Internet & Web Technology

FOR DIPLOMA STUDENTS

Lecture Notes Prepared

by

Subrata Parida (Sr. Lect.)

IT Dept.

JHARSUGUDA ENGINEERING SCHOOL, JHARSUGUDA

INTERNET AND WEB TECHNOLOGY (Syllabus)

Semester & Branch: 5th.sem CSE/IT
Theory: 4 Periods per Week
Total Periods: 60 Periods per Semester
Examination: 3 Hours

Teachers Assessment : 10 Marks
Class Test : 20 Marks
End Semester Exam : 70 Marks
TOTAL MARKS : 100 Marks

Objective: After completion of this course the student will be able to:

=> Understand the concept of Internet and its wide application in various areas.
=> Understand different internet connectivity and ISP.
=> Understand the Internet security and Applications
=> Know the methods of development of Portals using HTML
=> Know the Client side Scripting using JavaScript
=> Know the server side Scripting using PHP
=> Know the what is ASP and what can it do

1.0 Internet Basics
1.1 Computer network
1.2 Concept of Internet, Intranet, Modem
1.3. IP Address, Internet Domains, CIDR Notation, ISP, TCP/IP

2.0 Internet Connectivity & WWW
2.1 Introduction ot connectivity
2.2 Medium and methods of connectivity, ISDN, VSAT, RF Link
2.3 Working of Internet
2.4 Introduction to WWW, Application Level Protocol
2.5 Web Browser, URL, Hyper text, Hyperlinks, Hypermedia,
2.6 Search Engine, Proxy sever, CGI, URI, Dreamweaver

3.0 Internet Security
3.1 Introduction to security
3.2 Types of security, Authentication & Authorization
3.3 Firewalls, Encryption & Decryption, SSL

4.0 Internet Application
4.1 E-Mail, Email protocols
4.2 Telnet
4.3 FTP
4.4 Newsgroup
4.5 Chartroom
4.6 Internet Relay Chat
4.7 Video Conferencing
4.8 E-Commerce

5.0 Website Classifications
5.1 Static Websites
5.2 Dynamic websites
5.3 Web portals
5.4 Social Networking Sites
5.5 RSS Feed, Blog, Netiquette

6.0 Development of Portals Using HTML
6.1 Design a webpage, Good Web Design
6.2 HTML Introduction
6.3 HTML Tags, Anchor Tag, Table Tag
6.4 HTML Frames, Forms
6.5 Disadvantages of HTML
6.6 Separating style from structure with style sheets
6.7 CSS Rules, Types of CSS

7.0 Client side Scripting with JavaScript
7.1 Introduction to script, Client side Scripting, Types of Scripting
7.2 Variables in JavaScript, Built-in Function
7.3 Arrays in JavaScript, Conditional statements, Loops
7.4 Document Object Model
7.5 Creating Functions, objects in JavaScript
7.6 Event handling in JavaScript
7.7 Embedding JavaScript with HTML
7.8 Working with Cookies
7.9 Connecting database using JavaScript in HTML Page
7.10 Working with Browser, validating and submitting Forms

8.0 Server Side Scripting
8.1 Introduction to server side Scripting
8.2 Components of SSS
8.3 Difference between CSS and SSS
8.4 Server side Scripting method
8.5 JavaScript on server
8.6 SQL
9.0 Server Side Programming using PHP
9.1 Introduction to PHP
9.2 Variables, string, operator types
9.3 Conditional statement, Loops
9.4 Array
9.5 GET and POST Method and Sessions

Books Recommended:-

1. Internet and Web Designing By Neha Dutta, Adesh Pandey - (Katson Books)
2. Internet & Web page Design By Sisodia - (BPB Publication)
3. Web Technologies By U.K Roy - (Oxford Univ.Press)

Chapter-1

Internet basics

Computer Network:

Connections and Communications between set of computers are called Computer Network.

Connections between computers are done through wired or wireless medium and communication between computers are done via several protocols.

File and Resource Sharing are the main advantages of computer network.

The connections between computers are built by following several topologies such as bus, star, ring, mesh, hybrid as per the requirement of the organizations.

The communications between transmitter and receiver is simplex, half duplex or full duplex mode of communications.

The connections between computers may be point to point or multipoint.

Internet:

The Internet is a global system of interconnected computer networks that use the standard Internet Protocol Suite (TCP/IP).

It is a network of networks that consists of millions of private, public academic, business, and

Government networks.

The Internet carries a vast range of information resources and services, such as the inter-linked hypertext documents and applications of the World Wide Web (WWW), electronic mail, telephony, and file sharing.

The Internet has no single centralized governance in either technological implementation or policies for access and usage; each constituent network sets its own policies.

The technical supporting and standardization of the core protocols(IPv4 and IPv6) is an activity of the Internet Engineering Task Force (IETF), a non-profit organization.

To maintain interoperability, the principal name spaces of the Internet are administered by the Internet Corporation for Assigned Names and Numbers (ICANN).

Intranet:

An **intranet** is a computer network for sharing information, collaboration tools, operational systems, and other computing services within an organization.

It is usually for the exclusion of access by outsiders. The term is used in contrast to public networks, such as the Internet, but uses most of the same technology based on the Internet Protocol Suite.

However, an extranet is a controlled private network that allows access to partners, vendors and suppliers or an authorized set of customers.

It is normally a subset of the information accessible from an organization's intranet.

Modem:

A **modulator-demodulator**, or simply **modem**, is a computer hardware device that converts data from a digital format into a format suitable for an analog transmission medium such as telephone or radio.

A modem transmits data by modulating one or more carrier wave signals to encode digital information, while the receiver demodulates the signal to recreate the original digital information.

The goal is to produce a signal that can be transmitted easily and decoded reliably.

Modems can be used with almost any means of transmitting analog signals, from light-emitting diodes to radio.

WWW:

The World Wide Web, abbreviated as WWW and commonly known as the Web, is a system of interlinked hypertext documents accessed via the Internet.

With a web browser, one can view web pages that may contain text, images, videos, and other multimedia and navigate between them via hyperlinks.

IP address:

An IP address is **a unique address that identifies a device on the internet or a local network**. IP stands for "**Internet Protocol**," which is the set of rules governing the format of data sent via the internet or local network.

Internet Protocol version 4 (IPv4) defines an IP address as a 32-bit number. However, because of the growth of the Internet and the depletion of available IPv4 addresses, a new version of IP (IPv6), using 128 bits for the IP address, was standardized in 1998.

The IP address space is managed globally by the Internet Assigned Numbers Authority (IANA).

IPv4 addresses are usually represented in dot-decimal notation, consisting of four decimal numbers, each ranging from 0 to 255, separated by dots, e.g., *192.0.2.1*. Each part represents a group of 8 bits (an octet) of the address.

Subnet Masking or Subnetting identifies a network from an Ip address.

E.g. Ip address: 192.168.137.60

Subnet Mask: 255.255.255.0

There are two types of Ip-address: Class full and classless.

Above is the example of Classfull Ip-address where ip addresses are categorized into 5 classes:A, B, C, D,E and accordingly subnet masking is done for that ip-address.

Whereas, due to insufficient availability of ip-address, classless ip-address is evolved.

e.g. 192.168.137.60/24

where 24 is bits reserved for network address.

CIDR Notation (Classless Inter-Domain Routing):

CIDR notation is a compact representation of an IP address and its associated network mask.

The notation was invented by Phil Karn in the 1980s.

CIDR notation specifies an IP address, a slash ('/') character, and a decimal number. The decimal number is the count of consecutive leading *1*-bits (from left to right) in the network mask. The number can also be thought of as the width (in bits) of the network prefix. The IP address in CIDR notation is always represented according to the standards for IPv4 or IPv6.

For example:

- *198.51.100.14/24* represents the IPv4 address *198.51.100.14* and its associated network prefix *198.51.100.0*, or equivalently, its subnet mask *255.255.255.0*, which has 24 leading *1*-bits.
- the IPv4 block *198.51.100.0/22* represents the 1024 IPv4 addresses from *198.51.100.0* to *198.51.103.255*.

Internet Domain

An Internet domain name is a unique name of an organization or person on the Internet. The name is combined with a generic top-level domain (gTLD), such as .com or .org.

In order for a domain name to be visible on the Internet, it must first be registered with any one of hundreds of registrars, which will verify that the name is absolutely unique.

Unrestricted gTLDs	
.com	commercial
.net	network oriented
.org	non-profit organization

Restricted gTLDs	
.edu	accredited U.S. educational
.gov	U.S. government agencies
.mil	U.S. military
.int	international treaties (1988)

In simple terms, the **Internet domain** is part of a system in which Internet addresses are linked to specific locations on the Internet – servers, websites, e-mail servers etc.

An **Internet domain** name is your own Internet address which you have registered as part of a national or international Internet domain.

It is an integral part of a website and e-mail address and thus identifies you on the Internet.

ISP:

An **Internet service provider** (**ISP**) is an organization that provides a numerous of services for accessing, using, or participating in the Internet.

Internet services typically provided by ISPs can include Internet access, Internet transit, domain name registration, web hosting, Usenet service, and colocation.

An ISP typically serves as the access point or the gateway that provides a user access to everything available on the Internet.

Access provider ISPs provide Internet access, employing a range of technologies to connect users to their network. e.g. BSNl, Ortel, Jio

TCP/IP:

In computing, a protocol is a set of rules which is used by computers to communicate with each other across a network.

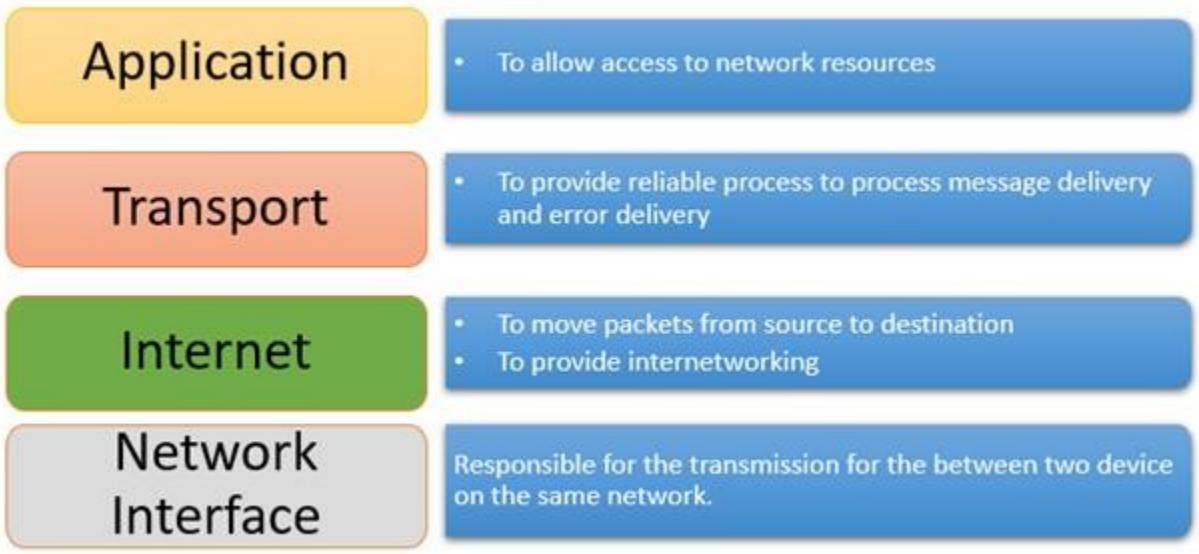
A protocol is a convention or standard that controls or enables the connection, communication, and data transfer between computing endpoints.

