# Computer Science

## Grade - 9



**Government of Nepal**

**Ministry of Education, Science & Technology**

**Curriculum Development Centre**

**Sanothimi, Bhaktapur**

**Publisher :** Government of Nepal
Ministry of Education, Science & Technology
**Curriculum Development Centre**
Sanothimi, Bhaktapur

**ISBN:**

**First Edition :** 2081 B.S.

**Printed at :**

**Price :**

# Preface

A textbook is a main tool for the use of students on a particular subject that is usually developed based on a curriculum to meet its curriculum standards. This 'Computer Science, Grade 9' textbook has been developed to address the main aims of Secondary Education. It aligns with the intent pursued by the National Curriculum Framework for School Education, 2076 and has been developed fully in accordance with the Secondary Level Optional Computer Science Curriculum, 2080. All components of each lesson in the textbook are equally important and they have the specific aim of presentation so as to fulfill the learning outcomes stated in the curriculum.

This textbook is prepared by a team that includes Mr. Sunil Kumar Gharti, Mr. Bimal Thapa, Mr. Pratik Man Singh Pradhan, Mr. Binaya Adhikari, Mr. Shyam Adhikari and Mr. Lav Dev Bhatta. Several people notably, the Director General, Mr. Ima Narayan Shrestha, the subject committee chairperson, Dr. Bed Prasad Dhakal and the subject committee members; Dr. Bal Krishna Subedi and MS. Pramila Bakhati contributed a lot in framing the book in this form. The contribution made by other experts and the practitioners; Dr. Shyam Prashad Acharya, Mr. Rom Kanta Pandey, MS. Nirmala Pokhrel, Mr. Yub Raj adhikari, MS. Salina Dangol and Mr. Shreehari Sharestha is also appreciated. The language of the book was edited by Mr. Nabin Kumar Khadka and the layout of the book was designed by Mr. Jayaram Kuikel. The Curriculum Development Centre extends sincere gratitude to all of them.

This textbook may be used to deliver the foundation for the content of the curriculum, the balance between theoretical and practical aspects is to be maintained by the teachers. This textbook can be used as the primary resource for classroom teaching but teachers are encouraged to make use of their own resources to supplement the computer science learning of their students. The Curriculum Development Centre always welcomes constructive feedback for the betterment of its publications.

2081 BS.

Curriculum Development Centre

Sanothimi, Bhaktapur

# Contents

# Unit 1 Computer System

*Figure 1.1*

***Let's think!***

a) What are the components of a computer system?

b) How are the computer systems used in a different field?

c) Which are the major input and output devices of a computer system?

Nowadays, science and technology are inseparable from our daily lives. The rapid development of computer technology has revolutionized the modern world. Computers are now used worldwide, regardless of a country's development level. In Nepal, computers play essential roles in business, banking, education, entertainment, scientific research, law, and air traffic control. Computers have become integral to every sphere of human activity, bringing about enormous changes.

Technology is at the heart of our existence. Computers serve as essential tools for accessing information, expressing creativity, collaborating, and monitoring learning outcomes. Understanding the factors behind this technology becomes even more crucial as we build smart cities.

## 1.1 Introduction to computer systems

"Computer" comes from the Latin word "Computare," meaning 'to calculate.' Nowadays, computers are everywhere and are important. We use computers at home, school, work, and other places. They're great at doing math fast and accurately. But they can do much more than that! They can make documents, videos, games, music, and presentations.

A computer is an electronic device that accepts raw data from users; processes raw data according to a given set of instructions, and stores and gives meaningful information. It is a programmable device that can perform specific tasks and generate results at a very high speed. In other words, a computer is an electronic device operating under the control of instructions stored in its memory that can accept data (input), process the data according to specified rules, produce information (output), and store the information for future use.

### Features of a computer

Computers are powerful and useful in all areas. The uses of computers are being increased with time. The main features or characteristics of computers are described below:

**i. Automatic**

The computer is an automatic device. It can do tasks automatically as instructed by software. It can automatically control various devices attached to the computer system.

**ii. Speed**

A computer works on low or high signals of electricity. These two different states of electric signals are represented on a computer by using binary numbers 1 and 0. The circuitry of a computer is designed such that electric signals travel at the speed of light, resulting in high processing speed.. It takes only a few seconds for calculations that we take hours to complete. The weather forecast that you see every day on TV is the result of the compilation and analysis of huge amounts of data on temperature, humidity, pressure, etc. from various places on computers. It takes a few minutes for the computer to process the huge amount of data and give the result.

It can process millions (1,000,000) of instructions or even more in a second.

The table below shows the speed of the computer in different units of time.

| Unit of Time | Part of Second | Power of 10 | |
|---|---|---|---|
| Millisecond (ms) | 1/1000 | 10-3 | One Thousandth |
| Microsecond (μ)s) | 1/1000000 | 10-6 | One Millionth |
| Nanosecond (ns) | 1/1000000000 | 10-9 | One Billionth |
| Picosecond (ps) | 1/1000000000000 | 10-12 | One Trillionth |
| Femtosecond (fs) | 1/1000000000000000 | 10-15 | One Quadrillionth |

### iii. Accuracy

A computer performs calculations accurately. It is a 100% accurate device. It does not make any kind of mistake and does not produce the wrong information. But sometimes, due to incorrect input (data or instruction) or faults in hardware or programs, a computer may produce incorrect output. The incorrect output produced due to incorrect input is referred to as Garbage In Garbage Out (GIGO). An error due to the malfunctioning of hardware or programs is referred to as a bug.

### iv. Storage

A computer has storage capacity. It can store a large volume of data, information, and programs on storage devices like hard disks, optical disks, Blu-ray disks and flash drives as files. These storage devices store data, information, and programs as long10 as desired by the user and can be used whenever required. The storage capacities (memory size) of storage devices are measured in bytes and higher units of bytes. One byte is needed to store one character. The different units of storage are shown below.

| | |
|---|---|
| 1 Bit = 1 or 0 | 4 Bits = 1 Nibble |
| 8 Bits = 1 Byte | 1024 Bytes = 1 KB |
| 1024 KB = 1 MB | 1024 MB = 1 GB |
| 1024 GB = 1 TB | 1024 TB = 1 Peta Byte |
| 1024 PB = 1 Exa Byte | 1024 EB = 1 Zetta Byte |
| 1024 ZB = 1 Yotta Byte | |

### v. Diligence

A computer can perform the task repeatedly without losing speed and accuracy for a long time. The capability of a computer to perform a task repeatedly without losing speed and accuracy for a long time is known as diligence. Due to this characteristic of a computer, it is also called a diligent machine. This capability of the computer makes it useful for quality control and process control tasks.

### vi. Versatile

A computer is a versatile machine. Versatility refers to the capability of doing various kinds of tasks. Due to the versatile nature of a computer, it is used in almost all fields. It can be used for calculation, preparation of documents, designing, researching, and entertaining. It can even be used for transferring emails, e-faxes, messages, and files from one computer to another computer in any place in the world.

## Functions of a computer system: Input, Process, Output, Storage (IPOS) with Basic Block Diagram

A computer is an electronic device that converts raw data into meaningful information carrying out various processes as instructed by the program. The block diagram of the IPOS cycle of the computer system as

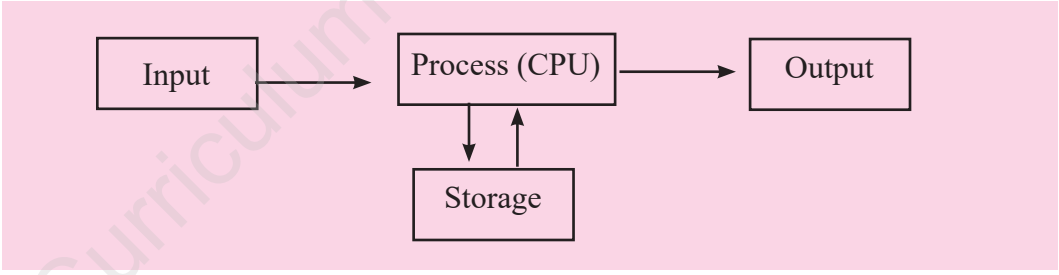



*Figure 1.2*

**Input:** Enter data into the computer.

**Process:** Process data to produce information.

**Output:** Display information.

**Storage:** Store data and information

## Application Areas of Computer Systems

Due to the versatile abilities of a computer, it has become one of the most important tools for people. People can perform tasks more quickly, correctly, efficiently, and systematically with the help of computers. Computers make people efficient and productive, and they also free up their time for other tasks. People are not only using computers for preparing documents, spreadsheets, presentations, graphics, etc. They are using computers for modern facilities like instant messaging, video chatting, email, online gaming, online shopping, ticketing, booking, checking and transferring money in banks, finding the latest news and information, etc. So, computers are used in nearly every field. The following are some important applications of computers:

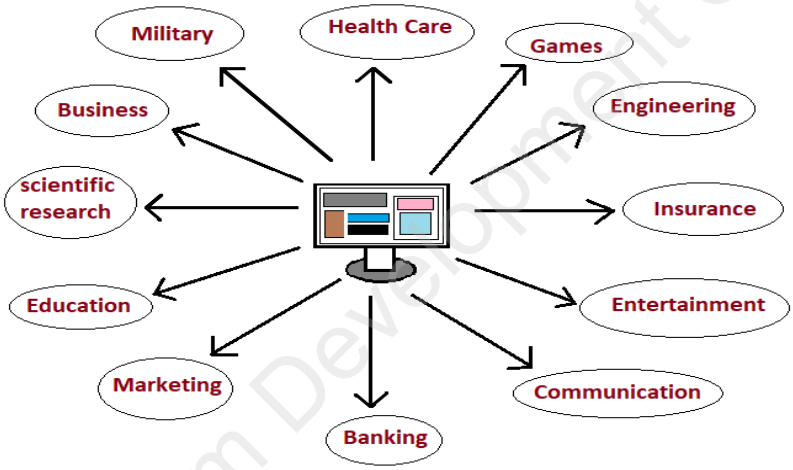



*Figure 1.3*

**Education:** Computers are used in the education sector for teaching and learning purposes. Using computers, students can learn different subjects through websites on the Internet. Teachers can prepare the teaching materials and present them effectively through multimedia computers. Computers are also used in education for keeping records of students, producing monthly bill statements, and results of examinations.

*Figure 1.4*

**Communication:** Computers play a vital role in communication. They help us send and receive information through the Internet. We use various communication services like email, video conferencing, net phone, chatting, and social media platforms such as Facebook, Twitter, and Instagram to stay connected with others.

*Figure 1.5*

**Banking:** Computers have revolutionized the banking sector. They handle various services like accounting, updating customer records, and generating bank statements. Automated computers revolutionize banking by improving efficiency, security, and customer service through online and mobile banking, advanced data analytics, and AI, reducing manual processing and costs.


*Figure 1.6*

**Medicine:** Computers are essential in healthcare. They help diagnose diseases through scans like CT scans and ultrasounds and monitor heart health with tools like ECGs. Hospitals use computers to manage patient records and medications, and they're even used in surgeries. In addition, computers play a crucial role in medical research, analyzing large datasets to develop new treatments and training surgeons with advanced simulations.

*Figure 1.7*

**Entertainment:** Computers are a big part of entertainment. We use them to play games, listen to music, watch films and movies, and even access web stuff. They provide immersive experiences through powerful graphics and virtual reality technology. Streaming platforms like Netflix and Spotify rely on computers

*Figure 1.8*

to deliver a vast library of entertainment. Social media platforms connect people worldwide, and computers empower creativity by enabling digital art creation and innovative product design.

*Figure 1.9*

**Scientific research:** Computers are essential in all forms of scientific study. Scientists use the Internet to obtain up-to-date information from all across the world. They rely on computers for weather forecasting, space exploration, and complex research in various scientific fields. Computers enable researchers to analyze vast amounts of data and conduct simulations that further our understanding of the world around us.

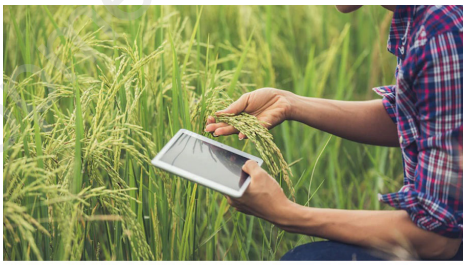


*Figure 1.10*

**Agriculture:** Computers have not only transformed modern occupations but have also brought significant changes in traditional sectors like agriculture. Computers are revolutionizing agriculture through precision farming, data-driven decision-making, and advanced technologies like GPS, drones, IoT, and AI, enhancing efficiency and productivity.

*Activity 1.1*

**Activity Outcome -**

- Able to describe computer components, features, and applications with diagrams/flowcharts.

**Required Resources:** Chart Papers, Meta Cards, Cardboards, Images, PowerPoint Presentation, etc.

**Procedure:**

Your teacher will break you into groups of 3-5 and let each group to present on one of the following topics:

1. Computer Features 2. Computer Components 3. Computer Applications

**Result:** Moderated discussion by the teacher for your reflection with feedback.

## Exercises

**1. Write the full forms of the following abbreviations.**

i. EC ii. MHz iii. GHz iv. ms v. ns vi. Ps

vii. Fs viii. GIGO ix. RAM x. IPOS xi. ATM

xii. CT scans

**2. Choose the correct answer.**

a. Which fraction of a second is equivalent to one millionth of a second?

i) Milliseconds ii) Microsecond

iii) Nanosecond iv) picoseconds

b. Incorrect input results in incorrect output. It is referred to as ….

i) FIFO ii) LIFO iii) GIGO iv) None of the above

c. How many kilobytes are in one megabyte?

i) 1030 ii) 1000 iii) 1048 iv) 1024

d. Repeatedly working capability of computers is known as ……..

i) storage ii) speed iii) diligence iv) versatility

e. What is the primary function of the CPU in a computer system?

i) Memory storage ii) Data input

iii) Arithmetic and logic operations iv) Graphics rendering

**3. Write short answers to these questions.**

a) Define the Computer.

b) How do IPOS work?

c) What is GIGO?

d) List any four computer features.

e) Why is a computer called a diligence machine?

f) List four computer-related uses.

g) Identify distinct computer storage units.

**4. Write long answers to these questions.**

a) Explain the working principles of computers with a block diagram and explain it in detail.

b) Describe the use of computers in the education field.

## 1.2 Input Devices

Input devices are tools for entering raw data and instructions into a computer system. These devices allow users to provide data for processing. Examples of input devices include the Mouse keyboard, joystick, microphone, etc.

### Mouse

The mouse is a popular handheld pointing input device used to interact with computers. It's used for tasks like pointing at objects, giving commands, and drawing or painting. A typical mouse has three buttons: the left button, the right button, and the scroll button. Users employ these buttons for actions such as clicking, double-clicking, right-clicking, dragging, and scrolling to communicate commands to the computer.

*Figure 1.11*

### Keyboard

The keyboard is a popular input device used to enter data into the computer by typing. The most common keyboard layout is the "QWERTY" keyboard, named after the first six letters in the top row of keys. A standard keyboard consists of 104 keys. A multimedia keyboard has more than 104 keys. The keys on a keyboard are grouped into five different groups as:

a. Alphabet keys
b. Numeric keys
c. Cursor-movement keys
d. Functional keys
e. Special purpose keys

*Figure 1.12*

All keys marked with A to Z and punctuation symbols are alphabet keys. Keys marked with 0 to 9 and mathematical symbols are numeric keys. Keys marked with F1 to F12 are the function keys, and they are used for entering commands. Keys marked with arrow signs (←, ↑, →, ↓) are cursor movement keys (i.e., also known as arrow keys), and they are used to move the cursor in the specified direction. The Space bar, Enter key, Back Space key, Caps Lock key, Esc key, Ctrl key, Alt key, Shift key, Delete key, Home key, End key, and Insert key are the special purpose keys.

### Joystick

A joystick is an input device commonly used for playing video games. It consists of a stick that can be moved in various directions and typically features additional buttons for controlling gameplay. Joysticks come in different types and configurations, tailored for different gaming experiences.

*Figure 1.13*

### Microphone

The microphone is a device used for capturing sound and converting it into a digital format that can be input into a computer. It is widely used for a variety of reasons, including voice chat and video conferencing via platforms such as Skype, Viber, Facebook Messenger, and others.

*Figure 1.14*

### Scanner

The scanner is an input device used to convert physical drawings or documents into digital format, allowing them to be entered into a computer. It functions by capturing images or text from a physical document and then transferring this data to the computer in a digital format.

*Figure 1.15*

### Webcam

A webcam is a digital device that captures real-time video and audio, essential for video conferencing, online streaming, content creation, remote work, online education, and social interactions.


*Figure 1.16*

### Touchscreen

A touch screen is a display panel that is sensitive to touch, covering the screen of a device. Initially popularized in smartphones, touch screens are now commonly found in laptops, notebooks, palmtops, and desktop computers. Users interact with touch screens by touching them with their fingers, allowing them to select objects on the screen and give commands to the computer system.


*Figure 1.17*

### Touchpad

A touchpad is a touch-sensitive pad found in laptops, also called a Glide pad or trackpad. It acts like a mouse, letting users move the cursor by sliding their fingers. It usually has left and right buttons for clicking.

*Figure 1.18*

## 1.3 Central Processing Unit (CPU)

The processing unit contains elements such as the CPU (Central Processing Unit), also known as the computer's brain. The CPU is the primary processing unit responsible for receiving data from the input unit, executing instructions provided by the user, and generating output information. Intel CPUs, AMD CPUs, Apple CPUs, ARM CPUs (Non-Apple), etc. are popular examples of CPUs.


*Figure 1.19*

### Functions of the CPU

The CPU (Central Processing Unit) performs the following primary functions:

i. **Fetching:** The CPU retrieves instructions or data from the computer's memory (RAM).

ii. **Decoding:** The CPU interprets the fetched instructions to understand what actions are required.

iii. **Executing:** The CPU performs the necessary actions, such as calculations, data movement, or interacting with other hardware.

iv. **Storing:** The CPU stores the results of executed instructions back into memory or sends them to an output device.

v. **Managing Interrupts:** The CPU handles interruptions from other hardware or software, ensuring smooth and efficient processing.

### Components of the CPU: CU, ALU, and MU

The speed of a computer depends mainly on its microprocessor. The microprocessor has three main parts.

a. Control Unit (CU)
b. Arithmetic Logic Unit (ALU)
c. Registers

**All these work together to process the computer's data.**

a. The Control Unit controls the working of all parts of a computer. It controls the movement of data inside as well as outside the CPU. It communicates between registers and the ALU and between the CPU and all the input, output, and storage devices. It directs all the parts of the computer system to perform their tasks.

b. The Arithmetic Logic Unit (ALU) performs all arithmetic operations like addition, subtraction, multiplication, division, and logical operations like greater than, less than, not equal, etc.

c. Registers are small memory units used to store immediate data, instructions, and results while processing data.

## 1.4 Motherboard and Data Bus

The motherboard is known as the PCB (Printed Circuit Board), or system board, or main circuit board. It serves as the foundation of the computer system. It hosts various components such as the microprocessor, slots, ports, and other essential chips. Often referred to as the "backbone" of the computer, the motherboard connects critical components like the ROM chip, network chip, and sound card, ensuring seamless communication and functionality within the system.

The motherboard serves as a central connector hub in a computer system, facilitating communication and collaboration among components.

i. Physical Connections: Sockets securely hold key components like the CPU and RAM. Expansion slots allow for additional hardware like graphics cards and network adapters. Dedicated ports connect external devices such as monitors, keyboards, and printers.

ii. Electrical Connections: Power distribution from the PSU ensures all components receive power through specific circuits and voltage regulators. Data transfer occurs via electrical signals flowing through the motherboard's network of pathways.

iii. Communication Hub: Chipsets like the Southbridge and Northbridge manage communication between peripherals and high-speed components like the CPU and RAM. Internal buses, such as PCIe and SATA, act as data transfer highways between components.

### Data Bus

Buses are connected lines or electronic pathways within a microprocessor chip that link its internal components. Data, address, and control are typically three types of buses in a microprocessor. The data bus is a key communication pathway in a computer system, enabling the transfer of digital data between components like the CPU, memory, and peripherals. Made of parallel lines, it allows for efficient and high-speed exchange of binary information. The width of the data bus, measured in bits, determines the amount of data transferred in one go, with wider buses typically offering faster communication and better system performance. In essence, the data bus plays a crucial role in ensuring smooth information flow and the execution of computing tasks.

**The data bus functions as follows:**

i. Transferring Data: It facilitates the transfer of data between different components such as the CPU, RAM, hard drive, and peripherals.

ii. Handling Different Data Types: Various types of buses manage different kinds of data and operate at varying speeds.

## 1.5 Computer memory

A computer memory is used for storing various types of data and information. There are different types of memory for storing data and information. Some memory stores data temporarily and some stores permanently. Some memories communicate

directly with CPU whereas some communicate indirectly with computer systems. The overall classification of computers can be done in two types:

a. Primary memory and

b. Secondary memory

## Primary memory

Primary memory, also known as the main memory or internal memory of a computer system, serves as the core storage component. It can be categorized as either temporary or permanent. Compared to secondary memory, primary memory has a relatively limited storage capacity. Its primary function is to temporarily store data and instructions during processing. Acting as the memory directly accessible by the CPU, primary memory facilitates the processor's interaction with running applications and services stored temporarily in specific memory locations. Upon booting up, primary memory loads all active applications, including the operating system, user interface, and background programs. Whenever a program or application initiates within the computer system, it is loaded into primary memory to interact with the operating system. The two types of primary memory are:

RAM (Random Access Memory)

ROM (Read Only Memory)

## RAM (Random Access Memory)

Random Access Memory (RAM) is the volatile memory or short-term memory of a computer system. It executes both reading and writing operations. It can read and store data randomly from any physical location within it. RAM holds data and instructions during processing. Since it is volatile memory, data stored in RAM is erased when the power supply is turned off. Hence, to retain files permanently, they must be stored on secondary storage devices like a hard disk. RAM is classified as R/W (read-write) memory since it allows both reading and writing operations. Upon powering on the CPU, system files stored on the hard disk are loaded into primary memory RAM. There are two main types of RAM: Static Random

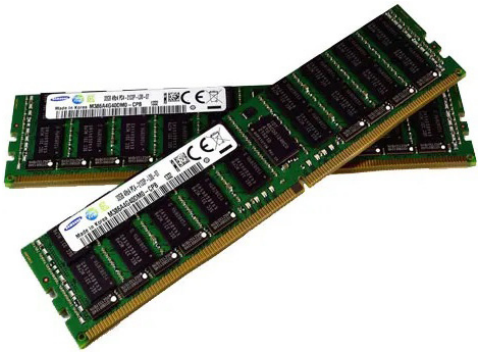


*Fig. 1.20 Random Access Memory*

Access Memory (SRAM) and Dynamic Random Access Memory (DRAM).

i. **SRAM:** Static Random Access Memory, retains data in a static state as long as power is supplied to the memory. This semiconductor-based memory stores data and instructions without the need for periodic refreshment, unlike DRAM. While SRAM is more costly than DRAM, it offers faster performance. Typically employed as cache memory, SRAM consumes less power compared to DRAM.

ii. **DRAM:** Dynamic Random Access Memory, is constructed using capacitors. Data stored in this type of RAM is prone to loss within a few milliseconds, even when the computer is powered on, necessitating periodic circuit refreshment. DRAM's design involves capacitors, where charge leakage occurs. While DRAM is less expensive and slower compared to SRAM, it finds common usage in PCs due to these characteristics.

### Difference between SRAM and DRAM

| SRAM | | DRAM | |
|---|---|---|---|
| a. | It is made up of transistors. | a. | It is made up of capacitors. |
| b. | It is more expensive. | b. | It is less expensive. |
| c. | SRAM doesn't need periodic refreshment. | c. | DRAM needs periodic refreshment. |
| d. | Charges don't get leaked from SRAM. | d. | Data charges get leaked from DRAM. |
| e. | SRAM are less dense. | e. | DRAM is denser. |
| f. | It is faster than DRAM. | f. | It is slower than SRAM. |
| g. | Data is stored in the form of voltage. | g. | Data is stored in the form of charge. |

### ROM (Read Only Memory)

ROM exclusively retains information that can solely be accessed for reading purposes. Altering data stored in ROM presents a challenge. Serving as the non-volatile memory of a computer system, ROM ensures that data and instructions remain intact even when the power supply is disconnected. Unlike volatile memory types, ROM doesn't require a constant power supply to preserve stored data. The

instructions within ROM are embedded during manufacturing by the respective manufacturer. Given its read-only nature, users cannot modify the contents of ROM. This type of memory commonly stores BIOS (Basic Input Output System) information essential for booting up the computer. The programs housed in ROM are known as firmware. Variants of ROM include PROM, EPROM, and EEPROM.

i. **PROM:** PROM, or Programmable Read-Only Memory, refers to a type of ROM with a unique characteristic: it can be programmed just once but read multiple times thereafter. Once data and instructions are stored in PROM, they become permanent and cannot be erased. Similar to other ROM types, PROM maintains its contents even when the power is off. Unlike ROM, which comes with pre-installed contents during manufacturing, PROM is initially blank upon manufacture. To write data onto PROM, a specialized device called a PROM programmer or PROM burner is employed. Once programmed, PROM cannot be altered or erased, making it suitable for storing permanent data in digital electronic devices.

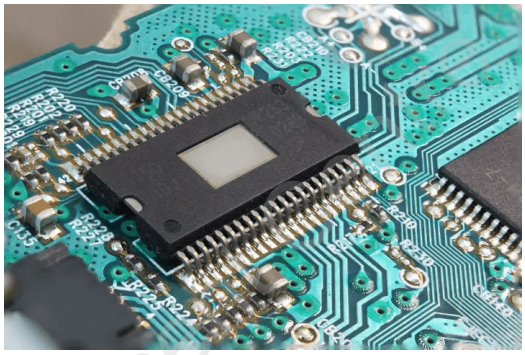


*Fig. 1.21. Read Only Memory*

ii. **EPROM:** Erasable Programmable Read-Only Memory (EPROM) is a form of ROM that offers the capability of being reprogrammed multiple times. However, reprogramming EPROM requires specialized techniques involving UV (ultraviolet) light rays. The contents stored on EPROM can be erased and rewritten using these UV rays. It's important to note that frequent erasures of EPROM can lead to wear and tear, potentially affecting its lifespan.

iii. **EEPROM:** Electrically Erasable Programmable Read-Only Memory (EEPROM) is a variation of EPROM that offers the capability to erase and write data electronically using electrical signals within a millisecond. Unlike traditional EPROM, EEPROM does not require special techniques or laboratory conditions to delete or write content onto it. This flexibility allows for the efficient modification of data without the need for external tools or UV light exposure. Whether it's a single byte of data or the entire content of the device, EEPROM provides the convenience of electrically erasable and writable memory.

## Firmware

Firmware refers to software that is embedded within a piece of hardware. In essence, it serves as software specifically designed to operate and control hardware components.

## Cache memory

The high speed memory between primary memory RAM and CPU is known as cache memory. The cache memory is used to store frequently accessed data and instruction during the time of processing by CPU. It is the high speed volatile memory available to the processor for fast processing. The storage size of cache memory is small. The CPU first searches data in cache for processing if not found there then looks in RAM and loads to cache memory. The main importance of cache memory is to make the CPU execution faster.

## CPU register

A CPU Register is a compact storage memory situated within the microprocessor that is distinguished by its high data transfer rate. It stores the data and instructions used by the microprocessor during its operation.

## Differentiating between primary and secondary memory.

| | Primary Memory | | Secondary Memory |
|---|---|---|---|
| a. | Primary memory is considered as the main memory of a computer system. | a. | Secondary memory is also known as auxiliary or backup memory. |
| b. | It can directly communicate with the CPU. | b. | Secondary memory indirectly communicates with the CPU. |
| c. | The storage capacity of primary memory is less than secondary memory. | c. | The storage capacity of secondary memory is higher than primary memory. |
| d. | Primary memory is faster than secondary memory. | d. | Secondary memory is slower than primary memory. |
| e. | They are more expensive. | e. | They are less expensive |
| f. | Information stored in primary memory cannot be moved from one place to another. | f. | Information stored in secondary memory can be transferred from one computer to another. |

| | | | |
|---|---|---|---|
| g. | Data and instructions to be currently executed are stored in Primary memory. | g. | Data and information to be stored permanently are kept in secondary memory. |
| h. | Primary memory can be both volatile and non-volatile. | h. | Secondary memory is non-volatile in nature. |
| i. | Primary Memory has faster access time. | i. | This memory has low access time. |
| j. | Examples: RAM, ROM, Cache | j. | Examples: Pen drive, CD, Hard Disk. |

### Units of memory measurement

The bit is the smallest unit of memory measurement, representing one binary digit (either 0 or 1).

| S. N. | Memory Unit | Description |
|---|---|---|
| 1. | Bit | 1 Bit = 0 or 1 |
| 2. | Nibble | 1 Nibble = 4 Bits |
| 3. | Byte | 1 Byte = 8 Bits |
| 4. | Kilobyte (KB) | 1 Kilobyte (KB) = 1024 Bytes |
| 5. | Megabyte (MB) | 1 Megabyte (MB) = 1024 KB |
| 6. | Gigabyte (GB) | 1 Gigabyte (GB) = 1024 MB |
| 7. | Terabyte (TB) | 1 Terabyte (TB)= 1024 GB |
| 8. | Petabyte (PB) | 1 Petabyte (PB) = 1024 TB |
| 9. | Yottabyte (YB) | 1 Yottabyte (YB) = 1024 PB |

## 1.6 Storage devices

Storage devices or Secondary memory, also known as auxiliary or backup memory, store data, and instructions permanently for future use. Unlike primary memory, it retains information even when the computer is turned off. Common types include magnetic tape, hard disks, SSDs (Solid State Drives), optical storage discs, and pen drives.

## Hard disk


*Fig. 1.22*

A hard disk is a popular storage device that holds enormous amounts of data indefinitely. It is constructed with aluminum materials and coated with a ferromagnetic element. Its surface consists of circular metal discs, or platters, which rotate while data is written or read. It offers substantial storage capacity and is measured in revolutions per minute (rpm) for speed, typically ranging from 5400 to 7200 rpm.

## SSD (Solid State Drive)


*Fig. 1.23*

Nowadays, hard disks are quickly being replaced by SSD (Solid State Drive) due to their faster performance, lower latency, and increased reliability. SSDs use semiconductor chips to store data, offering quicker read and write speeds compared to hard disks. Despite being more expensive, SSDs are preferred in devices like laptops, notebooks, and ultrabooks for their enhanced performance.

## Optical Storage Disc

An optical storage disc is a disc that uses laser light technology to store and retrieve data. The most commonly used optical storage discs are CDs, DVDs, Blu-ray disks, etc.

## CD (Compact Disc)

In 1980, Philips Company of the Netherland developed the first CD or optical disk, which was 12cm in diameter and had a 72-minute audio/video storage capacity. Laser ray technology is used to retrieve and store the data. The information on the optical disk is stored in the form of pits, which are tiny reflective bumps.

CD is 12cm in diameter and weight 18gm and width 1mm (millimeter). The data can be written on a CD only once and can be read many times. CD is made up of polycarbon and both surfaces are coated with aluminium reflected. Nowadays, CDs are commonly used in microcomputers. It can be used to transfer data from one computer to another. The storage capacity of CD is 700MB.

## DVD (Digital Versatile Disc)

DVD stands for Digital Versatile Disc or Digital Video Disc. Its storage capacity is higher than CD. It stores multiple layers of data. The various DVD storage capacities are

- Single Side Layer- 4.7 GB
- Single Side Double Layer-8.5 GB
- Double Layer Single Side- 9.4 GB
- Double Layer Double Side-17.08 GB

Philips, Sony, Toshiba/ Panasonics Companies developed first DVD in 1995 whose weight 16gm. DVD Drive is used to read/write the data on DVD.

## Difference between CD and DVD

| S.No. | CD | DVD |
|---|---|---|
| 1 | Its storage capacity is 700MB. | Its storage capacity is up to 20GB. |
| 2 | It is cheaper. | It is expensive. |
| 3 | It is used to store audio and software. | It is used to store video and software. |
| 4 | Its types are CD-R, CD-RW, etc. | Its types are DVD-RW, DVD+RW, etc. |
| 5 | It was developed by Philips and Sony. | It was developed by Philips, Sony, Toshiba, Panasonic, etc. |

## Flash memory

Flash memory is a type of non-volatile computer storage that can be electrically erased and reprogrammed. It is a crucial component in various electronic devices, including USB drives, memory cards, SSDs (Solid State Drives), and other storage devices. Flash memory retains stored data even when power is turned off.

## Pen drive

Flash memory is a form of non-volatile memory known for its ability to retain data even when power is turned off. This feature makes it ideal for storage and transferring data between personal computers and various digital devices. Its electronic reprogramming and erasing capability add to its versatility. Flash memory is commonly integrated into USB flash drives,

*Fig. 1.24*

MP3 players, digital cameras, and solid-state drives (SSDs). Data stored in flash memory is erased in blocks, necessitating the removal of entire blocks before new data can be written. Despite being more expensive than traditional hard drives and RAM, flash memory offers advantages such as durability, portability, and faster access times. These qualities have contributed to its widespread adoption in modern technology.

## Cloud storage

*Fig. 1.25*

Cloud storage refers to the online storage of data on remote servers accessed over the Internet. It allows users to upload, access, and manage their files and data from anywhere with an Internet connection. Third-party companies like Dropbox, Google Drive, Microsoft OneDrive, and Amazon S3 provide cloud storage services. These services let you change the amount of storage you use. Cloud storage is useful because you can access it from different devices, it syncs automatically, and it can be cheaper than traditional storage methods.

## Compare HDD Vs SSD Vs Flash memory

Hard Disk Drives (HDDs) have been used for storage for many years. They use spinning disks and read/write heads to access data. This method is reliable and cost-effective for large storage, but it has slower speeds, usually between 80 to 160 MB/s. HDDs are good for storing a lot of data, with sizes from 500 GB to 18 TB, and they are cheaper per gigabyte. However, their moving parts can fail more easily and are sensitive to physical shocks. They also use more power, especially when starting up. So, while HDDs are great for backup and storing lots of data, they are slower than newer storage technologies.

Solid State Drives (SSDs) are a big improvement in storage technology, using NAND flash memory with no moving parts. This makes them much faster, with read/write speeds from 200 MB/s to over 3,500 MB/s for NVMe SSDs, and they have lower latency. SSDs are more reliable and can handle physical shocks and vibrations better than HDDs. They also use less power, which is great for laptops and mobile devices. However, SSDs cost more per gigabyte, although prices are going down. Due to their speed and performance, SSDs are best for running operating systems, software, and games.

Flash memory includes USB flash drives, SD cards, and similar storage devices.

They use NAND flash technology, which means no moving parts and low latency. Flash memory devices have different speeds and capacities; USB 3.0 drives can reach up to 600 MB/s, and SD cards range from 10 MB/s to 300 MB/s. They are very durable and use little power, making them ideal for portable storage and battery-powered devices like cameras and smartphones. Although more expensive per gigabyte than HDDs, flash memory is often similar in cost to SSDs and is great for quick, removable storage.

## 1.7 Output devices

Output devices are essential components of a computer system, responsible for presenting processed data in a meaningful format to users. These devices convert digital information generated by the processing unit into human-readable forms. The primary output devices include monitors and printers, while other devices such as plotters, speakers, and headphones also serve specific output functions.

### Monitor

Monitors, also known as Visual Display Units (VDUs), are essential output devices in computer systems, presenting processed data to users in the form of text, videos, and graphics. Output displayed on a monitor's screen is referred to as soft copy output. Monitors come in two main types based on color capabilities: monochrome and color monitors.

*Fig. 1.26*

### LED and LCD Monitors

#### LCD (Liquid Crystal Display) Monitor

*Figure 1.27*

LCD monitors utilize liquid crystal technology to display images on a screen. They are known for their lightweight, thin profile, and energy efficiency. These monitors consist of two layers of glass or plastic substrates containing liquid crystals that form an image when activated. Backlighting in LCD monitors is typically provided by fluorescent lamps. LCD monitors offer advantages such as sharp image quality, thin design, and energy efficiency, making them popular in various visual display applications.

### LED (Light Emitting Diode) Monitor

LED monitors are a type of display technology commonly found in modern monitors, televisions, and other visual display devices. Unlike traditional LCD monitors that use fluorescent backlighting, LED monitors utilize light-emitting diodes to illuminate the screen. LED monitors are lighter, thinner, and more energy-efficient than CRT (cathode-ray tube) and LCD monitors. They offer higher image quality, brightness, and color reproduction, making them ideal for long-term use.

## Printer

A printer is an output device that displays the data or information on the paper. There are mainly two types of printers. They are impact and non-impact printers.

### Laser printer

Laser printers utilize a laser beam to create an image on a photosensitive drum, which is then transferred onto the paper using toner (powdered ink). Laser printers are known for their fast printing speeds and crisp, professional-quality prints. They are often used in office environments for high-volume printing.

*Fig. 1.28*

### Ink-jet printer

Inkjet printers use liquid ink sprayed onto the paper to create text and images. They are known for their versatility and ability to produce high-resolution prints. Inkjet printers are commonly used for everyday printing tasks, such as documents, photos, and graphics.


*Fig. 1.29*

### Dot-matrix printer

A dot matrix printer is a type of impact printer. It features a print head containing numerous pins. When the print head strikes an inked ribbon, it creates dotted characters on the paper. Dot matrix printers offer a speed range of 50 to 400 CPS (Characters per second).


*Fig. 1.30*

### 3D printer

3D printers are a unique type of non-impact printer that builds three-dimensional objects layer by layer using various materials such as plastic, metal, or resin. They are commonly used in manufacturing, prototyping, and design industries to create prototypes, models, and custom parts. 3D printers offer unparalleled flexibility and customization options for creating objects with complex details.

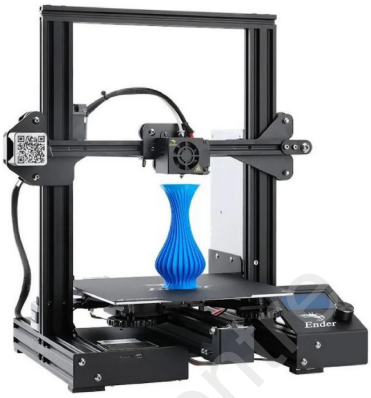



*Fig. 1.31*

### Speaker

A speaker is a device used for playing sound from computers or other electronic devices. It converts digital signals into audible sound. The quality of the sound we hear depends on the quality of the speaker itself. There are various types of speakers available, including those designed for high-volume, bass, or outdoor use.

*Fig. 1.32*

## 1.8 Peripheral devices

Peripheral devices are external hardware components that connect to a computer or other digital device to extend its capabilities. These devices provide additional functionalities and interact with the computer to enhance user experience. Peripheral devices can be categorized into input, output, storage, communication, and other specialized devices. They are essential for the overall functionality and usability of computer systems. They help with input, and output, storing data, and connecting to the Internet.

### Hardware ports

Hardware ports are sockets located on the exterior of the system unit that links to expansion boards. This architecture facilitates the connection of two devices by determining the design of the plug and socket, the types and purposes of wires involved, and the electrical signals transmitted across them. Ports are linked to the interior of the system unit. Hardware interfaces serve several key functions that include:

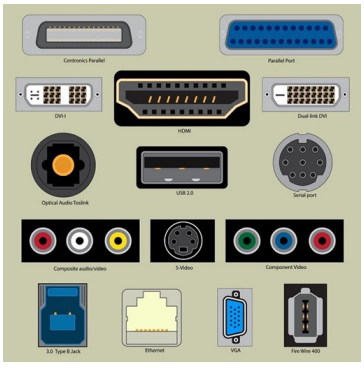



*Figure 1.33*

- Connecting peripheral devices.

- Providing electrical power to devices with low power requirements, like mice, keyboards, and pen drives.
- Transmitting data to and from peripheral devices.

### USB port

The Universal Serial Bus (USB) is used for short-distance digital data transfer. USB ports are versatile, allowing connection to a wide range of devices such as printers, cameras, keyboards, speakers, scanners, and mice.

*Figure 1.34*

### Type C port

The Type C port is a versatile connector found in many devices like phones and laptops. It's small and reversible, meaning you can plug it in either way. Type C supports fast charging, and high-speed data transfer, and can connect to various accessories.

*Figure 1.35*

### HDMI

The High Definition Multimedia Interface (HDMI) is a common interface for transmitting high-definition audio and video signals over a single cable. It is widely used to connect devices such as digital TVs, DVD players, Blu-Ray players, gaming consoles, and AppleTV to televisions.


*Figure 1.36*

### VGA port

The Video Graphics Array (VGA) port is a standard analog interface used to connect monitors, projectors, and other display devices to computers. It uses a 15-pin connector to carry analog signals for the red, green, and blue color channels (RGB), as well as horizontal and vertical synchronization signals. Despite being an older technology, VGA ports remain prevalent in many computing environments.

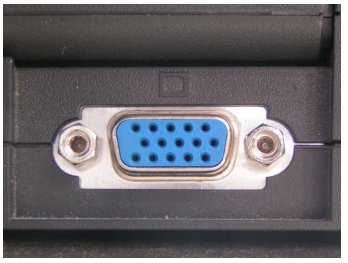


*Figure 1.37*

### Ethernet port

*Figure 1.38*

Ethernet ports are used for wired local area networking (LAN), enabling devices to communicate within a network. Ethernet technology defines how data packets are transmitted over network cables and the protocols used for communication among networked devices. Ethernet ports typically feature a physical connector supporting data transmission using Ethernet protocols.

