன் தானாகவே இயக்கப்படும். இந்த ஆக்கி செயற்கூறை அளபுருக்களுடனோ அல்லது இல்லாமலோ வரையறுக்கலாம்.

***__init__* வழிமுறையின் (ஆக்கி) பொதுவடிவம்**

def __init__(self, [args ……..]):

<statements>

எடுத்துக்காட்டு 10. 4: ஆக்கியை விளக்கும் நிரல்

```
class Sample:
	def __init__(self, num):
		print("Constructor of class Sample...")
		self.num=num
		print("The value is :", num)
S=Sample(10)
```

மேலே உள்ள இனக்குழு “Sample”, “num” என்னும் ஒரே ஒரு அளபுருவுடன் கூடிய ஆக்கியைக் கொண்டுள்ளது. ஆக்கி இயக்கப்படும் போது முதல் “print” கூற்று, “Constructor of class Sample….” என்பதை அச்சிடும். பிறகு, ஆக்கிக்கு அனுப்பப்படும் மதிப்பு self.num ல் இருத்தப்படுகிறது. இறுதியாக, கொடுக்கப்பட்ட சரத்துடன் அனுப்பட்ட மதிப்பையும் அச்சிடுகிறது. S என்ற பொருள் 10 என்ற மெய்யான அளபுருவுடன் உருவாக்கப்படும் போது மேலே உள்ள ஆக்கி தானாகவே இயக்கப்படும். பின்வரும் வெளியீடு கிடைக்கிறது.

Constructor of class Sample...

The value is : 10

ஆக்கியின் உள்ளே வரையறுக்கப்படும் இனக்குழு மாறியானது, இனக்குழுவில் உருவாக்கப்படும் பொருள்களின் எண்ணிக்கையைக் கொண்டிருக்கும்:

Chapter 10.indd 178 30-01-2020 19:40:01

எடுத்துக்காட்டு 10.5 : உருவாக்கப்பட்ட பொருள்களின் எண்ணிக்கையை கொண்டிருக்கும் இனக்குழு மாறியை விளக்கும் நிரல்

```
class Sample:
    num=0
    def __init__(self, var):
        Sample.num+=1
        self.var=var
        print("The object value is = ", var)
        print("The count of object created = ", Sample.num)

S1=Sample(15)
S2=Sample(35)
S3=Sample(45)
```

மேலே உள்ள நிரலில், num என்னும் இனக்குழு மாறியானது Sample என்னும் இனக்குழுவின் மூன்று பொருள்களுடன் பகிரப்படுகிறது. இதில் 0 என்ற தொடக்க மதிப்பு இருத்தப்பட்டு, ஒவ்வொரு முறையும் பொருள் உருவாக்கப்படும் போது num ன் மதிப்பு 1 அதிகரிக்கப்படுகிறது. அனைத்துப் பொருள்களும் மாறியை பகிர்வதால் ஒரு பொருளில் num மாறியில் செய்யப்படும் மாற்றமானது மற்ற பிற பொருள்களில் பிரதிபலிப்பதால், பின்வரும் வெளியீடு கிடைக்கும்.

வெளியீடு

```
The object value is          =   15
The count of object created  =   1
The object value is          =   35
The count of object created  =   2
The object value is          =   45
The count of object created  =   3
```

குறிப்பு: இனக்குழு மாறியானது *C++*ல் உள்ள *Static* வகையை ஒத்ததாகும்.

இனக்குழுவில் உருவாக்கப்பட்ட பொருளின் பயன்பாடு முடிவுக்கு வரும் போது அழிப்பி என்னும் சிறப்பு செயற்கூறு தானாகவே இயக்கப்படும். இது ஆக்கிக்கு முரணானது. பைத்தானில், __del__( ) செயற்கூறு அழிப்பியாகப் பயன்படுத்தப்படுகிறது.

எடுத்துக்காட்டு 10.6 : __del__( ) வழிமுறையை விளக்கும் நிரல்

```
class Sample:
        num=0
        def __init__(self, var):
                Sample.num+=1
                self.var=var
                print("The object value is = ", var)
                print("The value of class variable is= ", Sample.num)
        def __del__(self):
                Sample.num-=1
                print("Object with value %d is exit from the scope"%self.var)
S1=Sample(15)
S2=Sample(35)
S3=Sample(45)
```

10.7 Public மற்றும் Private தரவு உறுப்புகள்

இனக்குழுவின் உள்ளே வரையறுக்கப்படும் மாறி கொடாநிலையாக public மாறி ஆகும். இந்த மாறிகளை புள்ளி செயற்குறி(.) பயன்படுத்தி நிரலில் எங்கு வேண்டுமானாலும் அணுகலாம். இரட்டை அடிக்கீறலை முன்னொட்டாக கொண்ட மாறிகள் private ஆகும். இந்த மாறிகளை இனக்குழுவின் உள்ளே மட்டும் தான் அணுக முடியும்.

எடுத்துக்காட்டு 10.7 : private மற்றும் public மாறிகளை விளக்கும் நிரல்

```
class Sample:
        def __init__(self, n1, n2):
                self.n1=n1
                self.__n2=n2
        def display(self):
                print("Class variable 1 = ", self.n1)
                print("Class variable 2 = ", self.__n2)
S=Sample(12, 14)
S.display()
print("Value 1 = ", S.n1)
print("Value 2 = ", S.__n2)
```

மேலே உள்ள நிரலில் n1 மற்றும் n2 என இரண்டு இனக்குழு மாறிகள் அறிவிக்கப்பட்டுள்ளது. n1 என்பது public மாறி n2 என்பது Private மாறி ஆகும். Display ( ) உறுப்பு செயற்கூறு இந்த இரண்டு மாறிகளுக்கும் அனுப்பப்பட்ட மதிப்புகளைக் காண்பிப்பதற்காக வரையறுக்கப்பட்டுள்ளது.

இனக்குழு மாறி n2 private ஆக இருந்தபோதிலும் இனக்குழுவின் உள்ளே உள்ள print கூற்றுகள் வெற்றிகரமாக n1 மற்றும் n2 மாறிகளின் மதிப்புகளைக் காண்பிக்கிறது. ஏனென்றால், இந்த நிரலில் இனக்குழுவின் உள்ளே வரையறுக்கப்பட்ட வழிமுறையின் மூலம் n2 அழைக்கப்பட்டுள்ளது. ஆனால், இனக்குழுவின் வெளியில் இருந்து n2 வின் மதிப்பை அணுக முயற்சித்தால் பிழை செய்தியைக் காண்பிக்கும். ஏனென்றால் private மாறிகள் இனக்குழுவின் வெளியில் இருந்து அணுக முடியாது.

Chapter 10.indd 180 30-01-2020 19:40:01

வெளியீடு

```
Class variable 1 =  12
Class variable 2 =  14
Value 1 =  12

Traceback (most recent call last):
    File "D:/Python/Class-Test-04.py", line 12, in <module>
        print("Value 2 = ", S.__n2)
AttributeError: 'Sample' object has no attribute '__n2'
```

10.8 இனக்குழு மற்றும் பொருள் ஆகியவற்றை விளக்கும் மாதிரி நிரல்கள்

நிரல் 1: வட்டத்தின் பரப்பையும், சுற்றளவையும் கணக்கிடும் பைத்தான் நிரலை எழுதுக.

```
class Circle:
    pi=3.14
    def __init__(self,radius):
        self.radius=radius
    def area(self):
        return Circle.pi*(self.radius**2)
    def circumference(self):
        return 2*Circle.pi*self.radius
r=int(input("Enter Radius: "))
C=Circle(r)
print("The Area =",C.area())
print("The Circumference =", C.circumference())
```

வெளியீடு

```
Enter Radius: 5
The Area = 78.5
The Circumference = 31.400000000000002
```

நிரல் 2: உன்னுடைய பள்ளி நூலகத்தில் உள்ள புத்தகங்களை பதிவு செய்து, புத்தகங்களின் பட்டியலை காட்டும் பைத்தான் நிரலை எழுதுக

```
class Library:
    def __init__(self):
        self.bookname=""
        self.author=""

    def getdata(self):
        self.bookname = input("Enter Name of the Book: ")
        self.author = input("Enter Author of the Book: ")

    def display(self):
        print("Name of the Book: ",self.bookname)
        print("Author of the Book: ",self.author)
        print("\n")

book=[] #empty list
ch = 'y'
while(ch=='y'):
    print("1. Add New Book \n 2.Display Books")
    resp = int(input("Enter your choice : "))
if(resp==1):
    L=Library()
    L.getdata()
    book.append(L)
elif(resp==2):
    for x in book:
        x.display()
else:
    print("Invalid input....")
ch = input("Do you want continue....")
```

வெளியீடு:

```
1. Add New Book
2.Display Books
Enter your choice : 1
Enter Name of the Book: Programming in C++
Enter Author of the Book: K. Kannan
Do you want continue....y
1. Add New Book
2.Display Books
Enter your choice : 1
Enter Name of the Book: Learn Python
Enter Author of the Book: V.G.Ramakrishnan
Do you want continue....y
```


Chapter 10.indd 182 30-01-2020 19:40:01

```
1. Add New Book
 2.Display Books
Enter your choice : 1
Enter Name of the Book: Advanced Python
Enter Author of the Book: Dr. Vidhya
Do you want continue....y
1. Add New Book
 2.Display Books
Enter your choice : 1
Enter Name of the Book: Working with OpenOffice
Enter Author of the Book: N.V.Gowrisankar
Do you want continue....y
1. Add New Book
 2.Display Books
Enter your choice : 1
Enter Name of the Book: Data Structure
Enter Author of the Book: K.Lenin
Do you want continue....y
1. Add New Book
 2.Display Books
Enter your choice : 1
Enter Name of the Book: An Introduction to Database System
Enter Author of the Book: R.Sreenivasan
Do you want continue....y
1. Add New Book
 2.Display Books
Enter your choice : 2
Name of the Book:  Programming in C++
Author of the Book:  K. Kannan

Name of the Book:  Learn Python
Author of the Book:  V.G.Ramakrishnan

Name of the Book:  Advanced Python
Author of the Book:  Dr. Vidhya

Name of the Book:  Working with OpenOffice
Author of the Book:  N.V.Gowrisankar

Name of the Book:  Data Structure
Author of the Book:  K.Lenin

Name of the Book:  An Introduction to Database System
Author of the Book:  R.Sreenivasan

Do you want continue....n
```

நிரல் 3: சரத்தை உள்ளீடாகப் பெற்று, அதில் எத்தனை பெரிய எழுத்துக்கள், சிறிய எழுத்துக்கள், உயிரெழுத்துகள், மெய்யெழுத்துகள் மற்றும் இடைவெளிகள் உள்ளன என்று கண்டறியும் பைத்தான் நிரலை எழுதுக.

```
class String:
    def __init__(self):
        self.uppercase=0
        self.lowercase=0
        self.vowels=0
        self.consonants=0
        self.spaces=0
        self.string=""
    def getstr(self):
        self.string=str(input("Enter a String: "))
    def count_upper(self):
            for ch in self.string:
                if (ch.isupper()):
                    self.uppercase+=1
    def count_lower(self):
            for ch in self.string:
                if (ch.islower()):
                    self.lowercase+=1
    def count_vowels(self):
        for ch in self.string:
            if (ch in ('A', 'a', 'e', 'E', 'i', 'I', 'o', 'O', 'u','U')):
                self.vowels+=1
    def count_consonants(self):
        for ch in self.string:
            if (ch not in ('A', 'a', 'e', 'E', 'i', 'I', 'o', 'O', 'u','U')):
                self.consonants+=1
    def count_space(self):
        for ch in self.string:
            if (ch==" "):
                self.spaces+=1
    def execute(self):
        self.count_upper()
        self.count_lower()
        self.count_vowels()
        self.count_consonants()
        self.count_space()
```


Chapter 10.indd 184 30-01-2020 19:40:01

```
        def display(self):
            print("The given string contains...")
            print("%d Uppercase letters"%self.uppercase)
            print("%d Lowercase letters"%self.lowercase)
            print("%d Vowels"%self.vowels)
            print("%d Consonants"%self.consonants)
            print("%d Spaces"%self.spaces)
S = String()
S.getstr()
S.execute()
S.display()
```

வெளியீடு

```
        Enter a String:Welcome To Learn Computer Science
        The given string contains...
        5 Uppercase letters
        24 Lowercase letters
        12 Vowels
        17 Consonants
        3 Spaces
```

நிரல் 4: பொருட்களையும், அதன் அடக்க விலையையும் சேமிக்கவும், சேமிக்கப்பட்ட பொருட்களை திரையில் தோன்றவும் செய்ய வேண்டும். அப்போது அனைத்து பொருட்களையும் அளவுகளை உள்ளீடாக பெற்று இறுதியில் விலைப்பட்டியலை அச்சிடும் பைத்தான் நிரல் ஒன்றை எழுதுக.

```
class MyStore:
        __prod_code=[]
        __prod_name=[]
        __cost_price=[]
        __prod_quant=[]
        def getdata(self):
                self.p = int(input("Enter no. of products you need to store: "))
                for x in range(self.p):
                        self.__prod_code.append(int(input("Enter Product Code: ")))
                        self.__prod_name.append(str(input("Enter Product Name: ")))
                        self.__cost_price.append(int(input("Enter Cost price: ")))
```


Chapter 10.indd 185 30-01-2020 19:40:01

```
def display(self):
        print("Stock in Stores")
        print("----------------------------------------------------------")
        print("Product Code \t Product Name \t Cost Price")
        print("----------------------------------------------------------")
        for x in range(self.p):
                print(self.__prod_code[x], "\t\t", self.__prod_name[x], "\t\t", self.__
                                                        cost_price[x])
        print("----------------------------------------------------------")

def print_bill(self):
        total_price = 0
        for x in range(self.p):
                q=int(input("Enter the quantify for the product code %d : "%self.__
                                                        prod_code[x]))
self.__prod_quant.append(q)
total_price = total_price +self.__cost_price[x]*self.__prod_quant[x]
        print("            Invoice Receipt                ")
        print("-----------------------------------------------------------------------------")
        print("Product Code\t Product Name\t Cost Price\t Quantity \t Total
                                                                Amount")
        print("-----------------------------------------------------------------------------")
        for x in range(self.p):
                print(self.__prod_code[x], "\t\t", self.__prod_name[x], "\t\t",
                self.__cost_price[x], "\t\t", self.__prod_quant[x], "\t\t",
                self.__prod_quant[x]*self.__cost_price[x])
        print("-----------------------------------------------------------------------------")
        print("                                        Total Amount = ", total_price)
S=MyStore()
S.getdata()
S.display()
S.print_bill()
```


Chapter 10.indd 186 30-01-2020 19:40:01

வெளியீடு:

```
Enter no. of products you need to store: 5
Enter Product Code: 101
Enter Product Name: Product-A
Enter Cost price: 25
Enter Product Code: 201
Enter Product Name: Product-B
Enter Cost price: 35
Enter Product Code: 301
Enter Product Name: Product-C
Enter Cost price: 35
Enter Product Code: 401
Enter Product Name: Product-D
Enter Cost price: 50
Enter Product Code: 501
Enter Product Name: Product-E
Enter Cost price: 120

                    Stock in Stores
----------------------------------------------------------------------
Product Code        Product Name              Cost Price
----------------------------------------------------------------------
101                     Product-A              25
201                     Product-B              35
301                     Product-C              35
401                     Product-D              50
501                     Product-E              120
----------------------------------------------------------------------
```


Chapter 10.indd 187 30-01-2020 19:40:01

```
Enter the quantify for the product code 101 : 10
Enter the quantify for the product code 201 : 15
Enter the quantify for the product code 301 : 10
Enter the quantify for the product code 401 : 20
Enter the quantify for the product code 501 : 10

                         Invoice Receipt
--------------------------------------------------------------------------------
  Product Code  Product Name       Cost Price    Quantity   Total Amount
--------------------------------------------------------------------------------
  101             Product-A            25           10            250
  201             Product-B            35           15            525
  301             Product-C            35           10            350
  401             Product-D            50           20           1000
  501             Product-E           120           10           1200
--------------------------------------------------------------------------------
                                               Total Amount = 3325
--------------------------------------------------------------------------------
```

நினைவில் கொள்க

- பைத்தான் ஒரு பொருள் நோக்கு நிரலாக்க மொழி ஆகும்.
- இனக்குழுக்களும், பொருள்களும் பொருள் நோக்கு நிரலாக்கத்தின் முக்கிய அம்சங்களாகும்.
- பைத்தானில், class என்னும் சிறப்புச் சொல்லைப் பயன்படுத்தி இனக்குழு வரையறுக்கப்பட்டுள்ளது.
- இனக்குழுக்குள் வரையறுக்கப்படும் மாறிகள், இனக்குழு மாறிகள் எனவும் செயற்கூறுகள் வழிமுறைகள் எனவும் அழைக்கப்படுகிறது.
- ஒரு பொருளை உருவாக்கும் செயல்முறையே "சான்றுருவாக்கல்" எனப்படும்.
- ஒரு இனக்குழுவின் சான்றுரு பயன்பாட்டிற்கு வரும் பொழுது ஆக்கி என்னும் சிறப்புச் செயற்கூறு, தானாகவே இயக்கப்படுகிறது.
- இனக்குழுவில் உருவாக்கப்பட்ட பொருளின் பயன்பாடு முடிவுக்கு வரும் போது அழிப்பி என்னும் சிறப்பு செயற்கூறு தானாகவே இயக்கப்படும்.
- பைத்தானில், __del__( ) செயற்கூறு அழிப்பியாகப் பயன்படுத்தப்படுகிறது.
- இரட்டை அடிக்கீறினை முன்னொட்டாக கொண்ட மாறிகள் private மாறிகள் ஆகும்.

Chapter 10.indd 188 30-01-2020 19:40:01

செய்முறைப் பயிற்சி

1. இனக்குழுவைப் பயன்படுத்தி மாணவர்களின் பெயர் மற்றும் மதிப்பெண்களைப் பட்டியலாக சேமித்து மொத்த மதிப்பெண்ணை அச்சிடும் பைத்தான் நிரலை எழுதுக.

2. இனக்குழுவைப் பயன்படுத்தி முக்கோணத்தின் மூன்று பக்கங்களின் அளவை உள்ளீடாகப் பெற்று பரப்பளவைக் கண்டறியும் பைத்தான் நிரலை எழுதுக.

3. இரண்டு தூரங்களின் அளவை படிக்க, காண்பிக்க, கூட்ட மற்றும் கழிக்க உதவும் பட்டியல் முறை (Menu driven) நிரலை எழுதுக.

மதிப்பீடு

81GNU5

பகுதி–அ

சரியான விடையை தேர்ந்தெடுத்து எழுதுக (1 மதிப்பெண்)

1. பின்வருவனவற்றுள் எவை பொருள் நோக்கு நிரலாக்கத்தின் முக்கிய அம்சம் ஆகும்?

   (அ) ஆக்கி மற்றும் இனக்குழு (ஆ) ஆக்கி மற்றும் பொருள்

   (இ) இனக்குழு மற்றும் பொருள் (ஈ) ஆக்கி மற்றும் அழிப்பி

2. இனக்குழுவின் உள்ளே வரையறுக்கப்படும் செயற்கூறு எது:

   (அ) செயற்கூறு (ஆ) கூறு

   (இ) வழிமுறை (ஈ) பிரிவு

3. இனக்குழு உறுப்புகள் எந்த செயற்குறியின் மூலம் அணுகப்படுகிறது?

   (அ) & (ஆ) .

   (இ) # (ஈ) %

4. பொருள் உருவாக்கப்படும் போது தானாகவே இயக்கப்படும் செயற்கூறு எது?

   (அ) __object__( ) (ஆ) __del__( )

   (இ) __func__( ) (ஈ) __init__( )

5. private இனக்குழு மாறியின் முன்னொட்டு எது

   (அ) __ (ஆ) &&

   (இ) ## (ஈ) **

6. பின்வரும் வழிமுறைகளில் எது அழிப்பியாகப் பயன்படுகிறது?

   (அ) __init__( ) (ஆ) __dest__( )

   (இ) __rem__( ) (ஈ) __del__( )

Chapter 10.indd 189 30-01-2020 19:40:02

7. பின்வருவனவற்றுள் எந்த இனக்குழு அறிவிப்பு சரியானது?

(அ) class class_name (ஆ) class class_name<>

(இ) class class_name: (ஈ) class class_name[ ]

8. பின்வரும் நிரலின் வெளியீடு என்ன?

```
class Student:
        def __init__(self, name):
                self.name=name
                print(self.name)
S=Student("Tamil")
```

(அ) Error (ஆ) Tamil

(இ) name (ஈ) self

9. பின்வருவனவற்றுள் எது private இனக்குழு மாறி?

(அ) __num (ஆ) ##num

(இ) $$num (ஈ) &&num

10. பொருளை உருவாக்கும் செயல்முறை எது:

(அ) ஆக்கி (ஆ) அழிப்பு

(இ) மதிப்பிருத்தல் (ஈ) சான்றுருவாக்கல்

பகுதி–ஆ

அனைத்து வினாக்களுக்கும் விடையளி **(2 மதிப்பெண்)**

1. இனக்குழு என்றால் என்ன?
2. சான்றுருவாக்கல் என்றால் என்ன?
3. பின்வரும் நிரலின் வெளியீடு என்ன?

```
class Sample:
        __num=10
        def disp(self):
                print(self.__num)
S=Sample()
S.disp()
print(S.__num)
```

4. பைத்தானில் ஆக்கியை எவ்வாறு உருவாக்குவாய்?
5. அழிப்பியின் நோக்கம் என்ன?

பகுதி–இ

அனைத்து வினாக்களுக்கும் விடையளி **(3 மதிப்பெண்)**

1. இனக்குழு உறுப்புகள் என்றால் என்ன? அதனை எவ்வாறு வரையறுப்பாய்?
2. இரண்டு private இனக்குழு மாறிகளுடன், வழிமுறையை பயன்படுத்தி கூட்டுத் தொகை sum அச்சிடும் இனக்குழுவை வரையறுக்கவும்.
3. கொடுக்கப்பட்ட வெளியீட்டை பெற பின்வரும் நிரலில் உள்ள பிழைகளை காண்க.

```
class Fruits:
    def __init__(self, f1, f2):
        self.f1=f1
        self.f2=f2
    def display(self):
        print("Fruit 1 = %s, Fruit 2 = %s" %(self.f1, self.f2))
F = Fruits ('Apple', 'Mango')
del F.display
F.display()
```

வெளியீடு

```
    Fruit 1 = Apple,   Fruit 2 = Mango
```

4. பின்வரும் நிரலின் வெளியீடு என்ன?

```
class Greeting:
    def __init__(self, name):
        self.__name = name
    def display(self):
        print("Good Morning ", self.__name)
obj=Greeting('Bindu Madhavan')
obj.display()
```

5. பைத்தானில் ஆக்கி மற்றும் அழிப்பிகளை எவ்வாறு வரையறுப்பாய்?

பகுதி–ஈ

அனைத்து வினாக்களுக்கும் விடையளி **(5 மதிப்பெண்)**

1. எழுது பொருட்களை (stationary) சேர்க்க அல்லது நீக்கும் பட்டியல் முறை நிரல் ஒன்றை எழுதுக. பொருட்களின் பெயர் மற்றும் பிராண்ட்-யை ஒரு dictionary -யில் சேமிக்க வேண்டும்.

References

1. *https://docs.python.org/3/tutorial/index.html*
2. *https://www.techbeamers.com/python-tutorial-step-by-step/#tutorial-list*
3. *Python programming using problem solving approach – Reema Thareja – Oxford University press.*
4. *Python Crash Course – Eric Matthes – No starch press, San Francisco.*

அலகு IV

பாடம் 11

தரவுதள கருத்துருக்கள்

I88ZFD

கற்றலின் நோக்கங்கள்

இந்த பாடப்பகுதியை கற்றபின் மாணவர்கள் பின்வருவனவற்றை அறிந்து கொள்வார்கள்

- தரவுதளம் மற்றும் உறவுநிலை தரவுதளம் பற்றிய கருத்துருக்கள்
- தரவுதளத்தின் பல்வேறு கூறுகள்
- தரவுதள மாதிரிகளின் வகைகள்
- உறவுநிலையின் வகைகள்
- உறவுநிலை இயற்கணிதத்தின்(relational algebera) கருத்துருக்கள்

அறிமுகம்

தரவுதளம் என்பது ஒரு கணினி அமைப்பிலிருந்து மின்னணு முறையில் பெறப்பட்டு சேமிக்கப்படுகின்ற தரவுகளின் ஒழுங்குபடுத்தப்பட்ட தொகுப்பு ஆகும். "தரவுதளம்" என்பது DBMS, தரவுதள அமைப்பு அல்லது தரவுதளத்துடன் தொடர்புடைய பயன்பாடு போன்றவற்றை குறிப்பிடவும் பயன்படுகிறது. இவைகளுக்கிடையே உள்ள நெருங்கிய உறவுநிலையின் காரணமாக, தரவுதளம் என்ற சொல் தரவுதளம் மற்றும் தரவுதளத்தைக் கையாளுவதற்குப் பயன்படும். DBMS ஆகிய இரண்டையும் குறிப்பிட்ட பயன்படுகிறது. ஒரு பள்ளியில் வகுப்பறை பதிவேடு என்பது ஒரு தரவுதளமாகும். இதில், பெயர்கள் அகரவரிசையில் வரிசைப்படுத்தப் பட்டுள்ளன. மக்கள் நிகழ்வுகளை பதிவு செய்ய ஆரம்பித்ததிலிருந்து தரவுதளங்கள் பரவி வருகின்றன. இங்கு நாம் மின்னணு சார்ந்த வேலைகளில் கவனம் செலுத்துவோம்.

11.1 தரவு

தரவு என்பது கணிப்பொறியில் சேமிக்கப்பட்ட அடிப்படைத் துணுக்குகளாகும். தரவு, எந்தவொரு எழுத்து, உரை, வார்த்தை அல்லது எண்ணாகவோ இருக்க முடியும்.

எடுத்துக்காட்டு : 600006, DPI வளாகம், SCERT, சென்னை, கல்லூரி சாலை

11.2 தகவல்

தகவல் என்பது வடிவமைக்கப்பட்ட தரவாகும். இது தரவுகளை சில குறிப்பிடத்தக்க வழியில் பயன்படுத்த அனுமதிக்கிறது.

எடுத்துக்காட்டு

SCERT
கல்லூரி சாலை
DPI வளாகம்
சென்னை 600006

மேலே கொடுக்கப்பட்டுளள எடுத்துக்காட்டில் தரவுகள், சீரற்ற வார்த்தைகள் மற்றும் எண்களின் தொகுதியாக தோற்றமளிக்கிறது. எனினும் தரவானது, செயல்படுத்தப்பட்டு, திட்டமிடப்பட்டு மற்றும் வடிவமைக்கப்படும் பொழுது, அது SCERT நிறுவனத்தின் தொடர்பு முகவரியைப் பற்றிய ஒரு அர்த்தமுள்ள தகவல்களை தருகிறது.

11.3 தரவுதளம்

தரவுதளம் என்பது தரவுகளை எளிமையாக அணுக, நிர்வகிக்க, புதுப்பிக்கக் கூடிய, தொடர்புடைய தரவுகளின் களஞ்சியமாகும். தரவுதளம், தரவுகளை சேமிக்கும் நோக்கத்தோடு மென்பொருள் அல்லது வன்பொருளை தழுவியுள்ளது.

11.4 தரவுதள மேலாண்மை அமைப்பு (DBMS-DataBase Management System)

DBMS என்பது தரவுதளங்களை உருவாக்க, வரையறுக்க மற்றும் கையாளுவதற்கு

Chapter 11.indd 192 30-01-2020 20:40:40

அனுமதிக்கின்ற ஒரு மென்பொருளாகும். இது தரவுகளை எளிமையாக சேமிக்க, செயல்படுத்த மற்றும் பகுப்பாய்வு செய்ய பயனர்களை அனுமதிக்கிறது. DBMS, உருவாக்குதல் போன்ற பல தரவுதளத்தை வேலைகளை செய்வதற்கான இடைமுகம் (அ) கருவியை வழங்குகிறது. DBMS தரவுதளங்களுக்கு பாதுகாப்பையும் வழங்குகிறது. இது, பல பயனர்கள் உள்ள நிலையில் தரவு நிலைத்தன்மையையும் பராமரிக்கிறது.

DBMS மென்பொருளுக்கு எடுத்துக்காட்டுகள் : Foxpro, dbase.

11.4.1 தரவுதள மேலாண்மை அமைப்பின் பண்பியல்புகள்

1. அட்டவணைகளில் சேமிக்கப்பட்ட தரவுகள்	தரவுதளத்தில், தரவு எப்பொழுதும் நேரடியாக சேமிக்கிப்படுவதில்லை. தரவு தளத்தின் உள்ளே உருவாக்கப்பட்டுள்ள அட்டவணையில் தரவுகள் சேமிக்கப்பட்டிருக்கும். அட்டவணைகளுக்கு இடையே உறவுகளை ஏற்படுத்த DBMS அனுமதிக்கிறது. இது தரவுகளை மேலும் அர்த்தமுள்ளதாகவும், தொடர்புடையதாகவும் அமைக்கிறது.
2. மிகைமை (தேவைக்கு அதிகமானவற்றை) குறைத்தல்	நவீன உலகில் வன்வட்டுகளின் விலை மிகவும் குறைவு. ஆனால் முந்தைய காலங்களில் வன்வட்டுகள் மிகவும் விலை உயர்ந்தவை, ஆகையால் தரவுதளத்தில் உள்ள தேவையற்ற தரவுகள் பெரிய சிக்கலாக இருந்தது. ஆனால் DBMS தரவுகளை பிரித்தாலும் இயல்பு நிலையைப் பின்பற்றி தேவையற்ற தரவுகள் மீண்டும் இடம்பெறுவதைக் குறைக்கிறது.
3. தரவுகளின் நிலைத்தன்மை	தொடர்ச்சியாக புதுப்பிக்கப்படும் மற்றும் பராமரிக்கப்படும் நேரடி தரவின் (Live data) நிலைத்தன்மையை பராமரிப்பது ஒரு சவாலாக மாறும். ஆனால் DBMS தானாகவே அதை கையாளுகிறது.
4. பல பயனர்கள் மற்றும் ஒத்த அணுகுதலை ஆதரித்தல்	DBMS பல பயனர்களை ஒரே நேரத்தில் ஒரு தரவின் மீது வேலை (தரவுகளைப் புதுப்பித்தல், செருகுதல், நீக்குதல்) செய்ய அனுமதிக்கிறது. இருப்பினும் தரவின் நிலைத்தன்மைப் பராமரிப்பை கையாளுகிறது.
5. வினவல் மொழி	DBMS ஒரு எளிய வினவல் மொழியை பயனருக்கு வழங்குகிறது. இதைப்பயன்படுத்தி தரவுத்தளத்திலிருந்து தரவுகளை எளிமையாக பெறவும், சேர்க்கவும், நீக்கவும், புதுப்பிக்கவும் முடியும்.
6. பாதுகாப்பு	DBMS, தரவின் பாதுகாப்பை கவனித்துக் கொள்கிறது. அங்கீகாரமற்ற அணுகுதலில் இருந்து தரவுகளுக்கு பாதுகாப்பை வழங்குகிறது. வழக்கமான DBMSல், பல்வேறு அணுகலுக்கான அனுமதியுடன் பயனர் கணக்குகளை நாம் உருவாக்கமுடியும். இதனைப் பயன்படுத்தி பயனர் அணுகுதலை கட்டுப்படுத்தி நம்முடைய தரவுகளை எளிமையாக பாதுகாத்துக் கொள்ள முடியும்.
7. DBMS பரிவர்த்தனைகளை ஆதரித்தல்	இது நிகழ் உலக பயன்பாடுகளில் தரவு ஒருமைப்பாட்டை கையாள மற்றும் நிர்வகிக்க நம்மை அனுமதிக்கிறது. இங்கு பல்புரியாக்கம் (mutithreading) விரிவாக பயன்படுகிறது.

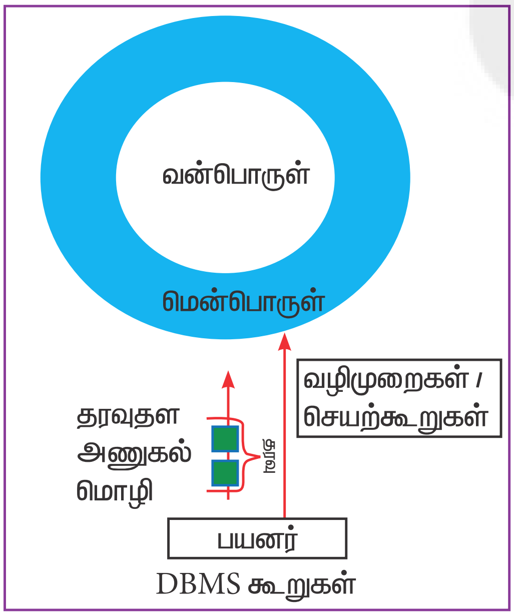
11.4.2 DBMS-ன் நிறைகள்

- பயன்பாட்டு நிரல்களைப் பிரித்தல்
- தரவுப்பிரதிகளை (Duplication) அல்லது தரவு மிகையைக் குறைத்தல்
- வினவல் மொழியைப் பயன்படுத்தி தரவை எளிமையாக திரும்பப் பெறுதல்
- தரவை உருவாக்கும் நேரம் மற்றும் பராமரித்தலைக் குறைத்தல்

11.4.3 DBMS-ன் கூறுகள்

தரவுத்தள மேலாண்மை அமைப்பு (DBMS) பின்வரும் ஐந்து கூறுகளாக பிரிக்கப்பட்டுள்ளது.

1. வன்பொருள்
2. மென்பொருள்
3. தரவு
4. வழிமுறைகள்/செயல்முறைகள்
5. தரவுத்தள அணுகல் மொழிகள்



படம் 11.1

1. வன்பொருள்: கணினி, வன்வட்டு, தரவுக்கான உள்ளீடு/வெளியீடு சாதனங்கள் மற்றும் தரவைச் சேமிக்கப் பயன்டும் வேறு பாகங்கள்

2. மென்பொருள்: அனைத்தையும் கட்டுப்படுத்தும் மிக முக்கியமான நிரலின் கூறு ஆகும். DBMS மென்பொருளானது தரவுத்தள அணுகல் மொழிகளைப் புரிந்து கொள்ள மற்றும் இயக்கத்திற்கான தரவுத்தள கட்டளைகளை நிரல் பெயர்க்க தகுதி வாய்ந்தது.

3. தரவு: இது DBMS ஐ வடிவமைப்பதற்கான வளம் ஆகும். DBMS தரவுகளை சேமிக்கவும், பயன்படுத்தவும் உருவாக்கப்படுகிறது.

4. வழிமுறைகள் / செயல்முறைகள்: இது தரவுத்தள மேலாண்மை அமைப்பை பயன்படுத்துவதற்கான பொதுவான கட்டளைகள் ஆகும். அதாவது, DBMS ஐ நிறுவுதல், தரவுத்தளங்களின் பிரதி எடுத்தலை (Backups) நிர்வகித்தல், அறிக்கை தயாரித்தல் போன்றவை ஆகும்.

5. தரவுத்தள அணுகல் மொழிகள்: இவை தரவுத்தளத்தில் சேமிக்கப்பட்ட தரவுகளை அணுகுதல், சேர்த்தல், புதுப்பித்தல் மற்றும் நீக்குதல் போன்ற கட்டளைகளை எழுதுவதற்கு பயன்படும் மொழிகள் ஆகும்.

11.5 தரவுத்தள கட்டமைப்பு

ஒரு அட்டவணையில் உள்ள தொடர்புடைய தரவுகளின் முழு தொகுப்பே அட்டவணை எனப்படும். இது தரவுகளை வரிசை மற்றும் நெடுவரிசைகளைக் கொண்ட கோப்பு (அ) அட்டவணை என்றும் அழைக்கப்படுகிறது. அட்டவணையின் ஒவ்வொரு வரிசையும் பதிவு என அழைக்கப்படுகிறது. இது ஒவ்வொரு தரவுத்தள பதிவிடுதலுக்குமான தரவின் தொகுதி ஆகும். ஒவ்வொரு நெடுவரிசையும் புலம் என அழைக்கப்படுகிறது. இது ஒவ்வொரு தரவின் வகையையும் குறிப்பிட்ட குழுக்களாக பிரிக்கிறது.

எ.கா. stu.No., stu.Name, stu.Age, stu.class, stu. sec.

ஒரு அட்டவணை உறவுகள், வரிசை எனவும் நெடுவரிசை எனவும் அழைக்கப்படுகிறது.

ஒரு அட்டவணை உறவுகள் (relation) எனப்படும்.

ஒரு வரிசை பதிவு (tuple) எனப்படும்.

Chapter 11.indd 194 30-01-2020 20:40:40

ஒரு நெடுவரிசை பண்புக்கூறு (attribute) எனப்படும்.

பண்புக்கூறு / நெடுவரிசை

StuNo	Stu Name	StuAge	StuClass	StuSec
1101	Kannan	14	9	A
1102	Ramakrishnan	14	9	A
1103	Vidhya	14	10	B
1104	Britto	15	9	A
1105	Padmaja	14	10	B

Tuple / வரிசை

அட்டவணை / உறவுகள்

தரவுதள அமைப்பு

படம் 11.2

11.6 தரவு மாதிரி

தரவு மாதிரி, முழுமையான செயல்பாட்டிற்குப் பிறகு ஒரு மென்பொருளில் இருந்து தரவு எவ்வாறு பெறப்படுகிறது என்பதை விளக்குகிறது.

- இது சிக்கலான நிகழ் உலக தரவு சேகரிக்கும் சூழலை எளிமையாக்குகிறது.
- தரவு மாதிரியின் முக்கிய நோக்கம், இறுதி மென்பொருள், அதன் முழுமையான உருவாக்கத்திற்குப் பிறகு எவ்வாறு இருக்கும் என்பதற்கான ஒரு மாதிரியைத் தருகிறது.

11.6.1 தரவு மாதிரியின் வகைகள்

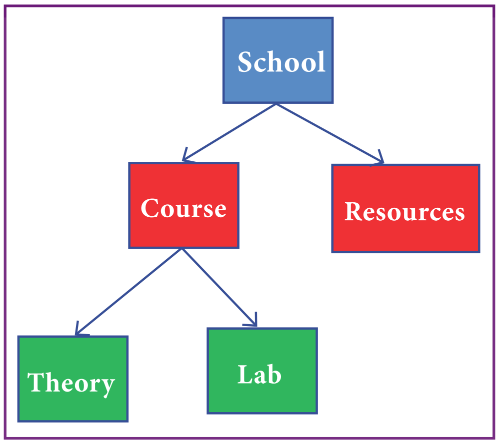
தரவு மாதிரியின் பல்வேறு வகைகள் பின்வருமாறு

1. படிநிலை தரவுதள மாதிரி (Hierarchical Model)
2. உறவுநிலை தரவுதள மாதிரி (Relational Model)
3. வலையமைப்பு தரவுதள மாதிரி (Network Model)
4. ER தரவுதள மாதிரி (Entity Relationship Model)
5. பொருள்நோக்கு தரவுதள மாதிரி (Object Model)

1. படிநிலை மாதிரி

இது IBM-ஆல் தகவல் மேலாண்மை அமைப்பு **Information Management System** போல உருவாக்கப்பட்டது. இந்த மாதிரியில் தரவு எளிமையான மரக்கிளை அமைப்பில் குறிப்பிடப்படுகிறது. இது ஒன்றிலிருந்து பல **(one to many)** உறவு நிலையை குறிக்கிறது. **(Parent - child)** அதாவது பெற்றோர் – குழந்தை உறவுநிலை. ஒரு குழந்தைக்கு ஒரு பெற்றொர் இருப்பர். ஆனால், ஒரு பெற்றோருக்கு பல குழந்தைகள் இருக்கலாம். இது முக்கியமாக IBM தலைமைக் கணிப்பொறியில் பயன்படுகிறது.

Chapter 11.indd 195 30-01-2020 20:40:40



படிநிலை மாதிரி படம். 11.3

2. உறவுநிலை மாதிரி

உறவுநிலை தரவுதள மாதிரி முதன்முதலில் E.F.Codd என்பவரால் 1970-ல் உருவாக்கப்பட்டது. தற்போது உலகம் முழுவதும் தரவுதள பயன்பாடுகளில் இத்தரவுதள மாதிரி பரவலாக பயன்படுத்தப்பட்டு வருகிறது.

உறவுநிலை தரவுதள மாதிரியில் தரவுகளின் அடிப்படை கட்டமைப்பு அட்டவணைகள் (உறவுகள்) ஆகும். ஒரு குறிப்பிட்ட வகையை சார்ந்த அனைத்து தகவல்களும் அவ்வட்டவணையின் வரிசைகளில் சேமிக்கப்படுகின்றன. எனவே, அட்டவணைகளை உறவுநிலை தரவுதள மாதிரியில் உறவுகள் (Relation) என்கிறோம். ஒரு உறவுநிலை தரவுகோல் குறிப்பிட்ட வரிசையிலான தரவுகளை தனித்தன்மையுடன் குறிக்கும் ஒரு பண்புக்கூறு ஆகும்.

Stu_id	Name	Age
1	Malar	17
2	Suncar	16
3	Velu	16

Subj_id	Name	Teacher
1	C++	Kannan
2	Php	Ramakrishnan
3	Python	Vidhya

Stu_id	Subj_id	Marks
1	1	92
1	2	89
3	2	96

உறவு நிலை தரவுதள மாதிரி
படம் 11.4

3. வலையமைப்பு மாதிரி

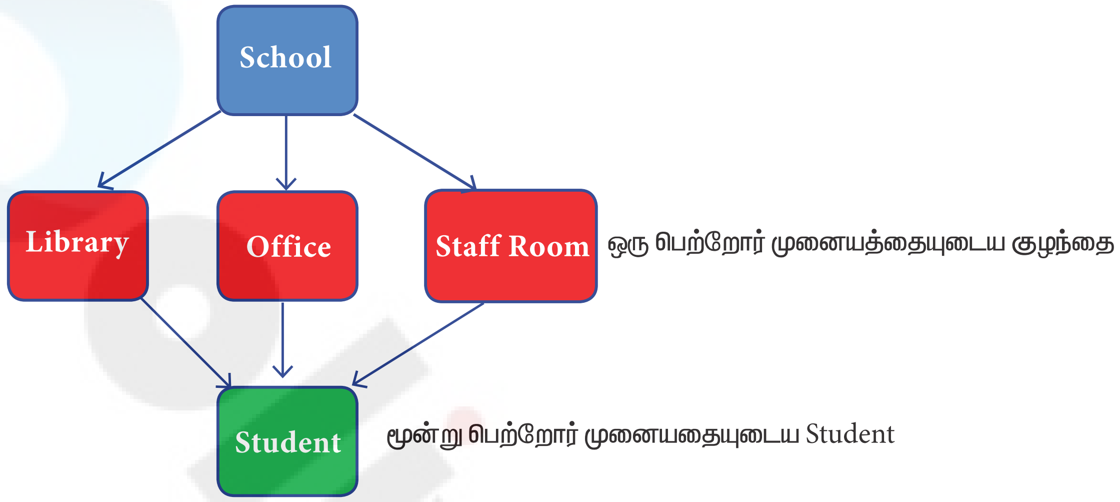
வலையமைப்பு தரவுத்தள மாதிரி படிநிலை தரவுத்தள மாதிரியின் விரிவாக்கப்பட்ட அமைப்பாகும். படிநிலை மற்றும் வலையமைப்பு மாதிரிக்கும் இடையே உள்ள வேறுபாடு.

- படிநிலை மாதிரியில், ஒரு குழந்தை பதிவு ஒரே ஒரு பெற்றோர் முனையத்தை மட்டுமே கொண்டிருக்கும்.

Chapter 11.indd 196 30-01-2020 20:40:40

• வலையமைப்பு மாதிரியில் ஒரு குழந்தைக்கு பல பெற்றோர் முனையங்கள் இருக்கலாம். இது தரவை பலவற்றிலிருந்து பலவற்றிற்கு (many to many) உறவு நிலையை குறிப்பிடுகிறது

• வலையமைப்பு மாதிரி எளிமையாகவும், விரைவாகவும் தரவுகளை அணுக பயன்படுகிறது



தரவுதள மாதிரி
படம் 11.5

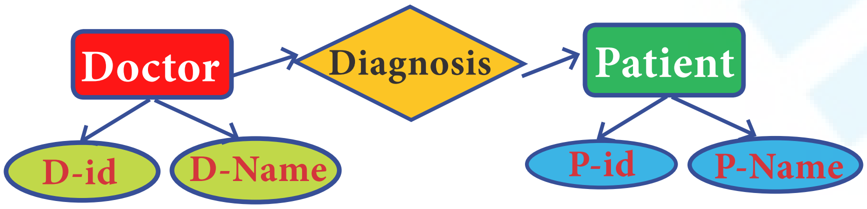
School என்பது பெற்றோர் முனையத்தை குறிக்கும்

Library, Office மற்றும் Staff room என்பன school- ன் குழந்தைகளாகும் (பெற்றோர் முனையம்)

Student என்பது library, office மற்றும் staff room என்பதன் குழந்தை (ஒன்றிலிருந்து பலவற்றிற்கான உறவுநிலை)

4. ER தரவுதள மாதிரி (Entity Relationship Model)

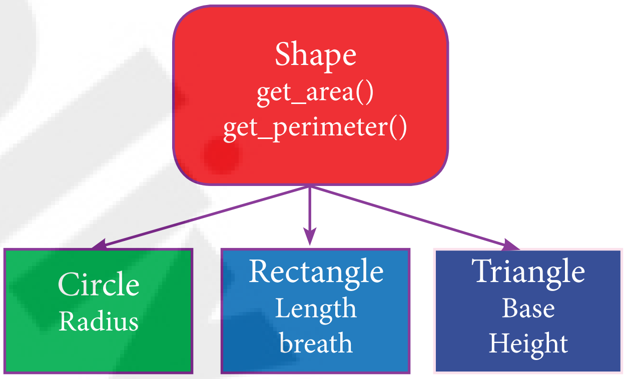
இந்த தரவு மாதிரியில், பொருளை உருப்படியாகவும் அதன் பண்புகளை, பண்புக் கூறுகளாகவும் பிரித்து உறவு நிலை உருவாக்கப்படுகிறது. 1976ல் Chen சென் என்பவரால் உருவாக்கப்பட்டது. இந்த மாதிரி தரவுத்தளத்திற்கான கருத்து வடிவமைப்பை உருவாக்க பயன்படுகிறது. தரவின் விளக்கப்படத்தை வடிவமைக்க மிகவும் எளிமையாக உள்ளது. ER மாதிரியைக் கொண்டு நிரலர் அமைப்பை எளிமையாக புரிந்து கொள்ள முடியும். செவ்வகம் உருப்படிகளைக் குறிக்கிறது. எடுத்துக்காட்டு Doctor, Patient நீள்வட்டம் பண்புக் கூறுகளைக் குறிக்கிறது. எடுத்துக்காட்டாக D-id, D-Name, P-id, P-Name பண்புக்கூறுகள் பண்பியல்புகளை விளக்குகிறது. ஒவ்வொரு உருப்படியும் தரவுத்தளத்தில் சேமிக்கப்பட்ட தரவின் முக்கிய பகுதியாகிறது. டைமண்ட் ER படங்களின் உறவுநிலையைக் குறிப்பிடுகிறது. எடுத்துக்காட்டு Doctor, Patient யை பரிசோதனை செய்கிறார்.



ER தரவுதள மாதிரி
படம் 11.6

5. பொருள்நோக்கு தரவுதள மாதிரி (Object Model)

இந்த மாதிரியானது தரவை பொருள்கள், பண்புக்கூறுகள், வழிமுறைகள், இனக்குழு மற்றும் மரபுரிமம் போன்ற வழிகளில் சேமிக்கிறது. இது மிகவும் சிக்கலான பயன்பாடுகளான புவியியல் தகவல் அமைப்பு (GIS- Geographic Information System) அறிவியல் சோதனைகள் (Scientific experiments) பொறியியல் வடிவமைப்பு (Engineering design) உற்பத்தி (Manufacturing) போன்றவற்றைக் கையாள்கிறது. இது கோப்பு மேலாண்மை அமைப்பில் பயன்படுகிறது. நிகழ்உலக பொருள்கள், பண்புக்கூறுகள், பண்பியல்புகளை குறிப்பிடுகிறது மற்றும் தெளிவான கூறுநிலை (modular structure) அமைப்பை வழங்குகிறது. இதில் ஏற்கனவே உள்ள குறிமுறையை எளிதாகப் பராமரிக்கவும், மாற்றி அமைக்கவும் முடியும்.



பொருள்நோக்கு தரவுதள மாதிரி படம் 11.7

பொருள்நோக்கு தரவுதள மாதிரியின் எடுத்துக்காட்டு Shape, Circle, Rectangle மற்றும் Triangle ஆகியவை இந்த மாதிரியில் உள்ள பொருள்களாகும்.

- **Circle** என்பது **radius** என்ற பண்புக்கூற்றையும்,
- **Rectangle** என்பது **length** மற்றும் **breath** என்ற பண்புக்கூறுகளையும்,
- **Triangle** என்பது **base** மற்றும் **height** என்ற பண்புக்கூறுகளையும் கொண்டுள்ளன.
- Circle, Rectangle மற்றும் Triangle ஆகிய பொருள்கள் Shape என்ற பொருளில் இருந்து தருவிக்கப்பட்டவைகளாகும்.

11.6.2 DBMS பயனர்களின் வகைகள்

தரவுத்தள நிர்வாகிகள் (DBA- Data Base Administrator)

தரவுத்தள நிர்வாகி அல்லது DBA என்பவர் முழு தரவுத்தள மேலாண்மை அமைப்பையும் நிர்வகிப்பவர் ஆவார். இவர் DBMS பாதுகாப்பு, உரிமங்களை நிர்வகித்தல், பயனர் கணக்குகள் மற்றும் அணுகல்களை நிர்வகித்தல் போன்றவற்றை கவனித்துக் கொள்கிறார்.

பயன்பாட்டு நிரலர்கள் அல்லது மென்பொருள் உருவாக்குபவர்கள்

இந்த பயனர்கள் DBMSன் பகுதிகளை உருவாக்குதல் மற்றும் வடிவமைத்தல் போன்ற வேலைகளில் ஈடுபடுகிறார்கள்.

Chapter 11.indd 198 30-01-2020 20:40:40

இறுதி நிலைப்பயனர்

அனைத்து நவீன பயன்பாடுகளிலும், வலை அல்லது கைபேசி, பயனர் தரவுகளைச் சேமிக்கிறது . பயனர் தரவுகள் (User data) சேகரிக்கப்பட்டு சேவையகத்தில் இயங்குகின்ற DBMS அமைப்பில் சேமிக்கப்படும் வகையில் பயன்பாடுகள் நிரலாக்கப்பட்டுள்ளன. இந்த பயனர்கள் தரவை சேமித்தல், திரும்ப பெறுதல், புதுப்பித்தல் மற்றும் நீக்குதல் போன்ற பல்வேறு செயல்களை கையாள்கிறார்கள்.

தரவுத்தள வடிவமைப்பாளர்கள்:

இவர்கள் தரவுத்தளத்தில் சேமிக்கப்படுகின்ற தரவுகளை கண்டறிந்து சரியான கட்டமைப்பை தேர்வு செய்து தரவை குறிப்பிடுவதற்கும் சேமிப்பதற்குமான பொறுப்பாளர்கள் ஆவர்

11.7 DBMSக்கும் RDBMSக்கும் இடையேயான வேறுபாடு

ஒப்பீட்டு அடிப்படை	DBMS	RDBMS
விரிவாக்கம்	Database Management System தரவுதள மேலாண்மை அமைப்பு	Relational DataBase Management System உறவுநிலை தரவுதள மேலாண்மை அமைப்பு
தரவு சேமிப்பு	ஊடுருவலின் மாதிரி அதாவது ஒன்றோடொன்று இணைக்கப்பட்ட பதிவுகளின் தரவு	உறவுநிலை மாதிரி (அட்டவணையில்) அட்டவணையில் வரிசை மற்றும் நெடுவரிசை உள்ள தரவுகள்.
மிகைமைத் தரவு	இடம் பெற்றுள்ளது	இடம் பெறவில்லை
இயல்பாக்கம்	செய்ய இயலாது	இயல்பாக்கத்தை பயன்படுத்துகிறது.
தரவு அணுகல்	அதிக நேரத்தை எடுத்துக் கொள்கிறது.	DBMS உடன் ஒப்பிடும் போது வேகமானது.
திறவு கோல்கள் மற்றும் குறியீடுகள்	பயன்படுத்தப்படவில்லை	உறவுநிலையை உருவாக்குவதற்குப் பயன்படுகிறது.
பரிவர்த்தனை மேலாண்மை	திறமையற்றது, பிழைகளைக் கொண்டது பாதுகாப்பற்றது.	திறமையானது மற்றும் பாதுகாப்பானது.
பரவல் தகவல்தளம்	ஒத்துழைக்காது	ஒத்துழைக்கும்
எடுத்துக்காட்டு	Dbase, FoxPro.	SQL server, Oracle, mysql, MariaDB, SQLite.

தரவுத்தள இயல்பாக்கம் முதலில் டாக்டர் எட்கர் எப் காட் (Dr. Edgar F Codd) என்பவரால் RDBMS-ன் உட்பகுதியாக முன் மொழியப்பட்டது, இது தரவு மிகைமையைக் குறைக்கிறது மற்றும் தரவு நிலைப்பாட்டை அதிகப்படுகிறது. இந்த விதிகளே E F Codd விதிகள் எனப்படும்.

Chapter 11.indd 199 30-01-2020 20:40:40

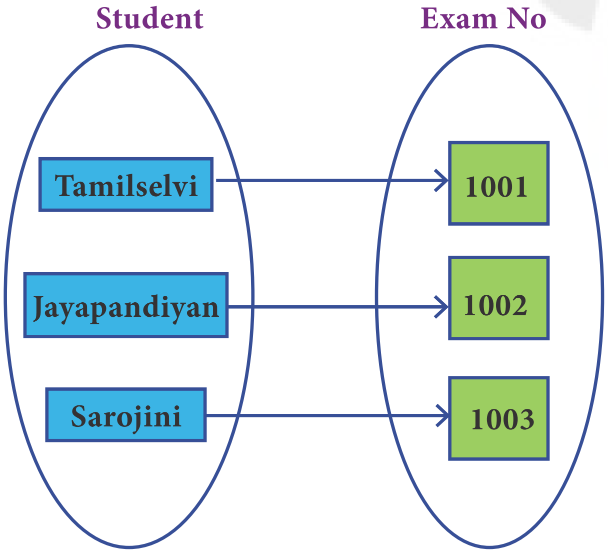
11.8 உறவு நிலைகளின் வகைகள்

தரவுத்தளத்தில் பயன்படும் உறவுநிலைகளின் வகைகள்.

1. ஒன்றுடன் ஒன்று உறவுநிலை (One-to-One Relationship)
2. ஒன்றுடன் பல உறவுநிலை (One-to-Many Relationship)
3. பலவற்றுடன் ஒன்று உறவுநிலை (Many-to-One Relationship)
4. பலவற்றுடன் பல உறவுநிலை (Many-to-Many Relationship)

1. ஒன்றுடன் ஒன்று உறவுநிலை

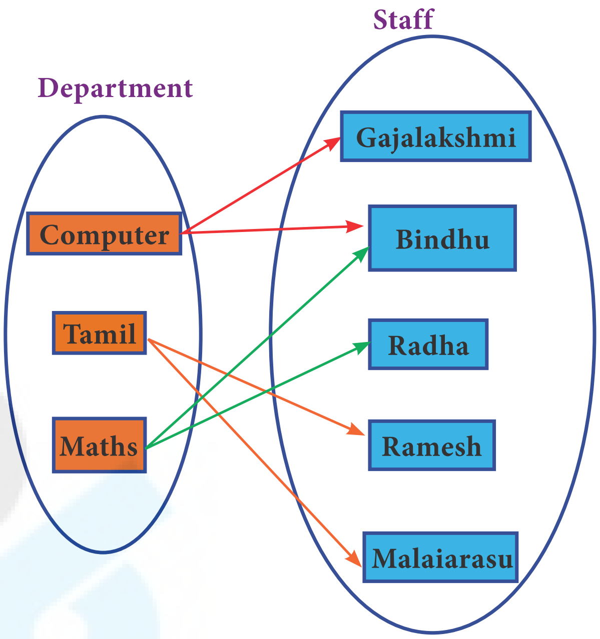
இந்த உறவுநிலையில், ஒரு பொருள் வேறு ஒரு பொருளுடன் மட்டுமே தொடர்பு கொண்டிருக்கும். ஒரு அட்டவணையின் ஒரு பதிவு மற்றொரு அட்டவணையின் ஒரு பதிவுடன் மட்டுமே இணைக்கப்பட்டிருக்கும். எடுத்துக்காட்டு ஒரு மாணவனுக்கு ஒரு தேர்வெண் மட்டுமே இருக்க முடியும்.



ஒன்றுடன் ஒன்று உறவுநிலை
படம் 11.8

2. ஒன்றுடன் பல உறவுநிலை

இந்த உறவுநிலையில் ஒரு பொருள் வேறு பல பொருள்களுடன் தொடர்பு கொண்டிருக்கும். அட்டவணை A-வின் ஒரு பதிவு அட்டவணை B யின் பல பதிவுகளுடன் தொடர்பு கொண்டிருக்கலாம். ஆனால் அட்டவணை Bயின் ஒரு பதிவு அட்டவணை A-வின் ஒரு பதிவுடன் மட்டுமே தொடர்பு கொண்டிருக்க முடியும். எடுத்துக்காட்டு: ஒரு துறை பல ஊழியர்களைக் கொண்டுள்ளது.



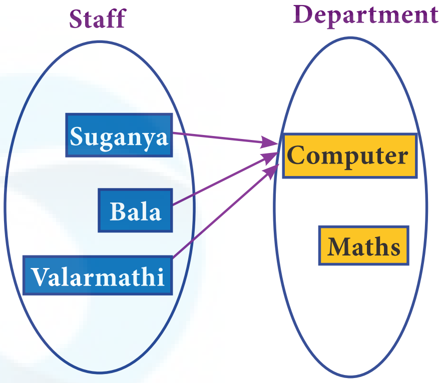
ஒன்றுடன் பல
உறவுநிலை படம் 11.9

3. பலவற்றுடன் ஒன்று உறவுநிலை

இந்த உறவுநிலையில் பல பொருள்கள் வேறு ஒரேயொரு பொருள்களுடன் மட்டுமே தொடர்பு கொண்டிருக்கும்.

எடுத்துக்காட்டு: ஒரு துறையில் வேலை செய்யும் ஊழியர்களின் எண்ணிக்கையில், ஊழியர்கள் அட்டவணையின் பல பதிவுகள் துறை அட்டவணையின் ஒரு பதிவுடன் மட்டுமே தொடர்பு படுத்தப்பட்டிருக்கும்.

Chapter 11.indd 200 30-01-2020 20:40:40



பலவற்றுடன் ஒன்று
உறவுநிலை படம் 11.10

4. பலவற்றுடன் பல உறவுநிலை

இந்த உறவுநிலையில் ஒரு அட்டவணையில் உள்ள பல பதிவுகள் மற்றொரு அட்டவணையில் உள்ள பல பதிவுகளுடன் தொடர்பு கொண்டிருக்கும்.

எடுத்துக்காட்டு .1. நுகர்வோர் மற்றும் விலை பொருள்

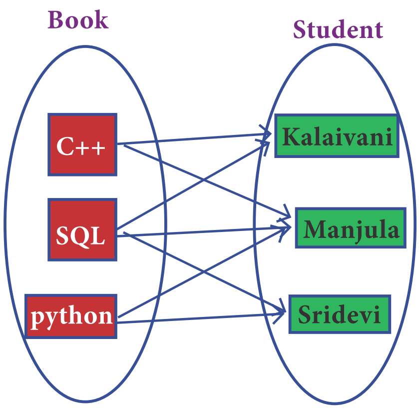
நுகர்வோர் பல விலைப்பொருள்களை வாங்குகிறார் மற்றும் விலைப்பொருள்கள் பல நுகர்வோரால் வாங்கப்படலாம்.

எடுத்துக்காட்டு 2 மாணவர்கள் மற்றும் பாடப்பிரிவுகள்

ஒரு மாணவன் பல பிரிவுகளில் பதியலாம் மற்றும் ஒரு பாடப்பிரிவு பல மாணவர்களைக் கொண்டிருக்கலாம்.

எடுத்துக்காட்டு 3: புத்தகங்கள் மற்றும் மாணவன்.

நூலகத்தில் உள்ள பல புத்தகங்கள் பல மாணவர்களுக்கு வழங்கப்படும்.



பலவற்றுடன் பல
உறவுநிலை படம் 11.11

உங்களுக்குத் தெரியுமா

உறவுநிலை மாதிரி, எட்கர்ஃபிராங்க் காட் (Edgar Frank Codd) (உறவுநிலை மாதிரியின் தந்தை) என்பவரால் கண்டுபிடிக்கப்பட்டது. இது தரவின் பொது மாதிரி ஆக உள்ளது. இது கிரிஸ்டேட் மற்றும் ஹக் டார்வென் மற்றும் பலரால் மேம்படுத்தப்பட்டது.

11.9 DBMS- ல் உறவுநிலை இயற்கணிதம்

உறவுநிலை இயற்கணிதம் என்றால் என்ன?

உறவுநிலை இயற்கணிதம் முதலில் எட்கர் எப் காட் என்பவரால் IBMல் உருவாக்கப்பட்டது. இது உறவுநிலை தரவுத்தளங்களில் சேமிக்கப்பட்ட தரவை மாதிரியாக்கம் செய்யவும் அதில் வினவல்களை வரையறுக்கவும் பயன்படுகிறது. உறவுநிலை இயற்கணிதம், செயல்முறை வினவல்மொழி என அழைக்கப்படுகிறது. இது SQLயைப் பயன்படுத்தி தரவுத்தள அட்டவணைகளில் வினவல்களைக் கொடுக்கவும் பயன்படுகிறது.

உறவுநிலை இயற்கணிதச் செயற்பாடுகள் விடையைப் பெறுவதற்காக ஒரு உறவுநிலையில் (அட்டவணை) தற்சுழற்சி முறையில் செய்யப்படுகின்றன. இந்த செயற்பாடுகளின் விடை ஒன்று அல்லது அதற்கு மேற்பட்ட உள்ளீடு உறவுகளால் ஏற்படுத்தப்பட்ட ஒரு புதிய உறவாகும்.

உறவுநிலை இயற்கணிதம் பல்வேறு தொகுதிகளாகப் பிரிக்கப்பட்டுள்ளது.

ஒரும உறவுநிலைச் செயற்பாடுகள்

SELECT (symbol : σ)

PROJECT (symbol : ∏)

Set தேற்றத்தில் இருந்து உறவுநிலை இயற்கணித செயற்பாடுகள்

- ஒட்டுதல் (∪)

Chapter 11.indd 201
30-01-2020 20:40:40

- வெட்டுதல் (∩)
- வேறுபாடு (−)
- கார்டீசியன் பெருக்கல் (X)

SELECT (symbol : σ)

பொதுவடிவம் σc (R): என்பது உறவுநிலை மற்றும் அதனுடைய பண்புக்கூறுகளின் நிபந்தனை C

இந்த select செயற்பாடு ஒரு நிபந்தனையின் அடிப்படையில் துணைத் தொகுதியை tuples களுடன் தேர்ந்தெடுக்க உதவுகிறது. நிபந்தனைகளை திருப்திபடுத்தாத tuples களை Select வடிகட்டுகிறது.

