DRAFT

WEB APPLICATIONS
(SUBJECT CODE-803)
CLASS-XI
SESSION 2024-25

TABLE OF CONTENTS

PRACTICAL MARKS DISTRIBUTION

CLASS XI

WEB APPLICATIONS (803)

Details	**Marks Distribution**
For Project work (**Do anyone**) • Create a website using HTML & CSS • Design a poster/e-invite/brochure/ advertisement using GIMP	10 Marks
Practical File **HTML Programs:** • Images • Lists • Tables • Hyperlinks • Form **JavaScript Programs:** Any 5 JavaScript programs based on the content.	10 marks
Viva-Voce	5 marks
Practical Exam based on HTML and CSS, GIMP & JavaScript	15 Marks

Learning Objectives

Chapter 1: Basics of Networking

After studying this chapter, the students will:

- Understand the concept of Networking and Data Communication.
- Get familiar with data communication terminology.
- Know about various Network Devices and types of Networks.
- Get familiar with Peer-to-Peer and Client-Server Web Architecture.
- Be aware of Network Threats and the Security Measures to be taken.

Chapter 2: Website Building Using HTML and CSS

After studying this chapter, the students will:

- Know about creating a webpage using the tags in HTML.
- Understand the usage of Images, Lists and Tables.
- Get familiar with Hyperlinks and Forms.
- Know how to embed audio and video files in a web page.
- Get familiar with CSS and the three ways to implement it.
- Understand the CSS Box Model using Div.
- Be able to use CSS in a webpage.

Chapter 3: Multimedia Design Using GIMP

After studying this chapter, the students will:

- Get familiar with various multimedia Design Tools.
- Especially know about GIMP and its Toolbox.
- Understand the concept of Filters in GIMP.
- Know how to work with Layers in GIMP.

Chapter 4: JavaScript Part 1

After studying this chapter, the students will:

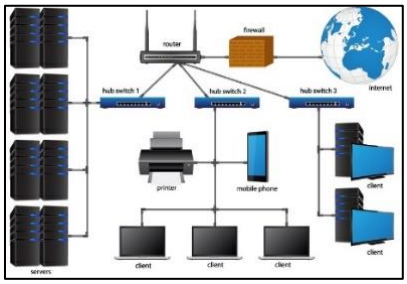
- Get introduced to scripting language JavaScript.
- Understand what JavaScript is and how it is interpreted.
- Get familiar with JavaScript features and advantages
- Be aware of the rules and common Errors in JavaScript
- Know how to Input and Output from the script
- Get familiar with the Data Types, Variables and Operators in JavaScript
- Understand Inbuilt functions in JavaScript
- Use Conditional Statements and Loops

CHAPTER 1
Networking Concepts

Topics covered

1.1 What is Networking?

A computer network is a group of devices connected with each other through different transmission mediums such as cables, radio waves, satellites etc. These devices can be computers, printers, scanners, plotters, mobile phones etc. A computer network is mostly used to send and receive data stored in other devices over the network. These devices are often referred as nodes.



There are numerous computer networks available, and these can be categorized according to their size as well as their purpose. The size of a network depends on the area, the number of computers, the purpose for which they are being networked etc. It could include devices within a single home or millions of devices spread across the world.

1.1.1 Concept of Communication:

The transmission of digital data between various devices is data communication. A telecommunications network allows computers to communicate and exchange data in many forms such as text, numbers, pictures, audio, and video. The connection between networked computing devices is established using both wired/cable media or wireless media. The best-known computer network is the Internet. When computer systems, and other peripheral digital devices are connected to form a network, they communicate and provide numerous advantages:

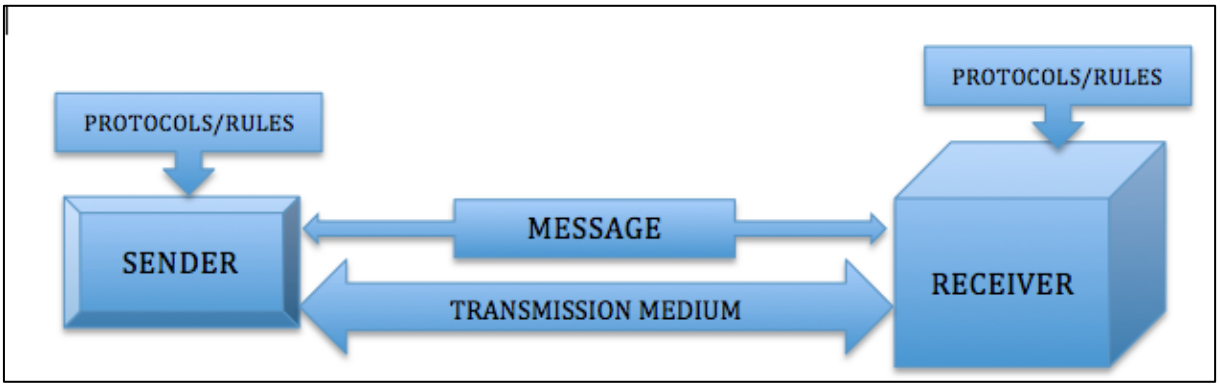
- Resource sharing such as printers and storage devices.
- Exchange of information through email and FTP.
- Information sharing by using Web or Internet.
- Video conferencing.
- Instant messaging.
- Parallel computing.
- Direct to Home Online content like Television channels, Netflix, Amazon Prime Disney Hotstar, Apple TV etc.

1.1.2 Components of Data Communication

When sharing information or communicating over a network we exchange data through five components.

1. **Message:** This refers to the information or data to be communicated which could be text, numbers, pictures, audio, and video is the message.
2. **Sender:** Here we are referring to the sender as the device that sends the data message. It can be a computer, workstation, mobile phone, camera, and so on.
3. **Receiver:** Here we are referring to the receiver as the device that receives the message. It can be a computer, workstation, mobile phone, television, and so on.
4. **Transmission/communication medium:** This is the medium or path which facilitates messages to travel from sender to receiver. Some examples of transmission media include twisted-pair wire, coaxial cable, fiber-optic cable, radio waves, microwaves and satellites. Mostly it is a combination of different transmission media.

5. **Protocol:** This is a set of rules that govern data communications. It represents an agreement between the communicating devices and without a protocol, two devices may be connected but not communicating like two people who are speaking two different languages. Most used protocols are TCP/IP, HTTPS, SMTP, IMAP, POP3.



Art Integration
Draw a creative version of the components of data communication

1.1.3 Internet

The internet is started off as "internetwork" and slowly became a globally connected network system that can transmit and receive information. It allows devices of all kinds to exchange information in real time and has found many applications that seem to be growing every day. Some of the most common applications are:

- Email and Instant messaging
- File Sharing and Data transfer
- Audio and video conferencing
- Online Forums and Social networking
- Movies, Music and Gaming
- Online shopping
- Online banking, Crypto currency and other financial services

1.2 Advantages and Disadvantages of Computer Networking

- **Data and Resource Sharing:** Files can be stored on a server or a central computer which can then be shared and made available to everyone on the network. Similarly a printer, scanner or any other resource on the network can be used by multiple devices.

- **Cost Saving:** Devices can be shared on the network for example a printer or scanner, this can save the cost of buying multiple machines.
- **Data Backup:** A network can help keep a backup of important information, so if for some reason one of the system crashes, it will be available on the network. These days it can be available on the cloud server through Google Drive, OneDrive Dropbox etc.
- **Storage capacity:** Since data can be kept on a server with enhanced storage capacity, the storage requirements of the other machines on the network will become low. Cloud storage available on the internet can increase the storage capacity manifold.
- **Sharing Security Setup:** All the devices on the network can be protected through a firewall, making them secure against virus attacks.

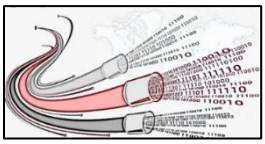
Disadvantages of Computer Networking

Some of the main disadvantages of Computer Networking are discussed below:

- **Expensive:** A network can be expensive while being setup, as the wires and the cost of the cable are high and sometimes equipment is also costly.
- **Management of the network:** Management of the network is quite difficult as it requires skilled persons to handle that large network. It requires training of people who are employed in this work.
- **Security Threats:** All networks especially Wide Area Networks(WAN), can face the threat of hacking or virus attacks.
- **Can Spread Virus:** Computer Networking can lead to the spreading of viruses to another computer through the network.
- **Loss of Information:** In case of a crash of the Computer Network, it can lead to the loss of information or not being able to access information for some time.

1.3 Data Communication Terminology

1.3.1 Communication Channel



A communication channel is used to transport an information signal, such as a digital bit stream, from one or several senders (or transmitters) to one or several receivers. The capacity of a channel to transmit information is measured by its bandwidth in Hertz or its data rate in bits per second.

The pathways used to communicate data or communication channels, use two types of media:

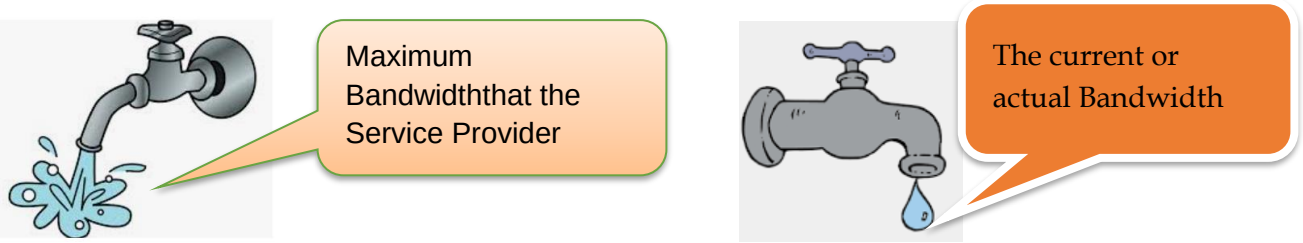
- A physical transmission medium such as a wire, or cable (twisted-pair wire, coaxial cable, and fiber-optic cable)
- A logical connection or broadcast over a multiplexed medium (microwave, satellite, radio, and infrared). The multiplexing divides the capacity of the

communication channel into several logical channels, one for each message signal or data stream to be transferred.

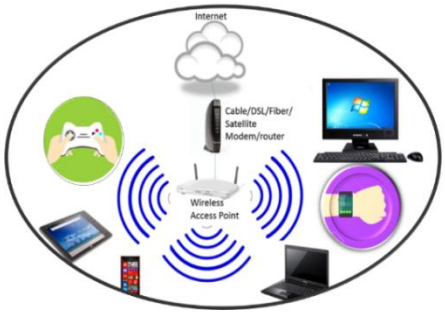
1.3.2 Bandwidth:

The amount of data that can be transferred over a network in a specific amount of time is termed as Bandwidth. There are different ways to measure bandwidth, some calculate current data flow, while others measure maximum flow, typical flow, or what could be called a good flow.

Bandwidth is expressed as a bitrate and measured in bits per second (bps), Mbps and Gbps. We can compare Bandwidth to water flowing through a tap then bandwidth is the rate at which water (data) flows through the tap (connection) under various circumstances. The amount of water(data) that possibly *can* flow through the tap represents the maximum bandwidth, while the amount of water(data) that *is currently* flowing through the pipe represents the current bandwidth.



Continuing with the above analogy, the water pipe to a home (the bandwidth) remains the same size, however when all home's taps are turned on (data downloads to the devices), the water pressure at each point ("speed" at each device) will reduce—again, because there's only so much water (bandwidth) available for the home (your network).



Sometimes when multiple people are on the same network -watching a movie, playing an online multiplayer video game, downloading files, watching online videos on their phone or syncing their smartwatch. It's likely that everyone will feel that the Wi-Fi speed is slow or constantly starting and stopping. This is because the same bandwidth is now divided among many users.

1.3.3 Data Transfer Rate

The data transfer rate (DTR) is the amount of digital data that is moved or transferred from one place to another in a given time. The data transfer rate can be viewed as the speed of travel of a given amount of data from one place to another

The data transfer rate of a network connection is measured in units of bits per second, and is abbreviated as bps instead of b/s. Network equipment manufacturers rate the maximum network bandwidth level their products support using the standard units of Kbps, Mbps, and Gbps.

- bps is **bits per second** and is the smallest unit measurement for data rate
  Kbps is one **kilobit per second equals 10^3 bits per second.**
- Mbps is one **megabit per second equals 10^3 Kbps or 10^6 bps**.
- Gbps is one **gigabit per second equals 10^3 Mbps, 10^6 Kbps or 10^9 bps**
- One **terabit per second (Tbps)** is equal to 1000 Gbps.

1 Kbps= 10^3 bps
1Mbps= 10^3 Kbps=10^6 bps
1 Gbps= 10^3 Mbps= 10^6 Kbps=10^9 bps
1Tbps=10^3 Gbps= 10^6 Mbps= 10^9 Kbps=10^{12} bps

To avoid confusion between bits and bytes, networking professionals always refer to network connection speeds in terms of bps (lowercase 'b') ratings. and not Bps which is Bytes per second.

- *one KBps equals one kilobyte per second = 8 Kbps*
- *one MBps equals one megabyte per second = 8 Mbps*
- *one GBps equals one gigabyte per second = 8 Gbps*

1.3.4 IP Address

An IP address (*internet protocol address*) is a standard numerical representation that uniquely identifies a specific interface on the network. An IP address is, generally represented as 4 octets of numbers from 0-255.

The IP address has two parts- the network part and the host part. The network part is to the left and describes the network or subnetwork in which the host is located. The host part is always on the right and describes the individual device within the IP network or IP subnet.

IP Address: 222.145.142.231

Think of the network part as a city, like Delhi and the host part is a specific location within the city for example the Red Fort.

Extension Activity

Try to find the IP address of different phones using the same Wi-Fi. Notice how only the last digit of the addresses will differ.

Do you know?

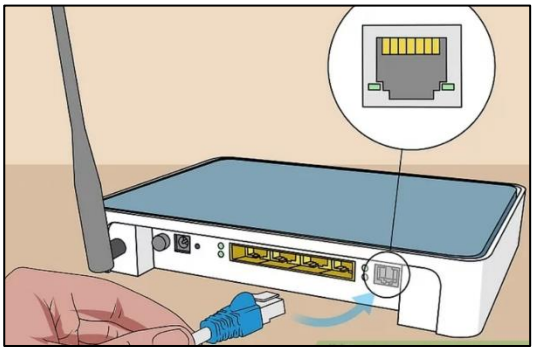
When you walk into a coffee shop, a hotel or an organization and use their Wi-Fi, you will temporarily hide your IP address and use that network's IP address as long as you are online.

1. 4 Network Devices

Modem

Modem is short for "Modulator-Demodulator" and it is a hardware component that allows a computer or another device, such as a router or switch, to connect to the Internet. It converts or "modulates" analog signals from a telephone or cable wire to digital data (1s and 0s) that a computer can recognize and it 'demodulates' or converts digital data from a computer or other devices into analog signals that can be sent over standard telephone lines.

Earlier modems were "dial-up," meaning they had to dial a phone number to connect to an ISP. These used the same frequencies as telephone calls, which limited their maximum data transfer rate to 56 Kbps. Dial-up modems also required full use of the local telephone line, meaning voice calls would interrupt the Internet connection. Modern modems are typically DSL (Digital Subscriber Line) and they operate over standard telephone lines, but use a wider frequency range



Modems are not needed for fiber optic connections because the signals are transmitted digitally from beginning to end.

Router

Routers can interconnect identical networks as well as to interconnect networks with different types of hardware. They can also connect LANs and WANs. This helps create a more efficient system of multiple smaller LANs rather than one big LAN.

Routers normally have a dynamically updating routing table based on which they make decisions on routing the data packets. Network routers receive, analyze, perform the traffic directing functions and forwards data packet from one network to its destination node.

Difference between Modem and Router

Modem	Router
Modems are commonly used to connect your home network to the internet.	Router can connect Multiple devices and routes the network traffic.
A modem converts analog to digital signals and vice versa.	Routers create and maintain a local area network.
A modem does not inspect the data packets before forwarding.	A router inspects all data packets before forwarding them.

Wired and Wireless Routers

Gateway

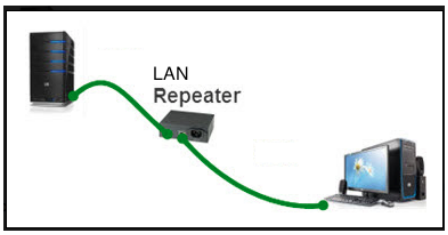
This is a device that connects two networks that may be working with different networking models. They take data from one system, interpret it, and transfer it to another system. Also called protocol converters gateways, they can operate at any network layer. Gateways are generally more complex than a switch or a router.

Repeater

As the signal is transmitted over a network it can become weak or corrupted, a repeater is a hardware device that can be used to regenerate the signal which is then copied bit by bit and regenerated to its the original strength.

This can help extend the length to which the signal can be transmitted over the same network.

However, repeaters do not amplify the signal they simply regenerate it.



Wired and Wireless Repeaters

Hub – This device is a multiport repeater and it connects multiple wires coming from different branches.

Hubs cannot filter data, so data packets are sent to all connected devices. Hubs are not intelligent devices so they cannot find out the best path for data packets, which leads to inefficiencies and wastage.

Hub

Switch – A switch is an intelligent hub with a buffer and a design that can boost its efficiency and performance. The switch can perform error checking before forwarding data, which makes it very efficient, as it does not forward packets that have errors and forwards good packets selectively to correct port only.

Switch

Difference between Hub and Switch

Hub	Switch
A hub is a repeater with multiple ports	This is a smart version of a Hub
It cannot filter data	It can perform error checking before forwarding data.
It sends the packets received to all connected devices	It has a buffer and forwards good packet selectively to correct port only.
Does not find the best path for the packets and is relatively inefficient	It boosts efficiency and performance.

Wi-Fi Card

You can think of Wi-Fi cards as being invisible cables that connect your computer to the antenna for a direct connection to the internet. Although these days most devices come preinstalled with one, but they can also be purchased relatively inexpensively and self-installed or simply inserted into a slot on the side of the computer.
Wi-Fi cards come in several forms and there are different cards for laptops, desktops and other devices.

The full form of Wi-Fi is Wireless Fidelity

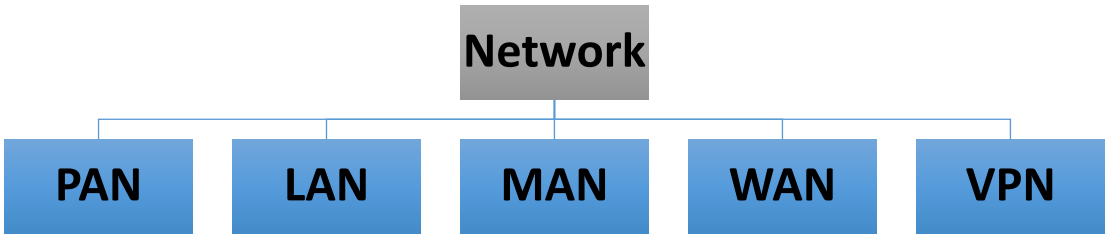
To understand the mechanism behind Wi-Fi cards, you should know how the wireless Internet itself works. Instead of transmitting data through a phone line, digital subscriber line (DSL) or high-speed cable, a wireless Internet network transmits data the same way that radios and mobile phones do, through radio waves.
Wireless signals typically travel between 23 meters to 46 meters. In a wide-open area with no obstacles, however, they have been known to transmit up to 305 meters and, with optimal conditions, a little more than 1.5 km.

Lease Line: A fixed bandwidth, dedicated connection mostly used for businesses with high data exchange requirements is a leased line. This is much more stable and reliable for providing uninterrupted, high speed, high bandwidth data. It is much more expensive than a normal broadband connection.

1.5 Types of Networks

The entire world has become interconnected through networks big and small. These networks can be categorized based on their dimension and purpose. From a network within a single home to millions of devices spread around the world they are there in every size.

The networks can be classified as:



1.5.1 PAN (Personal Area Network)

PAN is a computer network formed in small area such as room or a house in up to 10m. It generally consists of a computer, mobile, tablet, laptop etc. These personal devices can communicate among themselves or connect to the internet using PAN. It consists of personal devices placed in proximity and mostly devices like mobile phones, tablet, and laptop form a PAN. For example, transferring files between two mobile phones or from a mobile phone to a laptop or syncing a smart watch with an app on the mobile phone we create a PAN.

Advantages of PAN

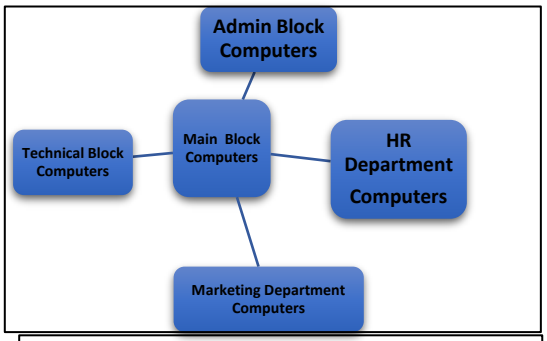
- Relatively safe and secure network
- This network is a short-range solution for up to ten meters
- Strictly restricted to a small area

Disadvantages of PAN

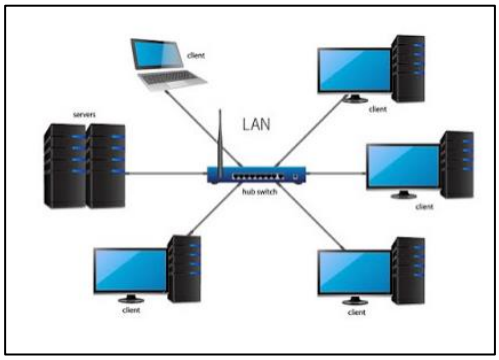
- Limited by distance
- May get interference from networks at the same radio bands.

1.5.2 LAN (Local Area Network)

This network is a group of computers and peripheral devices which are connected within a radius of 10km. It could be used in a small campus or blocks of an organization. It is used for sharing resources such as files, printers, software and applications. It is usually a private network belonging to an institution or organization.



Company Local Area Network



Office Local Area Network

Advantages of LAN

- Significantly reduces the cost of equipment and software shared over the network.
- Helps reduce redundancy of data which can stored on a single device with a backup on cloud.
- Data storage is secure and its transfer is fast and easy.

Disadvantages of LAN

- Installing a LAN can be expensive and tedious.
- Since there is a LAN admin privacy could be an issue.
- It is vulnerable to online attacks if the firewall is not in place.
- Constant maintenance and upgrade are required.

1.5.3 MAN(Metropolitan Area Network)

This network can be categorized as a large LAN which could spread over a city or region. This allows you to configure the network of up to 50 km radius using mostly optical fiber. Broadband networks provided by various companies such as Airtel, Jio, BSNL etc. within the city are examples of MAN.

Advantages of MAN

- Since it uses fiber optic cables the communication is high-speed.
- It can easily be extended to other networks and connected to wide area networks.

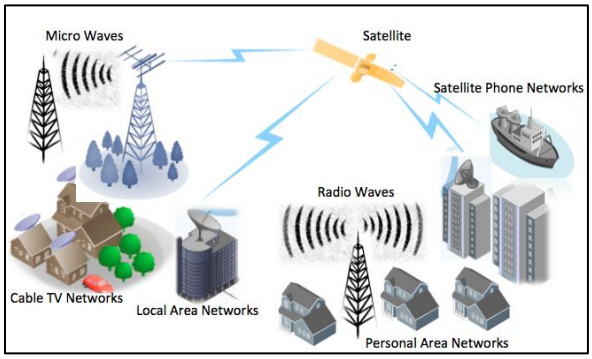
- It can be used to cover an entire city.

Disadvantages of MAN

- The extensive cabling may become very expensive.
- It is vulnerable to attacks from hackers.

1.5.4 WAN (Wide Area Network)

WAN (Wide Area Network) is another network that which is spread across a large geographical area such as cities, countries and continents. The internet or mobile phone network is an example of WAN.



WAN network system could be an extension of various MANs and LANS connected through Telephone cables, radio waves, Microwaves and satellites. It could include millions of web servers and cloud storage servers. It can make every network globally accessible.

Advantages of WAN

- The sheer coverage of WAN helps businesses communicate easily.
- Any device such as mobile phones, laptops, tablets, gaming consoles can connect to WAN.
- It can be really useful in connecting to remote and hazardous places.

Disadvantages of WAN

- Web Servers and cloud storage devices are expensive to set up initially.
- It's tough to maintain
- Is vulnerable to viruses, worms and trojans
- Is vulnerable to hackers.

Other Types of Networks: Apart from the above mentioned, there are some other important types of networks:

- VPN (Virtual Private Network
- WLAN (Wireless Local Area Network)
- (EPN) Enterprise private network
- (CAN) Campus Area Network

Comparing the Networks PAN with LAN, MAN and WAN

Parameters	PAN	LAN	MAN	WAN
Area Covered	Up to 10m radius	5-10Km radius	Covering a city (up to 100km radius)	Over cities, countries and continents
Cost	Cheap	Inexpensive	Cost Effective	Expensive
Communication Medium	Wireless or Wired	Coaxial and Ethernet	Optic Fibre, Radio waves, Microwaves	Optic Fibre, Radio waves, Satellites Microwaves
Devices used	Wi-Fi Router, Infrared, Bluetooth	Wi-Fi Router, Modem, Hub, Switch, Repeaters	Router, Gateway, Repeaters	Satellite receivers, Routers, Gateways, Radio wave and Microwave Towers

1.6 Web Architecture

Web Architecture can be defined as the conceptual structure of a network. The web architecture of the biggest network that is WWW or internet is a constantly changing medium that enables communication between different users and the technical interaction between different systems and subsystems.

Some of the main components that make up Web Architecture are:

The client −The web browser or any other application that the user uses to interact with a server is called a client. The client sends requests to the server and receives responses from the server.

The server −This is the computer or group of computers that host the website or web application. The server processes requests from the client and sends back the appropriate response.

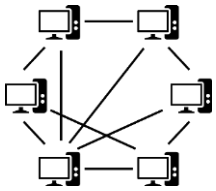
The network : The actual infrastructure that connects the client and the server and facilitates communication.

The database −The data that is stored to be retrieved by the website or web application.

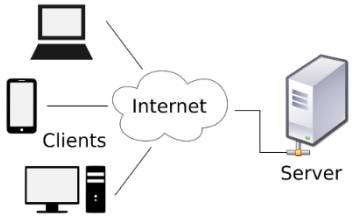
The Website or Web Application Structure−This refers to the way the website or web application is structured and organized, including the layout, navigation and overall appearance.

Types of Web Architecture

1.6.1 Peer to Peer Architecture : In a peer-to-peer (P2P) architecture two or more devices are connected and share resources without going through a separate server computer.



1.6.2 Client Server Architecture: This is an architecture in which the server hosts, delivers and manages most of the resources and services which are requested by the client. This type of architecture has one or more client computers or devices connected to a central server over a network or internet connection.



1.7 Network Threats and Security Measures

With the increase in use of the network for accessing data and resource sharing, security is becoming a prime concern. The Internet represents an insecure channel for exchanging information, which leads to a high risk of fraud, such as phishing, viruses, trojans, worms and more. In order to stay safe on any network, it is important to know the kind of risks one can face and the security measures that should be taken.

1.7.1 Virus and Malware

Virus

A virus is a software code that may harm any device by overwriting or corrupting the system files. A computer virus self-replicates by copying itself to another program, executable code or documents. The purpose of creating a computer virus is to infect vulnerable systems, gain admin control and steal user sensitive data.

One of the most common methods by which viruses spread is through emails – opening the attachment in the email, visiting an infected website, clicking on an executable file, or viewing an infected advertisement can cause the virus to spread to your system. Viruses also spread while connecting with already infected removable storage devices, such as USB drives.

It is quite easy and simple for the viruses to sneak into a computer by dodging the defence systems. A successful breach can cause serious issues for the user such as infecting other resources or system software, modifying or deleting key functions or applications and copy/delete or encrypt data.

Viruses can be classified according to the method that they use to infect a computer

- File viruses
- Boot sector viruses
- Macro viruses
- Script viruses

Malware

Malware is short for *mal*icious soft*ware*, meaning software that can be used to compromise computer functions, steal data, bypass access controls, or otherwise cause harm to the host computer. Malware is a broad term that refers to a variety of malicious programs. Malware breaches a network through a vulnerability, typically when a user clicks a dangerous link or email attachment that then installs risky software.

Once inside the system, malware can do the following:

- Blocks access to key components of the network (ransomware)
- Installs malware or additional harmful software
- Covertly obtains information by transmitting data from the hard drive (spyware)
- Disrupts certain components and renders the system inoperable

Some common types of malware are:

- Adware
- Bots
- Spyware
- Ransomware
- Viruses
- Trojan horses,
- Worms

A device infected with malware can exhibit any of the following symptoms:

- Device may slow down considerably
- Web Browser takes a long time to load sites.
- There is a problem connecting to networks
- The system keeps on crashing or hangs frequently
- Some files appear modified or deleted.
- Appearance of strange files, programs, or desktop icons
- Programs running, turning off, or reconfiguring themselves (malware will often reconfigure or turn off antivirus and firewall programs)
- Emails/messages get sent automatically and without user's knowledge

1.7.2 Phishing and Scam

Phishing

This is a cybercrime in which a person is contacted by phone, email, or text message by someone posing as a legitimate institution to lure individuals into providing sensitive data such as banking and credit card details, date of birth, personal identification number, Aadhaar card number, passwords etc.

Other than email and website phishing, there's also '**vishing**' (voice phishing), '**smishing**' (SMS Phishing) and several other phishing techniques cybercriminals are constantly coming up with.

Vishing: Voice	Smishing: SMS

Types of Phishing scams:

- **Big Prizes and Lucrative Offers:** The user gets a message that he/she has won money or an iPhone or an all-paid trip are ways to attract users to click on a link

- **Super deals only for a limited time:** One of the most common trait amongst cybercriminals is to create a sense of urgency and ask you to act fast. Messages of unbelievable deals, which are being offered for a very short period, are some ways to lure users to click and provide important personal information.
- **Misspelt popular website:** Sometimes cybercriminals come up with misspelt names of popular websites such as Government sites or retail stores or other

agencies. The site name may have one letter different but the user does not realize this and fills in their personal details on a fake site.

- **Unknown attachment:** If you see an attachment in an email, you weren't expecting or that doesn't make sense, don't open it! They often contain payloads like ransomware or other viruses.
- **Unusual Sender:** If the sender seems unusual or suspicious it's best not to open the mail.

To protect yourself against Phishing:

- Use spam filters in your email against spam or fraudulent mail.
- The settings of the web browser should allow only reliable websites to open up. Most popular Internet browsers can be customized with anti-phishing toolbars
- In case of misspelt websites, the user can report the matter to the actual website where legal actions can be taken against these fraudulent websites.
- If a link looks suspicious, hover over the URL first. Secure websites with a valid Secure Socket Layer (SSL) certificate begin with "https". Do not click on an insecure link.
- A bank will not ask for personal information via email so you should not comply with any such request.

Spam

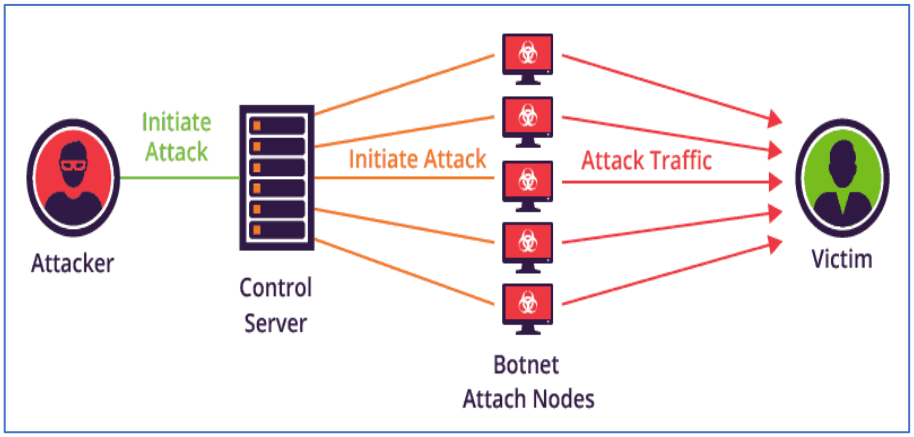
Spamming is the use of messaging systems to send multiple unsolicited messages (spam) to large numbers of recipients for the purpose of commercial advertising, or any other purpose. Spam emails are almost always commercial and driven by a financial motive. Spammers try to promote and sell questionable goods, make false claims and deceive recipients into believing something that's not true.

The most popular spam subjects include the following:

- Online Gambling and Games
- Adult Content
- Financial Services and Loans
- Online Courses and Degrees
- Work-from-home jobs
- Pharmaceutical Companies
- Cryptocurrencies

1.7.3 Denial of Service

A denial-of-service attack (DoS attack) is a cyber-attack in which the perpetrator seeks to make a machine or network resource unavailable to its intended users by temporarily or indefinitely disrupting services of a host connected to a network. Denial of service is typically accomplished by flooding the targeted network or resource with superfluous requests in an attempt to overload systems and prevent some or all legitimate requests from being fulfilled.



