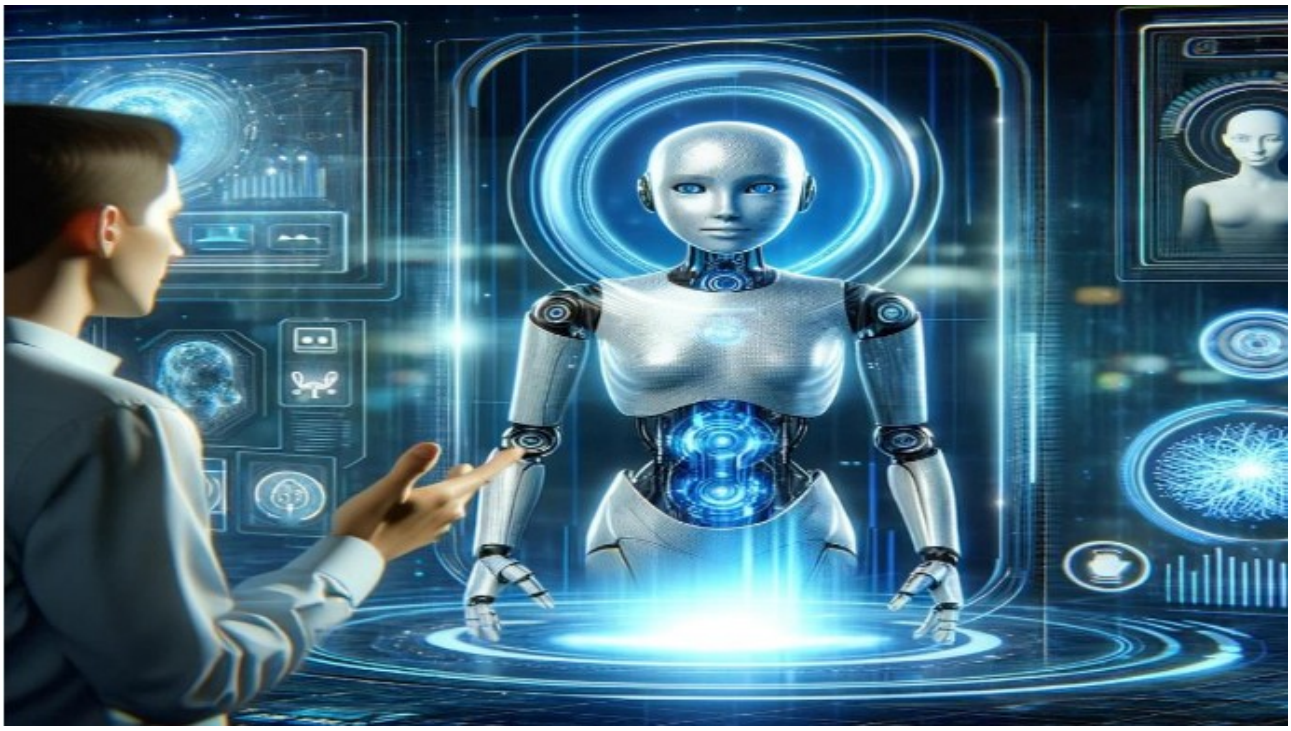
Draft Study Material

Artificial Intelligence Assistant

(QP Code: NIE/SSC/Q1003)

Sector: Information Technology-Information Technology Enable Services (IT-IteS)

Grade IX



विद्यया ऽमृतमश्नुते

एन सी ई आर टी
NCERT

PSS CENTRAL INSTITUTE OF VOCATIONAL EDUCATION
(a constituent unit of NCERT, under Ministry of Education, Government of India)
Shyamla Hills, Bhopal- 462 002, M.P., India
http://www.psscive.ac.in

Preface

Vocational Education is a dynamic and evolving field, and ensuring that every student has access to quality learning materials is of paramount importance. The journey of the PSS Central Institute of Vocational Education (PSSCIVE) toward producing comprehensive and inclusive study material is rigorous and time-consuming, requiring thorough research, expert consultation, and publication by the National Council of Educational Research and Training (NCERT). However, the absence of finalized study material should not impede the educational progress of our students. In response to this necessity, we present the draft study material, a provisional yet comprehensive guide, designed to bridge the gap between teaching and learning, until the official version of the study material is made available by the NCERT. The draft study material provides a structured and accessible set of materials for teachers and students to utilize in the interim period. The content is aligned with the prescribed curriculum to ensure that students remain on track with their learning objectives.

The contents of the modules are curated to provide continuity in education and maintain the momentum of teaching-learning in vocational education. It encompasses essential concepts and skills aligned with the curriculum and educational standards. We extend our gratitude to the academicians, vocational educators, subject matter experts, industry experts, academic consultants, and all other people who contributed their expertise and insights to the creation of the draft study material.

Teachers are encouraged to use the draft modules of the study material as a guide and supplement their teaching with additional resources and activities that cater to their students' unique learning styles and needs. Collaboration and feedback are vital; therefore, we welcome suggestions for improvement, especially by the teachers, in improving upon the content of the study material.

Deepak Paliwal
(Joint Director)
PSSCIVE, Bhopal

May, 2025

STUDY MATERIAL DEVELOPMENT COMMITTEE

Members

Deepak D. Shudhalwar, Professor (CSE), Department of Engineering and Technology, PSSCIVE, NCERT, Bhopal, Madhya Pradesh

Prakash Khanale, Sr Coordinator, School of Computer Science, YCMOU, Nashik, Ex-Professor and Head, Department of Computer Science, DSM College, Parbhani, Maharashtra

Member Coordinator

Deepak D. Shudhalwar, Professor (CSE), Head, Department of Engineering and Technology, PSSCIVE, NCERT, Bhopal, Madhya Pradesh

CONTENT

Module 1. Basics of AI Programming

Raju was checking his e-mails on his phone. He found that today me received many mails. Mails were automatically classified into different categories such as primary social, promotion, updates and spam. He was surprised how his phone can identify this. Later on he knows that it is possible because of Artificial Intelligence or AI.

AI is playing a very active role now in our day to day life. In this unit we will understand what is AI and how to program for AI using pictorial blocks.

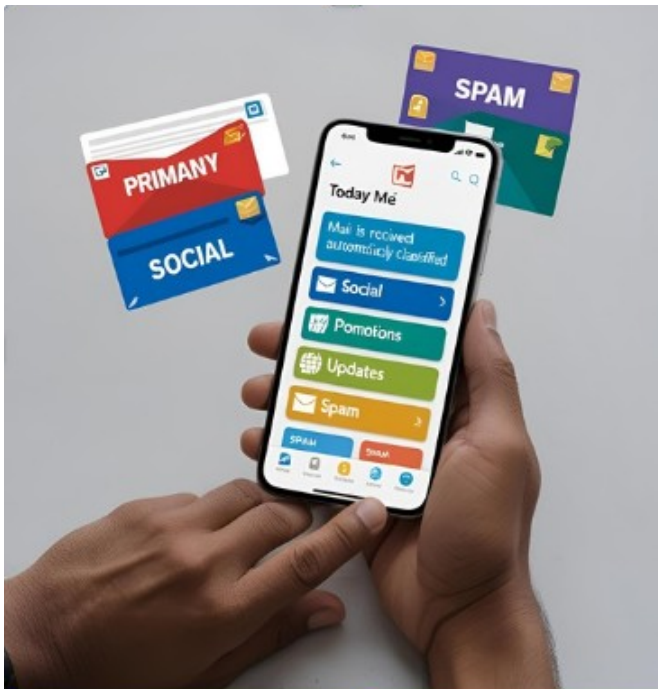


Fig. 1.1 : AI in Day to Day Life

Session 1. Introduction to Artificial Intelligence

Introduction to Artificial Intelligence

Artificial Intelligence is made up of two words *"Artificial"* and *"Intelligence"*. Artificial is something which is man-made, which does not occur naturally.

Intelligence can be defined as the 'ability to perceive or infer information. It is used to retain the knowledge that can be applied within an environment or context. Intelligence is the ability to interact with the world in the form of speech, vision, motion, and manipulation. It is the ability to model and reason the world. Intelligence is the ability to learn and solve problems. Due to intelligence one can learn and make decisions.

Artificial Intelligence (AI) is like giving a computer brain power.

Artificial Intelligence (AI) refers to the simulation of human intelligence in machines that are programmed to think and act like humans.

Fig. 1.1 Artificial Intelligence

AI is the technology that includes the following processes.

1. **Learning:** acquiring information and rules for using it. For example, machine learning.
2. **Reasoning:** using rules to reach a conclusion. For example, logical reasoning.
3. **Problem solving:** finding solutions to complex issues. For example, writing algorithms for stock market predictions.
4. **Perception:** understanding and interpreting input such as image and sound. For example, computer vision or image processing.
5. **Natural language processing:** enabling machines to understand human languages. For example, fuzzy languages.

Need of AI in School Education

In School Education, we normally teach those subjects that are useful for the students to make their career in that subject. Also, we teach those subjects that are frequently needed when he faces different real-life problems in the real world. The subject of artificial intelligence is developed over a long period of time of about 75 years. The use of AI in human lives is increased by many folds. Many people knowingly or unknowingly are making use of AI. Hence it is necessary that a student studying at school in class 9 should be aware of this topic. The course on AI can significantly enhance school education in the following ways.

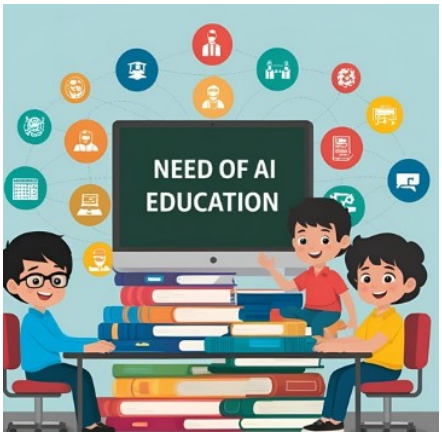


Fig. 1.2 : AI in School Education

1. **Professional development:** professionals in computer science or AI will be developed at an early stage.
2. **Ability of data analysis:** with AI, a student can easily analyse the data to obtain fruitful conclusions.
3. **Accessibility:** with AI a student can access many resources. Even students with special abilities can make use of AI tools.
4. **Intelligent learning:** a student can learn the subject even outside the classroom with the help of an intelligent learning environment.

The course on AI will also help to have an effective, efficient and inclusive learning environment for school students.

Practical Activity 1.1. Identification of AI technology

Material Required

Smartphone with internet connectivity

Procedure

Step 1. Connect your smartphone to WIFi or make use of built in Internet connectivity.

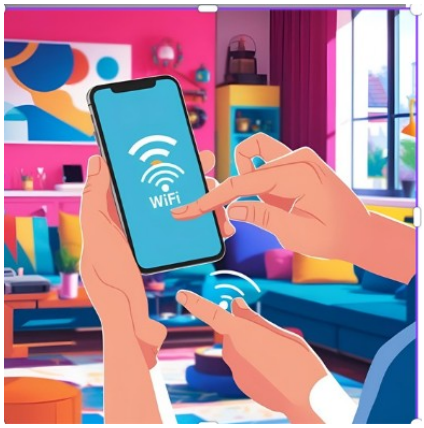


Fig. 1.3 : Connecting smartphone to WIF

Step 2. Give an oral command to make a call to your friend.

Fig. 1.4 : Calling using smartphone

Step 3. Observe that your phone on its own calls to your friend.

Fig. 1.5 : Smartphone using AI

Step 4. Note that mobile phone has used narrow AI

Practical Assignment

Make use of your phone to play songs on your internet by using oral commands.

Types of AI

Artificial Intelligence can be broadly divided into two categories – AI based on capability and AI based on functionality. The following image illustrates these types of AI as shown in Figure.

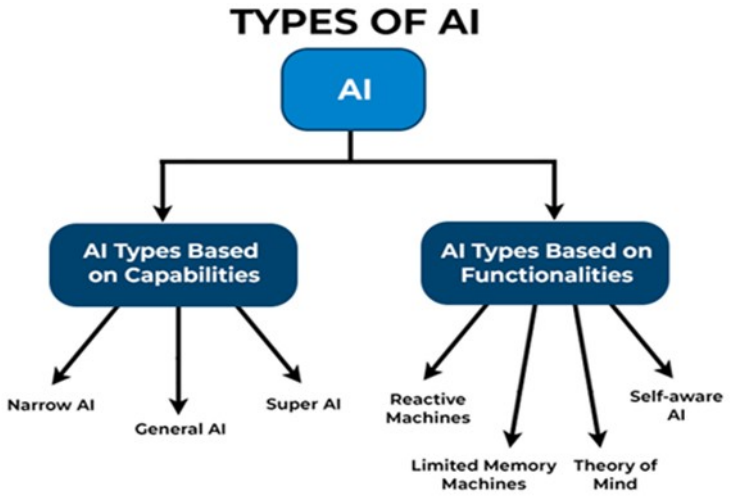


Fig. 1.6 : Types of AI

Based on capability, there are 3 types of AI — *(1) Narrow AI, (2) General AI and (3) Super AI.*

1. **Narrow AI:** Narrow AI is a goal-oriented AI. It is also known as weak AI. It refers to AI systems that are designed and trained for a specific task or a narrow set of tasks. Apple's Siri and Amazon Alexa are examples of narrow AI. They are good in speech recognition but operate with a limited pre-defined range of functions.
2. **General AI:** General AI, also known as Strong AI. It is Artificial General Intelligence (AGI). It can understand and learn any intellectual task that a human being can perform. Currently, no such system exists which can come under general AI. But in future it is possible that a general AI system may get developed over a period of time.
3. **Super AI:** Super AI is the AI that surpasses human intelligence. It can perform any task better than a human. Capabilities of a machine with super AI include thinking, reasoning, solving a puzzle, making judgments, learning, and communicating on its own. Today, super AI is a hypothetical concept but represents the future of AI.

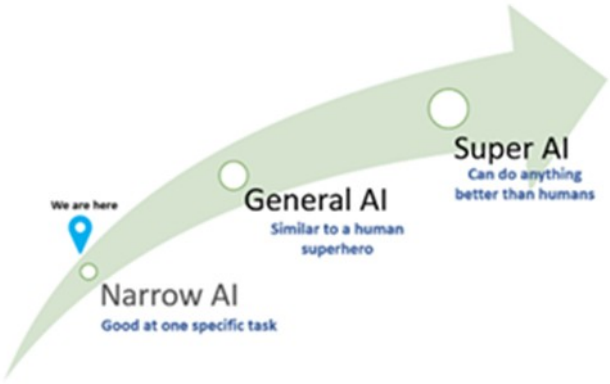


Fig. 1.7 : Types of AI based on capabilities.

Figure shows types of AI based on capabilities.

Types of AI — Based on functionality

Based on functionality, there are 4 types of AI — (1) Reactive Machines, (2) Limited Memory, (3) Theory of Mind and (4) Self-Aware AI.

1. **Reactive machines:** Reactive machines are AI systems that do not have any memory. These are basic AI types that do not store past experiences. These machines are designed to perform specific tasks. For example, programmable coffeemakers and washing machines are examples of reactive machines of AI.
2. **Limited memory machines:** This AI doesn't remember everything forever, but it uses its short-term memory to learn from the past. They have a built-in program running the memory. Self-driving cars are examples of limited memory AI. Another example is platforms such as Netflix to suggest movies based on the user's past preferences and behaviours.
3. Theory of Mind: These AI machines can socialize and understand human emotions. It will have the ability to cognitively understand somebody based on the environment. Machines with such abilities have not been developed yet.
4. Self-aware AI: This is similar to Super AI. Self-aware AI deals with super-intelligent machines with their consciousness, sentiments, emotions, and beliefs. Such systems are expected to be smarter than a human mind and may outperform us in assigned tasks.

Uses of AI

AI can be used in various disciplines. Some of them are as mentioned below.

1. **AI in Education:** With AI it is possible to have personalised learning experience. AI provides tutoring, support and it automates the administrative tasks.
2. **AI in Healthcare:** AI can be used in the diagnosis, predictive analysis of disease, personalised medicine and to manage the patient records.
3. **AI in Creative Fields:** AI can be used for creation of art, music, and literature.
4. **AI and Employment:** With AI it is possible to generate new employment opportunities such as data science, AI-powered robots, preventive and predictive maintenance.
5. **AI in Finance:** AI is used for fraud detection, credit scoring, algorithmic trading and personalised financial advice.
6. **AI in the Automotive industry:** AI powers autonomous vehicles where the car drives automatically. Traffic management systems can be automated by using AI systems.
7. **AI in E-Commerce:** AI can be used for virtual shopping and personalised shopping. It assists customers in locating products, and addressing inquiries.
8. **AI and Human-AI Collaboration:** Humans and AI will collaborate more closely in symbiotic partnerships. AI will aid professionals in decision-making, research, and data analysis, enhancing human capabilities.
9. **AI in Agriculture:** With AI, it is possible to practice precision farming through data analysis, crop monitoring and pest detection.
10. **AI in security:** Threat detection, surveillance, monitoring, access control, fraud detection can be enhanced by making the use of AI. Data security, cyber security, network security, cloud security can be performed in a more efficient manner by using AI.

Fig. 1.8 : Uses of AI

Practical Activity 1.1. Identification of use of AI in our daily activities.

Material Required

Pen, Paper

Procedure

Step 1. Your morning wakeup call can be managed by phone. It makes use of AI.

Step 2. Your email service provider makes use of AI for classification of mails.

Step 3. You can get entertained by AI by offering you your favourite music and TV programs.

Step 4. You can make use of AI in your studies for topic searching & notes.

Step 5. You will be updated automatically for your needs by AI.

Step 6. You can get connected with your family and friends where AI plays an important role.

Step 7. You can order your favourite food using AI.

Step 8. You can prepare travel plans and book tickets using AI chatbots.

Fig. 1.9 : AI in Daily activities

Practical Activity 1.2. Identify and use the various type of AI technology

Material Required

Pen, Paper, washing machine

Procedure

Step 1. Observe front panel of your automatic washing machine

Fig.

Step 2. Take note of different control buttons provided on the front panel.

Fig.

Step 3. Start the machine and observe how steps of washing are executed one after the other.

Fig.

Step 4. Note that this is a kind of reactive AI used in washing machines.

Practical Activity 1.3. Use of AI in Netflix.

Material Required

Pen, Paper, Netflix on TV

Procedure

Step 1. Watch some program on Netflix TV

Fig.

Step 2. Next time, when you switch ON TV, Netflix will suggest your programs based on your history.

Step 3. Note AI used is limited memory AI.

Practical Assignment 1.2

1. List down your daily activities from morning to evening. Identify where you can use AI.

2. List down the industries/areas where AI can be used effectively.

3. Suppose you want to swap values of two values and write possible solutions. Identify which solution is efficient.

4. Write steps for organizing a fund-raising event for a charitable institution.

5. Write steps for organizing a trip to hill station events by using problem decomposition.

A. AI in shops

AI can be used in Retail shops to enhance the customer experience, optimize operation and improve security.

1. Use of chatbots and virtual assistance to answer customer queries and provide product information.
2. Based on the customer purchase history, the AI system can suggest additional items for purchase.
3. With AI, we can automate the stock monitoring. This reduces over stocking and stock out.
4. Theft detection and prevention. AI can analyse the video to detect suspicious behaviour on unauthorised access.

Practical Activity 3.1. Demonstrate the use of AI at airport/IRCTC web.

Material Required: chatbot, Kiosk at airport/IRCTC website.

Procedure

Step 1. Visit chatbot kiosks at any airport/IRCTC web nearby you.

Fig.

Step 2. How boarding passes can be printed by using PNR.

Fig.

Step 3. Change of planes or new booking is also possible.

Step 4. Using AI chat bot can assist you for booking new train/air tickets.

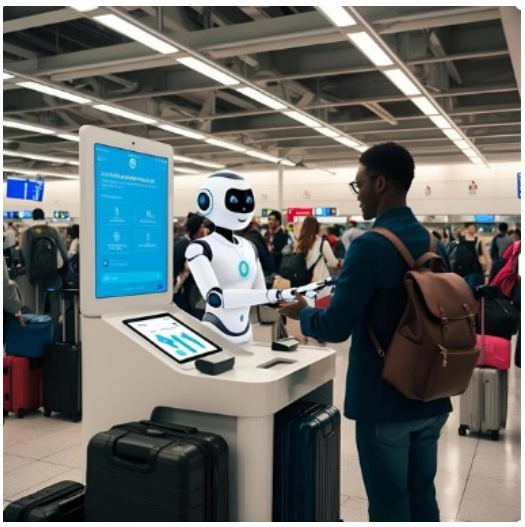


Fig. AI Chatbot at Airport

B. AI in office

AI can be used in offices to perform biometric attendance, document processing, and email sorting.

1. Every employee's attendance can be recorded automatically by using biometric devices such as fingerprint, scanner and face recognition system. It records in-time and exit-time in the office.
2. Document processing such as scanned documents or pdfs can be organised by using AI based optical character recognition devices. Invoice processing can also be performed by using AI based tools.
3. AI powered tools can help to manage email by categorizing it into different categories such as spam, junk, social, promotional and primary.

Practical Activity 3.2. Demonstrate the use of AI in Office

Material Required: Email Account, Computer/Laptop.

Procedure

Step 1. See how many emails you have received today.

Fig.

Step 2. Using AI, the email program automatically classifies it into spam, Junk, social, Promotional & Primary.

Fig.

Step 3. Send mail to your friend.

Step 4. Asking to check his email and see how it is classified.

C. AI in Industry

AI can be used in industries to enhance productivity, optimize processes, improve quality and reduce cost.

1. AI algorithms can analyse the data from equipment sensors to predict which machines are likely to fail. This will allow timely maintenance and reduce downtime of the equipment.
2. Computer vision and machine learning for identifying defects during the production. Hence quality can be improved.
3. AI powered robotics can be used to perform repetitive and dangerous tasks such as welding, painting and assembly.
4. Root optimisation, warehouse automation and demand forecasting for any product can be achieved through AI tools.

Practical Activity 3.3. Demonstrate the use of AI in industry/Car Industry.

Material Required: Industry visit/video or production

Procedure

Step 1. Visit nearby industries or watch their production video.

Fig.

Step 2. See how robot arms are performing welding of parts.

Fig.

Step 3. See how robots are used for painting cars.

Fig. AI in Car Industry

Introduction to Artificial Intelligence (AI) Assistant

With the evolution in AI technology a large number of jobs are available in the field of AI. AI Assistant is a job that will help the organisations with the digital tasks. For such jobs it is necessary to practice AI technology. This job role is

developed to provide the opportunity to choose the career in the field of AI and starting their career with AI Assistant.

This Qualification covers the fundamentals of Python programming and libraries like NumPy and Pandas used for data analysis. The course will lay stress on developing programming skills by providing practical exposure to the aspiring Python developers and also introduce to the concepts of Machine Learning. Participants from any background can develop the skills needed to become an AI Assistant.

Definition of AI Assistant

AI Assistant is personnel that will use AI technology to help people and organisations with their digital tasks.

Roles and Responsibilities of AI Assistant

An AI Assistant can perform a variety of tasks as given below.

1. Performing analysis of the data.
2. Creating contents.
3. Automating the tasks.
4. Scheduling of the tasks.
5. Answering the questions.
6. Managing to do lists.
7. Understand and learn preferences based on the data.
8. Understanding the data and making predictions of the future outcomes.
9. Making recommendations based on the history and preferences.
10. Understanding and processing of the natural language.

An AI Assistant must use the existing AI technology to resolve the various issues related with the digital tasks.

Knowledge and Skills required for AI Assistant

Following knowledge and skills are required for the AI Assistant.

1. Fundamentals of AI
2. Basics of programming
3. Scripting tools such as PictoBlox or Scratch
4. Concept of data and data science
5. Pre-processing of data
6. Concept of data analysis,
7. Data analysis tools

8. Concept of machine learning
9. Types of machine learning
10. Elementary Python programming
11. Introduction to NumPy,
12. Array manipulation and computation using NumPy,
13. Data manipulation using Pandas,
14. Neural Network
15. Applications of Neural Network
16. Machine Learning tools

Fig. Knowledge and Skills required for AI Assistant

Future Scope of AI Assistant

AI technology is getting developed at a rapid pace. It is being used across various industries from health care, finance, to education and cyber security. Now we can find that AI is everywhere. So, there is a large future scope for manpower in the field of AI. Some of the areas are as mentioned below.

1. Job Automation

With the advancement in machine learning, robotics and AI technology, most of the traditional jobs that are performed by humans are automated. Here we will require an AI assistant to assist the automation.

2. Cyber Security

Due to automation there is an increasing threat of cyber-attacks and data bridges. Organisations are using AI powered tools to enhance their security measures. Here also we require AI Assistant for better Cyber Security.

3. Healthcare

Shortage of medical professionals can be replaced by AI powered tools. These tools can analyse vast amounts of medical data to identify the patterns and predict the disease. In such a case an AI assistant is required for the hospital for such analysis.

4. Education

It is possible to offer a personalised learning environment that will suit the pace, style and need of the learner. This can be achieved by using AI technology. AI assistants in colleges and schools will help them to achieve this.

5. Enhanced Customer Experience

AI chatbots can provide 24x7 customer services and support which otherwise is difficult for human persons. They will also improve response time and reduce the wait time. By using natural language processing, human-like conversation is possible. AI Assistant will play a crucial role enhancing the customer experience.

Fig. Future scope of AI

Check your progress

A. Multiple choice questions

1. Intelligence is the ability to interact in the form of (a) speech, (b) vision, (c) motion, (d) manipulation (e) All of the above.
2. Which of the following type of AI is based on functionality (a) Self-Aware AI (b) Narrow AI, (c) General AI and (d) Super AI.
3. Which of the following type of AI is based on capability (a) Reactive Machines, (b) Limited Memory, (c) Theory of Mind and (d) Super AI.
4. AI is used in industries to (a) enhance productivity, (b) optimize processes, (c) improve quality (d) reduce cost (e) All of the above
5. AI powered robotics can be used to perform (a) welding, (b) painting (c) assembly (d) All of the above

6. AI is used in Retail shops to (a) enhance customer experience, (b) optimize operation (c) improve security (d) All of the above

B. Fill in the blanks

1. Intelligence is the ability to _____ or _____ information.
2. AGI stands for _____
3. Super AI is a _____ concept.
4. AI powered robotics can be used to perform _____ and _____tasks
5. AI is used for _____ and _____ shopping

C. State whether the following statement is True or False

1. Artificial Intelligence is the simulation of machine intelligence.
2. Students with special abilities can make use of AI tools.
3. Narrow AI is known as a strong AI.
4. Apple's Siri and Amazon Alexa are examples of general AI.
5. Super AI is the future of AI.

D. Answer the following questions in short

1. What is the need of AI in School Education?
2. State the differtent types of AI based on capabilities?
3. State the differtent types of AI based on functionalities?
4. List the few sectors where AI can be used.
5. What is AI Assistant?
6. List the few roles and responsibilities of AI Assistant.
7. List the few areas where there is a scope of AI.
8. How AI can be used in Healthcare sector?
9. How AI can be used in Education sector?

Session 2. Programming Basics

In this session we will understand how to write algorithms and flowcharts before we go for the discussion of block coding.

2.1 ALGORITHMS

In our day-to-day life we perform activities by following a certain sequence of steps. Examples of activities include getting ready for school, making breakfast, riding a bicycle, wearing a tie, solving a puzzle and so on. To complete each activity, we follow a sequence of steps.

Let us now consider an example to post a letter. The sequence of steps for this would be:

Fig: Posting of Letter

Step 1. Writing the letter

Step 2. Folding the letter

Step 3. Inserting the letter in an envelope

Step 4. Writing the address

Step 5. Sticking the stamp

Step 6. Posting the letter

Step (1), (2) and (3) in this algorithm should be followed in the given order. Step (4) and (5) can be interchanged. The reason for this is, you can either stick the stamp after writing the address or write the address after sticking the stamp.

Thus, there are steps in an algorithm which should be followed in a strict sequential order. Sometimes, if the order of some steps is changed, it does not affect the process and the output is the same.

Need of Algorithm

It is required to prepare an algorithm before writing a program. It prepares a roadmap of the program to be written, before actually writing the code. It helps the programmer to clearly visualise the instructions to be written. For example,

searching using a search engine, sending a message, finding a word in a document, booking a taxi through an app, performing online banking, playing computer games, all are based on algorithms.

Writing an algorithm is mostly considered as a first step to programming. It is easy to code the program from algorithms in a high-level programming language. If the algorithm is correct, the computer will run the program correctly, every time. So, the purpose of using an algorithm is to increase the reliability, accuracy and efficiency of obtaining solutions.

An algorithm is a set of finite steps which when carried out for a given set of initial conditions, produce the corresponding output. In our everyday life we observe that whenever we face a certain problem then there exists some method to solve the problem. This method consists of some finite logical steps. Similarly, in order to solve a problem by computer we need to decide the logical steps. Such a set of steps is called an algorithm.

Characteristics of a good algorithm

There are certain characteristics associated with the algorithm.

1. **Precision —** There should be a finite number of steps in the algorithm and they are precisely stated or defined. One step should not alter or change the structure of the other steps.
2. **Uniqueness —** Every step must be accurate and complete in itself. The results of each step are uniquely defined and only depend on the input and the result of the preceding steps.
3. **Finiteness —** the algorithm always stops after a finite number of steps. The logic should be capable of handling the worst conditions.
4. **Input —** Whenever needed there should be provision to accept the input data.
5. **Output —** After solving the problem the algorithm should produce the corresponding output.
6. **Terminate —** An algorithm should terminate in a finite time.

While writing an algorithm, it is required to clearly identify the following:

- The input to be taken from the user,
- Processing or computation to be performed to get the desired result,
- The output desired by the user

An algorithm therefore corresponds to the solution to a problem that is independent of any programming language. Algorithms can be written either by using the natural language like English or one can make the use of pseudo

language. It is possible that the algorithm can be represented graphically and it is being called as a flowchart.

Practical Activity 2.1. Write an algorithm to find the square of a number.

Procedure

Let us first identify the input, process and output:

Input: Number whose square is required

Process: Multiply the number by itself to get its square

Output: Square of the number

Algorithm to find the square of a number.

Step 1. Input a number and store it to num

Step 2. Compute num * num and store it in square

Step 3. Print square

Practical Activity 2.2. Write the algorithm for swapping of variables.

Suppose you have been given two variables A and B and asked to exchange the value of A and B.

The problem of interchanging the values occurs in many sorting and data manipulation algorithms. Suppose A is assigned with a value 172 and B is assigned with value 563. Our task is to replace the content of A with the value 563 and the content of B with the value 172.

Procedure

Step 1. Save the original value of "A" in variable 'T', where T is the temporary variable.

Step 2. Assign to 'A' the original value of "B".

Step 3. Assign to 'B' the original value of 'A' that is stored in T.

Step 4. Stop

After executing the above four steps we observe that the values of 'A' and 'B' are interchanged. This is called swapping of variables.

Assignment 2.1

Suppose your want to swap values of two variables. Write the possible solutions. Identify which solution is efficient.

Practical Activity 2.3. Compute the factorial of a number n, where n >= 1.

The factorial of a number n is given by the formula n! = 1 × 2 × 3 × 4 ×....× n where n >= 1.

i.e. n! = n x (n-1)!

Here initially we need to assume that the initial value of the product is equal to 1 and then we go on multiplying this product by 1, 2, 3, ..., until n. This can be achieved by using the following steps.

Procedure

Step 1. Read the value of variable n.

Step 2. Set product = 1 and count =1.

Step 3. While count <= n, repeatedly do:

a. Compute new product by multiplying count to the product.

Product = product x count.

b. Increase the count by 1.

Count = count + 1.

Step 4. Result is the recent product.

Step 5. Stop.

Assignment 2.2

Write an algorithm to find the addition of the set of numbers.

Write an algorithm to find the Fibonacci sequence.

2.2. REPRESENTATION OF ALGORITHMS

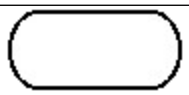
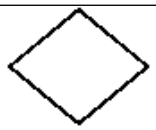
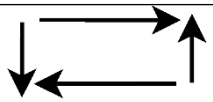
Using algorithms, the software designers or programmers analyse the problem and identify the logical steps that need to be followed to reach a solution. Once the steps are identified, the need is to write down these steps along with the required input and desired output. There are two common methods of representing an algorithm —flowchart and pseudocode. These methods can be used to represent an algorithm.

Flowchart

Flowchart is a graphical representation of an algorithm. This tool is used in the design of the program. To express different operations in the flowchart various standard symbols are used. These are graphical symbols. All symbols are connected among themselves in order to show the flow of information and processing. Each symbol represents a certain activity. For example, we can have a symbol of start and end, processing, decision, input and output and connectivity. Symbols are connected to each other by using flow lines. These lines have an arrow which indicates the direction of the data flow.

Different symbols as prescribed by American National Standard Institute (ANSI) which are frequently required while drawing flowchart are tabulated in table 4.1.

Table 2.1: Symbols used in the Flowchart

Flowchart Symbol	**Symbol Name**	**Description**
	Terminal (Start or Stop)	Terminals are the oval shape symbols used to represent start and stop of the flowchart.
	Process	Process symbols are represented by a rectangle and are used to represent processes such as Arithmetic operations or Data movements.
	Input/ Output	Input symbol is represented by a parallelogram, used to read input data and display the output.
	Decision	Decision symbol is represented by a diamond shape, used to check any condition or decide for which there are two options: true or false.
	Predefined process or function	It is used to define functions.
	Flow Lines	Flow lines are represented by an arrow used to connect symbols used in flowchart and indicate direction of flow.
	Connector	It is used to connect or join flow lines.

It is essential to follow certain guidelines while preparing the flowcharts as below.

1. Standard symbols should be used while drawing flowchart.
2. Ensure that flowchart has START (or BEGIN) and STOP (or END).
3. Flowchart should be neat, clean and easy to follow.
4. There should be no ambiguity.
5. The usual direction of flowchart is from top to bottom or from left to right.
6. The terminal symbol, that is, START/BEGIN or STOP/END should have only one flow line.
7. Only one flow line should come out from the process symbol.
8. Only one flow line should enter a decision symbol, but two or three flow-lines, one for each possible answer, can leave the decision symbol.

9. If the flowchart is lengthy and complex, the connector symbol should be used to reduce the number of flow lines.
10. Avoid intersection of flow lines.

Example of Flowchart

Flowchart example for finding whether a number is odd or even is shown below.

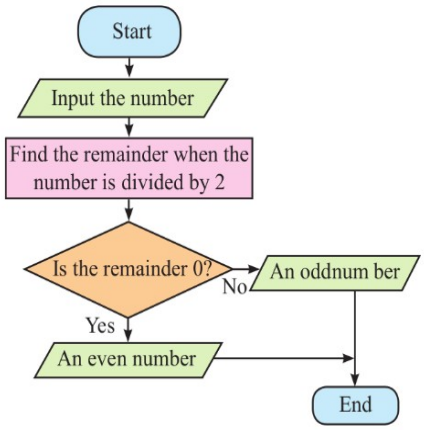


Fig. 2.1: Flowchart for finding whether a number is odd or even

Advantages of using flowchart

Following are the advantages of using flowchart.

1. Flowcharts are independent of any programming language. i.e. they are not related to a particular programming language. Hence, they can be implemented in any programming language.
2. Flowcharts are easy to understand. The data flow of the program can be easily understood by the flowchart.
3. Finding errors and recovering them is easy in the flowchart.
4. Whenever the program needs further modification, then through flowchart it is easy to incorporate such modifications.

Practical Activity 2.4. Write steps and draw the flowchart for organizing a fund-raising event for a charitable institution.

Solutions

Steps for organizing a fund-raising event for charitable institution

Step 1. Start

Step 2. Determine purpose

Step 3. Identify target audience

Step 4. Set timeline

Step 5. Choose venue

Step 6. Check if entertainment is required, then plan the activities and execute the event

Step 7. If entertainment is note required, then execute the event

Step 8. Stop the process

Flowchart for organizing a fund-raising event for charitable institution

Assignment 2.3

Write steps for organizing a trip to hill station events by using problem decomposition.

Practical Activity 2.5. Draw a flowchart for swapping of variables.

Draw the flowchart to interchange the values of two variables.

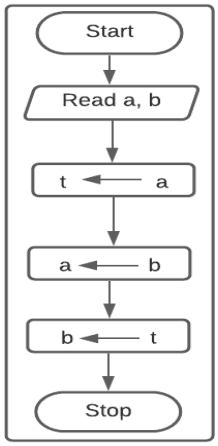


Fig. 2.2: Flowchart to interchange the values of two variables

Practical Activity 2.6. Draw a flowchart to compute the factorial of a number n, where n >= 1.

Solution – Factorial of a number n is given by the formula

$n! = 1 \times 2 \times 3 \times \ldots \times n$

First, we ask the number n whose factorial we have to calculate. Enter the number n. As the factorial of a number 1 is 1, assign the product to 1 and count to 1. Compute the Product by multiplying Product to Count. Increment the Count by 1 repeat the steps to compute the Product till the counter reaches the value n.

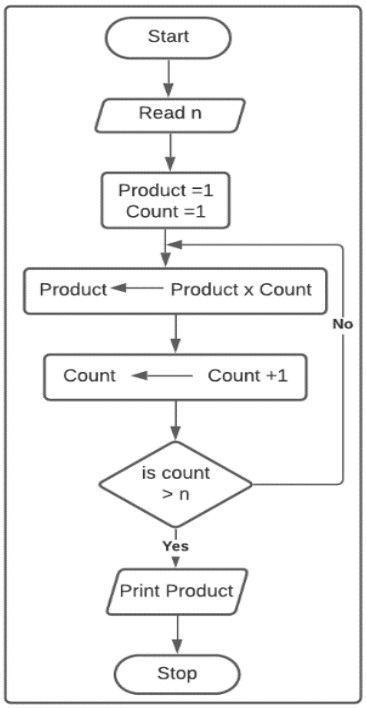


Fig. 2.3: Flowchart to find the factorial of a given number

Practical Activity 2.7. Draw the flowchart to find the largest of three given numbers

Solution – Consider the three numbers a, b, c. Following is the flowchart to find the largest of three given numbers.

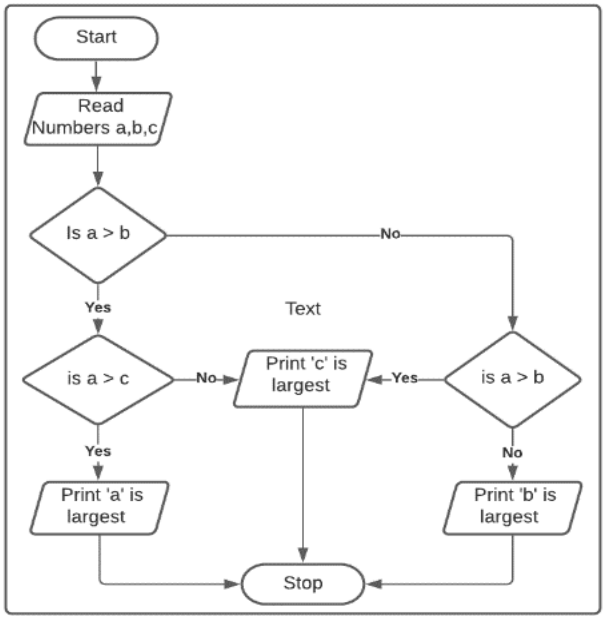


Fig. 2.4 : Flowchart to find the largest of three numbers

Practical Activity 2.8. Draw the flowchart to find the largest set of numbers.

Solution – Following is the flowchart which reads the number and assigns it to the largest. It allows the user to enter another number and then it compares the

new number with the existing largest number. If it is greater than the already stored number then it assigns the largest to the new number. The loop is executed till the user does choose to further enter the new numbers.

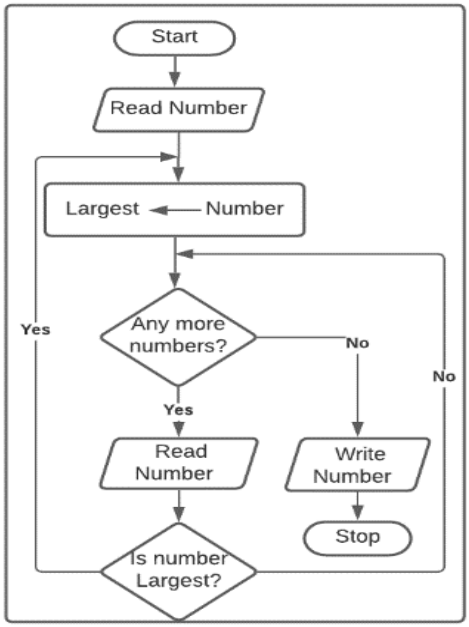


Fig. 2.5 : Flowchart to find the largest of set of number

Practical Activity 2.9. Draw the flowchart for generation of n Fibonacci numbers.

Solution – The Fibonacci sequence, also known as Fibonacci numbers, is defined as the sequence of numbers in which each number in the sequence is equal to the sum of two numbers before it. The Fibonacci Sequence is given as:

Fibonacci Sequence = 0, 1, 1, 2, 3, 5, 8, 13, 21, ...

We assign the first number as 0 and second number to 1. The following flowchart will generate the Fibonacci numbers for the given number n.