### Audio port

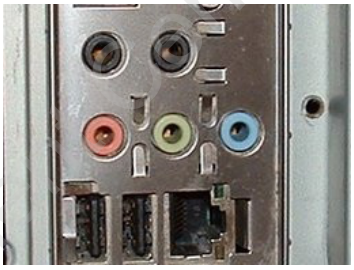


*Figure 1.39*

It is also known as audio jacks or audio connectors, audio ports are essential for connecting audio devices such as headphones, microphones, speakers, and other audio equipment to electronic devices. These ports facilitate the transmission of analog audio signals between devices, allowing users to listen to audio output or record audio input.



*Activity 1.2*

**Activity Outcome:**

Able to describe disassembling and reassembling computers to understand hardware components by the teacher.

**Required Resources:** Computer Case, Power Supply Unit (PSU), Motherboard, Central Processing Unit (CPU), Random Access Memory (RAM), Storage Drives: Hard Disk Drive (HDD) or Solid State Drive (SSD), Graphics Processing Unit (GPU), Cooling Systems, Optical Drives (optional), and Expansion Cards.

**Tools needed:**

- **Screwdrivers:** Typically Phillips head screwdrivers of various sizes.
- **Anti-static wrist strap:** Prevents static discharge from damaging components.
- **Tweezers:** For handling small screws and parts.
- **Thermal paste:** Used when reinstalling a CPU to maintain proper cooling.
- **Plastic pry tools:** For gently opening cases or removing components.

**Procedure:**

***Steps for disassembling***

1. **Power off and unplug:** Ensure the computer is completely powered down and unplugged.
2. **Remove the case cover:** Use the screwdriver to remove screws and slide off the cover.
3. **Disconnect power supply cables:** Detach cables connected to the motherboard and drives.
4. **Remove storage drives and GPU:** Unscrew and carefully slide out.
5. **Uninstall RAM:** Press the clips on either side of the RAM to release.
6. **Remove the CPU cooler and CPU:** Unscrew the cooler and lift it off, then unlock and remove the CPU carefully.
7. **Detach the motherboard:** Unscrew it from the case and lift it out.

**Steps for Reassembling**

1. **Place the motherboard:** Align it with standoffs in the case and screw it in.
2. **Install the CPU and cooler:** Place the CPU in the socket, apply thermal paste, and attach the cooler.
3. **Install RAM:** Insert it into the slots and press until the clips lock.
4. **Reinstall the GPU:** Slide it into the PCIe slot and secure it.
5. **Reconnect storage drives:** Slide them into the bays and connect power and data cables.
6. **Connect power supply cables:** Attach to the motherboard and components.
7. **Reattach the case cover:** Secure the cover with screws.

**Result:** Moderated discussion by the teacher for your reflection with feedback.

## Exercises

**1. Write Full forms of the following abbreviations:**

| | | | | |
|---|---|---|---|---|
| i. CPU | ii. BIOS | iii. HDD | iv. SSD | v. RAM |
| vi. CU | vii. USB | viii. ALU | ix. MU | x. PCB |
| xi. BIOS | xii. ROM | xiii. PSU | xiv. CMOS | xv. CD |
| xvi. DVD | xvii. KB | xviii. MB | xix.GB | xx. TB |
| xxi.YB | xxii. PB | xxiii.SRAM | xxiv. DRAM | xxv. VRAM |
| xxvi. EPROM | xxvii. EEPROM | xxviii. rpm | xxix. VDU | |
| xxx. LED | xxxi. LCD | xxxii. CPS | xxxiii. LCM | xxxiv. PPM |
| xxxv. Mbps | xxxvi. HDMI | xxxvii. LAN | | |

**2. Choose the correct answer.**

i. What is the primary function of the CPU in a computer system?

(a) Memory storage (b) Data input

(c) Arithmetic and logic operations (d) Graphics rendering

ii. Which of the following is a volatile memory in a computer system?

a. Hard Disk Drive (HDD)

b. Random Access Memory (RAM)

c. Solid State Drive (SSD)

d. Cache Memory

iii. What does the term "BIOS" stand for in a computer system?

a. Basic Input Output System b. Binary Input Output System

c. Base Input Output Software d. Basic Integrated Operating System

iv. Which of the following is a secondary storage device?

a. RAM b. CPU

c. USB Flash Drive d. Cache Memory

v. What is the role of the motherboard in a computer system?

a. Manage power supply

b. Connect all hardware components and provide communication between them

c. Execute software applications

d. Store data permanently

vii. What is the role of a monitor as an output device in a computer system?

a. Monitors capture and decode barcodes.

b. Monitors provide a visual display of digital information such as text, images, and videos.

c. Monitors regulate the power supply to other output devices.

d. Monitors convert hand-drawn sketches into digital format.

viii. What is the role of microphones as an input device in a computer system?

a. Microphones capture and decode barcodes.

b. Microphones record audio input, allowing for voice commands or communication.

c. Microphones read information from magnetic stripes.

d. Microphones are used for scanning documents.

ix. What is the primary function of a printer as an output device?

a. Convert printed or handwritten text into digital form

b. Display graphics on the screen

c. Produce a hard copy of digital documents or images

d. Record audio input

x. What is the primary function of a microprocessor in a personal computer?

a. Display graphics on the screen

b. Manage power supply to peripherals

d. Execute instructions and perform calculations

d. Control the cooling system

## 3. Write short answers to these questions.

a. What is the primary function of the CPU in a computer system?

b. Explain the difference between RAM and ROM in terms of functionality.

c. How does a hard drive differ from RAM in terms of data storage?

d. What is the role of a motherboard in a computer system?

e. Explain the concept of cache memory and its importance in CPU performance.

f. Define the term "bus" in the context of computer architecture.

g. What is the purpose of an input device in a computer system? Provide examples.

h. What is a peripheral device, and how does it enhance a computer system?

## 4. Write long answers to these questions.

a. Explain the architecture of a computer system, detailing the functions of the CPU, memory, input/output devices, and storage.

b. What is the role of an operating system in managing and coordinating the various components of a computer system with examples? Analyze the functions of an operating system.

c. Differentiate between primary and secondary memory.

d. Describe the role of cache memory, RAM, and storage devices. How do they contribute to overall system performance?

e. Explain the architecture of a CPU in detail.

## 1.9 Computer Software

Computer software refers to a set of instructions, programs, or data that enable a computer to perform specific tasks or functions. It is a crucial component of a computer system, working in conjunction with hardware to provide a platform for various applications and user interactions. Computer software is classified into two types. They are:

a. System software

b. Application software

### a. System software

System software serves as the backbone of computer operations, overseeing and supporting its functions. It acts as a vital bridge between users and computers, serving as a foundation for running different applications. It also plays a crucial role in organizing files and folders.

System software is crucial for ensuring computers operate smoothly. It encompasses a range of programs and tools essential for basic tasks, hardware resource management, and maintaining a reliable computing environment. It can be further categorized into operating systems, utility software, device drivers, and language processing software, etc.

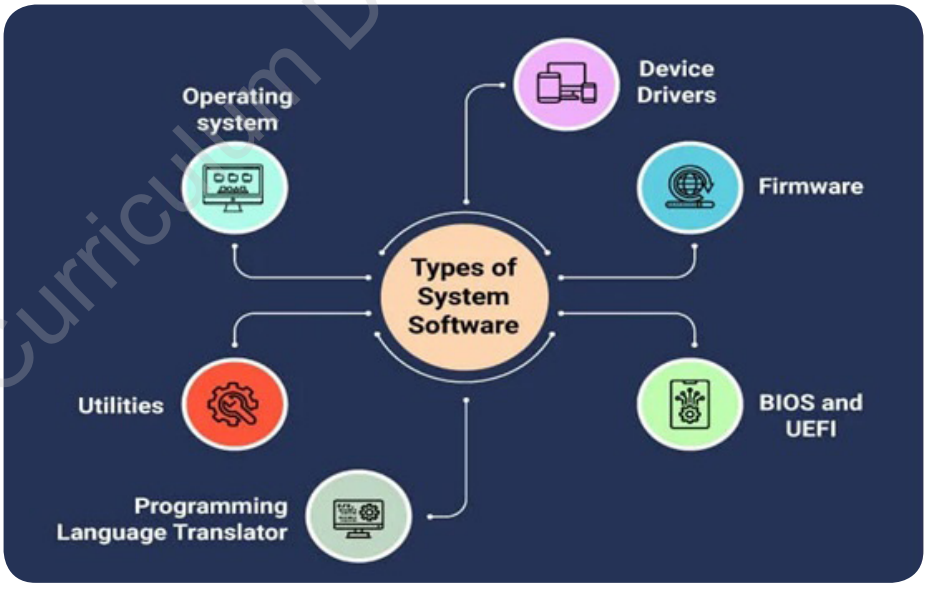



*Figure 1.40*

### i. Operating system

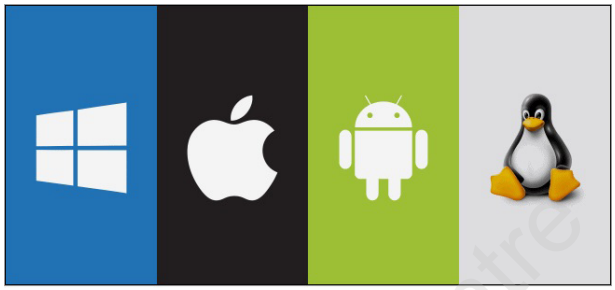


*Figure 1.41*

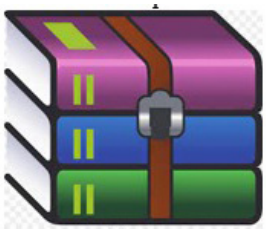
Operating system software is the system software that controls and manages the overall operation of a computer just like a traffic policeman controls the traffic on the road. It is the basic requirement of a computer. It is the first software that is loaded into the computer memory (RAM) when you switch a PC on and works till you shut down the computer. It controls hardware and application software. It acts as an interface between the user and the computer or between the software and the hardware. It makes hardware usable in computers and provides an environment to software and users for doing their tasks. It provides the environment in which you interact with a computer and application software. It is responsible for the smooth and efficient operation of the computer system. Without an operating system in a computer, we cannot operate the computer. Microsoft Windows OS, Macintosh OS X, Fedora, Ubuntu, Android, IBM OS/2, UNIX and LINUX are some examples of operating systems.

### ii. Language processor

A language processor (also called translator) is system software that translates programs written in assembly language or high level language into machine language. Assembler, interpreter and compiler are three types of language processors.

### iii. Utility software

Utility software is the helpful software that performs specific tasks related to the maintenance of computer hardware and data. It helps to keep a computer in a smooth functioning condition. It provides facilities for performing tasks like transferring data and file, recovering lost data and file, searching and removing computer viruses, disk management, etc. Backup software, Download Accelerator (DAP), SplitFile, Anti viruses, WinZip, WinRaR, Disk Defragmenter, Registry Cleaner, HistoryCleaner, CClearner, etc. are some examples of utility software.



WinRaR

CClearner


Avira Antivirus

*Figure 1.42*

### iv. Device drivers software

A device driver is a set of programs that is designed to control and manage the specific device. The device drivers make the devices usable. Every device or hardware like printer, sound card, display card, etc. has its own device driver. Suppose you have brought a new printer and to use the printer, you have to first install the device driver of the printer.


*Figure 1.43*

## b. Application software

Application software refers to programs tailored to meet specific user needs. This collection of programs is crafted to perform distinct tasks such as composing emails, editing photos, creating presentations, listening to music, engineering design, video editing, billing, and more. Packaged Software and Customized (or Tailored) Software are two types of application software.

### i. Packaged/general purpose application software

Packaged software is the readymade software developed for all general users to perform their generalized tasks. Software companies like Microsoft, Adobe, Dropbox, Corel, Oracle, Google, etc., use to develop packaged software. MS-Office, Adobe InDesign, Sybase, SQL Server, Oracle, CorelDraw, Adobe PhotoShop, 3DS Max, Maya, VLC Media Player, Google Meet, Zoom, Spotify, etc. are packaged software.


*Figure 1.44*

### ii. Customized/tailored software

*Figure 1.45*

**Customized (Tailored) software** is the application software which is designed to fulfill the specific requirements of an organization, office or individual. It is useful for the organization, office or person for whom it is developed. SEE Result Processing Software, Hospital Management Software, School Management Software, Bill Processing Software, Air Ticket Reservation Software, Banking Software, etc. are customized software.

## Open sources and proprietary software

Based on the accessibility of source code of software, there are two types of software. They are open source software (OSS) and closed source software (CSS).

Open source software is a type of software that allows its source code to be publicly accessible. It means anyone can view, modify and use the code freely. Linux, Apache, Audacity, OpenOffice, Android and others are open source software.

The close source software (also called proprietary software) is a type of software that does not allow its source to be accessed by any others except the creator or the authorized organization. The CSS usually comes with a license, and users are required to purchase the software to use it. This leads to higher costs and restrictions on the modification and usability of the software. Windows OS, Adobe Photoshop, iTunes, Microsoft Office Suite, AutoCAD, and many others are CSS.


*Figure 1.46*

**Differences between Open-source Software and Proprietary Software**

| Open-source software | Proprietary software |
|---|---|
| Open Source Software refers to the software in which users can view the source code which is freely available on the Internet. | Proprietary Software refers to the software that uses the proprietary and closely guarded code. Users cannot view the source code. |
| The code can be copied, modified, or deleted by other users and organizations. | Only the original authors of the software can access, copy, and alter that software. |
| Open source software is freely available on the Internet or users have to pay a nominal amount of fee to use Open Source Software. | Proprietary Software is also known as Licensed or Closed Source Software and users have to buy a license to use Proprietary Software. |
| There is very little security because anyone can view and modify the source code and because of availability, the source code is open for hackers to practice on. | Proprietary software can be fixed only by a vendor as its source code can only be accessed by the person, team, or organization who created it and maintains exclusive control over it. |

| | |
|---|---|
| Open source software provides better flexibility which means more freedom which encourages innovation. | There is a very limited scope of innovations with the restrictions and all. |
| Examples: Andriod, Linux, Unix, Libre Office, Drupal, GNOME | Examples: Windows OS, Mac OS, Adobe Flash Player, iTunes, Microsoft Office |

**Comparison between the features of system and application software**

| System software | Application software |
|---|---|
| It is designed to manage the resources of the computer system, like memory, I/O device process management, security, etc. | It is designed to fulfill the requirements of the end user for performing day-to-day tasks. |
| It is designed by writing programs in low-level language like machine-level language or assembly language. | It is designed by using high-level language and 4GL. |
| It executes when the system is powered on and runs until the system is powered off. | It executes according to the user's needs. It executes when the user starts and terminates when the user closes it. |
| It is installed on the computer system at the time when the operating system is installed. | It is installed as per the user's requirements. |
| It is capable of running independently. | It cannot run independently without system software. |
| It is independent of the application software, | It needs system software to execute. |
| It is crucial for the effective functioning of a system. | It is not extremely important for the functioning of the system. |

## Introduction to mobile and web applications

A computer program designed for a specific purpose is called an app or application. It is commonly installed on smartphones and tablets for various tasks. Mobile software includes apps and operating systems made for mobile devices like smartphones, tablets, and smartwatches. This software is designed to work with features like touchscreens, GPS, cameras, and mobile networks. It covers many

types of apps, such as games, productivity tools, social media, and navigation apps. It also includes operating systems like Android and iOS, which manage the device's hardware and software.

Web applications are software programs that run on web servers and are used through web browsers over the internet. Unlike traditional desktop applications, they don't need to be installed on your device. You use them through a web page interface. Examples include email services like Gmail, online office tools like Google Docs, and social media sites like Facebook. Web applications are convenient because you can use them on any device with a web browser and an internet connection.


*Figure 1.47*



***Activity 1.3***

**Activity Outcome:**

- Able to describe the main objectives of software.

Required Resources: Chart Papers, Meta Cards, Cardboards, Images, PowerPoint Presentation, etc.

**Procedure:**

Your teacher will divide you into a group of 3-5 each, and let your group collect names of software used in hospitals, hotels, the educational sector, and other different sectors surrounding you. Present and discuss in group about the main objectives of this software.

**Result:** Moderated discussion by the teacher for your reflection with feedback.

## Exercises

**1. Write the full forms of the following abbreviations:**

i. OS ii. CUI iii. DOS iv. GUI v. SPSS

vi. OSS vii. SQL viii. HTML ix. CSS

**2. Choose the correct answer.**

i. What is the purpose of an operating system in a computer system?

a. Manage hardware resources and provide a user interface

b. Perform complex mathematical calculations

c. Store and retrieve data

d. Display graphics on the screen

ii. Which type of software is designed to perform a specific task, such as word processing or spreadsheet calculations?

a. System software b. Application software

c. Utility software d. Firmware

iii. What is software?

a) A physical part of the computer

b) Instructions that tell a computer what to do

c) A type of input device

d) A tool for cleaning computer hardware

iv. Which of the following is an example of application software?

a) Microsoft Word b) CPU c) Hard Drive d) RAM

v. What is the role of the operating system?

a) To provide power to the computer

b) To manage hardware and software resources

c) To store data permanently

d) To print documents

vi. Which of the following is not software?

a) Keyboard b) Web Browser

c) Video Player d) Word Processor

**3. Write short answers to these questions.**

a. What does the term "open source" mean in software?

b. How does open-source software differ from proprietary software?

c. What is system software, and what role does it play in a computer system?

d. Define application software and provide examples of common applications.

e. What is the purpose of utility software, and give examples of utilities?

**4. Write long answers to these questions.**

a. Explain the types of computer software with their respective functions.

b. Differentiate between system software and application software with respective roles in a computer system.

c. Differentiate between open source and proprietary software models with examples of everyday applications that we use.

d. Explain the concept of utility software and its role in enhancing the performance and functionality of a computer system.

## Project Work

1. Collect names of software used in the hospital, hotel, educational sector and other different sectors surrounding you and group discussion about the main objectives of these software.

2. Take a short interview with your teachers, friends, parents and relatives and prepare a field report about what types of computers, laptops or other devices they are using, what types of Operating System the device installs. Also make a list of apps that they have used in their smartphones.

3. Carry out a practical task involving the assembly and disassembly of a computer system, and write a detailed report documenting each step of the process.

# Number System

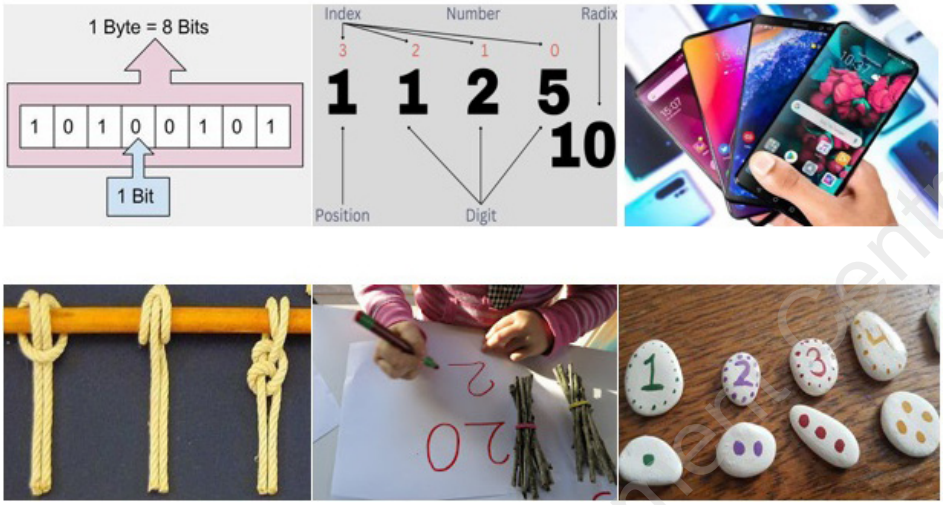



*Figure 2.1*

***Let's think!***

a) What is a digit? What is a computer word?

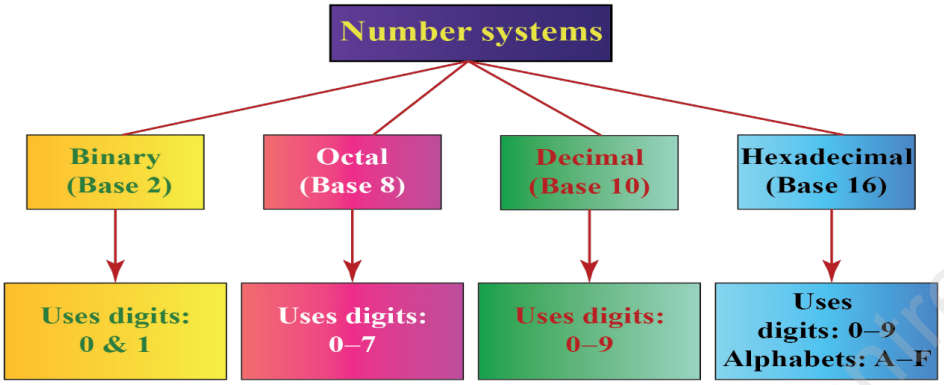
b) Define the base or radix of the number system.

c) Which language is used by computer systems, smartphones, and tablets?

The concept of numbers evolved long ago in human history when people started using their fingers, sticks, pebbles, knots of rope, and different symbols for counting with simple addition or subtraction. Over time, people invented various calculating devices. At present, we use different kinds of numbers every day for different purposes.

## 2.1 Concept of number system

These numbers belong to different number systems, which have different symbols and rules. A number system is a writing system with specific symbols and rules for expressing numbers and doing different calculations. For example, when we count things, we use the decimal number system, which has 10 symbols, from 0 to 9. That means we can make any number we want using the 10 symbols. The decimal number system is easy to use and understand. So, it is the most commonly used number system in our daily life.




Various number systems are currently in use. A number system is differentiated by its base or radix. The base is defined as the total number of digits available in the number system. There are mainly four types of number systems:

- Decimal number system
- Binary number system
- Octal number system
- Hexadecimal number system

## Decimal number system

The decimal number system is the common number system we use in our lives. It has a base of 10 since it uses digits ranging from 0 to 9. In this system, each digit's position represents a distinct power of 10, which includes units, tens, hundreds, thousands, and so on. For example, the decimal number 719 consists of the digit 9 in the units position, 1 in the tens position, and 7 in the hundreds position.

$$
\begin{aligned}
(719)_{10} &= (7 \times 100) + (1 \times 10) + (9 \times 1) \\
&= (7 \times 10^2) + (1 \times 10^1) + (9 \times 10^0) \\
&= 700 + 10 + 9 \\
&= 719
\end{aligned}
$$

## Binary number system

The binary number system uses only two digits: 0 and 1. Thus, it is a base-2 number system. It is used by computers. For example, $(110111)_2$ is a binary number.

## Octal number system

The octal number system is the base-8 number system that uses 8 digits (0 to 7). Some examples of octal numbers are $(207)_8$, $(5706)_8$, $(601022)_8$, etc.

## Hexadecimal number system

The hexadecimal number system is the base-16 number system that uses 16 symbols, i.e., 10 digits:0, 1, 2, 3, 4, 5, 6, 7, 8, 9, and 6 letters: A, B, C, D, E, F. The letters from A to F represent the values 10 to 15 respectively. Some examples of octal numbers are $(8F5)_{16}$, $(AC4)_{16}$, $(B52)_{16}$, $(ACD)_{16}$, etc.

| Hexadecimal | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Decimal | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |

## Number system chart

The following number system chart shows the base values and the digits of different number systems.

| Number system | Base | Symbols / Digits Used |
|---|---|---|
| Binary | 2 | 0, 1 |
| Octal | 8 | 0, 1, 2, 3, 4, 5, 6, 7 |
| Decimal | 10 | 0, 1, 2, 3,4, 5, 6, 7, 8, 9 |
| Hexadecimal | 16 | 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 and A, B, C, D, E, F |

## Application of number system conversion

Number system conversion is important in many fields, especially in computer science and digital technology. Computers use binary code to process information, so it's essential to understand how to convert between binary, decimal, octal, and hexadecimal systems. In digital logic design, circuits and components use binary and hexadecimal notations, making these conversions necessary. IP addresses in networking are often shown in hexadecimal or dotted-decimal notation, so knowing how to convert them is useful. Number system conversion is also used in cryptography, graphics, and signal processing to represent data efficiently. Overall, understanding number system conversion is crucial for anyone working with digital information.

## 2.2 Binary calculation

There are four main types of binary calculation:

i. Binary addition,

ii. Binary subtraction,

iii. Binary multiplication and

iv. Binary division.

### Binary addition

Binary numbers can be added in the same way as we add decimal numbers. The rules for adding binary numbers are

| A | B | A+B |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 with carry 1 (i.e 10) |

### Steps of binary addition:

Step1 Align the numbers that you have to add in the same way you align while adding decimal numbers.

Step2 Start with the two numbers from the right end position.

Step3 Add the numbers by following the rules of binary addition as given above.

Step4 After adding the first column from the right position, add a second column, and so on.

Step5 Repeat the above steps until all the columns are finished.

Examples of binary addition

$$\begin{array}{r} 1\ 1\ 0\ 0 \\ +\,0\ 1\ 0\ 1 \\ \hline 1\ 0\ 0\ 0\ 1 \end{array}$$

Hence, 1100 + 0101 = 1 0 0 0 1

### Binary subtraction

Binary numbers can be subtracted in the same way as we subtract decimal numbers. The rules for subtracting binary numbers are:

| A | B | A-B |
|---|---|---|
| 0 | 0 | 0 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |
| 0 | 1 | 1 with borrowing 1 from the left column |

***Example:***

```
  1 0 1 1 1
    -1 0 1
  ---------
  1 0 0 1 0
```

Hence, 1 0 1 1 1 – 1 0 1 = 1 0 0 1 0

## 2.3 Number conversion

People commonly use the decimal number system in their daily lives. Computers use binary, octal, and hexadecimal number systems according to their model or brand. We cannot easily understand binary number systems, this way computers can't understand decimal number systems. So there are different methods to convert one number system to another as follows:

### Decimal to Binary, Octal, Hexadecimal

Decimal to Binary number system conversion

To convert decimal numbers to binary numbers, follow the following steps:

Step 1. Divide the given decimal number by 2 and write down the remainder.

Step 2. Divide the quotient by 2 and again write down the remainder.

Step 3. Repeat the process until the quotient becomes zero.

Step 4. Write the remainder from bottom to top.

*Example:* $(46)_{10} = (?)_2$

| 2 | 46 | Remainder |
|---|---|---|
| 2 | 23 | 0 |
| 2 | 11 | 1 |
| 2 | 5 | 1 |
| 2 | 2 | 1 |
| 2 | 1 | 0 |
| | 0 | 1 |

Therefore, $(46)_{10} = (101110)_2$

## Decimal to Octal number system conversion

To convert decimal numbers to octal numbers, follow the following steps:

Step 1. Divide the given decimal number by 8 and write down the remainder.

Step 2. Divide the quotient by 8 and write down the remainder.

Step 3. Repeat the process until the quotient becomes zero.

Step 4. Write the remainder from bottom to top.

*Example:* $(345)_{10} = (?)_8$

| 8 | 345 | Remainder |
|---|---|---|
| 8 | 43 | 1 |
| 8 | 5 | 3 |
| | 0 | 5 |

Therefore, $(345)_{10} = (531)_8$

## Decimal to Hexadecimal number system conversion

To convert decimal numbers to hexadecimal numbers, follow the following steps:

Step 1. Divide the given decimal number by 16 and write down the remainder.

Step 2. Divide the quotient by 16 and write down the remainder.

Step 3. Repeat the process until the quotient becomes zero.

Step 4. Write the remainder from bottom to top.

***Example:*** $(88)_{10} = (?)_{16}$

| 16 | 88 | Remainder |
|---|---|---|
| 16 | 5 | 8 |
| | 0 | 5 |

Therefore, $(88)_{10} = (58)_{16}$

## Binary, Octal, Hexadecimal to Decimal

### Binary to Decimal number system conversion

To convert binary numbers to decimal numbers, follow the following steps:

Step 1. Multiply each binary digit with its place value (positive powers of two that is $2^0, 2^1, 2^2, 2^3, 2^4$ .....)

Step 2. Add all the products calculated in step 1 and the total number is the decimal equivalent number.

***Example:*** $(10011)_2 = (?)_{10}$

$(10011)_2 = 1 \times 2^4 + 0 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0.$

$= 1 \times 16 + 0 \times 8 + 0 \times 4 + 1 \times 2 + 1 \times 1.$

$= 16 + 0 + 0 + 2 + 1.$

$= 19.$

## Octal to Decimal number system conversion

To convert an octal number to a decimal number, follow the following steps:

Step 1. Multiply each octal digit with its place value (positive powers of eight that is $8^0, 8^1, 8^2, 8^3, 8^4$ .....)

Step 2. Add all the products calculated in step 1 and the total number is the decimal equivalent number.

*Example:* $\mathbf{(157)_8 = (?)^{10}}$

$(157)_8 = 1 \times 8^2 + 5 \times 8^1 + 7 \times 8^0.$

$= 1 \times 64 + 5 \times 8 + 7 \times 1.$

$= 64 + 40 + 7.$

$= 111.$

## Hexadecimal to Decimal number system conversion

To convert hexadecimal numbers to decimal numbers, follow the following steps:

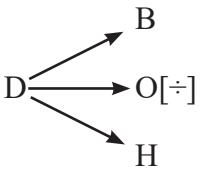
Step 1. Multiply each octal digit with its place value (positive powers of two that is $16^0$,$16^1$, $16^2$,$16^3$, $16^4$ ……)

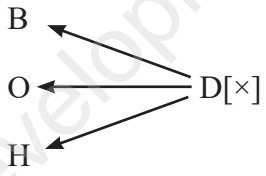
Step 2. Add all the products calculated in step 1 and the total number is the decimal equivalent number.

***Example:*** $(1AC)_{16} = (?)_{10}$

$$(1AC)_{16} = 1\times16^2 + 10 \times16^1 + 12 \times16^0$$
$$= 1 \times 256 + 10 \times 16 + 12 \times 1$$
$$= 256 + 160 + 12$$
$$= 428$$

***Note:***




| Decimal | Hexadecimal | Octal | Binary |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 |
| 2 | 2 | 2 | 10 |
| 3 | 3 | 3 | 11 |
| 4 | 4 | 4 | 100 |
| 5 | 5 | 5 | 101 |
| 6 | 6 | 6 | 110 |
| 7 | 7 | 7 | 111 |

| | | | |
|---|---|---|---|
| 8 | 8 | | 1000 |
| 9 | 9 | | 1001 |
| 10 | A | | 1010 |
| 11 | B | | 1011 |
| 12 | C | | 1100 |
| 13 | D | | 1101 |
| 14 | E | | 1110 |
| 15 | F | | 1111 |

## Binary to Hexadecimal and vice versa

Binary to Hexadecimal number system conversion

To convert binary numbers to hexadecimal numbers, follow the following steps:

Step 1. Arrange binary digits in groups of four from right to left.

Step 2. Write the respective hexadecimal number for each binary group.

Step 3. Give the result base 16.

***Example:*** **$(100000110101)_2 = (?)_{16}$**

1000 0011 0101

| 4 Digit Combination | 1000 | 0011 | 0101 |
|---|---|---|---|
| Octal Number | 8 | 3 | 5 |

Therefore, $(100000110101)_2 = (835)_{16}$

## Hexadecimal to Binary number system conversion

To convert hexadecimal numbers to binary numbers, follow the following steps:

Step 1. Write binary equivalent 4 digits group to each hexadecimal number.

Step 2. Give the result base 2.

***Example:*** **$(9A3)_{16} = (?)_2$**

| Hexadecimal number | 9 | A | 3 |
|---|---|---|---|
| Binary equivalent | 1001 | 1010 | 0011 |

Therefore, $(9A3)_{16} = (100110100011)_2$

*Activity*

**Activity Outcome -**

Able to describe number systems and binary calculations with diagrams/ flowcharts.

**Required Resources:** Chart Papers, Meta Cards, Cardboards, Images, PowerPoint Presentation, etc.

**Procedure:**

Your teacher will divide you into a group of 3-5 each, and let your group present on any of the following topics:

a. number system conversion

b. Binary calculation

**Result:** Moderated discussion by the teacher for your reflection with feedback.

## Exercises

**1. Choose the correct option.**

i. What is the result of the binary addition: 1101 + 1011?

a. $(11010)_2$ b. $(11000)_2$

c. $(11110)_2$ d. $(11101)_2$

ii. When adding two binary numbers, what is the carrying value in binary addition?

a. 100 b. 1 c. 10 d. 11

iii. In binary multiplication, what is the result when multiplying any binary digit by 0?

a. 1 b. 11 c. 0 d. The original digit

iv. What is the base of the octal number system?

a. 2 b. 7 c. 8 d. 10

v. What is binary equivalent to the octal number 64?

a. $(110100)_2$ b. $(1101111)_2$

c. $(1110001)_2$ d. $(11000101)_2$

vi. In the hexadecimal system, what does the symbol 'A' represent?

a. 11 b. 12 c. 10 d. 13

## 2. Answer these questions.

a) What is a number system?

b) Define the base or radix of the number system.

c) List out the different types of number systems.

d) What is a hexadecimal number system? Write down the symbols used in the hexadecimal number system.

## 3. Calculate the following as indicated:

**a. Perform the following binary addition:**

i. $(11110)_2 + (1001)_2$

ii. $(1011)_2 + (1001)_2$

iii. $(101011)_2 + (11011)_2$

iv. $(1010)_2 + (110)_2$

v. $(101001)_2 + (1110)_2$

vi. $(100001)_2 + (100011)_2$

vii. $(100111)_2 + (11010)_2$

viii. $(110001)_2 + (100101)_2$

**b. Perform the following binary subtraction:**

i. $(1100)_2 - (1001)_2$

ii. $(1001)_2 - (110)_2$

iii. $(11101)_2 - (1010)_2$

iv. $(101100)_2 - (10011)_2$

v. $(11111)_2 - (10110)_2$

vi. $(110011)_2 - (10100)_2$

vii. $(100100)_2 - (1110)_2$

viii. $(1000001)_2 - (10101)_2$

## 4. Convert the given numbers as indicated:

**a. Decimal to Binary Conversion**

i. $(56)_{10}$

ii. $(78)_{10}$

iii. $(123)_{10}$

iv. $(345)_{10}$

v. $(540)_{10}$

vi. $(572)_{10}$

vii. $(546)_{10}$

viii. $(1098)_{10}$

ix. $(2103)_{10}$

x. $(445)_{10}$

**b. Binary to Decimal Conversion**

i. $(1101)_2$

ii. $(1010)_2$

iii. $(10010)_2$

iv. $(10110)_2$

v. $(101001)_2$

vi. $(11100111)_2$

vii. $(111100)_2$

viii. $(10010011)_2$

ix. $(1011100)_2$

x. $(100110)_2$

**c. Decimal to Octal Conversion**

i. $(69)_{10}$ ii. $(216)_{10}$ iii. $(767)_{10}$

iv. $(79)_{10}$ v. $(443)_{10}$ vi.$(413)_{10}$

vii.$(765)_{10}$ viii.$(1334)_{10}$ ix.$(1825)_{10}$

x. $(2783)_{10}$

**d. Octal to Decimal Conversion**

i. $(124)_8$ ii.$(242)_8$ iii. $(333)_8$ iv. $(763)_8$

v. $(103)_8$ vi.$(451)_8$ vii. $(3401)_8$ viii. $(1045)_8$

ix. $(438)_8$ x. $(611)_8$

**e. Decimal to Hexadecimal Conversion**

i. $(55)_{10}$ ii. $(540)_{10}$ iii. $(225)_{10}$ iv. $(880)_{10}$

v. $(2046)_{10}$ vi. $(2024)_{10}$ vii.$(6678)_{10}$

**f. Hexadecimal to Decimal Conversion**

i. $(56)_{16}$ ii. $(67)_{16}$ iii. $(558)_{16}$ iv. $(B74)_{16}$

v. $(20D3)_{16}$ vi. $(DEF)_16$ vii. $(6E3)_{16}$ viii. $(63F)_{16}$

**g. Binary to Hexadecimal Conversion**

i. $(1000110)_2$ ii. $(11001)_2$ iii. $(1111000)_2$

iv. $(11110000111)_2$ v. $(101010110)_2$ vi. $(1110010110)_2$

vii. $(11011001)_2$ viii. $(1001100)_2$

**h. Hexadecimal to Binary Conversion**

i. $(D4)_{16}$ ii. (643)16 iii. (189)16 iv. (2BF)16

v. $(A9F)_{16}$ vi. (FACE)16 vii. (FB4)16 viii. (1B2)16

# Unit 3 Block Programming

Have you ever wondered how apps like Google Classroom, Duolingo, or educational games like Code Combat are created? While traditional programming requires writing complex lines of code, block programming makes coding as simple as snapping together pieces of a puzzle.

*Figure 3.1*

***Let's think!***

a. What do you think is easier—solving a puzzle or writing lines of code? Why?

b. Can you guess how block programming might help us create educational tools or fun learning games?

Can block programming be employed to create an educational game or simulation that teaches users about disaster readiness, response strategies, and resilience-building techniques?

In our rapidly advancing world, more and more people are getting access to computers. These computers help us do all sorts of things, like going online and using different programs such as Word, Excel, Notepad, Photoshop, etc. But have you ever wondered how games work and how awesome gadgets like smartwatches operate? Well, it's all because of programming. Programming is like giving instructions to computers to make them do all the amazing things we see and use every day. Learning programming lets you understand how computers work and how to make them do what you want. That's why programming is such a valuable skill to have in today's world.

Have you ever had a super cool idea for a game or a gadget, but felt lost on how to bring it to life? Don't worry, you're not alone! Starting with programming can feel like stepping into a new world with lots of tricky stuff to learn. It's like trying to understand a language with tons of confusing rules. But there is no need to worry! There is a friendlier and easier way to get into programming, and it is called block programming. Instead of typing out long lines of code, block programming lets you drag and drop colorful blocks to create your programs.

With block programming, you can make all sorts of cool stuff like games, animations, and gadgets without getting lost in complicated code. So, if you're ready to turn your ideas into reality, block programming is your new best friend!

**Have you ever played puzzles?**

Do you know Block programming is just like solving puzzles?

Adding blocks to code, just like solving puzzles in block programming, we add blocks of code to programs and instruct computers to perform some tasks.

## 3.1 Concept of block programming

Block programming is made up of two words: "Block" and "Programming". Blocks are like puzzle pieces that you can join to form a beautiful picture. On the other hand, programming is a way of giving human instructions to a computer or electronic device about what they should do. Block programming combines both these elements to provide a user-friendly way to programming.

*Figure 3.2: Concept of Block programming*

Block programming is a popular programming language made for beginners, especially children, to teach them the basics of programming by using colorful blocks to build computer programs, videos, and games. It supports the use of graphics, animation, and sound without the need to understand complex programming languages. Some examples of block programming platforms are Scratch, Blockly, and Snap.

***Let's try this!***

Which of the following is an example of block programming?

a) Blockly b) Snap! c) Scratch d) All of the above

### Advantages of block programming

a) The users do not need to worry about syntax or grammar like other programming languages.

b) The users have more time to focus on creativity, building logic, and their ability to solve problems.

c) The chances of human typing error are less since the users do not need to write codes.

d) Unlike other programming languages, block programming is colorful, visually appealing, and engaging for beginners and children.

e) Block programming offers an interactive platform to learn programming concepts so that the users can easily switch to text-based programming in the future.

## Applications of block programming

Block programming has many practical applications in various fields due to its user-friendly and accessible nature. Here are some applications of block programming:

### a) Educational purpose

Block programming is used in schools and educational programs to teach coding concepts. Its visual approach makes programming appealing for beginners. It helps them develop a strong foundation in programming so that they can easily transition to text-based coding languages in the future.


*Figure 3.3*

### b) Game development

Block programming is also used to create games. Users can make different characters, and elements, and build game logic. It allows them to focus more on gameplay logic without worrying about complex coding.

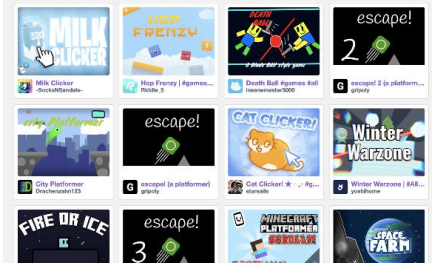



*Figure 3.4*

### c) Mobile App development

Block programming can be used to develop mobile applications. Users can design user interfaces, define functionalities, and assemble different code blocks to develop Android and iOS apps without the need for complex coding knowledge.

*Figure 3.5*

### d) Robotics

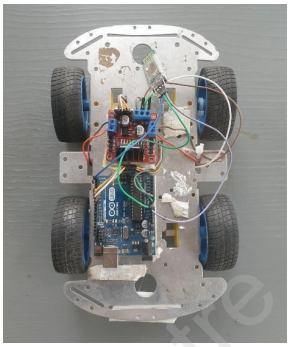


*Figure 3.6*

Block programming is also used to program robotic devices. Its visual programming tools allow users to design movements of robots, define their behaviors, and control sensors. There are devices like micro: bit and Arduino UNO, which can be used with block programming to create robotic projects. While this provides a learning platform for students, it also helps them build creativity and interest in the experimentation of robotics projects.

### e) Web development

*Figure 3.7*

With the help of block programming's visual interface, users can create interactive websites. They can define web page elements, user interactions, and data handling using visual blocks. This visual approach acts as a bridge for individuals to understand web development concepts before advancing to traditional text-based programming languages.

### f) Creative projects

*Figure 3.8*

Block programming is also used by artists to create interactive projects like animations. There are platforms like Scratch and Blockly, where you can find many projects created till date.

### g) Modeling and simulation

*Figure 3.9*

Modeling and simulation is the process of using physical or logical representation of any given system to generate data and help make predictions about the system. Block programming is a suitable platform that makes modeling and simulation easy by allowing users to define different parameters, interactions, and scenarios using visual blocks.

### h) Data science and visualization

Block programming can also be used to design algorithms or processes, arrange data, and perform analysis of that data using a visual interface. Those analyses can be visualized by creating interactive charts and graphs using block programming.