- The largest DoS attack to date happened in September 2017, when Google Cloud experienced an attack with a peak volume of 2.54 Tb/s(Terabytes per second).
- In February 2020, Amazon Web Services experienced an attack with a peak volume of 2.3 Tb/s.
- In February 2023, Cloudflare faced a 71 million/requests per second DOS attack.
- In October 2023 a DoS attack happened twice, once with a 201 million requests per second attack observed by Cloudflare and a 398 million requests per second attack observed by Google.

1.7.4 Security Tools

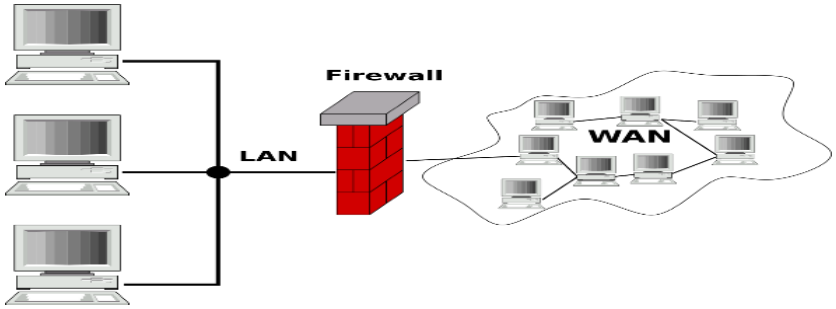
Antivirus

Anti-virus is software that aims to protect your system against malicious and potentially unwanted programs. It is responsible for detecting these malicious programs by searching for them, and removing them to keep the system protected.

The software operates by maintaining a database of malware definitions, which are automatically updated. It searches for any malicious program by scanning the files against the stored malware definitions for a match. In case of a match, they are declared as potentially harmful, and are disabled and removed depending upon anti-virus software settings.

Firewalls

A firewall aims at protecting the internal network of an organization, home, or individual from malicious traffic from external networks. A router or a computer (often dedicated to serve as a firewall) may be installed between external network and internal network for this purpose. Firewall inspects the network traffic, and allows only that data to pass through the network that does not violate the security constraint



Hardware firewall in the form of a router prevents malicious software from entering your network from outside the network. However, software firewalls installed on personal computers prevent unauthorized access or malwares from gaining access to personal computers. Network firewalls may also encrypt the incoming data by converting it to non-readable format, thus, adding further protection.

Password managers

A password manager is a software application that helps a user store and organize passwords. Password managers usually store passwords encrypted, requiring the user to create a master password; a single, ideally very strong password which grants the user access to their entire database of passwords.

Exercise

I Multiple Choice Questions

1. A number of devices connected with each other through some transmission medium is:
a)Group b)Network c)Nodes d)System

2. The information or data to be communicated through text, numbers, pictures, audio, and video is:
a) Sender b)Channel c)Message d)Receiver

3. The physical path by which a message travels from sender to receiver is:
a) Transmission medium b)Bandwidth c)Connection d)Switch

4. A set of rules that govern data communications and represent an agreement between the communicating devices is called:
a)Data Rules b)Transmission Laws c)Data Instructions d)Protocols

5. Which of these devices cannot be a sender:
a) Computer b)Stylus c)Mobile Phone d)Tablet

6. Which of the following statements are true about communication channels:
a) It is used send data or signal from one or several senders to one or several receivers.
b) A channel has a certain capacity for transmitting information
c) It can be either a physical transmission medium or to a logical connection such as a radio channel.
d) All of the above

7. The term for amount of data that can be transferred from one point to another within a network in a specific amount of time is:
a)Data transfer b)Baud c)Bandwidth d)Network Capacity

8. The speed of travel of a given amount of data from one place to another is called:
a)Data transfer Rate b)Baud c)Network Capacity d)Bandwidth

9. One terabit per second (Tbps) is not equal to:
a)10^3Gigabit per second b)10^8Megabit per second c)10^9 Kilobit per second d)10^{12} bits per second

10. Which of the following statements is incorrect:
a) The data transfer rate of a network connection is measured in units of bits per second
b) Network equipment manufacturers rate the bandwidth of products in Kbps/Mbps/Gbps.
c) 1 GBps = 1Gbps

d) Bits per second and is the smallest unit measurement for data rate

11. A numerical representation that uniquely identifies an interface or device on the network:
a)Address b)Token c)Protocol d)IP Address

12. Which of the following statements are true for IP Address:
a) IP address is represented as 4 octets of numbers from 0-255.
b) IP address is unique for a device.
c) The full form of IP is Internet Protocol.
d) All of the above

13. What is the full form of Wi-Fi?
a)Wired Fidelity b)Wireless Fidelity c)Wireless Filter d)It has no full form

14. A device that converts analog signals to digital and vice versa:
a)Router b)Switch c)Modem d)Hub

15. Routers connect:
a)LAN b) WAN c)LANs and WANs together d)None of these

16. Which of the following is not true about Gateways?
a) Not as complex as switch or router
b) Work upon different networking models
c) Are also called protocol converters
d) Take data from one system, interpret it, and transfer it to another system

17. Which network device has a buffer and performs error checking before forwarding data?
a) Hub b)Repeater c)Switch d)Router

18. LAN or Local Area Network is about
a) 10m radius b) 10 km radius c) 50 km radius d)No distance limitation

19. If you are sharing songs with your friend through Bluetooth it is a
a)LAN b)WAN c) PAN d) MAN

20. The internet is an example of :
a)LAN b)WAN c) PAN d) MAN

21. Various devices connected at home without a server is an example of :
a) Peer to Peer Architecture b)Client Server Architecture c) Wi Fi Architecture
d) None of these

22. Which of the following is not a Malware?
    a) Ransomware
    b) Viruses
    c) Trojan horses
    d) Instagram

23. _____ in the form of a router prevents malicious software from entering your network from outside the network:
a) Hardware firewall
b) Anti-Virus
c)Malware
d) Gateway

24. The _____ software operates by maintaining a database of malware definitions, which are automatically updated:
a)Operating System b)Application c)Anti-Virus d)Firewall

25. _____ is typically accomplished by flooding the targeted network or resource with superfluous requests in an attempt to overload systems and prevent some or all legitimate requests from being fulfilled.
a) Phishing b) DOS attack c) Adware d)Ransomware

II. Differentiate between

a) Modem and Router
b) Hub and Switch
c) PAN, LAN, MAN and WAN

III. Short answer questions

Q1. What is a computer network?
Q2. What is Internet?
Q3. What are some of the applications of internet?
Q4. What does the term data communication mean?
Q5. What are the five components of data sharing over a network?
Q6. What is a data communication channel?
Q7. What does the term Bandwidth signify?
Q8. Explain maximum and actual Bandwidth with the help of an example.
Q9. Why is actual Bandwidth usually less than maximum bandwidth ?
Q10. What is Data Transfer Rate?
Q11. What is a Protocol?
Q12. What is the significance of IP Addresses?
Q13. What are the different types of Computer Networks?
Q14. What is PAN (Personal Area Network)?
Q15. Describe the following Network devices briefly:

a) Repeater
b) Hub
c) Switch
d) Router
e) Gateway

Q16. What is Virus and how does it spread?

Q17. How does a firewall protect a network?

Q19. Discuss Phishing and different types of Phishing Scams

Q20. What is spamming?

IV. Answer the following Competency based / Application based Questions

1. A small organization in your neighbourhood has received a donation of 10 computers and a printer. They want to be able to offer internet and printing facilities in their newly built computer room, what suggestions can you give them. They will also need help with choosing the right network devices.
2. The internet in the computers in the school library is very slow, how can you help them identify the problem.
3. Your uncle is using online banking on his laptop, he does not have anti-virus installed, how will you convince him to have one installed.

V. Group Discussion

- Discuss the pros and cons of the Internet.
- Also discuss how IP addresses can be hidden.
- Discuss the Bandwidth plans offered by various service providers.

Chapter 2

Website Building Using HTML and CSS

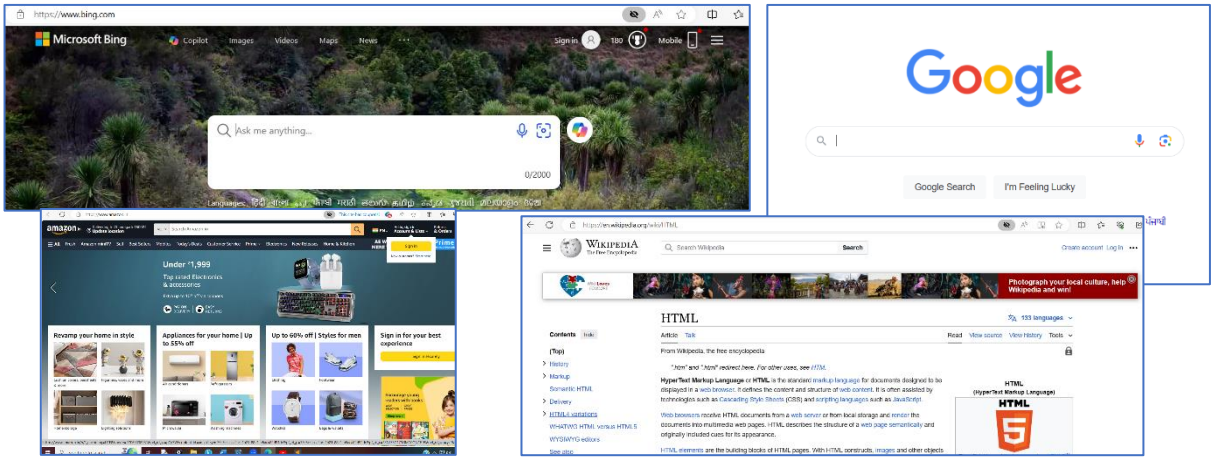
Topics Covered

2.1 Introduction to HTML
2.2 Basic Tags in HTML
2.3 Images
2.4 Lists
2.5 Tables
2.6 Div and Span Tags
2.7 Hyperlinks
2.8 Forms
2.9 Audio and Video Tags
2.10 Introduction to CSS
2.11 Advantages of CSS
2.12 Three ways to implement CSS
2.13 CSS Box Model using Div
2.14 CSS Syntax and Tags

2.1 Introduction to HTML

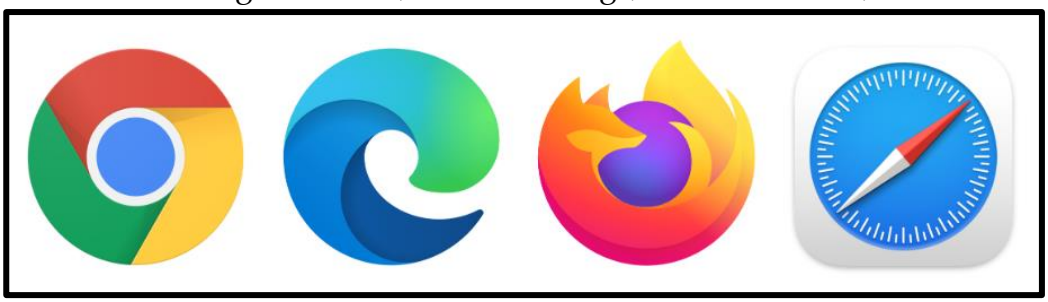
What is a website?

A website is a collection of web pages which contains the information about the particular organization or institution or any product. It contains the related content that is identified by a common domain name and published on one or many web servers.



Few examples of such websites are bing.com, wikipedia.org, google.com, amazon.com etc.

A website can be opened using the software known as Browser. Commonly used browsers are Google Chrome, Microsoft Edge, Mozilla Firefox, Safari etc.



Types of Website:

- Static Website
- Dynamic Website

A static website is one where web pages are delivered exactly as they are stored, with no real-time content changes. In contrast, a dynamic website generates content in real time, typically using databases and scripting languages to provide interactivity and personalized experiences.

Static Website	Dynamic Website
Same content is delivered every time the page is loaded.	Content may change every time the page is loaded.
No interaction with database possible.	Interaction with database possible.
It is faster to load as compared to dynamic website.	It is slower than the static website

Languages used for website development

The specific languages used for a website depend on its complexity and functionality. The most common programming languages used to develop websites are:

- **HTML (Hyper Text Markup Language) :** This forms the core structure of a web page, defining the content and its layout. Imagine it as the skeleton of a building.
- **CSS (Cascading Style Sheets) :** This language controls the visual aspects like fonts, colors, and layout that bring the website to life.
- **JavaScript :** This adds interactivity to web pages, enabling features like animations, dynamic content updates, and user input handling. It's like the electrical wiring and plumbing that make the website functional.

HTML

Hyper Text Markup Language or HTML is the foundational building block of web pages. It's a markup language, meaning it uses special tags to define the structure and content of a web page, telling the web browser how to display the information. It is a way of describing how a set of text, and images should be displayed to the viewer.

Brief History of HTML

In 1989, Tim Berners-Lee, a physicist at the European Organization for Nuclear Research, also known as CERN, wrote a memo proposing an Internet based hypertext system. In late 1990 he specified HTML and wrote the browser and server software.

The first publicly available description of HTML was a document called "HTML Tags", first mentioned on the Internet by Tim Berners-Lee in late 1991. It describes 18 elements comprising the initial,

HTML's key features:

- **Structure:** It defines the overall layout of a web page, separating it into sections like headings, paragraphs, lists, images, and more. Think of it as the skeleton of a building, providing the framework for the content.
- **Content:** While HTML doesn't directly display the content itself, it specifies the type of content within each section using various tags. For example, a <p> tag defines a paragraph, an <h1> tag defines a heading, and an <img> tag defines an image.
- **Simplicity:** HTML is a relatively easy-to-learn language with a basic syntax. It uses tags written within angle brackets (< and >) , making it accessible for beginners with no prior programming experience. It is platform-independent.
- **Collaboration:** HTML works hand-in-hand with other web development technologies like CSS (Cascading Style Sheets and JavaScript).
    - **CSS** controls the visual aspects of the website, like fonts, colors, and layout, adding the "style" to the HTML's structure.
    - **JavaScript** adds interactivity to the website, enabling features like animations, dynamic content updates, and user input handling.

HTML has no compiler or interpreter and they are browser dependent. We can create an HTML program in any text editor. Thereafter, we can run it on a browser like Edge, Chrome, etc. The file name for an HTML program should have .htm or .html extension.

Web Page

A web page is a document written in hypertext (also known as HTML) that you can see online, using a web browser. Most web pages include text, photos or videos, and links to other web pages. A group of many web pages managed by one person or company is a website. A web page consists of an HTML file with any image used on the page.

The HTML file (a normal text file) contains all the text to display and also acts as the 'glue' to hold the text and images together in the right places and display them in the right style. Writing an HTML file means composing the text you want to display, then inserting any tags you want in the right places. Tags tells a browser to do something special, like to show text in italic or bold in a larger font, show an image or to make a link to another web page.

Prerequisites to writing a HTML program

You will need a text editor, such as Notepad and an Internet browser, such as Internet Explorer or Netscape Navigator. You don't need any sophisticated software to write HTML code.

We create HTML documents using text editors and some popular editors are:

- Notepad
- Notepad++
- Sublime Text
- Text Edit(Mac OS)

HTML Tags

Tags have a simple structure and they begin with a "<" and end with a ">". Between the <> angular brackets are the tag name and may be some attributes, depending on the tag.

The general form of a tag is:
<TAGNAME ATTRIBUTEl="value1" ATTRIBUTE2="value2" ... >

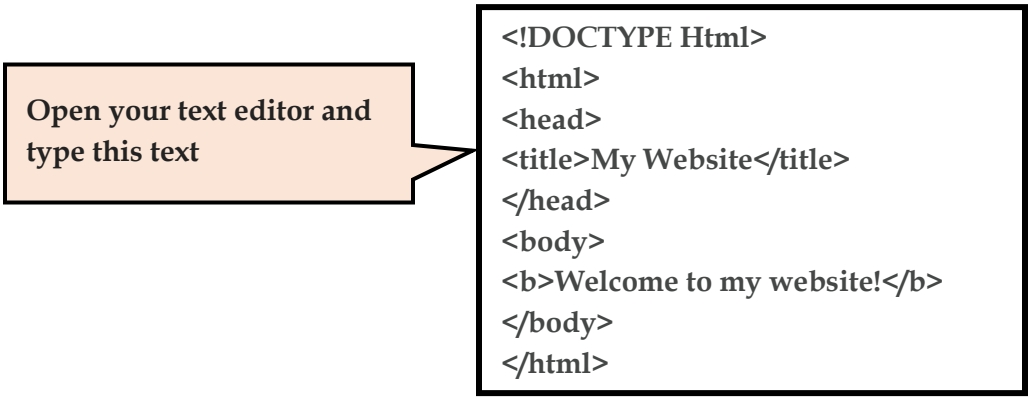
Tag names and attribute names in general are not case sensitive, but some attributes are case sensitive. The tag name must come first, but the order of the attributes doesn't matter.

HTML Element

An HTML element is defined by a start tag, some content, and an end tag. An HTML attribute provides additional information about an HTML element.

- All HTML elements can have **attributes**
- Attributes provide **additional information** about an element/tag

- Attributes are always specified in **the start tag**
- Attributes usually come in name/value pairs like: **name="value"**



Example Explained

What you just made is a basic html document. This is the minimum required information for a web document and all web documents should contain these basic components.

The <!DOCTYPE html> declaration defines that this document is an HTML5 document, this is an optional tag.

- The first tag in your html document is <html>. This tag tells your browser that this is the start of an html document.
- The last tag in your document is </html>. This tag tells your browser that this is the end of the html document.
- The text between the <head> tag and the </head> tag is header information.
- Header information is not displayed in the browser window.
- The text between the <title> tags is the title of your document.
- The <title> tag is used to uniquely identify each document and is also displayed in the title bar of the browser window.
- The text between the <body> tags is the text that will be displayed in your browser.
- The text between the and tags will be displayed in a bold font.

Running an HTML file using a browser involves a simple process:

1. **Save your HTML code :**

- Open a text editor like Notepad (Windows) or TextEdit (Mac).
- Type your HTML code in the editor.

- Save the code as a file with a .html or a .htm extension. For example, you can save it as first.html.

2. **Open the file in your browser :**

- **Double-clicking the file**: This is the most common way. In most operating systems, double-clicking the HTML file will automatically open it in your default web browser.
- **Right-click and select "Open with":** If double-clicking doesn't work or you want to use a specific browser, right-click the file, choose "Open with," and then select your preferred browser from the list.

3. **View the webpage:**

- Once you open the file in your browser, you should see the webpage rendered based on the HTML code you wrote. The browser interprets the tags and displays the content and structure as defined in your code.

Additional Tips:

- You can also drag and drop the HTML file onto your browser window to open it.
- If your code has errors, the browser might display an error message or not render the page correctly. You'll need to check your code and fix any mistakes to see the desired output.

2.2 Basic HTML Tags

Tags can be broadly divided into two categories as:

1. Container tags

These types of tags require a pair of tags that is a starting tag and an ending tag. These tags affect the content which is capsulized between the starting and ending tag. An ending tag is similar to that of a starting tag except that it begins with a slash (/) symbol. Some of the container tags are given below:

<HTML>.....</HTML>
<HEAD>.....</HEAD>
<TITLE>.....</TITLE>
<B>.....</B>

2. Empty tags

These types of tags require only a starting tag instead of a pair of tags. i.e. it does not require an ending tag. Some of the empty tags are given below:

<HR>
<IMG>
<LINK>

Some Important Tags in HTML

<HTML> Tag

The <HTML> tag identifies the document as an HTML document. <HTML> tag represents the root of an HTML document. So, it acts as the container tag for all other HTML elements. All HTML documents start with <HTML> tag and end with </HTML> tag.

Syntax

```
<HTML>
…….
other HTML Elements
……...
</HTML>
```

<HEAD> Tag

The <HEAD> tag defines the document header and does not affect the appearance of the document in the browser window. The <HEAD> element can also include a title for the document.

Syntax

```
<HEAD> Header of the document </HEAD>
```

<TITLE> Tag

The <TITLE> tag defines the title of the document. It is placed between <HEAD> and </HEAD> tags. Each document can have only one title which should identify the

document content in a general way. The title is not a part of the document text and cannot contain hypertext links or special markup commands. It must be simple text.

Syntax

<TITLE> Title of the Page </TITLE>

<BODY> Tag

The <BODY> tag defines the largest part of an HTML document, i.e. the body. This tag contains all the major content of the document such as text, images, lists, tables, hyperlinks, etc.

Syntax

<BODY> Body of the document </BODY>

HTML Background can be a color or an image and can be used as an attribute with the body tag.

Bgcolor

The bgcolor attribute specifies a background-color for an HTML page. The value of this attribute can be a hexadecimal number, an RGB value, or a color name: The lines above all set the background-color to black.

Example

```
<body bgcolor=”#000000”>
<body bgcolor=”rgb(0,0,0)”>
<body bgcolor=”black”>
```

All the above lines set the background-color to black.

Background

The background attribute can also specify a background-image for an HTML page. The value of this attribute is the URL of the image you want to use. If the image is smaller than the browser window, the image will repeat itself until it fills the entire browser window.

Example

```
<body background="http://profdevtrain.austincc.edu/html/graphics/clouds.gif">
```

Heading (H1 to H6) Tags in HTML

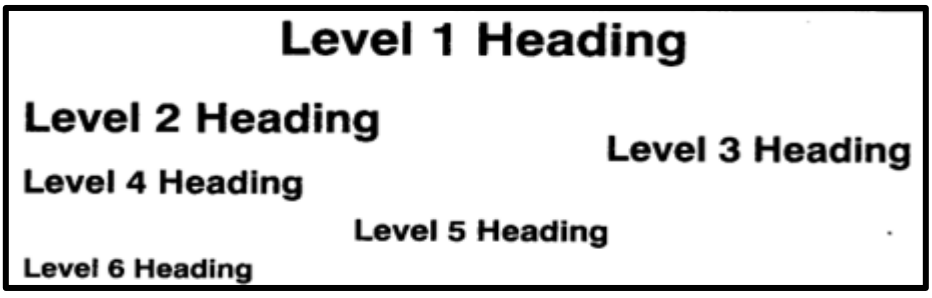
There are six levels of headings in HTML, which are numbered 1 to 6. The 1 numbered heading is the largest and bolder in fonts than normal body text. The first heading in each document should be tagged <H1>. Number 1 to 6 specifies the level of heading and attribute ALIGN sets the heading in the left side, right side or at the center.

If ALIGN = RIGHT, then the heading is aligned at RIGHT
ALIGN = LEFT, then the heading is aligned at LEFT
ALIGN = CENTER, then the heading is aligned at CENTER

H1……...H6, represents different levels of headings. They each differ by some factors such as typeface, font size and the space. Sample code to illustrate the use of headings is as follows:

```
<HTML>
<HEAD>
<TITLE> Headings in HTML </TITLE>
</HEAD>
<B0DY>
<H1 ALIGN = "CENTER"> Level 1 Heading </Hl>
<H2> Level 2 Heading </H2>
<H3 ALIGN = "RIGHT". Level 3 Heading </H3>
<H4> Level 4 Heading </H4>
<H5 ALIGN = "CENTER". Level 5 Heading </H5>
<H6> Level 6 Heading </H6>
</B0DY>
</HTML>
```

Output



Paragraph <P> Tag

This tag is used to add space between paragraphs and is a container tag. The align attribute specifies the alignment of the text within a paragraph. Use a paragraph marker <P> to end a line and also show a blank line before beginning anything else.

Syntax

<P align = "Left/Right/Center/Justify">
<P dir="ltr/rtl">

Attributes for <p> tag:

- **align:** This attribute specifies the horizontal alignment of the text within the paragraph.
  - Possible values: left (default) , center, right
- **dir:** This attribute specifies the text direction for the content within the paragraph.
  - Possible values: ltr (left-to-right) , rtl (right-to-left)

Example

```
<html>
<head>
<title>Paragraph Example</title>
</head>
<body>
<p align="center" dir="ltr">This paragraph is centered and written from left to right.</p>
<p align="right" dir="rtl"> This paragraph is right-aligned and written from right to left. .</p>
</body>
</html>
```

**Line Break
 Tag**

The
 tag (or Line Break tag) is used to simply end the current line and jump to the next line. The
 tag is useful for writing addresses split into house numbers, locality, street numbers, city etc.

Syntax

Horizontal Rule <HR> Tag

The <hr> tag in HTML is used to represent a thematic break between content sections on a webpage. It's commonly displayed as a horizontal line, visually separating different parts of your content. The <HR> tag produces a horizontal line spread across the width of the browser window.

Syntax

<HR>

Attributes

- **Align:** This attribute specifies the horizontal alignment of the line.
  - Possible values: left, center, right (default is left)
- **Size:** This attribute defines the height of the line in pixels. Default size of Horizontal Rule is 3 pixels.
- **Width:** This attribute defines the width of the line in pixels or as a percentage of the available space.
- **Color:** This attribute defines the color of the line.

Example

<hr align=”center” size = “5” width=”50%” color=”blue”>

This example will create a horizontal line that is:

- Center-aligned (align="center")
- 5 pixels high (size="5")
- Spans 50% of the available width (width="50%")
- Color is blue

<FONT> Tag

The <FONT> tag is used to change font sizes, font colors and font styles of the text in your web pages. The <FONT> tag provides no real functionality by itself but with the help of a few attributes, (i.e. size, color and face) it does so. The size, color and face attributes can all be used at once or individually, providing users with the ability to create dynamic font styles for any HTML element.

The font tag should not be used for modern web development. Instead, **you should leverage CSS properties like font-family, font-size, font-weight, color, and others to control the appearance of text in your web pages.** CSS offers more flexibility, control and separation of concerns, making it a better choice for styling.

Attributes:

Font Size

The <FONT> tag uses size attribute to specify the relative or absolute size of text. The size of font ranges from 1 (very small. to 7 (very large).

Syntax

<FONT SIZE = "2"> Text </FONT>

The Font size number from 1 to 7 defines the size of the text. By default, font size is 3

Font Color

The color attribute defines the color of the text inside a <FONT> element. To change the text color, you need to add the attribute color to the opening font tag and assign it a value for the color. The color values can be given either a standard color name e.g. red, blue, black, etc., or as RGB components e.g. # 000099. The RGB components are denoted by a preceding # sign followed by six-digit hex number.

Syntax

<FONT COLOR = "color name/RGB value of color">

Font Face

The face attribute defines the font or style of the text inside a <FONT> element. To change the font style from the default to a different style, simply add the attribute FACE to the opening FONT tag. As a value for the face attribute, you can use any specific font name such as "Verdana", "Arial" and many more. The font face value is not case sensitive.

Example

<p font size="3" color="blue" face="Arial"> It is an example using the font tag.>/p>

Comments in HTML

The comment tag is used to insert a comment in the HTML source code. A comment can be placed anywhere in the document and the browser will ignore everything inside the brackets. The comment tag in HTML is used to add explanatory notes or annotations within your HTML code. Comments are not interpreted by the browser and do not affect the appearance of your webpage. The comment has an exclamation point (!. in the opening tag, but not in the closing tag).

These comments are not displayed by the web browser when rendering the page, but they are visible when viewing the source code. To view the source code for this page, in your browser window, select View and then select Source. Do not put private information in comments, as anyone viewing the source code can easily see them. Do not put HTML tags inside your comments as well, since most browsers will think the comment ends with the first "›' character.

- Comments can make your code easier to understand for yourself and others who may need to work with it in the future.
- Comments can be helpful for communicating the rationale behind code decisions, facilitating collaboration among developers.
- Well-commented code can serve as a form of documentation, reducing the need for separate documentation files.

Example

<p> HTML Comments are used to add information, which is not meant to be viewed on the webpage<p>
<! This is a comment-->

Other HTML Tags

The <sup> and <sub> tags in HTML are used to format text, specifically for **superscript** and **subscript**, respectively. They are often used for:

- **Superscript:**
  - Denoting exponents in mathematical expressions (e.g., x^2).
  - Footnotes or endnotes referencing annotations at the bottom of the page.
  - Trademarks (e.g., TM).
- **Subscript:**
  - Chemical formulas (e.g., H_2O).
  - Units of measurement (e.g., m^2).
  - Indexing or referencing numbered items within text.

<sup> Tag (Superscript) :

- Raises the text **half a line** above the baseline of the surrounding text.
- Often rendered in a **smaller font size** for better readability.

Example

The formula for the area of a square is A ².

<sub> Tag (Subscript) :

- Lowers the text **half a line** below the baseline of the surrounding text.
- Often rendered in a **smaller font size** for better readability.

Example

The chemical formula for water is H ₂ O.

2.3 Images

The <img> tag is used to embed an image into an HTML document. Remember to replace "images/" with the actual path to your image files.

Required Attributes:

src: Specifies the path or URL to the image file. This is a required attribute.

Additional Attributes:

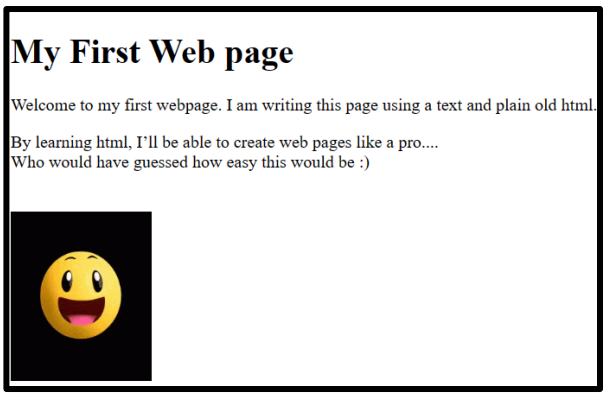
- **width:** Sets the width of the image in pixels.
- **height:** Sets the height of the image in pixels.
- **title:** Displays a tooltip when the user hovers over the image (optional) .
- **align:** An attribute for aligning the image. Set align to “left,” “center,” or “right” for horizontal alignment and “top”and “bottom” for vertical alignment.
- **border:** To add a border to an image in HTML, you can use the "border" attribute . It specifies the width of the border around the image, in pixels.
- **alt:** Provides alternative text that is displayed if the image cannot be loaded or if the user is using a screen reader. It's also essential for accessibility and for search engines to understand the image content.

Example <img src="image.jpg" alt="Example image" border="1">

Example:

```
<!DOCTYPE html>
<html>
<head>
<title>My First Webpage</title>
</head>
<body>
<h1>My First Web page</h1>
<p>Welcome to my first webpage. I am writing this page using a text
and plain old html.</p>
<p>By learning html, I'll be able to create web pages like a pro....<br>
Who would have guessed how easy this would be :. <br><br><br>

<imgsrc=C:\Users\Desktop\HTML\"Smiley.gif"  width=130  alt="Smiley" >

</body>
</html>
```

Output



2.4 Lists

Lists are essential components in HTML for organizing and presenting information in an ordered or unordered manner. They improve the readability of your web content, making it easier for users to navigate and understand the presented information.

1. Ordered List (<ol> tag) :

- Used for items that have a specific sequence or order.
- Each list item is represented by the <li> (list item) tag.
- The browser automatically numbers the items by default, starting from 1.
- These are also called numbered lists.

Attributes : Type, Start

type="1"	The list items will be numbered with numbers (default)
type="A"	The list items will be numbered with uppercase letters
type="a"	The list items will be numbered with lowercase letters
type="I"	The list items will be numbered with uppercase roman numbers
type="i"	The list items will be numbered with lowercase roman numbers

Example

```
<ol>
<li>Coffee</li>
<li>Milk</li>
<li>Tea</li>
</ol>
```

Output

1. Coffee
2. Milk
3. Tea

Example

```
<ol type="A">
<li>Coffee</li>
<li>Milk</li>
<li>Tea</li>
</ol>
```

```
<ol type="a">
<li>Coffee</li>
<li>Milk</li>
<li>Tea</li>
</ol>
```

```
<ol type="I">
<li>Coffee</li>
<li>Milk</li>
<li>Tea</li>
</ol>
```

```
<ol type="i">
<li>Coffee</li>
<li>Milk</li>
<li>Tea</li>
</ol>
```

The above HTML code in a browser looks like:

A. Coffee
B. Tea
C. Milk

a. Coffee
b. Tea
c. Milk

I. Coffee
II. Tea
III. Milk

i. Coffee
ii. Tea
iii. Milk

By default, an ordered list will start counting from 1. If you want to start counting from a specified number, you can use the **start attribute**:

```
<ol  start=2>
<li>Coffee</li>
<li>Milk</li>
<li>Tea</li>
</ol>
```

Output

2. Coffee
3. Milk
4. Tea

2. Unordered List (<ul>) tag :

- Used for items that don't have a specific order or sequence.
- Each list item is represented by the <li> tag.
- By default, unordered lists use bullet points (●) to mark each item.
- These are also called bulleted lists.

Types of Attributes for HTML Unordered List

- type ='disc' – Gives default bullet structure.
- type ='square' – Looks like solid box bullets.
- type ='circle' – Gives Hollow box structure.

Example

```
<ul>
<li>Coffee</li>
<li>Milk</li>
<li>Tea</li>
</ul>
```

```
<ul type='square'>
<li>Coffee</li>
<li>Milk</li>
<li>Tea</li>
</ul>
```

```
<ul type='circle'>
<li>Coffee</li>
<li>Milk</li>
<li>Tea</li>
</ul>
```

The above HTML code in a browser looks like:

- Coffee
- Milk
- Tea

- Coffee
- Tea
- Milk

- Coffee
- Tea
- Milk

3. Description List (<dl>**) tag :** Used for defining terms and their descriptions.

- The <dl> tag defines the entire description list.
- Each term is defined using the <dt> (definition term) tag.
- The corresponding description for each term is defined using the <dd> (definition description) tag.