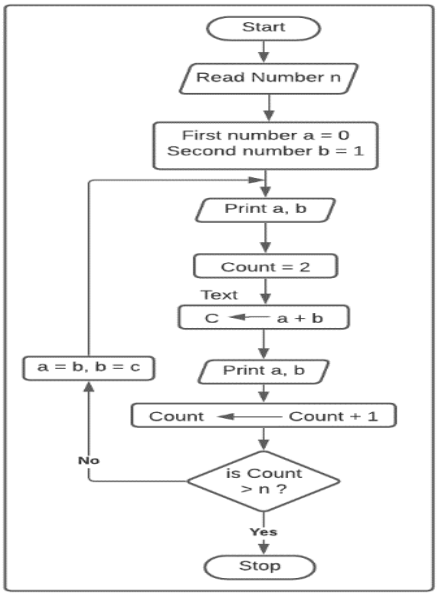


Fig. 2.6: *Flowchart for generation of Fibonacci numbers*

Assignment 2.5

Draw the flowchart to

1. Find the addition of the set of numbers.
2. Determine whether the given number is prime or not.
3. Arrange the numbers in ascending or descending order.
4. Count non-zero digits in a set of 100 digits.
5. Find the perimeter and area of a rectangle.
6. Calculate the new salary of an employee by adding an increment of 3% to the basic salary.

2.3. FLOW OF CONTROL

The flow of control depicts the flow of events as represented in the flow chart. The events can flow in a sequence, or on branch based on a decision or even repeat some part for a finite number of times. There are three types of flow control – sequence, selection and loop structure as shown in Figure 2.13.

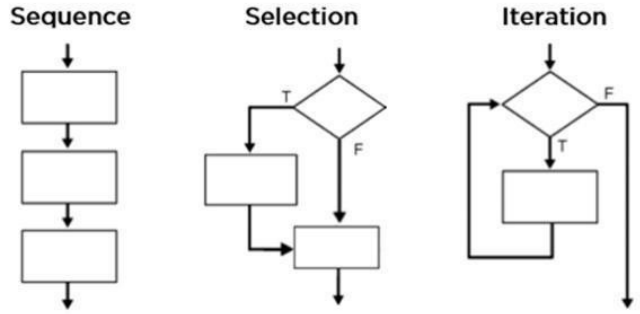


Fig. 2.7: Sequence, selection and loop

Sequence

When all the statements are executed one after another from the beginning to the end of an algorithm in a strict order is called a sequence. However, the statements in an algorithm may not always execute in a sequence. Sometimes the problem may require doing some routine tasks in a repeated manner or behave differently depending on the outcomes of previous steps.

Example:

1. Climbing up or down step by step when going on a staircase

2. Students who were admitted to class 1 of the school continue studies till class 12

Fig. 2.8: Example of sequence

Selection

Selection is a situation where step(s) are executed depending on whether a condition of an algorithm is satisfied or not. There are two choices; if the condition is satisfied, one is selected and if it is not satisfied, the other selection is selected.

Examples of selection

1. Admitting a child to Class 1:

```
If a child is below 6 years as at 1st July of that year
     The child cannot be admitted to school
If not
     The child can be admitted to school
```

2. Passing a subject:

```
If the mark scored is 35 or more
     It is a Pass
If not
     It is a Fail
```

3. Buying a book:

```
If you have money equal to or more than the price of the book
     You can buy the book
If not
     You cannot buy the book
```

4. Checking eligibility for voting.

```
Indian citizens get the right to vote after completing the age of
18.
If age is greater than or equal to 18,
     the person is eligible to vote
If age is less than 18,
     the person is not eligible to vote
```

5. Checking eligibility for marriage.

```
If you are boy then
```

```
    The age of eligibility for marriage should be more than 21.
If you are girl then
    The age of eligibility for marriage should be more than 18.
```

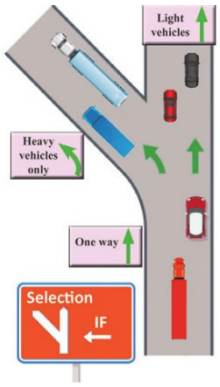


Fig. 2.9 : Example of selection

Assignment 2.6

1. Write down three real life situations which consist of selection.

2. In India, the age of eligibility for marriage of a boy is more than 21 years and for girls is more than 18 years. Select the most suitable word for the blanks given below.

If you are boy then

age of eligibility for marriage should be ..... (more/less) than 21

If you are girl then

age of eligibility for marriage should be ..... (more/less) than 21

Repetition

If one or several steps of an algorithm are repeated until a condition is satisfied, it is called repetition.

Examples

1. Let us consider the process of a class teacher marking the attendance register.

(1) Call the first name on the register

(2) Mark "P" if the student is present

(3) Mark "A" if the student is absent

(4) Call the name of the next student

(5) Repeat step (2) or (3) and (4) till the last name of the register is called

2. Let us consider the process of reading a paragraph and calculating the number of words you read.

(1) Read the first word of the paragraph

(2) Number of words = 1

(3) Read the next word

(4) Add 1 to the number of words

(5) Repeat step (3) and (4) till the end of the paragraph

(6) After reading the paragraph, indicate the number of words

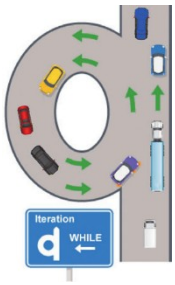


Fig. 2.10 : Example of repetition

Assignment 2.7

1. Write down two real life examples with steps that comprise repetition.

2. Fill in the blanks below related to repetition that output 5 times from 5 to 60.

a. n = 5

b. Output the value of n.

c. Add 5 to the value of n.

d. Repeat step number ..... and ..... till the value of n = 60.

2.4 ALTERNATIVE SOLUTION AND EFFICIENCY OF ALGORITHM

There can be more than one solution to a given problem, such solutions are called alternative solutions. Also, there can be more than one approach to solve a given problem. Hence there can be more than one algorithm.

Example 1. Let us examine the solution space to find the perimeter of a rectangle.

Let us analyse the input, process and output related to this problem.

Input: Length and width of the rectangle

Process: Calculating the perimeter

Output: Indicating the perimeter

Let us examine the solution space to calculate the perimeter.

Solution 1. Perimeter = length + width + length + width

Solution 2. Perimeter = length × 2 + width × 2

Solution 3. Perimeter = (length + width) × 2

All these three solutions are correct. Think of the difference between all these three solutions and decide which one is more efficient.

Example 2. Consider the problem of finding whether a given number is prime or not. There can be four different ways to write algorithms to check whether a given number is prime or not as below:

Solution 1. Starting with divisor 2, divide the given number (dividend) and check if there are any factors. Increase the divisor in each iteration and repeat the previous steps as long as divisor < dividend. If there is a factor, then the given number is not prime.

Solution 2. In *Solution 1*, instead of testing all the numbers till the dividend, only test up to half of the given value (dividend) because the divisor cannot be more than half of the dividend

Solution 3. In *Solution 1*, only test up to the square root of the dividend (numbers)

Solution 4. Given a prior list of prime numbers till 100, divide the given number by each number in the list. If not divisible by any number, then the number is a prime else it is not prime.

All these four methods can check if a given number is prime or not. In programming efficiency is an important concern and hence the selection of algorithms depends on the efficiency of the algorithm.

Solution 1, requires a large number of calculations hence requires more processing time, as it checks for all the numbers as long as the divisor is less than the number. If the given number is too large, this method will take more time to give the output.

Solution 2, is more efficient than *Solution 1*, as it checks for divisibility till half the number, and thus it reduces the time for computation of the prime number.

Solution 3, is even more efficient as it checks for divisibility till square root of the number, thereby further reducing the time taken.

As *Solution 4*, uses only the prime numbers smaller than the given number for divisibility, it further reduces the calculations. But in this method, we are required to store the list of prime numbers first. Thus, it takes additional memory even though it requires lesser calculations.

Hence, algorithms can be compared and analysed on the basis of the amount of processing time they need to run and the amount of memory that is needed to execute the algorithm. These are termed as time complexity and space complexity, respectively. The choice of an algorithm over another is done

depending on how efficient they are in terms of processing time required (time complexity) and the memory they utilize (space complexity).

2.5 CODING

Coding refers to the conversion of the algorithm to any high-level programming language by following the syntax of that programming language. Syntax is the set of rules or grammar that governs the formulation of the statements in the language, such as spellings, order of words, punctuation.

The programs are written in high-level programming language which are then converted to machine language or low-level language consisting of binary code with 0s and 1s which are understood by the computer. The high-level programming languages are close to natural languages and are easier to read, write, and maintain, but are not directly understood by the computer. A wide variety of high-level languages, such as FORTRAN, C, C++, Java, Python, are used by the programmer for software development.

A program written in a high-level language is called source code. This source code is translated into machine language using a compiler or an interpreter, so that it can be understood by the computer.

The selection of a programming language depends on the various factors such as the requirement of the client, the platform (OS) where the program will run, whether the application would be a desktop application, a mobile application or a web application.

Desktop and mobile applications are generally developed for a particular operating system and for certain hardware whereas the web applications are accessed in different devices using web browsers and may use resources available over cloud.

Besides, programs are developed not only to work on a computer, mobile or a web browser, but it may also be written for embedded systems like digital watches, mp3 players, traffic signals or vehicles, medical equipment and other smart devices. In such cases, we have to look for other specialised programming tools or sometimes write programs in assembly languages.

2.6 PROBLEM DECOMPOSITION

For a complex problem, the solution is not directly derivable. In such cases, we need to decompose it into simpler parts.

Let us look at the Railway reservation system. The complex task of designing a good railway reservation system is seen as designing the different components of the system and then making them work with each other effectively.

The basic idea of solving a complex problem by decomposition is to 'decompose' or break down a complex problem into smaller subproblems.

These sub problems are relatively easier to solve than the original problem. Finally, the sub-problems are combined in a logical way to obtain the solution for the bigger, main problem.

Breaking down a complex problem into subproblems helps each sub problem can be solved independently and by different programmers. Having different teams working on different sub problems can also be advantageous because specific sub problems can be assigned to teams who are experts in solving such problems.

There are many real-life problems which can be solved using decomposition. Examples include solving problems in mathematics and science, events management in school, weather forecasting, delivery management system, etc.

Once the individual sub problems are solved, it is necessary to test them for their correctness and integrate them to get the complete solution.

CHECK YOUR PROGRESS

A. Multiple choice questions

1. Which of the following symbol is used to represent start or stop process in the flowchart (a) oval shape (b) rectangle (c) parallelogram (d) diamond shape
2. Which of the following symbol is used to represent Input or Output in the flowchart (a) oval shape (b) rectangle (c) parallelogram (d) diamond shape
3. Which of the following symbol is used for decision making in the flowchart (a) oval shape (b) rectangle (c) parallelogram (d) Flow lines
4. Which of the following symbol is used for process in the flowchart (a) oval shape (b) rectangle (c) parallelogram (d) Flow lines
5. Which of the following is flow control statement (a) sequence (b) selection (c) loop (d) all of the above

B. Fill in the blanks

1. Algorithms is written by using the natural language like _____
2. The solution of problem represented in natural language is called as _____
3. In the modular approach, the project is divided into _____
4. Coding the program is the process of translating program into _____
5. The length of the program will depend upon the _____ of the problem

C. State whether true or false

1. Algorithm increase the reliability, accuracy and efficiency.
2. The steps in algorithm should be followed in strict sequential order.
3. The common methods to represent the algorithm are flowchart and pseudo-code.
4. A pseudo-code represents the basic functionality of program.
5. A program written in machine code is called as source code.

D. Short answer questions

1. What are the steps of problem-solving cycle?
2. What is modular approach for designing the solution of problem?
3. What are characteristics of good algorithm?
4. What is algorithm and flowchart?
5. State the guidelines to prepare the flowchart.
6. State the advantages of flowchart.
7. Give the real-life examples of sequence, selection and repetition.
8. What is meant by alternative solution?
9. List the five high level programming languages.
10. Draw the flowchart for the problems given in this chapter.

Session 3. Coding in Scratch

Scratch is a free educational block-based visual programming language. It makes coding fun and easy. It has a user-friendly interface with drag-and-drop functionality. It is the ideal tool for setting the first step into the world of programming. It is developed for children between the ages of 8 to 16. It is designed as an educational tool that enables children to produce projects.

Scratch allows users to create these projects and share them online. Scratch can be used in a web browser or downloaded to a web/mobile device.

Scratch makes it easy for students to create custom machine–learning models. We can learn coding in local languages like Hindi, Gujarati, and other regional languages.

Scratch is designed primarily for kids to learn the fundamentals of programming in a fun and interactive way. It is a great tool for beginners with no prior coding knowledge to get started with coding.

Scratch is a graphical and Python programming software that is the ideal companion for setting the first step into the world of programming. Its user-friendly interface and drag-and-drop functionality eliminate the need to memorize syntax and rules. As a result, they only require to focus on the problem at hand and develop skills such as logical reasoning and problem-solving.

Users can directly create projects on the official website https://scratch.mit.edu/ using a block-like interface.

3.1 Getting Started with Scratch

There are two ways to begin Scratch programming:

Method 1. Launch the Scratch Online Editor:

Open https://scratch.mit.edu/ in your browser and click on create. This will open the Scratch Online Editor. You can now begin creating scratch projects.

Method 2. Download the Scratch Offline Editor:

If you would like to create and save projects on Scratch without an internet connection you can download the Scratch Offline Editor also known as the Scratch App. Different versions of Scratch are available for download for both Windows and Mac.

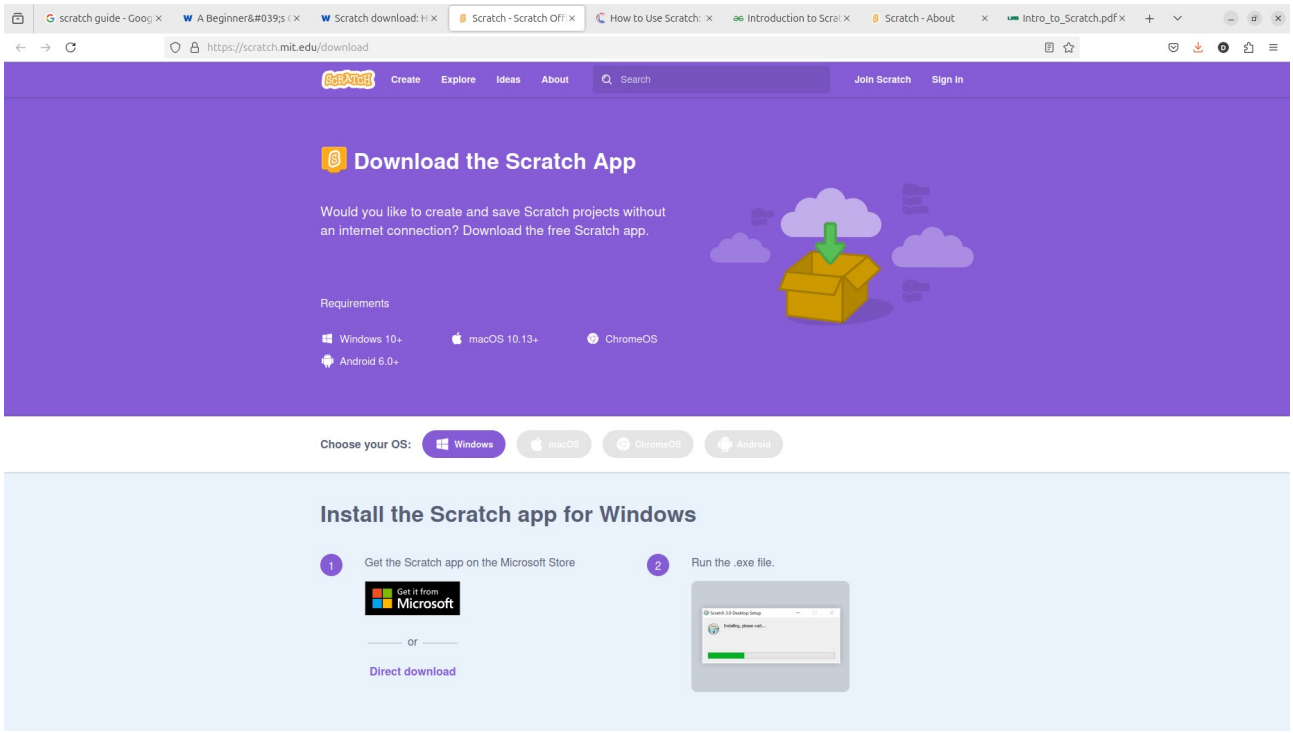
The steps by step process to download Scratch 3.0 is explained below:

Step 1. Visit scratch.mit.edu/download

The first step towards downloading the Scratch App is to click on https://scratch.mit.edu/download. As soon as, you click on the link given here, you will be directed to the official page of Scratch from where you can download Scratch 3.0.

Step 2. Choose your OS

As you can see from the image shown below, the message is pretty straightforward regarding download requirements. You have to choose your OS from all the 4 given options: Windows, macOS, ChromeOS, and Android.



Step 3. Install the Scratch App for Your OS

Click on your choice of OS. We have chosen Windows for this. As you can see, after clicking on Windows, it gives us 2 options: either to get the application from the Microsoft store or download it directly to your system.

Step 4. Running the Application

As you can see in the image shown below, Scratch advises users to run the application in order to complete the installation process. Go through your downloads and run the application by granting permission to it. The installation will take 3–4 minutes, depending on your system's compatibility. After the installation is complete, head over to sign in or Scratch login, if you already have an account, and start creating projects!

Creating Scratch account

The process of creating an account on the Scratch platform is known as a Scratch account. Either a teacher or student can set up or create their own

account to learn to code, save and share Scratch projects and even become part of many online communities, called Scratch Studios.

Types of Scratch Accounts

There are two types of Scratch accounts and they are:

1. ***For Students:*** To help children over the age of 8 learn about coding through fun and interactive projects.
2. ***For Teachers:*** To help children learn about coding from experienced coders who want to teach.

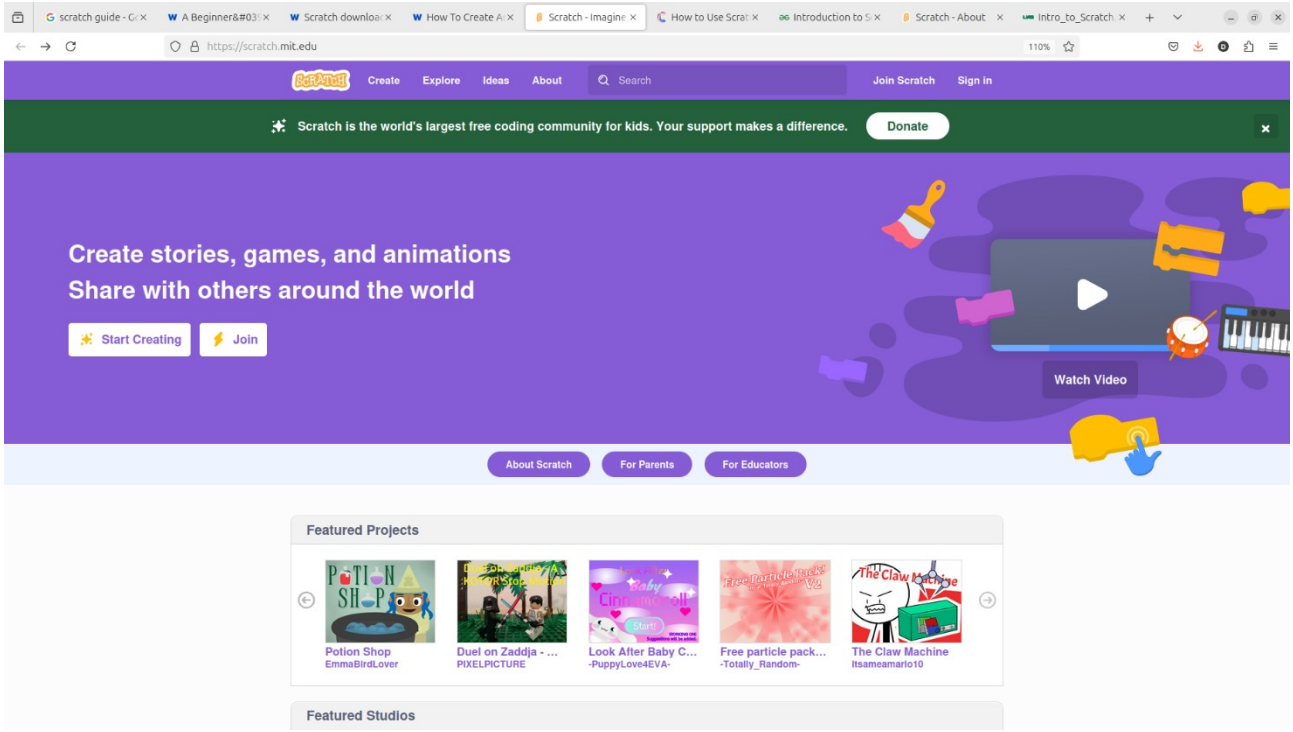
Steps to create your own Scratch account.

Step 1. Open scratch.mit.edu

Visit the Scratch official website for creating an account. Follow the link – https://scratch.mit.edu/

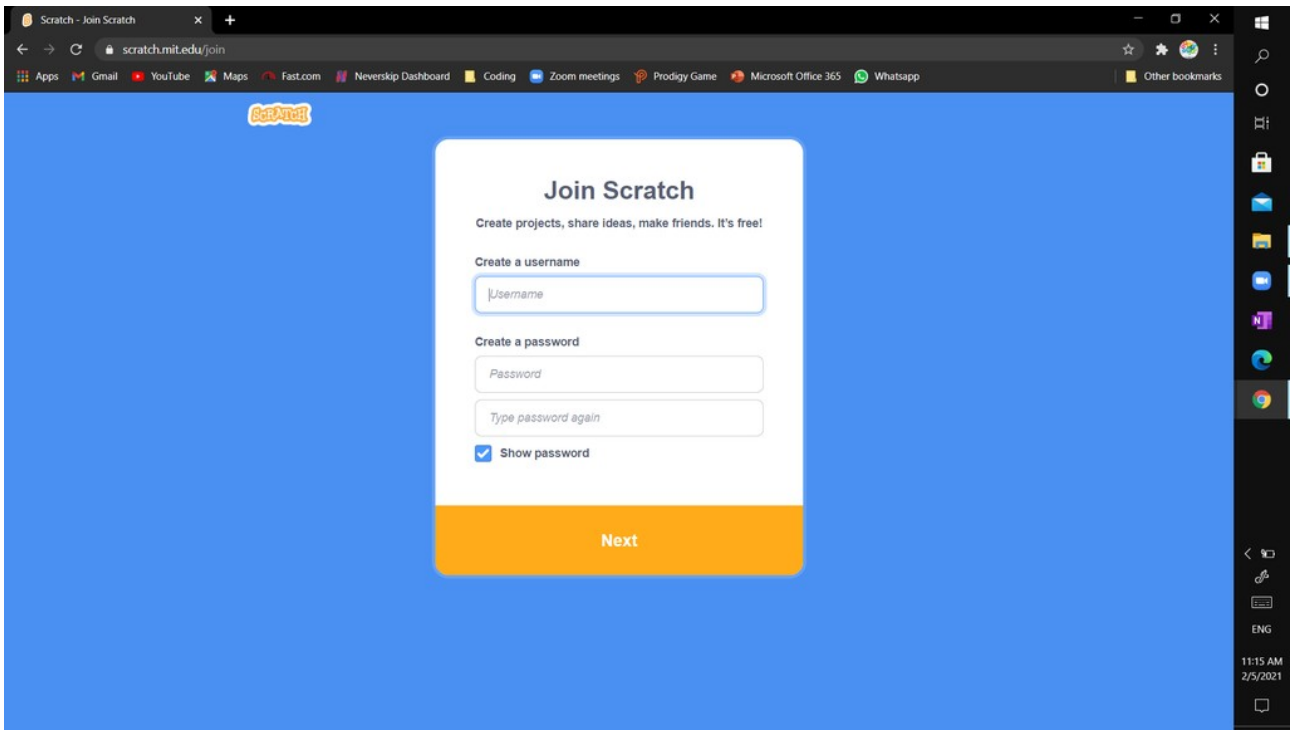
Step 2. Click on "Join Scratch"

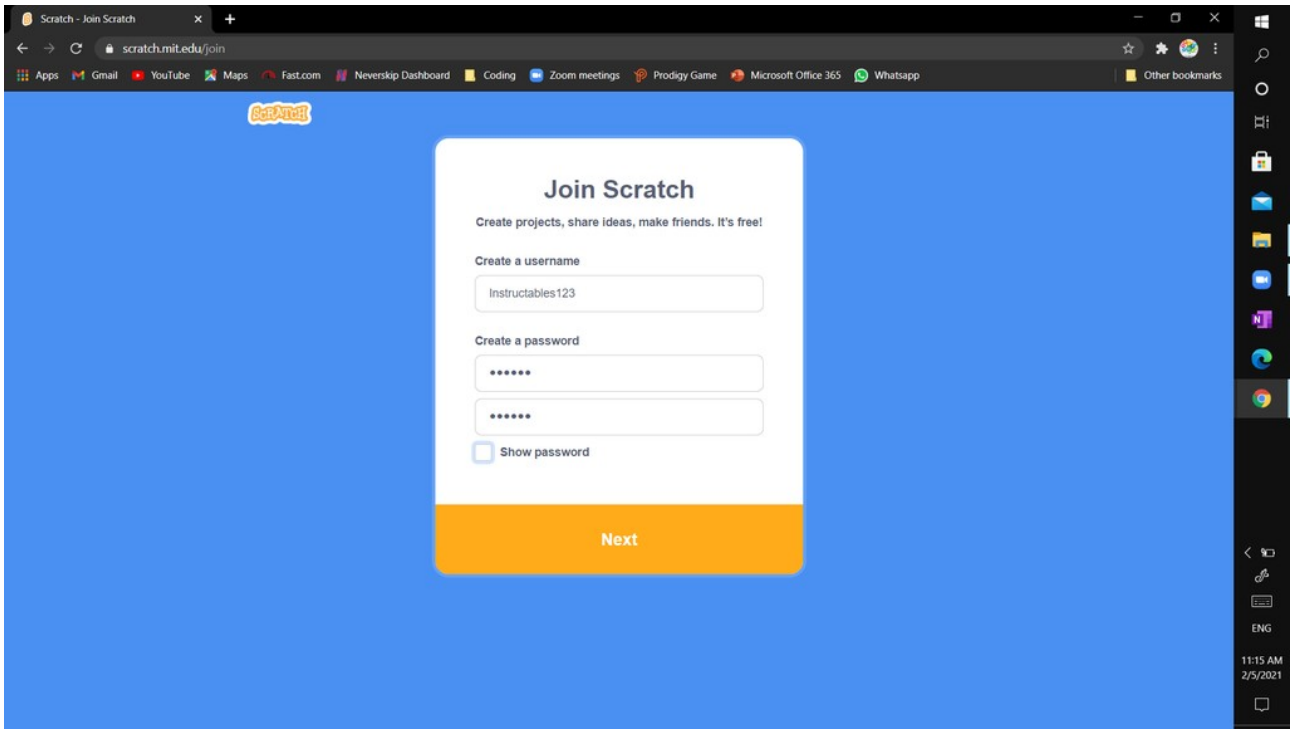
You will now see the Scratch homepage. You'll see a "Join Scratch" icon on the top-right corner of the page, as shown in the image below. Just click on that.



Step 3. Set up a "Username" & "Password"

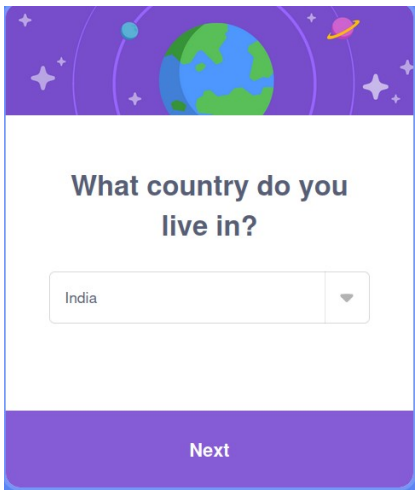
After you have clicked on the "Join Scratch" icon, it will take you to a window where you will have to enter a new username and a strong password. Scratch recommends not using your real name as a username, so get creative!





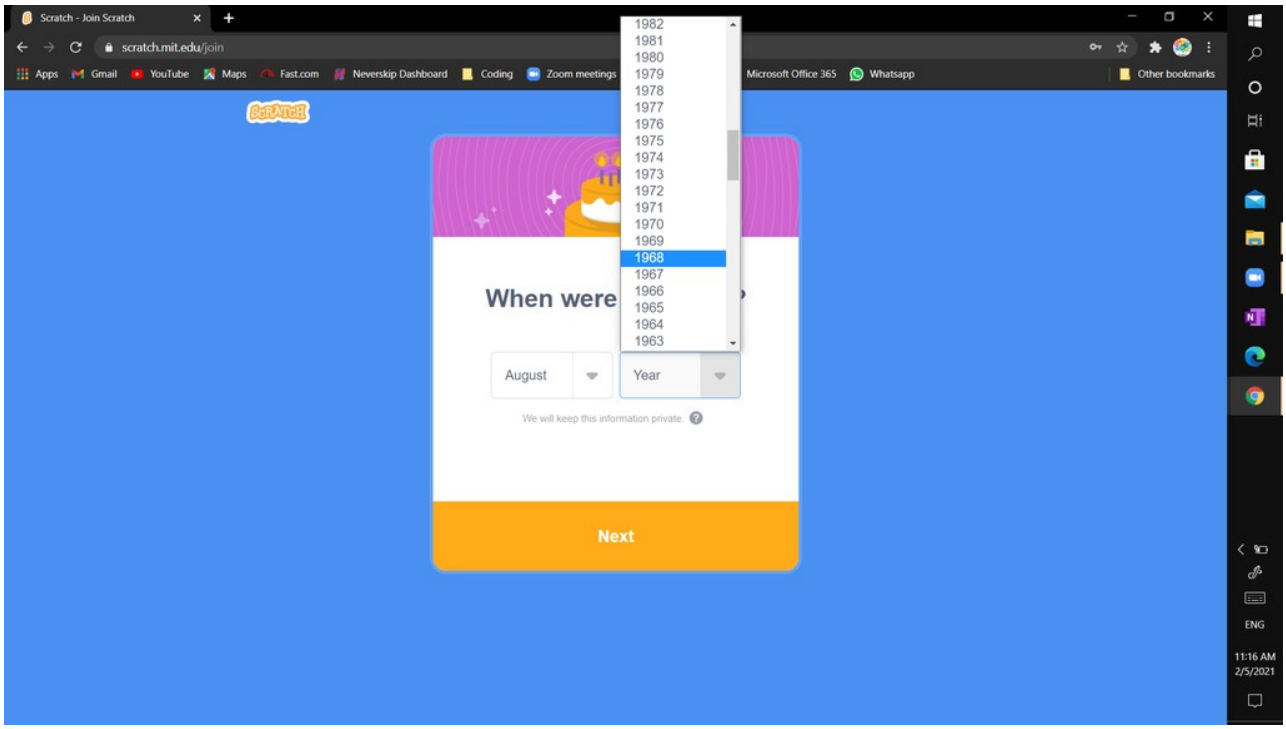
Step 4. Select your country

On the next couple of screens, you will have to put in your personal details. Don't worry, we will guide you through that. First, it will ask you what country you reside in. You have to select your country from the drop-down menu options.



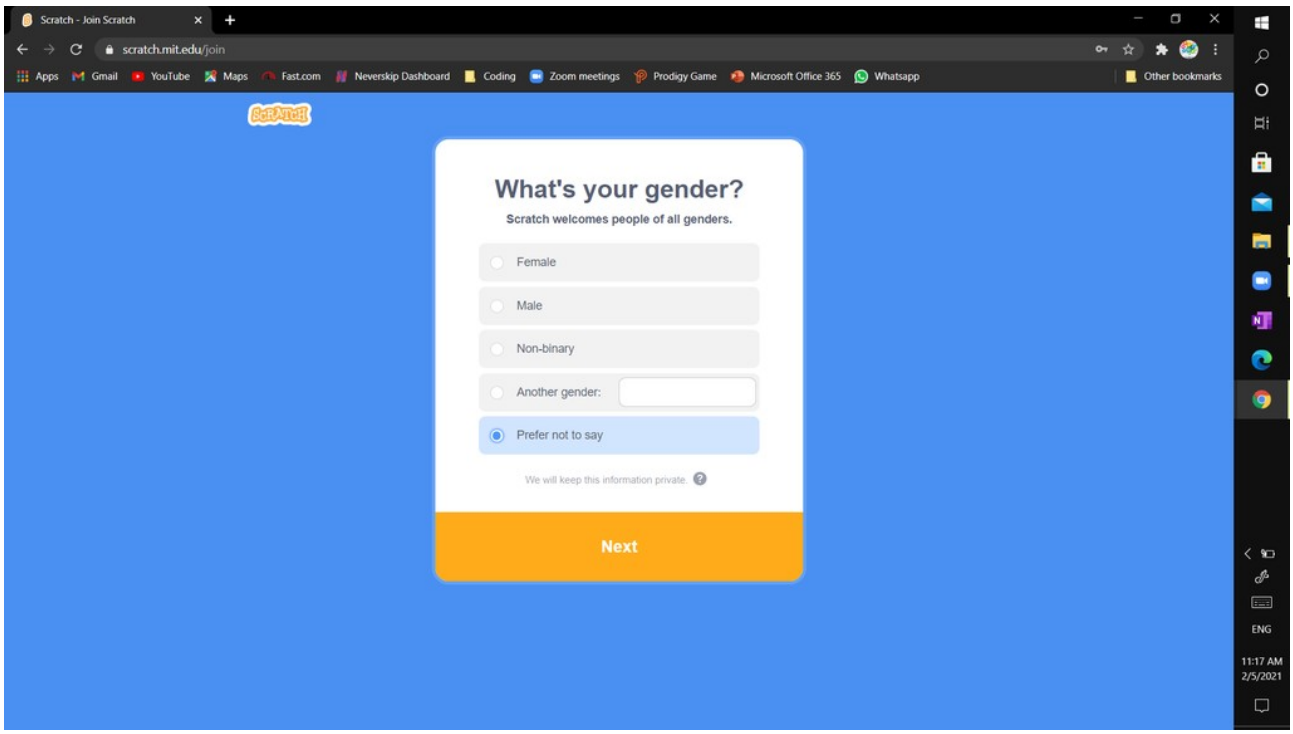
Step 5. Enter your birthday

Next, it will ask for your birthday. Again, you have to just select the month and year from the drop-down menu option.



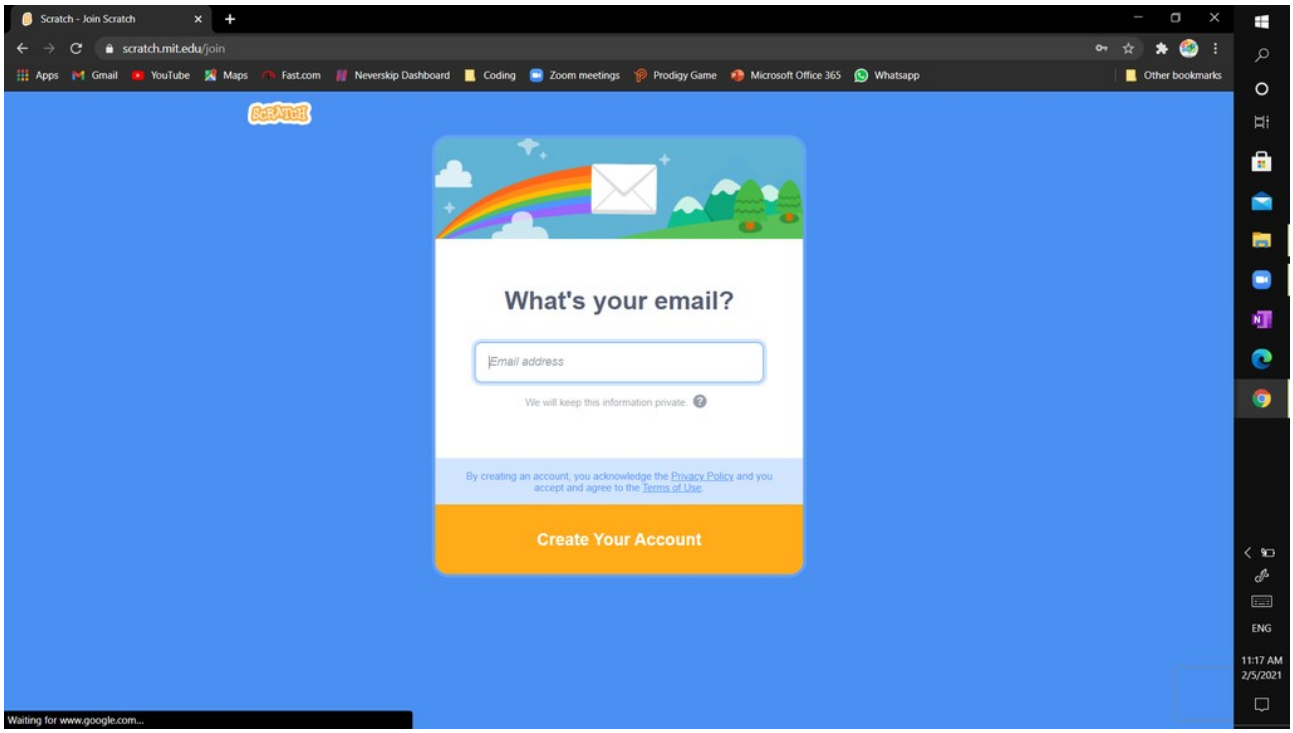
Step 6. Select your gender

Then you have to select your gender.



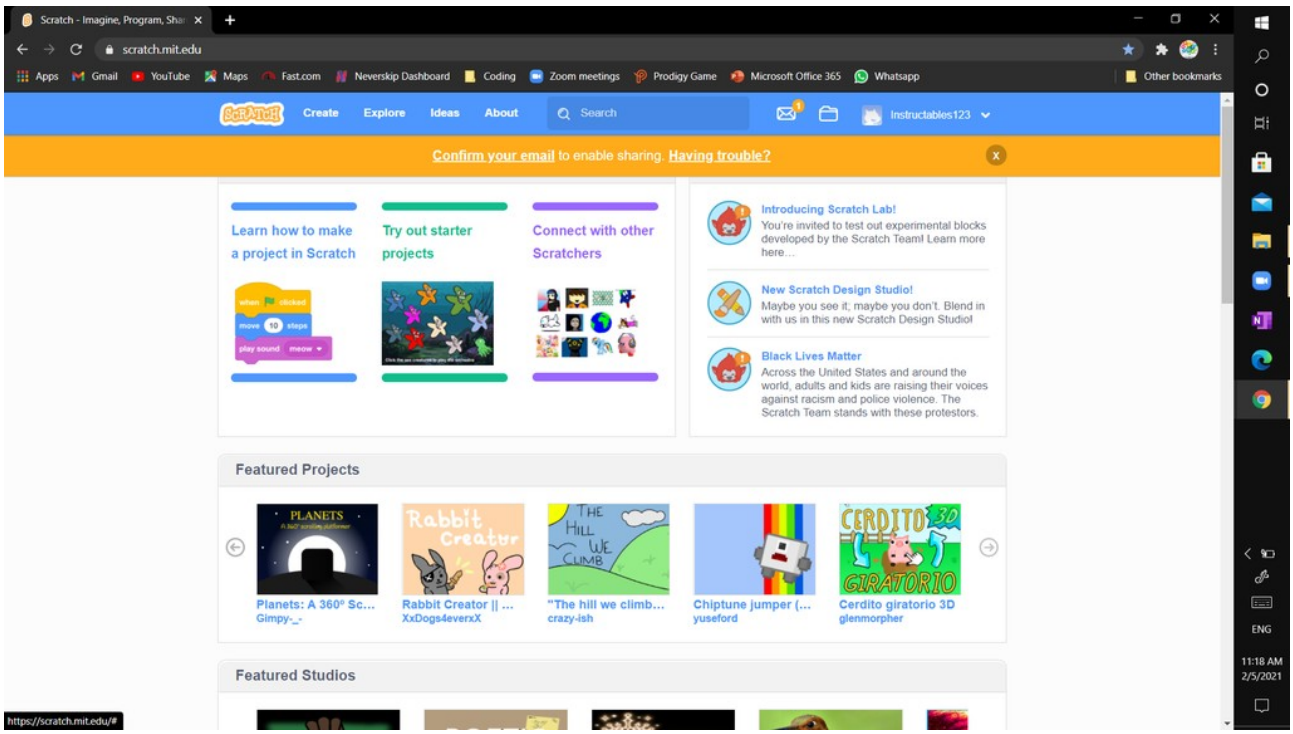
Step 7. Enter an email address

Then it will ask for your email ID. Enter an email address that you want to be associated with your Scratch account.