*Figure 3.10*

## 3.2 Introduction to Scratch

Today's world is about technology and innovation. More and more people are gaining access to computers and digital devices. Our devices enable diverse tasks from productivity software to creative tools like Photoshop. But have you wondered about the magic behind games and gadgets? The main thing is programming, giving computers instructions for incredible tasks. Learning programming unlocks understanding and empowers us to create digital solutions. Let's explore Scratch, a popular and user-friendly programming platform.


*Figure 3.11*

Scratch is one of the most popular and widely used block programming languages. It was developed by Lifelong Kindergarten Group at MIT Media Lab, and is an open-source platform, which means that the software is free and publicly available to everyone. It has a large and supportive community of users, who share their projects and help each other on the Scratch website. Scratch 3.0 is the current Version of Scratch. Scratch a real software development tool with a colorful interface and presentation style. It allows users to create various things including games and interesting gadgets while giving them an experience of programming. Even though Scratch is fun and engaging to use, users will face similar challenges like a professional programmer in terms of logic building and problem solving. With Scratch, you can also make your dream projects such as games and even controlling robots or devices a reality.

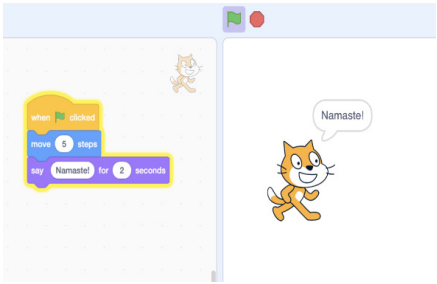



*Figure 3.12*

In the given figure, you can see a simple program that makes a cat character move and say," Namaste" with the help of Scratch.

In this section, we will learn how to build programs like this. However, first, let's learn how to download Scratch on your computer.

***Let's know more!***

Discuss with your classmates and list out any 4 different programming platforms like scratch on the concept of block programming. You can take the help of a teacher.

| | |
|---|---|
| a) | b) |
| c) | d) |

## Downloading Scratch on PC

Scratch is available for offline use on your computer. To use Scratch offline, you must download the Scratch application through their website, following these steps:

1. Go to your web browser and open the link: https://scratch.mit.edu/download.

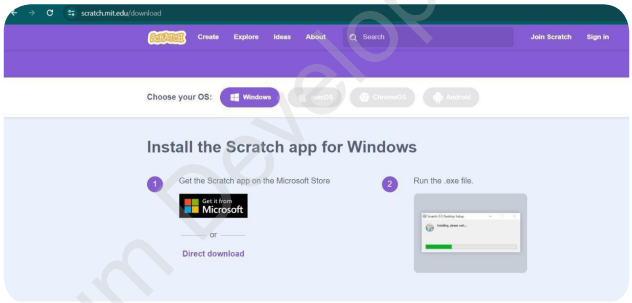



*Figure 3.13*

2. Download the Scratch installation file. Once downloaded, open the .exe file to install the application on your computer.

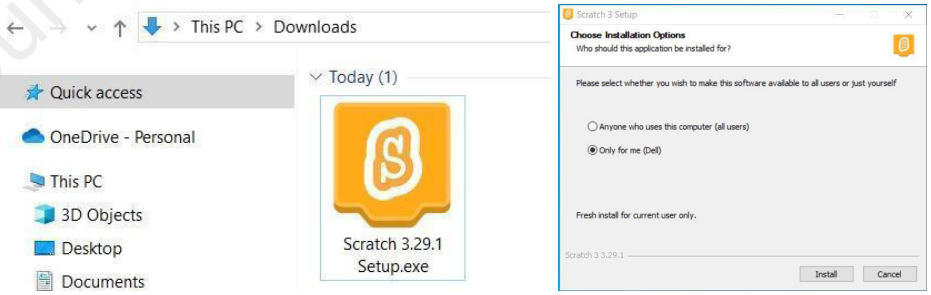



*Figure 3.14*

3. Once the installation is complete, open the Scratch application. You will find an interface as shown below, depending on your Scratch version.

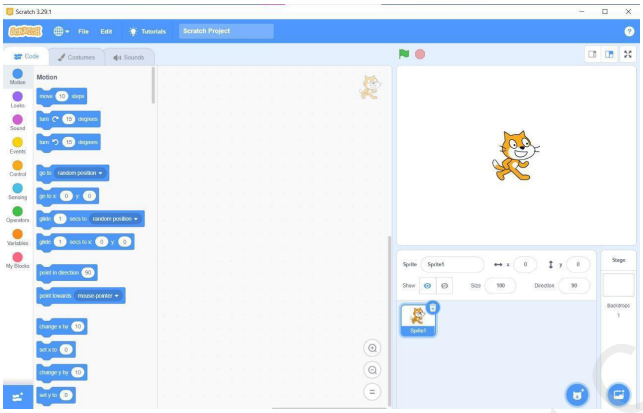


*Figure 3.15*

## Using Scratch online on PC

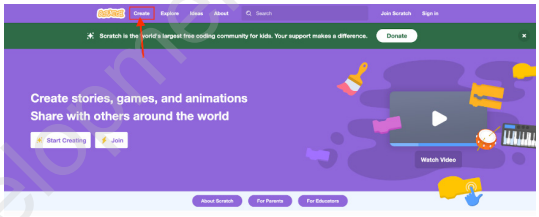



*Figure 3.16*

You can also use Scratch online on the internet from the Scratch website: https://scratch.mit.edu/. Once you are on the website, click on "Create" and start creating any projects that you want.

Remember, when using Scratch online, you will need to create an account to save your work or share your projects with others. To create an account, click on "Join scratch" on the top right corner and fill your details. Use the same details to login to your account later.

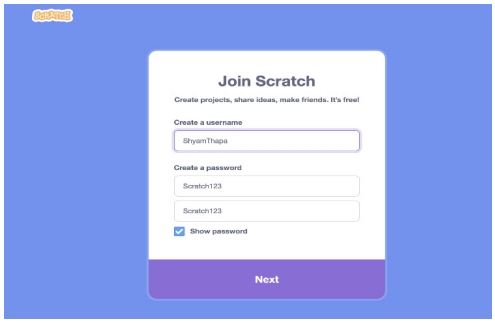



*Figure 3.17*

## Scratch for mobile.

Scratch also has a mobile version for users who prefer to create and explore projects on their mobile devices. To access Scratch on mobile, you can download the Scratch app from your device's app store. Once downloaded, you can open the app and start enjoying Scratch on your mobile device.


*Figure 3.18*

*Classroom activity: 3.1*

As you already learned how to download Scratch on both mobile phones and PCs, discuss with the class which version of Scratch you downloaded and where you saved it on both platforms.

## Features of Scratch

Scratch is a visual programming language that can be used to implement coding logic. It is useful for beginners, programmers, students, or content creators. Scratch programming helps to develop creative thinking, logical reasoning, and the ability to collaborate with others. Here are some of the features of Scratch programming:

a) **Drag and Drop Interface:** Helps to create code by assembling blocks.

b) **Event Driven:** Respond to events such as clicking, dragging, or pressing key.

c) **Data and Variable:** Scratch allows variables to store and manipulate data within a program.

d) **Loops and Condition:** Easily control the flow of program execution.

e) **Sounds:** Allows the addition of sound and music to enhance project interactivity.

f) **Community Sharing:** Programmers can use Scratch to share their projects with others, allowing everyone to explore the amazing creations from the global Scratch community.

## Scratch Interface

Scratch has a colorful and engaging interface. Both the online and offline platforms of Scratch are the same. The interface has four main components: stage, sprite, blocks palette, and script area.

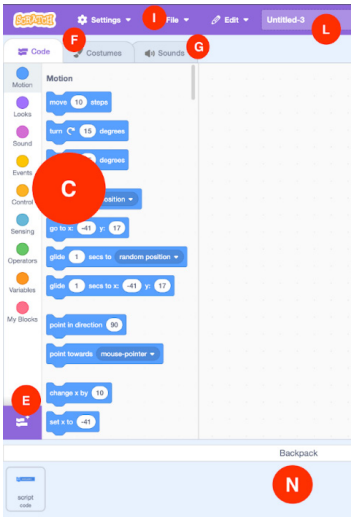



*Figure 3.19*

a. **Stage:** You can see the result of your program in the stage. You can do so by clicking on the green flag.

b. **Sprite:** Each object in Scratch is called a sprite. By default, you get a cat sprite, which can be changed through the "Choose a Sprite" option.

c. **Blocks palette:** The blocks palette is similar to the paint color palette used by painters. The blocks are represented by different colors, which you can use to program your sprites.

d. **Scripts area:** You can drag the blocks and drop it in this area, and connect them with other blocks to create your own piece of code.

e. **Extensions library:** You can add additional features and functionalities to your Scratch program through this library, such as Music, Pen, micro:bit, etc.

f. **Costumes tab:** This tab is also called the paint editor, where you can draw your own characters or sprite.

g. **Sound tab:** This tab is used for adding sound effects to your sprite or project.

h. **Sprite info pane:** This portion of Scratch shows all the sprites added to your project. You can select which sprite you want to program by clicking the sprite in this section.

i. **File tab:** This tab has options to open new projects, load old projects as well as save projects.

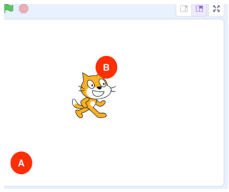



*Figure 3.20*

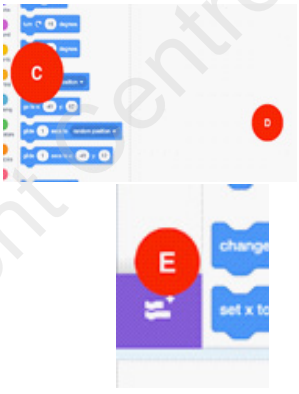



*Figure 3.21*

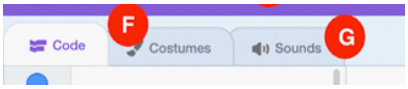



*Figure 3.22*

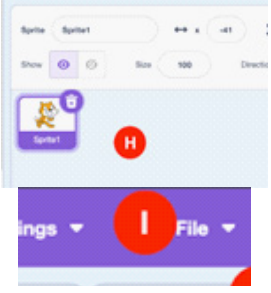



*Figure 3.23*

j. **Choose a Sprite:** This tab allows you to choose a different sprite from the Scratch's in-built library. There are a total of 339 sprites in the Scratch library.

k. **Choose a Backdrop:** A backdrop is an image which can be displayed on stage. This tab allows you to choose a backdrop from Scratch library.

l. **Project Name:** This tab displays the name of your project. You can simply change your project name by clicking on it.

m. **Green flag:** This button is like the play button for your sprite. If you choose the event block "when green flag clicked", you need to click this button to see the output of your code.

n. **Backpack:** This portion is a space to add your blocks, which can be used in different projects.

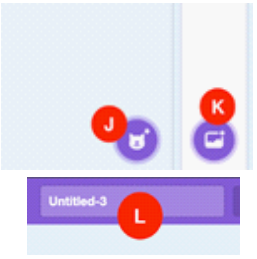



*Figure 3.24*

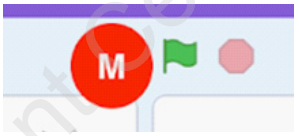



*Figure 3.25*

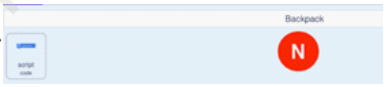



*Figure 3.26*

***Let's understand more!***

Discuss how the original sprite that is cat can be changed and what object can we keep instead of cat.

## Concept of Block

Block programming uses colorful blocks to show actions or commands. You put them together by dragging and dropping to make a list of things to do. It's like playing with building blocks, but you're building programs! This way of coding is easy for beginners, like you, to understand. For example, to move a character or make it perform an action like jump, simply you can select and connect the corresponding blocks. It helps you be creative and figure out problem solving skills.

***Assess yourself!***

As you already have knowledge on scratch interface, open scratch and try to change the sprite from cat to any other object and create a basic program like saying "Namaste!".

- Use the looks block "say Hello!" and change the value to "Namaste!"

## Component of Scratch

The block component of Scratch is one of the main components which makes this programming language unique. These are pre-programmed codes with unique functions that are represented in the form of colorful block size representation. These blocks are like puzzle pieces that you can join to form a beautiful picture.

In Scratch, there are a total of 9 different types of blocks available in the blocks pallete. These blocks are grouped together as per their functionality and can be used to control the sprite's behavior, such as:

- Motion block
- Looks block.
- Sound block.
- Events block
- Control block.
- Sensing block
- Operators block.
- Variables block
- My block

### 1. Motion block

This block makes the sprite or character move, turn, point, or glide the sprite on the stage. You can also set or change the sprite's position, direction, or rotation style.

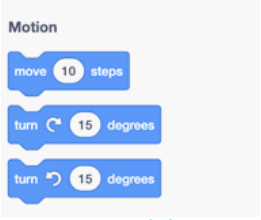
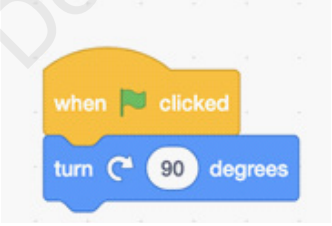



*Figure 3.27*

*For example:*

| | | |
|---|---|---|
|  |  |  |
| Default sprite in the beginning | A motion block "turn 90 degrees" combined with an event block "when clicked" | Output of the code when is clicked |

### 2. Looks blocks

This block allows you to change the appearance of the sprite, such as its costume, color, size, or visibility. You can also make the sprite say or think something through it. For example:

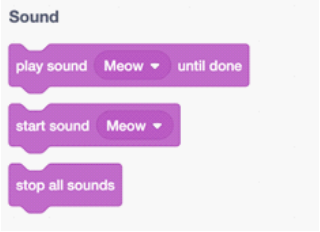
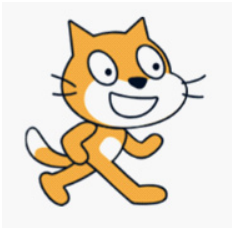
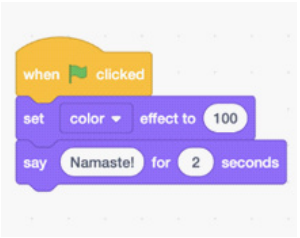



*Figure 3.28*

| | | |
|---|---|---|
|  |  |  |
| Default sprite in the beginning | 2 looks blocks "set color effect to 100" and "say Namaste! for 2 seconds" combined with with an event block "when clicked" | Output of the code when is clicked |

## 3. Sound block

This block lets you play, stop, or change the volume of sounds. You can also make the sprite play a note, a drum, or a sound effect, or record your own sound using the microphone. For example:

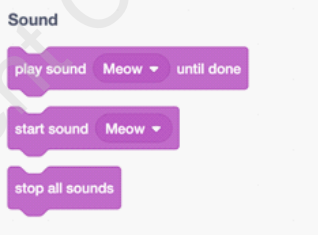
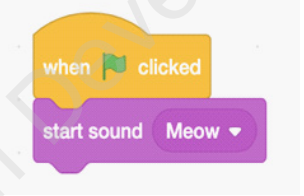



*Figure 3.29*

| | | |
|---|---|---|
|  |  | A sound block "start sound Meow" combined with an event block "when clicked"<br>Makes a sound when is clicked. |

## 4. Events block

This block allows you to trigger the actions in your project, such as when the green flag is clicked, when a key is pressed, or when a message is received. For example:

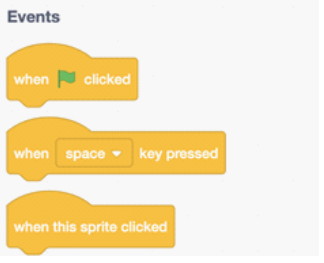
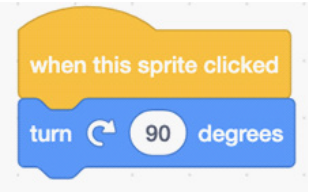



*Figure 3.30*

| | | |
|---|---|---|
|  |  |  |
| Default sprite in the beginning | Same program as the first example, but event block "when this sprite clicked" is used | Same output as the first example, but by clicking on the sprite |

## 5. Control block

This block allows you to control the flow of the sprite's code by repeating, waiting, or stopping. You can also use conditional statements, such as if, if-else, or wait until, to make the sprite do something only when a certain condition is true. For example:

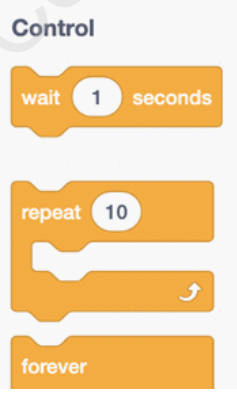
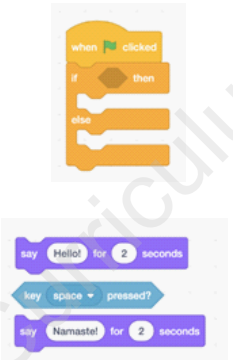
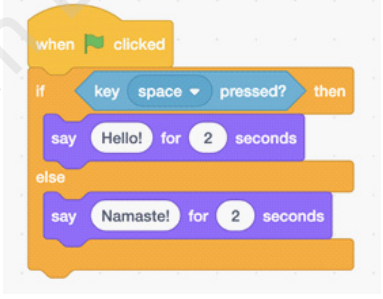


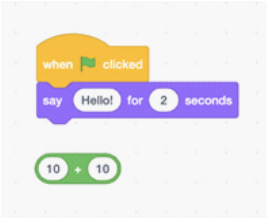
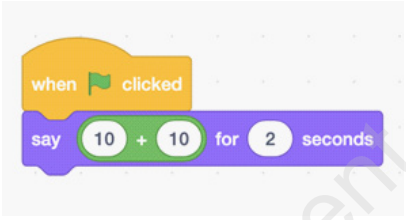

*Figure 3.31*

| | | |
|---|---|---|
|  |  |  |
| A control block "if ___ then () else ()" is combined with event block "when clicked". | A sensing block "key space pressed?" is dropped in "if condition". Then, 2 looks blocks "say Hello! for 2 seconds" and "say Namaste! for 2 seconds" combined in "if and else case" respectively. | Two outputs of program when clicked with and without pressing "space" on the keyboard |

## 6. Operators block


*Figure 3.32*

This block allows you to perform mathematical calculations, such as addition, subtraction, multiplication, division, or random numbers. You can also compare values, such as equal, greater, less, or between, as well as combine or manipulate text. For example:

| | | |
|---|---|---|
|   |   |   |
| A looks block "say Hello! for 2 seconds" combined with an event block "when clicked" | An operator block "10 + 10" dragged and dropped in place of default "Hello!" parameter in the looks block | Output of program displayed when is clicked |

***Let's anwer these questions!***

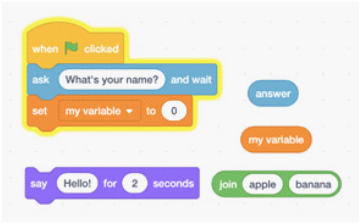
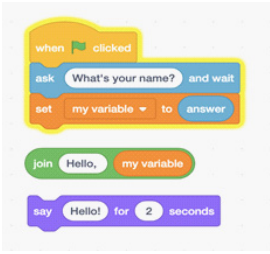
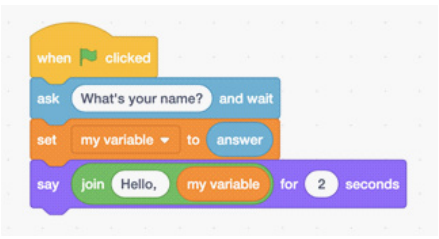
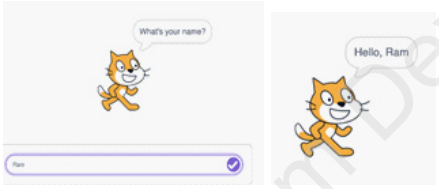
a. What are operator blocks?

b. What control block?

## 7. Variables block


*Figure 3.33*

This block allows you to create and use your own variables to keep track of scores, names, levels, or anything else that changes during the project. You can also use these blocks to set, change, or show the value of a variable. For example:

| | |
|---|---|
|  | A variable block "set my variable to 0" is combined with a sensing block "ask What's your name? and wait". We have a sensor block "answer" which takes the user input. |
|  | The variable parameter 0 is replaced with the sensor block "answer" to put the value in the variable block "my variable". The operator block "join apple banana" is also replaced with a text "Hello," and block "my variable". |
|  | All the blocks are connected together to form this piece of code. |
|  | The program asks the user to respond and replies, "Hello, name." |

***Let's check!***

Which blocks help you store data?

a) sensing block (b) variable block (c) control block (d) my block

## 8. My blocks

This block allows you to create and use custom blocks you make by yourself. You can use custom blocks to group together a set of blocks that you want to use multiple times or to create new blocks that perform a specific function. For example:

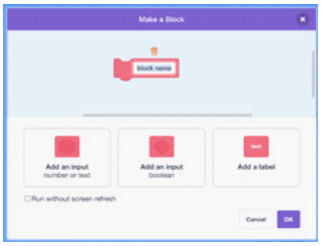



*Figure 3.34*

*Lab activity*

Create a program to add the sum of 2 numbers in which you have to add your own block of code, create a block of code that fulfills the need of the program.

Now that you have a basic understanding of Scratch's components, let us try to build a game using these components.

## Demonstration: Dustbin Game

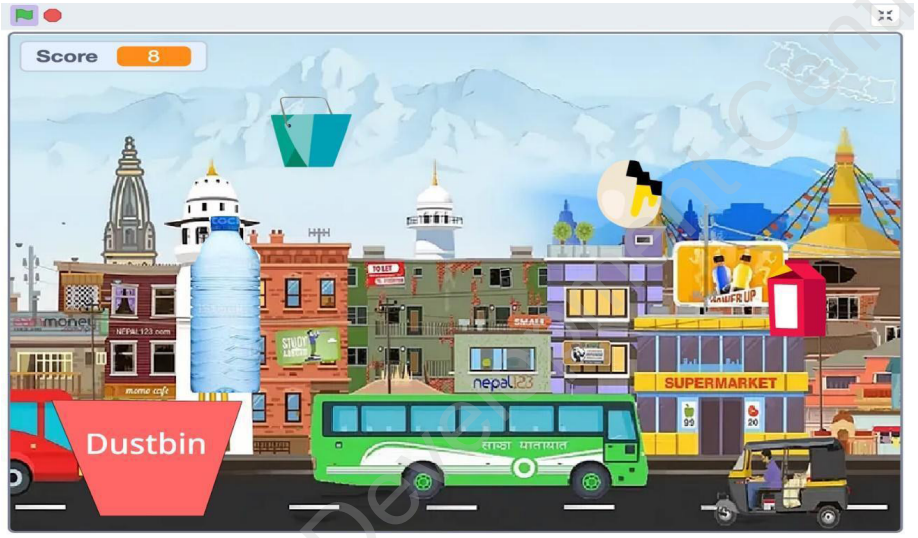



*Figure 3.35*

**Step 1:** By default, you get a cat sprite. Delete it from the sprite info pane, and click on "Choose a sprite" to add new sprites from the library or draw your own sprite: the dustbin.

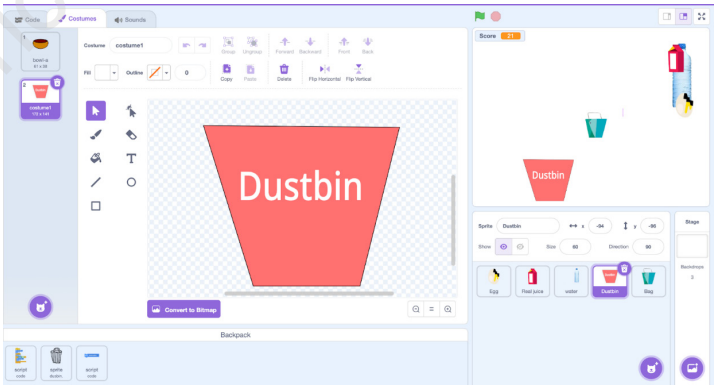



*Figure 3.36*

**Step 2:** Select the dustbin sprite from the sprite info pane to program with the code shown below. Find these blocks from the blocks palette and combine them as shown. You can find these blocks by referring to their color. For example, "when clicked" is an event block, which is yellow in color, look blocks like "hide " and "show " are purple in color and so on.

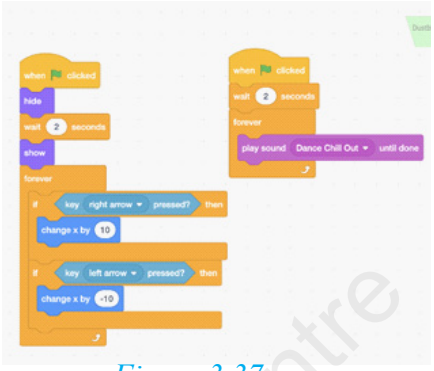



*Figure 3.37*

**Step 3:** Now, click on the other sprites (i.e. eggs, bottles, etc.) one by one to program them. Find the blocks shown below and combine them. You can copy and paste the same blocks for other sprites.


*Figure 3.38*

**Step 4:** You can also add a backdrop to your stage from the "Choose a backdrop" option.

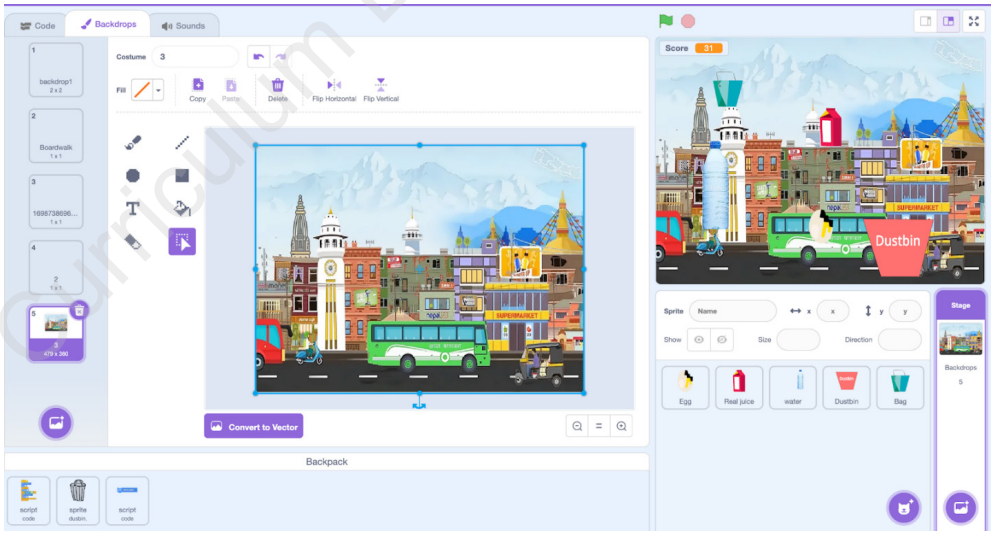



*Figure 3.39*

**Step 5:** Now that everything is complete, you must save your project so that you can view it later, or share it with others.

If you are using Scratch online, make sure you have created your account and logged in. Once you are logged in, you should see the "Save Now" option in the top right corner just above the stage.

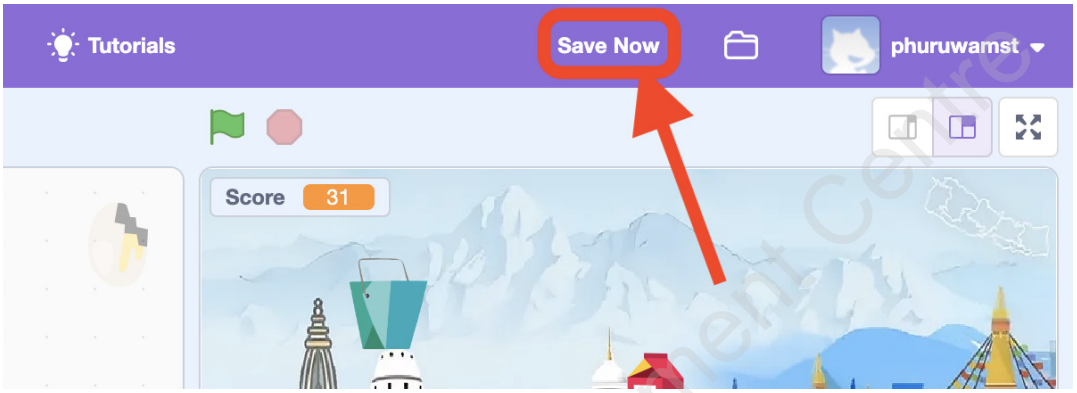



*Figure 3.40*

If you are using Scratch offline, simply click on "File" and save your project as ".sb3" extension for future use.

**Step 6:** Now that everything is completed, click on the and start playing.

***Activity: 3.1***

As we come to the end of the concept of scratch, you may have ideas for different games. But for now let's try to build a game, keeping the name of the game "Catch the fruit".

**Game Description:** Create a game where you control a basket to catch falling fruits for points. Use motion blocks to move the basket left and right with arrow keys. Use sprites for fruits falling from top to bottom, and event blocks for random appearances. Display points using a variable and sensing blocks. Make it challenging with speed variations and sound effects for catching or missing fruits.

## 3.4 Concept of Micro Bit

The micro bit is also stylized as micro: bit is a small device, not bigger than your calculator, that you can program and create interactive projects like games,

flashlights, robots, Tihar lights, etc. It is a tiny computer designed for learning how programming and hardware components can work together to create interactive projects. The way of creating such projects to interact with the world is also referred to as physical computing.

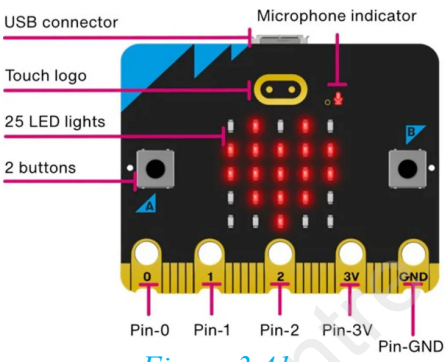


*Figure 3.41*

## History of Micro:bit

> ***Do You Know?***
>
> Micro:bit is also called BBC Micro:bit and it was developed aiming to encourage young people to be creative digitally and upgrade their programming skills.

Micro:bit was developed as a part of an educational initiative in the United Kingdom by the BBC (British Broadcasting Corporation) in 2015. This project was a collective work of several organizations, including the BBC, ARM, Microsoft, and several educational partners. The micro:bit was designed to be a simple and affordable device that could introduce students to coding and electronics in a practical, engaging and motivating way.

## Components of Micro:bit

**The components that makes physical computing possible are as follows:**

a) Microcontroller: It is a tiny computer that can program the micro:bit board to perform specific tasks.

b) 5x5 LED matrix: It has a grid of 25 lights that you can use to display text, numbers, images, or animations.

c) Sensors: It has light and temperature sensors that measure light and temperature around the micro:bit board.

d) Physical computing: Physical computing is about making systems that interact with the real world. It combines hardware (like sensors) with software to create interactive experiences, such as robotics or wearable tech (smartwatches, fitness trackers, augmented reality glasses, and health monitoring devices).

e) Buttons: It has two buttons that you can use to trigger your code when pressed.

f) Microphone: It can send and detect sounds or noises.

g) Pin connector: It has a set of holes that you can use to connect wires, or other components to the micro:bit.

***Let's answer!***

Microcontroller is a tiny _____.

a) Sensor (b) computer (c) software (d) All of the above


*Figure 3.42*

## Applications of Micro:bit

There are many applications of micro:bit in various fields. Some examples are:

### Education

Micro:bit is used as an educational tool to provide hands-on coding and electronics experience. This small device consists of sensors, LEDs, and wireless capabilities, which makes it ideal for students to create interactive projects, from games to cool gadgets solutions.

### Fitness

Micro:bit can also be used to make fitness gadgets. Due to its small design and programmable features, micro:bit can be used to create interesting fitness projects that promote movement and exercise. Some examples of fitness related projects using micro:bit are step counter, fitness game, workout tracker, etc.

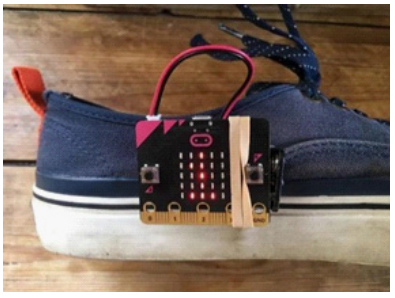


*Figure 3.43*

### Games

Micro:bit is also an ideal platform to make games. It allows users to code and create their own games using different built-in features, such as buttons, sensors, and LEDs. Some examples of games are rock paper scissors, maze and dice simulator.

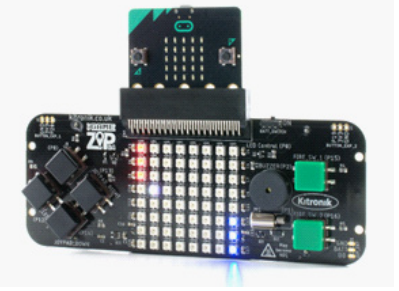


*Figure 3.44*

## Fashion

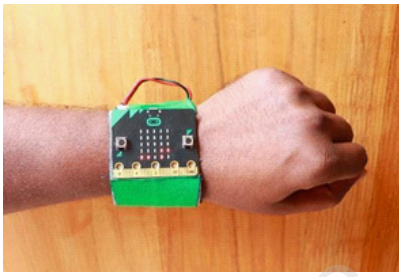


Figure 3.45

Micro:bit is also used in the world of fashion, with a unique combination of technology and style. This small programmable device allows fashion designers to put interactive elements into garments, accessories, and wearables with the help of built-in sensors, LEDs, and wireless capabilities.

## Music

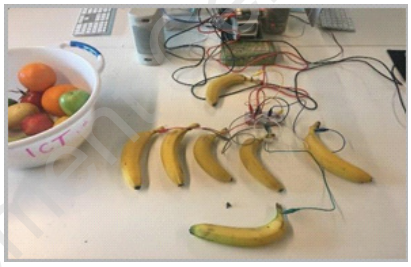


Figure 3.46

Micro:bit can be used to create interactive musical instruments, compose electronic beats, and experiment with different sounds. It has a built-in speaker to play sounds. You can find projects like a funny voice recorder, banana piano and jukebox on micro:bit's official website.

## Cooking

Figure 3.47

Micro:bit is not specifically designed for cooking applications but its features can be creatively used to enhance certain aspects of the cooking experience. For example, a digital egg timer, a temperature sensor to check cooking temperatures, or you can make it to display recipes for foods.

## Home and Garden

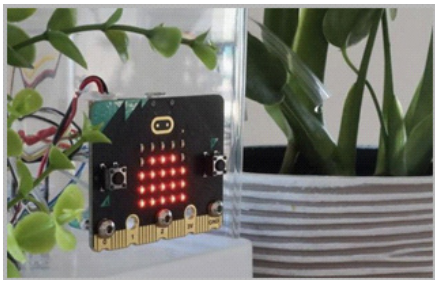


Figure 3.48

Micro:bit can be used to create smart home solutions, such as temperature and humidity monitors, interactive lighting systems, and much more. Similarly, micro:bit can also be used for smart gardening by monitoring the moisture of soil, exposure to sunlight, or even automating watering systems.

***Do you know?***

We can also control micro:bit devices through Scratch programming. In Scratch, there is an extension tab, where you can select "micro:bit" and get a new block named "micro:bit" on the blocks palette to create physical projects.

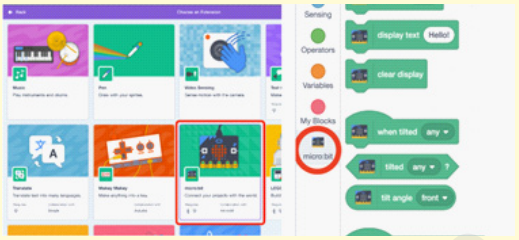



*Figure 3.49*

## 3.5. Concept of Arduino UNO

While micro: bit is more suitable for beginners and educational purposes, Arduino UNO is preferred for a wider range of projects such as LED flashlights, robotic vehicles, temperature sensors, etc. It is one of the most common and widely used Arduino boards.

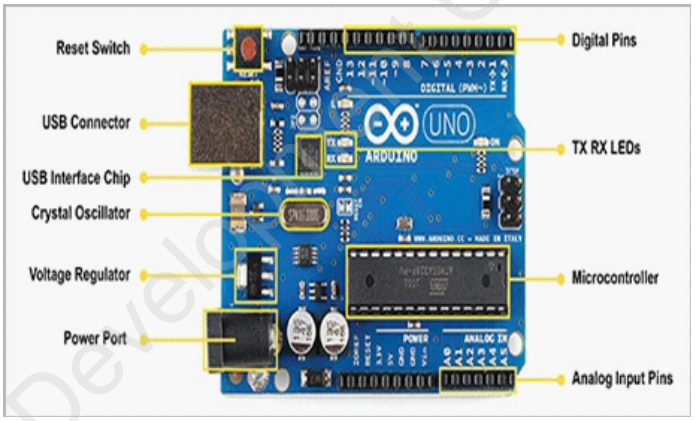



*Figure 3.50*

**Fun fact!**

The UNO board got its name from the Italian and Spanish word "uno," meaning "one." It's called UNO because it's the first and most basic Arduino board.

### History of Arduino

The Arduino UNO platform was developed by two engineers, David Cuartielles and Massimo Banzi in 2010. Their motive was to provide a tool, specifically for students, to help them learn programming with the Arduino Uno microcontroller and enhance their electronic skills at the same time. Their goal was to encourage students to apply their knowledge in real-world cases and help develop a practical understanding of programming and electronics.

## Types of components in Arduino UNO

**The components of Arduino UNO can be categorized in two parts: hardware and software.**

### a) Hardware

Arduino UNO has a microcontroller called "ATmega328P", which is like the "brain" of this Arduino board. It also consists of 14 digital pins, 6 analog pins, a USB port, a power jack, a reset button, a built-in LED, a crystal oscillator, and a voltage regulator.

### b) Software

Arduino UNO has a special software called Arduino IDE. This software helps you write codes for your Arduino and sends it to the board. You can also add extra features using ready-made sets of code called libraries.

**Do you know?**

Tinkercad is a free 3d modeling program where we can make virtual projects with Arduino and simulate the result. While coding, we can either use other programming languages like C, Python or we can even code blocks just like scratch and program Arduino.

## Applications of Arduino UNO

The Arduino UNO is versatile and easy to use. It means that an Arduino UNO can be used in various fields. Here are some common applications of the Arduino Uno:

### a. Education

Arduino Uno is widely used for educational purposes. It has a user-friendly interface and is used to teach both electronics and programming together.

### b. Home

Arduino Uno can be used in home automation projects, such as controlling lights, checking the temperature, and many other appliances.

*Figure 3.51*

### c. Robotics

Arduino Uno is popular for creating and testing robotic devices. It can be used to control motors, sensors, and other robotic components.

### d. Wearable technology

Arduino Uno is a small device, which makes it suitable for creating wearable technology devices. You can program an Arduino to make devices like digital watches, step counters, etc.

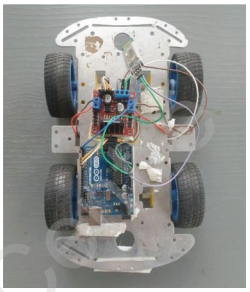


*Figure 3.52*

### e. Data gathering

Arduino Uno can also be used as a source of collecting data with the help of various sensors. This makes it suitable for monitoring the environment, motion detection, GPS tracking, etc.

### f. Gaming

Arduino Uno can be used to create simple games or gaming devices. It combines programming and electronics concepts in a fun and interactive way.

### g. Audio projects

Arduino Uno can also be used in creating audio-related projects, such as music controllers, sound installations, or walking piano.

*Figure 3.51*

### h. Testing product

Engineers and developers also use Arduino Uno to test electronic projects before making complex hardware.

***Activity: 3.2***

Build a smart dustbin designed to automatically open its lid when a user approaches with trash. To build the project you need Arduino's and other different components, collect all the components necessary which are necessary to build the project. You can take the help of your teacher to collect the components.

## Exercises

### 1. Write the full forms of the given abbreviations:

a. MIT b. LED c. IDE d. UNO e. IDE

### 2. Choose the best answer.

1. To make your program generate results, you can use event block called _____.

   (a) when clicked (b) when the sprite clicked

   (c) both (a) and (b) (d) none of the above

2. The background image in the stage area is called _____.

   (a) background (b) wallpaper

   (c) backdrop (d) design

3. Which type of block in Scratch is used to repeat actions?

   a) Motion Block b) Control Block

   c) Looks Block d) Sensing Block

4. What is the primary use of "sensing blocks" in Scratch?

   a) To control sprite movements

   b) To detect events like key presses or mouse clicks

   c) To display messages on the stage

   d) To add sound effects to a project

5. There are ____ extensions in Scratch programming.

   (a) 5 (b) 9 (c) 11 (d) 12

6. "ask What's your name? and wait" is an example of _____ block

   (a) sensing (b) control (c) operator (d) looks

7. Which block in Scratch for micro programming is used to display text or patterns on the LED matrix?

   a) Show LEDs Block  b) Display Block

   c) Animation Block  d) Matrix Block

8. What are devices like Micro:bit and Arduino used for?

   (a) Physical computing

   (b) Combining programming with physical devices

   (c) Creating interactive projects

   (d) All of the above

## 3. Write short answers to these questions.

a. What is scratch?

b. List down the components of Scratch interface.

c. What is a micro: bit? Give any two applications of micro: bit.

d. How does a micro: bit help in physical computing?

e. What is ATmega328P?

## 4. Write long answers to these questions.

a. Define block programming along with its features.

b. What are the applications of block programming? Explain in brief.

c. How is block programming different from traditional text-based programming?

d. What are blocks in block programming? Describe the block components of Scratch programming language.

e. What is an Arduino UNO? Write down its types of components.

f. What is a microcontroller? What is it used for?

g. You are given a program where the cat sprite rotates 90° and says, "Namaste" when the sprite is clicked. List and explain the block components used in this program.