STUDENT

Studno	Name	Course	Year
cs1	Kannan	Big Data	II
cs2	Gowri Shankar	R language	I
cs3	Lenin	Big Data	I
cs4	Padmaja	Python Programming	I

அட்டவணை 11.1

σ_{course} = "Big Data" (STUDENT)

Studno	Name	Course	Year
cs1	Kannan	Big Data	II
cs3	Lenin	Big Data	I

PROJECT (symbol : Π)

இந்த PROJECT செயற்பாடு குறிப்பிடப்பட்டுள்ள உள்ளீடு தொடர்புகளின் பண்புக்கூறுகளை நீக்குகிறது. இந்த செயற்பாடு கிடக்கை துணைத் தொகுதியின் ஒப்பீடுகளை வரையறுக்கிறது.

எடுத்துக்காட்டு 1 அட்டவணை 11.1ஐ பயன்படுத்தி

Π_{course} (STUDENT)

Result

Course
Big Data
R language
Python Programming

Chapter 11.indd 202 30-01-2020 20:40:41

குறிப்பு

விடையில் நகல் வரிசைகள் நீக்கப்பட்டுள்ளன.

எடுத்துக்காட்டு 2 (அட்டவணை 11.1)

$\Pi_{\text{studno, course}}$ (STUDENT)

விடை

Studno	Course
cs1	Big Data
cs2	R language
cs3	Big Data
cs4	Python Programming

ஒட்டுதல் (Symbol :∪)

இது A அல்லது B அட்டவணையில் உள்ள அனைத்து tuples-களையும் உள்ளடக்கியது. இது நகல்களையும் நீக்குகிறது. தொகுதி A ஒட்டுதல் தொகுதி B என்பது A ∪ B

எடுத்துக்காட்டு 3 (அட்டவணை 11.1ஐ பயன்படுத்தி)

பின்வரும் அட்டவணையை நோக்குக.

Table A		Table B	
Studno	**Name**	**Studno**	**Name**
cs1	Kannan	cs1	Kannan
cs3	Lenin	cs2	GowriShankarn
cs4	Padmaja	cs3	Lenin

அட்டவணை 11.2

விடை

Table A ∪ B	
Studno	**Name**
cs1	Kannan
cs2	GowriShankar
cs3	Lenin
cs4	Padmaja

வேறுபாடு (Symbol : -)

இது A மற்றும் B அட்டவணைகளை ஒப்பிட்டு A அட்டவணையில் உள்ள வேறுபட்ட tuples களை மட்டும் தருகிறது. B அட்டவணையைத் தவிர்க்கிறது. A பண்புக்கூறின் பெயர் B பண்புக்கூறின் பெயரோடு பொருந்த வேண்டும்.

Chapter 11.indd 203 30-01-2020 20:40:41

எடுத்துக்காட்டு 4 (அட்டவணை 11.2 ஐ பயன்படுத்தி)

விடை

Table A – B	
cs4	Padmaja

வெட்டு (symbol : ∩) A ∩ B

இது A மற்றும் B அட்டவணைகளில் பொதுவாக உள்ள அனைத்து tuplesகளையும் உள்ளடக்கியது.

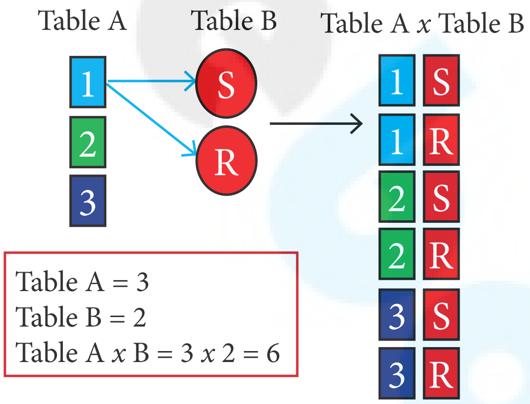
எடுத்துக்காட்டு 5 (அட்டவணை 11.2ஐ பயன்படுத்தி)

A ∩ B	
cs1	Kannan
cs3	Lenin

கார்டீசியன் பெருக்கல் (Symbol : X)

இரண்டு தொடர்புகளை சேர்க்க குறுக்குப் பெருக்கல் வழிவகுக்கிறது. இதன் விடை இரண்டு தொடர்புகளின் இணைப்பைக் கொண்டுள்ளது.

A x B என்பது A times B, இங்கு A தொடர்புகள் மற்றும் B தொடர்புகள் என்பன வேறுபட்ட பண்புக்கூறுகளாகும். இந்த வகை செயற்பாடுகள் இரண்டு தொடர்புகளிலிருந்து நெடுக்கைகளை ஒன்று சேர்க்க பயன்படுகிறது.



கார்டீசியன் பெருக்கல் படம் 11.12

Table A		Table B	
studno	**name**	**studno**	**subject**
cs1	Kannan	cs28	Big Data
cs2	Gowri Shankar	cs62	R language
cs4	Padmaja	cs25	python programming

அட்டவணை 11.3

Chapter 11.indd 204 30-01-2020 20:40:41

கார்டீசியன் பெருக்கல் : Table A x Table B

studno	name	course	subject
cs1	Kannan	cs28	Big Data
cs1	Kannan	cs62	R language
cs1	Kannan	cs25	python rogramming
cs2	Gowri Shankar	cs28	Big Data
cs2	Gowri Shankar	cs62	R language
cs2	Gowri Shankar	cs25	python programming
cs4	Padmaja	cs28	Big Data
cs4	Padmaja	cs62	R language
cs4	Padmaja	cs25	python programming

நினைவில் கொள்க

- DBMS என்பது பதிவுகளை கொண்டிருக்கும் கணிப்பொறிதழுவிய ஒரு அமைப்பாகும்.
- தரவு என்பது செயல்படுத்தப்படாத தரவுகளாகும். இது எந்த ஒரு எழுத்து, உரை, வார்த்தை அல்லது எண்ணாகவோ இருக்கலாம்.
- தகவல் என்பது செயற்படுத்தப்பட்ட, ஒருங்கிணைக்கப்பட்ட வடிவூட்டப்பட்ட தரவுகளாகும்.
- mysql, oracle, sql server, ibm db2 ஆகியவை RDBMSக்கான எடுத்துக்காட்டுகளாகும்.
- தரவுத்தளத்தில் உள்ள தரவின் நகலே மிகைமை எனப்படும்.
- தரவு நிலைத்தன்மை என்பது தரவுத்தளத்தில் அனைத்து இடங்களிலும் ஒத்த மதிப்புடைய தரவுகளாகும்.
- தரவு நிலைப்பாடு என்பது அனுமதியற்ற பயனர்களிடம் இருந்து பாதுகாப்பு வழங்குகிறது.
- அட்டவணை என்பது ஒரு தொடர்பு (relation) எனப்படும்.
- ஒரு வரிசை என்பது tuple என்று அழைக்கப்படுகிறது.
- ஒரு நெடுவரிசை என்பது பண்புக்கூறு எனப்படும்.
- படிநிலை, உறவுநிலை, வலையமைப்பு, ER மற்றும் பொருள் மாதிரி போன்றவை தரவு மாதிரியின் வகைகள் ஆகும்.
- படிநிலை மாதிரி ஒன்று ஒன்று உறவுநிலையைக் கொண்ட மரக்கிளை போன்ற கட்டமைப்பு உடையது. இது பெற்றோர் குழந்தை உறவுநிலை எனவும் அழைக்கப்படுகிறது.
- உறவுநிலை மாதிரி இது தரவை தொடர்புகள் அல்லது அட்டவணைகளாக குறிப்பிடுகிறது.
- வலையமைப்பு மாதிரி- இது படிநிலை மாதிரி போன்றதாகும். ஆனால் இது ஒன்றுக்கு மேற்பட்ட பெற்றோரை கொண்டிருக்க அனுமதிக்கிறது.
- ER- பொருள்கள், பண்புக்கூறுகள் மற்றும் தொடர்புகளைக் கொண்ட உயர்நிலை தரவு மாதிரி ஆகும்.
- பொருள் மாதிரி- தரவை பொருள்கள், இனக்குழுக்கள் மற்றும் மரபுரிம மாகச் சேமிக்கிறது.
- இயல்பாக்கம்- தரவுமிகைமைக் குறைக்கிறது மற்றும் தரவு நிலைப்பாட்டை அதிகரிக்கிறது.

நினைவில் கொள்க

- பல்வேறு வகையான உறவுநிலைகள்
- ஒன்று ஒன்று – ஒரு மாணவனுக்கு ஒரு தேர்வெண் மட்டுமே இருக்க முடியும்.
- ஒன்று பல- ஒரு துறையில் பல ஊழியர்கள் இருக்கலாம்.
- பல ஒன்று- பல ஊழியர்கள் ஒரே துறையில் பணி செய்யலாம்.
- பல பல – நூலகத்தில் பல புத்தகங்கள் பல மாணவர்களுக்கு வழங்கமுடியும்.
- தரவுத்தள இயல்பாக்கம் Dr.Edgar F Codd என்பவரால் முன்மொழியப்பட்டது.
- உறவுநிலை இயற்கணிதம் செயல்முறை வினவல் மொழி என அழைக்கப்படுகிறது. இது SQLயைப் பயன்படுத்தி தரவுத்தள அட்டவணைகளில் வினவல்களைக் கொடுக்கவும் பயன்படுகிறது.

மதிப்பீடு

BUEQ24

பகுதி–அ

சரியான விடையை தேர்ந்தெடுத்து எழுதுக **(1 மதிப்பெண்)**

1. DBMS-ன் விரிவாக்கம்?

   (அ) DataBase Management Symbol (ஆ) Database Managing System
   (இ) DataBase Management System (ஈ) DataBasic Management System

2 ஒரு அட்டவணை என்பது

   (அ) வரிசை (tuple) (ஆ) பண்புக்கூறுகள் (attribute)
   (இ) உறவுகள் (relation) (ஈ) அமைப்பு (entity)

3 எந்த தரவுத்தள மாதிரி பெற்றோர் குழந்தை உறவுநிலையை குறிப்பிடுகிறது?

   (அ) உறவுநிலை (ஆ) வலையமைப்பு
   (இ) படிநிலை (ஈ) பொருள்

4 உறவுநிலை தரவுத்தள மாதிரி முதலில் யாரால் முன்மொழியப்பட்டது?

   (அ) E F Codd (ஆ) E E Codd
   (இ) E F Cadd (ஈ) E F Codder

5 படிநிலை மாதிரி எந்த வகை உறவுநிலையை குறிப்பிடுகிறது?

   (அ) ஒன்று ஒன்று (ஆ) ஒன்று பல
   (இ) பல ஒன்று (ஈ) பல பல

6. உறவுநிலை தரவுத்தளத்தின் தந்தை என்று அழைக்கப்படுபவர் யார்?

   (அ) Chris Date (ஆ) Hugh Darween
   (இ) Edgar Frank Codd (ஈ) Edgar Frank Cadd

Chapter 11.indd 206 30-01-2020 20:40:41

7. பின்வருவனவற்றுள் எது RDBMS?

(அ) Dbase b) Foxpro

(இ) Microsoft Access (ஈ) SQLite

8 SELECT கூற்றுக்கு பயன்படும் சின்னம் எது?

(அ) σ (ஆ) Π (இ) X (ஈ) Ω

9 ஒரு tuple என்பது

(அ) table (ஆ) row

(இ) attribute (ஈ) field

10. ER மாதிரியை உருவாக்கியவர் யார்?

(அ) Chen (ஆ) EF Codd

(இ) Chend (ஈ) Chand

பகுதி–ஆ

அனைத்து வினாக்களுக்கும் விடையளி (2 மதிப்பெண்)

11. தரவுத்தளத்திற்கான சில எடுத்துக்காட்டுகளைக் குறிப்பிடுக.
12. RDBMS-ன் சில எடுத்துக்காட்டுகளைப் பட்டியலிடுக.
13. தரவு நிலைத் தன்மை என்றால் என்ன?
14. படிநிலை மற்றும் வலையமைப்பு தரவு மாதிரிக்கும் இடையே உள்ள வேறுபாடு என்ன?
15. இயல்பாக்கம் என்றால் என்ன?

பகுதி–இ

அனைத்து வினாக்களுக்கும் விடையளி (3 மதிப்பெண்)

16. Select மற்றும் Project செயற்பாடுகளின் வேறுபாடுகள் யாவை?
17. DBAவின் பணி என்ன?
18. கார்டீசியன் பெருக்கலை பொறுத்தமான எடுத்துக்காட்டுடன் விளக்குக.
19. பொருள் மாதிரியை எடுத்துக்காட்டுடன் விளக்குக.
20. RDBMS-ன் பல்வேறு வகையான பயனர்களைப் பற்றி குறிப்பு வரைக.

பகுதி ஈ

அனைத்து வினாக்களுக்கும் விடையளி (5 மதிப்பெண்)

21. தரவு மாதிரியின் பல்வேறு வகைகளை விளக்குக.
22. உறவுநிலையின் வகைகளை விவரி.
23. DBMS மற்றும் RDBMS வேறுபடுத்துக.
24. ஒட்டுதல், வெட்டுதல், வேறுபாடு மற்றும் கார்டீசியன் பெருக்கல் போன்றவற்றை பொருத்தமான எடுத்துக்காட்டுடன் விளக்குக.
25. DBMS ன் பண்பியல்புகளை விளக்குக

Chapter 11.indd 207 30-01-2020 20:40:41

அலகு IV

பாடம் 12

வினவல் அமைப்பு மொழி (SQL)

QFV3UM

கற்றலின் நோக்கங்கள்

இந்தப் பாடப்பகுதியைக் கற்றபின் மாணவர் அறிந்து கொள்வது:

- SQL வினவல் மொழியின் செயலாக்க திறன்.
- SQL-ன் கூறுகள்.
- புலங்கள் மற்றும் பதிவுகளை குறிப்பிட்டு ஒரு அட்டவணையை உருவாக்குதல்.
- ஒரு அட்டவணையில் பதிவுகளை சேர்த்தல், மேம்படுத்துதல் (புதுப்பித்தல்) மற்றும் நீக்குதல் போன்ற பல்வேறு கையாளுதலை செயற்படுத்துதல்.
- பல்வேறு நிபந்தனைகளையும் அவற்றை அட்டவணைகளில் பயன்படுத்தும் வழிமுறைகளையும் தெரிந்து கொள்ளுதல்.
- பல்வேறு உட்பிரிவுகளைப் பயன்படுத்துவதன் மூலம் அட்டவணையில் வினவல்களை உருவாக்குதல்.
- ஏற்கனவே உள்ள அட்டவணையின் கட்டமைப்பை மாற்றியமைக்க.
- ஒரு அட்டவணை மற்றும் பதிவுகளை நீக்குவதற்கான கட்டளைகளையும், செயல்படுத்தப்பட்ட கட்டளைகளை திரும்ப பெறும் கட்டளைகள் பற்றி அறிந்து கொள்ளுதல்.

12.1 SQL அறிமுகம்

வினவல் அமைப்பு மொழி (Structured Query Language) என்பது தரவு தளங்களை அணுகுதலுக்கும், கையாளுதலுக்கும் பயன்படுத்தப்படும் ஒரு வினவல் மொழியாகும். SQL பயனருக்கு தரவுதளங்களை உருவாக்குதல், திரும்ப எடுத்தல், மாற்றி அமைத்தல் மற்றும் தகவல்களை பரிமாற்றம் செய்தல் போன்ற செயற்பாடுகளை செயற்படுத்த அனுமதிக்கின்றது. உறவுநிலை தரவுதள மேலாண்மை அமைப்பில் (Relational DataBase Management System) உள்ள தரவுகளை நிர்வகிப்பதற்கும், அணுகுவதற்கும் வடிவமைக்கப்பட்ட ஒரு மொழியாகும்.

SQL பல பதிப்புகள் கொண்டது. SQL-ன் அசல் பதிப்பு IBMன் ஆய்வு மையத்தில் உருவாக்கப்பட்டது. மேலும், இது 1970 தொடக்கத்தில் Sequel என்று அழைக்கப்பட்டது. பின்னர், SQL என மாற்றப்பட்டது. American National Standard Institute (அமெரிக்க தேசிய தரநிலை நிறுவனம்) நிறுவனத்தால் 1986-ல் வெளியிடப்பட்ட SQL தரநிலை, 1992ல் புதுப்பிக்கப்பட்டு, SQLன் அண்மை பதிப்பு 2008 ல் வெளியிடப்பட்டு, SQL 2008 என பெயரிடப்பட்டுள்ளது.

Chapter 12.indd 208 30-01-2020 20:45:16

குறிப்பு

SQL ன் தற்போதைய அண்மைகால பதிப்பு, 2008 ஆம் ஆண்டு வெளியிடப்பட்டுள்ளது.

12.2 உறவுநிலை தரவுதள மேலாண்மை அமைப்பில் SQL -ன் பங்கு

RDBMS என்பதன் விரிவாக்கம் Relational DataBase Management System என்பதாகும். Oracle, MySQL, MS SQL Server, IBM DB2 மற்றும் மைக்ரோ சாப்ட் Access போன்றவை உறவுநிலை தரவுதள மேலாண்மை அமைப்பதற்கான RDBMS மென்பொருள் தொகுப்புகளாகும். அத்தகைய தரவுத்தளங்களுடன் தொடர்புக் கொண்டு தரவுகளை கையாள SQL மொழி பயன்படுகின்றது.

பொதுவாக, **தரவுத்தளம்** என்பது தொடர்புடைய தரவுகளை கொண்ட அட்டவணையின் தொகுப்பைக் கொண்ட ஒரு **களஞ்சியம்** எனப்படும். பிற பயன்பாடுகள் பயன்படுத்தும் வகையில் இவற்றை வினவல்கள் கொண்டு செயற்படுத்தி கொள்ளலாம். தரவுத்தள மேலாண்மை அமைப்பு தரவுதளத்தை உருவாக்குதல், நிர்வகிக்கும் திறன் மற்றும் தரவுத்தள பணித்தளத்தினை பயன்படுத்தும் திறன் போன்றவற்றை ஆதரிக்கிறது. RDBMS என்பது தொடர்புடைய தரவு கூறுகளை இணைக்கும் வரிசை அடிப்படையிலான அட்டவணை வடிவமைப்பைக் கொண்ட ஒரு வகை தரவுத்தள மேலாண்மை அமைப்பாகும். மேலும், Create, Read, Update மற்றும் Delete செயற்பாடுகளை ஒட்டுமொத்தமாக CRUD எனப்படும்.

RDBMS-ல் உள்ள தரவுகள், தரவுத்தள பொருட்களாக **அட்டவணைகளில்** (Tables) சேமிக்கப்படும். ஒரு அட்டவணை என்பது தொடர்புடைய தரவுப் பதிவுகளின் தொகுப்பாகும், அட்டவணை வரிசைகள் மற்றும் நெடுவரிசைகளை கொண்டிருக்கும்.

ஒரு அட்டவணையின் நெடுவரிசை **புலம்** (field) என்று குறிப்பிடப்படுகிறது, இது ஒவ்வொரு பதிவின் தேவையான தகவல்களை பராமரிக்கும் வகையில் வடிவமைக்கப்பட்டுள்ளது. ஒரு அட்டவணையின் நெடுவரிசையை ஒரு குறிப்பிட்ட புலத்தின் தொடர்புடைய அனைத்து தகவல்களையும் கொண்டிருக்கும். ஒரு மாணவர் அட்டவணையில் புலங்கள் AdmNo, studName, studAge, studclass, place ஆகிய புலங்களைக் கொண்டிருக்கலாம்.

ஒரு அட்டவணையில் உள்ள தொடர்புடைய புலங்கள் அல்லது நெடுவரிசைகளின் தொகுப்பைக் கொண்ட ஒரு வரிசை, **பதிவு** (Record) எனப்படும். ஒரு பதிவு என்பது மாணவர் அட்டவணையில் உள்ள ஒரு குறிப்பிட்ட மாணவரின் விவரங்களை கொண்டிருக்கும் ஒரு வரிசையாகும்.

12.3 SQL-ன் செயலாக்க திறன்கள்

SQLன் பல்வேறு செயலாக்க திறன்கள் :

1. **தரவு வரையறை மொழி** (DDL-**Data Definiton Language**) : SQL DDL-அட்டவணைகளை (Relational) உறவுநிலை திட்டங்களை (கட்டமைப்பு) வரையறுத்தல், நீக்குதல், மாற்றியமைத்தல் மற்றும் குறிப்புகளை உருவாக்கத்திற்கு தேவையான கட்டளைகளை வழங்கும் ஒரு வரையறை மொழியாகும்.
2. **தரவை கையாளுதல் மொழி (DML-Data Manipulation Language)** : SQL DML தரவுத்தளங்களில் வரிசைகளை (tuples)-ஐ சேர்க்கவும், நீக்கவும், மாற்றவும் கட்டளைகளை உள்ளடக்கியது.
3. **உட்பொதிக்கப்பட்ட தரவுக் கையாளுதல் மொழி** (EDML-**Embedded Data Manipulation Language**) **: SQLன்** உட்பொதிந்த வடிவம், உயர்நிலை நிரலாக்க மொழிகளில் பயன்படுத்தப்படுகிறது.

Chapter 12.indd 209 30-01-2020 20:45:16

4. **View வரையறை** : SQL அட்டவணையை பார்வையிடுவதற்கான View கட்டளைகளைக் கொண்டுள்ளது.

5. **அங்கீகாரம்** : SQL உறவுநிலைக்கான அணுகல் உரிமைகளுக்கான கட்டளைகளையும், அட்டவணைகளை பார்வையிடுதலுக்கான கட்டளைகளையும் கொண்டுள்ளது.

6. **ஒருமைப்பாடு** : SQL நிபந்தனையை பயன்படுத்தி ஒருமைப்பாட்டை சரிபார்த்தலுக்கு படிவங்களை வழங்குகிறது.

7. **பரிவர்த்தனை கட்டுப்பாடு** : கோப்பு பரிமாற்றங்களுக்கான கட்டளைகள் மற்றும் பரிவர்த்தனை செயலாக்கத்தின் கட்டுப்பாட்டையும் SQL கொண்டுள்ளது.

உங்களுக்குத் தெரியுமா SQL(Structured Query Language)-வினவல் அமைப்பு மொழி தரவுத்தளங்களை அணுகுதலுக்கான மொழி எனில், MySQL என்பது SQL Server, Oracle, Informix, Postgres, etc. போன்று ஒரு தரவுத்தள மேலாண்மை அமைப்பாகும். MySQL ஒரு உறவுநிலை தரவுத்தள மேலாண்மை அமைப்பு (RDBMS).

MYSQL- நிறுவுதல் இணைப்பு 1-ல் காண்க

12.4 ஒரு தரவுத்தளத்தை உருவாக்குதல்

1. ஒரு தரவுத்தளத்தை உருவாக்க, பின்வரும் கட்டளையை தூண்டுகுறியில் உள்ளிடவும்:

   CREATE DATABASE database_name;

   தரவுத்தளத்தின் பெயர்

   (எடுத்துக்காட்டாக) அட்டவணைகளை சேமித்து வைக்க ஒரு தரவுத்தளத்தை உருவாக்க:

   CREATE DATABASE stud;

2. தரவுத்தளத்துடன் செயலாற்ற, பின்வரும் கட்டளையை உள்ளிடவும்.

   USE DATABASE;

   எடுத்துக்காட்டாக, உருவாக்கப்பட்ட stud என்ற தரவுத்தளத்தை பயன்படுத்த கொடுக்க வேண்டிய கட்டளை

   USE stud;

TUJ62X

உங்களுக்குத் தெரியுமா WAMP என்பதன் விரிவாக்கம் "Windows, Apache, MySQL மற்றும் PHP". WAMP எனப்படும் சொல்லானது விண்டோஸ், அப்பாச்சி, MySQL, PHP என்ற சொற்களின் முதல் எழுத்துக்களை எடுத்து உருவாக்கப்பட்ட சுருக்கப் பெயர் ஆகும். இணைய வளர்ச்சிக்கும், உட்புற சோதனைக்கும் இது பயன்படுத்தப்படுகிறது. இணையதள நேரலை சேவைகளுக்கு பயன்படுத்திக் கொள்ளலாம்.

12.5 SQL-ன் கூறுகள்

SQL- கட்டளைகள் ஐந்து பொதுப்பிரிவுகளாக பிரிக்கப்பட்டுள்ளது:

Chapter 12.indd 210 30-01-2020 20:45:16

- DML - தரவு கையாளுதல் மொழி
- DDL - தரவு வரையறை மொழி
- DCL - தரவு கட்டுப்பாட்டு மொழி
- TCL - பரிவர்த்தனை கட்டுப்பாட்டு மொழி
- DQL - தரவு வினவல் மொழி

12.5.1 தரவு வரையறை மொழி

தரவு வரையறை மொழி (DDL) தரவுத்தள அமைப்பு அல்லது திட்ட வடிவமைப்பினை வரையறுக்கும் SQL கூற்றுகளை கொண்டிருக்கும். இது தரவுத்தள திட்ட வடிவமைப்பின் விளக்கங்களுடன் செயல்படுகிறது. மேலும், தரவுத்தளங்களில் உள்ள தரவுத்தள உறுப்புகளில் கட்டமைப்பை உருவாக்குவதற்கும், மாற்றியமைப்பதற்கும் பயன்படுகிறது.

DDL தரவுத்தள அமைப்பு சேமிப்பு வடிவமைப்பையும், அணுகல் முறைகளையும் குறிப்பிடப் பயன்படும் வரையறைகளை வழங்குகிறது.

ஒரு DDL கீழ்க்காணும் செயல்பாடுகளை செயல்படுத்துகிறது :

1. தரவு உருப்படி, பிரிவு, பதிவு மற்றும் தரவுத்தள கோப்பு போன்ற தரவு பிரிவு வகைகளை அடையாளம் காண வேண்டும்.
2. ஒவ்வொரு தரவு உருப்படி, பதிவு, கோப்பு மற்றும் தரவுத்தள வகைகளுக்கு ஒரு தனித்தன்மையான பெயரை கொடுக்கிறது.
3. இது சரியான தரவு வகையை குறிப்பிட வேண்டும்.
4. இது தரவு உருப்படியின் அளவை வரையறுக்க வேண்டும்.
5. ஒரு தரவு உருப்படி பயன்படுத்தக்கூடிய மதிப்புகளின் வரம்பை இது வரையறுக்கலாம்.
6. அங்கீகரிக்கப்படாத தரவு பதிவுகளை தடுப்பதற்கான தனியுரிமையை குறிப்பிடலாம்.

தரவு வரையறை மொழியின் கீழ்வரும் SQL கட்டளைகள் பின்வருமாறு:

CREATE	தரவுத்தளத்தில் அட்டவணைகளை உருவாக்க
ALTER	தரவுத்தளத்தின் வடிவமைப்பை மாற்றியமைக்க
DROP	தரவுத்தளத்தில் உள்ள அட்டவணைகளை நீக்க
TRUNCATE	ஒரு அட்டவணையில் உள்ள அனைத்து பதிவுகளையும் அழிக்கும். மேலும், அவற்றுக்கென ஒதுக்கப்பட்ட இடத்தையும் விடுவிக்கும்.

Chapter 12.indd 211 30-01-2020 20:45:16

12.5.2 தரவு கையாளுதல் மொழி

தரவு கையாளுதல் மொழி (DML) என்பது ஒரு தரவுத்தளத்தில் உள்ள தரவுகளை சேர்த்தல், அழித்தல் மற்றும் மாற்றியமைப்பதற்காக பயன்படும் ஒரு கணிப்பொறி நிரலாக்கு மொழியாகும். SQL-ல் தரவு கையாளுதல் மொழி தரவுத்தள அட்டவணையில் சேமித்து வைக்கப்பட்டுள்ள தரவுகளை மாற்றியமைக்க பயன்படும் SQL தரவு மாற்று கூற்றுகளை கொண்டிருக்கும்.

தரவுத்தள திட்ட வடிவமைப்பின்படி, தரவுத்தளம் உருவாக்கிய பின்னர், DML ல் குறிப்பிட்டுள்ள செயல்முறைகளின் தொகுப்பை பயன்படுத்தி தரவுகளை கையாளலாம்.

தரவுகளை கையாளுதல் என்பது,

- தரவுத்தளத்தில் புதிய தகவலை சேர்த்தல்
- தரவுத்தளத்தில் சேமித்து வைக்கப்பட்டுள்ள தகவல்களை திரும்ப பெறுதல்.
- தரவுத்தளத்திலிருந்து வேண்டாத தகவல்களை நீக்குதல்.
- தரவுத்தளத்தில் சேமித்து வைக்கப்பட்டுள்ள தரவுகளை மாற்றி அமைத்தல்.

DML (தரவு கையாளுதல் மொழி) அடிப்படையில் இரண்டு வகைப்படும்:

செயல்முறையுடனான DML – தேவைப்படும் தரவையும், அதனை பெறும் வழிமுறையும் பயனரால் குறிப்பிடப்பட வேண்டும்.

செயல்முறை அல்லாத DML - பயனர் தேவைப்படும் தரவை மட்டும் குறிப்பிட்டால் போதுமானது அதனை பெற தேவையான வழி முறையை குறிப்பிடத் தேவையில்லை.

SQL-லில் உள்ள தரவு கையாளுதல் மொழியின் கட்டளைகள்:

INSERT	ஒரு அட்டவணையில் தரவுகளை நுழைத்தல்.
UPDATE	அட்டவணையில் ஏற்கனவே உள்ள தரவுகளை புதுப்பித்தல்.
DELETE	அட்டவணையில் உள்ள அனைத்து பதிவுகளையும் நீக்கும். ஆனால் அவற்றிற்கு ஒதுக்கப்பட்ட நினைவக பகுதியை விடுவிக்காது.

12.5.3 தரவு கட்டுப்பாட்டு மொழி

ஒரு தரவு கட்டுப்பாடு மொழி என்பது தரவுத்தளத்தில் சேமித்து வைக்கப்பட்டுள்ள தரவுகளின் அணுகுதலை கட்டுப்படுத்தும் ஒரு நிரலாக்க மொழியாகும். இது தரவுத்தளத்தை அணுகுவதற்கான சிறப்புரிமையை கட்டுப்படுத்துகிறது (அங்கீகாரம்). இத்தகைய சிறப்புரிமைகள் வரிசைத்தொடர்களை உருவாக்கல், அட்டவணைகளை பார்வையிடுதல் போன்ற அனைத்து தரவுத்தள செயல்பாடுகளை செயல்படுத்துவதைக் குறிக்கும்.

SQL-லில் உள்ள தரவு கட்டுப்பாட்டு மொழியின் கட்டளைகள்:

GRANT	ஒரு குறிப்பிட்ட பணியை செய்ய ஒன்று அல்லது அதற்கு மேற்பட்ட பயனர்களுக்கு அனுமதி வழங்கும்.
REVOKE	Grant கட்டளையினால் தரப்பட்டட அணுகல் அனுமதி திரும்பப் பெறப்படும்.

Chapter 12.indd 212 30-01-2020 20:45:16

12.5.4 பரிவர்த்தனைக்கான கட்டுப்பாட்டு மொழி

தரவுத்தளத்தில் உள்ள பரிவர்த்தனைகளை (TCL) நிர்வகிக்க பரிவர்த்தனை கட்டுப்பாட்டு மொழி பயன்படுகிறது. DML கூற்றுகளின் மூலம் அட்டவணையில் உள்ள தரவுகளுக்கு செய்யப்படும் மாற்றங்களை நிர்வகிக்க அவை பயன்படுத்தப்படுகின்றன.

SQL – லில் உள்ள பரிவர்த்தனை கட்டுப்பாட்டு மொழியின் கட்டளைகள்:

Commit	தரவுத்தள பரிவர்த்தனையை நிரந்தரமாக சேமிக்கும்.
Roll back	ஒரு தரவுத்தளத்தை முந்தைய commit நிலைவரை மீட்டெடுக்கும்.
Save point	Rollback செய்வதற்கு ஏதுவாக தரவுத்தள பரிவர்த்தனையை தற்காலிமாக சேமிக்கும்.

12.5.5 தரவு வினவல் மொழி

தரவு வினவல் மொழி ஒரு தரவுத்தளத்தில் உள்ள தரவுகளை வினவுதலுக்கும், மீட்டெடுப்பதற்கான கட்டளைகளை கொண்டுள்ளது. தரவு வினவல் மொழியில் உள்ள அத்தகைய ஒரு SQL கட்டளை SELECT ஆகும்.

Select	அட்டவணையிலுள்ள பதிவுகளை வெளிக்காட்டும்.

12.6 தரவு வகைகள்

தரவுத்தளத்தில் தரவுகள், அவற்றில் இருத்தி வைக்கப்படும் மதிப்புகளின் அடிப்படையில் சேமிக்கப்படும். இது தரவின் தரவு வகையாக அடையாளப் படுத்தப்படுகிறது (அல்லது) ஒவ்வொரு புலமும் ஒரு தரவு வகையைச் சேர்ந்ததாக இருக்கலாம். கொடுக்கப்பட்ட புலத்தில் உள்ள அனைத்து மதிப்புகளையும் ஒரே வகையாக இருக்க வேண்டும்.

SQL ன் ANSI தரநிலை உரை மற்றும் எண் தரவு வகைகளை மட்டுமே அங்கீகரிக்கிறது. ஆனால் சில தனியார் நிரல்கள் தேதி மற்றும் நேரம் போன்ற தரவு வகைகளை பயன்படுத்துகின்றன. ANSI தரவு வகைகள் கீழே அட்டவணை 12.1-ல் பட்டியலிடப்பட்டுள்ளன.

தரவு வகை	விளக்கம்
char (Character)	நிலையான அகலம் கொண்ட சர மதிப்பு. இந்த வகையின் மதிப்பு ஒற்றை மேற்கொள் குறிகளுக்குள் தரப்படும். (எ.கா.) Anu's என்பது 'Anu''S' என்று குறிப்பிடப்படும்.
varchar	மாறும் அகலம் கொண்ட குறியுரு சரம். இது char தரவு வகையை போன்று இருந்தாலும் தரவின் அளவிற்கேற்ப பதியும் அளவு மாறும்.

dec (Decimal)	15.12,0.123 போன்ற பின்ன எண்களை இது குறிக்கும். இங்கு அளவுக்கான செயலுருபு இரண்டு பகுதிகளை கொண்டுள்ளது. துல்லியம் மற்றும் அளவுகோல். அளவு செயலுருபுவில் வரும் துல்லியம் எத்தனை இலக்க எண் என்பதையும், அளவு கோல் தசமபுள்ளிக்கு வலப்பக்கம் அதிகபட்சமாக எத்தனை இலக்கங்கள் இடம் பெற வேண்டும் என்பதை குறிக்கம். size(5,2) என்பதில் துல்லியம் 5 ஆகவும் மற்றும் அளவுகோல் 2 எனவும் எடுத்துக் கொள்ளும். அளவுக்கோல் துல்லியத்தை விட அதிகமாக இருக்கக் கூடாது.
numeric	இது decimal தரவு வகை போன்றே இருக்கும். ஆனால் இலக்கங்களின் எண்ணிக்கை துல்லிய செயலுருபுவைக் காட்டிலும் அதிகமாக இருத்தல் கூடாது.
int (Integer)	இது தசமபுள்ளி இல்லாத ஒரு முழு எண்ணைக் குறிக்கும். இது size அளபுரு பயன்படுத்தாது.
smallint	இது integer வகையை போன்று காணப்பட்டாலும் கொடாநிலை அளவு integer ஐ விட சிறியதாக இருக்கும்.
float	இது பதின்ம எண் அமைப்பில் ஒரு மிதவைப் புள்ளி எண்ணை 10ன் அடுக்குகளில் கொண்டிருக்கும். மேலும், அதிகபட்சமாக துல்லியம் 64 இலக்கங்கள் வரை வரையறுக்கப்படலாம்.
real	இது float வகையைப் போன்றது. இருப்பினும், size செயலுருபை பயன்படுத்திக் கொள்ளாது. அதிகபட்சம், துல்லியம் 64 இலக்கங்கள் வரையறுக்கப்படலாம்.
double	இது real வகையைப் போன்றது. ஆனால் துல்லியம் 64 இலக்கங்களுக்கு கூடுதலாக இருக்கலாம்.

அட்டவணை *12.1*

12.7 SQLகட்டளைகளும்அதன்செயல்பாடுகளும்

தரவுகளை சேமித்து வைக்க ஒரே வழி அட்டவணை என்பதால், அனைத்து தகவல்களும் அட்டவணை வடிவில் ஒருங்கிணைக்கப்பட வேண்டும். SQL தரவுத்தளத்துடன் செயலாற்றுவதற்கான முன்வரையறுக்கப்பட்ட கட்டளைகளின் தொகுதியை வழங்குகிறது.

சிறப்புச் சொற்கள்	இவை SQL ல் சிறப்பு பொருளைக் கொண்ட கட்டளைகளாக அறியப்படுகிறது.
கட்டளைகள்	கூற்றுகள் என அறியப்படும் இவை பயனரால் தரவுத்தளத்திற்க கொடுக்கப்படும் அறிவுறுத்தல்களைக் குறிக்கும்.
Clauses	இவைச் சிறப்புச் சொல்லுடன் தொடங்கி, சிறப்புச் சொல்லையும், செயலுருபையும் கொண்டிருக்கும்.
செயலுருபுகள்	இவை clause ஐ நிறைவுப் பெற செய்ய கொடுக்கப்படும் மதிப்புகளை குறிக்கும்.

Chapter 12.indd 214 30-01-2020 20:45:16

12.7.1 DDL கட்டளைகள்

CREATE TABLE கட்டளைகள்

CREATE TABLE கட்டளையை பயன்படுத்தி ஒரு அட்டவணையை உருவாக்கலாம். ஒரு அட்டவணையை உருவாக்கும் போது, அதன் நெடுவரிசைகளுக்கு பெயரிட்டு, தரவு வகைகளையும், அளவினையும் குறிப்பிட வேண்டும். ஒவ்வொரு அட்டவணையும் குறைந்த பட்சம் ஒரு நெடுவரிசையை கொண்டிருக்க வேண்டும். CREATE TABLE கட்டளையின் தொடரியல்.

```
CREATE TABLE <table-name>
(<column name><data type>[<size>]
(<column name><data type>[<size>]......
);
```

மேற்கண்ட தொடரியலைப் பயன்படுத்தி, ஒரு வகுப்பில் பயிலும் மாணவர்களின் விவரங்களை ஒரு தரவுத்தளத்தில் சேமிக்கலாம். இதற்கு முதலில் நாம் ஒரு அட்டவணையை உருவாக்க வேண்டும். ஒரு மாணவர் குறித்த அட்டவணையில் மாணவர் தொடர்பான விவரங்களாக சேர்க்கை எண் (Admission number)- இதை சுருக்கி Admno என்றும், மாணவரின் பெயர் (Name), பாலினம் (gender), வயது (Age) போன்றவை எடுத்துக்கொள்ளலாம். Admno, Name, Gender, Age மற்றும் Place என்ற புலங்களை கொண்டுள்ள ஒரு அட்டவணையை நாம் உருவாக்கலாம்.

SQL கட்டளை பின்வருமாறு இருக்கும்:

```
CREATE TABLE Student

    (Admno    integer,
       Name  char(20),
       Gender  char(1),
       Age     integer,
       Place char(10),
       );
```

மேலே உள்ளது, எந்த கட்டுப்பாடுகளும் இல்லாத ஒரு எளிய அட்டவணை அமைப்பாகும். ஒரு அட்டவணையில் அமைந்திருக்கும் புலங்களுக்கான தரவுகளை சில கட்டுப்பாடுகளை ஏற்படுத்தி கட்டுப்படுத்தலாம். இது தரவுத்தளத்திலுள்ள தரவுகளின் துல்லியத்தன்மை மற்றும் நம்பகத்தன்மையை உறுதி செய்கிறது. கட்டுப்பாடுகள் நெடுவரிசை அல்லது அட்டவணை அளவிலானதாக இருக்கக்கூடும்.

குறிப்பு

கட்டுப்பாடுகள் என்பது ஒரு புலம் அல்லது புலங்களின் தொகுதிக்கு பயன்படுத்தக்கூடிய நிபந்தனையை குறிக்கும்.

நெடுவரிசை கட்டுப்பாடு	தனிப்பட்ட நெடுவரிசைக்கு மட்டுமே நெடுவரிசை கட்டுப்பாட்டை பயன்படுத்த முடியும்.
அட்டவணை கட்டுப்பாடு	அட்டவணை கட்டுப்பாடு ஒன்று அல்லது அதற்கு மேற்பட்ட நெடுவரிசைகளுக்கு பயன்படுத்த முடியும்.

Chapter 12.indd 215 30-01-2020 20:45:16

கட்டுப்பாட்டுடன் உருவாக்கப்பட்ட ஒரு அட்டவணையின் தொடரியல் கீழே கொடுக்கப்பட்டுள்ளது.

```
CREATE TABLE <table-name>
(<column name><data type>[<size>]<column constraint>,
(<column name><data type>[<size>]<column constraint>……
<table constraint>(<column name>,[<column name>….]).....
);
```

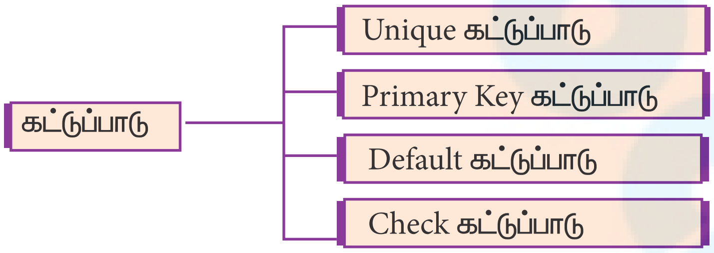
"NOT NULL" நெடுவரிசை கட்டுப்பாட்டுடன் கூடிய மாணவர் அட்டவணை கீழே தரப்பட்டுள்ளது. இந்த கட்டுப்பாடு ஒரு புலம் எப்பொழுதும் ஒரு மதிப்பைக் கொண்டிருக்க வேண்டும் என்பதை அறிவுறுத்துகிறது.

```
CREATE TABLE Student
(
Admno integer    NOT NULL PRIMARY KEY, → Primary Key constraint
Name char(20) NOT NULL,
Gender char(1),
Age integer,
Place char(10),
);
```

மேற்கண்ட கட்டளை அமைப்பை பயன்படுத்தி உருவாக்கப்பட்ட student என்ற அட்டவணையில் integer வகையை சார்ந்த Admno என்ற புலமும், char வகையை சார்ந்த Name என்ற புலமும், NOT NULL என்ற கட்டுப்பாட்டுடன் வரையறுக்கப்பட்டுள்ளது. இதன் பொருள் என்னவெனில், அந்த இரண்டு புலங்களும் கட்டாயமாக மதிப்புகளை கொண்டிருக்க வேண்டும். Gender, Age மற்றும் Place என்ற புலங்களுக்கு எந்த வித கட்டுப்பாடுகளும் இல்லை.

12.7.2 கட்டுப்பாடுகளின் வகைகள்

கட்டுப்பாடுகள் தரவுத்தள ஒருங்கிணைப்பை உறுதி செய்வதால், அவற்றை தரவுத்தள ஒருங்கிணைப்பு கட்டுப்பாடுகள் எனவும் அழைக்கலாம். கட்டுப்பாடுகள் பல வகைப்படும். அவை



12.7.2.1 Unique கட்டுப்பாடு

இந்த கட்டுப்பாடு குறிப்பிட்ட நெடுவரிசைகளில் எந்த இரு வரிசைகளும் ஒரே மதிப்பை கொண்டிருக்காது என்பதை உறுதி செய்கிறது. உதாரணத்திற்கு, student அட்டவணையில் Admno என்ற புலத்திற்கு UNIQUE கட்டுப்பாட்டை பயன்படுத்தி, ஒரே சேர்க்கை எண் இரு மாணவர்களுக்கு தரப்படாததை உறுதி செய்து கொள்ளலாம். இந்த கட்டுப்பாட்டை கீழ்க்கண்டவாறு பயன்படுத்திக் கொள்ளலாம்:

Chapter 12.indd 216 30-01-2020 20:45:16

```
CREATE TABLE Student
(
Admno integer NOT NULL UNIQUE, → Unique constraint
Name char (20) NOT NULL,
Gender char (1),
Age integer,
Place char (10),
);
```

UNIQUE கட்டுப்பாட்டை NOT NULL என்று அறிவிக்கப்பட்ட புலங்களுக்கு மட்டுமே பயன்படுத்த முடியும்.

ஒரு தனிப்பட்ட புலத்திற்கு இரு கட்டுப்பாடுகளை பயன்படுத்தினால் அப்புலம் பல கட்டுப்பாடுகளைக் கொண்ட புலம் என அழைக்கலாம். மேற்கண்ட கட்டணை அமைப்பில், NOT NULL மற்றும் UNIQUE எனும் பல கட்டுப்பாடுகள் எனும் பல கட்டுப்பாடுகள் Admno என்ற ஒரு தனித்த புலத்தில் பயன்படுத்தப்பட்டுள்ளது. கட்டுப்பாடுகள் இடைவெளிக் கொண்டு பிரிக்கப்பட்டு புல வரையறுப்பின் இறுதியில் காற்புள்ளி சேர்க்கப்படுகிறது. Admno என்ற புலத்திற்கு இந்த இரண்டு கட்டுப்பாடுகளை சேர்ப்பதால், இந்த புலம் ஏதேனும் சில மதிப்புகளைக் கொண்டிருக்க வேண்டும் அதே சமயம் இரட்டையான மதிப்புகள் இருத்தல் கூடாது.

12.7.2.2 Primary Key கட்டுப்பாடு

தரவுத்தளத்திலுள்ள ஒரு பதிவை தனித்தன்மையோடு அடையாளம் காட்ட ஒரு புலத்தினை primary key என்ற கட்டுப்பாட்டுடன் அறிவிக்க வேண்டும். இது unique கட்டுப்பாட்டினை போன்றே காணப்பட்டாலும் அட்டவணையின் ஒரு புலத்தை மட்டுமே primary key ஆக குறிப்பிட முடியும். primary key கட்டுப்பாடு வெற்று (NULL) மதிப்புகளை அனுமதிக்காது என்பதால், primary key என்ற கட்டுப்பாட்டுடன் அறிவிக்கப்படும் புலம், NOT NULL என்ற கட்டுப்பாட்டையும் கொண்டிருக்க வேண்டும்.

மாணவர் அட்டவணையில் primary key கட்டுப்பாட்டை பயன்படுத்துதலை விளக்கும் எடுத்துக்காட்டு:

```
CREATE TABLE Student
(
Admno integer    NOT NULL PRIMARY KEY, → Primary Key constraint
Name char(20) NOT NULL,
Gender char(1),
Age integer,
Place char(10),
);
```

மேற்கண்ட எடுத்துக்காட்டில், Admno என்ற புலம் primary key என்று வரையறுக்கப்பட்டுள்ளதால், ஒரு பதிவை தனித்தன்மையோடு அடையாளம் காட்ட முடிகிறது. மேலும் NOT NULL என்றும் வரையறுக்கப்பட்டுள்ளதால், இந்த புலத்தின் மதிப்பு காலியாக இருக்கக் கூடாது.

12.7.2.3 DEFAULT கட்டுப்பாடு

DEFAULT கட்டுப்பாடு ஒரு புலத்தில் முன்னியல்பு மதிப்புகளை இருத்தி வைக்க பயன்படுகிறது. DEFAULT கட்டுப்பாட்டை கொண்ட குறிப்பிட்ட புலத்திற்கு மதிப்பு கொடுக்காவிடில், தானாகவே முன்னியல்பு

Chapter 12.indd 217 30-01-2020 20:45:16

மதிப்பு அந்த புலத்தில் இருத்தப்பட்டு விடும். மாணவர் அட்டவணையில் Default கட்டுப்பாடு பயன்படுத்துதலை விளக்கும் எடுத்துக்காட்டு:

```
CREATE TABLE Student
(
Admno integer   NOT NULL  PRIMARY KEY,
Name char(20) NOT NULL,
Gender char(1),
Age integer  DEFAULT "17", → Default Constraint
Place char(10),
);
```

மேற்கண்ட எடுத்துக்காட்டில், "Age" என்ற புலத்திற்கு 17 என்று முன்னியல்பு மதிப்பு கொடுக்கப்பட்டுள்ளதால், அந்த புலத்தில் பயனரால் எந்த மதிப்பும் உள்ளிடப்படவில்லை எனில், "Age" என்ற புலம் தானாகவே 17 என்ற மதிப்பை எடுத்துக் கொள்ளும்.

12.7.2.4 Check கட்டுப்பாடு

இந்தக் கட்டுப்பாடு ஒரு புலத்திற்கான மதிப்பின் வரம்பை நிர்ணயிக்க உதவுகிறது. CHECK கட்டுப்பாட்டை ஒரு நெடுவரிசைக்கு வரையறுத்தால், அப்புலத்தில் வரையறுக்கப்பட்ட மதிப்புகளை மட்டுமே உள்ளிட அனுமதிக்கும்.

மாணவர் அட்டவணையில் CHECK கட்டுப்பாட்டை பயன்படுத்துதலை விளக்கும் எடுத்துக்காட்டு

```
CREATE TABLE Student
(
Admno integer   NOT NULL PRIMARY KEY
Name char(20) NOT NULL,
Gender char(1),
Age integer (CHECK Age <=19), → Check Constraint
Place char(10),
);
```

மேற்கண்ட எடுத்துக்காட்டில், Age என்ற புலத்தில் CHECK கட்டுப்பாட்டை பயன்படுத்தி, Age என்ற புலத்திற்கு உள்ளிடும் மதிப்பு 19க்கு குறைவாகவோ அல்லது சம்மாகவோ இருக்க வேண்டும் என வரையறுக்கப்படுகிறது.

குறிப்பு

CHECK கட்டுப்பாடு ஒப்பீட்டு மற்றும் தருக்க செயற்குறிகளை நிபந்தனையை வரையறுக்க பயன்படுத்திக் கொள்ளலாம்.

12.7.2.5 அட்டவணை கட்டுப்பாடு

ஒரு அட்டவணையில் உள்ள புலங்களின் குழுவிற்கு கட்டுப்பாட்டைப் பயன்படுத்தினால், அதனை அட்டவணைக் கட்டுப்பாடு என்கிறோம். பொதுவாக, அட்டவணை கட்டுப்பாடு அட்டவணை வரையறுப்பின் இறுதியில் குறிப்பிடப்படும். Admno, Firstname, Lastname, Gender, Age, Place என்ற புலங்களை கொண்ட student 1 என்ற ஒரு புதிய அட்டவணையை எடுத்துக் கொள்வோம்.

Chapter 12.indd 218 30-01-2020 20:45:16

```
CREATE TABLE Student1
(
Admno integer NOT NULL,
Firstname char(20),
Lastname char(20),
Gender char(1),
Age integer,
Place char(10),
PRIMARY KEY (Firstname, Lastname) → Table constraint
);
```

மேற்கண்ட எடுத்துக்காட்டில், primary key என்று வரையறுக்கப்பட்டுள்ள Firstname, Lastname என்ற இருபுலங்கள், அட்டவணை கட்டுப்பாடுடையதாக அமைந்துள்ளன.

12.7.3 DML கட்டளைகள்

ஒரு அட்டவணையின் திட்டமைப்பு (அல்லது) வடிவமைப்பு உருவாக்கப்பட்டவுடன், அதில் தரவுகளை சேர்க்கலாம். அட்டவணையில் வரிசைகளை சேர்த்தல், நீக்குதல், புதுப்பித்தல் போன்ற செயற்பாடுகளுக்கு DML கட்டளைகள் பயன்படுகிறது.

12.7.3.1 INSERT கட்டளை

INSERT கட்டளை தரவுத்தளத்தில் புதிய தரவை சேர்ப்பதற்கும், அல்லது ஒரு அட்டவணையில் புதிய பதிவுகளை சேர்ப்பதற்கு உதவுகிறது. இந்த கட்டளையை எவ்வாறு பயன்படுத்துவது என்பது கீழே குறிப்பிடப்பட்டுள்ளது:

```
INSERT INTO <table-name> [column-list] VALUES (values);
```

INSERT INTO Student (Admno, Name, Gender, Age, Place)

VALUES (100, 'Ashish', 'M', 17, 'Chennai');

INSERT INTO Student (Admno, Name, Gender, Age, Place)

VALUES (101, 'Adarsh', 'M', 18, 'Delhi');

அட்டவணையில் இரு புதிய பதிவுகள் சேர்க்கப்பட்டுள்ளதை கீழ்காணலாம்:

Admno	Name	Gender	Age	Place
100	Ashish	M	17	Chennai
101	Adarsh	M	18	Delhi

மதிப்புகளின் வரிசைமுறை CREATE TABLE கட்டளையில் உள்ள நெடுவரிசையின் வரிசை முறையுடன் பொருந்தி இருக்க வேண்டும். அனைத்து நெடுவரிசைகளுக்கும் தரவு சேர்க்கபட வேண்டுமாயின், நெடுவரிசை பெயர்களைக் குறிப்பிடுவது கட்டாயமில்லை. மாணவர் அட்டவணையில் தரவுகளை பின்வரும் கட்டளையின் மூலமும் உள்ளிடலாம்.

INSERT INTO Student VALUES (102, 'Akshith', 'M', 17, 'Bangalore');

102	Akshith	M	17	Bangalore

மேற்கண்ட கட்டளை மாணவர் அட்டவணையில் ஒரு பதிவை சேர்க்கிறது.

ஒரு பதிவினுடைய சில நெடுவரிசைகளுக்கு மட்டும் தரவை சேர்க்க, நெடுவரிசையின் பெயரையும், அதற்கான தரவையும் கீழ்க்கண்டவாறு தரலாம்.

Chapter 12.indd 219 30-01-2020 20:45:16

INSERT INTO Student(Admno, Name, Place) VALUES (103, 'Ayush', 'Delhi');

103	Ayush	M	18	Delhi

மேற்கண்ட கட்டளை பின்வரும் பதிவினை Gender என்ற புலத்திற்கு முன்னியல்பு மதிப்பான 'M' என்றும், Age என்ற புலத்திற்கு முன்னியல்பு மதிப்பான 18 என்றும் சேர்க்கும்.

INSERT INTO Student (Admno, Name, Place) VALUES (104, 'Abinandh', 'Chennai');

104	Abinandh	M	18	Chennai

மாணவர் அட்டவணை பின்வரும் தரவைக் கொண்டிருக்கும்:

Admno	Name	Gender	Age	Place
100	Ashish	M	17	Chennai
101	Adarsh	M	18	Delhi
102	Akshith	M	17	Bangalore
103	Ayush	M	18	Delhi
104	Abinandh	M	18	Chennai

INSERT கட்டளையில் கொடுக்கப்படாத புலங்கள், முன்னியல்பு மதிப்புகள் அதற்கு வரையறுக்கப்பட்டிருந்தால் அவற்றை எடுத்துக் கொள்ளும், இல்லையேல் NULL மதிப்பை இருத்திக் கொள்ளும்.

உங்களுக்குத் தெரியுமா?

INSERT கட்டளையில், விடுப்பட்ட புலங்கள் வரையறுக்கப்பட்ட முன்னியல்பு மதிப்பையோ அல்லது NULL மதிப்பையோ கொண்டிருக்கும்.

12.7.3.2 DELETE கட்டளை

DELETE கட்டளை அட்டவணையிலிருந்து ஒன்று அல்லது அதற்கு மேற்பட்ட பதிவுகளை நீக்குகிறது. இது தனிப்பட்ட புலத்தை மட்டும் நீக்காமல், வரிசை முழுவதையும் நீக்குவதால், புலத்திற்கான செயலுருபு தேவையில்லை.

DELETE கட்டளையின் பயன்பாடு பின்வருமாறு :

DELETE FROM table-name WHERE condition;

உதாரணத்திற்கு, **104** என்ற சேர்க்கை எண் கொண்ட பதிவை நீக்குவதற்கான கட்டளை பின்வருமாறு:

DELETE FROM Student WHERE Admno=104;

104	Abinandh	M	18	Chennai

மேற்கண்ட பதிவு மாணவர் அட்டவணையிலிருந்து நீக்கப்படும்.

அட்டவணையில் உள்ள அனைத்து வரிசைகளையும் நீக்க, பயன்படும் கட்டளை :

DELETE * FROM Student;

Chapter 12.indd 220 30-01-2020 20:45:16

இப்பொழுது அட்டவணை வெறுமையாக காணப்படும். DROP கட்டளையை (12.7.4.3 பகுதியில் பார்க்கலாம்) பயன்படுத்தி. இந்த அட்டவணையை முழுவதுமாக அழிக்க முடியும்.

12.7.3.3 UPDATE கட்டளை

ஒரு தரவுத்தளத்தில் உள்ள சில அல்லது அனைத்து தரவு மதிப்புகளையும் UPDATE கட்டளை புதுப்பித்தல் செய்யும்(புதுப்பிக்கும்). அது அட்டவணையில் உள்ள ஒன்று அல்லது அதற்கு மேற்பட்ட பதிவுகளையும் புதுப்பிக்க முடியும். UPDATE கட்டளை, WHERE clause ஐ பயன்படுத்தி மாற்றப்பட வேண்டிய வரிசைகளையும், SET சிறப்புச் சொல்லைப் பயன்படுத்தி புதிய தரவையும் குறிப்பிடுகிறது. UPDATE கட்டளையை பின்வருமாறு பயன்படுத்தலாம்:

> ***UPDATE <table-name> SET column-name = value, column-name = value,… WHERE condition;***

உதாரணத்திற்கு, பின்வரும் புலங்களை புதுப்பிக்க:

UPDATE Student SET Age = 20 WHERE Place = "Bangalore";

மேற்கண்ட கட்டளை **"Bangalore"** என்ற இடத்தில் இருக்கும் மாணவர்களுடைய வயதை **20** என மாற்றி அமைக்கும். புதுப்பிக்கப்பட்ட அட்டவணை பின்வருமாறு:

Admno	Name	Gender	Age	Place
100	Ashish	M	17	Chennai
101	Adarsh	M	18	Delhi
102	Akshith	M	20	Bangalore
103	Ayush	M	18	Delhi

ஒன்றுக்கு மேற்பட்ட புலங்களை புதுப்பிக்க, SET clause பயன்படுத்தி அவற்றிற்கான மதிப்புகளை காற்புள்ளியால் பிரித்து குறிப்பிடலாம். உதாரணத்திற்கு, ஒரு மாணவர் அட்டவணையில், பல்வேறு புலங்களை புதுப்பித்தலுக்கான கட்டளை பின்வருமாறு :

UPDATE Student SET Age=18, Place = 'Chennai' WHERE Admno = 102;

102	Akshith	M	18	Chennai

மேற்கண்ட கட்டளை கீழ்க்கண்ட முறையில் பதிவை மாற்றி அமைக்கிறது.

Admno	Name	Gender	Age	Place
100	Ashish	M	17	Chennai
101	Adarsh	M	18	Delhi
102	Akshith	M	18	Chennai
103	Ayush	M	18	Delhi

Chapter 12.indd 221 30-01-2020 20:45:17

12.7.4 சில கூடுதல் DDL கட்டளைகள்

12.7.4.1 ALTER கட்டளை

ALTER கட்டளையைப் பயன்படுத்தி அட்டவணை வடிவமைப்பில் நெடுவரிசையை சேர்த்தல், ஏற்கனவே இருக்கும் புலத்திற்கு மறு பெயரிடல், எந்த ஒரு நெடுவரிசையின் தரவு வகையையோ அல்லது அளவினையோ மாற்றி அமைத்தல் நெடுவரிசையை அட்டவணையிலிருந்து நீக்குதல் போன்ற மாற்றங்களை செய்யலாம். அது கீழ்க்காணும் முறையில் பயன்படுத்தப்படுகிறது:

ALTER TABLE <table-name> ADD <column-name><data type><size>;

மாணவர் அட்டவணையில் 'char' என்ற தரவுவகை சார்ந்த "address" என்ற நெடுவரிசையை சேர்க்க, கட்டளையமைப்பு

ALTER TABLE Student ADD Address char;

அட்டவணையில் ஏற்கனவே இருக்கும் நெடுவரிசையை மாற்றியமைக்க, ALTER TABLE கட்டளை MODIFY clause உடன் பயன்படுத்தப்படுகிறது.

ALTER <table-name> MODIFY<column-name><data type><size>;

ALTER TABLE Student MODIFY Address char (25);

மேற்கண்ட கட்டளை மாணவர் அட்டவணையில் உள்ள address நெடுவரிசை இப்பொழுது 25 எழுத்துக்கள் வரை தேக்கி வைத்துக் கொள்ளும் வகையில் மாற்றி அமைக்கும்.

ALTER கட்டளை கீழ்க்காணும் முறையில் ஏற்கனவே உள்ள நெடுவரிசைக்கு மறுபெயரிட பயன்படுகிறது:

ALTER <table-name> RENAME old-column-name TO new-column-name;

உதாரணத்திற்கு, address என்ற பெயரைக் கொண்ட நெடுவரிசைக்கு city என்று மறுபெயரிட உதவும் கட்டளை :

ALTER TABLE Student RENAME Address TO City;

ஒரு நெடுவரிசை அல்லது அனைத்து நெடுவரிசைகளையும் நீக்க, ALTER கட்டளையைப் பயன்படுத்தலாம். உதாரணத்திற்கு, ALTER TABLE-ன் DROP COLUMN ஐ பயன்படுத்தி, குறிப்பிட்ட நெடுவரிசையை நீக்கலாம். கட்டளை:

ALTER <table-name> DROP COLUMN <column-name>;

மாணவர் அட்டவணையில் city என்ற நெடுவரிசையை நீக்க, பயன்படும் கட்டளை:

ALTER TABLE Student DROP COLUMN City;

12.7.4.2 TRUNCATE கட்டளை

TRUNCATE கட்டளை அட்டவணையிலுள்ள அனைத்து வரிசைகளையும் நீக்கப் பயன்படுகிறது. அட்டவணையின் வடிவமைப்பு மாறாமல், ஒதுக்கீடு செய்யப்பட்ட இடம் விடுவிக்கப்படும் TRUNCATE கட்டளையின் தொடரியல்

TRUNCATE TABLE table-name;

உதாரணத்திற்கு, மாணவர் அட்டவணையில் உள்ள அனைத்து பதிவுகளையும், அட்டவணையையும் நீக்குவதற்கான கட்டளை :

TRUNCATE TABLE Student;

மாணவர் அட்டவணை நீக்கப்பட்டு, நினைவகப்பகுதி விடுவிக்கப்படுகிறது.