Step 8. Welcome to your Scratch account

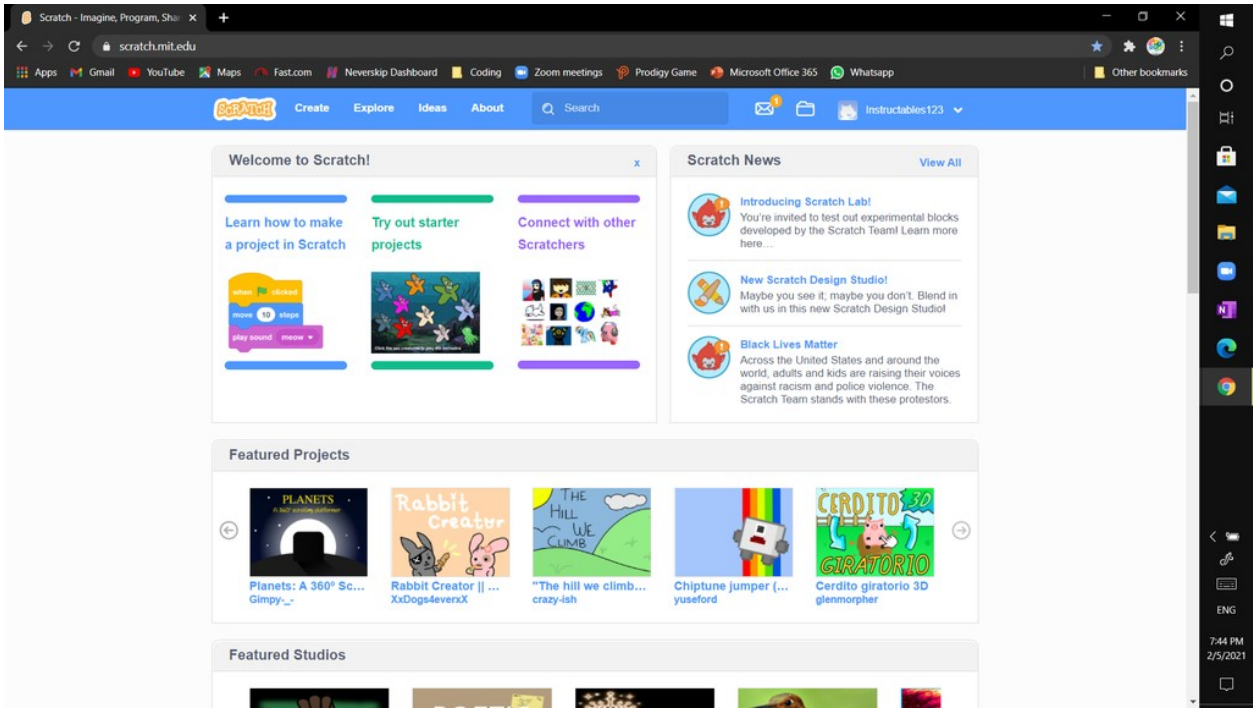
After that, you have to click on “Next.” It will show you that now you are logged in & can start creating projects! Click on “Get Started” for the next step.



Step 9. Confirm your email address

On the next screen, you'll see a notification that if you enable sharing, then you have to confirm your mail-id. Click on "confirm your mail".

Open Gmail Sign in with your account which you have used to sign in the scratch account.



At the top Ribbon bar you can see a change before and after signing in. The **Join scratch** and **Sign in** are gone. You may see the following extra options.

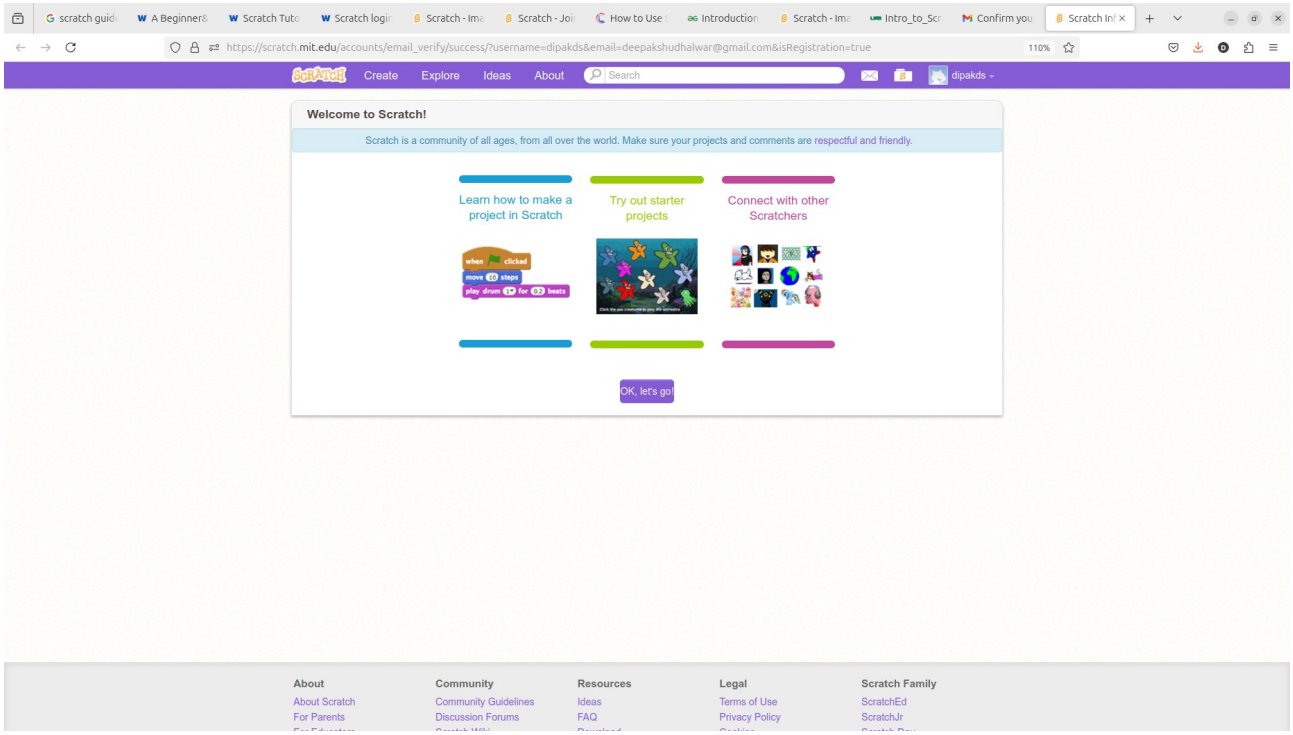
- *Messages (Icon is a mail icon)* – Here you can see all you scratch messages

- *My stuff (Icon is a file icon)* – Here you can see all your created project and studios
- *Profile icon* – Here you can see your profile name and your profile picture. If you click on it, you can see some more options like profile, my stuff, account settings and sign out

Step 10. Start creating & sharing projects

You will now see a window pop-up saying you will just have to head to your email & confirm your ID. If you haven't received the confirmation mail, then you also have the option of resending the confirmation mail.

After you have confirmed your mail, get started on your first new project & share it with your peers!



Creating a teacher's Scratch account

If you want to teach students about Scratch, you can create a teacher account and become a teacher in four simple steps.

Step 1. Go to the homepage at scratch.mit.edu.

Step 2. Click on the "**for educators**" icon.

Step 3. After that, you'll have to create the account by entering a new username and a strong password.

Step 4. After this, Scratch will verify your profile and approve you as an educator, which will take a day or two. Then you can only be a teacher and set up classes and add students to your classes.

Logging into new Scratch account

Steps to login your Scratch account are as below.

Step 1. Visit the official website of Scratch – scratch.mit.edu.

Step 2. Here you can see the homepage with the "sign-in" option in the right-hand corner of the screen. Click on it.

Step 3. In the pop-up window, enter your newly created credentials, username & passwords.

Step 4. Click on the "Sign-in" button and you are logged in to the Scratch account.

Scratch Online Editor

The Scratch Online Editor is the interface you use to build your games, animations, and stories in Scratch programming. When you launch the Scratch editor through the web browser, the default Scratch programming interface will open up.

Scratch programming interface contains:

***Code Tab*:** This is the main code editor and default tab of the Scratch online editor. This section is used to access all the code blocks such as Motion blocks, Looks Blocks, Control Blocks, Operator Blocks, and so on.

***Costumes Tab*:** The costume tab opens up a costume editor. This is a place where you can make and modify sprites. Scratch has tons of in-built sprites and you can always import a new one from your drive or make something completely new.

***Sounds Tab*:** Sounds tab opens up a sound editor. This is a place where you can make and modify sounds. Scratch has tons of in-built sounds that can be used to make games and animations. You can always import something new from your drive or create something entirely new.

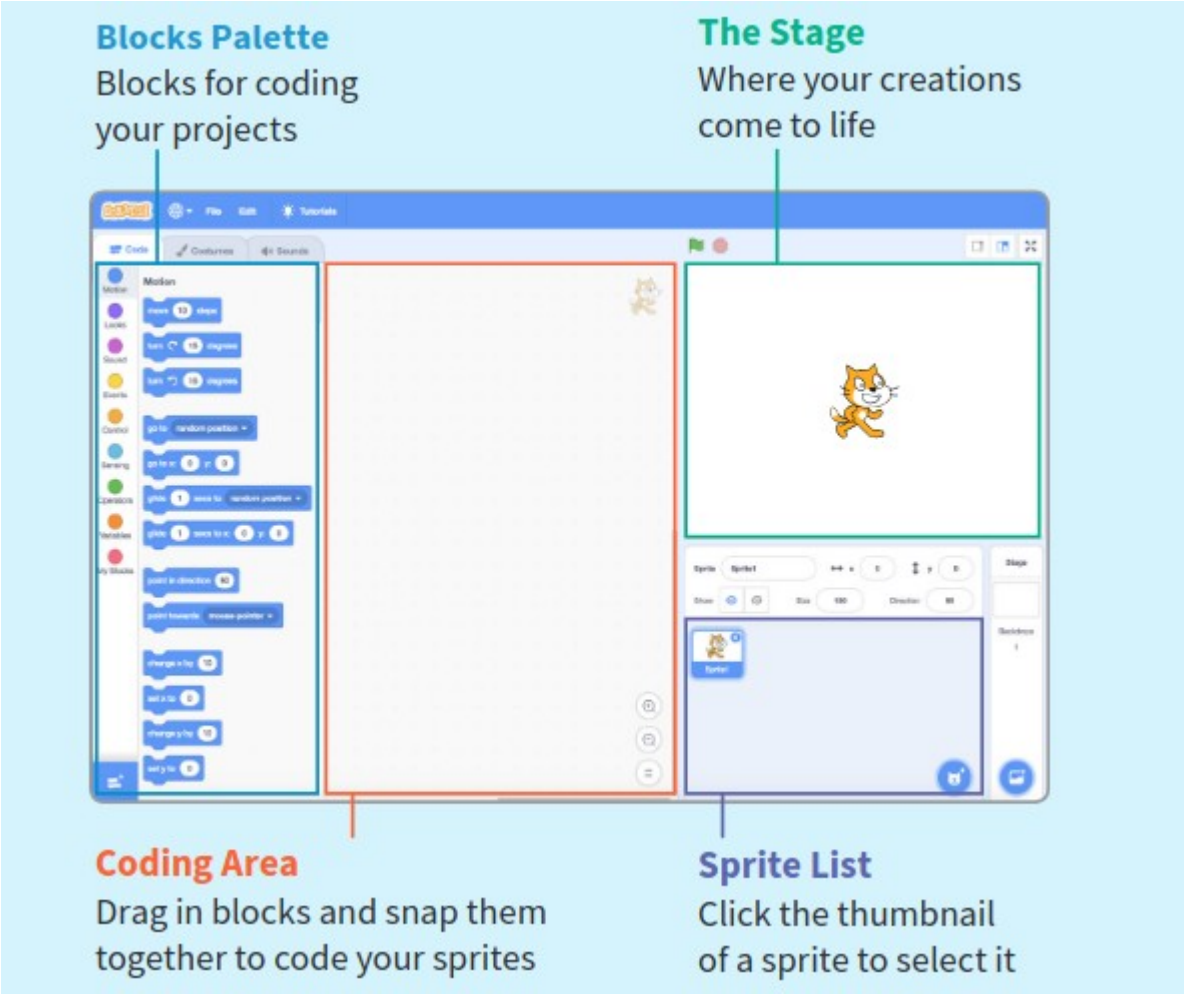
***Green Flag (Start Button)*:** This is used to start the code execution and see the result on the preview panel.

Red Dot (Stop Button): This is used to stop the code execution.

Components of Scratch Online Editor

When you launch the Scratch editor you will see the editor interface comprising of different sections as shown in the image below:

There are four main components of Scratch: Block Palette, Code Area, Stage, Sprite Pane.



Block Palette

The Block Palette is the area on the left of the screen when the Code button is pressed. On the left, there is an area that contains the nine-block categories in Scratch. These blocks can be dragged into the Code Area to make scripts.

Code Area

The Code Area is the large space to the right of the Block Palette. It is an area for storing blocks that run the project. Blocks can be dragged from the Block Palette into the Code Area and arranged to form scripts. Each sprite has its own script, and sprites can be selected using the Sprite Pane.

Stage

The stage is the area where the sprites are displayed and perform their actions. It is located at the top of the area to the right of the Code Area.

Sprite Pane

The Sprite Pane is located to the right of the Code Area and at the bottom of the stage. It allows one to quickly switch sprites and view the different scripts each sprite contains.

Menu Bar or The Header

Menu Bar is used to open new projects, edit existing projects, and save and share them with other Scratchers. The menu bar comprises of the following elements

- Scratch
- Translate
- File menu
- Edit Menu
- Tutorials button
- Project Name
- Share button
- See Project Page
- My stuff
- Profile

Scratch Components

Scratch project is the generic term used to refer to any program written in Scratch such as a game or a story or animation or any other application. Any Scratch project is built using three fundamental components:

1. Sprites
2. Backdrops
3. Code Blocks

Let us understand each of these building components of Scratch programming one by one

Sprites in Scratch Programming

Sprites are the key objects (images) that are programmed in Scratch. Sprites can range from animals, people, objects, and things like sports, music, food, and fashion. Scratch has a rich library of inbuilt sprites which come attached with a set of code blocks. For example, a drum kit sprite can be used to play different levels of bass, high tom, and low tom.

You can also edit an existing sprite, import one from your drive, or make a new one using the costume editor. Sprites can be selected from the Sprite library as shown below:

Backdrops in Scratch Programming

Backdrops are creative backgrounds that you can use for your game or story. You can choose from an existing backdrop, paint a new one, or upload from an

existing file on your computer or drive. When a backdrop is added to a scratch project, it appears on the stage.

Scratch has tons of in-built backdrops across categories such as fantasy, sports, outdoors, indoors, space, underwater, patterns, and much more.

Code Blocks in Scratch Programming

Scratch Code blocks are used to perform actions on sprites in a game or story. Scratch provides code blocks such as motion blocks, which can be used to set a sprite into motion, or sound blocks, which can be used to add sound effects to a sprite, or event blocks, which are used to define when the code is run.

Code blocks can be used by dragging them into your project to create a script. A collection of code blocks that interlock with each other to perform a certain action or story or game is called a script.

Scratch provides 9 types of code blocks:

1. Motion Blocks
2. Looks Blocks
3. Sound Blocks
4. Event Blocks
5. Control Blocks
6. Sensing Blocks
7. Operator Blocks
8. Variable Blocks
9. My Blocks

Let us look at the coding blocks one by one:

1. Motion Blocks

These are used to move the sprite on the stage. Let us look a some of the popular motion blocks

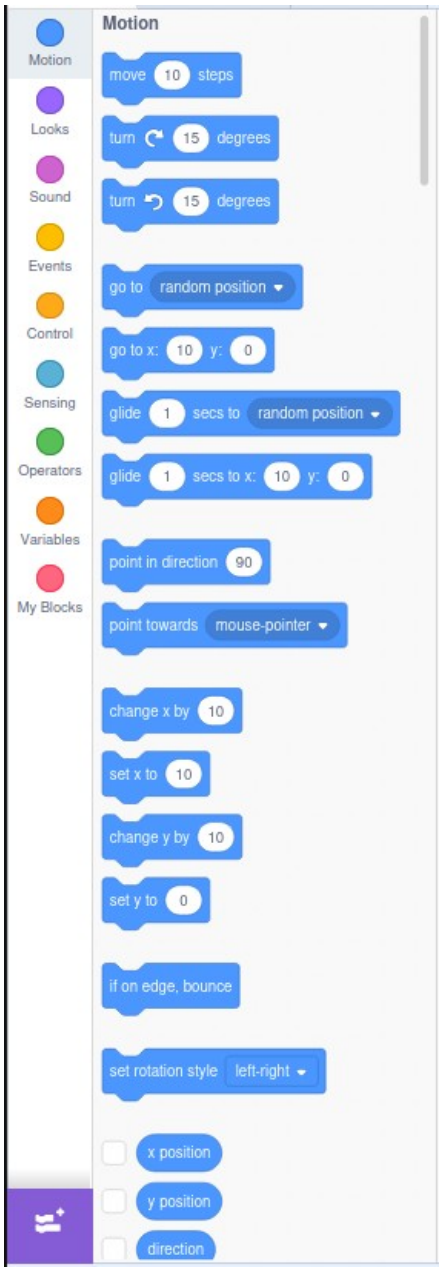
Move steps block: this block is used to move the sprite by a finite number of steps.

Turn degrees blocks: these are used to turn the sprite by a specified angle in a clockwise or anti-clockwise direction.

Go to block: is used to move the sprite to a specific position on the stage

Glide to blocks: is used to move the sprite to a specific position using a gliding movement

If on edge, bounce block: This is one of the most popular motion blocks used in game design. If the sprite hits any of the edges on the stage it will bounce back onto the stage.



2. Looks Blocks

These are used to define how the sprite and background will look. Let's look at some of the popular Looks blocks

Say & think blocks: Say blocks can be used to represent speech by sprites

Switch costume block: This is used to change the costume of a sprite. This block comes in very handy to denote the motion of a sprite such as a bird flapping its wings where each costume can be defined as a stage of motion in the bird's flight

Change effect by block: This block is used to change the color, brightness, or pixelation of the sprite

3. Sound Blocks

These are used to add sound effects to a sprite e.g.

Play sound block: will play selected sound example a “magic spell” sound until done

Change pitch effect block: is used to change the pitch by specified points

Set volume to block: is used to set the sound effect volume to the desired level.

4. Events Blocks

Events blocks are used to define the triggers when the script should run. All the other blocks will have no meaning till an event block is used to define when a script is to be run. Some examples of events and associated blocks are as below:

When the green flag clicked block

When key pressed block

When backdrop switches block

Broadcast message and wait block

5. Control Blocks

These are the logic blocks of Scratch allowing power over when and how the script is run e.g.

Repeat block: is used to define that the code within the repeat block will run a specified number of times (10 times in the example below)

If then else block: This is a conditional block and is used to perform a certain action only if the required condition is met or else perform the alternative action defined under the else block.

Wait until block: This is a pause block in Scratch programming. It can be used to pause code execution for a defined time period or until a certain condition is met.

Repeat until block: This block will loop until the specified statement is true in which case the code beneath the block, if any, will be executed.

Repeat until block: This block will loop until the specified statement is true in which case the code beneath the block, if any, will be executed.

When I start as a clone block

This control block will be activated by the clones when clones are created.

Create clone of () block

This control block creates a clone of the sprite in the argument.

Delete this clone block

This control block deletes the clone it runs in. Besides clicking on the green flag or stop sign, this block is the only way to delete clones.

6. Sensing blocks

Sensing blocks are used in Scratch programming to identify and measure how different objects (sprites & colors) within a game or story are interacting with each other or to detect certain keyboard and mouse movements. For example:

Ask & wait block: This block can be used to ask for input from the user such as the name and store this information. Later the stored information can be retrieved and displayed on the screen.

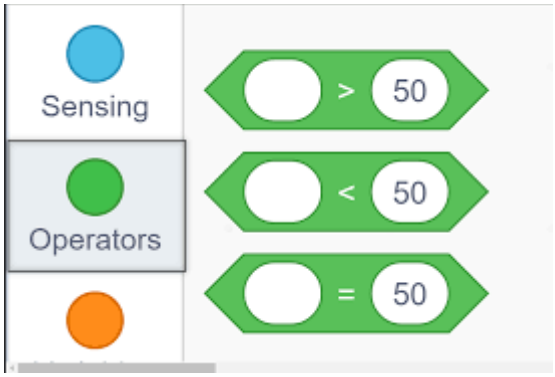
Touching color block: This block can be used to check if a sprite is touching the specified color. This is very useful in game development where actions can be defined if a collision occurs between two sprites

Distance to block: This block is used to calculate and report the distance in pixels between objects (sprites).

7. Operator Blocks

Operator blocks are used to:

- Perform Arithmetic functions such as addition, subtraction, multiplication & division.
- Perform value comparison
- Perform string operations such as joining two strings, calculating the length of a string, or identifying characters within a string.



8. Variable Blocks

Variable blocks are used to make variables and set, change and report values of variables. Variables are temporary stores of value in a program. Let's look at the key variable blocks.

Set to block: This is used to set the initial value of a variable. This can be a numeric value or string.

Change by block: This block is used to change the value of a variable by a specified value.

Show variable block: This block can be used to show the variable on the stage.

Hide variable block: This is the opposite of the show variable. It is used to hide the variable value from the stage.

9. My Blocks

This feature is used to create new custom blocks (called procedures) which can be created using a combination of standard Scratch blocks. The custom procedures can be saved in Scratch and used directly in a script. This saves the time and effort of duplicating blocks of code if the same sequence of actions is to be performed multiple times within a script.

Scratch Projects

Scratch project is the generic term used to denote any creation such as a game, story or animation created in Scratch. Let's look at Scratch projects in detail:

What is a Scratch Project

Any creation, such as a story, a game, or an animation, is a Project in Scratch. The process of making Scratch projects is called "Scratching". Projects are made using the Scratch Editor through the use of Scratch components such as sprites, backdrops and code blocks. Projects can also be saved and shared once you set up your account in Scratch.

Starting A Scratch Project

You can start a scratch project simply by clicking on "Create" in the navbar of the Scratch home page. This will immediately take you to the Scratch editor where you can start coding your project. You can choose the default Cat sprite in the editor or replace it with another sprite to get started.

Downloading A Scratch Project

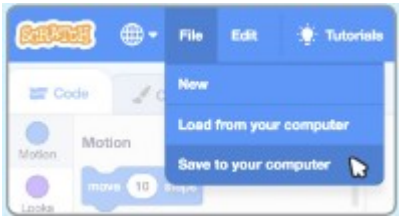
You can also save a Scratch project offline on your devices such as a laptop or a desktop computer. Click on the "Save to My Computer" button in the file menu on the scratch editor. This will download the project on your device in the default downloads folder. You can change the folder manually to a folder name and location that you want. The scratch project will be saved as a .sb3 file.

Saving and Sharing Your Scratch Projects

Scratch online editor auto-saves the work that you do. It is however always a good practice to manually save the project to prevent any loss of work. You will need to be logged in to Scratch to save a project. The easiest way to save a

project is to save it online by clicking on “Save Now” located in the top right corner of the scratch online editor.

Click the File menu and select **Save to your computer**. To open a project, click **Load from your computer**.



Further Scratch allows Scratchers to share their saved project with other Scratchers. A key prerequisite is that the projects must be made in Scratch only and not on any Scratch modifications.

The steps to share a project are as follows

Step 1. Go to the My Stuff page from the Scratch editor by clicking on the <> icon.

Step 2. The My Stuff page will open with a list of projects. Select and click on the required project to be shared.

Step 3. Add instructions for other users on how to experience your project (game/story/animation).

Step 4. Include notes and credits about how you created your project, as well as if you borrowed ideas, scripts, or artwork from other users. It is always good practice to credit users whose work you have built upon.

Step 5. Click on the “**share**” button on the orange bar above the project.

Voila!!! Your project is now available for the world to experience. A unique link is also auto-generated for the shared project, which you can copy and share with your friends. Scratch also generates an embed code that you can insert into your website to list the project on your website. Read our detailed step-by-step guide on how to save and share a project on Scratch here.

Note: Projects cannot be shared from the offline editor. A shared project is viewable to all Scratchers, even if they are not signed in. A shared project can be used by anyone to remix the project.

Unsharring & Delete Your Scratch Project

The steps to delete a Scratch project are as follows:

Step 1. Go to the My Stuff page from the Scratch editor by clicking on the <> icon

Step 2. My Stuff page will open with a list of projects

Step 3. Scroll down until you see the project/projects you want to Delete

Step 4. If your project is shared it must be first unshared before deleting. Click on the "unshare" button next to the project then click "delete"

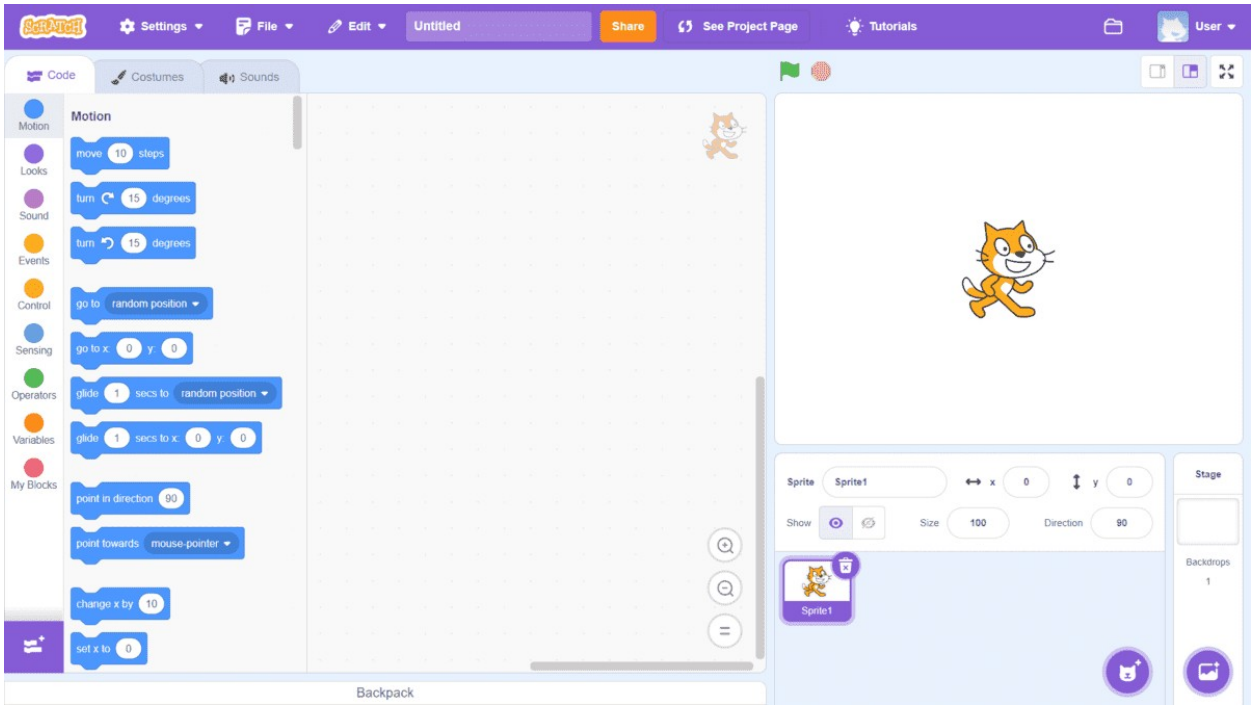
Step 5. As with other software to permanently delete the project, go to "trash" and click on the "Empty Trash" button. You will be prompted to verify your account password to permanently delete the projects.

Note: You cannot recover the projects you have permanently deleted. Scratch recommends using Contact Us with an explanation if you want to recover a permanently deleted project as the Scratch Team can still recover it.

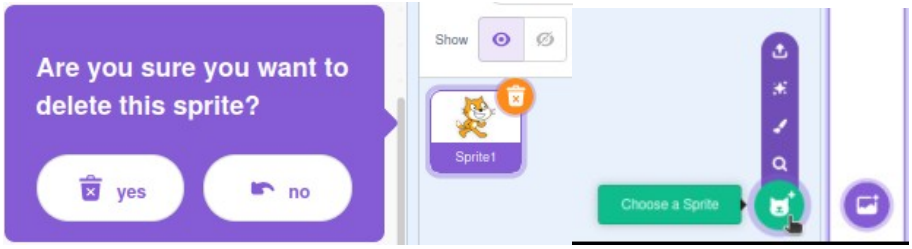
Creating Your First Scratch Project

Now that we have a good understanding of Scratch, and its various components such as Sprites, Backdrops, and Code Blocks, let us create our first Scratch Project:

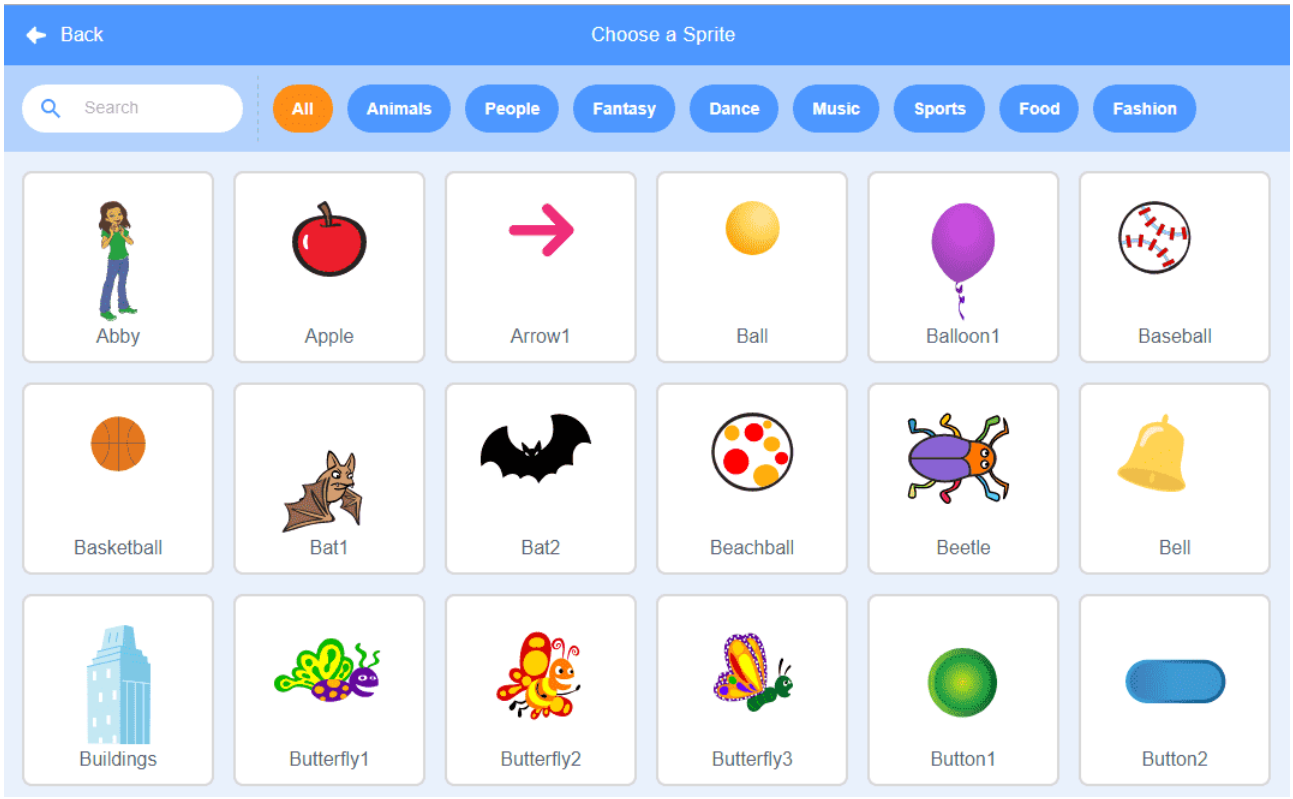
Step 1. Open the Scratch editor. You will see an empty code area and a default cat sprite in the code area.



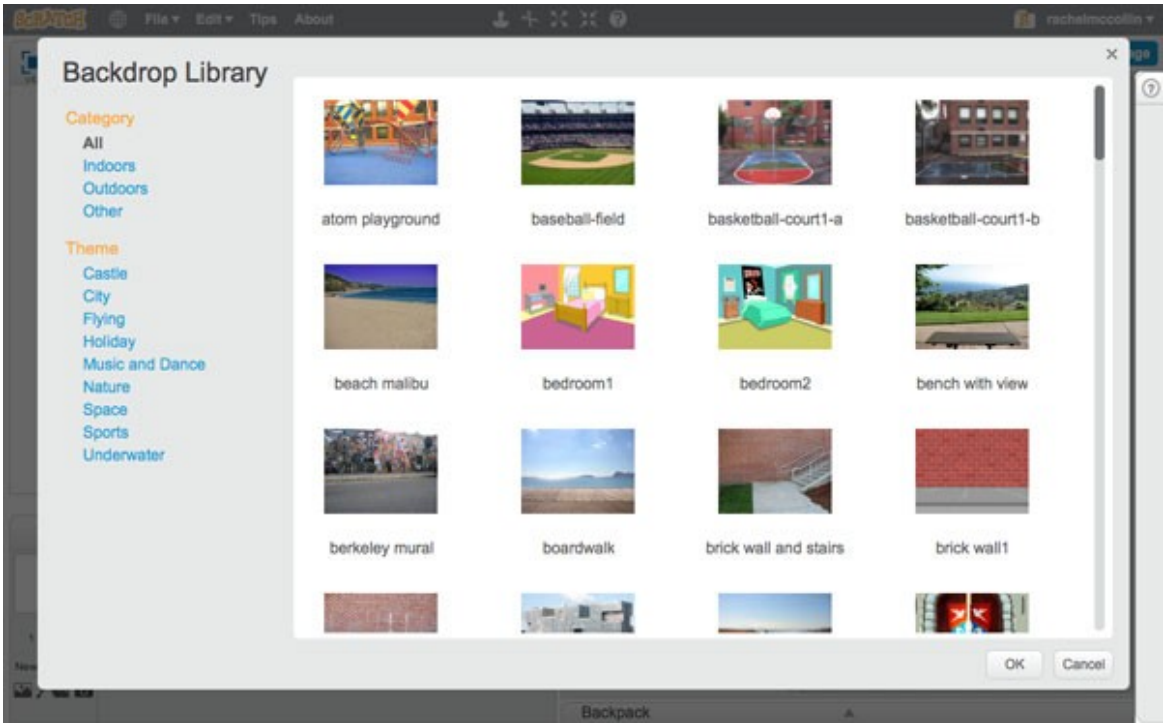
Step 2. Choose a sprite for your project.



When Scratch opens the default, sprite is the Cat sprite. Delete the Cat sprite and select another sprite from the sprite library. We will select a fish sprite.

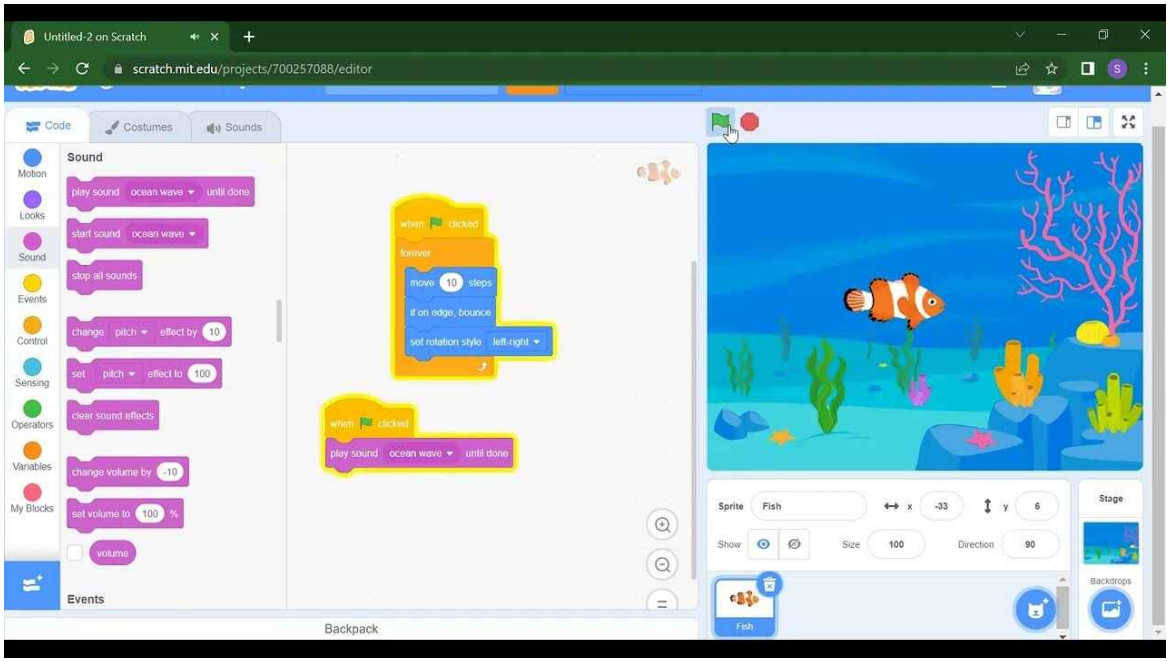


Step 3. Add a backdrop to your project. Go to the backdrop selector and select the "underwater 2" backdrop.



Select "underwater 2" backdrop from the backdrop selector

Step 4. Add code to your project. Select the fish sprite and Drag the "when green flag clicked block" to the code area. This is the control event that will trigger the execution of the code. Now add code to the fish sprite to move forward 100 steps. Drag the "move steps" block from the motion blocks and set it to 100 steps.

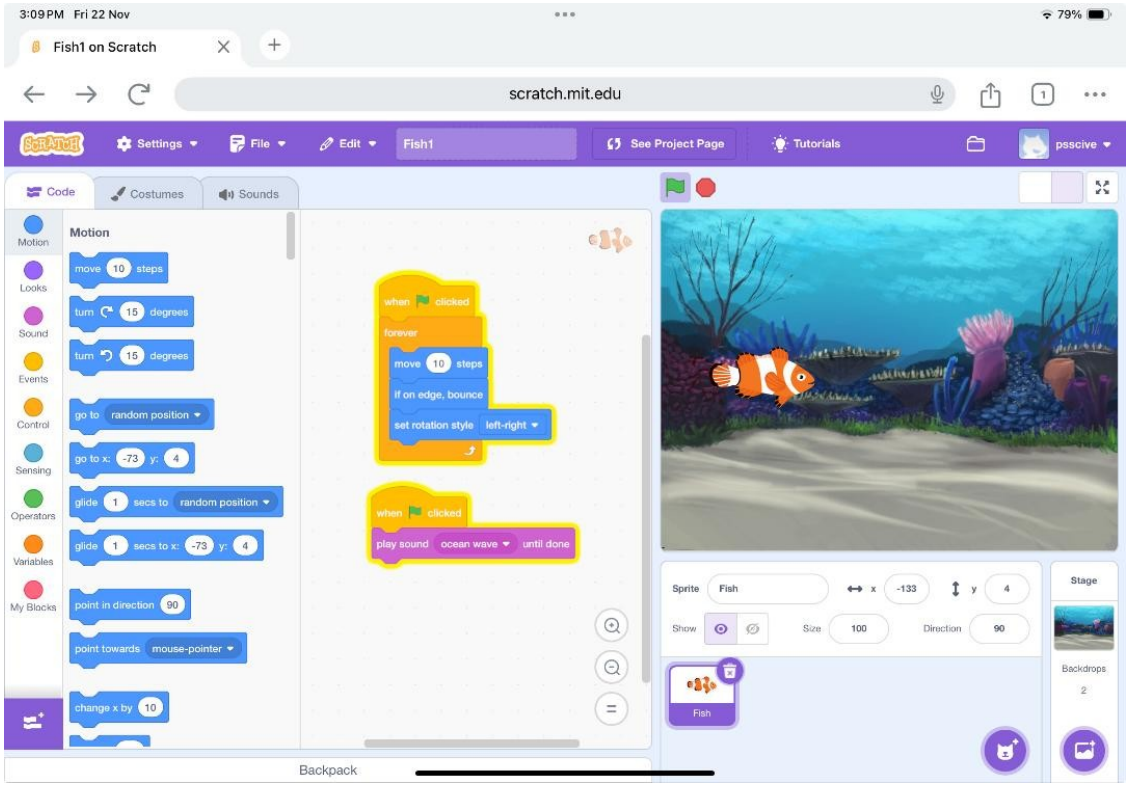


Select fish sprite and drag move steps block

Step 5. Now Add sound to your project. Drag the "start sound" block from the sound blocks and set it to "ocean wave". In Scratch, every sprite comes with its predefined list of sounds. For fish sprites, these include "ocean waves" and "bubbles". You can also add custom sounds to a sprite.

Step 6. Execute the project by clicking on the flag. After clicking on the flag the project gets executed and the output is as shown below.

Step 7. Save your project. If you are signed in online the project will get saved online else it will prompt you to save on your computer.



Check your progress

A. Multiple choice questions

1. The official website of scratch is (a) https://scratch.com (b) https://scratch.mit.edu/ (c) https://scratch.in (d) https://scratch.edu
2. Scratch accounts can be created by (a) Students and Parents (b) Students and Teachers (c) Teachers and Parents (d) Teachers and School
3. In Scratch, which of the following tab is used to start the code execution and see the result on the preview panel (a) Code tab (b) Costumes tab (c) Sound tab (d) Green flag
4. In Scratch, which of the following tab is used to make games and animations. (a) Code tab (b) Costumes tab (c) Sound tab (d) Green flag
5. Scratch project is built using the fundamental components (a) Sprites (b) Backdrops (c) Code Blocks (d) All of the above

B. Fill in the blanks

1. In Scratch, Code tab is used to access all the _____.
2. In Scratch, _____ are the key objects used for coding.
3. Scratch provides ____ types of code blocks.
4. Looks blocks are used to define _____.
5. Scratch has a rich library of _____ sprites.

C. State whether true or false

1. Scratch is a free educational block-based visual programming language. (T)
2. Scratch is a graphical and Python programming software. (T)
3. In Scratch, the Code tab is a place to make and modify sprites. (F)
4. In Scratch, Red Dot is a Stop button used to stop the code execution. (T)
5. In Scratch you cannot import or use a new sprite. (F)

D. Short answer questions

1. List the 9 types of code blocks in Scratch.
2. State the use of Motion Blocks in Scratch
3. State the use of Looks Blocks in Scratch
4. State the use of Sound Blocks in Scratch
5. State the use of Event Blocks in Scratch
6. State the use of Control Blocks in Scratch
7. State the use of Operator Blocks in Scratch
8. State the use of VariableBlocks in Scratch
9. State the use of Sensing Blocks in Scratch
10. State the use of My Blocks in Scratch

Module 2. Basic Python Programming

Python is a very simple programming language, developed by **Guido van Rossum** in 1989. It is named after the comedy television show Monty Python's Flying Circus and not after the Python snake. It is easy to learn to all who are new to programming.