## Lab Activities (Practical)

1. Create simple games using different blocks of Scratch.

   **a) MAZE game**

   Game description: In this game, you control a sprite that moves through a maze to reach a goal. You have to avoid touching the walls of the maze, or you have to start over.

   - Draw or use a sprite from the library and draw a backdrop that looks like a maze using the paint editor. You can also use the looks block to change the color or brightness of the maze.

   - Use the motion blocks to make it move when the arrow keys are pressed. You can also use the sensing blocks to keep the sprite inside the maze, and to detect when the sprite touches the walls or the goal.

   - To make the game more challenging, you can use the control blocks to add a timer, and the variable blocks to show it on the stage. You can also use the sensing blocks to detect when the timer runs out, and the control blocks to stop the game. You can also use the sound blocks to play sounds when the sprite moves, touches the walls, or reaches the goal.

2. **Demonstrate online free coding simulation tools and its working mechanism.**

   - For Micro: bit,

   a) Go to https://makecode.microbit.org and select the Micro:bit.

   b) Show the code editor, where you can either write codes or drag and drop blocks.

   c) Show Micro:bit's physical parts like LED, pins, etc. in the simulator.

   d) Make a basic program using blocks, and run the simulator.

   e) Show how the program would work on a physical micro:bit.

- **For Arduino UNO,**

a) Go to https://www.tinkercad.com/circuits.

b) Show the code editor and simulator.

c) Show Arduino UNO's physical parts like breadboard, LEDs, resistors, etc. in the simulator.

d) Write a basic Arduino code like RGB traffic lights, and show how the circuit works.

## *Project works*

**1. Develop an individual project using the Scratch tool.**

a) Think of different ideas for your project. It has to be a game project, such as jump game, flappy bird, snake game.

b) Make sure you try to apply all the block components that you have learnt in the chapter.

c) Save your work from time-to-time, so that you do not lose your work.

d) Once you complete your project, publish it online in your Scratch account. If using offline, download your work and share the "<filename>.sb3" with your teacher.

Unit 4

# Web Technology

## 4.1 Concept of Web technology

***Let's think!***

a. What is the web, and what does it mean to you?

b. How can IT help students develop skills?

c. What features would you add to a school or community website?

In today's digital age, the internet has become an integral part of our lives, providing us with access to vast amounts of information, communication, and entertainment. Websites are the building blocks of the internet, serving as platforms for sharing information, conducting business, and connecting people across the globe. Understanding web technology is essential for anyone looking to create, manage, or navigate the online world effectively.

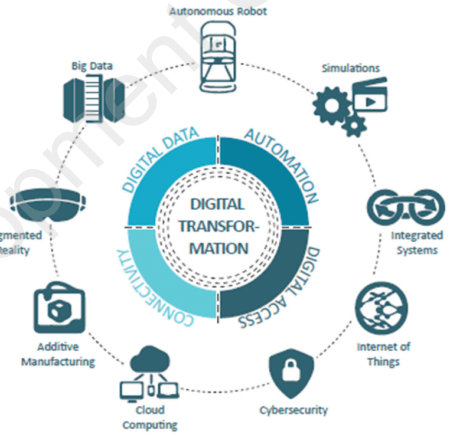



*Figure 4.1*

The word "web" is about how everything is connected online—information, people, and devices. And when we say "technology," we mean the tools, computer programs, and languages that make this connected web work and create our experiences on the internet.

"Web technology refers to the tools, software, protocols, and languages that individuals, businesses, and organizations use to create, develop, and manage applications and content on the Internet."

It encompasses a broad range of elements from fundamental building blocks like HTML (Hypertext Markup Language) for structuring content and CSS (Cascading Style Sheets) for presentation and design to dynamic scripting languages such as JavaScript for interactivity. Server-side technologies like PHP, Python, and Node. js handle data processing while web development frameworks like React, Angular,

and Vue.js provide structured approaches to building powerful applications.


*Figure 4.2*

The landscape of web technology also includes considerations for web security, responsive design for various devices, and adherence to web standards, ensuring a consistent and secure online experience for the user. Overall, web technology is the backbone of our digital interactions, shaping the way we access information, communicate, and engage with the vast digital ecosystem of the internet.

**Some examples of web technologies are:**

a. **HTML:** Hypertext Markup Language, the standard language for creating web pages

b. **CSS:** Cascading Style Sheets, the language for styling and formatting web pages

c. **JavaScript:** A scripting language for adding functionality, interactivity, and logic to web pages.

d. **PHP:** A server-side scripting language for creating dynamic web pages and web applications.

e. **Python:** A versatile programming language commonly used for web development, both on the server side and in frameworks like Django

f. **MySQL:** A database management system for storing and retrieving data for web applications.

g. **WordPress:** A web content management system for creating and managing websites and blogs.

The future of web technology is like an exciting journey that will change how we do things online. It is full of creative ideas that will make our digital experiences even better.

***Let's discuss!***

Discuss the future of technology in the class. Think about the devices and gadgets you use every day. How do you think they'll be changed and improved in the coming years? How will these changes affect our lives? Let's share our thoughts and ideas with each other.

## 4.2 Web Development Life Cycle

The web development life cycle is a systematic process of creating and maintaining a website. It is a planned method that provides a fast and easy development, improvement, and maintenance of a website while taking user experience, functionality, and adaptability into consideration. Thus, it provides a roadmap for web developers, designers, and stakeholders to collaborate and deliver successful online solutions.


*Figure 4.3*

a. AI and Machine Learning Integration: Smart websites are being developed that will use AI and machine learning to give users personalized content along with suggestions and do things automatically.

a. Progressive Web Apps (PWAs): - Progressive web apps are the websites that work like an app directly in web browsers. They work even when you're offline and can send you notifications.

b. Augmented Reality (AR) and Virtual Reality (VR): - Websites using AR and VR features are being developed. Such websites will provide us with experiences that feel real and let us interact with web content in exciting ways.

c. Responsiveness: Websites will keep getting updated making them a lot more responsive on different devices over time.

d. Edge computing: Edge computing will enable computers to do their work close to where the information is created which will make the performance of web applications much faster and more efficient.

e. Web security: Shortly, websites will get smarter about keeping our data and information safe from online threats. Web technologies will evolve to provide

advanced security measures such as stronger security codes, better ways to check who you are, and breach prevention.

***Did you know?***

The first website went live on August 6, 1991.

## Stages of WDLC

1. **Gathering information:** In this stage, important information like target audience data, content requirements, and website structure are gathered.

2. **Planning:** In this stage, a basic layout and design for the website are created, describing its general structure.

3. **Design and layout:** This stage involves selecting fonts, colors, and images to give the website a look that goes well with its content or brand.

4. **Content creation:** In this stage, the actual content for the website is created using text, images, audio, and videos.

5. **Development:** In this stage, developers use code to convert the design and content of websites into a useful platform.

6. **Testing:** It is the process of finding errors in the website. The website functionality is tested, and errors are fixed if found.

7. **Maintenance and updating:** A website requires constant maintenance and changes to keep the content up-to-date and maintain smooth functionality.

**Let's explore!**

Make a group in class and research the topic of web development life cycle methodologies, and why they are important. Can you name some common methodologies used in web development? How do these methodologies differ in their approach? Give examples of companies or projects that have successfully implemented these methodologies. All researchers give the presentation in class and share their chosen methodology.

## DNS (Domain Name System)

In simple terms, the DNS (Domain Name System) is a system that gives a unique

and suitable name to a website. The DNS translates the name of a website to the IP (Internet Protocol) address that a computer can understand. When the name of a website is entered, the computer sends a request to the DNS server for the IP address of the website's server. There are many types of Domain Name. Some of them are listed below:

1. .com (Commercial)
2. . internet (Network)
3. .np (Country Code Top-Level Domain for Nepal)
4. .edu (Education)
5. .gov (Government)
6. .org (Organization

***Activity 4.1:***

We have learned about the different types of domains. Now, explore the website of the government side of Nepal. Your task is to conduct research and gather information about various government sectors of Nepal. You can choose to focus on areas like education, health, transportation, or any other sector that interests you. Once you've gathered the information, fill out the below table and discuss it in the classroom.

| Department Name | Link of the website | Why is it important? |
|---|---|---|
| Inland Revenue Department | https://taxpayerportal.ird.gov.np/taxpayer/app.html | It helps in the payment of tax to the government |

## DNS registration process

A website needs to go through the DNS registration process to have a globally accessible name. A Domain Name Server translates identifiable names, such as domain names, into numeric IP addresses. We can register through many different registrars like nestnepal.com, prabhuhost.com, and gurkhahost.com. The steps for registration of a domain name are given below:

a. **Choose the domain name:** Choose a domain name that is unique and memorable that identifies your website.

b. **Check the availability:** Verify the availability of the chosen domain using the registrar's domain checker tool.

c. **Choose a domain name registrar:** Choose a trusted registrar to officially register your chosen domain name.

d. **Buy and register your domain name:** Once availability has been verified and a registrar has been chosen, buy and register your domain name.

e. **Do not lose your domain name:** Renew the domain registration before it expires so that you don't lose your domain name.

> ***Did you know?***
>
> We can freely register the .np domain in Nepal. https://register.com.np/

## 4.3 Concept of UI/UX

To understand the concept of UI/UX, we must understand the front end and back end, which are the foundations of every digital creation.

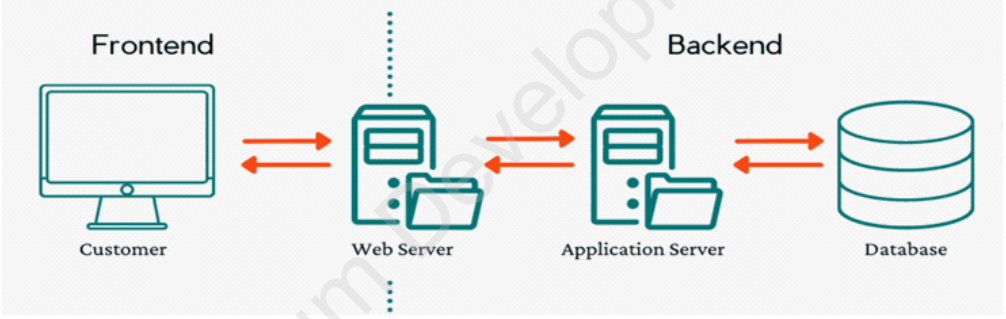



*Figure: 4.4: - concept of UI/UX design.*

### Front End:

Imagine building a house: the front end is like its sleek exterior. But what about the inner workings? That's the back end, hidden from view yet essential for functionality. It's the complex network of wiring, plumbing, and structural support that ensures the house stands strong. Just as the front end catches the eye, the back-end powers everything behind the scenes, making the house - or in the digital world,

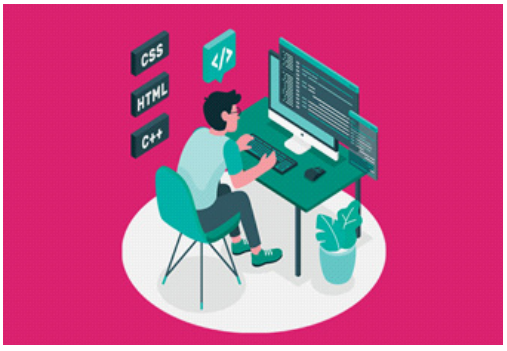



*Figure 4.5*

the website functional and livable. The part of a software application or website that users interact with directly is known as the front end. The front end is what you see on your screen like the buttons you click, the text you read, the images you view, and the overall layout. The front end helps in creating a positive and engaging user experience. The UI/UX is also a part of the front-end design.

## Back End

Back End is the part of a software application or website that is responsible for handling operations that are not directly visible to users. The back end does all the important tasks that keep the application or website running smoothly. The backend contains data, operating syntax, functions, and logic of the application.

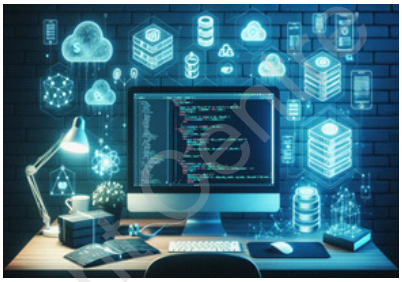


*Figure 4.6*

***Activities 4.2:***

Think of it as a treasure hunt on the internet. Your job is to find out about the tools that help build websites. First, look for the tools that make websites look good, like colors and shapes. Then, find out what tools are used to keep the website running on the server. When you find these tools, fill out the table below, tell your friends about them, and talk about why they are important.

| Tools to make website look good | Tools that help the website keep running |
|---|---|
| | |

## UI (User Interface):

UI stands for User Interface. The point of interaction between a user and a digital device or software application is known as a User Interface. It focuses on a product's visual, interactive elements to create interfaces that are visually appealing, user-friendly, and easy to navigate.

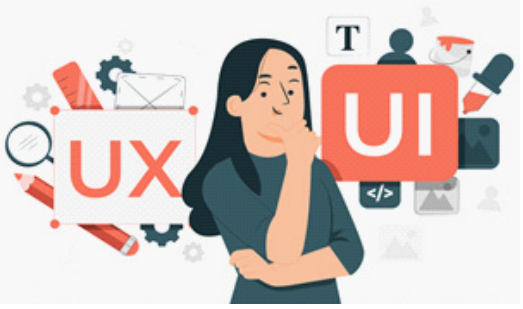



*Figure 4.7*

## UX (User Experience):

UX stands for User Experience. The overall experience of a user when interacting with a product, service, or system is known as User Experience (UX). It focuses on understanding the user's needs, behaviors, and preferences to enhance user satisfaction.

*Figure 4.8*

## Advantages of UI/UX design

a. It increases user engagement with attractive designs.

b. It enhances user experience.

c. It helps in designing a unique look for branding.

d. It helps in making websites and apps easy to use and navigate.

e. It saves development time and cost.

In this way, the UI helps in designing the visual elements such as the buttons, colors, and layouts you see whereas UX helps in making sure that using the website is easy and enjoyable. UI and UX work together to transform the front end into an interactive and user-friendly experience.

## Software used for UI/UX design

**The following software can be used for UI/UX design:**

a. **Figma:** It is used for collaborative design where multiple people can work on the same project simultaneously.

b. **Adobe XD:** It is used for creating different interfaces, buttons, and layouts.

c. **Balsamiq:** It is a software used for creating wireframes.

**Question**

i. Why is UI/UX important in web technologies?

ii. Who will create the UI/UX design in web development?

iii. In which stage of WDLC, UI/UX design comes under?

***Let's get more information!***

Back in the 1960s, a clever computer scientist named **Douglas Engelbart** had a vision. He imagined a way for people to interact with computers using pictures, buttons, and menus. This idea became the foundation for what we now call UI design. Think of it as creating the look and feel of software— just like arranging furniture in a room.

## Concept of wireframe

A wireframe is an outline or a skeleton of a UI, showing the basic structure and layout of a web page or application. Wireframes are used in the early stage of the design process to outline the placement of elements without getting into details of visual design or color. We can use different software such as Sketch, Figma, and Balsamiq to design wireframes.

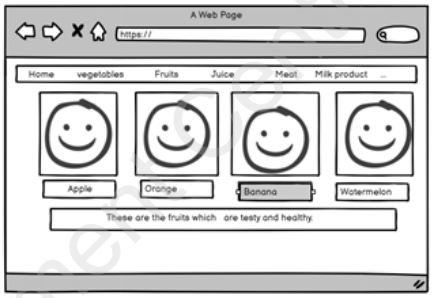



*Figure 4.9 Wireframe*

**Let's check these questions!**

a. Why is wire frame important in web development?

b. In which stage, Wireframe does come under?

c. Who will create the wireframe in the software development?

## Concept of wireframe design

Wireframe design is the process of creating a simplified representation of a user interface (UI) or webpage. Here are the key aspects of wireframe design:

1. **Structural framework:** The structural framework refers to the basic layout and organization of a web page.
2. **User flow and navigation:** A user flow is a visual representation of how a user navigates and completes tasks within a digital product while navigation refers to the process of moving through and interacting with a digital interface.
3. **Quick interaction:** Wireframes are less detailed than the final designs. So, they allow for quick repetition and changes.
4. **Prototyping:** Prototyping is the process of creating an opening model or sample of a product, system, or application to test and confirm the design concepts

before full development. Wireframes can be used to generate interactive prototypes.

5. **Cost-effective exploration:** A wireframe helps in finding problems early in the design process to avoid costly mistakes later on.

### Benefits of wireframe

a. **Easy to draw:** Wireframe has a simple and clean layout and consists of simple screen elements.

b. **Easy to understand:** Wireframe is focused on the essential elements and structure without any visual details or decoration.

c. **Easy to modify:** You do not need to have an idea of any programming language to make changes to a wireframe.

d. **No coding required:** You can draw the wireframe as if you are using a drawing tool.

## 4.4 HTML

### Introduction to HTML

Hypertext Markup Language (HTML) is a language used to build web pages. Hypertext is a text which contains links to other texts and the term markup refers to the collection of symbols or codes in a document that provides instructions on how the document should be formatted and processed. Thus, a markup language is a way to add notes, comments, or marks to a document, text, or other piece of data to a webpage.

***Let's get more information!***

- DHTML (or Dynamic HTML) and XML (or Extensible mark-up language) are the other popular mark-up languages.
- There are many specialized software packages like Dreamweaver, frontpage, coffee cup, etc., which are used to create web documents.

HTML is a computer language that uses predefined tags to define the structure of web documents which is the industry standard computer language for creating the skeleton of web pages. The markup has no direct effect on how a computer

operates, but it does help to structure and present the content. Tim Berners-Lee created HTML in 1990 at CERN, the European Particle Physics Laboratory in Geneva, Switzerland. The World Wide Web Consortium (W3C) is the organization that controls the development of HTML. Without HTML, the World Wide Web would not exist.

HTML was developed by **Tim Berners-Lee** in **1991**. It was first released publicly in **1993** and became the standard language for creating web pages. HTML 2.0 was published in **1995**, which included more features and remained the standard until **19972**. The development of HTML has been ongoing, with HTML5 being the latest major version.

To create HTML documents, you can use a simple text editor like Notepad or use special software such as:

1. **Sublime Text:** It is a feature-rich text editor that is both versatile and fast.
2. **Visual Studio Code:** It is a popular software that provides different tools to create different software and websites.
3. **Notepad++:** Notepad++ is a popular text editor that can be used for creating HTML documents.


*Figure 4.10.*

***HTML is commonly used for:***

a. Creating and designing the basic layout of a webpage

b. Creating and formatting the text content on web pages

c. Embedding the images, graphics, sounds, videos on the web

d. Adding tables, forms, and hyperlinks on the web.

*Activity 4.3*

As we already discussed about the different types of text-editor tools, let's get ready to dive into the world of web development by setting up Notepad ++ on your computer now.

## HTML Tags

Each element on a webpage is defined by an HTML code or tag. HTML tags tell browsers how to display data. HTML tags are predefined syntax and commands that perform specific tasks and are enclosed by angular brackets. A tag begins with a (less than) sign and ends with a > (greater than) sign and information on the screen and defines a webpage's appearance, layout, and flow. A tag consists of three parts: element (identification of tag), attribute, and value. HTML tags are **not case sensitive** which means fonts written within the tag can be both uppercase and lowercase. HTML tags are of two types:

### a. Container or Paired Tag

A container or paired tag is a type of HTML tag that has both an opening and a closing tag, such as  and . The opening tag starts with a less than sign (<) such as  tag and the closing tag starts with a slash and a less than sign (</) such as . The opening tag activates the implementation of the tag while the closing tag stops it.

### b. Empty or Unpaired Tag

The tag which does not have its corresponding closing tag is known as an Empty or Unpaired tag. It has only the opening tag, but no closing tag. It is used to perform specific actions, functions, or tasks on the webpage. An empty tag is also called a self-closing tag or stand-alone tag. For example, <br> inserts a blank line and <img> tag inserts an image in a web page. Both do not have a closing tag.

*Activities 4.4*

Discuss in the classroom and list down some of the container or paired tag and empty or unpaired tag.

| SN | Paired tags | Uses | Unpaired tags | Uses |
|---|---|---|---|---|
| 1 | | | | |

## HTML Attributes

In HTML, attributes are special words used in the opening tags to control or modify the behavior of the tags. For example, you can specify the FONT (size, color, or face) by including the appropriate attribute with the HTML code.

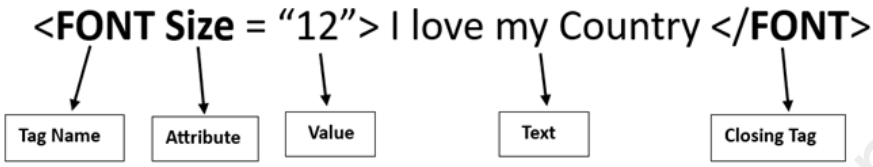



*Figure 4.11 HTML Attribute.*

## Structure of HTML

Every language has a certain structure for proper and smooth functioning. Likewise, HTML has its own standard structure. While creating a web page, HTM be nested inside L elements can other elements to build a hierarchical structure.

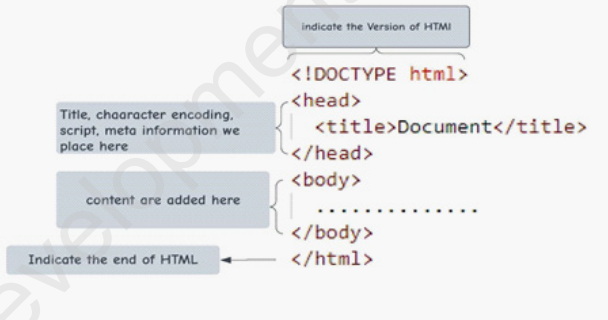



*Figure 4.11*

The basic HTML structure consists of four main parts: the doctype declaration, the <html> elements, the <head> element, and the <body> element.

***Lab task***

Imagine you're creating a web page to showcase "mySecondTeacher" in a big and bold way. Can you write some HTML code to make it happen? Once you're done, don't forget to run it to see your creation come to live on the screen.

## Creating HTML document in Sublime Text

To create an HTML file, all you need is a basic text editor. Different operating systems have their own text editors: **SimpleText** for Mac OS, **Vim and Emacs** for UNIX/Linux, and Notepad for Windows operating system. Additionally, there are HTML development tools like PHPEdit, Textpad, Editplus, and Notepad++, Sublime text, VS Code etc. HTML files typically have extensions like **.html** or **.htm.**

Sublime Text is a program for writing and editing code on your computer. Sublime Text has features like syntax highlighting, which colors different parts of your code to make it easier to read, autocomplete, which helps you complete your code faster, and customization options, which let you change how the program looks and works.

Here are the steps to create and save an HTML document in **Sublime Text Editor**:

1. Open Sublime Text Editor on your Computer
   (To download Sublime Text Editor,

   Go to: https://www.sublimetext.com/download )

2. Go to the File Menu and click on the Save option.

3. Sublime Text will display save as a dialogue box asking for a file name as shown in the figure below.

4. In the file name text box, type any name followed by .html or .htm extension and choose the drive location where you want to save the file.

5. Click the Save button.

6. Now you are ready to write HTML code in Sublime text.

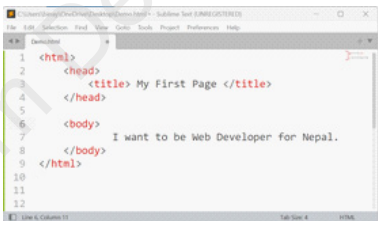



*Figure: 4.13*

To view your web page, double click on your file and you can see the output of the HTML document in your default browser.

## Organizing Text in HTML document

### Comment Tag

The comment tag is used to insert a comment in the HTML source code. It is a way for developers to include notes or remarks within the HTML code for various purposes, such as explanations, reminders, or temporary instructions.

< ! - - Your Comment Here…... - - >

The comment tag requires an exclamation mark (!) after the opening bracket.

***Example 1:***

```
<html>
    <head>
        <title> Example of Comment  </title>
    </head>

    <body>
        <!-- Body section of HTML contain all the informations of Web page -->
            There are two types of comments
            1. Single line Comment <br>
            2. Multiline comment
    </body>
</html>
```

*Figure: 4.14: - Example of Single-line comment*

The **<br> tag** in HTML is a special code that tells the webpage to start a new line. It is like pressing the enter key on your keyboard. You don't need to write anything after the <br> tag, because it is a **self-closing tag**.

The web browser only identifies a single pace between the words or texts even if there are many spaces given between words. To add additional spaces between the text you can use  . Using single   is equal to one space, double ** ** provides double space and so on.

**Paragraph Tag <P>**: It is used **to create paragraphs** in an html document. A paragraph is a group of sentences about a specific topic. The <p> tag helps you arrange your text on a webpage. It puts the sentences that belong together in one box. When you use the <p> tag, you're saying, "This is a paragraph."

**ALIGN** attributes with values **"LEFT"**, **"CENTER"** or **"RIGHT"** can be used to set the alignment of the paragraph. By default, the alignment of a paragraph is left.

### *Lab work!*

As we already learned about the structure of HTML, you are assigned to write the HTML structure, comment on each line in the structure, and submit it to the teacher. After the completion of the task discuss in the class about: -

a. Why are comments important in Coding?

b. What's the difference between single-line and multi-line comments?

## Text Formatting Tags

A tag is a command in a web page that performs specific actions and tells the browser to do. After we include the structure tags, we are ready to start placing basic content in the document body. To work with text appearing in a web browser, we use text formatting tags. Text formatting tags are special-purpose tags that are used to change, define, or enhance the visual appearance and style of content on a web page. Using text formatting tags, we can make our web text content bold, italic, etc.

**Heading Tags (<H1> …. <H6>)**: These tags are used to provide headings in an HTML document. HTML supports six different levels of headings. The biggest header format is <H1> and the smallest is <H6>. All the styles of headings tags appear in **BOLDFACE** and the size of the heading depends on the level chosen, i.e., <H1> to <H6>

| HTML Input | Output: |
|---|---|
| <h1> This is heading one. </ h1> | This is heading one. |
| <h2> This is heading two. </ h2> | This is heading two. |
| <h3> This is heading three. </ h3> | This is heading three. |
| <h4> This is heading four. </ h4> | This is heading four. |
| <h5> This is heading five. </ h5> | This is heading five. |
| <h6> This is heading six. </ h6> | This is heading six. |

Horizontal Lines <HR>: It is used to draw horizontal lines or rulers to separate contents in an HTML page. It draws a line across the page from left to right and is a self-closing tag. The attributes of the <HR> tag is:

| Attributes | Description |
|---|---|
| ALIGN | It is used to align the line on the browser screen (aligned to the center by default).<br>**ALIGN=LEFT** will align the line to the left of the screen.<br>**ALIGN=RIGHT** will align the line to the right of the screen.<br>**ALIGN=CENTER** will align the line to the center of the screen. |

| | |
|---|---|
| SIZE | It is used to change the size of the line. |
| WIDTH | It is used to set the width of the line. It can be set to a fixed number of pixels, or a percentage of the available screen width. |
| COLOR | It is used to change the color of the line. |
| | |

## HTML Input

<HR ALIGN = " LEFT " WIDTH= "25" SIZE = " 3 " color = " Blue ">

Nepal is the Birthplace of Lord Buddha.

## Text styles

You can use these tags to change how your text looks. <B>…</B> makes it bold, <I>… </ I> makes it italic, <U>…</ U> makes it underline. The <SUP>…</SUP> and <SUB>…</SUB> tags are used for superscript and subscript, respectively. For example, H <SUB> 2 </SUB> O and E = mc <SUP> 2 </SUP>.

***HTML Input***

 This text is bold. 

 This text is italicized. 

<u> This text is underlined. </u>

A<sup> 2 </sup> B <sup> 2 </sup>

H <sub> 2 </sub> O

**Output:**

This text is bold.

This text is italicized.

This text is underlined.

$A^2B^2$

$H_2O$

< FONT > Tag: Controlling Font Size, Color and Face

All text specified within the tags<FONT> and </FONT> will appear in the font face, size and color as specified in the attributes of the <FONT> tag. The attributes are:

| | |
|---|---|
| **FACE** | Sets the font to the specified font name |
| **SIZE** | Sets the size of the text (The default size of the font is 3) |
| **COLOR** | Sets the color of the text. Colors can be set by their name or using hexadecimal codes. |

### *HTML Input*

<FONT face = "Times New Roman" Size = "5" color = "Red"> Talent is not what you have, talent is what you do. </FONT>

You can use color codes to choose the color of your text or background on your web page. RGB color codes are special codes that have six numbers or letters. The RGB color codes tell the computer how much red, green, and blue should be in a color.

**Here are some of the basic color codes:**

| | | | |
|---|---|---|---|
| Black | #000000 | White | #FFFFFF |
| Red | #FF0000 | Green | #00CC00 |
| Blue | #0000FF | Yellow | #FFFFF00 |
| Silver | #C0C0C0 | Lime | #00FF00 |
| Gray | #808080 | Maroon | #800000 |
| Purple | #800080 | Aqua | #00FFFF |

### HTML Input

< Font color = " #800000 "> Maroon </Font>

< Font color = " #808080 "> Gray </Font>

< Font color = " #00CC00 "> Green </Font>

### The Marquee Tag:

The <marquee> tag in HTML makes your text, image, graphics, and videos move or scroll on your web page.

| Attribute | Description | Values |
|---|---|---|
| Behavior | Scrolling behavior | Alternate, slide, scroll |
| Direction | Scrolling direction | Left, right, up, down |
| Bgcolor | Background color | Color name or value |
| scrolldelay | Delay in scrolling | A number in millisecond |
| Height | Height of scroll area | A number in pixels |
| width | Width of scroll area | A number in pixels |

### HTML Input

<MARQUEE> I will be a web developer. </MARQUEE>

<marquee behavior = “scroll” direction = “down” bgcolor = “ #CCFF00 “

scrolldelay = “1000”> I am Marquee Running Text! </ marquee>

### *Let’s have Quiz!*

**Which tag is used to create a scrolling text or image in HTML?**

a. <scroll>
b. <animate>
c. <marquee>
d. <slide>

**Which attribute in HTML is used to horizontally align text or images within an element?**

a. Align
b. Center
c. Horizontal-align
d. Text-align

***Activity:4.5***

Make a marquee to display the future of learning, mySecondTeacher.

## 4.4.5 Anchor, List, Table, Image Tags and Their Properties

### 1. Anchor Tag

The anchor tag is an HTML element used to create hyperlinks on web pages. Hyperlinks are used to direct users to different sections of the same webpage, another webpage or various types of data such as files, images, or documents. The anchor tag is denoted by the <a> element.

**The browser distinguishes hyperlinks from the normal text. Hyperlink has the following features:**

a. By default, hyperlinked texts appear blue in color.

b. The hyperlink text/ images are underlined.

c. When the mouse pointer hovers over the hyperlink text, it changes into the shape of the pointing hand.

**Syntax: < a href = " URL/ path of page or document"> text displayed on page </a>**

**Attributes of anchor tag:**

**href:** It specifies the URL of the page or document the link navigates or jumps to.

**target**: Specifies where to open the linked page or document. The values of the target attribute is:

| | |
|---|---|
| **_blank:** | Opens the linked document in a new browser window or tab |
| **_ parent:** | Opens the linked document in the parent frame or window |
| **_ self:** | Opens the linked document in the same frame or window |
| **_ top:** | Opens the linked document in the full body of the window, breaking out of any frames |

**Hyperlinks can be of two types:**

a. **External link:** It links to an external web document of the same or different website.

b. Internal link: It links to the specific location of the same web page.

## External Document References

***Example 2:***

< A HREF = " Link_Check.HTML "> Visit my Home Page </A>

***HTML INPUT***

```
<HTML>
<HEAD>
    <TITLE> Using External Links </TITLE>
</HEAD>
<BODY BGCOLOR = " SkyBlue ">
<H2 ALIGN = "CENTER"> Visit the following links to explore Nepal: </H2>
<P> <A HREF = " http://www.ntb.gov.np"> Nepal Tourism Board Website </A> </P>
<P> <A HREF = " http://www.explorenepal.com"> Explore Nepal </A> </P>
<P> <A HREF = " http://mofa.gov.np "> Ministry of Foreign Affairs Website </A> </P>
</BODY>
</HTML>
```

## Intra-page links or bookmarks

Intra-page links or bookmarks are used to direct the user to a different location on the same page. It is also possible to jump to a particular location on another web page using bookmarks.

## Using Name Anchor

To use intra-page links, a name anchor is used. For instance, <A NAME = "top"> </A>. Here, it is necessary to give a name at the specific point on the page where the tag occurs. The </A> tag must be included, but no text is required between <A> and </A>.

<A HREF = " #TOP "> TOP <A>

The # symbol identifies the word "TOP" as a named anchor point within the current document, rather than a separate page. When a reader clicks on "TOP", the web

browser will display the part of the page starting with the <A NAME = “top”> tag.

***Example 3:***

```
HTML Input
<!DOCTYPE html>
<HTML>
<HEAD>
  <TITLE> Internal Link Example </TITLE>
</HEAD>
<BODY>
<nav>
  <!-- Internal links in the navigation menu -->
  <UL>
    <LI> <A HREF = “ #section1 “> Section 1 </A></LI>
     <LI> <A HREF = “ #section2 “> Section 2</A></LI>
  </UL>
</nav>
<!-- Content for Section 1 - - >
<H2 id = “section1”> Section 1</H2>
<P> This is the content of Section 1.</P>
<!-- Link back to the top of the page - ->
<P> < A HREF = “#top”> Back to Top </A> </P>
<!-- Content for Section 2 →
<H2 id = “section2”> Section 2 </h2>
<P> This is the content of Section 2.</P>
<!-- Link back to the top of the page - - >
<P> <A HREF = “ #top “> Back to Top </A> </P>
</BODY>
</HTML>
```

### Email link

You can use the <A> tag to make links that open an email. When you click on an email link, it starts a new message addressed to the set email. The email address should be written after 'mailto:' in the <A HREF> tag.

***Example 4:***

```
HTML Input
<HTML>
    <HEAD>
        <TITLE> USING EMAIL LINK </TITLE>
    </HEAD>
<BODY>
Send your email to <A HREF = mailto:info@moecdc.gov.np> Curriculum Development Center, Sanothimi Bhaktapur </A>
</BODY>
</HTML>
```

## 2. HTML Lists

A list in HTML is a way of grouping related items in an ordered or unordered form. There are three types of lists: **unordered (with bullet points)**, **ordered (with numbers)**, and **description lists**. Each item in the list is marked with a specific tag, like <li> for unordered and ordered lists, or <dt> and <dd> for description lists.

### Unordered List <UL> (Bullets)

An Unordered list starts with the tag <UL> and ends with </UL>. Each list of items starts with the tag <LI>. The attributes that can be specified with <LI> are:

| | | |
|---|---|---|
| **TYPE** | : | Specifies the type of bullet<br>**TYPE** = FILLAROUND provides a solid round black bullet.<br>**TYPE** = SQUARE provides a solid square blacked bullet.<br>**TYPE** = CIRCLE provides a circle bullet. |

***Example 5:*** **HTML Input**

<B> Some of the Heavenly Places of Nepal </B>

<UL TYPE = "FILLROUND">

<LI> Surma Sarovar </LI>

<LI> Aalital </LI>

<LI> Ramaroshan </LI>

</UL>

**Output:**

Some of the Heavenly Places of Nepal

I. Surma Sarovar

II. Aalital

III. Ramaroshan

**Ordered Lists <OL> (Numbering)**

An ordered list starts with the tag <OL> and ends with </OL>, each list item starts with the tag <LI>. The attributes that can be specified with <LI> are:

| | |
|---|---|
| **TYPE** | Controls the numbering scheme to be used<br>TYPE = " 1 " provides a list in counting numbers (1, 2,....)<br>TYPE = " A " provides a list with UPPERCASE letters (A, B,...)<br>TYPE = " a " provides a list with lowercase letters (a, b,....)<br>TYPE = " I " provides a list with UPPERCASE Roman Numbers (I, II,..)<br>TYPE = " i " provides a list with lowercase Roman Numbers (i, ii, iii,..) |

| | |
|---|---|
| **START** | Alters the numbering sequence (A list can be started with any numeric value.) |
| **VALUES** | Changes the numbering sequence in the middle of an ordered list |

***Example 6:*** **HTML Input**

```
<B> Some of the Heavenly Places of Nepal </B>
<UL TYPE = " 1 "  START = " 3 ">
<LI> Surma Sarovar </LI>
<LI> Aalital </LI>
   <LI> Ramaroshan </LI>
</UL>
```

**Output:**

Some of the Heavenly Places of Nepal

3. Surma Sarovar

4. Aalital

5. Ramaroshan

## Definition Lists

Definition list values appear within tags <DL> and </DL>. It consists of two parts:

| | |
|---|---|
| Definition term | Appears after the tag <DT> |
| Definition description | Appears after the tag <DD> |

***Example 7:*** **HTML Input**

```
  <DL>
  <DT> Front End
<DD> Where user interacts with the system
  <DT> Back End
<DD> Where the logic of the system is written.
 </DL>
```

**Output:**

Front End

Where user interacts with the system

Back End

Where the logic of the system is written

As per the requirement, we can also use Nested List, a list inside another list.

### 3. Adding Images to HTML Document

In HTML, besides text, we can add still or moving pictures to a webpage. The common picture formats are (dot) .gif and (dot) .jpg. Software such as Gif Constructor or Adobe Photoshop can be used to create and save images in these formats. To put an image on a webpage, we use the <img> tag and set the SRC attribute to the image file name (e.g. 'filename.gif' or 'filename.jpg').

Syntax: <IMG SRC = "Path of Image / Source of Image ">

The <img> tag in HTML is used to embed images or pictures into a webpage.

<IMG> tag takes the following attributes:

| Attributes | Value |
|---|---|
| **SRC** | Path or location and name of the image file |
| **ALIGN** | Defines the position of the image: left / centre / right |
| **BORDER** | Defines the size of border of the inserted image |
| **WIDTH** | Defines the width of an image (in pixel value or percentage of the browser's window) |
| **HEIGHT** | Defines the height of an image (in pixel value or percentage of the browser's window) |
| **HSPACE** | Defines a space of specified size to the left and the right of the image |
| **VSPACE** | Defines a space of specified size to the top and the button of the image |
| **ALT** | Defines the alternate text-based information if the user is unable to view the image due to server or networking issues. |

*Example 8:* **HTML Input**

```
<HTML>
<HEAD>
 <TITLE> Education in Nation Building </TITLE>
</HEAD>
<BODY>
 <MARQUEE BGCOLOR = yellow BEHAVIOR = alternate DIRECTION = right  WIDTH = 300> Education Enlightens the Students!!! </MARQUEE>
<BR> <BR>
<HR color = Blue >
<H2 ALIGN = center> Inserting image in HTML Document </H2>
<IMG SRC = " images/myImage.jpg "  HEIGHT = 120  WIDTH = 200  ALT = " Computer Education">
<BR> <BR>
<IMG  SRC  =  "  images / myImage2.jpg "  BORDER = 5>
</BODY>
</HTML>
```

**Note:** While saving images, you can give the full path or save the image file in your HTML document's location.

## 4. HTML Tables

Table is the combination of **ROWs** and **COLUMNs**. Tables are commonly used objects to define the layout of a webpage. A table in HTML starts with <TABLE> tag and closes with </TABLE> tag. Each heading row of a table is defined between **<TH> </TH>** tags. Each row of a table is defined between the **<TR>** and **</TR>** tags. Likewise, each column data of a table is defined between the **<TD>** and **</TD>** tag.

| Sunday | Monday | Tuesday | Wednesday |
|---|---|---|---|
| 5 | 7 | 10 | 21 |
| 11 | 13 | 3 | 4 |

### Header Row:

Each table has a table header. It is defined using <TH> </TH> tag. The content of the table header row is automatically centred and appears in boldface.

Data cells hold data to be displayed in the table. A data row is defined using <TR> </TR> tags. The text matter displayed in a data row is left-justified by default. Any special formatting like boldface or italics is done by including appropriate formatting tags inside the <TR> </TR> tags. An image can also be displayed in a data cell.

***The attributes that can be included in the <TABLE> tag are:***

| Attributes | Description |
|---|---|
| **BORDER** | It defines the border and its thickness of table. The border thickness is specified in pixels. Default size of the border is 0.<br>**Example: <TABLE BORDER = 3 >** |
| | |
| **BGCOLOR** | It defines the background color of the table.<br>**Example: <TABLE BGCOLOR = pink >** |
| **HEIGHT** | It defines the height of the table. Value for height attribute can be defined either in pixel (absolute) or in percentage (relative). If Height is not specified the data cell is adjusted based on the cell data value.<br>**Example: <TABLE HEIGHT = 350 >** |
| **WIDTH** | It defines the width of the table. Value for width attribute can be defined either in pixel (absolute) or in percentage (relative). If width is not specified the data cell is adjusted based on the cell data value.<br>**Example: <TABLE HEIGHT = 200 >** |
| **ALIGN** | It defines the horizontal alignment of the table. The value of Align attribute can set to LEFT, CENTER, RIGHT<br>**Example: <TABLE ALIGN = LEFT>** |

| | |
|---|---|
| **CELLPADDING** | It defines the space between content within the cell and border of the cells.<br>**Example: <TABLE CELLPADDING = 5 >** |
| **CELLSPACING** | It defines the space between the cells of the table.<br>**Example: <TABLE CELLSPACING = 3 >** |
| **COLSPAN** | It lets a table cell occupy or stretch over multiple columns horizontally. It is applied to the <td> or <th> elements inside a table. |
| **ROWSPAN** | It lets a table cell occupy or stretch over multiple rows vertically. It is applied to the <td> or <th> elements inside a table. |
| **BACKGROUND** | It defines the background image of the table.<br>**Example: <TABLE BACKGROUND = "img.jpg">** |

## Caption Tag

A caption for a <table> in HTML is defined using the <caption> tag. The caption is usually shown on top of the table and gives a short title or summary of what the table contains. The <caption> tag is inserted right after the opening <table> tag. The table caption can be made to appear above or below the table structure with the help of the attribute ALIGN as given below:

| | |
|---|---|
| **ALIGN** | It controls the placing of the caption with respect to the table.<br>***Example:*** **ALIGN = BOTTOM** will place the caption immediately below the table.<br>**ALIGN = TOP** will place the caption immediately above the table. |

## VALIGN and ALIGN

The **VALIGN attribute** controls how the content is aligned vertically inside a table cell. It can be used on single cells (<td> or <th>), rows (<tr>), or the whole table.