Example

```
<dl>
<dt>Christmas</dt>
<dd>- A festival celebrating the birth of Jesus</dd>
<dt>Diwali</dt>
<dd>- A festival of lights celebrating the return of Lord Ram to Ayodhya</dd>
<dt>Eid</dt>
<dd>- A festival celebrated after the end of the Ramzan</dd>
</dl>
```

The above HTML code in a browser looks like:

Christmas
- A festival celebrating the birth of Jesus
Diwali
- A festival of lights celebrating the return of Lord Ram to Ayodhya
Eid
- A festival celebrated after the end of the Ramzan

2.5 Tables

HTML tables, defined using the **<table>** tag and its associated elements, provide a way to structure and present tabular data in a web page. They allow you to organize information into rows and columns, enhancing readability and clarity for users. We can create tables and display text, number, etc., in tabular form using the following:

- Tables are defined with the <TABLE> container tag.
- For adding a new row to the table use <TR> container tag (TR is for Table row) .
- For adding a new column in a row use <TH> container tag (TH is for Table Heading) or use <TD> container tag (TD is for Table Data).

```
<html>
<body>
<h3>A basic HTML table for Datesheet</h3>
<table>
<tr>
<th>Subject</th>
<th>Code</th>
<th>Date</th></tr>
<tr>
<td>Web Applications</td>
<td>803</td>
<td>March 4, 2025</td></tr>
<tr>
<td>Computer Science</td>
<td>083</td>
<td>April 5, 2025</td></tr>
</table>
</body></html>
```

Output

A basic HTML table for Datesheet

Subject	Code	Date
Web Applications	803	March 4, 2025
Computer Science	083	April 5, 2025

Here's an example of an HTML table with a border:

```
<!DOCTYPE html>
<html>
<head>
<title>Demo of an HTML Doc</title>
</head>
<body>
<h2>HTML Table</h2>
<table border=2>
<tr>
<th>School</th>
<th>Contact</th>
<th>Email</th></tr>
<tr>
<td>Green Meadows</td>
<td>Shona Manko</td>
<td>Shona@greenmeadows.org</td></tr>
<tr>
<td>Plenum International</td>
<td>Peter Chawang</td>
<td>PChawang@plenum.com</td></tr>
<tr>
<td>The Scholars Circle</td>
<td>Hardik Jagan</td>
<td>HJ@SCircle.in</td></tr>
</table>
</body></html>
```

The Output will be :

HTML Table

School	Contact	Email
Green Meadows	Shona Manko	Shona@greenmeadows.org
Plenum International	Peter Chawang	PChawang@plenum.com
The Scholars Circle	Hardik Jagan	HJ@SCircle.in

In a Table

- Each data cell contains whatever you want i.e. links, images, lists or even other tables.
- Rows are defined from top to bottom and cells are defined from left to right. If you want lines to show up between the table cells, use the border attribute in the <TABLE> tag.
- Most browsers do not require the ending </TR> or </TD> tags; they assume one cell or row ends when the next one begins. So, you might see tables written without those end tags (though the </TABLE> end tag is still required) .

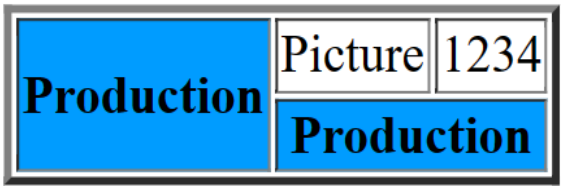
Cells that Span Multiple Columns or Rows

Sometimes, you may want one cell to span more than one column across or more than one row deep. In these cases, use the colspan and rowspan attributes of the <TD> tag. Then, just skip defining the cells that the large cell would overlay.

Example :

```
<!DOCTYPE html>
<html>
<head>
<title>My First Webpage</title>
</head>
<body>
<TABLE Border=3>
<TR>
<TH ROWSPAN=3 BGCOLOR="#99CCFF">Production</TH><TD>Picture</TD>
<TD>1234</TD>
</TR>
<TR>
<TH COLSPAN=3 BGCOLOR="#99CCFF">Production</TH></TR>
</TABLE>
</body>
</html>
```

The output will be:



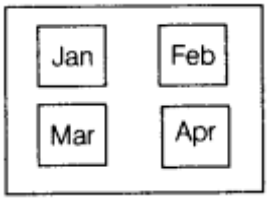
Cellspacing and Cellpadding

The cellspacing tag is used to create space between different cells within your table and the cellpadding tag controls or sets the amount of space between the contents of the cell and the cell wall. By default, cellpadding is 1. Cellpadding is usually more effective than cellspacing for spreading out the content of the table.

Example :

```
<HTML>
<Body>
<Table Border =“1” CELLSPACING = “10” CELLPADDING=“15”>
<TR>
<TD>Jan</TD>
<TD>Feb</TD>
</TR>
<TR>
<TD>Mar</TD>
<TD>Apr</TD>
</TR>
</TABLE>
</Body>
</HTML>
```

Output



Aligning Cell Contents within the Cells

Usually, all cell contents are left justified and vertically centered by default. To set the horizontal or vertical placement within the <TD> tag, use the align and valign attributes, respectively.

- align can be left, right or center.
- valign can be top, middle, bottom or baseline (aligned to baseline of the text) e.g. this borderless grocery receipt lines up the prices on the right margin

Example :

```
<TABLE>
<TR>
<TD>Laundry detergent</TD>
<TD ALIGN ="right”>Rs. 5</TD>
</TR>
<TR>
<TD>cat food</TD>
<TD ALIGN=“right”>Rs. 128.00</TD>
</TR>
</TABLE>
```

Output

laundry detergent	Rs. 5
cat food	Rs. 128.00

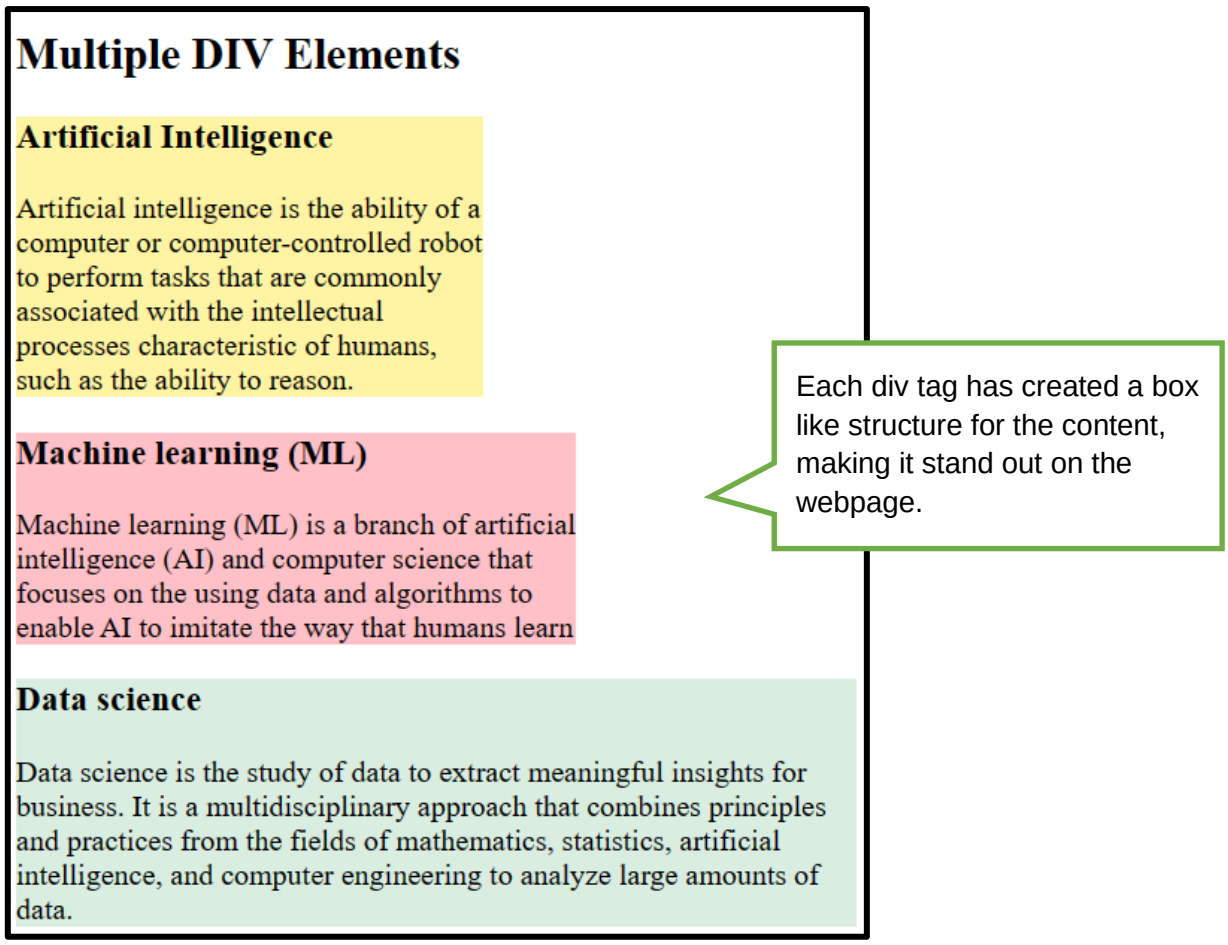
You can also use the align and valign attributes in the <TR> tag, to affect all cells in that row.

2.6 Div and Span Tags

The Website created with HTML should be well organized into sections or blocks so as to look professional.

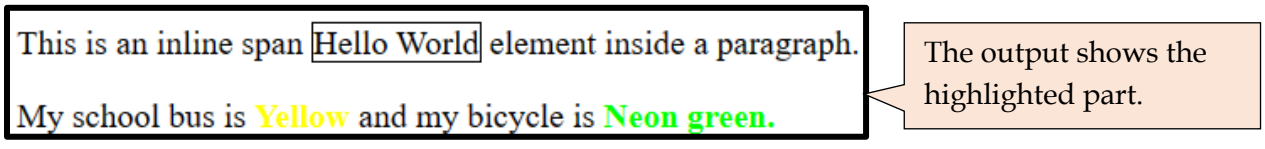
The ** tag** is an extremely important tag used as a container to group sections of a webpage together. We can use the div tag multiple times to create neat and organized sections as given in the example below:

```
<html>
<body>
<h2>Multiple DIV Elements</h2>

<div style="background-color:#FFF4A3; width:250px">
<h3>Artificial Intelligence</h2>
<p>Artificial intelligence is the ability of a computer or computer-controlled robot to perform tasks that are commonly associated with the intellectual processes characteristic of humans, such as the ability to reason.</p>
</div>

<div style="background-color:#FFC0C7; width:300px">
<h3>Machine Learning</h3>
<p>Machine learning (ML) is a branch of artificial intelligence (AI)  and computer science that focuses on the using data and algorithms to enable AI to imitate the way that humans learn</p>
</div>

<div style="background-color:#D9EEE1;width:450px">
<h3>Data Science</h3>
<p>Data science is the study of data to extract meaningful insights for business. It is a multidisciplinary approach that combines principles and practices from the fields of mathematics, statistics, artificial intelligence, and computer engineering to analyze large amounts of data.</p>
</div>
</body>
</html>
```



The **…… element is** also used to make some part of a paragraph stand out. It is an inline element, and will not start on a new line and only takes up as much width as necessary.

```
<html>
<body>
<p>This is an inline span <span style="border: 1px solid black">Hello World</span> element inside a paragraph.</p>
<p>My school bus is  <span style="color:Yellow;font-weight:bold;">Yellow</span> and my bicycle is <span style="color:lime;font-weight:bold;"> Neon green.</span></p>
</body>
</html>
```



2.7 Hyperlinks

In HTML, links or hyperlinks are created using the**<a> (anchor) tag**. They allow users to navigate between different web pages or sections within the same page, fostering interactivity and providing access to additional information. Almost any web content can be converted to a link, so that when clicked (or otherwise activated) it will make the web browser go to another web address (URL). A link does not have to be text. It can be an image or any other HTML element.

<a> tag and its attributes:

- **href (required)** : Specifies the URL of the target resource (web page, file, etc.) that the link points to.
- **target (optional)** : Defines where the linked resource should be opened.

The **href attribute** is the cornerstone of creating hyperlinks or links in HTML using the <a> (anchor) tag. It dictates the destination or target resource the link should point to when the user clicks on it.

Example

 Visit Wikipedia

Types of Links:

Absolute links: Specify the full URL (including protocol, domain name, and path of the target resource (e.g., href="https://www.example.com/page.html") .

Relative links: Specify the path relative to the current page's location (e.g., href="about.html") assuming "about.html" is in the same directory as the current page.

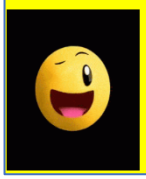
Using Image as a Hyperlink

```
<HTML>
<HEAD><TITLE> Welcome to links </TITLE></HEAD>
<Body BGCOLOR=”CYAN”>
<FONT SIZE=10>
<a href=”RegistrationForm.html”> Click here to Register </a><br><br><br>
Or click on the winking smiley below to Register! <br><br>
</FONT>
<a href=”RegistrationForm.html”><img src=”Smiley.gif”></a>
</BODY>
</HTML>
```

Output

Click here to Register

Or click on the winking smiley below to Register!



The image is also a hyperlink

Why are HTML links called hyperlinks?

- You can click on a link and jump to another document.
- When you move the mouse over a link, the mouse arrow will turn into a little hand.
- You can have different colours for links and visited links.

Using the target Attribute

The target attribute specifies where to open the linked document.
The target attribute can have one of the following values:

- _blank - Opens the linked document in a new window or tab
- _self - Opens the linked document in the same window/tab/frame as it was clicked (this is default)
- _parent - Opens the linked document in the same window where the link was clicked.
- _top - Opens the linked document in the full body of the window.

```
<html>
<head><title>Hyperlink</title></head>
<body>
<p>
<a href="https://www.cbse.nic.in" target="-self">CBSE homepage using self-
</a><br><br>
<a href="https://cbse.nic.in " target="_blank"> Visit CBSE!</a> CBSE homepage in a
anew tab</a><br><br>
<a href="https://cbse.nic.in " target="_top"> CBSE homepage in full window </a>
</p>
</body>
</html>
```

There are two types of linking in HTML - Internal and External.

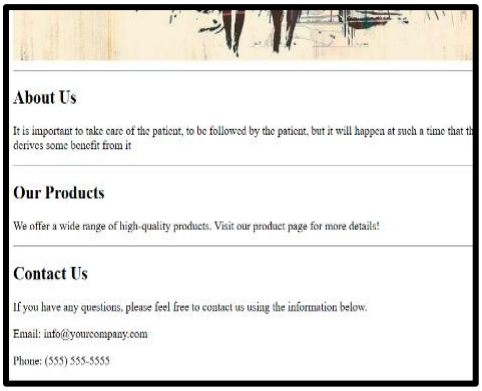
Internal Linking : A web page is linked within the same web page. It is done by using an absolute path or relative path of a link. The internal link name is followed by the hash sign(#) . It is done by assigning an id to refer to a section of the webpage, which is referred to as an internal link to the same page.

Syntax:

Example:

```
<html>
<body>
<h1>Welcome to our Website!</h1>
<p>We offer a variety of content. Explore different sections using the links below:
</p>

<ul>
  <li><a href="#about">Learn About Us</a></li>
  <li><a href="#products">See Our Products</a></li>
  <li><a href="#contact">Contact Us</a></li>
</ul>

<img src=company.jpg>

<h2 id="about">About Us</h2>
<p>  It is important to take care of the patient, to be followed by the patient, but it will happen at such a time that there is a lot of work and pain. For to come to the smallest detail, no one should practice any kind of work unless he derives some benefit from it </p>

<hr>

<h2 id="products">Our Products</h2>
<p>We offer a wide range of high-quality products. Visit our product page for more details!</p>

<hr>

<h2 id="contact">Contact Us</h2>
<p>If you have any questions, please feel free to contact us using the information below.</p>
<p>Email: info@yourcompany.com</p>
<p>Phone: (555.  555-5555)</p>

</body>
</html>
```

Output:



This code will display a webpage with:

- A heading "Welcome to our Website!"
- A paragraph introducing the website's content.
- An unordered list containing three links:
  - "Learn About Us" (jumps to the section with the id "about").
  - "See Our Products" (jumps to the section with the id "products").
  - "Contact Us" (jumps to the section with the id "contact") .
- A horizontal line for separation.
- Three sections with headings (About Us, Our Products, Contact Us. identified by their respective IDs)

Clicking on the links in the list will jump you to the corresponding section on the same web page, demonstrating internal linking.

External Linking

The Anchor Tag and the “href” Attribute An anchor can point to any resource on the Web: an HTML page, an image, a sound file, a movie, etc.

Syntax:
 Text to be displayed
Example
 Visit Swayam!

The <a> tag is used to create an anchor to link from, the href attribute is used to tell the address of the document or page we are linking to, and the words between the open and close of the anchor tag will be displayed as a hyperlink.

Example:

```
<!DOCTYPE html>
<html>
<body>
<h1>My Favorite Websites</h1>
<ul>
  <li><a href="https://www.w3schools.com/">W3Schools - Learn HTML, CSS, JavaScript</a></li>
  <li><a href="https://www.youtube.com/">YouTube - Watch Videos Online</a></li>
  <li><a href="https://www.wikipedia.org/">Wikipedia - The Free Encyclopedia</a></li>
</ul>
</body>
</html>
```

Output:
This code will display a webpage with the following elements:

1. **Heading:** "My Favorite Websites"
2. **Unordered List:** A list with three links:

- W3Schools-Learn HTML, CSS, JavaScript(link to https://www.w3schools.com/)
- YouTube - Watch Videos Online (link to YouTube: https://www.youtube.com/)
- Wikipedia - The Free Encyclopedia (link to https://www.wikipedia.org/)

2.8 Inserting Audio and Video in a webpage

The audio and video elements are embedded using the dedicated audio and video tags, respectively. These tags offer a standard and flexible way to incorporate multimedia content into your web pages.

1. Audio:

- Use the <audio> tag to embed audio content.
- You can specify multiple audio sources using the <source> tag within the <audio> element. This allows you to provide different audio formats (e.g., MP3, Ogg for wider browser compatibility)

Example:

```
<audio controls>
<source src="audio.mp3" type="audio/mpeg">
<source src="audio.ogg" type="audio/ogg">
Your browser does not support the audio element.
</audio>
```

- The <audio> tag defines the audio element.
- The controls attribute adds playback controls (play, pause, volume. to the audio player)
- Each <source> tag specifies an alternative audio source:
  - src attribute defines the path to the audio file.
  - type attribute indicates the audio format (e.g., "audio/mpeg", "audio/ogg") .
- The text between the opening and closing <audio> tags is displayed if the browser doesn't support the audio element.

2. Video:

- Use the <video> tag to embed video content.
- Similar to the audio element, you can use multiple <source> tags within the <video> element to offer different video formats (e.g., MP4, Webm for better compatibility)

Example:

```
<video controls width=”640” height=”480”>
<source src=”video.mp4” type=”video/mp4”>
<source src=”video.webm” type=”video/webm”>
Your browser does not support the video element.
</video>
```

- The <video> tag defines the video element.
- The controls attribute adds playback controls (play, pause, volume. to the video player)
- Each <source> tag specifies an alternative video source:
  - src attribute defines the path to the video file.
  - type attribute indicates the video format (e.g., "video/mp4", "video/webm")
- The width and height attributes are optional and specify the initial size of the video player.
- The text between the opening and closing <video> tags is displayed if the browser doesn't support the video element.

You can use other attributes with both audio and video tags for further customization, such as:

- autoplay: Starts playback automatically.
- loop: Loops the audio/video playback.
- muted: Starts playback muted.

Example

```
<!DOCTYPE html>
<html>
<body>
<h2>Media Tags</h2>

<audio controlsautoplay>
  <source src="your_audio.mp3" type="audio/mpeg">
  <source src="your_audio.ogg" type="audio/ogg">
  Your browser does not support the audio element.
</audio>
<br>
<video width="400" height="300" controlsautoplay>
  <source src="your_video.mp4" type="video/mp4">
  <source src="your_video.webm" type="video/webm">
  Your browser does not support the video element.
</video>
</body>
</html>
```

You can use loop or muted in place of autoplay

You can use loop or muted in place of autoplay

- Replace "your_audio.mp3" and "your_audio.ogg" with the actual filenames of your audio files.
- Replace "your_video.mp4" and "your_video.webm" with the actual filenames of your video files.

This code will display two sections on your webpage:

1. **Audio Example:** This section will embed an audio player. If your browser supports the audio format (MP3 in this case) , you'll see controls to play, pause, and adjust the volume. Browsers that don't support MP3 will try the OGG format. If neither format is supported, a message will display indicating that the audio element is not supported.
2. **Video Example:** This section will embed a video player with controls to play, pause, adjust volume, and control playback in full screen. Similar to the audio, the browser will try the MP4 format first and then WebM. Browsers that don't support either format will display a message indicating that the video element is not supported.

2.9 Forms

HTML forms provide a way for users to interact with your web page by submitting information. They consist of various input elements that allow users to enter data, along with other elements to structure and submit the data. It is a container element started by <FORM> tag and ended by </FORM>tag. It is used to create a form on a web page.

Syntax:

<FORM NAME=”FormName” ACTION=”URL” METHOD=”method”>

Attributes of FORM element are discussed below:

1. NAME -This specifies the name of the form. But this name will not be displayed on the form. As there can be more than one FORMs in an HTML document, a name is required to differentiate one form from another. The NAME attribute is optional if there is only one FORM on the web page.

NAME = "FormName"

2. ACTION - This specifies the URL where the form-data is sent when the form is submitted.This URL is also called the destination of the form.

ACTION = "URL"

3. METHOD - This specifies how the form-data is submitted. Form-data can be submitted using the methods get or post. With METHOD = "get", the form-data is submitted as URL variables, and with METHOD = "post", the form-data is submitted as an HTTP post.

METHOD = "method"

Input Elements

The <form> tag in HTML is essential for creating interactive web pages where users can submit information. Within the form, various input elements allow users to enter and interact with data.

- Various input elements allow users to enter data:
  - type="text": Single-line text input.
  - type="password": Password input (text is masked) .
  - type="email": Email address input.
  - type="checkbox": Checkbox for selecting/deselecting options.
  - type="radio": Radio buttons for selecting one of multiple options.
  - type="file": File upload input.
  - Many other types exist, like text area for multi-line text, select for dropdowns, etc.
- Each input element typically has attributes like name (to identify the data , value (pre-filled value , and others specific to the type)

1.Text Input (type="text")

- Creates a single-line text field for users to enter free-form text.

Example:

```
<label for=”name”>Name:</label>
<input type=”text” id=”name” name=”name” placeholder=”Enter your name”>
```

- **label**: Associates the label text "Name:" with the input field for better accessibility.
- **id**: Unique identifier for the input element (optional. .
- **name**: Attribute used to identify the data submitted from this field (important for processing. .
- **placeholder**: Sets a hint or placeholder text displayed within the field until the user enters their input.

2. Password Input (type="password")

- Creates a single-line text field where the entered characters are masked (usually with dots or asterisks) for security reasons, typically used for passwords.

Example:

```
<label for=”password”>Password:</label>
<input type=”password” id=”password” name=”password” required>
```

- **required**: Ensures the user must enter a value before submitting the form.

3. Email Input (type="email")

- Creates a single-line text field where the user can enter an email address. Browsers may perform basic validation (e.g., checking for "@" and "." symbols) .

Example:

```
<label for=”email”>Email:</label>
<input type=”email” id=”email” name=”email” required>
```

4. Checkbox (type="checkbox")

- Creates a checkbox element that the user can click to select or deselect an option. Multiple checkboxes can be used within a form to allow users to choose multiple options.

Example:

```
<label for=”agree”> I agree to the terms and conditions.</label>
<input type=”checkbox” id=”agree” name=”agree”>
```

- When checked, the submitted data for the checkbox will typically be a value like "on" or "true".

5. Radio Button (type="radio")

- Creates a radio button element. Multiple radio buttons with the same name attribute can be used within a group, where only one selection can be made at a time.

Example:

```
<p>Select your preferred language:</p>
<label for="lang_en">English</label>
<input type="radio" id="lang_en" name="language" value="english" checked>
<label for="lang_es">Spanish</label>
<input type="radio" id="lang_es" name="language" value="spanish">
```

- **value**: Assigns a specific value to be submitted when the radio button is selected.
- **checked**: Makes the first radio button selected by default (optional).

6. Text Area (type="textarea")

- Creates a multi-line text input field for users to enter extended text, often used for comments, feedback, or longer descriptions.

Example:

```
<label for="message">Message:</label>
<textarea id="message" name="message" rows="5" cols="30" required></textarea>
```

- **rows**: Sets the number of visible rows for the text area.
- **cols**: Sets the number of visible columns for the text area.

7. File Upload (type="file")

- Creates a button or input field that allows users to select a file from their local device for upload to the server.

Example:

```
<label for="file">Upload a document:</label>
<input type="file" id="file" name="document">
```

Submit Button: Creates a button that triggers form submission.

- <input type="submit" value="Submit">

Forms start with the <FORM> tag and end with the </FORM> tag. In the form, you can still put any HTML code you want, but you can also use these tags to define input fields that contain interactive controls that enable a user to submit information to a web server. Let's look at some examples:

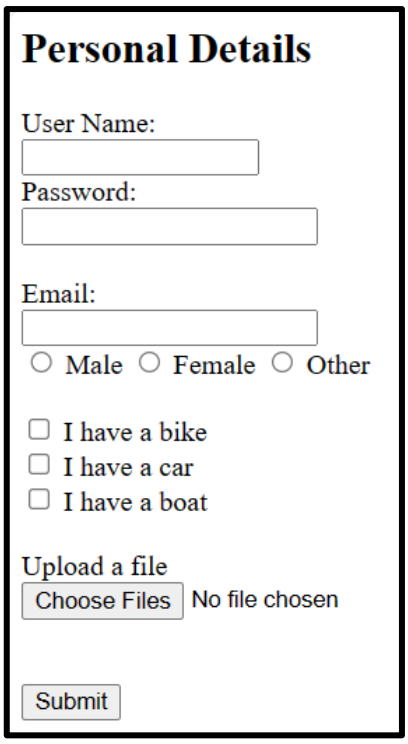
Example 1:

```
<html>
<head>
<title>Form in HTML</title>
</head>
<body>
<form>
<label for="firstname">First Name: </label><br/>
<input type="text" id="firstname" name="firstname"/><br/>
<label for="lastname">Last Name: </label><br/>
<input type="text" id="lastname" name="lastname"/><br/>
<br>
<label for="gender">Gender: </label>
<input type="radio" id="gender" name="gender" value="male"/>Male
<input type="radio" id="gender" name="gender" value="female"/>Female <br/>
<br>
    Game Chosen:<br>
<input type="checkbox" id="cricket" name="cricket" value="cricket"/>
<label for="cricket">Cricket</label><br>
<input type="checkbox" id="football" name="football" value="football"/>
<label for="football">Football</label><br>
<input type="checkbox" id="hockey" name="hockey" value="hockey"/>
<label for="hockey">Hockey</label>
<br><br>
     Enter your address:<br>
<textarea rows="2" cols="20"></textarea>
<br><br>
<input type="submit" value="Enter your Info">
</form>
</body></html>
```

Example 2:

```
<html>
<body>
<h2>Personal Details</h2>
<form>
<label for="name">User Name:</label><br>
<input type="text" id="name" name="name"  maxlength="20" size="15" required><br>

<label for="password">Password:</label><br>
<input type="password" id="pwd" name="pwd" value=""><br><br>

<label for="email">Email:</label><br>
<input type="email" id="email" ><br>

<input type="radio" id="male" name="gender" value="male">
<label for="male">Male</label>
<input type="radio" id="female" name="gender" value="female" >
<label for="female">Female</label>
<input type="radio" id="other" name="gender" value="other">
<label for="other">Other</label><br><br>

<input type="checkbox" id="vehicle1" name="vehicle1" value="Bike">
<label for="vehicle1"> I have a bike</label><br>
<input type="checkbox" id="vehicle2" name="vehicle2" value="Car">
<label for="vehicle2"> I have a car</label><br>
<input type="checkbox" id="vehicle3" name="vehicle3" value="Boat">
<label for="vehicle3"> I have a boat</label><br><br>

<label for="files">Upload a file</label><br>
<input type="file" id="files" multiple><br><br><br>

<input type="submit" value="Submit">

</form>
</h3>
</body>
</html>
```

Output:



Special Characters

How do you display the < and > characters? If you just type them in your HTML file, the browser will think you are starting or ending a tag. You have got to escape the characters, as it's called, by typing special sequences of characters in their place. When displaying your page, the browser translates the sequences back into the characters you need.
All special character sequences start with *&' (ampersand) and end with ; (semicolon) and in between is the name of the special character, e.g. '>' means the greater than symbol, '<' means the less than symbol, '"' means double quotes and '&' means the ampersand itself, e.g. the line to display the < character, use the sequence <.

Example

```
<html>
<body>
<p>Displaying &euro;
<p>Displaying &gt and &lt;</p>
<p>Displaying &quot;</p>
</body>
</html>
```

The output will be:

Displaying €

Displaying > and <

Displaying "

2.10 Introduction to CSS

CSS stands for Cascading Style Sheets and it is used for creating a unique style for your website. This means that you can specify in advance how the different headings, paragraphs, tables, lists, forms etc. will look. This will bring uniformity and make your website look really professional. Using the latest version of CSS, you can create websites that will look great on any device that is desktop, laptop, tablet or smartphone.

Here's a look at two very popular websites without CSS

1. 

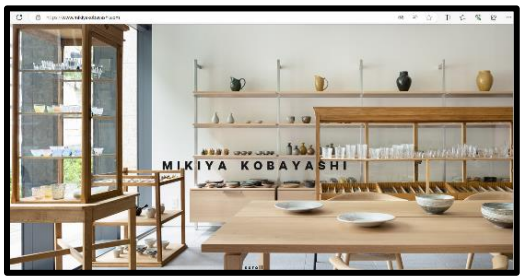
2. 

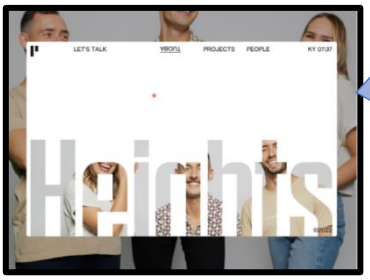
If you couldn't guess the first one is Wikipedia while the second one is Amazon.
Hope you understand the importance of CSS in making the websites glamourous!!

While Hypertext Markup Language (HTML) is used to structure a web document (defining things like headlines and paragraphs, and allowing you to embed images, video, and other media) , Cascading Style Sheet language comes through and specifies your document's style — page layouts, colors, and fonts etc.

2.11 Advantages of CSS

Improves the aesthetics of a Website: Nobody wants to look at an unattractive, dull and boring looking unprofessional website, so to attract traffic to your website, you need an attractive website. Without CSS that's not possible.





Most professional websites use a lot of style sheets.

Easy updates across multiple pages: By specifying the styles of each HTML element using a style sheet, we don't need to spend time changing each element individually. We can easily change the style sheet and it gets reflected everywhere in the website.
Faster loading times: With style sheets created in .css files, the website becomes less heavy and its loading time becomes faster.

Responsive Websites: When our website opens in different devices such as laptops, tablets, phones etc., the size of the screen changes and this can make the website look bad. The biggest advantage of CSS is its ability to create responsive designs that adapt to various screen sizes in devices.

Did You Know?
Responsive Web Design(RWD) makes your web page look good on all devices and it uses only HTML and CSS.
Scan the QR code to learn more.

2.12 Three ways to implement CSS

The very first thing we need to do is to tell the HTML document that we have some CSS rules we want it to use. There are three different ways to apply CSS or to insert a style sheet in an HTML document:

- External CSS – Inserting a .css file Link Tag
- Internal CSS- Inserting info within a Style Tag
- Inline CSS– Using style attribute within a Tag

External CSS

Create a file in the same folder as your HTML document and save it as styles.css. The .css extension shows that this is a CSS file.
Then give the following tag within the head section of the Html file.

```
<link rel="stylesheet" href="styles.css">
```

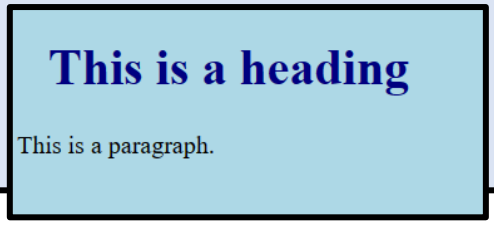
This **<link>** element tells the browser that we have a style sheet, using the **rel** attribute, and the location of that style sheet as the value of the **href** attribute.

External CSS Example:

```
<!DOCTYPE html>
<html>
<head>
<link rel="stylesheet" href="mystyle.css">
</head>
<body>
<h1>This is a heading</h1>
<p>This is a paragraph.</p>
</body>
</html>
```

The "mystyle.css" file

```
body {
  background-color: lightblue;
        }
h1,h3,h5,hr {
  color: navy;
  margin-left: 20px;
}
```



Internal CSS : An internal style sheet may be used if one single HTML page has a unique style. The internal style is defined inside the <style> element, inside the head section.