12.7.4.3 DROP TABLE கட்டளை

DROP TABLE கட்டளை தரவுத்தளத்திலிருந்து ஒரு அட்டவணையை நீக்கப் பயன்படுகிறது. ஒரு அட்டவணையை **DROP** செய்தால், அட்டவணையில் உள்ள அனைத்து வரிசைகளும் நீக்கப்பட்டு, அட்டவணையின் வடிவமைப்பு தரவுத்தளத்திலிருந்து நீக்கப்படும். அட்டவணையை **DROP** செய்த பின்பு, அதனை திரும்ப பெற இயலாது. ஆதலால், **DROP TABLE** கட்டளையை பயன்படுத்தும் போது எச்சரிக்கையாக இருத்தல் வேண்டும். ஒரு அட்டவணையை **DROP** செய்ய வேண்டுமெனில், அது காலி அட்டவணையாக இருக்க வேண்டும். அட்டவணையின் அனைத்து வரிசைகளையும் **DELETE** கட்டளை கொண்டு நீக்கப்பட்டிருக்க வேண்டும்.

அனைத்து வரிசைகளையும் நீக்க, தேவையான கட்டளை:

DELETE * FROM Student;

அனைத்து வரிசைகளும் நீக்கப்பட்டப் பின்பு, கீழ்க்காணும் முறையில் DROP TABLE கட்டளையை பயன்படுத்தி அட்டவணையை நீக்கலாம்.

DROP TABLE table-name;

உதாரணத்திற்கு மாணவர் அட்டவணையை நீக்க:

DROP TABLE Student;

DELETE, TRUNCATE AND DROP கூற்று:

DELETE	**DELETE** கட்டளை **WHERE** clause-ல் கொடுக்கப்பட்டிருக்கும் நிபந்தனையின் அடிப்படையில், அட்டவணையின் குறிப்பிட்ட வரிசைகளை மட்டும் நீக்கும் அல்லது எந்த வித நிபந்தனையும் கொடுக்கப்படவில்லையெனில், அட்டவணையின் அனைத்து வரிசைகளையும் நீக்கி விடும். ஆனால் அட்டவணைக்கென ஒதுக்கப்பட்ட இடத்தை விடுவிக்காது.
TRUNCATE	TRUNCATE கட்டளை அட்டவணையின் அனைத்து வரிசைகளையும் நீக்கி, வடிவமைப்பை மாற்றாமல், அட்டவணைக்கென ஒதுக்கீடு செய்யப்பட்ட இடத்தை விடுவிக்கும்.
DROP	DROP கட்டளை தரவுத்தளத்திலிருந்து ஒரு உறுப்பை நீக்க பயன்படுகிறது. ஒரு அட்டவணையை DROP செய்தால், அட்டவணையில் உள்ள அனைத்து வரிசைகளும் நீக்கப்பட்டு, அட்டவணையின் வடிவமைப்பு தரவுத்தளத்திலிருந்து நீக்கப்படும். ஒரு அட்டவணையை DROP செய்துவிட்ட பின்பு, அதனை திரும்பப் பெற இயலாது.

12.7.5 DQL கட்டளை – SELECT கட்டளை

SQL-ல் செயலாற்றும் போது, வினவல்களை உருவாக்குவதும், தரவை மீட்டெடுப்பதும் மிக முக்கியமான பணிகளில் ஒன்றாகும். ஒரு வினவல் என்பது தரவுத்தள அட்டவணையிலிருந்து நாம் விரும்பும் தீர்வை பெற கொடுக்கப்படும் கட்டளையாகும். SQL SELECT கட்டளை, தரவுத்தள அட்டவணையில் உள்ள

Chapter 12.indd 223 30-01-2020 20:45:17

தரவினை வினவலாகவோ அல்லது மீட்டெடுக்கவோ பயன்படும் கட்டளையாகும். இது ஒன்று அல்லது அதற்கு மேற்பட்ட அட்டவணைகளிலிருந்து பதிவுகளின் துணைக் கூறுகளை மீட்டெடுக்கப் பயன்படுகிறது. SELECT கட்டளையை பல வடிவங்களில் பயன்படுத்தலாம்.

SELECT கட்டளையின் தொடரியல் :

SELECT <column-list>FROM<table-name>;

- table-name என்பது தகவல் பெறப்பட்ட அட்டவணையின் பெயர்.
- Column-list என்பது தரவு பெறப்பட்ட ஒன்று அல்லது அதற்கு மேற்பட்ட நெடுவரிசைகளை உள்ளடக்கியது.

உதாரணமாக மாணவர் அட்டவணையில் இருந்து சேர்க்கை எண் மற்றும் மாணவர்களின் பெயரை மட்டுமே பார்க்க, கட்டளை பின்வருமாறு கொடுக்கப்பட்டுள்ளது:

மாணவர் அட்டவணை பின்வரும் தரவைக் கொண்டிருந்தால்:

Admno	Name	Gender	Age	Place
100	Ashish	M	17	Chennai
101	Adarsh	M	18	Delhi
102	Akshith	M	17	Bangalore
103	Ayush	M	18	Delhi
104	Abinandh	M	18	Chennai
105	Revathi	F	19	Chennai
106	Devika	F	19	Bangalore
107	Hema	F	17	Chennai

SELECT Admno, Name FROM Student;

மேற்கண்ட SELECT கட்டளை பின்வரும் தகவலை வெளிக்காட்டும்:

Admno	Name
100	Ashish
101	Adarsh
102	Akshith
103	Ayush
104	Abinandh
105	Revathi
106	Devika
107	Hema

அட்டவணையின் அனைத்து புலங்களையும், வரிசைகளையும் பார்க்க, SELECT கட்டளை இவ்வாறு அமையும்.

SELECT * FROM STUDENT;

Chapter 12.indd 224 30-01-2020 20:45:17

12.7.5.1 சிறப்புச் சொல் DISTINCT

DISTINCT சிறப்புச் சொல்லுடன் SELECT கட்டளையைப் பயன்படுத்தி ஒரு அட்டவணையில் இரட்டிப்பு வரிசைகளை உருவாகாமல் தடுக்கலாம். இது தரவு இரட்டித்தலைத் தவிர்க்கின்றது. உதாரணத்திற்கு,

SELECT * DISTINCT place FROM student; என்ற கட்டளை பின்வரும் தரவை வெளிக்காட்டும்:

SELECT * FROM Student;

Place
Chennai
Bangalore
Delhi

மேற்கண்ட வெளியீட்டில், place என்னும் புலத்தில் இரட்டிப்பு வரிசைகள் இல்லாத்தைக் காணலாம். DISTINCT என்ற சிறப்புச் சொல்லை பயன்படுத்தும் போது NULL மதிப்புகள் அதிக எண்ணிக்கையில் இருப்பினும், ஒரே ஒரு NULL மதிப்பு மட்டுமே திருப்பி அனுப்பப்படும்.

12.7.5.2 சிறப்புச் சொல் ALL

ALL என்ற சிறப்புச் சொல் இரட்டிப்பு வரிசைகளையும் இருத்தி வைக்கிறது. அது இரட்டிப்பு பதிவுகளை கருத்தில் கொள்ளாது, அட்டவணையின் ஒவ்வொரு வரிசையையும் வெளிக்காட்டும்.

SELECT ALL Place FROM Student;

மேற்கண்ட கட்டளை அட்டவணையில் உள்ள ஒவ்வொரு வரிசையிலும் உள்ள place என்ற புலத்தின் அனைத்து மதிப்புகளையும்(இரட்டிப்புப் பதிவுகளையும் உள்ளடக்கி) வெளிக்காட்டும்.

Place
Chennai
Delhi
Bangalore
Delhi
Chennai
Chennai
Bangalore
Chennai

SELECT கட்டளையிலுள்ள **WHERE** clause, விரும்பிய தீர்வை பெறுவதற்கான விதிமுறைகளை குறிப்பிடுகிறது. WHERE clause யுடனான SELECT கட்டளையின் பொது வடிவம்:

SELECT <column-name>[,<column-name>,….] FROM <table-name>WHERE condition>;

Chapter 12.indd 225 30-01-2020 20:45:17

உதாரணத்திற்கு, மாணவர்களின் சேர்க்கை எண்ணையும், சென்னையை சேர்ந்த மாணவர்களின் பெயர்களை மட்டும் வெளிக்காட்ட, SELECT கட்டளை கீழ்க்காணும் முறையில் பயன்படுத்தப்படுகிறது:

SELECT Admno, Name, Place FROM Student WHERE Place =”Chennai”;

Admno	Name	Place
100	Ashish	Chennai
104	Abinandh	Chennai
105	Revathi	Chennai
107	Hema	Chennai

SELECT Admno, Name, Age FROM Student WHERE Age >= 18;

Admno	Name	Age
101	Adarsh	18
103	Ayush	18
104	Abinandh	18
105	Revathi	19
106	Devika	19

ஒப்பீட்டு இயக்கிகள் =, <, <=, >, >=, <> போன்றவை இரண்டு மதிப்புகளை ஒப்பீட SELECT கட்டளையுடன் WHERE clause பயன்படுத்தலாம். WHERE clauseல் தேடல் நிபந்தனைகளை OR, AND மற்றும் NOT போன்ற தருக்க செயற்குறிகள் ஒன்றிணைக்க பயன்படுகின்றன. உதாரணத்திற்கு,

SELECT Admno, Name, Age, Place FROM Student WHERE (Age>=18 AND Place = "Delhi");

Admno	Name	Age	Place
101	Adarsh	18	Delhi
103	Ayush	18	Delhi

The **SELECT** கட்டளையை கீழ்க்காணும் முறைகளிலும் பயன்படுத்தலாம்:

SELECT Admno, Name, Age, Place FROM Student WHERE (Age>=18 OR Place ="Delhi");

SELECT Admno, Name, Place FROM Student WHERE (NOT Place ="Delhi");

12.7.5.3 BETWEEN மற்றும் NOT சிறப்புச் சொற்கள்

BETWEEN என்ற சிறப்புச் சொல் மதிப்புகளின் பரப்பை வரையறுக்கின்றது, இந்த பரப்பினுள் உள்ள மதிப்புகள் கொண்ட வரிசைகளை உள்ளடக்கும். மதிப்புகளின் பரப்பான அதிகபட்ச மதிப்பு மற்றும் குறைந்தபட்ச மதிப்பிற்கு இடையே உள்ள நிபந்தனையை உள்ளடக்கும்.

SELECT Admno, Name, Gender, Age, Place FROM Student WHERE Age BETWEEN 18 AND 19;

Admno	Name	Gender	Age	Place
101	Adarsh	M	18	Delhi
103	Ayush	M	18	Delhi
104	Abinandh	M	18	Chennai
105	Revathi	F	19	Chennai
106	Devika	F	19	Bangalore

Chapter 12.indd 226 30-01-2020 20:45:17

BETWEEN இயக்கியின் எதிர் இயக்கியாக NOT BETWEEN செயல்படும், நிபந்தனையை நிறைவு செய்யாத பதிவுகளை இது வெளிக்காட்டும்.

SELECT Admno, Name, Age FROM Student WHERE Age NOT BETWEEN 18 AND 19;

Admno	Name	Age
100	Ashish	17
102	Akshith	17
107	Hema	17

12.7.5.4 IN சிறப்புச் சொல்

IN சிறப்புச் சொல் பதிவின் மதிப்புகளுடன் பொருந்தக்கூடிய வகையில் மதிப்புகளின் பட்டியலைக் குறிப்பிடப் பயன்படுகிறது. வேறு சொற்களில் கூறுவதெனில், இது ஒன்றுக்கு மேற்பட்ட மதிப்புகளுடன் ஒரு நெடுவரிசையை ஒப்பிடப் பயன்படுகிறது. இது OR நிபந்தனைக்கு நிகரானது.

உதாரணத்திற்கு :

SELECT Admno, Name, Place FROM Student WHERE Place IN ("Chennai", "Delhi");

Admno	Name	Place
100	Ashish	Chennai
101	Adarsh	Delhi
103	Ayush	Delhi
104	Abinandh	Chennai
105	Revathi	Chennai
107	Hema	Chennai

NOT IN சிறப்புச் சொல் பட்டியலுடன் பொருந்தாத மதிப்புகளைக் கொண்ட பதிவுகளை மட்டுமே வெளிக்காட்டும். உதாரணத்திற்கு,

SELECT Admno, Name, Place FROM Student WHERE Place NOT IN ("Chennai", "Delhi");

என்ற கட்டளை "Chennai" மற்றும் "Delhi" ஆகிய இடங்களை தவிர மற்ற மாணவர்களின் பட்டியலை வெளியிடும்.

Admno	Name	Place
102	Akshith	Bangalore
106	Devika	Bangalore

NULL மதிப்பு :

ஒரு அட்டவணையில் NULL மதிப்பையுடைய புலத்தை WHERE clauseல் **IS NULL** என்பதைப் பயன்படுத்தி தேடலாம். உதாரணத்திற்கு, age என்ற புலத்தில் மதிப்புகள் கொடுக்கப்படாத அனைத்து மாணவர்களின் பட்டியலை வெளியிட, பயன்படுத்த வேண்டிய கட்டளை:

SELECT * FROM Student WHERE Age IS NULL;

குறிப்பு

IS NOT NULL என்பதை பயன்படுத்தி அட்டவணையில் உள்ள NULL அல்லாத மதிப்புகளை பட்டியலிடலாம்.

12.7.5.5 ORDER BY clause

SQL-ன் ORDER BY clause பயன்படுத்தி தரவுகளை ஒன்று அல்லது அதற்கு மேற்பட்ட நெடுவரிசைகளின் அடிப்படையில் ஏறுவரிசையிலோ அல்லது இறங்கு வரிசையிலோ வரிசைப்படுத்தலாம்.

1. கொடாநிலையாக, **ORDER BY** தரவை ஏறுவரிசையில் வரிசைப்படுத்தும்.
2. தரவினை **DESC** என்ற சிறப்புச் சொல்லைப் பயன்படுத்தி இறங்கு வரிசையிலும்,

ASC என்ற சிறப்புச் சொல்லைப் பயன்படுத்தி ஏறுவரிசையிலும் வரிசைப்படுத்தலாம்.

ORDER BY ஐ இவ்வாறு பயன்படுத்தலாம் :

```
SELECT <column-name>[,<column-name>,….] FROM <table-name>ORDER BY
<column1>,<column2>,…ASC| DESC ;
```

உதாரணத்திற்கு :

மாணவர்களின் பெயரை அகர வரிசைப்படுத்திய பட்டியலை வெளியிட பயன்படும் கட்டளையமைப்பு

SELECT * FROM Student ORDER BY Name;

வரிசைப்படுத்தப்பட்ட மாணவர் அட்டவணை பின்வருமாறு:

Admno	Name	Gender	Age	Place
104	Abinandh	M	18	Chennai
101	Adarsh	M	18	Delhi
102	Akshith	M	17	Bangalore
100	Ashish	M	17	Chennai
103	Ayush	M	18	Delhi
106	Devika	F	19	Bangalore
107	Hema	F	17	Chennai
105	Revathi	F	19	Chennai

குறிப்பு

ORDER BY clause பயன்படுத்தினால் அட்டவணைக்கு எந்த பாதிப்பையும் ஏற்படுத்தாது.

12.7.5.6 WHERE clause

WHERE clause பதிவுகளை வடிகட்டுதலுக்கு பயன்படுகிறது. இது கொடுக்கப்பட்ட நிபந்தனையை நிறைவு செய்யும் பதிவுகளை மட்டும் பிரித்தெடுக்க உதவுகிறது. உதாரணத்திற்கு, மாணவர் அட்டவணையில்

Chapter 12.indd 228 30-01-2020 20:45:17

18 மற்றும் அதற்கு மேற்பட்ட வயதுடைய மாணவர்களின் பட்டியலை, அகர வரிசைப்படுத்தி ஏறுவரிசையில் வெளியிடும் கட்டளை பின்வருமாறு:

SELECT * FROM Student WHERE Age>=18 ORDER BY Name;

Admno	Name	Gender	Age	Place
104	Abinandh	M	18	Chennai
101	Adarsh	M	18	Delhi
103	Ayush	M	18	Delhi
106	Devika	F	19	Bangalore
105	Revathi	F	19	Chennai

18 மற்றும் அதற்கு மேற்பட்ட வயதுடைய மாணவர்களின் பட்டியலை, அகர வரிசைப்படி இறங்கு வரிசையில், வெளியிடும் கட்டளை பின்வருமாறு :

SELECT * FROM Student WHERE Age>=18 ORDER BY Name DESC;

Admno	Name	Gender	Age	Place
105	Revathi	F	19	Chennai
106	Devika	F	19	Bangalore
103	Ayush	M	18	Delhi
101	Adarsh	M	18	Delhi
104	Abinandh	M	18	Chennai

குறிப்பு

பற்பல புலங்களின் மீதும் வரிசையாக்கம் செய்ய்யலாம்.

12.7.5.7 GROUP BY clause

SELECT கூற்றுடன் **GROUP BY** clause பயன்படுத்தி வரிசைகள் அல்லது நெடுவரிசைகளில் ஒரே மாதிரியான மதிப்புகளில் கொண்ட மாணவர்களை ஒரு தொகுதியாக பிரிக்கலாம் அல்லது ஒரு அட்டவணையை குழுக்களாக பிரிக்கலாம். உதாரணத்திற்கு ஒரு வகுப்பில உள்ள ஆண் மாணவர்கள் அல்லது பெண் மாணவிகளின் எண்ணிக்கையைத் தெரிந்துக் கொள்ள, GROUP BY ஐ பயன்படுத்தலாம்.

GROUP BY clause ன் தொடரியல்

SELECT <column-names> FROM <table-name> GROUP BY <column-name>HAVING [condition];

மேற்கண்ட கட்டளையை மாணவர் அட்டவணையில் பயன்படுத்த :

SELECT Gender FROM Student GROUP BY Gender;

மேற்கூறிய கட்டளையின் வெளியீடு பின்வருமாறு:

Gender
M
F

இரண்டு முடிவுகள் மட்டுமே திரும்ப பெறப்பட்டுள்ளது என்பதை கருத்தில் கொள்ளவும். இதன் காரணம் என்னவெனில், நாம் 'ஆண்' மற்றும் 'பெண்' என்ற இரு பாலினங்களையே பெற்றுள்ளோம். GROUP BY clause அனைத்து 'M' மாணவர்களையும் தொகுத்து, ஒரு வரிசையை மட்டும் வெளியிட்டுள்ளது. இது போன்றே 'F' மாணவர்களுக்கும் வெளியீடு அமையும்.

உதாரணத்திற்கு, மாணவர் அட்டவணையில் ஆண், பெண் மாணவர்களின் எண்ணிக்கையை கண்டறிய கீழ்காணும் கட்டளை தரப்பட்டுள்ளது :

SELECT Gender, count(*) FROM Student GROUP BY Gender;

Gender	count(*)
M	5
F	3

குறிப்பு

NULL மதிப்புகளையும் சேர்த்துக்கொள்ள COUNT உடன் * என்ற குறியீடு பயன்படுத்தப்பட்டுள்ளது.

மேற்கண்ட SELECT கூற்ற ஆண் மற்றும் பெண் மாணாக்கர்களின் எண்ணிக்கையை வெளியீடும்.

12.7.5.8 HAVING clause

SELECT கட்டளையுடன் HAVING மற்றும் GROUP BY Clause-க்களை பயன்படுத்தி குழுக்களுக்கு நிபந்தனையை இருத்தலாம் மற்றும் பயனர் ஒன்றிணைந்த செயற்கூறுகளையும் பயன்படுத்தலாம்.

உதாரணத்திற்கு, Chennai ஐ சேர்ந்த ஆண் மற்றும் பெண் மாணாக்கர்களின் எண்ணிக்கையை கண்டறிய

SELECT Gender , count(*) FROM Student GROUP BY Gender HAVING Place = 'Chennai';

Gender	count(*)
M	2
F	2

மேற்கண்ட வெளியீடு மாணவர் அட்டவணையில் சென்னையை சேர்ந்த ஆண் மற்றும் பெண் மாணாக்கர்களின் எண்ணிக்கையை காண்பிக்கிறது.

Chapter 12.indd 230 30-01-2020 20:45:17

12.7.6 TCL commands

12.7.6.1 COMMIT command

COMMIT கட்டளை, தரவுத்தளத்திலுள்ள எந்த ஒரு நடவடிக்கையும் நிரந்தரமாக சேமிக்கப்பயன்படுகிறது. **INSERT, UPDATE, DELETE** போன்ற DML கட்டளைகளைப் பயன்படுத்தி செய்யப்பட்ட மாற்றங்கள் நிரந்தரமல்ல. SQL தூண்டுகுறியில் COMMIT கட்டளையை கொடுத்த பிறகுதான், இது நிரந்தரமாக குறிக்கப்படுகிறது. COMMIT கட்டளையை கொடுத்து விட்டால், செய்யப்பட்ட மாற்றங்கள் திரும்பப் பெற இயலாது. **COMMIT** கட்டளை இவ்வாறு பயன்படுத்தப்படுகிறது.

```
COMMIT;
```

12.7.6.2 ROLLBACK கட்டளை

ROLLBACK கட்டளை கடைசி COMMIT கட்டளைக்கு பிறகு கொடுத்த அனைத்து கட்டளைகளையும் திரும்ப பெறும். இந்த கட்டளையுடன் SAVEPOINT கட்டளையும் பயன்படுத்தி ஒரு குறிப்பிட்ட savepoint இடத்திற்கு கொண்டு செல்ல முடியும்.

```
ROLL BACK TO  savepoint name;
```

12.7.6.3 SAVEPOINT கட்டளை

SAVEPOINT கட்டளை ஒரு நடவடிக்கையை தற்காலிகமாக சேமித்து வைப்பதால், நாம் தேவைப்படும் சமயங்களில் திரும்பவும் அந்த நிலைக்கு செல்லலாம். நம்முடைய அட்டவணையின் மாறுபட்ட நிலைகளை வெவ்வேறு பெயர்களை பயன்படுத்தி எந்த சமயத்திலும் சேமித்துக் கொள்ளலாம். மேலும், **ROLLBACK** கட்டளையைப் பயன்படுத்தி, மீண்டும் அந்த நிலையை சென்றடையலாம்.

```
SAVEPOINT savepoint_name;
```

பின்வரும் தரவைக் கொண்ட மாணவர் அட்டவணையில் COMMIT, SAVEPOINT மற்றும் ROLLBACK கட்டளைகளை காண்பிக்கும் எடுத்துக்காட்டு:

Admno	Name	Gender	Age	Place
105	Revathi	F	19	Chennai
106	Devika	F	19	Bangalore
103	Ayush	M	18	Delhi
101	Adarsh	M	18	Delhi
104	Abinandh	M	18	Chennai

```
INSERT INTO Student  VALUES (107, 'Beena', 'F', 20 , 'Cochin');
COMMIT;
```

Admno	Name	Gender	Age	Place
105	Revathi	F	19	Chennai
106	Devika	F	19	Bangalore

103	Ayush	M	18	Delhi
101	Adarsh	M	18	Delhi
104	Abinandh	M	18	Chennai
107	Beena	F	20	Cochin

UPDATE Student SET Name = 'Mini' WHERE Admno=105;

SAVEPOINT A;

Admno	Name	Gender	Age	Place
105	Mini	F	19	Chennai
106	Devika	F	19	Bangalore
103	Ayush	M	18	Delhi
101	Adarsh	M	18	Delhi
104	Abinandh	M	18	Chennai
107	Beena	F	20	Cochin

INSERT INTO Student VALUES(108, 'Jisha', 'F', 19, 'Delhi');

SAVEPOINT B;

Admno	Name	Gender	Age	Place
105	Mini	F	19	Chennai
106	Devika	F	19	Bangalore
103	Ayush	M	18	Delhi
101	Adarsh	M	18	Delhi
104	vAbinandh	M	18	Chennai
107	Beena	F	20	Cochin
108	Jisha	F	19	Delhi

ROLLBACK TO A;

Admno	Name	Gender	Age	Place
105	Mini	F	19	Chennai
106	Devika	F	19	Bangalore
103	Ayush	M	18	Delhi
101	Adarsh	M	18	Delhi
104	Abinandh	M	18	Chennai
107	Beena	F	20	Cochin

Chapter 12.indd 232 30-01-2020 20:45:17

நினைவில் கொள்க:

- *SQL* என்பது உறவுநிலைத் தரவுத்தளங்களை உருவாக்கவும், செயல்படுத்தவும் உதவும் மொழியாகும்.
- *MySQL* என்பது ஒரு தரவுத்தள மேலாண்மை அமைப்பாகும்.
- *SQL*ன் பல்வேறு கூறுகளானவை தரவு வரையறை மொழி *(DDL)*, தரவு கையாளுதல் மொழி *(DML)*, தரவு வினவல் மொழி *(DQL)*, பரிவர்த்தனைக் கட்டுப்பாட்டு மொழி *(TCL)*, தரவு கட்டுப்பாட்டு மொழி *(DCL)*.
- *DDL* அட்டவணைகளை உருவாக்குதல் மற்றும் அழித்தலுக்கான கூற்றுகளை வழங்குகிறது.
- *DML* ஒரு அட்டவணையில் தரவை சேர்த்தல், புதுப்பித்தல் மற்றும் நீக்குதலுக்கான கூற்றுகளை வழங்குகிறது.
- *DCL* தரவு அணுகுதலுக்கான அங்கீகார கட்டளைகளை வழங்குகிறது.
- தரவுத்தளத்திலுள்ள பரிவர்த்தனைகளை மேலாண்மை செய்ய *TCL* கட்டளைகள் பயன்படுகின்றன.
- *DQL* கட்டளைகள் தரவுத்தளத்தில் வினவல்கள் உருவாக்க உதவுகின்றன.
- *CREATE TABLE* கட்டளை ஒரு புதிய அட்டவணையை உருவாக்குகிறது.
- *SELECT* கட்டளை வினவல்களை உருவாக்க உதவுகிறது. மேலும், அதனை **WHERE clause** பயன்படுத்தி பல்வேறு வகைகளில் கட்டுப்படுத்த முடியும்..
- *INSERT* கட்டளை அட்டவணையில் வரிசைகளை சேர்க்க அனுமதிக்கிறது.
- *DELETE* கட்டளை அட்டவணையில் வரிசைகளை நீக்க அனுமதிக்கிறது.
- *UPDATE* கட்டளை ஏற்கனவே உள்ள வரிசைகளின் மதிப்புகளுக்கு மாற்றங்கள் செய்ய அனுமதிக்கின்றது.
- *ALERT TABLE* கட்டளை அட்டவணையின் வடிவமைப்பை மாற்றியமைக்க அனுமதிக்கிறது.
- *TRUNCATE* கட்டளை அனைத்து பதிவுகள் மற்றும் அட்டவணையை நீக்கி, ஒதுக்கீடு செய்த இடத்தை விடுவிக்கப் பயன்படுகிறது.
- *DROP TABLE* கட்டளை தரவுத்தளத்திலிருந்து ஒரு அட்டவணையை நீக்குகிறது.
- *COMMIT* கட்டளை எந்தவொரு பரிவர்த்தனையையும் தரவுத்தளத்தில் நிரந்தரமாக சேமித்து வைக்கிறது.
- *SAVEPOINT* கட்டளை தேவையானப் போது, அந்த குறிப்பிட்ட இடத்திற்கு ஒரு குறிப்பிட்ட இடத்தில் பரிவர்த்தனையை தற்காலிகமாக சேமிக்கிறது.
- *ROLLBACK* தரவுத்தளத்தை அதன் கடைசி *COMMIT* கட்டளைக்கு பிறகு கொடுத்த அனைத்து கட்டளைகளையும் திரும்ப பெறும்.

செய்முறைப் பயிற்சி

1. பணியாளர் எண், பணியாளர் பெயர், பதவி, பணியில் சேர்ந்த தேதி, அடிப்படை ஊதியம் ஆகிய புலங்களைக் கொண்ட ஒரு பணியாளர் அட்டவணையை உருவாக்குக.
2. மேற்கூறிய அட்டவணையில் பணியாளர் எண்ணை முதன்மை திறவுகோல் (primary key) ஏற்படுத்திக் கொள்ளவும். மேலும், ஏதேனும் ஒரு புலம் NOT NULL-லாக குறிப்பிடுக.
3. மேலாளர்களாக இருக்கும் அனைத்து பணியாளர்களின் பட்டியலை தயாரிக்கவும்.

Chapter 12.indd 233 30-01-2020 20:45:17

செய்முறைப் பயிற்சி

1. மாணவர் அட்டவணைக்கு **name, age, place** மற்றும் **admno** என்ற புல வரிசைப்படி வினவலை உருவாக்கு.
2. மாணவர் அட்டவணையில் **18** வயதுக்கு மேற்பட்ட மாணவர் பற்றிய விவரங்களை வெளியிட ஒரு வினவலை உருவாக்கவும். **City** என்ற புலத்திற்கு **unique** கட்டுப்பாட்டை வரையறுத்துக் கொள்ளவும்.

F93S9E

மதிப்பீடு

பகுதி–அ

சரியான விடையை தேர்ந்தெடுத்து எழுதுக **(1 மதிப்பெண்)**

1. எந்த கட்டளைகள் அட்டவணை வடிவமைப்பை உருவாக்குதல், உறவுநிலையை நீக்குதல் மற்றும் உறவுநிலை திட்ட வடிவமைப்பை மாற்றுதல் போன்ற செயற்பாடுகளுக்கான வரையறைகளை வழங்குகிறது?

   அ) DDL   ஆ) DML

   இ) DCL   ஈ) DQL

2. எந்த கட்டளை அட்டவணையின் வடிவமைப்பை மாற்றி அமைக்க அனுமதிக்கிறது?

   அ) SELECT   ஆ) ORDER BY

   இ) MODIFY   ஈ) ALTER

3. அட்டவணையை நீக்க பயன்படுத்த வேண்டிய கட்டளை

   அ) DROP   ஆ) DELETE

   இ) DELETES ALL   ஈ) ALTER TABLE

4. வினவல்களை உருவாக்க பயன்படுவது

   அ) SELECT   ஆ) ORDER BY

   இ) MODIFY   ஈ) ALTER

5. ஒரு தரவுத்தளத்தில் உள்ள தரவை வரிசையாக்கம் செய்ய பயன்படும் clause

   அ) SORT BY   ஆ) ORDER BY

   இ) GROUP BY   ஈ) SELECT

பகுதி–ஆ

அனைத்து வினாக்களுக்கும் விடையளி **(2 மதிப்பெண்)**

1. 18 வயதிற்கும் குறைவாக உள்ள அனைத்து மாணவர்களின் தரவினை வரிசைப்படி தெரிவு செய்யும் ஒரு வினவலை எழுதுக.
2. Unique மற்றும் Primary Key கட்டுபாடுகளை வேறுபடுத்துக
3. அட்டவணை கட்டுப்பாட்டிற்கும், நெடுவரிசை கட்டுப்பாட்டிற்கும் உள்ள வேறுபாடுகளை எழுதுக?

Chapter 12.indd 234 30-01-2020 20:45:18

4. எந்த SQL கூறு, அட்டவணையை உருவாக்கவும், அவற்றில் மதிப்புகளை சேர்க்கவும் அனுமதிக்கும்?
5. SQL மற்றும் MySQLக்கு உள்ள வேறுபாடுகள் யாவை?

பகுதி–இ

அனைத்து வினாக்களுக்கும் விடையளி (3 மதிப்பெண்)

1. கட்டுப்பாடு என்றால் என்ன? Primary Key கட்டுப்பாடு பற்றி சிறுகுறிப்பு வரைக.
2. ஒரு புதிய புலத்தை சேர்ப்பதன் மூலம் மாணவர் அட்டவணை கட்டமைப்பை மாற்றி அமைக்க ஒரு SQL கூற்றை எழுதுக.
3. ஏதேனும் மூன்று DDL கட்டளைகளை எழுதுக.
4. Savepoint கட்டளையின் பயன்பாட்டை ஒரு எடுத்துக்காட்டுடன் எழுதுக.
5. DISTINCT சிறப்புச் சொல்லை பயன்படுத்தி ஒரு SQL கூற்றினை எழுதுக.

பகுதி–ஈ

அனைத்து வினாக்களுக்கும் விடையளி (5 மதிப்பெண்)

1. பல்வேறு வகையான கட்டுப்பாடுகளையும், அதன் செயல்பாடுகளையும் எழுதுக.
2. கீழ்காணும் பணியாளர் அட்டவணையை கருத்தில் கொண்டு, (i) முதல் (v) வரையிலான வினாக்களுக்கு SQL கட்டளைகளை எழுதுக.

EMP CODE	NAME	DESIG	PAY	ALLO WANCE
S1001	Hariharan	Supervisor	29000	12000
P1002	Shaji	Operator	10000	5500
P1003	Prasad	Operator	12000	6500
C1004	Manjima	Clerk	8000	4500
M1005	Ratheesh	Mechanic	20000	7000

(i) அனைத்து பணியாளர்களின் விவரங்களை அவர்கள் பெறும் சம்பளங்களின் இறங்கு வரிசையில் காண்பிக்க.
(ii) 5000 முதல் 7000 வரை ALLOWANCE பெறும் அனைத்து பணியாளர்களின் விவரங்களை காண்பிக்க.
(iii) mechanic வகையை சார்ந்த பணியாளர்களை நீக்க.
(iv) ஒரு புதிய வரிசையை உருவாக்க.
(v) operators வகையை சார்ந்த அனைத்து பணியாளர்களின் விவரங்களை காண்பிக்க.

3. SQLன் கூறுகள்? ஒவ்வொன்றிற்கும் கட்டளைகளை எழுதுக.
4. மாணவர் அட்டவணையில் பின்வரும் SQL கூற்றுகளை கட்டமைக்கவும்.
   (i) SELECT கூற்று GROUP BY clause பயன்படுத்தி
   (ii) SELECT கூற்று ORDER BY clause பயன்படுத்தி
5. பணியாளர்களுக்கான ஏதேனும் ஐந்து புலங்களைக் கொண்ட ஒரு அட்டவணையை உருவாக்க ஒரு SQL கூற்றினை எழுதி, அந்த பணியாளர் அட்டவணைக்கு ஒரு அட்டவணை கட்டுப்பாட்டை உருவாக்கவும்.

Chapter 12.indd 235 30-01-2020 20:45:18

அலகு IV

பாடம் 13

பைத்தான் மற்றும் CSV கோப்புகள்

52BGN6

கற்றலின் நோக்கங்கள்

இப்பாடத்தை கற்றறிந்த பின் மாணவர்கள் தெரிந்து கொள்வது.

- CSV என்றால் என்ன? அதை பற்றி புரிந்து கொள்ளுதல்
- பைத்தான் நிரலில் CSV கோப்புகளை இறக்கம் (integer) பற்றி அறிதல்.
- பைத்தான் நிரல்களை பிழைத்திருத்தி இயக்குதல்.

13.1 அறிமுகம்

பைத்தான் மொழியானது மிக அதிக நூலகத் செயற்கூறுகள் கொண்டுள்ளது. அவற்றுள் ஒன்றான CSV செயற்கூறானது பைத்தான் நிரலர் CSV கோப்புகளை பற்றி முழுமையாக அறிந்து கொள்ள உதவுகின்றது. CSV (Comma Separated values) கோப்பில் ஒவ்வொரு வரியும் காற்புள்ளி அல்லது வேறு பிரிப்பான்களை பயன்படுத்தி பிரிக்கப்பட்ட பல்வேறு புலங்களைக் கொண்டு பயனரால் புரிந்து கொள்ள கூடிய உரைக்கோப்பாகும். ஒவ்வொரு வரியை ஒரு வரிசையாகவும் ஒவ்வொரு புலத்தையும் நெடுவரிசையாகவும் கருதிக் கொள்ளலாம். CSV செயற்கூறுகளை பயன்படுத்தி பெரும்பாலான CSV கோப்புகளை எழுதவும், படிக்கவும் முடியும்.

13.2 CSV மற்றும் XLS கோப்புகளுக்கிடையேயான வேறுபாடு

Comma-Separated Values (CSV) மற்றும் eXceL Sheets(XLS) கோப்புகளுக்கிடையேயான வேறுபாடுகள்.

எக்ஸெல்	CSV
கோப்பின் அனைத்து அட்டவணைத்தாளிலுள்ள பொருளடக்கம் மற்றும் வடிவூட்டல்களை உள்ளடக்கிய தகவல்களை இரு நிலை வடிவில் கொண்ட கோப்பாகும்.	CSV வடிவத்தில் காற்புள்ளிகளால் பிரிக்கப்பட்ட தொடர்ச்சியான மதிப்புகளைக் கொண்ட எளிய உரை வடிவ கோப்பாகும்.
XLS கோப்புகள் அவற்றை படிப்பதற்காக அவற்றை உருவாக்கப்பட்ட பயன்பாடுகளை கொண்டு மட்டுமே படிக்க முடியும். அது போன்றே எழுதுதலும் ஆகும்.	CSV கோப்புகளை Windows இயக்க அமைப்பில் உள்ள notepad, MS Excel, OpenOffice போன்ற உரைப் பதிப்பான்களைக் கொண்டு திறக்கலாம்.
Excel அட்டவணைச்செயலி அதற்கென உருவாக்கப்பட்டுள்ள தனியுரிம வடிவமைப்பு வகையில் கோப்பினை சேமிக்கும். (அதாவது xls அல்லது xlsx).	அட்டவணை வடிவ தகவல்களை .csv என்ற நீட்டிப்புடன் பிரிக்கப்பட்ட உரைக் கோப்புகளாக சேமிக்கும் வடிவம் CSV ஆகும்.

Excel ஆனது தரவை பெறும் இறக்கம் செய்யும் (import) போது அதிக நினைவக இடத்தை எடுத்துக் கொள்ளும்.	CSV கோப்புகளை இறக்கம் செய்யும் போது வேகமாக செயல்படும். குறைவான நினைவக இடத்தை எடுத்துக் கொள்ளும்.

உங்களுக்குத் தெரியுமா?

Excel வடிவத்தில் சேமிக்கப்பட்ட கோப்புகளை உரைப்பதிப்பானை கொண்டு திறக்கவோ அல்லது பதிப்பாய்வு செய்யவோ முடியாது.

13.3 CSV கோப்புகளின் பயன்பாடுகள்

தரவுத்தளம் அல்லது அட்டவணைச் செயலியில் உள்ள அட்டவணை வகை தரவுகளை சேமிக்க ஒரு எளிய கோப்பு வடிவமாக **CSV** பயன்படுகிறது. எளிய உரைவடிவ கோப்பாக இருப்பதனால் நாம் பயன்படுத்தும் மென்பொருளை பொருட்படுத்தாமல் அட்டவணைச் செயலி அல்லது தரவுத்தளத்தில் இறக்கம் செய்ய இது எளியதாகும்.

CSV கோப்புகளை திறப்பதற்கும் அவற்றை எளிதாக படிப்பதற்கும் அட்டவணைச் செயலியான மைக்ரோ சாஃப்ட் எக்ஸெல் அல்லது ஏதேனும் ஒரு உரை பதிப்பான் அல்லது ஒரு ஏதேனும் ஒரு தரவுத்தளத்தை பயன்படுத்தலாம்.

குறிப்பு

CSV கோப்பு விளக்கப்படங்கள் அல்லது வரைபடங்களை தேக்கி வைக்காது. வடிவூட்டலைக் கொண்டிராத உரைகள், வாய்பாடுகள் மற்றும் மேக்ரோக்கள் போன்றவற்றைக் கொண்டிராத தரவுகளை மட்டும் சேமிக்கும். ஆங்கிலத்தில் MODULES, FUNCTIONS, PROCEDURES என்ற வார்த்தைகள்(Functions) செயற்கூறுகளைக் குறிக்கும் பொதுவான சொற்களாகும்.

உங்களுக்குத் தெரியுமா?

CSV கோப்பானது FLAT FILE எனவும் அழைக்கப்படும். தரவுகளை அட்டவணையாக சேமிக்கும் நிரல்களான மைக்ரோ சாஃப்ட் எக்ஸெல் அல்லது ஓபன்ஆஃபீஸ் கால்க் போன்றவற்றில் CSV வடிவ கோப்புகளை இறக்கம் (import) மற்றும் ஏற்றம் (export) செய்து கொள்ள முடியும்.

13.4 Notepad உரை பதிப்பானை பயன்படுத்தி CSV கோப்புகளை உருவாக்குதல் (அல்லது எதேனும் ஒரு உரை பதிப்பான்)

CSV கோப்பானது ஓர் உரைக் கோப்பாகும். எனவே ஏதேனும் ஒரு உரைபதிப்பானை பயன்படுத்தி CSV கோப்புகளை உருவாக்கவோ அல்லது பதிப்பாய்வு செய்யவோ முடியும். ஆனால் CSV கோப்பானது அட்டவணைச் செயலி அல்லது தரவுத்தளத்தை ஏற்றம் செய்வதன் மூலமை உருவாக்கப்படும்.

13.4.1 சாதாரண CSV கோப்பினை உருவாக்குதல்

Notepad உரைப்பதிப்பனை கொண்டு CSV கோப்பினை உருவாக்க முதலில் புதிய ஆவணத்தை பின்வரும் வழிமுறையை பயன்படுத்தி திறக்கலாம்.

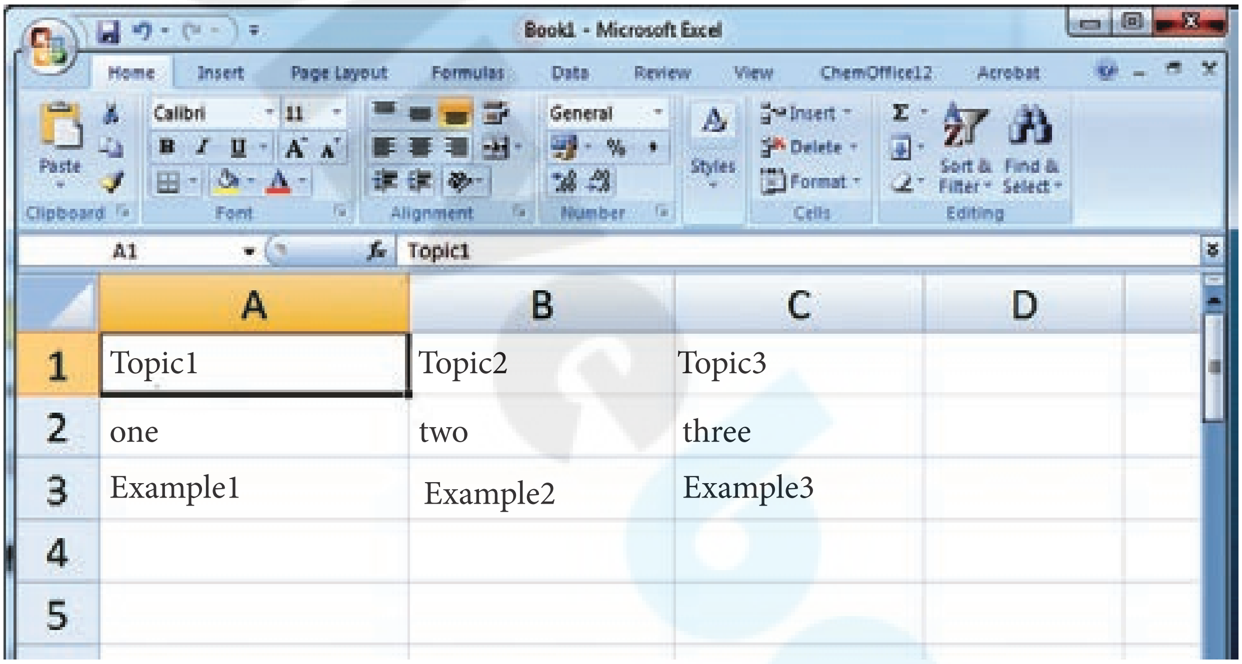
Chapter 13.indd 237 30-01-2020 20:48:27

File →New or ctrl +N.

அடுத்து, கோப்பில் உள்ளிடப்பட வேண்டிய தரவின் ஒவ்வொரு வரி அல்லது வரிசையிலுருள்ள மதிப்புகளை காற்புள்ளியால் பிரிக்கப்பட்டு உள்ளீடு செய்யவும். எடுத்துக்காட்டாக, கீழ்க்கண்ட விவரங்களை கருதி கொள்க.

> Topic1, Topic2, Topic3
> one, two, three
> Example1, Example2, Example3

மேற்கண்ட கோப்பினை .csv என்ற நீட்டிப்புடன் சேமிக்கவும். சேமிக்கப்பட்ட கோப்பினை மைக்ரோ சாப்ட் எக்ஸெல் அல்லது ஏதேனும் ஒரு அட்டவணைச் செயலி மென்பொருளை பயன்படுத்தி கோப்பினை திறக்கவும். மைக்ரோ சாப்ட் எக்ஸெலை பயன்படுத்தி மேற்கண்ட கோப்பினை திறக்கவும். உள்ளிடப்பட்ட தரவானது பின்வருமாறு அட்டவணை வடிவில் உருவாக்கப்பட்டிருக்கும்.



படம். *13.1 MS*-எக்ஸெல் *(MS EXCEL)* பயன்பாட்டை பயன்படுத்தி திறக்கப்பட்ட *CSV* கோப்பு

CSV கோப்பில் உள்ள புலத்தின் தரவுகள் காற்புள்ளியை தன்னுடன் கொண்டிருந்தால், காற்புள்ளியுடன் தரவினை வெளிப்படுத்த அத்தகைய பல தரவுகளை இரட்டை மேற்கோள் குறியுடன் (“) கொடுக்கவும். தரவின் ஒரு பகுதியாக காற்புள்ளியை கொள்ளவும் புலங்களை பிரிக்க பயன்படும் பிரிப்பானான காற்புள்ளியில் இருந்து இதை பிரித்து காட்டும்.

13.4.2 காற்புள்ளியுடன் கூடிய தரவினைக் கொண்ட CSV கோப்பினை உருவாக்குதல்

எடுத்துக்காட்டாக, முகவரி புலம் காற்புள்ளியை கொண்டுள்ளதாக கருத்துக் கொள்வோம். கீழே கொடுக்கப்பட்டுள்ள எடுத்துக்காட்டு நம் தரவாக இருப்பின்

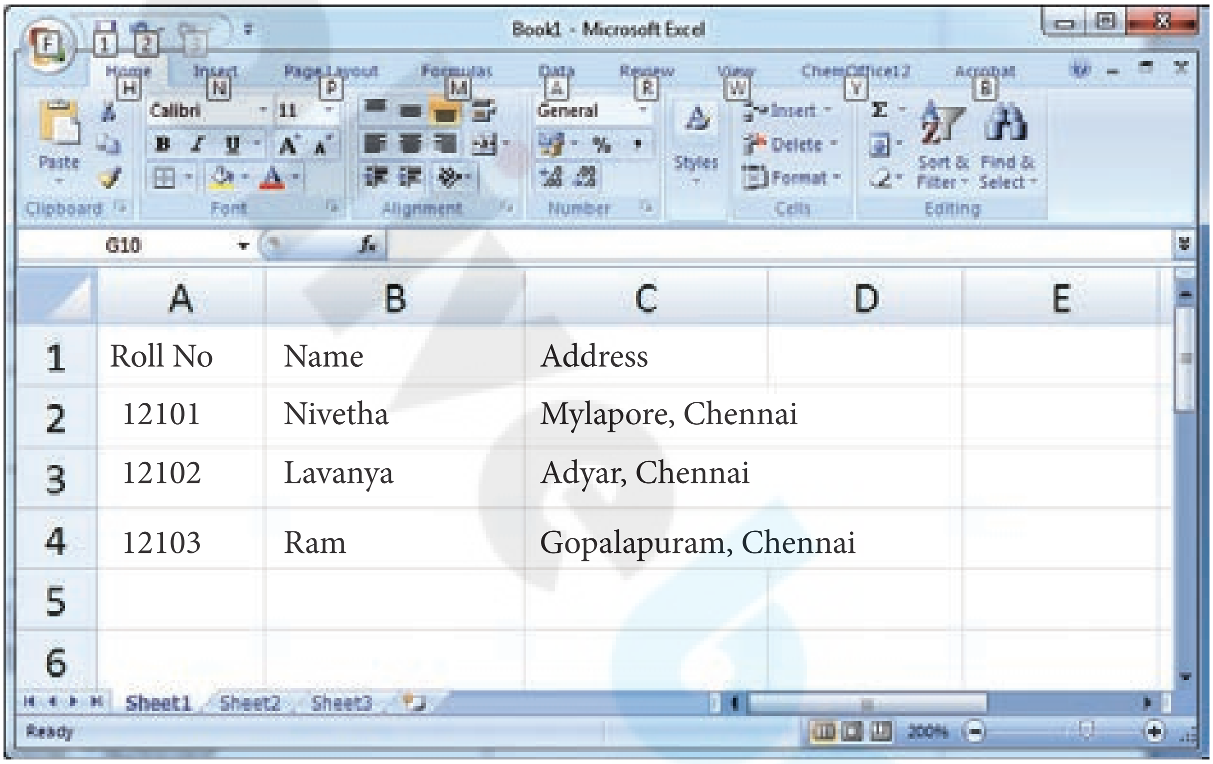
RollNo	Name	Address
12101	Nivetha	Mylapore, Chennai
12102	Lavanya	Adyar, Chennai
12103	Ram	Gopalapuram, Chennai

Chapter 13.indd 238 30-01-2020 20:48:27

முகவரி புலத்தில் உள்ள தரவுடன் காற்புள்ளியையும் சேர்த்து வெளிப்படுத்த அத்தரவினை இரட்டை மேற்கோள் குறிகளுக்குள் கொடுக்கவும். எடுத்துக்காட்டாக,

```
RollNo, Name, Address
12101, Nivetha, “Mylapore, Chennai”
12102, Lavanya, “Adyar, Chennai”
12103, Ram, “Gopalapuram, Chennai”
```

இரட்டை மேற்கோள் குறிகளுக்குள் கொடுக்கப்பட்ட தரவுகள் மட்டும் காற்புள்ளியுடன் வெளிப்படுத்தப்பட்டிருப்பதை காணலாம். MS-எக்ஸெல் பயன்பாட்டை பயன்படுத்தி மேற்கண்ட கோப்பினை திறக்கும் போது அது பின்வருமாறு தோற்றமளிக்கும்.



படம் *13.2* எக்ஸெல் *CSV* புலத்தின் தரவானது காற்புள்ளியுடன் வெளிப்படுத்துதல்.

புலத்தரவுகளை புதிய வரியில் இருப்பின் அவற்றை வெளிப்படுத்த மேற்கண்ட முறையே பயன்படுத்தலாம். ஏதேனும் புலத்தில் உள்ள தரவினை புதிய வரியில் வெளிப்படுத்த வேண்டுமெனில் புலத்தின் தரவினை இரட்டை மேற்கோள் குறிகளுக்குள் கொடுக்க வேண்டும்.

எடுத்துக்காட்டு,

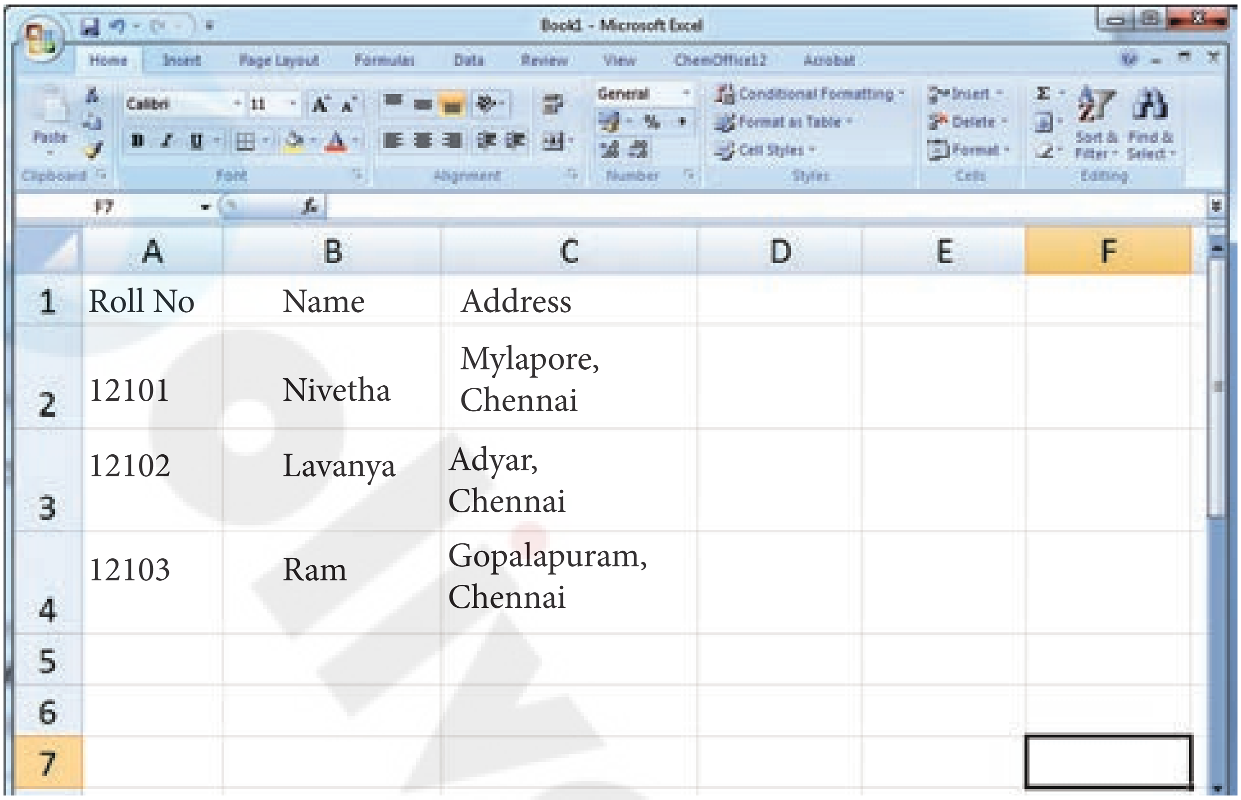
RollNo	Name	Address
12101	Nivetha	Mylapore, Chennai
12102	Lavanya	Adyar, Chennai
12103	Ram	Gopalapuram, Chennai

CSV கோப்பில் இது பின்வருமாறு எழுதப்பட வேண்டும்.

```
RollNo, Name, Address
12101, Nivetha, “Mylapore, Chennai”
12102, Lavanya, “Adyar, Chennai”
12103, Ram, “Gopalapuram, Chennai”
```


Chapter 13.indd 239 30-01-2020 20:48:27

வெளியீடானது பின்வருமாறு தோற்றமளிக்கும்



படம் *13.3* எக்ஸெல் *CSV* புலத்தரவானது புதிய வரியில் தோற்றமளித்தல்

13.4.3 இரட்டை மேற்கோள் குறிகளுடன் கூடிய தரவினை கொண்ட CSV கோப்பினை உருவாக்குதல்.

புலத்தில் உள்ள தரவானது இரட்டை மேற்கோள் குறிகளை ஒரு பகுதியாக தன்னுடன் கொண்டிருந்தால், இரட்டை மேற்கோள் குறியானது இரட்டிப்பாக்கப்படல் வேண்டும். இதன் மூலம் அவைகள் சரியாக விளக்கப்படலாம்.

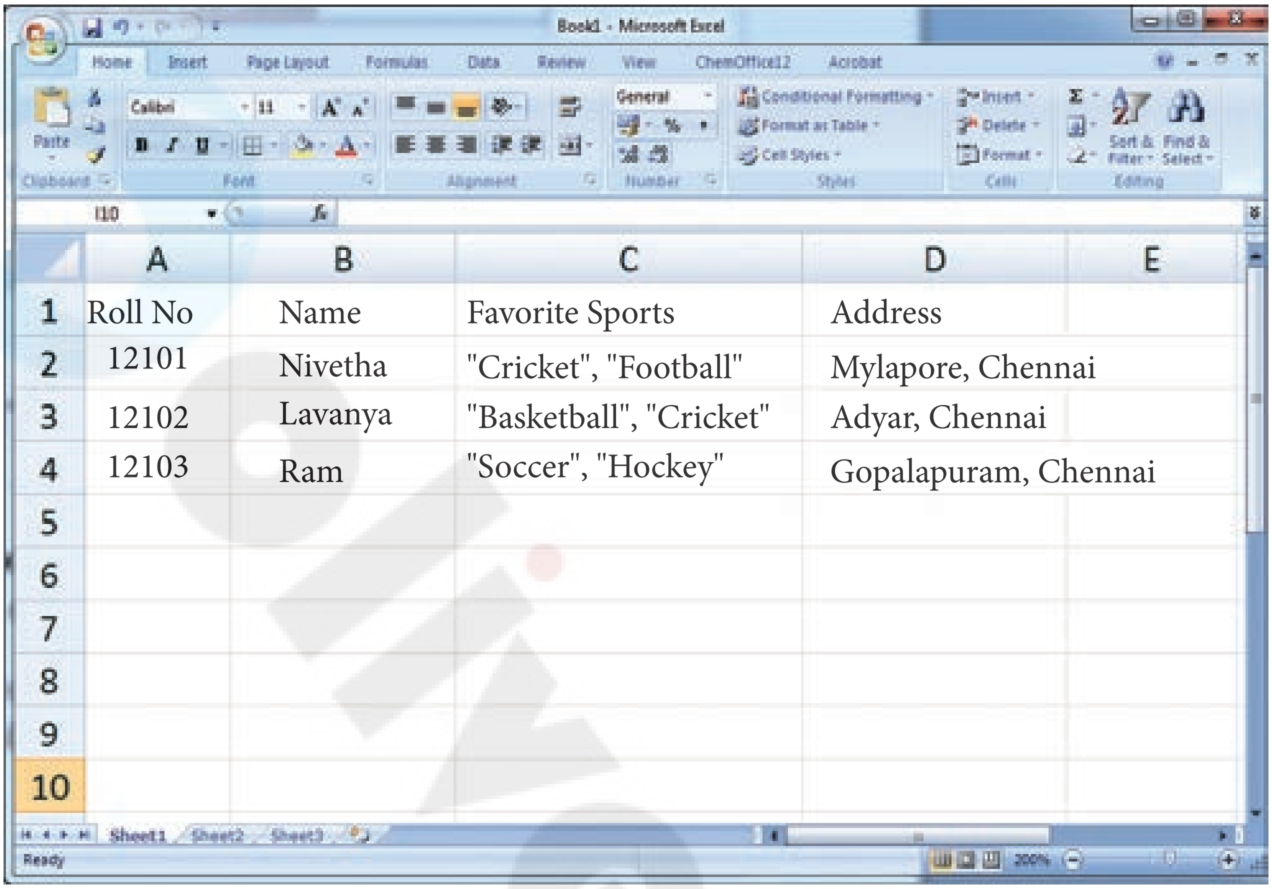
எடுத்துக்காட்டாக, கீழேயுள்ள தரவை கொடுக்க வேண்டுமெனில்,

Roll No	Name	Favorite Sports	Address
12101	Nivetha	“Cricket”, “Football”	Mylapore Chennai
12102	Lavanya	“Basketball”, “Cricket”	Adyar Chennai
12103	Ram	“Soccer”, “Hockey”	Gopala puram Chennai

இது csv கோப்பில் பின்வருமாறு உருவாக்கப்பட வேண்டும்.

```
RollNo, Name, FavoriteSports, Address
12101, Nivetha,”””” Cricket ”””,””” Football ””””, Mylapore chennai
12102, Lavanya,”””” Basketball ”””,””” Cricket ””””, Adyar chennai
12103, Ram,”””” Soccer”””,””” Hockey””””, Gopalapuram chennai
```

வெளியீடானது



	A	B	C	D	E
1	Roll No	Name	Favorite Sports	Address	
2	12101	Nivetha	"Cricket", "Football"	Mylapore, Chennai	
3	12102	Lavanya	"Basketball", "Cricket"	Adyar, Chennai	
4	12103	Ram	"Soccer", "Hockey"	Gopalapuram, Chennai	
5					
6					
7					
8					
9					
10					

படம் *13.4 CSV* புலத்தரவுகள் இரட்டை மேற்கோள் குறிகளுடன் *Excel*-ல் தோற்றமளித்தல்

13.4.4 CSV கோப்பிலுள்ள தரவை வடிவூட்டம் செய்ய பின்பற்ற வேண்டிய விதிமுறைகள்

1. ஒவ்வொரு வரிசையும் (தரவின் வரிசை) புதிய வரியில் இருத்த அந்த வரியை நுழைவு பொத்தானை அழுத்த வேண்டும். எடுத்துக்காட்டு :↵

   xxx,yyy↵

   ↵ என்ற குறியீடானது நுழைவு விசையை குறிக்கும்.

2. கோப்பினில் உள்ள கடைசி பதிவானது வரிமுறிவு/வரி செலுத்தி பிரிப்பானைக் கொண்டிருக்கலாம் அல்லது இல்லாமலும் இருக்கலாம்.எடுத்துக்காட்டாக:

   ppp, qqq ↵

   yyy, xxx

3. சாதாரண பதிவு வரிசைகளின் வடிவங்களை போன்று கோப்பின் முதல் வரிசையில் தோன்றக்கூடிய விருப்பத்தலைப்பு வரிசை இருக்கலாம். கோப்பின் தலைப்பானது புலங்களின் தொடர்புடைய பெயரினை கொண்டிருக்க வேண்டும். மேலும் பதிவுகளிலுள்ள புலங்களின் எண்ணிக்கையில் மீதமுள்ள கோப்பில் இருத்தல் வேண்டும். ↵

Chapter 13.indd 241 30-01-2020 20:48:27

எடுத்துக்காட்டு

```
aaa,bbb,ccc ↵
zzz,yyy,xxx CRLF( Carriage Return and Line Feed)
```

4. தலைப்பு மற்றும் ஒவ்வொரு பதிவிலும் ஒன்று அல்லது அதற்கு மேற்பட்ட புலங்கள் காற்புள்ளியால் பிரிக்கப்பட்டிருக்கலாம். இடைவெளியானது புலத்தின் ஒரு பகுதியான கருதப்படும் மேலும் நிராகரிக்கப்பட மாட்டாது. பதிவின் கடைசி காற்புலத்தை தொடர்ந்து காற்புள்ளி இடம் பெறல் கூடாது. எடுத்துக்காட்டு: Red , Blue

5. ஒவ்வொரு புலமும் இரட்டை மேற்கோள் குறிகளுக்குள் கொடுக்கப்படலாம் அல்லது கொடுக்கப்படாமல் இருக்கலாம். புலமானது இரட்டை மேற்கோள் குறிகளுக்குள் தரப்படவில்லையெனில், புலங்களில் இரட்டை மேற்கோள் குறியானது தோன்றாது: எடுத்துக்காட்டு:

```
"Red","Blue","Green"↵      # இரட்டை மேற்கோள் குறிகளுடன் உள்ள புலத்தின் தரவு
Black,White,Yellow         # இரட்டை மேற்கோள் குறிகள் இல்லாத புலத்தின் தரவு
```

6. புலங்களில் வரிதிருப்பி (CRLF), இரட்டை மேற்கோள் குறி மற்றும் காற்புள்ளி போன்றவைகளை கொண்டிருந்தால் அவைகள் இரட்டை மேற்கோள் குறிகளுக்குள் கொடுக்கப்பட வேண்டும். எடுத்துக்காட்டு:

```
Red, “,”, Blue CRLF        # காற்புள்ளியே ஒரு புலத்தின் மதிப்பாக இருப்பதால் அதை இரட்டை
                             மேற்கோள் குறிக்குள் கொடுக்கப்பட வேண்டும்.
Red, Blue , Green
```

7. புலமானது இரட்டை மேற்கோள் குறிக்களுக்குள் கொடுக்கப்பட்டிருக்க வேண்டுமெனில், இரட்டை மேற்கோள் குறிகளுக்குள் உள்ள புலமானது மற்றொரு இரட்டை மேற்கோள் குறிகளுக்குள் அமைக்கப்பட வேண்டும்.

```
Red, “ “Blue” ”, Green     # இரட்டை மேற்கோள்குறி ஒரு புலத்தின் மதிப்பாக இருப்பதால் அதை
                             மற்றொரு இரட்டை மேற்கோள் குறிக்குள் அமைக்கப்பட வேண்டும்
, , White
```

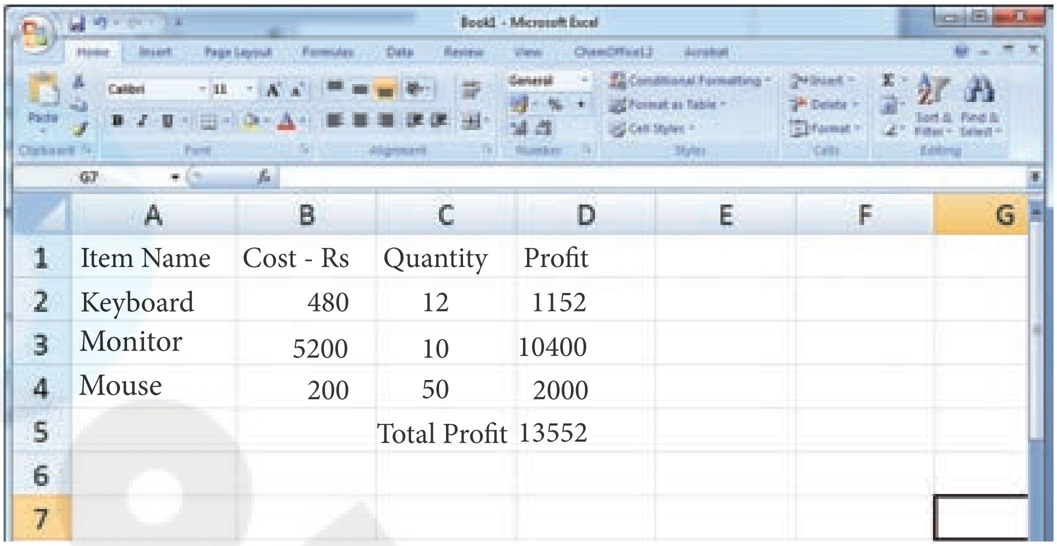
குறிப்பு

மேலே கொடுக்கப்பட்டுள்ள எடுத்துக்காட்டில் („White) உள்ள இறுதி வரிசையில் இரண்டு காற்புள்ளிகளுடன் தொடங்கியுள்ளது, ஏனென்றால் நமது அட்டவணையில் உள்ள பதிவின் முதல் இரண்டு வெற்று பதிவுகளாக இருக்கின்றது. இந்த காற்புள்ளிகளை அழிக்க வேண்டாம். இந்த இரண்டு காற்புகள்ளிகளும் ஒரு வரியிலிருந்து மற்றொரு வரியை உணர்த்த புலங்களாக பயன்படுகின்றது. இவற்றை தவிர்க்க கூடாது.