The **VALIGN** attribute can have these values:

| | |
|---|---|
| **top** | The content is placed at the top of the cell. |
| **middle** | The content is centred vertically in the cell. |
| **bottom** | The content is placed at the bottom of the cell. |

*Example 9:* **VALIGN**

```
<TD VALIGN = "top"> Top-aligned content </TD>
<TD VALIGN = "middle"> Middle-align content </TD>
<TD VALIGN = "bottom"> Bottom-align content </TD>
```

The **ALIGN attribute** controls how the content is aligned horizontally inside a table cell. It can be used on single cells, rows, or the whole table.

***The ALIGN attribute can have these values:***

| | |
|---|---|
| **left** | The content is placed at the left edge of the cell. |
| **center** | The content is centered horizontally in the cell. |
| **right** | The content is placed at the right edge of the cell |

*Example 10:* **ALIGN**

```
<TD ALIGN = "left"> Left-aligned content </TD>
<TD ALIGN = "center"> Center-aligned content </TD>
<TD ALIGN = "right"> Right-aligned content </TD>
```

*Example 11:* **Sample table in HTML**

```
<!DOCTYPE HTML>
<HTML>
<HEAD>
    <TITLE> VISIT NEPAL - FAMOUS PLACES </TITLE>
</HEAD>
<BODY>

<TABLE BORDER = "1">
 <CAPTION> VISIT NEPAL - FAMOUS PLACES </CAPTION>
 <TR>
```

```
    <TH> PLACE </TH>
    <TH> DESCRIPTION </TH>
    <TH> HIGHLIGHTS </TH>
  </TR>
  <TR>
    <TD> Pashupatinath Temple </TD>
    <TD> Hindu Temple in Kathmandu </TD>
    <TD> Ancient Architecture </TD>
  </TR>
   <TR>
    <TD> Boudhanath Stupa </TD>
    <TD> Center for Buddhism</TD>
    <TD> Religious Rituals </TD>
  </TR>
</TABLE>
</BODY>
</HTML>
```

VISIT NEPAL - FAMOUS PLACES

| PLACE | DESCRIPTION | HIGHLIGHTS |
|---|---|---|
| Pashupatinath Temple | Hindu temple in kathmandu . | Ancient architecture |
| Boudhanath Stupa | Center for Buddhism. | Religious Rituals. |

*Figure: 4.15*

***Example 12:*** **Using the CELLSPACING attribute in HTML**

```
<!DOCTYPE HTML>
<HTML>
<HEAD>
    <TITLE>TABLE EXAMPLES</TITLE>
</HEAD>
```

```
<BODY>
<CENTER> <H3> TABLE WITHOUT CELLSPACING </H3> </CENTER>
<TABLE BORDER = "1" WIDTH = "30%" ALIGN = "CENTER">
 <CAPTION> Without Cellspacing </CAPTION>
 <TR >
    <TH> NAME </TH>
    <TH> AGE </TH>
 </TR>
 <TR ALIGN = "CENTER">
    <TD> Krishna </TD>
     <TD> 25 </TD>
 </TR>
 <TR ALIGN = "CENTER">
     <TD> Shiva </TD>
    <TD> 30 </TD>
 </TR>
 <TR ALIGN = "CENTER">
    <TD> Ram </TD>
     <TD> 22 </TD>
 </TR>
</TABLE>
<HR>
<CENTER> <H3> TABLE WITH CELLSPACING </H3> </CENTER>
<TABLE BORDER = "1" WIDTH = "30%" ALIGN = "CENTER"
CELLSPACING = "10">
 <CAPTION> With Cellspacing </CAPTION>
 <TR>
    <TH> NAME</TH>
```

**TABLE WITHOUT CELLSPACING**

Without Cellspacing

| NAME | AGE |
|---|---|
| Krishna | 25 |
| Siva | 30 |
| Ram | 22 |

**TABLE WITH CELLSPACING**

With Cellspacing

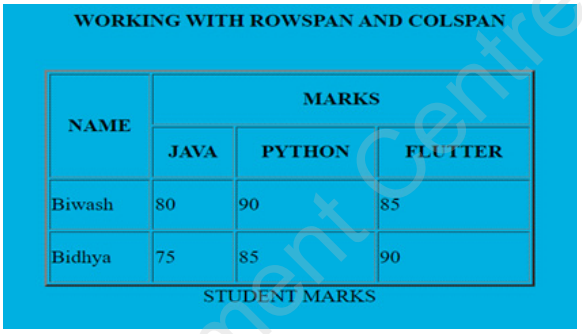
| NAME | AGE |
|---|---|
| Krishna | 25 |
| Siva | 30 |
| Ram | 22 |

```
    <TH> AGE</TH>
  </TR>
  <TR ALIGN = “CENTER”>
    <TD> Krishna </TD>
     <TD> 25 </TD>
  </TR>
  <TR ALIGN = “ CENTER”>
     <TD> Siva </TD>
    <TD> 30 </TD>
  </TR>
  <TR ALIGN = “ CENTER”>
    <TD> Ram </TD>
     <TD> 22 </TD>
  </TR>
</TABLE>
</BODY>
</HTML>
```

***Example 13:*** **Using ROWSPAN and COLSPAN**

```
<!DOCTYPE HTML>
<HTML>
<HEAD>
     <TITLE> WORKING WITH ROWSPAN AND COLSPAN </TITLE>
</HEAD>
<BODY BGCOLOR = “ SKYBLUE “>
<CENTER> <B> WORKING WITH ROWSPAN AND COLSPAN </B> <./
CENTER>
<TABLE BORDER = “ 2 “  WIDTH = “ 70% “ ALIGN = “ CENTER” HEIGHT
= “200”>
```

```
<CAPTION ALIGN = "BOTTOM"> STUDENT MARKS </CAPTION>
<TR ALIGN = "CENTER">
   <TH ROWSPAN = "2"> NAME </TH>
   <TH COLSPAN = "3"> MARKS </TH>
</TR>
<TR>
   <TH> JAVA </TH>
   <TH> PYTHON </TH>
    <TH> FLUTTER </TH>
</TR>
<TR>
   <TD> Bidhya </TD>
   <TD> 80 </TD>
   <TD> 90 </TD>
   <TD> 85 </TD>
</TR>
<TR>
   <TD> Biwash </TD>
   <TD> 75 </TD>
   <TD> 85 </TD>
   <TD> 90 </TD>
</TR>
</TABLE>

</BODY>
</HTML>
```



## Form and Div Tag

# 1. HTML <FORM> tag

You have probably seen and filled out different forms on websites, like **login form, contact form, registration form, search form** and so on. The main purpose of these web-based forms is **to get data and information** from the end users and send them to a server for processing.

HTML forms are a client-side object that provide an interface between client and server. <FORM> tag is used to create a form in HTML documents. Form tag can contain various input elements along with other HTML contents. This is why the form element is also known as a **container** for different types of input element.

The <input> element or tag in HTML is a flexible form tag that can make different kinds of input fields inside a <form> tag. The type attribute with input element decides the type of input field that includes text boxes, checkboxes, radio buttons, submit buttons, and more.

The opening tag of the FORM tag includes the attributes: name, method and action. Name is used for defining the name of the form. The Method attribute can be either GET or POST. Which determines how the form data is being sent to the server. The Action attribute is a pointer to the script that processes the form on the server side.

*Activity 4.6:*

Research on the internet and collect two or more forms and try to fill out the form. It is better if student find the gov sites form.

## Form elements in HTML

### a. Text Fields

Text fields enable end users / clients to type text into a single line input field. A text field is an input field where users can type text or alphanumeric characters.

To create a text field, you can either use TYPE = “TEXT” in the <INPUT> element or leave off the TYPE specification altogether. The default TYPE for the <INPUT> tag is “text”.

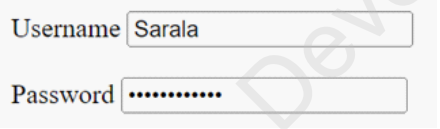
***Example:*** <INPUT TYPE = " TEXT " NAME = "UserName" SIZE =" 50" >

| | |
|---|---|
| **TYPE** | Determines what type of data is allowed for input |
| **NAME** | Determines the name of the field that passes to the script processing the form |
| **SIZE** | Indicates the length of the text-entry field in characters (The default field size is 20 characters) |

***Example 14:*** **Using "text" INPUT type**

```
<FORM>
    <label> Username </label>
<Input type = " text "  Name = " UserName "> <BR>
<label> Password </label>
<Input type = " password "  Name = "UserPass">
</FORM>
```

Output:




*Note:*

**< input type = " Password ">** lets users type in sensitive information, like passwords, securely. The actual characters they type are not shown.

**Lab work!**

We can add activities here like input username, age, phone number and other.

**b. Text Area**

The <textarea> tag in HTML creates a multiline text input field in a form. Users can type and send multiple lines of text with it, so it is good for longer texts or free-form content. For example, if you want to create a form that enables readers to compose electronic mail, you might use a text area for the body of the message.

**Syntax:** <Textarea Name = " BodyArea" Rows = "7" Cols = " 30"

Enter your message Here…. </Textarea>

**<TextArea> Attributes**

| | |
|---|---|
| **NAME** | The name to be sent when the form is submitted |
| **ROWS** | The height of the text area element (number of rows) |
| **COLS** | The width of the text area element in the column (number of columns) |

**Lab work!**

Make a text-area for the message for the above form.

***Example 15:*** **Using "Textarea " INPUT type**

```
<FORM>
    <b> Comments: </b> <br>
    <textarea  rows =  " 7 "  cols = "25"  name = "details">
        Write your comment here…..
    </textarea>
</FORM>
```

**Output:**

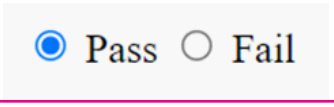

**c. Radio Button**

A radio button in an HTML form is an input element that lets users pick only one option from a group of options that cannot be picked together. If one radio button in a list is selected, all the rest of the radio buttons in the same list get deselected.

***Example 15:*** **Using "Radio" INPUT type**

```
<FORM>
<INPUT TYPE = "Radio" Name = "result" Value = "PASS"> Pass
<INPUT TYPE = "Radio" Name = "result"    Value = "Fail"> Fail
</FORM>
```

**Output:**




***Activities: 4.7***

Add the radio button for the age i.e., Male, Female, and others in the form.

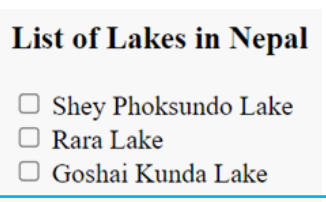
**d. CheckBox**

Unlike Radio Button, checkbox allows us to choose multiple items at a time in a Checkbox list. Each checkbox can be either checked or unchecked.

***Example 16:*** **Using "Checkbox" INPUT type**

```
<h3> List of Lakes in Nepal </h3>
<FORM>
<Input type = "Checkbox" Name = "Shey Phoksundo Lake"> Shey Phoksundo Lake <br>
<Input type = "Checkbox" Name = "Rara Lake">  Rara Lake  <br>
<Input type = "Checkbox" Name = "Goisai Kunda Lake"> Goshai Kunda Lake <br>
</FORM>
```

**Output:**




***Activity:4.8***

We can add the hobbies in the form such as programming language, guitar, football, and others in the above form.

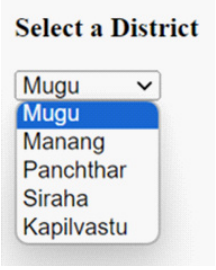
**e. Select Box**

It is used to create pull - down menus and scrolling lists. The <select> tag in HTML creates a dropdown list or a select box in a form. Users can pick one or more options from a list of options that are already set. The <option> tags inside the <select> tag make the options for the dropdown list or select box. <SELECT> elements work like radio buttons or checkboxes, but the interface is different.

***Example 17:*** **Using "select " INPUT type**

```
<HTML>
    <HEAD>
        <TITLE> Working with Select Box </TITLE>
    </HEAD>
    <Body>
        <form>
            <p> <b> Select a District </b></p>
            <SELECT Name = "District">
                <option value = "MUGU"> Mugu </option>
                <Option value = "Manang"> Manang </Option>
                <Option value = "Panchthar"> Panchthar </Option>
                <Option value = "Siraha"> Siraha </Option>
                <Option value = "Kapilvastu"> Kapilvastu </Option>
            </SELECT>
        </form>
    </Body>
</HTML>
```

**Output:**



*Activity:4.9*

Make the address in the form and make the select box in the form.

**f. Buttons**

The <button> tag in HTML creates a button that can be clicked on a form or anywhere on a webpage. The button can have text, images, or other HTML tags inside it. It is used to perform specific tasks or perform events when it is clicked. There are two types of Buttons:

i. **Submit Button:** It tells the browsers to send the form data to the server. You should include at least one submit button on every form.

**Syntax:** <input type = " submit " value = " submit " >

**Note:** Value attribute is used to give the text display in a button.

ii. Reset Button: It is used for resetting the elements of a form. It allows the user to fill new entries all over again.

**Syntax:** <input type = " reset " value = " reset ">

***Example 18:*** **Using "select" INPUT type**

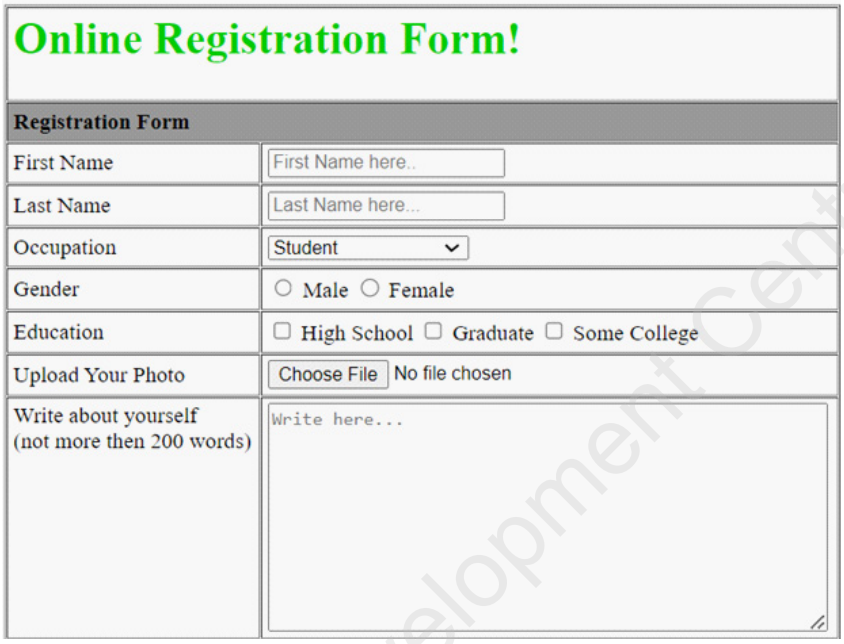
```
<form>
    <input type = "submit"   name = "submit"   value = "Submit">
    <input type = "reset"    name = "reset"    value = " Reset">
</form>
```

Output:


Add the submit and reset button on the form at last.

### Lab work!

**Let's make a form like below.**




## 2. HTML <DIV> Tag

**Context:**

The  (division) tag in HTML is a division or container element that is used to divide a web document logically. It helps to o**rganize and arrange content on a webpage**. It is a block-level container that can contain other HTML elements.

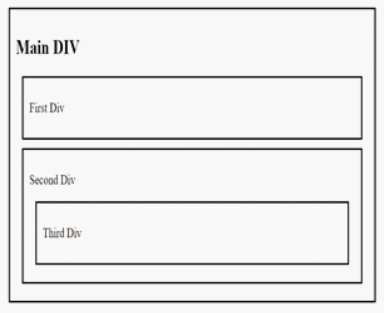
The div tag can be styled with CSS to change its appearance, size, position, and layout. It can also have an id or a class attribute to identify it or group it with other elements.

**Advantages of using DIV tag**

a. Grouping Content
b. Layout and Structure
c. Responsive Design
d. Styling with CSS
e. JavaScript Interaction

***Example 21:*** **Sample Layout using <DIV> Tag**

```
<!DOCTYPE html>
<html>
<head>
        <title> DIV Example </title>
</head>
<body>

<! - - Parent DIV -  ->
<div style = “border: 2px solid #000;   margin: 10px;  padding: 10px;”>

<h2> Main DIV </h2>

   <!-- First Div -->
   <div style = “border: 2px solid #000;  margin: 10px;  padding: 10px;”>
      <p> First Div </p>
      <!-- Content of the first div goes here -->
   </div>

   <!-- Second Div -->
   <div style = “border: 2px solid #000;  margin: 10px;  padding: 10px;”>
      <p> Second Div </p>
      <!-- Content of the second div goes here -->

      <!-- Third Div inside Second Div -->
      <div style=”border: 2px solid #000; margin: 10px; padding: 10px;”>
         <p> Third Div </p>
         <!-- Content of the third div goes here -->
      </div>
   </div>
</div>

</body>
</html>
```



- <Div> Defines a division or a section in an HTML document.

- To Style a specific HTML element directly, you can use the style attribute with inline CSS.

***Let's know***

a. Why is div needed while writing HTML code? With its advantages?

b. Write a HTML code using the div function to print The height of Sagarmatha is 8848 m.

## 4.5. Cascading Style Sheets (CSS)

**Dynamic Hypertext Markup Language**

DHTML stands for **Dynamic Hypertext Markup Language**. It is a technology that allows web developers to create interactive and animated websites. DHTML uses HTML to define the structure of a web page, such as tables, frames, paragraphs, lists, and so on. It also **uses CSS to control the appearance and layout of the HTML elements**, such as colors, fonts, margins, and positions. In addition, it uses scripting languages, such as JavaScript and VBScript, to access and modify the HTML elements and CSS styles dynamically based on user actions or events.

### Cascading Style Sheet

**Context:**

HTML is a language that uses tags to create different elements on a web page, such as text, links, images, and videos. CSS is another language that works with HTML to style and layout these elements. You can use CSS to change the color, font, and size of a paragraph element. For example,

**< p style = " color : #ff0000; font-weight: bold "> CSS helps to define Style </p>**

CSS is a language that works with HTML to style and format web pages. It **can change the appearance of elements** such as colors, layout, and fonts. It can also help to **define responsiveness** which means CSS adjusts the presentation to suit different devices, such as desktops, mobiles, or printers.

As said in the definition, CSS can be defined in such a way that it **can change layout, colors, fonts or any other styles based on the size of the screen** (PC

Screen, Tablet screen and mobile screen) or for printers. In a desktop computer, screen size is large whereas in mobile screen size is small and accordingly HTML content should be presented to webpage viewers. CSS is used to define this type of presentation. Also, if a viewer is trying to print a web page, the presentation of the page can be made different using CSS to make it printer-friendly.

The **Style assignment is done** with **<STYLE>…</STYLE>** tags. Between the tags, specific style attributes are listed. The Style tags are written within the <HEAD>…</HEAD> tags.

### CSS Syntax:

```
<STYLE   Type = "text/css">
Tag {attribute : value;  attribute : value; …}
.
.
.
</STYLE>
```

**Attribute:** Font, Color, Background, Text, Border, Margin and List

**Let's answer these questions!**

Why is CSS important in web development?

What are the advantages of CSS?

### A CSS rule is formed from:

- A set of properties, which have values set to update how the HTML content is displayed. For example, I want my element's text color to be white, and its background to be grey.
- A selector, which selects the element(s) you want to apply the updated property values to. For example, I want to apply my CSS rule to all the paragraphs in my HTML document.
- A set of CSS rules contained within a stylesheet determines how a web page should look.

Let's make things clear with an example. First of all, let's take a simple HTML document, containing an <h1> and a <p>

***Example 1:***

```
<!DOCTYPE html>
<html>
 <head>
                <title> My First CSS Example </title>
  <style>
     h1 {
                 color : blue;
          background-color : yellow;
          border : 1px solid black;
              }
      p {
          color: red;
              }
  </style>
 </head>
 <body>
    <h1> Hello World! </h1>
    <p> This is my first CSS example.  </p>
 </body>
</html>
```

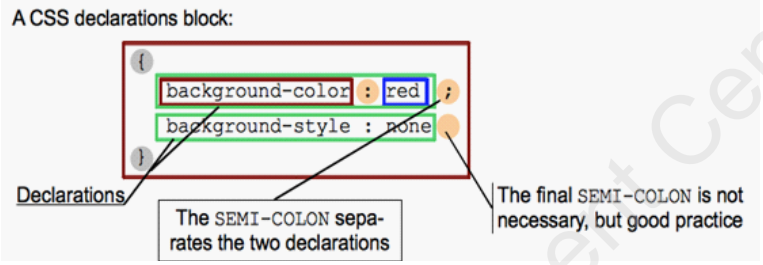

## 4.5.1. Properties of CSS

CSS is a language that works with HTML to style and format web pages. It has two main components:

**Properties:** These are the names of the aspects of the web page that you want to modify, such as font, width, background color, etc.

**Values:** These are the specific settings that you assign to each property, such as Arial, 50%, blue, etc.

A property and a value together form a CSS declaration. You can group multiple CSS declarations inside a CSS Declaration Block. You can also use selectors to specify which HTML elements the CSS Declaration Block applies to. This creates a CSS Rule Set (or CSS Rule). Each property has a predefined set of values that are valid for it.




## 4.5.2. Selectors

CSS selectors are expressions that identify, target, and select HTML elements on a web page, and apply styles to them. Selectors are part of CSS which enable you to change the appearance of elements without changing the HTML code. Selectors are essential for creating the rules that control the style and layout of a web page.

### Types of CSS Selectors:

1. **Element selector**

   It is used to select HTML elements based on the element name.

   ***Example:***

```
b {
        /* Styles applied to all <b> elements */
}
```

2. **ID Selector**

   It is used to target a specific HTML element based on its unique ID attribute. As each element's ID must be unique within a page, the ID selector allows the

selection of a single, distinct element. To select an element with a specific id, write a hash (#) character, followed by the id of the element.

***Example:***

```
#header {
        /* Styles applied to the <p> element with id = “header” */
}

<Body>
        <p id = “ header “> I am Id Selector with Id name check  </p>
</Body>
```

### 3. Class Selector

The class selector is used to select and style HTML elements that have a certain class attribute. To select elements with a certain class, a dot (.) is written followed by the class name.

***Example:***

```
.check {
        /* Styles applied to all elements with class = “ check “ */
}

<Body>
    <p class = “ check “> I am Class Selector with class name check </p>
</Body>
```

### 4. Universal Selector

The universal selector (*) in CSS is a symbol that is used to select all HTML elements on a web page. It matches any element type, allowing you to apply styles to every element without specifying individual element names, classes, or IDs.

*Example:*

```
* {
        /* Styles applied to all elements */
            property : value;
        /* Additional styles */
}
```

## 4.5.3. Types of CSS:

There are three different ways to apply CSS to an HTML document:

Inline CSS, Internal CSS, and External CSS

### i. Inline CSS

It is a way of styling HTML elements directly into an HTML document. You can do this by adding the style attribute to the HTML tags. Inline CSS lets you set specific styles for a certain element without changing other elements on the page.

***Example:***

```
<!DOCTYPE html>
<html>
  <head>
    <title> My CSS Experiment </title>
  </head>
  <body>
    <h1 style =" color :blue ; background-color :yellow ; border: 1px solid black;">
Good Morning Class !  </h1>
   <p style = " color : red; "> This is my first CSS example of Inline styles  </p>
  </body>
</html>
```

### ii. Internal CSS

Internal CSS is a method of styling HTML elements within the HTML document itself. Instead of styling each element individually, internal CSS allows you to define styles for multiple elements within a single <style> tag in the <head> section of the HTML document.

By adding CSS code inside the opening <head> and closing <head> tag inside an HTML file, we can work with numbers of HTML elements at a time in order to define styles on the same web page.

***Example:***

```
<!DOCTYPE html>
<html>
 <head>
  <meta charset = "utf-8">
  <title>My First CSS Try Page </title>
    <style>
          h1 {
                    color : blue;
              background-color : yellow;
              border : 1px solid black;
               }
         p {
              color : red;
              }
 </style>
</head>

<body>
    <h1> Hello World! </h1>
     <p> This is my first CSS example of an Internal stylesheet. </p>
</body>
</html>
```

### iii. External CSS

External CSS is a method of styling HTML elements by placing the CSS code in a separate external file with a .css extension. This file is then linked to the HTML document using the <link> tag in the <head> section.

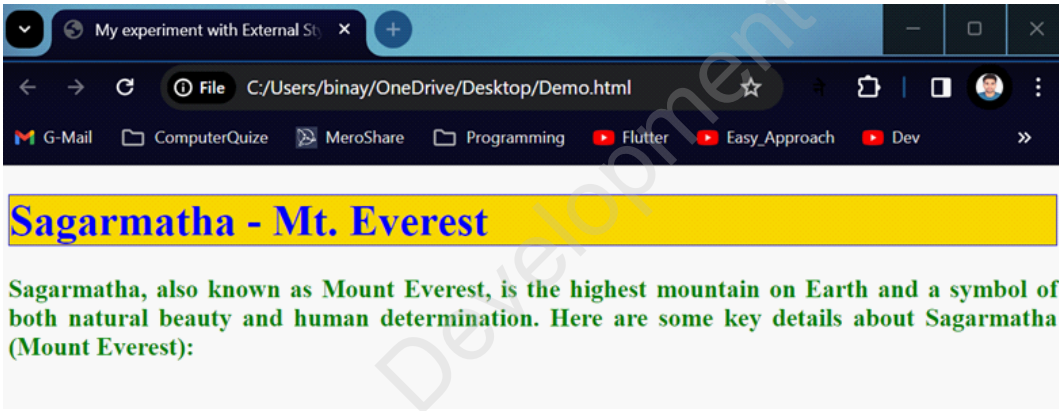
Unlike inline CSS or internal CSS, external CSS allows for the separation of style and content, making it easier to manage and maintain styles across multiple web pages. This approach promotes code reusability, as the same CSS file can be linked to multiple HTML documents, ensuring consistent styling throughout the website.

***Example:***

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset = “utf-8”>
    <title> My Experiment with External Stylesheet </title>
          <link rel = “stylesheet” href = “mystyle.css”>
  </head>
  <body>
    <h1>  Visit Nepal 2020!  </h1>
    <p> Sagarmatha, also known as Mt. Everest, is the highest mountain on Earth and a
symbol of both natural beauty and human determination. Here are some key
details about Sagarmatha (Mount Everest): </p>
  </body>
</html>
```

**And (create a new file) code for mystyle.css file:**

```
h1 {
  color: blue;
  background-color: gold;
  border: 1px solid black;
}
p {
   font-size :14pt;   font-weight : bold;   text-align : justify;  color : red;
}
```




### i. Common CSS properties reference

| Property | Values | Used with Elements |
|---|---|---|
| **color** | #RRGGBB (Red, Green, Blue hex values) | any element that contains text |
| **text-align** | left \| right \| centre \| justify | block elements like h1.. h6, p, li, etc. |
| **text-decoration** | none \| underline \| overline \| line-through \| blink \| inherit | mostly with a (anchor) elements |
| **text-transformation** | none \| capitalize \| uppercase \| lowercase | any element that contains text |
| **line-height** | % or px | block elements h1..h6, p, li, etc. |

| | | |
|---|---|---|
| **letter-spacing** | normal or px value | any element that contains text |
| **font-family** | font or font-family [, font or font-family ...] | any element that contains text |
| **font-size** | px or em value | any element that contains text |
| **font-style** | normal \| italic \| oblique | any element that contains text |
| **font-weight** | normal \| bold | any element that contains text |
| **background-color** | #RRGGBB (Red, Green, Blue hex values) | any element with a background |
| **background-image** | url("[image url]") | mostly with body |
| **background-repeat** | repeat \| repeat-x \| repeat-y \| no-repeat | mostly with body |
| **background-position** | left \| center \| right \| top \| center \| bottom | mostly with body |
| **list-style-type** | disc \| square \| circle | ul |
| **list-style-type** | decimal \| lower-roman \| upper-roman \| lower-alpha \| upper-alpha | ol |

### ii. CSS Box Model

CSS is a language that makes web pages look nice. It uses the box model to arrange HTML elements on the page. Each element is like a box with four parts:

There are several properties in CSS that are often grouped together as **box properties** which control the way browsers layout the page using the CSS **box model**.

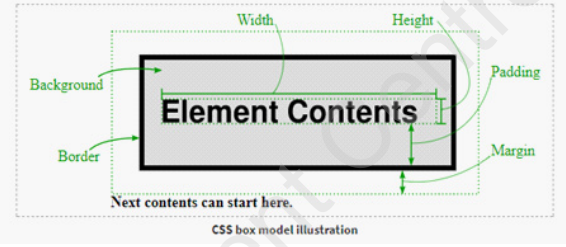
For example, if we have an element (like a **<h2> Element Contents </h2>**) on the page, here are the parts of its 'box' when drawn:

**Content:** It is what you see inside the box, such as words, pictures, or videos.

**Padding:** It is the cushion around the content. It makes the box bigger and more comfortable.

**Border:** It is the frame around the content and padding. It shows where the box ends and the next one begins.

**Margin:** It is the gap between the boxes. It gives some breathing room and keeps the boxes from touching.

CSS box model illustration

**CSS code:**

```
h2 {
    padding : 1em ;
    border  : medium dashed black ;
    background-color : grey ;
}
```

It will make the **<h2>** have a gray background, with **1 em** of space between the text and the border. The space inside the border (even if it is invisible because we do not have one) is covered with the background color and is controlled by the **padding** properties.

If we want more space separating this element (and its border) from the stuff around it, we would have to increase the **margin** values. It's easy to mistake margin for padding, especially when there is no border or background color. If you are trying to change the space around an element (especially when the browser's default CSS has some space there), try setting both margin and padding to zero and work from there.

```
ul {
        margin : 0em;
                padding : 0em;
}
```

**iii. CSS measurement units**

Here are some common units of measurement used in CSS:

**em -** The current font size. Another unit, an ex is half the text size.

**px -** One screen pixel (dot) on the display. (Note: for some very high-resolution devices, real "pixels" are very small, so this length is adjusted to be close to the size of a pixel on a traditional display.)

**mm -** A millimeter. There are also units for centimeters, inches, etc. (Note: This is the browser's best guess, but might be inaccurate depending on the scaling of the display/projector/phone/etc. For example, if you display your screen on a projector, the "millimeter" suddenly becomes much larger.)

As much as possible, it is better to specify measurements using ems and pxs. These are the only units that do not require any kind of note.

```
p {
      line-height: 1.5em;
  }
blockquote {
                margin-left : 2em;
                border-left : 0.25em solid black;
}

# logoimg {
        width : 120px;
        height : 160px;
        float : left;
        margin-left : 1em;
}
```

Here is an example page that we can style with some of the properties above:

***Example 5:***

```
<!DOCTYPE html>
<html>
<head>
<meta charset = “UTF-8” />
<title> CSS Properties </title>
<link rel = “stylesheet”  href = “css-prop.css” />
</head>

<body>
<h1> CSS Properties </h1>
<h2> Goals </h2>
<p> This is a page that we’re using to demonstrate various CSS properties and techniques. Because of that, it is probably going to be ugly. </p>
<h2> Results </h2>
<p> Yes, it is turning out rather ugly, but it is important to demonstrate some CSS stuff. Here are some of the new things: </p>

<ul>
<li> more CSS properties </li>
<li> the box model </li>
<li> the units of length used </li>
</ul>
</body>
</html>
```

**CSS to go with it:**

```
body {
  font-family : “Helvetica”, sans-serif;
}

h1 {
  text-align : center ;
  font-weight : bold ;
  background-color : silver;
  color : teal ;
  padding : 0.25em ;
}