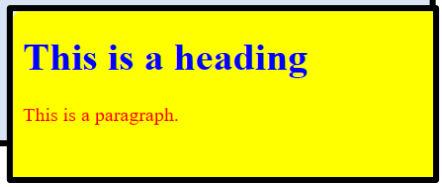
Internal CSS Example:

```
<!DOCTYPE html>
<html>
<head>
<style>
body {
  background-color: linen;
}
h1 {
  color: maroon;
  margin-left: 40px;
}
</style>
</head>
<body>
<h1>This is a heading</h1>
<p>This is a paragraph.</p>
</body>
```

Inline CSS: An inline style may be used to apply a unique style for a single element.To use inline styles, add the style attribute to the relevant element. The style attribute can contain any CSS property.

Inline CSS Example:

```
<!DOCTYPE html>
<html>
<bodybgcolor=”Yellow”>

<h1 style="color:blue;text-align:center;">This is a heading</h1>
<p style="color:red;">This is a paragraph.</p>

</body>
</html>
```



Multiple Style Sheets

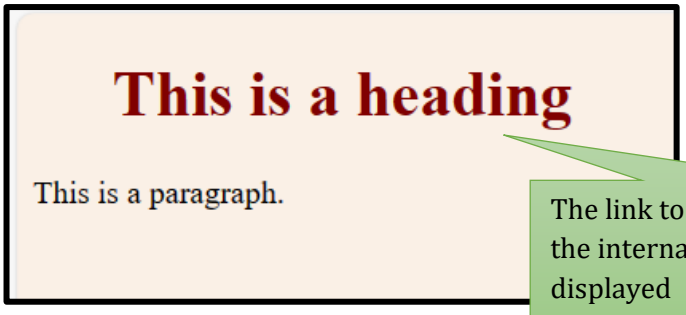
What style will be used when there is more than one style specified for an HTML element?

All the styles in a page will "cascade" into a new "virtual" style sheet by the following rules, where number one has the highest priority:

1. Inline style (inside an HTML element)
2. External and internal style sheets (in the head section whichever comes later)
3. Browser default

So, an inline style has the highest priority, and will override external and internal styles and browser defaults.

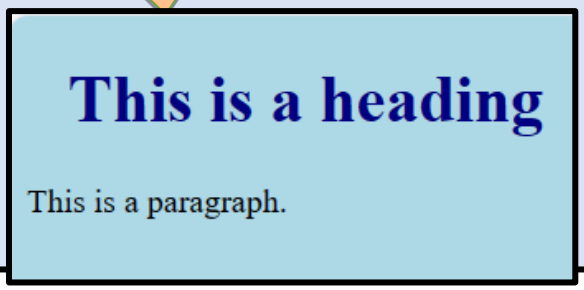
```
<!DOCTYPE html>
<html>
<head>
<link rel="stylesheet" href="mystyle.css">
<style>
body {
  background-color: linen;
}
h1 {
  color: maroon;
  margin-left: 40px;
}
</style>
</head>
<body>
<h1>This is a heading</h1>
<p>This is a paragraph.</p>
```



The link to the external style sheet is followed by the internal style sheet so its style will be displayed

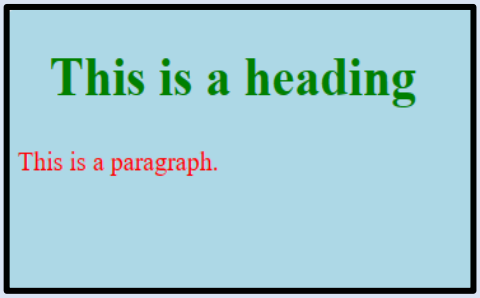
```
<!DOCTYPE html>
<html>
<head>
<style>
body {
  background-color: linen;
}
h1 {
  color: maroon;
  margin-left: 40px;
}
</style>
<link rel="stylesheet" href="mystyle.css">
</head>
<body>
<h1>This is a heading</h1>
<p>This is a paragraph.</p>
```

The link to the external style sheet is later so it will be displayed




```
<!DOCTYPE html>
<html>
<head>

<style>
body {
  background-color: linen;
}
h1 {
  color: maroon;
  margin-left: 40px;
}
</style>
<link rel="stylesheet" href="mystyle.css">
</head>
<body>
<h1 style="color:green; text-align:left;">This is a heading</h1>
<p style="color:red;">This is a paragraph.</p>
</body>
</html>
```

The inline style will be displayed for heading and paragraphs. The background colour will be displayed from the external style sheet since there is no inline stvle in the

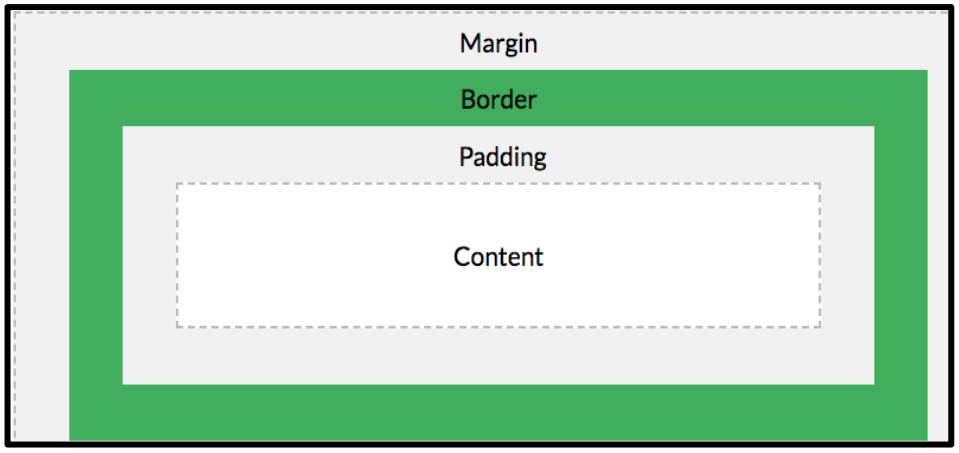


2.13 The CSS Box Model using div

In CSS, the term "box model" is used when talking about design and layout. The CSS box model is essentially a box that wraps around every HTML element. It consists of:

- **Content** - The content of the box, where text and images appear
- **Padding** - Clears an area around the content. The padding is transparent
- **Border** - A border that goes around the padding and content
- **Margin** - Clears an area outside the border. The margin is transparent

The image below illustrates the box model:



It is implemented in CSS by defining properties of the tag within the style tag.

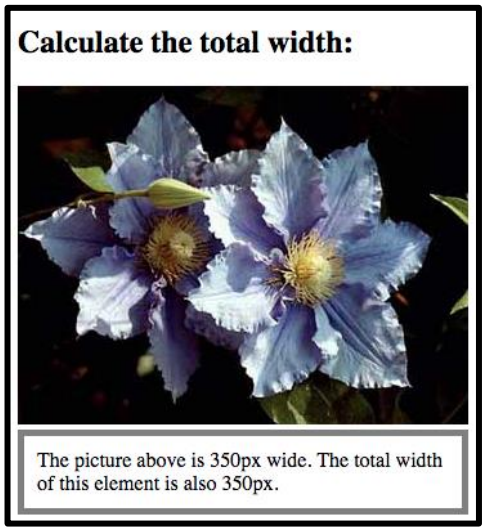
```
div {
  width: 320px;
  padding: 10px;
  border: 5px solid gray;
  margin: 0px;
}
```

The box model allows us to add a border around elements, and to define space between elements.

Setting the Width and Height of an Element using the box model

In order to set the width and height of an element correctly in all browsers, you need to calculate the padding and margins accordingly.

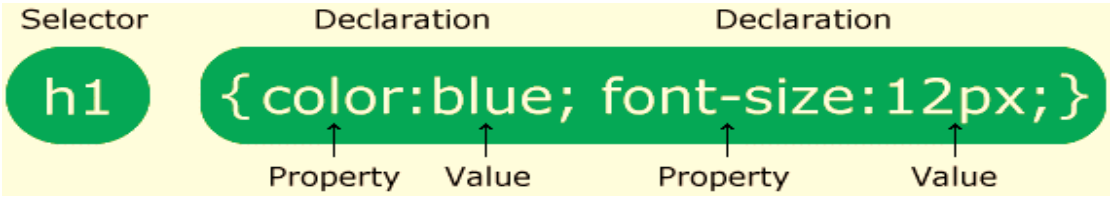
```
<!DOCTYPE html>
<html>
<head>
<style>
div {
  width: 320px;
  padding: 10px;
  border: 5px solid gray;
  margin: 0;
}
</style>
</head>
<body>

<h2>Calculate the total width:</h2>

<imgsrc="klematis4_big.jpg" width="350" height="263" alt="Klematis">
<div>The picture above is 350px wide. The total width of this element is also
350px.</div>

</body>
</html>
```



320px (width)
+ 20px (left + right padding)
+ 10px (left + right border)
+ 0px (left + right margin)
= 350px

2.14 CSS Syntax and Tags

- A CSS rule consists of a selector and a declaration block.
- The selector points to the HTML tag you want to style.
- The declaration block contains one or more declarations separated by semicolons.
- Each declaration includes a CSS property name and a value, separated by a colon.
- Multiple CSS declarations are separated with semicolons, and the entire declaration block is surrounded by curly braces.



CSS Comments

Comments are used to explain the code, and may help when you edit the source code at a later date. Comments are ignored by browsers. A CSS comment is placed inside the <style> element, and starts with /* and ends with */. We can have single line, multi-line comments and comments in the same line as attributes of styles.

```
<!DOCTYPE html>
<html>
<head>
<style>
p {
  color: blue;  /* Set paragraph color to blue */
}
/* This is
a multi-line
comment */
</style>
</head>
<body>
<!-- These paragraphs will be blue -->

<p>Hello World!</p>
<p>This paragraph is styled with CSS.</p>
<p>CSS comments are not shown in the output.</p>
</body></html>
```

Hello World!

This paragraph is styled with CSS.

CSS comments are not shown in the output.

Some commonly used CSS tags:

CSS Properties for text in Paragraphs, Body, Headings etc.	
CSS property	**Description**
text-align	This property is implemented to specify a horizontal alignment in your text and can be values left, right, center or justify.
color	This property is implemented for assigning colours to your texts.
text-indent	The text-indent property can be applied to indent the text of your paragraph, heading etc.
text-transform	This CSS property is implemented for specifying the uppercase as well as the lowercase letters of your text.
line-height	This CSS line-height property is applied to assign space between lines.
letter-spacing	The letter-spacing property is implemented for specifying the space involving the characters within your HTML text.
word-spacing	The word-spacing property is implemented for specifying the space involving the words between your texts.

CSS Properties for Lists.	
CSS property	**Description**
list-style-image	This property specifies an image as the list item marker.
list-style-type	This property specifies the type of list item marker.

CSS Properties for Tables.	
CSS property	**Description**
border	This property specifies the border in pixels.
width	This property specifies the span/width of the screen as %.
text-align	This property specifies the horizontal alignment of the text in the columns.
vertical-align	This property specifies the vertical alignment of the text in the columns.
height	This property specifies the height of the cell in pixels.
padding	This property is used to control the space between the border and the content in a table

Some examples of the CSS properties for HTML elements:

```
p {
   text-align:left;
text-indent: 40px;}
body {
   text-align: center;
  color: red;}
h2 {
   text-align: right;
  color: aqua;}
```

```
table {
  border: 1px solid;
  width: 100%;}
th, td {
  text-align: center;
  height: 50px;
  vertical-align: bottom;
  padding: 15px;}
```

CSS Colors

With CSS, colors can be specified in different ways:

- As RGB values
- By color names
- As hexadecimal values

As RGB values

RGB color values are supported in all browsers.

An RGB color value is specified with: rgb(RED , GREEN , BLUE)

Each parameter defines the intensity of the color as an integer between 0 and 255.

For example, rgb(0,0,255) is rendered as blue, because the blue parameter is set to its highest value (255) and the others are set to 0.

Color	RGB	Color
	rgb(255,0,0)	Red
	rgb(0,255,0)	Green
	rgb(0,0,255)	Blue

As Hexadecimal Values

Hexadecimal color values are also supported in all browsers.

A hexadecimal color is specified with: #RRGGBB)

RR (red) , GG (green) and BB (blue) are hexadecimal integers between 00 and FF specifying the intensity of the color.

For example, #0000FF is displayed as blue, because the blue component is set to its highest value (FF) and the others are set to 00.

Color	HEX	RGB	Color
	#FF0000	rgb(255,0,0)	Red
	#00FF00	rgb(0,255,0)	Green
	#0000FF	rgb(0,0,255)	Blue

As Color Names : The color names are supported by All Browsers. Modern browsers support 140 color names, some of these are:

Some examples of CSS : Creating a List using an image instead of bullets:

```
<!DOCTYPE html>
<html>
<head>
<style>
ul {  list-style: url("flower.jpeg". ;}
li:hover{background-color: grey;}
</style>
</head>
<body>
<h2>The list-style Property</h2>
<ul>
<h3>
<li>Coffee</li>
<li>Tea</li>
<li>Coca Cola</li>
<h3>
</ul>
</body>
</html>
```

list-style: url("flower.jpeg") will display flowers instead of bullets
li:hover will help highlight the row of the list when the pointer hovers over it.

The list-style Property

- Coffee
- Tea
- Coca Cola

The list-style Property

- Coffee
- Tea
- Coca Cola

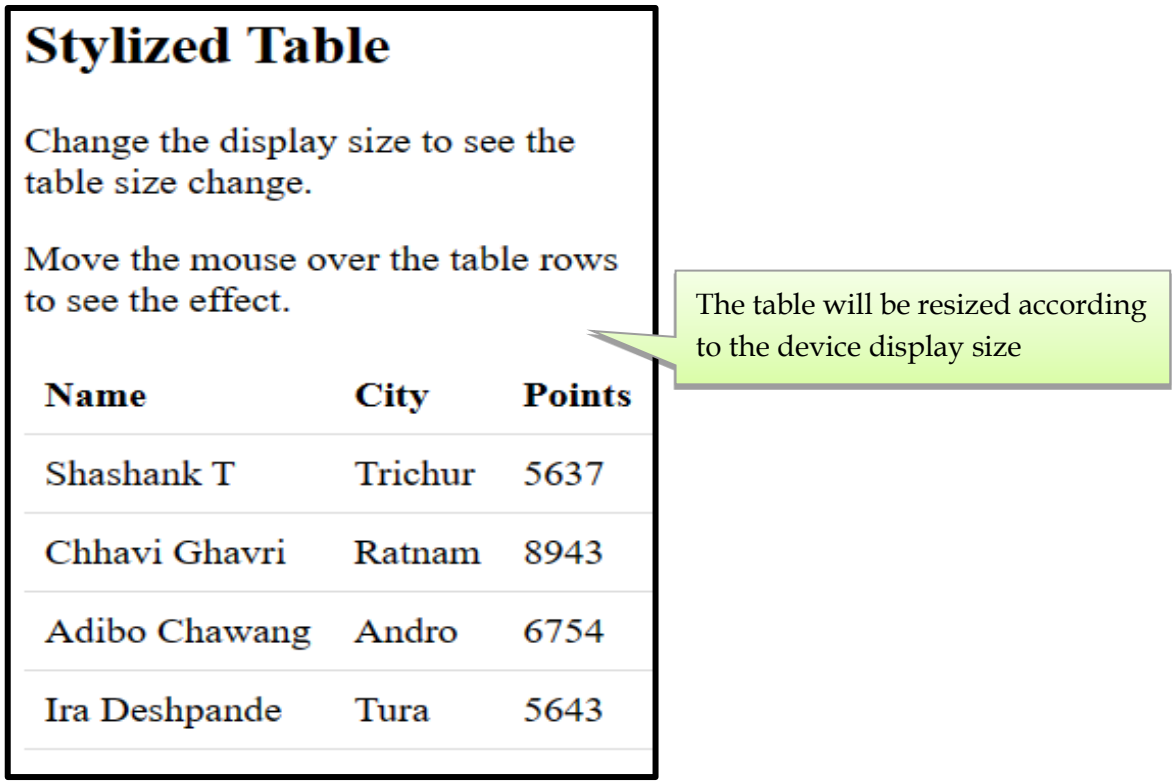
Creating a table that will resize the display in different devices and the row gets highlight when the mouse hovers over it.

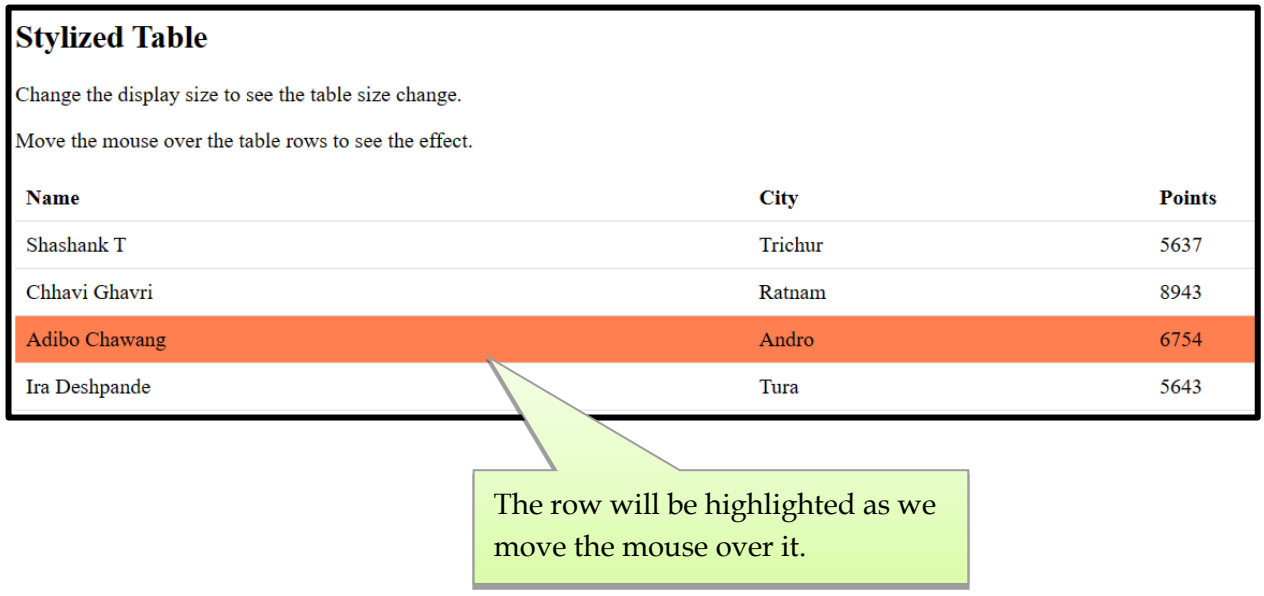
```
<html>
<head>
<style>
table {
  border-collapse: collapse;
  width: 100%;}
th, td {
  padding: 8px;
  text-align: left;
  border-bottom: 1px solid #ddd;}
tr:hover {background-color: coral;}
</style>
</head>
<body>
<h2>Stylized Table</h2>
<p>Change the display size to see the table size change.</p>
<p>Move the mouse over the table rows to see the effect.</p>
<table>
<tr>
<th>Name</th>
<th>City</th>
<th>Points</th></tr>
<tr>
<td>Shashank T</td>
<td>Trichur</td>
<td>5637</td></tr>
<tr>
<td>Chhavi Ghavri</td>
<td>Ratnam</td>
<td>8943</td></tr>
<tr>
<td>AdiboChawang</td>
<td>Andro</td>
<td>6754</td></tr>
<tr>
<td>Ira Deshpande</td>
<td>Tura</td>
<td>5643</td></tr>
</table>
</body></html>
```

border-collapse: collapse will show single line border.

This property will help resize the table according to the device display size.

tr:hoverwill help highlight the row of the table in coral colour when the pointer hovers over it.


Stylized Table
Change the display size to see the table size change.
Move the mouse over the table rows to see the effect.
Name City Points
Shashank T Trichur 5637
Chhavi Ghavri Ratnam 8943
Adibo Chawang Andro 6754
Ira Deshpande Tura 5643
The table will be resized according to the device display size


Stylized Table
Change the display size to see the table size change.
Move the mouse over the table rows to see the effect.
Name City Points
Shashank T Trichur 5637
Chhavi Ghavri Ratnam 8943
Adibo Chawang Andro 6754
Ira Deshpande Tura 5643
The row will be highlighted as we move the mouse over it.

Exercise

I. Multiple Choice Question

1. Which HTML tag defines the title of the webpage (displayed on the browser tab) ?
a. <name>
b. <title>
c. <heading>
d. <webpage_title>

2. Which HTML tag defines a paragraph?
a. <para>
b. <paragraph>
c. <p>
d. <text>

3. How do you make text bold in HTML?
a. <bold>text</bold>
b. text
c. <bl>text</bl>
d. Both and <bl> can be used.

4. How do you create a horizontal line separator in HTML?
a. <line>
b. <hr>
c. <separator>
d. <divider>

5. Which HTML tag is used to create an image?
a. <picture>
b. <image>
c. <img>
d. <visual>

6. What attribute is required in the <img> tag to specify the image source?
a. <source>
b. <path>
c. <file>
d. <src>

7. Which HTML tag defines a comment that is not displayed on the webpage?
a. <!—comment -->
b. <comment>text</comment>
c. // comment
d. <hidden>text</hidden>

8. Which HTML tag is used to format text as italic?
a. <i>
b. <italic>
c. <italics>
d. <it>

9. Which HTML tag defines a table element?
a. <table_data>
b. <table>
c. <data_table>
d. <TD>

11. What tag defines a table row?
a. <tr>
b. <row>
c. <td>
d. <TR>

11. What tag defines a table header cell?
a. <td>
b. <th>
c. <header_cell>
d. <TDHeader>

12. To span a cell across multiple columns, which attribute would you use on the <td> or <th> tag?
a. rowspan
b. colspan
c. merge
d. spread

13. By default, do tables have borders around them?
a. Yes
b. No
c. Depends on the browser
d. Only if a border attribute is set

14. Which HTML tags define ordered lists?
a. and
b. and
c. and
d. <item> and </item>

15. Which HTML tags define unordered lists?
a. and
b. and
c. and
d. <item> and </item>

16. What tag defines a list item within an ordered or unordered list?
a. and
b. and
c. and
d. <item> and </item>

17.How can you specify a different numbering style for an ordered list ?
a. Use the style attribute.
b. Use a separate tag for different numbering styles.
c. Ordered lists only support numeric styles.
d. Use the type attribute with the <ol> tag.

18. What attribute is used to specify a starting number for an ordered list?
a. start
b. value
c. begin
d. number

19. What symbol is used as a bullet point by default in unordered lists by default?
a. Square (❑).
b. Circle (●).
c. Disc (◦).
d. Triangle (▲).

20. What is the correct way to create an ordered list with numbered items?
a. <list type="ordered">...</list>
b. ...
c. <numbered_list>...</numbered_list>
d. <olist>...</olist>

21. Which HTML tag is used to create a hyperlink?
a. <url>
b. <link>
c. <a>
d. <href>

22. What attribute specifies the destination URL of a hyperlink?
a. <destination>
b. <href>
c. <url>
d. <linkto>

23. Which attribute specifies the text displayed for the hyperlink?
a. <text>
b. The text is automatically generated by the browser.
c. The text is defined within the <a> tag itself.
d. <content>

24. What happens when a user clicks on a hyperlink?
a. The browser displays an error message.
b. The browser opens the document specified in the <href> attribute.
c. The text of the hyperlink is highlighted.
d. The behaviour depends on the specific browser settings.

25. Can hyperlinks link to local elements within the same webpage?
a. No, hyperlinks can only link to external websites.
b. Yes, you can use an ID selector within the <href> attribute to link to a specific element on the same page.
c. You can only link to external websites and other webpages on the website.
d. We need other scripting languages for linking on the same webpage.

26. How can you specify that a link should open in a new browser tab or window?
a. By using a specific attribute with the <a> tag (e.g., <a target="_blank">. .
b. There is no way to control how links open in HTML.
c. This behaviour depends on the user's browser settings.

27. What is the difference between an absolute URL and a relative URL in the context of hyperlinks?
a. Absolute URLs are shorter and easier to remember.
b. Absolute URLs specify the complete web address (including protocol, domain name, and path. , while relative URLs are relative to the current webpage's location.
c. Absolute URLs cannot be used in HTML tags.
d. Relative URLs specify the complete web address (including protocol, domain name, and path. , while absolute URLs are relative to the current webpage's location.

28. Which HTML tag creates a hyperlink (link to another webpage or resource) ?
a. <link>
b. <url>
c. <a>
d<connect>

29. Which HTML tag defines the start and end of a form?
a. <form_data>
b. <form>
c. <input_form>
d. <data_collection>

30. What attribute specifies the method used to send form data (e.g., GET or POST) ?
a. <action>
b. <method>
c. <data_type>
d. <send_method>

31. What attribute specifies the URL of the program that will process the submitted form data?
a. <destination>
b. <action>
c. <processor>
d. <submit_to>

32. Which HTML tag is used to create a text input field in a form?
a. <text_input>
b. <input type="text">
c. <data_entry>
d. <field>

33. What attribute is used to specify the label for a form element?
a. <label_text>
b. <label for="element_id"> (with corresponding element ID).
c. <data_name>
d. There is no way to add labels to form elements in HTML.

34. How can you create a radio button input field in a form?
a. <radio_button>
b. <input type="radio">
c. <option type="radio">
d. <choice>

35. What is the purpose of the <textarea> element in a form?
a. To create a single-line text input field.
b. To create a multi-line text input field.
c. To define a label for another form element.
d. To upload a file.

36. How can you allow users to upload a file through a form?
a. By using a specific attribute with the <input> tag (e.g., <input type="file">. .
b. There is no way to allow file uploads in HTML forms.
c. This functionality requires additional scripting beyond HTML.

37. What is the difference between the GET and POST methods for form submission?
a. There is no functional difference.
b. GET appends form data to the URL, while POST sends data separately.
c. GET is for sending data to the server, while POST is for retrieving data from the server.
d. GET is more secure than POST.

38. What happens when a user submits a form?
a. The form disappears from the webpage.
b. The form data is sent to the program specified in the <action> attribute for processing.
c. The browser displays a confirmation message.
d. The behaviour depends on the specific form elements used.

39. What does CSS stand for?
a. Creative Style Sheets
b. Computer Style Sheets
c. Cascading Style Sheets
d. Colorful Style Sheets

40. What does the "Cascading" in CSS stand for?
a. The ability to apply multiple styles to an element
b. The hierarchical structure of HTML elements
c. The priority system for resolving conflicting styles
d. The process of applying styles from parent to child elements

41. Which tag do you use to include an external CSS file in your HTML document?
a. <css> tag
b. <style> tag
c. <link> tag
d. <script> tag

42. Which CSS property is used to change the text color of an element?
a. text-color
b. color
c. font-color
d. text-style

43. Which CSS property is used to control the spacing between lines of text?
a. line-height
b. text-spacing
c. line-spacing
d. text-line

44. Which CSS property specifies the type of list item marker?
a. list-style-type
b. list-style-image
c. list-style
d. list-type

45. Which CSS property is used to specify the space between contents and the border of a table?
a. border-spacing
b. cell-padding
c. padding
d. border-collapse

II. Answer the following questions:

1. Describe the purpose of the <head> and <body> sections in an HTML document.
2. Explain how comments are used in HTML code. What is their purpose?
3. Describe the purpose of a form in HTML. What are some common form elements used for user input?
4. Explain the difference between the GET and POST methods for submitting form data. When might you use each method?
5. You are creating a feedback form for your school website. What HTML elements would you use to collect user information like name, email, and their feedback message?
6. Explain how you would use an HTML table to display a timetable for your school classes.
7. Imagine you are creating a webpage for a fictional band. How would you use HTML elements to embed a music player and display the band's upcoming tour dates?
8. Describe the difference between heading tags (H1-H6) in HTML. How would you use them to structure your webpage content?
9. Explain how to create a bulleted list and a numbered list in HTML. What are the advantages of using lists for web page content?
10. How can you create a hyperlink in HTML? Explain the different attributes you can use with the <a> tag.

11. Describe how to create and format a table in HTML. What are the benefits of using tables for web page content?
12. What is CSS? Give its three advantages.
13. Differentiate between Inline, Internal, and External CSS.
14. What is the purpose of the <style> tag in HTML?
15. Explain the concept of the CSS Box Model and its components.
16. How can you change the font size, color, and background color of elements using CSS?
17. What are the three different ways to give colors in CSS?
18. How can you create a border around an image using CSS?

III. Do the following practically

1. Create a webpage for a fictional restaurant. Include:

- A heading with the restaurant's name.
- An image of the restaurant's logo.
- A navigation bar with links to sections like "Menu," "About Us," and "Contact Us" (don't worry about creating the linked pages yet).
- A short description of the restaurant's cuisine.

2. Create a webpage for a musician showcasing their work. Include:

- A heading with the musician's name.
- An image of the musician.
- A section with a brief biography using semantic tags (e.g., <h2>Biography</h2> followed by paragraphs).
- An embedded audio player to play some of their music (you can search for embedding options from online music services).
- Links to social media profiles.

3. Design a product page for an online store. Include:

- A product image with a title and description.
- A drop-down menu to select variations (e.g., size, color).
- A button to add the product to the cart.
- A table displaying product specifications (e.g., material, dimensions).

4.You're building a simple website for a bakery. The home page should have a large heading with the bakery's name and a short description underneath.

- Create a basic HTML structure for the heading and description using appropriate HTML elements (e.g., <h1>, <p>)
- Write external CSS code to style the heading and description:

  - Set a large font size and a bold font weight for the heading.
  - Set a smaller font size for the description.
  - Make the heading text a dark brown color (#8B4513) and the description text a medium grey (#696969).
  - Center both the heading and description horizontally on the page.

5. You are designing a personal website and want to create a stylish contact section. Use CSS to style the contact section with the following features:

- Heading: Create a heading "Get in Touch" with a large font size and a cool blue color (#3399FF).
- Contact Information: Style a list of contact details including email address and phone number. Use bullet points and make the text slightly smaller than the heading.
- Social Media Links: Include icons for popular social media platforms like Facebook, Twitter, and Instagram. Style the icons to have a consistent size and spacing.

- Remember to link your external CSS file to your HTML document.
- You can find free social media icons online in various formats (PNG, SVG).
- Use properties like color, font-size, to style the contact information list.

CHAPTER 3
Multimedia Design Using GIMP

Topics Covered:

3.1 Multimedia Design Tools
3.2 Introduction to GIMP
3.3 GIMP Tool Box
3.4 Filters
3.5 Working with Layers

3.1 Multimedia Design Tools

Multimedia is made up of two words- **MULTI + MEDIA**.

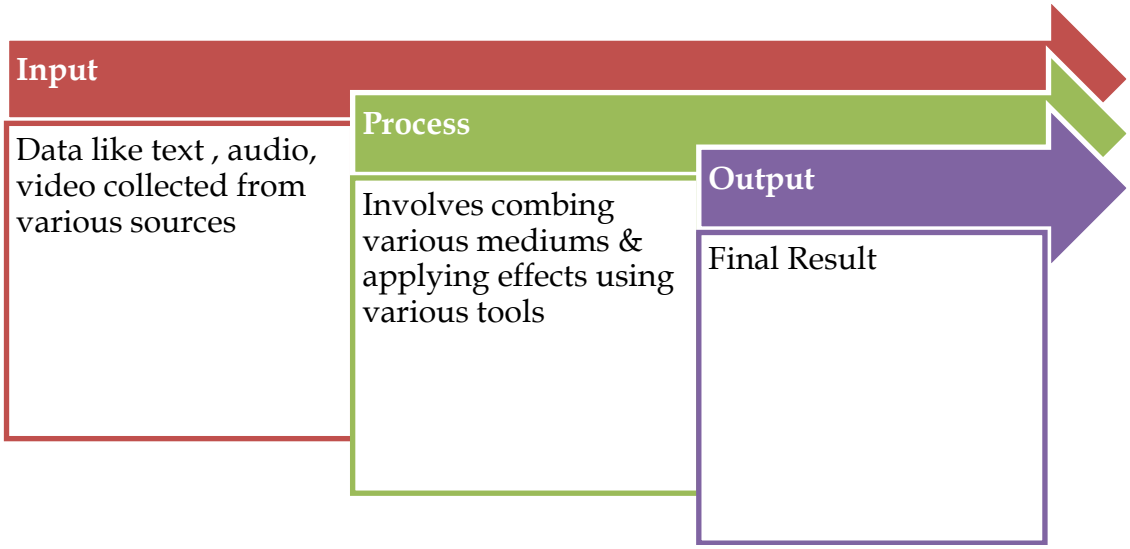
- **Multi** means **Many**
- **Media** means **Medium of Communication**

To share knowledge with one another, we have to communicate. Multiple communication channels are part of multimedia. Media such as text, photos, music, video, and animation can all be used to communicate.

Multimedia design is the skill of combining several types of media. In addition to many other interactive applications, it is utilized in marketing, advertising, video games, television, movies, websites, and informative material.