13.5 எக்ஸெலை பயன்படுத்தி CSV கோப்பினை உருவாக்குதல்

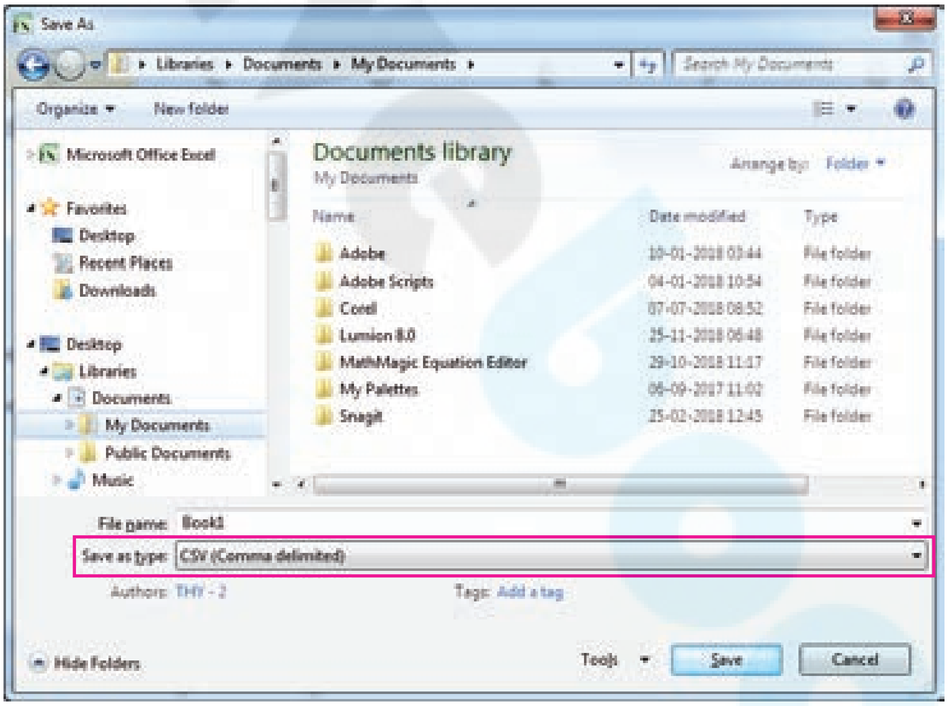
மைக்ரோ சாப்ட் எக்ஸெல் பயன்பாட்டை பயன்படுத்தி CSV கோப்புகளை உருவாக்க, **எக்ஸெல்** பயன்பாட்டை திறந்து வடிவத்தை CSV வடிவத்தில் சேமிக்க விரும்பும் கோப்பினை திறக்கவும். எடுத்துக்காட்டாக, எக்ஸெல் அட்டவணைத்தாள் மாதிரி தரவுகளைக் கொண்டுள்ளது.

Chapter 13.indd 242 30-01-2020 20:48:27



படம் *13.5* மாதிரி தரவுகளைக் கொண்டுள்ள அட்டவணைத்தாள்

அட்டவணைத் தாளில் தரவுகளை உள்ளிட்ட பிறகு **File** → **Save As** என்ற விருப்பத்தினை தேர்வு செய்யவும். "Save as type option" என்பதில் CSV காற்புள்ளியால் பிரிக்கப்பட்டு தேர்வு செய்யவும். கோப்பின் பெயருடன் CSV நீட்டிப்பினை தேர்வு செய்யவும்.



படம் *13.6 Save As* உரையாடல் பெட்டி

கோப்பினை சேமித்த பிறகு அக்கோப்பினை உரைப்பதிப்பானை பயன்படுத்தி எளிதாக திறக்கலாம் மேலும் பதிப்பாய்வு செய்யவும் முடியும். இதன் உள்ளடக்கமானது பின்வருபவனவற்றை ஒத்திருக்கும்.

```
Item Name, Cost-Rs, Quantity, Profit
Keyboard, 480, 12, 1152
Monitor, 5200, 10, 10400
Mouse, 200, 50, 2000
,,Total Profit =,13552
```


Chapter 13.indd 243 30-01-2020 20:48:27

13.5.1 மைக்ரோ சாப்ட் எக்ஸெல் கோப்பானது CSV கோப்பாக மாற்றப்பட்டுள்ளது

மைக்ரோ சாப்ட் எக்ஸெல் பயன்பாடானது கணினியில் நிறுவப்பட்டிருந்தால் CSV கோப்பின் மீது இருமுறை கிளிக் செய்தவுடன் தாமாகவே எக்ஸெல் பயன்பாட்டை பயன்படுத்தி CSV கோப்பானது திறக்கும்.

CSV கோப்பினை திறக்கும் போது மேல்மீட்பு பெட்டியில் **Open with** தோன்றினால் அதில் **மைக்ரோ சாப்ட் எக்ஸெல்** என்பதை தேர்வு செய்யவும்.

இம்முறைக்கு மாற்றாக MICROSOFT EXCEL பயன்பாட்டை திறந்து பட்டிப்பட்டையில் இருந்து File → Open என்பதை தேர்வு செய்து, அதில் CSV கோப்பினை தேர்வு செய்யவும். கோப்பின் பெயரானது தோன்றவில்லை எனில் கோப்பின் நீட்டிப்பினை மாற்றம் செய்து (*.prn, *.txt, *.csv) திறக்கலாம்.

உங்களுக்குத் தெரியுமா?

மைக்ரோ சாப்ட் எக்ஸெல் மற்றும் ஸ்டார் ஆ∴பீஸ் கால்க் என்ற இரண்டு பயன்பாடுகளுமே கணினியில் நிறுவப்பட்டிருந்தால் CSV கோப்பானது தானமைவாக எக்ஸெல் பயன்பாட்டின் மூலம் திறக்கப்படும்.

13.6 பைத்தான் மூலம் CSV கோப்பில் படிக்க மற்றும் எழுத

CSV கோப்புகளில் பல்வேறு செயல்பாடுகளை செய்ய பைத்தான் ஒரு **csv** என்ற செயற்கூற்றை வழங்கியுள்ளது. CSV கோப்பிலிருந்து அல்லது CSV கோப்பில் தரவினை படித்தல், எழுதுதல் மற்றும் பல செயல்பாடுகளை செய்வதற்கு CSV-ன் நூலகமானது பொருட்கள் மற்றும் பிற குறிமுறைகளை கொண்டுள்ளது.

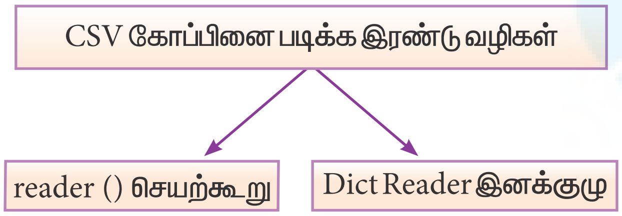
உங்களுக்குத் தெரியுமா?

CSV கோப்புகளானது நிகழ் நேர பயன்பாடுகளில் பரவலாக பயன்படுத்தப்படுகின்றன. ஏனெனில் அவை செயல்படுத்துவதற்கு எளிமையானவை.

13.6.1 பைத்தான் பயன்படுத்தி CSV கோப்பினை படித்தல்

CSV கோப்பினை படிக்க இரண்டு வழி முறைகள் உள்ளன.

1. CSV தொகுதி படித்தல் செயற்கூறை பயன்படுத்துதல்.
2. **DictReader** இனக்குழுவை பயன்படுத்துதல்.



படம் 13.7 CSV கோப்பினை படிக்கும் வழிமுறைகள்

கோப்பினில் உள்ள தகவல்களை படிப்பதற்கோ அல்லது கோப்பினில் தகவல்களை எழுதுவதற்கோ முதலில் கோப்பினை திறக்க வேண்டும். படித்தல் செயல் பாடானது முடிந்த பிறகு கோப்பானது மூடப்பட வேண்டும். இதன் மூலம் கோப்பிற்கு நினைவகத்தில் ஒதுக்கப்பட்டுள்ள இடம் விடுவிக்கப்படும்.

Chapter 13.indd 244 30-01-2020 20:48:27

எனவே பைத்தானில் கோப்பு செயல்பாடானது பின்வரும் படிநிலைகளில் செயல்படுத்தப்படுகிறது.

- படிநிலை 1 — கோப்பை திறக்கவும்
- படிநிலை 2 — படிக்க அல்லது எழுதும் செயல்பாட்டை செய்யவும்
- படிநிலை 3 — கோப்பை மூடிவும்

குறிப்பு

கோப்பினை திறக்கும் open கட்டளையில் கோப்பின் பெயர் அல்லது கோப்பின் முழுமையான பாதையானது இரட்டை மேற்கோள் குறிகளுக்குள் அல்லது ஒற்றை மேற்கோள் குறிகளுக்குள் கொடுக்கப்படல் வேண்டும்.

open() என்ற உள்ளிணைந்த செயற்கூறு பைத்தானில் ஒரு கோப்பை திறக்கும் இந்த செயற்கூறு கோப்பு பொருள் அல்லது திரும்பி கொடுக்கும், இதை பயன்படுத்தி கோப்பின் தரவுகளை படிக்கவோ அல்லது மாற்றவோ முடியும்.

எடுத்துக்காட்டாக,

```
>>> f = open("sample.txt")     # தற்போது கோப்புறையில் கோப்பை திறக்க மற்றும் இதில் f என்பது
                                 கோப்பு பொருளை குறிக்கும்.
>>> f = open('c:\pyprg\ch13sample5.csv')     # கோப்பின் முழுமையான பாதையைக் குறிக்கும்.
```

கோப்பை திறப்பதற்கான முறைமையை பயனர் குறிப்பிடலாம். பயனர் கோப்பினை படிப்பதற்கு 'r', எழுதுவதற்கு 'w' அல்லது கோப்பின் இறுதியில் சேர்க்க 'a' என்ற கோப்பு முறைமையை குறிப்பிடலாம். 'உரை அல்லது இருபரிமாண' (text or binary) என்ற ஏதேனும் ஒரு முறைமையில் கோப்பினை திறக்க முறைமையை குறிப்பிடலாம்.

உரை முறைமை தானமைவான கோப்பு பிடிக்கு முறைமையாகும். இதில், தரவை கோப்பிலிருந்து படிக்கும் போது தரவு சரங்களாக படிக்கப்படும். உரை அல்லது கோப்புகளான படங்கள் அல்லது .exe கோப்புகளை படிக்கும் போது அவற்றை இருமநிலை முறைமையில் கையாளுதல் செய்ய வேண்டும்.

பைத்தானில் கோப்பு முறைமைகள்

முறை	விளக்கம்
'r'	படிப்பதற்க மட்டுமே ஒரு கோப்பினை திறக்கவும் (தானமைவு நிலை)
'w'	கோப்பில் தரவுகளை எழுதுவதற்கு திறக்கவும். குறிப்பிடப்பட்ட கோப்பு இல்லையெனில் புதிய கோப்பினை உருவாக்கும். கோப்பில் தரவுகள் இருப்பின் அவை அழிக்கப்படும்.
'x'	தனித்துவமானபடைப்பிற்காககோப்பினைதிறக்கவும்.கோப்பு முன்பே உருவாக்கப்பட்டிருந்தால் இந்த செயல் முறையானது தோல்வியடையும்.
'a'	கோப்பின் தரவுகளை அழிக்காமல் அதன் இறுதியில் புதிய தரவுகளை சேர்ப்பதற்கு திறக்கவும். குறிப்பிடப்பட்ட கோப்பு இல்லையெனில் புதிய கோப்பினை உருவாக்கும்.

Chapter 13.indd 245 30-01-2020 20:48:28

't'	உரை முறைமையில் கோப்பு திறக்கவும் (தானமைவு நிலை)
'b'	இருமநிலை முறைமையில் கோப்பினை திறக்கவும்.
'+'	புதுப்பித்தலிற்காக கோப்பினை திறக்கவும் (படித்தல் மற்றும் எழுதுதல்)

f=**open("sample.txt")** # *'r'* அல்லது *'rt'* என்ற முறைமைகளுக்கு நிகர்

f = open("sample.txt",'w') # உரை முறைமையில் கோப்பில் எழுத.

f = open("image1.bmp",'r+b') # இருமநிலை முறைமையில் படித்தல் மற்றும் எழுதுதலுக்காக

நினைவகத்தின் பயன்பாட்டில் இல்லாத பொருள்களை **(objects)** சேகரிக்கவும் மற்றும் அந்த நினைவக பகுதியை சுத்தம் செய்யவும், பைத்தானில் தேவையற்ற நினைவக பகுதியை சேகரிக்கும் வசதி **(Garbage collector)** உள்ளது. ஆனால் இந்த வசதி கோப்புகளை மூடும் என்று பயனர் நம்ப வேண்டாம்.

```
f = open("test.txt")     # எந்த முறைமையும் குறிப்பிடப்படவில்லையெனில் தானமைவாக
                         rt என்ற முறைமை பயன்படும்
# n காப்பில் செயற்பாடுகள் செய்யவும்.
f.close()
```

மேலே குறிப்பிட்டுள்ள வழிமுறை பாதுகாப்பானது அல்ல, ஏதேனும் ஒரு பிழை ஏற்படுமெனில் அந்த செயற்பாட்டிலிருந்து நிரலானது கோப்பை மூடாமல் வெளியேறும். இதை சிறந்த முறையில் கையாள **with** கூற்றை பயன்படுத்தலாம். **with** தொகுதியிலிருந்து வெளியேறும் போது அதனுள்ளே கொடுக்கப்பட்டுள்ள கோப்பு மூடப்படும். செயற்கூற்றை வெளிப்படையாக கொடுக்க வேண்டியதில்லை. கோப்பு உள்ளமைவாகவே மூடப்படும்.

```
with open("test.txt",'r') as f:
# கோப்பு சார்ந்த செயல்பாடுகளை செய்ய
# என்பது கோப்பு பொருளாகும்.
```

கோப்பினை மூடும் போது அதனுடன் இணைக்கப்பட்டுள்ள வளங்கள் பைத்தானில் **close()** செயற்கூறு மூலம் விடுவிக்கப்படும்.

```
f = open("sample.txt")
# கோப்பு சார்ந்த செயல்பாடுகள் செய்ய
f.close()
```

13.6.1.1 CSV-ன் Reader செயற்கூறு

CSV கோப்பின் உள்ளடக்கத்தை படிக்க csv.reader() என்ற முறையானது பயன்படுத்தப்படுகிறது. reader() செயற்கூறானது கோப்பின் ஒவ்வொரு வரியையும் படித்து அவற்றை நெடுவரிசைகளின் பட்டியலாக (List) அமைக்கும். மாறியின் தரவிற்கு தேவையான நெடுவரிசையை தேர்வு செய்யலாம்.. இச்செயற்கூறை பயன்படுத்தி பயனர் கோப்பின் தரவுகளில் உள்ள இரட்டை மேற்கோள் குறி (“ ”), (|) மற்றும் (,) போன்ற பல்வேறு வடிவமைப்புகளை பயன்படுத்தி படிக்கலாம்.

csv.reader() செயற்கூறின் தொடரியல்

```
csv.reader(fileobject, delimiter, fmtparams)
```


Chapter 13.indd 246 30-01-2020 20:48:28

இங்கு,

file object :- கோப்பின் பாதையையும் முறைமையும் திருப்பி அனுப்பும்.

delimiter :- இது விருப்பத்தேர்வு அளபுருவாகும். இது செந்தர வரம்புக் குறிகளை கொண்டிருக்கும். (;|) போன்றவை மற்றவைகள் தவிர்க்கப்படும்.

fmtparams: கொடாநிலை மதிப்புகளை மேற்பதிப்பு செய்ய/நீக்க பயன்படும் விருப்பத் தேர்வு அளபுருவாகும்.

1. CSV கோப்பு – கொடாநிலை பிரிப்பானான காற்புள்ளியை கொண்ட தரவு.
2. CSV கோப்பு – தொடக்கத்தில் இடம் கொண்ட தரவுகள்
3. CSV கோப்பு – மேற்கோளுடன் கூடிய தரவுகள்
4. CSV கோப்பு – தனிப்பயன் பிரிப்பான கொண்ட தரவு.

(a) கொடாநிலை பிரிப்பான் காற்புள்ளியுடன் கூடிய CSV (,) கோப்புகள்

பின்வரும் நிரலானது கொடாநிலை பிரிப்பானான காற்புள்ளியுடன் கூடிய "sample1.csv" என்ற கோப்பினை படித்து ஒவ்வொரு வரிசையாக அச்சிடும் நிரல்.

```
#importing csv
import csv
#opening the csv file which is in different location with read mode
with open('c:\pyprg\sample1.csv', 'r') as F:
 #other way to open the file is f= ('c:\pyprg\sample1.csv', 'r')
    reader = csv.reader(F)
    for row in reader :
# printing each line of the Data row by row
       print(row)
F.close()
```

வெளியீடு

```
    ['SNO', 'NAME', 'CITY']
    ['12101', 'RAM', 'CHENNAI']
    ['12102', 'LAVANYA', 'TIRUCHY']
    ['12103', 'LAKSHMAN', 'MADURAI']
```

(b) தொடக்கத்தில் இடைவெளிகளுடன் கூடிய தரவினைக் கொண்ட CSV கோப்பு

பின்வரும் sample2.csv என்ற கோப்பினை கருத்தில் கொள்ளவும்.

NOTEPAD பயன்பாட்டில் மூலம் திறக்கும் போது பின்வருமாறு தரவினை கொண்டிருக்கும்.

Topic1,	**Topic2,**	**Topic3,**
one,	two,	three
Example1,	Example2,	Example3

பின்வரும் நிரலானது பைத்தானில் உள்ள csv.reader() என்ற செயற்கூறு மூலம் கோப்பினை படிக்கிறது.

```
import csv
csv.register_dialect('myDialect',delimiter = ',',skipinitialspace=True)
F=open('c:\pyprg\sample2.csv','r')
reader = csv.reader(F, dialect='myDialect')
for row in reader:
        print(row)
F.close()
```

வெளியீடு

```
['Topic1', 'Topic2', 'Topic3']
['one', 'two', 'three']
['Example1', 'Example2', 'Example3']
```

"sample2.csv" கோப்பில் காண்பதை போல் பிரிப்பானான காற்புள்ளிக்கு அடுத்து வரும் வெற்று இடைவெளிகளால் வெளியீடும் வெற்று இடைவெளிகளுடன் தோன்றும்.

வெற்று இடைவெளிகளானது CSV செயற்கூறினுள் உள்ள csv.register_dialect() என்ற இனக்குழுவின் மூலம் நீக்கலாம். dialect ஆனது CSV கோப்பினை படிப்பதற்கான வழிமுறைகளை விவரிக்கிறது. dialects-ல் "skipinitialspace" என்ற அளபுருவானது பிரிப்பானிற்கு அடுத்து வரும் வெற்று இடைவெளிகளை நீக்க உதவுகிறது.

குறிப்பு

தானமைவாக "skipinitialspace" என்பதன் மதிப்பு false ஆகும்.

கீழேயுள்ள நிரலானது **"sample2.csv"** என்ற கோப்பில், பிரிப்பானற்க்கு பிறகுயுள்ள வெற்றிடைவெளிகளை படிக்கின்றது.

```
import csv
csv.register_dialect('myDialect', delimiter = ',' , skipinitialspace=True)
F=open('c:\pyprg\sample2.csv','r')
reader = csv.reader(F, dialect='myDialect')
for row in reader:
        print(row)
F.close()
```

வெளியீடு

```
['Topic1', 'Topic2', 'Topic3']
['one', 'two', 'three']
['Example1', 'Example2', 'Example3']
```

மேற்கண்ட நிரலில் புதிய dialect ஆனது (delimiter=' , ') காற்புள்ளி பிரிப்பாணை கொண்டுள்ளது. மேலும் d skipinitialspace என்பதன் மதிப்பு True என அமைக்கப்பட்டுள்ளது. இது பிரிப்பானிற்கு அடுத்து வரும் வெற்று இடைவெளிகளை தவிர்த்திடுமாறு நிரல் பெயர்ப்பிக்கு உணர்த்தும்.

Chapter 13.indd 248 30-01-2020 20:48:28

குறிப்பு

csv –ல் படிப்பதற்கு மற்றும் எழுதுவதற்கு உள்ள அளபுருக்களை வரையறுக்க dialect என்ற இனக்குழு CSV செயற்கூறில் பயன்படுகிறது. இது தரவை வடிவூட்டம் செய்வதற்கான பல்வேறு உருவாக்க, சேமிக்க மற்றும் மறுபயனாக்க பயன்படும் அளபுருக்களை கொண்டுள்ளது.

(c) மேற்கோள் குறிகளுடன் CSV கோப்பு

CSV செயற்கூறில் உள்ள csv.register_dialect() என்ற புதிய இனக்குழுவை பதிவதன் மூலம் இரட்டை மேற்கோள் குறிகளுடன் கூடிய CSV கோப்பினை படிக்கலாம்.

quotes.csv என்ற கோப்பானது பின்வரும் தரவுகளை கொண்டிருக்கும்.

```
SNO,Quotes
1, "The secret to getting ahead is getting started."
2, "Excellence is a continuous process and not an accident."
3, "Work hard dream big never give up and believe yourself."
4, "Failure is the opportunity to begin again more intelligently."
5, "The successful warrior is the average man, with laser-like focus."
```

பின்வரும் நிரலானது காற்புள்ளியை பிரிப்பானாக கொண்டிருக்கும் “quotes.csv” என்ற கோப்பினை படிக்கும். ஆனால் இரட்டை மேற்கோள் குறிகளானது மற்றுமொரு மேற்கோள் குறிகளுக்குள் கொடுக்கப்பட்டிருக்கும்.

```
import csv
csv.register_dialect('myDialect', delimiter = ',' ,quoting=csv.QUOTE_ALL,
skipinitialspace=True)
f=open('c:\pyprg\quotes.csv','r')
reader = csv.reader(f, dialect='myDialect')
for row in reader:
    print(row)
```

வெளியீடு

```
['SNO', 'Quotes']
['1', 'The secret to getting ahead is getting started.']
['2', 'Excellence is a continuous process and not an accident.']
['3', 'Work hard dream big never give up and believe yourself.']
['4', 'Failure is the opportunity to begin again more intelligently.']
['5', 'The successful warrior is the average man, with laser-like focus. ']
```

மேற்கண்ட நிரலில் myDialect என்ற பெயருடன் கூடிய புதிய Dialect பதிவு செய்யப்பட்டுள்ளது. csv.QUOTE_ALL என்ற கோப்பினை பயன்படுத்தும் போது அனைத்து குறியுருக்களும் மேற்கோள் குறிக்கஅடுத்து வெளிப்படுத்தலாம்.

Chapter 13.indd 249 30-01-2020 20:48:28

(d) தனிப்பயனாக்கப்பட்ட பிரிப்பானைக் கொண்ட CSV கோப்பு

csv.register_dialect() என்பதன் உதவியுடன் புதிய **dialect**-ஐ பதிவு செய்து தனிப்பயனாக்கப்பட்ட பிரிப்பானைக் கொண்ட CSV கோப்புகளை படிக்க முடியும்.

பின்வரும் **"sample4.csv"** என்ற கோப்பானது பைப் | **(Pipe symbol)** குறியீட்டைக் கொண்டு புலங்கள் பிரிக்கப்பட்டுள்ளது.

```
Roll No | Name | City
12101 | Arun | Chennai
12102 | Meena | Kovai
12103 | Ram | Nellai
```

பின்வரும் நிரலானது பயனரால் வரையறுக்கப்பட்டுள்ள பிரிப்பானைக் கொண்டுள்ள |(Pipe symbol) "sample4.csv" என்ற கோப்பினை படிக்கும்.

```
import csv
csv.register_dialect('myDialect', delimiter = '|')
with open('c:\pyprg\sample4.csv', 'r') as f:
    reader = csv.reader(f, dialect='myDialect')
    for row in reader:
        print(row)
f.close()
```

வெளியீடு

```
['RollNo', 'Name', 'City']
['12101', 'Arun', 'Chennai']
['12102', 'Meena', 'Kovai']
['12103', 'Ram', 'Nellai']
```

மேற்கண்ட நிரலில் My Dialect என்ற பெயரில் புதிய dialect ஆனது பதிவு செய்யப்பட்டுள்ளது. இதில் | (Pipe symbol) ஆனது புலத்தை பிரிக்கும் பிரிப்பானாக அமைக்கப்பட்டுள்ளது.

13.6.2 ஒரு கோப்பினில் குறிப்பிட்ட புலத்தை மட்டும் படித்தல்

"sample5.csv" என்ற கோப்பினில் உள்ள Item Name மற்றும் profit போன்ற குறிப்பிட்ட புலங்களை மட்டும் படிக்க கீழ்க்காணும் வழிமுறைகளை செய்தல் வேண்டும்.

```
import csv
#opening the csv file which is in different location with read mode
f=open("c:\pyprg\ch13sample5.csv",'r')
#reading the File with the help of csv.reader()
readFile=csv.reader(f)
#printing the selected column
for col in readFile :
    print (col[0],col[3])
f.close()
```


Chapter 13.indd 250 30-01-2020 20:48:28

sample5.csv File in Excel

	A	B	C	D
1	item Nam	Cost-Rs	Quantity	Profit
2	Keyboard	480	12	1152
3	Monitor	5200	10	10400
4	Mouse	200	50	2000

sample5.csv File with selected col

வெளியீடு

```
Item Name Profit
Keyboard 1152
Monitor 10400
Mouse 2000
```

13.6.3 CSV கோப்பினை படித்தல் மற்றும் பட்டியலாக சேமித்தல்

இப்பகுதியில் CSV கோப்பினை படித்தல் மற்றும் கோப்பின் உள்ளடக்கத்தை பட்டியலாக சேமித்தல் முறையை பற்றி படிக்க இருக்கிறோம். பட்டியலாக சேமிப்பதற்கான தொடரியல்

```
list = []                  # வெற்று list-ஆக தொடங்கவும்
list.append(element)       # append() பயன்படுத்தி உருப்புகளை சேர்க்கவும்.
```

எடுத்துக்காட்டு பின்வரும் நிரலானது "sample.csv" கோப்பின் அனைத்து வரிசையில் உள்ள மதிப்புகளையும் பட்டியலாக சேமிக்கும்.

```
import csv
# other way of declaring the filename
inFile= 'c:\pyprg\sample.csv'
F=open(inFile,'r')
reader = csv.reader(F)
# declaring array
arrayValue = []
# displaying the content of the list
for row in reader:
    arrayValue.append(row)
    print(row)
F.close()
```


Chapter 13.indd 251 30-01-2020 20:48:28

sample.csv opened in MS-Excel

	A	B	C
1	Topic 1	Topic 2	Topic 3
2	One	two	three
3	Example 1	Example 2	Example 3
4			

sample5.csv File with selected col

வெளியீடு

['Topic1', 'Topic2', 'Topic3']
[' one', 'two', 'three']
['Example1', 'Example2', 'Example3']

குறிப்பு

வரிசைப்படுத்தப்பட்ட உறுப்புகளின் வரிசையை மாற்றக்கூடிய தரவு கட்டமைப்பு பட்டியல் (List) ஆகும்.

உங்களுக்குத் தெரியுமா?

நிலையுருக்களின் பட்டியலானது சதுர வடிவ அடைப்புக் குறிகளுக்குள் எழுதப்படல் வேண்டும். List-ஆக சரங்களை ஒத்ததாகும்.

13.6.4 வரிசையாக்கம் செய்வதற்காக CSV கோப்பினை படித்தது புலத்தின் மதிப்புகளை List-ல் சேமித்தல்

இந்த நிரலானது "sample6.csv" என்ற கோப்பினில் இருந்து பயனரால் தரப்படும் எண்ணிற்குரிய நெடுவரிசையின் பொருளடக்கத்தை படித்து அதை List -ல் சேமிக்கும்.

	A	B	C	D
1	Item Name	Cost - Rs	Quantity	Profit
2	Keyboard	480	12	1152
3	Monitor	5200	10	10400
4	Mouse	200	50	2000
5			Total Profit	13552

படம் *13.6.4* வரிசையாக்கம் செய்வதற்கு *CSV* கோப்பன் தரவுகளின் பட்டியல்

வரிசையின் தலைப்புகளும் சேர்த்து வரிசைப்படுத்தப்படுவதை தவிர்க்க வரிசைபடுத்தும் போது முதல் வரிசையானது தவிர்க்கப்படல் வேண்டும். "next()" என்ற கட்டளை மூலம் இதை செயல்படுத்த முடியும். List -ஆனது வரிசையப்படுத்தப்பட்டு வெளியிடப்பட்டுள்ளது.

தலைப்பு நெடுவரிசையை தவிர்த்து பயனர் தேர்ந்தெடுத்த நெடுவரிசையை வரிசையாக்கம் செய்ய

Chapter 13.indd 252 30-01-2020 20:48:28

```
# sort a selected column given by user leaving the header column in descending order
import csv
# other way of declaring the filename
inFile= 'c:\pyprg\sample6.csv'
# opening the csv file which is in the same location where the content of the python file is there.
```

```
python file
F=open(inFile,'r')
# reading the File with the help of csv.reader()
reader = csv.reader(F)
# skipping the first row(heading)
next(reader)
# declaring a list
arrayValue = []
a = int(input ("Enter the column number 1 to 3:-"))
# sorting a particular column-cost
for row in reader:
        arrayValue.append(row[a])
arrayValue.sort(reverse=True)
for row in arrayValue:
        print (row)
F.close()
```

வெளியீடு

```
        Enter the column number between 0 to 3:- 2
        50
        12
        10
```

குறிப்பு

ஒரு சிஎஸ்வி கோப்பில் ஒரு குறிப்பிட்ட நெடுவரிசையைப் படித்து அதன் முடிவை ஏறுவரிசையில் காண்பி

உங்களுக்குத் தெரியுமா?

list_name.sort() கட்டளையானது பட்டியலில் உள்ள உறுப்புகளை ஏறு வரிசையில் (ascending) வரிசைப்படுத்தும்.

list_name.sort(reverse) கட்டளையானது பட்டியலில் உள்ள உறுப்புகளை இறங்கு (descending) வரிசையில் ஒழுங்குபடுத்தும்.

Chapter 13.indd 253 30-01-2020 20:48:28

13.6.5 CSV கோப்பினில் உள்ள ஒரு குறிப்பிட்ட புலத்தை வரிசையாக்கம் செய்தல்.

இந்நிலையானது "sample8.csv" என்ற கோப்பின் முழு பொருளடக்கத்தையும் படித்து பட்டியலுக்கு மாற்றம் செய்யும். அதன் பிறகு quantity என்ற புலத்தில் உள்ள வரிசைகளின் பட்டியலானது வரிசையாக்கம் செய்யப்பட்டு ஏறு வரிசையில் வெளிப்படுத்தப்படும். ஒன்றுக்கு மேற்பட்ட புலங்களை வரிசைப்படுத்த itemgetter என்பதில் ஒன்றுக்கு மேற்பட்ட சுட்டு எண்களை குறிப்பிடுவதன் மூலம் நிறைவேற்றலாம். operator.itemgetter(col_no) என்ற கோப்பின் பொருளடக்கமானது

	A	B
1	Item Name	Quantity
2	Keyboard	48
3	Monitor	52
4	Mouse	20

sample8.csv in Notepad

ItemName ,Quantity
Keyboard, 48
Monitor,52
Mouse ,20

படம் *13.6.5 CSV* கோப்பு தரவினை வரிசையாக்க *list*-ல் சேர்த்தல்

பின்வரும் நிரலானது **operator.itemgetter(col_no)** என்பதை பயன்படுத்தி மேற்குறிப்பிட்ட நிரலை நிறைவேற்றுகிறது.

```
#Program to sort the entire row by using a specified column.
# declaring multiple header files
import csv ,operator
#One more way to read the file
data = csv.reader(open('c:\PYPRG\sample8.csv'))
next(data)                        #(to omit the header)
#using operator module for sorting multiple columns
sortedlist = sorted (data, key=operator.itemgetter(1))      # 1 specifies we want to sort
                                                              # according to  second column

for row in sortedlist:
       print(row)
```

வெளியீடு

```
       ['Mouse ', '20']
       ['Keyboard ', '48']
       ['Monitor', '52']
```

குறிப்பு

sorted() முறையானது / செயல்கூறானது ஒரு குறிப்பிட்ட வரிசையில் கொடுக்கப்பட்டபடி உறுப்புகளை ஏறுவரிசையிலோ அல்லது இறங்கு வரிசையிலோ வரிசைப்படுத்தும். sort() முறையானது sorted() முறையை ஒத்திருக்கும். ஒரு வேறுபாடானது sort() முறை எந்த மதிப்பையும் திரும்ப தராது மேலும்மூல பட்டியலில் மாற்றத்தை ஏற்படுத்தும்.

Chapter 13.indd 254 30-01-2020 20:48:28

செய்முறைப்பயிற்சி: "sample8.csv" என்ற கோப்பினில் "cost" எனும் மேலும் ஒரு புலத்தை சேர்த்து அதன் பொருள் இறங்கு வரிசையில் வரிசையாக்கம் செய்ய பின்வரும் தொடரியலானது பயன்படுகிறது.

தொடரியல்

sortedlist = sorted(data, key=operator.itemgetter(Col_number),reverse=True)

13.6.6 CSV கோப்பினை Dictionary படிக்க

CSV செயற்கூறிலுள்ள DictReader இனக்குழுவை பயன்படுத்தி CSV கோப்பை படித்தல்.இதன் செயல்பாடு reader() இனக்குழு செயல்பாட்டை ஒத்திருக்கும். ஆனால் இது ஒரு பொருளை உருவாக்கி அதை Dictionaryயில் இணைக்கும்.

DictReader CSV கோப்பில் உள்ள முதல் வரியை காற்புள்ளியை இந்த வரியின் Dictionary திறவுகோள் (Dictionary Key) பயன்படுத்தி படிக்கும். அடுத்தடுத்துள்ள வரிசையில் உள்ள நெடுவரிசையானது Dictionaryயின் மதிப்புகளாக செயல்பட்டு அவற்றை உரிய திறவுகோள் மூலம் அணுக முடியும் (புலத்தின் பெயர்)

CSV கோப்பின் முதல் வரிசையானது புலப்பெயரை கொண்டிருக்கவில்லையெனில் புலப்பெயர் அளபுருவை DictReader's ஆக்கிக்கு அனுப்பி Dictionary திறவுகோள்களை நாமாக ஒதுக்க முடியும்.

csv.reader() மற்றும் DictReader() க்கு இடையேயான முக்கிய வேறுபாட்டை எளிமையாக கூறுவதெனில் csv.reader மற்றும் csv.writer ஆனது பட்டியல் (list/tuple) பதிவுடன் வேலை செய்யும். csv.DictReader மற்றும் csv.DictWriter ஆனது அகராதியில் வேலை செய்யும்.

```
import csv
filename = 'c:\pyprg\sample8.csv'
input_file =csv.DictReader(open(filename,'r'))
for row in input_file:
     print(dict(row))                    #dict() to print data
```

வெளியீடு

```
{'ItemName ': 'Keyboard ', 'Quantity': '48'}
{'ItemName ': 'Monitor', 'Quantity': '52'}
{'ItemName ': 'Mouse ', 'Quantity': '20'}
```

csv.DictReader மற்றும் csv.DictWriter இரண்டும் கூடுதல் அளபுருவாக புலப்பெயரினை பெற்று Dictionary திறவுகோளாக பயன்படுத்தும்.

(எடுத்துக்காட்டாக) "sample8.csv" என்ற கோப்பானது Dictionary -யை படித்தல்.

செயல்முறை: மேலே கொடுக்கப்பட்டுள்ள நிரலில் dict() செயல்கூறில் நீக்கவும் மற்றும் print(row) பயன்படுத்தவும். கீழே கொடுக்கப்பட்டுள்ள வெளியீடை தரும்.

```
OrderedDict([('ItemName ', 'Keyboard '), ('Quantity', '48')])
OrderedDict([('ItemName ', 'Monitor'), ('Quantity', '52')])
OrderedDict([('ItemName ', 'Mouse '), ('Quantity', '20')])
```

13.6.7 Dictionary பயனர் வரையறுத்த பிரிப்பானை பயன்படுத்தி CSV கோப்பினை படித்தல்

புதிய Dialects ஐ பதிவு செய்து DictReader() செயற்கூறில் நாம் அதை பயன்படுத்த முடியும். "sample8.csv" கோப்பானது பின்வரும் வடிவத்தில் இருந்தால்

```
ItemName|Quantity
Keyboard|48
Monitor|52
Mouse|20
```

"sample8.csv"(படம் 13.9) ஒரு புதிய dialect பயன்படுத்தி Dictionary படிக்கலாம்.

```
import csv
csv.register_dialect('myDialect',delimiter = '|',skipinitialspace=True)
filename = 'c:\pyprg\ch13\sample8.csv'
with open(filename, 'r') as csvfile:
        reader = csv.DictReader(csvfile, dialect='myDialect')
        for row in reader:
            print(dict(row))
csvfile.close()
```

வெளியீடு

```
        {'ItemName ,Quantity': 'Keyboard ,48'}
        {'ItemName ,Quantity': 'Monitor,52'}
        {'ItemName ,Quantity': 'Mouse ,20'}
```

குறிப்பு

DictReader() தானமைவாக ordereddict என்பதை வெளியீடும். Dictionaryயின் ஒரு துணைகுழுவாக Ordereddict செயல்படும் இது உள்ளடக்கத்தைக்கும் அவற்றை சேமிக்கும் வரிசையில் சேர்க்கும். Ordereddict நீக்க Dict() யை பயன்படுத்தவும்.

13.7 பல்வேறு வகையான CSV கோப்பினுள் தரவுகளை எழுதுதல்

பைத்தானது CSV கோப்பில் பணியாற்றுவதற்கு எளிமையான வழிகளை வழங்குகிறது. மேலும் CSV கோப்பில் தரவுகளை படிக்க மற்றும் எழுத CSV கோப்பில் தரவுகளை படிக்க மற்றும் எழுத CSV கூறினை கொண்டுள்ளது. இதற்கு முந்தைய தலைப்புகளில் CSV கோப்பில் இருந்து தரவுகளை எவ்வாறு படிப்பது என்பதை பற்றி பயின்றோம். அதை போலவே ஒரு புதிய CSV கோப்பினை உருவாக்கவும் (அல்லது) ஏற்கனவே உள்ள கோப்பில் மாற்றங்கள் செய்யவும் முடியும்.

Chapter 13.indd 256 30-01-2020 20:48:28

1 சாதாரணமான ஒரு புதிய CSV கோப்பினை உருவாக்குதல்.

2 ஏற்கனவே உள்ள கோப்பினை திருத்துதல்.

3 மேற்கோள் குறியுடன் CSV கோப்பினில் எழுதுதல்.

4 தனிப்பயன் பிரிப்பான்களுடன் CSV கோப்பினில் எழுதுதல்.

5 Lineterminator உடன் CSV கோப்பினில் எழுதுதல்.

6 Quotechars உடன் CSV கோப்பினில் எழுதுதல்.

7 CSV கோப்பினை Dictionaryயில் எழுதுதல்.

8 இயக்க நேரத்தில் தரவினை பெற்று கோப்பில் எழுதுதல்.

13.7.1 ஒரு புதிய சாதாரண CSV கோப்பினை உருவாக்குதல்

நீங்கள் **CSV** கோப்பினுள் சேமிப்பதற்காக தரவுகளின் தொகுப்பை கொண்டிருக்கும் போது, இந்த சூழ்நிலையில் கொண்டிருக்கும் போது, இந்த சூழ்நிலையில் எதிர்முறையாக **write()**செயற்கூறை பயன்படுத்தவும்.

csv.writer() செயற்கூறு **writer** பொருளை திருப்பி அனுப்பும் அது பயனர் தரவை சரங்களை பிரிப்பானுடன் மாற்றி கோப்பு போன்ற பொருளை தரும். **writerow()** செயற்கூறு ஒரு தரவின் வரியை ஒரு குறிப்பிட்ட கோப்பினுள் எழுதும்.

csv.writer() முறையின் தொடரியல்

The syntax for csv.writer() is

```
csv.writer (fileobject, delimiter, fmtparams)
```

இங்கு,

fileobject :-	கோப்பின் பாதையையும் முறைமையும் அனுப்பும்.
delimiter :-	| (pipe) போன்ற நிலையான Dialects பிரிப்பான்களை கொண்டிருக்கும் விருப்ப அளபுருவாகும். மற்றவை நிராகரிக்கப்படும்.
fmtparams :	விருப்ப அளபுருவாகும். skipinitialspace, quoting போன்ற dialects களின் தானமைவான மதிப்புகளை மாற்றியமைக்க செய்ய உதவுகிறது. மற்றவை (மாற்ற) நிராகரிக்கப்படும்

CSV உறுப்பான writer() முறையை பயன்படுத்தி தானமைவு பிரிப்பானான காற்புள்ளியைச் சாதாரண CSV கோப்பினை நீங்கள் உருவாக்க முடியும்.

(எடுத்துக்காட்டு)

Chapter 13.indd 257 30-01-2020 20:48:28

பின்வரும் பைத்தான் நிரலானது தரவுகளின் பட்டியலை "Pupil.csv" எனப்படும் CSV கோப்பிற்கு மாற்றம் செய்யும். மேலும் மதிப்பு பிரிப்பானக , (காற்புள்ளி) பயன்படுத்தப்பட்டுள்ளது.

```
Import csv
csvData = [['Student', 'Age'], ['Dhanush', '17'], ['Kalyani', '18'], ['Ram', '15']]
with open('c:\pyprg\ch13\Pupil.csv', 'w') as CF:
    writer = csv.writer(CF)              # CF என்பது கோப்பு பொருளாகும்
    writer.writerows(csvData)            # csv என்பது List-ன் பெயர்
CF.close()
```

உரைபதிப்பானை பயன்படுத்தி "Pupil.csv" என்ற கோப்பினை திறக்கும் போது அதன் பொருடக்கமானது பின்வருமாறு தோன்றும்.

Student, Age
Dhanush, 17
Kalyani, 18
Ram, 15

மேலே கொடுக்கப்பட்டுள்ள நிரலில் csv.writer() செயற்கூறு எல்லா தரவையும் "csvData" என்ற list சரங்களாக மாற்றி, உள்ளடக்கத்தை ஒரு கோப்பில் பொருளாக உருவாக்கும். writerows () செயற்கூறு எல்லா தரவையும் ஒரு புதிய CSV கோப்பான "Pupil.csv"-ல் எழுதும்.

குறிப்பு

writerow() முறையானது ஒரு சமயத்தில் ஒரு வரிசையை மட்டுமே எழுதும். அனைத்து தரவுகளையும் ஒரே சமயத்தில் எழுத விரும்பினால் writerows() முறையை பயன்படுத்தலாம்.

13.7.2 இருக்கும் கோப்பை மாற்றியமைத்தல்

தற்போதுள்ள கோப்பின் தரவில் சில மாற்றங்களைச் செய்வது அல்லது கூடுதல் தரவைச் சேர்ப்பது மாற்றம் என அழைக்கப்படுகிறது .எடுத்துக்காட்டக "student.csv" கோப்பு பின்வரும் தரவைக் கொண்டுள்ளது.

Roll No, Name, City
1, Harshini, Chennai
2, Adhith, Mumbai
3, Dhuruv, Bangalore
4, Krishna, Tiruchy
5, Venkat, Madurai

மேற்கண்ட நிரலானது "student.csv" கோப்பினில் ஒரு புதிய வரிசையை சேர்க்கும். இப்பயன்பாட்டிற்காக மட்டுமே CSV கோப்பானது 'a' append திறக்கப்படுகிறது. Append முறையானது "student.csv" கோப்பின் கடைசி வரியின் அடுத்து புதிய வரிசையில் மதிப்புகளை எழுதும்.

குறிப்பு

The writerow() முறை ஒரு நேரத்தில் ஒரு வரிசையை எழுதுகிறது. நீங்கள் எல்லா தரவையும் ஒரே நேரத்தில் எழுத வேண்டும் என்றால் நீங்கள் writerows()முறையைப் பயன்படுத்தலாம்.

Chapter 13.indd 258 30-01-2020 20:48:28

13.7.3 ஏற்கனவே உள்ள கோப்பினில் மாற்றங்கள் செய்தல்

ஏற்கனவே உள்ள கோப்பினில் சில மாற்றங்களை செய்தல் அல்லது மேலும் புதிய தரவுகளை சேர்த்தல் ஆகியவை மாற்றம் செய்தல் எனப்படும்.

S No, Name, City,
1, Harshini, Chennai
2, Adhith, Mumbai
3, Dhuruv, Bangalore
4, Krishna, Tiruchy
5, Venkat, Madurai
6, Sajini, Madurai

"student.csv" என்ற கோப்பானது பின்வருமாறு தரவுகளை கொண்டிருக்கும்.

```
import csv
row = ['3', 'Meena','Bangalore']
with open('student.csv', 'r') as readFile:
        reader = csv.reader(readFile)
        lines = list(reader)   # list()- ஒவ்வொரு வரிசையான தரவை ஒரு Listஆக சேமிக்கும்
        lines[3] = row
with open('student.csv', 'w') as writeFile:
        # பயனர் தரவை பிரிப்பானுடன் மாற்றியமைத்து திருப்பி அனுப்பும் writer பொருள்
        writer = csv.writer(writeFile)
        #writerows() செயற்கூறு csv கோப்பில் ஒன்றிக்கு
        writer.writerows(lines)
readFile.close()
writeFile.close()
```

பின்வரும் நிரலானது student.csv என்ற கோப்பில் ஏற்கனவே உள்ள வரிசைகளின் மதிப்புகளை மாற்றம் செய்கிறது.

S No,	Name,	City,
1,	Harshini,	Chennai
2,	Adhith,	Mumbai
3,	menna ,	Bangalore
4,	Krishna,	Tiruchy
5,	Venkat,	Madurai

உரைப்பதிப்பானில் "student.csv" என்ற கோப்பினை திறக்கும் போது வெளியீடானது பின்வருமாறு தோற்றமளிக்கும். மேற்கண்ட நிரலில் "student.csv" என்ற கோப்பின் மூன்றாவது வரிசையானது மாற்றம் செய்யப்பட்டு சேமிக்கப்பட்டுள்ளது. முதலில் csv.reader() என்ற செயற்கூறு மூலம் "student.csv" என்ற

கோப்பானது படிக்கப்படும். list() என்பதன் மூலம் கோப்பின் அனைத்து வரிசைகளும் சேமிக்கப்படும். “lines[3] = row” என்ற கூற்றானது கோப்பினில் உள்ள 3வது வரிசையை row என்பதில் உள்ள பொருளடக்கத்தை மாற்றம் செய்யும். writerows (lines) என்பதை கோப்பு பொருளான writer பயன்படுத்தி “student.csv” கோப்பு மதிப்புகளை list-ல் எழுதும்.

(i) புதிய வரிசையை சேர்த்தல்

சில நேரங்களில் ஏற்கனவே உள்ள CSV கோப்பினில் புதிய வரிசையை சேர்க்க வேண்டிய தேவை இருக்கலாம். புதிய வரிசையானது கோப்பின் இறுதியில் சேர்க்கப்படும் செயல்முறையை ஒரு வரிசை இறுதியில் சேர்த்தல் (appending) எனப்படும்.

பின்வரும் நிரலானது “student.csv” கோப்பில் ஒரு புதிய வரிசையை சேர்க்கும்.

```
import csv
row = ['6', 'Sajini ', 'Madurai']
with open('student.csv', 'a') as CF:      # append முறைமையில் தரவுகளை இறுதியில் சேர்க்க
 writer = csv.writer(CF)
writer.writerow(row)  # writerow() செயற்கூறில் தரவை ஒரு வரிசையாக கோப்பில் எழுத
CF.close()
```

student.csv” கோப்பானது உரைப்பதிப்பானை பயன்படுத்தி திறக்கும் போது அது பின்வருமாறு தோன்றும்.

```
Student, Age
Dhanush, 17
Kalyani, 18
Ram, 15
```

மேற்கண்ட நிரலானது csv.writer () பட்டியலில் உள்ள எல்லா தரவையும் சரங்களாக மாற்றுகிறது மற்றும் உள்ளடக்கத்தை பொருள் போன்ற கோப்பாக உருவாக்குகிறது. writerrowse () முறை அனைத்து தரவையும் புதிய csv கோப்பு “Pupil.csv” க்கு எழுதுகிறது.

? உங்களுக்குத் தெரியுமா?

'w' write முறைமை புதிய கோப்பினை உருவாக்க உதவுகிறது. ஏற்கனவே அக்கோப்பானது உருவாக்கப்பட்டிருந்தால் அக்கோப்பின் பெயரில் புதிய கோப்பினை உருவாக்கும். 'a' append முறைமையானது கோப்பானது ஏற்கனவே உருவாக்கப்பட்டிருந்தால் அதன் இறுதியில் தரவினை சேர்க்கும். இல்லையெனில் புதிய கோப்பினை உருவாக்கும்.

குறிப்பு

writerow() செயற்கூறானது 1-பரிமாண தரவினையும் மற்றும் writerows() செயற்கூறானது இருபரிமாண தரவுகளையும் கோப்பினில் எழுதும்.

13.7.3 மேற்கோள் குறிகளுடன் CSV கோப்பு

csv.register_dialect() இனக்குழுவை பயன்படுத்தி புதிய Dialectஐ பதிவு செய்வதன் மூலம் மேற்கோள் குறிகளுடன் csv கோப்பினை எழுதலாம். பின்வரும் நிரலானது இதை விளக்குகிறது.

```
import csv
info = [['SNO', 'Person', 'DOB'],
['1', 'Madhu', '18/12/2001'],
['2', 'Sowmya','19/2/1998'],
['3', 'Sangeetha','20/3/1999'],
['4', 'Eshwar', '21/4/2000'],
['5', 'Anand', '22/5/2001']]
csv.register_dialect('myDialect',quoting=csv.QUOTE_ALL)
with open('c:\pyprg\ch13\person.csv', 'w') as f:
        writer = csv.writer(f, dialect='myDialect')
        for row in info:
            writer.writerow(row)
f.close()
```

"person.csv" கோப்பினை திறக்கும் போது பின்வரும் வெளியீட்டை நாம் பெறலாம்:

```
"SNO", "Person", "DOB"
"1", "Madhu", "18/12/2001"
"2", "Sowmya", "19/2/1998"
"3", "Sangeetha", "20/3/1999"
"4", "Eshwar", "21/4/2000"
"5", "Anand", "22/5/2001"
```

மேற்கண்ட நிரலில் புதிய Dialect ஆனது mydialect என்ற பெயரில் அறிவிக்கப்பட்டுள்ளது. Quoting=csv.QUOTE_ALL என்ற dialect ஆனது அனைத்து மதிப்புகளையும் இரட்டை மேற்கோள் குறிகளுடன் எழுத அனுமதிக்கின்றது.

13.7.4 பயனர் விரும்பும் பிரிப்பானுடன் கூடிய CSV கோப்பு

பிரிப்பான் என்பது சரம் ஆகும். இது புலங்களை பிரிக்க பயன்படுகிறது. இதன் கொடாநிலை மதிப்பு காற்புள்ளி(,) ஆகும். **csv.register_dialect()** என்பதன் உதவியுடன் புதிய **dialect()**ஐ அறிவித்து பயனர் விரும்பும் பிரிப்பானை CSVகோப்பினில் பயன்படுத்த முடியும் பின்வரும் எடுத்துக்காட்டு நிரலானது **pipe(|)** பிரிப்பானை பயன்படுத்தி எழுதப்பட்டுள்ளது.

```
import csv
info = [['SNO', 'Person', 'DOB'],
['1', 'Madhu', '18/12/2001'],
['2', 'Sowmya','19/2/1998'],
['3', 'Sangeetha','20/3/1999'],
['4', 'Eshwar', '21/4/2000'],
['5', 'Anand', '22/5/2001']]
csv.register_dialect('myDialect',delimiter = '|')
with open('c:\pyprg\ch13\dob.csv', 'w') as f:
        writer = csv.writer(f, dialect='myDialect')
        for row in info:
            writer.writerow(row)
f.close()
```


Chapter 13.indd 261 30-01-2020 20:48:29

"dob.csv" கோப்பினை நாம் திறக்கும் போது, பின்வருமாறு வெளியீட்டை பெறலாம்.

```
SNO|Person|DOB
1|Madhu|18/12/2001
2|Sowmya|19/2/1998
3|Sangeetha|20/3/1999
4|Eshwar|21/4/2000
5|Anand|22/5/2001
```

மேற்கண்ட pipe(|) பிரிப்பானைக் கொண்டு புதிய dialect ஆனது அறிவிக்கப்பட்டுள்ளது. அதன் பிறகு "info" list -யை CSV கோப்பினுள் எழுதப்பட்டுள்ளது.

குறிப்பு

dialect அளபுருவான skipinitialspace என்பதன் மதிப்பு True-ஆக இருக்கும் போது, பிரிப்பானை தொடர்ந்து வரும் வெற்று இடைவெளிகளை நீக்கும் இதன் கொடாநிலை மதிப்பு False ஆகும்.

13.7.5 வரி முறிப்பானுடன் கூடிய CSV கோப்பு

வரி முறிப்பானானது வரியினை முடித்து வைக்கப் பயன்படுகிறது. இதன் கொடாநிலை மதிப்பு **\r** அல்லது **\n** ஆகும். CSV செயற்கூறில் உள்ள **csv.register_dialect()** என்ற இனக்குழுவை பயன்படுத்தி புதிய **dialect** ஐ அறிவிப்பதன் மூலம் வரி முறிப்பானை கொண்டு CSV கோப்பினை எழுத முடியும்.

```
import csv
Data = [['Fruit', 'Quantity'], ['Apple', '5'], ['Banana', '7'], ['Mango', '8']]
csv.register_dialect('myDialect', delimiter = '|', lineterminator = '\n')
with open('c:\pyprg\ch13\line.csv', 'w') as f:
    writer = csv.writer(f, dialect='myDialect')
    writer.writerows(Data)
f.close()
```

line.csv கோப்பினை திறக்கும் போது நாம் பின்வரும் வெளியீட்டை பெற முடியும்.

```
Fruit|Quantity
Apple|5
Banana|7
Mango|8
```

மேற்கண்ட நிரலில் புதிய dialect ஆன "myDialect uses the delimiter='|' என்பது பிரிப்பானாக |(pipe) குறியீட்டாக கொண்டுள்ளது. இது புலங்களை பிரிக்க பயன்படுகிறது. வரி முறிப்பானானது line terminator='\r\n\r\n' வரிசையை பிரிப்பதுடன் ஒவ்வொரு இரண்டு வரிகளுக்கு அடுத்து தரவினை வெளிப்படுத்தும்.

குறிப்பு

பைத்தான் CSV செயற்கூறானது \r\n, \n அல்லது \r ஆகியவற்றை மட்டுமே வரி முறிப்பானாக ஏற்றுக் கொள்ளும்.

13.7.6 மேற்கோள் குறிகளுடன் கூடிய CSV கோப்பு

CSV செயற்கூறினில் உள்ள **csv.register_dialect()** என்ற இனக்குழுவை பயன்படுத்தி புதிய dialect-ஐ அறிவிப்பதன் மூலம் மேற்கோள் குறிகளை csv கோப்பினில் பயன்படுத்த முடியும்.

```
import csv
csvData = [['SNO','Items'], ['1','Pen'], ['2','Book'], ['3','Pencil']]
csv.register_dialect('myDialect',delimiter = '|',quotechar = '"', quoting=csv.
    QUOTE_ALL)
with open('c:\pyprg\ch13\quote.csv', 'w') as csvFile:
    writer = csv.writer(csvFile, dialect='myDialect')
    writer.writerows(csvData)
print("writing completed")
csvFile.close()
```

"quote.csv" என்ற கோப்பினை நாம் notepad உரைபதிப்பானில் திறக்கும் போது வெளியீடானது பின்வருமாறு தோன்றும்:

```
"SNO|"Items
1|"Pen"
2|"Book"
3|"Pencil"
```

மேற்கண்ட நிரலில் myDialect ஆனது pipe(|) குறியீட்டை பிரிப்பானாகவும் quotechar ஆனது இரட்டை மேற்கோள் குறியையும் கோப்பினுள் எழுதும்.

13.7.7 CSV கோப்பினை Dictionary-ல் எழுதுதல்

CSV கூறினில் உள்ள DictWriter() இனக்குழுவை பயன்படுத்தி csv கோப்பினை Dictionary-ல் எழுதலாம்.

திறவுகோளானது புலப்பெயர் அளபுருக்களால் வழங்கப்படுகிறது. பின்வரும் நிரலானது ஒரு கோப்பினுள் Dictionary-யை எழுதுவதற்கு பயன்படுகிறது.

```
import csv
data = [{'MOUNTAIN' : 'Everest', 'HEIGHT': '8848'},
    {'MOUNTAIN' : 'Anamudi ', 'HEIGHT': '2695'},
    {'MOUNTAIN' : 'Kanchenjunga', 'HEIGHT': '8586'}]
with open('c:\pyprg\ch13\peak.csv', 'w') as CF:
    fields = ['MOUNTAIN', 'HEIGHT']
    w = csv.DictWriter(CF, fieldnames=fields)
    w.writeheader()
    w.writerows(data)
print("writing completed")
CF.close()
```


Chapter 13.indd 263 30-01-2020 20:48:29

"peak.csv" என்ற கோப்பினை **notepad** உரைப்பதிப்பானில் திறக்கும் போது பின்வருமாறு வெளியீட்டை பெறலாம்.

MOUNTAIN,	**HEIGHT**
Everest,	8848
Anamudi,	2695
Kanchenjunga,	8586

"peak.csv" என்ற கோப்பினை NOTEPAD உரைப்பதிப்பானில் திறக்கும் போது பின்வருமாறு வெளியீட்டை பெறலாம்.

மேற்கண்ட நிரலில் புலப்பெயரானது csv கோப்பின் ஒவ்வொரு நெடுவரிசையின் தலைப்பாக உள்ளது. DictWriter() பயன்படுத்தி அகராதியில் உள்ள தரவுகளை "peak.csv" கோப்பினில் எழுதலாம்.

13.7.7.1 தனிப்பயன் Dialects உடன் CSV கோப்பாக Dictionary-ல் எழுதுதல்

```
import csv
csv.register_dialect('myDialect', delimiter = '|', quoting=csv.QUOTE_ALL)
with open('c:\pyprg\ch13\grade.csv', 'w') as csvfile:
    fieldnames = ['Name', 'Grade']
    writer = csv.DictWriter(csvfile, fieldnames=fieldnames, dialect="myDialect")
        writer.writeheader()
    writer.writerows([{'Grade': 'B', 'Name': 'Anu'},
                      {'Grade': 'A', 'Name': 'Beena'},
                      {'Grade': 'C', 'Name': 'Tarun'}])
print("writing completed")
```

grade.csv கோப்பினை திறக்கும் போது அது பின்வருமாறு வெளியீட்டை கொண்டிருக்கும்.

"Name"|	**"Grade"**
"Anu"|	"B"
"Beena"|	"A"
"Tarun"|	"C"

மேற்கண்ட நிரலில் **myDialect** என்ற தனிப்பயன் **pipe(|) dialect** ஆனது பிரிப்பானாக **pipe(|)** குறியீட்டை கொண்டுள்ளது.

மேலும் புலப்பெயரானது **csv** கோப்பானது நெடுவரிசையின் பெயர்களாக பயன்படுத்தப்படுகிறது. இறுதியாக **DictWriter()** ஐ பயன்படுத்தி அகராதியில் உள்ள தரவுகளை **"grade.csv"** கோப்பினில் எழுதலாம்.