The Internet Protocol Suite is the set of communications protocols used for the Internet and other similar networks. It is commonly also known as TCP/IP named from two of the most important protocols in it: The Transmission Control Protocol (TCP) and the Internet Protocol (IP), which were the first two networking protocols defined in this standard.

The **Internet protocol suite,** commonly known as **TCP/IP,** is the set of communications protocols used in the Internet and similar computer networks.

The Internet protocol suite provides end-to-end data communication specifying how data should be packetized, addressed, transmitted, routed, and received.

TCP/IP Stack is specifically designed as a model to offer highly reliable and end-to-end byte stream over an unreliable internetwork.



1. The message would start at the top of the protocol stack on your computer and work it's way downward.
2. If the message to be sent is long, each stack layer that the message passes through may break the message up into smaller chunks of data. This is because data sent over the Internet (and most computer networks) are sent in manageable chunks. On the Internet, these chunks of data are known as **packets**.

3. The packets would go through the Application Layer and continue to the TCP layer. Each packet is assigned a **port number**. We need to know which program on the destination computer needs to receive the message because it will be listening on a specific port.
4. After going through the TCP layer, the packets proceed to the IP layer. This is where each packet receives it's destination address, 5.6.7.8.
5. Now that our message packets have a port number and an IP address, they are ready to be sent over the Internet. The hardware layer takes care of turning our packets containing the alphabetic text of our message into electronic signals and transmitting them over the phone line.
6. On the other end of the phone line your ISP has a direct connection to the Internet. The ISPs **router** examines the destination address in each packet and determines where to send it. Often, the packet's next stop is another router. More on routers and Internet infrastructure later.
7. Eventually, the packets reach computer 5.6.7.8. Here, the packets start at the bottom of the destination computer's TCP/IP stack and work upwards.
8. As the packets go upwards through the stack, all routing data that the sending computer's stack added (such as IP address and port number) is stripped from the packets.
9. When the data reaches the top of the stack, the packets have been re-assembled into their original form, "Hello computer 5.6.7.8!"

Chapter 2

Internet Connectivity & WWW

Introduction to Connectivity:

Internet connections are of different types depending upon the cost, speed, availability, etc.

Medium of Connectivity:

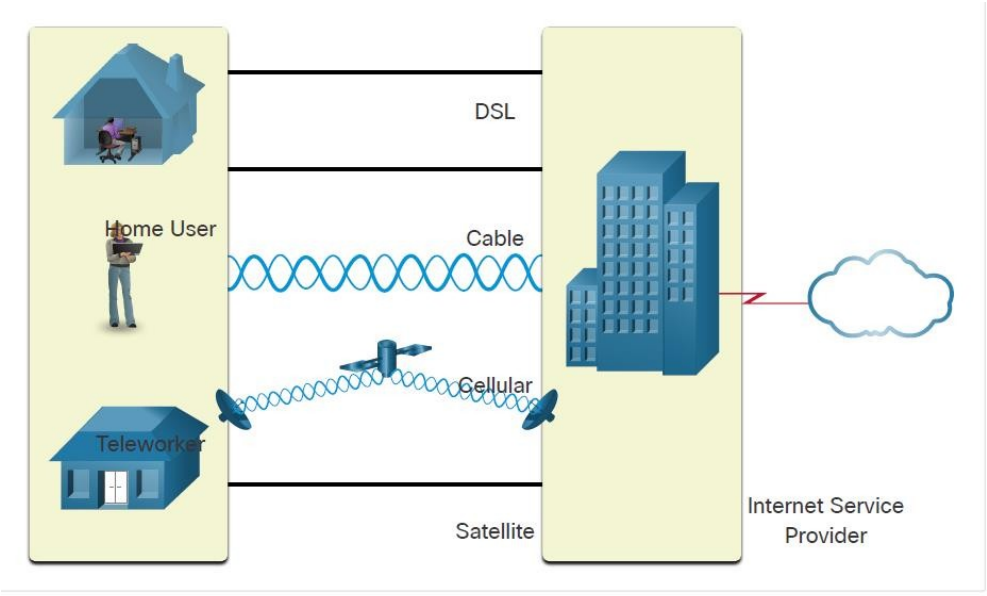
Communication transmits across a network on media. The media provides the channel over which the message travels from source to destination.

Modern networks primarily use three types of media to interconnect devices:

- **Metal wires within cables** - Data is encoded into electrical impulses.
- **Glass or plastic fibers within cables (fiber-optic cable)** - Data is encoded into pulses of light.
- **Wireless transmission** - Data is encoded via modulation of specific frequencies of electromagnetic waves.

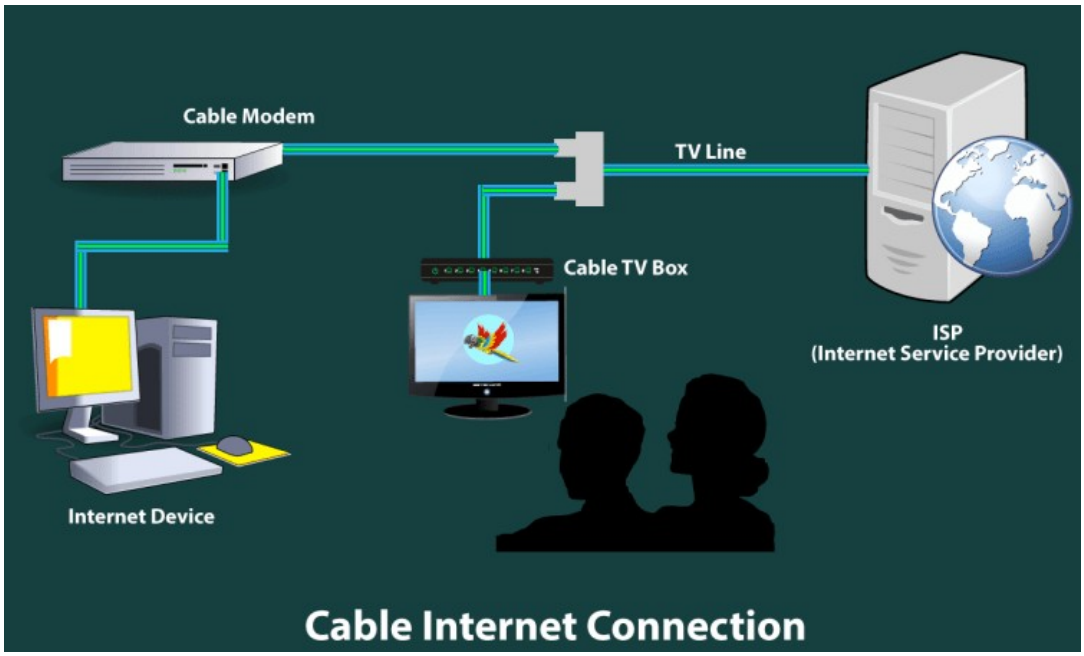
Method of Connectivity:

The figure illustrates common connection options for small office and home office users.



- **Cable** - Typically offered by cable television service providers, the internet data signal transmits on the same cable that delivers cable television.

  It provides a high bandwidth, high availability, and an always-on connection to the internet.



Broadband Cable is typically provided by the same provider that provides cable television to its customers.

Here we use cable modem/wifi router combo.

Because of the connection of your neighbors and you are on the same cable line so during the peak hours you have experienced slow speed. The reason behind is that your neighbors are also using at the same time.

- **DSL** - Digital Subscriber Lines also provide high bandwidth, high availability, and an always-on connection to the internet.

DSL runs over a telephone line.

In general, small office and home office users connect using Asymmetrical DSL (ADSL), which means that the download speed is faster than the upload speed.

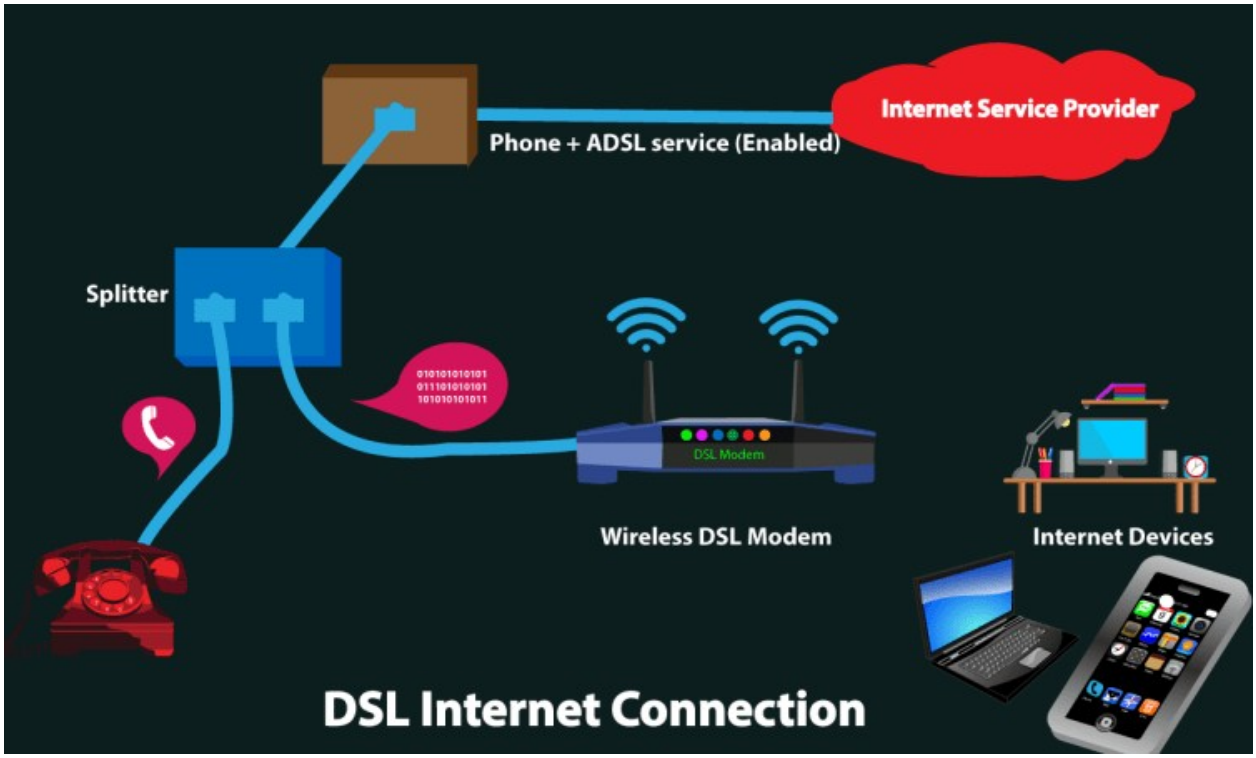
In a DSL internet connection, both voice and internet data can flow over the same telephone line at the same time.