Integrating two or more media types needs technical and creative abilities in multimedia design.

Rendering the **input** to multimedia design tools and **processing** it to produce the **final output** is part of multimedia development.



Why is Multimedia important?

Multimedia content helps to

- Enhanced engagement.
- Improved understanding
- Better knowledge retention
- Cater to diverse learning styles.
- Improved Problem Solving

It's important to understand the fundamentals of multimedia design, such as layout and composition and color theory, in order to produce multimedia content that is effective by adding:

- Typography
- Graphics and Pictures
- Animation and Video

To produce engaging and useful digital material, multimedia designers need to possess both technical and artistic abilities. This is a dynamic and exciting area. You may produce visually appealing and interactive multimedia material that can be utilized for a variety of reasons by learning the fundamentals of multimedia design and becoming proficient with the tools and software used in the industry.

You might have noticed that photographs in marriage albums, magazines or catalogues of any cosmetic company are so different from common pictures. A normal looking person looks beautiful, dull skin turns bright; wrinkles disappear from the face of old & eventually a person looks stunning than actual.

Can you imagine how does it happened? Well, this alteration is made possible with the image editing software.

An image editor is a software program to edit & manipulate an image or graphic. You can change the attributes of an image, like size, colour, effects, contrast, brightness etc. Image editing software is also known as Photo Editing software.

In this chapter, our focus will be on learning open-source multimedia design software GIMP for manipulating and enhancing multimedia features of images.

3.2 Introduction to GIMP

GNU Image Manipulation Program is referred to as GIMP. Versions for Mac, Windows, and Linux are available. It is used for specialized activities such as picture retouching and editing, free-form sketching, cropping, and resizing, and converting between various image formats. Both professional and beginner photographers will find this program to be very useful. In addition, anyone who uses picture editing software for fun as well as graphic designers, artists, and website developers can utilize it.

Spencer Kimball and **Peter Mattis** were the developers of GIMP. Since then, updates have been made on a regular basis. At the time of writing this book the latest GIMP version available is 2.10.36.

Features of GIMP:

Some features of GIMP are:

- Has user friendly interface.
- Image editing can be done in less time & with little effort.
- It has powerful tools to:
  - change the background and captions to your photographs.
  - enhance or change the colour of an image by adjusting the brightness & contrast, colour balance, hue and saturation levels, curves etc.
  - crop or resize pictures without losing the quality of an image.
  - merge two or more images.
- Has a full screen editing mode, which make it possible to enhance all the elements of the image.
- Supports a wide range of file formats, including gif, jpeg, png, tiff, bmp etc. GIMP allows the creation of animated gifs by placing each frame of an animation in its own layer.
- To save disk space, any format can be saved in GIMP with an extension, such as zip, gz. This compresses the files and reduces its size. These files are suitable for web-based applications.

Installing GIMP:

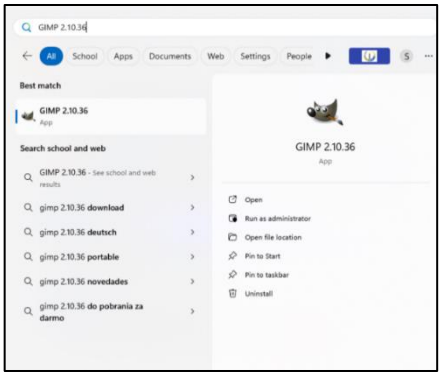
To install GIMP visit its site https://www.gimp.org/downloads/ .

- Select your operating system from:

Show downloads for GNU/Linux | macOS | **Microsoft Windows** | All

- Click on download.
- Run the executable file from the downloads.

Starting GIMP:

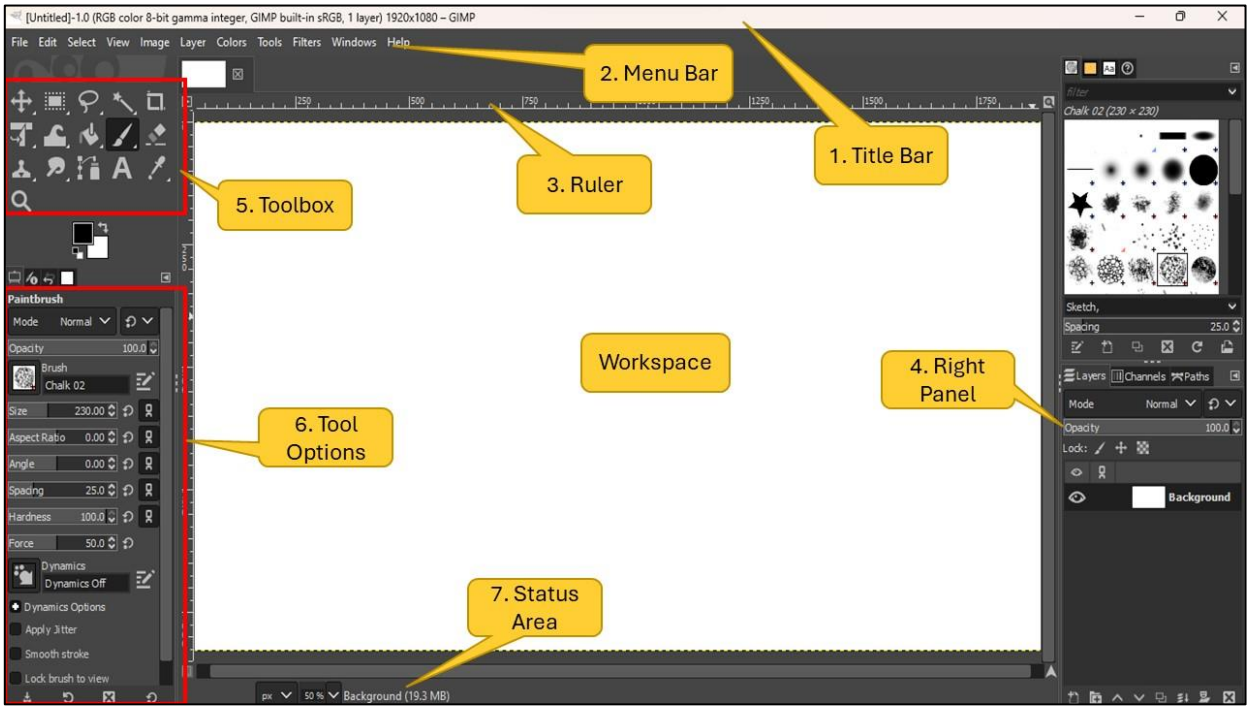


To start GIMP, follow the given steps:

- Click on **Show Application icon**& type **GIMP** in the search bar.
- The **GIMP Image Manipulator** application icon appears. Click on the icon to open it.
- The GIMP 2.10.36 welcome screen appears followed by the **GNU Image Manipulation Program** window.

Components of GIMP:

The main components of GIMP are:



1. **Title Bar:** The topmost horizontal bar that displays the name of the program.
2. **Menu Bar/Image Menu:** The Menu bar contains menus that stores the various options to manipulate the images. There are 11 menus: File, Edit, Select, View, Image, Layer, Colors, Tools, Filters, Windows and Help.
3. **Ruler:** They are used to determine the co-ordinates within the image & default unit of rulers is pixel. Rulers are shown above and to the left of an image.
4. **Right Panel:** Contains Brushes, Patterns, Fonts, and Document History dialogs together in a multi-tab dock.
5. **Toolbox:** The toolbox in GIMP is the main editing tool and contains the most used controls and menus. By default, it's situated in the upper left corner of the GIMP window.
6. **Tool Options:** Shows options for the currently selected tool.
7. **Status Area:** It is placed at the bottom of the GIMP window. Displays the original name of the picture & system memory consumed by the image.

GIMP Docks:

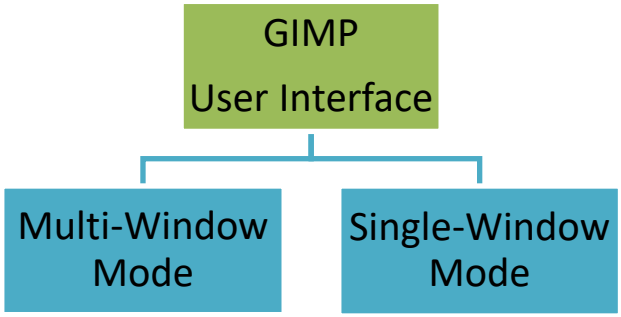
GIMP comes with three default docks:

- the Brushes, Patterns, Fonts, Gradients and Document History dock in the upper part of the right panel;
- the lower portion of the right panel is where the Layers, Channels, and Paths dock.

- the Tool Options, Device Status, Undo History, and Images dock under the Toolbox in the left panel.

A dock is a container which can hold a collection of dialogs together at a specific location. To activate any dialog, just click on its tab located on the dock.

GIMP Window Modes:
There are two options for the GIMP user interface:

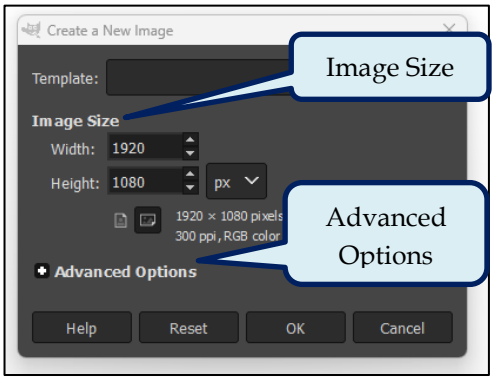


Multi-Window Mode: When you open GIMP for the first time, it opens in the multi-window by default. In the multi-window mode, the left and right panels are located separately on the screen and an image window is present in the middle. You can move these panels anywhere on the screen.
Single-Window Mode: In this mode, the left and right tools panels are fixed, and you cannot move them. But you can increase or decrease their width.

You can enable single-window mode through Windows > Single-Window Mode in the image menu bar.

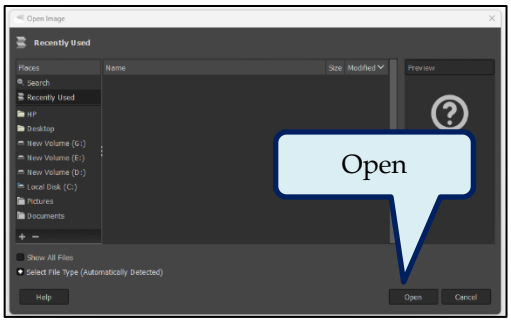
Creating A New File:
To open a new file steps are:

- Click the **File Menu > Select "New Option"** or press **Ctrl + N**.
- A dialog box will appear.
- Specify the template, width and height under the image size section. Properties of the image like resolution, color space, precision etc. in Advanced options.
- Click on **OK** button.



Resolution is the number of pixels that display per inch or centimeter for an image, or pixels per inch (PPI).

Opening An Image File:



To open an image file:

- Click on **File menu > Select Open**option or Press **Ctrl + O**.
- A dialog box will appear.
- Browse the file and click on its name.
- Click on the **Open** button.

Saving A File:

To save a file:

- Select the **File Menu > Save**Option
- The **Save Image**dialog box appears.
- Type the name of the file in the **Name** box.
- Select the location of your file from the **Save in folder**option.
- Click on **Save** button.

The file extension of an image file in GIMP is **.xcf**. It stands for Experimental Computing Facility.

3.3 GIMP Toolbox

The **GIMP** provides tools that will crop, zoom, erase, smudge, draw, measure, blur, and more.

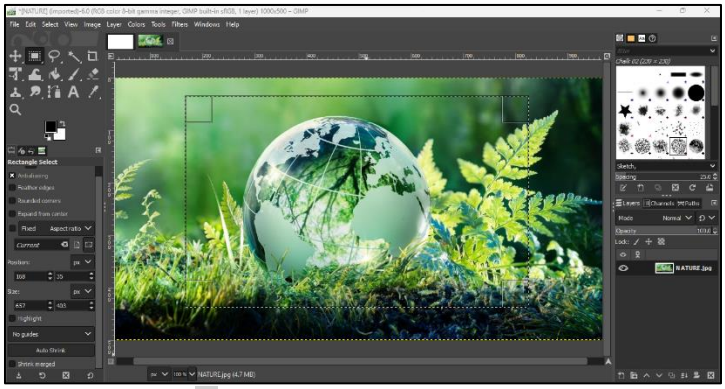
Selection Tools:

To work on selected portions of the active layer without affecting unselected areas, selection tools are used.

Seven tools are available for selection:

- the Rectangle Select;
- the Ellipse Select;
- the Free Select (the Lasso);
- the Select Contiguous Regions (the Magic Wand) ;
- the Select by Color;
- the Select Shapes from Image (Intelligent Scissors) and
- the Foreground Select.

Rectangle Select Tool:



The Rectangle Selection tool is used to select rectangular regions of the active layer.

Steps to use Rectangle Tool are:

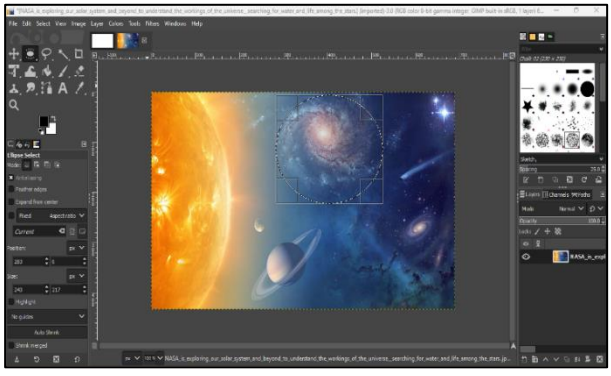
- To select the tool from the image menu bar: Go to **Tools > Selection Tools > Rectangular Select Tool**or click on the tool icon in the Toolbox or press the keyboard shortcut **R**
- After selection the mouse pointer is displayed like this: as soon as it is over the image.
- Drag the mouse to select a rectangular (or square) shape.
- When the mouse button is released, a dotted line ("marching ants") outlines the selection.
- The selection has four rectangular corners that are known as resizing handles. You can adjust the selected image's size with these handles.
- Click inside the selection or press the Enter key to exit the editing mode after making changes to it.

While editing an image if you want to revert one action Undo option can be used. If you want to undo multiple changes at once, you can use the Undo History options by using Edit>Undo History.

Ellipse Select Tool:

The purpose of the Ellipse Selection tool is to select elliptical and circular areas within a picture.

Steps to Use the Ellipse Selection Tool:



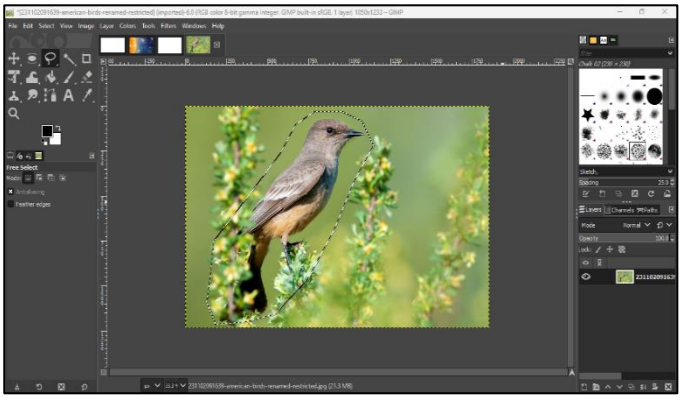
- Choose **Tools > Selection Tools > Ellipse**from the picture menu bar; or Click the tool icon in the ToolBox or Press the keyboard shortcut **E.**
- As soon as the mouse pointer crosses a picture after selection, it displays as a circular icon.
- An elliptical selection is outlined by a dotted line, also known as "marching ants," when the mouse button is released.

To deselect the selection, click anywhere outside the dotted line.

Free Select (The Lasso):
You can draw a free hand drawing selection using the cursor while using the Free Selection tool, also known as Lasso.

Steps to use Lasso Tool are:

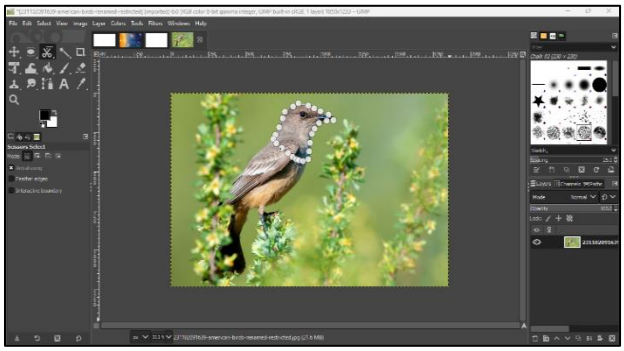
- To select from the image menu bar go to **Tools > Selection Tools > Free Select**, or click on the tool icon in the ToolBox, or press the keyboard shortcut **F**.
- To begin, click on the image for the starting point.
- To make a freehand selection, keep the mouse button pressed (click and drag).
- Closing the selection requires releasing the mouse button when the end point is above the starting point.



- A moving cross appears along with the end point when you click on it. The selection can be made bigger by dragging.

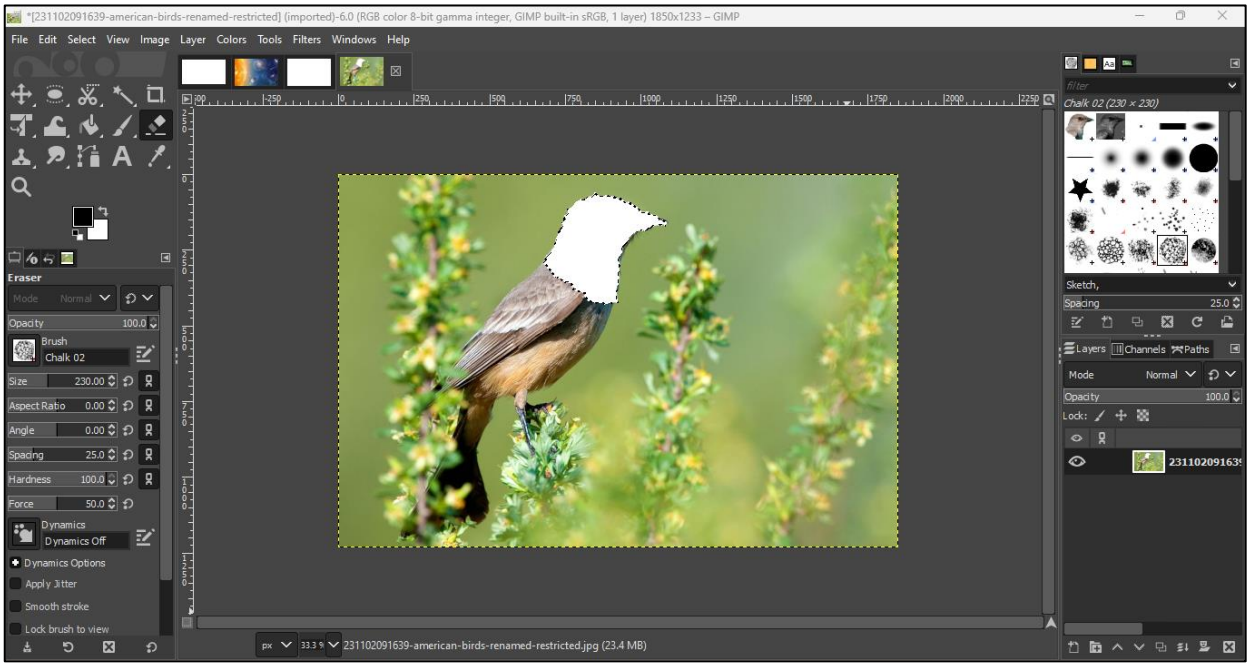
Scissors Select Tool:
The Scissors Select Tool is used to select shapes using intelligent edge fitting.
Steps to use the Intelligent Scissors Tool are:

- To use Intelligent Scissors, move to **Tools > Selection Tools > IntelligentScissor** from the image menu bar or by clicking on the tool icon in the ToolBox, or by using the keyboard shortcut **I**.
- Upon each left-click, a new control point is generated and linked to the previous control point by a curve that aims to mark the edges present in the image.
- Select the first point to complete the task.
- When the selection is closed the pointer shape changes according to its position: inside , on the boundary , and outside .



- Pressing the Enter key will enable you to exit the editing mode.
- To use the selected part in another image, go to **Edit>Cut** (or Press **Ctrl + X** key combination) option.

Note: Increase the number of control nodes to obtain a more accurate selection.



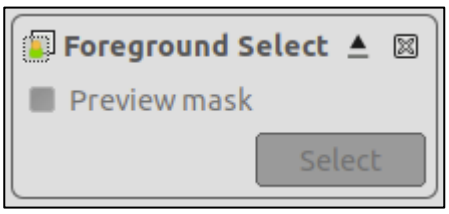
Foreground Select Tool:

An object's backdrop will sometimes have to be removed. Perhaps you would want the item to be on a clear backdrop or one with a simple color scheme.

With this tool, you can extract the foreground from a selection or the active layer.

Steps to use the Foreground Select tool:

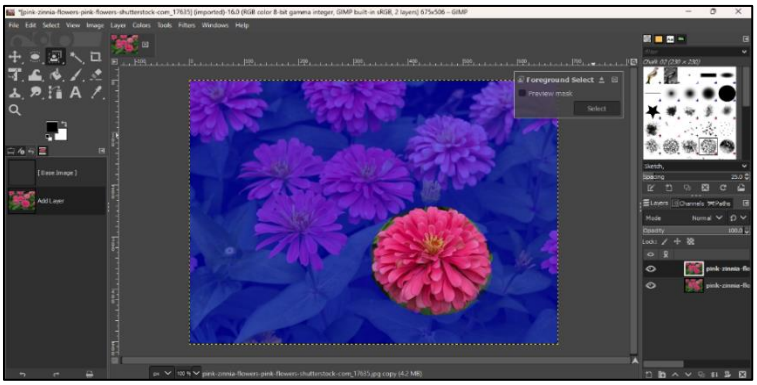
- Select the tool icon in the Toolbox or using the picture menu's **Tools >Selection Tools > Foreground Select** option.
- A little popup appears as soon as you click on start drawing the selection:



- The only button that is active before the selection is complete is the close button **x**, which lets you cancel and go back to the original image (you may also do this by using the **Esc** key).
- The first selection must be closed for it to be finished. The mouse pointer becomes yellow as it approaches the dot where you first made the selection. If

you want to end the selection, click this. Alternatively, you can double-click to end the selection.

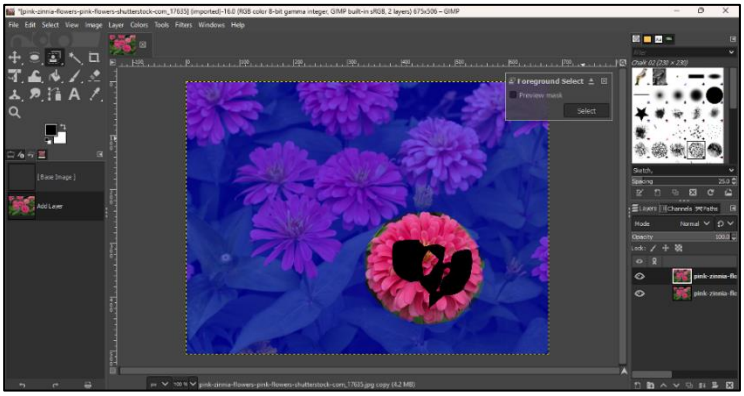
- To create the mask from your selection, press **Enter**, or double-click inside the selection:
- A circle-shaped Paintbrush symbol now appears when the mouse pointer moves. The backdrop may be seen in the dark blue (or any other color) section. Your chosen foreground area and a small portion of the backdrop are covered by the lighter region. The dark blue region is known as the "Unknown pixels area" since it is outside the selected area.
- When the image is seen in a small window, the following choices will become available:

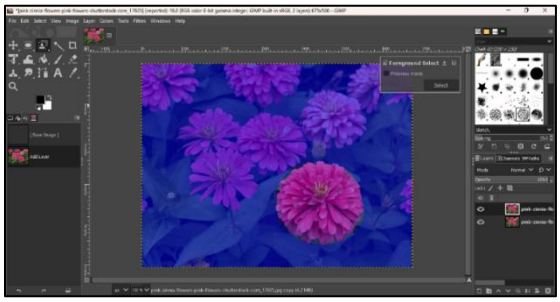


  - A Preview mask checkbox that allows you to choose whether to get a preview of the extraction progress in the foreground or not. You can do the same thing by pressing the **Enter** key.
  - After selecting the foreground, a Select button will be used to create the extraction.
- At this point, we begin the process of extracting the foreground: draw a line through it. The Tool Options section allows you to change the brush size. Moving across the

colors that will be retained for the extraction, draw a continuous line across the chosen foreground region.

- It only takes a few strokes to create this one-color, distinguishing item from the background:

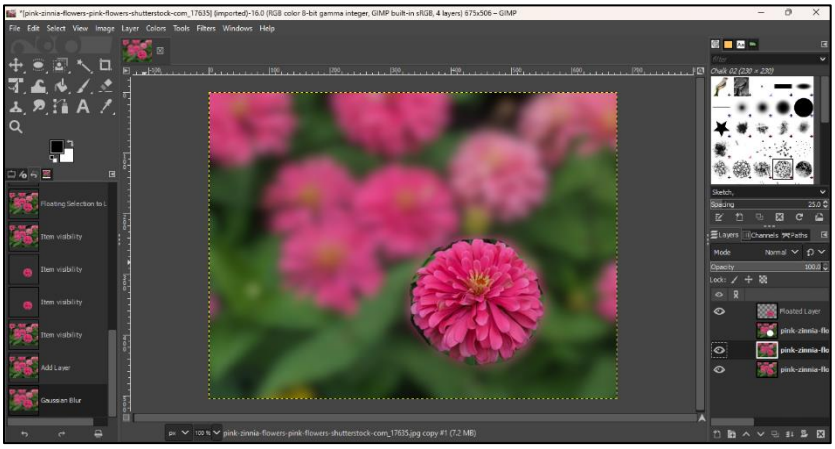




- Toggle the preview *button, or* press **Enter** to verify the result.

Note: You can draw directly on this **Preview mask** and see the result immediately.

- You can cancel it by pressing **Esc**.



Copying The Selection Using the Move Tool:

The Move Tool is used to move or copy the selected part of an image from one location to another.

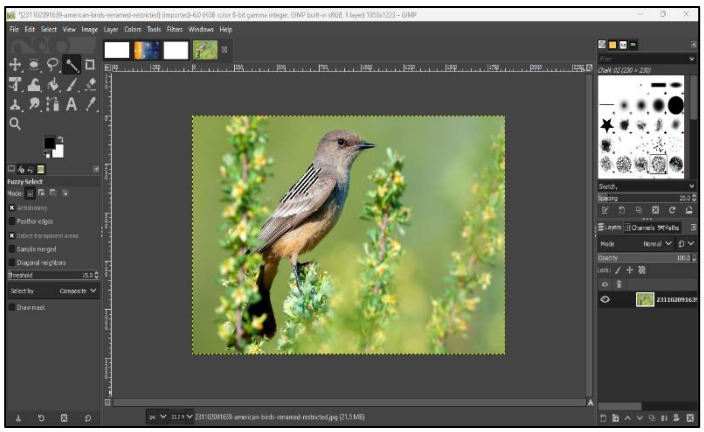
Steps to use the tool are:

- Select the object that you want to copy.
- Select the **Edit> Copy**and **Edit >Paste As> New Layer**option to paste the object.
- The new layer will be pasted under the Layers tab with the name Clipboard.
- Click on the Move Tool on the ToolBox.

- Place the pointer on the image and drag it to another location. As you release the mouse, the copy of the image will be placed to the new location.

Fuzzy Select Tool (Magic Wand):
Based on color similarity, the Fuzzy pick (Magic Wand) tool is used to select areas of the active layer or picture.

Steps to use the Magic Wand Tool are:



- Go to **Tools > Selection Tools > Fuzzy** Select from the picture menu bar, or by clicking on the tool icon in the ToolBox, or by using the keyboard shortcut **U**.
- The mouse cursor takes on the form of a magic wand.
- Click anywhere on the picture.
- The adjacent areas of identically colored pixels are selected.

Select by Color Tool:

It selects areas of an image based on color similarity.
To use the Select by Color Tool, follow these steps:

• To select Go To **Tools > Selection Tools > By Color**Select from picture menu bar or You may access the tool by either clicking on its icon in the ToolBox or by pressing the **Shift + O**keyboard shortcut.
• Select an area of the picture with a click.
• A dotted line will surround the selected area.

Crop Tool:
The Crop tool is used for removing unwanted areas outside the selection.
Steps to crop an image are:

- Click the tool icon in the Toolbox or press **Shift+C** or select **Tools > Transform Tools > Crop**from the picture menu bar;
- Click and drag to create a rectangle as required.
- Double-click inside the rectangle or press **Enter** to finish cropping.

Unified Transform Tool:

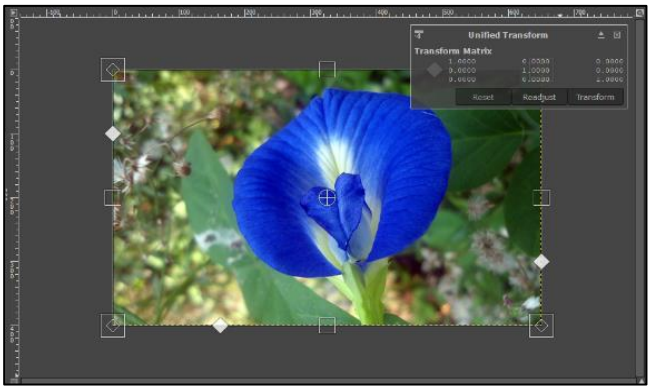
This tool performs one or more of the following operations simultaneously in a single operation: rotate, scale, shear, and perspective.

There are many ways to make the tool active:

- From the image menu: **Tools > Transform Tools > Unified** or use the **Shift+L** keyboard shortcut or can access the Tool by clicking on its icon

Several kinds of **handles** appear on the edges:

-  Diamonds for shearing
-  Squares for scaling.
-  Small diamonds for changing perspective, in large squares for Scaling.



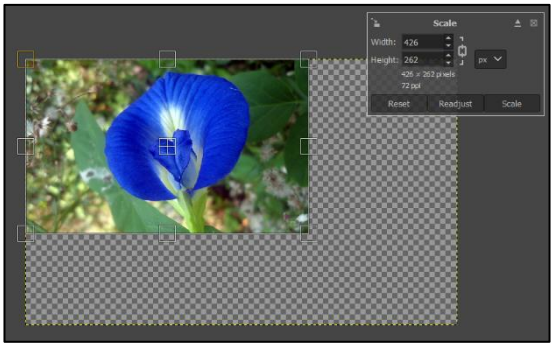
Let's discuss some important tools from it:

Scale Image:

The Scale Image command is used to resize the image.

Steps to scale the image are:

- Go **to Tools > Transform Tools >Scale**or use shortcut key combination -**Shift+S**
- Select the image.
- A Scale dialog box will appear.
- Click on Scale button after Specifying the width and height of the image in the dialog box.



Flip Tool:

This tool is used to flip layers or selections horizontally or vertically. Reflections and mirror images can be produced using it.

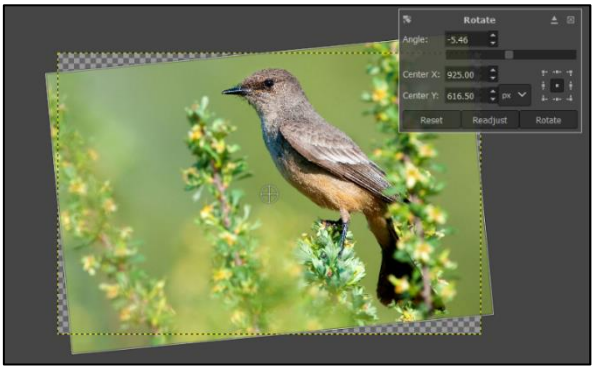
Steps to flip the image:

1. Use **Tools → Transform Tools → Flip**, or use the  button on the toolbox.
2. Click inside the canvas.
3. Use the controls in the Tool Options dockable to switch between Horizontal and Vertical modes.

Rotate Tool:

The Rotate tool in GIMP can be used to rotate an active layer, a selection, or a path. The rotation angle and axis can be changed.

Steps to rotate an image:



- For selection of Rotate tool, **Go to Tools > Transform Tools > Rotate** in the image menu bar , or by clicking the tool icon:  in the Toolbox, or by using the **Shift+R** key combination.
- Click on the image or selection with the Rotate tool.
- Set the rotation axis, rotation angle, and position of the rotation center in the Rotation adjustment dialog box,
- Click on Rotate button.

By pressing Ctrl key rotation angle will get increased by 15 degrees.

Warp Transform Tool:

The Warp Transform tool in GIMP is a brush-like tool that can warp an image. It can also partially remove warping and adjust its size and strength.

There are different possibilities to activate the tool:

- from the image-menu: **Tools > Transform > Warp Transform,**
- by clicking the tool icon in toolbox:  ,
- or by clicking on the **W** keyboard shortcut.

Bucket Fill Tool:

The Bucket Fill tool in GIMP fills a selection with the current foreground color. It can also fill with the background color by **Shift+ clicking**and using the Bucket tool.