Python is a dynamic, interpreted (bytecode-compiled) language. It is platform independent, free and open source, easy to learn, achieve the more tasks through less coding, fast in execution. There are no type declarations of variables, parameters, functions, or methods in source code. This makes the code short and flexible.

Python is a general-purpose language, which can be used in various domains including: Web applications, Big data applications, Data science, Desktop software, Mobile apps, Machine learning and AI. The SQL can be used for querying data from relational databases.

In our daily life we may require different tools to perform different tasks such as scissor for cutting paper, screw driver for tightening of screw, removing the hook, scaling the fish or making use of the scissors. If you can get one tool to perform plenty of task then that will be the most preferred tool. Swiss knife is such a multi-functional tool having various tools such as blade, opener, screwdriver, reamer and others one in all. In the same way Python is such a programming language which is object oriented, functional programming and also declarative programming language.

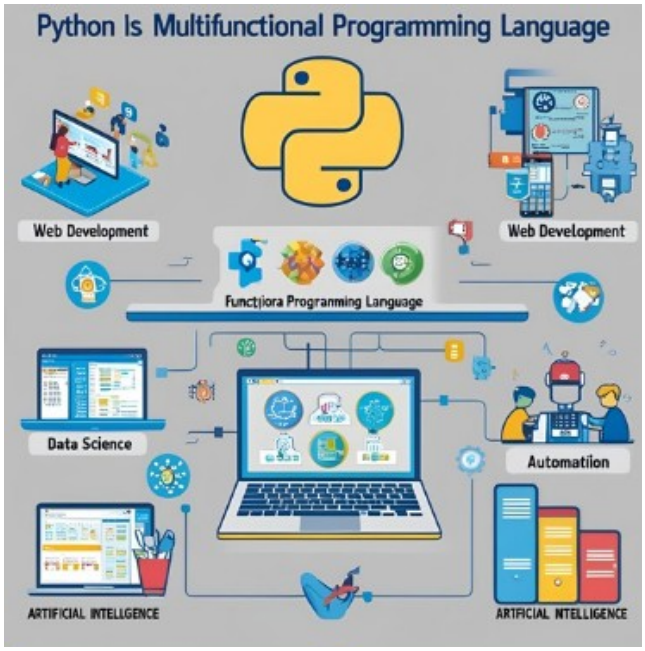


Fig. : Python: A Multifunctional Programming Language

Session 1. Introduction to Python

In the previous unit, you learnt about the different methodologies for programming. A programming language is a set of grammatical rules for instructing a computer to perform specific tasks. It is used code the programs in computer to perform a specific task to produce the desired output. There are many different programming languages such as BASIC, Pascal, C, C++, Java, Ruby, Python, and many others.

In this session you will understand the basics of Python programming, its features, and installation of Python. You will also learn the Keywords, Identifiers, and Variables in Python. The Data Types, Operators, and Statements in Python is also explained. You will be able to run the sample code in Python interpreter as well as write the programs in IDLE and run to get the output.

1.1 Introduction to Python Programming Language

A computer program is a set of instructions written in any programming language to perform a specific task after its execution by a computer.

The first version of Python was labeled as **"0.9.0"**. The first official version **1.0** was released in January 1994. The current version of Python is **3.13**, released on October 7, 2024.

Python for AI

Artificial intelligence (AI) is the most recent technology. Many AI applications are being developed in various area. There are various programming languages like Lisp, Prolog, C++, Java and Python, which can be used for developing applications of AI. It is the robust programming language used in many domains from web applications, data analysis, data science, machine learning, and AI. Python is the most suitable language because of the following features.

Features of Python programming

Python is an interpreted, high-level, general-purpose programming language. It allows solving complex problems in less time and fewer lines of code. It is simple and easy-to-learn. Python can be used to build all types of applications ranging from small and simple scripts to complex machine learning algorithms. Python source code is available under the GNU General Public License (GPL).

Readable and Easy to Learn – Python is a very readable language. Python programs are easy to understand as they have a clearly defined syntax and relatively simple structure. It is an interpreted language, as Python programs are executed by an interpreter.

Interactive Mode – Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.

Case Sensitive – Python is case-sensitive language, means, the variables "NUMBER" and "number" are different in Python. Python uses indentation for blocks and nested blocks.

Free and Open Source – Python is a free and open source programming language. It can be downloaded from its official website for free to use.

Cross platform – Python is portable and platform independent. It can run on various operating systems such as Mac, Windows, Linux, Unix and any hardware platforms. This makes it a cross platform and portable language.

Large standard library – Python comes with a large standard library of predefined functions that has some handy codes and functions which can be used while writing code in Python. Python is also helpful in web development. Many popular web services and applications are built using Python.

Supports exception handling – Python supports exception handling which means we can write less error prone code and can test various scenarios that can cause an exception later on.

Portability and Compatibility – Python can run on a wide variety of operating systems and hardware platforms, has the same interface on all platforms.

Extendable. We can add low-level modules to the Python interpreter. These modules enable programmers to customize their tools to be more efficient.

Automatic memory management – Python supports automatic memory management which means the memory is cleared and freed automatically.

Applications of Python

Python is used for a large number of applications as below:

1.2 Working with Python

Python is a cross-platform programming language, means, it runs on multiple platforms like Windows, MacOS, Linux and has even been ported to the Java and .NET virtual machines. It is also available online in cloud environment.

To write and run Python program, we need to have Python interpreter installed in our computer.

Practical Activity 1.1. Install the Python compiler (binaries)

Resources required

A Desktop or Laptop computer with operating system installed, Internet Connectivity

Procedure

Step 1. Download Python from its official website *https://www.python.org/downloads*. Select appropriate download link as per Operating System [Windows 32 Bit/64 Bit, Apple iOS].

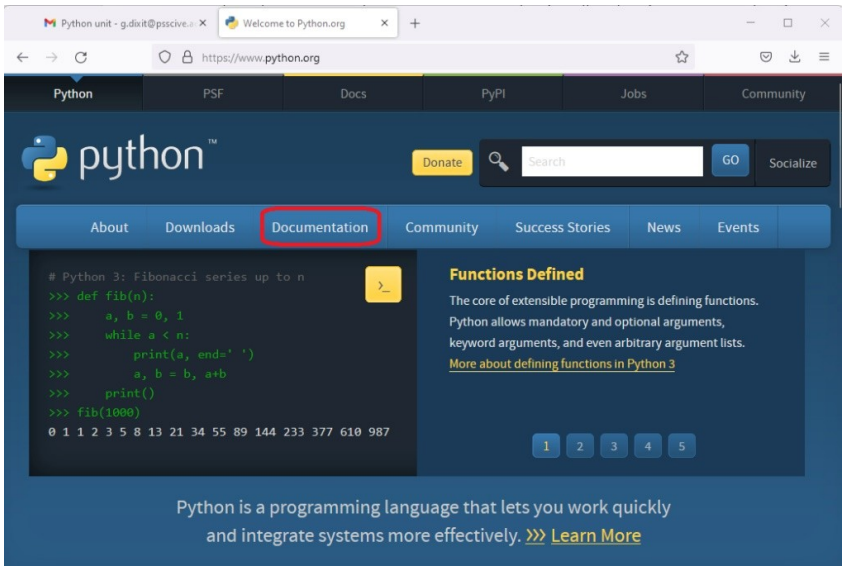


Fig. 1.1 Official website of Python

Step 2. For Windows 64 Bit OS, download latest version using the link

https://www.python.org/ftp/python/3.13.2/python-3.13.2-amd64.exe

Step 3. Download the installer. Once it is downloaded, run the Python installer.

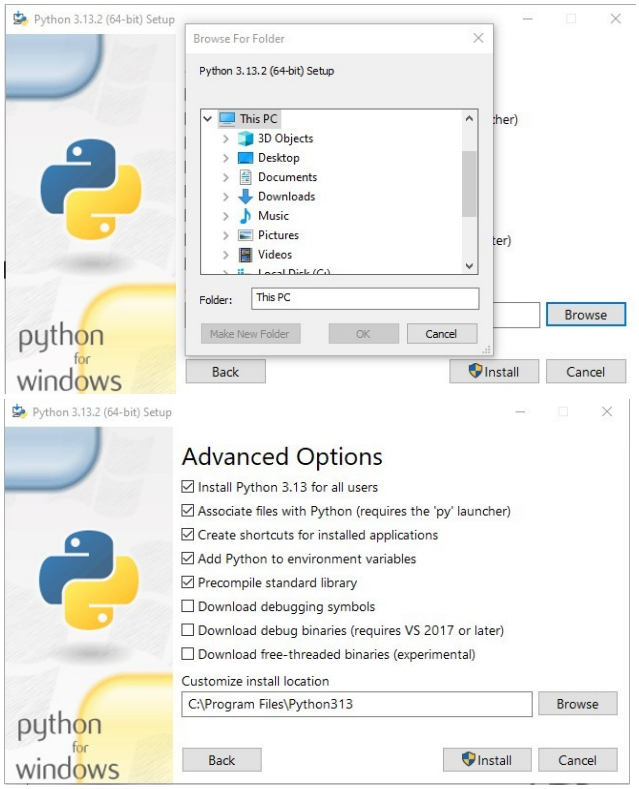
Step 4. Check the **Install launcher for all users** check box. Further, you may check the **Add Python 3.13.2** to path check box to include the interpreter in the execution path as shown in Figure 1.2 (a).

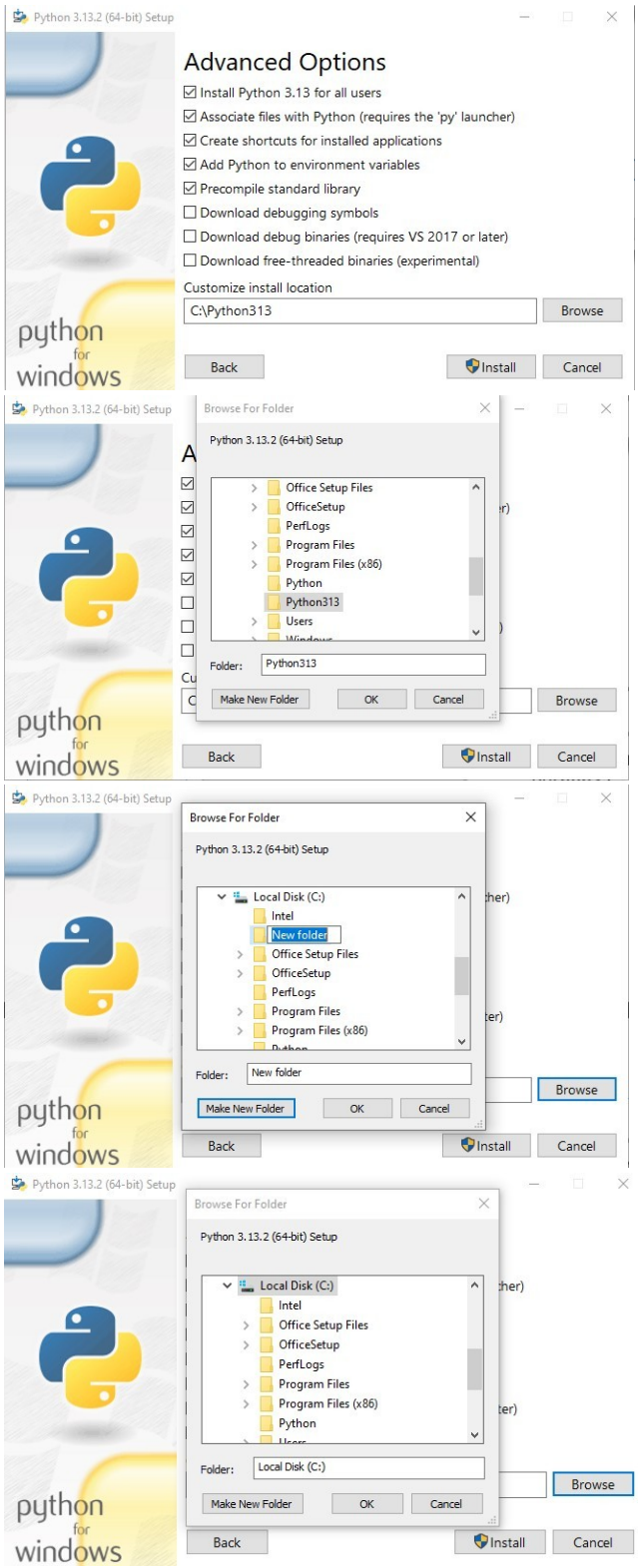
Step 5. Select **Customize installation**.

Fig. 1.2 : Installation of Python

Step 6. After selecting the Advanced options, click **Install** to start installation. Follow the instructions to complete the installation process as shown in Figure 1.3.

Step 7. Once the installation is over, a window will appear showing the **Python Setup Successful**.




Python 3.13.2 (64-bit) Setup
Advanced Options
Install Python 3.13 for all users
Associate files with Python (requires the 'py' launcher)
Create shortcuts for installed applications
Add Python to environment variables
Precompile standard library
Download debugging symbols
Download debug binaries (requires VS 2017 or later)
Download free-threaded binaries (experimental)
Customize install location
C:\Python313
Browse
Back
Install
Cancel
python for windows
Browse For Folder
Python 3.13.2 (64-bit) Setup
Office Setup Files
OfficeSetup
PerfLogs
Program Files
Program Files (x86)
Python
Python313
Users
Folder:
Make New Folder
OK
Cancel
Local Disk (C:)
Intel
New folder

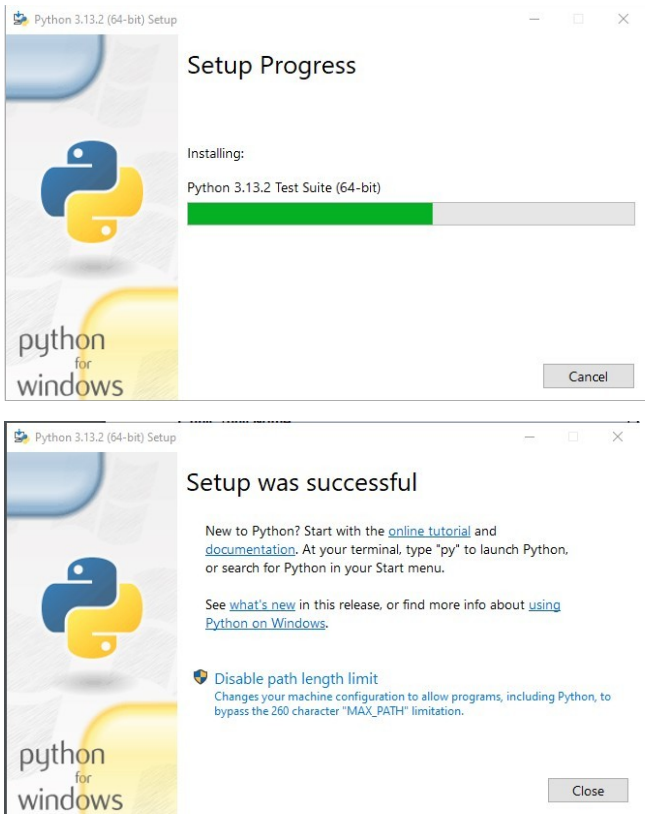


Fig. 1.3 : Installation of Python

Run in the Integrated Development Environment (IDE)

After installing Python, you can start developing Python programs. An alternate way to reach python is to search for *"Python"* in the start menu and clicking on IDLE (Python 3.13.3 64-bit). You can start coding in Python using the Integrated Development and Learning Environment (IDLE**)** as shown in Figure 1.4.

IDLE is Python's Integrated Development and Learning Environment. It is coded in Python itself, using the tkinter GUI toolkit. It is cross-platform i.e. it works mostly same on Windows, Linux, and MacOS. Python shell window (interactive interpreter) with colorizing of code input, output, and error messages.

1.3 Python Interpreter Mode

Python interpreter can be used in two modes – ***Interactive mode*** and ***Script mode***. Interactive mode allows execution of individual statement instantaneously. Whereas, the script mode allows writing more than one instruction in a file called Python source code file that can be executed.

1.3.1 Interactive Mode

To work in the interactive mode, simply type Python statement on the **>>>** prompt directly and press ***Enter*** key. The interpreter executes the statement and displays the result(s), as shown in Figure 1.4. Working in the interactive mode is convenient for testing a single line code for instant execution. But in the

interactive mode, we cannot save the programming code for future use and we have to retype the statements to run them again.

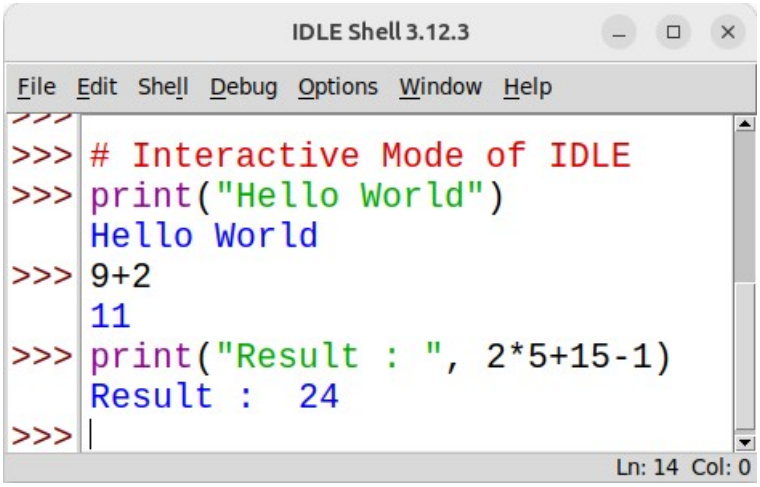


Fig. 1.4 : Interactive mode of IDLE

You can see the above example, Python IDLE Shell account has **>>>** as Python prompt, where simple mathematical expressions and single line Python commands can be written and can be executed simply by pressing enter.

The first statement print ("Hello World") written on the first Python prompt shows Hello World as output in the next line.

The second expression 9+2 written on the second Python prompt shows 11 as output in the next line.

The third statement print("Result:", 2*5+15-1) written on the fourth Python prompt shows

Result : 24

Assignment

Find the result of (75+85+65)/3 i.e. the average of three marks

Find the result of 22/7 * 5 * 5 i.e. the area of circle having radius as 5

Find the result of "RAVI"+"Kant"

Find the result of "###" * 3

1.3.2 Script Mode

In the script mode, you have to write a Python program in the editor and save the file with ***.py*** extension. Then use the interpreter to execute it by entering the file name on the python prompt (>>>). By default, the Python program is saved in the Python installation folder. Let us see steps to run Python interpreter in script mode.

Working in interactive mode is easy to test small pieces of code immediately. But for coding the large programs it is always better to use the script mode, so that you can save and modify the code whenever required.

Note: Result produced by Interpreter in both the modes, viz., Interactive and script mode is exactly the same.

Python Script/Program: Python statements written in a particular sequence to solve a problem is known as Python Script/Program.

To write a Python script/program, open a new file using **File >> New File**, type a sequence of Python statements for solving a problem, save it with a meaningful name using **File >> Save**, and finally run the program using **Run >> Run Module** to view the output of the program.

Following steps illustrate to write your first python program.

Step 1. Create a new file by clicking on **File > New File** menu in IDLE as shown in Figure 1.5.

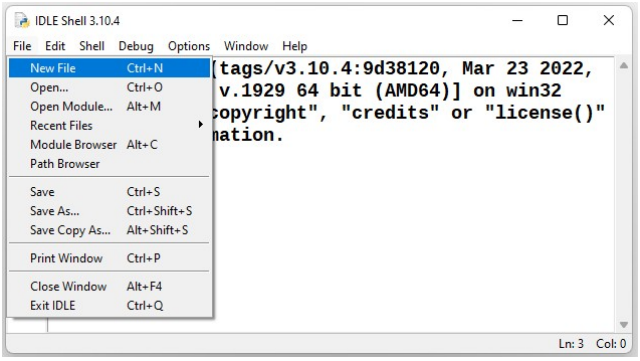


Fig. 1.5 : Creating a new file in Python

Step 2. A new file will open in editor. Enter the Python code as shown in Figure 1.6.

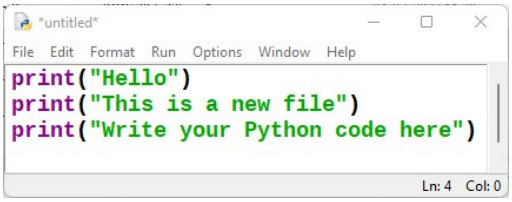


Fig. 1.6 : Coding a Python program

Step 3. To save the file, click on **File > Save** menu option. The **Save as** window will open as shown in Figure 1.7. Select the folder in which you want to save the file. Type the file name and click on Save button as shown in Figure 1.7.

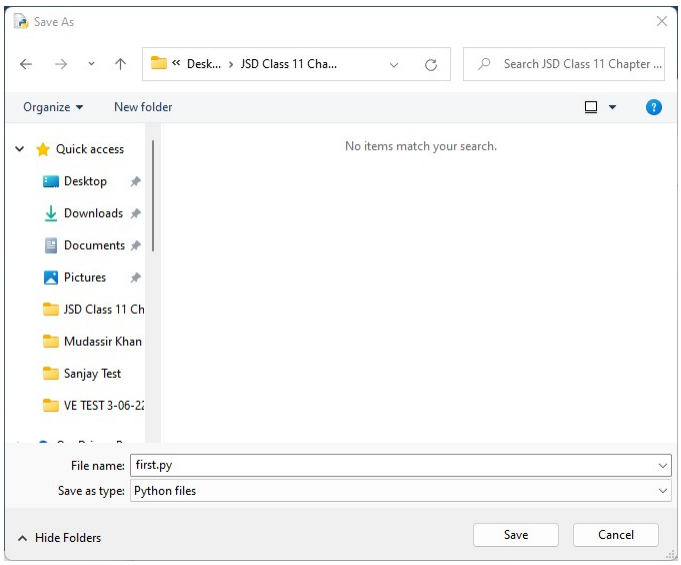


Fig. 1.7: Saving a Python program

Step 4. After saving the file, click **Run Module** from the Run menu as shown in Figure 1.8. You can also press **F5 key** to run the program.

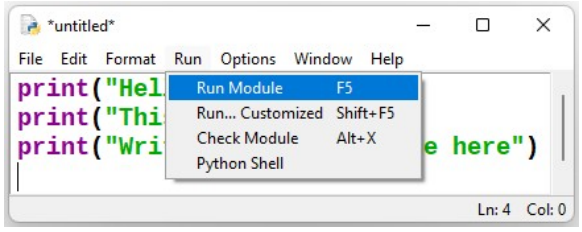


Fig. 1.8: Running a Python program

Step 5. The output appears on shell as shown in Figure 1.9.

Fig. 1.9 : Showing output of Python program

Now you are familiar with both mode of Python Interpreter, let us see the general structure of Python Program.

1.4 Structure of a Python program

In general, the interpreter reads and executes the Python statements line by line i.e. sequentially, however there are some statements that can alter this behavior like conditional statements.

Python Statement and Comments

Python program consists of statements and comments. The indentation is important in statements. The comments are used to explain the specific code.

Python Statement

Instructions written in the source code for execution are called statements. Python statements are written line by line such that one statement end up in that line only. The interpreter considers the *'new line character'* as the terminator of one instruction. However, writing multiple statements per line is also possible in script mode of Python IDLE.

There are different types of statements in the Python programming language like Assignment statement, Conditional statement, Looping statements etc. These help the user to get the required output. For example, n = 50 is an assignment statement.

Multi-line statement

In Python, end of a statement is marked by a newline character.

However, Statements in Python can be extended to one or more lines using parentheses (), braces {}, square brackets [], semi-colon (;), continuation character slash (\). When we need to do long calculations and cannot fit these statements into one line, we can make use of these characters.

Type of Multi-line Statement	**Usage**
Using Continuation Character (/)	`S = 1 + 2 + 3 + \  4 + 5 + 6 + \ 7 + 8 + 9`
Using Parentheses ()	`n = (1 * 2 * 3 + 4 – 5)`
Using Square Brackets []	`footballer = ['MESSI', 'NEYMAR', 'SUAREZ']`
Using braces {}	`X = {1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9}`
Using Semicolons (;)	`flag = 2; ropes = 3; pole = 4`

Python Comments

Comments are used to add remarks or note in the source code. Comments are not executed by interpreter. They are added to understand the source code for others. They are used primarily to document the meaning and purpose of source code and its input and output requirements, so as to remember later how it functions and how to use it.

For large and complex software, several programmers are working in teams and sometimes, a programmer has to work on the program written by other programmer. In such situations, documentations in the form of comments is useful to understand the logic of program.

In Python, we use the hash (#) symbol to start writing a comment. Everything following the # till the end of that line is treated as a comment and the interpreter simply ignores it while executing the statement.

Single line comments

```
# This is single line comment
```

Multi line comments

```
# This is multi line comment
# This is multi line comment
```

1.5 Python Keywords and Identifiers

In Python, keywords are reserved words and identifiers are the names given to variables or functions.

Keywords

Keywords are the reserved words in Python used by Python interpreter to recognize the structure of the program. The keyword can be used in program for the purpose it has been defined. As Python is case sensitive, keywords must be written exactly as given in Table 1.1.

Table 1.1 Python keywords

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	
break	except	in	raise	

Identifiers

In programming languages, identifiers are names used to identify a variable, function, or other entities in a program. The rules for naming an identifier in Python are as follows:

1. Identifiers can be combination of letters in lowercase (a to z) or uppercase (A to Z) or digits (0 to 9) or an uderscore (_). We cannot use special

symbols like !, @, #, $, %, in name of an identifier. Identifier cannot start with digit. For example 1count is invalid identifier, count1 is a valid identifier.

2. Identifier can be of any length. However, it is preferred to keep it short and meaningful.
3. Keywords cannot be used as identifiers.
4. Python is a case-sensitive language. This means, Variable and variable are not the same.
5. Always name identifiers that make sense.

For example, to find the average of marks obtained by a student in three subjects, choose the identifiers as marks1, marks2, marks3 and avg rather than a, b, c, or A, B, C.

```
avg = (marks1 + marks2 + marks3)/3
```

Similarly, to calculate the area of a rectangle, use identifier names, such as area, length, breadth instead of single alphabets as identifiers for clarity and more readability.

area = length * breadth

1.6. Variables and Constants

A variable is a named location used to store data in the memory. You can think of variables as a container that holds data which can be changed later throughout programming. In Python an assignment statement is used to create new variables and assign specific values to them. For example, a = 5 is a simple assignment operator that assigns the value 5 on the right to the variable a on the left.

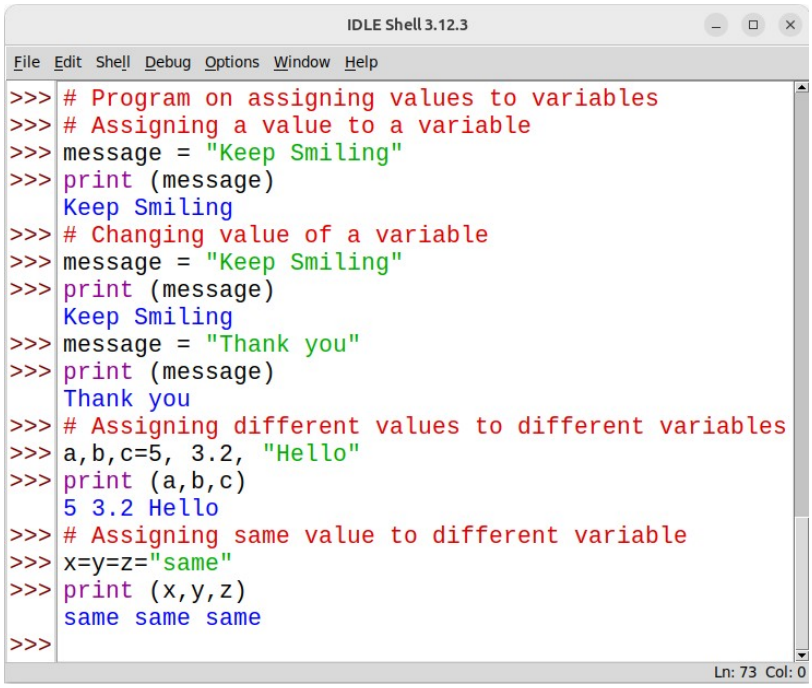
A variable in a program is uniquely identified by a name (identifier). Variable in Python refers to an object — an item or element that is stored in the memory. Value of a variable can be a string (e.g., 'b', 'Global Citizen'), numeric (e.g., 345) or any combination of alphanumeric characters (CD67).

Task	Python Code	Result
Assigning a value to a variable	`message = "Keep Smiling"` `print (message)`	`Keep Smiling`

Changing value of a variable	`message = "Keep Smiling"`
	`print (message)`
	`message = "Thank you"`
	`print (message)`
		`Keep Smiling`
		`Thank you`

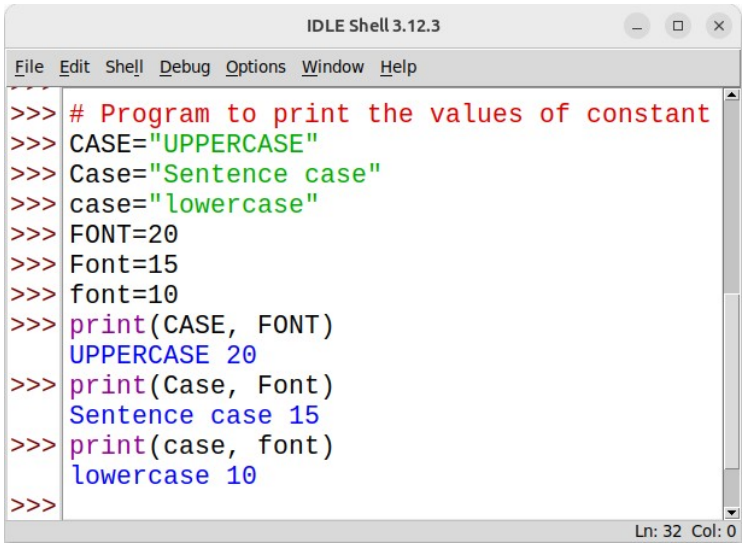
Assigning different	`a,b,c=5, 3.2, "Hello"`	`5`

values to different variables	print(a) print(b) print(c)	 3.2 “Hello”
Assigning same value to different variable	x=y=z= "Same" print(x) print(y) print(z)	Same Same Same



Constants:

A constant is a type of variable whose value cannot be changed in the program. In Python, constants are usually declared and assigned. Following example illustrate to declare and print the values of constant.

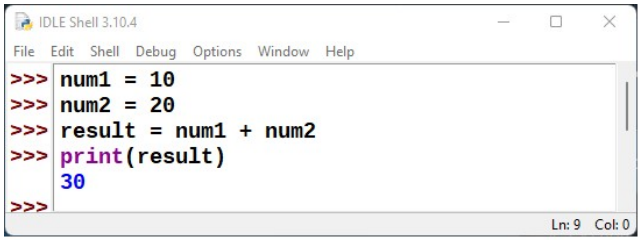


Rules and Naming convention for variables and constants

1. Create a name that makes sense. Suppose, area makes more sense than a.
2. Use camelCase notation to declare a variable. It starts with lowercase letter. *For example:* myName
3. Use capital letters where possible to declare a constant. For example: PI
4. Never use special symbols like !, @, #, $, %, etc.
5. Constant and variable names should have combination of letters in lowercase or uppercase or digits or an underscore (_).

Program. Write a Python code to find the sum of two numbers.

The following program illustrates to find the sum of two numbers.



Check your progress

A. Multiple Choice Questions

1. Which type of Programming does Python support? (a) object-oriented programming (b) structured programming (c) functional programming (d) all of the mentioned
2. Which of the following is the correct extension of the Python file? (a) .python (b) .pl (c) .py (d) .p
3. Is Python code compiled or interpreted? (a) Python code is both compiled and interpreted (b) Python code is neither compiled nor interpreted (c) Python code is only compiled (d) Python code is only interpreted
4. All keywords in Python are in _____ (a) Capitalized (b) lower case (c) UPPER CASE (d) None of the mentioned
5. What will be the value of the following Python expression? 4 + 3 % 5 (a) 7 (b) 2 (c) 4 (d) 1
6. Which of the following is used to define a block of code in Python language? (a) Indentation (b) Key (c) Brackets (d) All of the mentioned
7. Which of the following character is used to give single-line comments in Python? (a) // (b) # (c) ! (d) /*
8. Which of the following is an invalid variable? (a) my_string_1 (b) 1st_string (c) foo (d) _
9. Which of the following is not a keyword? (a) eval (b) assert (c) nonlocal (d) pass

10. Special meaning words of Python, fixed for specific functionality are called (a) identifiers (b) functions (c) keywords (d) literals
11. Names given to different parts of Python program are (a) Identifiers (b) functions (c) Keywords (d) literals
12. Data items having fixed value are called (a) Identifiers (b) functions (c) Keywords (d) literals
13. Which of the following is escape sequence for newline character? (a) \a (b) \t (c) \n (d) \b
14. Which of the following cannot be used for naming an identifier in Python (a) Digits (0-9) (b) Letters (A-Z) (c) Underscore (_) (d) Special symbols(!, @, #, $, %)
15. Variables whose values can be changed after they are created and assigned are called (a) mutable (b) immutable (c) static (d) dynamic

B. Fill in the Blanks

1. Python interpreter can be used in _____ mode and _____ mode.
2. Python statements written in a particular sequence to solve a problem is known as Python _____.
3. Keywords are the _____ words in Python.
4. Python variables support _____ typing.
5. The lines beginning with a certain character, and which are ignored by a compiler and not executed, are called _____.

C. State whether the following statements are True or False

1. Python source code file can be executed in interactive mode.
2. Python program has *.py* extension.
3. Script mode is used to code and modify the code of python programs.
4. In Python, keywords can be used as identifiers.
5. In Python, Identifier can be of any length.
6. Python is a case-sensitive language.
7. The variable names in Python are of fixed length.
8. The identifiers in Python can begin with an underscore.
9. In Python, a keyword cannot be renamed.
10. In Python, a variable can contain values of different types at different times.

Session 2. Data Types and Operators

2.1 Data Types

Every value belongs to a specific data type in Python. Data type identifies the type of data values a variable can hold and the operations that can be performed on that data. Figure 2.1 enlists the data types available in Python.

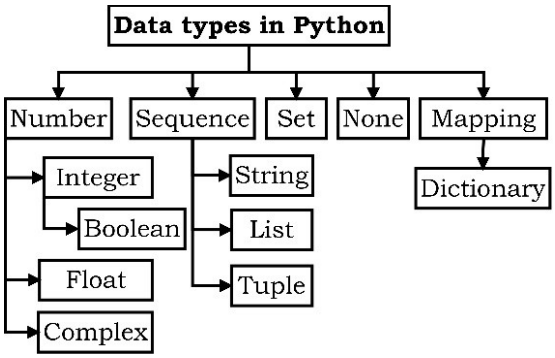


Fig. 2.1: Data types in Python

2.1.1 Number

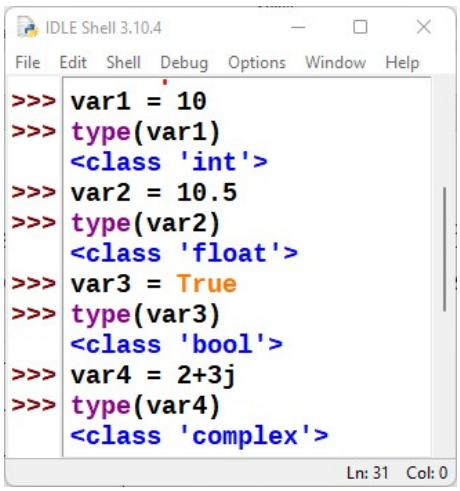
Number data type stores numerical values only. It is further classified into three different types: int, float and complex. Boolean data type (bool) is a subtype of integer. It is a unique data type, consisting of two constants, True and False. Boolean True value is non-zero, non-null and non-empty. Boolean False is the value zero.

Table 2.1 Numeric data types

Type/Class	Description	Examples
int	integer numbers	–12, –3, 0, 125, 2
float	real or floating-point numbers	–2.04, 4.0, 14.23
complex	complex numbers	3 + 4i, 2 – 2i
bool	Boolean values	

Let us execute few statements in interactive mode of Python IDLE to determine the data type of the variable using built-in function type ().

Example 2.1



Variables of simple data types like integer, float, boolean that hold single value. But such variables are not useful to hold a long list of information, such as months in a year, student names in a class, names and numbers in a phone book or the list of artifacts in a museum. For this, Python provides other data types like *tuples, lists, dictionaries* and *sets.*

2.1.2 Sequence

A Python sequence is an ordered collection of items, where each item is indexed by an integer. The three types of sequence data types available in Python are *Strings, Lists* and *Tuples*. We will learn about each of them in detail in later Sessions. A brief introduction to these data types is as follows:

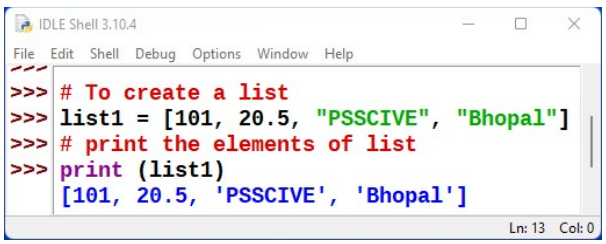
String – String is a group of characters. These characters may be alphabets, digits or special characters including spaces. String values are enclosed either in single quotation marks (e.g., 'Hello') or in double quotation marks (e.g., "Hello"). The quotes are not a part of the string; they are used to mark the beginning and end of the string for the interpreter. For example,

```
>>> str1 = 'Hello Friend'
>>> str2 = "452"
```

It is not possible to perform numerical operations on strings, even when the string contains a numeric value, as in str2.

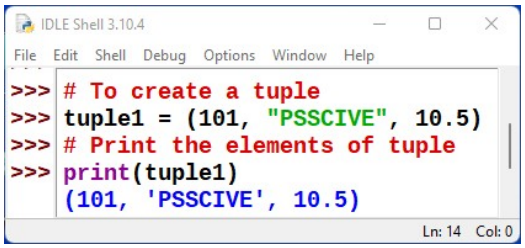
List – List is a sequence of items separated by commas and the items are enclosed in square brackets [ ].

Example 2.2



Tuple – Tuple is a sequence of items separated by commas and items are enclosed in parenthesis (). This is unlike list, where values are enclosed in brackets []. Once created, we cannot change the tuple.

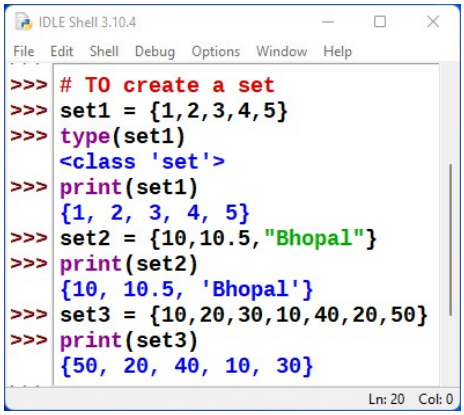
Example 2.3



2.1.3 Set

Set is an unordered collection of items separated by commas and the items are enclosed in curly brackets { }. A set is similar to list, except that it cannot have duplicate entries. Once created, elements of a set cannot be changed.

Example 2.5

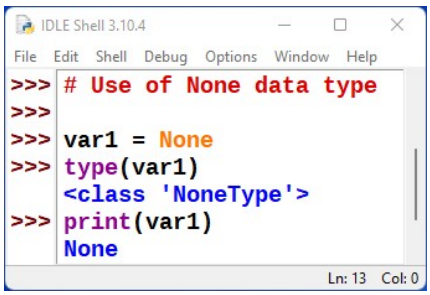


In the above example, set1 is a collection of 5 integers. The set2 is collection of different data types of elements. You must have noticed that elements of set2 have been displayed in an order different from the order in which they have entered. Reason of this is that set is unordered. If you run the same code again, it is possible that you will get an output with the elements arranged in a different order. A set does not allow duplicate values, that you may observe for set3. Here, all the duplicate values have been removed from set3.

2.1.4 None

None is a special data type with a single value. It is used to signify the absence of value in a situation. None supports no special operations, and it is neither False nor 0 (zero).

Example 2.6



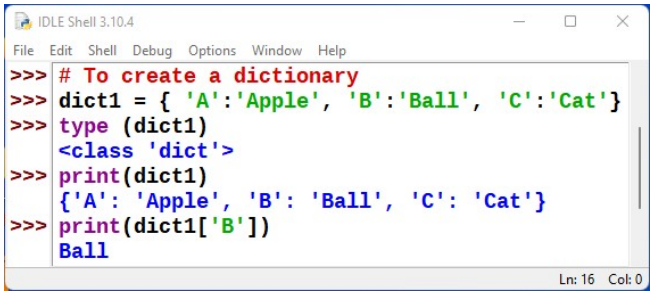
2.1.5 Mapping

Mapping is an unordered data type in Python. Currently, there is only one standard mapping data type in Python called dictionary.