It has a DSL modem which filters the voice and data.

DSL represents a high-speed connection, much faster than the ancient dial-up connection, but it is not as fast as a cable connection.

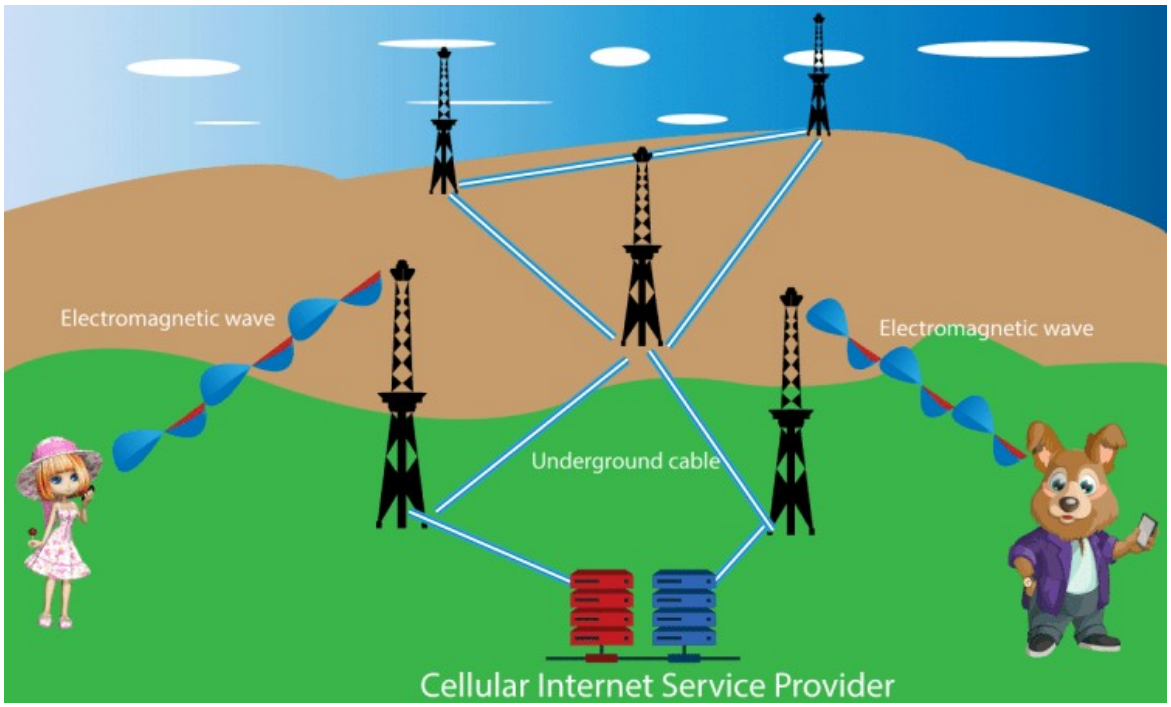
So due to DSL modem, you can go on the internet and talk on the phone at the same time (Modem convert analog signal to digital signal).

Speed of DSL varies; depending upon the place where you live, but on an average, download speed varies from 5Mbps to 100 Mbps.



- **Cellular** - Cellular internet access uses a cell phone network to connect. Wherever you can get a cellular signal, you can get cellular internet access.

  Performance is limited by the capabilities of the phone and the cell tower to which it is connected.



In this network, radio waves are used for transfer signal to and from the mobile phone.

In cellular technology geographic area is divided into hexagonal cells, with it each cell having its tower and frequency slot.

Generally, these cellular towers are connected through a wire or more specifically optical fiber cable. These optical fiber cables are laid under the ground or ocean for providing national or international connectivity.

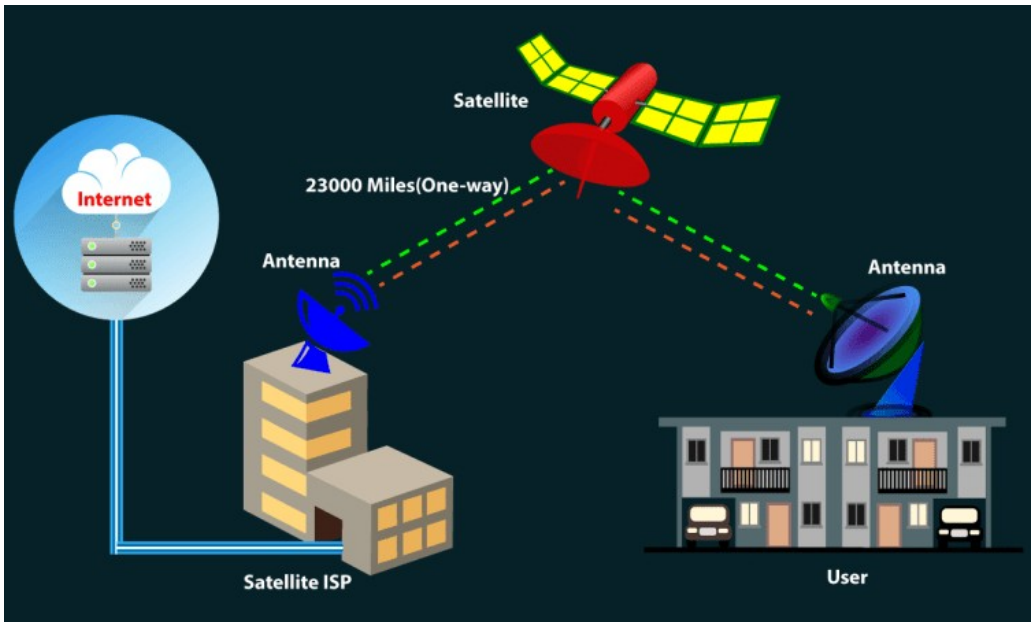
During the use of the internet; Smartphone transmits a signal in the form of electromagnetic waves.

The electromagnetic wave produces by your phone are picked up by the tower and system on the tower convert them high-frequency light pulses.

These light pulses are further transferred to the base transceiver for the further signal transferring.

After that signal transferred to the destination server, for what you searched, after that reverse process takes place.

- **Satellite** - The availability of satellite internet access is a benefit in those areas that would otherwise have no internet connectivity at all. Satellite dishes require a clear line of sight to the satellite.



Satellite internet is the internet which uses the path of the outer space. Satellite internet signal travels more than 46,000 miles (From ISP to user through satellite).

In rural area satellite internet will be the best option for high speed, it is a faster option than dial-up.

- **Dial-up Telephone** - An inexpensive option that uses any phone line and a modem. The low bandwidth provided by a dial-up modem connection is not sufficient for large data transfer, although it is useful for mobile access while traveling.

The dial-up line uses PSTN (Public Switched Telephone Network).

PSTN may be of the telephone line, fiber optic, Microwave transmission or communications satellite.

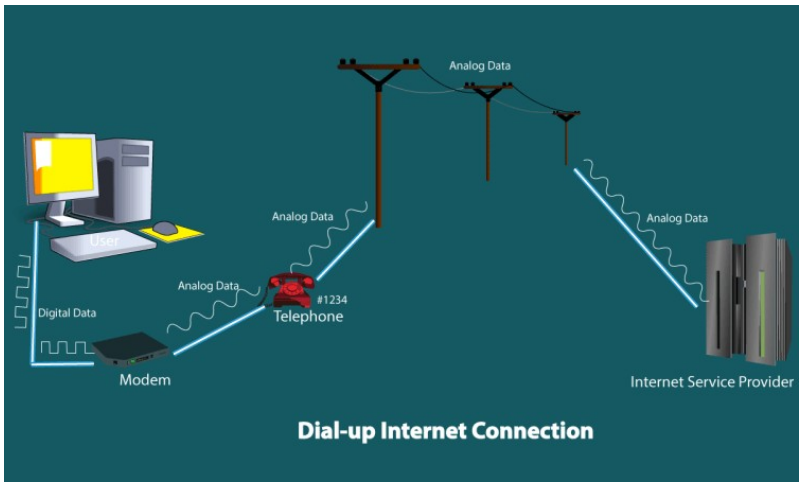
Dial-up internet connections are connected through a telephone line.

For accessing the internet, you have to dial a specific number, and Your ISP will open the internet line for you.

Here is the same line for internet and telephone, so at a time only one task can take place either you can run internet or talk on the telephone. It is the main disadvantage of a Dial-up connection.

It is the most inexpensive form of the internet connection, apart from it, its speed is also very slow from 28 Kbps to 56 Kbps.

It has a very high ping (Latency) speed so we cannot watch online movies and play online games.



Corporate connection options differ from home user options. Businesses may require higher bandwidth, dedicated bandwidth, and managed services. Connection options that are available differ depending on the type of service providers located nearby.

Wireless Broadband

In many areas where cable and DSL are not available, wireless may be used to connect to the internet.

Wireless Internet Service Provider

A Wireless Internet Service Provider (WISP) is an ISP that connects subscribers to a designated access point or hot spot using similar wireless technologies found in home wireless local area networks (WLANs). WISPs are more commonly found in rural environments where DSL or cable services are not available.

Although a separate transmission tower may be installed for the antenna, typically the antenna is attached to an existing elevated structure, such as a water tower or a radio tower. A small dish or antenna is installed on the subscriber's roof in range of the WISP transmitter. The subscriber's access unit is connected to the wired network inside the home. From the perspective of the home user, the setup is not much different than DSL or cable service. The main difference is that the connection from the home to the ISP is wireless instead of a physical cable.

Wireless Broadband Service

This solution uses the same cellular technology as a smart phone. An antenna is installed outside the house providing either wireless or wired connectivity for devices in the home. In many areas, home wireless broadband is competing directly with DSL and cable services.

ISDN,

ISDN stands for Integrated Service Digital Network.

This connection is nearly similar to the DSL.

ISDN is a set of international communication standards designed in 1980 and improved in the 1990s.