h2 {
  border : medium dotted teal;
  font-weight : normal ;
  padding : 0.1em ;
}
```

Try experimenting with CSS on this page (or one you have created).

**DID YOU KNOW?**

Short-hand properties in CSS allow you to set multiple related properties using a single declaration. For example, instead of writing separate lines of code for font-size, font-family, and font-weight, you can use the font property to set them all at once.

## Exercises

**1. Write down the full forms of:**

a. HTML c. UI e. UX g. DNS i. HTTP

b. WDLC d. W3C f. URL h. RGB j. JS

**2. Choose the correct option.**

a. What does HTML stand for?

i. Hyperlinks and Text Markup Language

ii. Hypertext Markup Language

iii. Home Tool Markup Language

iv. Hyperpage Markup Language

b. Which of the following is a commonly used text editor for web development?

i. Microsoft Word ii. Notepad

iii. Sublime Text iv. Photoshop

c. Choose the correct HTML tag for the largest heading.

i. <h3> ii. <h1>

iii. <big heading> iv. <h6>

d. What is the correct HTML tag for inserting a horizontal line?

i. <br> ii. <hr> iii. <ar> iv. <tr>

e. What is the correct HTML for adding a background color?

i. <body bgcolor=”yellow”>

ii. <background> yellow </background>

iii. <body color=”yellow”>

iv. <body background = “yellow”>

f. Choose the correct HTML tag to give base value.

i. <Power> ii. <sup> iii. <base> iv. <sub>

g. Choose the correct HTML tag to create a hyperlink.

i. <a> ii. <b> iii. <c> iv. <d>

h. What is the correct HTML for making a hyperlink?

i. <a url="http://www..ntb.gov.np"> Nepal Tourism Board </a>

ii. <a href="http://www..ntb.gov.np"> Nepal Tourism Board </a>

iii. <a name="http://www..ntb.gov.np"> Nepal Tourism Board </a>

iv. <a> http://www..ntb.gov.np </a>

i. How can you make an email link?

i. <mail> xxx@yyy </mail> ii. <a href="mailto:xxx@yyy">

iii. <mail href = "xxx@yyy"> iv. <a href="xxx@yyy">

j. How can you open a link in the same browser window?

i. <a href = "url" target = "_blank">

ii. <a href = "url" target = "new">

iii. <a href = "url" same>

iv. <a href ="url" target= "_self ">

k. Which symbol is used to denote an ID selector in CSS?

a. & b. # c. ! d. @

l. What is the primary purpose of a wireframe in web development?

i. To test functionality ii. To showcase final design

iii. To outline basic structure and layout

iv. To implement coding

m. In web design, what does "UI" primarily focus on?

i. User Interaction ii. User Exploration

iv. User Expression v. User Experience

n. There are _____ types of CSS.

i. one iii. two ii. three iv. four

o. Which of the following CSS tags is used for creating External Style Sheets?

i. <style src="mystyle.css">

ii. <stylesheet> mystyle.css </stylesheet>

iii. <link rel= stylesheet href= "stylesheet file name">

iv. <link rel = stylesheet.html href = "url">

## 3. Write short answers to these questions.

a. What is DNS?

b. Define HTML tag.

c. What is the main function of <HEAD> tag?

d. What is an attribute?

e. Which tag is used for breaking lines or paragraphs in HTML?

f. What is an Empty tag?

g. What is a container tag?

h. Write down the syntax of the comment tag.

i. What is the output of (a+b) <sup> 2 </sup> tag?

j. What is the function of <TR> tag?

k. Write down the use of Form in web pages.

## 4. Write long answers to these questions.

a. What is HTML? List out any three uses of HTML.

b. Write down the structure of HTML.

c. Explain Web Development Life Cycle.

d. List out the importance of UI/UX.

e. How is the FONT tag used? List out the attributes of the FONT tag with its attributes.

f. What are the different types of Input types in FORM?

g. What is the <DIV> tag? Write down the importance of the DIV tag.

h. What is CSS? List out its advantages.

i. Explains the types of CSS.

j. What are CSS selectors? Why is it important?

## *Project work*

### Develop a Static Web Page Using HTML and CSS.

This project will help you learn the basics of web development, which is a valuable skill in today's digital world. You will learn how to use HTML and CSS to create a

website that showcases your personal portfolio. You will also learn how to design a website that is user-friendly, responsive, and attractive.

## Website Overview

In this project, you will create a website for yourself. It will have four pages: Home, Gallery, Contact, and Information. Each page has a different job. The Home page welcomes visitors, the Gallery shows your personal portfolio images, the Contact page lets people reach out to you, and the Information page shares important details about you. This project is a chance for you to be creative and show off your skills while helping yourself shine online!

1. Your task is to design and develop a static website for yourself.
2. The website should consist of four main pages: Home, Gallery, Contact, and Information.
3. Each page should have a consistent navigation bar (navbar) and footer design.
4. Utilize HTML for structuring the content and CSS for styling and layout.

## WireFrame

**Design Wireframe of your Webpage.**

### Navbar

A navbar is a part of a website that helps you find other pages on the site. It usually has the site name and some buttons at the top. You can click on the buttons to go to different pages.

### Perform:

1. Create a navbar that is consistent across all pages.
2. Include menu items such as Home, Gallery, Contact, and Information.
3. Ensure that the navbar is responsive and functional, allowing users to navigate between pages easily.

### Footer

A footer is a part of a website that you can find at the bottom of a page. It has extra information that is not in the main part of the page. Some things you might see in a footer are: Copyright Information, Contact Information, Navigation Links, Social Media Icons/Links, Sitemap, etc.

## Perform:

**Include essential information in the footer such as:**

1. Contact details: Provide your email, and phone number for users to reach out.
3. Social media links: Include icons or links to your social media profiles (e.g., Facebook, Twitter, Instagram).
4. Ownership information: Add a line indicating who owns the website and when it was created.

## Home Page

The homepage is the first page you see when you visit a website. Home page is also known as the Index page. It is like the front door, welcoming you in and showing you around. On the homepage, you'll find menus to explore different parts of the site, links to important pages, and highlights of what the website is all about.

## Perform:

- At least 1 image should be inserted as a cover image of the page.
- Home page should contain a welcome message section.
- Home page should contain a section for upcoming events.
- Home page should contain a social media icon where it should hyperlink with the social media site.
- Home page should contain a footer.

## Gallery Page

A gallery page on a website serves as a visual showcase for various forms of multimedia content. Typically, it displays collections of images, videos, or other visual elements related to a specific theme, event, product, or service.

- Page should contain different sections such as Events, Head of the Department, and others.
- There should be 2 or 3 sections.
- Page should require at least 3 or 4 photos with captions, which are needed for each section.

## Information Page

In this section, there should be a static table where your information is to be mentioned. A sample structure for information page is given below:

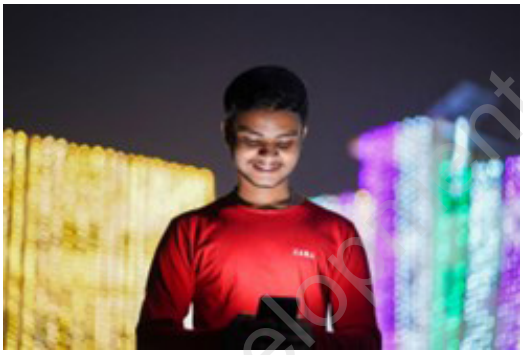
## Contact page:

- There should be form.
- In the form, there should be first name, last name, phone number, address, email, radio button- and drop-down menu and message box.
- There should be a button to submit.

**To complete this project, you need to do the following tasks:**

1. Create a folder for your project and name it “my-porfolio.html”.
2. Create four HTML files and name them “index.html”, “gallery.html”, “contact.html”, and “information.html” and store them in myProject folder.
3. Create a CSS file, name it “style.css” and store it.
4. Link the CSS file to all the HTML files using the <link> tag.
5. Add the basic HTML structure to each file.

# Unit 5 Internet and Social Media

**Let's think!**

a) How has the internet changed the way we access and share information compared to traditional methods, like newspapers and letters in Nepal?

b) In what ways do you think social media has impacted our daily communication and relationships with others?

c) Why do you think understanding online tools, such as web browsers, search engines, and email, is essential for navigating the digital world?

We read newspapers in our daily lives through which we can get information from all around the world which is limited. But with access to the internet, we can read newspapers from all around the world through our mobile phones, tablets, laptops, and other electronic devices.

*Figure:5.1*

The Internet is a global communication system connecting thousands of individual networks, forming a vast network of networks. It facilitates the exchange of messages, real-time communication, data sharing, and access to vast information repositories.

***Do You Know?***

Approximately 4.32 billion people use their mobile to devices access the internet.

Internet and social media explore the foundational concepts governing online communication. It begins with an overview of the World Wide Web (WWW), web browsers, search engines, and URLs. Students delve into remote login applications, web protocols, and Virtual Private Networks (VPNs). The unit also emphasizes the significance of email and provides hands-on tasks, delving into social media's diverse applications. Online video meetings are introduced with examples and practical tasks, leading to exploring blogs and their key features. This comprehensive unit provides students with essential insights into the interconnected web of the internet and social media, enhancing their ability to navigate the digital landscape effectively.

## 5.1 Concept of WWW, Web Browser, Search Engine, and URL

### World Wide Web (WWW)

The World Wide Web (WWW) refers to a network of interlinked hypertext documents and multimedia content accessible online.


*Figure:5.2*

Users use web browsers to search, view, and download information. WWW or World Wide Web was developed in the European Particle Physics Laboratory (Geneva, Switzerland) in 1989 A.D. The Web has evolved to become one of the most widely utilised Internet services in recent years.

## Web Browser

A web browser is a special computer program that helps you find and access information on the internet. A browser is like a tool that allows us to explore and read information from the web on our computer. It is also used to interpret and open the documents. Mosaic was the 1st web browser and Google Chrome, Safari, and Mozilla Firefox are the most popular browsers in use today.


*Figure:5.3*

***Do you know?***

Mosaic, which is the first web browser started developing in the late 1992 which was founded by Marc Lowell Andreessen.

## Search Engine

A search engine is client-side software that helps us find data and information on the internet based on keywords provided by users. When we type in words or phrases, it shows us web pages related to what we are looking for.

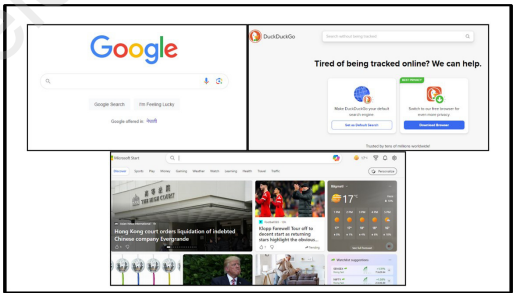



*Figure:5.4*

Google, Bing, Baidu and DuckDuckGo are some examples of popular search engines. Most search engines provide not only text results but also images, videos, news, maps, and shopping services. There are also some specialized search engines such as KidRex, a search engine for school students.

***Let's check!***

Which of the following is the most popular search engine?

a) Bing b) Google c) Yahoo d) Microsoft

Search engines use advanced techniques to help users find relevant information online. For instance, they use algorithms to match keywords entered by users with

the content on web pages to check the relevance of the information found on the web. Moreover, users can refine their searches using Boolean operators like "AND," "OR," and "NOT," ensuring that specific keywords must or must not be present in the results.

Additionally, putting quotes around a phrase narrows down searches, while wildcard characters like "*" broaden them to include variations of a word. Users can also limit searches to specific websites using the "site:" operator, or find particular file types using "filetype:" followed by the extension, like PDF or DOC.

*Activity 5.1*

**Activity Outcome -**

- Understand the impact of different search operators on search engine results by analyzing result counts.

Required Resources: Mobile phone or computer with internet access and a search engine (e.g., Google).etc.

1. Open a search engine on your device (such as Google).
2. Enter the keyword "**search engine**" using different search operators as instructed below.
3. After each search, note down the result count displayed by the search engine.
   - **Without Quotes**: Type search engine without any quotation marks.
   - **Using Quotation Marks (" ")**: Type "search engine" with quotation marks around the entire phrase.
   - **Using "filetype:" Operator**: Type "search engine:pdf" to look for PDF files related to the term.
   - **Using Asterisk (*)**: Type "search engine*" to include variations of the keyword.

| Quote | Keyword | Result number |
|---|---|---|
| Without using qoute | search engine | |
| Using " " Quote | "search engine" | |
| Using " filetype:" Quote | "search engine:pdf" | |
| Using "*" Quote | "search engine*" | |

**Result:** Group discussion on the impact of different search operators and how they influence search results, with each group sharing their recorded data.

## URL

A Uniform Resource Locator (URL) is like an internet address that helps you find and access things online. It is what you type into a web browser to reach specific resources, like websites or files. It is commonly used in web browsers to navigate to specific web pages.


*Figure: 5.5*

**Here are the key components and concepts associated with a URL:**

a. **Scheme/Protocol:** It is the method used to access things, such as web pages (HTTP), secure pages (HTTPS), or files (FTP).

b. **Domain:** This is the server's name, like "www.example.com." "www" can be a part of it, and "example.com" is the main name.

c. **Top-Level Domain (TLD):** It's the last part of the domain, showing the website type, like ".com," ".org," or ".edu."

d. **Path:** This is where the specific thing is on the server, like "/path/to/resource."

**Here's an example of a complete URL and its components:**

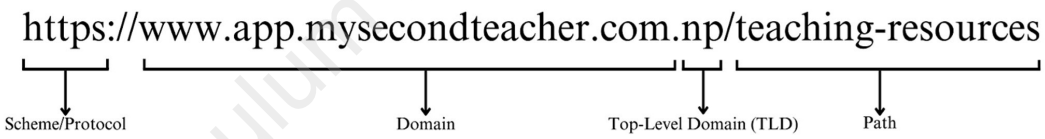



*Figure: 5.6*

*Activity 5.2*

**Activity Outcome -**

- Access and analyze online resources, observing available content and resource types on the specified website.

***Required Resources:*** *Personal computer or mobile phone with internet access, etc.*

1. Open a web browser on your device.

2. Enter the following URL in the address bar: https://mysecondteacher.com.np/teaching-resources
3. Review the content and teaching resources available on the page, noting specific types of resources, topics covered, and any additional observations.
4. Record your findings in the table below:

| URL | Result/Findings |
|---|---|
| https://mysecondteacher.com.np/teaching-resources | |
| | |

**Result:** Group reflection on the types and quality of resources found on the page, with discussion on their potential usefulness for academic learning.

## 5.2 Concept of Remote Login

**Remote Login**

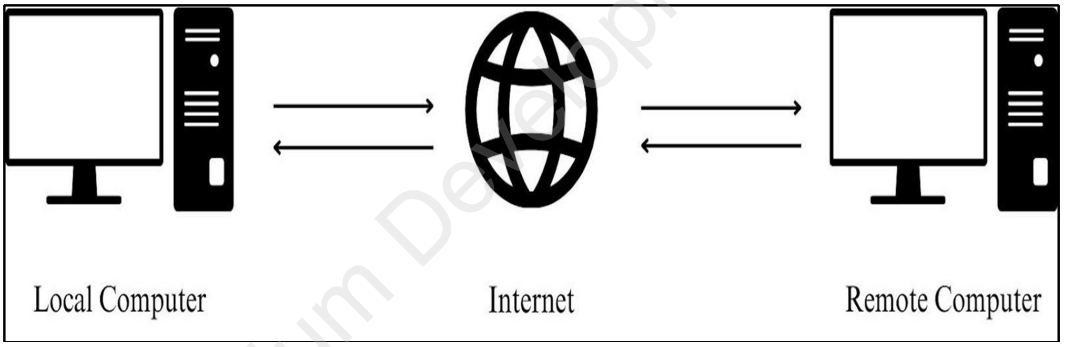



*Figure: 5.7*

Remote Login is a process in which a user can log into a remote site i.e. computer and use services that are available on the remote computer. With the help of remote login, a user can understand the result of transferring and the result of processing from the remote computer to the local computer.

### Examples of Remote Login

**Some examples of remote login software are:**

**AnyDesk:** AnyDesk is a remote desktop software application that allows users to access and control a computer or device from another location.

**TeamViewer:** TeamViewer is another software that allows us to remotely access and control any computer or device. TeamViewer is similar to AnyDesk, but it also has features such as online meetings, video calls, and cloud storage.

*Activity 5.3*

**Activity Outcome -**

- Learn to set up an Anydesk account and use it to connect to a remote computer.

Required Resources: Personal computer or mobile phone with internet access, etc.

**Procedure:**

**Step 1: Sign in/create an account.**

1. Visit Anydesk and download the application on your PC.
2. Open Anydesk and choose to sign in using your Google account or create a new account if you do not have one.

**Step 2: Connect from local to remote computer**

1. Open the Anydesk application on your computer.
2. Copy your Anydesk Address from the application window.
3. Click the Invite button located near the address field.
4. Paste the Address of another Anydesk account with which you want to connect.
5. Select the profile for Screen Sharing.
6. Click on Invite and wait for the other user's approval to begin the remote session.

**Result:** Teacher-led discussion and reflection on the experience, addressing any issues encountered, and clarifying remote access best practices and security considerations.

## 5.3 Concept of protocols

### Web protocols

Figure:5.8

A protocol is like a set of rules that guides computers to communicate with each other on the internet. It defines how data is organized and moved between computers, making sure it is done efficiently, safely, and reliably. Different protocols are used for various online activities like sending files, browsing the web, emails, and streaming.

**Examples of web protocols**

Some of the examples of web protocols are:

### HTTP


Figure:5.9

HTTP stands for Hypertext Transfer Protocol, which is a set of rules for how web browsers and servers communicate with each other. It works like a request-and-response system: the browser asks for something, and the server gives it. However, because HTTP isn't secure, others might be able to see or change the information being sent.

### HTTPS

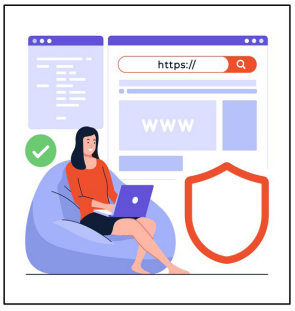



Figure:5.10

HTTPS, or Hypertext Transfer Protocol Secure, is a protocol ensuring secure communication between web browsers and servers. This protocol is employed for the safe transfer of various resources, such as web pages, from a web server to a web browser.

*Activity 5.4*

**Activity Outcome -**

- Recognize the difference between HTTP and HTTPS protocols by identifying websites using each protocol.

**Required Resources:** Personal computer or mobile phone with internet access, etc.

**Procedure:**

1. Open a web browser on your device.
2. Use a search engine to find one website that uses **HTTP** (without encryption) and another that uses **HTTPS** (with encryption).
3. Note the URL of each website, observing the security indicators (such as a padlock icon) present in HTTPS sites but not HTTP ones.
4. Record the URLs in the table below:

| Protocol | Website URL |
|---|---|
| HTTP | |
| HTTPS | |
| | |

**Result:** Group discussion on the significance of HTTPS for security, focusing on data encryption, privacy, and protection against potential threats associated with HTTP.

## 5.4 Email and its uses

### Email

Email, short for electronic mail, is a popular way for people to exchange digital messages using devices like computers, tablets, and smartphones. It consists of a sender, a recipient, a message body, and sometimes attachments.

*Figure: 5.11*

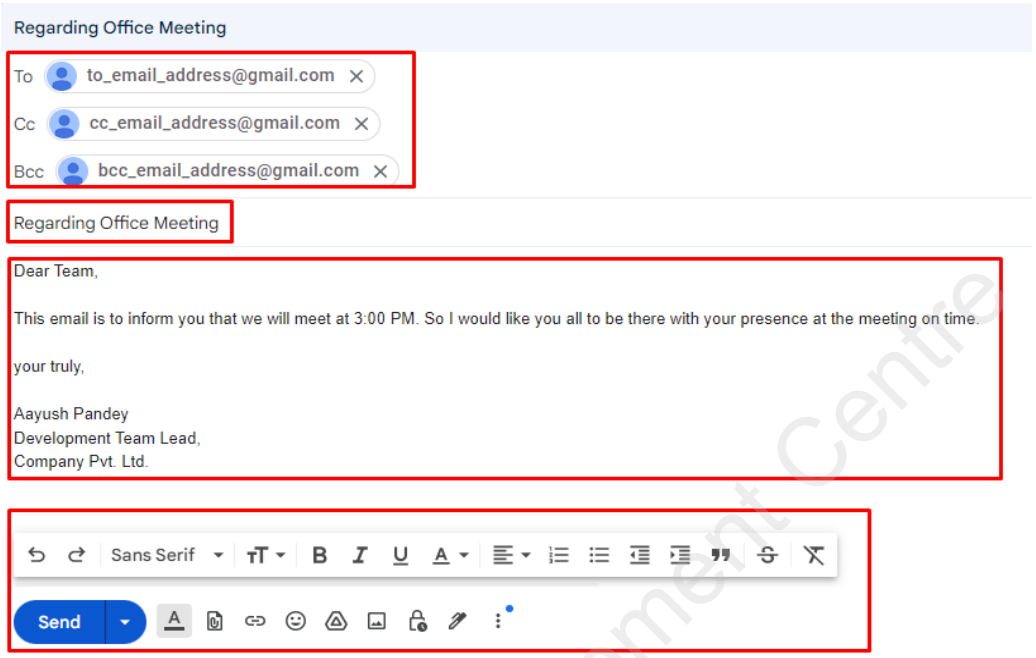



*Figure: 5.12*

Here,
In email, the "To" field indicates the primary recipient, while "Cc" (carbon copy) allows sending a copy to additional recipients who are not the main addressee but should still be informed. "Bcc" (blind carbon copy) serves a similar purpose as Cc but hides the additional recipients from others.

Common email platforms like Outlook facilitate various actions such as "Reply" for responding to a message, "Forward" for sharing it with others, "Adv" for marking an email as an advertisement, "Dis" for marking it as spam or unwanted, and "Attach" for including files with the message. These features contribute to the versatility and functionality of email communication.

## Uses of email

**Some of the uses of email are:**

a. **Personal Communication:** Email facilitates sharing thoughts and news, congratulating, inviting, and expressing sympathy with friends and family.

b. **Education:** Email provides students with a convenient way to ask questions and communicate with mentors, teachers, and classmates.

c. **Business:** Email is essential for business communication, connecting with suppliers, investors, and customers, and facilitating transactions and brand promotion.

d. **Marketing:** Email is an effective marketing tool, distributing surveys, offers, and newsletters to enhance client satisfaction and loyalty.

e. **Entertainment:** Email allows fans to stay connected with their favourite celebrities, receive updates, and participate in fan club activities.

*Activity 5.5*

**Activity Outcome -**

- Learn to write a well-structured, formal email to inform a teacher about an absence.

**Required Resources:** Personal computer or mobile phone with internet access, etc.

**Procedure:**

1. Open your email application.
2. Write a clear **subject line** like: "Absence Notification for [Date(s)]."
3. Start with a **polite greeting** (e.g., "Dear [Teacher's Name],").
4. In the **body of the email**, include:
5. Reason for your absence (e.g., illness, family commitment).
6. The specific date(s) you will be absent.
7. A brief statement about catching up on missed work or assignments.
8. Conclude with a **closing statement** (e.g., "Thank you for your understanding").
9. **Sign off** with "Best regards" or "Sincerely," followed by your full name and class information.

**Result:** Teacher reviews students' emails, providing feedback on the formality, clarity, and completeness of each email draft.

## 5.5 Appropriate usage of social media and safely accessing web sites

### Social media

Before the invention of the internet, people used to communicate through paper

mail. The use of communication through such means would sometimes consume a lot of time resulting in late delivery. With the invention of various social applications we can now send messages in one click.

Social media is like a big online community where people can connect, share, and talk to each other. We can make friends, share pictures, videos, and messages, and stay updated on what's happening with people we know or things we're interested in. Examples include Facebook, Instagram, Snapchat, Twitter, and WhatsApp.

**Let's learn!**

Look up and research the figure, and answer the following questions:

a. What is LinkedIn?

b. Do you know LinkedIn provides learning platform?

c. How can we make the best LinkedIn profile?

## Pros of social media

a. **Global Connectivity:** It enables instant communication and connection globally, fostering relationships.

b. **Information sharing:** It provides a powerful platform for sharing information, and news, and promoting social causes.

c. **Business opportunities:** It facilitates marketing and brand exposure for businesses, reaching a vast audience.

d. **Diverse content:** It offers entertainment, educational content, and news, catering to a wide range of interests.

e. **Self-expression:** It provides individuals with a space for showcasing talents and expressing creativity

f. **Real-time updates:** It serves as a valuable source for real-time updates and information.

## Cons of social media

a. **Isolation and loneliness:** Despite connectivity, online interactions may contribute to feelings of isolation and loneliness.

b. **Misinformation:** The spread of fake news and misinformation is a significant drawback.

c. **Cyberbullying:** Social media can be a platform for cyberbullying, harassment, and privacy invasion.

d. **Mental health impact:** Excessive use of social media may lead to mental health issues, such as anxiety and depression.

e. **Productivity loss:** The addictive nature of social media can result in productivity loss.

f. **Body image issues:** Frequently seeing carefully selected content on social media can make people feel bad about themselves and lead to concerns about their body image and self-esteem.

*Activity 5.6*

**Activity Outcome -**

- Locate key contact information for the Cyber Bureau of Nepal, including its website, email, and contact number.

**Required Resources:** Personal computer or mobile phone with internet access, etc.

**Procedure:**

1. Open a search engine on your device (e.g., Google).
2. Type the keywords "cyber bureau of Nepal" into the search bar.
3. Identify and note the following details about the Cyber Bureau of Nepal:
   - **Link of the Website: www.nepalpolice.gov.np**
   - **Email Address of the Cyber Bureau of Nepal**
   - **Contact Number of the Cyber Bureau of Nepal**
4. Record your findings in the table below:

| | |
|---|---|
| Link of the website | www.nepalpolice.gov.np |
| E-mail of Cyber bureau of Nepal | |
| Contact Number of cyber bureau of Nepal. | |

**Result:** Group discussion on the importance of knowing how to contact cyber authorities and when it may be necessary to reach out to the Cyber Bureau.

## 5.6 Introduction to video conferencing tools

### Concept of online video meeting

> ***Fun fact!***
>
> The first video call was made between Pittsburgh Mayor Peter Flaherty and Chairman and CEO John Harper of Alcoa on June 30, 1970.

An online video meeting is a way of talking to each other over the internet where we can see and hear each other at the same time. It's kind of like having a chat face-to-face, but instead of being in the same room, we're looking at each other on a computer or phone screen.

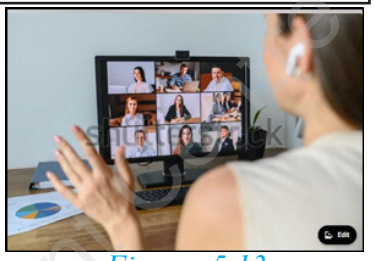


*Figure: 5.13*

When participating in online video meetings, one needs to observe proper etiquette, ethics, and dressing sense. The etiquette includes being punctual, muting microphones when not speaking to minimize background noise, and actively participating in discussions. Likewise, the ethics in online meetings include being respectful of others' opinions, avoiding disruptive behavior, and maintaining a professional attitude. Additionally, students should be mindful of their dressing sense, wearing suitable formal attire helps create a positive and respectful online environment, contributing to a successful virtual meeting experience.

### Examples of online video meetings

**Some examples of online video meetings are:**

a. **Zoom:** Zoom is a software that allows us to host and join online video meetings with up to 100 participants. It is a widely used video conferencing platform that enables users to conduct virtual meetings, webinars, and online collaboration. It also allows us to record and save the meetings for later viewing or sharing.

b. **Google Meet:** Google Meet is a video conferencing platform developed by Google. It is designed for virtual meetings, remote collaboration, and online communication.

c. **Microsoft Teams:** Microsoft Teams is a collaboration platform developed by Microsoft. It is designed to facilitate communication, collaboration, and teamwork within organizations. It also allows us to record and save the meetings to OneDrive or SharePoint.

*Activity 5.7*

**Activity Outcome -**

- Learn to set up, manage, and invite participants to a Google Meet session.

**Required Resources:** Computer or mobile device with internet access and a Google account, etc.

**Procedure:**

**Step 1: Sign in/create an account.**

1. Visit Google Meet.
2. Sign in using your Google account or create a new account if you do not have one.

**Step 2: Start a meeting.**

1. Click on "Start a Meeting" or "New Meeting" to begin a session.
2. Allow the browser or app to access your camera and microphone if prompted.
3. Click on "Join Now" to officially start the meeting.

**Step 3: Share the meeting link.**

1. Once the meeting has started, copy the meeting link from the screen.
2. Share the link with participants through email or messaging.
3. Alternatively, use the option to invite participants directly by entering their email addresses.

**Step 4: Add participants during the meeting.**

1. During the meeting, click on the "Participants" icon.
2. Select "Invite" to add participants by their email addresses.

**Result:** Teacher-moderated reflection on the ease of setting up and managing online meetings, with tips for effective virtual communication and meeting management.

## 5.7 Concept of blogs and its features

### Concept of blog

Some of us have a hobby of exploring different places and capturing those special moments on our cameras or smartphones. Those captured videos or photos can be turned into a blog and we can upload them on different social media platforms and websites. We can also gain widespread fame and capital by carrying out such activities.

A blog is like an online journal or diary where people can share their thoughts, ideas, experiences, or information on a specific topic. Blogs can cover a wide range of subjects, such as personal stories, hobbies, travel, technology, or any other area of interest.

### Features of blog

**Some of the features of a blog are:**

a. **Posts:** Blogs consist of individual entries called posts. Each post focuses on a specific topic or idea.

b. **Authorship:** Each post is written by an author, who is usually the person maintaining the blog. Authors share their thoughts, experiences, or knowledge.

c. **Comments:** Readers can leave comments on blog posts, creating a space for interaction and discussion.

d. **Categories:** Blogs often organize posts into categories based on topics. This helps readers find content related to their interests.

e. **Archives:** Blogs have archives that allow users to access older posts. It is like a digital history of the blog's content.

f. **Tags:** Bloggers use tags to label posts with keywords, making it easier for readers to search for specific topics.

g. **Images and multimedia:** Blog posts can include images, videos, and other multimedia elements to enhance content and engage readers.

h. **Social sharing:** Blog platforms usually include options for readers to share posts on social media, spreading the content to a wider audience.

i. Subscribe: Readers can subscribe to a blog to receive updates when new posts are published.

## Exercises

**1. Write full forms of the following abbreviations.**

a) ARPANET b) HTML c) SMTP

d) DNS e) ISP f) HTTPS

**2. Choose the correct answer.**

a) Which of the following is a correct term that describes the multimedia content and interlinked collection of hypertext documents on the Internet?

i. Internet ii. World Wide Web (WWW)

iii. Web Browser iv. Search Engine

b) The main purpose of the internet is to …… .

i. send emails ii. play games

iii. access and display hypertext documents on the web

iv. edit images

c) Which of the following is a secure version of HTTP, providing encrypted communication?

i. HTTP ii. HTTPS iii. FTP iv. SMTP

d) …….. is the primary purpose of a Virtual Private Network (VPN).

i. Securely connect to the internet and access resources

ii. used for online meetings.

iii. Play online games

iv. Edits images online

e) The main purpose of social media is …….

i. editing documents

ii. connecting, sharing, and talking to others online

iii. playing online games

iv. sending emails

f) Which of the following is not an example of a VPN?

i. NordVPN ii. Google Meet VPN

iii. ExpressVPN iv. Super unlimited proxy

g) Which of the following applications are used for online meetings?

i. Google Meet ii. Zoom

iii. Skype iv. All of the above

h) …….. are individual entries called in a blog?

i. Pages ii. Sections

iii. Posts iv. Articles

i) Why is blogging used?

i. Share blog posts on social media

ii. Allow readers to interact and discuss

iii. Categorize posts based on topics

iv. All of the above.

## 3. Write short answers to these questions.

a. What is the Internet? Write its two uses.

b. What is a WWW? List out the components of www.

c. What is a web browser? Give an example.

d. Write down some examples of online meeting platforms.

e. What are Web Protocols? List out the examples of the Web Protocols?

f. Write down some uses of email?

## 4. Write long answers to these questions.

a. Define Search Engine. What are the uses of search engines? Write with an example.

b. Explain the concept of Remote Login.

c. Write down the difference between HTTP and HTTPs.

g. What is social media? List out the uses of social media with its examples.

h. Define blogs. Write down the advantages and disadvantages of blogging.

## 5. Lab Activities

**a) Apply digital signature and secure email using Gmail, Outlook or similar mail system.**

**Step:**

1. Open your Gmail account.
2. Go to Settings.
3. Click on See all settings.
4. Click on the General.
5. Go to Signature. There you saw the option create new
6. Click on Create New option – Add new signature – Type Signature name – Click on Create option.
7. After that on the left side the content area is present on the content area Type the Name, Address, and own positions and organization name too.

8. After the area is filled then go to the bellow of the page and enter on the save change.

**b) Demonstrate the mechanism of searching for different learning materials from the internet.**

**Step:**

1. Open your web browser: Launch your preferred web browser (e.g., Google Chrome, Mozilla Firefox, Safari).

2. Go to Search engine: Navigate to a search engine's website, such as Google (for this demonstration, we'll use Google).

3. Enter your search query: In the search bar, type your learning query. For example, let's say you're looking for materials on "Python programming for beginners."

4. Review search results: Press "Enter" or click on the search icon to view the search results.

5. Explore results: Browse through the search results to find relevant learning materials. Pay attention to the titles and snippets provided.

6. Refine your search: If the initial results are not what you're looking for, consider refining your search query. For instance, you might add specific terms like "Python programming tutorials" or "Python basics PDF."

7. Use filters: Utilize search engine filters to narrow down results. You can filter by type (videos, images, news) or time (past hour, past 24 hours, past week) to get the most relevant content.

8. Visit websites: Click on the search result links to visit websites that seem promising. Look for reputable sources such as educational institutions, official documentation, or well-known learning platforms.

9. Download or save materials: Once you find the desired learning materials, you may download PDFs, videos, or other resources as needed. Follow the specific instructions on the website to access or save the content.

10. Explore additional resources: Scroll through the search results and explore different websites to gather a variety of learning materials. Consider bookmarking useful websites for future reference.

**c) Creating a profile on other platforms:**

**Steps:**

1. Choose the platform: Identify the social media platform you want to join (e.g., Instagram, Twitter).

2. Visit the platform's website: Go to the official website of the chosen platform.

3. Sign up: Follow the sign-up process, providing the required information.

4. Customize your profile: Add a profile picture, bio, and other relevant details.

5. Connect with others: Find and connect with friends, family, or other users. Explore and Contribute: Start exploring content, following accounts, and contributing to the platform.

**d) Create a virtual meeting using any application such as Zoom, Meet, or Teams.**

**Zoom**

**Step 1: Sign in/create an account.**

1. Visit the Zoom website (https://zoom.us/).

2. Sign in if you have an account or create a new account.

**Step 2: Schedule a meeting.**

1. After logging in, click on "Schedule a New Meeting" or "Host a Meeting" in the top right corner.

2. Fill in the meeting details, such as topic, date, time, and other settings.

3. Click "Save" to schedule the meeting.

**Step 3: Invite participants.**

1. Once the meeting is scheduled, you'll see options to invite participants.

2. Send invites via email, copy the invitation link, or use other options.

**Step 4: Start the meeting.**

1. At the scheduled time, return to the Zoom website or open the Zoom app.

2. Click on "Start" to begin the meeting.

***Project work***

**Project Outcome -**

- Learn to create a YouTube channel, upload videos, and engage with the YouTube community.

**Required Resources:** Computer or mobile device with internet access, Google account, video files, and images for channel customization, etc.

**Procedure:**

**Step 1: Sign in to Google.**

- Go to Google's sign-in page.
- If you already have a Google account, sign in using your credentials.
- If you don't have a Google account, click on "Create account" and follow the steps to set up a new Google account.
- Remember that YouTube requires a Google account since it is a Google-owned platform.

**Step 2: Go to YouTube.**

- Open your web browser and go to YouTube.

- Sign in with your Google account if you aren't signed in automatically.

**Step 3: Create a channel.**

- Click on your profile picture in the top right corner of the YouTube homepage.
- From the dropdown menu, select "Create a channel."
- You'll be prompted to set up your channel; click on "Get Started" to begin.

**Step 4: Set up your channel.**

- Choose a unique channel name that reflects the type of content you plan to upload.
- Upload a profile picture that visually represents your channel (e.g., a logo or relevant image).
- Customize your channel settings by following the prompts. This includes privacy settings, your display name, and any other initial configuration options.

**Step 5: Upload a video.**

- Prepare a video you want to share on your channel. This can be a short introduction, tutorial, or any content relevant to your theme.
- Click on the camera icon with a plus (+) at the top of the page and select "Upload video."
- Select the video file from your device and follow the prompts to add a title, description, and tags to improve visibility.
- Review YouTube's video settings for privacy (Public, unlisted, private), age restrictions, and monetization if applicable.

**Step 6: Customize your channel.**

- Go to the YouTube Studio (found in your profile dropdown) to access channel customization options.
- Add a channel description to inform viewers about the content they can expect.

- Customize your channel layout (e.g., featured sections) to make it easy for visitors to browse your videos.
- Add links to your social media profiles or website by going to the "Basic Info" section in the customization menu.
- Upload channel art (banner) for a visually appealing homepage.

**Step 7: Start sharing.**

- Share your video on social media platforms or with friends to increase visibility.
- Engage with the YouTube community by responding to comments, liking relevant content, and collaborating with other creators.
- Use the YouTube Studio to monitor viewer statistics and get insights into your audience.

**Result:**

Reflect on your experience setting up the channel, customizing it, and sharing content. Consider what types of videos you want to produce in the future and how you will engage with your audience.

# Cyber Security and Digital Citizenship

***Let's think!***

a) Discuss why cybersecurity is essential for individuals and businesses in Nepal.

b) Explain how you think your online activities impact your privacy and security.

c) Discuss the responsibilities that come with being a digital citizen.

## 6.1. Concept of cybersecurity

Cybersecurity refers to the practice of protecting computer systems, networks, and digital information from theft, damage, unauthorized access, and various forms of cyberattack. Every day, there are a lot of cyberattacks happening online. These attacks can target big companies, regular people, and small businesses. They can cause a lot of damage and can happen to anyone, anywhere in the world.


*Figure 6.1*

People and businesses can use different tools and methods to keep their digital information safe and defend against cyber threats. There are various types of cybersecurity, like network

security, application security, information security, and operational security. It's crucial for everyone to realize how important cybersecurity is and to make sure they're using all the methods available to defend against cyberattacks.

***Let's check!***

Which of the following is not cyber security?

a. Network security
b. Application security.
b. Information security
c. Quantum security.

## 6.2. Concept of cybercrime

Cybercrime simply refers to the criminal activities carried out by means of computers or the Internet. When someone uses a computer to do bad things, it can hurt other people's safety and money. Cyber-attacks come in different forms, like stealing someone's identity, tricking them with fake emails, stopping websites from working, stealing information, putting bad software on computers without permission, or taking over social media accounts to post bad stuff. In Nepal, the ETA (Electronic Transaction Act), 2063 handles controlling the cybercrime related issues and helps in drafting and implementing laws against cybercrime.

## 6.3 Prevention methods from cybercrime

Cybercrime is a serious threat we deal with daily, and even cybersecurity experts can struggle to fix everything. That's why it's wise to try to stop cybercrime before it starts. Here are some steps to prevent it:

### a. Use a strong password.

Password is a set of alphanumeric characters usually used to conform users' identity. Password is just as we secure our homes with locks, safeguarding our digital assets requires strong passwords to protect valuable data from threats. Strong, unique passwords should be used for online accounts and devices, changed regularly, and not shared. These passwords, along with usernames, are called credentials and are essential for accessing email, websites, financial accounts, and more. Keeping passwords secure helps prevent unauthorized access and misuse of personal information.


*Figure 6.2*

**Here are some tips to ensure our passwords are secure and strong:**

a. Do not use a sequence for example 1234 or abcd which can be easily guessed.

b. Try to include numbers, symbols, and both uppercase and lowercase letters.

c. Avoid using words that can be found in the dictionary. For example, words such as **admin** and **password** are considered very weak passwords.

d. Use a longer password. Your password should be at least six characters long, although for extra security it should be even longer.

e. Use Combination of Upper Case and Lower case letters including Numbers and special characters.

**Fun fact!**

"qwerty", "123456" and "password" are like the evergreen rock stars of the digital world. Despite being about as secure as leaving your front door wide open in a busy city, they still manage to rock the charts as the most used passwords. It's like they're the classics that never go out of style, even though security experts are constantly screaming for a change in tune!

## Multi-factor authentication (MFA)

Multi-factor authentication (MFA) is an authentication method that adds an extra layer of security by requiring users to provide two or more verification methods to access an online account or application. It's considered one of the most important security measures because even if someone knows your password, they would still need access to your mobile phone or require your fingerprint or face scan to gain access to your account.


*Figure 6.3*

**Types of multifactor authentication**

**1. Thing you know**

a. Password b. Pin

2. **Things you have**

a. Badge b. Smartphone

3. **Things you are**

a. Biometrics such as fingerprint b. Voice c. Retina of eye

### Software updates

Keeping your operating system and internet security up to date is extremely important. Make sure to regularly update your applications and operating system, such as Windows, Android, iOS, Linux, and others, with the latest system updates. Turning on automatic updates can help prevent potential attacks on outdated software.

### Authentication

Identification of an individual usually based by username and password is known as authentication. Having a strong authentication system in place makes it difficult for others to access your computer system without your permission.

### Use of Firewall

A firewall is a security tool that filters network connections to block unauthorized users from accessing a device, network, or private data. It can come in different forms such as hardware, software, or as a feature within an operating system like Windows Firewall or iOS Firewall. Enabling a firewall on a device helps protect it from unwanted internet traffic. This helps prevent computers from viruses.

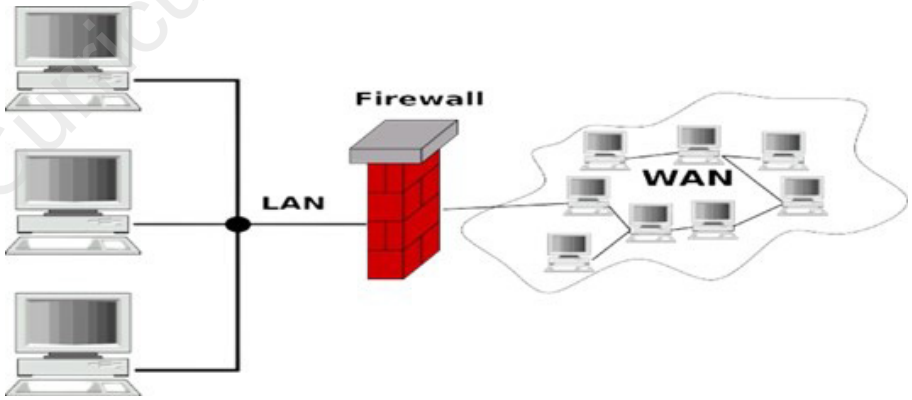



*Figure 6.4*

*Activity 6.1*

**Activity Outcome**

- Gain hands-on experience with antivirus software and firewall settings, understanding their functions in protecting devices from threats.

**Required Resources:** Computer or device with antivirus software and firewall options (such as Windows Defender or a third-party antivirus, Internet connection (if needed for updates) Projector or screen (optional, for class demonstration), etc.

**Part 1: Understand use of antivirus software**

**Procedure:**

**Open the antivirus program:** You should open the antivirus software installed on the classroom's device. If there isn't one, they may use Windows Defender or a free trial of an antivirus program that you can use.

**a) Demonstrate a quick scan:**

- Run a quick scan of the system and observe any threats detected.

**b) Full scan option:**

- See the difference between a quick scan and a full scan.

**c) Review and quarantine threats:**

- If any threats are detected, review the results and quarantine or removal the viruses.

**d) Update the antivirus:**

- Update the virus definitions manually if and understand that keeping software updated is crucial for protecting against new threats.

**Part 2: Understand use of a firewall (software or hardware)**

**procedure:**

**a) Access firewall settings:**

- Open firewall settings on the device (e.g., in Windows, access "Windows Defender Firewall" through Control Panel).

**b) Configure basic firewall rules:**

- Allow or block applications from accessing the network using firewall rules.

- For example, block an application temporarily and show how to unblock it afterward.

**c) Demonstrate inbound and outbound rules (if available):**

- Create custom inbound or outbound rules to control access to specific services or IP addresses.

**d) Enabling/disabling the firewall:**

- Use the option to turn the firewall on or off but explain the risks of disabling it and emphasize the importance of keeping it enabled.

**Result:** Moderated reflection and discussion about your experience using these tools.

## 6.4 Safe web browsing techniques

In today's digital world, the internet is no longer a safe place for individuals or businesses. There are a number of websites hacked daily. Apart from these, cyberattacks and ransomware attacks also happen daily. So, it is crucial to have a good knowledge about cybersecurity practices. To prevent such attacks, here are some safe browsing techniques that help minimize cyber threats:

### a. Use a secure connection

*Figure 6.5*

A secure connection refers to the private communication between your device and a server using encryption. Encrypted communication prevents anyone from listening to or changing your data. You can also check if a connection is secure by looking for a lock icon or a https prefix in the address bar of a website. Using a virtual private network (VPN) from trusted providers like NordVPN, Proton VPN, and Mozilla VPN can also protect the user's privacy to some extent.

### b. Use a secure browser

*Figure 6.6*

A secure browser protects user privacy and security while surfing the web. The secure browser has features such as private browsing, tracking protection, Ad-blocking, password management, etc. Some examples of secure browsers are Mozilla Firefox, Google Chrome, Safari, Microsoft Edge, Opera, Brave, etc.

### c. Use a secure search engine

A secure search engine is a web-based tool that allows users to search for information on the internet while prioritizing privacy and security. It may also filter out harmful or inappropriate websites from the search results. Some examples of popular search engines are Google, Bing, and DuckDuckGo.


*Figure 6.7*

> ***Do you know?***
>
> Google reigns as the undisputed champion of search engines across the globe. Additionally, a whopping 65.76% of internet users prefer to surf the web using Google Chrome, making it the top choice for web browsing worldwide.

### d. Use secure websites.

A secure website prioritizes user privacy and security during visits by incorporating various features such as encryption, authentication, and verification. To determine if a website is secure, look for indicators like a lock icon or "https" prefix in the address bar, indicating that the connection is encrypted. Additionally, review the website's privacy policy, terms of service, and user reviews to check how it handles your data and ensures security measures are in place.


*Figure 6.8*

> ***Activity 6.2***
>
> **Activity Outcome -**
>
> - Demonstrate safe browsing techniques by following recommended security practices.
>
> **Required Resources:** Computer with internet access, browser, VPN application (if needed), ad blocker software or extensions, instructional materials etc.
>
> **Procedure:**
>
> Your teacher will introduce various safe browsing techniques and demonstrate each step.

You will work individually or in pairs to practice the following techniques:

a) **Check browser version and updates:** Verify that the browser is up to date and perform updates if necessary.

b) **Exercise caution with downloads:** Review and discuss best practices for downloading files and applications securely.

c) **Clear browser cookies:** Learn to delete browser cookies to maintain privacy.

d) **Use an Ad blocker:** Install or enable an ad blocker to protect privacy from invasive ads.

e) **Create strong passwords and enable MFA:** Practice creating secure passwords and setting up Multi-Factor Authentication (MFA) where available.

f) **Use a trusted VPN (if required):** Learn how to set up and connect to a reliable VPN for added security when needed.

**Result:** Moderated reflection and discussion for feedback on each technique practiced, addressing questions and reinforcing safe browsing practices.

## 6.5 Concept of Digital Citizen

A Digital Citizen is someone who knows how to use the internet and digital technology responsibly. They actively engage in the digital society by creating, collaborating, and benefiting from digital content and resources. To be a good digital citizen, consider following these guidelines:

*Figure 6.9*

a. Think before you post on social media platforms. Be mindful of what you share and its potential impact.

b. Avoid sharing personal information excessively to reduce the risk of identity theft.

c. Use multiple search engines to limit tracking of your search history and enhance privacy.

d. Regularly change passwords to make it harder for cybercriminals to access and steal your data.

e. Report any unlawful or inappropriate behavior to the authorities to help maintain a safer online environment.

## 6.6 Concept of Netiquette and online behaviors

Netiquette, derived from "network" and "etiquette," refers to a set of guidelines for appropriate online behavior, while internet ethics focuses on the proper use of online resources in digital communities. It's a recommended rule for both personal and professional internet usage, guiding users on platforms like Facebook, Discord, and online forums. Just like in-person manners, online manners are equally important in family, society, and professional settings. Despite challenges in online communication, the fundamental principle of treating others with respect remains constant.

**Here are some examples of good netiquette and online behavior:**

1. Have respectful communication.
2. Avoid offensive content.
3. Respect privacy, don't share personal info.
4. Avoid spamming or flooding.
5. Give credit when sharing others' work.
6. Think before posting.
7. Report abusive behavior.
8. Be open to constructive criticism.
9. Maintain a positive presence online.

## 6.7 Concept of digital footprint and privacy in online

Just like when you walk on sand or mud, each step you take leaves an impression, indicating where you've been and what you've done. Your digital footprint is like leaving footprints as you walk through the internet. It shows where you've been and what you've done online.

*Figure 6.10*

A digital footprint is defined as an information record generated by an individual's online activities. These activities include online searches, social media visits, online shopping, phone calls, sending emails, etc. A digital footprint is sometimes called "**A Digital Shadow**" or "**Electronic Footprint**". Every internet activity creates some sort of digital footprint that can be used to trace a person's online activities as well as their devices. There are two types of digital footprints, and they are:

I. **Active digital footprints:** It comprises information that an internet user knowingly leaves behind. Since the user knowingly provided information, the user is also informed of the digital footprint they have left behind. Social media posts, phone calls, and emails are a few examples of active digital footprints.

II. **Passive digital footprints:** These consist of information that an internet user unknowingly leaves behind. These kinds of footprints are difficult to monitor and manage since they might be obtained without the approval of the user. Web searches, online shopping, location data, and fitness trackers are some examples of passive digital footprints.

*Activity 6.3*

**Activity Outcome -**

- Understand the concept of digital footprints by gathering and analyzing information about digital identities and device usage.

**Required Resources:** Notebook or digital spreadsheet for data collection, internet-enabled device, etc.

**Procedure:**

Your teacher will divide you in pairs or small groups, and you will collect digital footprint information from classmates:

a) You will have to collect the following information:

- **Name:** First and last names.
- **Email ID:** Primary email address.
- **Device:** Types of devices they frequently use (e.g., smartphone, laptop).
- **Social Media Username:** Main usernames or handles used on social media platforms.

b) Complete the following data table:

| Name | Email ID | Device | Social Media Username |
|---|---|---|---|
| | | | |
| | | | |

c) **Privacy Reminder:** Students should obtain permission from their peers before collecting and recording their information and handle this data responsibly.

**Result:** Moderated discussion on digital footprints, privacy, and online safety, encouraging students to reflect on how their information is shared and used online.

Online privacy is a fundamental right in today's digital world, focusing on safeguarding one's private life, home, and communication. Specifically, it pertains to the protection of an individual's personal information stored on the internet, including data on digital devices and services like web searches and social networking sites. Privacy concerns are a significant issue in the modern era, with digital footprints serving as the primary avenue for cybercriminals to target private data. While there are advantages to digital connectivity, such as enhanced communication and access to information, there are also disadvantages, including the risk of privacy breaches and unauthorized access to personal data.

*Activity 6.4*

**Activity Outcome -**

- Conduct research and identify differences between regular browsing mode and incognito mode.

**Required Resources:** Computer or mobile device with internet access, etc.

**Procedure:**

Your teacher will divide you in pairs or small groups, and you will collect digital footprint information from classmates:

a) **Research:** Individually or in pairs, you will conduct research on incognito mode, exploring its purpose and functionality.

**Focus points should include:**

- How incognito mode works.
- Key differences from regular browsing mode.
- Limitations of incognito mode in terms of privacy and security.

**b) Practical Exploration:**

- Open a browser in regular mode and note down key features, visible history, and saved data (such as cookies, saved passwords).
- Switch to incognito mode and compare differences, focusing on what data is saved, visibility of browsing history, and session behavior.

**c) Complete the Following Comparison Table:**

| Regular Browsing Mode | Incognito Mode |
|---|---|
| Saves browsing history and cookies | Doesn't save browsing history |
| Retains login sessions and form data | Ends login sessions after closing |

**Incognito Mode Shortcut Key:**

Shortcut keys for incognito mode:

Shift + Ctrl + N

**Result:** Moderated discussion on the advantages and limitations of using incognito mode, with students sharing observations and insights.

**Advantages:**

**a. Health monitoring**

Smart watches and health applications utilize digital footprints to monitor and analyse users' fitness and health data, aiding in better health management.

**b. Social connections**

Digital footprints help build and foster connections with others, particularly on social networking platforms, facilitating communication and relationship-building.

**c. Targeted advertising**

Companies can target advertisements based on customer browsing behaviors, improving the relevance of ads and potentially increasing sales effectiveness.

**d. Law enforcement support**

Police can utilize online records to aid in investigations and prevent illegal activities, leveraging digital footprints to gather evidence and ensure public safety.

## Disadvantages

**a. Security vulnerabilities**

Digital footprints expose individuals to security risks, enabling others to track their online activities with ease, potentially leading to privacy breaches and identity theft.

**b. Legal and ethical concerns**

The use of digital footprints raises legal and moral questions regarding the handling of sensitive information, prompting debates over privacy rights and data protection laws.

**c. Exploitative business practices**

Businesses may profit from selling user data without adequately compensating individuals for the value of their private information, raising concerns about fairness and ethical business practices.

**d. Cybercriminal exploitation**

Hackers can exploit the data contained in digital footprints to perpetrate fraud and other criminal activities, posing significant risks to individuals' financial and personal security.

**e. Spying and online abuse**

Detailed digital footprints can be used for spying and online abuse, jeopardizing individuals' safety and exposing them to various forms of cyber harassment and exploitation.

**Here are some practical steps you can take to protect yourself from the risks of digital footprints:**

a. **Know security rules:** Understand how your personal information is handled by websites and services you use.

b. **Update privacy settings:** Adjust privacy settings on social media to control who can see your information.

c. **Secure devices:** Use strong, unique passwords for each account and enable extra security measures like multi-factor authentication.

d. **Remove old accounts:** Delete unused online profiles to reduce exposure of personal information.

e. **Update software:** Keep your devices and apps updated to fix security flaws.

f. **Disable location tracking:** Turn off location services on apps and devices when not needed to prevent constant tracking.

**Do you know?**

If a person publishes or displays material against morals, etiquette, or hatred on a computer, internet, and other electronic media, the culprit can be punished with a fine of 1 lakh rupees or imprisonment for up to 5 years or both as per Electronic Transactions Act 2063.

## Exercises

**1. Write full forms of the following abbreviations:**

a. MFA b. 2FA c. ETA

d. VPN e. DDoS f. XSS

**2. Choose the correct answer.**

a. The full form of ETA is _____.

i. Electronic Transaction Act

ii. Electric Transmission Action

iii. Electronic Transform Act

iv. Electronic Transaction Awareness

b. Who registers complaints of cybercrime in Nepal?

i. Nepal Telecommunication Authority

ii. Local police stations

iii. Cyber Bureau of Nepal.

iv. The Ministry of Home affair.

c. _____ is not an example of antivirus software.

i. Kaspersky ii. Firewall

iii. Avast iv. Norton 360

d. Which of the following is not a cybercrime?

i. Accessing child pornography contents ii. Authentication

iii. Phishing iv. Brute force attack

e. Cyber law is commonly known as _____.

i. law of the internet ii. digital Legislation

iii. prevention method of cyber crime iv. ways of using the internet

f. The Electronic Transactions Act of Nepal also consists of _____.

i. constitution ii. cyber laws

iii. computer software iv. transaction records

g. The combination of username and _____ is basically known as a user credential.

i. password ii. security question

iii. biometric iv. phone number

h. When was the Electronic Transactions Act published?

i. 2063 B.S ii. 2053 B.S iii. 2062 B.S iv. 2058 B.S

## 3. Write short answers to these questions.

i. What is Cybersecurity?

ii. Explain Cyber Law by relating it with the constitution.

iii. What is a Firewall? Why is it different from antivirus software?

iv. You use digital devices on a daily basis. What do you know about digital society?

v. What is malware? Write down the different types of malwares.

## 4. Write long answers to these questions.

a. Define malware and describe its types.

b. Explain about digital footprint and write the advantages and disadvantages of it.

c. Explain the safe browsing techniques.

d. Define Cybercrime. Explain the prevention method from cybercrime.

e. Explain about netiquette guidelines.

***Project work***

**Project Outcome -**

- Conduct research on cybercrime trends, explore real-world case studies, and develop awareness of prevention strategies.

**Required Resources:** Internet-enabled device (laptop, tablet, or computer), access to online research resources or databases, notebook or digital document for notes, etc.

**Procedure:**

Your teacher will divide you in pairs or small groups

**a) Topic Selection and Group Division:**

- The teacher will assign you to groups, with each group focusing on one specific type of cybercrime, such as phishing, identity theft, ransomware, social media hacking, or online scams.

**b) Research Phase:**

- Each group will conduct online research to find recent case studies, articles, and reports about their assigned type of cybercrime.

- Gather information on:
    - Definition and characteristics of the specific cybercrime
    - **Real-life case study:** An example where the cybercrime impacted individuals or organizations
    - **Consequences:** The effects on victims and organizations, such as financial loss, reputational damage, or data breaches
    - **Prevention Strategies:** Methods and tools used to prevent or reduce risks related to the cybercrime

**c) Documentation and Analysis:**

- Organize findings in a document or presentation with the following sections:
    - **Introduction:** Brief overview of the assigned cybercrime
    - **Case Study:** Description of a real-world example, detailing the incident and its impact
    - **Consequences:** Analyze how the cybercrime affected the victims and the broader implications
    - **Prevention:** Outline strategies, tools, and best practices to avoid falling victim to this type of cybercrime.

**d) Create a Report and Prevention Guide:**

- Compile the research into a concise report. Include visuals, like charts or infographics, to show trends in cybercrime.
- Prepare a one-page prevention guide summarizing the key prevention tips for the specific cybercrime. This guide should be practical and easy for classmates to understand.

**e) Presentation and Class Discussion:**

- Each group will present their findings to the class, explaining the nature of their assigned cybercrime, its impact, and effective prevention strategies.
- After each presentation, the teacher will facilitate a class discussion on the broader importance of cybersecurity and personal responsibility online.

**Result:** After the presentations, students will participate in a moderated discussion,

led by the teacher, to reflect on key findings from each group's research. This discussion will cover:

- **Trends of cybercrime:** Observing patterns across different types of cybercrimes and identifying which crimes are most prevalent today.
- **Impact on individuals and organizations:** Discussing the social, financial, and emotional toll cybercrimes can take on victims.
- **Effectiveness of prevention strategies:** Evaluating the practicality and success of different prevention methods, including which strategies may be most applicable in students' own digital lives.
- **Personal reflection on digital safety:** Encouraging students to share how the research has shaped their understanding of cybersecurity and what actions they might take to improve their own online safety.

# Concept of Programming

*Figure 7.1*

***Let's think!***

a) What do you think programming is?

b) Why do you think programming is important in today's world?

c) Define algorithm.

Just like people use languages like Nepali, Newari, Maithili, Bhojpuri, English, and French to talk to each other, we use programming languages to communicate with computers. These languages are what computers understand. With programming languages, we can make all sorts of computer software. Everything from operating systems like Windows and Linux to application software like Office Package and web browsers is made using programming languages.

**Fun Fact!**

The first commercially available language was FORTRAN (FORMULA Translation), developed in 1956 (first manual appeared in 1956, but first developed in 1954) by a team led by John Backus at IBM.

## 7.1 Introduction to programming languages

A programming language is a language consisting of a set of instructions provided by the user that tells a computer what task to do and how to do it. Python, Java, PHP, C++, etc. are some examples of popular programming languages.

**Programming:** It is the process of providing detailed instructions to a computer step by step to do specific tasks.

**Programmer:** A programmer is a person who is involved in writing computer programs.

**Syntax:** The rule for writing commands is called syntax.

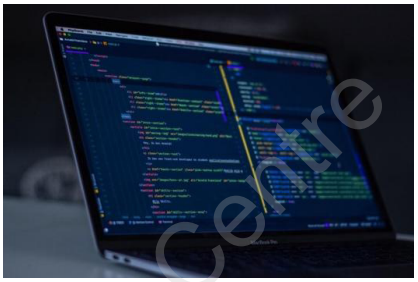


*Figure 7.2*

**Let's check it!**

Which is not a programming language?

a. C b. HTML

c. Javad d. Python

There are two main types of programming languages: high-level and low-level.

### High-level programming language

a. These languages use structures and commands that resemble human language, making them easier for people to understand.

b. Programs written in high-level languages are called program code, and they're readable to humans but not to computers.

### Low-level programming language

a. Low-level languages mainly consist of 0s and 1s, which are directly understood by computers.

b. Programs are written in the form of machine code, which is difficult for humans to comprehend without specialized knowledge.

To make high-level language programs understandable to computers, they need to be converted into low-level programming languages. This conversion is done using language translators like interpreters and compilers.

## Programming tools: Flowchart and Algorithm

An algorithm is like a recipe for computer programming. It's a step-by-step guide that tells the computer exactly what to do to solve a problem or computer task. Just like a recipe has instructions for making a dish, an algorithm has instructions for the computer to follow. A flowchart is a diagram that shows the steps of an algorithm in a visual way. It uses different shapes to represent different types of actions and arrows to show the direction of the process. Think of it as a map that guides you through the steps of the algorithms. An algorithm and a flowchart are tools in programming. They are helpful for building logic to solve problems before programming.

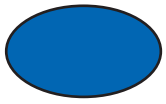
### Algorithm

- An algorithm is a set of step-by-step instructions designed to solve a specific problem or perform a particular task.
- It begins with "start" and ends with "stop."
- The instructions are written in simple, general language (like spoken languages).

### Flowchart

- A flowchart is defined as the pictorial and graphical representation of an algorithm.
- It utilizes shapes (oval, rectangle, parallelogram, etc.) to depict various elements.
- The shapes have distinct meanings in a flowchart.
- The flowcharts correspond to algorithms, visually representing their steps and structure.

**The table below shows shapes used in flowchart and their meaning:**

| Name | Shapes | Meaning | Uses |
|---|---|---|---|
| Oval |  | Start/End | Marks the start and end points of a flowchart. |
| Parallelogram | | Input/Output | Handles input/output, taking user input and displaying output. |

| | | | |
|---|---|---|---|
| Rectangle |  | Processing | Represents processing or calculation steps in the flow. |
| Arrow |  | Direction of flow | Shows the direction of flow or sequence in the flowchart. |
| Diamond |  | Decision/ Condition Check | Checks conditions and represents decision-making points. |
| Circle |  | Connector | Act as a connector joining components of the flow chart. |

**Example of algorithm and flowchart**

Problem: To display the greater number between two numbers

**Algorithm:**

**Step 1:** Start

**Step 2:** Read the first number (num1)

**Step 3:** Read the second number (num2)

**Step 4:** Is num1 > num2?

Yes: Display num1 is greater.

No: Display num2 is greater.

**Step 5:** Stop

**Flowchart:**

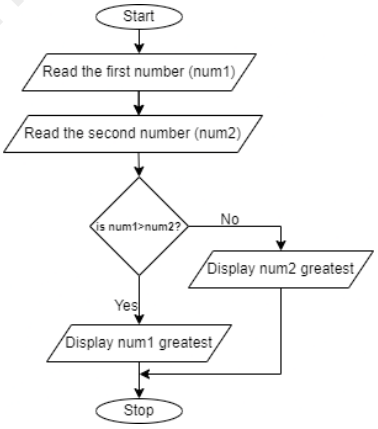



*Figure 7.3*

Here, the problem is to find the greater number between two numbers using programming logic. In order to solve the problem, firstly, the programming logic is written in terms of algorithm and visualized through the corresponding flowchart. The logic involves:

a. Getting two numbers num1 and num2, from the user

b. Checking if num1 is greater than num2 and making a decision on its basis

c. Displaying the result is to the user

*Activity 7.1*

**Activity Outcome -**

Able to list down step and draw

**Required Resources:** Chart Papers, Meta Cards, Cardboards, Image, etc.

**Procedure:**

Create an algorithm and flowchart to illustrate the process of making tea.

**Result:** Show the drawing to the teacher for further review

## Coding, testing and debugging

1. **Coding:** Coding is the process of writing instructions in a programming language to create software or applications. These instructions, known as code, tell the computer exactly what tasks to perform.

2. **Testing:** Once the code is written, testing is conducted to ensure the program functions as expected. This involves running the program to identify any errors or bugs and verifying that all parts of the program work correctly.

3. **Debugging:** If testing reveals any errors or bugs, the process of debugging begins. Debugging involves locating the source of the problem in the code, understanding why it occurred, and then making the necessary corrections to fix it.

These three different steps are crucial for developing software that is reliable and functions smoothly.

## Compiler and Interpreter

Imagine two friends: one speaks Nepali only, and the other speaks Sanskrit only. They want to chat but need someone who understands both languages to translate for them. Similarly, in computer programming, compilers and interpreters act as translators for computers.

## Compiler

For a program written in high level languages, we must convert it into its equivalent machine language program before we can execute it on a computer. We can convert high-level programming languages with the help of a translator program called

a compiler. A compiler is a translator of computer programs that translate the entire high-level language into machine level programming language in a single operation on a computer. Some of the most commonly widely used compiled languages are JAVA, C#, C, C++, FORTRAN, etc.


*Figure 7.4*

**Some examples of compiler are:**

a. Microsoft visual studio

a. GNU compiler collection (GCC)

a. Common Business oriented languages (COBOL)

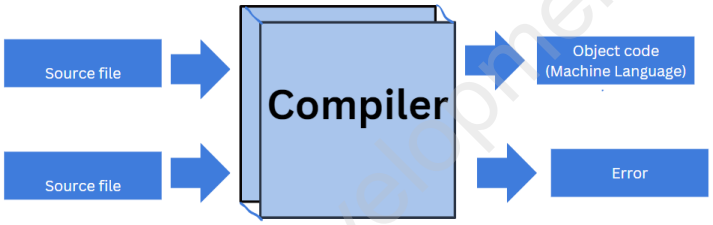



*Figure 7.5*

## Interpreter

Imagine, you're telling a friend how to cook in a language they don't fully understand. You'd have to explain each step one by one, and they would follow your instructions steps by steps. An interpreter works similarly with computer code. It takes one instruction (line of code), translates it, and then moves on to the other line, rather like explaining each step of the cooking process as your friend does it. This method can be slower than having all the instructions translated at once, but it's useful because it can handle changes more easily. An interpreter is also more portable than a compiler as it is not processor dependent, you can work between hardware architectures. The most frequently used interpreted language is QBASIC. It required less memory.

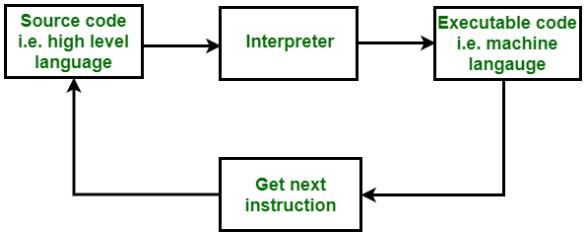



*Figure 7.6*

**Some examples of interpreter are:**

a. JavaScript

b. Ruby

c. List processing (LISP)

d. MATLAB

e. Python

Compiler and interpreter translate program code into machine code. They convert high-level programming language (user-friendly language) into low-level programming language (language understood by computers, like 0s and 1s). While both are language translators, they work differently.

A compiler translates the entire program at once, producing an executable file that can run independently. An interpreter, on the other hand, translates code line-by-line as it executes, without creating an independent file. Both tools serve the same purpose: enabling computers to understand and execute human-written code.

**The table below shows the difference between compiler and interpreter.**

| Compiler | Interpreter |
|---|---|
| A compiler reads and translates program code to machine code all at once. | An interpreter reads and translates program code to machine code line by line. |
| If there's an error in the program code, it displays errors at the end. | If there is an error in the program code, it displays while running the program. |
| Compiled program's execution time is comparatively faster. | Interpreted program's execution time is comparatively slower. |
| Compiled programs have separate files to store machine code. | An interpreted program does not have a separate file to store machine code. |

**Do you know?**

An Assembler is a special tool that translates programs written in Assembly language into machine code, which the computer can understand.

## Introduction to Python

*Figure 7.7*

Python, a programming language introduced by Guido van Rossum in 1991, has become widely popular due to its simplicity and versatility. With Python, we can solve a variety of problems and even create our own games. What makes Python stand out is its easy-to-understand syntax, which is similar to everyday language. This makes it particularly beginner-friendly, perfect for those who are new to coding. Additionally, Python comes with a wealth of built-in tools and libraries. These tools allow programmers to perform many different tasks without having to start from scratch each time. This saves time and effort, making Python a powerful tool for programming tasks.

Python is widely used in web development, data science and machine learning, artificial intelligence, data analysis, automation and scripting, game development, web scraping, cyber security, etc.

Python has a simple to understand coding style, applying indentation instead of braces to define blocks of code. This unique feature increases readability by supporting reliable indentation practices. Developers find Python's clear and straight forward syntax easy to write concise and understandable. For example, in a loop or conditional statement, indentation signifies the scope of the block.

**Do you know?**

The name of the Python programming language comes from an old BBC television comedy sketch series called Monty Python's Flying Circus.

## Features of Python

**a. Easy to read and write**

Python uses simple and understandable syntax, making it easy for programmers to write and understand the code.

**b. Versatile**

Python can be used for a wide range of tasks like simple automating systems to complex web development, data analysis, and Artificial Intelligence.

**c. Beginner-friendly**

Python uses simple syntax, making it a great choice for those who are new to programming.

**d. Extensive standard library**

Python has an extensive standard library of pre-written code that offers programmers ready-made solutions, without requiring them to write code from ground level.

**e. Rich ecosystem**

Python has a vast collection of libraries and ready-made structures known as frameworks that provide ready-to-use tools for programmers.

**Do you know?**

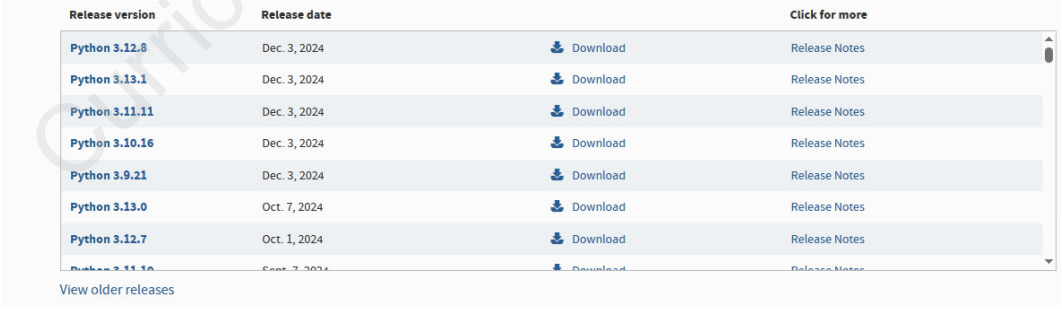
Python is one of the few programming languages recognized by Google as an official language.

## Installation process of Python

### Download python installer:

Visit the official Python website ( https://www.Python.org/downloads/ )and download the latest version of Python 3.x for Windows. The website will automatically detect your operating system and offer the appropriate installer for your system (32-bit or 64-bit).

### Download and install suitable IDE for Python.



| Release version | Release date | | Click for more |
|---|---|---|---|
| Python 3.12.8 | Dec. 3, 2024 | Download | Release Notes |
| Python 3.13.1 | Dec. 3, 2024 | Download | Release Notes |
| Python 3.11.11 | Dec. 3, 2024 | Download | Release Notes |
| Python 3.10.16 | Dec. 3, 2024 | Download | Release Notes |
| Python 3.9.21 | Dec. 3, 2024 | Download | Release Notes |
| Python 3.13.0 | Oct. 7, 2024 | Download | Release Notes |
| Python 3.12.7 | Oct. 1, 2024 | Download | Release Notes |

View older releases

*Figure 7.8*

## Downloading python from official website

**Run the installer:**

Locate the downloaded installer file (usually in your Downloads folder) and double-click on it to run the installation process.

**Install Python:**

Select your desired installation settings, click on Install to begin the installation process. This may take a few minutes.

***Note:*** *You can download python file by going through this link below: https://www.Python.org/downloads/*

**Comments in Python**

In Python, comments are brief messages that programmers write in their code to explain what the code is doing, like leaving helpful hints for others to understand the program. Comments are indicated by the '#' sign, and the interpreter ignores them, treating them as notes for the programmer. They don't affect the actual program.

**There are two types of comments in Python.**

### Single line Comment

In Python, a short note or comment starts with the '#' symbol and goes until the end of that line. If the note is longer and needs more than one line, each additional line should also begin with a '#' symbol. There should not be whitespace after '#'.

***Example 1:***