Chapter 13.indd 264 30-01-2020 20:48:29

13.7.8 இயங்கு நேரத்தில் தரவினை பெற்று CSV கோப்பினில் எழுதுதல்

நிரலின் இயங்கு நேரத்தில் கூட நாம் தரவுகளை விசைப்பலகை வழியாக பெற்று CSV கோப்பினில் எழுதலாம். எடுத்துக்காட்டாக "dynamicfile.csv" என்ற கோப்பு பின்வரும் நிரலானது பயனரிடமிருந்து விசைப்பலகை, வழியாக தரவினை பெற்று கோப்பினில் எழுதுகிறது, மேலும் அது கோப்பின் உள்ளடக்கத்தையும் வெளிப்படுத்தும்.

```
import csv
with open('c:\pyprg\ch13\dynamicfile.csv', 'w') as f:
    w = csv.writer(f)
    ans='y'
    while (ans=='y'):
        name = input("Name?: ")
        date = input("Date of birth: ")
        place = input("Place: ")
        w.writerow([name, date, place])
        ans=input("Do you want to enter more y/n?: ")
    F=open('c:\pyprg\ch13\dynamicfile.csv','r')
    reader = csv.reader(F)
    for row in reader:
        print(row)
    F.close()
```

வெளியீடு

```
Name?: Nivethitha
Date of birth: 12/12/2001
Place: Chennai
Do you want to enter more y/n?: y
Name?: Leena
Date of birth: 15/10/2001
Place: Nagercoil
Do you want to enter more y/n?: y
Name?: Padma
Date of birth: 18/08/2001
Place: Kumbakonam
Do you want to enter more y/n?: n
['Nivethitha', '12/12/2001', 'Chennai']
[]
['Leena', '15/10/2001', 'Nagercoil']
[]
['Padma', '18/08/2001', 'Kumbakonam']
```

MS-Excel வெளியீடு

	A	B	C
1	Nivethitha	12/12/2001	Chennai
2			
3	Leena	15/10/2001	Nagercoil
4			
5	Padma	18/08/2001	Kumbakonam
6			

Chapter 13.indd 265 30-01-2020 20:48:29

நினைவில் கொள்க

- CSV கோப்பானது ஒவ்வொரு வரிசையிலும் காற்புள்ளி அல்லது வேறு ஏதேனும் ஒரு பிரிப்பானைக் கொண்டு பிரிக்கப்பட்ட பல புலங்களை கொண்டுள்ள பயனர் படிக்கக் கூடிய உரை கோப்பாகும்.
- எக்ஸெல் கோப்பானது ஒரு இருமநிலை வடிவ கோப்பாகும். அதே சமயம் csv கோப்பானது ஒரு எளிய உரை வடிவ கோப்பாக இருக்கும்.
- CSV கோப்பினை படிக்க இரண்டு வழிமுறைகள் உள்ளன. **csv.reader()** செயற்கூறை பயன்படுத்துதல் மற்றும் **DictReader** இனக்குழுவை பயன்படுத்துதல்.
- எழுதுதல் மற்றும் படித்தல் செயல்பாடுகளில் CSV கோப்பின் கொடாநிலை உரை முறையாகும்.
- உரை அல்லாத கோப்பு உருவப்படம் அல்லது இயங்குநிலை கோப்புகளை கையாளும் போது இருமநிலை முறைமையானது பயன்படுத்தப்படும்.
- நினைவகத்தில் பயன்பாட்டில் இல்லாத பொருள்களை **(Objects)** சேகரிக்கவும் மற்றும் அந்த நினைவக பகுதியை சுத்தம் செய்யவும் பைத்தானில் தேவையற்ற நினைவக பகுதியை சேகரிக்கும் வசதி (Garbage collector) உள்ளது.
- close() முறையானது கோப்புடன் இணைக்கப்பட்டுள்ள வளங்களை விடுவிக்கும்.
- CSV கோப்பானது தானமைவாக எக்ஸெல் பயன்பாட்டில் திறக்கும்.
- CSV நூலகமானது CSV கோப்பினில் உள்ள தரவுகளை படிப்பதற்கும், CSV கோப்பினில் தரவுகளை எழுதுவதற்கும் செயல்படுத்துவதற்கும் தேவையான பொருள்களையும் மற்றும் குறிமுறைகளையும் கொண்டிருக்கும்.
- “skipinitialspace” பிரிப்பானிற்கு பிறகு உள்ள வெற்று இடைவெளியை நீக்க பயன்படும்.
- ஒன்றிக்கு மேற்பட்ட நெடுவரிசையை வரிசையாக்கம் செய்ய operator.itemgetter() பயன்படுகிறது.
- Dictionary யில் தரவுகளை குறிக்க யில் தரவுகளை குறிக்க CSV செயற்கூறின் DictReader() என்ற இனக்குழு ஒரு பொருளை உருவாக்குகின்றது.
- CSV கோப்பில் உள்ள தனிப்பயனாக்கப்பட்ட பிரிப்பான் csv.register_dialect() மூலம் படிக்கப்படுகிறது.
- itemgetter() ஒன்றிக்கு மேற்பட்ட சுட்டெடன் மூலம் ஒன்றிற்கு மேற்பட்ட நெடுவரிசைகளை வரிசையாக்கம் செய்ய பயன்படும்.
- csv.reader மற்றும் csv.writer list/tuple-ல் வேலை செய்யும், ஆனால் csv.DictReader மற்றும் csv. DictWriter dictionary-ல் வேலை செய்யும் .
- csv.DictReader மற்றும் csv.DictWriter கூடுதல் அளபுருக்காக புலத்தின் பெயரை எடுத்துக்கொண்டு அவற்றை Dictionary-யின் திறவுகோளாக பயன்படுத்துகிறது.

Chapter 13.indd 266 30-01-2020 20:48:29

- dict() செயற்கூறு dictionary வடிவத்தில் தரவினை எந்தவொரு வரிசையிலும் அமைக்காமல் வெளியீடும்.
- csv.writer() செயற்கூறு பயனரின் தரவை பிரிப்பானுடன் கூடிய சரங்களாக மாற்றியமைக்க ஒரு writer பொருளை திருப்பி அனுப்பும்.
- writerow() செயற்கூறு ஒரு வரிசை மட்டுமே ஒரு சமயத்தில் எழுதும். writerows() செயற்கூறு அனைத்து தரவையும் ஒரே முறையில் எழுதும்.
- கோப்பின் இறுதியில் ஒரு வரிசையை சேர்பதை ஒரு வரிசை சேர்த்தல் (appending) எனப்படும்.

செய்முறைப் பயிற்சி

1. Namelist.csv கோப்பை படித்து அதில் உள்ள பெயர் புலத்தில் வரிசையாக்கம் செய்து அதை list-ல் எழுதி வெளியீடு செய்வதற்கான பைத்தான் நிரல் எழுதுக.

	A	B	C
1	SNO	NAME	OCCUPATION
2	1	NIVETHITHA	ENGINEER
3	2	ADHITH	DOCTOR
4	3	LAVANYA	SINGER
5	4	VIDHYA	TEACHER
6	5	BINDHU	LECTURER

2. ஐந்து மாணவர்களின் ஐந்து பாடத்தின் மதிப்பெண்களை மாணவரின் பெயருடன் உள்ளீடு செய்ய பைத்தான் நிரலை எழுதுக. மேலும் மதிப்பெண்ணின் கூட்டுத்தொகையை கண்டறிந்து எல்லா விவரங்களையும் CSV கோப்பில் எழுது.

Chapter 13.indd 267 30-01-2020 20:48:29

மதிப்பீடு

பகுதி–அ

8V9HV5

சரியான விடையை தேர்ந்தெடுத்து எழுதுக **(1 மதிப்பெண்)**

1. CSV கோப்பானது பின்வருபவனவற்றுள் எவ்வாறு அழைக்கப்பட்டுள்ளது ....

   (அ) Flat File (ஆ) 3D File

   (இ) String File (ஈ) Random File

2. CRLF என்பதன் விரிவாக்கம்

   (அ) Control Return and Line Feed

   (ஆ) Carriage Return and Form Feed

   (இ) Control Router and Line Feed

   (ஈ) Carriage Return and Line Feed

3. பின்வருபவனவற்றுள் எந்த செயற்கூறானது CSV கோப்பினில் பல்வேறு செயல்பாடுகளை செய்ய பைத்தானால் வழங்கப்பட்டுள்ளது ஆகும்?

   (அ) py (ஆ) xls

   (இ) csv (ஈ) os

4. உருவப்படம் அல்லது இயங்குநிலை கோப்பு போன்று உரை அல்லாத கோப்புகளை கையாள பின்வரும் எந்த முறைமையானது பயன்படுகிறது?

   (அ) உரை (ஆ) இருமநிலை

   (இ) xls (ஈ) csv

5. கோப்பினில் ஒரு வரிசையை தவிர்க்க பயன்படும் கட்டளை

   (அ) next() (ஆ) skip()

   (இ) omit() (ஈ) bounce()

6. பின்வருபவனவற்றுள் CSV செயற்கூறில் writer() முறையால் வழங்கப்பட்டுள்ள வரிமுறிப்பான் எது?

   (அ) Line Terminator (ஆ) Enter key

   (இ) Form feed (ஈ) Data Terminator

7. பின்வரும் நிரலின் வெளியீடு யாது?

   “city.csv” என்ற கோப்பில் கீழேயுள்ள விவரங்களை கொண்டிருப்பின்

   ```
   import csv
   d=csv.reader (open("C.\PYPRG\ch/3City.CSV")
   ```


Chapter 13.indd 268 30-01-2020 20:48:29

```
chennai,mylapore
mumbai,andheri
next(d)
for row in d:
    print(row)
```

(அ) chennai,mylapore

(ஆ) mumbai,andheri

(இ) chennai
mumbai

(ஈ) chennai,mylapore
mumbai,andheri

8. Dictionary தரவுகளை குறிக்க இவற்றுள் எது ஒரு பொருளை உருவாக்குகின்றது?

(அ) listreader()

(ஆ) reader()

(இ) tuplereader()

(ஈ) DictReader ()

9. ஏற்கனவே உள்ள கோப்பில் உள்ள தரவுகளில் சில மாற்றங்கள் செய்வதும் அல்லது மேலும் தரவை சேர்ப்பது இவ்வாறு அழைக்கலாம்

(அ) பதிப்பித்தல்

(ஆ) இறுதியில் சேர்த்தல்

(இ) மாற்றம் செய்தல்

(ஈ) திருத்துதல்

10. test.csv என்ற கோப்பில் பின்வரும் நிரல் என்ன விவரத்தை எழுதும்.

```
import csv
D = [['Exam'],['Quarterly'],['Halfyearly']]
csv.register_dialect('M',lineterminator = '\n')
with open('c:\pyprg\ch13\line2.csv', 'w') as f:
    wr = csv.writer(f,dialect='M')
    wr.writerows(D)
f.close()
```

(அ) Exam Quarterly Halfyearly

(ஆ) Exam Quarterly Halfyearly

(இ) E
Q
H

(ஈ) Exam,
Quarterly,
Halfyearly

பகுதி–ஆ

அனைத்து வினாக்களுக்கும் விடையளி (2 மதிப்பெண்)

1. CSV கோப்பு என்றால் என்ன?
2. பைத்தான் மூலம் CSV கோப்பை படிப்பதற்கான இரு வழிகளை குறிப்பிடுக.
3. கோப்பின் கொடாநிலை முறைமைகளை குறிப்பிடுக.
4. next() செயற்கூறின் பயன்பாடு என்ன?
5. csv கோப்பில் ஒன்றிக்கு மேற்பட்ட நெடுவரிசையை எவ்வாறு வரிசையாக்கம் செய்வாய்? எடுத்துக்காட்டுத் தருக.

Chapter 13.indd 269 30-01-2020 20:48:29

பகுதி–இ

அனைத்து வினாக்களுக்கும் விடையளி **(3 மதிப்பெண்)**

1. பைத்தானிலுள்ள open() செயற்கூற்றை பற்றி குறிப்பு எழுதுக. மேலும் இதன் இரண்டு வழிமுறைகளின் வேறுபாடுகள் என்ன?
2. ஏற்கனவே உள்ள கோப்பில் மாற்றம் செய்யும், பைத்தான் நிரலை எழுதுக.
3. காற்புள்ளியை (,) தானமைவு பிரிப்பானாக கொண்டுள்ள CSV கோப்பினை படிப்பதற்கான பைத்தான் நிரலை எழுதுக.
4. write மற்றும் append mode முறைமைகளின் வேறுபாடு என்ன?
5. reader() மற்றும் DictReader() செயற்கூற்றின் வேறுபாடு என்ன?

பகுதி–ஈ

அனைத்து வினாக்களுக்கும் விடையளி **(5 மதிப்பெண்)**

1. எக்ஸெல் மற்றும் CSV கோப்பின் வேறுபாடு என்ன?
2. பல்வேறு கோப்பு முறைமைகளின் பொருள்களை பட்டியலிடுக.
3. பைத்தானில் ஒரு கோப்பை படிப்பதற்கான பல்வேறு வழிமுறைகளை எழுதுக.
4. தனிப்பயனாக்கம் பிரிப்பானுடன் கூடிய CSV கோப்பை எழுதுவதற்கான பைத்தான் நிரலை எழுதுக.
5. CSV கோப்பிலுள்ள தரவை வடிவமைப்பதற்கு பின்பற்றவேண்டிய விதிமுறைகளை எழுதுக.

REFERENCES

1. *Python for Data Analysis, Data Wrangling with Pandas, NumPy, and IPython By William McKinney*
2. *CSV File Reading and Writing - Python 3.7.0 documentation*
3. *https://docs.python.org*

Chapter 13.indd 270 30-01-2020 20:48:29

அலகு V

பாடம் 14

பைத்தானில் C++ நிரல்களை இறக்கம் செய்தல்

XNH79W

கற்றலின் நோக்கங்கள்

இந்த பாடப்பகுதியை கற்றபின் மாணவர்கள பின்வருவனவற்றை அறிந்து கொள்வார்கள்

- உறை இடுதல் (wrapping) என்றால் என்ன என்பதை புரிந்துக் கொள்ளுதல்.
- C++ செயற்கூறுகளையும் இனக்குழுக்களையும் பைத்தான் நிரல்களுக்குள் தருவித்து கொள்ளுதல்.
- இருநிரலாக்க மொழிகளிலும் செயலாக்கவல்ல சூழலை உருவாக்குதல்.
- பைத்தான் நிரல்களை இயக்குதல் மற்றும் பிழைத்திருத்தம் செய்தல்.

14.1 அறிமுகம்

பைத்தான் மற்றும் C++ ஒரு பொதுப்பயன் நிரலாக்க மொழியாகும். இருப்பினும் பைத்தான் C++ மொழியைக் காட்டிலும் முற்றிலும் மாறுபட்டது.

வ.எண்	பைத்தான்	C++
1	பொதுவாக, பைத்தான் ஒரு 'வரி மொழி மாற்றி' மொழியாகும்.	பொதுவாக C++ மொழி, ஒரு தொகுப்பு மொழியாகும்.
2	பைத்தான் மாறும் தன்மைக் கொண்ட மொழியாகும்.	C++ நிலையான தன்மைக் கொண்ட தொகுக்கப்பட்ட மொழி.
3	மாறியை அறிவிக்கும் போது, அது சார்ந்த தரவின வகையை குறிப்பிடத் தேவையில்லை	மாறியை அறிவிக்கும் போது, தரவினத்தை குறிப்பிட வேண்டும்.
4	இது Scripting மற்றும் பொதுப் பயன் மொழியென இருவகையிலும் செயல்பட முடியும்.	இது ஒரு பொதுப்பயன் நிரலாக்க மொழியாகும்.

ஆயினும், இவ்விரு மொழிகளும் ஒன்றுக்-கொன்று சரியான வகையில் பொருந்திக் கொள்ளும். பைத்தான் , பெரும்பாலும் scripting அல்லது "glue" மொழியாக பயன்படுத்தப்படுகிறது. அதாவது, மேல் நிலையிலுள்ள நிரல் பெரும்பாலும் C அல்லது C++ மொழிகளில் எழுதப்பட்டுள்ள செயல் முறைகளை அழைக்கும். இது தருக்க கருத்தை தற்போதைய குறிமுறையில் (எடுத்துக்காட்டாக C++ மொழியில் எழுதப்பட்டுள்ள நிரல்) எழுதும் போது உதவிகரமாக இருக்கும். ஆனால், பைத்தான் நிரலின் மூலம் அழைத்து, கையாளலாம்.

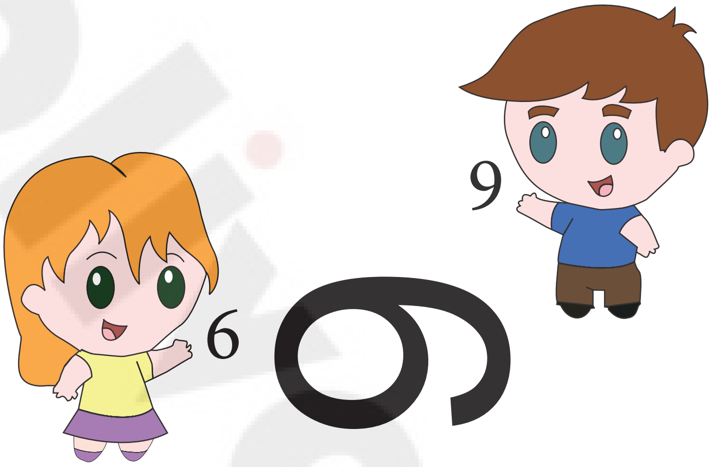
Chapter 14.indd 271 10-02-2020 20:04:00

14.2 Scripting மொழி

ஒரு scripting மொழி என்பது பிற நிரலாக்க மொழிகளுடன் ஒருங்கிணைப்பதற்கும், தொடர்பு கொள்வதற்கும் வடிவமைக்கப்பட்ட ஒரு நிரலாக்க மொழியாகும். ஜாவாஸ்கிரிப்ட், VBஸ்கிரிப்ட், PHP, பெர்ல், பைத்தான், ரூபி, ASP மற்றும் Tcl ஆகியவை மிகவும் பரவலாக பயன்படுத்தப்படும் Scripting மொழியாகும். பொதுவாக, ஒரு Scripting மொழி மற்ற நிரலாக்க மொழிகளுடன் இணைக்கப்பட்டு பயன்படுத்தப்படுவதால், பெரும்பாலும், அவை HTML, ஜாவா அல்லது C++ மொழிகளுடன் இணைந்து காணப்படுகின்றது.

14.2.1 ஸ்கிரிப்டிங் மற்றும் நிரலாக்க மொழிகளுக்கிடையே உள்ள வேறுபாடு

Scripting மொழி மற்றும் நிரலாக்க மொழி கீழ்க்காணும் வரைப்படத்தைப் போன்று காணப்படும்.



அடிப்படையில், அனைத்து Scripting மொழிகளும் நிரலாக்க மொழிகளாகும். இரண்டுக்குமிடையே தத்துவார்த்தமான வேறுபாடு என்னவெனில், Scripting மொழிக்கு தொகுத்தல் படிநிலைத் தேவைப்படாது, மாறாக விளக்கம் தேவைப்படும்.. உதாரணத்திற்கு, ஒரு C++ நிரல் இயக்குவதற்கும் முன்பாக, தொகுக்கப்பட வேண்டும். அதே சமயம் ஜாவாஸ்கிரிப்ட் அல்லது பைத்தான் போன்ற Scripting மொழிகள் தொகுக்கப்பட்ட தேவையில்லை. நிரலாக்க மொழிக்கு தொகுப்பான் / நிரல்பெயர்ப்பி தேவைப்படுவது போல Scripting மொழிக்கு வரி மொழி மாற்றித் தேவைப்படுகிறது. கொடுக்கப்பட்ட மொழி நிரலாக்க மொழியா அல்லது Scripting மொழியா என்பது அவை பயன்படுத்தப்படும் சூழலைப் பொறுத்தது.

14.3 Scripting மொழிகளின் பயன்பாடுகள்

1. ஒரு நிரலில் சில செயல்பாடுகளை தானியங்குப்படுத்துதல்.
2. தரவு தொகுப்பிலிருந்து தகவலைப் பிரித்தெடுத்தல்.
3. பழமையான நிரலாக்க மொழிகளுடன் ஒப்பிடும் போது, குறைந்த நிரல் குறிமுறையைக் கொண்டது.
4. பயன்பாடுகளுக்கு புதிய செயல்பாடுகளை கொண்டு வர முடியும். மேலும், சிக்கலான அமைப்புகளை ஒருங்கமைக்க முடியும்.

பைத்தான் என்பது உண்மையில், எந்த ஒரு தற்போதைய கணினி இயக்க அமைப்பிலும் பயன்படுத்தக்கூடிய ஒரு வரிமொழி மாற்று, உயர்நிலை, பொது பயன்பாட்டு, நிரலாக்க மொழி ஆகும். இது உரை, எண்கள், படங்கள் மற்றும் அறிவியல் சார்ந்த தரவுகளைச் செயலாக்கப்பயன்படுகிறது. தற்போது, பெரும்பான்மையான, பெரிய பயன்பாடுகள் பிரத்யேகமாக பைத்தான் மொழியில் எழுதப்படுகின்றன.

Chapter 14.indd 272 10-02-2020 20:04:01

14.4 C++மீது பைத்தான் பண்புக்கூறுகள்

- பைத்தான் தேவையற்ற (garbage) மதிப்புகளைச் சேகரிக்கும் தானியங்கியைப் பயன்படுத்துகிறது.. இந்த பண்புக்கூறு C++ ல் கிடையாது.
- C++ நிலையான வகையைச் சார்ந்த மொழி, ஆனால் பைத்தான் ஒரு மாறக்கூடிய வகையைச் சார்ந்த மொழியாகும்.
- பைத்தான் வரி மொழி மாற்றி மூலம் இயங்குகிறது. ஆனால் C++ மொழி முன் தொகுக்கப்பட்டது.
- C++ நிரல் குறிமுறையைக் காட்டிலும் பைத்தான் குறிமுறை 5லிருந்து 10 தடவைகள் (மடங்குகள்) குறைவானது.
- பைத்தானில், வெளிப்படையாக தரவினங்களை அறிவிக்க தேவையில்லை. ஆனால் C++-ல் அவை அறிவிக்கப்பட வேண்டும்.
- பைத்தானில், ஒரு செயற்கூறு எந்த வகை செயலுருபையும் ஏற்கும். மேலும், முன்னதாக எந்த ஒரு அறிவிப்பும் இல்லாமல் பல மதிப்புகளை திருப்பியனுப்பும். ஆனால் C++ return கூற்று ஒரே ஒரு மதிப்பை மட்டுமே திருப்பியனுப்பும்.

குறிப்பு

பைத்தான், நினைவக இடத்தை விடுவிக்க, தேவையற்ற பொருட்களை (உள்ளமைக்கப்பட்ட வகைகள் அல்லது இனக்குழு சான்றுருக்கள்) தானமைவாக நீக்குகிறது.அவ்வப்போது, பைத்தான், நினைவகப் பகுதிகளை விடுவித்து, பயன்பாட்டில் இல்லாதவைகளை மீட்டெடுக்கும் செயல்முறை தேவையற்ற மதிப்புகளை சேகரித்தல் (Garbage collection) என்றழைக்கப்படுகிறது.

14.5 பைத்தானில் C++ கோப்புகளைத் தருவித்துக் கொள்ளுதல்

பைத்தான் நிரலில் C++ நிரலை தருவித்துக் கொள்ளுதலைப் பைத்தானில் C++ -யை உறைஇடுதல் என்கிறோம். C++ நிரல்களுக்கான பைத்தான் இடைமுகங்களை அல்லது உறை இடுதலை பல வழிகளில் உருவாக்கலாம். பொதுவாக பயன்படுத்தப்படும் இடைமுகங்களாவன

- Python-C-API (API-Application Programming Interface C நிரல்களுடன் தொடர்பு கொள்ள)
- Ctypes (c நிரல்களுடன் தொடர்பு கொள்ள)
- SWIG (Simplified Wrapper Interface Generator- C மற்றும் C++ இரண்டு மொழிகளுக்கும்)
- Cython (C-நீட்டிப்புக்களை எழுதுவதற்கான ஒரு பைத்தான் போன்ற மொழியாகும்.)
- Boost. Python (Python மற்றும் C++ தொடர்பு கொள்வதற்கான கட்டமைப்பு)
- MinGW(விண்டோஸ்-க்கான குறைந்த பட்ச GNU)

14.5.1 MinGW இடைமுகம்

MinGW இயக்க நேர தலைப்புக் கோப்புகளின் தொகுப்பைக் குறிக்கிறது. விண்டோஸ் இயக்க முறைமையில் இயங்கும் வகையில், C , C++ மற்றும் FORTRAN நிரல் குறிமுறைகளைத் தொகுக்கவும், இணைக்கவும் இது பயன்படுகிறது.

MGPVNN

Chapter 14.indd 273 10-02-2020 20:04:01

MinGw-W64 (MinGW இன் பதிப்பு) windows இன் C++ க்கு சிறந்த தொகுப்பான்/ நிரல் பெயர்ப்பி ஆகும். C++ நிரல்களை தொகுத்து, இயக்க, விண்டோஸ் இயக்க முறைமைக்கு 'g++' தொகுப்பான் தேவை MinGW g++ஐ பயன்படுத்தி பைத்தான் நிரல் மூலம் C++ நிரல்களை தொகுத்து, இயக்க அனுமதிக்கிறது.

C++ நிரல் குறிமுறையைக் கொண்ட பைத்தான் நிரல்களை minGW-w64 project திட்டப்பணி (run முனையம்) மூலம் மட்டுமே இயக்க முடியும். பைத்தான் நிரல்கள் இயக்கப்பட வேண்டிய கட்டளை வரி சாளரத்தை **run முனையம்** திறந்து வைக்கும்.

> g++ என்பது GCC- யை(GNU C தொகுப்பான்) அழைக்கும் நிரல்.இது தானாகவே தேவையான C++ நூலக கோப்புகளை இலக்கு நிரலுடன் இணைக்கிறது.

Refer installation of MinGW in Annexure -2

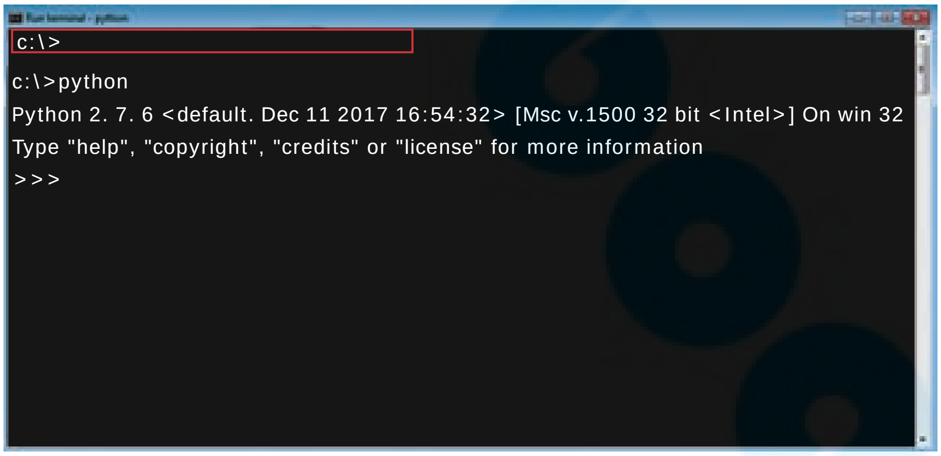
14.5.2 C++ நிரலை பைத்தான் மூலம் இயக்குதல்

1. MinGW run-ன் முனையத்தை இரட்டைக் கிளிக் செய்யவும்
2. பைத்தான் மென்பொருள் அமைந்திருக்கும் (python.exe) கோப்புறைக்கு செல்லவும். இந்த உதாரணத்தில், பைத்தான் கோப்புறை C:\Program Files\OpenOffiice4\Python என்ற அமைவிடத்தில் காணலாம்.

   c:\> கோப்புறையிலிருந்து பைத்தான் அமைந்திருக்கும் கோப்புறைக்கு மாறுவதற்கான தொடரியல்

   cd <absolute path>

இதில் "cd" கட்டளை change directory என்பதையும் absolute path என்பது பைத்தான் நிறுவப்பட்டிருக்கும் முழுமையான பாதையையும் குறிக்கும்.



படம் *14.5.2.1 -11*

இந்த எடுத்துக்காட்டில், பைத்தான் கோப்புறை அமைந்துள்ள இடத்திற்கு சென்று, பின்வரும் கட்டளையைத் தட்டச்சு செய்ய வேண்டும். "cd C:\>". மேற்கண்ட சாளரத்தில் உயர்த்திக்காட்டிருப்பதைக் காண்க.

உதாரணத்திற்கு, Pali.cpp என்ற C++ நிரலை படிக்கும் pycpp.py என்ற பைத்தான் நிரலை எடுத்துக் கொள்ளுங்கள். Pali.cpp என்ற நிரல் ஓர் எண்ணைப் பெற்றுக் கொண்டு, அது பாலிண்ட்ரோமா இல்லையா என்பதை வெளியிடும். உதாரணத்திற்கு, 232 என்ற எண்ணை உள்ளீடாக கொடுத்தால், palindrome என்ற வெளியீடு கிடைக்கும். C++ நிரல் pali நோட்பேடில் தட்டச்சு செய்யப்பட்டு, pali.cpp என்ற பெயரில்

Chapter 14.indd 274 10-02-2020 20:04:01

சேமிக்கப்படும். அதைப் போன்றே, pycpp.py என்று பைத்தான் நிரல் குறிமுறையும் நோட்பேடில் தட்டச்சு செய்யப்பட்டு, pycpp.py என்று சேமிக்கப்படுகிறது.

3. நம் நிரலை இயக்க, run முனையத்தை இரட்டை கிளிக் செய்யவும். பாதையை பைத்தான் கோப்புறை அமைவிடத்திற்கும் மாற்றவும். பைத்தானை இயக்குவதற்கான தொடரியல்,

Python <filename.py> -i <C++ filename without cpp extension>

இதில்,

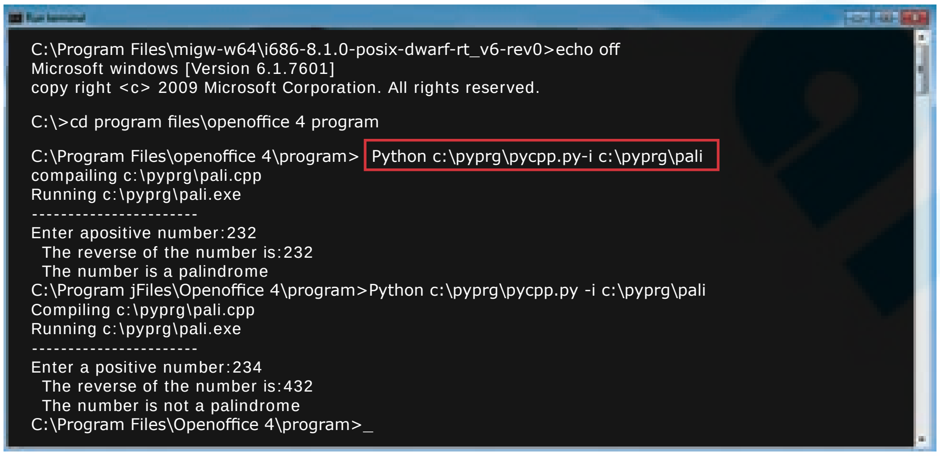
Python	கட்டளை வரியிலிருந்து பைத்தான் நிரலை செயல்படுத்துவதற்கான சிறப்புச் சொல்
filename.py	செயல்படுத்த வேண்டிய பைத்தான் நிரலின் பெயர்
- i	உள்ளீட்டு முறைமை
C++ filename without cpp extension	தொகுக்கப்பட்டு செயல்படுத்தப்பட வேண்டிய C++ நிரலின் பெயர்.

உதாரணத்திற்கு, கட்டளை தூண்டுக்குறியில், "Python pycpp.py –i pali" என்று உள்ளிட்டு enter விசையை அழுத்தவும். தொகுத்தல் வெற்றிகரமாக முடிந்தால் நீங்கள் எதிர்பார்த்த வெளியீடு கிடைக்கும் இல்லையேல் பிழைச் செய்தி வெளியிடப்படும்.

குறிப்பு

இயங்கு (execution) கட்டளையில், உள்ளீட்டு கோப்பிற்கு நீட்டிப்பு தேவைப்படாது. உதாரணத்திற்கு, "pali.cpp" என்பதற்கு பதிலாக "pali" என்ற பெயரைக் கொடுத்தால் போதுமானது.

நாம் pycpp.py மற்றும் pali.cpp என்ற எடுத்துக்காட்டுகள் மூலம் நிரலை இயக்குதல் பற்றி பார்ப்போம். இந்த இரு நிரல்களும் c:\pyprg என்ற கோப்புறையில் இருத்தி வைக்கப்பட்டுள்ளன. இந்த நிரல்கள் பைத்தானில் "exe file" அமைந்திருக்கும் அதே கோப்புறையில் இல்லாவிடில், இயக்க நேரத்தின் போது கோப்புகளின் முழுமையான பாதையைக் குறிப்பிடுதல் அவசியம். வெளியீடு கீழ்க்கண்டவாறு இருக்கும்.



படம்14.2

Chapter 14.indd 275 10-02-2020 20:04:01

குறிப்பு

கட்டளை சாளரத்தில் உள்ள திரையை அழிக்க (clear) செய்ய cls கட்டளையைப்பயன்படுத்தவும்.

நாம் இப்பொழுது C++ நிரல் குறிமுறையைத் தொகுப்பதற்கான பைத்தான் நிரலை எவ்வாறு எழுதுவது என்பதைப் பார்ப்போம்.

14.6 C++ நிரலைத் தருவித்துக் கொள்வதற்கான பைத்தான் நிரல்

பைத்தான் பல கூறுநிலைகளைக் கொண்டுள்ளது. ஒரு சிக்கலுக்கு நிரலர்கள் பல தரப்பட்ட கூறுநிலைகளைத் தங்கள் வசதிக்கேற்ப பயன்படுத்திக் கொள்ள பைத்தான் அனுமதிக்கிறது. இப்பொழுது நாம் கற்றுக் கொள்ளவிருக்கும் பைத்தான் நிரல், அடிப்படை பைத்தான் நிரலில் நீங்கள் அறிந்திராத சில புதிய கட்டளைகளைக் கொண்டுள்ளது. இந்த அதிகாரத்தில், நிரல்கள் இரண்டு வெவ்வேறு மொழிகளின் ஒருங்கிணைப்பைப் பெற்றுள்ளதால், os,sys மற்றும் getopt போன்ற கூறுநிலைகளைத் தருவித்துக் கொள்ள வேண்டும்.

14.6.1 கூறுநிலை

கூறுநிலை நிரலாக்கம் என்பது உங்கள் குறிமுறையை சிறுசிறு பகுதிகளாக பிரிப்பதற்கான மென்பொருள் வடிவமைப்பு நுட்பமாகும். இந்த பகுதிகள் கூறுநிலைகள் என்றழைக்கப்படுகின்றன. இந்த பிரித்தலின் போது கூறுநிலைகள் என்றழைக்கப்படுகிறது. இந்த பிரித்தலின் போது கூறுநிலைகள் மற்ற பிற கூறுநிலைகளின் மீது குறைந்த அளவிலோ அல்லது இல்லாமாலோ சார்ந்திருப்பதை கவனத்தில் கொள்ள வேண்டும். வேறுவகையில், கூறவேண்டுமெனில் சார்பு நிலையை குறைத்தலே நோக்கமாகும்.

ஆனால், நாம் பைத்தானில் கூறுநிலைகளை எவ்வாறு உருவாக்குவோம்? பைத்தானில் கூற்றுகள் மற்றும் வரையறைகளைக் கொண்டுள்ள கோப்பு கூறுநிலை எனப்படுகிறது. பைத்தான் குறிமுறையை கொண்டிருக்கும் கோப்பு எடுத்துக்காட்டாக, factorial.py என்பதனை கூறுநிலை என்றும், factorial என்பது கூறுநிலையின் பெயரைக் குறிக்கும். நாம் கூறுநிலைகளைக் கையாளக்கூடிய, ஒருங்கமைக்கப்பட்ட சிறிய கோப்புகளாக பிரித்து பயன்படுத்துகிறோம். மேலும், கூறுநிலை, குறிமுறையின் மறுபயனபாக்கத்தை வழங்குகிறது. வெவ்வேறு நிரல்களில், நாம் மிகுதியாக பயன்படுத்தும் செயற்கூறு வரையறைகளை நகலெடுப்பதற்கு பதிலாக, அவற்றை ஒரு கூறுநிலையில் வரையறுத்து, தருவித்துக் கொள்ளலாம்.

எடுத்துக்காட்டு

```
def fact(n):
        f=1
        if n == 0:
        return 0
elif n == 1:
        return 1
        else:
                for i in range(1, n+1):
                        f= f*i
                print (f)
```

வெளியீடு

```
        >>>fact (5)
        120
```

மேற்கண்ட எடுத்துக்காட்டு factorial.py என பெயரிடப்பட்டுள்ளது.

Chapter 14.indd 276
10-02-2020 20:04:01

14.6.2 பைத்தானில் கூறுநிலைகளை எவ்வாறு தருவித்துக் கொள்வது?

ஒரு கூறுநிலைக்குள் மற்றொரு கூறுநிலையின் வரையறைகளைத் தருவித்துக் கொள்ளலாம். அவ்வாறு செய்வதற்கு நாம் 'import' என்ற சிறப்பு சொல்லைப் பயன்படுத்த முடியும். நாம் முன்னதாக வரையறுத்த factorial என்ற கூறுநிலையைத் தருவித்துக் கொள்ள, பைத்தான் தூண்டுகுறியில் கீழ்க்கண்டவாறு தட்டச்சு செய்யவும்.

>>> import factorial

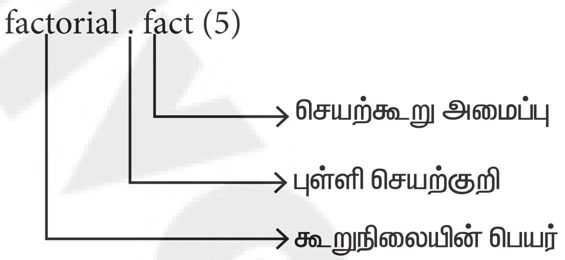
கூறுநிலைக்குள் வரையறுக்கப்பட்ட செயற்கூறுகளை, கூறுநிலையின் பெயரைப் பயன்படுத்தி அணுக முடியும். செயற்கூறுகளை அணுக புள்ளிச் செயற்குறி பயன்படுத்தப்படுகிறது கூறுநிலையிலிருந்து செயற்கூறுகளை அணுகுவதற்கான தொடரியல்

<module name> . <function name>

எடுத்துக்காட்டாக:

>>> factorial.fact(5)

120



பைத்தான் அடிப்படை (உள்ளிணைந்த) கூறுநிலைகள் பலவற்றைக் கொண்டுள்ளது. பயனர் வரையறுத்த கூறுநிலைகளைத் தருவிப்பது போன்றே அடிப்படை கூறுநிலைகளையும் தருவித்துக் கொள்ளலாம். நீங்கள் தற்போது C++ குறிமுறையை இயக்கும் பைத்தான் நிரலுக்கு தேவையான அடிப்படை கூறுநிலைகளைக் காணப் போகிறீர்கள்.

(i) பைத்தானின் sys கூறுநிலை

இந்த கூறுநிலை வரிமொழி மாற்றியால் பயன்படுத்தப்படுகிறது.

sys.argv

sys.argv என்பது பைத்தான் நிரலுக்கு அனுப்பி வைக்கப்படும் கட்டளை வரி செயலுருபுகளின் பட்டியலாகும். argv கட்டளை வரி உள்ளீட்டு வழியாக வரும் உருப்படிகள் அனைத்தையும் கொண்டிருக்கும். இது , அடிப்படையில், நிரலின் கட்டளை வரி செயலுருபுகளைக் கொண்ட ஓர் அணியாகும்.

sys.argv ஐ பயன்படுத்த, முதலில் நீங்கள் sys கூறுநிலையைத் தருவித்துக் கொள்ள வேண்டும். முதல் செயலுருபு sys.argv[0] எப்பொழுதும் செயல்படுத்த வேண்டிய நிரலின் பெயராக இருக்கும். மேலும், sys.argv[1] என்பது நிரலுக்கு (இங்கு இது C++கோப்பு) அனுப்பப்படும் முதல் செயலுருபு ஆகும்.

main(sys.argv[1])	நிரல் கோப்பையும் (பைத்தான் நிரல்), உள்ளீட்டு கோப்பினை (C++ கோப்பு) பட்டியலாகவும் (அணி) பெற்றுக் கொள்கிறது. argv[0] அனுப்பப்பட தேவையில்லாத பைத்தான் நிரலைக் கொண்டிருக்கும். ஏனெனில் default-main மூலக் குறிமுறையைக் குறிக்கும். செயல்படுத்த வேண்டிய கோப்பின் பெயரைக் குறிக்கும்.

Chapter 14.indd 277 10-02-2020 20:04:01

(ii) பைத்தானில் OS கூறுநிலை

பைத்தானில் இருக்கும் OS கூறுநிலை இயக்க முறைமையை சார்பு செயல்பாட்டுடன் பயன்படுத்துவதற்கான ஒரு வழிமுறையை வழங்குகிறது.

பைத்தான் இயங்கிக் கொண்டிருக்கும்போது, விண்டோஸ் இயக்க முறைமையுடன் OS கூறுநிலை ஊடாட அனுமதிக்கும் செயற்கூறுகளாவன.

os.system() : செயல்தளத்தில்(Shell)C++ தொகுத்தலுக்கான கட்டளையை இயக்கும். (இங்கு இது ஒரு கட்டளை சாளரம்) உதாரணத்திற்கு, C++ நிரலைத் தொகுக்க g++ தொகுப்பி செயல்படுத்தப்பட வேண்டும்.

இதை செயல்படுத்த, பின்வரும் கட்டளைப் பயன்படுத்தப்படுகின்றது.

```
os.system ('g++' + <variable_name1> '-<mode>' + <variable_name2>
```

இங்கு,

os.system :-	OS கூறுநிலையில் வரையறுக்கப்பட்டுள்ள system() செயற்கூறு
g++ :-	Windows இயக்கமுறைமையில் C++ நிரலை தொகுப்பதற்கான பொதுவான தொகுப்பி
variable_name1:-	(.cpp) என்ற நீட்டிப்பு இல்லாமல் C++ கோப்பின் பெயர் சர வடிவமைப்பில் குறிப்பிடப்பட்டிருக்கும்.
mode :-	உள்ளீடு/வெளியீட்டு பாங்கினை குறிக்கிறது. இங்கு இது hypen முன்னொட்டுடன் 0 என்று உள்ளது.
variable_name2 :-	(.exe) என்ற நீட்டிப்பு இல்லாமல் இயக்குக் கோப்புப் பெயர் சர வடிவமைப்பில் குறிப்பிடப்பட்டிருக்கும்.

உதாரணமாக, C++ நிரலை தொகுத்து, இயக்குவதற்கான கட்டளை கீழே கொடுக்கப்பட்டுள்ளது.

os.system('g++ ' + cpp_file + ' -o ' + exe_file)	g++ தொகுப்பான் cpp_file என்ற கோப்பினைத் தொகுக்கும்.–o(output) send to exe_file

குறிப்பு

OS.system()ல் '+' என்பது அனைத்து சரங்களும் ஒரே சரமாக இணைக்கப்பட்டு பட்டியலாக அனுப்பப்படுகிறது என்பதைக் குறிக்கிறது.

(iii) பைத்தான் getopt கூறுநிலை

பைத்தானில் getopt கூறுநிலை கட்டளை வரி தேர்வுகளையும், செயலுருபுகளையும் பிரித்தெடுக்க உங்களுக்கு உதவும். இந்த கூறுநிலை கட்டளைவரி செயலுருபு பிரித்தெடுத்தலை செயல்படுத்த செயற்கூறுகளை வழங்குகிறது.

getopt.getopt வழிமுறை

இந்த வழிமுறை கட்டளை வரி தேர்வுகளையும், செயலுருபுகள் பட்டியலையும் பிரித்தெடுக்கும். இந்த வழிமுறைக்கான தொடரியல் பின்வருமாறு

Chapter 14.indd 278 10-02-2020 20:04:01

<opts>,<args>=getopt.getopt(argv, options, [long_options])

செயலுருபுகளின் விவரங்களை இங்கே காணலாம்.

argv – இது பிரிக்கப்பட வேண்டிய செயலுருபின் மதிப்புகளின் பட்டியலைக் குறிக்கும். நமது நிரலில், கட்டளை முழுமையும் பட்டியலாக அனுப்பப்படுகிறது.

options – இது பைத்தான் நிரல் உள்ளீடு அல்லது வெளியீட்டிற்கான தேர்வு எழுத்துக்களின் சரமாகும். இங்கு 'i' அல்லது 'o' போன்ற தேர்வுகளும், அதைத் தொடர்ந்து முக்காற்புள்ளியும் (:) அமைந்திருக்கும். இங்கு (:) பாங்கினை குறிப்பிடப் பயன்படுகிறது.

long_options – இந்த அளபுரு சரங்களின் பட்டியலுடன் செலுத்தப்படுகிறது. Long options-ன் செயலுருபைத் தொடர்ந்து ('=') என்ற சமக்குறி இடம்பெற வேண்டும். நம்முடைய நிரலில் C++ கோப்பின் பெயர் சரமாக செலுத்தப்படும். மேலும், அது ஒரு உள்ளீடு கோப்பு என்பதைக் குறிக்க, அதனுடன் 'i' என்ற தேர்வும் செலுத்தப்படும்.

getopt() method returns value consisting of two elements. வழிமுறை இரண்டு உறுப்புகளை கொண்டுள்ள மதிப்புகளை திருப்பியனுப்பும். இவை ஒவ்வொன்றும் தனித்தனியாக opts மற்றும் args என்ற இரண்டு வெவ்வேறு பட்டியலில் (அணிகள்) சேமிக்கப்படும். Opts பாங்கு,பாதைப் போன்ற பிரிக்கப்பட்ட சரங்களின் பட்டியலைக் கொண்டிருக்கும். Args, தவறான பாதை அல்லது பாங்கின் காரணமாக பிரிக்கப்பட முடியாத எந்தவொரு சரத்தின் பட்டியலைக் கொண்டிருக்கும். geopt() வழிமுறையின் மூலம் சரங்களை பிரித்தெடுக்கும் போது பிழையேதும் இல்லாவிட்டால் args வெற்று அணியாக அமையும்.

உதாரணத்திற்கு, p4 என்ற C++ கோப்பினை, கட்டளை வரியில் இயக்கப்போகும் பைத்தான் நிரல், பின்வரும் ஒன்றைப் போல getopt() வழிமுறையைக் கொண்டிருக்கும்.

opts, args = getopt.getopt (argv, "i:",['ifile='])

where **opts** contains	[('-i', 'c:\\pyprg\\p4')] என்பதைக் கொண்டிருக்கும்.
-i :-	தேர்வு பாங்கினைத் தொடர்ந்து இடம் பெற்றிருக்க வேண்டும்.
'c:\\pyprg\\p4'	மதிப்பு C++ கோப்பின் முழுமையான பாதை

நம்முடைய எடுத்துக்காட்டுகளில், கட்டளை வரி கட்டளைகளை அனைத்தும் பிரிக்கப்பட்டிருப்பதாலும், செயலுருபு மீதமில்லாமலிருப்பதாலும், இரண்டாவது செயலுருபு args வெறுமையாக இருக்கும் [ ]. Print() கட்டளையைப் பயன்படுத்தி args வெளிக்காட்டப்பட்ட செய்திருந்தால், வெளியீடு [ ] என இருக்கும்.

```
>>>print(args)
[]
```

குறிப்பு

பைத்தானில் அடிப்படை கூறுநிலைகளின் பட்டியலையும், அவை எதற்காக பயன்படுத்தப்படுகிறது என்பதையும் நீங்கள் சரிபார்த்துக் கொள்ள முடியும். இந்த கோப்புகள் பைத்தானை நீங்கள் நிறுவி இருக்கும் அமைவிடத்தில், Lib கோப்புறையில் இருக்கும் C++ நிரலை wrap செய்வதற்கான மேலும் சில கட்டளைகள்.

Chapter 14.indd 279 10-02-2020 20:04:01

Some more command for wrapping C++ code

```
if __name__=='__main__':
        main(sys.argv[1:])
```

__name__ (A Special variable) in Python

பைத்தானில் main() செயற்கூறு இல்லையென்பதால், வரிமொழி மாற்றிக்கு (interpreter) பைத்தான் நிரலை இயக்குவதற்கான கட்டளைக் கொடுக்கும்போது, 0 (பூஜ்யம்) நிலையில் 0 level உள்தள்ளப்பட்ட குறிமுறை இயக்கப்பட வேண்டும். இருப்பினும், வரிமொழி மாற்றி இதனை செய்வதற்கு முன், சில சிறப்பு மாறிகளை வரையறுத்துக்கொள்ளும். __name__ என்பது அத்தகைய சிறப்பு மாறி, தானமைவாக, கோப்பின் பெயரை இருத்திக் கொள்ளும். மூலக்கோப்பு முதன்மை நிரலாக இயக்கப்படுமேயானால், வரிமொழி மாற்றி, __name__மாறிக்கு "__main__" என்ற மதிப்பினை அமைத்துக் கொடுக்கும்.

__name__ என்பது தற்போதைய கூறுநிலையை மதிப்பீடு செய்யும் ஒரு உள்ளிணைந்த மாறியாகும். இதனால் தற்போதைய script தானாகவே இயங்கி கொண்டிருக்கிறதா என்பதை சோதிக்கப்பயன்படுகிறது.

உதாரணத்திற்கு பின்வருவனவற்றை நோக்குங்கள்

```
if __name__ == '__main__':
        main (sys.argv[1:])
```

கட்டளை வரியில் பைத்தான் நிரல், தானே முதலில் செயல்பட போகிறதென்றால், __main__ அந்த பைத்தான் நிரலின் பெயரைக் கொண்டிருக்கும். மேலும், பைத்தானில் சிறப்பு மாறியான, __name__ என்பதும் பைத்தான் நிரலின் பெயரைக் கொண்டிருக்கும். மேலும், பைத்தானின் சிறப்பு மாறியான _name_ என்பதும் பைத்தான் நிரலின் பெயரைக் கொண்டிருக்கும். நிபந்தனை சரியெனில் அது main அழைக்கப்பட்டு, C++ கோப்பினை செயலுருபாக அனுப்பி வைக்கப்படும்.

குறிப்பு

sys.argv[1:] - பெருக்குப்பின் கோப்பு பெயர் அனைத்தையும் பெறவும்.
sys.argv[0:]-Script name (பைத்தான் நிரல்)
நீங்கள் பாடம் 8-ல் String slicing என்று படித்ததை நினைவில் கொள்ளுங்கள்

14.7 பாய்வு கட்டுப்பாட்டுகூற்றுகளை கொண்ட C++ நிரல்களை இயக்கும் பைத்தான் நிரல்

நாம் இப்பொழுது , ஒரு C++குறிமுறையை படித்து, இயக்கி, அதன் தீர்வைக் கண்டறிய ஒரு சோதனை நிரலை எழுதலாம். கொடுக்கப்பட்டுள்ள எண் பாலிண்ட்ரோமா இல்லையா என்பதைச் சோதிக்கும் C++ நிரலை இயக்குவதற்கான படிநிலைகள் கீழே தரப்பட்டுள்ளது.

Chapter 14.indd 280 10-02-2020 20:04:01

படிநிலை 1 — உள்ளீடு செய்யப்பட்ட எண் பாலிண்ட்ரோமா இல்லையா என சோதிக்கும் C++ நிரலை நோடபேடில் தட்டச்சு செய்து **pali.cpp.cpp** என்று சேமிக்கவும்.

படிநிலை 2 — பைத்தான் நிரலை தட்டச்சு செய்து pali.py என்று சேமிக்கவும்.

படிநிலை 3 — Run முனையத்தை Run terminal கிளிக் செய்து கட்டளை சாளரத்தை திறக்கவும்.

படிநிலை 4 — python.pali.py-ipali-cpp என்ற கட்டளையை தட்டச்சு செய்யவும்.

எடுத்துக்காட்டு:- 14.7.1 – ஏதேனும் ஓர் எண்ணை உள்ளீடு செய்து, while மடக்கினை பயன்படுத்தி அந்த எண் பாலிண்ட்ரோமா இல்லையா என்பதை கண்டறிய ஒரு C++ நிரலை எழுதுக.

```
/*. To check whether the number is palindrome or not using while loop.*/
//Now select File->New in Notepad and type the C++ program
#include <iostream>
using namespace std;
int main()
{
int n, num, digit, rev = 0;
cout<< "Enter a positive number: ";
cin>>num;
n = num;
    while(num)
    {  digit = num % 10;
       rev = (rev * 10) + digit;
       num = num / 10;    }
cout<< " The reverse of the number is: " << rev <<endl;
    if (n == rev)
       cout<< " The number is a palindrome";
    else
       cout<< " The number is not a palindrome";
    return 0;
}
// Save this file as pali_cpp.cpp
```

இப்பொழுது நோட்பேடில் *File->New* என்பதை தேர்ந்தெடுத்து நிரலை தட்டச்சு செய்யவும்.

கோப்பினை *pali.py* . என்று சேமிக்கவும். இது ஒரு *.cpp* கோப்பைத் தொகுத்து, இயக்குவதற்கான நிரல்

Chapter 14.indd 281 10-02-2020 20:04:01

```
# Python c:\pyprg\pali.py -i c:\pyprg\pali_cpp
import sys, os, getopt
def main(argv):
    opts, args = getopt.getopt(argv, "i:")
    for o, a in opts:
        if o in "-i":
            run(a)
def run(a):
    inp_file=a+'.cpp'
    exe_file=a+'.exe'
    os.system('g++  ' + inp_file + '  -o  ' + exe_file)
    os.system(exe_file)
if __name__=='__main__':
        main(sys.argv[1:])
```

Output of the above program

Output 1

```
C:\Users\Dell>python c:\pyprg\pali.py -i c:\pyprg\pali_cpp
Enter a positive number: 56765
The reverse of the number is: 56765
The number is a palindrome
```

Output 2

```
C:\Users\Dell>python c:\pyprg\ch14\pali.py -i c:\pyprg\pali_cpp
Enter a positive number: 56756
The reverse of the number is: 65765
The number is not a palindrome
```

பைத்தான் நிரல்	எப்படி வேலை செய்கிறது (அல்லது) வேலை செய்யும் விதம்
import sys, os, getopt	தேவையான செயற்கூறுகளை பயன்படுத்த sys, os மற்றும் getopt ஆகிய துணை நிரல்களை இணைத்துக்கொள்ள வேண்டும்.
def main(argv):	main( ) செயற்கூறு வரையறை. argv உள்ளீட்டு கோப்புநிலை மற்றும் C++ நிரல் கோப்பு ஆகியவற்றை ஒரு லிஸ்ட்-ஆக பெறும். எடுத்துக்காட்டு: ['i', 'C:\pyprg\pali_cpp']
opts, args = getopt. getopt(argv, "i:")	getopt( ), opts-ல் உள்ள கட்டளையை விருப்பத்தேர்வு மற்றும் செயலுருபு என இரண்டாக பிரிக்கின்றது. எடுத்துக்காட்டு: ['i', 'C:\pyprg\pali_cpp'] பிழை ஏதுமில்லை எனவே, 'args' [] என காட்டும்.
for o, a in opts:	'o' கோப்பு நிலையையும், 'a' என்பது C++ நிரலின் பாதையையும் கொண்டிருக்கும். அதாவது, print("o= ",o) என்பது o= -i எனவும், print("a= ", a) என்பது a= C:\pyprg\pali_cpp எனவும் காட்டும்.
if o in ("-i"):	o=='i' மெய்யா என சோதிக்கும்.
run(a)	C++ நிரலை அனுப்பி, run( ) அழைக்கப்படும்
def run():	run() செயற்கூறு வரையறை இங்கு தொடங்கிறது.
inp_file=a+'.cpp'	inp_file என்ற மாறி C++ நிரலை, .cpp என்ற நீட்சியுடன் இணைத்து பெற்றிருக்கும். எடுத்துக்காட்டு: print(inp_file) என்பது C:\pyprg\pali_cpp.cpp எனக்காட்டும்.
exe_file=a+'.exe'	exe_file என்ற மாறி, C++ நிரலை, .exe என்ற நீட்சியுடன் இணைத்து பெற்றிருக்கும். எடுத்துக்காட்டு: print(exe_file) என்பது C:\pyprg\pali_cpp.exe எனக்காட்டும்.
os.system('g++' + inp_file + '-o' + exe_file)	g++ நிரல் பெயர்ப்பி, inp_fileல் கொடுக்கப்பட்ட C++ நிரலை தொகுத்து, இயங்கு நிரலாக exe_fileல் சேமிக்கும்.
os.system(exe_file)	exe நிரல்கோப்பை இயக்கும்.
if __name__ == '__main__':	__name__ பைத்தான் நிரலின் பெயரை சேமிக்கும். __main__ இந்த மாறியும், பைத்தான் நிரலின் பெயரை சேமிக்கும். இந்த இரண்டு மாறிகளின் மதிப்பும் நிகரானதாக இருப்பின்.
main(sys.argv[1:])	பைத்தான் நிரல் பெயரை தவிர்த்து, கட்டளை இடைமுகத்தின் வழியே செயலுருபை அனுப்பி, main( ) அழைக்கப்படும். argv[1:] என்பது -i C:\pyprg\pali_cpp எனப் பெற்றிருக்கும்.

பாலிண்ட்ரோம் நிரலில் பயன்படுத்தியதைப் போலவே இடமாற்று அணிக்கோவையை இயக்குவதற்கான பைத்தான் நிரல் இருப்பதை நீங்கள் கவனித்திருப்பீர்கள் இதிலிருந்து நீங்கள் புரிந்துக் கொண்டது என்ன?

இந்த பைத்தான் நிரல் முக்கியமாக 'i'/'o' போன்ற பாங்குகளுடன் கூடிய C++கோப்பை படிக்கப் பயன்படுத்தப்பட்டுள்ளது. இது கட்டளை வரியிலிருக்கும் ஒவ்வொரு மதிப்பையும் பிரித்தெடுத்து, "opts" எனப்படும் பட்டியலுக்கு அளபுருவாக அனுப்பி வைக்கும். "long option ifile" அதன் பாங்குடன், C++குறிமுறை முழுவதையும் படிப்பதற்கான குறிப்பாக அமைகிறது.

for மடக்கிலுள்ள பட்டியல் 'o' பாங்கு (i') ஐயும் 'a' எனும் மாறி, முழுமையான பாதையுடனான C++ நிரல் கோப்பையும் கொண்டிருக்கும். உதாரணத்திற்கு [<'c:\pyprg\cpp_file'>,' trans_cpp']) cpp/exe போன்ற நீட்டிப்புகள் பைத்தான் scriptனால் இணைக்கப்படுகின்றன.

__name__ variable பைத்தான் Script-ன் 'main' வரையறையிலிருந்து நிரலைத் தொடங்குமாறு அறிவுறுத்துகிறது. 'main' வரையறை நீட்டிப்புகளை பிரித்தெடுத்தல் மற்றும் இணைத்தல் செயல்பாடுகளை செய்கிறது. "run" வரையறை "g++" தொகுப்பானை செயல்படுத்தி, exe கோப்பை உருவாக்குறிது. "os" கூறுநிலையின் system() கட்டளை exe கோப்பை செயல்படுத்தி, விரும்பிய வெளியீட்டை பெற்றுத் தருகிறது. பைத்தான் Scriptனைக் கொண்டு கோப்பு நீட்புக்களை இணைத்துக் கொள்ள முடியுமாதலால், C நிரல்களையும் நாம் செயல்படுத்திக் கொள்ளலாம்.

14.8 பைத்தான் எவ்வாறு C++ நிரல்களின் பிழைகளைக் கையாள்கிறது

பைத்தான் வெற்றிகரமான C++ நிரலை இயக்குவது மட்டுமல்லாது, தொகுத்தலின் போது C++ கூற்றில் ஏதேனும் பிழைகள் இருப்பின் அவற்றை வெளியிடவும் உதவுகிறது. உதாரணத்திற்கு, கீழ்க்காணும் C++ நிரலின் ஒரு பிழை உள்ளது. பைத்தான் மூலம் தொகுக்கும் போது என்ன நேரும் என்பதைப் பார்க்கலாம்.

எடுத்துக்காட்டு 14.8.1

```
// C++ program to print the message Hello
//Now select File→New in Notepad and type the C++ program
#include<iostream>
using namespace std;
int main()
{
    std::cout<<"hello"
    return 0;
}
// Save this file as hello.cpp
```


Chapter 14.indd 284 10-02-2020 20:04:02

```
#Now select File→New in Notepad and type the Python program as main.py
# Program that compiles and executes a .cpp file
 # Python main.py -i  hello

import sys, os, getopt
def main(argv):
    opts, args = getopt.getopt(argv, "i:")
    for o, a in opts:
        if o in "-i":
            run(a)
def run(a):
    inp_file=a+'.cpp'
    exe_file=a+'.exe'
    os.system('g++  ' + inp_file + '  -o  ' + exe_file)
    os.system(exe_file)
if __name__=='__main__':
        main(sys.argv[1:])
```

மேற்கண்ட நிரலின் வெளியீடு

```
C:\Users\Dell>python c:\pyprg\main.py -i c:\pyprg\ch14\hello
c:\pyprg\hello.cpp: In function 'int main()':
c:\pyprg\hello.cpp:6:21: error: expected ';' before 'return'
  std::cout<<"hello"
                    ^
                    ;
  return 0;
  ~~~~~~
'c:\pyprg\hello.exe' is not recognized as an internal or external command,
operable program or batch file.
```

குறிப்பு

மேற்கண்ட நிரலில், பைத்தான் உள்ள பிழைகளை வெளிக்காட்ட உதவுகிறது. பிழைகள் அவற்றின் வரிசை எண்களுடன் வெளியிடப்படுகின்றன. அவற்றின் வரிசை எண், பைத்தான் script-லிருந்து தொடங்குகிறது.

Chapter 14.indd 285 10-02-2020 20:04:02

14.9 அணியைக் கொண்டுள்ள C++ நிரலை இயக்கும் பைத்தான் நிரல்

கொடுக்கப்பட்ட எண் பாலிண்ட்ரோம் எண்ணா அல்லது இல்லையா என்று கண்டறிந்த நம்முடைய முந்தைய நிரலில், நீங்கள் கட்டுப்பாட்டு கட்டமைப்புக் கூற்றுகளை நிங்கள் பயன்படுத்தியுள்ளீர்கள். இப்பொழுது நீங்கள் அணியைக் கொண்டுள்ள C++ நிரலை இயக்கப் போகிறீர்கள்.

எடுத்துக்காட்டு :- 14.9.1

ஓர் அணியின் இடமாற்று அணிக்கோவையை (இருபரிமாக அணி) பெற ஒரு C++ நிரல் எழுதுக. இப்பொழுது, நோட்பேடில் File → New என்று தேர்ந்தெடுக்கப்படும் C++ நிரலை உள்ளீடு செய்யவும்.