The tool can be used as:

- You can select the Bucket Fill tool from the image-menu through: **Tools > Paint Tools > Bucket Fill**or by clicking the tool icon: in the toolbox or by pressing the **Shift** + **B** keys.
- Choose the desired colour or pattern & fill.

By default, foreground colour is black and background colour is white.

Paint Brush Tool:

The paintbrush tool paints fuzzy brush strokes.

Steps to use Paint Brush Tool are:

- To select the Paintbrush Tool, from the image-menu: Go to **Tools > Paint Tools > Paintbrush**or by clicking the tool icon: or by using the **P** keyboard shortcut.
- Select the required color,
- Press the left mouse button and drag the paintbrush icon over the canvas to create brush strokes.

Ink Tool:

The Ink tool uses a simulation of an ink pen with a controllable nib to paint solid brush strokes.

Steps to use the Ink tool are:

- To select the tool Go to **Tools > Paint Tools >Ink** in the image-menu or click on the tool icon: in Toolbox, or by using the **K** keyboard shortcut.
- Click and hold the left button of the mouse.
- Adjust the nib's size, shape, and angle of the stroke.
- Select the required color from the Foreground Color box.
- Click on **OK** button.

Eraser Tool:

The eraser tool can be used to remove areas of an active layer or a selected portion of a layer.

Steps to use Eraser Tool:

- You can activate the tool from the image menu through **Tools > Paint Tools > Eraser;**or from the Toolbox by clicking on the tool icon ; or from the keyboard using the shortcut **Shift** + **E**.
- Drag the mouse over the part of the image that you want to erase.

Clone Tool:

The Clone tool in GIMP uses the current brush to copy from an image or pattern.

Steps to use clone tool are:

- You can access the tool from the image menu through **Tools > Paint Tools > Clone**. or by clicking on the tool icon in Toolbox or by pressing the **c** keyboard shortcut.
- Set the brush style and size from the Tools Option pane.
- To define the source point press **CTRL** key and click on the image.
- Release the **CTRL** key and drag the mouse pointer to the location where we need to draw the clone.

Healing Tool:

The healing tool in GIMP is used for photo editing, such as: Spot removal, Photo refixing, Photo repair, Wrinkles removal, Removing blemishes, spots, and more.

Steps to use healing tool are:

- Select the tool from the image-menu: **Tools > Paint tools > Heal**, or by clicking the tool icon: in the Toolbox, or by clicking on the **H** keyboard shortcut.
- Set the brush shape and size from the Tools Option pane.
- Take the pointer to the normal area which is free from imperfections. Press the **CTRL** key to define the source point.

- Release the **CTRL** key and drag the mouse pointer to the defective area of the image to fix the imperfections.

Smudge Tool:

The Smudge tool in GIMP is like finger painting. It allows you to drag color around and blend it with the surrounding areas.

Steps to use the Smudge tool are:

- Select it through **Tools > Paint Tools > Smudge** in the image menu, or by clicking on the tool icon: in Toolbox, or by pressing the **S** key on keyboard.
- Select the required brush type & size.
- Drag the tool over the area of the image you want to smudge.

Dodge/Burn Tool:

The Dodge/Burn tool in GIMP is a paint tool that lightens or darkens colors in an image.

Steps to use the Dodge/ Burn tool are:

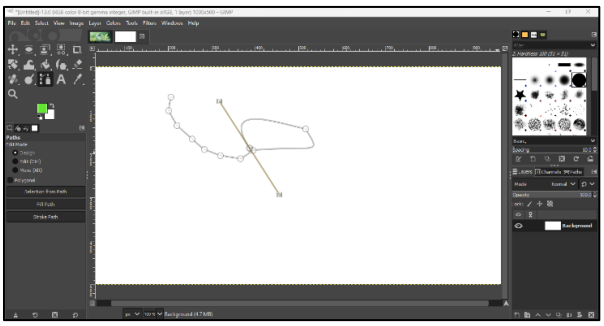
- To activate the tool from the image-menu: Go to **Tools > Paint Tools > Dodge/ Burn**, or by clicking the tool icon: , or by using the **Shift+D** keyboard shortcut.
- Select the brush style and size from the Tools Options pane.
- Move over the area that you want to brighten by holding down the left mouse button.

Note: **Ctrl** Toggle between dodge or burn types. The type will remain switched until **Ctrl** is released.

Path Tools:

The Path tool in GIMP is a tool that allows users to create complex shapes, such as Bézier curves, geometrical figures, and different polygonal shapes.

Steps to use the Path tool:



- To use it, Go to **Tools > Paths** from the menu, or select the icon in toolbox, or press the hotkey **B**.
- When the Paths tool is selected, the mouse cursor changes into a crosshair with a curve by default.
- Left click in the image to create the first point of the path. Move the mouse to a new point and left click to create another point linked to the previous point.

Text Tool:

GIMP's Text tool adds text to an image in a new layer. You can edit text directly on the canvas.

Steps to use Text tool are:

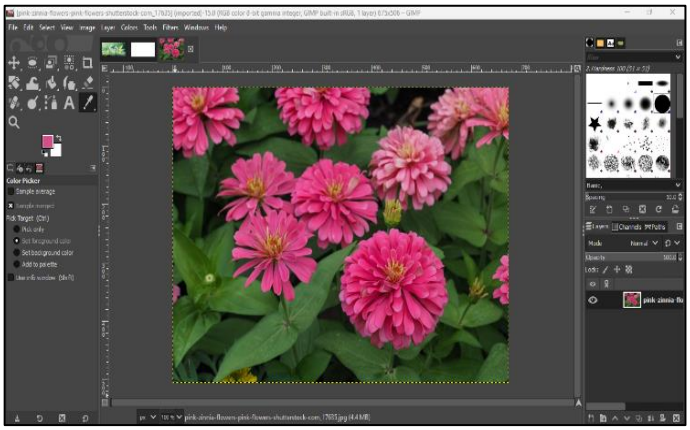
- Can select from the image menu through **Tools > Text**, or by clicking the tool icon in Toolbox, or by using the **T** keyboard shortcut.
- Click on the canvas where you want to add the text.
- Choose the required font style, size, colour etc. using the tools from the Tools Option or Font Bar.

Color Picker Tool:

GIMP's Color Picker Tool is used to select a color on an image.

Steps to use Color Picker tool are:

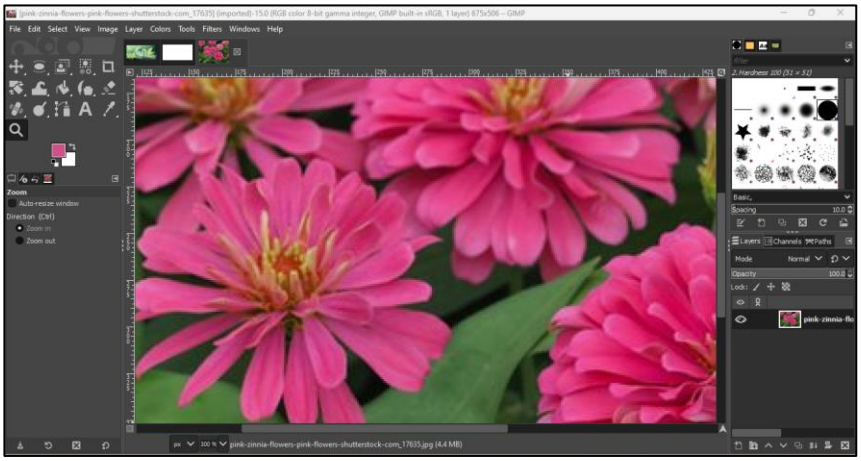
- In the image menu Go to **Tools > Color Picker**., or by clicking the tool icon in Toolbox, or by pressing the **O** keyboard shortcut,
- Click on any part of the image to choose the colour.
- Selected colour will be displayed in the foreground colour box.



By pressing the Shift key along with the Color Picker Tool, the color picker information window gets opened.

Zoom Tool:

The zoom tool in GIMP magnifies a selected part of an image.



Steps to use Zoom tool are:

- You can get to the Zoom Tool from the image-menu through: **Tools > Zoom**, or by clicking the tool icon: in Toolbox.
- Click on the image to increase the size of the image.

- To reduce the size of the image, press the CTRL key and click on the image. The mouse pointer will change into a (-) sign along with the magnifying glass.

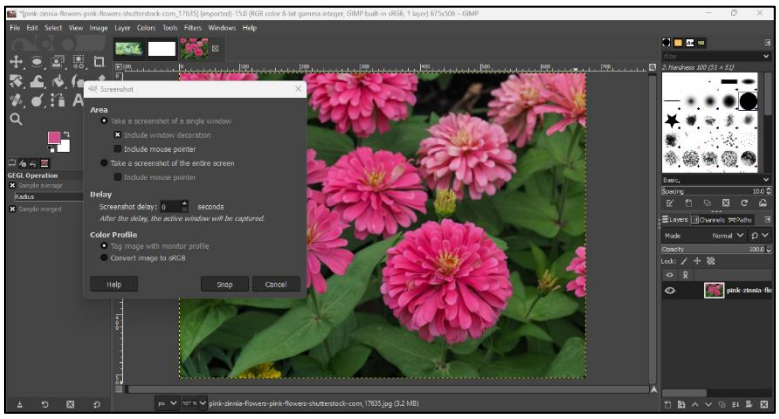
Some Additional Tools:

Screenshot:

The Print Screen keyboard key captures the screen and puts it in the clipboard.

Steps to take the screenshot:

- Click on **File> Create> Screenshot.**
- The Screenshot dialog box will appear.
- Choose the required options for Area, Delay & Color Profile
- Click on Snap.



Color Tool:

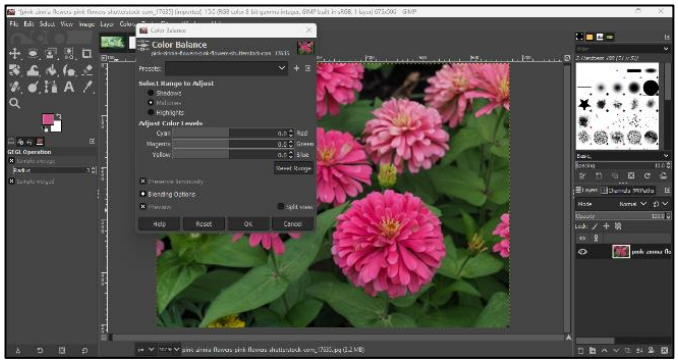
The Colors menu in GIMP contains commands that affect the color of an image. The menu includes:

Color balance, Color temperature, Hue-Croma, Hue-Saturation, Saturation, Exposure, Shadows-Highlights, Brightness-contrast.

Let's discuss some:

Color Balance:

GIMP's color balance tool can modify the color balance of a layer or selection. It can help correct colors in digital photos.



Steps to use Color Balance are:

- Go to **Colors > Select Color Balance**
- Color Balance dialog box will appear.
- Adjust the color levels.
- Click on **OK.**

Brightness And Contrast:

The Brightness-Contrast tool adjusts the brightness and contrast levels for the active layer or selection.

Steps to use Brightness and Contrast are:

- Go to **Colors > Select Brightness & Contrast option**
- Dialog box will appear.
- Adjust the levels.
- Click on **OK**.

Invert:

Inverts all pixel colors and brightness values in the current layer. This makes dark areas bright and bright areas dark. To do so, Go to **Colors → Select Invert.**

3.4 Filters

A filter is a special kind of tool designed to take an input layer or image, apply a mathematical algorithm to it, and return the input layer or image in a modified format.

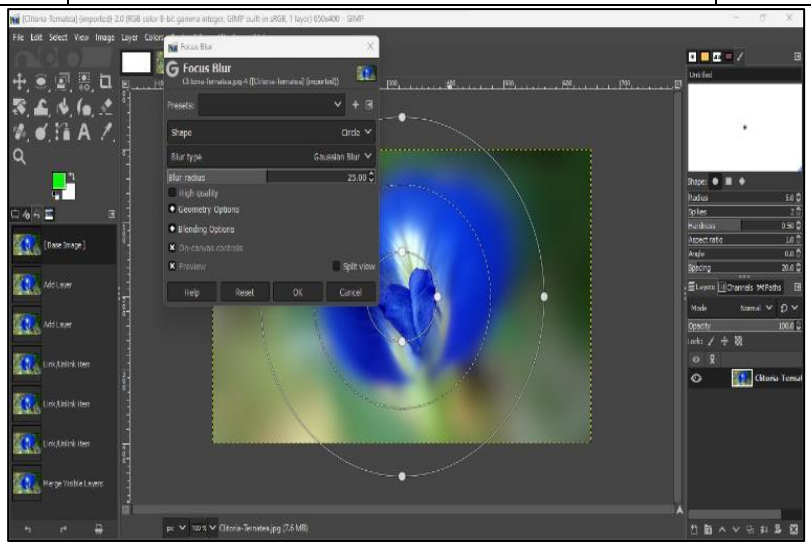
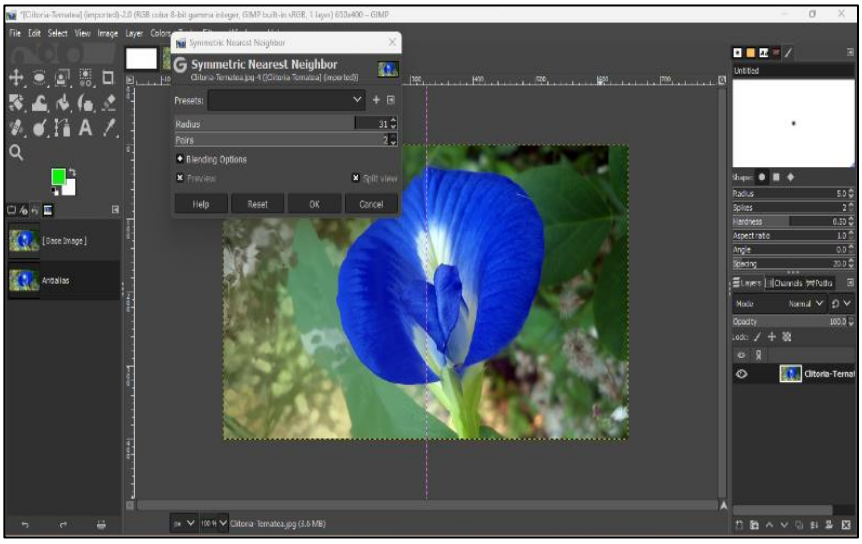
The filters are divided into several categories in GIMP:

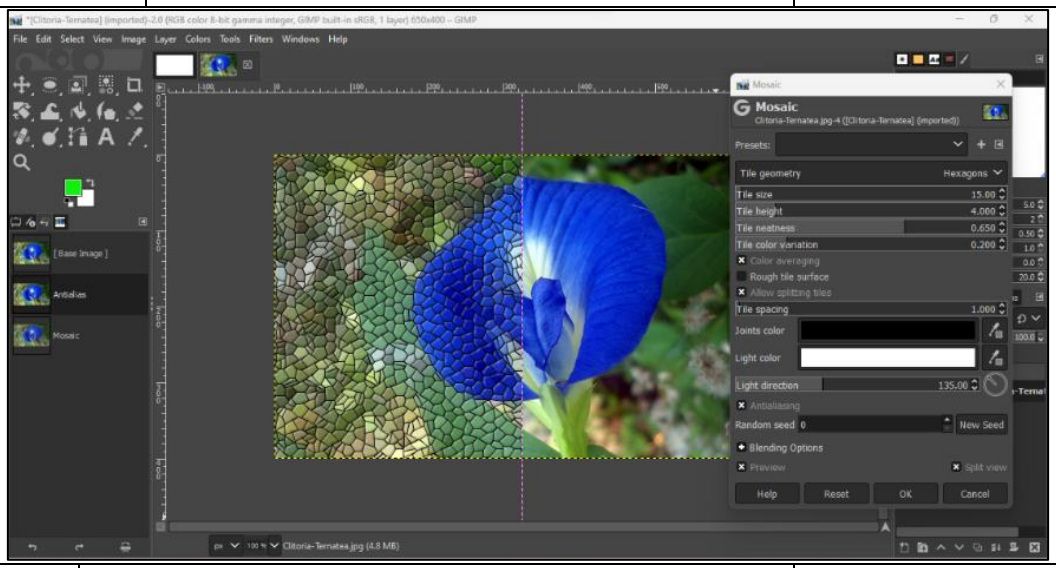
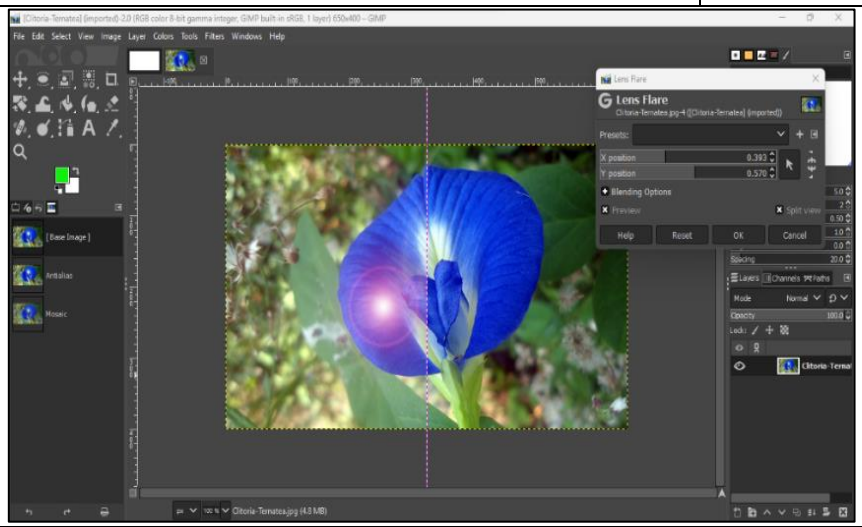
- Blur Filters
- Enhance Filters
- Distort Filters
- Light and Shadow Filters
- Noise Filters
- Edge-Detect Filters
- Generic Filters
- Combine Filters
- Artistic Filters
- Decor Filters
- Map Filters
- Rendering Filters
- Web Filters
- Animation Filters

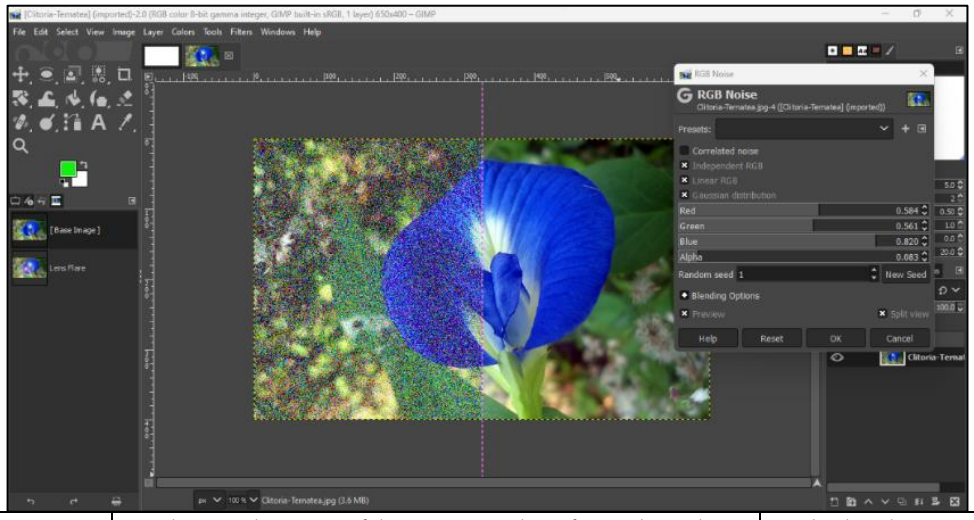
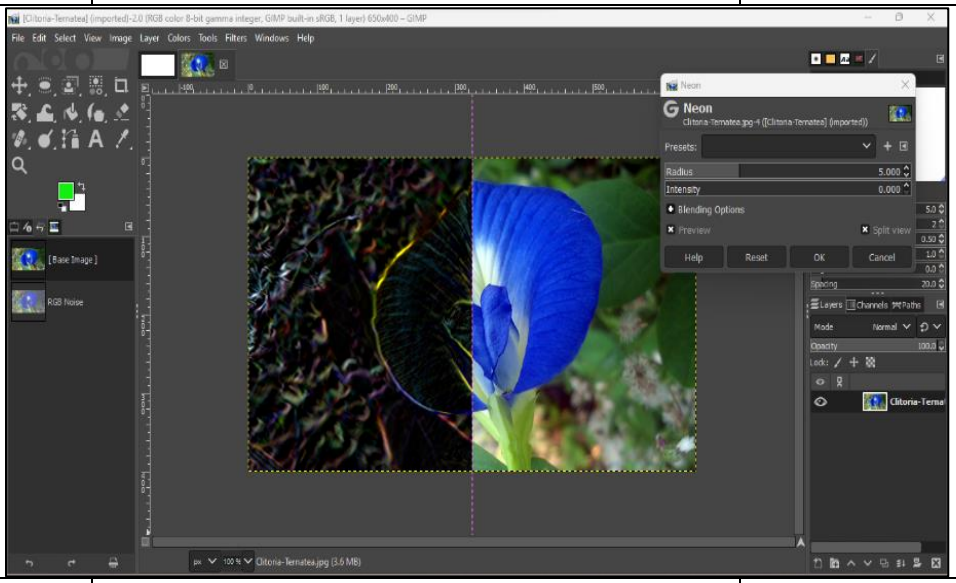
Steps to use Filters:

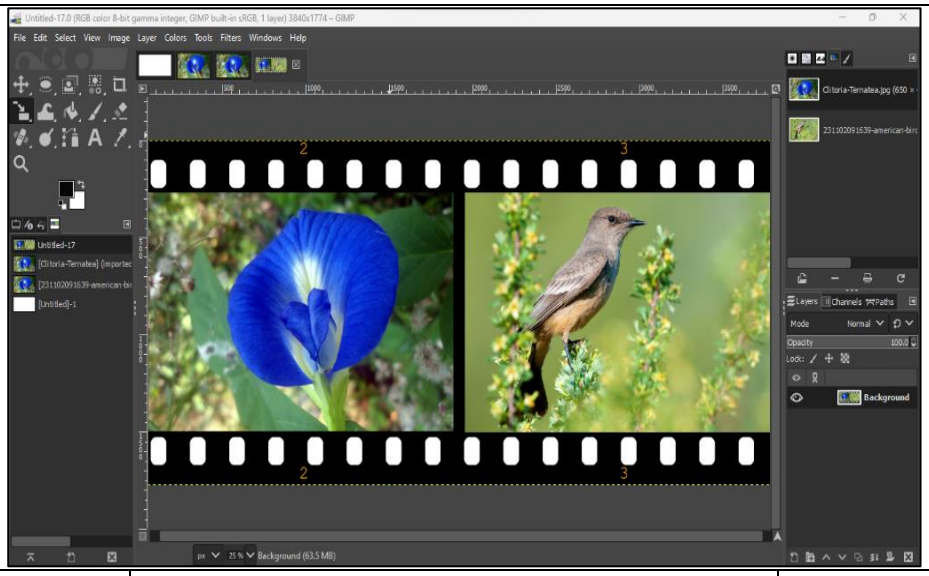
- Select the image.
- Go to filters in the image menu.
- Select the required filter.

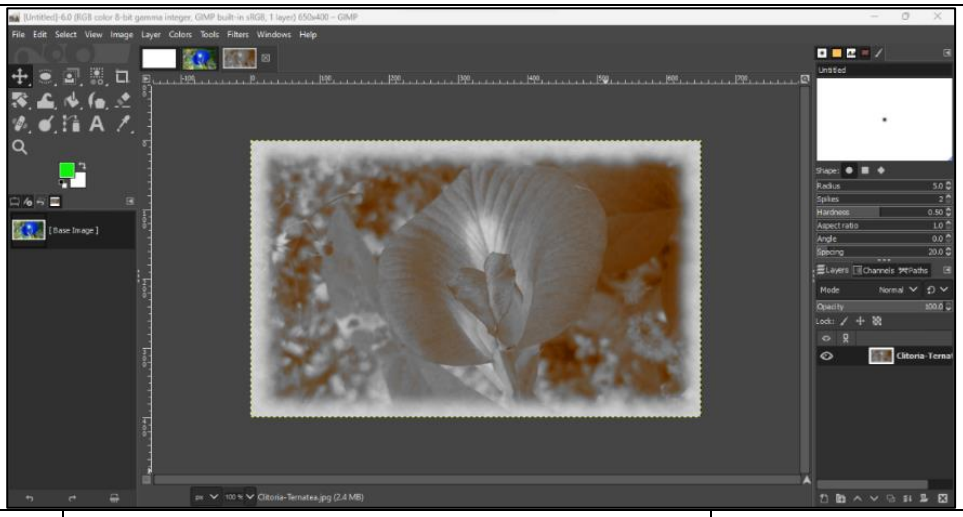
Some commonly used filters:

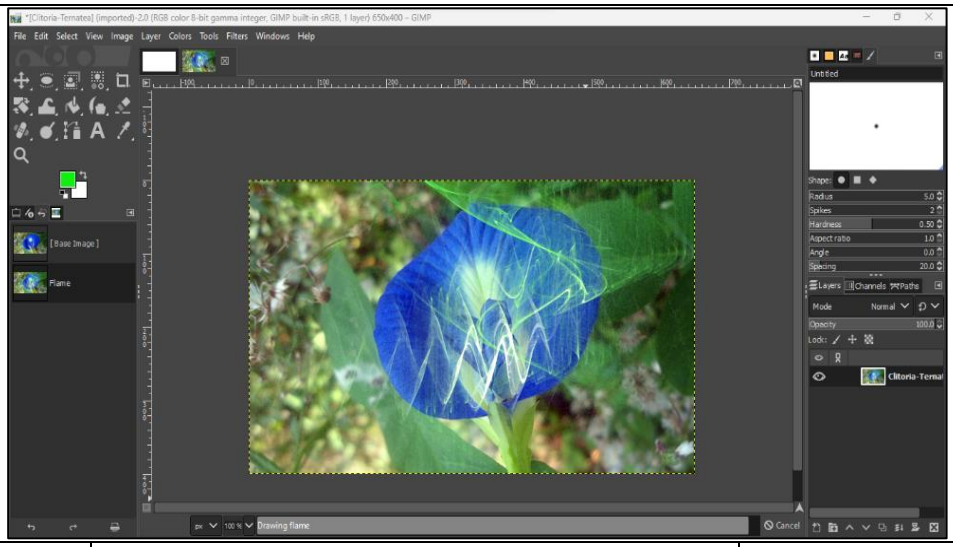
Filter	Description	Sub Filters
Blur Filters	This is a set of filters that blur images, or parts of them, in various ways. If there is a selection, only the selected parts of an image will be blurred.	Circular Motion Blur Tileable Blur Zoom Blur Lens Blur Selective Gaussian Blur
		
Enhance Filters	It is a plug-in that compensates for imperfections in an image, such as noise, dust, interlacing, and lack of sharpness.	High Pass Sharpen NL Filter Red Eye
		

Distort Filters	Distort filters are used to transform an image in a variety of ways. The distort filter sub-menu contains tools for transforming an image. These filters can warp, twist, and pull layers in different directions.	Whirl and Pinch Polar Coordinates Ripple Lens Distortion Waves Curve Bend
		
Light and Shadow Filters	The "Light and Shadow" menu consists of two groups of filters: **Light effects filters**, which render several illumination effects of the image. **Shadow effect filters**, which create various kinds of shadows.	Drop Shadow Lighting Effects Lens Flare Gradient Flare Long Shadow Drop Shadow (Legacy)
		

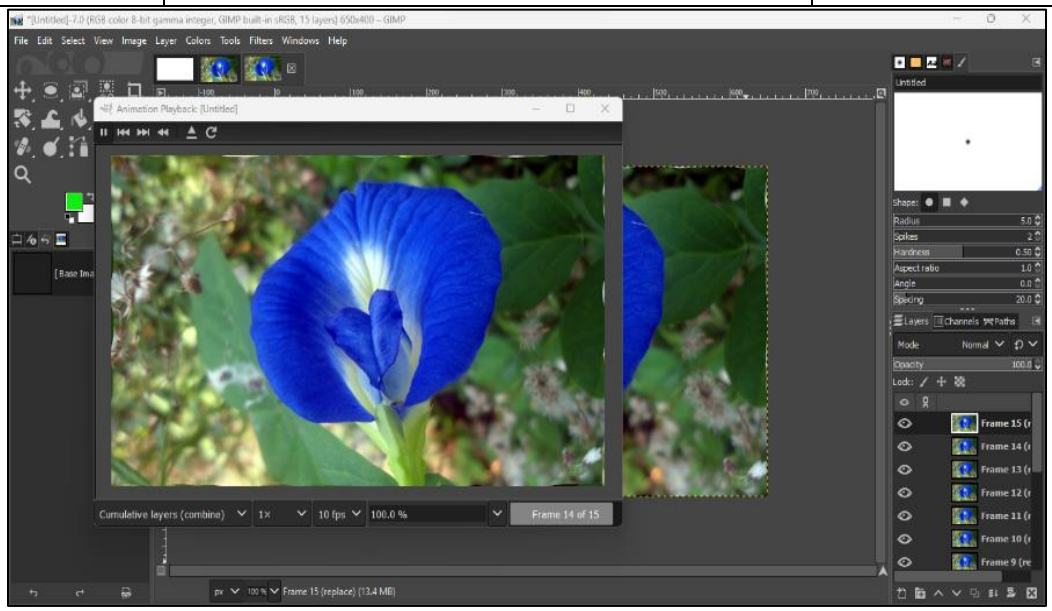
Noise Filters	Noise filters can be used to add texture to an image, or to remove noise from an image.	RGB Noise HSV Noise
		
Edge-Detect Filters	Edge detect filters search for borders between different colors and so can detect where significant changes in intensity or color occur.	Sobel Filter Laplacian Filter Edge Detection (Gradient)
		
Combine Filters	The Combine tool is used to combine the two or more images into one image.	Depth Merge Filmstrip

<table>
<tr><td colspan="3">  </td></tr>
<tr><td>Artistic Filters</td><td>These filters can be used to add artistic effects to an image, such as making it look like a painting or a sketch.</td><td>Cubism
Oilify
Apply Canvas
Watercolor
Cartoon</td></tr>
<tr><td colspan="3">  </td></tr>
<tr><td>Decor Filters</td><td>Decor filters are a type of filter that creates decorative borders and special effects.</td><td>Add Bevel
Add Border
Coffee Stain
Fog
Fuzzy Border
Old Photo</td></tr>
</table>

		
Map Filters	Map filters use an object named *map* to modify an image: you map the image to the object. So, you can create 3D effects by mapping your image to another previously embossed image ("Bumpmap" Filter) or to a sphere ("Map Object" filter). You can also map a part of the image elsewhere into the same image ("Illusion" and "Tile Seamless" filters), bend a text along a curve ("Displace" filter).	Bump map Fractal trace Illusion Displace Little planet Panorama projection Paper title
		
Render Filters	The render tool is used to create patterns from scratch. Most of the filters and effects work on a layer or a selection, but they create the effect from scratch.	Plasma Fractal Trace Maze Newsprint Line Nova



Animation Filters	Animation filters in GIMP are animation helpers that allow you to optimize and view your animations	Blend Burn-In Rippling Spinning Globe Waves



3.5 Working with Layers

In GIMP, layers are a stack of slides that contain parts of an image. They can be used to add and remove parts to an image without affecting the rest. Layers can also be used to experiment with different effects.

Creating A New Layer

Steps to create new layer:

- Go to **Layer Menu> Select New Layer**
- New Layer dialog box will appear
- Specify layer name, width & height of the layer
- Click on **OK** button.

To enable the layers tab press Ctrl + L

Renaming A Layer

Steps to rename a layer:

- Select the layer from Layers Tab & Right Click on it.
- Select Edit Layer Attributes
- A dialog box will appear.
- In the Layer name Text Box type the name
- Click on **OK** button.

Duplicating A Layer

It is used to create a copy of the layer.

Steps to duplicate a layer:

- Select the layer from Layers Tab & Right Click on it.
- Select Duplicate Layer Option
- or Press **Shift + Ctrl + D** combination to insert a duplicate layer.

A small thumbnail view of an image is displayed on Layer tab. A new layer is added above the previous layer. You can work on any layer by selecting it from the Layers tab in the right panel.

Making Layers Visible/ Invisible

You can make the layer Visible/ Invisible(hidden) by clicking on the Eye icon in the Layers Tab before the layer name. If the layer is hidden you cannot apply any tool on it.

Merging Layers

Merging means combining two or more layers into a single layer.

Steps to merge a layer:

- Select the layer from Layers Tab
- Press Shift Key to Select Second Layer
- Right Click on it
- Select Merge Visible Layers Option

Deleting A Layer

Steps to delete a layer:

- Select the layer from Layers Tab
- Go to **Layer Menu> Select delete option**
- or Click on Delete this Layer Button on Layers Tab

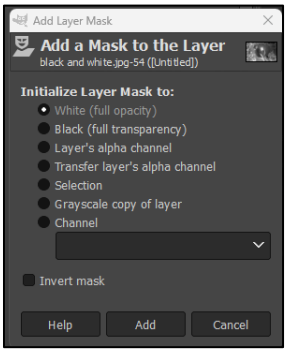
The selected layer will be deleted.