```
#This is an example of a single line comment.

print("Nepal has CULTURAL DIVERSITY")
```

***Example 2:***

```
#This is an example of a Single line comment.

#This program prints what's special about Nepal.

print("Nepal is a famous for ADVENTURE TOURISM")
```

### Multi-line comment

In Python, when a single-line comment is not sufficient and needs to go in multiple lines, it can be challenging to add a '#' at the beginning of each line. In such cases, Python allows the use of triple single quotes (''') at the beginning and end of the comment to extend it over multiple lines.

***Example:***

''' This is an example to demonstrate the use of multiple line comments.

This program prints what's special about Nepal. '''

**print**("Nepal is home for various ethnic groups, languages, and traditions.")

## Keywords

**In programming, keywords are reserved words that have predefined meanings**. Keywords cannot be used as identifiers. Some of the common Python keywords are listed below:

***Let's check!***

What are keywords in Python?

a. Words used as comment in code
b. Identifiers for variables
c. Special words reserved for specific purposes
d. User-defined functions

## I/O statements and string formatting

In Python, Input/Output(I/O) statements and string formatting are essential for interacting with users and presenting information in a clear and ordered method.

### Input/Output statements (I/O)

I/O statements help us to interact with the program. They allow us to provide our data and instructions to the program as well as display the output on screen. In Python, the statement used to give data and instruction to the program is an "input" statement and the statement to display the output on screen is "print".

## 1. Print Statement

The print statement in Python is used to display information on the screen. We use a 'print' statement to display the output of data that we want to show.

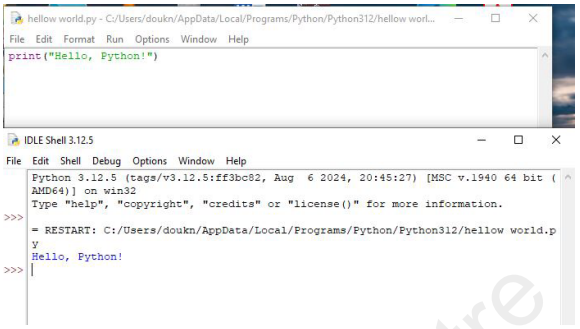



*Figure 7.9*

***Example:*** **print**("Hello, Python!")

## 2. Input Function

The 'input' function allows us to provide input to the program.

We can give data to the program as we need.

***Example:*** number = **input**("Enter number: ") **String Formatting**:

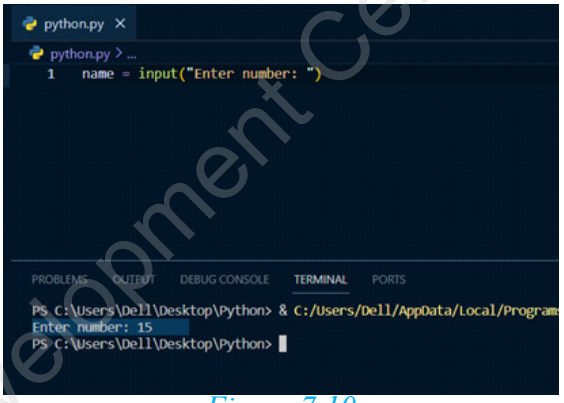



*Figure 7.10*

String formatting in Python is like creating a special message where you fill in the blanks with the information you want. It involves using a message template, allowing you to insert different names or numbers at specific spots to make your messages more interesting and customized.

### 1. Old-style formatting

In earlier times, we used '%' to insert values into a string. This was the old way of formatting the string values.

name= "Ram"

age = 25

print("My name is %s and I am %d years old." % (name, age)

Output: my name is Ram and I am 25 years old.

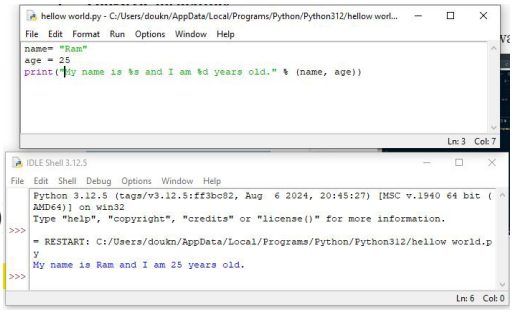



*Figure 7.11*

**2. New-style formatting (str.format)**

The new style string format uses '{ }' placeholders and the 'format' method. It is used in newer generations of formatting.

name = "Sita"

age = 30

print("My name is {} and I am {} years old.".format(name, age))

**Output:** My name is Sita, and I am 30 years old

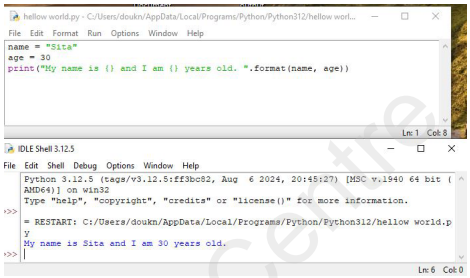



*Figure 7.12*

**3. Formatted string literals**

This is the most widely used method in string formatting in the current time. In formatted string literal we use 'f' before the string and embedding expressions inside { }. It is the most popular method of using string format.

#Formatted String Literals (f-strings)

name = "Shiva"

age = 35

print (f"My name is {name} and I am {age} years old.")

**Output:** My name is Shiva, and I am 35 years old.

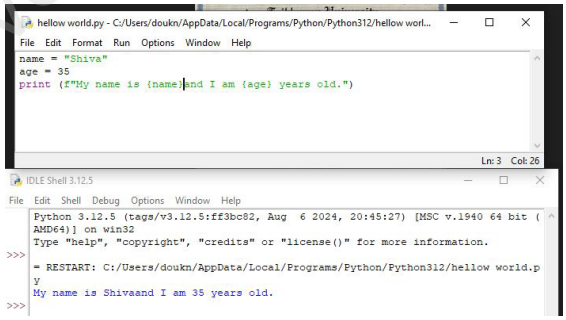



*Figure 7.13*

## Data types and variables

Imagine you have a collection of different types of fruits in your kitchen. Each fruit has its own unique characteristics, such as color, taste, and size. In computer programming, data types are like the different types of fruits. Just as you sort fruits in categories like apples, oranges, and bananas, programmers use data types to organize and categorize different kinds of information. Our data can be in various forms like numbers, decimals, alphabets or special characters.

**The classification of data based on its nature is called datatype.** Python also offers various data type which are listed as:

**Integer (int):** It is a whole number ranging from negative infinity to positive infinity.

***Examples:*** -,...,-3,-2,-1,0,1,2,3,....,

**Float (float):** It is numbers with decimals.

***Examples:*** 3.14, -0.5, 1.567.

**String (str):** It consists of alphabets, special characters, alphanumeric values which are enclosed in double quotes.

***Examples:*** "hello" , "Python@", "Mahendranagar1", "@#@#kathmandu"

**Boolean (bool):** It only provides True or False values.

***Example:*** is_student = True, has_mobile=False

## Identifier

Identifiers are names given to program units such as variables, functions, classes, or other entities. They are not predefined in the programming language but are defined by programmers themselves.

## Rules for identifier

i. An identifier name must start with a letter or the underscore character.

ii. An identifier name cannot start with a number.

iii. An identifier name can only contain alpha-numeric characters and underscores (A-z, 0-9, and _ ).

iv. Python is case sensitive so identifier names are also case-sensitive. (age, Age and AGE are three different identifiers).

v. An identifier name cannot be any of the Python keywords.

**Valid identifiers**

A1 B34 First_Name x_1

**Invalid identifiers**

1A 34BA int print first-name def

## Variables

Variables are like containers that hold information. Imagine you have labelled boxes where you can keep different types of items. Those boxes contain different categories of item and you have labelled them accordingly. Variables are similar to these boxes as they hold different types of data.

Python doesn't have any command for declaring a variable. A variable is created when the value is assigned to it.

***Example:***

a =15

Let's consider the example of data type and variable with the example given below:

| Variables | Description |
|---|---|
| age | **'age'** is the variable name of the integer data type, and it holds the integer value 15. In Python, it is represented as:<br>**age=15** |
| | |
| Height | **'height'** is the variable name of the float data type, and it holds the float value "5.7". In Python, it is represented as :<br>**height=5.7** |
| Fname | **'fname'** is the variable name of the string data type, and it holds the string value "Sanjog". In Python, it is represented as:<br>**fname="Sanjog"** |
| is_student | '**is_student**' is the variable name of the boolean data type, and it holds the Boolean value True. In Python, it is represented as:<br>**is_student = True.** |

## Concept of Type Casting

Imagine you're developing a program for a bakery where customers can place orders for cakes. You receive orders in the form of strings, but you need to convert them into numerical values (integers or floats) to perform calculations and manage inventory. Type casting is the process of converting a variable from one data type to another. In this process, the compiler automatically adjusts the data type based

on the program's requirements. For example, if we assign an integer (int) value to a float variable, the compiler will convert the int value to a float. Type casting allows programmers to explicitly control these conversions, ensuring the computer understands the kind of data needed for specific tasks.

**There are two types of casting: Implicit casting and explicit casting.**

## Implicit casting

Implicit type casting, also known as automatic type conversion, occurs when the Python interpreter automatically converts one data type to another in certain situations.

***Examples:***

```
x = 10          # integer
y = 5.5         # float
z = x + y       # the integer 'x' is cast to a float for the addition.

print(z)
```

## Explicit casting

In explicit type casting, the user intentionally converts the data type of a variable to another data type. Following are the approaches of casting the data type intentionally:

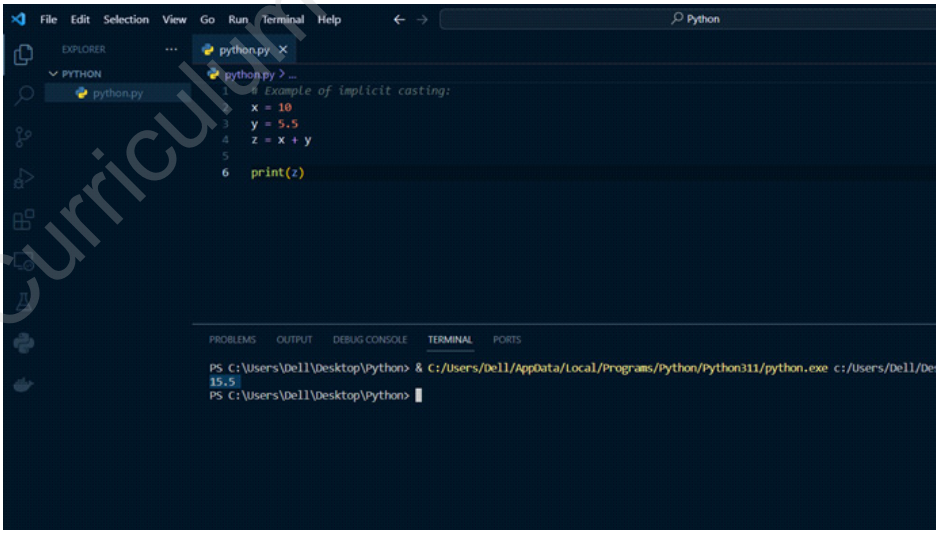



*Figure 7.13*

1. **int(x):** Converts x to an integer.

```
x = 4.5
int(x)                    #cast x to int
print(x)
```

2. **float(x):** Converts x to a floating-point number.

```
x = 4
float(x)          #cast x to float
print(x)
```

3. **str(x):** Converts x to a string.

```
x = 11
str(x)                    #cast x to string
print(x)
```

## Operators and Expressions: Arithmetic, Relational, Logical, Assignment

Operators are special symbols that we use to do different things with numbers and words which allows us to perform specific actions. They are like tools in our programming toolbox that help us to work with information.

### Some of the operator types are described below:

1. **Arithmetic operator**

Arithmetic operator is used in Python to do mathematical operations. We use arithmetic operators as special symbols to do basic math. It is like having a set of tools for simple calculation.

**Using arithmetic operators in Python:**

| Operators | Description | Example |
|---|---|---|
| + | It is used for addition. | print(5 + 3) |
| - | It is used for subtraction. | print(7 - 2) |
| * | It is used for multiplication. | print(4 * 6) |
| / | It is used for division. | print(10 / 2) |
| % | It is used for modulus. | print(5 % 3) |
| ** | It is used for exponentials. | print(3 ** 2) |

**Simple program to find the sum of the two number for the user input**

```
num1 = float(input("Enter first number: "))
num2 = float(input("Enter second number: "))
sum = num1 + num2
print("The sum of {0} and {1} is {2}".format(num1, num2, sum))
```

*Activity 7.2*

**Activity Outcome -**

Able to write a appropriate program

**Procedure:**

Write a simple code to print (Hello Class! My name is Your Name). Then comment in every step of code using single line comment and multi line comment to explain the purpose of the program and how it works..

**Result:** complete program.

## 2. Relational operator

Relational operator is used to check and compare values. These operators check the relationship between two things and tell us if they are equal, greater than or less than each other.

**Here are the main relational operators:**

| Name | Operator | Description | Example |
|---|---|---|---|
| Equal to | == | Two things are exactly the same | (5 == 5) |
| Not equal to | != | Two things are not the same | (3 != 5) |
| Greater than (separate) | > | One is greater than the other | (7 > 5) |
| Less than | < | One is less than the other | (3 < 9) |
| Greater than or equal to | >= | Is greater or equal to another | (8 >= 8) |
| Less than or equal to | <= | Is less or equal to another | (4 <= 6) |

## 3. Logical operator

Logical operators in Python are used to combine conditions and make decisions based on different situations. This operator is like a tool that helps us make decisions based on different situations. There are 3 main logical operators, 'and', 'or', and 'not'.

**AND:**

Both the conditions must be true for the result to be true in the "and" operator.

Example: x = (5<2) and (5>3)

Result: False

**Truth table for 'AND' operators**

| A | B | A and B |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

**OR:**

Only one of the conditions needs to be true for the result to be true in the "or" operator.

Example: (5<2) or (5>3)

Result: True

**Truth table for 'OR' operators**

| A | B | A and B |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

**NOT:**

The logical operator "**not**" provides the opposite result of a given condition.

***Example:*** not(5<2)

**Result:** True

**Truth table for 'NOT' operators**

| A | $\bar{A}$ |
|---|---|
| 0 | 1 |
| 1 | 0 |

## 4. Assignment operator

| Operator | Name | Example |
|---|---|---|
| = | Assignment Operator | a = 7 |
| += | Addition Assignment | a += 1 # a = a + 1 |
| -= | Subtraction Assignment | a -= 3 # a = a – 3 |
| *= | Multiplication Assignment | a *= 4 # a = a * 4 |
| /= | Division Assignment | a /= 3 # a = a / 3 |
| %= | Remainder Assignment | a %= 10 # a = a % 10 |
| **= | Exponent Assignment | a **= 10 # a = a ** 10 |

Suppose we put ingredients like salt, sugar, spices, and turmeric in different containers. To know what is there inside the particular container we put labels outside the containers.

**Assignment operators are used to assign values to variables.**

# Using the assignment operator

***Example***

```
# assign 10 to a
a = 10
# assign 5 to b
b = 5
# assign the sum of a and b to a
a += b      # a = a + b
print(a)
```

**Output: 15**

***Example:***

In this example, "my_age" is assigned the value 15. Here, my_age is like a box, and the assignment operator "=" is putting the value 15 inside that box. Now, whenever you use my_age in your program, it's like opening the box and finding the number 15.

So, the assignment operator is like a labelling machine in Python that helps us keep things organised by giving names to data.

## Expressions

An expression in Python is like a formula that tells the computer to do something with numbers and words. It is like a command that produces a value. They are like the building blocks of our code, telling the computer what to do with the information we provide.

**Here, are some examples of expressions:**

**Maths expression:** result = 5 + 3

**Text expression:** greeting = "Hello"

**Combining expression:** combined = (5 * 3) + "Python"

| Algebraic expression | Python expression |
|---|---|
| A + B - C | A + B – C |
| A x B ÷ C | A * B / C |
| (a+b) (a-b) | (a+b) * (a-b) |
| | I = (P*T*R)/100 |

### Operands

In programming, operands are values or variables that operators operate on. Operands refer to the values or entities that are operated upon by an operator. They are the variables that utilize the operators.

***Example:*** **add = 5 + 3**

Here, '5' and '3' are operands and '+' is an operator, and it is performing an 'addition' operation.

***Let's check***

In the Python statement x=a + 5 - b, a and b are_____

a. Term b)Operators c) Operands d) Equation

## Conditional statement

A conditional statement is like a decision making tool that helps our program choose what to do based on conditions provided by the user. We can ask a question and provide different answers based on the conditions provided. The most fundamental form of conditional statement is **if** statement, "**if - else**" statement and "**if - elif - else**" statement.

The statement to be executed follows the indentation rule of Python.

### if statement

"if statement" is a conditional statement that gives us output based on the requirement of the condition that we provide. "if statement" is written using the if keyword and after that condition is provided and ends with indentation.

### Syntax

if condition:

#statement to be executed when the condition is true

## Flowchart:

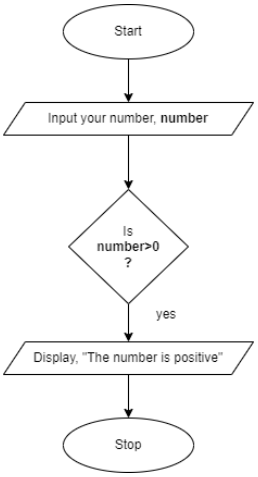



*Figure 7.14*

***Example:***

#if statement program to check if a number is positive

number = int(input("Enter a number: "))

if number > 0:

print("The number is positive.")

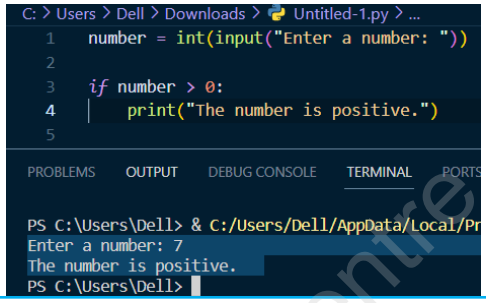



*Figure 7.15*

## if-else condition

**Syntax**

**if** condition:

#statement to be executed when the condition is true

**else:**

#statement to be executed when the condition is false

## Flowchart:

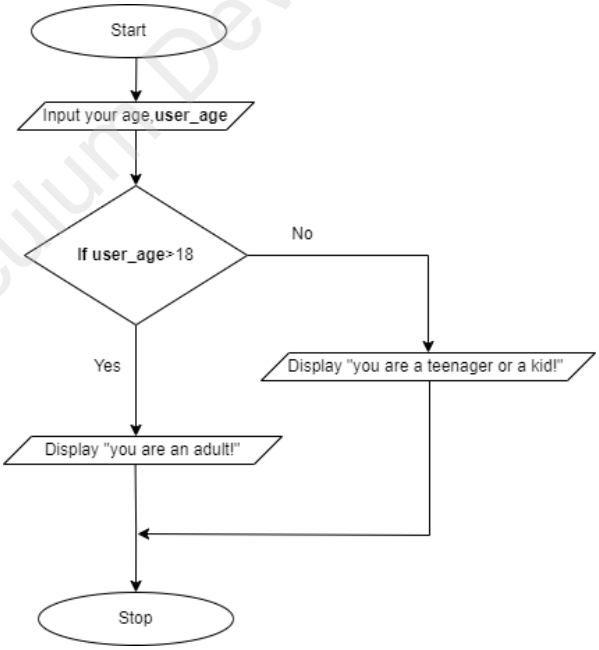



*Figure 7.16*

***Example:***

```
# Asking age to the user and providing results.
user_age = int(input("How old are you? "))
if  user_age >= 18:
    print("You are an adult!")
else:
    print("You are a teenager or  a kid.")
```

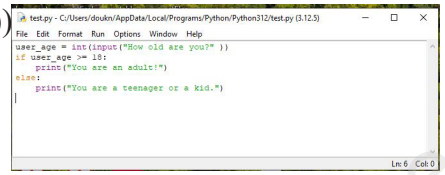



*Figure 7.17*

## if-elif-else condition

**Syntax**

```
if condition1:
    #code to be executed if condition1 is True
elif condition2:
    #code to be executed if condition2 is True
else:
    #code to be executed when condition1 and condition2 are False
```

## Flowchart:

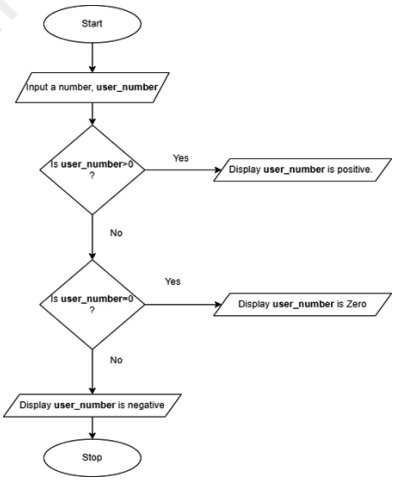



*Figure 7.18*

***Example***

Checking the number's category, whether it is positive, negative or zero.