Dictionary

Dictionary in Python holds data items in key-value pairs. Items in a dictionary are enclosed in curly brackets { }. Dictionaries permit faster access to data. Every key is separated from its value using a colon (:) sign. The key: value pairs of a dictionary can be accessed using the key. The keys are usually stringing and their values can be any data type. In order to access any value in the dictionary, we have to specify its key in square brackets [].

Example 2.7



2.2 Mutable and Immutable Data Types

Sometimes we may require to change or update the values of certain variables used in a program. However, for certain data types, Python does not allow to change the values once a variable of that type has been created and values are assigned.

Variables whose values can be changed after they are created and assigned are called mutable. Variables whose values cannot be changed after they are created and assigned are called immutable. When an attempt is made to update the value of an immutable variable, the old variable is destroyed and a new variable is created by the same name in memory. Python data types can be classified into mutable and immutable as shown in Figure 2.1.

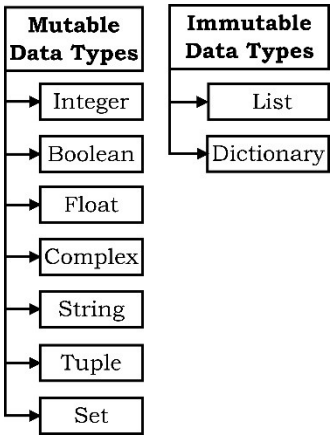


Fig. 2.2 : Classification of data types

Let us now see what happens when an attempt is made to update the value of a variable. `>>> num1 = 300` This statement will create an object with value 300 and the object is referenced by the identifier `num1` as shown in Figure 10.13.

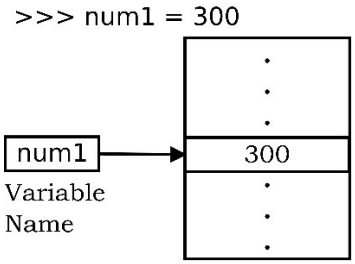


Fig. 2.3 : Object and its identifier

The statement `num2 = num1` will make `num2` refer to the value 300, also being referred by num1, and stored at memory location number, say 1000. So, num1 shares the referenced location with `num2` as shown in Figure 2.4.

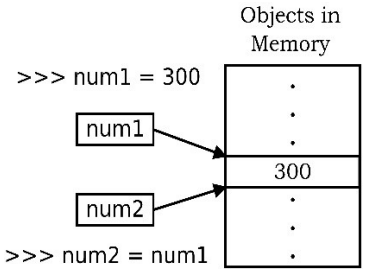


Fig. 2.4 : Variables with same value have same identifier

In this manner Python makes the assignment effective by copying only the reference, and not the data:

```
>>> num1 = num2 + 100
```

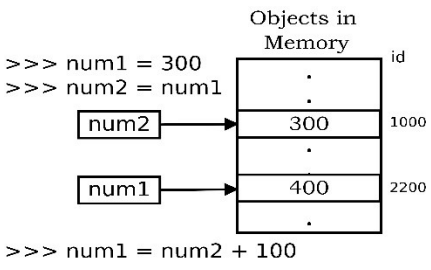


Fig. 2.5 : Variables with different values have different identifiers

This statement *num1 = num2 + 100* links the variable num1 to a new object stored at memory location number say 2200 having a value 400. As num1 is an integer, which is an immutable type, it is rebuilt, as shown in Figure 2.5.

Usage of Python Data Types

Each data type has some specific properties. Appropriate use of data type is depending on the situation. It is preferred to use *lists* for a simple iterative collection of data that may go for frequent modifications. For example, if we store the names of students of a class in a list, then it is easy to update the list when some new students join or some leave the course. *Tuples* are used when we do not need any change in the data. For example, names of months in a year. When we need uniqueness of elements and to avoid duplicity it is preferable to use *sets*, for example, list of artifacts in a museum. If our data is being constantly modified or we need a fast lookup based on a custom key or we need a logical association between the key: value pair, it is advised to use *dictionaries*. A mobile phone book is a good application of dictionary.

2.3 OPERATORS

An operator is used to perform specific mathematical or logical operation on values are called operands. For example, in the expression *"10 + num"*, the value 10, and the variable *num* are operands and the + (plus) sign is an operator. Python supports several types of operators as discussed below.

2.3.1 Arithmetic Operators

Python supports arithmetic operators (+, -, *, /) that are used to perform the four basic arithmetic operations as well as modulus division (%), floor division (//) and exponentiation (**).

Addition operator (+) – Adds the two numeric values on either side of the operator. This operator can also be used to concatenate two strings on either side of the operator.

Subtraction operator (–) - Subtracts the operand on the right from the operand on the left.

Multiplication operator (*) - Multiplies the two values on both side of the operator. Repeats the item on left of the operator if first operand is a string and second operand is an integer value.

Division operator (/) - Divides the operand on the left by the operand on the right and returns the quotient.

Modulus operator (%) - Divides the operand on the left by the operand on the

right and returns the remainder.

Exponent operator (**) - Performs exponential (power) calculation on operands. That is, raise the operand on the left to the power of the operand on the right.

The following code as shown in Figure 2.6 demonstrates the use of *Arithmetic Operators* in Python.

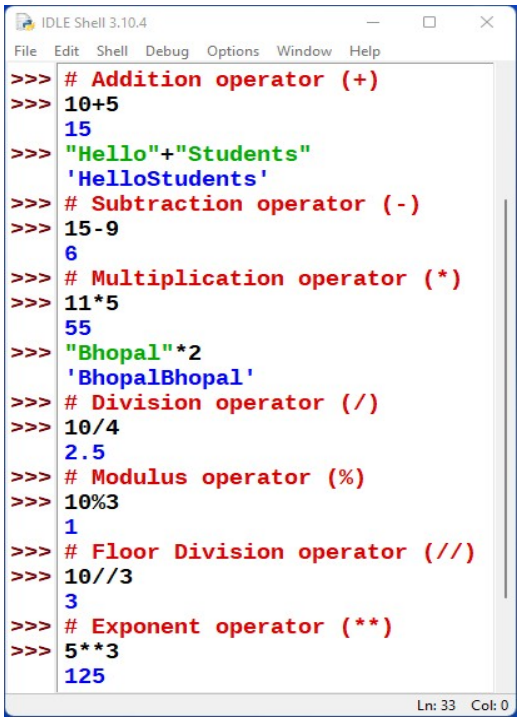


Fig. 2.6 Demonstration of Arithmetic Operators

2.3.2 Relational Operators

Relational operator compares the values of the operands on its either side and determines the relationship among them.

Equals to (==) - If the values of two operands are equal, then the condition is True, otherwise it is False.

Not equal to (!=) - If values of two operands are not equal, then condition is True, otherwise it is False.

Greater than (>) - If the value of the left-side operand is greater than the value of the right- side operand, then condition is True, otherwise it is False.

Less than (<) - If the value of the left-side operand is less than the value of the right- side operand, then condition is True, otherwise it is False operand, then condition is True, otherwise it is False.

Greater than or equal to (>=) - If the value of the left-side operand is greater than or equal to the value of the right-side.

Less than or equal to (<=) - If the value of the left operand is less than or equal to the value of the right operand, then is True otherwise it is False.

Assume the Python variables num1 = 10, num2 = 20, num3 = 10, str1 = "Hello", str2 = "Students" and str3 = “Hello”.

The following code as shown in Figure 2.7 demonstrates the use of ***Relational Operators*** in Python.

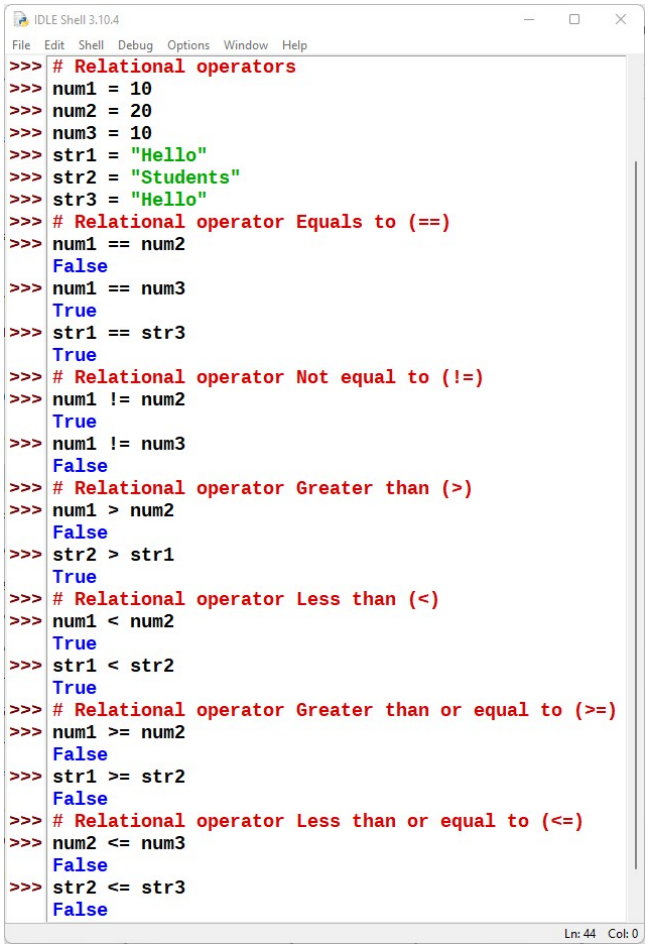


Fig. 2.7 : Demonstration of Relational Operators

2.3.3 Assignment Operators

Assignment operator assigns or changes the value of the variable on its left. Table 2.3 shows various assignment operators available in python

Table 2.3 Assignment operators.

=	Assigns value from right-side operand to left-side operand.
+=	It adds the value of right-side operand to the left-side operand and assigns the result to the left-side operand. In other words, x += y is same as x = x + y.
-=	It subtracts the value of right-side operand from the left-side operand and assigns the result to left-side operand. In other words, x-=y is same as x=x-y.

*=	It subtracts the value of right-side operand from the left-side operand and assigns the result to left-side operand. In other words, x-=y is same as x=x-y.
/=	It multiplies the value of right-side operand with the value of left-side operand and assigns the result to left-side operand. In other words, x*=y is same as x=x*y.
%=	It performs modulus operation using two operands and assigns the results to left-side operand. In other words, x % =y is same as x = x % y.
**=	It performs exponential (power) calculation on operators and assigns value to the left-side operand. In other words, x **= y is same as x = x ** y.

The following code as shown in Figure 2.8 demonstrates the use of ***Assignment Operators*** in Python.

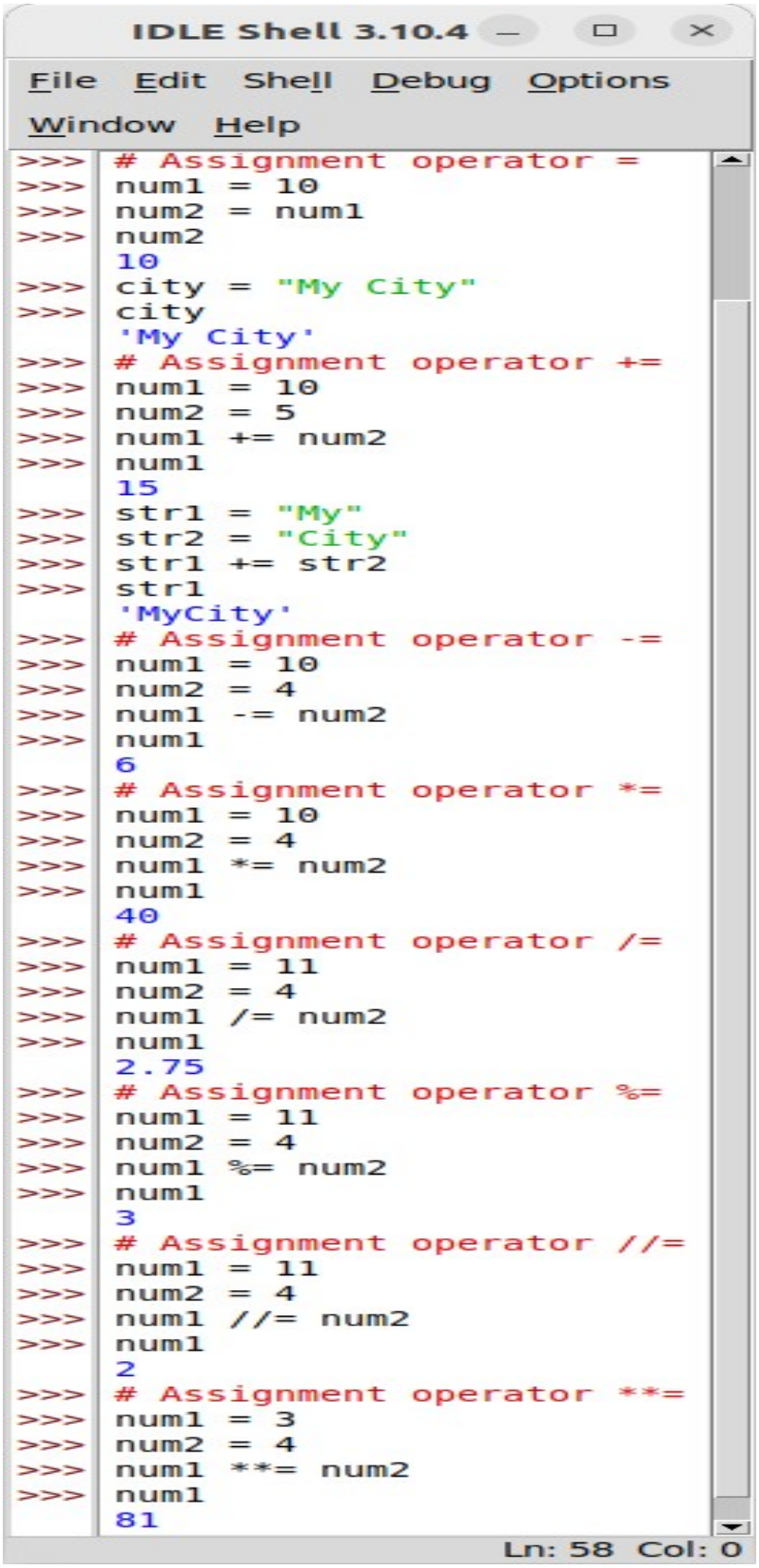


Fig. 2.8 Demonstration of Assignment operators

2.3.4 Logical Operators

There are three logical operators “and, or, not” supported by Python. These are

to be written in lower case only. The logical operator evaluates to either True or False based on the logical operands on either side. Every value is logically either True or False. By default, all values are True except None, False, 0 (zero), empty collections "", (), [], {}, and few other special values. So, if we say num1 = 10, num2 = -5, then both num1 and num2 are logically True. If we have num3 = 0, than num3 is logically False.

Logical AND (and) - If both the conditions are True, then expression becomes True.

Logical OR (or) - If any of the two condition is True, then expression becomes True.

Logical NOT (not) - It is used to reverse the logical state of its operand.

The following code as shown in Figure 2.9 demonstrates the use of ***Logical Operators*** in Python.

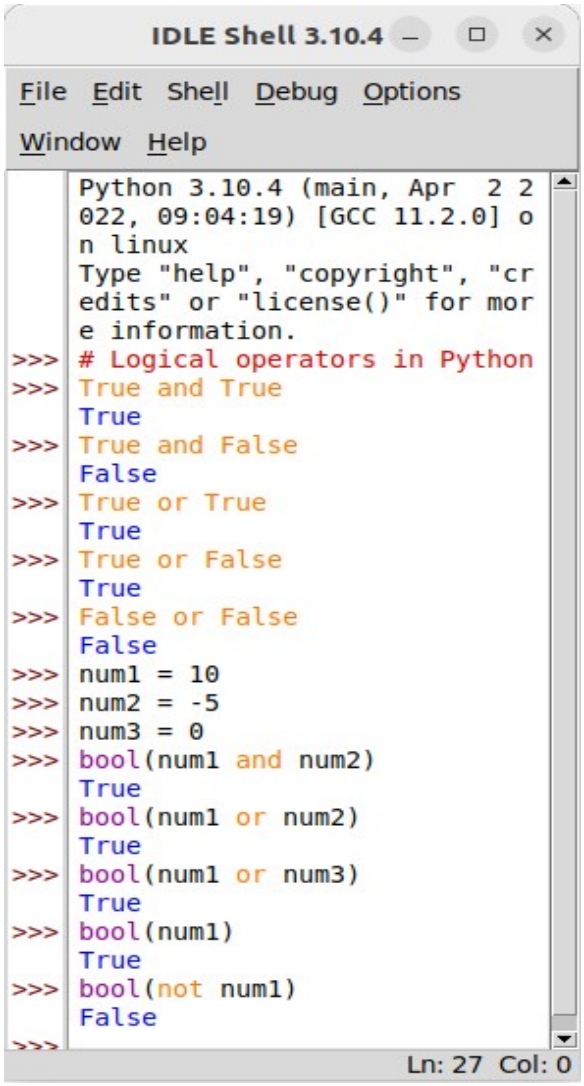


Fig. 2.9 : Demonstration of Logical operators

2.3.5 Identity Operators

Identity operators are used to determine whether the value of a variable is of a certain type or not. Identity operators can also be used to determine whether two variables are referring to the same object or not. There are two identity operators explained in table 10.4.

Table 2.4 Identity operators in Python

Operator	Description	Example

is	Evaluates True if the variables on either side of the operator point towards the same memory location and False otherwise. var1 is var2 results to True if id(var1) is equal to id(var2).	`>>> num1 = 10`
		`>>> num2 = 5`
		`>>> num3 = num1`
		`>>> num2 is num1`
		`False`
		`>>> num3 is num1`
		`True`

is not	Evaluates to False if the variables on either side of the operator point to the same memory location and True otherwise. var1 is not var2 results to True if id(var1) is not equal to id(var2).	`>>> num1 = 10`
		`>>> num2 = 5`
		`>>> num3 = num1`
		`>>> num2 is not num1`
		`True`
		`>>> num3 is not num1`
		`False`

2.3.6 Membership Operators

Membership operators are used to check if a value is a member of the given sequence or not. There are two membership operators explained in Table 2.5.

Table 2.5 Membership operators in Python

Operator	Description	Example (Try in Lab)

in	Returns True if the variable/value is found in the specified sequence and False otherwise.	`>>> a = [10,20,30,40,50]`
		`>>> 10 in a`
		`True`
		`>>> 25 in a`
		`False`

not in	Returns True if the variable/value is not found in the specified sequence and False otherwise.	`>>> a = [10,20,30,40,50]`
		`>>> 10 not in a`
		`False`
		`>>> 25 not in a`
		`True`

2.3.7 Bitwise Operators

Bitwise operators are the only operators which works on equivalent binary value of the integer operands. First of all, integer operands are converted into binary then respected operator works on bit by bit, hence the name is bitwise operators. The result is also converted into decimal format.

The code snippet demonstrates in table 1.6 shows the use of various bitwise operators in Python.

Table 2.6 Bitwise operators in Python

Operator	Description	Example

Bitwise
& (AND)
	Returns 1 if both the bits are 1 else 0.
	Here x=10 and y=12.
	x = 1 0 1 0 (10 in decimal)
	y = 1 1 0 0 (12 in decimal)
	z = 1 0 0 0 (8 in decimal)
		>>> x = 10
		>>> y = 12
		>>> z= x & y
		>>> z
		8

Bitwise
| (OR)
	Returns 1 if either of the bit is 1 else 0.
	x = 1 0 1 0 (10 in decimal)
	y = 1 1 0 0 (12 in decimal)
	z = 1 1 1 0 (14 in decimal)
		>>> x = 10
		>>> y = 12
		>>> z = x | y
		>>> z
		14

Bitwise
~ (NOT)
	Returns one's complement of the number.
	a = 1010 (Binary)
	z = ~a
	= ~ 1010
	= - (1010 + 1)
	= - 11
		>>> x = 10
		>>> z = ~x;
		>>> z
		-11

Bitwise
^ (XOR)
	Returns 1 if one of the bits is 1 and the other is 0 else returns false.
	x = 1 0 1 0 (10 in decimal)
	y = 1 1 0 0 (12 in decimal)
	z = 0 1 1 0 (6 in decimal)
		>>> x = 10
		>>> y = 12
		>>> z = x ^ y
		>>> z
		6

Bitwise
>>(RIGHT)
	Shifts the bits of the number to the right and fills 0 on voids left (fills 1 in the case of a negative number) as a result. Similar effect as of dividing the number with some power of two.	>>> x = 10
		>>> z = x >> 2
		>>> z
		2

Bitwise
<<(LEFT)
	Shifts the bits of the number to the left and fills 0 on voids right as a result. Similar effect as of multiplying the number with some power of two.	>>> x = 10
		>>> z = x << 2
		>>> z
		40

2.4 Expressions

An expression is defined as a combination of constants, variables, and operators. An expression always evaluates to a value. A value or a standalone variable is also considered as an expression but a standalone operator is not an expression. Some examples of valid expressions are given below.

(a) 100

(b) num

(c) num – 20.4

(d) 3.0 + 3.14

(e) 23/3 -5 * 7(14 -2)

(f) "Global" + "Citizen"

4.1 Precedence of Operators

Evaluation of the expression is based on precedence of operators. When an expression contains different types of operators, precedence determines which operator should be applied first. Higher precedence operator is evaluated before the lower precedence operator. Most of the operators studied till now are binary operators with two operands. The unary operators need only one operand, and they have a higher precedence than the binary operators. The minus (-) as well as + (plus) operators can act as both unary and binary operators, but “not” is a unary logical operator.

```
#Depth is using - (minus) as unary operator
Value = -Depth
```

#not is a unary operator, negates True print (not (True))

Table 2.7 lists precedence of all operators from highest to lowest.

Table 2.7 Precedence of all operators in Python

Precedence	Operators	Description
1	**	Exponentiation (raised to the power)
2	~ ,+, -	Complement, unary plus and unary minus
3	* ,/, %, //	Multiply, divide, modulo and floor division
4	+, -	Addition and subtraction
5	<= ,< ,> ,>=	Relational operators
6	== ,!=	Equality operators

7	=, %=, /=, //=, -=, +=, *=, **=	Assignment operators
8	is, is not	Identity operators
9	in, not in	Membership operators
10	not, or, and	Logical operators

Parenthesis can be used to override the precedence of operators. The expression within () is evaluated first. For operators with equal precedence, the expression is evaluated from left to right.

Example 2.8.

How will Python evaluate the following expression? 20 + 30 * 40

```
= 20 + (30 * 40)     #Step 1
#precedence of * is more than that of +
= 20 + 1200          #Step 2
= 1220               #Step 3
```

Example 2.9.

How will Python evaluate the following expression? 20 – 30 + 40

Solution:

The two operators (–) and (+) have equal precedence. Thus, the first operator, i.e., subtraction is applied before the second operator, i.e., addition (left to right).

```
= (20 – 30) + 40     #Step 1
= - 10 + 40          #Step 2
= 30                 #Step 3
```

Example 2.10.

How will Python evaluate the following expression? (20 + 30) * 40

Solution:

```
= (20 + 30) * 40     # Step 1
#using parenthesis (), we have forced precedence of + to be more
than that of *
= 50 * 40            # Step 2
= 2000               # Step 3
```

Example 2.11.

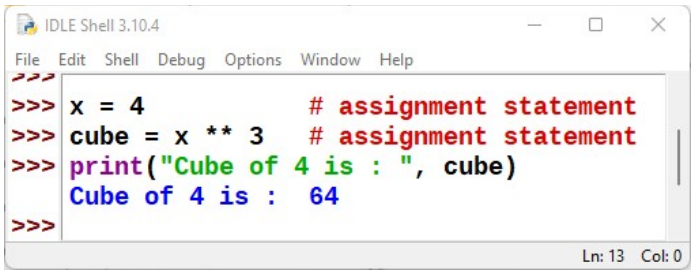
How will the following expression be evaluated in Python? 15.0 / 4 + (8 + 3.0)

Solution:

```
15.0 / 4 + (8.0 + 3.0    #Step 1
15.0 / 4.0 + 11.0        #Step 2
3.75 + 11.0              #Step 3
14.75                    #Step 4
```

2.5 Input Output Statements

In Python, a statement is a unit of code that the Python interpreter can execute. Following example illustrate the statements in Python.

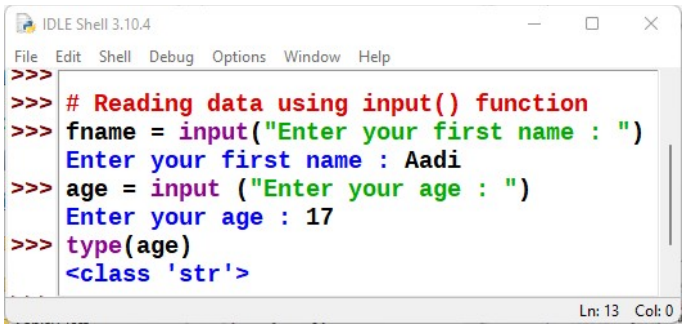


Sometimes, a program needs to interact with the user to receive data which is processed to give the desired output. In Python, the input () function prompts the user to enter data. It accepts all user input as string. The user may enter a number or a string but the input () function treats them as strings only. The syntax for input statement is:

```
input ([Prompt])
```

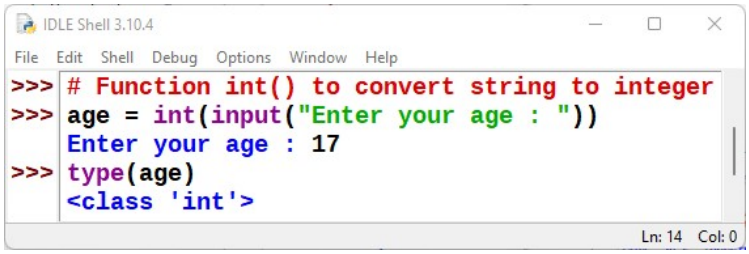
Prompt is the string we may like to display on the screen prior to taking input, and it is optional. When a prompt is specified, first it is displayed on the screen after which the user can enter data. The input() takes exactly what is typed from the keyboard, converts it into a string and assigns it to the variable on left-hand side of the assignment operator (=). Entering data for the input function is terminated by pressing the enter key.

Example 1.2



In Example 1.2, the variable fname will get the string 'Aadi', entered by the user. Similarly, the variable age will get the string '17'. We can typecast or change the datatype of the string data accepted from user to an appropriate numeric value. The code in Example 1.3 convert the accepted string to an integer. If the user enters any non-numeric value, an error will be generated.

Example 1.3



Python uses the print () function to output data to standard output device — the screen. More about function will be covered in Chapter 12. The function print () evaluates the expression before displaying it on the screen. The print () outputs a complete line and then moves to the next line for subsequent output. The syntax for print () is:

```
print (value [, ..., sep = ' ', end = '\n'])
```

`sep:` The optional parameter sep is a separator between the output values. We can use a character, integer or a string as a separator. The default separator is space.

`end:` This is also optional and it allows us to specify any string to be appended after the last value. The default is a new line.

Example 1.4

In *Example 1.4*, the third print function is concatenating strings, and we use + (plus) between two strings to concatenate them. The fourth print function also appears to be concatenating strings but uses commas (,) between strings. Actually, here we are passing multiple arguments, separated by commas to the print function. As arguments can be of different types, hence the print function accepts integer (16) along with strings here. But in case the print statement has values of different types and '+' is used instead of comma, it will generate an error as discussed in the next section under explicit conversion.

CHECK YOUR PROGRESS

A. Multiple Choice Questions

1. Which of these in not a core data type? (a) Lists (b) Dictionary (c) Tuples (d) Class
2. Which is the correct operator for power(x^y)? (a) x^y (b) x**y (c) x^^y (d) None of the mentioned
3. What is the answer to this expression, 22 % 3 is? (a) 7 (b) 1 (c) 0 (d) 5
4. Operators with the same precedence are evaluated in which manner? Left to Right (b) Right to Left (c) Can't say (d) None of the mentioned
5. What is the output of this expression, 3*1**3? (a) 27 (b) 9 (c) 3 (d) 1
6. Which one of the following has the same precedence level? (a) Addition and Subtraction (b) Multiplication, Division and Addition (c) Multiplication, Division, Addition and Subtraction (d) Addition and Multiplication
7. Which one of the following has the highest precedence in the expression? (a) Exponential (b) Addition (c) Multiplication (d) Parentheses
8. Which data type in Python is used to store a sequence of characters? (a) Integer (b) Float (c) String (d) Boolean
9. What will be the output of the following code is (a) 2 (b) 2.5 (c) 3 (d) 3.5

```
>> a=10
>> b=4
>> print (a/b)
```

10. What will be the output of the following code is (a) 4 (b) 6 (c) 10 (d) 14

```
>> a=10
>> b=4
>> a+=b
>> print(a)
```

11. What will be the output of the following code is (a) 0 (b) 1 (c) 01 (d) 10

```
>> a=0
>> b=1
>> print (a and b)
```

12. Which of the following Python statement has syntax error (a) 10+20 (b) (10+20) (c) (10+20 (d) ((10+20)) (c)
13. Wrong output are indicators of (a) Syntax errors (b) Logical errors (c) Runtime error (d) Compile time error
14. Division by zero is a (a) Syntax error (b) Logical error (c) Runtime error (d)

Compile error

15. What is the output of the following code snippet? (a) int (b) Integer (c) float (d) Float
16. What is What will be the output of the following code snippet? (a) set (b) tuple (c) list (d) dictionary

```
>> x = {"apple", "banana", "cherry"}
>> print(type(x))
```

17. What is the output of the following code snippet? (a) dictionary (b) Dictionary (c) dict (d) Dict

```
>> x = {"name": "John", "age": 30}
>> print (type(x))
```

18. What is the data type of the variable "x" in the following code? (a) Integer (b) String (c) Float (d) Boolean
19. What will be the output of the following code? (a) 30 (b) "1020" (c) Error (d) "10200"

```
>> x = 10
>> y = "20"
>> print(str(x) + y)
```

20. What will be the output of the following code? (a) [1, 2, 3] (b) [1, 2, 3, 4] (c) [1, 2, 3, 4, 4] (d) [1, 2, 3, 3]

```
>> x = [1, 2, 3]
>> y = x.copy()
>> x.append(4)
>> print(y)
```

B. Fill in the Blanks

1. A string is a _____ data type in Python.
2. The data type _____ is used to store a collection of items, where each item is indexed by a key?
3. In Python the data type _____ in is mutable?
4. In Python the data type _____ represents a sequence of characters
5. In Python, the _____ function prompts the user to enter data.

C. State whether the following statements are True or False

1. Mathematical operations can be directly performed on a string in Python without conversion?
2. In Python the data type "set" is used to store a collection of items.
3. In Python, strings can be concatenated using the "+" operator.

4. In Python, the data type tuple is mutable?
5. In Python, input () function accepts all user input as string.

D. Programming Questions

1. Write the Python assignment statements to:
   a) Assign 10 to variable length and 20 to variable breadth.
   b) Assign the average of values of variables length and breadth to a variable sum.
   c) Assign a list containing strings 'Paper', 'Gel Pen', and 'Eraser' to a variable stationery.
   d) Assign strings 'Mohandas', 'Karamchand', and 'Gandhi' to variables first, middle and last
   e) Assign the concatenated value of string variables first, middle and last to variable full name. Make sure to incorporate blank spaces appropriately between different parts of the names.

2. Write the output of the following:

```
a) >> num1 = 4
   >> num2 = num1 + 1
   >> num1 = 2 print (num1, num2)
b) >> num1, num2 = 2, 6
   >> num1, num2 = num2, num1 + 2
   >> print (num1, num2)
c) >> num1, num2 = 2, 3
   >> num3, num2 = num1, num3 + 1
   >> print (num1, num2, num3)
```

3. Which data type will be used to represent the following data values and why?

Data values	**Data type**	**Reason**
Number of months in a year		
Resident of Delhi or not		
Mobile number		
Pocket money		
Volume of a sphere		
Perimeter of a square		
Name of the student		
Address of the student		

4. What will be the output of statement print(num1) in the following example when num1 = 4, num2 = 3, num3 = 2

```
>> num1 += num2 + num
>> num1 = num1 ** (num2 + num3)
```

```
>> num1 **= num2 + num3
>> num1 = '5' + '5'
>> num1 = 2+9*((3*12)-8)/10
>> num1 = 24 // 4 // 2
>> num1 = float(10)
>> num1 = int('3.14')
```

5. What will be the output of following statements

```
>> print('Bye'  == 'BYE')
>> print(10 != 9 and 20 >= 20)
>> print(10 + 6 * 2 ** 2 != 9//4 -3 and 29 >= 29/9)
>> print(5 % 10 + 10 < 50 and 29 <= 29)
>> print((0 < 6) or (not(10 == 6) and 10<0)))
```

Module 3. Data Literacy

Rakesh was accompanying his father for opening a bank account. The Bank manager ask for Aadhar card of his father. He further captures face image and Aadhar card for authentication of his father. Automatically all the information of his father appears on screen. Rakesh was surprised to see how fast Aadhar authenticates the person.

Fig: Aadhar Authentication

In this unit we are going to discuss the concept of data and data science. We will see how data science is going to change of our world.

Session 1. Basics of Data Literacy

1.1 Introduction to Data Literacy

Data

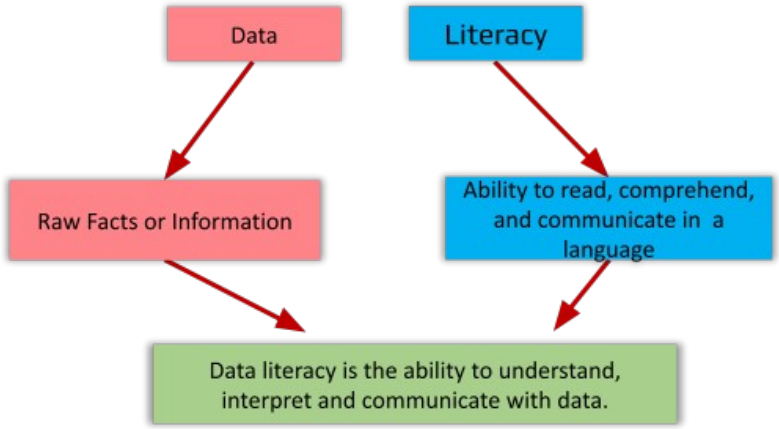
The term data is not a new term. It is actually used since 1640 in literature, but in 1946 it was first used in relation with “transmittable and storable computer information”.

The word data is taken from Latin word “datum”. Data means facts or set of values that are collected for processing. Data can be measured, collected, reported analysed and visualized. Data is available in a raw form. Data in this form is not very useful.

When the data is processed, we get the information. This information leads to knowledge to understand.

Data Literacy

Data literacy means to know all about what we can do with data. We can able to collect, analyze, and show data in ways that make sense.



In real life, data can be in different formats such as number, text, image, audio, video as shown in the Figure 1.1. Observe that these different types are used to describe things.

Data type	**Example**
Number	52354872
Text	Hello world

Image	
Audio	
Video	

Fig. 1.1 : Various types of data

If data is well organized and processed then it becomes useful. Several times you need to write your personal information. This personal information is particularly required when you apply for a job in an organisation. This personal information is also called bio-data or resume. Following illustration shows some basic entities to be written in bio-data. Observe that this bio data contains the data elements of different types such as characters (., -, ‘, ‘ ’), capital and small letters (D, e, G, o….), numbers (11, 2, 9, 1990….), date and image.

PRACTICAL ACTIVITY 1.1. Identify data types for the given data values

Materials Required

Pen, Paper, Photo, Personal details

Procedure

1. Prepare your personal details consisting of Name, Father’s Name, Photo and other details as shown in the following table.
2. Identify the data type for each of the data values provided by you for your personal details.
3. Record the details as shown in the following table.

Personal Details		
Data Item	**Data Value**	**Data type**
Name	Ms. Devanshi Ghosh	Text
Photo		Image
Father's Name	Mr. Aryan Ghosh	Character with text

Mother's Name	Mrs. Poorva Ghosh	Character with text
Permanent Address	House Number 215, APJ Kalam Marg, Delhi	Character with text
Present Address	House Number 115, Kasturba Nagar, Bhopal	Character with text
Date Of Birth	9 December, 1990	Date with number and text
Qualification	B. Sc.	Character with text
Height	158 CM	Number with text
Weight	60 kg	Number with text
Contact Number	987654321	Number

Examples of Data

In the earlier illustration we have seen that your personal information contains different types of data. In real life we came across many examples where data is required. For example, you require your personal data in preparation of Aadhaar card, passport and school ID.

There are two types of data. Some data elements are called individual data. For example, your personal data is individual data. The examples of personal data is shown in Figure 1.1.

Fig. 1.1 (a) : Aadhaar card

Fig. 1.1 (b) : Passport ID

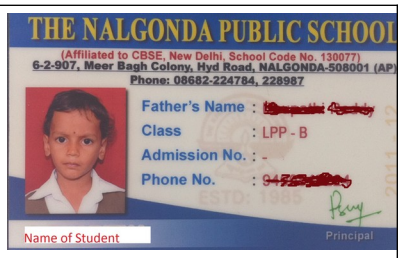


Fig. 1.1 (c) : School ID

Demographic data

The other type of the data is demographic data. The demographic data represents a large population. There are many examples of demographic data such as “Energy survey data”, “Gross domestic product and stock market data”. The examples of various demographic data are shown in Figure 1.2

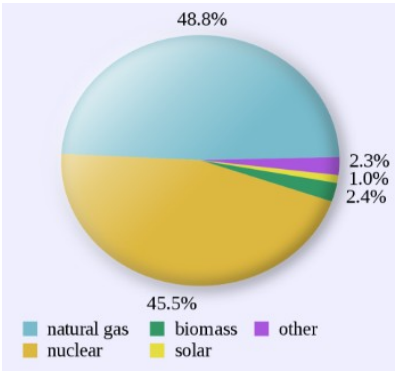


Fig. 1.2 (a) : Pie chart for Energy survey Data

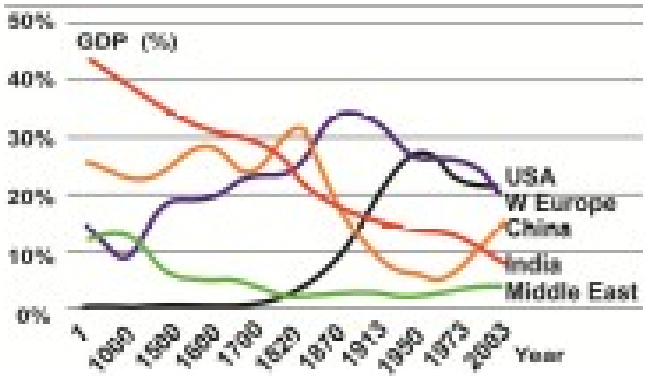


Fig. 1.2 (b) : Line graph of Gross Domestic Product

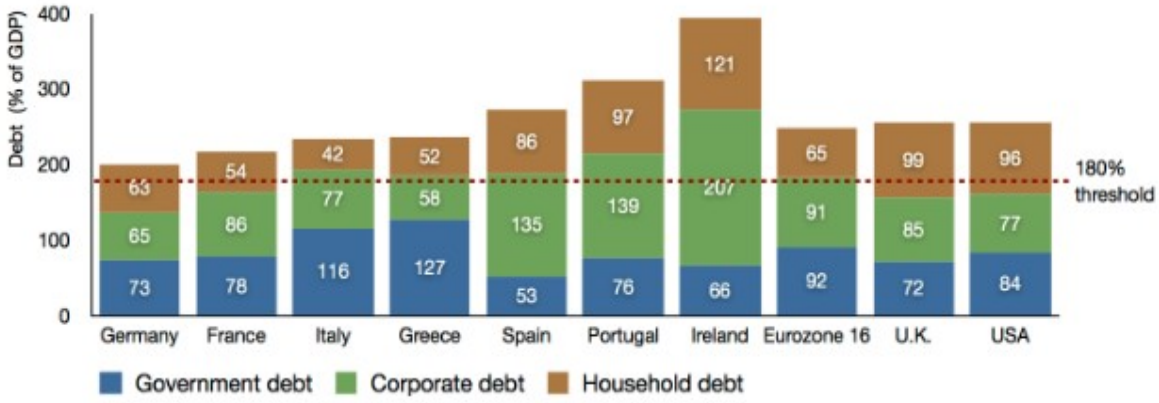


Fig. 1.2 (c) : Bar graph of Gross domestic product

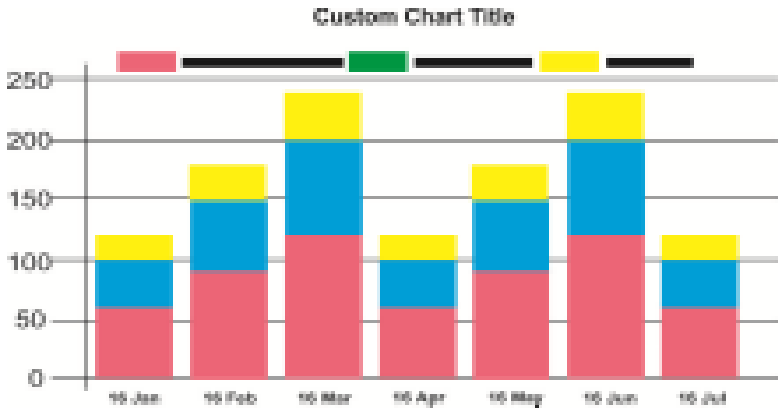


Fig. 1.2 (d) : Line graph for stock market data

Observe that the data can be visualized in various formats such as pie chart, bar graph and line graph.

Age, society, education, gender and marital status can be considered as demographic features. Data associated with these demographic features is called demographic data. Demographic data refers to the public data that normally represents certain demographic features for a large population. By using this data, we can easily understand some common features of a large population.