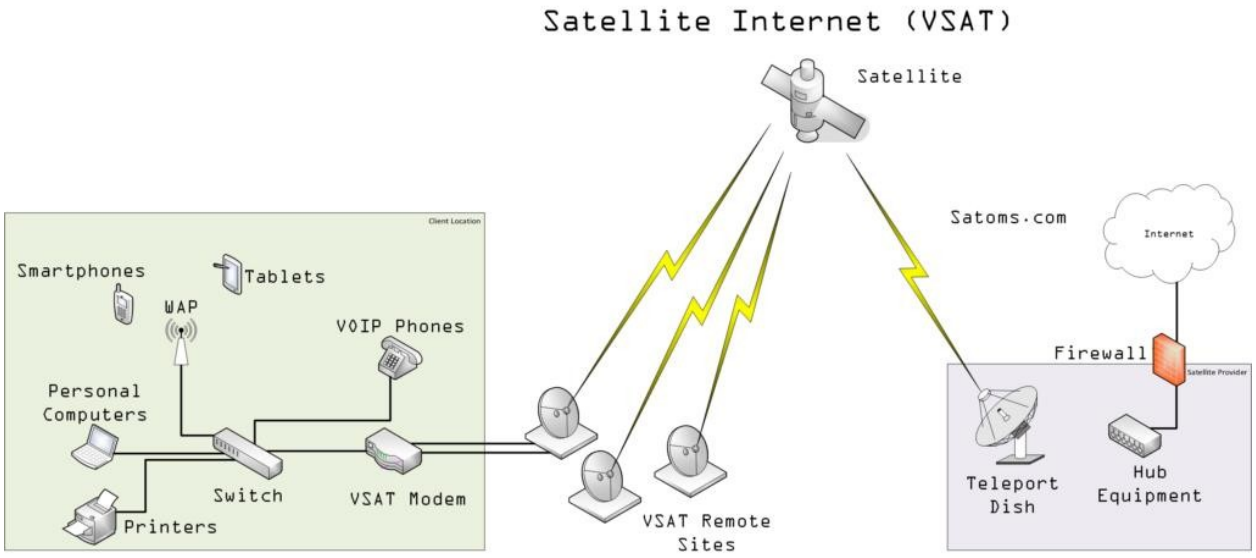
It is a digital network to transmit voice, image, video, and text over the existing circuit-switched PSTN telephone network.

Different types of devices, such as analog phone, a fax machine, or a computer, are connected to an ISDN terminal adaptor, which converts different types of data into ISDN format so that they can share the same digital PSTN network.

VSAT, RF Link

VSAT is an acronym for **Very Small Aperture Terminal,** and generally refers to a satellite dish of 3.8m or less located at the remote location. Satellite communication is used for internet connection in remote areas of the world where a connection to the traditional copper or fibre network is not possible. It is also used as a backup communication link for mission critical systems such as remote autonomous mine sites and offshore Oil and Gas rigs.
VSAT links are used for both internet data and voice (VoIP – Voice over IP) traffic.

There are many different types and sizes of satellite dish and equipment to choice depending on the frequency band, satellite orbit, bandwidth, network topology, etc.
Internet speed can vary between 128kbps or up to 72Mbps and above for luxury cruise liners.

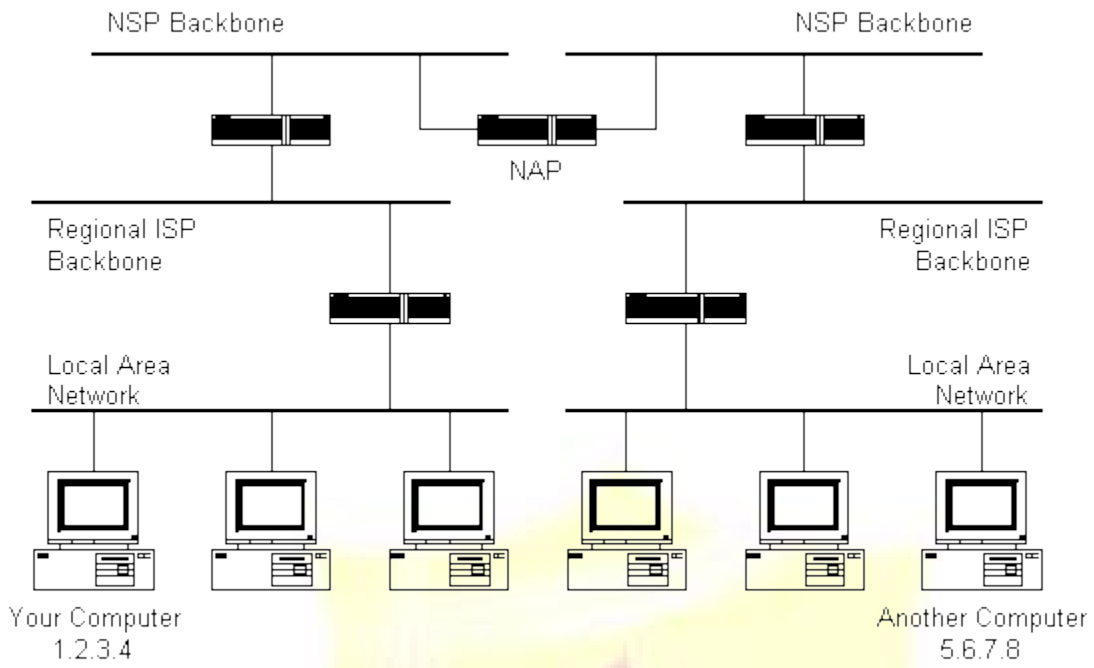


Working of Internet:

The Internet backbone is made up of many large networks which interconnect with each other. These large networks are known as **Network Service Providers** or **NSP**s.

E.g. UUNet, CerfNet, IBM, BBN Planet, SprintNet, PSINet

Each NSP is required to connect to **Network Access Points** or **NAP**s. At the NAPs, packet traffic may jump from one NSP's backbone to another NSP's backbone.



A router is usually connected between networks to route packets between them. Each router knows about it's sub-networks and which IP addresses they use. The black boxes connecting the backbones are routers.

When a packet arrives at a router, the router examines the IP address and checks it's routing table. If the network containing the IP address is found, the packet is sent to that network.

If the network containing the IP address is not found, then the router sends the packet on a default route, usually up the backbone hierarchy to the next router. Hopefully the next router will know where to send the packet. If it does not, again the packet is routed upwards until it reaches a NSP backbone.

The routers connected to the NSP backbones hold the largest routing tables and here the packet will be routed to the correct backbone, where it will begin its journey 'downward' through smaller and smaller networks until it finds it's destination.

The DNS is a distributed database which keeps track of computer's names and their corresponding IP addresses on the Internet.

Many computers connected to the Internet host part of the DNS database and the software that allows others to access it. These computers are known as DNS servers.

When an Internet connection is setup (e.g. for a LAN or Dial-Up Networking in Windows), one primary and one or more secondary DNS servers are usually specified as part of the installation. This way, any Internet applications that need domain name resolution will be able to function correctly. For example, when you enter a web address into your web browser, the browser first connects to your primary DNS server. After obtaining the IP address for the domain name you entered, the browser then connects to the target computer and requests the web page you wanted.

Introduction to WWW

World Wide Web, which is also known as a Web, is a collection of websites or web pages stored in web servers and connected to local computers through the internet.

These websites contain text pages, digital images, audios, videos, etc. Users can access the content of these sites from any part of the world over the internet using their devices such as computers, laptops, cell phones, etc. The WWW, along with internet, enables the retrieval and display of text and media to your device.

The building blocks of the Web are web pages which are formatted in HTML and connected by links called "hypertext" or hyperlinks and accessed by HTTP. These links are electronic connections that link related pieces of information so that users can access the desired information quickly. Hypertext offers the advantage to select a word or phrase from text and thus to access other pages that provide additional information related to that word or phrase.

A web page is given an online address called a Uniform Resource Locator (URL). A particular collection of web pages that belong to a specific URL is called a website, e.g., *www.facebook.com*, *www.google.com*, etc. So, the World Wide Web is like a huge electronic book whose pages are stored on multiple servers across the world.

Internet is entirely different from WWW. It is a worldwide network of devices like computers, laptops, tablets, etc. It enables users to send emails to other users and chat with them online. For example, when we send an email or chatting with someone online, we are using the internet. But, when we have opened a website like google.com for information, you are using the World Wide Web; a network of servers over the internet. You request a webpage from your computer using a browser, and the server renders that page to your browser.

Application Level Protocol

HTTP:

One of the most commonly used services on the Internet is the World Wide Web (WWW). The application protocol that makes the web work is **Hypertext Transfer Protocol** or **HTTP**.

HTTP is the protocol that web browsers and web servers use to communicate with each other over the Internet. It is an application level protocol because it sits on top of the TCP layer in the protocol stack and is used by specific applications to talk to one another. In this case the applications are web browsers and web servers.

HTTP is a connectionless text based protocol. Clients (web browsers) send requests to web servers for web elements such as web pages and images. After the request is serviced by a server, the connection between client and server across the Internet is disconnected. A new connection must be made for each request.

When we type a URL into a web browser, this is what happens:

1. If the URL contains a domain name, the browser first connects to a domain name server and retrieves the corresponding IP address for the web server.
2. The web browser connects to the web server and sends an HTTP request (via the protocol stack) for the desired web page.
3. The web server receives the request and checks for the desired page. If the page exists, the web server sends it. If the server cannot find the requested page, it will send an HTTP 404 error message. (404 means 'Page Not Found' as anyone who has surfed the web probably knows.)
4. The web browser receives the page back and the connection is closed.
5. The browser then parses through the page and looks for other page elements it needs to complete the web page. These usually include images, applets, etc.
6. For each element needed, the browser makes additional connections and HTTP requests to the server for each element.
7. When the browser has finished loading all images, applets, etc. the page will be completely loaded in the browser window.

SMTP:

E-mail uses an application level protocol called **Simple Mail Transfer Protocol** or **SMTP**. SMTP is also a text based protocol, but unlike HTTP, SMTP is connection oriented. SMTP is also more complicated than HTTP.

When we open our mail client to read our e-mail, this is what typically happens:

1. The mail client (Netscape Mail, Lotus Notes, Microsoft Outlook, etc.) opens a connection to it's default mail server. The mail server's IP address or domain name is typically setup when the mail client is installed.
2. The mail server will always transmit the first message to identify itself.
3. The client will send an SMTP HELO command to which the server will respond with a 250 OK message.
4. Depending on whether the client is checking mail, sending mail, etc. the appropriate SMTP commands will be sent to the server, which will respond accordingly.
5. This request/response transaction will continue until the client sends an SMTP QUIT command. The server will then say goodbye and the connection will be closed.

Web Browser

A **web browser** (commonly referred to as a **browser**) is application software for accessing the World Wide Web. When a user requests a web page from a particular website, the web browser retrieves the necessary content from a web server and then displays the page on the user's device.