Layer Masking

Layer Masks are an essential tool in photo editing. They allow us to manage transparency and other components selectively. It is useful for removing the background of the image. The layer masks can be three types white, black, or Gray, representing full transparency, full opacity, or partial transparency. So, the layer mask makes the image transparent, opaque and partially transparent.

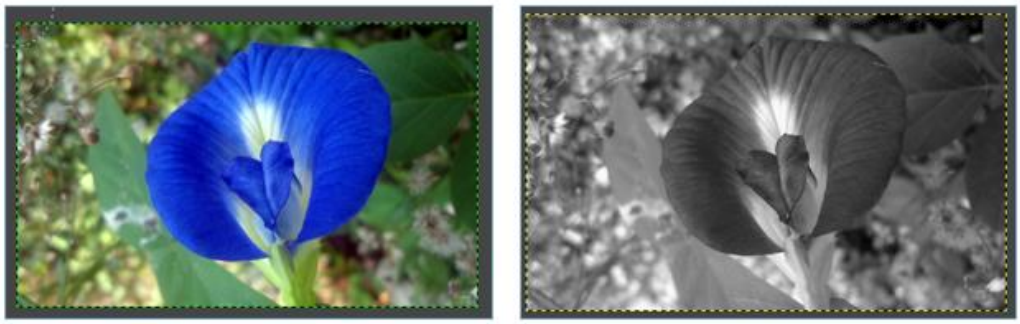
Steps for Layer Mask:

- Insert an image to the background layer.
- Add one more layer.
- Convert the same image in the grayscale mode. (Right Click on the image > select mode from the context menu> grayscale mode)



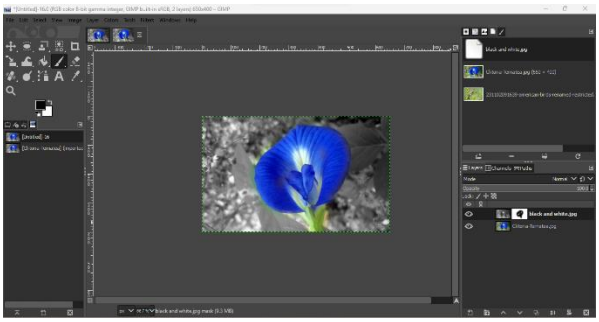
- Right-click on the layer and select the Add Layer Mask option
- Select the type of mask to add in the dialog window.

  - **White (full opacity)**
    With this option, the layer mask will make all of the layer fully opaque. That means that you will not notice any difference in the appearance of the layer until you paint on the layer mask.
  - **Black (full transparency)**
    With this option, the layer mask will make all of the layer fully transparent. This is represented in the image by a checkered pattern on which you will need to paint to make any part of the layer visible.



Layer 1 Layer 2

- Select the mask
  - Use Brush if you have selected **a White (full opacity) mask.**
  - Use Easer if you have selected **Black (full transparency)mask.**
- As you paint, the Layer Mask icon will change to reflect the brush strokes that you are applying.
- You should see the image changing visibly as transparent areas become opaque again.



Exercise

I. Multiple Choice Questions:

1. The shortcut key to use the Select by Color Tool is _____.
   a. Ctrl + O
   b. Ctrl + S
   c. Shift + O
   d. Shift + S
2. The _____ tool is used to select the areas of the image based on colour similarity.
   a. Fuzzy Select
   b. Select by Color
   c. Free Hand
   d. Lasso
3. You can invert the selection by using the _____ key combination.
   a. Shift + I
   b. Ctrl + I
   c. Ctrl + M
   d. Alt + I
4. The Shift key allows you to change the mode between horizontal and vertical flipping.
   a. True
   b. False
5. If you press the Ctrl key and use the Bucket Fill Tool, it will select the _____ colour.
   a. Foreground
   b. Background
   c. Base
   d. Palette
6. _____ filters can be used to add texture to an image, or to remove noise from an image.
   a. Blur
   b. Artistic
   c. Noise
   d. Web
7. Is the correct key combination to select the Flip Tool _____?
   a. Shift + F
   b. Ctrl + F
   c. Alt + F
   d. Shift + Ctrl + D
8. Which tool is used to enlarge or shrink the size of the image?
   a. Zoom b. Flip
   c. Clone d. Scale

9. Which tool uses simulation of an Ink pen with a controllable nib to paint?
    a. Clone
    b. Colour Picker
    c. Ink
    d. Color Balance
10. _____ is the correct combination to duplicate a layer.
    a. Shift + D
    b. Ctrl + D
    c. Shift + Ctrl +D
    d. Shift + Alt + D
11. By default, the foreground colour is white and background colour is black in GIMP.
    a. True
    b. False
12. The _____ effect lightens the colours and creates a soft tone in the selected image.
    a. Burn
    b. Scale
    c. Dodge
    d. Paint Brush
13. The _____ filters can be used to add artistic effects to an image, such as making it look like a painting or a sketch.
    a. Blur
    b. Animation
    c. Artistic
    d. Decor
14. Selection tools are used to select an image or any part of it.
    a. True
    b. False
15. _____ are the transparent sheets that can hold objects and are stacked on top of each other.
    a. Pages
    b. Slides
    c. Filters
    d. Layers
16. Lens Flare is a subtype of _____ filter.
    a. Decor
    b. Blur
    c. Animation
    d. Map
17. _____ is visible only when a new file is created.
    a. Brush
    b. Eraser

c. Ruler
d. Menu

18. If you want to do multiple changes at once you should use _____ option.
    a. Undo
    b. Undo History
    c. Revert
    d. Revert All
19. _____ filters are a type of filter that create decorative borders and special effects.
    a. Blur
    b. Map
    c. Décor
    d. Web
20. The file extension for an image file is .xcl.
    a. True
    b. False

II. Answer the Following Questions:

1. What is GIMP? Briefly describe the features of GIMP.
2. Explain the use of Scissors Select tool.
3. What is the purpose of the Crop tool?
4. Explain the use of the Fuzzy Select tool.
5. What is the purpose of inverting a selection?
6. Explain in brief the utility of layers.
7. How can you take screenshot in GIMP?
8. Differentiate between Blur filter and Décor filter.

III. Application-Based Questions:

1. Myra has designed a card for her mother on Women's Day. She wants to change the colour of the flowers from blue to pink. Suggest a tool with which she can perform the task?
2. Vasu has inserted an image in a file, and he wants to create a copy of that image on the same layer. Suggest to him a tool that will fulfill his requirements.
3. Tiya has inserted an image of birds. She wants to do free-hand drawing selection of the part of an image. Suggest the tool which will serve her purpose.

IV. Practical Exercise:

1. Create a poster on World Environment Day.
2. Design a birthday card for your friend using an animation filter.
3. Create a collage on Women empowerment.
4. Design an advertisement on Health and Fitness.

V. Group Discussion:

- How do you stay motivated with repetitive tasks like editing and improving images?
- Share tips and tricks for streamlining the editing process, including keyboard shortcuts, presets, and automation.
- What are the most important skills every photo editor should possess?
- Discuss the ethical implications of image editing, including manipulation, photojournalism ethics, and digital alterations.
- Share common challenges faced in image editing and brainstorm solutions.
- Speculate on emerging technologies and trends that may shape the future of image editing, such as AI-driven editing tools and virtual reality.

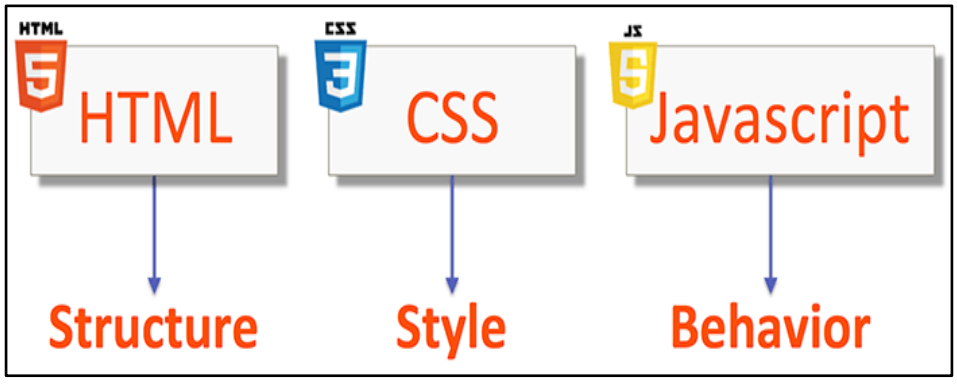
CHAPTER 4
JavaScript Part 1

Topics Covered

4.1 Introduction to JavaScript

JavaScript is a computer programming language used to make websites and applications dynamic and interactive. It is unique because it can run directly in your browser and also on a server. Along with Hypertext Markup Language (HTML) and Cascading Style Sheets (CSS), JavaScript is one of the most commonly used programming languages of the Internet.

HTML provides the basic layout, structure, and content of a website. CSS provides design, fonts, colors, effects, and other visual elements. JavaScript brings dynamism and interactivity to the website. For example, pop-ups, animations, video and social media embeds, drop-down menus, and many other website components are created using JavaScript.



4.1.1 History of JavaScript

JavaScript is developed in 1995 by Brendan Eich, a computer scientist and programmer at Netscape Communications Corporation. The initial name of the JavaScript was 'Mocha'. After that, it changed to 'LiveScript', and then 'JavaScript'. Over the years, JavaScript has evolved and become more powerful, with the addition of new features such as object-oriented programming, regular expressions, and support for asynchronous programming. Today, JavaScript is used in a wide range of applications, including web development, mobile app development, and server-side programming.
JavaScript language is interpreted and executed by the browser.

4.1.2 JavaScript as an Interpreted Language

An interpreted language is one that does not require compiling into machine language. It is executed by an interpreter that reads the source code and converts it into a form that is directly executed. The interpreter executes the code line by line. Any error that is found by the interpreter on any line of code, will stop the further execution of the program. JavaScript is directly executed by the browser which interprets the program instruction by instruction leading to slower execution of the program.

Static Website

A static website is a type of website that delivers the same content to all users. The content is stored on the web server and is shown on the user's browser as is, without any change of content. Static websites are simple to create and host, and are typically used for informational websites, such as brochure websites or portfolios. They don't require any server-side processing or database interactions, making them fast and easy to maintain.

Dynamic Website

A dynamic website is a type of website that generates content based on user interactions or other events. It generates custom content for each user based on their actions or preferences. Dynamic websites are typically created using server-side programming languages such as PHP, JavaScript, or Python, and they interact with a database to retrieve and store data. Examples of dynamic websites include social media platforms, e-commerce sites, and content management systems.

Static Website	Dynamic Website
Content of Web pages cannot be changed at runtime.	Content of Web pages can be changed at runtime.
No interaction with the database possible.	Interaction with database possible.

It is faster to load as compared to dynamic websites as it is more text based.	It is slower than a static website because it has images, audio & video content.
Cheaper Development costs.	More Development costs.
Blog Websites, Newsletter Contents, Brochure websites	Facebook, MakemyTrip, TimesofIndia

4.1.3 Features of JavaScript

1. **Easy to Learn:** JavaScript is easy to learn and the syntax of JavaScript is very simple. A programming beginner can choose JavaScript as their first programming language.
2. **Case Sensitive:** JavaScript is a highly case-sensitive programming language, which means that the identifiers, keywords, variables, and function names must be written correctly.
3. **OS Support:** It is supported by several operating systems including, Windows, macOS, etc.
4. **Control Statements:** It has control statements like if, if-else, if-else-if, switch case, and loop which helps users to write complex code using these control statements.
5. **Client and Server-Side Support:** Client-side or front-end web developers use their programming talents to create visually appealing websites for users. This means they build homepages, shopping pages, and slideshows. Server-side or back-end developers create, design, and manage server-side code responsible for data exchange. You can use JavaScript for both client-side and server-side scripting. This means it's helpful when designing website layouts and managing server-side code for data exchange.
6. **Object-Oriented:** It is an object-oriented programming language, which means that it uses objects to represent real-world entities and concepts.
7. **Browser Support:** All popular web browsers support JavaScript as they provide built-in execution environments.
8. **Functional Programming:** In JavaScript, a function can be assigned to variables just like any other data type, a function can accept another function as a parameter and can also return a function. This provides you with the ability to code in functional programming style.
9. **Dynamic:** JavaScript is a high-level and dynamic programming language, which means that the type of a variable is determined at runtime, rather than being specified in the code.

Advantages of JavaScript

1. It is relatively easy to learn and use.
2. It can be used for client-side and server-side i.e. front end and back end.
3. It provides dynamism and interactivity on websites.
4. It runs on multiple platforms and devices. It is supported by all browsers.
5. There are many libraries, frameworks, and APIs available to facilitate tasks.
6. It can create visually appealing web projects and create drag & drop components like sliders etc. to make the website more professional.

4.2 Prerequisites for Executing JavaScript Programs

The biggest advantage of JavaScript is that there is no need to purchase any tool to develop the JavaScript application. These tools are freely available and can be easily obtained.
A simple text editor like window notepad is required to create the JavaScript code. An online text editor can be used alternatively. A HTML editor is another tool which can edit the HTML source code and add the JavaScript code in the web page. An example of an HTML editor is Dreamweaver which has lots of features and drag and drop facilities to make the task easy. Microsoft Visual Web Developer Express is an advanced page editor which can color the important JavaScript words and validate the code and load the page into the web browser to view the changes or contents.

4.3 Introduction to Script Tag

JavaScript can be implemented using **<script>… </script>** tags. The <script> tag containing JavaScript can be placed anywhere within the web page, but it is normally recommended that it should be kept within the <head> tags. The <script> tag alerts the browser program to start interpreting all the text between these tags as script commands.

The syntax of a JavaScript segment in HyperText Markup Language (HTML) is as follows:

```
<script language="JavaScript" type="text/JavaScript">
        JavaScript code
</script>
```

Since JavaScript is the default scripting language in HTML, we can work within <script></script> tags.

The <SCRIPT> tag takes two important attributes:

1. Language- This attribute specifies the scripting language. Typically, its value will be JavaScript. The recent versions of HTML have phased out the use of this attribute and have become optional.
2. Type – This attribute is used to indicate the scripting language and its value should be set to "text/JavaScript". All current browsers are updated and JavaScript enabled by default. This attribute too is optional.

There are **three** different places in the HTML document where scripts can be used.

1. **Body of the page:** In this case when page is loaded in the browser then output is displayed as the part of the HTML document.
2. **Header of the page:** In this case code is written in the form of a function (groups of JavaScript statements but treated as a single unit and referred to in the other script in the same page).
3. **As external file:** In this case JavaScript code is written in another file having .js extension. This file is included in a script tag by specifying the file name.

Writing first JavaScript program

```
<script>
    document.write("<i>hello World!</i>");
// First JavaScript Program to print Hello World on the screen
</script>
```

Output on the browser : hello World!

1. Open any editor such as notepad and write the above program.
2. Save the program with the .html extension in a proper folder or subfolder on a drive like C:\JavaScript\myprograms.
3. Open the web browser like Google Chrome, I
4. Double Click the file you have created and saved in step 2, and then you can see the output of your program.

4.3.1 JavaScript Syntax and Rules

JavaScript syntax is the set of rules that define a structured JavaScript. Here are some tips to remember when writing your first JavaScript program.

1. **Case Sensitivity:** JavaScript is case sensitive i.e., uppercase letters and lower-case letters have different meanings. For example, the word "alert" has a lowercase "a". So, if we type the word with an uppercase "A", then the alert box will not be displayed and the JavaScript code won't get executed.
2. **Whitespace & Line Break:** You can use spaces, tabs, and newlines anywhere in the JavaScript Program. The JavaScript interpreter ignores them. Use tabs &

spaces to neatly format or indent your code. It makes the code easy to read & understand.

3. **Comments:** The JavaScript allows us to add single line comments or Multi line comments.

Single-line comments (//) − Any text between a // and the end of a line is treated as a comment.

```
<script>
        //This program shows single line comments
</script>
```

Multi-line comments (/* */) − These comments may span multiple lines.

```
<script>
        /*This is an example of
          multiline comment */
</script>
```

4.3.2 Common Errors

Since JavaScript is an interpreted language, there are some common mistakes that wont display the code correctly on your browser.

1. Missing quotation marks - document.write(“hello world)
2. Case sensitive - document -> cannot be written as Document
3. Missing parenthesis/brackets - document.write(“hello world”
4. Missing <script> tag in the program (either opening or closing)
5. Variable names misspelt.

Internal & External Java Script

Internal JavaScript: JavaScript can be added directly to the HTML file by writing the code inside the <script> tag . We can place the <script> tag either inside <head> or the <body> tag according to the need.

Example:

```
<html>
<head>
<title>Internal JavaScript</title>
</head>
<script>
   /*Internal JavaScript*/
    document.write("Hello Friends!How are you?");
</script>
</html>
```

Output: Hello Friends! How are you?

External JavaScript: A JavaScript program can be written in a file and saved with the .js extension. This file can be then linked inside the <HEAD> and </HEAD> tags of the HTML document in which we want to add this code. The SRC attribute of the <script> tag allows to give the path of the JavaScript file.

Example:
Open the text editor and type the following:

```
document. write("Good Morning Friends!")
```

Save the above file as tryExternal.js

Open another document in your text editor and type the following code:

```
<html>
<head>
<script src="tryExternal.js">
</script>
</head>
</html>
```

Save the above file as Show.html and open it in your browser to see the output.

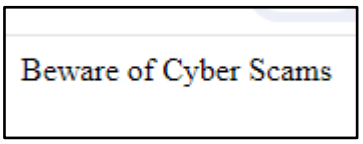
Output: Good Morning Friends!

4.4 Input and Output from the Script

1. **document.write method**
   JavaScript has access to the document property of the window object. The document property returns the document object of the window which is being used. When an HTML element is loaded onto the web browser then it becomes a part of the document object. The document object is the root node of the HTML page.
   One can access the document object either by using window.document or just document. *For example:* window.document.close() or document.close()
   The document object has a large range of properties or methods which a developer can make use of.The document.write() in JavaScript helps to write a JavaScript program or HTML expression into the document.

Example 1: We can write a string into a web page.

```
<html><head>
<script>
    document. write("Beware of Cyber Scams")
</script>
</head></html>
```



Example 2: We can also write HTML content and display it on a web page.

```
<html><head>
<script>
    document.write("<b>Beware of Cyber Scams<b>");
</script>
</head></html>
```

2. Dialog boxes

(i) **alert() dialog box:** An alert dialog box displays a short message or notification to the user. Alert box gives only one button "**OK**" to select and proceed. It just gives some information necessary to the user and it can be invoked at various events possible in JavaScript.

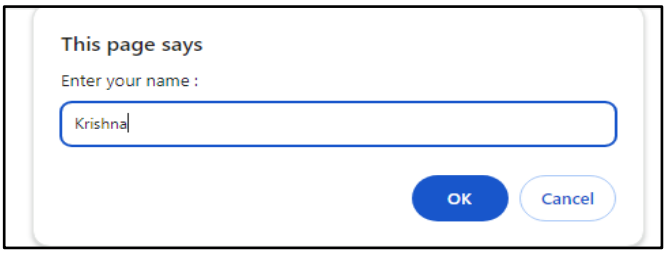
Example:

```
<html>
<head>
<title> JavaScript Alert Dialog Box
</title></head>
<body>
<script>
    alert("You have chosen an Alert box");
</script></body></html>
```

(ii) **prompt() dialog box:** The prompt dialog box is used to get some input from the user. It has two buttons, one for "OK" and the other for "Cancel". This method returns the text entered in the input field when the user clicks the OK button. It will return null if the user clicks the Cancel button. If the user clicks the OK button without entering any text, an empty string is returned. For this reason, its result is usually assigned to a variable when it is used. If you enter a number in the prompt box, it will be stored as a string, as it always returns a string.

Example:

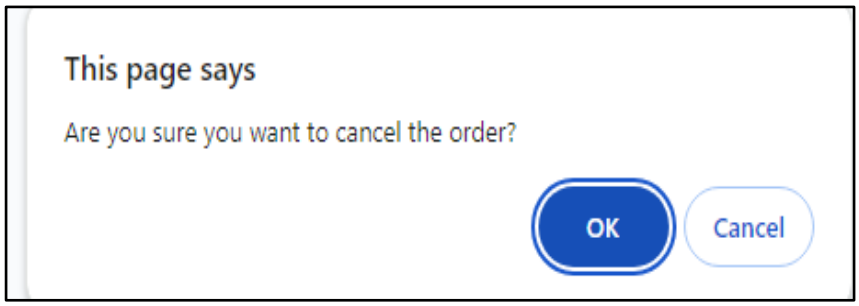
```
<html>
<head>
<script>
   var yourname = prompt("Enter your name : ");
   var age = prompt("Enter your age :");
   document.write("Your name is : " + yourname+"<br>");
   document.write("Your age is : " + age);
</script>
</head></html>
```



(iii) **confirm() dialog box:** A confirmation dialog box is mostly used to take the user's confirmation on any option. It displays a dialog box with two buttons: **OK** and **Cancel**. If the user clicks on the OK button, it will return true. If the user clicks on the Cancel button, it will return false. It returns a Boolean value depending on whether the user clicks OK(true) or CANCEL(false) button.

Example:

```
<html>
<head>
<title> JavaScript Confirm Dialog Box </title></head>
<body>
<script>
 var result = confirm("Are you sure you want to cancel the order?");
 if(result==true)
 {
   document.write("Order IS Canceled!");
 }
 else
 {
  document.write("Order NOT Canceled");
 }
</script></body></html>
```



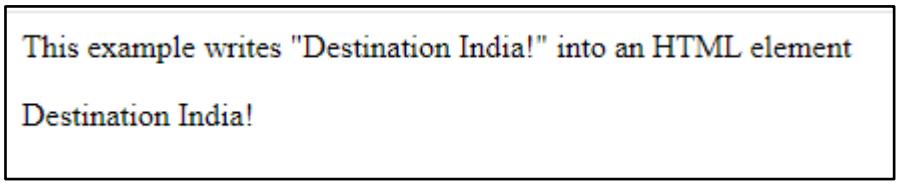
3. **Interacting with HTML**
   The HTML DOM (Document Object Model) provides a way to interact with and manipulate the elements of an HTML document using JavaScript. It allows you to access, modify, and add elements dynamically, change styles and classes, handle events, and perform other operations on the document. One of the most common DOM functions you'll use is the document.getElementById() function, which returns the element with the ID you pass in as a parameter.

Output a value from JavaScript into an HTML element, in this case in the paragraph.

Example 1:

```
<html>
<body>
<p>This example writes "Destination India!" into an HTML element</p>
<p id="demo"></p>
<script>
document.getElementById("demo").innerHTML = "Destination India!";
</script></body></html>
```

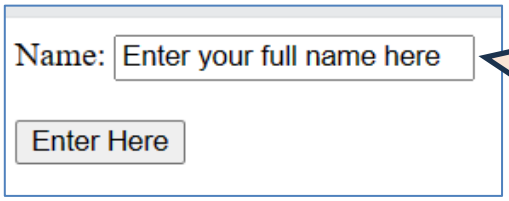
This string is displayed in the paragraph with ID demo



Example 2:

```
<html>
<head>
<script>
function myFunction()
{
   document.getElementById("demo").innerHTML="Hello World";
}
</script>
</head>
<body>
<p>Click the button</p>
<button onclick="myFunction()">Click me</button>
<p id="demo"></p>
</body>
</html>
```

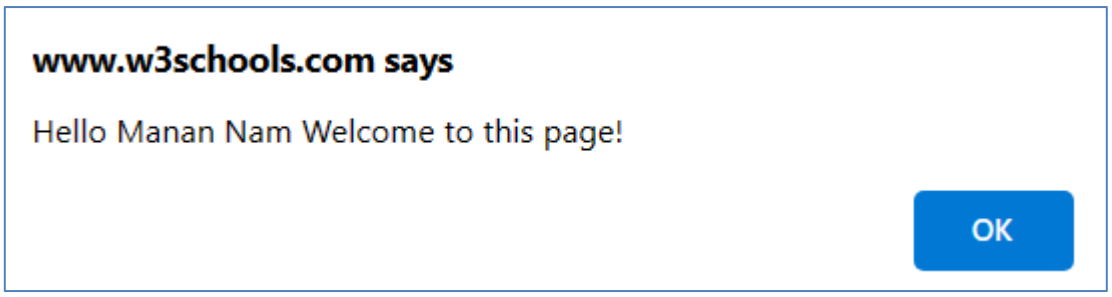
Output a value from an HTML form, in this case in a text box into an alert box in JavaScript.



The value entered in this textbox will be stored in a variable in JavaScript using document.getElementById("myText").value

```
<!DOCTYPE html>
<html>
<body>
Name: <input type="text" id="myText" value="Enter your full name here">
<button onclick="myFunction()">Enter Here</button>

<script>
function myFunction()
{

 name=document.getElementById("myText").value
 alert("Hello " + name +" Welcome to this page!")
}
</script>
</body>
</html>
```

Also see how the strings and variable (along with spaces) are concatenated with the + sign to get a perfect message in the alert box



4.5 Data Types

Data types in JavaScript define the data type that a variable can store. JavaScript supports different data types. JavaScript includes Primitive and Non-Primitive data types. The primitive data types in JavaScript include string, number, Boolean, undefined & null. The non-primitive data type includes the object, array and functions.

Primitive Data Types

Numbers: A number data type can be an integer, a floating-point value, an exponential value, a 'NaN' or a 'Infinity'.

```
1. var a=250  // integer value
2. var b=25.5  // a number containing a decimal
3. var c = 10e4 //  an exponential value which evaluates to 10*10000;
```

There are special numeric values e.g. ***NaN*** *and* ***Infinity***. If a number is divided by 0, the resulting value is infinity.

```
1. 5/0;    // results in infinity
2. The type of infinity is a number
3. typeof(infinity);   // returns number
```

A 'NaN' results when we try to perform an operation on a number with a non-numeric value

```
1. 'hi' * 5; // returns NaN
2. typeof(NaN);  // returns a number
```

String:
The string data type in JavaScript can be any group of characters enclosed by a single or double-quotes or by backticks.

```
1. var str1 = "This is a string1";  // This is a string primitive type or string literal
2. var str2= 'This is a string2';
3. var str3 = `This is a string3`;
```

Boolean

The Boolean data type has only two values, true and false. It is mostly used to check a logical condition. Thus Booleans are logical data types which can be used for comparison of two variables or to check a condition. The true and false imply a 'yes' for 'true' and a 'no' for 'false'. When we check the data type of 'true' or 'false' using typeof operator, it returns a Boolean.

```
1. typeof(true) // returns Boolean
2. typeof(false) // returns Boolean
```

Let's see an example of comparison statement:

```
var a =5;
var b=6;
a==b // returns false
```

Undefined

Undefined data type means a variable that is not defined. The variable is declared but doesn't contain any value. In the following example, the variable 'a' has been declared but hasn't been assigned a value yet.

```
var a;
document.write(a) // This will return undefined.
```

Null

The Null in JavaScript is a data type that is represented by only one value, the 'null' itself. A null value means no value.

```
var a = null;
document.write(a)   // This returns null
typeof(a)
```

Non-Primitive Data Types

These types of data type are complex in nature which consist of more than one component. Objects, arrays and functions are examples of composite data types. Object contains properties and methods; array contains a sequential list of elements and functions contains a collection of statements.

4.6 Variables

JavaScript variables are used to store data that can be changed later on. The variables can be thought of as named containers. You can place data into these containers and then refer to the data simply by naming the container. JavaScript variables must have unique names. These names are called Identifiers.

Variable Naming Convention

JavaScript has some rules when we give the name to the variable. Following factor must be considered while naming the variable.

1. **Case Sensitivity:** JavaScript variables are case sensitive. When name is given to the variable then one must be careful to use the same case in the JavaScript otherwise JavaScript considers it as a new variable and returns the error. For example numvalue, Numvalue, numValue NUMVALUE are four different variables.
2. **Use Valid Characters:** A variable must begin with underscore or letter only. All other characters are invalid to start the variable name. Variables must not contain blank spaces. For example, _numvalue, num2value, numvalue are valid names while 2numvalue, _num value, num value are invalid names.
3. **Avoid Reserved Words:** Reserved words are the special words which have specific meaning in JavaScript. These words cannot be used as a variable name. For example 'while' is a reserved keyword hence cannot be used as a variable name.

break	else	new	var
case	final	return	void
catch	for	switch	while
continue	function	this	with
default	if	throw	delete
in	try	do	instance of
type of	interface	null	undefined

Table: list of reserved keywords in JavaScript

Variable Declaration and Initialization

1. Variables in JavaScript can be defined using the keyword **var (**Note that var is the keyword in JavaScript)
2. The equal to (=) sign is used to assign a value to a variable.
3. Users can either, separately declare the variable and then assign values to it or straight-away declare and initialize the variables.

4. JavaScript variables can hold a value of any data type. For example, you can store the value of number, string, Boolean, object, etc. data type values in JavaScript variables.
5. The value type of a variable can change during the execution of a program and JavaScript takes care of it automatically.

Examples of variables

```
var X;  //defines a variable X, and by default no value is assigned to this variable
        // the variable X has data type undefined

var y = 100;  //defines a variable Y and assigns the integer value of 100 to it

var customer  // declare the variable using the var keyword
customer="Rose"     //assigned a string to variable customer
var sum = x + y ; // sum is a variable that stores the sum of variables x and y
var check=true;  //assigning a Boolean value
```

JavaScript Variable Scope

- **Global Variables** − A global variable has global scope which means it can be defined anywhere in your JavaScript code.
- **Local Variables** − A local variable will be visible only within a function where it is defined. Function parameters are always local to that function.

JavaScript Literals

JavaScript Literals are the values that are assigned to a variable and depending on what literal we assign to a variable its data type will be fixed.
For example:

```
var a= "Roman Numbers"
var b=45
var x=90.78
var c=[34,56,78]
var d=true
```

So a JavaScript Literal can be a numeric, string, floating-point value, an array, Boolean value or even an object.

4.7 Operators

In JavaScript, an **operator** is a symbol that performs an operation on one or more operands, such as variables or values, and returns a result. Let us take a simple expression: 5 **+ 3** is equal to 8. Here 5 and 3 are called **operands**, and '+' is called the **operator**.

Some of the operators that JavaScript supports are:

- Arithmetic Operators
- Comparison (or Relational) Operators
- Logical Operators
- Assignment Operators

Arithmetic Operators

There are following arithmetic operators supported by JavaScript language: assume variable a hold 10 and variable b holds 20 then:

Operator	Description	Example
+ (Addition)	Adds two operands	a + b will give 30
- (Subtraction)	Subtracts second operand from the first	a - b will give -10
* (Multiplication)	Multiplies both operands	a * b will give 200
/ (Division)	Divides numerator by denominator.	b / a will give 2
% (Modulus)	Modulus operator gives the remainder after an integer division.	b % a will give 0
++ (Increment)	Increment operator, increases integer value by one	a++ will give 11
– (Decrement)	Decrement operator, decreases integer value by one	a-- will give 9
** (Exponent)	Can be used to give one value raised to power another value.	a**2 will give 10^2 or 100

Comparison Operators

These are some of the comparison operators supported by JavaScript language. Assume variable a hold 10 and variable b holds 20

Operator	Description	Example
== (Equal)	Checks if the values of two operands are equal or not, if yes then the condition becomes true.	(a == b) is false
!= (Not Equal)	Checks if the values of two operands are equal or not, if values are not equal then the	(a != b) is true

	condition becomes true.	
> (Greater than)	Checks if the value of the left operand is greater than the value of right operand, if yes then the condition becomes true.	(a > b) is false
< (Less than)	Checks if the value of the left operand is less than the value of the right operand, if yes then the condition becomes true.	(a < b) is true
>= (Greater than or equal to)	Checks if the value of left operand is greater than or equal to the value of right operand, if yes then the condition becomes true.	(a >= b) is false
<= (Less than or equal to)	Checks if the value of left operand is less than or equal to the value of right operand, if yes then the condition becomes true.	(a <= b) is true

Comparison statements always return or output a Boolean value, that is true or false.

Logical Operators: These are some of the logical operators supported by JavaScript language. Assume variable a hold 10 and variable b holds 20 then:

Operator	**Description**	**Example**
&& (logical AND)	If both the operands are non-zero numbers then and returns/outputs the second number. For zero it returns 0. In other cases it gives true only if both conditions are true otherwise it gives false.	(10&&20) will output 20 (20&&10) will output 10 (10>9&&7<6) will output false (1<=5 && 2==2.0) will output true
|| (logical OR)	If both the operands are non-zero numbers then and returns/outputs the first number. If one of them is zero it returns the non-zero number. In other cases it gives false only if both conditions are false otherwise it gives true.	(10||20) will output 10 (20||10) will output 20 (45<=49||7<9) will output true (1>=5||2==2.7)will output false

! (logical NOT)	It is used to reverse the logical state of its operand. if a condition is true then logical not operator will make it false.	! (10>9&&7<6) will output true ! (45<=49||7<9) will output false

Assignment Operators: In JavaScript, an assignment operator is used to assign a value to a variable.