```
#include <iostream>
using namespace std;
int main()
{
int a[3][3], i, j;
	for(i=0; i<3; i++)
	{
	for(j=0; j<3; j++)
	{   cout<<"enter the value for array["<<i+1<<"]"<<"["<<j+1<<"] :";
	    cin>>a[i][j];
	}
}
	system("cls");
cout<<"\n\nOriginal Array\n";
for(i=0; i<3; i++)    {
for(j=0; j<3; j++)
cout<<a[i][j]<<' ';
cout<<endl;   }
cout<<"\n\n The Transpose of Matrix\n";
 for (int i = 0; i < 3; i++)
 {
	for (int j = 0; j < 3; j++)
	    cout<<a[j][i]<<' ';
cout<<endl ;
}
	return 0;
}
```

இநத் கோப்பை **trans_cpp.cpp** என்று சேமிக்கவும். இப்பொழுது நோட்பேடில் **File → New** என்று தேர்ந்தெடுக்கும் பைத்தான் நிரலை உள்ளீடு செய்யவும்.

File ஐ **transpose.py** என்று சேமிக்கவும். இது **.cpp file**ஐ தொகுத்து, இயக்குவதற்கான நிரலாகும்.

 # Python tanspose.py -i trans_cpp

Chapter 14.indd 286
10-02-2020 20:04:02

```
import sys, os, getopt
def main(argv):
    opts, args = getopt.getopt(argv, "i:")
    for o, a in opts:
        if o in "-i":
            run(a)
def run(a):
    inp_file=a+'.cpp'
    exe_file=a+'.exe'
if __name__=='__main__':
        main(sys.argv[1:])
```

மேற்கண்ட நிரலின் வெளியீடு

```
Original Array
    1 2 3
    4 5 6
    7 8 9
The Transpose of Matrix
    1 4 7
    2 5 8
    3 6 9
```

14.10 செயற்கூறுகளைக் கொண்ட C++ நிரல்களை இயக்கும் பைத்தான் நிரல்

இப்பொழுது நீங்கள் செயற்கூறுகள் மற்றும் செயற்கூறு அழைப்புகள் கொண்ட C++ நிரலை இயக்க பைத்தான் Scriptஐ சோதிக்க போகிறீர்கள்.

எடுத்துக்காட்டு 14.10.1 – பயனர் வரையறுத்த செயற்கூறினைக் கொண்டு ஒரு எண்ணின் கனத்தினை கண்டறியும் C++ நிரலை எழுதுக.

```
#include <iostream>
using namespace std;
// Function declaration
int cube(int num);
int main()
{
int num;
int c;
cout<<"Enter any number: "<<endl;
```


Chapter 14.indd 287 10-02-2020 20:04:02

```
cin>>num;
c = cube(num);
cout<<"Cube of " <<num<< " is "<<c;
return 0;
}
//Function to find cube of any number
int cube(int num)
{
    return (num * num * num);
}
// Save this file as cube_file.cpp
#Now select File→New in Notepad and type the Python program
# Save the File as fun.py
# Program that compiles and executes a .cpp file
 # Python fun.py -i c:\pyprg\cube_file.cpp
import sys, os, getopt
def main(argv):
  opts, args = getopt.getopt(argv, "i:")
  for o, a in opts:
    if o in "-i":
      run(a)
def run(a):
  inp_file=a+'.cpp'
  exe_file=a+'.exe'
  os.system('g++  ' + inp_file + '  -o  ' + exe_file)
  os.system(exe_file)
if __name__=='__main__':
    main(sys.argv[1:])
```

மேற்கண்ட நிரலின் வெளியீடு

```
Enter any number:
5
Cube of 5 is 125
```


Chapter 14.indd 288 10-02-2020 20:04:02

14.11 ஓர் இனக்குழுவின் மரபுரிமத்தை விளக்கும் பைத்தான் நிரல்

நீங்கள் இப்பொழுது C++-ன் இனக்குழு நிரலை செயல்படுத்த போகிறீர்கள். இந்த இனக்குழு நிரல் பலநிலை மரபுரிமத்தைக் கொண்டுள்ளது. மேலும், இது பைத்தான் Scriptஐ பயன்படுத்தி வெளியீட்டை தருகிறது.

எடுத்துக்காட்டு 14.11.1 - பலநிலை மரபுரிமத்தை செயல்படுத்தும் C++நிரல்

```
// C++ program to implement Multilevel Inheritance
//Now select File→New in Notepad and type the C++ program
#include <iostream>
using namespace std;

// base class
class Vehicle
{
        public:
        Vehicle()
        {
            cout<< "This is a Vehicle" <<endl;
        }
};
```

```
class threeWheeler: public Vehicle
{  public:
threeWheeler()
        {
            cout<<"Objects with 3 wheels are vehicles"<<endl;
        }
};
// sub class derived from two base classes
class Auto: public threeWheeler{
        public:
            Auto()
        {
            cout<<"Auto has 3 Wheels"<<endl;
        }
};
// main function
int main()
{
        //creating object of sub class will invoke the constructor of base classes
        Auto obj;
        return 0;
}
```

```
// Save this file as inheri_cpp.cpp
//Now select File → New in Notepad and type the Python program
# Save the File as classpy.py
# Python classpy.py -i  inheri_cpp  command to execute c++ program
import sys, os, getopt
def main(argv):
    opts, args = getopt.getopt(argv, "i:")
    for o, a in opts:
        if o in "-i":
            run(a)
def run(a):
    inp_file=a+'.cpp'
    exe_file=a+'.exe'
    os.system('g++  ' + inp_file + '  -o  ' + exe_file)
    os.system(exe_file)
if __name__=='__main__':
        main(sys.argv[1:])
```

மேற்கண்ட நிரலின் வெளியீடு

```
This is a Vehicle
Objects with 3 wheels are vehicles
Auto has 3 Wheels
```

இந்த எடுத்துக்காட்டிலிருந்து, C++ நிரல்களை ஒருங்கிணைக்கப் பயன்படுத்தப்படும் பைத்தான் Script பொதுவானது என்பதை புரிந்துக் கொள்ளலாம். பைத்தான் Script கோப்பு மற்றும் C++ (cpp) கோப்பின் பெயர்கள் மட்டும் மாற்றியமைக்கப்பட்டுள்ளது. ஒருவருடைய கணினியில் முன்பே C++ கோப்பு இருந்தால், C++குறிமுறையை உள்ளீடு செய்வது கட்டாயமில்லை என்பதை கவனத்தில் கொள்ளவும். இந்தப் பாடப்பகுதியில், பைத்தான் மூலம் சாதாரண செயற்கூறுவைப் போல் மரபுரிமத்தை போன்ற அனைத்து வகை c++ நிரல்களிலும் செயல்படுத்த முடியும் என்பதைக் காண்பிக்க C++ நிரல் குறிமுறை வெளிப்படையாக குறிப்பிடப்பட்டுள்ளது. இத்தகைய பைத்தான் Script ஐ பயன்படுத்தி, நீங்கள் C நிரல்களையும் தொகுத்து, செயல்படுத்திக் கொள்ளலாம். பைத்தான், அதன் IDE இல்லாமலேயே தன்னிச்சையாக C++ நிரல் கோப்பை செயல்படுத்துவதால், அதனை Scripting மொழி எனவும் அழைக்கலாம்.

Chapter 14.indd 290 10-02-2020 20:04:02

நினைவில் கொள்க

- C++ ஒரு தொகுப்பு மொழி மாற்றி மொழி, பைத்தான் என்பது வரி மொழி மாற்றி மொழியாகும்.
- C++ நிலையான தன்மைக் கொண்ட தொகுக்கப்பட்ட மொழி, ஆனால் பைத்தான் என்பது மாறும் தன்மை கொண்ட மொழி.
- C++ போன்ற நிலையான வகையை சார்ந்த மொழியில் நிரலர் கணிப்பொறிக்கு ஒவ்வொரு தரவு மதிப்பும் பயன்படுத்தக்கூடிய தரவின வகையை வெளிப்படையாக தெரிவித்தல் அவசியமாகிறது.
- பைத்தான் போன்று மாறும் நிலைக் கொண்ட மொழிக்கு தரவு மதிப்பிற்கான தரவு வகையை வெளிப்படையாக கொடுக்கப்பட்ட வேண்டிய தேவையில்லை .
- பைத்தான் தரவு மதிப்பின் அடிப்படையில் மாறியை கையாள்கிறது.
- ஒரு Scripting மொழி என்பது பிற நிரலாக்க மொழிகளுடன் ஒருங்கிணைப்பதற்கும், தொடர்பு கொள்வதற்கும் வடிவமைக்கப்பட்ட ஒரு நிரலாக்க மொழியாகும்.
- MinGW இயக்க நேர தலைப்புக் கோப்புகளின் தொகுப்பைக் குறிக்கிறது. விண்டோஸ் இயக்க முறைமையில் இயங்கும் வகையில் C, C++ மற்றும் FORTRAN நிரல் குறிமுறைகளை தொகுக்கவும், தொடர்புறுத்தவும் இது பயன்படுகிறது.
- (.) புள்ளி செயற்குறி தருவிக்கப்பட்ட கூறுநிலையின் செயற்கூறுகளை அணுகுவதற்கு பயன்படுகிறது.
- sys கூறுநிலை வரிமொழி மாற்றியால் பயன்படுத்தப்படும் மாறிகளுக்கும், வரிமொழி மாற்றியுடன் வலுவாக ஊடாடும் செயற்கூறுகளுக்கும் அணுகுதலை வழங்குகிறது.
- பைத்தானில் OS கூறுநிலை இயக்க அமைப்பின் சார்பு நிலை செயல்பாட்டுடன் பயன்படுத்த வழிவகை வழங்குகிறது.
- பைத்தானில் getopt கூறுநிலை கட்டளைவரி தேர்வுகளையும் செயலுருபுகளையும் பிரித்தெடுக்க உதவுகிறது.

செய்முறைப் பயிற்சி

1. கீழ்க்காணும் வரையறுப்புகளுடன் student என்னும் ஓர் இனக்குழுவை வரையறுக்க ஒரு C++ நிரலை எழுதவும்.

    Protected உறுப்பு

    Rno integer

    Public members

    void Readno(int); to accept roll number and assign to Rno

    void Writeno(); To display Rno.

    The class Test is derived Publically from the Student class contains the following details

    Protected உறுப்புகள்

    Mark1 float

    Mark2 float

    Public உறுப்புகள்

void Readmark(float, float); To accept mark1 and mark2

void Writemark(); To display the marks

கீழ்க்காணும் வரையறுப்புகளுடன் sports என்னும் இனக்குழுவை உருவாக்கவும்.

Protected உறுப்புகள்

score integer

Public உறுப்புகள்

void Readscore(int); To accept the score

void Writescore(); To display the score

test இனக்குழுவிலிருந்து public அணுகியல்புடன் தருவிக்கப்பட்ட Result என்னும் இனக்குழு கீழ்க்காணும் வரையறுப்புகளை கொண்டுள்ளது

Private உறுப்புகள்

Total float

Public உறுப்புகள்

void display() assign the sum of mark1, mark2, score in total.

invokeWriteno(), Writemark() and Writescore(). Display the total also.

Total ஐ திரையில் காட்டும்.

C++ நிரலினை hybrid என்ற பெயரில் சேமிக்கவும். Hybrid.cpp என்ற கோப்பை செயல்படுத்த ஒரு பைத்தான் நிரலை எழுதவும்.

2. Write a C++ program to print boundary elements of a matrix and name the file as Border.cpp. Write a python program to execute the Border.cpp

மதிப்பீடு

QVDXVY

பகுதி–அ

சரியான விடையை தேர்ந்தெடுத்து எழுதுக (1 மதிப்பெண்)

1. பின்வருவனவற்றுள் எது Scripting மொழி அல்ல?

   (அ) ஜாவாஸ்கிரிப்ட் (ஆ) PHP (இ) பெர்ல் (ஈ) HTML

2. பைத்தான் நிரலில் C++ நிரலை தருவித்தல் எவ்வாறு அழைக்கப்படுகிறது?

   (அ) wrapping செய்தல் (ஆ) பதிவிறக்கம் செய்தல்

   (இ) இணைத்தல் (ஈ) பிரித்தல்

3. API ன் விரிவாக்கம் is

   (அ) Application Programming Interpreter

   (ஆ) Application Programming Interface

   (இ) Application Performing Interface

Chapter 14.indd 292 10-02-2020 20:04:02

(ஈ) Application Programming Interlink

4. பைத்தான் மற்றும் C++ நிரல்களை இடைமுகப்படுத்துவதற்கான கட்டமைப்பு

   (அ) Ctypes (ஆ) SWIG (இ) Cython (ஈ) Boost

5. பின்வருவனவற்றுள் எது உங்கள் குறிமுறையை தனித்தனி பகுதிகளாக பிரித்தெடுப்பதற்கான மென்பொருள் வடிவமைப்பு தொழில்நுட்பம்?

   (அ) பொருள்நோக்கு நிரலாக்கம்

   (ஆ) கூறுநிலை நிரலாக்கம்

   (இ) குறைந்த நிலை மொழி நிரலாக்கம்

   (ஈ) செயல்முறை நோக்கு நிரலாக்கம்

6. நீங்கள் விண்டோஸ் இயக்க முறைமையுடன் தொடர்பு கொள்ள எந்த கூறுநிலை அனுமதிக்கிறது?

   (அ) OS கூறுநிலை (ஆ) sys கூறுநிலை (இ) csv கூறுநிலை (ஈ) getopt கூறுநிலை

7. சரங்களை எந்த மாதிரியாக பிரிக்கும்பொழுது பிழையின்றி அமைந்தால், getopt( ) வெற்று அணியை திருப்பி அனுப்பும்?

   (அ) argv மாறி (ஆ) opt மாறி (இ) args மாறி (ஈ) ifile மாறி

8. பின்வரும் நிரல் பகுதியில் உள்ள செயற்கூறின் பெயரை அடையாளம் காண்க.

```
if __name__ =='__main__':
        main(sys.argv[1:])
```

   (அ) main(sys.argv[1:]) (ஆ) __name__ (இ) __main__ (ஈ) argv

9. கீழ்க்கண்டவற்றுள் எது உரை, எண்கள், படங்கள் மற்றும் அறிவியல் சார்ந்த தரவுகளை செயலாக்கப் பயன்படும்?

   (அ) HTML (ஆ) C (இ) C++ (ஈ) PYTHON

10. __name__ இது எதனை கொண்டுள்ளது ?

   (அ) c++ filename (ஆ) main() name

   (இ) python filename (ஈ) os module name

பகுதி–ஆ

அனைத்து வினாக்களுக்கும் விடையளி (2 மதிப்பெண்)

1. Scripting மொழிக்கும், மற்ற நிரலாக்க மொழிக்கும் உள்ள தத்துவர்த்த வேறுபாடு யாது?
2. தொகுப்பான் மற்றும் வரிமொழி மாற்றியை வேறுபடுத்துக.
3. விரிவாக்கம் தருக (i) SWIG (ii) MinGW
4. கூறுநிலைகளின் பயன் யாது?
5. cd கட்டளையின் பயன் யாது? எடுத்துக்காட்டு தருக.

Chapter 14.indd 293 10-02-2020 20:04:02

பகுதி–இ

அனைத்து வினாக்களுக்கும் விடையளி (3 மதிப்பெண்)

1. பைத்தான் மற்றும் C++ வேறுபடுத்துக
2. Scripting மொழியின் பயன்பாடுகள் யாவை?
3. MinGW என்றால் என்ன? அதன் பயன் யாது?
4. கீழ்க்காணும் கூற்றில் கூறுநிலை, செயற்குறி, வரையிறையின் பெயர் ஆகியவற்றை அடையாளம் காண்க.

   welcome.display()
5. sys.argv என்றால் என்ன?

பகுதி ஈ

அனைத்து வினாக்களுக்கும் விடையளி (5 மதிப்பெண்)

1 பைத்தானில் ஏதேனும் 5 பண்புக்கூறுகளை கூறவும்.

2. பின்வரும் கட்டளை ஒவ்வொன்றையும் விளக்கவும்.

   Python <filename.py> -<i> <C++ filename without cpp extension>

3. பைத்தானில், sys,os,getopt கூறுநிலைகளின் தேவை என்ன என்பதை விளக்குக.

4. getopt() என்ற செயற்கூறின் தொடரியலை எழுதி, அதன் செயலுருபுகளையும், திருப்பியனுப்பும் மதிப்புகளையும் விளக்குக.

5. கீழ்க்காணும் c++ நிரலை செயல்படுத்த ஒரு பைத்தான் நிரலை எழுதவும்.

```
#include <iostream>
using namespace std;
int main()
{ cout<<"WELCOME";
return(0);
}
```

   The above C++ program is saved in a file welcome.cpp

REFERENCES

1. *Learn Python The Hard Way by Zed Shaw*
2. *Python Programming Advanced by Adam Stuart or Powerful Python by Aaron Maxwell*
3. *https://docs.python.org*

அலகு V

பாடம் 15

SQL மூலம் தரவுகளைக் கையாளுதல்

2369H7

கற்றலின் நோக்கங்கள்

இந்த பாடப்பகுதியைக் கற்றபின் மாணவர்கள், பைத்தான் ஸ்கிரிப்ட்டைப் பயன்படுத்தி,

- ஒரு தரவு அட்டவணையை உருவாக்கி அதில் புதிய வரிசைகளைத் தரவுத் தளத்தில் சேர்த்தல்.
- ஒரு தரவு அட்டவணையில் உள்ள பதிவுகளை மேம்படுத்துதல் மற்றும் நீக்குதல்.
- ஒரு தரவு அட்டவணையில் வினவல்.
- CSV கோப்பில் வினவலை எழுதுதல்.

15.1 அறிமுகம்

ஒரு தரவுத்தளம் என்பது ஒருங்கிணைக்கப்பட்ட தரவுகளின் தொகுப்பாகும். “தரவுத்தளம்” என்ற கூறு தரவுகளை அல்லது தரவுத்தள மேலாண்மை அமைப்பைக் குறிக்கும். பயனர்கள் மற்றும் தரவுத்தளத்திற்கு இடையே உள்ள தொடர்புக்கு பயன்படும் மென்பொருள் பயன்பாடே தரவுத்தள மேலாண்மை அமைப்பாகும். பயனர்கள் மனிதர்களாக இருக்க வேண்டிய அவசியமில்லை, பிற நிரல்களாகவோ அல்லது பயன்பாடுகளாகவோ இருக்கலாம். பைத்தான் நிரல் எவ்வாறு SQL தரவுத்தளத்துடன் பயனராக தொடர்பு கொள்கிறது என்பதை இனி கற்கலாம்.

15.2 SQLite

SQLite என்பது எளிய உறவுநிலை தரவுத்தள அமைப்பாகும். இது தரவுகளை முறையான தரவுக்கோப்புகளாகவும் கணினியின் உட்புற நினைவகத்தில் கூட சேமித்து வைக்கும். இது MySQL அல்லது Oracle போன்று தனித்த தரவுத்தள சேவையக நிரலாக இல்லாமல் உள்ளிணைந்த பயன்பாடாக வடிவமைக்கப்பட்டுள்ளது. வேகமாகவும், மிகுந்த சோதிக்கப்பட்டதாகவும் மற்றும் நெகிழ்வானதாகவும் உள்ளதால் SQLite-ல் வேலை செய்வது எளிதாகும். SQLite ற்காக பைத்தான் சிறப்பான நூலகத்தைக் கொண்டுள்ளது.

SQLite யைப் பயன்படுத்த,

படிநிலை 1	sqlite3யை இணைக்கவும்
படிநிலை 2	connect( ) வழிமுறையைப் பயன்படுத்தி இணைப்பை உருவாக்கி தரவுத்தளத்தின் பெயரை அனுப்பவும்.
படிநிலை 3	cursor= connection.cursor( ) என்றக் கூற்றைப் பயன்படுத்தி cursor என்னும் பொருளை அனுப்பவும்.

- படிநிலை 2 ல் தரவுத்தளத்தை இணைத்தல் என்பது, அணுக வேண்டிய தரவுத்தளத்தின் பெயரைஅனுப்புதல் என்பதாகும். அவ்வாறு அனுப்பும்போது, அத்தரவுத்தளம் ஏற்கனவே இருக்குமாயின் அது இணைக்கப்படும். இல்லாவிடில், பைத்தான், கொடுக்கப்பட்ட பெயரில் ஒரு புதிய தரவுத்தளத்தை உருவாக்கும்.
- படிநிலை3ல் cursor என்பது ஒரு கட்டுப்பாட்டு அமைப்பாகும், இது தரவுத்தளப் பதிவுகளை இணைக்கப் பயன்படுகிறது.
- பைத்தானில், cursor மிக முக்கிய பங்குவகின்றது. அனைத்து கட்டளைகளும் cursor பொருள் மூலமே மட்டுமே இயக்கப்படும்.

ஒரு அட்டவணையை உருவாக்க, தரவுத்தளத்தில் ஒரு பொருளை உருவாக்கி அதற்கான SQL கட்டளைகளை எழுத வேண்டும்.

எடுத்துக்காட்டு:- sql_comm = "SQL statement"

கட்டளைகளை நிறைவேற்ற cursor வழிமுறையைப் பயன்படுத்தி sql கட்டளையின் பெயரை அளபுருக்களாக அனுப்ப வேண்டும். sql_comm்ல் நிறைய கட்டளைகள் சேமிக்கப்பட்டு அவை நிறைவேற்றப்படும். அனைத்து செயல்களையும் முடித்த பின்னர் மாற்றங்களைக் கோப்பில் சேமித்து பின்னர் இணைப்பை மூட வேண்டும்.

15.3 SQLite டைப் பயன்படுத்தி தரவுத்தளத்தை உருவாக்குதல்

பின்வரும் எடுத்துக்காட்டு sqlite3 பயன்படுத்தி பைத்தானில் ஒரு தரவு தளத்தை எவ்வாறு இணைப்பது என்பதை விளக்குகிறது.

```
# Python code to demonstrate table creation and  insertions with SQL
# importing module
import sqlite3
# connecting to the database
connection = sqlite3.connect ("Academy.db")
# cursor
cursor = connection.cursor()
```

மேற்கண்ட எடுத்துக்காட்டில், "Academy" என்ற பெயரில் தரவுத்தளம் உருவாக்கப்பட்டுள்ளது. இது SQL-ன் "CREATE DATABASE Academy;" என்ற கட்டளைக்கு இணையானது. அனுப்புவது. "sqlite3. connect ('Academy.db')" என்ற கூற்றை அழைக்கும் போது ஏற்கனவே உருவாக்கிய தரவுத்தளத்தைத் திறக்கும்.

15.3.1 அட்டவணையை உருவாக்குதல்

ஒரு வெற்று தரவுத்தளத்தை உருவாக்கிய பிறகு, முடிந்த வரை ஒன்று அல்லது அதற்கு மேற்பட்ட அட்டவணைகளைச் சேர்க்கலாம். அதற்கான கட்டளை அமைப்பு

CREATE TABLE Student (

RollnoINTEGER, SnameVARCHAR(20), GradeCHAR(1), gender CHAR(1),

Chapter 15.indd 296 30-01-2020 20:58:04

Average float(5.2), birth_date DATE, PRIMARY KEY (Rollno));

மேற்காண் இந்த முறையில் SQL கட்டளைகளில் செய்யப்படுவதைப் போலவே நேரடியாக பைத்தானில் செய்யலாம். SQL அல்லது SQLiteக்கு கட்டளைகளை அனுப்ப cursor பொருள்தேவை. பொதுவாக SQL மற்றும் தரவுத்தளத்திலுள்ள cursor என்பது, தரவுத்தளப் பதிவுகளின் மீது செயல்படும் ஒரு கட்டுப்பாட்டு அமைப்பாகும்.

குறிப்பு

SQL-ன் அனைத்து கட்டளைகளையும் செயல்படுத்த cursor பயன்படுகிறது.

இணைப்புக்கான cursor( ) வழிமுறையை அழைத்து cursor பொருளை உருவாக்கலாம்.

Cursor விடைத்தொகுப்பில் இருந்து பதிவுகளை எடுக்கப்பயன்படுகிறது.

பைத்தானில் SQL கட்டளைகளை மூன்று மேற்கொள் அடைப்புக் குறிகளுக்குள் சரமாக வரையறுக்க வேண்டும்.

எடுத்தக்காட்டு 15.3.1

```
import sqlite3
sql_command = """
CREATE TABLE Student (
Rollno INTEGER PRIMARY KEY ,
Sname VARCHAR(20),
Grade CHAR(1),
gender CHAR(1),
Average DECIMAL(5,2),
birth_date DATE);"""
```

மேலேயுள்ள எடுத்துக்காட்டில்,

Emp_no என்ற புலம் "INTEGER PRIMARY KEY" (முழு எண் முதன்மைத் திறவுகோல்) என வரையறுக்கப்பட்டுள்ளது. SQLite3 ல் இது போல் பெயரிடப்பட்ட நெடுவரிசையானது தானாகவே அதிகரித்துக் கொள்ளும்.

வேறு விதமாக கூற வேண்டுமெனில், அட்டவணையில் உள்ள ஒரு நெடுவரிசை INTEGER PRIMARY KEY, என்று அறிவிக்கப்பட்டு, எப்பொழுதெல்லாம் NULL என்ற மதிப்பு உள்ளீடாகப் பயன்படுத்தப்படுகிறதோ, அந்த NULL மதிப்பு தானாகவே அந்த நெடுவரிசையில் இதுவரை பயன்படுத்தப்பட்ட மிக உயர்ந்த மதிப்பைவிட ஒன்று மிகுந்து முழு எண்ணாக இருக்கும். வெற்று அட்டவணை எனில் 1 என்ற மதிப்பு பயன்படுத்தப்படும்.

15.3.2 பதிவுகளைச் சேர்த்தல் (Adding Records)

இப்பொழுது அட்டவணையுடன் கூடிய ஒரு தரவுத்தளம் எந்த தரவுகளும் இல்லாமல் உள்ளது. "INSERT" கட்டளையை SQLite ல் அனுப்புவதன் மூலம் அட்டவணையில் தரவுகளை உள்ளிடலாம். execute() செயற்கூறு கொடுக்கப்பட்ட SQL கட்டளையை செயல்டுத்தும்

Chapter 15.indd 297 30-01-2020 20:58:04

எடுத்தக்காட்டு 15.3.2 -1

```
import sqlite3
connection = sqlite3.connect ("Academy.db")
cursor = connection.cursor()
# delete
cursor.execute ("""DROP TABLE Student;""")
sql_command = """
CREATE TABLE Student (
Rollno INTEGER PRIMARY KEY , Sname VARCHAR(20), Grade CHAR(1),
gender CHAR(1), Average DECIMAL (5, 2), birth_date DATE);"""
cursor.execute(sql_command)
    sql_command = """INSERT INTO Student (Rollno, Sname, Grade, gender, Average, birth_
    date) VALUES (NULL, "Akshay", "B", "M","87.8", "2001-12-12");""" cursor.execute(sql_
    command)
    sql_command = """INSERT INTO Student (Rollno, Sname, Grade, gender, Average,
    birth_date) VALUES (NULL, "Aravind", "A", "M","92.50","2000-08-17");""" cursor.
    execute(sql_command)
# never forget this, if you want the changes to be saved:
connection.commit()
connection.close()
print("STUDENT TABLE CREATED")
```

வெளியீடு

```
    STUDENT TABLE CREATED
```

பின்வரும் எடுத்துக்காட்டு 15.3.2 ஒரு முழுமையான உதாரணமாகும். இந்த நிரலை இயக்க "Academy.db" கோப்பை வேண்டும் அல்லது SQL கட்டளையில் உள்ள DROP TABLE என்ற வரியின் குறிப்புரையாக்குக.

பெரும்பாலான நிகழ்வுகளில் கருத்தியலாக ஒரு SQL அட்டவணையில் தரவுகளை இடையில் செருக முடியாது. Insert கூற்றைப்பயன்படுத்தி, சில பைத்தான் தரவு இனங்களில் உள்ள பலத்தரவுகளை உள்ளிடலாம்.

பின்வரும் எடுத்துக்காட்டில், Academy.db என்ற தரவுத்தளம் மற்றும் Student அட்டவணை ஏற்கனவே உள்ளது எனக்கொள்க. நபர்களின் தரவுகளைக்கொண்ட List-யைINSERT கூற்றைப் பயன்படுத்தி உள்ளிடப்பட்டுள்ளது.

Chapter 15.indd 298 30-01-2020 20:58:05

எடுத்தக்காட்டு 15.3.2-2

```
import sqlite3
connection = sqlite3.connect("Academy.db")
cursor = connection.cursor()
student_data = [("BASKAR", "C", "M","75.2","1998-05-17"),
        ("SAJINI", "A", "F","95.6","2002-11-01"),
        ("VARUN", "B", "M","80.6","2001-03-14"),
        ("PRIYA", "A", "F","98.6","2002-01-01"),
        ("TARUN", "D", "M","62.3","1999-02-01") ]
for p in student_data:
        format_str = """INSERT INTO Student (Rollno, Sname, Grade, gender,Average, birth_
        date)  VALUES (NULL,"{name}", "{gr}", "{gender}","{avg}","{birthdate}");"""
        sql_command=format_str.format(name=p[0],gr=p[1],gender=p[2],avg=p[3],
        birthdate = p[4])
cursor.execute(sql_command)
connection.commit()
connection.close()
print("RECORDS ADDED TO STUDENT TABLE ")
```

வெளியீடு

```
RECORDS ADDED TO STUDENT TABLE
```

15.4 பைத்தானை பயன்படுத்தி SQL வினவல்

இப்பொழுது student அட்டவணையில் வினவலைப் பயன்படுத்துதலைப்பற்றி ,0கற்கலாம். பதிவுகளைச் சேர்ப்பதைப் போல அவற்றை திரும்ப பெறுவதும் மிக எளிது. Execute செயற்கூறு SQL கட்டளைகளை பயன்படுத்தி அனைத்து தரவுகளையும் அட்டவணையில் இருந்து பெறுகிறது.

15.4.1 SELECT வினவல்

SELECT கூற்று **SQL**-ல் மிக அதிகமாக பயன்படும் கூற்று ஆகும். இந்த கூற்று தரவுத் தளத்திலிருந்து தரவை பெற பயன்படுத்தப்படுகிறது. இந்த கூற்றுக்கான கட்டளை அமைப்பு **"Select * from table_name"** அட்டவணையின் அனைத்து தரவுகளையும் பட்டியலாக ஒரு பொருளில் பெற முடியும்.

"sql_Academy_query.py" என்ற சேமிக்கப்பட்ட நிரலை இயக்கினால் மெய்யான தரவைப் பொறுத்து பின்வரும் வெளியீடு தோன்றும்.

பைதான் கோப்பு உருவாக்கப்படும் கோப்புறையில் தான் தரவுத்தள கோப்புகளும் உருவாக்கப்படும் என்பதை கவனத்தில் கொள்ள வேண்டும். கோப்பின் பாதையை மாற்ற விரும்பினால் கோப்பை திறக்கும் போது மாற்றம் செய்ய வேண்டும்.

எடுத்தக்காட்டு 15.4.1-1

```
#save the file as "sql_Academy_query.py"
# cursor object
crsr = connection.cursor()
# execute the command to fetch all the data from the table Student
crsr.execute("SELECT * FROM Student")
# store all the fetched data in the ans variable
ans= crsr.fetchall()
# loop to print all the data
for i in ans:
        print(i)
```

15.4.1.1 fetchall() பயன்படுத்தி அனைத்து பதிவுகளையும் காண்பித்தல்

fetchall() செயற்கூறு அனைத்து வரிசைகளையும் தரவுத்தள அட்டவணையில் இருந்து பெற பயன்படுகிறது.

எடுத்தக்காட்டு 15.4.1.1-1

```
import sqlite3
connection = sqlite3.connect("Academy.db")
cursor = connection.cursor()
cursor.execute("SELECT * FROM student")
print("fetchall:")
result = cursor.fetchall()
for r in result:
    print(r)
```

வெளியீடு

```
fetchall:
(1, 'Akshay', 'B', 'M', 87.8, '2001-12-12')
(2, 'Aravind', 'A', 'M', 92.5, '2000-08-17')
(3, 'BASKAR', 'C', 'M', 75.2, '1998-05-17')
(4, 'SAJINI', 'A', 'F', 95.6, '2002-11-01')
(5, 'VARUN', 'B', 'M', 80.6, '2001-03-14')
(6, 'PRIYA', 'A', 'F', 98.6, '2002-01-01')
(7, 'TARUN', 'D', 'M', 62.3, '1999-02-01')
```


Chapter 15.indd 300 30-01-2020 20:58:05

குறிப்பு

cursor.fetchall() -fetchall () செயற்கூறு அனைத்து வரிசைகளையும் பெற பயன்படுகிறது.

cursor.fetchone() - fetchone () செயற்கூறு வினவல் முடிவுத் தொகுதியின் உள்ளே உள்ள அடுத்த வரிசையைக் கொடுக்கும் (அல்லது) எந்த வரிசையும் இல்லை என்றால் None என்ற மதிப்பை விடையாகக் கொடுக்கும்.

cursor.fetchmany() செயற்கூறு முடிவுத் தொகுதியில் மீதம் உள்ள வரிசைகளின் எண்ணிக்கையைக் கொடுக்கும்.

15.4.1.2 fetchone()-னைப் பயன்படுத்தி ஒரு பதிவைக் காண்பித்தல்

fetchone() செயற்கூறு வினவல் முடிவுத் தொகுதியின் உள்ளே உள்ள அடுத்த வரிசையைக் கொடுக்கும் (அல்லது) எந்த வரிசையும் இல்லை என்றால் None என்ற மதிப்பை கொடுக்கும்.

எடுத்தக்காட்டு 15.4.1.2-1

```
import sqlite3
connection = sqlite3.connect("Academy.db")
cursor = connection.cursor()
cursor.execute("SELECT * FROM student")
print("\nfetch one:")
res = cursor.fetchone()
print(res)
```

வெளியீடு

```
fetch one:
(1, 'Akshay', 'B', 'M', 87.8, '2001-12-12')
```

15.4.1.3 fetchone()-னைப் பயன்படுத்தி அனைத்து பதிவுகளையும் காண்பித்தல்

while மடக்கு மற்றும் fetchone() செயற்கூறு பயன்படுத்தி அட்டவணையில் உள்ள அனைத்து பதிவுகளையும் காண முடியும்.

எடுத்தக்காட்டு 15.4.1.3 -1

```
import sqlite3
connection = sqlite3.connect("Academy.db")
cursor = connection.cursor()
cursor.execute("SELECT * FROM student")
print("fetching all records one by one:")
result = cursor.fetchone()
while result is not None:
    print(result)
    result = cursor.fetchone()
```

வெளியீடு

```
fetching all records one by one:
(1, 'Akshay', 'B', 'M', 87.8, '2001-12-12')
(2, 'Aravind', 'A', 'M', 92.5, '2000-08-17')
(3, 'BASKAR', 'C', 'M', 75.2, '1998-05-17')
(4, 'SAJINI', 'A', 'F', 95.6, '2002-11-01')
(5, 'VARUN', 'B', 'M', 80.6, '2001-03-14')
(6, 'PRIYA', 'A', 'F', 98.6, '2002-01-01')
(7, 'TARUN', 'D', 'M', 62.3, '1999-02-01')
```

15.4.1.4 fetchmany() பயன்படுத்தி பதிவுகளைக் காண்பித்தல்

குறிப்பிட்ட எண்ணிக்கையிலான பதிவுகளைக் காண்பிக்க fetchmany() செயற்கூறு பயன்படுகிறது. இந்த செயற்கூறு முடிவுத் தொகுதியில் மீதம் உள்ள வரிசைகளின் எண்ணிக்கையைக் கொடுக்கும்.

எடுத்தக்காட்டு 15.4.1.4-1: பயன்படுத்தி பதிவுகளின் பொருளடக்கத்தைக் காண்பிக்கும் நிரல்

```
import sqlite3
connection = sqlite3.connect("Academy.db")
cursor = connection.cursor()
cursor.execute("SELECT * FROM student")
print("fetching first 3 records:")
result = cursor.fetchmany(3)
print(result)
```

வெளியீடு

```
fetching first 3 records:
[(1, 'Akshay', 'B', 'M', 87.8, '2001-12-12'), (2, 'Aravind', 'A', 'M', 92.5, '2000-08-17'), (3, 'BASKAR', 'C', 'M', 75.2, '1998-05-17')]
```

எடுத்தக்காட்டு 15.4.1.4-2: மடக்கு பயன்படுத்தாமல், பதிவுகளின் பொருளடக்கத்தை புதியவரியில் காண்பிக்கும் நிரல்

```
import sqlite3
connection = sqlite3.connect("Academy.db")
cursor = connection.cursor()
cursor.execute("SELECT * FROM student")
print("fetching first 3 records:")
result = cursor.fetchmany(3)
print(*result,sep="\n")
```

வெளியீடு

```
fetching first 3 records:
(1, 'Akshay', 'B', 'M', 87.8, '2001-12-12')
(2, 'Aravind', 'A', 'M', 92.5, '2000-08-17')
(3, 'BASKAR', 'C', 'M', 75.2, '1998-05-17')
```

குறிப்பு

மடக்குகளைப் பயன்படுத்தாமல் இடைவெளிகளுடன் கூடிய உறுப்புகளின் பட்டியலை ஒரே வரிசையில் அச்சிடுவதற்கு குறியீடு =”\n” or sep=”, ” பயன்படுகிறது.

15.4.2 SQL-ல் துணைநிலை கூற்று (CLAUSES)

SQL வழங்கும் பலவகையான துணைநிலை கூற்றுகள் SELECT கூற்றுகளில் பயன்படுத்தப்படுகிறது. SQLite ல் ஏறக்குறைய அனைத்து வகை துணை நிலை கூற்றுகளும் செயல்படுகிறது. அதிகமாக பயன்படுத்தப்படும் துணை நிலை கூற்றுகள் பின்வருமாறு,

Chapter 15.indd 302 30-01-2020 20:58:05

- DISTINCT
- WHERE
- GROUP BY
- ORDER BY.
- HAVING

15.4.2.1 SQL DISTINCT துணைநிலை கூற்று

ஒரு குறிப்பிட்ட நெடுவரிசை அல்லது அட்டவணையில் உள்ள இரட்டிப்பு மதிப்புகளைத் தவிர்ப்பதற்காக DISTINCT துணை நிலை கூற்று பயன்படுகிறது. இந்த சிறப்பு சொல்லைப் பயன்படுத்தும் போது தனித்த மதிப்புகளை பெற முடியும்.

பின்வரும் எடுத்துக்காட்டில், “student table” என்ற அட்டவணையில் இருந்து மாணவர்கள் பெற்ற தரவரிசை காண்பிக்கப்பட்டுள்ளது.

எடுத்தக்காட்டு 15.4.2.1-1

```
import sqlite3
connection = sqlite3.connect("Academy.db")
cursor = connection.cursor()
cursor.execute("SELECT DISTINCT (Grade) FROM student")
result = cursor.fetchall()
print(result)
```

வெளியீடு

```
[('B',), ('A',), ('C',), ('D',)]
```

மேற்கண்ட எடுத்துக்காட்டில் “distinct” என்ற சிறப்பு சொல்லைப் பயன்படுத்தவில்லை எனில் 4 பதிவுகளுக்குப் பதிலாக 7 பதிவுகள் காண்பிக்கப்பட்டிருக்கும். ஏனெனில், மூல அட்டவணையில் இரட்டிப்பு மதிப்புகளுடன் 7 பதிவுகள் உள்ளன.

15.4.2.2 SQL WHERE துணைநிலை கூற்று

குறிப்பிட்ட நிபந்தனைகளை நிறைவேற்றும் பதிவுகளை மட்டுமே பிரித்தெடுக்க WHERE துணைநிலை கூற்று பயன்படுகிறது. பின்வரும் எடுத்துக்காட்டில் “student table” தரவுத்தளத்தில் இருந்து மாணவர்களின் தரவரிசையை மட்டுமே பிரித்தெடுப்பதைக் காணலாம்.

```
import sqlite3
connection = sqlite3.connect("Academy.db")
cursor = connection.cursor()
cursor.execute("SELECT DISTINCT (Grade) FROM student where gender='M'")
result = cursor.fetchall()
print(*result,sep="\n")
```

வெளியீடு

```
('B',)
('A',)
('C',)
('D',)
```

15.4.2.3 SQL Group By துணை நிலை கூற்று

GROUP BY துணை நிலை கூற்றானது குறிப்பிட்ட பதிவுகளைச் சுருக்கமான வரிசைகளைக் கொண்ட குழுவாக சேர்க்கிறது. இது ஒவ்வொரு குழுவிற்கும் ஒரு பதிவை கொடுக்கிறது. இது மதிப்பீட்டு செயற்கூறுகள் (COUNT, MAX, MIN, SUM, AVG) விடைத்தொகுதிகளை ஒன்று அல்லது அதற்கு மேற்பட்ட நெடுவரிசை குழுக்களாக மாற்றபயன்படுகிறது. பின்வரும் எடுத்துக்காட்டு student அட்டவணையில் இருந்து ஆண் மற்றும் பெண் எண்ணிக்கையைக் கணக்கிட்டு விடையைக் காண்பிக்கிறது.

எடுத்தக்காட்டு 15.4.2.3 -1

```
import sqlite3
connection = sqlite3.connect("Academy.db")
cursor = connection.cursor()
cursor.execute("SELECT gender,count(gender) FROM student Group BY gender")
result = cursor.fetchall()
print(*result,sep="\n")
```

வெளியீடு

```
    ('F', 2)
    ('M', 5)
```

15.4.2.4 SQL ORDER BY துணை நிலைக் கூறு

ORDER BY துணை நிலைக்கூற்று குறிப்பிட்ட புலங்களில் உள்ள தரவுகளை வரிசையாக்க SELECT கூற்றுடன் சேர்ந்து பயன்படுகிறது. விடைத் தொகுதியை ஏறு வரிசை அல்லது இறங்கு வரிசையில் வரிசையாக்கம் செய்ய உதவுகிறது. பின்வரும் எடுத்துக்காட்டு, மாணவர் பெயர் மற்றும் வரிசை எண் ஆகியவற்றை அகர வரிசைப்படி காண்பிக்கிறது.

Chapter 15.indd 304 30-01-2020 20:58:05

எடுத்தக்காட்டு 15.4.2.4 -1

```
import sqlite3
connection = sqlite3.connect("Academy.db")
cursor = connection.cursor()
cursor.execute("SELECT Rollno,sname  FROM student Order BY sname")
result = cursor.fetchall()
print(*result,sep="\n")
```

வெளியீடு

```
(1, 'Akshay')
(2, 'Aravind')
(3, 'BASKAR')
(6, 'PRIYA')
(4, 'SAJINI')
(7, 'TARUN')
(5, 'VARUN')
```

15.4.2.5 SQL HAVING துணை நிலைக்கூறு

குழு சார்புகளைப் பொறுத்து தரவுகளை வடிகட்ட HAVING துணை நிலைக்கூற்று பயன்படுகிறது. இது WHERE நிபந்தனை கூற்றை ஒத்ததாகும் ஆனால் குழு சார்புகளுடன் பயன்படுகிறது. குழு சார்புகளை WHERE துணை நிலைக் கூற்றில் பயன்படுத்த முடியாது. ஆனால், HAVING துணை நிலைக் கூற்றில் பயன்படுத்த முடியும்.

எடுத்தக்காட்டு 15.4.2.5 -1

```
import sqlite3
connection = sqlite3.connect("Academy.db")
cursor = connection.cursor()
cursor.execute("SELECT GENDER,COUNT(GENDER) FROM Student GROUP BY
GENDER HAVING COUNT(GENDER)>3")
result = cursor.fetchall()
co = [i[0] for i in cursor.description]
print(co)
print(result)
```

வெளியீடு

```
['gender', 'COUNT(GENDER)']
[('M', 5)]
```

15.5 SQL AND, OR மற்றும் NOT செயற்குறிகள்

WHERE துணைநிலைக் கூற்று AND, OR, மற்றும் NOT செயற்குறிகளுடன் இணைந்து செயல்படும்.

AND மற்றும் OR செயற்குறிகள் பதிவுகளை ஒன்றிற்கு மேற்பட்ட நிபந்தனைகளைப் பொறுத்து வடிகட்ட பயன்படுகிறது. பின்வரும் எடுத்துக்காட்டு, student அட்டவணையில் இருந்து 'A' அல்லது 'B' தரவகை பெற்ற மாணவர்களைத் தவிர மற்ற மாணவர்களின் விவரங்களைக் காண்பிக்கிறது.

WHERE துணை நிலைக் கூற்று NOT செயற்குறியுடன் உள்ள எடுத்துக்காட்டு

எடுத்தக்காட்டு15.5 -1

```
import sqlite3
connection = sqlite3.connect("Academy.db")
cursor = connection.cursor()
cursor.execute("SELECT * FROM student where grade<>'A' and Grade<>'B'")
result = cursor.fetchall()
print(*result,sep="\n")
```

வெளியீடு

```
(3, 'BASKAR', 'C', 'M', 75.2, '1998-05-17')
(7, 'TARUN', 'D', 'M', 62.3, '1999-02-01')
```

WHERE துணை நிலைக்கூற்று AND செயற்குறியுடன் உள்ள எடுத்துக்காட்டு

பின்வரும் எடுத்துக்காட்டு, சராசரியாக 80 லிருந்து 90 விழுக்காடு (இரண்டு வரம்பையும் சேர்த்து) வாங்கிய மாணவர்களின் பெயர், பதிவு எண், அவர்களின் சராசரி ஆகியவற்றைக் காண்பிக்கிறது.

எடுத்தக்காட்டு 15.5 -2

```
import sqlite3
connection = sqlite3.connect("Academy.db")
cursor = connection.cursor()
cursor.execute("SELECT Rollno,Sname,Average  FROM student WHERE
(Average>=80 AND Average<=90)")
result = cursor.fetchall()
print(*result,sep="\n")
```

வெளியீடு

```
(1, 'Akshay', 87.8)
(5, 'VARUN', 80.6)
```

WHERE துணை நிலைக்கூற்று OR செயற்குறியுடன் உள்ள எடுத்துக்காட்டு

பின்வரும் எடுத்துக்காட்டில் 2001ல் பிறந்த மாணவர்களின் பெயர் மற்றும் தரவரிசை காண்பிக்கப்பட்டுள்ளது.

எடுத்தக்காட்டு 15.5 -3

```
import sqlite3
connection = sqlite3.connect("Academy.db")
cursor = connection.cursor()
cursor.execute("SELECT Rollno,sname  FROM student WHERE (Average<60 OR
Average>70)")
result = cursor.fetchall()
print(*result,sep="\n")
```

OUTPUT

```
(1, 'Akshay')
(2, 'Aravind')
(3, 'BASKAR')
(4, 'SAJINI')
(5, 'VARUN')
(6, 'PRIYA')
```

15.6 தேதி உள்ள நெடுவரிசையில் வினவல்

பின்வரும் எடுத்துக்காட்டில் 2001ல் பிறந்த மாணவர்களின் பெயர் மற்றும் தரவரிசை காண்பிக்கப்பட்டுள்ளது.

எடுத்தக்காட்டு 15.6 -1

```
import sqlite3
connection = sqlite3.connect("Academy.db")
cursor = connection.cursor()
cursor.execute("SELECT Rollno,sname FROM student
WHERE(Birth_date>='2001-01-01' AND Birth_date<='2001-12-01')")
result = cursor.fetchall()
print(*result,sep="\n")
```

வெளியீடு

```
(5, 'VARUN')
```

15.7 மதிப்பீட்டுச் சார்புகள் (Aggregate Functions)

இந்த சார்புகள் நெடுவரிசையில் உள்ள மதிப்புகளைக் கொண்டு செயல்பாடுகளைச் செய்து ஒரே ஒரு மதிப்பை விடையாகக் கொடுக்கும்.

- COUNT()
- AVG()
- SUM()
- MAX()
- MIN()

15.7.1 COUNT() சார்பு

WHERE துணை நிலை கூற்றில் குறிப்பிட்டுள்ள நிபந்தனையை நிறைவேற்றும், அட்டவணையில் உள்ள வரிசைகளின் எண்ணிக்கையை SQL COUNT() சார்பு விடையாகக் கொடுக்கும்.

எடுத்தக்காட்டு 15.7.1-1

எடுத்தக்காட்டு 1 : இந்த எடுத்துக்காட்டில் வரிசைகளின் எண்ணிக்கை கணக்கிடப்பட்டுள்ளது.

```
import sqlite3
connection = sqlite3.connect("Academy.db")
cursor = connection.cursor()
cursor.execute("SELECT COUNT(*)  FROM student ")
result = cursor.fetchall()
print(result)
```

வெளியீடு

```
[(7,)]
```

எடுத்தக்காட்டு 15.7.1-2

எடுத்துக்காட்டு 2 : இந்த எடுத்துக்காட்டில் குறிப்பிட்ட நெடுவரிசையைக் கொண்ட பதிவுகளின் எண்ணிக்கை கணக்கிடப்பட்டுள்ளது.

```
import sqlite3
connection = sqlite3.connect("Academy.db")
cursor = connection.cursor()
cursor.execute("SELECT COUNT(AVERAGE)  FROM student ")
result = cursor.fetchall()
print(result
```

வெளியீடு

```
[(7,)]
```

குறிப்பு

NULL மதிப்புகள் கணக்கிடப்படமாட்டாது. மேலே உள்ள எடுத்துக்காட்டில் ஒரு வேளை பதிவில் NULL மதிப்பு இருந்தால் வெளியீடு 6 என இருக்கும்.

15.7.2AVG():

பின்வரும் பைதான் நிரலில் உள்ள கூற்று அனைத்து மாணவர்களின் சராசரி மதிப்பெண்ணைக் கண்டு பிடிக்கிறது.

Chapter 15.indd 308 30-01-2020 20:58:05

எடுத்தக்காட்டு 15.7.2-1

```
import sqlite3
connection = sqlite3.connect("Academy.db")
cursor = connection.cursor()
cursor.execute("SELECT AVG(AVERAGE)  FROM student ")
result = cursor.fetchall()
print(result)
```

வெளியீடு

```
[(84.65714285714286,)]
```

குறிப்பு

NULL மதிப்புகள் கணக்கிடப்படமாட்டாது.

15.7.3 SUM():

பின்வரும் பைதான் நிரலில் உள்ள SQL கூற்று சராசரி புலத்தில் உள்ள சராசரிகளின் கூட்டுத் தொகையைக் கணக்கிடுகிறது.

எடுத்தக்காட்டு 15.7.1-3

```
import sqlite3
connection = sqlite3.connect("Academy.db")
cursor = connection.cursor()
cursor.execute("SELECT SUM(AVERAGE)  FROM student ")
result = cursor.fetchall()
print(result)
```

வெளியீடு

```
[(592.6,)]
```

குறிப்பு

NULL மதிப்புகள் கணக்கிடப்படமாட்டாது.

15.7.4 MAX() மற்றும் MIN() செயற்கூறுகள்

The MAX() சார்பு தேர்ந்தெடுக்கப்பட்ட நெடுவரிசையில் உள்ள மிகப்பெரிய மதிப்பை விடையாகக் கொடுக்கும்.

The MIN() சார்பு தேர்ந்தெடுக்கப்பட்ட நெடுவரிசையில் உள்ள மிகச்சிறிய மதிப்பை விடையாகக் கொடுக்கும் .

சராசரி மதிப்பைக் கொண்ட மாணவரின் பெயர்களைக் காண்பிக்கிறது.

எடுத்தக்காட்டு 15.7.4-1

```
import sqlite3
connection = sqlite3.connect("Organization.db")
cursor = connection.cursor()
print("Displaying the name of the Highest Average")
cursor.execute("SELECT sname,max(AVERAGE)  FROM student ")
result = cursor.fetchall()
print(result)
print("Displaying the name of the Least Average")
cursor.execute("SELECT sname,min(AVERAGE)  FROM student ")
result = cursor.fetchall()
print(result)
```

வெளியீடு

```
Displaying the name of the Highest Average
[('PRIYA', 98.6)]
Displaying the name of the Least Average
[('TARUN', 62.3)]
```

15.8 பதிவுகளைப் புதுப்பித்தல்

பைதான் ஸ்கிரிப்ட் மூலமாக பதிவுகளைப் புதுப்பிக்க முடியும். பின்வரும் எடுத்துக்காட்டில் “student” அட்டவணையில் “Priya” என்ற பெயர் “Priyanka” என புதுப்பிக்கப்படுகிறது.

எடுத்தக்காட்டு 15.8 -1

```
# code for update operation
import sqlite3
# database name to be passed as parameter
conn = sqlite3.connect("Academy.db")
# update the student record
conn.execute("UPDATE Student SET sname ='Priyanka' where Rollno='6' ")
conn.commit()
print ("Total number of rows updated :", conn.total_changes)
cursor = conn.execute("SELECT * FROM Student")
for row in cursor:
 print (row)
conn.close()
```


Chapter 15.indd 310
30-01-2020 20:58:05

வெளியீடு

```
Total number of rows updated : 1
(1, 'Akshay', 'B', 'M', 87.8, '2001-12-12')
(2, 'Aravind', 'A', 'M', 92.5, '2000-08-17')
(3, 'BASKAR', 'C', 'M', 75.2, '1998-05-17')
(4, 'SAJINI', 'A', 'F', 95.6, '2002-11-01')
(5, 'VARUN', 'B', 'M', 80.6, '2001-03-14')
(6, 'Priyanka', 'A', 'F', 98.6, '2002-01-01')
(7, 'TARUN', 'D', 'M', 62.3, '1999-02-01')
```

குறிப்பு

இந்தப் பாடப்பகுதி முழுவதும் SQL வினவல்களை விளக்குவதற்காக student என்ற தரவு அட்டவணை எடுத்துக்காட்டாக பயன்படுத்தப்பட்டுள்ளது என்பதை நினைவில் கொள்க. எடுத்துக்காட்டு 15.3.2-1. பதிவுகள் கொண்ட student தரவு அட்டவணை பயன்படுத்தப்பட்டுள்ளது.

15.9 நீக்குதல் செயல்பாடு

SQL கட்டளைகளைக் கொண்டு பதிவுகளை நீக்குவது போல பைதானிலும் நீக்க முடியும். பின்வரும் எடுத்துக்காட்டில் “student ” அட்டவணையில் Rollno 2 புலத்தில் உள்ள பொருளடக்கத்தை நீக்குவது காட்டப்பட்டுள்ளது.

எடுத்தக்காட்டு 15.9-1

```
# code for delete operation
import sqlite3
  # database name to be passed as parameter
conn = sqlite3.connect("Academy.db")
# delete student record from database
conn.execute("DELETE from Student where Rollno='2'")
conn.commit()
print("Total number of rows deleted :", conn.total_changes)
cursor =conn.execute("SELECT * FROM Student")
for row in cursor:
        print(row)
conn.close()
```

வெளியீடு

```
Total number of rows deleted : 1
(1, 'Akshay', 'B', 'M', 87.8, '2001-12-12')
(3, 'BASKAR', 'C', 'M', 75.2, '1998-05-17')
(4, 'SAJINI', 'A', 'F', 95.6, '2002-11-01')
(5, 'VARUN', 'B', 'M', 80.6, '2001-03-14')
(6, 'Priyanka', 'A', 'F', 98.6, '2002-01-01')
(7, 'TARUN', 'D', 'M', 62.3, '1999-02-01')
```

15.10 பயனரால் உள்ளிடப்படும் தரவு

பின்வரும் எடுத்துக்காட்டில் இயங்கு நேரத்தில் பைதான் input() கட்டளையைப் பயன்படுத்தி தரவுகளைப் பெற்று அதை "Person" என்ற அட்டவணையில் எழுதுதல்

எடுத்தக்காட்டு 15.10 -1

```
# code for executing query using input data
import sqlite3
# creates a database in RAM
con =sqlite3.connect("Academy.db")
cur =con.cursor()
cur.execute("DROP Table person")
cur.execute("create table person (name, age, id)")
print("Enter 5 students names:")
who =[input() for i in range(5)]
print("Enter their ages respectively:")
age =[int(input()) for i in range(5)]
print("Enter their ids respectively:")
p_id =[int(input())for i in range(5)]
n =len(who)
for i in range(n):
        # This is the q-mark style:
        cur.execute("insert into person values (?, ?, ?)", (who[i], age[i], p_id[i]))
# And this is the named style:
cur.execute("select * from person")
# Fetches all entries from table
print("Displaying All the Records From Person Table")
print (*cur.fetchall(), sep='\n' )
```


Chapter 15.indd 312 30-01-2020 20:58:05

வெளியீடு

```
Enter 5 students names:
        RAM
        KEERTHANA
        KRISHNA
        HARISH
        GIRISH
Enter their ages respectively:
        28
        12
        21
        18
        16
Enter their ids respectively:
        1
        2
        3
        4
        5
Displaying All the Records From Person Table
        ('RAM', 28, 1)
        ('KEERTHANA', 12, 2)
        ('KRISHNA', 21, 3)
        ('HARISH', 18, 4)
        ('GIRISH', 16, 5)
```

ஏற்கனவே உள்ள "student" என்ற அட்டவணையில் Name புலத்தில் சில மாற்றங்களைச் செய்து பதிவுகளைச் சேர்க்கலாம். இதை செய்ய Create table கூற்றைக் குறிப்புரையில் (Comment) எழுதவும்.

குறிப்பு

Execute (sql[, parameters]) :- இது ஒற்றை SQL கூற்றை இயக்குகிறது. SQL கூற்று அளபுருக்களை கொண்டிருக்கலாம் (அதாவது நிலையுருக்களுக்குப் பதிலாக இடநிரப்பிகள்). sqlite3 இரண்டு வகையான இடநிரப்பிகளை கொண்டுள்ளது. அவை, கேள்விக் குறிகள் (?) மற்றும் பெயரிடப்பட்ட இடநிரப்பிகள் (:name). எ.கா who= :who ,age=:age

15.11 பல அட்டவணைகளைப் பயன்படுத்தி வினவல்

ஒன்றுக்கு மேற்பட்ட அட்டவணைகளைச் சேர்த்து பைத்தானில் வினவலை மேற்கொள்ள முடியும். பின்வரும் எடுத்துக்காட்டில் மாணவர்களின் Rollno, Duty, Age ஆகிய விவரங்களைக் கொண்ட "Appointments" அட்டவணையை உருவாக்குவோம். "student" மற்றும் "Appointments" அட்டவணைகளை ஒன்றிணைத்து நெடுவரிசையின் தலைப்புகளுடன் கூடிய இறுதி அட்டவணையைக் காணலாம்.

எடுத்தக்காட்டு 15.11-1

```
import sqlite3
connection = sqlite3.connect("Academy.db")
cursor = connection.cursor()
cursor.execute("""DROP TABLE Appointment;""")
sql_command = """
CREATE TABLE Appointment(rollnointprimarkey,Dutyvarchar(10),age int)"""
cursor.execute(sql_command)
sql_command = """INSERT INTO Appointment (Rollno,Duty ,age )
        VALUES ("1", "Prefect", "17");"""
cursor.execute(sql_command)
sql_command = """INSERT INTO Appointment (Rollno, Duty, age)
        VALUES ("2", "Secretary", "16");"""
cursor.execute(sql_command)
#  never forget this, if you want the changes to be saved:
connection.commit()
cursor.execute("SELECT student.rollno,student.sname,Appointment.
      Duty,Appointment.Age FROM student,Appointment where student.
      rollno=Appointment.rollno")
#print (cursor.description) to display the field names of the table
co = [i[0] for i in cursor.description]
print(co)
#  Field informations can be read from cursor.description.
result = cursor.fetchall()
for r in result:
        print(r)
```

வெளியீடு

```
        ['Rollno', 'Sname', 'Duty', 'age']
        (1, 'Akshay', 'Prefect', 17)
        (2, 'Aravind', 'Secretary', 16)
```

குறிப்பு

cursor.description ஒவ்வொரு நெடுவரிசையின் தலைப்புகளின் விவரங்களைக் கொண்டிருக்கும். இது பதிவாக சேமிக்கப்படும். முதல் பதிவு நெடுவரிசையின் பெயர்கைளக் குறிக்கும் (index=0) இதை பயன்படுத்தி அட்டவணையின் புலங்களின் பெயர்களைக் காண்பிக்க முடியும்.

15.12 வினவலை CSV கோப்புடன் ஒருங்கிணைத்தல்

CSV கோப்பிலும் வினவலின் விடையைச் சேமிக்கலாம். வினவலின் வெளியீட்டை அட்டவணையில் காண்பிப்பதற்கு பயன்படும். பின்வரும் எடுத்துக்காட்டில் student அட்டவணையில் உள்ள gender நெடுவரிசையைப் பயன்படுத்தி இறங்கு வரிசையில் வரிசையாக்கம் செய்து பிறகு பதிவுகளை அகரவரிசைப்படி அமைக்கப்பட்டுள்ளது. இந்த வினவலின் வெளியீட்டை "SQL.CSV" என்னும் CSV கோப்பில் எழுதலாம். இந்த கோப்பினை MS-Excel மூலம் திறந்து முடிவுகளை காணலாம் அல்லது ஸ்கிரிப்ட்டைப் பயன்படுத்தியும் CSV கோப்புகளைத் திறந்து முடிவுகளைக் காணலாம்.

Chapter 15.indd 314 30-01-2020 20:58:05

எடுத்தக்காட்டு 15.12 -1

```
import sqlite3
import io                                    # to access replace()
import csv
# CREATING CSV FILE
d=open('c:/pyprg/sql.csv','w')
c=csv.writer(d)
connection = sqlite3.connect("Academy.db")
cursor = connection.cursor()
# a=Connection.cursor()
cursor.execute("SELECT *  FROM student ORDER BY GENDER DESC,SNAME")
# WRITING THE COLUMN HEADING
co = [i[0] for i in cursor.description]
c.writerow(co)
data=cursor.fetchall()
for item in data:
        c.writerow(item)
d.close()
# Reading the CSV File
# replace() is used to eliminate the newline character at the end of each row
with open('c:/pyprg/sql.csv', "r", newline=None) as fd:
        for line in fd:
                line = line.replace("\n", "")
                print(line)
cursor.close()
connection.close()
```

வெளியீடு

```
Rollno,Sname,Grade,gender,Average,birth_date
1, Akshay, B, M, 87.8, 2001-12-12
2, Aravind, A, M, 92.5, 2000-08-17
3, BASKAR, C, M, 75.2, 1998-05-17
7, TARUN, D, M, 62.3, 1999-02-01
5, VARUN, B, M, 80.6, 2001-03-14
6, PRIYA, A, F, 98.6, 2002-01-01
4, SAJINI, A, F, 95.6, 2002-11-01
```

இதுவரை, பைத்தான் நிரல் எடுத்துக்காட்டுகளில் SQL கட்டளைகளைப் பயன்படுத்துவதைப் பார்த்தோம். ஏறக்குறைய அனைத்து SQL கட்டளைகளையும் பயன்படுத்துவதைப் பார்த்தோம். ஏறக்குறைய அனைத்து SQL கட்டளைகளும் பைதான் SQLite கூறுகளைப் பயன்படுத்தி நிறைவேற்றப்பட்டுள்ளது. SQL பாடத்தில் விவாதித்த பிற கட்டளைகளையும் முயற்சிக்கவும்.

எடுத்துக்காட்டு 15.12 -2 கோப்பினை ("sqlexcel.csv") MS-Excel மூலம் திறந்து அதனுடைய வெளியீட்டை பார்க்கவும். (எடுத்துக்காட்டு: 15.12.1 Script நிரலை ஒத்ததாகும்)

```
import sqlite3
import io                                   #to access replace()
import csv
# database name to be passed as parameter
conn = sqlite3.connect("Academy.db")
print("Content of the table before sorting and writing in CSV file")
cursor = conn.execute("SELECT * FROM Student")
for row in cursor:
 print (row)
#  CREATING CSV FILE
d=open('c:\\pyprg\\sqlexcel.csv','w')
c=csv.writer(d)
cursor = conn.cursor()
cursor.execute("SELECT *  FROM student ORDER BY GENDER DESC,SNAME")
#WRITING THE COLUMN HEADING
co = [i[0] for i in cursor.description]
c.writerow(co)
data=cursor.fetchall()
for item in data:
        c.writerow(item)
d.close()
print("sqlexcel.csv File is created open by visiting c:\\pyprg\\sqlexcel.csv")
conn.close()
```

வெளியீடு