A web browser is not the same thing as a search engine, though the two are often confused. A search engine is a website that provides links to other websites. However, to connect to a website's server and display its web pages, a user must have a web browser installed.

URL

A **Uniform Resource Locator** (**URL**), often termed a **web address**, is a reference to a web resource that specifies its location on a computer network and a mechanism for retrieving it.

URLs occur most commonly to reference web pages (http) but are also used for file transfer (ftp), email (mailto), database access (JDBC), and many other applications.

Most web browsers display the URL of a web page above the page in an address bar. A typical URL could have the form `http://www.example.com/index.html`, which indicates a protocol (`http`), a hostname (`www.example.com`), and a file name (`index.html`).

Hyper text:

Hypertext is text that links to other information. By clicking on a link in a hypertext document, a user can quickly jump to different content.

Hypertext documents are interconnected by hyperlinks, which are typically activated by a mouse click, keypress set, or screen touch.

Apart from text, the term "hypertext" is also sometimes used to describe tables, images, and other presentational content formats with integrated hyperlinks.

Hypertext is one of the key underlying concepts of the World Wide Web, where Web pages are often written in the Hypertext Markup Language (HTML).

Hyperlinks:

In computing, a **hyperlink**, or simply a **link**, is a reference to data that the user can follow by clicking or tapping.

A hyperlink points to a whole document or to a specific element within a document. Hypertext is text with hyperlinks. The text that is linked from is called anchor text.

A software system that is used for viewing and creating hypertext is a *hypertext system*, and to create a hyperlink is *to hyperlink* (or simply *to link*). A user following hyperlinks is said to *navigate* or *browse* the hypertext.

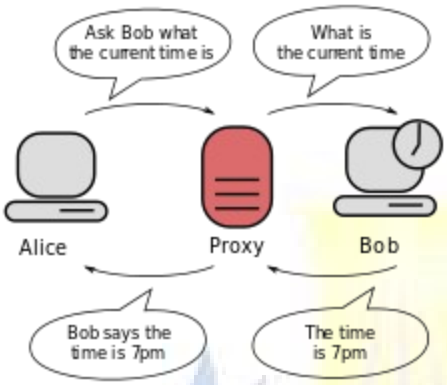
Hypermedia:

Hypermedia, an extension of the term hypertext, that includes graphics, audio, video, plain text and hyperlinks. This designation contrasts with the broader term *multimedia*, which may include non-interactive linear presentations as well as hypermedia. The World Wide Web is a classic example of hypermedia to access web content.

Search Engine:

A **search engine** is a software system that is designed to carry out web searches. They search the World Wide Web in a systematic way for particular information specified in a textual web search query.

Proxy Server:



In computer networking, a **proxy server** is a server application that acts as an intermediary between a client requesting a resource and the server providing that resource.

Instead of connecting directly to a server, the client directs the request to the proxy server, which evaluates the request and performs the required network transactions. This serves as a method to simplify or control the complexity of the request, or provide additional benefits such as load balancing, privacy, or security.

A proxy server functions on behalf of the client when requesting service, potentially masking the true origin of the request to the resource server.

CGI

The **Common Gateway Interface (CGI)** provides the middleware between WWW servers and external databases and information sources.

When a Web user submits a Web form on a web page that uses CGI. The form's data is sent to the Web server within an HTTP request with a URL denoting a CGI script. The Web server then launches the CGI script in a new computer process, passing the form data to it. The output of the CGI script, usually in the form of HTML, is returned by the script to the Web server, and the server relays it back to the browser as its response to the browser's request.

This process or convention for passing data back and forth between the server and the application is called the common gateway interface (CGI).

Advantages of CGI:

- The advanced tasks are currently a lot easier to perform in CGI than in Java.
- It is always easier to use the code already written than to write your own.
- CGI specifies that the programs can be written in any language, and on any platform, as long as they conform to the specification.
- CGI-based counters and CGI code to perform simple tasks are available in plenty.

Disadvantages of CGI:

There are some disadvantages of CGI which are given below:

- In Common Gateway Interface each page load incurs overhead by having to load the programs into memory.
- Generally, data cannot be easily cached in memory between page loads.
- There is a huge existing code base, much of it in Perl.
- CGI uses up a lot of processing time.

URI

A **Uniform Resource Identifier** (**URI**) is a unique sequence of characters that identifies a logical or physical resource used by web technologies.

URIs may be used to identify anything, including real-world objects, such as people and places, concepts, or information resources such as web pages and books.

Some URIs provide a means of locating and retrieving information resources on a network (either on the Internet or on another private network, such as a computer filesystem or an Intranet); these are Uniform Resource Locators (URLs). A URL provides the location of the resource. A URI identifies the resource by name at the specified location or URL.

Dreamweaver

Dreamweaver is an application that lets us design, code, and manage websites.

Its a piece of software that combines different tools to make web design and development easier.

This is very useful because it allows us to build the skeleton of a website quickly and without the need to write a single line of code.

HTML

Planning for designing web pages:

Breaking Up Your Content into Main Topics

With your goals in mind, try to organize your content into main topics or sections, chunking related information together under a single topic.

Ideas for Organization and Navigation

At this point, you should have a good idea of what you want to talk about as well as a list of topics. The next step is to actually start structuring the information you have into a set of web pages. Before you do that, however, consider some standard structures that have been used in other help systems and online tools. This section describes some of these structures, their various features, some important considerations, including the following

Model and Structure of a Web site:

You need to know what the following terms mean and how they apply to the body of work you're developing for the Web:

Website: A collection of one or more web pages linked together in a meaningful way that, as a whole, describes a body of information or creates an overall effect.

Web server: A computer on the Internet or an intranet that delivers Web pages and other files in response to browser requests.

Web page: A single document on a website, usually consisting of an HTML document and any items that are displayed within that document such as inline images.

Home page: The entry page for a website, which can link to additional pages on the same website or pages on other sites.

Developing websites:

Designing a website, like designing a book outline, a building plan, or a painting, can sometimes be a complex and involved process. Having a plan before you begin can help you keep the details straight and help you develop the finished product with fewer false starts. Today, you learned how to put together a simple plan and structure for creating a set of web pages, including the following:

• Deciding what sort of content to present

• Coming up with a set of goals for that content

• Deciding on a set of topics

• Organizing and storyboarding the website

Basic HTML: HTML stands for Hypertext Markup Language. The idea here is that most documents have common elements for example, titles, paragraphs, and lists. Before you start writing, therefore, you can identify and define the set of elements in that document and give them appropriate names.

How Markup Works

HTML is a markup language. Writing in a markup language means that you start with the text of your page and add special tags around words and paragraphs. The tags indicate the different parts of the page and produce different effects in the browser. HTML has a defined set of tags you can use. You can't make up your own tags to create new styles or features.

What HTML Files Look Like

Pages written in HTML are plain text files (ASCII), which means that they contain no

platform- or program-specific information. Any editor that supports text can read them. HTML files contain the following:

• The text of the page itself

• HTML tags that indicate page elements, structure, formatting, and hypertext links to other pages or to included media. Most HTML tags look something like the following:

<thetagname>affected text</thetagname>

The tag name itself (here, thetagname) is enclosed in brackets (< >). HTML tags generally have a beginning and an ending tag surrounding the text they affect. The beginning tag "turns on" a feature (such as headings, bold, and so on), and the ending tag turns it off. Closing tags have the tag name preceded by a slash (/). The opening tag (for example, for paragraphs) and closing tag (for example, for paragraphs) compose what is officially called an HTML element.

Text Formatting and HTML

When an HTML page is parsed by a browser, any formatting you might have done by hand that is, any extra spaces, tabs, returns, and so on is ignored. The only thing that specifies formatting in an HTML page is an HTML tag. If you spend hours carefully editing a plain text file to have nicely formatted paragraphs and columns of numbers but don't include any tags, when a web browser loads the page, all the text will flow into one paragraph. All your work will have been in vain.

The advantage of having all white space (spaces, tabs, returns) ignored is that you can put your tags wherever you want. The following examples all produce the same output. Try them!

<h1>If music be the food of love, play on.</h1>

<h1>
If music be the food of love, play on.
</h1>

<h1>
If music be the food of love, play on. </h1>

<h1> If music be the food of love,
play on. </h1 >

Structuring Your HTML

The DOCTYPE Identifier

Although it's not a page structure tag, the XHTML 1.0 recommendation includes one additional requirement for your web pages. The first line of each page must include a DOCTYPE identifier that defines the XHTML 1.0 version to which your page conforms, and the document type definition (DTD) that defines the specification. This is followed by the <html>, <head>, and <body> tags. In the following example, the XHTML 1.0 Strict document type appears before the page structure tags:

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/strict.dtd">
<html>
<head>
<title>Page Title</title>
</head>
<body>
...your page content...
</body>
</html>

Three types of HTML 4.01 document types are specified in the XHTML 1.0 specification:

Strict, Transitional, and Frameset.

The <html> Tag

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/transitional.dtd">
<html>
...your page...
</html>

Attributes:

1- <html lang=”en-us”>

HTML comment tag

Comments are not displayed by the browser, but they can help document your HTML source code.

<!- - This is a comment- ->

The <head> Tag

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"

"http://www.w3.org/TR/xhtml1/DTD/transitional.dtd">

<html>

<head>

<title>This is the Title. It will be explained later on</title>

</head>

...your page...

</html>

The <body> Tag

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"

"http://www.w3.org/TR/xhtml1/DTD/transitional.dtd">

<html>

<head>

<title>This is the Title. It will be explained later on</title>

</head>

<body>

...your page...

</body>

</html>

The Title

Each HTML page needs a title to indicate what the page describes. It appears in the title bar of the browser when people view the web page.

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"

"http://www.w3.org/TR/xhtml1/DTD/transitional.dtd">

<html>

<head>

<title>The Lion, The Witch, and the Wardrobe</title>

</head>

<body>

...your page...

</body>

</html>

Headings

Headings are used to add titles to sections of a page. Search engines use the headings to index the structure and content of your web pages.

It is important to use headings to show the document structure.

<h1> headings should be used for main headings, followed by <h2> headings, then the less important <h3>, and so on.

HTML defines six levels of headings.

```
<h1 style="font-size:60px;">Heading 1</h1>
```

<h1 style="background-color:Tomato;">Tomato</h1>

```
<h1 style="border:2px solid Tomato;">Hello World</h1>
```

Heading tags look like the following:

<h1>Movies</h1>

<h2>Action/Adventure</h2>

<h3>Caper</h3>

<h3>Sports</h3>

<h3>Thriller</h3>

<h3>War</h3>

<h2>Comedy</h2>

<h3>Romantic Comedy</h3>

<h3>Slapstick</h3>

<h2>Drama</h2>

<h3>Buddy Movies</h3>

<h3>Mystery</h3>

<h3>Romance</h3>

<h2>Horror</h2>

Horizontal Rule

The <hr> tag defines a thematic break in an HTML page, and is most often displayed as a horizontal rule.