For example, India is said to be a young country because the average age of the Indian population is around *30-35* years.

By collecting the gender information of the population of a country then one can easily infer the percentages of males and females in that country. This information is useful for framing the common policies and taking decisions for an organisation. Figure 1.49 (a) and (b) gives the illustration of male and female workforce demographic data in rural and urban India.

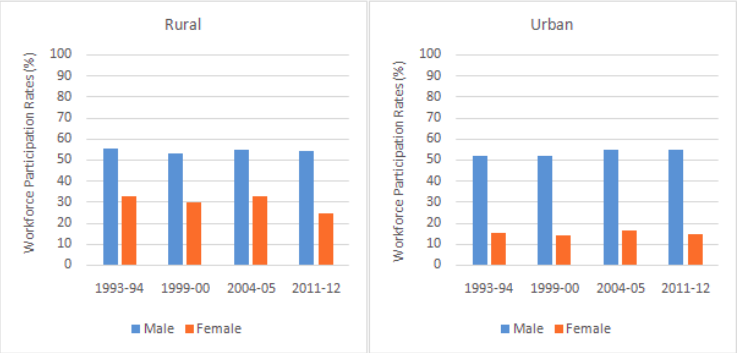


Fig. 1.3 (a): Male and female workforce demographic data in rural and (b) urban India

From this data one can understand that the number of working females is less in urban areas and more in rural areas. Also, we can infer that the number of working males is more than females both in rural and urban India. Such types of useful conclusions can be easily drawn from the demographic data. These conclusions are useful for the government or any organisations to make their decisions or to frame their policies.

PRACTICAL ACTIVITY 1.2

From the given data identify the personal data and demographic data

Materials Required

Pen, Paper, Photo, Given data

Procedure

1. Observe the given data carefully and then decide whether it is personal data or demographic data.
2. Write the appropriate name of the data in front of the data item.

Data item/ picture	**Data type (Personal/Demographic)**
Name of a person	Personal data
Photo of a person	Personal data
Population of India	Demographic data

Average age of Indians	Demographic data
Literacy rate of your state	Demographic data
Educational qualification	Personal data

PRACTICAL ACTIVITY 1.3

Visualise the data using bar graph, pie chart and line graph

Materials Required

Computer system with spreadsheet software, Paper, Given data

Procedure

Step 1. Record the maximum and minimum temperature in your town on all weekdays as shown in the table below.

Day	**Max.Temperature in degree Celsius**	**Min.Temperature in degree Celsius**
Sunday	41.2	29.1
Monday	40.6	29.6
Tuesday	40.8	29.5
Wednesday	41.1	30
Thursday	41.2	30.1
Friday	41	29.9
Saturday	41.3	30.3

Step 2. Enter the data in the spreadsheet and draw a line graph to visualise the data.

Step 3. Enter the data in the spreadsheet and draw a pie chart to visualise the data.

Step 4. Enter the data in the spreadsheet and draw a bar graph to visualise the data.

Step 5. Record your observations in visualising the data.

Types of Data

Data can be of variety of types such as – *(1) Digital or Electronic Data, (2) Qualitative or descriptive data (3) Quantitative or measurable data*

1. Digital or Electronic Data

The data that is captured by electronic devices and can be viewed and processed by using computing is called digital or electronic data. Any discrete data can be called digital data. Electronic data is an electronically created digital file. Nowadays many websites use One Time Password (OTP) at the time of login. Basically, OTP is computer generated code which is used for authentication of users. Generally, at the time of money transaction it helps for secure as well as authentication as shown in Figure 1.4.

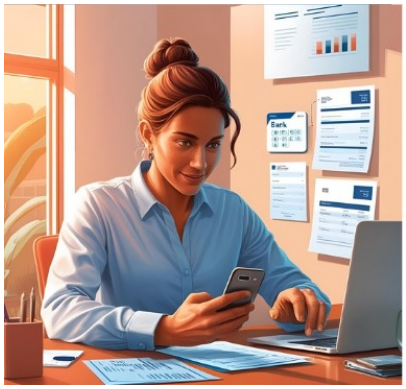


Figure 1.4 : OTP Process Flow

OTP Process Flow

Enter a bank website address such as https: www.....

Step 1. Enter login ID and Password into your bank's website.

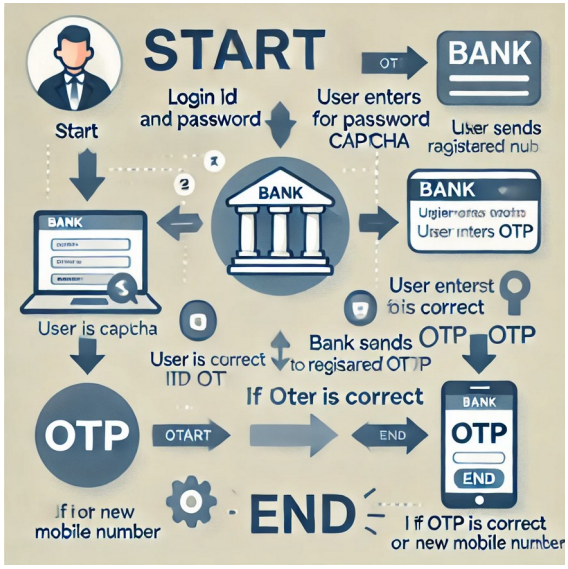
Step 2. Website will ask you to enter captcha.

Step 3. Now the bank will send OTP (One Time Password) to your registered mobile number.

Step 4. Enter this OTP and then only you can access your account.

Step 5. Correct entering of OTP indicates the authentication is completed.

Flow Diagram



OTP is used in various fields for example when use Mobile App like (Bharat Interface for Money (BHIM), Swachh Bharat Abhiyaan, Goods and Services Tax (GST) Rate Finder, Unified Mobile Application for New-age Governance (UMANG), Passport Mobile Application (mPassportSeva), Online (Right to Information) RTI, MyGov, Startup India and use Automated Transaction Machine (ATM) or when user login in website. Some of Indian Govt apps for citizens of India are as shown in Figure 1.5, that makes use of OTP .

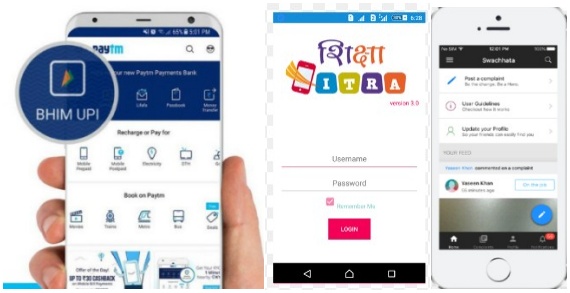


Fig. 1.5 : Apps that make use of OTP

Digital data can be either quantitative data or qualitative data.

Qualitative data

Whenever Words and Descriptions are used for data it is called "Qualitative data". Qualitative data can be observed but is subjective and therefore difficult to use for the purposes of making comparisons.

Qualitative data is data concerned with descriptions, which is difficult to get computed and used for observation.

For example, the softness of your skin, colour of eyes, face image, thumb image, finger print image, iris image, palm image, are all qualitative data. Observe that qualitative data cannot be expressed in the form of numbers but we use a natural language description to express it. The examples of qualitative data is shown in Figure 1.6.

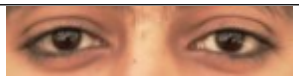


Fig. 1.6 (a) color of eyes

Fig. 1.6 (b) face image

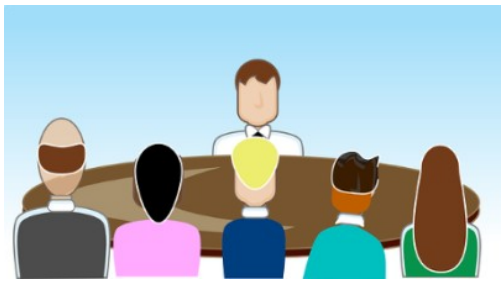
Fig. 1.6 (c) thumb image

Fig. 1.6 (e) : iris image

Fig. 1.6 (d) : Finger print image		Fig. 1.6 (f) : Palm image

Fig. 1.6 : Different types of qualitative data

Qualitative data can be collected by using various methods such as group discussion, interviews and a survey. Computer machines and peripherals can also be used for collection of qualitative data such as face image data, thumb image, finger print image, iris image and palm image as shown in Figure 1.6 (f). Figure 1.7 (a), (b), (c) and (d) shows the various ways to collect the qualitative data.



Applications of Qualitative data

Qualitative data is used in various applications such as scientific research, health and biomedical research, marketing research, education, entertainment, banking, energy and government sectors. For example, the Government of India uses qualitative data for preparation of Aadhaar card, passport and driving license.

PRACTICAL ACTIVITY 1.4

Capturing qualitative data

Materials Required

Digital camera or mobile phone and computer system

Procedure

Step 1. Capture the face image of your classmates through a digital camera or using a mobile phone.

Step 2. Record your voice using your mobile phone.

Step 3. Capture the video of your friend while working.

Step 4. Transfer the picture, audio and video files from the device to the computer.

Step 5. Prepare the list of these data items in tabular form and indicate it as a qualitative data.

Quantitative data

Quantitative data is data that is expressed only with numbers. Quantitative data is information that can be measured and represented numerically.

Numerical data can be easily counted and therefore can be easily analysed by using analytical methods. Quantitative data is also called statistical data. Statistical analysis techniques can be applied to quantitative data. In real life several times we represent the measurement in the form of numbers. For example, height, weight, age and ID number are all examples of quantitative data. Salary, Crop yield, Area, Air pollution index are also examples of quantitative data. Quantitative data can be represented in the form of graphs and tables. Figure 1.8 shows examples of qualitative data.

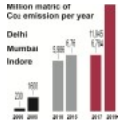
Feature	Height	Weight	Age	ID number	Salary	Crop yield	Area	Air Pollution
Value	5.6 feet	60 kg	27 years	12	Rs. 35,000	1.2 tons	2500 Sq. Feet.	59 micrograms per cubic meter (µg/m3)
Related image								

Fig. 1.8 : Examples of qualitative data

Observe that when quantitative data is expressed in numbers with specific units. Unit depends upon the physical quantity that is measured. For example, weight has a unit of kilogram and area is measured in square feet.

PRACTICAL ACTIVITY 1.5

Prepare a list of quantitative data

Materials Required

Word or Excel Software, Computer, Printer

Procedure

Step 1. Prepare the Employee table with various fields such as Emp id, Employee salary, Pin code, Bank Account Number, and Mobile Number as shown below.

Employee ID	Employee Salary	Pin Code	Bank Account Number	Mobile Number
101	70000	462011	12454784244	1236544654
102	85000	462024	12544557177	NIL
103	85000	462042	12325336868	1232125564
104	80000	465265	12345657887	1235454787
105	80000	454454	12345467878	1234546578
106	40000	454775	12345547878	1234545678

Step 2. Observe that all values in the fields are represented by numbers. Therefore, this data is quantitative data.

Step 3. From such data you can easily infer the following information.

- Highest salary in an organization is Rs. 85000/-
- Lowest salary in an organization is Rs. 40000/-
- Number employees having same salary are 2 and their amount is Rs. 85000/-
- Number of employees having mobile phones is 5.

Data Science

Data Science is a branch of Science that combines various fields. It is a combination of Mathematics, Statistics and Computer Science. It is used to extract meaningful insights and knowledge from the data.

Data science is widely used in AI for the purpose of prediction. For example, disease prediction can be done by using patient data. We can do the risk assessment by using the stock market analysis. Sentiment analysis, Customer behaviour prediction can also be achieved by using data science.

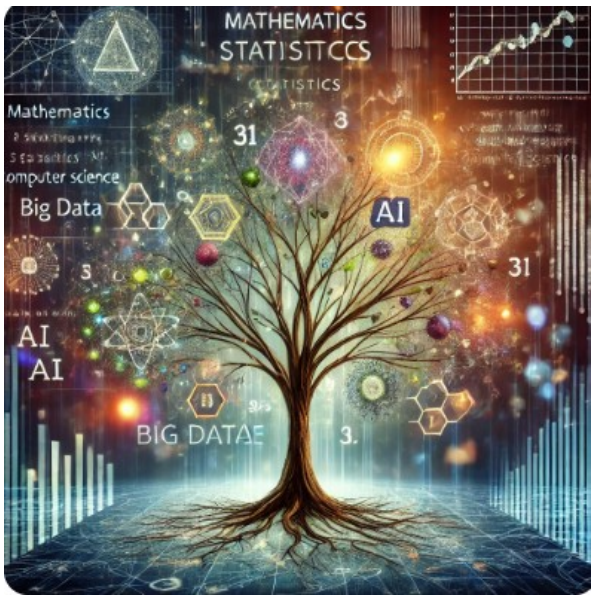


Fig. 1.9: Data Science

There are various components that constitute data science.

1. ***Data collection*** **–** Gathering of raw data from databases, web pages and sensors. There are various tools available for data collection. The DBMS tools such as SQL, Web Scraping can be used to store the data.
2. ***Data pre-processing*** – The collected data is pre-processed to find out the missing or duplicate values. Pandas and NumPy in Python can be used for data pre-processing.
3. ***Data Analysis*** **–** After preprocessing the data can be analysed to obtain patterns or relationships.
4. ***Feature extraction*** **–** Features can be extracted from the data by using tools such as Personal Component Analysis (PCA) and Scikit-Learn.
5. ***Statistical Analysis*** **–** Various statistical methods can be used for analysis of data by using SciPy of Python.
6. ***Machine Learning*** **–** Predictive models using supervised or unsupervised learning by using tools such as PyTorch and Tensorflow.

Fig. Components of Data Science

Practical Activity 1.6

Illustrate the use of Data Science in Health care

Rahul visited the hospital with his friend as his friend was suffering from illness. He found the following steps are carried out by the hospital.

Step 1. Patient data was entered on the computer.

Step 2. Several tests were performed such as blood test, urine test etc.

Step 3. Doctor collected reports online on a computer screen.

Step 4. The data was processed and suggestions were given to the doctor regarding diagnosis of disease.

Step 5. Doctors record it carefully and according to diagnosis of disease, treatment is suggested.

Fig. Data Science in Health care

Practical Assignment

Observe carefully the technologies that you are using in daily lives. Identify where data mining is used.

Data Mining

Data Mining is a process of extraction of meaningful features or patterns from large data sets. It uses statistics and Machine Learning. Data mining is a very common technique that is used in various AI applications such as given below.

1. ***Classification*** – Classifying into categories based on input features.
2. ***Clustering*** – Grouping similar items together.
3. ***Regression*** – Predicting numerical outcome based on input feature.
4. ***Text mining*** – Extracting knowledge from textual data.

Data mining can be widely used in various AI applications such as business and marketing, healthcare, finance, social media, and telecommunication.

Fig. 1.10 : Data mining

Check your progress

A. Multiple choice questions

1. In real life in which of the format data cannot be represented (a) binary code (b) text, (c) image, (d) audio, (e) video
2. Your biodata written on paper does not have the data of type (a) numberic (b) text (c) image (d) video
3. Data can be visualized in the form of (a) pie chart, (b) bar graph (c) line graph (d) All of the above
4. Which of the following is not the example of qualitative data (a) Age (b) softness of skin, (c) colour of eyes, (d) face image, (e) thumb image

5. Qualitative data can be used for preparation of (a) Aadhaar card, (b) Passport (c) Driving license (d) All of the above

B. Fill in the blanks

1. Processing of data produce _____ .
2. The demographic data represents a _____.
3. Digital data is also called as _____
4. Qualitative data is also called as _____
5. Quantitative data is also called as _____
6. OTP stands for _____
7. Any discrete data can be called _____
8. Digital data can be either _____ or _____ data.
9. Data science is widely used in AI for the purpose of _____.
10. Pandas and NumPy in Python can be used for _____

C. State whether true or false

1. Demographic data refers to the private data of a person.
2. Demographic data can be used to know the common features of a large population.
3. The useful conclusions can be drawn from demographic data.
4. Electronic data is an electronically created digital file.
5. Any discrete data can be called digital data.
6. Quantitative data is difficult to for the purposes of making comparisons.
7. Qualitative data can be easily analysed by using analytical methods.
8. Statistical analysis techniques cannot be applied to quantitative data.
9. Data Mining is a process of extraction of meaningful features or patterns from large data sets.
10. Age, society, education, gender and marital status are the examples of demographic data.

D. Answer the following questions in short

1. What is Data?
2. List the different types of data?
3. Give the exmaple of qualitative and quntitative data.
4. Give the few exmaple of Data science in real life.
5. List the various components of data science.
6. List the few applications of data mining.
7. What is data pre-processing?
8. What is data analysis?
9. What is feature extration?
10. List the tools of Machine Learning?

Session 2. Data Pre-processing

Introduction

The raw data that is collected from the various sources is noisy and not suitable for analysis. Hence such data needs to be pre-processed. Data preprocessing is a step of transforming the raw data into a clean and usable format. Such data can be used for the purpose of analysis and it is also used for machine learning models.

The steps that are involved in the data pre-processing are as given below.

1. ***Data collection*** – collect the data from various sources such as databases, files or websites.
2. ***Data cleaning*** – handle missing values, remove duplicates and correct the data errors.
3. ***Data transformation*** – normalised the data or use scale features for uniform distribution.
4. ***Feature extraction*** – identify and extract features among the data values.
5. ***Data reduction*** – reduce the dimensionality of the data.
6. ***Data splitting*** – split the data into training, validation and testing.

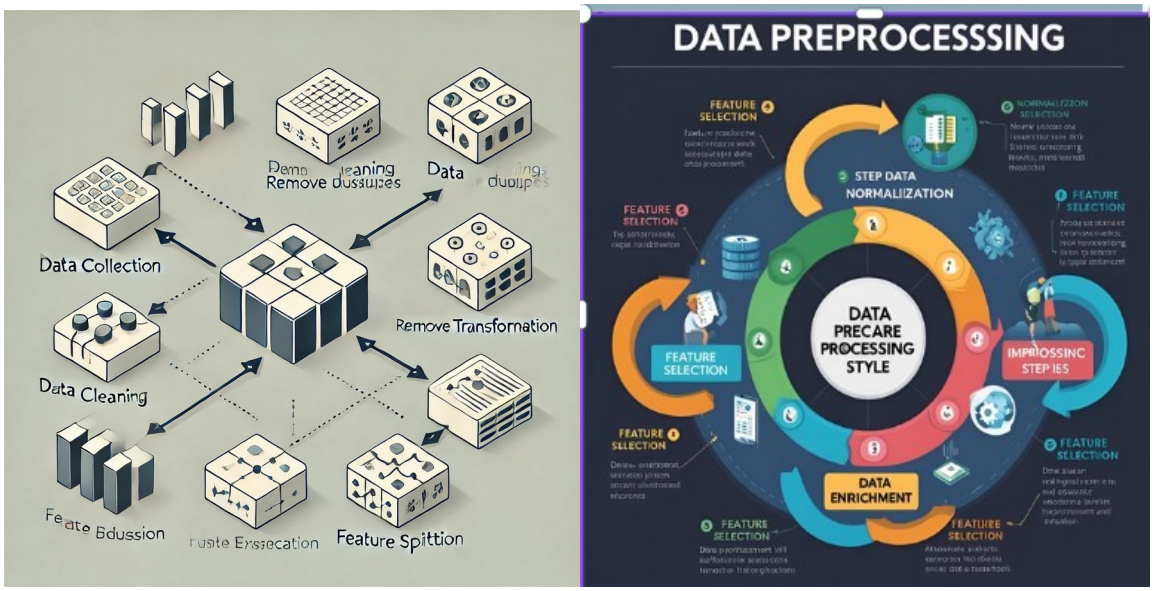


Fig. 2.1 : Data Pre-processing

Example of Data Pre-processing:

Example: Customer Data Preprocessing

Scenario:

A company collects customer information from various sources (databases, surveys, websites). The dataset contains missing values, duplicate records, and

inconsistent formats. The goal is to prepare the data for a machine learning model.

1. Data Collection

Sources:

Databases (Customer transactions)

Files (CSV, Excel reports)

Websites (User behavior tracking)

Example Dataset (Raw Data):

Cust. ID	**Name**	**Age**	**Income**	**Purchase History**	**City**	**Gender**
101	Meena	25	50000	Yes	Mumbai	Female
102	Rahul		60000	No	Pune	Male
103	Dipak	30	70000	Yes	Mumbai	Male
104	Meena	25	50000	Yes	Mumbai	Female
105	Deepali	28		Yes	Delhi	Female

2. Data Cleaning

Handle missing values: Fill missing age/income with median or mean values.

Remove duplicates: The duplicate row for Meena is removed.

Correct errors: Standardize city names, fix incorrect data formats.

Cleaned Dataset:

Cust. ID	**Name**	**Age**	**Income**	**Purchase History**	**City**	**Gender**
101	Meena	25	50000	Yes	Mumbai	Female
102	Rahul	27	60000	No	Pune	Male
103	Dipak	30	70000	Yes	Mumbai	Male
105	Deepali	28	65000	Yes	Delhi	Female

3. Data Transformation

Normalize income to a scale of 0 to 1.

Convert categorical variables (Gender, Purchase_History) to numerical values (One-Hot Encoding).

Transformed Dataset:

Cust. ID	Age	Income (Scaled)	Purchase History (1=Yes, 0=No)	City Mumbai	City Pune	City Delhi	Gender Female	Gender Male
101	25	0.25	1	1	0	0	1	0
102	27	0.50	0	0	1	0	0	1
103	30	0.75	1	1	0	0	0	1
105	28	0.65	1	0	0	1	1	0

4. Feature Extraction

Selecting important features for training, such as Age, Income, and Purchase_History.

Removing less relevant columns like Customer_ID.

Final Feature Set:

Age	Income (Scaled)	Purchase History (1=Yes, 0=No)
25	0.25	1
27	0.50	0
30	0.75	1
28	0.65	1

5. Data Reduction

Reduce dimensionality using PCA (Principal Component Analysis) if needed.

6. Data Splitting

Split the dataset into Training (70%), Validation (15%), and Testing (15%) sets.

Set	Rows	Used for
Training	3	Model Training
Testing	1	Model evaluation

Final Processed Data is Ready for Machine Learning. This dataset is now clean, standardized, and split for model training.

Data Understanding

Data Understanding is an important phase in data science. In this phase the data is explored to get the insights into the data sets. By using data understanding we can determine quality, relevance and potential of the data for solving the problem.

Data Understanding involves classification of data into different categories – numerical, categorical, ordinal, and textual data.

If the data contains only numbers then we can easily determine certain statistical features such as mean, mode, standard deviation, Min and Max.

Categorical data can be divided into different groups or subgroups. For example, we can have nonnumerical categories of data which are often recognised by labels or names. For example, Red, Blue and Green. Sometimes we use numbers for representation of categories, such as 1 for Male 2 for Female. Other examples of categorical data are Marital status, Blood type and Geographic region. Following figure shows the different categories of data.

Data Type	**Definition**	**Example**
Numerical (Quantitative) Data	Data that represents numbers and can be measured or counted.	Age (25 years), Salary (50,000), Height (5.7 ft), Temperature (30°C)
Categorical (Nominal) Data	Data that represents categories without a meaningful order.	Gender (Male, Female), City (Mumbai, Pune), Car Brand (Toyota, Maruti)
Ordinal Data	Categorical data with a meaningful order but without equal differences between values.	Education Level (High School < Bachelor's < Master's), Customer Satisfaction (Low<Medium<High)
Textual (Unstructured) Data	Data in text form that requires processing to extract meaning	Customer Reviews ("Great product!"), Tweets ("#TrendingTopic"), Product Descriptions ("Soft cotton t-shirt")

Fig. 2.2 : Data Understanding Examples

Numerical data can be verified and can be categorised into different formats such as integer, float, string or real number. The numerical data can also be of the category date and time.

Ordinal data represents the data with natural order or ranking. This data has a meaningful sequence such as from low to high or small to large. For example, the educational level is an ordinal data as given below.

A high school is < bachelor's degree < master's degree < Ph. D.

Customer satisfaction level is also a kind of ordinal data.

Unsatisfied < Neutral < Satisfied < Very satisfied.

Health status is also a kind of ordinal data.

Poor health < Fair health < Good health < Excellent health.

For ordinal data we can use statistical techniques such as Median, Mode, and Range.

Textual data is data that is in the form of natural language text. Textual data comes from various sources such as documents, email, social media posts and chats. Textual data is unstructured and its required specialized techniques for processing and analysis. For example, sentences, Paragraphs or Documents can be a Textual data. Words are the main elements in the textual data. Textual data can be classified by using the Context analysis. Grammar, Syntax, Schematics and culture references can also be used for classification. The classification of text consists of techniques such as tokenization, lower casing and stop word removal.

Tokenization refers to breaking the text into smaller words. Lower casing refers to conversion of text to lower case. Stop word removal refers to removal of the non-significant word such as 'the' and 'and'.

Textual data analysis consists of text classification which categorizes the text into predefined labels such as spam or not spam. Sentiment analysis determines the sentiments such as positive, negative or neutral in the text. Topic modelling discovers the topic in the collection of documents. Named entity recognition recognizes names, dates, location and organization in the text. Text summarization concise the summary of a longer text. Text clustering groups similar text data together. Textual data is used in business, healthcare, social media monitoring and in search engines.

Example of Text Preprocessing

Scenario: A company wants to classify customer reviews as Positive or Negative based on sentiment analysis.

1. Sample Dataset (Customer Reviews)

Review ID	Review Text	Sentiment Label
1	"I love this product! It works perfectly."	Positive
2	"Terrible quality. I regret buying it."	Negative
3	"Very satisfied! Fast delivery and great support."	Positive
4	"Not as described. Waste of money."	Negative

2. Text Preprocessing Steps:

(i) Convert text to lowercase

(ii) Remove punctuation and special characters

(iii) Tokenization (Splitting text into words)

(iv) Remove stopwords (e.g., "the", "is", "and")

(v) Convert words into numerical format (TF-IDF or Word Embeddings)

3. Output of Text Preprocessing

Original Review	Processed Review	Sentiment Label (0=Negative, 1=Positive)
"I love this product! It works perfectly."	"i love this product it works perfectly"	1
"Terrible quality. I regret buying it."	"terrible quality i regret buying it"	0
"Very satisfied! Fast delivery and great support."	"very satisfied fast delivery and great support"	1
"Not as described. Waste of money."	"not as described waste of money"	0

Characteristics of data sets

Characteristics of data sets refer to the attributes or properties of the data. These properties define the structure, quality and content of the data. Proper understanding of characteristics of the datasets is useful for preprocessing.

The structure of the dataset can consist of rows and columns. Each row represents an observation of records. Each column represents a variable or feature. Following figure shows the structure of the data.

Example Structure of a Dataset

Column Name	**Data Type**	**Description**
Review_ID	Integer	Unique identifier for each review
Review_Text	String (Text)	The actual text data (customer feedback, product reviews, etc.)
Processed_Text	String (Text)	The cleaned text after preprocessing (lowercasing, punctuation removal, etc.)
Sentiment	Categorical (0/1)	The label for classification (e.g., Positive = 1, Negative = 0)

Fig. 2.5: Structure of data set

The dimensions of the datasets represent its shape. The data type can be numeric, categorical, ordinal or mixed. The data can also be textual or boolean data. Number of columns and variables can be a feature count. The data quality can be completeness, consistency, accuracy, timeliness and uniqueness.

Data Cleaning

Data cleaning is a process of detecting, connecting, removing errors, inconsistency and inaccuracy of the data sets. It is used to improve its quality. It makes data suitable for data analysis.

Following are the goals of data cleaning

1. *Accuracy* – Ensuring data values are correct and free of errors.
2. *Consistency* – Ensuring data is consistent in standard formats and units.
3. *Completeness* – Ensuring there are no missing values.
4. *Relevance* – Remove irrelevant data.
5. *Uniqueness* – Remove duplicate records.
6. *Integrity* – Maintain valid relationships among the data elements.

Whenever there is absence of data then such data elements can be removed by using deletion. Outlier is a datapoint that deviates from the majority of the data in the datasets. It is normally high or low. For example, in a dataset of heights, a

value of 300 cm can be an outlier. Following figures show the data cleaning process.

The tools that are used for data cleaning are spreadsheets tools such as Excel and google sheets. In python, pandas and numpy are also used for data cleaning. For databases, Sql can be used for the cleaning of large datasets.

Example of Data Quality

Consider a data set.

1. Example Dataset with Data Quality Issues

Customer_ID	Name	Age	Email	Mobile Number	Aadhaar Number	Purchase_Amount (₹)	Join_Date	Location	Last_Purchase_Date
101	Amit	25	amit@email.com	9876543210	123456789101	1200	2023-02-10	Mumbai	2023-10-15
102	Priya		priya@email.com	8765432109		4500	2022-08-15		2023-11-10
103	Rahul	29	rahul@email.com	7654321098	567812344321	-850	2023-06-20	Bengaluru	2023-05-20
104	Neha	40	neha.email.com	6543210987	345678905678	3000	2021-12-05	Delhi	2023-10-05
105	Aarav	22	aarav@email.com	5432109876	789045671234	600	2024-01-01	Kolkata	2024-02-01
101	Amit	25	amit@email.com	9876543210	123456789101	1200	2023-02-10	Mumbai	2023-10-15

2. Data Quality Issues & Fixes

Dimension	Problem	Example from Dataset	Solution
Completeness	Missing values	Age missing for Priya, Location missing	Fill with average (for numerical) or most frequent value (for

			categorical)
Consistency	Inconsistent formats	neha.email.com (incorrect email format)	Use regex to validate and correct email format
Accuracy	Incorrect values	-850 in Purchase_Amount (negative values)	Convert negative values to absolute or remove incorrect records
Timeliness	Outdated information	Last_Purchase_Date of 2023-05-20 may be outdated for an active customer	Update records regularly and verify data sources
Uniqueness	Duplicate records	Customer_ID 101 appears twice for Amit	Remove duplicates based on unique Customer_IDs

3. Fixed Dataset After Cleaning

Customer_ID	Name	Age	Email	Mobile Number	Aadhaar Number	Purchase_Amount (₹)	Join_Date	Location	Last_Purchase_Date
101	Amit	25	amit@email.com	9876543210	123456789101	1200	2023-02-10	Mumbai	2023-10-15
102	Priya	30	priya@email.com	8765432109	432187659012	4500	2022-08-15	Chennai	2023-11-10
103	Rahul	29	rahul@email.com	7654321098	567812344321	850	2023-06-20	Bengaluru	2023-11-10
104	Neha	40	neha@email.com	6543210987	345678905678	3000	2021-12-05	Delhi	2023-10-05
105	Aarav	22	aarav@email.com	5432109876	789045671234	600	2024-01-01	Kolkata	2024-02-01

Target Variables

Target variables are also known as dependent variables or outcomes in the given dataset. They represent the result or output. For example, in the customer model the target variable can be customer - yes / no. Dependent variables are those variables of which the value is influenced by other variables.

Predicting the house price, estimating number of items sold, predicting stock price can be the target variables. Following figure shows the different forms of target variables. Target variables are used in supervised learning and for evaluation of model performance.

Example: Predicting Loan Approval (Binary Classification)

In a loan approval dataset, the target variable is Loan_Status, which indicates whether a loan is approved (1) or not (0).

Customer_ID	Age	Income (₹)	Credit_Score	Loan_Amount (₹)	Loan_Status (Target)
101	25	30,000	750	500,000	1 (Approved)
102	32	40,000	680	700,000	0 (Rejected)
103	45	80,000	800	600,000	1 (Approved)
104	28	20,000	600	300,000	0 (Rejected)
105	35	50,000	720	450,000	1 (Approved)

Target Variable: Loan_Status (1 = Approved, 0 = Rejected)
Features: Age, Income, Credit_Score, Loan_Amount

2. Example: Predicting House Prices (Regression)

For a house price prediction model, the target variable is House_Price, which is a continuous value (₹).

House_ID	Area (sq ft)	Bedrooms	Location	Age of House (years)	House_Price (Target) (₹)
201	1200	2	Mumbai	5	85,00,000
202	2000	3	Bangalore	3	1,20,00,000
203	1500	2	Delhi	10	95,00,000

204	1800	3	Pune	8	1,05,00,000
205	1100	1	Chennai	12	75,00,000

Target Variable: House_Price (Continuous)

Features: Area, Bedrooms, Location, Age of House

Example of Data Understanding: E-commerce Customer Dataset

Let's consider an Indian e-commerce platform's customer dataset, where the goal is to understand customer spending behavior.

Sample Dataset

Customer_ID	Name	Age	Gender	City	State	Purchase_Amount (₹)	Payment_Mode	Product_Category	Review_Rating	Signup_Date	Last_Purchase_Date
101	Aman	24	Male	Mumbai	Maharashtra	5,200	UPI	Electronics	4.5	2021-05-10	2024-02-15
102	Priya	32	Female	Delhi	Delhi	8,900	Credit Card	Fashion	3.8	2020-08-20	2024-01-25
103	Ramesh	41	Male	Bangalore	Karnataka	1,200	Cash on Delivery	Groceries	4.0	2019-11-12	2024-02-10
104	Sanya	29	Female	Chennai	Tamil Nadu	12,500	Debit Card	Electronics	4.7	2022-02-05	2024-02-20
105	Yusuf	37	Male	Hyderabad	Telangana	3,400	UPI	Home & Kitchen	3.5	2018-07-15	2024-01-30

2. Data Understanding Steps

A. Identifying Data Types

Column Name	Data Type	Example Value	Type Category
Customer_ID	Integer	101	Numerical (Unique ID)
Name	String	"Aman"	Textual
Age	Integer	24	Numerical
Gender	Categorical	Male/Female	Categorical
City	Categorical	"Mumbai"	Categorical
State	Categorical	"Maharashtra"	Categorical
Purchase_Amount	Float/Integer	5,200	Numerical (Continuous)
Payment_Mode	Categorical	"UPI"	Categorical
Product_Category	Categorical	"Electronics"	Categorical
Review_Rating	Float	4.5	Ordinal (1-5)

Signup_Date	Date	2021-05-10	DateTime
Last_Purchase_Date	Date	2024-02-15	DateTime

B. Data Quality Checks (Indian Context)

Data Quality Dimension	Issues Found	Solution
Completeness	Some customers may not have given a review rating.	Fill missing reviews with an average rating.
Consistency	Payment_Mode has inconsistent spelling (UPI vs upi).	Standardize categories.
Accuracy	Negative values in Purchase_Amount are invalid.	Convert negatives to absolute values or remove incorrect rows.
Timeliness	Last_Purchase_Date might be too old for some customers.	Flag inactive users for re-engagement.
Uniqueness	Duplicate entries in Customer_ID.	Remove duplicate records.

C. Data Distribution & Insights

1. Age Distribution of Customers

- Most customers fall in the 25-40 age range, indicating young professionals and middle-aged buyers.
- Can target college students & working professionals for discounts.

2. Popular Payment Methods

Payment Mode	Number of Customers
UPI	2
Credit Card	1
Debit Card	1
Cash on Delivery	1

UPI is the most preferred payment method in India. Older customers might prefer Credit/Debit Cards. COD (Cash on Delivery) is still used, indicating trust issues with online payments.

3. Customer Engagement

1. Frequent Shoppers: Customers who made a purchase in the last 3 months are active users.
2. Inactive Users: If Last_Purchase_Date > 1 year old, they should receive marketing emails or discount offers.

4. Insights for Business Strategy

High-value customers spend more on Electronics & Fashion → Offer premium deals.

UPI is growing as a payment method → Improve UPI discounts & cashback offers. Inactive users need personalized offers to re-engage them.

Cities like Mumbai, Delhi, Bangalore have high-value transactions → Target ads here.

Check your progress

A. Multiple choice questions

1. Collecting of the data from various sources such as database, files or websites (a) Data cleaning (b) Data reduction (c) Data collection (d) Data splitting answer
2. Which of the following is NOT a step-in data pre-processing? (a) Data collection (b) Data encryption (c) Data cleaning (d) Data transformation
3. What type of data is described as having a natural order or ranking? (a) Categorical Data (b) Numerical Data (c) Ordinal Data (d) Textual Data
4. Which data type includes reviews, tweets, and descriptions? (a) Numerical (b) Ordinal (c) Categorical (d) Textual
5. The most common techniques for converting categorical variables to numerical values are? (a) One-Hot Encoding (b) Normalization (c) Scaling (d) Aggregation
6. What is the purpose of data splitting in pre-processing? (a) Training (b) Validation (c) Testing Sets (d) All above
7. What tool can be used in Python for data cleaning? (a) Matplotlib (b) Sql (c) Pandas and numpy (d) TensorFlow

B. Fill in the blanks

1. Data transformation helps to _____or_____ features for uniform distribution.
2. Tokenization refers to breaking the text into _____.

3. The structure of the dataset can consist of _____and _____.
4. The Row represents an _____ of records.
5. The Column represents a _____or _____.
6. Databases _____ can be used for the cleaning of large datasets.

C. State whether true or false

1. Data pre-processing helps in converting raw data into a clean and usable format.
2. Data collection involves gathering data from various sources like databases, files, or websites.
3. Tokenization is splitting text into multiple datasets words or tokens.
4. Data reduction increases the dimensionality of the data.
5. Ordinal Data has with a meaningful order but without equal differences between values.

D. Short Question Answer

1. What are the steps involved in data pre-processing?
2. Give some examples of data understanding techniques.
3. What is the main purpose of data transformation in pre-processing?
4. List a few methods used for data understanding.
5. What are the key characteristics of a data set?

Module 4. Maths for AI

A company has launched a new product in the market. An advertisement was made to make it popular. But instead of an increase in sales, the company finds the sales go on decreasing. Company collected data. From data analysis I fully realize that the change in advertisement is necessary. This change further increases the sale of product.

Fig: Product Advertisement

In this unit we are going to discuss various aspects of data analysis such as its importance, work glow, structure and various data analysis tools.

Session 1. Data Analysis

Data Analysis

Data Analysis is a process of inspecting, cleansing, transforming and modelling the data. It is used to restore the useful information that is used for decision making.

Importance of data analysis

Data analysis is a very important process that is needed to be carried out in various stages. The importance of data analysis is as given below.

1. ***Improved Decision Making*** – most of the decisions are taken on assumptions. Data analysis provides an insight into the actual data that will lead to the more accurate decisions.
2. ***Improved Efficiency and Performance*** – Through data analysis organisations can find out where the resources are wasted. Data analysis will streamline the processes and optimize operations. This will lead to improved efficiency and performance.
3. ***Predicting Results*** – Through data analysis we can predict the future of the brand or organisation.
4. ***Enhance Problem Solving*** – Data analysis can help to uncover the underlying causes of the problem. Hence the problem solving is enhanced by using data analysis.
5. ***Better Resource Allocation*** – Resource allocation can be effectively performed by using data analysis. Smarter allocation with higher returns is possible through data analysis.
6. ***Competitive advantages*** – organisations can gain a competitive age over the others by identifying gaps in the market so that they can outperform arrivals. This can be achieved through data analysis.

Fig. 1.1: Importance of data analysis

Data Analysis Workflow

Following steps are followed in the data analysis workflow.

Data Collection: collect the data from various sources such as sensors for surveillance.

Data Pre-processing: remove inaccuracies or missing values.

Formatting: format the data for analysis.

Analysis: use statistical tools to detect patterns within the data.

Transformation: normalise the data.

Modelling: Apply machine learning models or statistical models to make predictions.

Interpretation: present the conclusion.

Fig. 1.3: Data Analysis workflow

Data Analysis with respect to AI

In an AI system, we need to collect the data from various sources such as sensors. A large amount of data gets collected in AI. Such data needs to be analysed to extract patterns and we need to prepare it for the machine learning process. This data is also used for extracting features. These features are used further for recognition or classification. Statistical techniques are frequently used in AI for the purpose of data analysis. Some of these techniques are discussed in this chapter and some advanced techniques will be discussed in Grade X.

Fig. 1.4: Data analysis in AI

Data Structure

Everybody has one or other kind of identity card. Let us take an example of a unique ID Aadhaar card issued by the Government of India. In this card you can easily observe that there are a lot of data items such as Name, Date of Birth (DOB), Gender as Male or Female, your photograph and Aadhar number which is unique.

Fig. 1.5: Data structures used in Aadhaar card

In real life we always want to deal with different sets of values such as the name of a student or person. Every person living in India has an Aadhaar number, every city or village in India has a pin code and also every student will score some marks in examinations. Whenever we are dealing with different situations then we are dealing with different sets of values.

Data Structure is the specialized means of organizing and storing data in computers to access the data easily. If the data is stored in an organised manner then it is possible to perform operations on the data more efficiently. There are different ways to organize data and accordingly there are different types of data structures. Some examples of Data Structures are arrays, Linked List, Stack, Queue, etc. Data Structures are widely used in almost every aspect of Computer Science i.e. Operating System, Compiler Design, Artificial intelligence, Graphics and many more. It is the fundamental and key topic to begin with in software

development. The software developers should have a sound knowledge of data structures. Computer programming languages are used to implement algorithms on computers.

Data Structures are the main part of many computer science algorithms as they enable the programmers to handle the data in an efficient way. It plays a vital role in enhancing the performance of software or a program as the main function of the software is to store and retrieve the user's data as fast as possible.

Practical Activity 1.1: Identify data elements in Your school ID card.

Material Required

School ID, Pen, Paper

Procedure

Step 1. Observe your school ID card carefully.

Step 2. Note the contents of ID card such as

Name:	Roll Number: .....
Address:	Photo: .....
School Name:	Schol Symbol : .....
Class:	Symbol: .....
Year:	Signature: .....

Step 3. Note the different data elements in each data item.