Operator	Description	Example
= (Simple Assignment)	Assigns values from the right-side operand to the left side operand.	z = x + y will assign the value of x + y into z
+= (Add and Assignment)	It adds the right operand to the left operand and assigns the result to the left operand.	z += x is equivalent to z = z + x
−= (Subtract and Assignment)	It subtracts the right operand from the left operand and assigns the result to the left operand.	z -= x is equivalent to z = z - x
*=(Multiply and Assignment)	It multiplies the right operand with the left operand and assigns the result to the left operand.	z *= x is equivalent to z = z * x
/= (Divide and Assignment)	It divides the left operand with the right operand and assigns the result to the left operand.	z /= x is equivalent to z = z / x
%= (Modules and Assignment)	It takes modulus using two operands and assigns the result to the left operand.	z %= x is equivalent to z = z % x

Some important differences between Math and programming

- The **"="** symbol represents **equal to** in Math but in programming, it means we are **assigning** a value to a variable which will be stored in the memory to recall later.
- The left-hand side of the equal to sign is always a variable and the right-hand side may be a number, a string, a variable or an expression.

```
x=10
y=6
x=x+1      //here x on the left side will be assigned value 10+1 that is 11
x=x + y    //here x on the left side will be assigned value 11+6 that is 17
```

- The **"=="** symbol is used to compare something on the left side to the right side.

Conditional operator (? :)

The conditional operator is also called the ternary operator, containing the 3 parts. The first part contains the condition and executes the second part if the condition evaluates the true; Otherwise, it executes the third part.

Syntax

Condition ? First statement : Second statement

Parameters

- **Condition** − It is a conditional statement.
- **First Statement** − If the conditional statement evaluates true, First Statement will be executed.
- **Second Statement** − If the conditional statement evaluates false, the Second statement will be executed.

  For example

  status = (marks >= 30) ? "Pass" : "Fail"

  The statement assigns value "Pass" to the variable status if marks are 30 or more. Otherwise, it assigns the value of "Fail" to status.

typeof operator

typeof in JavaScript is an operator used for type checking and returns the data type of the operand passed to it. The operand can be any variable, function, or object whose type you want to find out using the typeof operator.

Syntax

typeof (operand)

Here is the list of return values for the type of operator.

Type	String returned by type of
number	"number"
string	"string"
boolean	"boolean"
object	"object"
Function	"function"
Undefined	"undefined"
null	"object"

Some examples with Arithmetic operators

```
<!DOCTYPE html>
<html><body>
<h3>JavaScript Arithmetic Operators</h3>
<script>
var a=10;
var b=5
c=a+b;
document.write("The value of a+b is: ", c)
document.write("<br>")
c=a-b;
document.write("The value of a-b is: ", c)
document.write("<br>")
c=a*b;
document.write("The value of a*b is: ", c)
document.write("<br>")
c=a/b;
document.write("The value of a/b is: ", c)
document.write("<br>")
c=a%b;
document.write("The value of a%b is: ", c)
document.write("<br>")

document.write("The value of a++ is: ", a++)
document.write("<br>")
document.write("The value of a++ is: ", a)
document.write("<br>")
document.write("The value of ++a is: ", ++a)
document.write("<br>")
document.write("The value of --b is: ", --b)
document.write("<br>")
</script></body>
</html>
```

Observe how the increment operator written after the variable **a** will not increase the value in place, it will be reflected the next time variable **a** is output.

However, the increment or decrement operator written before the variable will increase or decrease the value in place.

JavaScript Arithmetic Operators

The value of a+b is: 15
The value of a-b is: 5
The value of a*b is: 50
The value of a/b is: 2
The value of a%b is: 0
The value of a++ is: 10
The value of a++ is: 11
The value of ++a is: 12
The value of --b is: 4

Increase reflected the next time variable **a** is output.

The increment or decrement reflected in place.

Some examples with Comparison/Relational operators

```
<!DOCTYPE html>
<html>
<body>
<h3>JavaScript Comparison Operators</h3>
<script>
var a;
a=10;
b=5;
document.write("The value of a==b is: ", a==b)
document.write("<br>")
document.write("The value of a!=b is: ", a!=b)
document.write("<br>")
document.write("The value of a>b is: ", a>b)
document.write("<br>")
document.write('The value of 1=="1" is: ', 1=="1" )
document.write("<br>")
document.write('The value of 1==="1" is: ', 1==="1" )
document.write("<br>")
</script>
</body>
</html>
```

JavaScript Comparison Operators

The value of a==b is: false
The value of a!=b is: true
The value of a>b is: true
The value of 1=="1" is: true
The value of 1==="1" is: false

Some examples with Logical operators

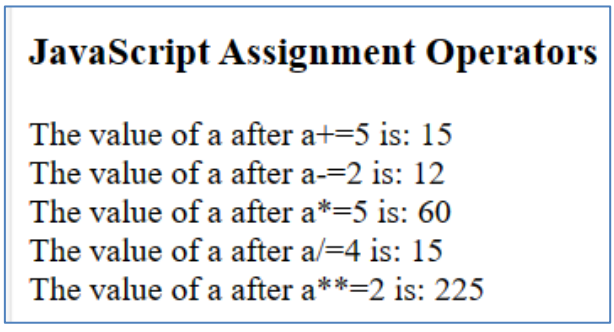
```
<!DOCTYPE html>
<html>
<body>
<h3>Logical Operators in Javascript</h3>
<script>

document.write("0&&5 will output: ",0&&5);
document.write("<br>")
document.write("0||5 will output: ",0||5);
document.write("<br>")
document.write("3&&5 will output: ",3&&5);
document.write("<br>")
document.write("3||5 will output: ",3||5);
document.write("<br>")

document.write("The && operator returns: ",(4>2)&&(4<2));
document.write("<br>")
document.write("The || operator returns: ",(5>2)||(2<2));
document.write("<br>")
document.write("The not with && operator returns: ",!((4>2)&&(4<2)));
document.write("<br>")
document.write("The not with ||operator returns: ",!((5>2)||(2<2)));
</script></body></html>
```

Logical Operators in Javascript

0&&5 will output: 0
0||5 will output: 5
3&&5 will output: 5
3||5 will output: 3
The && operator returns: false
The || operator returns: true
The not with && operator returns: true
The not with ||operator returns: false

Some examples with Assignment operators

```
<!DOCTYPE html>
<html>
<body>
<h3>JavaScript Assignment Operators</h3>
<script>
var a;
a=10;
a+=5;
document.write("The value of a after a+=5 is: ", a)
document.write("<br>")
a-=3
document.write("The value of a after a-=2 is: ", a)
document.write("<br>")
a*=5
document.write("The value of a after a*=5 is: ", a)
document.write("<br>")
a/=4
document.write("The value of a after a/=4 is: ", a)
document.write("<br>")
a**=2
document.write("The value of a after a**=2 is: ", a)
document.write("<br>")
</script>
</body>
</html>
```



Operator Precedence & Associativity

Operator precedence: It ensures the priority of the operators to be executed when a single expression contains multiple operators. So, whatever expressions have higher priority, the compiler executes it first over other operators and then executes the operators with the lower precedence.

Operator Associativity: It refers to the order in which an operator evaluates its operand i.e. from left to right or right to left. When an expression has an operator with equal precedence then association normally is left to right.
For example, the expression 6 + 3 * 10 / 2 will be written as (6 +((3 * 10) / 2)) and operator will be evaluated from left to right.

Operator precedence and associativity in JavaScript			
Operator	Operator Use	Operator Associativity	Operator precedence
( )	Parenthesis	Left to right	1
**	Exponent	Right to left	2
++	Increment	Right to left	3
--	Decrement	Right to left	
!	Logical NOT	Right to left	
/	Division	Left to right	4
*	Multiplication	Left to right	
%	Modulus	Left to right	
+	Addition	Left to right	5
-	Subtraction	Left to right	
>, >=	Greater than, greater than or equal to	Left to right	6
<, <=	Less than, less than or equal to	Left to right	
==	Equality(comparison	Left to right	7
!=	Inequality	Left to right	
&&	Logical AND	Left to right	8
||	Logical OR	Left to right	9
?:	Conditional operator	Left to right	10
=	Assignment	Right to left	11
*=, /=, %=, +=, -=,	Assignment according to the preceding operator	Right to left	12

4.8 Inbuilt functions in JavaScript

parseInt(): This function converts a string into an integer (a whole number) based on the specified radix (base). The radix is optional, default is 10. If the first character of the string cannot be converted to a number, it will return a NaN (Not a Number)
For example:
parseInt ("20") returns 20
parseInt ("34.12") returns 34
parseInt ("20 Pens") returns 20
parseInt ("Pens 20") returns NaN

parseFloat(): It accepts the string and converts it into a floating-point number. If the string does not contain a numeral value or If the first character of the string is not a Number, then it returns NaN.
For example:
parseFloat("65") returns 65
parseFloat("65.33") returns 65.33
parseFloat("14 inch") returns 14
parseFloat("inch 14") returns NaN

valueOf(): This method returns a number of a given variable.

For example:

```
var x =123;
x.valueOf() // will return 123;

var num=NaN
num.valueOf() // will return NaN
```

isNaN(): This method checks whether the passing argument is a number or not. It returns true or false value as a result.
For example:

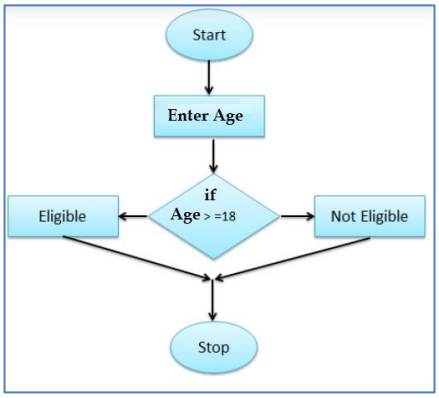
```
document. write(isNaN("Bottles"))  //will return true
document. write(isNaN(22)) //will return false
document. write(isNaN("13/12/2024")) //will return true
document.write(isNaN(-67)) //will return false
```

4.9 Control of Flow using Conditional Statements

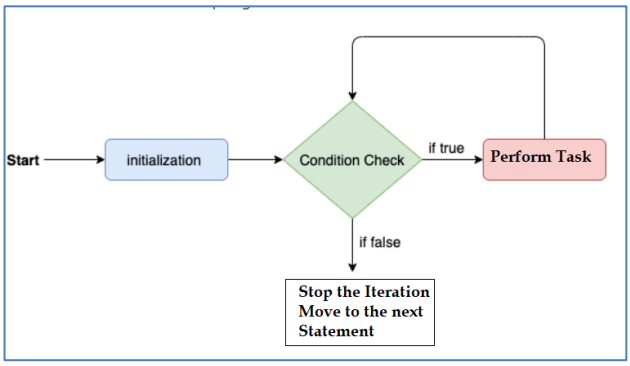
Sometimes we need to write code which can take certain actions based on a variety of scenarios. Control of Flow or Control Flow Structures is an important concept in programming that allows you to run your code under different conditions or until a certain condition is met.

Types of Control Statements in JavaScript:

- **Conditional Statement:** These statements are used for decision-making. Decision-making statements evaluate the statements and control the program flow depending upon the result of the condition provided.

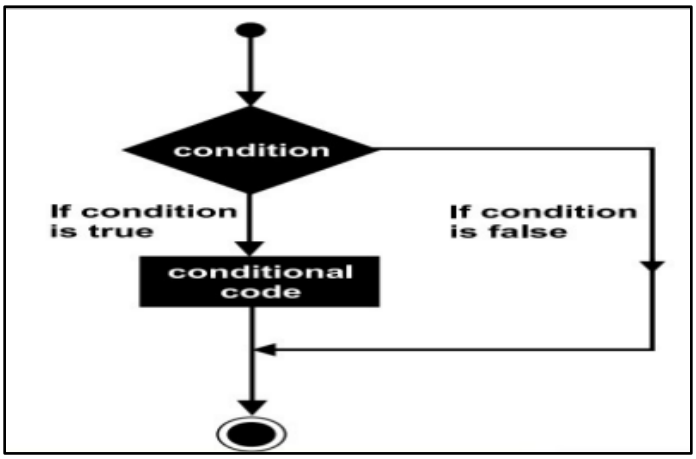


- **Iterative Statement:** This is a statement that iterates repeatedly until a condition is met. It executes the block of code repeatedly while some condition evaluates to true. The execution of the set of instructions depends upon a particular condition.



Conditional Statements:

1.Using if..else statements: The if-else statement will perform some action for a specific condition. If the condition is satisfied, then a particular code of action will be executed. If the condition is not satisfied, it will execute another code of action that satisfies that particular condition.



Example: If the hour is less than 18, create a "Good day" greeting, otherwise "Good evening":

```
if (hour < 18)
 {
  greeting = "Good day";
 }
else
 {
  greeting = "Good evening";
 }
```

The else if statement can be used to specify a new condition if the first condition is false.
Example: If time is less than 10:00, create a "Good morning" greeting, if not, but time is less than 20:00, create a "Good day" greeting, otherwise a "Good evening":

```
if (time < 10)
{
  greeting = "Good morning";
}
else if (time < 20)
{
  greeting = "Good day";
}
else
{
  greeting = "Good evening";
}
```

Example: Calculate discount price base on total amount. If the amount is less than 10000 then no discount. If the amount is less than 20000 then the discount is 5%. If the amount

is less than 30000 then the discount is 10%. If the amount is more than 50000 then the discount is 15%.

```
<html>
<body>
<p>Calculate Discount Total Amount Entered by User </p>
<script>
var tamt =prompt ("Enter Total Amount", "0");

if (tamt == 0) {
document.write("Invalid Amount");
}
else if (tamt < 10000) {
document.write("No Discount");
}
else if (tamt < 20000) {
disAmt = tamt - (tamt *.05);
document.write(" Total Amount after Discount = " + disAmt) ;
}
else if (tamt < 30000) {
disAmt = tamt - (tamt *.1);
document.write(" Total Amount after Discount = " + disAmt) ;
}
else {
disAmt = tamt - (tamt *.15);
document.write(" Total Amount after Discount = " + disAmt) ;
}
</script>
</body>
</html>
```

Important Points to remember:

In the **if-else if-else** construct there can be only one **if** statement and one **else** statement but there can be multiple **else if** statements. The **else** statement does not check a condition.

if(condition)
….statements…..
else if(condition)
….statements…..

else if(condition)
….statements…..

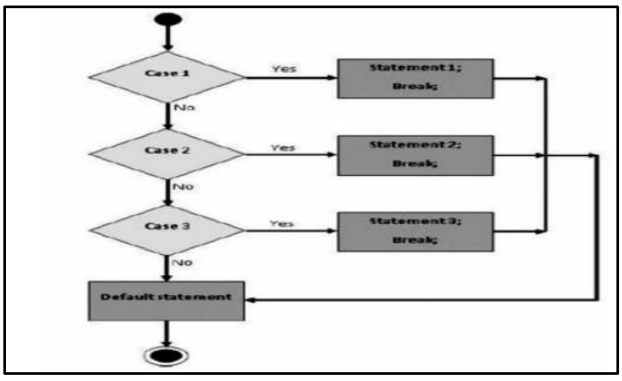
else if(condition)
….statements…..
else
….statements…..

1. **Using switch…case statement:** The switch statement is used to select one of many blocks of the code to be executed in a program. The switch expression is evaluated once and the value of the expression is compared with the values of each case. If a match is found, then the associated block of code is executed, otherwise the default code block is executed.

Syntax:

```
switch (expression)
{
 case value1:
  // statement 1;
    break;
 case value2:
  // statement 2;
    break;
 default:
   // default code block;
}
```

- Expression is the value that you want to compare.
- Case value1, case value2, etc., represent the possible values of the expression.
- **break** statement terminates the switch statement. Without it, execution will continue into the next case.
- **Default** specifies the code to run if none of the cases match the expression.



Example: The following program asks the user to input the day of the week starting with Sunday as 1. It displays the day as a string and displays a message for an invalid number.

```
<!DOCTYPE html>
<html><body>
<h2>Displaying the Day of the Week</h2>
<p id="demo"></p>
<script>
var day;
var daynum=prompt("Enter the Day of the Week (Sunday-1)")
switch(daynum) {
 case 1:
   day = "Sunday";
   break;
 case 2:
   day = "Monday";
   break;
```

```
  case 3:
    day = "Tuesday";
    break;
  case 4:
    day = "Wednesday";
    break;
  case 5:
    day = "Thursday";
    break;
  case 6:
    day = "Friday";
    break;
  case  7:
    day = "Saturday";
   break;
   default:
     document.write("Invalid Number")
}
document.getElementById("demo").innerHTML = "Today is " + day;
</script></body></html>
```

Example: Based on the grade entered show the remark as per the student grade

```
<!DOCTYPE html>
<html><body>
<script>
var grade=prompt("Enter the grade A/B/C");
 switch(grade)
 { case 'A':
   result="Excellent"
   break;
   case 'B':
   result="Very Good"
   break;
   case 'C':
   result="Good"
   break;
   default:
   result="No Grade"
  }
document.write(result);
</script>
</body></html>
```

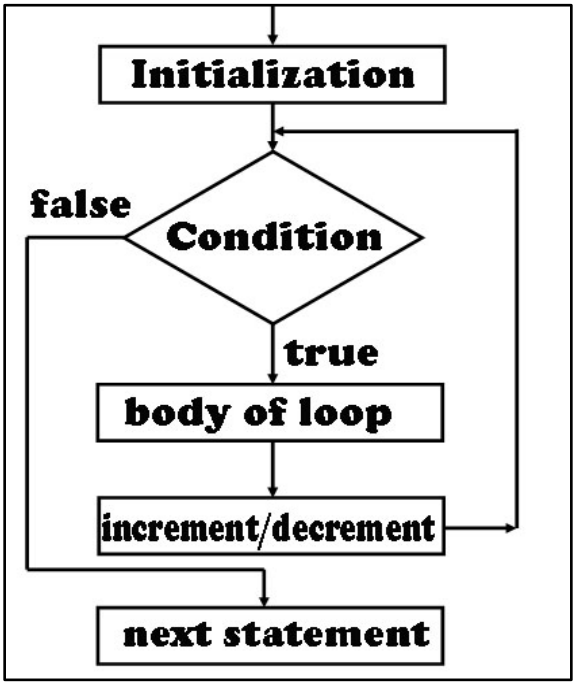
4.10 Control of Flow using Loops

Iterative Statements: These statements repeat a set of instructions a certain number of times. These statements are also called looping statements or loops. A loop has 3 parts as explained below:

- **Initialization:** Initializes a counter variable to start the loop. This is done only once in a loop.
- **Condition:** Specifies a condition that must evaluate to true for the next iteration. If the condition evaluates to true, the statements inside the loop body will execute, If the condition evaluates to false, the loop will terminate.
- **Iteration:** Increase or decrease the counter variable after every pass of the loop.

1. **for loop:** The for loop is used to repeat the set of instructions for a fixed number of times. The syntax to use the for loop is as follows:

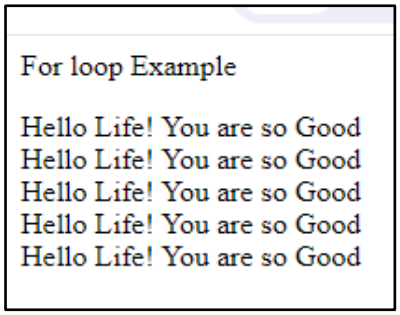
```
for (Initialization; condition; iteration)
{
  //set of JavaScript statements
}
```



Example: To display the text "Hello Life! You are so Good" for 5 times

```
<html>
<body>
<p> For loop Example</p>
<script>
for (i = 0; i < 5; i++) {
document.write("Hello Life! You are so Good" +"<br>"); }
</script>
</body>
</html>
```

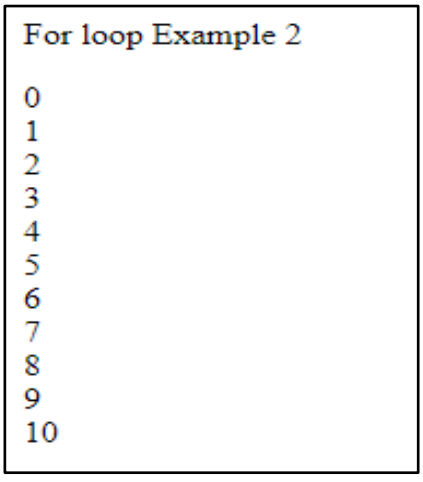
Output



Example: A program to display numbers from 1 to 10

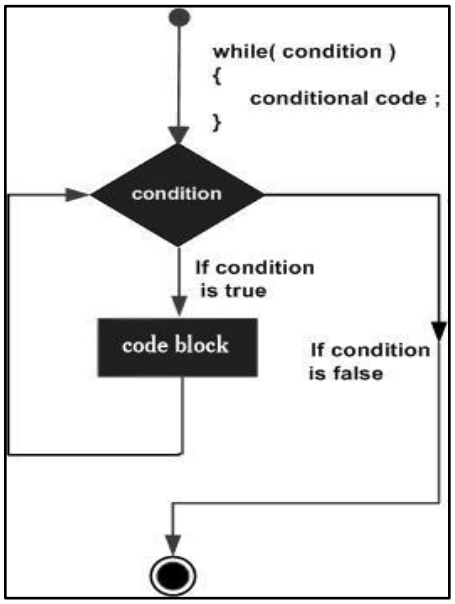
```
<html>
<body>
<p> For loop Example 2</p>
<script>
for (i = 0; i <=10; i++) {
document.write(i +"<br>"); }
</script>
</body>
</html>
```

Output:



2. **while loop:** The while loop is also used to repeat a set of instructions until a conditional statement returns true. Once the condition returns false, the loop is terminated. The syntax to use the while loop is as follows:

```
Initialization;
while (condition)
{
  //set of JavaScript statements
  iteration;
}
```



Why while loops?
In advanced programming while loops are used when we may not have a fixed number of iterations. For example in a video game we want the user click a certain key to end the game, we can use a while loop to make the game go on and it will break only if that key is pressed. In for loops the number of iterations is mostly fixed.

Example: A program to display the square of numbers between 1 to 5

```
<html>
<body>
<p> While loop Example 1</p>
<script>
var i=1
while (i<=5)
{
 var square= i * i;
 document.write("Square of "+i+" "+square"<br>");
 i++;
}
</script></body></html>
```

Output:

While loop Example 1

Square of 1: 1
Square of 2: 4
Square of 3: 9
Square of 4: 16
Square of 5: 25

Example: The same example of for Loop can be executed with the use of while loop

```
<html>
<body>
<p> While loop Example 2</p>
<script>
var i=1
while (i<=5)
{
    document.write("Hello Life! You are so Good" +"<br>");
    i++;
}
</script>
</body>
</html>
```

While loop Example 2

Hello Life! You are so Good
Hello Life! You are so Good
Hello Life! You are so Good
Hello Life! You are so Good
Hello Life! You are so Good

Use of break and continue statements in loops

The break statement can be used within a loop to stop execution based on a condition being true. Since the loop is stopped, the statements after the loop will follow.

```
<!DOCTYPE html>
<html>
<body>
<h2>JavaScript Loops</h2>
<p>A loop with a <b>break</b> statement.</p>
<p id="demo"></p>
<script>
let text = "";
for (let i = 0; i < 10; i++)
{
  if (i === 3)
{
break;
}
  text += "The number is " + i + "<br>";
}
document.getElementById("demo").innerHTML = text;
</script>
</body>
</html>
```

Output:

JavaScript Loops

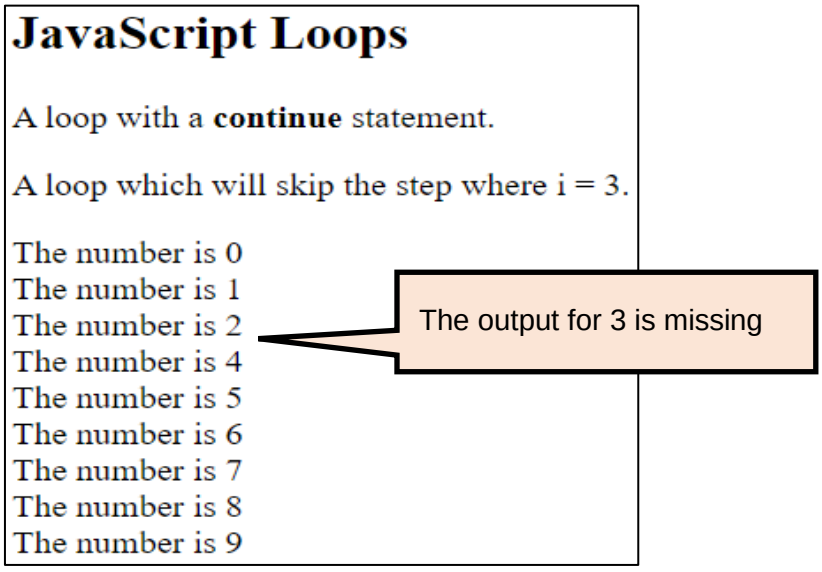
A loop with a **break** statement.

The number is 0
The number is 1
The number is 2

The loop stops as soon as i becomes equal to 3

The continue statement is used with a loop to ignore execution when a condition is true. This statement will not stop the execution of the rest of the loop.

```
<!DOCTYPE html>
<html>
<body>
<h2>JavaScript Loops</h2>

<p>A loop with a <b>continue</b> statement.</p>
<p>A loop which will skip the step where i = 3.</p>
<p id="demo"></p>
<script>
let text = "";
for (let i = 0; i < 10; i++)
{
  if (i === 3)
 {
continue;
 }
  text += "The number is " + i + "<br>";
}
document.getElementById("demo").innerHTML = text;
</script>
</body>
</html>
```



Exercise

I. Multiple Choice Questions

1. Which of the following keywords is used to initialize variables in JavaScript?
   a. for
   b. getElement
   c. var
   d. Boolean

2. _____ can provide the information to the user or ask him to input the value required by a program.
   a. Browser
   b. Dialog box
   c. Value box
   d. Window box

3. A multiline comment in JavaScript is written as:
   a. /* Welcome to JavaScript
   b. // Welcome to JavaScript
   c. / Welcome to JavaScript
   d. /* Welcome to JavaScript */

4. JavaScript is an _____ computer programming language.
   a. Interpreted
   b. Compiler
   c. Assembler
   d. None of the above

5. _____ is used to convert string to integer value.
   a. Integer()
   b. ParseInt()
   c. parseInt()
   d. Stringtoinit()

6. What will be the output of the following JavaScript code?
   var s= "welcome";
   var x=30;
   alert(s+x);
   a. welcome 30
   b. welcome30
   c. welcome
   d. 30

7. What are the three important parts in a for loop in JavaScript?
    a. initialization; assignment; increment
    b. initialization; condition; increment
    c. initialization; increment; condition
    d. condition; increment; initialization

8. When a variable is declared without assigning value then it contains the _____ value which may cause the runtime error.
    a. Null
    b. Zero
    c. Undefined
    d. New
9. A _____ is a set of statements enclosed in braces as a single statement.
    a. block
    b. expression
    c. box
    d. Sequence

10. <script> tag can be inserted in the:
    a. Head section
    b. Body section
    c. Both a & b
    d. None of these

11. An alternative to an if..else statement is:
    a. loop
    b. conditional operator
    c. typeof operator
    d. arithmetic operator

12. An external JavaScript file will have the extension as:
    a. .html
    b. .JavaScript
    c. .jc
    d. .js

13. If a number is divided by 0, the resulting value is _____..
    a. zero
    b. infinity
    c. null
    d. none of the above

14. The output of the following will be:
    document.write("english"*75);

a. infinity
b. english75
c. NaN
d. english*75

15. _____statement terminates the switch statement, without which the execution will continue into the next case.
    a. end
    b. break
    c. stop
    d. continue

16. In the switch..case statement,_____ specifies the code to run if none of the cases match the expression.
    a. omit
    b. skip
    c. fail
    d. default

17. Which of the following is not a valid variable name?
    a. grade_pay
    b. 2gradepay
    c. gradepay
    d. GradePay

18. Which of the following was also the name of JavaScript?
    a. LiveScript
    b. VBscript
    c. BasicScript
    d. None of the above

19. There can be only one if statement in if-else if-else construct.
    a. True
    b. False

20. _____ operator and gives the remainder of after an integer division.
    a. /
    b. %
    c. &&
    d. <=

II. Find the Output of the following code snippets:

i) 21 + 10 ** 2 / 4 – 45

ii) !(true || false && true)

iii) 15%7+10<19%8+6*2

iv)

```
var x=17,y=12;
      x--;
      x+=10;
        y+=x
       y++
document.write(y)
```

v)

```
<!DOCTYPE html>
<html>
<body>
<p> While loop</p>
<script>
var i=1;
while(i<= 5)
{
if( i==3)
continue
i++
document.write("The number is " + i+ "<br>");
}
</script>
</body>
</html>
```

vi)

```
<script>
 for (i-0; i<5; i++)
 {
   document.write("@" + "<br>");
}
</script>
```

vii)

```
var i=10;
var j=2;
while(i > j)
{
  i = i / j;
  document.write(i+" ");
}
```

III. Very Short Answer Questions

1. Who developed JavaScript? What was the initial name of JavaScript?
2. Why is JavaScript called an Interpreted language?
3. Write any 3 advantages of JavaScript
4. What are the tools required to write and view a JavaScript program?
5. What is meant by "External JavaScript file"?
6. What are the types of comments in JavaScript?
7. List the common errors that are usually made in a JavaScript program.
8. What are Data types?
9. Which are the non-primitive data types?
10. What are variables? Declare a variable subject and give it a value as "Web Application".
11. Differentiate between local and global variables.
12. What are different types of operators in JavaScript?
13. What is the use of typeof operator?
14. What is the difference between operator precedence and associativity?
15. Differentiate between conditional statements and iterative statements.

IV. Short Answer Questions

1. Differentiate between a static and dynamic website.
2. Explain any 4 features of JavaScript.
3. Explain the tag used to write a JavaScript program? Write about its attributes also.
4. Explain the document.write method
5. Differentiate between the alert(), prompt() and confirm() dialog boxes.
6. Explain the primitive data types of JavaScript.
7. What are the rules for naming the variables?
8. Explain the conditional operator with an example.

V. Write the JavaScript program for the following:

1. Write a program to print numbers from 1 to 20.
2. Write a program to print numbers from 1 to 20 in reverse order.
3. Write a program to print sum of numbers between 1 to 20
4. Write a program to find if the number entered by the user is even or odd.
5. Write a program to find the sum of all even and odd numbers between 1 to 50.
6. Write a program to calculate the area of a triangle. **Formula: (base/height)/2**

7. Write a program to accept a number from the user. Based on the choice 'S' or 'C' entered by the user, calculate and display:
   i. Square of the number
   ii. Cube of the number
8. Write a program to accept 2 numbers from the user and calculate the area of a rectangle and square.
   i. Area of rectangle = length & breadth
   ii. Area of square = side * side
9. Write a program to check if a number entered is positive, negative or zero.
10. Accept 3 numbers from the user and write a program to find the largest/smallest of 3 numbers
11. Write a program to display a multiplication table for a number.
12. Write a program to find the area and circumference of a circle. **(pi = 3.14)**
    i. Area of circle = 2 * pi * r
    ii. Circumference of circle = pi * r * r
13. Write a program to find the factorial of a number entered by the user.
14. Use the loop to print the numbers 8,11,14,17,20, . . . ,83,86,89
15. Write a program to accept salary and grade from the user. Based on the grade add the bonus to the salary and display net salary

Grade	**Bonus**
A	30% of salary
B	35% of salary
C	20% of salary
D	15% of salary

16. Calculate discount price base on total amount. If the amount is less than 10000 then no discount. If the amount is less than 20000 then the discount is 5%. If the amount is less than 30000 then the discount is 10%. If the amount is more than 50000 then the discount is 15%.

VI. Answer the following questions

1. Amit has written the following code in JavaScript, but it is not getting displayed on the browser. He is not able to find the errors in his program. Find the errors in the following code and rewrite the correct code:

```
<script>
    var prod, counter;
    prod=2;
    ctr =2;
   while (ctr<=9)
      prod= prod * ctr;
       ctr = ctr+2;
       document.right(prod + ", "+ ctr +"<br>");
   }
```

2. Give the JavaScript statement for each of the following:
    a. A variable 'sum' that stores the sum of variables 'A' & 'B'
    b. To print the data type of variable 'k'
    c. To take 'age' as input from the user

3. Rohit, a student of class 12, is learning JavaScript. During examination, he has been assigned an incomplete JavaScript code (shown below). Help him in completing the code :

Incomplete Code

```
var bs=_____(prompt("Enter salary")); --------- line 1
var da,hra,gs;
if(bs>=10000 _____ bs<=20000) ---------- line 2
{
  da=bs*15/100;
  hra=bs*10/100;
}
_____ (bs>=20001 && bs<=30000) ---------- line 3
{
da=bs*12/100;
hra=bs*8/100;
}
else if(bs>=30000)
{
da=bs*10/100;
hra=bs*6/100;
}
gs=bs+da+hra;
_____(gs); ---------- line 4
```

4. The following program is given using if...else if. Use the same program to convert using switch..case without changing the meaning of the program:

```
<script>
        var grade=parseInt(prompt("Enter grade of employee"))
        if (grade=="A")
        document.write("You are Vice President")
        else if(grade=="B")
        document.write("You are LEVEL 1 Officer")
        else if(grade=="C")
        document.write("You are a Trainee")
        else
        document.write("You are not employed")
</script>
```