```
Content of the table before sorting and writing in CSV file
(1, 'Akshay', 'B', 'M', 87.8, '2001-12-12')
(2, 'Aravind', 'A', 'M', 92.5, '2000-08-17')
(3, 'BASKAR', 'C', 'M', 75.2, '1998-05-17')
(4, 'SAJINI', 'A', 'F', 95.6, '2002-11-01')
(5, 'VARUN', 'B', 'M', 80.6, '2001-03-14')
(6, 'Priyanka', 'A', 'F', 98.6, '2002-01-01')
(7, 'TARUN', 'D', 'M', 62.3, '1999-02-01')
sqlexcel.csv File is created open by visiting c:\\pyprg\\sqlexcel.csv
```


Chapter 15.indd 316 30-01-2020 20:58:05

OUTPUT THROUGH EXCEL

	A	B	C	D	E	F
1	Rollno	Sname	Grade	gender	Average	birth_date
2						
3	1	Akshay	B	M	87.8	2001-12-12
4						
5	2	Aravind	A	M	92.5	2000-08-17
6						
7	3	BASKAR	C	M	75.2	1998-05-17
8						
9	7	TARUN	D	M	62.3	1999-02-01
10						
11	5	VARUN	B	M	80.6	2001-03-14
12						
13	6	PRIYA	A	F	98.6	2002-01-01
14						
15	4	SAJINI	A	F	95.6	200-11-01

15.3 அட்டவணை பட்டியல்

தரவுத்தளத்தில் உருவாக்கப்பட்ட அட்டவணைகளின் பட்டியல்களை காண்பிப்பதற்கு, பின்வரும் நிரலை (எ.கா. 15.3.1) பயன்படுத்தலாம்.

எடுத்துக்காட்டு 15.3 – 1

```
import sqlite3
con = sqlite3.connect('Academy.db')
cursor = con.cursor()
cursor.execute("SELECT name FROM sqlite_master WHERE type='table';")
print(cursor.fetchall())
```

வெளியீடு

```
[('Student',), ('Appointment',), ('Person',)]
```

மேற்கண்ட நிரல் 'Academy.db' என்ற தரவுத்தளத்தில் உருவாக்கப்பட்ட அனைத்து அட்டவணைகளின் பெயர்களை திரையில் காண்பிக்கும். நமது தரவுதள அட்டவணைகளின் முக்கிய தகவல்களை முதன்மை அட்டவணை கொண்டிருக்கும். sqlite_master எனப்படும்.

இதுவரை நாம் பைத்தான் நிரல் எடுத்துக்காட்டுகளில் SQL கட்டளைகளை பயன்படுத்துவதைப் பார்த்தோம். ஏறக்குறைய அனைத்து SRL கட்டளை பயன்படுத்துவதைப் பார்த்தோம். ஏறக்குறைய அனைத்து SQL கட்டளை பைத்தான் SQLite கூறுகளைப் பயன்படுத்தி நிறைவேற்றப்பட்டுள்ளது. SQL பாடத்தில் விவாதித்த பிற கட்டளைகளையும் முயற்சிக்கவும்.

Chapter 15.indd 317 30-01-2020 20:58:06

நினைவில் கொள்க

- ஒரு தரவுத்தளம் என்பது ஒருங்கிணைக்கப்பட்ட தரவுகளின் தொகுப்பாகும்.
- தரவுத்தளத்தின் பயனர்களாக மனிதர்கள், பிற நிரல்கள் அல்லது பயன்பாடுகள் இருக்கலாம்.
- எளிய உறவுநிலை தரவுத்தள அமைப்பான இது தரவுகளை முறையான தரவுக் கோப்புகளாக சேமித்து வைக்கும்.
- தரவுத்தளத்திலுள்ள cursor என்பது தரவுத்தள பதிவுகளின் மீது செயல்படும் ஒரு கட்டுப்பாட்டு அமைப்பாகும். SQL –ன் அனைத்து கட்டளைகளையும் செயல்படுத்த இது பயன்படுகிறது.
- அட்டவணையிலுள்ள தரவு ஒற்றை அல்லது இரட்டை மேற்கோள்குறியுடன் உள்ளதால், பைத்தானிலுள்ள SQL கட்டளைகள் மூன்று மேற்கோள் குறியினால் குறிக்கப்படும்.
- SQL-ல் மிகவும் பொதுவாக பயன்படுத்தக்கூடிய கூற்று Select ஆகும்.
- தரவுத்தளத்திலுள்ள அட்டவணையிலிருந்து தரவுகளைப் பெற SQL கூற்றான Select பயன்படுகிறது.
- GROUP BY துணைநிலை கூற்று குறிப்பிட்ட பதிவுகளைச் சுருக்கமான வரிசைகளைக் கொண்ட குழுவாக சேர்க்கிறது.
- ORDER BY துணை நிலைக்கூற்று SELECT கூற்றுடன் சேர்ந்து குறிப்பிட்ட புலங்களில் உள்ள தரவுகளை முறையாக வரிசையாக்கம் செய்ய பயன்படுகிறது.
- Having துணைநிலைக்கூற்று GROUP செயற்கூறின் அடிப்படையில் தரவுகளை வடிகட்ட பயன்படுகிறது.
- Where துணை நிலைக்கூற்று 'Group by' துணை நிலைக்கூற்றுடன் பயன்படுத்த முடியாது.
- WHERE துணைநிலைக்கூற்று AND, OR, மற்றும் NOT செயற்குறிகளுடன் இணைத்துப் பயன்படுத்தலாம்.
- ஒன்றிற்கு மேற்பட்ட நிபந்தனைகளின் அடிப்படையில் பதிவுகளை வடிகட்ட 'AND' மற்றும் 'OR' செயற்குறிகள் பயன்படுகின்றன.
- மதிப்பீட்டுச் சார்புகள் நெடுவரிசையில் உள்ள மதிப்புகளைக் கொண்டு செயல்பாடுகளைச் செய்து ஒரே ஒரு மதிப்பை விடையாகக் கொடுக்கும்.
- COUNT() சார்பு அட்டவணையிலுள்ள வரிசைகளின் எண்ணிக்கையை திருப்பி அனுப்பும்.
- AVG() சார்பு அட்டவணையிலுள்ள தேர்ந்தெடுக்கப்பட்ட புலத்தின் பதிவுகளின் சராசரியை / கணக்கிடப்பயன்படுகிறது.
- SUM() சார்பு அட்டவணையிலுள்ள தேர்ந்தெடுக்கப்பட்ட புலத்தின் பதிவுகளின் கூட்டுத் தொகையைக் கணக்கிடுகிறது.
- MAX() சார்பு தேர்ந்தெடுக்கப்பட்ட புலத்தின் பெரிய மதிப்பை திருப்பி அனுப்பும்.
- MIN() சார்பு தேர்ந்தெடுக்கப்பட்ட புலத்தின் சிறிய மதிப்பை திருப்பி அனுப்பும்
- sqlite_master என்பது முதன்மை அட்டவணையாகும். இது நமது தரவுத்தள அட்டவணைகளின் முக்கிய தகவல்களை கொண்டிருக்கும்.
- பைத்தானில், கோப்பின் பாதையை '/' அல்லது '\\' குறியீடு குறிக்கப்படுகிறது. எடுத்துக்காட்டாக, பாதையை 'c:/pyprg/sql.csv', அல்லது c:\\pyprg\\sql.csv' எனக் குறிப்பிடலாம்.

Chapter 15.indd 318 30-01-2020 20:58:06

செய்முறைப் பயிற்சி

1. பயனரிடமிருந்து விவரங்களைப் பெற்று, பைத்தானைப் பயன்படுத்தி CSV கோப்பாக பின்வரும அட்டவணையில் சேமிக்க ஊடாடும் நிரலை உருவாக்கவும்

   தரவுத்தளத்தின் பெயர்;- DB1

   அட்டவணை பெயர் : Customer

Cust_Id	Cust_Name	Address	Phone_no	City
C008	Sandeep	14/1 Pritam Pura	41206819	Delhi
C010	Anurag Basu	15A, Park Road	61281921	Kolkata
C012	Hrithik	7/2 Vasant Nagar	26121949	Delhi

2. பின்வரும் GAMES அட்டவணையை எடுத்துக் கொள்வோம். (i) லிருந்து (iv) வரையிலான SQL வினவல்களுக்கான வெளியீட்டை கொடுக்கும் பைத்தான் நிரலை எழுதுக.

   Table: GAMES

Gcode	Name	GameName	Number	PrizeMoney	ScheduleDate
101	Padmaja	Carom Board	2	5000	01-23-2014
102	Vidhya	Badminton	2	12000	12-12-2013
103	Guru	Table Tennis	4	8000	02-14-2014
105	Keerthana	Carom Board	2	9000	01-01-2014
108	Krishna	Table Tennis	4	25000	03-19-2014

(i) அனைத்து Gamename மற்றும் Gcodes-களும் Schedule date-ன் இறங்கு வரிசையில் பதிவுகளை திரையிடவும்.

(ii) எந்த விளையாட்டின் பரிசுத்தொகை 7000- விட அதிகமாக இருக்கிறதா அந்த விளையாட்டுகளின் விவரங்கள் அடங்கிய பதிவுகளைத் திரையிடவும்.

(iii) விளையாட்டு வீரர்களின் name மற்றும் gamename-ன் ஏறுவரிசையில் பதிவுகளை திரையிடவும்.

(iv) ஒவ்வொரு விளையாட்டு குழுவின் வீர்ருக்கான பரிசுத்தொகை அடங்கிய பதிவுகளை திரையிடவும் (புலம் 4-ல் கொடுக்கப்பட்டுள்ளதைப் போல்)

(v) GameName-ன் அடிப்படையில் அனைத்து பதிவுகளையும் திரையிடவும்.

மதிப்பீடு

பகுதி–அ

சரியான விடையை தேர்ந்தெடுத்து எழுதுக (1 மதிப்பெண்)

C9XK3F

1. பின்வரும் எது ஒருங்கிணைக்கப்பட்ட தரவுகளின் தொகுப்பாகும்?

   (அ) தரவுத்தளம் (ஆ) DBMS (இ) தகவல் (ஈ) பதிவுகள்

2. SQLite எந்த தரவுத்தள அமைப்பைச் சார்ந்தது?

   (அ) ஒற்றைக் கோப்பு தரவுத்தளம் (ஆ) உறவுநிலை தரவுத்தளம்

   (இ) படிநிலை தரவுத்தளம் (ஈ) பொருள்நோக்கு தரவுத்தளம்

3. பின்வரும் எந்த கட்டுப்பாட்டு அமைப்பு தரவுத்தளத்திலிருந்து பதிவுகளைப் பெற்றுத்தர பயன்படுகிறது?

   (A) சுட்டு (ஆ) திறவுகோல்

   (இ) Cursor (ஈ) செருகும் புள்ளி

4. பதிவுகளில் உள்ள மதிப்புகளில் செய்யப்படும் மாற்றங்களை சேமிக்கப் பயன்படும் கட்டளை எது?

   (அ) Save (ஆ) Save As (இ) Commit (ஈ) Oblige

5. சில செயல்பாடுகளை SQL கட்டளைகள் செய்வதற்கு பின்வரும் எது இயக்கப்படுகிறது?

   (அ) execute() (ஆ) key() (இ) cursor() (ஈ) run()

6. பின்வரும் எந்த சார்பு அட்டவணையிலுள்ள தேர்ந்தெடுக்கப்பட்ட புலத்தின் பதிவுகளின் சராசரியைக் கொடுக்கிறது?

   (அ) Add() (ஆ) SUM() (இ) AVG() (ஈ) AVERAGE()

7. எந்த செயற்கூறு தேர்ந்தெடுக்கப்பட்ட புலத்தின் பெரிய மதிப்பைத் திருப்பி அனுப்பும்

   (அ) MAX() (ஆ) LARGE()

   (இ) HIGH() (ஈ) MAXIMUM()

8. பின்வரும் எது முதன்மை அட்டவணை?

   (அ) sqlite_master (ஆ) sql_master

   (இ) main_master (ஈ) master_main

9. SQL-ல் மிகவும் பொதுவாக பயன்படுத்தப்படும் கூற்று எது?

   (அ) cursor (ஆ) select (இ) execute (ஈ) commit

10. பின்வரும் எந்த துணை நிலைக்கூற்று நகல்களைத் தவிர்க்கும்?

    (அ) Distinct (ஆ) Remove (இ) Where (ஈ) GroupBy

Chapter 15.indd 320 30-01-2020 20:58:06

பகுதி–ஆ

அனைத்து வினாக்களுக்கும் விடையளி (2 மதிப்பெண்)

1. தரவுத்தளத்தைப் பயன்படுத்தும் பயனர்களை குறிப்பிடவும்.
2. தரவுத்தளத்தை இணைக்க பயன்படும் முறைகள் யாவை? எடுத்துக்காட்டு தருக.
3. புலத்தை "INTEGER PRIMARY KEY" என அறிவிப்பதன் நன்மை என்ன?
4. அட்டவணையில் பதிவுகளை விரிவுப்படுத்துவதற்கான கட்டளையை எழுதுக. எடுத்துக்காட்டு தருக.
5. தரவுத்தள அட்டவணையிலிருந்து அனைத்து பதிவுகளையும் பெறுவதற்கான வழிமுறை எது?

பகுதி–இ

அனைத்து வினாக்களுக்கும் விடையளி (3 மதிப்பெண்)

1. SQLite என்றால் என்ன ? இதன் நன்மைகள் யாவை?
2. fetchone() மற்றும் fetchmany() வேறுபடுத்துக.
3. Where துணைநிலைக்கூற்றின் பயன் என்ன?where கூற்றைப் பயன்படுத்தி ஒரு பைத்தான் கூற்றை எழுதவும்.
4. பின்வரும் விவரங்களை படிக்கவும். அதன் அடிப்படையில் துறைவாரியாக பதிவுகளை திரையிட பைத்தான் ஸ்கிரிப்ட்டை எழுதவும்.

   தரவுத்தள பெயர் :- organization.db

   அட்டவணை பெயர் :- Employee

   புலங்கள் :- Eno, EmpName, Esal, Dept
5. பின்வரும் விவரங்களை படிக்கவும் அதன் அடிப்படையில் பதிவுகளை Eno இறங்குவரிசையில் திரையிட பைத்தான் ஸ்கிரிப்ட்டை எழுதவும்.

   தரவுத்தள பெயர் :- organization.db

   அட்டவணை பெயர் :- Employee

   புலங்கள் :- Eno, EmpName, Esal, Dept

பகுதி ஈ

அனைத்து வினாக்களுக்கும் விடையளி (5 மதிப்பெண்)

1. SQLite பற்றி விரிவாக எழுதவும். அதனை பயன்படுத்தும் படிநிலைகளை எழுதுக.
2. fetchmany() பயன்படுத்தி பின்வரும் அட்டவணையிலுள்ள அனைத்து பதிவுகளையும் திரையிடுவதற்கான பைத்தான் ஸ்கிரிப்ட்டை எழுதவும்

Icode	ItemName	Rate
1003	Scanner	10500
1004	Speaker	3000

1005	Printer	8000
1008	Monitor	15000
1010	Mouse	700

3. HAVING துணைநிலைக்கூற்றின் பயன் யாது? எடுத்துக்காட்டு தருக.

4. பின்வரும் குறிப்புகளைக் கொண்டு ITEM என்ற அட்டவணையை உருவாக்க பைத்தான் ஸ்கிரிப்ட்டை எழுதவும்.

   அட்டவணைக்கு ஒரு பதிவை சேர்க்கவும்.

   தரவுத்தளத்தின் பெயர் :- ABC

   அட்டவணையின் பெயர் :- Item

   நெடுவரிசையின் பெயர் மற்றும் விவரங்கள் :-

Icode	:-	integer and act as primary key
Item Name	:-	Character with length 25
Rate	:-	Integer
Record to be added	:-	1008, Monitor,15000

5. பின்வரும் supplier மற்றும் Item அட்டவணைகளை கவனித்து, (i) மற்றும் (ii) வினாக்களுக்கு பைத்தான் ஸ்கிரிப்ட்டை எழுதவும்.

SUPPLIER				
Suppno	**Name**	**City**	**Icode**	**SuppQty**
S001	Prasad	Delhi	1008	100
S002	Anu	Bangalore	1010	200
S003	Shahid	Bangalore	1008	175
S004	Akila	Hydrabad	1005	195
S005	Girish	Hydrabad	1003	25
S006	Shylaja	Chennai	1008	180
S007	Lavanya	Mumbai	1005	325

i) டெல்லியில் வசிக்காத மொத்த விற்பனையாளர்களின் Name,City மற்றும் Itemname களை திரையிடவும்.

ii) அகிலாவின் suppQty யில் உள்ள மதிப்போடு 40-யை அதிகரிக்கும்

References

1. *The Definitive Guide to SQLite by Michael Owens*
2. *Programming for Beginners: 2 Manuscripts: SQL & Python by Byron Francis*
3. *Tutorialspoint.com*

Chapter 15.indd 322 30-01-2020 20:58:06

அலகு V

பாடம் 16

தரவு காட்சிப்படுத்துதல் :PYPLOT பயன்படுத்தி கோட்டு வரைபடம், வட்ட உரைப்படம் மற்றும் பட்டை வரைப்படம் உருவாக்குதல்

5W4AP6

கற்றலின் நோக்கங்கள்

இந்தப் பாடப்பகுதியைக் கற்றபின் மாணவர்கள் அறிந்துக் கொள்வது,

- தரவு காட்சிப்படுத்துதல் வரையறுக்க
- தரவு காட்சிப்படுத்துதல் வகைகளை பட்டியலிட.
- தரவு காட்சிப்படுத்துதல் பயன்களை பட்டியலிட.
- Matplotlib உள்ள காட்சிப்படுத்தலின் வகைகள் .
- Matplotlib நிறுவுதல்.
- Matplotlib இறக்கம் பற்றி அறிந்து கொள்ளுதல்.
- தரவு காட்சிப்படுத்துதல் தொகுப்புகளை வகைப்படுத்துதல்.
- Matplotlib கொண்டு பல்வேறு வகை வரைபடங்கள் (plot) உருவாக்குதல்

16.1 தரவு காட்சிப்படுத்துதல் வரையறை

தரவு காட்சிப்படுத்துதல் என்பது தரவு மற்றும் தகவல்களை வரைகலையாக உருவாக்குகின்றது. தரவு காட்சிப்படுத்தலின் முக்கிய நோக்கம் பயனாளர்களுக்கு தகவல்களை படக் காட்சி முறையில் காண்பிப்பது. இதற்கு, தரவை காட்சிப்படுத்துதல், புள்ளியியல் வரைகலை முறையைப் பயன்படுத்துகிறது. இம்முறையில் எண்வகை தரவு, புள்ளி, கோடு அல்லது பட்டையைக் கொண்டு, குறியாக்கப்பட்டு, அளவைக்குரிய செய்திகளை காட்சிப்படுத்துவதன் மூலம் அறிவிக்கலாம்.

தரவு காட்சிப்படுத்தலின் வகைகள்

- வரைப்படங்கள் (Charts)
- அட்டவணைகள் (Tables)
- வரைகலை (Graphs)
- நிலப்படங்கள் (Maps)
- இன்போகிராபிக்ஸ் (Infographics)
- டேஷ்போர்ட் (Dashboards)

Chapter 16.indd 323 10-02-2020 20:06:50

தரவு காட்சிப்படுத்தலின் பயன்கள்

- தரவு காட்சிப்படுத்தல் பயனர்கள் தரவுகளை எளிதாக கூர்ந்து ஆய்வு செய்யவும், உட்பொருளை வெளிப்படுத்த உதவுகிறது.
- இது சிக்கலான தரவுகளை புரிந்துக் கொண்டு அவற்றை பயன்படுத்திக் கொள்ள வழி செய்கிறது.
- தரவு காட்சிப்படுத்தல் பல்வேறு வரைப்படங்களைக் கொண்டு தரவு மாறிகளுக்கு இடையே உள்ள உறவு நிலையை வெளிப்படுத்துகிறது.

உங்களுக்குத் தெரியுமா?

இன்போகிராபிக்ஸ் → இன்போகிராபிக்ஸ் என்பது தகவல்களை வரைகலை முறையில் உருவமைக்கப் பயன்படுகிறது.

டேஷ்போர்ட் → டேஷ்போர்ட் அனைத்து வளங்களையும் ஒன்றுபட்ட ஒற்றை காட்சி திரையில் காண்பிக்க பயன்படுகிறது. தரவு காட்சிப்படுத்துதல் மற்றும் டேஷ்போர்ட் சிக்கலான யோசனை மற்றும் கருத்துக்களை எளிமையான காட்சி வடிவத்தில் மாற்றும் உரையில் கண்டறியபட முடியாத. வடிவங்கள் மற்றும் உறவு நிலைகளை டேஷ்போர்ட் ஒரே பார்வையில் கண்டறிய பயன்படும்.

Matplotlib அறிமுகம் — பைத்தானில் தரவு காட்சிப்படுத்தல்

பைத்தானில் Matplotlib என்பது பிரபலமான தரவு காட்சிப்படுத்தல் நூலகம் ஆகும். குறைந்து அளவிலான குறிமுறைகள் கொண்டு வரைப்படங்களை உருவாக்க முடியும்.

Matplotlib காட்சிப்படுத்தல் வகைகள்

Matplotlib ல் பல்வேறு வகையான காட்சிப்படுத்தல் உள்ளன. அவை

- வரி வரைவிடம் (Line plot)
- ஸ்கேட்டர் வரைவிடம் (Scatter plot)
- ஹிஸ்டோகிராம் (Histogram)
- பெட்டி வரைவிடம் (Box plot)
- பட்டை வரைப்படம் (Bar chart)
- வட்ட வரைப்படம் (Pie chart)

உங்களுக்குத் தெரியுமா?

ஸ்கேட்டர் வரைவிடம்: ஸ்கேட்டர் வரைவு என்பது தரவுகளை புள்ளிகளின் தொகுப்பாக காட்டுகிறது. ஒரு புள்ளியின் நிலை அதன் இரு பரிமாண மதிப்பைப் பொறுத்தது, ஒவ்வொரு மதிப்பும் கிடைமட்ட அல்லது செங்குத்து பரிமாணத்தின் நிலையை குறிக்கும்.

பெட்டி வரைவிடம்: பெட்டி வரைவிடம் என்பது சிறிய முதல் கால்மானம், சராசரி, மூன்றாம் கால்மானம் மற்றும் பெரிய (**minimum, first quartile, median, third quartile and maximum**) ஆகிய ஐந்து எண்கள் திரட்டைக் கொண்டு தரவுகளின் பகிர்வை காட்டுகிறது.

Matplotlib நிறுவுதல்

Pip பயன்படுத்தி matplotlib நாம் நிறுவ முடியும். Pip என்பது பைத்தான் தொகுப்புகளை நிறுவுவதற்கான ஒரு மேலாண்மை மென்பொருள் ஆகும்.

குறிப்பு

PiP நிறுவுதல் முறையை இணைப்பில் கொடுக்கப்பட்டுள்ளது

16.2 தொடங்குதல்

Matplotlib நிறுவிய பின், Matplotlib பதிவிறக்கம் செய்து பின்வரும் கட்டளையை பயன்படுத்தி குறியீட்டை தொடங்கலாம்.

import matplotlib.pyplot as plt

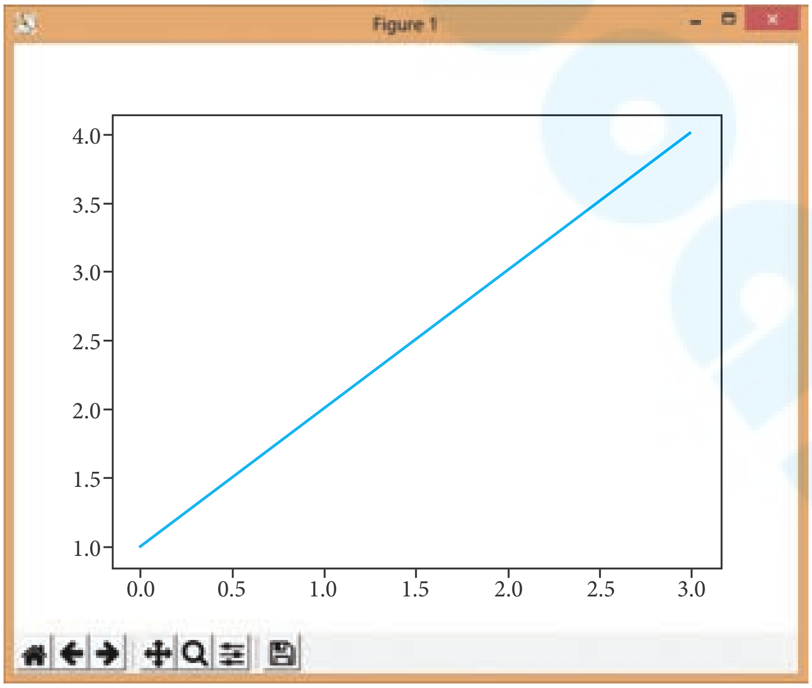
இப்போது உங்கள் பணியிடத்தில் Matplotlib பதிவிறக்கம் ஆகிவிட்டது. இப்பொழுது வரைவிடத்தை காண்பிக்க வேண்டும். பைத்தான் ஸ்கிரிப்டிலிருந்து matplotlib ஐ பயன்படுத்தி, lt.show() என்ற முறையை கொண்டு வரைவிடத்தை காட்டலாம்.

எடுத்துக்காட்டு

```
import matplotlib.pyplot as plt
plt.plot([1,2,3,4])
plt.show()
```

வெளியீடு

இந்த சாளரம் ஒரு matplotlib சாளரம் ஆகும். இது வரைப்படத்தை பார்க்க உதவுகிறது. நீங்கள் வரைப்படத்தில் மீது சுட்டியை நகர்த்தினால், ஒருங்கிணைப்பு புள்ளிகளை வலது புற ஓரத்தில் காணலாம்.



படம் *16.1*

ஏன் x-அச்சின் வரம்பு 0-3 மற்றும் y-அச்சின் வரம்பு 1-4 என வரக்கூடும் என்று நீங்கள் வியக்கலாம். நீங்கள் plot() கட்டளைக்கு ஒற்றை பட்டியல் அல்லது அணியாக மதிப்புகளை கொடுக்கும் போது, y அச்சின் தொடர் மதிப்புகளாக matplotlib எடுத்துக் கொண்டு, x அச்சின் மதிப்புகளை உருவாக்கும். பைத்தான் வரம்புகள் 0 உடன் தொடங்கும் என்பதால், இயல்பாகவே x திசை மற்றும் Y திசையை போன்று ஒரே நீளம் கொண்டதாக இருக்கும். எனவே x ன் தரவு மதிப்புகள் [0,1,2,3].

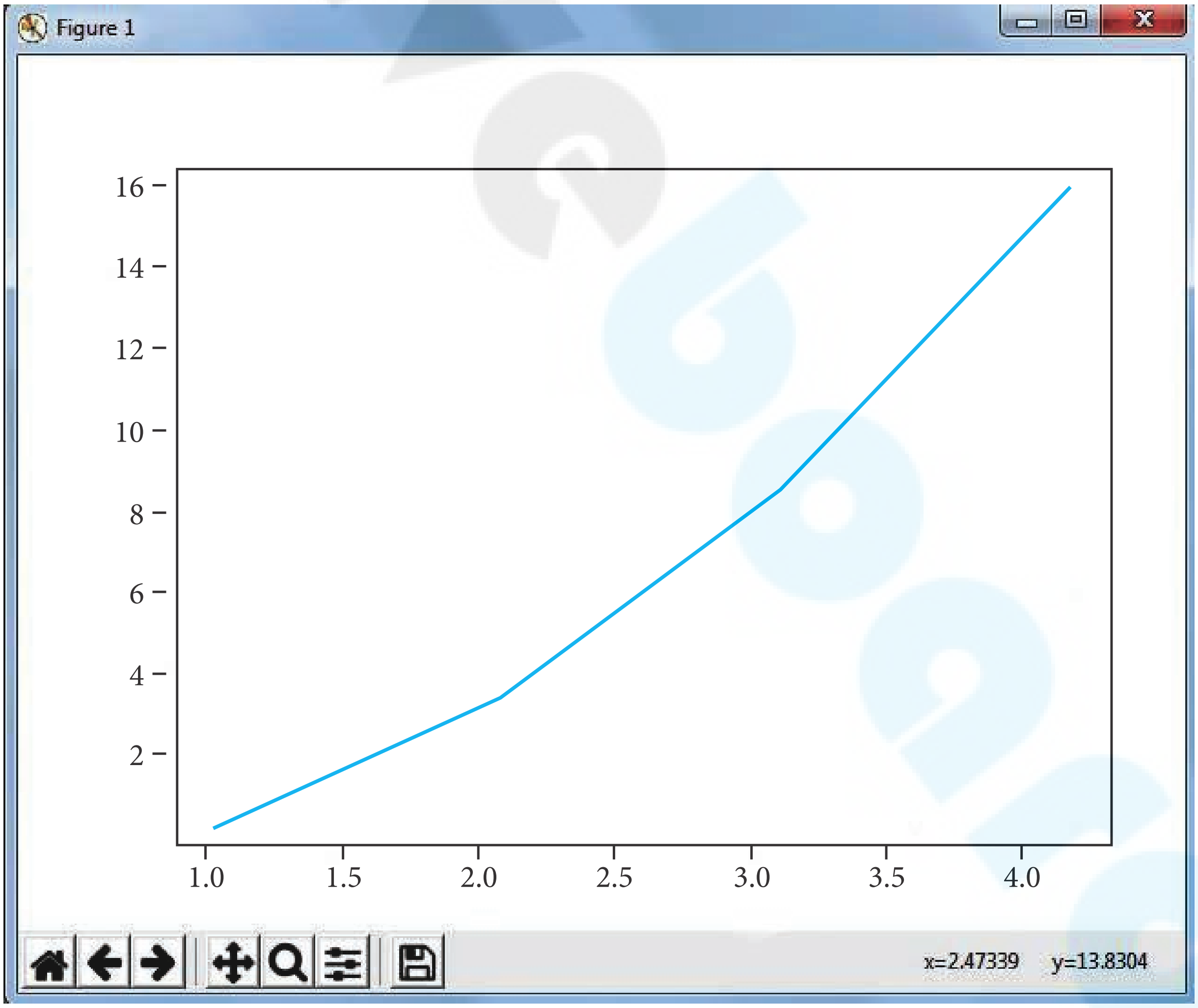
plot() என்பது ஒரு சக்திவாய்ந்த கட்டளையாகும். அது எண்ணற்ற அளபுருக்களை ஏற்கும்.

Program

எடுத்துக்காட்டாக, x versus y அச்சில் வரைய நீங்கள் கீழ்க்கண்ட கட்டளையை கொடுக்க வேண்டும்:

```
import matplotlib.pyplot as plt
plt.plot([1,2,3,4], [1,4,9,16])
plt.show()
```

கட்டளை பல அளபுருக்களை எடுத்துக் கொள்ளும். ஆனால் முதல் இரண்டு மதிப்பும் 'x' மற்றும் 'y' ஆயத்தொலைவுகளை குறிக்கும். இதன் பொருள், இந்த பட்டியல்களின் படி (1,1), (2,4), (3,9) மற்றும் (4,16) ஆகியத் தொலைவுகளைக் கொண்டிருக்கும்.



படம் 16.2

Chapter 16.indd 326
10-02-2020 20:06:50

இரண்டு கோடுகளை வரைய

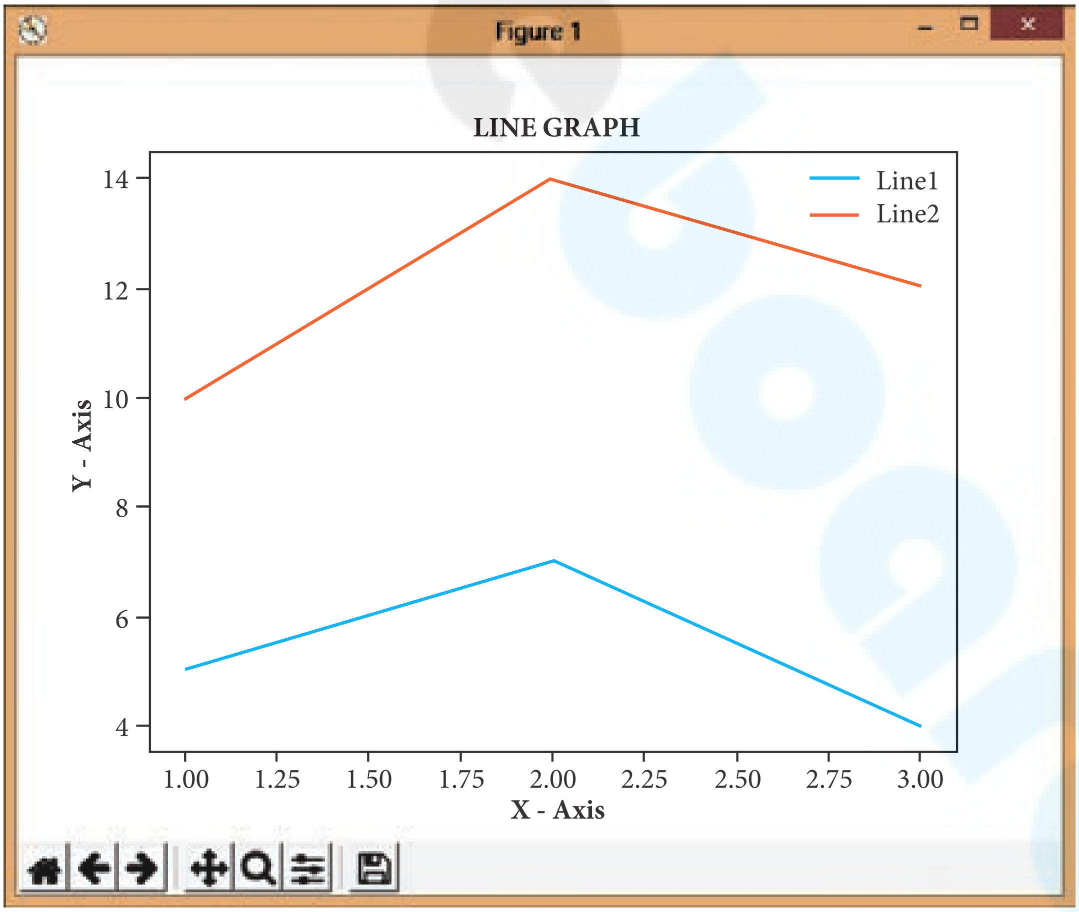
இரண்டு கோடுகளை வரைய, பின்வரும் குறியீட்டை பயன்படுத்தலாம்:

எடுத்துக்காட்டு

```
import matplotlib.pyplot as plt
        x = [1,2,3]
        y = [5,7,4]
        x2 = [1,2,3]
        y2 = [10,14,12]
        plt.plot(x, y, label='Line 1')
        plt.plot(x2, y2, label='Line 2')
        plt.xlabel('X-Axis')
        plt.ylabel('Y-Axis')
        plt.title('LINE GRAPH')
        plt.legend()
        plt.show()
```

வெளியீடு

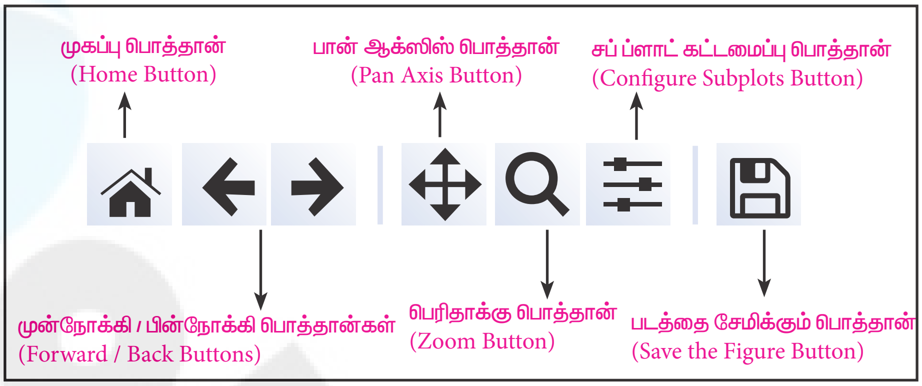
plt.xlabel மற்றும் plt.ylabel, கட்டளையைக் கொண்டு அச்சுகளுக்கு முறையே பெயரினை வழங்கலாம். plt.title கட்டளையைக் கொண்டு வரைப்படத்திற்கு தலைப்பினை வழங்கலாம், பின்னர் plt.legend() கட்டளையைக் கொண்டு கொடாநிலை புனைவுகள் (Legends) செயலாக்கலாம்.



படம் *16.3*

வெளியீடு திரையில் உள்ள பொத்தான்கள்

வெளியீட்டு திரையில், நீங்கள் கீழ் இடது மூலையில் சில பொத்தான்களைக் காணலாம். இந்த பொத்தான்களின் பயன்பாட்டை கீழே பார்க்கலாம்.



படம் *16.4*

முகப்பு பொத்தான் (Home Button)	→	வரைப்படத்தில் உலாவ தொடங்கிய உடன் இப்பொத்தான் உதவும். இப்பொத்தானை பயன்படுத்தி அசல் காட்சி திறையை எப்பொழுது வேண்டுமானாலும் பெறலாம்.
முன்னோக்கி/பின்நோக்கி (Forward/Backward buttons)	→	இப்பொத்தான்கள் உலவிகளில் காணப்படும் முன்னோக்கி/பின்நோக்கி பொத்தான்களைப் போல் பயன்படுகிறது. இப்பொத்தானைப் பயன்படுத்தி முந்தய இடத்திற்கோ அல்லது முன்னோக்கி செல்லவோ முடியும்.
பான் ஆக்ஸிஸ் பொத்தான் (Pan Axis Button)	→	குறுக்கு வடிவம் போன்ற தோற்றத்தை கொண்ட இப்பொத்தானை கிளிக் செய்து கொண்டே இழுத்து வரைபடத்தினுள் சுற்றி நகரலாம்.
பெரிதாக்கு பொத்தான் (Zoom Button)	→	இப்பொத்தான் தேர்ந்தெடுத்து, பெரிதாக்க வேண்டிய சதுரப்பரப்பினை கிளிக் செய்து நகர்த்த வேண்டும். பெரிதாக்குவதற்கு இடது கிளிக் செய்து நகர்த்தவும். மாற்றாக சிறிதாக்க விரும்பினால் வலது கிளிக் செய்து நகர்த்த வேண்டும்.
சப் ப்ளாட் கட்டமைப்பு பொத்தான் (Configure Subplots Button)	→	இப்பொத்தான், படம் மற்றும் வரைவிடத்திற்கு கிடையே உள்ள இடைவெளியை கட்டமைக்க உதவுகிறது.
படத்தை சேமிக்கும் பொத்தான் (Save Figure button)	→	இப்பொத்தான் படங்களை பல்வேறு வடிவங்களில் சேமிக்க உதவும்.

16.3 சிறப்பு வரைபடங்கள் (plot) வகைகள்

Matplotlib உங்களை ஹிஸ்டோகிராம், ஸ்கேட்டர் வரைவிடம், பட்டை விளக்கப்படம், பட்டை வரைப்படம் போன்ற பல்வேறு வகையான வரைவிடங்களை உருவாக்க உதவும்.

வரி விளக்கப்படம்

வரி வரைப்படம் அல்லது வரி விளக்கப்படம் என்பது தகவல்களை, "குறிப்பான்கள்" (Mancer) என்று அழைக்கப்படும் தரவு புள்ளிகளின் தொடரை நேர் கோட்டின் இணைப்பதன் மூலம் காட்டுகிறது. வரி

வரைப்படம் தரவுகளின் மாற்றத்தை குறிப்பிட்ட காலத்தில் நிகழக் கூடியதை காட்டும். எனவே, காலவரிசைப்படி இக்கோடுகள் அமைக்கப்படும்.

எடுத்துக்காட்டு: வரி வரைவிடம்

```
import matplotlib.pyplot as plt
        years = [2014, 2015, 2016, 2017, 2018]
        total_populations = [8939007, 8954518, 8960387, 8956741, 8943721]
        plt.plot (years, total_populations)
        plt.title ("Year vs Population in India")
        plt.xlabel ("Year")
        plt.ylabel ("Total Population")
        plt.show()
```

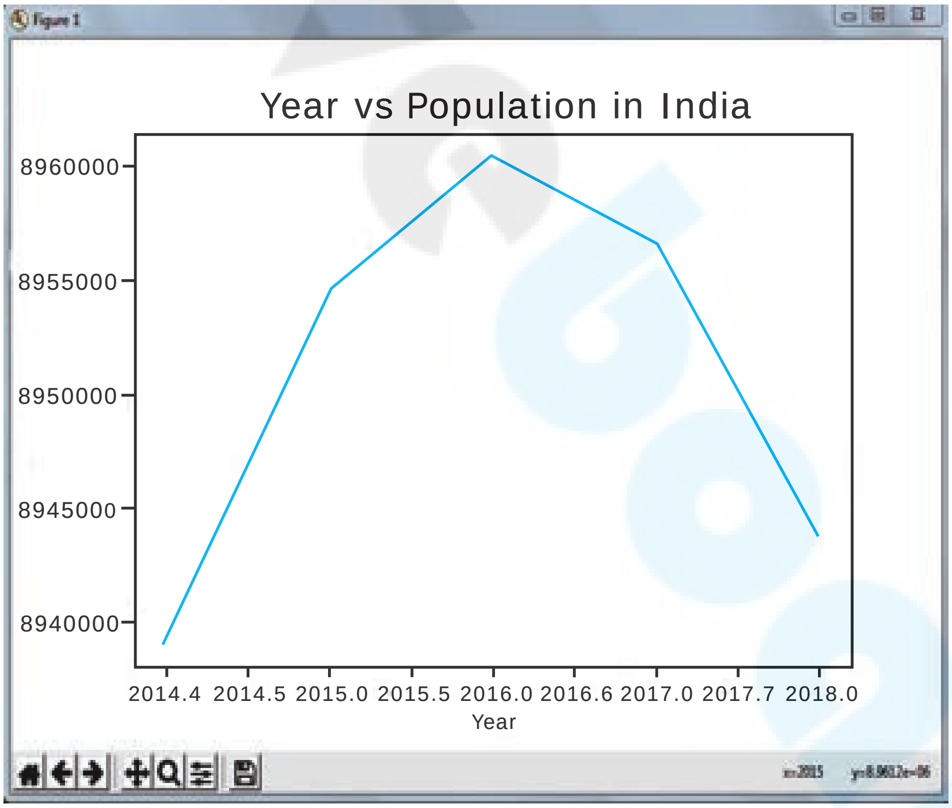
இந்த நிரலில்,

Plt.title() → வரைப்படத்தின் தலைப்பை குறிப்பிடுகிறது.

Plt.xlabel() → X-அச்சின் தலைப்பை குறிப்பிடுகிறது.

Plt.ylabel() → Y-அச்சின் தலைப்பை குறிப்பிடுகிறது.

வெளியீடு



படம் *16.5*

பட்டை வரைப்படம்

பட்டை வரைவிடம் (அல்லது பட்டை வரைப்படம்) என்பது மிகவும் பொதுவான வரைவிடம் வகையாகும். இது எண் மாறிக்கும், வகை மாறிகளுக்கும் இடையே உள்ள உறவை வெளிப்படுத்துகிறது. பட்டை விளக்கப்படம் வகைப்படுத்தப்பட்ட தரவுகளை செவ்வக பட்டையாக காட்டும். இப்பட்டைகள்

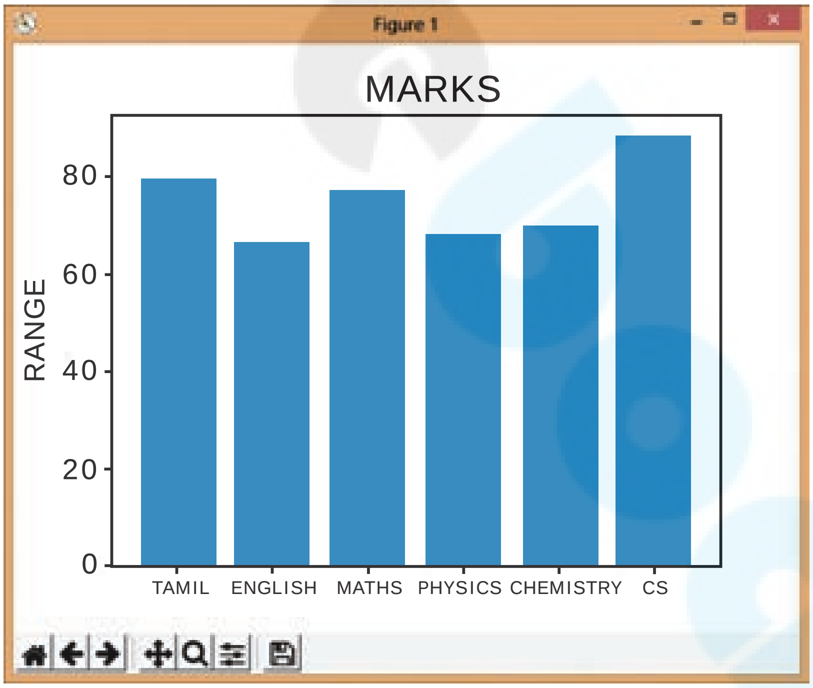
செங்குத்தாகவோ அல்லது கிடை மட்டமாகவோ தொகுக்கப்படலாம். இம்முறை கொடுக்கப்பட்ட எண் மதிப்புகளை பல்வேறு வகையுடன் ஒப்பிட்டு பார்க்க உதவும்.

Matplotlib-ல் பட்டை விளக்கப்படத்தை உருவாக்க, plt.bar() என்ற செயற்கூறினை பயன்படுத்த வேண்டும்.

எடுத்துக்காட்டு

```
import matplotlib.pyplot as plt
    # Our data
    labels = ["TAMIL", "ENGLISH", "MATHS", "PHYSICS", "CHEMISTRY", "CS"]
    usage = [79.8, 67.3, 77.8, 68.4, 70.2, 88.5]
    # Generating the y positions. Later, we'll use them to replace them with labels.
    y_positions = range (len(labels))
    # Creating our bar plot
    plt.bar (y_positions, usage)
    plt.xticks (y_positions, labels)
    plt.ylabel ("RANGE")
    plt.title ("MARKS")
    plt.show()
```

வெளியீடு



படம் *16.6*

மேலே குறிப்பிட்டுள்ள குறியீடு பின்வருவதைக் குறிக்கிறது :

Labels → பட்டையின் தலைப்பை குறிப்பிடுகிறது.

Usage → பட்டையின் பெயருக்கு மதிப்பினை ஒதுக்குகிறது

Xticks → x-அச்சில் டிக் குறியை குறிப்பிட்ட மதிப்பில் காட்டுகிறது. பின்னர் ஒவ்வொரு டிக் குறிக்கும் பெயர் குறிப்பிடப்படுகிறது.

Range → எண் தொடரை உருவாக்குகிறது.

உங்களுக்குத் தெரியுமா?

பட்டை வரைப்படம் மற்றும ஹிஸ்டோக்ராம் ஆகிய இரண்டும் தரவுகளை படவடிவில் காண்பிக்க உதவும்.

ஹிஸ்டோகிராம் மற்றும் பட்டை வரைபடங்களுக்கு இடையேயான முக்கிய வேறுபாடுகள்

ஹிஸ்டோகிராம் மற்றும் பட்டை வரைபடங்களுக்கு இடையே உள்ள வேறுபாடுகளை பின்வருமாறு குறிப்பிடலாம்:

1. ஹிஸ்டோகிராம் வரைப்படம், எண் வகை தரவுகளுக்கு இடையேயான அதிர்வெண்ணை பட்டை வடிவு வரைப்படத்தில் காட்டும். பட்டை வரைப்படம் பல்வேறு வகையான தரவுகளை ஒப்பீட பயன்படுகிறது.
2. ஹிஸ்டோக்ராம், மாறிகளின் தொடருக்கு இடையேயான அதிர்வெண் பகிர்வை காண்பிக்கும். மாறாக பட்டை வரைபடம் வெவ்வேறான மாறிகளை ஒப்பிடும் படத்தை காண்பிக்கும்.
3. ஹிஸ்டோக்ராம் எண்வகை தரவுகளிலும், பட்டை வரைப்படம் வகைப்படுத்தப்பட்ட தரவுகளின் செயலாற்றும்.
4. ஹிஸ்டோக்ராம் உள்ள பட்டைகளுக்கு இடையே இடைவெளியிருப்பதில்லை. ஆனால் பட்டை வரைப்படத்தில் தோன்றும் பட்டைகளிடையே முறையான இடைவெளிகள் மூலம் தரவுகளுக்குயிடையான தொடர்பின்மையை காண்பிக்கும்.
5. ஹிஸ்டோகிராம் எண்கள் உறுப்புகளாக அவை தரவுகளின் தொடர்புகளை வெளிப்படுத்த ஒன்றாக வகைப்படுத்தப்பட்டிருக்கும். பட்டை வரைபடத்தில் தரவுகள் தனி உறுப்புகளாக கருதப்படுகிறது.
6. பட்டை வரைப்படத்தில் தொகுதிகளை உச்சமதிப்பிலிருந்து குறைந்த மதிப்பீர்க்கு மறு வரியைக்கம் செய்ய முடியும். ஆனால் ஹிஸ்டோக்ராமில் இவை தொடர் வரிசையாக இருப்பதால் இது சாத்தியமில்லை.
7. ஹிஸடோக்ராமில் செவ்வக தொகுதியின் அகலம் ஒரே அளவில் இல்லாதிருக்கலாம். ஆனால் பட்டை வரைப்படத்தில் எப்போதும் அகலம் ஒரே மாதிரியானதாக இருக்கும்.

வட்ட வரைப்படம்

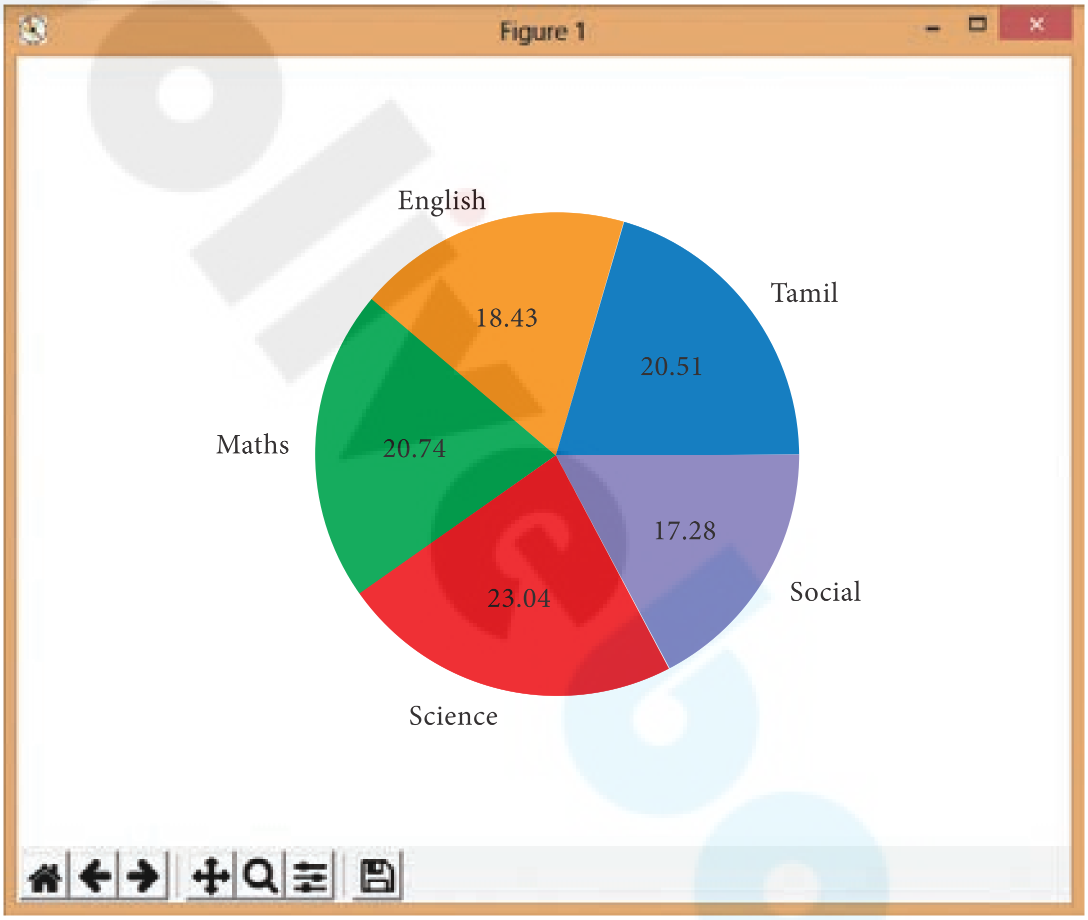
வட்ட வரைப்படம் ஒரு பொதுவான விளக்கப்பட வகையாகும். வட்ட வடிவில் தோன்றும் இந்த வரைப்படம் எண் விகிதத்தை விளக்கும் விதத்தில் துண்டுகளாக பிரிக்கப்பட்டிருக்கும். வட்ட வரைப்படத்தில் உள்ள துண்டுகள் முழு படத்துடன் உள்ள உறவை வெளிக்காட்டும்.

Matplotlib கொண்டு வட்ட வரைப்படத்தை உருவாக்க நாம் plt.pie() செயற்கூறினை பயன்படுத்த வேண்டும். autopct அளபுரு, Python சரம் வடிவமைப்பை பயன்படுத்தி மதிப்பை சதவீதத்தில் காட்டும்.

எடுத்துக்காட்டு

```
import matplotlib.pyplot as plt
        sizes = [89, 80, 90, 100, 75]
        labels = ["Tamil", "English", "Maths", "Science", "Social"]
        plt.pie (sizes, labels = labels, autopct = "%.2f")
        plt.axes().set_aspect ("equal")
        plt.show()
```

வெளியீடு



படம் *16.7*

செய்முறைப் பயிற்சி

1. தலைப்பு, x மற்றும் y அச்சுகளுக்கு பெயர்களை கொண்ட வரைவிடத்தை உருவாக்கவும்.
2. உனது சமீப தேர்வு மதிப்பெண்ணை கொண்டு வட்ட வரைப்படம் வரையவும்.
3. கடந்த 10 ஆண்டுகளில் 12ஆம் வகுப்பில் கணினி அறிவியலின் பாடத்தின் கல்வித்திறனை விளக்கும் கோட்டு வரைபடத்தை வரையவும்.

4. ஒரு வாரத்தின் உள்ள கணினி அறிவியல் பாட வேளையை விளக்கும் பட்டை வரைப்படத்தை வரையவும்.

மதிப்பீடு

பகுதி–அ

சரியான விடையை தேர்ந்தெடுத்து எழுதுக (1 மதிப்பெண்)

1. 2D வரைபடத்தை உருவாக்க பயன்படும் பைத்தான் தொகுப்பு எது?

   (அ) matplotlib.pyplot (ஆ) matplotlib.pip

   (இ) matplotlib.numpy (ஈ) matplotlib.plt

2. பைத்தான் தொகுப்பிற்கு அல்லது தொகுதிக்கு ஏற்ற தொகுப்பு மேலாண்மை மென்பொருளை தேர்ந்தெடுக்கவும்

   (அ) Matplotlib (ஆ) PIP (இ) plt.show() (ஈ) பைத்தான் தொகுப்பு

3. பின்வரும் குறியீட்டை படிக்கவும் இந்த குறியீட்டின் நோக்கத்தை கண்டறிந்து சரியான தேர்வை தேர்ந்தெடுக்கவும்.

   C:\Users\YourName\AppData\Local\Programs\Python\Python36-32\Scripts>pip –version

   (அ) PIP நிறுவப்பட்டுள்ளாத என கண்டறியும் (ஆ) PIP யை நிறுவும்

   (இ) தொகுப்பை பதிவிறக்கம் செய்யும் (ஈ) PIP பதிப்பை காண உதவும்

4. பின்வரும் குறியீட்டை படிக்கவும். இந்த குறியீட்டின் நோக்கத்தை கண்டறிந்து பின்வரும் சரியான தேர்வை தேர்ந்தெடுக்கவும்.

   C:\Users\Your Name\AppData\Local\Programs\Python\Python36-32\Scripts>pip list

   (அ) நிறுவப்பட்டுள்ள தொகுப்புகளை பட்டியலிடும்

   (ஆ) பட்டியல் கட்டளை

   (இ) PIPயை நிறுவும்

   (ஈ) நிறுவப்பட்டிருக்கும் தொகுப்புகள்

5. Matplotlib ஐ நிறுவ, கட்டளை துண்டுக்குறியில் பின்வரும் செயல்பாடு உள்ளிடப்படும் போது, “U” என்பது எதை குறிக்கிறது?

   Python –m pip install –U pip

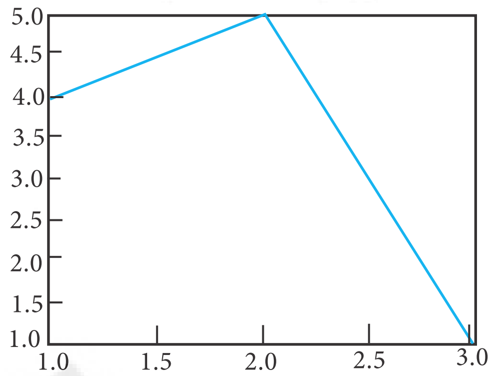
   (அ) pipயின் சமீபத்திய மதிப்பை பதிவிறக்கும்.

   (ஆ) pip யை சமீபத்திய பதிப்பிற்கு மேம்படுத்தும்

   (இ) pipயை அகற்றும்

   (ஈ) matplotlibயை சமிபத்திய பதிப்பிற்கு மேம்படுத்தும்.

6. பின்வரும் வெளியீட்டை உற்றுநோக்கு, இந்த வெளியீட்டை பெற சரியான குறியீட்டை தேர்ந்தெடுக்கவும்.



(அ) import matplotlib.pyplot as plt

plt.plot([1,2,3],[4,5,1])

plt.show()

(ஆ) import matplotlib.pyplot as plt

plt.plot([1,2],[4,5])

plt.show()

(இ) import matplotlib.pyplot as plt

plt.plot([2,3],[5,1])

plt.show()

(ஈ) import matplotlib.pyplot as plt

plt.plot([1,3],[4,1])

plt.show()

7. பின்வரும் குறியீட்டை படிக்கவும்:

(அ) import matplotlib.pyplot as plt

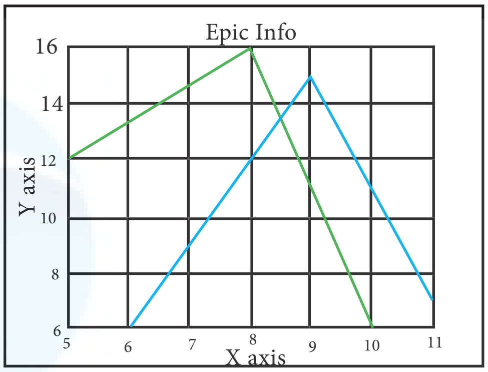
(ஆ) plt.plot(3,2)

(இ) plt.show()

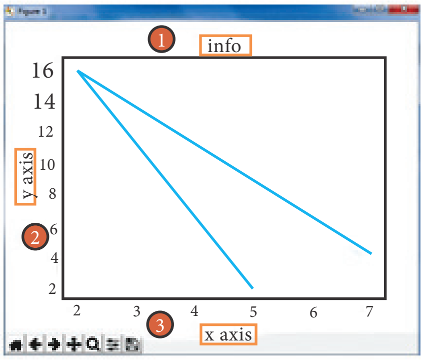
மேலே காணும் குறியீட்டின் வெளியீட்டை கண்டறியவும்.

Chapter 16.indd 334 10-02-2020 20:06:51

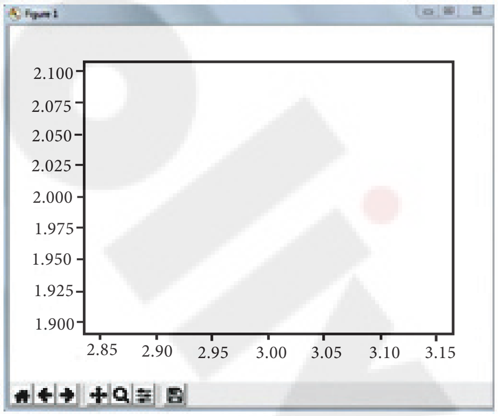
(அ)



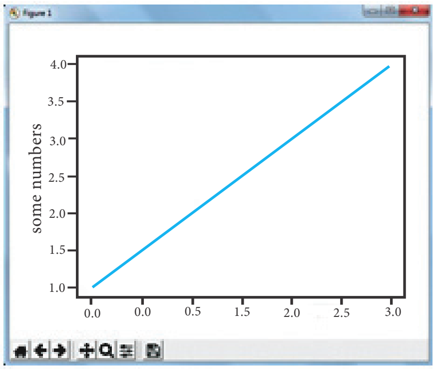
(ஆ)



(இ)



(ஈ)



8. ஒரு தொகுதியை செயல்படுத்த எந்த விசை பயன்படும்

(அ) F6 (ஆ) F4 (இ) F3 (ஈ) F5

9. பின்வரும் குறிப்புகளைப் படித்து சரியான விளக்கப்படத்தை கண்டறியவும்.

Hint 1: இந்த விளக்கப்படம் கால இடைவெளியை காட்டிலும் தரவுகளின் மாற்றத்தை காட்சிப்படுத்தும்.

Hint 2: இவ்வகை விளக்கப்படத்தில் காலவரிசைப்படி கோடுகள் இணைக்கப்பட்டிருக்கும்.

(அ) Line chart (ஆ) Bar chart (இ) Pie chart (ஈ) Scatter plot

10. பின்வரும் கூற்றை படித்து, வட்ட வரைப்படத்திற்காக சரியான தேர்வை தேர்ந்தெடுக்கவும்.

கூற்று A: plt.pie() செயற்கூற்றை பயன்படுத்தி Matplotlibல் வட்ட வரைப்படம் வரையலாம்.

கூற்று B: autopct அளபுரு பைத்தான் சரம் வடிவமைப்பை பயன்படுத்தி சதவீத மதிப்பை காட்டும்.

(அ) கூற்று A சரி (ஆ) கூற்று B சரி

(இ) இரு கூற்றும் வரி (ஈ) இரு கூற்றும் தவறு

பகுதி – ஆ

அனைத்து வினாக்களுக்கும் விடையளி (2 மதிப்பெண்)

1. தரவு காட்சிபடுத்துதல்-வரையறு.
2. தரவு காட்சிப்படுத்துதல் வகையை பட்டியலிடுக.
3. Matplotlib யுள்ள காட்சிப்படுத்துதல் வகைகளை பட்டியலிடுக.

4. Matplotlib யை எவ்வாறு நிறுவலாம்?

5. plt.plot([1,2,3,4]), plt.plot([1,2,3,4], [1,4,9,16]) ஆகிய இரு செயற்கூறுகளிடேயேயான வேறுபாட்டை எழுதுக.

பகுதி – இ

அனைத்து வினாக்களுக்கும் விடையளி (3 மதிப்பெண்)

1. பின்வரும் தரவு காட்சிப்படுத்துதல் வரைவிடத்தின் வெளியீட்டை வரையவும்.