Paragraphs

As of the HTML 4.01 standard, paragraph tags are two-sided (...), and

indicates the beginning of the paragraph. The closing tag is no longer optional, so rather than using to indicate where one paragraph ends and another begins, you enclose each paragraph within a tag.

Input

The dragon fell to the ground, releasing an anguished cry and seething in pain. The thrust of Enigern's sword proved fatal as the dragon breathed its last breath. Now Enigern was free to release Lady Aelfleada from her imprisonment in the dragon's lair.

Attributes:

1-

2- `<p title="I'm a tooltip">This is a paragraph.</p>` (The value of the title attribute will be displayed as a tooltip when you mouse over the element)

3- `<p style="font-family:courier;">This is a paragraph.</p>`

Formatting:

<b> The HTML <b> element defines bold text, without any extra importance.

<i> The content inside is typically displayed in italic.

****The content inside is typically displayed in bold. Defines text with strong importance.

-The content inside is typically displayed in italic.

<mark>: The HTML <mark> element defines text that should be marked or highlighted

<del>: he HTML <del> element defines text that has been deleted from a document. Browsers will usually strike a line through deleted text

<sub> The HTML <sub> element defines subscript text

```
<p>This is <sub>subscripted</sub> text.</p>
```

<sup> The HTML <sup> element defines superscript text.

```
<p>This is <sup>superscripted</sup> text.</p>
```

<abbr>: The HTML <abbr> tag defines an abbreviation or an acronym. The title attribute show the description for the abbreviation/acronym when you mouse over the element.

```
<p>The <abbr title="World Health Organization">WHO</abbr> was founded in 1948.</p>
```

**Line Break
**

The HTML <br> element defines a line break.

```
<p>This is<br>a paragraph<br>with line breaks.</p>
```

Preformatted Text <pre>

The HTML `<pre>` element defines preformatted text.

The text inside a `<pre>` element is displayed in a fixed-width font (usually Courier), and it preserves both spaces and line breaks:

```
<pre>
  My Bonnie lies over the ocean.

  My Bonnie lies over the sea.

  My Bonnie lies over the ocean.

  Oh, bring back my Bonnie to me.
</pre>
```

Image:

Images displayed on the Web should be converted to one of the formats supported by most browsers: GIF, JPEG, or PNG. GIF and JPEG are the popular standards, and every graphical browser supports them. PNG is a newer image format that was created in response to some patent issue s with the GIF format.

The most important attribute of the <img> tag is src, which is the URL of the image you want to include. Paths to images are derived in the same way as the paths in the href attribute of links. So, to point to a GIF file named image.gif in the same directory as the HTML document, you can use the following HTML tag:

<img src="image.gif">

The alt attribute of <img> provides an alternate text for an image

```
<img src="pic_trulli.jpg" alt="Italian Trulli">
<img src="img_girl.jpg" alt="Girl in a jacket" width="500" height="600">
```

Input:

```
<p><img src="house.jpg" alt="House of Terror" /></p>
<h1>Welcome to The Halloween House of Terror!!</h1>
```

Links/ Anchor tag:

To create a link in an HTML page, we use the HTML link tag <a>.... The <a> tag often is called an anchor tag because it also can be used to create anchors for links.

Input

Go back to

Main Menu

By default, links will appear as follows in all browsers:

- An unvisited link is underlined and blue
- A visited link is underlined and purple
- An active link is underlined and red

The target attribute of <a> specifies where to open the linked document.

The target attribute can have one of the following values:

- _self - Default. Opens the document in the same window/tab as it was clicked
- _blank - Opens the document in a new window or tab
- _parent - Opens the document in the parent frame
- _top - Opens the document in the full body of the window

```
<a href="https://www.w3schools.com/" target="_blank">Visit W3Schools!</a>
```

Absolute Vs Relative URL

Absolute: Use a full URL to link to a web page.

Relative: Link to a page located in the html folder on the current web site

```
<h2>Absolute URLs</h2>
<p><a href="https://www.w3.org/">W3C</a></p>
<p><a href="https://www.google.com/">Google</a></p>

<h2>Relative URLs</h2>
<p><a href="html_images.asp">HTML Images</a></p>
<p><a href="/css/default.asp">CSS Tutorial</a></p>
```

```
Link to a mail
<a href="mailto:someone@example.com">Send email</a>
```

Creating Bookmark:

Bookmarks can be useful if a web page is very long.

To create a bookmark - first create the bookmark, then add a link to it.

```
<h2 id="C4">Chapter 4</h2>

<a href="#C4">Jump to Chapter 4</a>

Or

<a href="html_demo.html#C4">Jump to Chapter 4</a>
```

When the link is clicked, the page will scroll down or up to the location with the bookmark.

Lists:

HTML 4.01 defines these three types of lists:

• Numbered or ordered lists, which are typically labeled with numbers

• Bulleted or unordered lists, which are typically labeled with bullets or some other symbol

• Glossary lists, in which each item in the list has a term and a definition for that term, arranged so that the term is somehow highlighted or drawn out from the text

List Tags

All the list tags have the following common elements:

• The entire list is surrounded by the appropriate opening and closing tag for the type of list (for example, <ul> and for unordered lists, or <ol> and for ordered lists).

• Each list item within the list has its own tag:

<dt> and <dd> for the glossary lists, and <li> for all the other lists.

Input

• Installing Your New Operating System

•

• Insert the CD-ROM into your CD-ROM drive.

• Choose RUN.

• Enter the drive letter of your CD-ROM (example: D:\), followed by SETUP.EXE.

• Follow the prompts in the setup program.

• Reboot your computer after all files are installed.

• Cross your fingers.

•

Customizing Ordered Lists

You can customize ordered lists in two main ways: how they're numbered and the number with which the list starts. HTML 3.2 provides the type attribute that can take one of five values to de fine which type of numbering to use on the list:

• "1" Specifies that standard Arabic numerals should be used to number the list (that is, 1, 2, 3, 4, and so on)

• "a" Specifies that lowercase letters should be used to number the list (that is, a, b, c, d, and so on)

• "A" Specifies that uppercase letters should be used to number the list (that is, A, B, C, D, and so on)

• "i" Specifies that lowercase Roman numerals should be used to number the list (that is, i, ii, iii, iv, and so on)

• "I" Specifies that uppercase Roman numerals should be used to number the list (that is, I, II, III, IV, and so on)

You can specify types of numbering in the <ol> tag, as follows: <ol type="a">. By default type="1" is assumed.

Input

The Days of the Week in French:

<ol type="I">

Lundi

Mardi

Mercredi

Jeudi

Vendredi

Samedi

Dimanche

Input

The Last Six Months of the Year (and the Beginning of the NextYear):

```
<ol type="I" start="7">
<li>July</li>
<li>August</li>
<li>September</li>
<li>October</li>
<li>November</li>
<li>December</li>
<li type="1">January</li>
</ol>
```

Tables:

Table Parts

Before getting into the actual HTML code to create a table, let's look at the following terms so that we both know what we're talking about:

• The caption indicates what the table is about: for example, "Voting Statistics" or "Toy Distribution Per Room" Captions are optional.

• The table headings label the rows, columns, or both. Usually they're in an emphasized font that's different from the rest of the table. They're optional.

• Table cells are the individual squares in the table. A cell can contain normal table data or a table heading.

• Table data is the values in the table itself. The combination of the table headings and table data makes up the sum of the table.

The <table> Element

The to create a table in HTML, you use <table>...</table> element to enclose the code for an optional caption, and then add the contents of the table itself:

```
<table>
...table caption (optional) and contents...
</table>
```

Rows and Cells

The cells within each row are created using one of two elements:

• <th>...</th> elements are used for heading cells. Generally, browsers center the contents of a <th> cell and render any text in the cell in boldface.

• <td>...</td> elements are used for data cells. TD stands for table data.

Input

```
<tr>
<th>Name</th>
<td>Alison</td>
<td>Tom</td>
<td>Susan</td>
</tr>
<tr>
<th>Height</th>
<td>5'4"</td>
<td>6'0"</td>
<td>5'1"</td>
</tr>
<tr>
<th>Weight</th>
<td>140</td>
<td>165</td>
<td>97</td>
</tr>
<tr>
<th>Eye Color</th>
<td>Blue</td>
<td>Blue</td>
<td>Brown</td>
</tr>
```

Setting Table Widths

To make a table as wide as the browser window, you add the width attribute to the table, as shown in the following line of code:

```
<table border="1" width="100%">
```

Changing Table Borders

You can change the width of the border drawn around the table. If border has a numeric value, the border around the outside of the table is drawn with that pixel width. The

default is border="1". border="0" suppresses the border, just as if you had omitted the border attribute altogether.

Input

<table border="10" width="100%">

Cell Padding

The cell padding attribute defines the amount of space between the edges of the cells and the content inside a cell.

Input

<table cellpadding="10" border="1">

Cell Spacing

Cell spacing is similar to cell padding except that it affects the amount of space between cells that is, the width of the space between the inner and outer lines that make up the table border.

Input

<table cellpadding="10" border="4" cellspacing="8">

Spanning Multiple Rows or Columns

The tables you've created up to this point all had one value per cell or the occasional empty cell. You also can create cells that span multiple rows or columns within the table. Those spanned cells then can hold headings that have subheadings in the next row or column or you can create other special effects within the table layout.

Input

```
<html>
<head>
<title>Row and Column Spans</title>
</head>
<body>
<table border="1" summary="span example">
<tr>
<th colspan="2">Gender</th>
</tr>
<tr>
<th>Male</th>
```

<th>Female</th>
</tr>
<tr>
<td>15</td>
<td>23</td>
</tr>
</table>
</body>
</html>

Forms:

Using the <form> Tag

To accept input from a user, you must wrap all of your input fields inside a <form> tag. The purpose of the <form> tag is to indicate where and how the user's input should be sent. First, let's look at how the <form> tag affects page layout. Forms are block-level elements.

Input

Please enter your username <form><input /> and password <input /></form> to log in.

The two most commonly used attributes of the <form> tag are action and method. Both of these attributes are optional. The following example shows how the <form> tag is typically used:

<form action="someaction" method="get or post">
content, form controls, and other HTML elements
</form>

action specifies the URL to which the form is submitted. Again, remember that for the form to be submitted successfully, the script must be in the exact location you specify and must work properly.

The method attribute supports two values: get or post. The method indicates how the form data should be packaged in the request that's sent back to the server. The get method appends the form data to the URL in the request.

Creating Text Controls

Text controls enable you to gather information from a user in small quantities. This

control type creates a single-line text input field in which users can type information, such as their name or a search term.