```
user_number = int(input("Enter a number: "))
if user_number > 0:
    print("The number is positive.")
elif user_number == 0:
    print("The number is zero.")
else:
    print("The number is negative.")
```

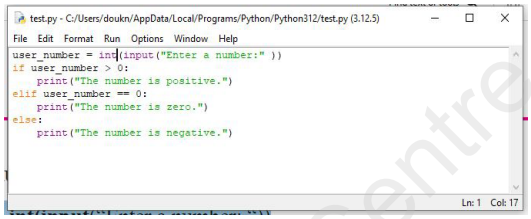



*Figure 7.19*

## Nested if (if inside if)

Nested if statement is a construct where we put another if statement inside an existing if statement. It is used to test multiple criteria and increase the number of possible outcomes. It helps in decision making using multiple conditions.

**Syntax:**

```
if condition1:
#code to be executed if condition1 is True
    if condition:
        #code to be executed if condition 2 is True
    else:
        #code to be executed when condition2 is False
    else:
    #code to be executed when condition1 and condition2 are False
```

## Flowchart:

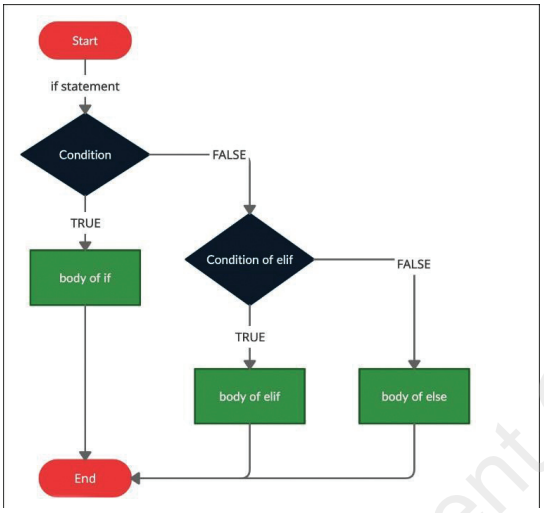



*Figure 7.20*

***Example:***

```
age = int(input("Enter your age: "))
if age >= 16:
    print("You are eligible for citizenship.")
    if age >= 18:
        print("You are eligible to cast a vote.")
    else:
        print("You are not eligible to cast vote.")
else:
    print("You are a minor.")
```

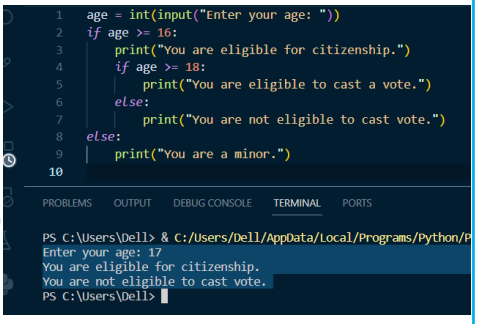



*Figure 7.21*

## Iteration

Iteration means doing things repeatedly. Iteration is the process of repeating a particular task until a specified condition is satisfied. It allows a program to perform a task multiple times until a required condition is satisfied. Iteration is like having a helper that repeats tasks for you, making your program more efficient and saving

you from writing the same thing repeatedly. The most fundamental example of iteration is the "for" loop and "while" loop.

**Do you know?**

The first instance of loop was used by Ada Lovelace to calculate Bernoulli numbers' Ada Lovelace, a London-born mathematician, is also referred to as the first programmer in the world.

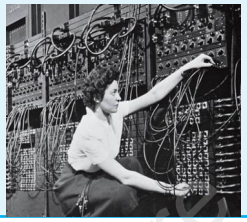


*Figure 7.21*

## For loop

"For loop" is used when we know how many times we want to repeat a block of code. A for loop in Python is a control flow structure that allows us to iterate over a sequence. It simplifies the process of repeatedly executing a set of statements for each time in the sequence. The basic idea is to loop through each element in the sequence, executing a block of code for each iteration. The for loop is particularly useful when the number of iterations is known.

"**for**" keyword is used in the for loop to give the condition.

**Syntax:**

**for** item **in** sequence:

    # Code to do something with each item

Using **for** loop to print "jump" five times

**for** x in **range**(5):

    **print**("Jump!")

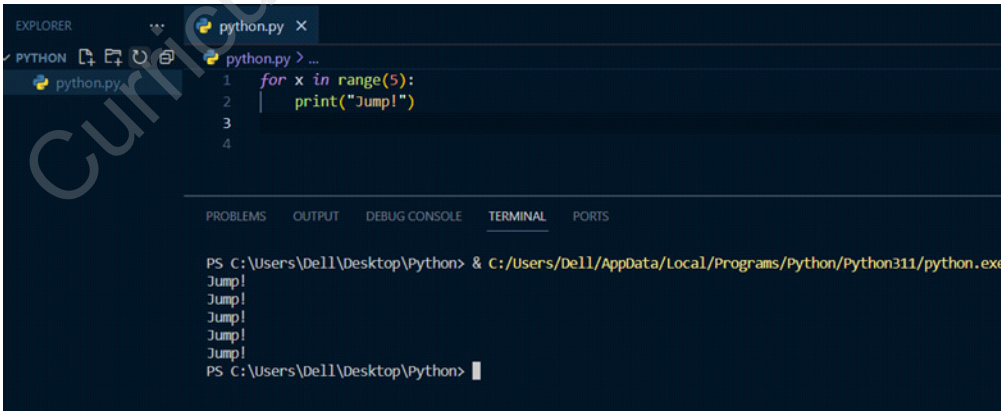



*Figure 7.23*

In Python, range is a function that helps you make a list of numbers in a certain order. It returns a sequence of numbers, starting from 0 by default and increments from 1 (by default), and stops before a given number by user.

### Pass

In Python, the **pass** keyword is a null operation or a no-operation statement. It acts as a placeholder where some code is required but no action is necessary. It is often used when a statement is required by Python syntax, but you don't want to execute any code.

***Example:***

```
if condition:
    # Some code here
else:
    pass  # Nothing happens in the "else" case
```

### Continue

In Python, the continue keyword is used in loops (such as for or while loops). It is used to skip the rest of the code inside the loop for the current iteration and move on to the next iteration. It allows us to bypass certain parts of the loop based on a condition without exiting the loop entirely.

***Example:***

```
for number in range (1, 6):
    if number == 3:
        continue # Skip the rest of the loop for number 3.
    print(number)
```

### Break

In Python, the break keyword is used in loops (such as for or while loops) to exit the loop early, even if the loop's condition hasn't been fully satisfied. It allows us to terminate the loop based on a certain condition.

***Example:***

```
for number in range (1, 6):
    if number == 3:
        break # Exit the loop when number is 3
    print(number)
```

## While loop

"While loop" in Python is a control structure that allows us to repeatedly execute a block of code as long as certain conditions remain true. *It's like having a set of instructions for a computer to keep doing something over and over until a specific goal is achieved or a condition is no longer met.* It provides a flexible way to handle tasks where the number of iterations is not known in advance.

**Syntax**

**while condition:**

```
    # Code to be executed while the condition is true
    # This code is indented and forms the body of the loop
    # It continues executing as long as the condition remains true
    # Remember to update the condition to eventually become false, or you might
      end up with an infinite loop
```

```
# Using while loop to print numbers from 1 to 5
count = 1
while count <= 5:
    print(count)
    count += 1
```

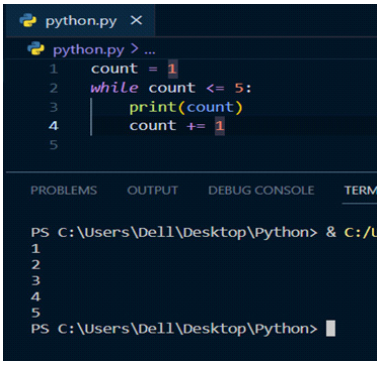



*Figure 7.24*

**Difference between for loop and while loop.**

| For loop | While loop |
|---|---|
| For loop is used when we know the number of iterations. | While loop is used when we don't know the number of iterations. |
| This loop iterates an infinite number of times if the condition is not specified. | If the condition is not specified, it shows compilation error. |
| The increment is done after the execution of the statement. | The increment can be done before or after the execution of a statement. |
| The nature of increment is simple. | The nature of increment is complex. |
| Initialisation can be in or out of the loop. | Initialisation is always out of the loop. |

***Let's check!***

Which loop is used when the number of iterations is known beforehand?

a. For loop b. While loop c. Do-while loop d.None of the above

## Python list

Python list is one of the built-in data types in Python used to store multiple data in a single variable. A list can contain heterogeneous types of elements which means elements can have different data-types like integers, strings, floats, Booleans etc.

List elements are enclosed by square brackets and elements are separated by comma.

***Example:***

thislist=[“Computer”, “Science”, 20, True]

print(thislist)

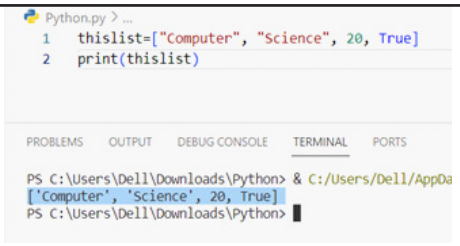



*Figure 7.25*

***Let's know more!***

In Python, lists support a rich set of operations that are common to all sequence types, including tuples, strings, and ranges. These operations are known as common sequence operations.

## Features of Python list

List elements are indexed. The index always begins from 0. So, the first list element always has index[0] so as the second list element has index[1] and so on.

Note: The index helps to access the list elements.

***Example:***

thislist=[“Computer”, “Science”, 20, True]

print(thislist[2])

```
Python.py > ...
1   thislist=["Computer", "Science", 20, True]
2   print(thislist[2])

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS C:\Users\Dell\Downloads\Python> & C:/Users/Dell/AppDa
20
PS C:\Users\Dell\Downloads\Python>
```

*Figure 7.26*

The output is 20 because the element associated with index [2] is 20.

List can have duplicate elements.

***Example:***

thislist =[“Computer”, “Science”, 20, True, “Computer”]

print(thislist)

```
Python.py > ...
1   thislist=["Computer", "Science", 20, True, "Computer"]
2   print(thislist)

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS C:\Users\Dell\Downloads\Python> & C:/Users/Dell/AppData/Local/
['Computer', 'Science', 20, True, 'Computer']
PS C:\Users\Dell\Downloads\Python>
```

*Figure 7.27*

Lists are ordered. If we add new elements in the list, the new element will hold the last position of the list. To add an element to the list, we use the append() method which is joined using “.” operator with the list variable.

***Example:***

thislist=[“Computer”, “Science”, 20, True]

thislist.append(“Social”)  #add new element in list

print(thislist)

```
Python.py > ...
1   thislist=["Computer", "Science", 20, True]
2   thislist.append("Social")
3   print(thislist)

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS C:\Users\Dell\Downloads\Python> & C:/Users/Dell/AppDat
['Computer', 'Science', 20, True, 'Social']
PS C:\Users\Dell\Downloads\Python>
```

*Figure 7.28*

To remove elements from list, del keyword is used preceding the list variable specifying the list index.

***Example:***

thislist=[“Computer”, “Science”, 20, True, “Social”]

**del** thislist[4]  #removes the element from list

**print**(thislist)

```
Python.py > ...
1  thislist=["Computer", "Science", 20, True, "Social"]
2  del thislist[4]      #removes the element from list
3  print(thislist)

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS C:\Users\Dell\Downloads\Python> & C:/Users/Dell/AppData/Local/
['Computer', 'Science', 20, True]
PS C:\Users\Dell\Downloads\Python>
```

*Figure 7.29*

## List length

To find the numbers of elements present in the list, we use len() function. The len() function gives us the number of list elements present in the list.

***Example:***

thislist=[“Computer”, “Science”, 20, True]

a=len(thislist)

print(a)

```
Python.py > ...
1  thislist=["Computer", "Science", 20, True]
2  a=len(thislist)
3  print(a)

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS C:\Users\Dell\Downloads\Python> & C:/Users/Dell/AppD
4
PS C:\Users\Dell\Downloads\Python>
```

*Figure 7.30*

## Looping through list

We can loop through the list elements using the for loop to print all the elements one by one.

***Example:***

thislist=[“Computer”, “Science”, 20, True]

for X in thislist:

print(X)

```
Python.py > ...
1  thislist=["Computer", "Science", 20, True]
2  a=len(thislist)
3  print(a)

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS C:\Users\Dell\Downloads\Python> & C:/Users/Dell/AppD
4
PS C:\Users\Dell\Downloads\Python>
```

*Figure 7.31*

*Activity 7.3*

**Activity Outcome -**

Able to write a appropriate program

**Required Resources:** to insert another fruit grape in the list.

**Procedure:**

Write a program to add new items in the list.

**Result:** a complete program

## Python Dictionary

Picture a real-life dictionary where you look up words to find their meanings. In Python, a dictionary is like that, but instead of words and meanings, it stores pairs of keys and values. Think of keys as the words and values as their meanings. The key-value pairs are stored together so that you can quickly look up a key and get its corresponding value.

In Python, dictionaries are used to store data in **Key:Value pair**.

A dictionary is a collection of data in Key:Value pairs, written within curly braces.

***Example:***

**Key:value** pairs are written within the curly braces. Here,

- The "student" is key and value associated to it is "pen"
- The "teacher" is key and value associated to it is "marker"
- The "tailor" is key and value associated to it is "needle"

In this way we can add as many key:value pairs as we want.

***Note:*** The values in the dictionary can be of any data type.

***Example:***

```
mydict= {
    “student” : “pen”,
    “teacher” : “marker”
    “tailor” : “needle”,
    “marks” : 30,
    “is_present” : True
}
print(mydict)
```

```
Python.py > ...
1  mydict= {
2      "student" : "pen",
3      "teacher" : "marker",
4      "tailor" : "needle",
5      "marks" : 30,
6      "is_present" : True
7  }
8  print(mydict)

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS C:\Users\Dell\Downloads\Python> & C:/Users/Dell/AppData/Local/Programs/Python/Python311/pyth
{'student': 'pen', 'teacher': 'marker', 'tailor': 'needle', 'marks': 30, 'is_present': True}
PS C:\Users\Dell\Downloads\Python>
```

*Figure 7.32*

## Features of dictionary

Duplicate keys are not allowed: In the Python dictionary, two values with the same key cannot exist. If the same key has two or more values, the new value will replace the old value.

***Example:***

```
mydict={
  “student” : “pen”,
  “teacher” : “marker”,
  “tailor” : “needle”,
  “marks” : 30,
  “is_present” : True,
  “teacher” : “duster”
  }
print(mydict)
```

```
Python.py > ...
1  mydict={
2      "student" : "pen",
3      "teacher" : "marker",
4      "tailor" : "needle",
5      "marks" : 30,
6      "is_present" : True,
7      "teacher" : "duster"
8      }
9  print(mydict)

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS C:\Users\Dell\Downloads\Python> & C:/Users/Dell/AppData/Local/Programs/Python/Python311/pyth
{'student': 'pen', 'teacher': 'duster', 'tailor': 'needle', 'marks': 30, 'is_present': True}
PS C:\Users\Dell\Downloads\Python>
```

*Figure 7.33*

The older value “marker” of key “teacher” gets replaced by “duster”.

## Dictionaries are changeable:

After the dictionary has been created, we can add or remove values. A new value is added to the dictionary using a new key index and value associated with it.

***Example:***

```
mydict={
  “student” : “pen”,
  “teacher” : “marker”,
  “tailor” : “needle”,
  “marks” : 30,
  “is_present” : True
}
mydict[“mechanic”]=”tools”
print(mydict)
```


*Figure 7.34*

The del keyword is used to remove specified key names in the dictionary.

***Example:***

```
mydict={
  “student” : “pen”,
  “teacher” : “marker”,
  “tailor”  : “needle”,
  “marks” : 30,
  “is_present” : True,
  “mechanic” : “tools”
}
del mydict[“mechanic”]
print(mydict)
```


*Figure 7.35*

## Accessing the value in dictionary

The value in the dictionary can be accessed by referring to its key name, by placing the key name inside a square bracket.

***Example:***

```
mydict={
    “student” : “pen”,
    “teacher” : “marker”,
    “tailor” : “needle”,
    “marks” : 30,
    “is_present” : True
}
a= mydict[“teacher”]
print(a)
```

```
Python.py > mydict
1   mydict={
2       "student" : "pen",
3       "teacher" : "marker",
4       "tailor" : "needle",
5       "marks" : 30,
6       "is_present" : True
7   }
8   a= mydict["teacher"]
9   print(a)

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS C:\Users\Dell\Downloads\Python> & C:/Users/Dell/AppData
marker
PS C:\Users\Dell\Downloads\Python>
```

*Figure 7.36*

## Dictionary length

To determine the numbers of key:values present in the dictionary, len() function is used.

***Example:***

```
mydict={
    “student” : “pen”,
    “teacher” : “marker”,
    “tailor” : “needle”,
    “marks” : 30,
    “is_present” : True
    }
b=len(mydict)
print(b)
```

```
Python.py > ...
1   mydict={
2       "student" : "pen",
3       "teacher" : "marker",
4       "tailor" : "needle",
5       "marks" : 30,
6       "is_present" : True
7       }
8   b=len(mydict)
9   print(b)

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS C:\Users\Dell\Downloads\Python> & C:/Users/Dell/AppDa
5
PS C:\Users\Dell\Downloads\Python>
```

*Figure 7.37*

## Looping through Dictionary

We can loop through the dictionary using the for loop to print all the elements one by one.

***Example:***

```
mydict={
    “student” : “pen”,
    “teacher” : “marker”,
    “tailor” : “needle”,
    “marks” : 30,
    “is_present” : True
}
for X in mydict:
    print(X)
```

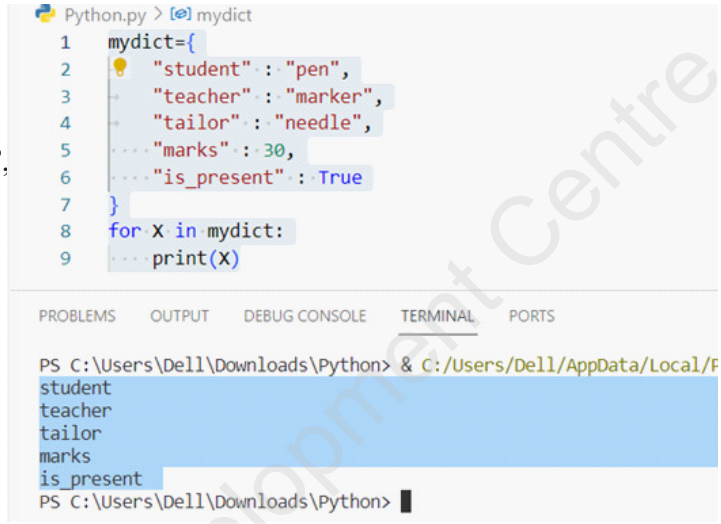


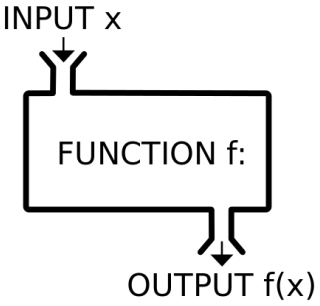

*Figure 7.38*

## Difference between list and dictionary

| List | Dictionary |
|---|---|
| A list stores elements in a sequence. | A dictionary stores data in key:value pairs. |
| List elements are accessed by index. | Dictionary values are accessed by keys. |
| List elements have an order. | Dictionary values don’t have any specific order. |
| List elements can be added and removed. | Dictionary values are added as key:value pair and removed by key . |

## Uses of Library functions

### String functions




A collection of related assertions that carry out a mathematical, analytical, or evaluative operation is known as a function. An assortment of proclamations called Python. Capabilities returns the specific errand. Function names meet the same standards as variable names do. The objective is to define a function and group-specific frequently performed actions. Python includes the following some built-in methods to manipulate strings −

**center()**

The **center()** method returns a string which is padded with the specified character.

**syntax**

string.center(width[, fillchar])

center() Parameters

The center() method takes two arguments:

- **width** - length of the string with padded characters
- **fillchar** (optional) - padding character

The fillchar argument is optional. If it's not provided, space is taken as the default argument.

***Example:*** center() Method With Default fillchar

string = "Python is awesome"

new_string = string.center(24)

print("Centered String: ", new_string)

Output

**Centered String: Python is awesome.**

**upper()**

Python string method **upper()** returns a copy of the string in which all case-based characters have been uppercased.

**Syntax:** str.upper()

***Example:***

str = " this is a string example....wow!!!"

print str.upper()

**Output:**

THIS IS STRING EXAMPLE....WOW!!!

**lower()**

Python string method **lower()** returns a copy of the string in which all case-based characters have been lowercased.

**Syntax:** str.lower()

***Example:***

str = "THIS IS STRING EXAMPLE....WOW!!!"

print str.lower()

**Output:**

This is a strong example....wow!!!

**len()**

Python string method **len()** returns the length of the string.

**Syntax:** len(str)

***Example:***

str = “this is string example....wow!!!”

print “Length of the string: “, len(str)

**Output:**

Length of the string: 32

## Numeric and mathematical functions

In python a math module is used to access mathematical functions. All methods of math() function are used for integer or real type objects but not for complex numbers.

**Numeric functions:** Some of the built-in numeric functions are:

sum(): Sums the items of an iterable

total = sum([1, 2, 3, 4, 5])

**output:**

total is 15

abs(): Returns the absolute value of a number.

absolute = abs(-10)

**output:**

absolute is 10

**round()**: Rounds a number to a specified number of decimal places.

rounded = round(3.14159, 2)

**output:**

rounded is 3.14

**max() and min()**: Return the largest and smallest items in an iterable or the largest/ smallest of two or more arguments.

maximum = max(1, 2, 3, 4, 5)

**output:**

maximum is 5

minimum = min(1, 2, 3, 4, 5)

**output:**

minimum is 1

## Mathematical functions

**math.sqrt():** Returns the square root of a number.

import math

root = math.sqrt(16)

**output:**

root is 4.0

**math.ceil()** and **math.floor():** Return the ceiling or floor of a number.

*ceiling = math.ceil(3.14)*

**output:**

ceiling is 4

floor = math.floor(3.14)

**output:**

floor is 3

**math.factorial():** Returns the factorial of a number.

factorial = math.factorial(5)

**output:**

factorial is 120

**math.pow():** Returns a number raised to the power of another number.

power = math.pow(2, 3)

**output:**

power is 8.0

**math.sin(), math.cos(), math.tan()**: Return the sine, cosine, and tangent of a number (in radians).

sine = math.sin(math.pi / 2)

**output:**

sine is 1.0

cosine = math.cos(0)

**output:**

cosine is 1.0

tangent = math.tan(math.pi / 4)

**output:**

tangent is 1.0

**math.log():** Returns the natural logarithm of a number.

natural_log = math.log(10)

**output:**

natural_log is 2.302585092994046

## Practice Tasks

### 1. Adding two numbers (Constant variable initialization)

```
number_1 = 5
number_2 = 7
add = number_1 + number_2
print ("The addition of number_1 and number_2 is: ", add)
```

**2. Adding two numbers asking from the user**

```
number_1 = int(input("Enter the first number: "))
number_2 = int(input("Enter the second number: "))
result = number_1 + number_2
print (f "The sum of {number_1} and {number_2} is: {result}")
```

**3. Calculating the area of a circle**

```
# Hint: pi = 3.14159 * (radius ** 2)
radius = float (input ("Enter the radius of the circle: "))
area = 3.14159 * (radius ** 2)    # Using formula
print ("The area of the circle is: ", area)
```

**1. Calculating the area of the rectangle**

```
length = float(input("Enter the length of the rectangle: "))
width = float(input("Enter the width of the rectangle: "))
area = length * width          # Using formula
print("The area of the rectangle is: ", area)
```

**5. Printing the name of user by asking name (string formatting)**

```
user_name = input ("Enter your name: ")
greeting = (f"Hello! {user_name}")
print(greeting)
```

**6. A program that takes current temperature as input and displays "it's a warm day" if it's above 25.**

```
temperature = float(input("Enter the current temperature in Celsius: "))
if temperature > 25:
   print("It's a warm day.")
```

**7. A program that shows odd or even numbers using an if statement**

```
number = int(input("Enter an integer: "))
if number % 2 == 0:
   print("The number is even.")
```

```
if number % 2 != 0:
    print("The number is odd.")
```

**8. Finding the greatest between two numbers asking from the user**

```
number_1 = input("Enter the first number: ")
number_2 = input("Enter the second number: ")
if (number_1 > number_2):
    greater_number = number_1
else:
    greater_number = number_2
print("The greater number is: ", greater_number)
```

**9. A program to check whether the given number is inside range (10 -100) or not**

```
number = float(input("Enter a number: "))
if 10 <= number <= 100:
    print ("The number is within the range.")
else:
    print ("The number is outside the range.")
```

**10. A program to check whether the number is divisible by 7 or not**

```
number = int(input("Enter a number: "))
if number % 7 == 0:
    print(f"{number} is divisible by 7.")
else:
    print(f"{number} is not divisible by 7.")
```

**11. Finding the greatest between three numbers asking from the user**

```
number_1 = int(input("Enter the first number: "))
number_2 = int(input("Enter the second number: "))
number_3 = int(input("Enter the third number: "))
```

```
if (number_1 > number_2) and (number_1 > number_3):
    print("number_1 is greater.", number_1)
elif (number_2 > number_1) and (number_2 > number_3):
    print("number_2 is greater.", number_2)
else:
    print("number_3 is greater.", number_3)
```

## 12. A program to check whether year is leap year or not

```
year = int (input ("Enter the year: "))
if year % 100 == 0:
    if year % 400 == 0:
        print ("Entered year is a leap year")
    else:
        print ("Entered year is not a leap year")
else:
    if year % 4 == 0:
        print ("Entered year is a leap year")
    else:
        print ("Entered year is not a leap year")
```

## 13. A program to find the middle number among three numbers

```
num1 = float(input("Enter the first number: "))
num2 = float(input("Enter the second number: "))
num3 = float(input("Enter the third number: "))
if num1 <= num2 <= num3 or num3 <= num2 <= num1:
    middle_number = num2
elif num2 <= num1 <= num3 or num3 <= num1 <= num2:
    middle_number = num1
else:
```

```
    middle_number = num3
print(f"The middle number among {num1}, {num2}, and {num3} is: {middle_
number}")
```

### 14. A program that takes students' scores as input and provides grades based on certain criteria

```
score = float(input("Enter the student's score: "))
if score >= 90:
    grade = 'A'
elif score >= 80:
    grade = 'B'
elif score >= 70:
    grade = 'C'
elif score >= 60:
    grade = 'D'
else:
    grade = 'F'
print (f "The student's grade is: {grade}")
```

### 15. A program to classify triangle based on angles

```
angle1 = float (input ("Enter the first angle: "))
angle2 = float (input ("Enter the second angle: "))
angle3 = float (input ("Enter the third angle: "))
if angle1 + angle2 + angle3 == 180:
    if angle1 == angle2 == angle3:
        print ("It's an Equilateral triangle.")
    elif angle1 == angle2 or angle2 == angle3 or angle1 == angle3:
        print ("It's an Isosceles triangle.")
    else:
        print ("It's a Scalene triangle.")
```

```
else:
    print ("Invalid triangle angles.")
```

**16 . A program to print numbers from 1 to 10 using a for loop**

```
for number in range (1, 11):
    print(number)
```

**17. A program to print the sum of the first 10 natural numbers using a for loop**

```
sum_of_numbers = 0
for number in range(1, 11):
    sum_of_numbers += number
print("The sum of the first 10 natural numbers is:", sum_of_numbers)
```

OR

```
sum_of_numbers = 0
for number in range (1, 11):
    sum_of_numbers =   sum_of_numbers + number
print ("The sum of the first 10 natural numbers is : " , sum_of_numbers)
```

**18. A program to print even numbers in a range**

```
(# range(2, 11, 2) the no. 2 at end tells program to skip number by 2)
for i in range(2, 11, 2):
    print(i)
```

**19. A program to print numbers (10-1) in reverse order**

```
(# range(10, 1, -1) the no. -1 at end tells program to count numbers in reverse
order)
for i in range(10, 1, -1):
    print(i)
```

### 20. A program to print factorials of 10

```
num = 10
factorial = 1
for i in range(1, num + 1):
    factorial *= i
print (f "Factorial of {num}: {factorial}")
```

### 21. A program to print star pattern as shown in the figure for i in range (1, 6):

```
# Program to print star pattern
for i in range (1, 6):
Print ('*' * i)
```

### 22. A program to calculate average of given numbers

```
numbers = [5, 8, 12, 3, 6]
total = sum(numbers)
average = total / len(numbers)          # "len" keyword gives total length  of numbers
print("Average:", average)
```

### 23. A program to find the sum of N numbers

```
N = int(input("Enter the value of N: "))
total = 0
for i in range(N):
    num = float (input (f "Enter number {i+1}: "))
    total += num
print (f "The sum of {N} numbers is: {total}")
```

### 24. A program to print numbers from 1 to 15 using a while loop

```
number = 1
while number <= 15:
    print(number)
    number += 1
```

**25. A program to print the first 10 integers and their squares using a while loop**

```
i = 1
while i <= 10:
    square = i ** 2
    print(f "Number: {i}, Square: {square}")
    i += 1
```

**26. A program to print the sum of the first 10 natural numbers using a while loop**

```
sum_of_numbers = 0
number = 1
while number <= 10:
    sum_of_numbers += number
    number += 1
print ("The sum of the first 10 natural numbers is: ", sum_of_numbers)
```

**27. A program to print products of digits of numbers accepted from users**

```
number = int (input ("Enter a number: "))
product = 1
while number > 0:
    digit = number % 10
    product *= digit
    number //= 10
print ("Product of the digits:", product)
```

**28. A program to reverse the number from user using while loop**

```
number = int (input ("Enter a number: "))
reversed_number = 0
while number > 0:
    digit = number % 10
```

```
    reversed_number = reversed_number * 10 + digit
    number //= 10
print ("Reversed number : ", reversed_number)
```

### 29. A program to print Fibonacci series till n terms provided by the user

A Fibonacci series begins with 0 and 1. After that, each number is obtained by adding the two previous numbers together.

Each term in the series is obtained by adding the two previous terms. For example, 0 + 1 = 1, 1 + 1 = 2, 1 + 2 = 3, 3+5=8, 5+8=13 and so on.

0, 1, 1, 2, 3, 5, 8, 13

```
n = int(input("Enter the number of terms in the Fibonacci series: "))
first_term = 0
second_term = 1
count = 0
while count < n:
    print (first_term, end=", ")
    next_term = first_term + second_term
    first_term = second_term
    second_term = next_term
    count += 1
```

### 30. A program to check if a number is an Armstrong number using a while loop (Armstrong number is a number that is equal to the sum of cubes of its digits.

153:

$1^3 + 5^3 + 3^3$

1 + 125 + 27 = 153

)

```
number = int (input ("Enter a number: "))
original_number = number
```

```
num_digits = len (str (number))
sum_of_digits = 0
while number > 0:
    digit = number % 10
    sum_of_digits += digit ** num_digits
    number //= 10
if original_number == sum_of_digits:
    print (f "{original_number} is an Armstrong number.")
else:
    print(f"{original_number} is not an Armstrong number.")
```

## Practical Task

Download Python to your computer and install it.

## Exercises

**1. Answer these questions.**

a) Explain programming languages with its types.

b) Differentiate between compiler and interpreter.

c) Define algorithm and flowchart.

d) Write five symbols with their functions of flowchart.

e) Create an algorithm to find whether the given number is odd or even. Also make flowchart from the algorithm.

f) Explain Python programming with its features.

g) Describe the use of input/output statements in Python.

h) Define formatted string literals. Demonstrate its use in a simple program where the user inputs their name and the output is displayed "Hello (user_name)! Welcome to our program".

i) Explain the data types with examples.

j) Describe typecasting with its types.

k) List the types of operators. Explain any one of them in detail with examples.

l) Explain a conditional statement with an example.

m) Define iteration. Differentiate for and while loop.

n) Differentiate between list and dictionary.

o) Given is the list elements.

[Kathmandu, Biratnagar, Nepalgunj, Pokhara, Butwal, Birendranagar, Mahendranagar, Balefi ]

a. Add list item Malangawa.

b. Remove list item Balefi

c. Print the list items one by one using a loop.

## 2. Write the names of technical terms:

a. A set of instructions to perform a specific task.

b. The set of rules to write a program.

c. Language translator that reads and translates program code to machine code line by line

d. The graphical or pictorial representation of a program

e. Text added in source code for the purpose of providing information or explanation that is ignored by the Python interpreter during execution

f. A data type that only provides true or false value

g. A special symbol that is used to check the relationship between two variables

h. The values or entities that operators act upon in programming

i. A decision-making control structure that helps our program to choose what to do

j. The process of repeating a task until a specified condition is met

## 3. Write true or false for the following statements.

a. Programming language is used to give commands to the computer.

b. Computers understand high level programming easily.

c. Compiler and interpreter translate programming code to machine code.

d. Algorithms are a graphical representation of instructions.

e. Python is a machine understandable programming language.

f. Print statement is used to display output in Python.

g. In Python, "int" data type denotes alphabetical characters such as hello.

h. Type casting is the conversion of one data type to another.

i. In AND operator all the conditions must be true for the result to be true.

j. In Python, if statement is used to provide a condition that, when false, allows a block of code to be executed.

k. A for loop in Python is used to repeat a process until a specified time given by the user.

l. Dictionaries have key : value pairs.

m. List items cannot be changed.

n. In the dictionary the data can be deleted by value.

## 4. Write full forms of the following.

i. LLL ii. HLL

iii. I/O iv. IDE

## 5. Choose the correct answer.

a. In the context of programming language, what does the term "syntax" refer to?

(i) The meaning of the code

(ii) A set of instructions

(iii) Efficiency of code execution

(iv) Structure and rules of code

b) A computer understands …….

(i) high level programming language

(ii) low level programming language

(iii) natural human language

(iv) symbols and pictures

c) How does a flowchart represent a process in a program?

(i) It provides a set of instructions for the computer.

(ii) It visually shows the steps and decision points in a process.

(iii) It directly executes the algorithm on the computer.

(iv) It is a high-level programming language.

d) How does an algorithm differ from flowchart?

(i) Algorithm is visual, while a flowchart is textual representation.

(ii) An algorithm and flowchart are interchangeable terms.

(iii) Algorithm is programming language, while flowchart is abstract concept.

(iv) Algorithm is a step-by-step procedure, while flowchart is a programming language.

e) Python is a ……. programming language.

(i) natural (ii) low-level

(iii) high-level (iv) easy

f) What are keywords in Python?

(i) Words used as comment in code

(ii) Identifiers for variables

(iii) Special words reserved for specific purpose

(iv) User-defined functions

g) Input function is used in Python to …….

(i) retrieve user input

(ii) display the information

(iii) define variable

(iv) perform mathematical calculation

h) What data type in Python is suitable for representing text and characters?

(i) Float (ii) Boolean

(iii) Int (iv) String

i) What will be the result of the following expression: float("3.14")?

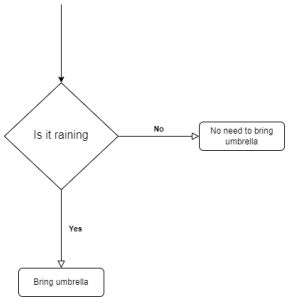
(i) 3.14 (ii) 3 (iii) "3.14" (iv) Error

j) What is the function of the NOT operator in Python?

(i) Check for equality (ii) Give True value

(iii) Convert expression to string (iv) Negate Boolean expression

k) What does the given image represent?




(i) It is a flowchart telling us to bring umbrella.

(ii) It is a flowchart giving us information about rain.

(iii) It is a flowchart showing decision to make based on the condition of rain.

(iv) It is a flowchart telling us not to bring umbrella.

L) How are elements accessed in a Python list?

(i) By keys (ii) By values

(iii) By indices (iv) By attributes

M) Which of the following statements accurately describes the mutability of Python lists and dictionaries?

(i) Lists are immutable, while dictionaries are mutable.

(ii) Both lists and dictionaries are immutable.

(iii) Both lists and dictionaries are mutable.

(iv) Lists are mutable, while dictionaries are immutable.

N) What is the primary purpose of Python lists?

(i) Storing data in key:value pairs

(ii) Storing data in a sequence

(iii) Storing data in a unordered manner

(iv) Associating keys with values

## 6. Practise these practical questions.

**Input/Output**

a) Write a program to ask input as "name" from the user and greet the user as he/she provides their name.

b) Write a program to input 3 sides of triangle and print area

c) Write a program to input a radius and find the area of circumference of circle

d) Write a program to input 3 digits as integers and calculate their sum and average.

e) Write a program to input a diameter and print area of circle and circumference of circle

**Constant value and Operators**

a) Write a program to input a radius and find the area of circumference of the circle.

b) Write a program to show the use of arithmetic operators. (Perform add, subtract, multiply, divide, square and modulus between two numbers)

c) Write a program to show the use of a comparison operator between two variables. (Use the following operators. ==, <=, >=, !=)

**If…..else Statement**

a) Write a program using the if statement to check whether the given number is positive.

b) Write a program that takes people's age as input and checks whether they are eligible to vote. (Voting age is 18 and above)

c) Write a program to check whether the given number is even or odd using an if-else statement.

d) Write a program to input two numbers and find the smallest number.

e) Write a program that takes students' marks as an input and checks whether the student passed or failed. (40 or greater is pass marks)

f) Write a program that takes 3 numbers as inputs and displays the smallest number.

g) Make a program like a traffic light using an if-elif-else statement. Ask input from the user on providing color. (green=go, red=stop and orange=be ready, any other color=invalid color)

h) Write a program to input practical and theory marks of Computer and check pass or fail. You can also validate whether a user has entered valid marks or not. (valid marks should be checked for both theory and practical)

**For loop**

a) Write a program to print numbers from 1-15.

b) Write a program to print an input name 20 times.

c) Write a program to print the squares of numbers from 1 to 5 using a for loop.

d) Write a program to print first 20 odd numbers

e) Write a program that takes an integer as input and prints the multiplication table for that number.

**While loop**

a) Write a program to print numbers from 1-20.

b) Write a program to print first 30 even numbers.

c) Write a program to calculate the factorial of a given number.

d) Write a program to print the sum of the first 20 odd numbers.

e) Write a program to input multi digits numbers and display the sum of the digits.

**Lists**

1. Write a Python program to create a list of 5 integers and print the list.
2. Write a Python program to print the second and fourth elements of a list.
3. Write a Python program to append a new element to the end of a list and print the updated list.
4. Write a Python program to print the first three elements of a list using slicing.
5. Write a Python program to print the length of a list.
6. Write a Python program to iterate through a list and print each element.
7. Write a Python program to create a list of squares of numbers from 1 to 10 using list comprehension.
8. Write a Python program to remove the third element from a list and print the updated list.

**Dictionaries**

1. Write a Python program to create a dictionary with 3 key-value pairs and

print the dictionary.

2. Write a Python program to print the value associated with a specific key in a dictionary.

3. Write a Python program to add a new key-value pair to a dictionary and print the updated dictionary.

4. Write a Python program to remove a key-value pair from a dictionary using the del statement and print the updated dictionary.

5. Write a Python program to iterate through a dictionary and print all the keys.

6. Write a Python program to create a dictionary where the keys are numbers from 1 to 5 and the values are their squares using dictionary comprehension.

7. Write a Python program to check if a specific key exists in a dictionary.

8. Write a Python program to merge two dictionaries and print the resulting dictionary.

## *Project Work*

Develop a money exchange system in Python and write a report on it. Your project should include functionalities for converting between various currencies using exchange rates.

### Steps to Follow:

**Understand the Requirements:**

1. Research and decide on the features to include, such as currency conversion, displaying exchange rates, and handling errors.
2. Identify the currencies to support (e.g., USD, EUR, GBP, INR, NPR).

**Design the Program:**

1. Outline the structure of your program using pseudocode or a flowchart.
2. Include functions for key features like fetching exchange rates, converting currencies, and displaying results.

**Write the Code:**

1. Develop the program in Python, ensuring proper use of functions, loops, and conditionals.
2. Organize the code into logical sections with meaningful variable and function names.
3. Add comments for clarity.

**Test the Program:**

1. Provide sample inputs and validate the outputs.
2. Ensure the program handles errors as well

**Document the Analysis:**

1. Describe whether the program executed without errors.
2. Provide examples of inputs and outputs to demonstrate functionality.
3. Assess whether the code is well-organized and easy to follow, noting any areas for improvement.

**Suggest Improvements:**

1. Propose additional features, such as live rate updates, historical rate tracking, or a graphical user interface.
2. Offer suggestions on how your peer could enhance the program's functionality or code readability.

-0-