```
import matplotlib.pyplot as plt
plt.bar([1,3,5,7,9],[5,2,7,8,2], label="Example one")
plt.bar([2,4,6,8,10],[8,6,2,5,6], label="Example two", color='g')
plt.legend()
plt.xlabel('bar number')
plt.ylabel('bar height')
plt.title('Epic Graph\nAnother Line! Whoa')
plt.show()
```

2. தரவு காட்சிப்படுத்தலின் மூன்று பயன்பாட்டை எழுதவும்.

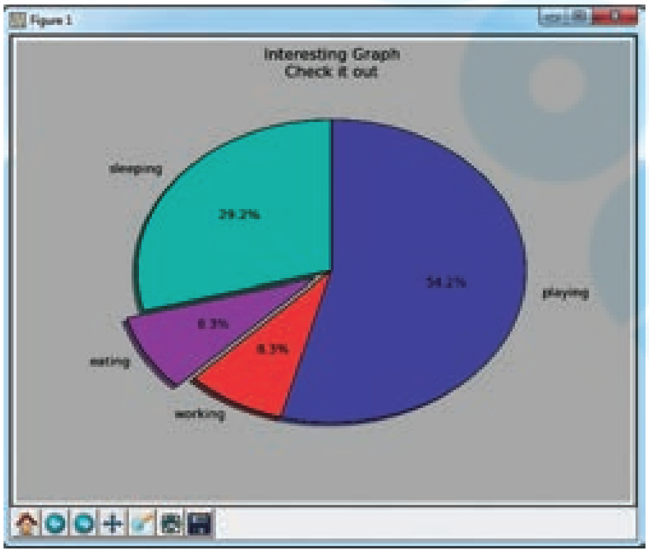
3. பின்வருவனவற்றை குறியீடை எழுதவும்:

   a. PIP உனது கணினியில் நிறுவப்பட்டிருப்பதை சோதிக்க.

   b. உனது கணினியில் நிறுவியுள்ள PIPயின் பதிப்பை அறிய

   c. Matplotlib யின் தொகுதியினை பட்டியலிட.

4. பின்வரும் வட்ட வரைப்படத்தை வெளியீடாக பெற குறியீடு எழுதவும்.



Chapter 16.indd 336 10-02-2020 20:06:52

பகுதி– ஈ

அனைத்து வினாக்களுக்கும் விடையளி (5 மதிப்பெண்)

1. Matplotlib யை பயன்படுத்தும் pyplot வகைகளை விரிவாக விவரி.
2. Matplotlib திரையில் காணப்படும் பல்வேறு பொத்தான்களை விளக்கு.
3. பின்வரும் செயற்கூறுகளின் பயன்பாட்டை எழுது:

   (அ) plt.xlabel

   (ஆ) plt.ylabel

   (இ) plt.title

   (ஈ) plt.legend()

   (உ) plt.show()

Reference

1. https://towards datascience.com / data - science - with - python - intro -to- data -visualization-and-matplotlib-5f799b7c6d82.
2. https://heartbeat.fritz.ai/introduction-to-matplotlib-data-visualization-in-python-d9143287ae39.
3. https://python programming.net / legends - titles - labels - matplotlib - tutorial/? completed=/matplotlib-intro-tutorial/.
4. https://keydifferences.com/difference-between-histogram-and-bar-graph.html.

சொற்களஞ்சியம்

கலைச்சொற்கள்		பொருள்
அளபுருக்கள்		செயற்கூறு வரையறைணுல் அறிவிக்கப்பட்டுள்ள மாறி
செயலுருபு		செயற்கூறுவிற்கு அனுப்பப்படும் மாறியின் உண்மை மதிப்பு
இடைமுகம்		ஒரு பொருள் செய்யக்கூடிய நடவடிக்கையின் தொகுப்பு இடைமுகம் ஆகும். ஆனால் அவற்றை உண்மையில் செய்வது இல்லை.
செயல்படுத்துதல்		இடைமுகத்தில் வரையறுக்கப்பட்டுள்ள கட்டளைகளை நிறைவேற்றுகிறது
Pure செயற்கூறு		ஒரே மாதிரியான அதே அளபுருக்களை அனுப்பும் போது சரியான விடையைத் தரும் செயற்கூறு *Pure* செயற்கூறு எனப்படும்.
Impure செயற்கூறு		செயற்கூறுக்கு அளபுருக்களை அனுப்பாத போதும், செயற்கூறின் உள்ளே உள்ள மாறியானது பக்க விளைவுகளை ஏற்படுத்தும். இந்த வகைச் செயற்கூறு *impure* செயற்கூறு எனப்படும்.
கண்ஸ்டரக்டர்		அருவமாக்கம் தரவு வகையை உருவாக்கும் செயற்கூறு.
செலக்டார்ஸ்		தரவு வகையிலிருந்து தகவலை மீட்டெடுக்கும் செயற்கூறு
லிஸ்ட்		வரிசைப்படுத்தப்பட்ட தொடர் உருப்புகளைக் கொண்ட மற்றும் தன்மையுடைய தரவு கட்டமைப்பாகும்.
Tuple		இது மாற்றவியலாத பொருள்களின் வரிசை. இவை பட்டியல்களை போன்றவை. *( )* அடைப்புக்குறிகளுக்குள் அதன் மதிப்புகள் வரையறுப்பப்படும்.
ஸ்டாக்		ஸ்டாக் *(சில* நேரங்களில், புஸ்டவுன் ஸ்டாக் என்று அழைக்கப்படும்*)* என்பது வரிசைப்படுத்தப்பட்ட உறுப்புகளின் சேகரிப்பு ஆகும், இதில் புதிய உறுப்பு சேர்த்தும் மற்றும் ஏற்கனவே இருக்கும். உறுப்புகளை நீக்குதலும் ஒரே சமயத்தில் நடைபெறும். இந்த முனை டாப் என்று பொதுவாக குறிப்பிடப்படுகிறது. டாப் முனையின் எதிர் முடிவு பேஸ் என்று அறியப்படுகிறது. இந்த வரிசைமுறை கொள்கை *LIFO, last-in first-out* என்று அழைக்கப்படுகிறது.
கியூ		கியூ என்பது ஸ்டாக் போன்ற ஒரு அப்ஸ்டராக்ட் தரவு கட்டமைப்பாகும். ஸ்டாக் போல் அல்லது கியூவின் இரு முனைகளிலும் திறந்திருக்கும். ஒரு முனை எப்பொழுதும்
CWI		*Centrum Wiskunde and Informatics*
IDLE		*Integrated Development Learning Environment*
தூண்டுக்குறி		<<< என்ற குறியுரு நிரல் பெயர்ப்பியில் உள்ளீட்டை ஏற்றுக் கொள்ள தயார் என்பதை பயனருக்கு காட்டுகிறது.

Glossary.indd 338 30-01-2020 21:02:22

ஸ்கிரிப்ட்		கோப்பில் சேமிக்கப்பட்ட பைத்தான் நிரல்
ஊடாடும் முறைமை		தூண்டு குறியில் கட்டளை மற்றும் கோவைகளை உள்ளீடுவதன் மூலம் பைத்தான் மொழிப்பெயர்ப்பியை பயன்படுத்தும் முறை.
தொடரியல்		நிரலின் கட்டளை அமைப்பு
தொடரியல் பிழை		ஒரு நிரலில் பிழை இருப்பின் அதை பைத்தான் நிரல் பெயர்ப்பி புரிந்து கொள்ளாது.
வில்லைகள்		நிரலின் கட்டளை அமைப்பின் அடிப்படை கூறுகளாகும்.
வரிசை		ஒன்றன் பின் ஒன்றாக
மாற்று		இரண்டு தேர்வுகளில் ஏதேனும் ஒன்று
மடக்குச் செயல்		திரும்பச் செய்தல்
தொகுதி		கூற்றுகளின் தொகுப்பு
மாறி		மதிப்புகளைச் சேமிக்கும் நினைவகப்பெட்டி
பைத்தான் தொடங்கும் குறி		>>>
சரி		தருக்க மதிப்பு *1*
தவறு		தருக்க மதிப்பு *0*
elif		*else…if*
பூலியன்		தருக்கம்
range()		தொடக்க மதிப்பிலிருந்து மேல்வரம்பு *1* வரை நிறை வேற்றப்படும்.
break		கட்டுப்பாட்டை நிறுத்தும்.
continue		மீதமுள்ள பகுதியைத் தவிர்த்து, அடுத்த மடக்கைத் தொடங்குதல்
pass		செயற்கூறு மற்றும் மடக்குகளில் இட ஒதுக்கீட்டிற்காக பயன்படுகிறது.
def		செயற்கூறை வரையறுப்பதற்கான சிறப்புச் சொல்
return		*return* கூற்று செயற்கூறினை விட்டு வெளியேறி அழைப்புக்கூற்றுக்கு மதிப்பைத் திருப்பி அனுப்பும்.
பின்னலான தொகுதி		தொகுதியினுள்ளே உள்ள மற்றொரு தொகுதி
sqrt()		உள்ளிணைந்த கணித செயற்கூறான இது, நேர்ம எண்ணின் வர்க்க மூலத்தைத் திருப்பி அனுப்புகிறது
id()		கொடுக்கப்பட்ட பொருளின் நினைவக முகவரியைத் திருப்பி அனுப்பும்.
eval()		சரத்தின் மதிப்பை மதிப்பீடு செய்கிறது.
முழுதளாவிய வரையெல்லை		நிரலில் எங்கு வேண்டுமானாலும் பயன்படுத்தக்கூடிய மாறி
உள்ளமை வரையெல்லை		செயற்கூறு உடற்பகுதியில் அல்லது தொகுதியின் உள்ளே அறிவிக்கப்படும் மாறி
லாம்டா		சிறிய மற்றும் ஒரு தடவை பெயரில்லாமல் உருவாக்கி பயன்படுத்தப்படும் செயற்கூறு
தற்சுழற்சி செயற்கூறு		தன்னைத் தானே அழைத்துக் கொள்ளும் செயற்கூறு

Glossary.indd 339 30-01-2020 21:02:22

List	மாற்றம் செய்யக்கூடிய வரிசையான மதிப்புகளின் தொகுப்பாகும்.
Tuples	மாற்றம் செய்ய முடியாத வரிசையான மதிப்புகளின் தொகுப்பாகும்.
Dictionary	திறவுகோல் மற்றும் மதிப்பு இணைகளின் தொகுப்பு
Key	***Dictionary***-ல் உள்ள மதிப்புகளுக்கு படமிடப்பட்ட தரவு.
பொருள் நோக்கு நிரலாக்கம்	நிகழ் உலக பொருள்களைப் பொறுத்த கணினி நிரல் கோட்பாடுகள்
இனக்குழு	பொருள்களின் வார்ப்புரு
பொருள்	தரவு மற்றும் செயற்கூறுகளின் தொகுப்பு
பண்புக்கூறு	பொருளை உருவாக்கும் தரவு உறுப்புகள்
இனக்குழு மாறி	இனக்குழுவின் உள்ளேஅறிவிக்கப்படும் சாதாரண மாறி
வழிமுறை	இனக்குழுவிற்குள்ளே அறிவித்து வரையறுக்கப்படும் செயற்கூறு.
சான்றுருவாக்கல்	ஒரு பொருளை உருவாக்கும் செயல்முறை
ஆக்கி	பொருள் பயன்பாட்டுக்கு வரும் பொழுது தானாகவே இயக்கப்படும் சிறப்புச் செயற்கூறு.
அழிப்பி	பொருளின் பயன்பாடு முடிவுக்கு வரும் பொழுது தானாகவே இயக்கப்படும் சிறப்புச் செயற்கூறு
DBMS	தரவுதள மேலாண்மை அமைப்பு *(Database Management System)*
RDBMS	உறவுநிலைத் தரவுதள மேலாண்மை அமைப்பு *(Relational Database Management System)*
GIS	புவியியல் தகவல் அமைப்பு *(Geographic Information System)*
DBA	தரவுதள நிர்வாகி *(DataBase Adminstrator)*
உறவுநிலை தரவுத்தளத்தின் தந்தை	*Dr. Edgar Frank Codd*
Select (σ)	இது ஒரு நிபந்தனையின் அடிப்படையில் துணைத் தொகுதியை ***tuples***களுடன் தேர்ந்தெடுக்க உதவுகிறது.
Projection (π)	இது குறிப்பிடப்பட்டுள்ள உள்ளீடு தொடர்புகளின் பண்புக்கூறுகளை நீக்குகிறது.
ஒட்டுதல்(∪)	இது அட்டவணை ***A*** அல்லது ***B*** ல் உள்ள அனைத்து ***tuples*** களையும் உள்ளடக்கியது.
வேறுபாடு *(-)*	இது ***A*** மற்றும் ***B*** அட்டவணைகளை ஒப்பிட்டு உள்ள வேறுபட்ட ***tuples*** களை மட்டும் தருகிறது.
வெட்டுதல் *(∩)*	இது ***A*** மற்றும் ***B*** அட்டவணைகளில் பொதுவாக உள்ள அனைத்து ***tuples*** களையும் உள்ளடக்கியது.
கார்டீசியன் பெருக்கல் *(X)*	இரண்டு தொடர்புகளைச் சேர்க்க இது வழிவகுக்கிறது.
CRUD	*Create, Read, Update and Delete*
ஒருமைப்பாடு *(Integrity)*	முழுமையான மற்றும் பிரிக்கப்படாத

Glossary.indd 340 30-01-2020 21:02:22

பதிக்கப்பட்டது *(Embedded)*		உறுதியாக இணைக்கப்பட்டுள்ளது
அங்கீகாரம் *(Authorization)*		அணுகலுக்கு அனுமதி கொடுப்பது
அமைப்பு முறை *(Schema)*		அமைப்பு அல்லது வடிவம்
நிபந்தனை *(Constraint)*		கட்டுப்பாடு அல்லது வரம்பு
தரவு இரட்டித்தல் *(Redundant)*		மிகையான தரவு
இணைப்பு *(Conjunction)*		உடன்பாடு, ஒன்றி பொருந்துதல்
Enter வரிசை		***Enter*** வரிசை அல்லது ***newline*** புதிய வரிகளை ***CSV*** கோப்பில் உருவாக்குவதற்கு பயன்படும்.
csv.reader()		***reader*** செயற்கூறு கோப்பில் உள்ள ஒவ்வொரு வரியையும் எடுத்து அவற்றை நெடுவரிசைகளின் பட்டியலாக மாற்றும்.
csv.register_dialect()		***CSV*** கோப்பில் படிப்பதற்கான வடிவத்தை dialect விவரிக்கின்றது.
append()		***List*** இறுதியில் ஒரு உறுப்பை சேர்க்க பயன்படும்
next()		***next()*** செயற்கூறு சுழற்சியில் உள்ள அடுத்த உறுப்பினை திருப்பி அனுப்பும். மேலும் ***CSV*** கோப்பில் ஒரு வரிசையைத் தவிர்க்கவும் பயன்படும்.
sort()		***sort*** செயற்கூறு ***list***யை ஏறுமுகமாக, இறங்குமுகமாக அல்லது பயனர் வரையறுத்த வரிசையில் வரிசையாக்கம் செய்ய பயன்படும்.
operator.itemgetter(col_no)		***CSV*** கோப்பில் ஒன்றிக்கு மேற்பட்ட நெடுவரிசையில் வரிசையாக்கம் செய்யும்.
DicReader()		***CSV*** கோப்பில் முதல் வரியை படிக்கவும் மற்றும் அந்த வரியில் காற்புள்ளியால் பிரிக்கப்பட்டுள்ள மதிப்புகளை ***dictionary***-யின் திறவுகோலாக பயன்படுத்தும்.
dict()		***Orderdict*** இல்லாமல் ***dictionary***-யில் உள்ள தரவை வெளியிட பயன்படும்.
Writerow()		கோப்பில் ஒரு வரிசையாகவுள்ள தரவை எழுதும் செயற்கூறு
writerows()		பல்வேறு வரிசைகளிலான தரவை கோப்பில் எழுதும் செயற்கூறு
skipinitialspace=true		***True*** என்ற மதிப்பெணில் பிரிப்பானிற்கு பிறகுயுள்ள வெற்று இடைவெளியை தவிர்க்கும். இதன் கொடாநிலை மதிப்பு ***false*** ஆகும்.
Csv.Quote_All		***csv.quote_all*** பிரிப்பான் அமைக்கப்பட்டிருந்தால், ***writerow()*** அனைத்து புலங்களையும் பிரிப்பானுடன் எழுதும்.
DictWriter()		***Dictionary***-ன் தரவை ***CSV*** கோப்பில் எழுதும்.
Parse		உள்ளீட்டு தரவை சிறு துணுக்குகளாக பிரிப்பதால் அவற்றை எளிமையாக சேமிக்க அல்லது கையாள முடியும்.
ஸ்டைரல் ***(stride)***		நீளமான படிநிலை
சரநிலையுறு ***(string)***		எழுத்துக்கள், எண்கள் அல்லது குறயீடுகளின் வரிசை
கீழெட்டு ***(subscript)***		சுட்டெண்

Glossary.indd 341 30-01-2020 21:02:22

இணைப்பு - 1

பைத்தானில் உள்ள செயற்கூறுகள்

உள்ளிணைந்த செயற்கூறுகள்

செயற்கூறு		பயன்பாடு
abs()		ஒரு எண்ணின் துல்லியமான மதிப்பை திரும்பதரும்
all()		அனைத்து உறுப்புகளின் மதிப்பும் உண்மை எனில் உண்மை என்ற மதிப்பை திரும்ப தரும்
any()		ஏதேனும் உறுப்பின் மதிப்பு உண்மையா என சோதிக்கும்.
ascii()		அச்சிடக்கூடிய குறிப்புகளை கொண்ட சரத்தை திரும்ப தரும்.
bin()		முழு எண் மதிப்பை இருநிலை எண்களாக மாற்றும்.
bool()		ஒரு மதிப்பை பூலியன் மதிப்பாக மாற்றித் தரும்.
bytearray()		கொடுக்கப்பட்ட பைட்டு அளவில் அணியை திரும்ப தரும்.
bytes()		மாற்ற முடியாத பைட்டுகளை திரும்ப தரும்.
callable()		பொருள்கள் அழைக்கத் தக்கவையா என சரிபாக்கும்.
chr()		முழு எண் மதிப்பிற்குரிய குறியுருவை திரும்ப அனுப்பும்.
classmethod()		கொடுக்கப்பட்ட செயற்கூறின் இனக்குழு முறையை திரும்ப தரும்.
compile()		பைத்தான் குறிமுறை பொருளை திருப்பும்
complex()		கலப்பு எண்ணை உருவாக்குகிறது.
delattr()		பொருளின் பண்புக்கூறுகளை நீக்குகிறது
dir()		பொருளின் பண்புக்கூறுகளை திரும்ப அனுப்ப முயற்சிக்கும்.
dlvmod()		ஈவு மற்றும் மீதிகளின் தொகுப்புகளை திரும்பத் தரும்.
enumerate()		எண்ணுரு *(enumerate)* பொருளை திருப்பும்
eval()		நிரலினுள் பைத்தான் குறிமுறையை இயங்கச் செய்யும்.
exec()		உடனடியாக உருவாக்கப்படும் நிரலை இயங்கச் செய்யும்.
filter()		எந்த உறுப்புகள் உண்மை என்ற மதிப்பை பெறுகிறதோ அதை குழுவாக்கும்.
float()		கொடுக்கப்பட்ட எண் சரத்திலிருந்து தசம எண் மதிப்பை திரும்ப தரும்.
format()		ஒரு மதிப்பின் வடிவூட்டப்பட்ட முறையை திரும்ப தரும்.
getattr()		ஒரு பொருளின் குறிப்பிட்ட பண்புக்கூறின் மதிப்பை திரும்ப தரும்.
globals()		நடப்பு குளோபல் குறியீட்டு அட்டவணையின் நூலகத்தை திருப்பும்
hasattr()		பொருளானது அளபுருக்களை பெற்றிருந்தால் மதிப்பை திரும்ப தரும்.
hash()		பொருளின் *hash* மதிப்புகளை திரும்ப தரும்.
help()		உள்ளிணைந்த உதவி அமைப்பை அழைக்கும்.
hex()		முழு எண் மதிப்பை பதினறும மதிப்பாக மாற்றி திரும்பத்தரும்
id()		பொருளின் அடையாளத்தை திரும்ப தரும்.

Annexure I.indd 342 30-01-2020 17:46:08

isinstance()		பொருளானது இனக்குழுவின் சான்றுருவா என சோதிக்கும்.
issubclass()		பொருளானது இனக்குழுவின் துணைக்குழுவா என சோதிக்கும்
iter()		ஒரு பொருளானது பலமுறை செயலாற்றும் எண்ணிக்கையை திரும்பத் தரும்.
len()		பொருளின் நீளத்தை திரும்பத் தரும்.
locals()		அகராதியில் உள்ள குறியீடுகளின் அட்டவணைத் திரும்பத் தரும்.
map()		செயல்பாடுகளை செயல்படுத்தி ஒரு பட்டியலை திரும்பத் தரும்.
max()		பெரிய மதிப்பினைக் கொண்ட உறுப்பை திரும்பத் தரும்.
memory view()		அளபுருவின் நினைவக அமைவிடத்தை திரும்பத் தரும்.
min()		சிறிய மதிப்பினைக் கொண்ட உறுப்பினை வெளிப்படுத்தும்.
next()		சுழற்சியின் அடுத்த உறுப்பினை திரும்ப தரும்.
object()		பயன்பாடில்லா பொருளினை உருவாக்கும்.
oct()		முழு எண் மதிப்பை எண்ம எண் மதிப்பாக திரும்பத் தரும்.
open()		கோப்பின் பொருளை திரும்பத் தரும்.
ord()		ஒருங்கமை குறியுருக்களுக்கு உண்டான ஒருங்கமை மதிப்பை திரும்பத் தரும்.
pow()		y -ன் அடுக்கு x -ன் மதிப்பை திருப்பும்
print()		கொடுக்கப்பட்ட பொருளினை அச்சிடும் .
property()		அளபுருவின் பண்புக்கூறை திரும்பத் தரும்
range()		தொடக்க மற்றும் முடிவு மதிப்புகளுக்கிடையேயான வரிசையான முழு எண் மதிப்புகளை திரும்பத் தரும்.
repr()		பொருளை அச்சிடும்.
reversed()		தலைகீழ் சுழற்சியின் திரும்பத்தரும்.
round()		மிதப்புப்புள்ளி எண்ணை n இலக்க முழு எண்ணாக திரும்பத் தரும்.
set()		பைத்தான் *Set*-ஐ திரும்ப தரும்.
setattr()		பொருளின் பண்புக்கூறுகளுக்கு மதிப்பினை தர பயன்படுகிறது.
slice()		குறிப்பிட்ட வரம்பினில் ஒரு துணை பொருளை உருவாக்கும்.
sorted()		குறிப்பிட்ட சுழற்சியின் வரிசையாக்கப்பட்ட பட்டியலை திரும்ப தரும்.
staticmethod()		செயற்கூறில் இருந்து Static முறையை உருவாக்கும்.
str()		பொருளின் முறையற்ற குறிப்பை திரும்பத்தரும்.
sum()		சுழற்சியில் உருப்படியை சேர்க்க பயன்படுகிறது.
super()		மீ இனக்குழுவானது பெற்றோர் இனக்குழுவை குறிப்பிட அனுமதிக்கிறது.
type()		பொருளின் வகையை திரும்ப தரும்.
vars()		இனக்குழுவின் dict-attributes களை திரும்பத் தரும்.
import()		மேம்பட்ட செயல்கூறானது import() மூலம் அழைக்கப்படும்.
maketrans()		மொழிப்பெயர்ப்பு அட்டவணையை திரும்பத் தரும்.
partition()		Tuple-ஐ திரும்பத் தரும்.

Annexure I.indd 343 30-01-2020 17:46:08

replace()		துணைச் சரத்தை பதிலீடு செய்யும்.
rfind()		துணைச்சரத்தின் பெரிய சுட்டு எண்ணை திரும்ப தரும்.
rindex()		துணை சரத்தின் உயர் சுட்டெண்ணை திருப்பும்
rjust()		கொடுக்கப்பட்ட அகலத்தில் சரத்தினை வலது இசைவில் வெளிப்படுத்தும்.
rsplit()		இடமிருந்து சரத்தினை பிரிக்கும்.
splitlines()		வரியின் இறுதியுடன் சரத்தினை பிரிக்க உதவும்.
startswith()		ஒரு சரமானது குறிப்பிட்ட சரத்துடன் துவங்குகிறதா சோதிக்கும்.
strip()		முன்னொட்டு *(Leading)* மற்றும் பின்னொட்டு *(Trailing)* குறியுருக்களை நீக்கும்
		சர (STRING) செயற்கூறுகள்
capitalize()		சரத்தின் முதல் குறியுருவை மேல் எழுத்தாக மாற்றும்.
casefold()		சரத்தின் அனைத்து எழுத்துரு நிலை (பெரிய எழுத்து அல்லது சிறிய எழுத்து) யையும் நீக்கும்
center()		குறிப்பிடப்படும் எழுத்துருவை கொண்டு ஒரு சரம் நிரப்பப்படும்
count()		கொடுக்கப்பட்ட சரத்தில் ஒரு துணைச் சரமானது தோன்றும் எண்ணிக்கையை குறிக்கும்.
encode()		கொடுக்கப்பட்ட சரத்தின் மறை குறியாக்கம் செய்யப்பட்ட சரத்தை திரும்ப தரும்.
endswitch()		சரமானது குறிப்பிட்ட பின்னொட்டுடன்/ குறியுருவுடன் முடிவடைகிறதா என்பதை சோதிக்கும்.
expandtabs()		Tab குறியுருக்களை வெற்றிடக்குறியுருவைக் கொண்டு மாற்றியிடும்.
find()		துணைச்சரமானது முதலில் தோன்றும் இடத்தின் சுட்டு மதிப்பை திரும்ப தரும்.
format()		சரத்தை பார்ப்பதற்கு இனிமையான தோற்றத்தில் தோன்ற செய்யும்
format_map()		அகராதியைக் கொண்டு சரத்தை வடிவூட்டும்.
index()		துணைச் சரத்தின் சுட்டு எண் மதிப்பை திரும்ப தரும்.
input()		ஒரு வரி சரத்தினை படித்து திரும்ப தரும்.
int()		ஒரு எண் அல்லது சரத்திலிருந்து முழு எண் மதிப்பை திரும்ப தரும்.
isalnum()		Alphanumeric குறியுரு உள்ளதா என சோதிக்கும்.
isalpha()		அனைத்து குறியுருக்கும் எழுத்துக்களா என்பதை சோதிக்கும்.
isdecimal()		பதின்ம எண் குறியுருக்களை சோதிக்கும்.
isdigit()		எண் குறியுருக்களை சோதிக்கும்.
isidentifier()		தகுதியான குறிப்பெயரா என சோதிக்கும்.
islower()		சரத்தினுள் உள்ள அனைத்து குறியுருக்களும் கீழ்நிலை எழுத்துக்களா என சோதிக்கும்.
isnumeric()		எண் வகை குறியுருக்களை சோதிக்கும்.
isprintable()		அச்சிடும் குறியுருக்களை சோதிக்கும்.
isspace()		பதின்ம எண் குறியுருக்களை சோதிக்கும்.
istitle()		எண் குறியுருக்களை சோதிக்கும்.

Annexure I.indd 344 30-01-2020 17:46:08

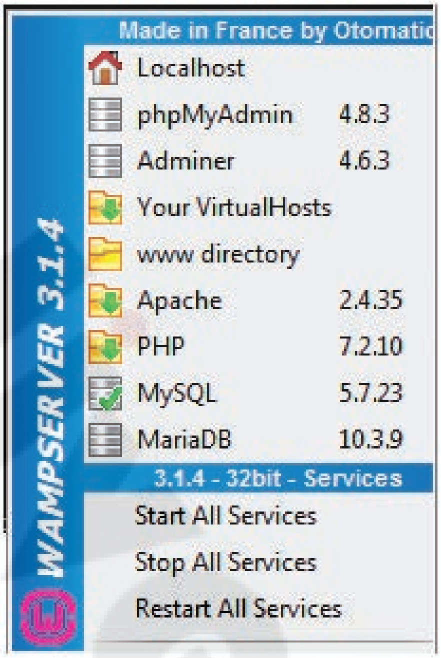
isupper()		அனைத்து குறியுருக்களும் பெரிய எழுத்துக்கள் எனில் உண்மை என்ற மதிப்பை தரும்.
join()		இணைக்கப்பட்ட சரத்தை திரும்பத் தரும்.
ljust()		கொடுக்கப்பட்ட அகலத்தில் சரத்தினை இடது இசைவினில் வெளிப்படுத்தும்.
lower()		கீழ்நிலை எழுத்துக்களை கொண்ட சரங்களை வெளிப்படுத்தும்.
lstrip()		முன்னெழுத்துக்களை திரும்பத் தரும்.
swapcase()		மேல்நிலை எழுத்துக்களை கீழ்நிலை எழுத்துக்களாகவும் கீழ்நிலை எழுத்துக்களை மேல்நிலை எழுத்துக்களாகவும் மாற்றும்.
title()		சரத்தை தலைப்பெழுத்துகளில் திருப்பும்
translate()		குறிக்கப்பட்ட எழுத்துருக்களின் சரமாக திருப்பும்
upper()		மேல் எழுத்துக்களை கொண்ட சரத்தை திரும்ப தரும்.
zfill()		சுழியங்களுடன் நிரப்பப்பட்ட சரத்தின் நகலை திருப்பும்
LIST செயற்கூறுகள்		
append()		*List*-ல் ஒரு உறுப்பைச் சேர்க்கும்.
clear()		*List*-ல் இருந்து அனைத்து உருப்படிகளையும் நீக்கும்.
copy()		*List*-ல் நிழல் மதிப்புகளைத் திருப்பும்.
count()		*List*-ல் உள்ள உறுப்புகளின் நிகழ்வு எண்ணிக்கையை திருப்பும்.
extend()		ஒரு *List*-ன் உறுப்பை மற்றொரு *List*-ல் சேர்க்கும்.
index()		*List*-ல் உள்ள உறுப்புகளின் சிறிய சுட்டெண்ணை திருப்பும்.
insert()		*List*-ல் உறுப்புகளைச் செருகும்.
list()		பைத்தானில் *List*-ஐ உருவாக்கும்.
pop()		கொடுக்கப்பட்ட சுட்டெண்ணில் உள்ள உறுப்பை நீக்கும்.
remove()		*List*-ல் இருந்து உறுப்பை நீக்கும்.
reverse()		*List*-ஐ தலைகீழாக மாற்றும்.
slice()		*range()*ல் குறிப்பிட்டுள்ளவாறு *slice* பொருளை உருவாக்கும்.
sort()		*List*-ல் உள்ள உறுப்புகளை வரிசையாக்கம் செய்யும்.
TUPLE செயற்கூறுகள்		
count()		*Tuple*ல் உள்ள உறுப்புகளின் எண்ணிக்கையை திருப்பும்.
index()		*Tuple*-ல் உள்ள உறுப்புகளின் மிகச் சிறிய சுட்டெண்ணை திருப்பும்.
slice()		*Range()*ல் குறிப்பிட்டுள்ளவாறு *slice* பொருளை உருவாக்கும்.
tuple()		*Tuple*ஐ உருவாக்கும்.
zip()		*tuple*ன் சுழற்சி திருப்பும்

Annexure I.indd 345 30-01-2020 17:46:08

இணைப்பு - 2

12.4 MySQL ஐ நிறுவிக் கொண்டு, ஒரு தரவுத்தளத்தை உருவாக்குதல்:

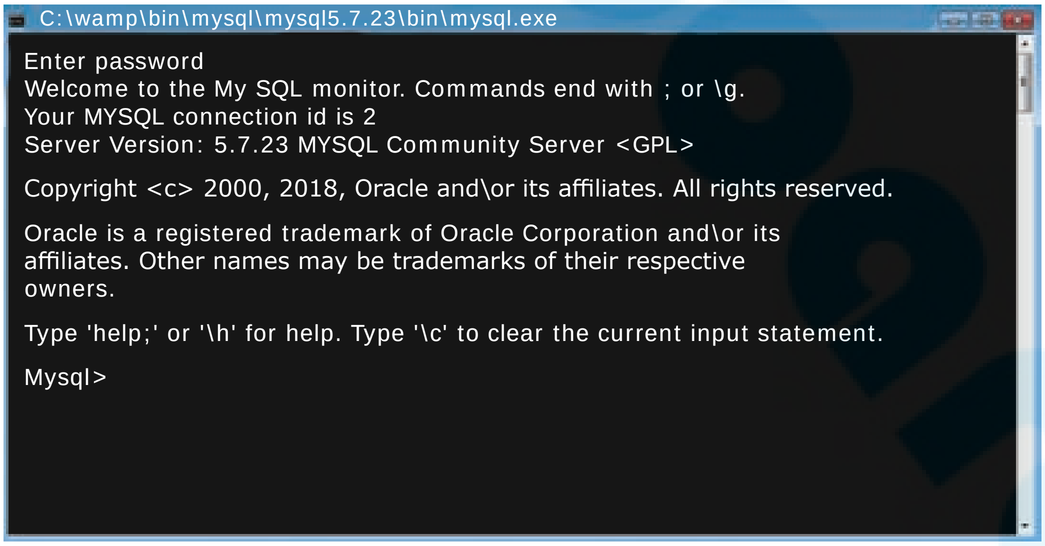
- கணினி கட்டமைப்பின் அடிப்படையில் WAMPஐ பதிவிறக்கம் செய்து, அதனை உங்கள் கணினியில் நிறுவவும்.



MW8QPZ

படம் 12.1

- MySQL முனையத்தில் மூல பயனராக (root user) உள்புகுக.
- Enter பொத்தானை அழுத்தவும்.(கடவுச்சொல்லை உள்ளிட வேண்டிய தேவையில்லை)
- SQL கட்டளைகளை உள்ளிடுவதற்கான MySQL தூண்டுகுறி தோன்றும்.



படம் 12.2

இணைப்பு 3

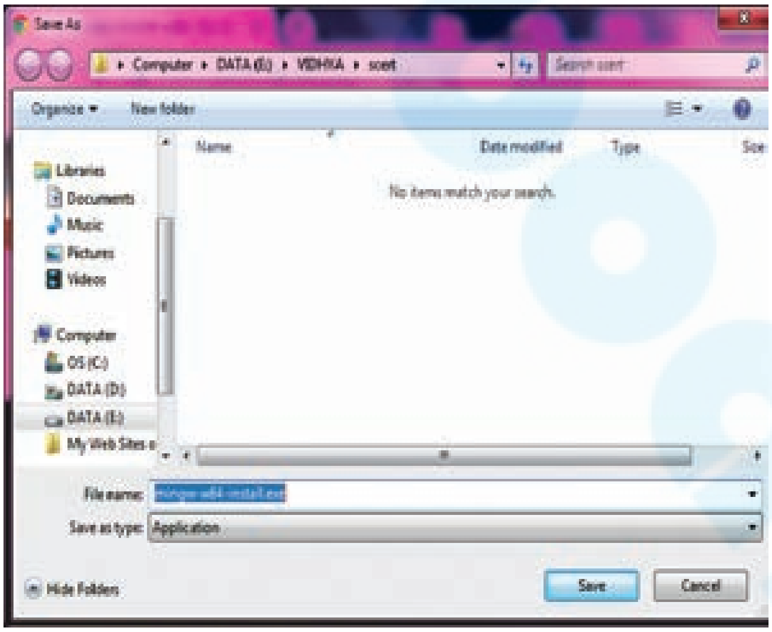
14.5.2 32 மற்றும் 64 பிட் விண்டோஸில் MinGW-w64ஐ நிறுவுவதற்கான படிநிலைகள்

படிநிலை1: ஏதேனும் ஒரு தேடுபொறியில் (https://sourceforge.net/projects/mingw-w64) என தட்டச்சு செய்து, இரட்டை கிளிக் செய்யவும்.

Y412B8

படம் -2

படிநிலை 2 : முகப்புப் பக்கத்தில் தோன்றும் **Download** என்னும் பதிவிறக்கப் பொத்தானை கிளிக் செய்யவும். உங்கள் கொடாநிலை பதிவிறக்க கோப்புறையில் கோப்பு பதிவிறக்கமாகும் அல்லது **save as** உரையாடல் பெட்டி தோன்றும்.



படம் 3

படிநிலை 3 : கோப்பு mingw-get-setup.exe . என தோன்றும். SourceForge வலை தளத்தை பார்வையிடும்

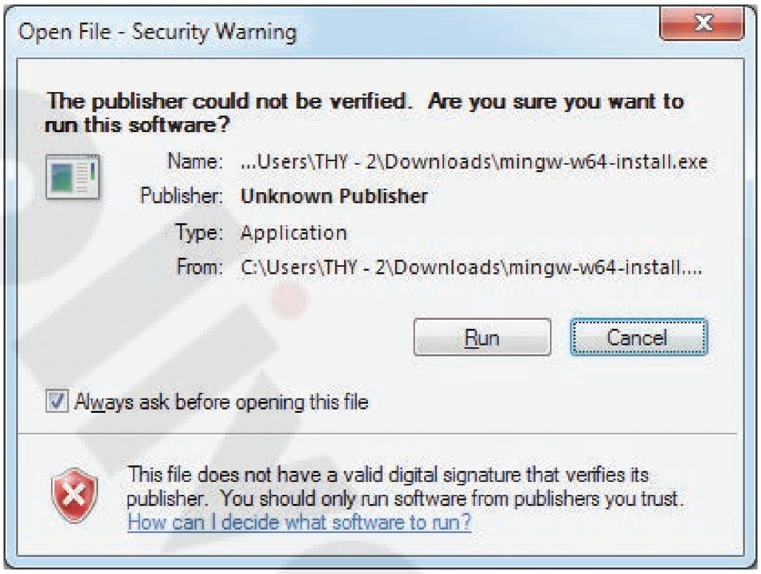
Annexure III.indd 347 30-01-2020 21:20:46

சாளரத்தை மூடவும். இந்த கோப்பை நிரந்தர இட அமைவிற்கு நகர்த்தவும். இதனால் MinGW நம் கணிப்பொறியில் நிறுவிக் கொள்ளலாம். (தேவைப்பட்டால், பிறகு மீண்டும் நிறுவலாம்)

படிநிலை 4: கொடுக்கப்பட்ட வழிமுறைகளின்படி நிறுவுதலை துவக்கவும்.

14.5.2.1 நிறுவுதல்

1. mingw-get-setup.exe குறும்படத்தை இரட்டைக் கிளிக் செய்யவும். பின்வரும் சாளரம் தோன்றும்.



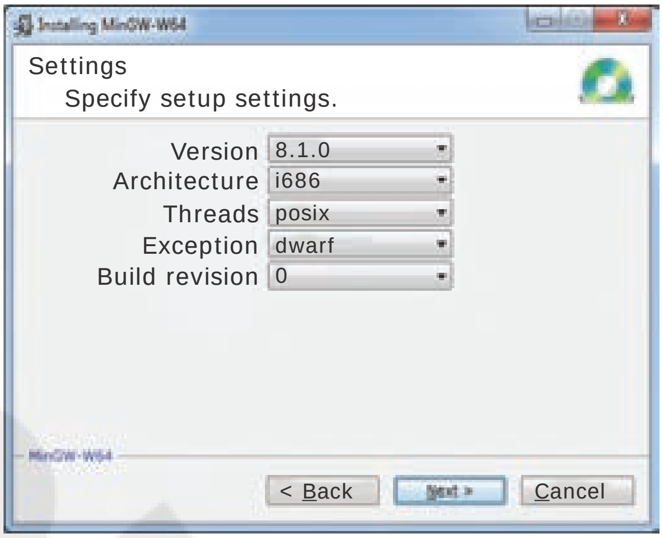
படம் 3

2. Runஐ கிளிக் செய்யவும். பின்வரும் மேல்மீட்பு சாளரத்தில் next பொத்தானை கிளிக் செய்யவும்.

படம் 4

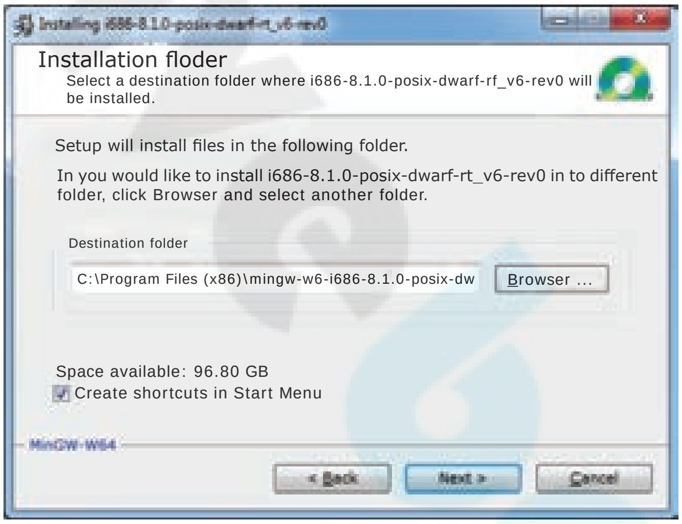
3. அமைவு அமைப்புகளை குறிப்பிடும் மேல்மீட்பு சாளரம் தோன்றும்.

Annexure III.indd 348 30-01-2020 21:20:46



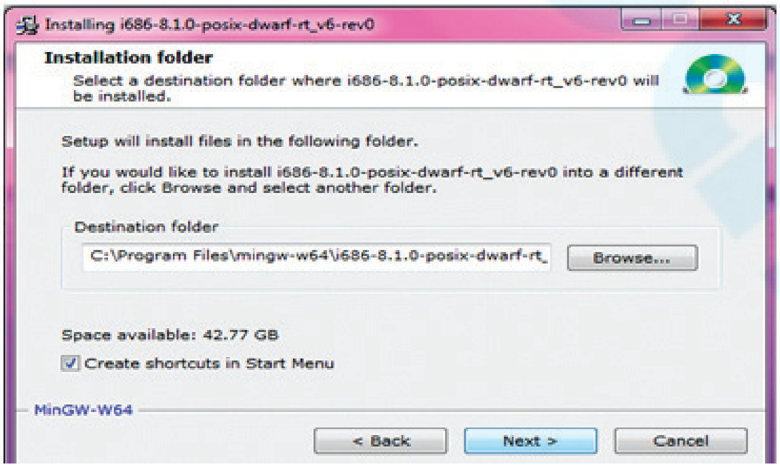
படம் 5

4. MinGW-W64 ஐ நிறுவ, பின்வரும் சாளரத்தில் இலக்கு கோப்புரையைத் தேர்ந்தெடுக்கவும் Next பொத்தானை கிளிக் செய்யவும்.



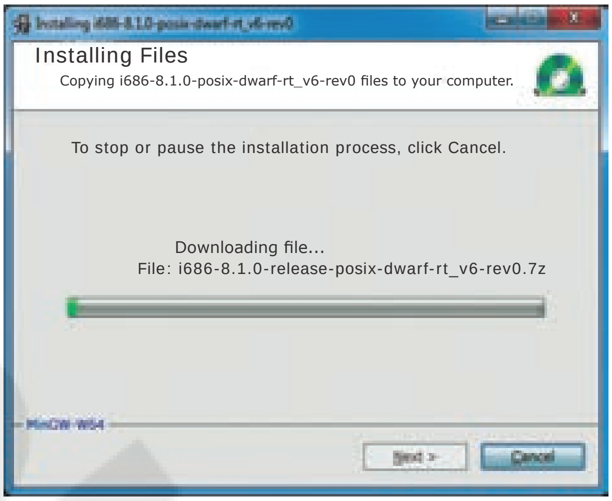
படம் 6

5. நிறுவுதல் நிறைவாகும் வரை காத்திருக்கவும்



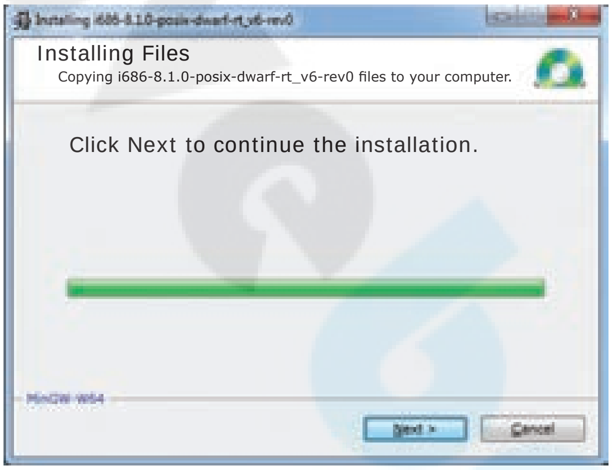
Annexure III.indd 349 30-01-2020 21:20:46

6. பைத்தான் ஒளிர் வூட்டி காண்பிக்கப்படும் போது, கிளிக் செய்யவும்.



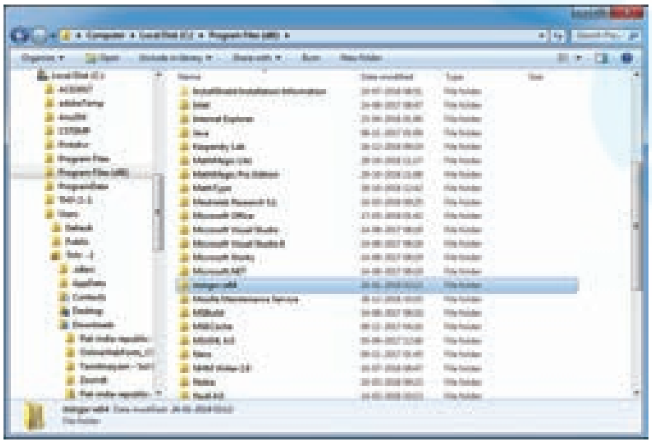
படம் 7

7 உங்கள் கணினியில் கோப்புறையை கண்டறியவும். உதாரணமாக, இங்கு mingw64 பின்வரும் பாதையில் அமைந்துள்ளது c:\Program Files\mingw-w64



படம் 8

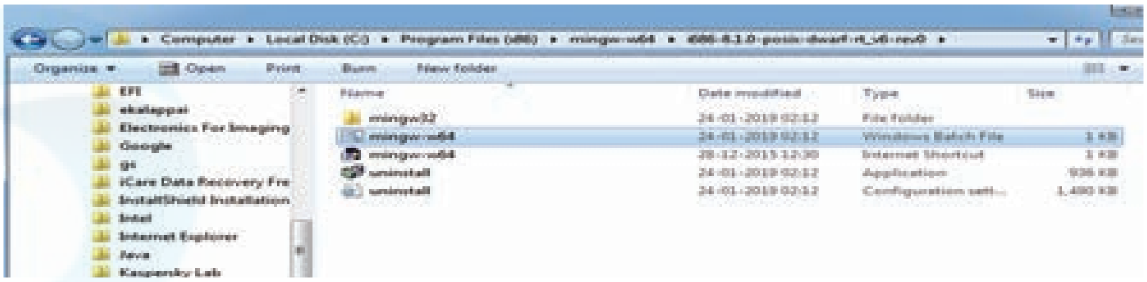
8. கோப்புறையை திறந்து, batch கோப்பை இரட்டை கிளிக் செய்யவும் நிரல் செயல்படுத்தப்படும்.



படம் 9

Annexure III.indd 350 30-01-2020 21:20:47

9. Open the folder and double click the batch file. The program will get executed.



படம் 10

10 திரைமுகப்பில் runக்கான சுருக்குவழி உருவாக்கப்படும் இரட்டை கிளிக் செய்து “command window” திறக்கவும்.

படம் 11

11 பைத்தான் நிரலை கட்டளை சாளரத்தின் வழியாக மட்டுமே இயக்க முடியும். ஏனெனில், இது C++ நிரல் போன்ற மற்ற நிரலாக்க மொழிகளின் நிரலைக் கொண்டுள்ளது. இந்த கட்டளை சாளரம் (Run முனையம்) பைத்தான் நிரலை பயன்படுத்தி, C++ நிரலை தொகுத்து, செயல்படுத்த இயக்க நேரத்தில் g++ தொகுப்பானை அழைக்கும்.

படம் 12

Annexure III.indd 351 30-01-2020 21:20:47

இணைப்பு 4

PIPஐ பதிவிறக்கம் செய்து நிறுவுதல்

முதலில் உங்கள் கணினியில் pip நிறுவப்பட்டுள்ளதா என்பதை அறிய வேண்டும். அவ்வாறு இருப்பின் pip-யை மேம்படுத்த வேண்டும். இதை செய்ய நீங்கள் கட்டளை தூண்டுகுறியை துவங்க வேண்டும். பைத்தான் கோப்புரை பாதை, விண்டோஸ் கோப்புரை பாதையுடன் இணைக்கப்பட்டிருந்தால் மட்டுமே கட்டளைகளை வேலைச் செய்யும். இதை pip நிறுவுவதற்கு அல்லது மேம்படுத்துவதற்கு முன் உறுதிப்படுத்திக் கொள்ள வேண்டும்.

pip நிறுவப்பட்டுள்ளதை சோதித்தறிதல்

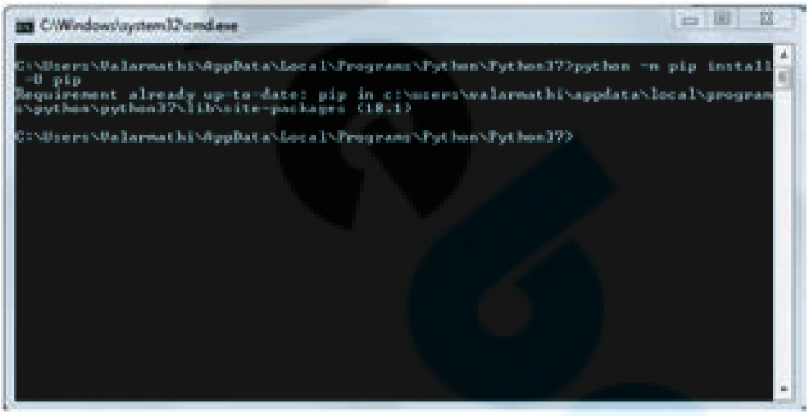
உங்கள் கணினியில் pip ஏற்கனவே நிறுவப்பட்டிருப்பதை சோதிக்க, கட்டளை தூண்டுகுறியை பைத்தான் ஸ்கிரிப்ட் கோப்புரைக்கு செலுத்த வேண்டும்.

குறிப்பு : pip நிறுவுதல் முறையை இணைப்பில் கொடுக்கப்பட்டுள்ளது.

பின்வரும் கட்டளைகளை கட்டளை தூண்டுகுறியை பயன்படுத்தி pip யின் புதிய பதிப்பை நிறுவ முடியும்.

Python –m pip install –U pip

-U என்பது pip யை புதிய பதிப்பை மேம்படுத்துவதை குறிக்கிறது.

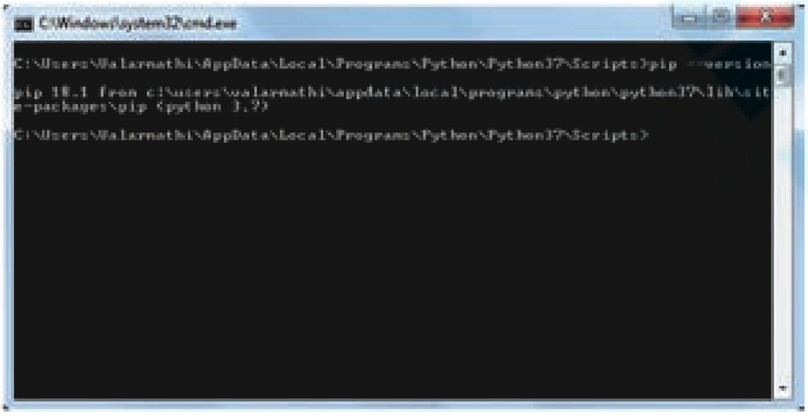


PIP பதிப்பை அறிய:

உங்கள் கணினியில் pipயின் பதிப்பை அறிந்துக்கொள்ள பின்வரும் கட்டளையை உள்ளிடவும்:

C:\Users\Your Name\AppData\Local\Programs\Python\Python36-32\Scripts>pip --version

கட்டளை தூண்டுகுறியில் வெளியீடு இவ்வாறு தோன்றும்:



Annexure IV.indd 352 10-02-2020 20:11:56

வெளியீடு திரையில் Pip யின் பதிப்பை காணலாம்.

Matplotlib நிறுவ, கீழ்க்கண்ட கட்டளை தூண்டுகுறியில் உள்ளிடவும்.

Python –m pip install –U matplotlib

இந்த கட்டளை matplotlibயை மூல நூலகத்திலிருந்து பதிவிறக்கும். ஏற்கனவே நிறுவப்பட்டிருந்தால் பின்வரும் திரை தோன்றும்.

```
C:\Windows\system32\cmd.exe

C:\Users\Valarmathi\AppData\Local\Programs\Python\Python37>python -m pip install
 -U matplotlib
Requirement already up-to-date: matplotlib in c:\users\valarmathi\appdata\local\
programs\python\python37\lib\site-packages (3.0.2)
Requirement already satisfied, skipping upgrade: python-dateutil>=2.1 in c:\user
s\valarmathi\appdata\local\programs\python\python37\lib\site-packages (from matp
lotlib) (2.7.5)
Requirement already satisfied, skipping upgrade: pyparsing!=2.0.4,!=2.1.2,!=2.1.
6,>=2.0.1 in c:\users\valarmathi\appdata\local\programs\python\python37\lib\site
-packages (from matplotlib) (2.3.0)
Requirement already satisfied, skipping upgrade: kiwisolver>=1.0.1 in c:\users\v
alarmathi\appdata\local\programs\python\python37\lib\site-packages (from matplot
lib) (1.0.1)
Requirement already satisfied, skipping upgrade: numpy>=1.10.0 in c:\users\valar
mathi\appdata\local\programs\python\python37\lib\site-packages (from matplotlib)
 (1.15.4)
Requirement already satisfied, skipping upgrade: cycler>=0.10 in c:\users\valarm
athi\appdata\local\programs\python\python37\lib\site-packages (from matplotlib)
(0.10.0)
Requirement already satisfied, skipping upgrade: six>=1.5 in c:\users\valarmathi
\appdata\local\programs\python\python37\lib\site-packages (from python-dateutil>
=2.1->matplotlib) (1.12.0)
Requirement already satisfied, skipping upgrade: setuptools in c:\users\valarmat
hi\appdata\local\programs\python\python37\lib\site-packages (from kiwisolver>=1.
0.1->matplotlib) (39.0.1)

C:\Users\Valarmathi\AppData\Local\Programs\Python\Python37>
```

தொகுப்புகளின் பட்டியல்

உங்கள் கணினியில் நிறுவப்பட்டுள்ள தொகுப்புகளை பார்வையிட, list கட்டளையை பயன்படுத்தலாம்.

C:\Users\YourName\AppData\Local\Programs\Python\Python36-32\Scripts>pip list

வெளியீடு திரையில் உங்கள் கணினியில் நிறுவப்பட்டுள்ள தொகுப்புகளின் பட்டியலைக் காண்பிக்கும்.

```
C:\Windows\system32\cmd.exe

C:\Users\Valarmathi\AppData\Local\Programs\Python\Python37\Scripts>pip list
Package         Version
--------------- -------
cycler          0.10.0
kiwisolver      1.0.1
matplotlib      3.0.2
numpy           1.15.4
pip             18.1
pyparsing       2.3.0
python-dateutil 2.7.5
setuptools      39.0.1
six             1.12.0
wheel           0.32.3

C:\Users\Valarmathi\AppData\Local\Programs\Python\Python37\Scripts>_
```


Annexure IV.indd 353 10-02-2020 20:11:57

கணினி அறிவியல், மேல்நிலை இரண்டாம் ஆண்டு
வல்லுநர்கள், மேலாய்வாளர்கள் மற்றும் நூலாசிரியர்கள் பெயர் பட்டியல்

பாடத் தயாரிப்புக்குழு தலைவர்

முனைவர் **T. V. கோபால்,** பேராசிரியர்,
கணினி அறிவியல் மற்றும் தொழில்நுட்பத் துறை,
பொறியியல் கல்லூரி, அண்ணா பல்கலைகழகம், கிண்டி,சென்னை.

மேலாய்வாளர்கள்

முனைவர். **ரஞ்சனி பார்த்தசாரதி**
பேராசிரியர், தகவல் மற்றும் அறிவியல் தொழில்நுட்பம்,
பொறியியல் கல்லூரி, அண்ணா பல்கலைகழகம், கிண்டி,சென்னை.

பாடப்பொருள் வல்லுநர்கள்

பேரா. பி.மலையரசு
இணைப் பேராசிரியர் மற்றும் துறைத் தலைவர், கணினி அறிவியல் துறை,
அரசு கலை மற்றும் அறிவியல் கல்லூரி, பெரும்பாக்கம், சென்னை.

முனைவர் **ம. ரமேஷ்குமார்**
உதவிப் பேராசிரியர் மற்றும் துறைத்தலைவர்,
கணினி அறிவியல் துறை,
அரசினர் ஆடவர் கலைக் கல்லூரி (தன்னாட்சி),
நந்தனம், சென்னை.

முனைவர் **.பெ. இராதா**
உதவி பேராசிரியர், தகவல் தொழில்நுட்பவியல் துறை,
அரசினர் கலை மற்றும் அறிவியல் கல்லூரி, கோயம்புத்தூர்.

முனைவர். **நெஸ்டர் ஜெயக்குமார்**
இணைப் பேராசிரியர் மற்றும் துறைத்தலைவர்,
கணிப்பொறியியல் துறை, இலயோலா கல்லூரி, சென்னை.

தமிழ் மொழிபெயர்ப்பாளர்கள்

S சிவசரவணன்
முதுகலைப் பட்டதாரி ஆசிரியர்,
சென்னை மேல்நிலைப் பள்ளி,
எம்.ஜி.ஆர். நகர், சென்னை.

D. ஜெயபாண்டியன்
முதுகலைப் பட்டதாரி ஆசிரியர்,
அரசு மேல்நிலைப் பள்ளி, சோழிங்கநல்லூர், காஞ்சிபுரம் மாவட்டம்.

T.K. சரோஜினி
முதுகலைப் பட்டதாரி ஆசிரியர்,
அரசு மேல்நிலைப் பள்ளி, பீர்க்கன்காரணை, காஞ்சிபுரம் மாவட்டம்.

C.ஸ்ரீதேவி
முதுகலைப் பட்டதாரி ஆசிரியர்,
அரசு முன்மாதிரி மேல்நிலைப் பள்ளி,
திருவல்லிக்கேணி, சென்னை.

A.G. கலைவாணி
முதுகலைப் பட்டதாரி ஆசிரியர்,
சென்னை பெண்கள் மேல்நிலைப் பள்ளி, மார்க்கெட் தெரு, பெரம்பூர்,
சென்னை.

K.V. மஞ்சுளா
முதுகலைப் பட்டதாரி ஆசிரியர்,
சென்னை மேல்நிலைப் பள்ளி,
துளைமேடு, சென்னை

கலை மற்றும் வடிவமைப்புக் குழு

வடிவமைப்பு

THY Designers and computers Chennai

In-House – QC
ராஜேஷ் தங்கப்பன்
பிரசாந்த் P
அருண் காமராஜ் பழனிசாமி
ர. மதன் ராஜ்

ஒருங்கிணைப்பு

ரமேஷ் முனிசாமி

நூலாசிரியர்கள்

க. கண்ணன்
முதுகலைப் பட்டதாரி ஆசிரியர்,
சென்னை பெண்கள் மேனிலைப் பள்ளி,
இராட்லர் தெரு, சென்னை.

வி . கோ. இராமகிருஷ்ணன்
முதுகலைப் பட்டதாரி ஆசிரியர்,
கர்நாடக சங்கா மேல்நிலைப்பள்ளி,
தி. நகர், சென்னை.

முனைவர் **ஹ. வித்யா**
முதுகலைப் பட்டதாரி ஆசிரியர்,
டி ஏ வி ஆண்கள் மேல்நிலைப் பள்ளி,
கோபாலபுரம், சென்னை

இரா.சீனிவாசன்
முதுகலைப் பட்டதாரி ஆசிரியர்
சாந்தோம் மேல்நிலைப் பள்ளி,
சென்னை.

நு . வீ. கௌரி சங்கர்
முதுகலைப் பட்டதாரி ஆசிரியர்,
சென்னை பெண்கள் மேல்நிலைப் பள்ளி,
நுங்கம்பாக்கம், சென்னை.

கோ. லெனின்
முதுகலைப் பட்டதாரி ஆசிரியர்,
சென்னை பெண்கள் மேல்நிலைப்பள்ளி, சைதாப்பேட்டை, சென்னை.

பிந்து மோகன்தாஸ்
முதுகலைப் பட்டதாரி ஆசிரியர்,
விஜயந்தா மாதிரி மேல்நிலைப்பள்ளி,
எச்.வி.எஃப் எஸ்டேட், ஆவடி, சென்னை.

இரா. கஜலஷ்மி
முதுகலைப் பட்டதாரி ஆசிரியர்,
ஜெயகோபால் கரோடிய இந்து வித்யாலயா மேல்நிலைப்பள்ளி,
மேற்கு மாம்பலம், சென்னை.

முனைவர் **க.எ. வளர்மதி**
முதுகலைப் பட்டதாரி ஆசிரியர், வேலம்மாள் வித்யாஷ்ரம் பள்ளி,
சூரப்பட்டு, சென்னை.

பாடநூல் ஒருங்கிணைப்பாளர்கள்

திரு.இரவிகுமார் ஆறுமுகம், முதல்வர்
மாவட்ட ஆசிரியர் கல்வி மற்றும் பயிற்சி நிறுவனம்,
மாயனூர், கரூர் மாவட்டம்.

திருமதி R. தமிழ்ச்செல்வி
பட்டதாரி ஆசிரியை, அரசு உயர் நிலைப் பள்ளி,
பூனாம்பாளையம், திருச்சி மாவட்டம்.

விரைவுக்குறியீடு மேலாண்மைக்குழு

இரா. ஜெகநாதன், இ.நி.ஆ,
ஊ.ஒ.ந.நி.பள்ளி, கணேசபுரம்,
போளூர் , திருவண்ணாமலை மாவட்டம்.

ஜே.எப்.பால்எட்வின் ராய், ப.ஆ.
ஊ.ஒ.ந.நி.பள்ளி, ராக்கிப்பட்டி, வீரபாண்டி, சேலம்

வ.பத்மாவதி ப.ஆ,
அ.உ.நி. பள்ளி, வெற்றியூர், திருமானூர்,அரியலூர்.

தட்டச்சர்

த. மீனா,
SCERT, சென்னை

இந்நூல் 80 ஜி.எஸ்.எம். எலிகண்ட் மேப்லித்தோ தாளில் அச்சிடப்பட்டுள்ளது.
ஆப்செட் முறையில் அச்சிட்டோர்:

XII Std Computer Science Vol-1 TM Acknowledgement 186.indd 354 10-02-2020 20:16:29

FREE Ebooks

Download Now

Current Affairs

Explore Now

Free Notes + Mock Tests + Mindmaps + Current Affairs

For TNPSC Group II/IIA, IV & VAO Exams

Web

APP

BLOG

Your one-stop destination for all exam related information & preparation resources.

Explore Now

FORUM

Interact with peers & experts, exchange scores & improve your preparation.

Explore Now

www.OliveBoard.in