Input

Enter your pet's name:

<input type="text" name="petname" /></ p>

Creating Password Controls

The password and text field types are identical in every way except that the data entered in a password field is masked so that someone looking over the shoulder of the person entering information can't see the value that was typed into the field.

Input

Enter your password: <input type="password" name="userpassword" size="8" maxlength="8" />

Creating Submit Buttons

Submit buttons are used to indicate that the user is finished filling out the form. Setting the type attribute of the form to submit places a submit button on the page with the default label determined by the browser, usually Submit Query. To change the button text, use the value attribute and enter your own label, as follows:

<input type="submit" value="Send Form Data" />

Creating Reset Buttons

Reset buttons set all the form controls to their default values. These are the values included in the value attributes of each field in the form (or in the case of selectable fields, the values that are preselected). As with the Submit button, you can change the label of a Reset button to one of your own choosing by using the value attribute, like this:

<input type="reset" value="Clear Form" />

Creating Check Box Controls

Check boxes are fields that can be set to two states: on and off. To create a check box, set the input tag's type attribute to checkbox. The name attribute is also required, as shown in the following example:

Input

Check to receive SPAM email <input type="checkbox" name="spam" />

Creating Radio Buttons

Radio buttons, which generally appear in groups, are designed so that when one button in

the group is selected, the other buttons in the group are automatically unselected. They enable you to provide users with a list of options from which only one option can be selected. To create a radio button, set the type attribute of an <input> tag to radio. To create a radio button group, set the name attributes of all the fields in the group to the same value. To cre ate a radio button group with three options, the following code is used:

Input

```
<p>Select a color:<br />
<input type="radio" name="color" value="red" /> Red<br />
<input type="radio" name="color" value="blue" /> Blue<br />
<input type="radio" name="color" value="green" /> Green<br />
</p>
```

Creating Menus with select and option

The select element creates a menu that can be configured to enable users to select one or more options from a pull-down menu or a scrollable menu that shows several options at once. The <select> tag defines how the menu will be displayed and the name of the parameter associated with the field. The <option> tag is used to add selections to the menu. The default appearance of select lists is to display a pull-down list that enables the user to select one of the options. Here's an example of how one is created:

Input

```
<p>Please pick a travel destination:
<select name="location">
<option>Indiana</option>
<option>Fuji</option>
<option>Timbuktu</option >
<option>Alaska</option>
</select>
</p>
```

Frames for designing a good website:

The first HTML document you need to create is called the frameset document. In this document, you define the layout of your frames, and the locations of the documents to be initially loaded in each frame. Each of the three HTML documents other than the frameset document, the ones that load in the frames, contain normal HTML tags that

define the contents of each separate frame area. These documents are referenced by the frameset document.

The <frameset> Tag

To create a frameset document, you begin with the <frameset> tag. When used in an HTML document, the <frameset> tag replaces the <body> tag, as shown in the following code:

```
<html>
<head>
<title>Page Title</title>
</head>
<frameset>
.. your frameset goes here ...
</frameset>
</html>
```

It's important that you understand up front how a frameset document differs from a normal HTML document. If you include a <frameset> tag in an HTML document, you cannot include a <body> tag also.

The cols Attribute

When you define a <frameset> tag, you must include one of two attributes as part of the tag definition. The first of these attributes is the cols attribute, which takes the following form:

```
<frameset cols="column width, column width, ...">
```

Input

```
<html>
<head>
<title>Three Columns</title>
</head>
<frameset cols="100,50%,*">
<frame src="leftcol.html">
<frame src="midcol.html">
<frame src="rightcol.html">
</frameset>
```

</html>

The rows Attribute

The rows attribute works the same as the cols attribute, except that it splits the screen into horizontal frames rather than vertical ones. To split the screen into two frames of equal height, you would write the following:

Input

```
<html>
<head>
<title>Two Rows</title>
</head>
<frameset rows="50%,50%">
<frame src="toprow.html">
<frame src="botrow.html">
</frameset>
</html>
```

The <frame> Tag

After you have your basic frameset laid out, you need to associate an HTML document with each frame by using the <frame> tag, which takes the following form:

```
<frame src="document URL">
```

For each frame defined in the <frameset> tag, you must include a corresponding <frame> tag, as shown in the following:

Input

```
<html>
<head>
<title>The FRAME Tag</title>
</head>
<frameset rows="*,*,*">
<frame src="document1.html" />
<frame src="document2.html" />
<frame src="document3.html" />
</frameset>
</html>
```

Changing Frame Borders

Start with the <frame> tag. By using two attributes, bordercolor and frameborder, you can turn borders on and off and specify their color. You can assign bordercolor any valid color value, either as a name or a hexadecimal triplet. frameborder takes two possible values:

1(to display borders) or 0 (to turn off the display of borders).

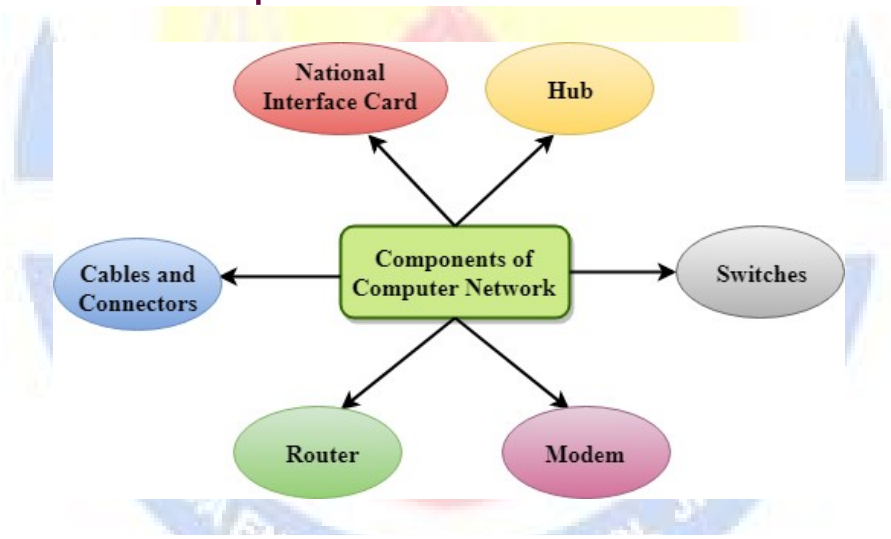
```
<html>
<head>
<title>Conflicting Borders</title>
</head>
<frameset frameborder="0" rows="*,*,*">
<frame frameborder="1" bordercolor="yellow" src="document1.html">
<frame bordercolor="#cc3333" src="document2.html">
<frame src="document3.html">
</frameset>
</html>
```

UNIT – 1 INTERNET BASICS

Computer Network

- **Computer Network** is a group of computers connected with each other through wires, optical fibres or optical links so that various devices can interact with each other through a network.
- The aim of the computer network is the sharing of resources among various devices.
- In the case of computer network technology, there are several types of networks that vary from simple to complex level.

Components of Computer Network:

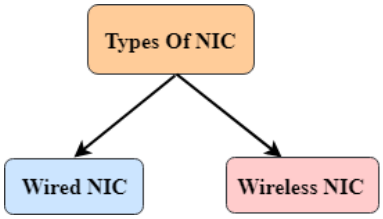


Major components of a computer network are:

NIC

- NIC stands for network interface card.
- NIC is a hardware component used to connect a computer with another computer onto a network
- It can support a transfer rate of 10,100 to 1000 Mb/s.
- The MAC address or physical address is encoded on the network card chip which is assigned by the IEEE to identify a network card uniquely. The MAC address is stored in the PROM (Programmable read-only memory).

There are two types of NIC:



1. Wired NIC
2. Wireless NIC

Wired NIC: The Wired NIC is present inside the motherboard. Cables and connectors are used with wired NIC to transfer data.

Wireless NIC: The wireless NIC contains the antenna to obtain the connection over the wireless network. For example, laptop computer contains the wireless NIC.

Hub

Hub is a central device that splits the network connection into multiple devices. When computer requests for information from a computer, it sends the request to the Hub. Hub distributes this request to all the interconnected computers.

Switches

Switch is a networking device that groups all the devices over the network to transfer the data to another device. A switch is better than Hub as it does not broadcast the message over the network, i.e., it sends the message to the device for which it belongs to. Therefore, we can say that switch sends the message directly from source to the destination.

Cables and connectors

Cable is a transmission media that transmits the communication signals. **There are three types of cables:**

- **Twisted pair cable:** It is a high-speed cable that transmits the data over **1Gbps** or more.
- **Coaxial cable:** Coaxial cable resembles like a TV installation cable. Coaxial cable is more expensive than twisted pair cable, but it provides the high data transmission speed.
- **Fibre optic cable:** Fibre optic cable is a high-speed cable that transmits the data using light beams. It provides high data transmission speed as compared to other cables. It is more expensive as compared to other cables, so it is installed at the government level.

Router

- A router is a hardware device which is used to connect a LAN with an internet connection. It is used to receive, analyze and forward the incoming packets to another network.
- A router works in a **Layer 3 (Network layer)** of the OSI Reference model.
- A router forwards the packet based on the information available in the routing table.
- It determines the best path from the available paths for the transmission of the packet.

Advantages of Router:

- **Security:** The information which is transmitted to the network will traverse the entire cable, but the only specified device which has been addressed can read the data.
- **Reliability:** If the server has stopped functioning, the network goes down, but no other networks are affected that are served by the router.
- **Performance:** Router enhances the overall performance of the network. Suppose there are 24 workstations in a network generates a same amount of traffic. This increases the traffic load on the network. Router splits the single network into two networks of 12 workstations each, reduces the traffic load by half.
- **Network range**

Modem

- A modem is a hardware device that allows the computer to connect to the internet over the existing telephone line.
- A modem is not integrated with the motherboard rather than it is installed on the PCI slot found on the motherboard.
- It stands for Modulator/Demodulator. It converts the digital data into an analog signal over the telephone lines.

Based on the differences in speed and transmission rate, a modem can be classified in the following categories:

- Standard PC modem or Dial-up modem
- Cellular Modem
- Cable modem

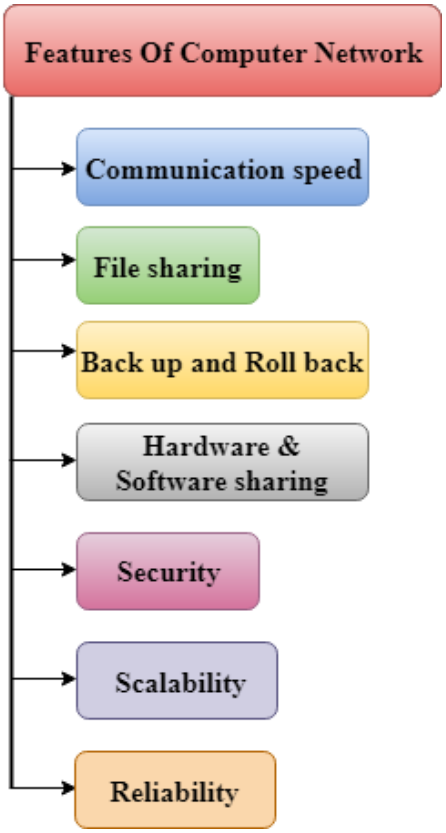
Uses of Computer Network

- **Resource sharing:** Resource sharing is the sharing of resources such as programs, printers, and data among the users on the network without the requirement of the physical location of the resource and user.
- **Server-Client model:** Computer networking is used in the **server-client model**. A server is a central computer used to store the information and maintained by the system administrator. Clients are the machines used to access the information stored in the server remotely.
- **Communication medium:** Computer network behaves as a communication medium among the users. For example, a company contains more than one computer has an email system which the employees use for daily communication.
- **E-commerce:** Computer network is also important in businesses. We can do the business over the internet. For example, amazon.com is doing their business over the internet, i.e., they are doing their business over the internet.