Name: Characters

Address: Strings and numbers

School name: Strings

Class: Characters

Year: Numbers

Roll No.: Numbers

Photo: Image

Symbol: Image

Signature: Hand signature

Step 4. Prepare a word file of the content of the ID card.

Basic Terminology

It is important to understand the basic terms before knowing all about the data structures. The data items are classified into primitive and group data items.

Primitive data item – Data is the elementary value or the collection of values. The data elements or items that cannot be further divided are called the primitive data elements. For example, *"Roll Number"* of a student is a unique number and it cannot be further divided into different items.

Group data item – The data elements that can be further divided into data items are called the group items. For example, the *"Student Name"* which can be further divided into three parts such as, First Name, Middle Name and Last Name, such data items can be called as a group data item.

Attribute and Entity – The data items with some property or attributes are called *"Entity"*. This entity can be assigned numeric or non-numeric values. For example, for the property or attribute *"Age"*, you can assign any numeric values such as 17.

The entities with similar attributes are called the entity set. For example, students in a class, or employees in the organisation.

Every attribute of an entity set has a range of values. For example, the range of age can be from 0 to 100. The data with the given attribute is called information or processed data. The data of the student such as marks obtained in the subject can be processed to obtain the information in the form of result as *"pass"* or *"fail"*.

Practical Activity 1.2. Identify the type of data item in school ID card.

Material Required

School ID, Pen, Paper

Procedure

Step 1. Note content of ID card as given in above activity.

Step 2. Identify the type of data item.

Step 3. Roll No. : Primitive data item

Name : Group data item

Age : Value

School name : Group data item

Elementary data organization

Data can be organized in linear data structure the elements are into hierarchy of fields, records and files. Figure 1.6, shows the file containing three records with each record having three fields. Every field has a certain data value associated with it. Let us discuss these terms in detail.

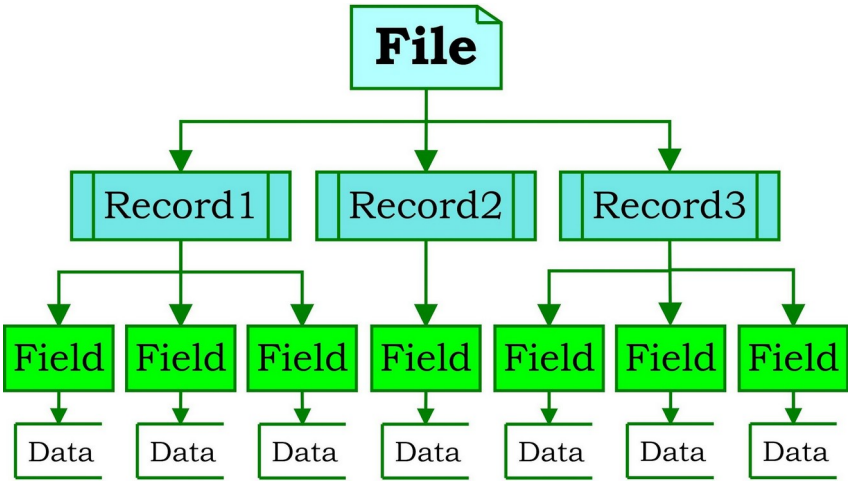


Fig. 1.6 Elementary data organization

Field – Field is a single elementary unit of information representing the attribute of an entity. There can be any number of fields for any single entity. For example, Roll number, Name of student, and Marks scored by the student are different fields.

Record – Record is a collection of field values. For example, for **student entities,** Roll Number, Name and Marks scored are the three different fields. Then the data value for these fields such as "4301, "Anil", "71" form one record.

A record may contain many fields but any one field has a unique value that will not be repeated for other records called *"Primary Key"*. For example, the Roll Number of a student cannot be repeated for other students and hence it is unique and forms a primary key for this record.

File – A File is a collection of various records of one type of entity. For example, the data of several students for these three fields form the several records and collection of all the records will form the file.

A file can have fixed length records or variable length records. Many times, the length of the records is fixed but it is also possible to have a variable length record. It is necessary to specify the minimum and maximum length for each field of the record. For example, we can have 4 digits for Roll Number, 20 characters for Name and 2 digits for Marks.

Practical Activity 1.3 : Identify Field, Records and data in a file.

Material Required

A file, Pen, Paper

Procedure

Step 1. Suppose you have given a file of employee records in an organization.

File: Employee Record

S.No.	Employee No.	Name of employee	Department	DOB	DOJ	Salary
1.						
2.						
3.						

Step 2. In above file

Fields are: Employee No.
Name of employee
Department
DOB
DOJ
Salary

Step 3. Records: Each Row is a record

Step 4. Employee numbers are unique for each record and hence it can be the primary key.

Practical Assignment

Suppose you have been given a file of student records. Identify files and the primary key of records.

In Aadhar card, determine primitive and non-primitive data elements.

Need of Data Structures

Fields, Records and *Files* are the elementary data organisation techniques. But these techniques are not sufficient to process all types of data in many applications in real life. Hence, we require more complex data structures such as arrays, stacks, queues, linked lists, trees, and graphs. As applications are getting complex and the amount of data is increasing day by day, following problems may arise.

Processor speed – To handle a very large amount of data, high speed processing is required, but as the data is growing day by day to the billions of files per entity, processors may fail to deal with that much amount of data.

Data Search – Consider an inventory size of 106 items in a store, if our application needs to search for a particular item, it needs to traverse 106 items every time, resulting in slowing down the search process.

Multiple requests – If thousands of users are searching the data simultaneously on a web server, then there are the chances that a very large server can be failed during that process

In order to solve the above problems, data structures are used. Data is organized to form a data structure in such a way that all items are not required to be searched and required data can be searched instantly.

Advantages of Data Structures

Efficiency – Efficiency of a program depends upon the choice of data structures. For example: suppose, we have some data and we need to perform the search for a particular record. In that case, if we organize our data in an array, we will have to search sequentially element by element. Hence, using arrays may not be very efficient here. There are better data structures which can make the search process efficient like ordered arrays, binary search trees or hash tables.

Re-usability – Data structures are reusable, i.e. once we have implemented a particular data structure, we can use it at any other place. Implementation of data structures can be compiled into libraries which can be used by different clients.

Abstraction – Data structure is specified by the ADT which provides a level of abstraction. The client program uses the data structure through interface only, without getting into the implementation details.

Classification of Data Structure

Data Structures can be classified basically into two categories – *primitive* and *non-primitive* data structure as shown in Figure 1.7. The data structures that cannot be further divided into other sub-items are called the primitive data structures or simple data structures or simply data types. For example, it is not possible to further divide the integer or real number. The primitive data structures have integer, real, character and Boolean data types. The non-primitive data structures are categorized in linear and nonlinear data structures.

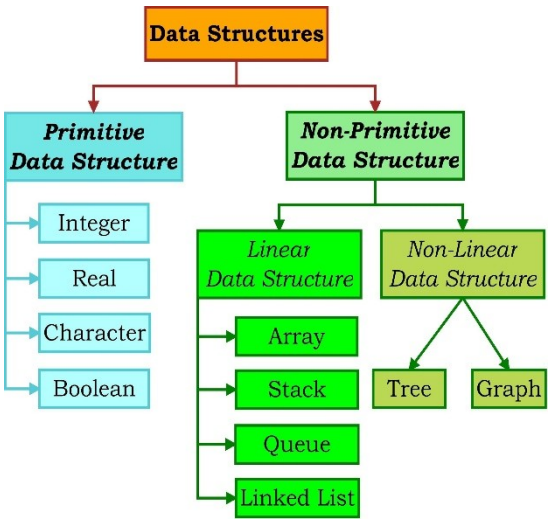


Fig. 1.7: Classification of data structure

Linear Data Structures

A data structure is called linear if all of its elements are arranged in the linear order or in a sequence. In linear data structures, the elements are stored in a non-hierarchical way where each element has the successors and predecessors except the first and last element. For example, the stacks, queues and linked list are the linear data structure.

Types of Linear Data Structures are given below:

Arrays – An array is a collection of similar types of data items and each data item is called an element of the array. The data type of the element may be any valid data type like char, int, float or double.

The elements of array share the same variable name but each one carries a different index number known as subscript. The array can be one dimensional, two dimensional or multidimensional.

The individual elements of the array age are:

age [0], age [1], age [2], age [3], ... age [98], age [99].

Stack – Stack is a linear list in which insertion and deletions are allowed only at one end, called **top**. A stack is an abstract data type (ADT), which can be implemented in most programming languages. It is named as stack because it behaves like a real-world stack, for example: – piles of plates or deck of cards.

Queue – Queue is a linear list in which elements can be inserted only at one end called **rear** and deleted only at the other end called **front**.

It is an abstract data structure, similar to stack. Queue is opened at both ends therefore it follows First-In-First-Out (FIFO) methodology for storing the data items.

Linked List – Linked list is a linear data structure which is used to maintain a list in the memory. It can be seen as the collection of nodes stored at non-contiguous memory locations. Each node of the list contains a pointer to its adjacent node.

Non-Linear Data Structures

This data structure does not form a sequence i.e. each item or element is connected with two or more other items in a non-linear arrangement. The data elements are not arranged in sequential structure. Trees and graphs are the nonlinear data structures.

Trees – Trees are multilevel data structures with a hierarchical relationship among its elements known as nodes. The bottommost nodes in the hierarchy are called **leaf nodes** while the topmost node is called **root node**. Each node contains pointers to point adjacent nodes.

Tree data structure is based on the parent-child relationship among the nodes. Each node in the tree can have more than one child except the leaf nodes. Each node must have a parent node except the root node. Trees can be classified into many categories which will be discussed later in this unit.

Graphs – Graphs can be defined as the pictorial representation of the set of elements represented by vertices connected by the links known as edge. A graph is different from a tree in the sense that a graph can have cycles while the tree cannot have one.

Practical Activity 1.4: Match the following

A	B
1. Linear Array	a. Pictorial representation of set of elements
2. Stack	b. Element insertion at rear end
3. Queue	c. Collection of similar data elements
4. Linked List	d. Insertion and deletion at one end only
5. Tree	e. Multilevel data structure
6. Graphs	f. Each node contains pointer

Ans: 1. (c) 2. (d) 3. (b) 4. (f) 5. (e) 6. (a)

Operations on data structure

It is possible to write computer programs to perform any operation on the data structure. For this it is required to write an algorithm. Algorithm is a set of steps written to perform the operation. Every algorithm is associated with complexity.

The complexity of an algorithm refers to the time and space requirement to execute that algorithm. The most common operations that can be performed on data structures are as follows.

Traversing – Every data structure contains a set of data elements. Traversing the data structure means visiting each element of the data structure in order to perform some specific operation like searching or sorting. For example, if we need to calculate the average of the marks obtained by a student in 6 different subjects, we need to traverse the complete array of marks and calculate the total sum, and then we will divide that sum by the number of subjects i.e. 6, in order to find the average.

Insertion – Insertion can be defined as the process of adding the elements to the data structure at any location. If the size of the data structure is **n** then we can only insert **n-1** data elements into it.

Deletion – The process of removing an element from the data structure is called Deletion. We can delete an element from the data structure at any random location. If we try to delete an element from an empty data structure then **underflow** occurs.

Searching – The process of finding the location of an element within the data structure is called Searching. There are two algorithms to perform searching, Linear Search and Binary Search.

Sorting – The process of arranging the data structure in a specific order is known as Sorting. There are many algorithms that can be used to perform sorting, for example, insertion sort, selection sort, bubble sort, etc.

Merging – When two lists List A and List B of size M and N respectively, of similar type of elements, clubbed or joined to produce the third list, List C of size (M+N), then this process is called merging.

Practical Activity 1.5 : Match the following

A	B
1. Insertion	a. Removing an element
2. Deletion	b. Finding an element
3. Searching	c. Arranging elements
4. Sorting	d. Joining elements
5. Merging	e. Adding an element

1. (e) 2. (a) 3. (b) 4. (c) 5. (d)

Statistical analysis of Data

By using certain statistical and mathematical tools we can easily measure certain features of the data such as Central tendency, Dispersion, and Probability. These techniques are discussed below.

Central tendency – Mean, Mode, Median

A data is a collection of values. Frequently we need to find and mean, Mode and Median called as the measure of the central tendency. It is actually used in indication of the central point of data.

Mean : It is called the average of the given set of values. If

Mean – It is defined as the average value of the data set. If there are *n* number of data values given to you then the mean can be computed by using the formula.

$$\underline{X} = (x_1 + x_2 + x_3 + \ldots\ldots + x_n)/\ n = \frac{\sum_{x=1}^{n} x_i}{n}$$

For example, the given data set is 1, 4, 4, 6, 10 then its mean is

$\underline{X}$= (1+4+4+6+10)/5 = 25/5 = 5

Median – Median of the given data set is the middle number that splits the data set into two halves.

Steps to find the Median.

Step 1. Arrange the data in the increasing order.

Step 2. Determine the number of data values in the data set which is equal to n.

Step 3. If *n* is odd then median is the middle number.

Step 4. If *n* is even, then the median is the average of the two middle numbers.

Practical Activity 1.6. Compute Mean, Median and Mode for the given data set

Given data set is 34, 22, 15, 25, 10

Step 1. The increasing order of the data is 10, 15, 22, 25, 34

Step 2. There are five numbers in the data set i.e., n = 5.

Step 3. Therefore, the median is equal to the median is middle number 22.

Given data set 19, 34, 22, 15, 25, 10

Step 1. The increasing order of the data is 10, 15, 19, 22, 25, 34

Step 2. There are six numbers in the data set i.e., n = 6.

Step 3. Therefore, there are two middle numbers 19 and 22. Therefore median is equal to (19+22)/2=20.5

Observe that mean and median numbers are not from the data set. Mean and median can have only one value. Mean is affected by the extreme values while the median is resistant.

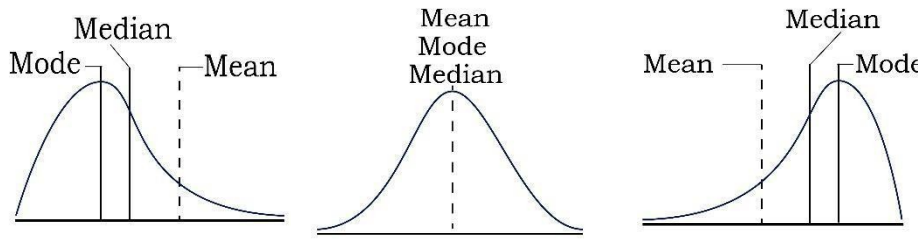
Mode – The most frequent number in the data set is called mode.

Example, suppose we have been provided with the data set 19, 19, 34, 3, 22, 10, 15, 25, 10, 6. In this data set the number that occurs the most is 10, and therefore, Mode is equal to 10.

If the two different numbers appear the same number of times in the given data set, then both of them are called as the mode of the given data set.

Observe that mode is always the number from the data set. It is also possible that mode can have 0 or 1 or more than one value and it is being called 0 Mode, 1 Mode, 2 Mode, and so on.

The relationship between Mean, Median and Mode can be graphically represented in Figure 1.8.



(a) Positive skew *(b) Symmetric distribution* *(c) Negative skew*

Fig. 1.8 Graphical representation of Mean, Median and Mode

2. Measurement of Deviation – Deviation is defined as the difference between the value x and the population mean (μ).

Deviation = $x - \mu$

Deviation can be measured by the parameters called variance and standard deviation.

Variance – It is a measure of how far the values of the data set are from the mean, on average. The average of the squared deviation is the population variance.

Population variance $\sigma^2 = \sum_{x=1}^{n} \frac{(x_i - (\bar{x}))^2}{n}$

Sample variance $S^2 = \sum_{x=1}^{n} \frac{(x_i - (\bar{x}))^2}{n-1}$

Standard deviation – The square root of the variance is called standard deviation.

The population standard deviation

$\sigma = \sqrt{\sigma^2} = \sqrt{\sum_{x=1}^{n} \frac{(x_i - (\bar{x}))^2}{n}}$

Sample standard deviation

$S = \sqrt{S^2} = \sqrt{\sum_{x=1}^{n} \frac{(x_i - (\bar{x}))^2}{n-1}}$

Steps for computing variance and standard deviation.

Step 1. Compute the sample mean $\underline{x}$

Step 2. Calculate the difference $xi - \underline{x}$ for each value in the data set.

Step 3. Calculate the squared difference $(xi - \underline{x})^2$ for each value in the data set.

Step 4. Sum the squared differences $\sum_{i=1}^{n} (x_i - \underline{x})^2$

Step 5. Divide the sum of squared differences to get the variance S^2

Step 6. Compute the square root of the variance to get the standard deviation S.

Example: Compute the variance and standard deviation of the following data in spreadsheet.

x	xi - $\underline{x}$	$(xi - \underline{x})^2$
15	-38	1444
25	-28	784
35	-18	324
45	-8	64
55	2	4

60	7	49
65	12	144
70	17	289
75	22	484
85	32	1024
∑x = 530		∑(xi - $\underline{x}$)^2 = 4610

Total No of item (N)=10

So, Mean ($\underline{X}$) = $\frac{\sum x}{n} = ¿ \frac{530}{10} = 53$

Variance ^{2} = $\frac{\sum (x_i - \underline{x})2}{N} = ¿ \frac{4610}{10} = 461$

Standard Deviation= ^{2} = 461 = 21.47091055

Probability

Probability is the measure of the likelihood of an event occurring. It is expressed by a value between 0 to 1. 0 means impossible and 1 means a certain event. Probability technique is very popular and is used in statistics, machine learning, decision making and every day reasoning to quantify uncertainty.

Probability of event A P(A) is defined as the ratio of number of favourable outcomes to total number of outcomes.

P(A) = Number of favourable outcomes / Total number of outcomes

For example, if the sample space S = {1,2,3,4,5,6} and event A = {2,4,6}, then probability of event A, P(A) = 3/6 = 1/2 = 0.5 or 50%

Consider another example of flipping a coin. Probability of getting at least one head in two flips can be computed as given below.

The sample space S = {HH, HT, TH, TT}

The favourable outcomes are A = {HH, HT, TH}

Therefore, the probability of A i.e. at least one head, P(A) = 3/4 = 0.75 or 75%

Fig. 1.9: Flipping of coin

Practical Activity 1.7: Find the Probability of getting 3 n rolling a die.

Solution

Sample space S = {1, 2, 3, 4, 5, 6}

No. of outcomes n(s) = 6

Let A is the event of getting 3.

n(A) = 1

A= {3}

P (A) = n(A)/n(S) = 1/6

Probability of getting 3 on rolling a die is 1/6 = 0.166 or 1.66%

Practical Activity 1.8: Draw a random card from a pack of cards. What is the probability that the card drawn is a face card?

Solution

There are 52 cards.

Total outcomes n(S) = 52

Let A is an event of drawing face cards.

n(A) = 4*3 = 12 (consider Jack, Queen & King only)

P (A) = n (A)/n(S) = 12/52 = 3/13 = 0.230 or 23%

Normalization of Data

Definition

Normalization is a data preprocessing technique used to scale numerical data into a specific range, usually between 0 and 1 or -1 and 1. This process ensures that different features contribute equally to machine learning models and prevents features with larger values from dominating the results.

Importance of Normalization

1. ***Improves Model Performance*** **–** Helps machine learning algorithms converge faster.
2. ***Eliminates Bias*** **–** Prevents features with larger scales from influencing the model disproportionately.

3. ***Enhances Comparability*** – Makes different datasets comparable, especially when units vary.
4. ***Reduces Computational Complexity*** – Speeds up calculations in distance-based algorithms like k-NN and K-Means.

Common Normalization Techniques

1. **Min-Max Scaling:**
    - Formula:

$$X' = \frac{X - X_{\min}}{X_{\max} - X_{\min}}$$

    - Scales values between 0 and 1.
2. ***Z*-Score Normalization (Standardization):**
    - Formula: Here we use mean and standard deviation.

$$X' = \frac{X - \mu}{\sigma}$$

    - Centers data around mean (0) with a standard deviation of 1.
3. **Decimal Scaling:**
    - Formula:

$$X' = \frac{X}{10^j}$$

    - Scales values based on the maximum absolute value.

Use Cases of Normalization:

1. **Machine Learning:** Used in neural networks, k-NN and SVM.
2. **Image Processing:** Adjusts pixel intensity for better model training.
3. **Finance:** Normalizes stock prices and financial indicators for comparison.

Example of Data Normalisation

Consider a data set:

Sr. No.	**Height**	**Weight**
1	150	60
2	160	72
3	170	85
4	180	90
5	190	100

Above data can be normalized by using Min-Max scaling as given below:

Sr. No.	Height	Weight
1	0.00	0.00
2	0.25	0.27
3	0.50	0.57
4	0.75	0.67
5	1.00	1.00

Above data can be normalized by using Z-Score Normalisation as given below:

Sr. No.	Height	Weight
1	-1.41	-1.27
2	-0.71	-0.42
3	0.00	0.44
4	0.71	0.81
5	1.41	1.27

Application of data analysis

Data analysis can help the organisation to improve their performance in uncertain environments. Hence data analysis is widely used in almost all different types of organisations. Some of them are as given below.

1. Business – In business data analysis can be used for sales forecasting, churn prediction and customer segmentation.

2. Healthcare – data analysis can be used for disease prediction, treatment optimisation and hospital management.

3. Banks/ finance – In banks or financial analysis, data analysis can be used for detection of frauds, credit scoring and risk management.

4. Retail – in the retail sector data analysis can be used for price optimisation and supply chain optimisation.

5. Education – in education data analysis can be used for student performance analysis and for curriculum development.

6. Sports – in sports data analysis can be used for player performance analysis and for prediction of the performance result, sports management and sports events organisation.

Practical Assignment

List down applications of data analysis.

Check in your Progress

A. Multiple choice questions

1. Which step collect the data from various sources such as sensors for surveillance in the data analysis workflow? (a) Transformation (b) Modelling (c) Data Collection (d) Interpretation
2. Which of the following statistical tools to detect patterns within the data in the data analysis workflow? (a) Modelling (b) Data Pre-processing (c) Transformation (d) Analysis
3. What kind of techniques are frequently used in AI for the purpose of data analysis? (a) Statistical (b) Historical (c) Chemical (d) Theoretical
4. Which of the following is a primitive data element? (a) Student Name (b) Roll Number (c) Record (d) File
5. What is a "field" unit of information representing the attribute of an entity in data structure? (a) Records (b) File (c) Single elementary (d) Primary Key
6. Which of the following Collection of similar data elements? (a) Linear array (b) Stack (c) Queue (d) Tree
7. Which of the following Multilevel data structure? (a) Linear array (b) Stack (c) Queue (d) Tree
8. Which of the following Joining elements data structure? (a) Insertion (b) Deletion (c) Merging (d) Sorting
9. A dataset is the middle number that split the data set into two halves. (a) Mean (b) Median (c) Mode (d) Deviation
10. When flipping a coin twice, how many possible outcomes are there? (a) 2 (b) 4 (c) 6 (d) 8

B. Fill in the blanks

1. Data analysis is a process of inspecting, _____, transforming and modelling data.
2. The Data Analysis Workflow missing values are removed is called _____.
3. Data analysis, we can _____ outcomes of a brand or organisation.
4. a single elementary unit of information representing the attribute of an entity _____.
5. Finding an element in data structure _____ .
6. sum of a set of values divided by the total number of values _____
7. _____ normalization uses the maximum absolute value of data.

8. Statistical and mathematical tools such as mean, mode, and median measure _____.
9. A linear data structure where insertion and deletion are done at one end _____.
10. The square root of the variance is called _____.

C. State whether true or false

1. Data analysis helps improve decision making by reducing reliance on assumptions.
2. Data transformation includes normalizing the data.
3. Linked lists collection of nodes stored at contiguous memory locations.
4. Graphs and Trees are examples of non-linear data structures.
5. Tree data structure is based on the only parent relationship among the nodes.
6. The process of arranging the data structure in a specific order is known as Merging.
7. Traversing means visiting each element of a data structure.
8. The most frequent number in the data set is called mean.
9. Deviation can be measured by the parameters called variance and standard deviation.
10. Probability value always lie between 0 and 1.

D. Short Question Answer

1. What is the importance of data analysis in AI?
2. What is elementary data organization in a data structure?
3. How is data structure classified as a linear data structure?
4. How is data structure classified as a non-linear data structure?
5. What are the different operations that can be performed on a data structure?
6. What is mean in statistics?
7. What is median in statistics?
8. What is mode in statistics?
9. What is deviation and standard deviation in statistics?
10. A single card is drawn at random from a standard deck of 52 playing cards. What is the probability that the card drawn is a face card?

Session 2. Data Visualization

The data mostly appears in the form of numbers or values. Such data is difficult to understand or interpret. The data interpretation can be performed by visualizing the data in the form of charts or graphs. By using certain tools, we can visualise the data. Let us first discuss the tools that are available for analysis of data and for visualisation of the data.

Tools for data analysis

There are a variety of tools available for data analysis. Some commonly used tools are – Spreadsheet, Python data analysis tools such as NumPy, Pandas, Matplotlib and SQL data analysis tools.

1. Spreadsheet

spreadsheet is a versatile tool that can be used for data organisation, analysis and visualisation. It consists of rows and columns so that the data can be stored in structured form. By using formulae and functions we can sort and filter the data. Also, we can perform calculations by using built in functions. Spreadsheet support for data visualisation in the form of bar charts, line graphs, pie charts and scatter plots. The data can be imported or exported in various file formats. The most popular spreadsheets are Microsoft Excel, Google Sheets and LibreOffice Calc, Number in Apple.

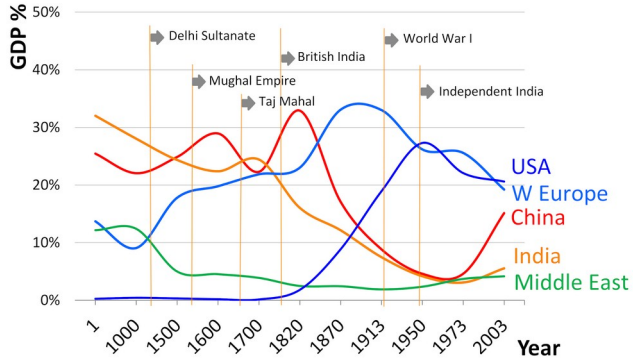


Fig. 2.1: Spreadsheet graphs

Use of Spreadsheets

1. Data Entry and Storage: Used for storing large amounts of data in tabular format such as, sales records and employee details. Example: Maintaining student marks in a school.

2. Data Analysis and Calculation: Perform basic arithmetic such as, addition, subtraction, multiplication and division. Advanced functions like SUM, AVERAGE, COUNT, IF, VLOOKUP help in complex calculations.

3. Data Visualization (Charts & Graphs): Create bar charts, pie charts, line graphs to represent data visually. Example: Analyzing monthly sales trends using a graph.

4. Sorting & Filtering Data: Sort data in ascending or descending order (e.g., ranking students by marks). Apply filters to show only relevant data (e.g., sales of a specific product).

5. Conditional Formatting: Highlight specific values based on conditions (e.g., mark students scoring below 40% in red).

6. Automation with Formulas & Functions: Use Excel formulas to automate calculations. Example: =SUM(A1:A10) to calculate the total of column A.

7. Financial Planning and Budgeting: Used for creating budget plans, tracking expenses, and managing accounts. Example: Monthly household budget tracking.

8. Data Validation and Cleaning: Remove duplicate data and identify missing values. Example: Ensuring phone numbers in a contact list are correctly formatted.

9. Collaboration and Sharing: Google Sheets allows multiple users to edit and view in real time.
Used for teamwork in business and project management.

10. Business and Scientific Applications: Inventory management, payroll processing, project scheduling. Scientific data analysis for research and statistics.

Spreadsheets is used for:

1. Easy to Use – No coding required.
2. Efficient – Quick calculations and analysis.
3. Flexible – Can handle small to medium datasets.
4. Widely Used – Used in schools, businesses, and research.

Python Data Analysis tools

1. NumPy

In core Python libraries, there are tools for data analysis. NumPy (Numerical Python) is a tool that is used to handle numerical computation and array manipulation. It provides efficient operations on large multidimensional arrays and matrices. It provides various functions like Linear Algebra, Fourier Transformation and Random Number generation.

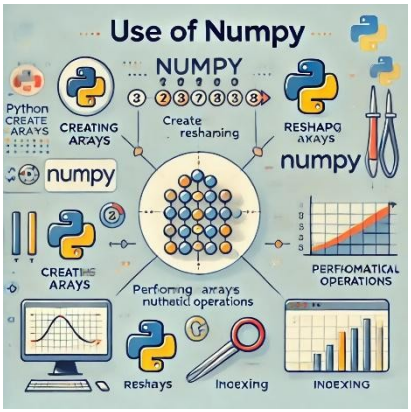


Fig. 2.2: NumPy

Use of NumPy

1. *Creating Arrays (Faster than Lists):* NumPy arrays (ndarray) are more efficient than Python lists. Supports 1D, 2D, and multi-dimensional arrays.
2. *Mathematical Operations:* Perform element-wise operations: Addition, Subtraction, Multiplication, Division.
3. *Statistical & Aggregate Functions:* Mean, median, sum, min, max, standard deviation, etc.
4. *Reshaping & Resizing Arrays:* Convert 1D array to 2D, or change the shape of an array.
5. *Indexing & Slicing (Faster Data Access):* Retrieve specific values or slices from arrays.
6. *Working with Large Datasets (Big Data & AI):* Used in Data Science, Machine Learning, Image Processing, Finance, and Scientific Computing. Faster and memory-efficient compared to lists.

NumPy is used for:

1. *Speed:* Performs operations faster than Python lists.
2. *Efficiency:* Uses less memory for storing large datasets.
3. *Scalability:* Works well with large-scale data processing.
4. *Integration:* Compatible with Pandas, Matplotlib, and SciPy.

2. Pandas

Pandas is a Python tool that is used for data manipulation and analysis for tabular data. It provides data structures such as dataframe (2D) and series (1D). Through Pandas you can write the data in multiple formats.

Fig. 2.3: Pandas

Use of Pandas

Pandas is a powerful Python library used for data manipulation, analysis, and preprocessing. It provides data structures like Series and DataFrame to handle large datasets efficiently.

Key Uses of Pandas

1. ***Data Loading and Storage:*** Read and write data from CSV, Excel, SQL, JSON formats.
2. ***Data Cleaning and Preprocessing:*** Handle missing values, duplicate entries, and incorrect data types.
3. ***Data Filtering and Selection:*** Extract specific rows and columns based on conditions.
4. ***Data Sorting and Aggregation:*** Sort data by specific columns. Perform operations like sum, mean, count, and grouping.
5. ***Data Transformation:*** Modify column values, rename columns, or apply custom functions.
6. **Merging and Joining Data:** Combine multiple datasets for better analysis.
7. ***Data Visualization (Integration with Matplotlib & Seaborn):*** Create charts and graphs to represent data.

Pandas are used for

1. Fast and efficient data handling
2. Easy data cleaning and preprocessing
3. Works well with large datasets
4. Integration with NumPy, Matplotlib, and Machine Learning libraries

3. MatPlotLib

In Python MatPlotLib is used for data visualization and plotting. By using this tool, we can create static, animated and interactive visualisations.

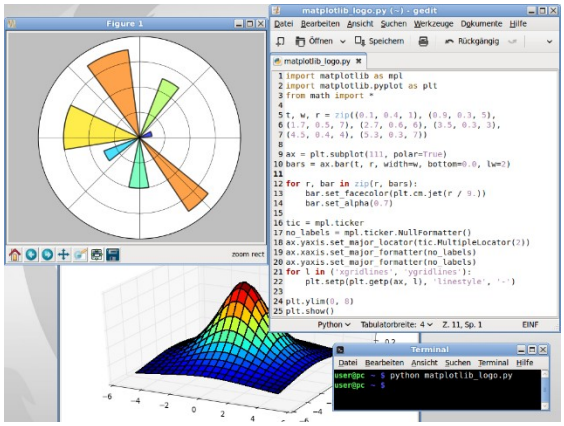


Fig. 2.4: Matplotlib

Uses of Matplotlib

Matplotlib is a powerful Python library used for data visualization. It helps create graphs, charts, and plots to represent data visually, making it easier to analyze trends and patterns.

1. Creating Different Types of Graphs:

Line Plots – Used for tracking trends over time.
Bar Charts – Comparing different categories.
Scatter Plots – Analyzing relationships between two variables.
Pie Charts – Representing proportions.

2. Data Visualization for Analysis: Helps in identifying trends, outliers, and relationships in data.
Example: Analyzing monthly sales using a bar chart.

3. Customization of Graphs: Add titles, labels, legends, grid lines, and colors to make plots more informative.

4. Visualizing Large Datasets: Used in Data Science and Machine Learning to explore large datasets efficiently.

5. Statistical Analysis & Distribution: Create histograms and box plots to understand data distribution.

6. Real-time Data Plotting: Used in finance, engineering, and IoT applications to visualize live data streams.

Matplotlib are used for:

Easy to use – Simple syntax to generate plots.
Highly customizable – Modify colors, styles, and annotations.

Supports multiple formats – Save graphs as PNG, PDF, SVG, etc.
Works with Pandas & NumPy – Perfect for data science and AI projects.

3. SQL data analysis tools

SQL is the structured query language. It is the most widely used tool for data analysis. It has the ability to interact with relational databases. It can extract data and can manipulate it. SQL can also analyse the data that is stored in the tables of the database.

SQL server management studio (SSMS) is a data inspection tool. It can query the data and can retrieve that data. Data visualization is also possible by using these tools. Insertion of the data, updating of the data, deletion of the data can also be achieved through this tool. MySQL, PostgreSQL, Dbeaver, Datagrip, TableAU and PowerBI are certain other SQL data analysis tools that can be used for analysis and visualisation of the data.

Fig. SQL data analysis tools (MySQL, PostgreSQL, Dbeaver, Datagrip, TableAU and PowerBI)

Uses of SQL Data Analysis Tools

1. Data Storage & Retrieval: Store large volumes of structured data in databases. Retrieve specific records using SELECT queries.
2. Data Filtering & Sorting: Filter data using WHERE conditions. Sort results using ORDER BY.
3. Data Aggregation & Summarization: Use functions like SUM(), COUNT(), AVG(), MIN(), MAX() to summarize data.
4. Data Cleaning & Transformation: Remove duplicates, handle missing values, and format data using SQL functions.
5. Data Joining & Merging: Combine multiple tables using JOIN operations.
6. Business Intelligence & Reporting: Extract insights for decision-making by

running complex queries. Used in finance, marketing, healthcare, and e-commerce.

7. Integration with Data Visualization Tools: SQL data can be integrated with Power BI, Tableau, and Python (Pandas, Matplotlib) for visualization.

Popular SQL Data Analysis Tools:

MySQL – Open-source, widely used.
PostgreSQL – Advanced features for large-scale data.
Microsoft SQL Server – Enterprise-level tool for analytics.
SQLite – Lightweight and portable.
Google BigQuery – Cloud-based SQL analysis tool.

Now we will illustrate the spreadsheet tool through Practical Activity 2.1.

Steps to create the charts – Bar chart, Line chart or Pie chart

Step 1. Open the spreadsheet package Microsoft Excel or LibreOffice Calc with a new worksheet.

Step 2. Enter the data in the worksheet at appropriate rows and columns

Step 3. Save the worksheet by proper name.

Step 4. To display the data graphically in the form of a bar chart, line chart or pie chart, select the data cells by dragging the mouse.

Step 5. Click on **Insert > Chart** and select the desired chart option from the selection window.

Step 6. Then click on the Finish button. The data will be displayed graphically as per the selection of bar chart, line chart or pie chart.

Practical Activity 2.2: Represent the given data values graphically in bar chart, line graph and pie chart using spreadsheet software.

Step 1. You have been provided the marks of a few students as given below.

Student	Marks
David	74
Hiba	78
Kannan	69
Kushaal	69
Manpreet	62
Pawani	75
Smitha	74
Vibhanshu	81

Draw a bar chart Application of data analysis using the spreadsheet software to represent the given data graphically as shown in Figure 2.5.

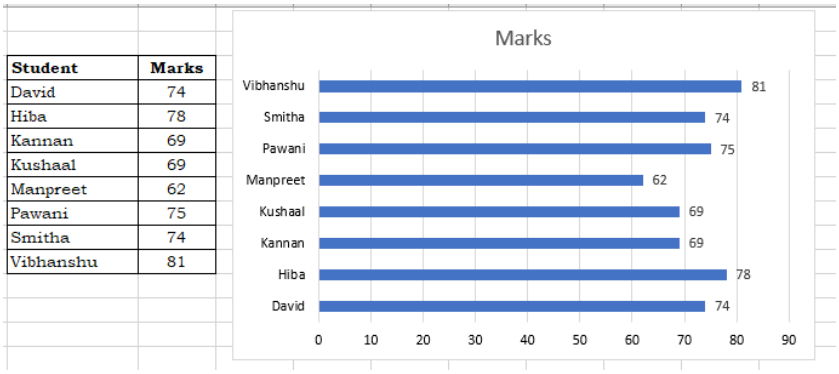


Fig. 2.5 : Bar chart drawn for the given data values

Step 2. You have been provided the data of temperature recorded in one week as given below.

Date	Temperature
21/07/21	33.5
22/07/21	34.2
23/07/21	36.4
24/07/21	38.1
25/07/21	35.9
26/07/21	36.8
27/07/21	34.2
28/07/21	39.5
29/07/21	34.8
30/07/21	36.5
31/07/21	37.5

Draw a line chart using the spreadsheet software to represent the given data graphically as shown in Figure 2.6.

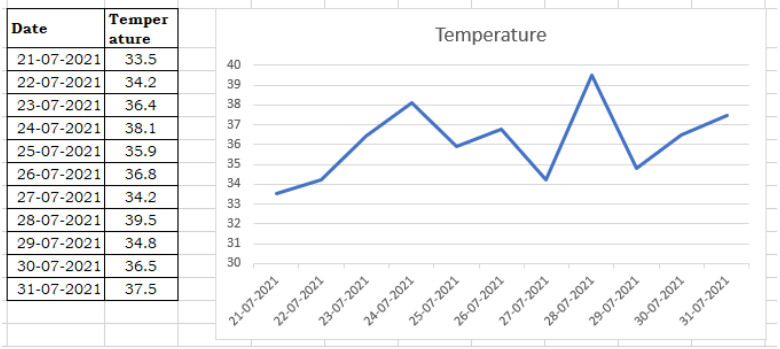


Fig. 2.6 : Line chart drawn for the given data values

Step 3. You have been provided the data of percentage of market share of different computer company as given below.

Company	% in market share
HP	22
Dell	33
Lenovo	13
Asus	15
Acer	17

Draw a pie chart using the spreadsheet software to represent the given data graphically as shown in Figure 2.7.

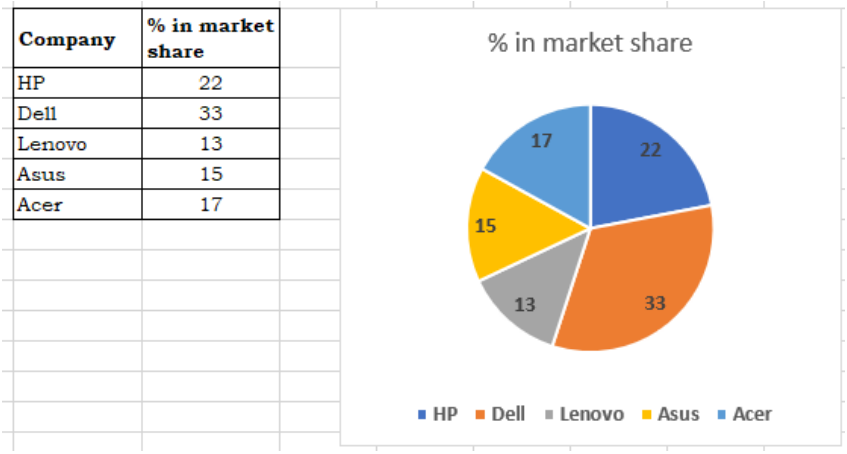


Fig. 2.7 : Pie-chart drawn for the given data values

Check in your Progress

A. Multiple choice questions

1. Which of the following is NOT a spreadsheet application? (a) Google Sheets (b) LibreOffice Calc (c) Microsoft Word (d) Microsoft Excel
2. Which function in Spreadsheets additions a range of cells? (a) COUNT () (b) VLOOKUP () (c) AVERAGE () (d) SUM ()
3. Which Python library is used for data manipulation and analysis for tabular data? (a) NumPy (b) Pandas (c) MySQL (d) Matplotlib
4. What type of Graphs is used to Representing proportions? (a) Pie chart (b) Line chart (c) Bar chart (d) Scatter plot
5. Which Graph is use to tracking trends over time? (a) Bar chart (b) Pie chart (c) Line chart (d) Scatter plot
6. What is the full form of SQL? (a) Structured Quick Language (b) Simple Query Language (c) Structured Query Language (d) Standard Question

Language

7. Which Python library is used for data visualization and plotting graphs? (a) Pandas (b) Matplotlib (c) NumPy (d) MySQL
8. Which of the following is a lightweight and portable SQL Data Analysis tool? (a) MySQL (b) Microsoft SQL Server (c) PostgreSQL (d) SQLite
9. The function that is not a Statistical & Aggregate Function in NumPy is? (a) Mean (b) Median (c) Multiplication (d) Standard deviation
10. How to convert a 1D array to a 2D array or change the shape of an array, using Python and NumPy? (a) Reshaping & Resizing Arrays (b) Mathematical Operations (c) Statistical (d) Indexing & Slicing Array

B. Fill in the blanks

1. In a spreadsheet, data is stored in _____ and _____.
2. _____ is used for numerical computation and array manipulation in Python.
3. _____ is used for creating line plots, Bar charts, Scatter plots and pie charts in Python.
4. NumPy is provides various functions like Linear Algebra, _____ and Random Number generation.
5. Structured query language ability to interact with _____.

C. State whether true or false

1. Spreadsheets can be used to create graphs, charts and plots.
2. NumPy arrays are supports 1D, 2D and multi-dimensional arrays.
3. In Python MatPlotLib is used for data manipulation and analysis.
4. Pandas uses for Integration with NumPy, Matplotlib, and Machine Learning libraries.
5. Pandas does not support read and write data from CSV, Excel, SQL, JSON Formats.
6. SQL server management studio (SSMS) is a data inspection tool.

D. Short Answer Questions

1. What is a spreadsheet used for in data analysis?
2. What is SQL used for in data analysis?
3. What is NumPy used for in Python data analysis?
4. What are the key uses of Pandas?
5. What are the uses of Matplotlib?

Module 5. Machine Learning

Sachin was watching a news channel. There was news that a certain school in Bangalore has purchased a Robot to teach to the students. He was surprised that even a robot can teach like a human teacher. Later on, Sachin understood that this is possible because of machine learning.

Fig: Robot Teacher

In this unit, the students will be able to understand the concept of Machine Learning and its connection with AI. We will also discuss the importance of machine learning, applications of machine learning in various industries, and its workflows. You will understand the different ways in which machines can learn or can be taught using different methodologies. The main types of machine learning methods – Supervised, Unsupervised and Reinforcement methods will be introduced to the students.

Session 1. Introduction to Machine Learning

Introduction

Machine Learning (ML) is a part of artificial intelligence (AI) that focuses on teaching computers to learn from data and make decisions without being explicitly programmed. Unlike traditional programming where developers provide precise instructions, ML algorithms learn from patterns and relationships in data. This allows them to generalize and make decisions on new, unseen data.

Machine learning is a technique that uses statistical models and algorithms to enable the computer to improve their performance by using the input data. It does not require an explicit program for each specific task.

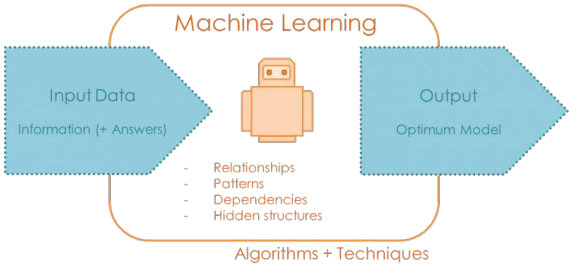


Fig. 1.1: Machine Learning (Redraw)

The input data to the computer can be in the form of text, images, numbers, audio or video. This data set is used to create an experience so that machines can understand relationships, patterns, dependencies and hidden structure among the data elements.

For example, we can use machine learning in identification of the disease for given medical data.

How a Machine Learns

Let's take an example of Email Spam Detection to understand how a machine learns.

Step 1. Data Collection

The machine is given a dataset of emails, labeled as Spam (1) or Not Spam (0).

Email Text	Label (Spam or Not Spam)
"Congratulations! You won the lottery."	1 (Spam)
"Meeting at 5 PM, see you there!"	0 (Not Spam)
"Get free gifts now, click the link!"	1 (Spam)
"Project update attached, please review."	0 (Not Spam)

Step 2. Feature Extraction

The machine identifies important words like "win", "lottery", "free", "click", which are common in spam emails.

Step 3. Training the Model

Using a Machine Learning Algorithm such as, Naïve Bayes, Decision Trees, the machine learns the pattern from past emails.

If an email contains words like "win" or "free", it's likely Spam.
If an email contains words like "meeting" or "project", it's likely Not Spam.

Step 4: Testing & Predictions

Now, when a new email arrives, the machine checks the words in the email and predicts whether it is Spam or Not Spam.

Example:
New Email: "You won a free iPhone! Click to claim."
Machine detects "won" & "free" Predicts Spam (1).

New Email: "Team meeting at 10 AM tomorrow."
Machine detects "meeting" Predicts Not Spam (0).

Step 5: Continuous Learning for improvement

If a user corrects a wrongly classified email, the machine updates its learning and improves future predictions.

This is how Machine Learning works! It learns from past data, finds patterns, and makes predictions on new data.

Importance of Machine Learning

1. Data driven decision making – ML algorithms learn from various types of data, including images, text, sensor readings, and historical records. Instead of hardcoding rules, ML models identify patterns and relationships within the data to make predictions or decisions.

2. Automation of processes – Some common ML algorithms include decision trees, neural networks, and support vector machines. Trained models serve as representations of the learned data, such as recognizing handwritten digits using a neural network. Many manual processes can be automated by using machine learning.

3. Improved efficiency and productivity – The applications of ML are vast and diverse. It powers recommendation systems like those used by Netflix, speech recognition, medical diagnosis, and autonomous vehicles. ML is also behind chatbots, personalized ads, and fraud detection systems. It ultimately improves the efficiency and productivity of organisations.

4. Enhance accuracy – ML transforms data into knowledge, enabling computers to learn, adapt, and make decisions more accurately.

5. Scalability – Artificial intelligence (AI) and machine learning (ML) have significantly impacted various aspects of our lives. From transportation and finance to healthcare and entertainment, AI algorithms are pervasive. They power self-driving cars, fraud detection systems, personalized shopping experiences, and virtual assistants like Siri and Alexa. As technology continues to evolve, the influence of AI and ML is only expected to grow, shaping the future of our society and culture.

Workflow of Machine Learning

Workflow of Machine Learning involves a series of steps that are designed to develop, evaluate and deploy a machine learning model.

Step 1. Problem definition – clearly define the problem that needs to be solved by using a machine learning model. State the objectives such as classification, regression or clustering. State the key matrices such as accuracy, precision and recall. The expected output can be predicted by label or probability.

Step 2. Data collection – get the relevant and sufficient data from various sources such as database and web scraping. Ensure that the data set contains features. Collect diverse and representative data.

Step 3. Data preprocessing – clean, transform and prepare the data for analysis and modelling.

Step 4. Data analysis – understand the patterns in data and visualise the data.

Step 5. Model selection – choose the appropriate algorithm depending upon the nature of the problem such as support vector machine or neural network, or K-means clustering.

Step 6. Model training – train the chosen model for training the data to learn the patterns.

Step 7. Model evaluation – access the performance of the model by using accuracy, precision and recall.

Step 8. Model deployment – integrate the trained model into production for real world use.

Step 9. Maintenance – continuously monitor the model performance and update it.

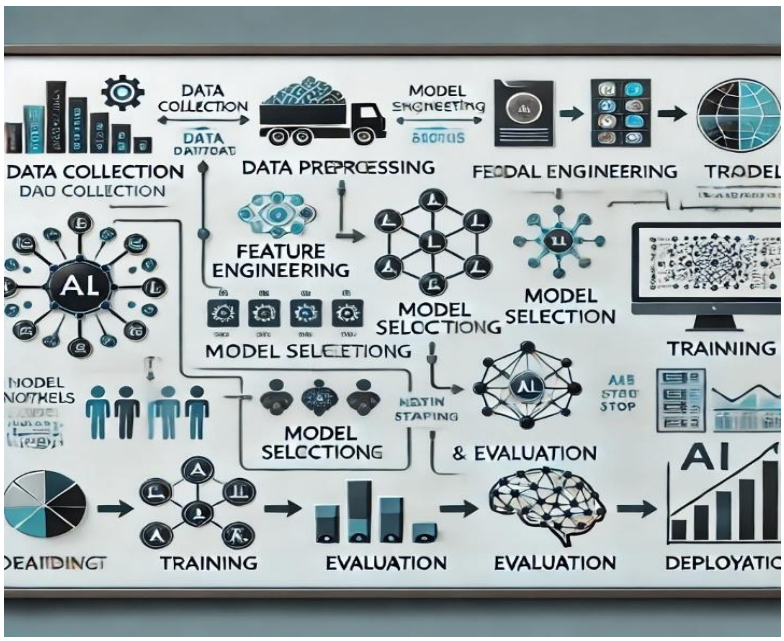


Fig. 1.2: Machine Learning Workflow

Applications of machine learning in various industries

ML can be used across diverse industries and fields. Some of the applications of ML are given as below.

1. Healthcare – ML can be used for disease diagnosis, personalised medicine, forecasting patient outcome, drug discovery and medical image analysis.

2. Bank and Finance – ML can be used for fraud detection, credit scoring, algorithmic trading, risk management and for solving customer queries.

3. Retail and E-Commerce – ML can be used to suggest products based on user preference and browsing history. For example, Amazon and Netflix use ML algorithms. Price optimization and customer segmentation i. e. Grouping of customers based on purchase behaviour can be done by using ML.

4. Transportation – Self driving cars can make use of ML algorithms. Traffic prediction can be performed by using ML.

5. Manufacturing – Detection of defects in production and supply chain optimisation can be achieved through ML.

6. Energy – Smart grids, prediction of energy production from solar can be achieved through ML.

7. Entertainment – Suggestion of movies, enhancement in gaming experience and smart content creation can be achieved through ML.

8. Security – Cyber security, biometric authentication, detection of unusual activities can be performed by ML.

9. NLP – language translation, sentiment analysis, virtual assistant can make use of ML.

10. Environment – Climate modelling, wildlife conservation, waste management can be achieved through ML.

11. Human resource – selection of candidates, retention of employees, analysing productivity can be achieved through ML.

Fig: Applications of Machine Learning

Check in your Progress

A. Multiple choice questions

1. What is Machine Learning a part of? (a) Database Management (b) Artificial Intelligence (c) Web Development (d) Data Entry
2. In machine learning, cleaning, transforming, and preparing the data for analysis and modeling is called (a) Data Pre-processing (b) Data analysis (c) Data Collection (d) Model evaluation
3. In machine learning, understanding the patterns in data and visualizing the data is known as (a) Data Pre-processing (b) Data analysis (c) Data Collection (d) Model evaluation
4. In machine learning, input data to the computer can be in the form of (a) Text (b) image (c) Numbers (d) All above
5. Which industry uses machine learning for fraud detection, credit scoring, algorithmic trading, and risk management? (a) Healthcare (b) Retail (c) Banking and Finance (d) Entertainment

B. Fill in the blanks

1. Machine Learning is a part of _____.
2. Machine learning is a technique that uses statistical models and _____.

3. In Machine Learning, _____ to the computer can be in the form of text, images, numbers, audio or video.
4. In a Data Collection get the relevant and sufficient data from various sources _____ and _____.
5. Clean, transform and prepare the data for analysis and modelling is called _____.

C. State whether true or false

1. Machine Learning helps computers make decisions based on data.
2. Machine Learning requires computers to be explicitly programmed for every task.
3. Machine learning models not identify patterns and relationships within the data to make predictions or decisions.
4. Retail companies like Amazon and Netflix use ML algorithms to suggest to suggest products based on user preference and browsing history.
5. Self-driving cars do not use Machine Learning algorithms.

D. Short Answer Questions

1. What is Machine Learning?
2. What is the importance of Machine Learning in real life?
3. What is Problem Definition in the workflow of Machine Learning?
4. Explain Retail and E-Commerce in application of Machine Learning.
5. What can be used for Machine Learning in Transportation?

Session 2. Types of Machine Learning

Types of Machine Learning

There are different types of machine learning such as Supervised Learning, Un-Supervised Learning, Reinforcement learning, and Deep Learning.

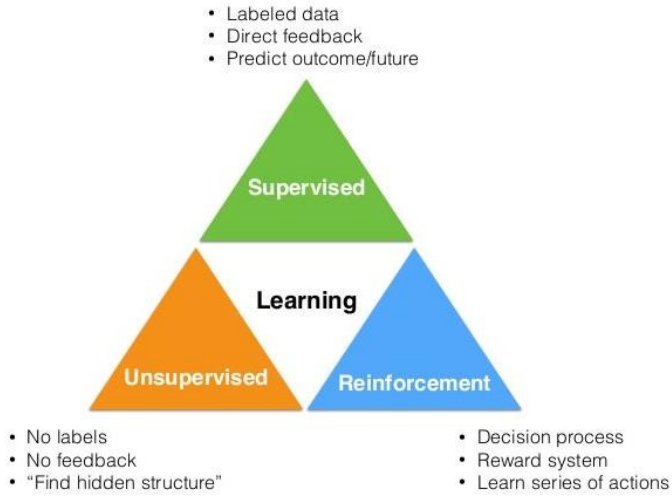


Fig. 2.1: Types of Machine Learning

Supervised Learning

It is a type of machine learning. In supervised learning an algorithm is trained on the labelled data. That is the training data set includes – input-output pairs. The model learns to predict the output based on the input. For example, a face recognition system can recognise the given input face if the face is available in the face database.

Fig. 2.2: Face recognition

The goal of supervised learning is to learn a mapping function from input variables to output variables so that the model can make predictions on unseen data. Examples of supervised learning algorithms include linear regression, logistic regression, decision trees, support vector machines (SVM), and neural networks.

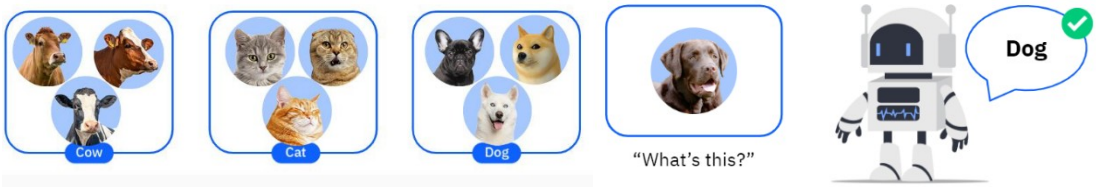


Fig. 2.3: Supervised Learning

Figure x shows that when images of cows, cats and dogs are used for training and when we ask a machine an unknown image then the machine can classify it as Dog.

Another example of supervised learning is handwritten digit recognition.

How It Works:

1. ***Labeled Dataset:*** The system is trained on thousands of images of handwritten digits (0-9), each labeled with the correct number.
2. ***Feature Extraction:*** The model analyzes pixel values, edges, and patterns in the images.
3. ***Training the Model:*** A machine learning algorithm (such as a neural network or support vector machine) learns to associate image features with digit labels.
4. ***Prediction:*** When given a new handwritten digit, the model predicts the number.
5. ***Accuracy Improvement:*** With more training data and optimization, the model gets better at recognizing digits accurately.

This technique is widely used in applications like postal mail sorting, bank check processing, and CAPTCHA solving.

Un-Supervised Learning

It is also a type of machine learning. It deals with unlabelled data. In unsupervised learning the algorithm tries to learn the underlined structure or distribution in the data without explicit output labels. For example, clustering of objects is unsupervised learning.

Fig. 2.4: Unsupervised learning

Figure 2.4 shows that when images are used for training then features such as pointed ears, folded ears and color are used for classification or recognition.

The goal of unsupervised learning is to explore and discover inherent structures or relationships within the data, such as clusters, associations, or anomalies. Examples of unsupervised learning algorithms include k-means clustering, hierarchical clustering, principal component analysis (PCA), and autoencoders.

Example:

An example of unsupervised learning is customer segmentation in marketing.

How It Works:

1. ***No Labeled Data:*** The system receives customer data (e.g., age, purchase history, browsing behavior) but without predefined categories.
2. ***Pattern Discovery:*** The algorithm (like K-Means Clustering) groups customers with similar behaviors.
3. ***Segmentation:*** The system identifies clusters, such as:
    - Frequent Shoppers (high spenders, regular buyers)
    - Discount Seekers (only buy during sales)
    - Casual Browsers (visit often but buy rarely)
4. Business Application: Companies use these insights to tailor marketing strategies, such as personalized promotions for different groups.

Other unsupervised learning examples:

- Anomaly Detection such as, fraud detection in banking.
- Topic Modeling that is, grouping similar news articles.
- Recommendation Systems such as Netflix suggest movies based on viewing patterns.

Reinforcement Learning

Reinforcement learning is a type of machine learning where an agent learns to make decisions by interacting with an environment to maximize cumulative rewards. In reinforcement learning, the agent learns through trial and error by taking actions and receiving feedback from the environment in the form of rewards or penalties. The goal of reinforcement learning is to learn a policy or strategy that guides the agent to take actions that lead to the highest cumulative reward over time. Reinforcement learning is commonly used in scenarios where the agent must make a sequence of decisions over time, such as playing games, controlling robots, or managing financial portfolios.

Examples of reinforcement learning algorithms include Q-learning, deep Q-networks (DQN), policy gradients, and actor-critic methods.

Following figure shows an example of a Reinforcement learning environment. There is a Robot and the bottle on the floor. Now the robot wants to get the bottle (Reward) by trial and error. So, the robot moves in a different direction to get the bottle. The movement of the robot is an Action. Finally, the Robot was able to get the bottle (Reward) in upward direction.

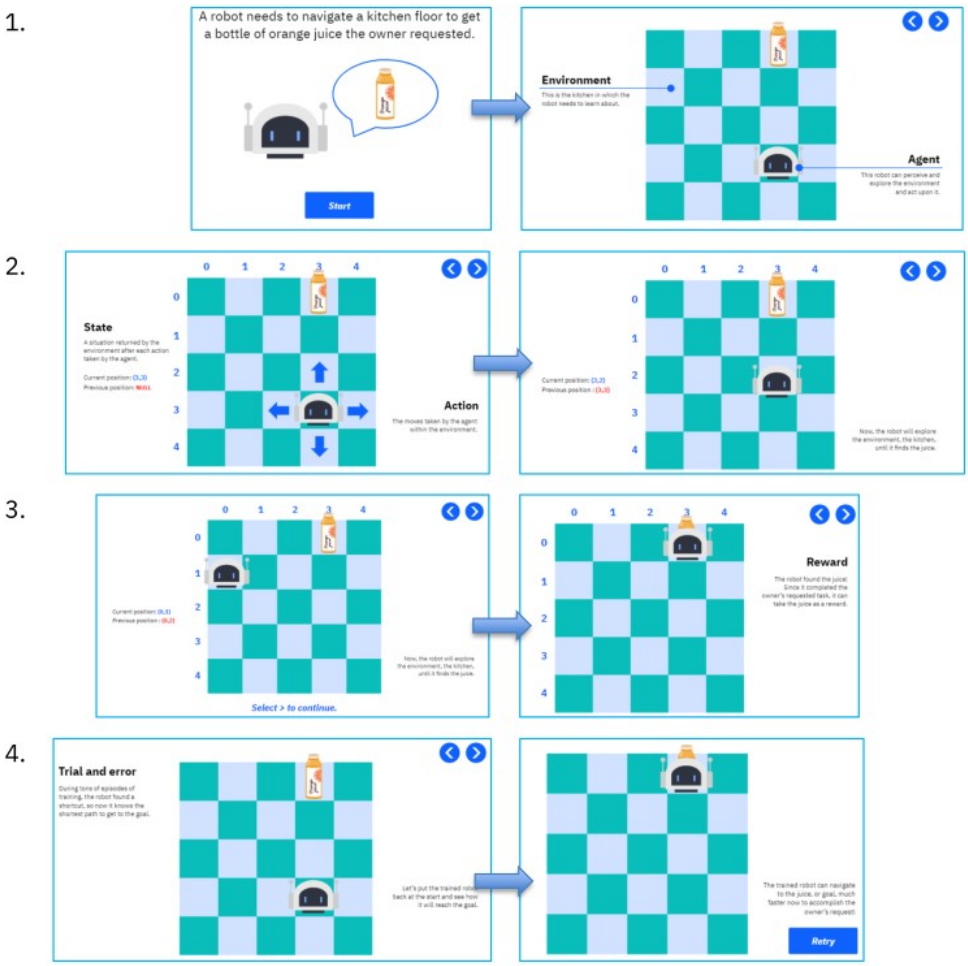


Fig. 2.5: Reinforcement learning

A classic example of reinforcement learning (RL) is training a robot to walk.

How It Works:

1. *Agent:* The robot is the learning agent.
2. *Environment:* The virtual or real-world space where the robot moves.
3. *Actions:* The robot tries different movements (e.g., shifting weight, adjusting legs).
4. *Rewards and Penalties:*
    - If the robot takes a step without falling Receives a positive reward.
    - If the robot loses balance and falls Gets a penalty.

5. *Learning Process:*
    - The robot continuously improves by learning which movements maximize rewards.
    - Over time, it optimizes its walking ability without explicit programming.

Real-World Applications of RL:

- Self-driving cars that learn to drive safely in traffic.
- Game AI where AI defeats human players in Chess.
- Robotic control for optimizing industrial robots for efficiency.
- Stock trading bots for learning to maximize investment returns.

Practical Activity 2.1. Illustrate the clustering of images.

Material Required: Pen paper, 10 photographs.

Procedure

Step 1. Take 10 Photographs.

Step 2. Observe the first photo and classify it as Face image/Nature photo/ Building photo/ others

Step 3. Repeat step 2 for all 10 photos.

Step 4. Write a number of Face photos/Nature Photos/Building Photos & others.

Step 5. You are making use of unsupervised learning.

Deep Learning

It is a subset of machine learning. Deep learning makes use of neural networks. It is used to model complex patterns in the data. For example, image recognition makes use of deep learning.

Example:

An example of deep learning is speech recognition, like how Siri, Google Assistant, and Alexa understand human voice commands.

How It Works:

1. *Neural Network Architecture:*
    - A deep learning model, such as a Recurrent Neural Network (RNN) or Transformer (like in GPT models), processes audio data.
2. *Feature Extraction:*
    - The system analyzes voice patterns, phonetics, and speech structures.
3. *Training with Large Datasets:*
    - The model is trained on thousands of hours of speech data in multiple languages and accents.
4. *Prediction and Output:*
    - When a user speaks, the system converts the audio into text and determines the appropriate response.
5. *Continuous Learning:*
    - The AI improves over time as it processes more speech inputs and user feedback.

Other Deep Learning Applications:

- Self-driving cars (real-time object detection with CNNs)
- Medical imaging (detecting diseases from X-rays & MRIs)
- Deepfake technology (generating realistic face-swaps)
- Machine translation (Google Translate using Transformers)

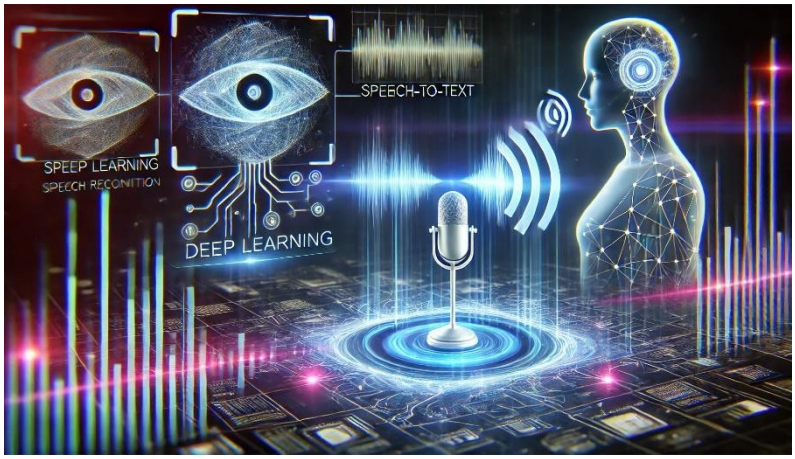


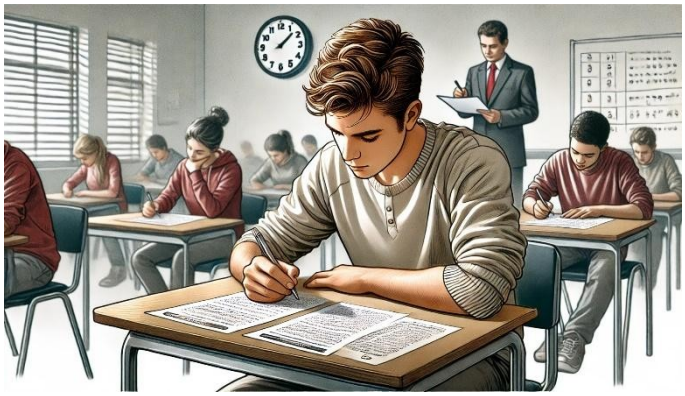
Fig. 2.9: Deep learning

Practical Activity 2.2: Illustrate the Prediction of your exam result.

Material Required: Pen, Paper

Procedure

Step 1. Suppose you have appeared for the 1st terminal examination in your school.



Step 2. At the end of the examination, take question papers and apply your thoughts on how much you can score in each paper.

Step 3. Write your prediction of marks.

Step 4. After declaration of result compare actual result with your prediction.

Step 5. Observe that you have used your brain for prediction.

Image Processing: It is a process that involves manipulation and analysis of digital data sets for images. It is used to improve the quality of image, extract information or achieve a desired effect.

Fig. Image processing example (blur to clear image)

Image Segmentation: It is a process of dividing a given image into meaningful regions or objects. This step helps to isolate different parts of the image. Detection of edges, counters or animals in image can be done by using segmentation.

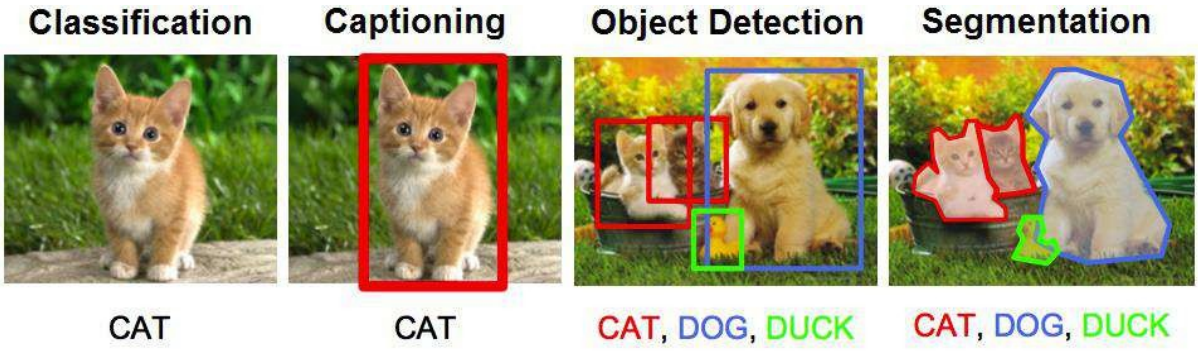


Fig. 2.12: Segmentation

Practical Activity 2.3: Illustrate the concept of Deep Learning

Material Required: Pen, Paper

Procedure

Step 1. Make get together of nearby child of play school and their parents.

Step 2. Ask all parents to get mixed with each other.

Step 3. Now ask children to identify their parents.

Step 4. All children can easily identify their parents.

Step 5. Actually, they are making use of deep learning for identification.

Object Detection: It is a technique that is used to identify and locate objects within an image. Position of each object is identified and it draws the bounding boxes around them. This technique is useful in autonomous driving, security surveillance and augmented reality. Face Detection is one example of object detection.

Image Captioning: It is a process that involves generating descriptive textual captions for the given image. It accurately describes the content of an image.

Image Classification: It is a technique that is used to assign a label and category to an image based on its visual contents. It is very widely used in medical diagnosis, autonomous driving, security systems and in social media.

Fig. 2.16: Face tagging in facebook

Practical Activity 2.4: Illustrate the concept of Image Segmentation

Material Required: Pen, Paper, Photo containing animals.

Procedure

Step 1. Collect 10 Photo containing animals

Step 2. By pen draw borders for each animal image in photos.

Step 3. Count the number of animals.

Step 4. Making each animal's image is just nothing but image segmentation.

Computer Vision: It is a field of AI that enables machines to interpret and understand the visual information from images and videos. Computer vision is widely used in image classification, object detection and face recognition.

Face Recognition: It is a computer vision technology. It identifies or verifies a person's identity using their facial features. It has various applications such as security systems, access control, social media tagging and customer interaction.

Practical Activity 2.5: Illustrate the concept of Object Detection

Material Required: Pen, Paper, Photos

Procedure

Step 1. Take 10 photos containing various images.

Step 2. By using a pen, mark face images in those photos.

Step 3. Count the number of fake images.

Step 4. This process is called face detection or object detection.

Speech Recognition: It is an AI technology that enables machines to understand and interpret human speech. It converts spoken language into text. It allows users to interact with devices through voice commands. It is very widely used in various applications such as Apple Siri, Google Assistant and Amazon Alexa. There are several speeches to text writers or programs available.

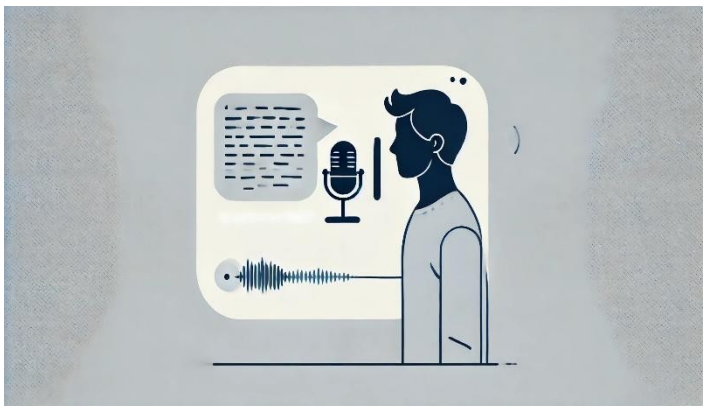


Fig. 2.24: Speech to text image

Natural Language Processing: It is a sub field of AI. It focuses on interaction between computers and humans through natural language like English. It enables machines to understand, interpret, generate and respond to human language in a meaningful and useful way. NLP is a combination of computational linguistics and machine learning.

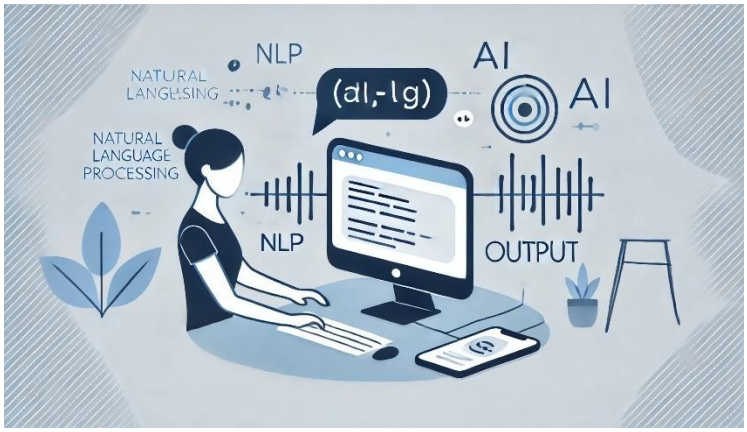


Fig. 2.25: Natural Language Processing

Practical Activity 2.6: Illustrate the concept of Computer vision

Material Required: Pen, Paper, Photographs

Procedure

Step 1. Collect 10 Photographs.

Step 2. Take the first photo and observe it.

Step 3. Now write what this photo contains.

Step 4. Repeat the process for all 10 Photos.

Video Processing: It is a process that involves analysing, manipulating and enhancing video data. It is used to extract meaningful information for performing a specific task. It is used in surveillance, entertainment, healthcare, autonomous driving and sports analytics.

Fig. 2.28: Autonomous driving

Internet of Things: IoT is a network of physical devices, sensors and other objects connected to the internet. They can collect and exchange data. It is used in various fields such as smart homes, healthcare, and agriculture.

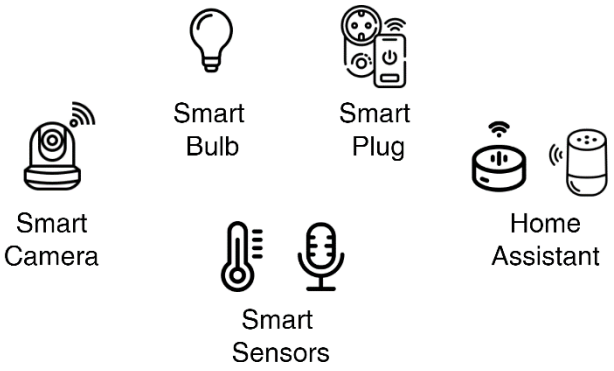


Fig. 2.29: Internet of Things

Translator: It is a multilingual machine translation service developed by many organisations such as Google, Microsoft and Apple. It provides instant translation for text, speech, images and entire web pages across over 100 languages. This software makes use of neural machine translation to enhance the accuracy.

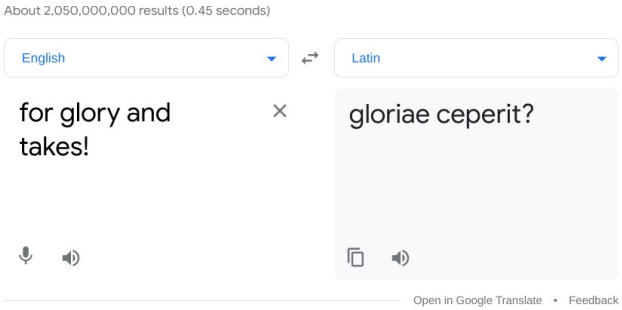


Fig. 2.30: Google Translator

Practical Activity 2.10. Illustrate the concept of Speech Recognition

Material Required: Mobile phone, speech to text converter

Procedure

Step 1. Check your mobile phone for app speech to text converter.

Step 2. Run the app.

Step 3. Talk any sentence in English and see that it is converted into text as given below.

Step 4. Your phone can also understand your oral commands also.

Practical Activity 2.11. Illustrate the concept of Google Translator

Material Required: Computer/Mobile phone, Google translator

Procedure

Step 1. Check the Google translator on your computer/mobile phone.

Step 2. Type English sentences.

Step 3. Now translate it into different languages such as Hindi, Marathi, Japanese, Chinese ...

Step 4. Observe that the translator makes use of AI for translation.

Check your Progress

A. Multiple choice questions

1. Which type of Machine Learning is used in face recognition? (a) Un-Supervised Learning (b) Reinforcement Learning (c) Supervised Learning (d) Deep Learning
2. Which machine learning technique involves learning through receiving feedback in the form of rewards or penalties? (a) Un-Supervised Learning (b) Reinforcement Learning (c) Supervised Learning (d) Deep Learning
3. Which type of Machine Learning is used for clustering of objects? (a) Un-Supervised Learning (b) Reinforcement Learning (c) Supervised Learning (d) Deep Learning
4. Which of the following is a subset of machine learning that makes use of neural networks? (a) Un-Supervised Learning (b) Reinforcement Learning (c) Supervised Learning (d) Deep Learning
5. Which type of Machine Learning uses the K-Means Clustering algorithm? (a) Un-Supervised Learning (b) Reinforcement Learning (c) Supervised Learning (d) Deep Learning
6. Which of the following is a Deep Learning application? (a) Real-time object detection with CNNs (b) Detecting diseases from X-rays & MRIs (c) Google Translate using transformers (d) all of the above

7. Which Deep Learning application is used for generating realistic face-swaps? (a) Self-driving cars (b) Medical Imaging) (c) Deepfake technology (d) Machine Translation
8. Which technique identifies objects in an image and draws bounding boxes around them? (a) Image Classification (b) Object Detection (c) Image Captioning (d) Image Enhancement
9. Which application is used in security systems, access control, social media tagging, and customer interaction? (a) Face Recognition (b) Computer vision (c) Speech Recognition (d) Image Captioning.
10. Which application includes examples like Apple Siri, Google Assistant, and Amazon Alexa? (a) Face Recognition (b) Computer vision (c) Speech Recognition (d) Image Captioning.

B. Fill in the blanks

1. Deep learning is a subset of _____.
2. Neural networks are used in _____.
3. _____ is a subfield of AI that deals with the interaction between computers and human language.
4. The technology that enables devices to convert spoken language into text is called _____.
5. The network of connected physical devices is called _____.

C. State whether true or false

1. Handwritten digit recognition is an example of supervised learning.
2. Clustering of objects is a technique used in supervised learning.
3. RNN full from for Recurrent Neural Network.
4. NLP stands for Neural Language Processing.
5. Translator tools like Google Translate use deep learning techniques.

D. Short Question Answers

1. Give an example of a Supervised Learning application.
2. What is Unsupervised Learning?
3. What is Reinforcement Learning?
4. What is Deep Learning? Give an example of a Deep Learning application.
5. What is Natural Language Processing (NLP)?

Answer Key

Module 1. Basics of AI Programming

Session 1. Introduction to Artificial Intelligence

A. Multiple choice questions

1. (e) 2. (a) 3. (d) 4. (e) 5. (d) 6. (d)

B. Fill in the blanks

1. (perceive, infer) 2. (Artificial General Intelligence) 3. (hypothetical) 4. (repetitive, dangerous) 5. (virtual, personalised)

C. State whether the following statement is True or False

1. (F) 2. (T) 3. (F) 4. (F) 5. (T)

Session 2. Programming Basics

A. Multiple choice questions

1. (a) 2. (c) 3. (d) 4. (b) 5. (d)

B. Fill in the blanks

1. (English) 2. (algorithm) 3. (pieces) 4. (programming language) 5. (complexity).

C. State whether the following statement is True or False

1. (T) 2. (T) 3. (T) 4. (T) 5. (F)

Session 3. Coding in Scratch

A. Multiple choice questions

1. (b) 2. (b) 3. (d) 4. (c) 5. (d)

B. Fill in the blanks

1. (code blocks) 2. (Sprites) 3. (9) 4. (background) 5. (inbuilt)

C. State whether the following statement is True or False

1. (T) 2. (T) 3. (F) 4. (T) 5. (T)

Module 2. Basic Python Programming

Session 1. Introduction to Python

A. Multiple choice questions

1. (d) 2. (c) 3. (a) 4. (d) 5. (a) 6. (a) 7. (b) 8. (b) 9. (a) 10. (c) 11. (a) 12. (d) 13. (c)14. (d) 15. (a)

B. Fill in the blanks

1. (Interactive, Script) 2. (Script/Program) 3. (reserved) 4. (dynamic) 5. (comments)

C. State whether true or false

1. (F) 2. (T) 3. (T) 4. (F) 5. (T) 6. (T) 7. (F) 8. (T) 9. (F) 10. (T)

Session 2. Data Types and Operators

A. Multiple Choice Questions

1. (d) 2. (b) 3. (b) 4. (a) 5. (a) 6. (c) 7. (a) 8. (c) 9. (b)10. (d) 11. (a) 12. (c) 13. (b) 14. (b) 15. (c) 16. (a) 17. (c) 18. (d) 19. (b) 20. (a)

B. Fill in the Blanks

1. (sequence) 2. (dictionary) 3. (list) 4. (string) 5. (input())

C. State whether the following statements are True or False

1. (F) 2. (T) 3. (T) 4. (F) 5. (T)

Module 3. Data Literacy

Session 1. Basics of Data Literacy

A. Multiple choice questions

1. (a) 2. (d) 3. (d) 4. (a) 5. (d)

B. Fill in the blanks

1. (information) 2. (large population) 3. (electronic data) 4. (descriptive data) 5. (measurable data) 6. (One Time Password) 7. (digital data) 8. (quantitative, qualitative) 9. (prediction) 10. (data pre-processing)

C. State whether true or false

1. (F) 2. (T) 3. (T) 4. (T) 5. (T) 6. (F) 7. (F) 8. (F) 9. (T) 10. (T)

Session 2. Data Pre-processing

A. Multiple choice questions

1. (c) 2. (b) 3. (C) 4. (d) 5. (a) 6. (d) 7. (c)

B. Fill in the blanks

1. Normalised, scale 2. Smaller words 3. Rows, Columns 4. Observation 5. Variable, Feature 6. SQL

C. State whether true or false

1. (T) 2. (T) 3. (F) 4. (F) 5. (T)

Module 4. Maths for AI

Session 1. Data Analysis

A. Multiple choice questions

1. (c) 2. (d) 3. (a) 4. (b) 5. (c) 6. (a) 7. (d) 8. (c) 9. (b) 10. (b)

B. Fill in the blanks

1. Cleansing 2. Data Pre-processing 3. Predict future 4. Field 5. Searching 6. Mean 7. Decimal Scaling 8. Central Tendency 9. Stack 10. Standard deviation

C. State whether true or false

1. (T) 2. (T) 3. (F) 4. (T) 5. (F) 6. (F) 7. (T) 8. (F) 9. (T) 10. (T)

Session 2. Data Visualization

A. Multiple choice questions

1. (c) 2. (d) 3. (b) 4. (a) 5. (c) 6. (c) 7. (b) 8. (d) 9. (c) 10. (a)

B. Fill in the blanks

1. Row, Columns 2. NumPy 3. Matplotlib 4. Fourier Transformation 5. Relational databases

C. State whether true or false

1. (T) 2. (T) 3. (F) 4. (T) 5. (F) 6. (T)

Class IX, Module 5. Machine Learning

Session 1. Introduction to Machine Learning

A. Multiple choice questions

1. (b) 2. (a) 3. (b) 4. (d) 5. (c)

B. Fill in the blanks

1. Artificial Intelligence 2. Algorithms 3. Input data 4. Data base, web scraping 5. Data preprocessing

C. State whether true or false

1. (T) 2. (F) 3. (F) 4. (T) 5. (F)

Session 2. Types of Machine Learning

A. Multiple choice questions

1. (c) 2. (b) 3. (a) 4. (d) 5. (a) 6. (d) 7. (C) 8. (b) 9. (a) 10. (c)

B. Fill in the blanks

1. Machine Learning 2. Deep Learning 3. Natural Language Processing 4. Speech Recognition 5. Internet of Things (IoT)

C. State whether true or false

1. (T) 2. (F) 3. (T) 4. (F) 5. (T)