Features of Computer network

A list Of Computer network features is given below.

- Communication speed
- File sharing
- Back up and Roll back is easy
- Software and Hardware sharing
- Security
- Scalability
- Reliability



Communication speed

Network provides us to communicate over the network in a fast and efficient manner. For example, we can do video conferencing, email messaging, etc. over the internet. Therefore, the computer network is a great way to share our knowledge and ideas.

File sharing

File sharing is one of the major advantage of the computer network. Computer network provides us to share the files with each other.

Back up and Roll back is easy

Since the files are stored in the main server which is centrally located. Therefore, it is easy to take the back up from the main server.

Software and Hardware sharing

We can install the applications on the main server, therefore, the user can access the applications centrally. So, we do not need to install the software on every machine. Similarly, hardware can also be shared.

Security

Network allows the security by ensuring that the user has the right to access the certain files and applications.

Scalability

Scalability means that we can add the new components on the network. Network must be scalable so that we can extend the network by adding new devices. But, it decreases the speed of the connection and data of the transmission speed also decreases, this increases the chances of error occurring. This problem can be overcome by using the routing or switching devices.

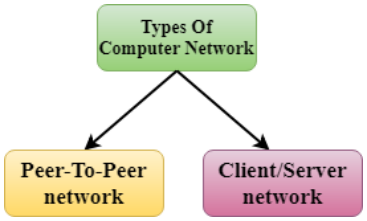
Reliability

Computer network can use the alternative source for the data communication in case of any hardware failure.

Computer Network Architecture

Computer Network Architecture is defined as the physical and logical design of the software, hardware, protocols, and media of the transmission of data. Simply we can say that how computers are organized and how tasks are allocated to the computer.

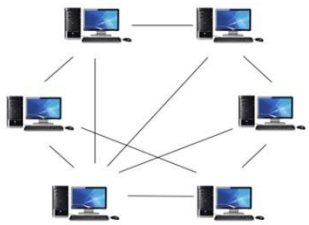
The two types of network architectures are used:



- Peer-To-Peer network
- Client/Server network

Peer-To-Peer network

- Peer-To-Peer network is a network in which all the computers are linked together with equal privilege and responsibilities for processing the data.
- Peer-To-Peer network is useful for small environments, usually up to 10 computers.
- Peer-To-Peer network has no dedicated server.
- Special permissions are assigned to each computer for sharing the resources, but this can lead to a problem if the computer with the resource is down.



Advantages of Peer-To-Peer Network:

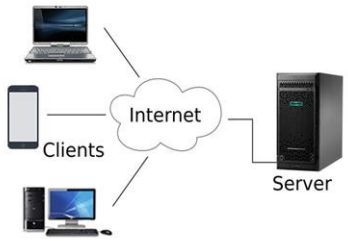
- It is less costly as it does not contain any dedicated server.
- If one computer stops working but, other computers will not stop working.
- It is easy to set up and maintain as each computer manages itself.

Disadvantages of Peer-To-Peer Network:

- In the case of Peer-To-Peer network, it does not contain the centralized system. Therefore, it cannot back up the data as the data is different in different locations.
- It has a security issue as the device is managed itself.

Client/Server Network

- Client/Server network is a network model designed for the end users called clients, to access the resources such as songs, video, etc. from a central computer known as Server.
- The central controller is known as a **server** while all other computers in the network are called **clients**.
- A server performs all the major operations such as security and network management.
- A server is responsible for managing all the resources such as files, directories, printer, etc.
- All the clients communicate with each other through a server. For example, if client1 wants to send some data to client 2, then it first sends the request to the server for the permission. The server sends the response to the client 1 to initiate its communication with the client 2.



Advantages of Client/Server network:

- A Client/Server network contains the centralized system. Therefore, we can back up the data easily.
- A Client/Server network has a dedicated server that improves the overall performance of the whole system.
- Security is better in Client/Server network as a single server administers the shared resources.
- It also increases the speed of the sharing resources.

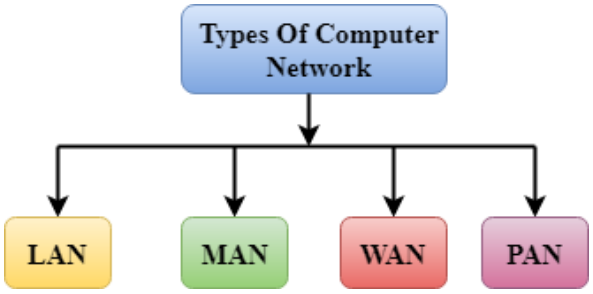
Disadvantages of Client/Server network:

- Client/Server network is expensive as it requires the server with large memory.
- A server has a Network Operating System(NOS) to provide the resources to the clients, but the cost of NOS is very high.
- It requires a dedicated network administrator to manage all the resources.

Computer Network Types

A computer network is a group of computers linked to each other that enables the computer to communicate with another computer and share their resources, data, and applications.

A computer network can be categorized by their size. A **computer network** is mainly of **four types**:



- LAN(Local Area Network)
- PAN(Personal Area Network)
- MAN(Metropolitan Area Network)
- WAN(Wide Area Network)

LAN(Local Area Network)

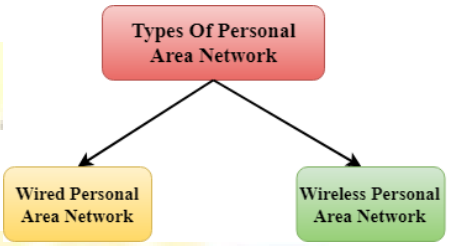
- Local Area Network is a group of computers connected to each other in a small area such as building, office.
- LAN is used for connecting two or more personal computers through a communication medium such as twisted pair, coaxial cable, etc.
- It is less costly as it is built with inexpensive hardware such as hubs, network adapters, and ethernet cables.
- The data is transferred at an extremely faster rate in Local Area Network.
- Local Area Network provides higher security.

PAN(Personal Area Network)

- Personal Area Network is a network arranged within an individual person, typically within a range of 10 meters.
- Personal Area Network is used for connecting the computer devices of personal use is known as Personal Area Network.

- **Thomas Zimmerman** was the first research scientist to bring the idea of the Personal Area Network.
- Personal Area Network covers an area of **30 feet**.
- Personal computer devices that are used to develop the personal area network are the laptop, mobile phones, media player and play stations.

There are two types of Personal Area Network:



- Wired Personal Area Network
- Wireless Personal Area Network

Wireless Personal Area Network: Wireless Personal Area Network is developed by simply using wireless technologies such as WiFi, Bluetooth. It is a low range network.

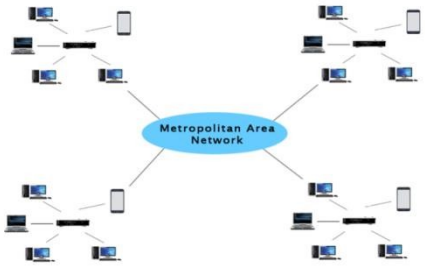
Wired Personal Area Network: Wired Personal Area Network is created by using the USB.

Examples of Personal Area Network:

- **Body Area Network:** Body Area Network is a network that moves with a person. **For example**, a mobile network moves with a person. Suppose a person establishes a network connection and then creates a connection with another device to share the information.
- **Offline Network:** An offline network can be created inside the home, so it is also known as a **home network**. A home network is designed to integrate the devices such as printers, computer, television but they are not connected to the internet.
- **Small Home Office:** It is used to connect a variety of devices to the internet and to a corporate network using a VPN

MAN(Metropolitan Area Network)

- A metropolitan area network is a network that covers a larger geographic area by interconnecting a different LAN to form a larger network.
- Government agencies use MAN to connect to the citizens and private industries.
- In MAN, various LANs are connected to each other through a telephone exchange line.



- The most widely used protocols in MAN are RS-232, Frame Relay, ATM, ISDN, OC-3, ADSL, etc.
- It has a higher range than Local Area Network(LAN).

Uses of Metropolitan Area Network:

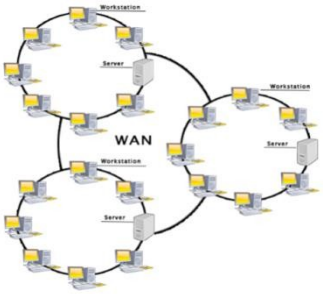
- MAN is used in communication between the banks in a city.
- It can be used in an Airline Reservation.
- It can be used in a college within a city.
- It can also be used for communication in the military.

WAN(Wide Area Network)

- A Wide Area Network is a network that extends over a large geographical area such as states or countries.
- A Wide Area Network is quite bigger network than the LAN.
- A Wide Area Network is not limited to a single location, but it spans over a large geographical area through a telephone line, fibre optic cable or satellite links.
- The internet is one of the biggest WAN in the world.
- A Wide Area Network is widely used in the field of Business, government, and education.

Examples of Wide Area Network:

- **Mobile Broadband:** A 4G network is widely used across a region or country.



- **Last mile:** A telecom company is used to provide the internet services to the customers in hundreds of cities by connecting their home with fiber.
- **Private network:** A bank provides a private network that connects the 44 offices. This network is made by using the telephone leased line provided by the telecom company.

Advantages of Wide Area Network:

Following are the advantages of the Wide Area Network:

- **Geographical area:** A Wide Area Network provides a large geographical area. Suppose if the branch of our office is in a different city then we can connect with them through WAN. The internet provides a leased line through which we can connect with another branch.
- **Centralized data:** In case of WAN network, data is centralized. Therefore, we do not need to buy the emails, files or back up servers.
- **Get updated files:** Software companies work on the live server. Therefore, the programmers get the updated files within seconds.
- **Exchange messages:** In a WAN network, messages are transmitted fast. The web application like Facebook, Whatsapp, Skype allows you to communicate with friends.
- **Sharing of software and resources:** In WAN network, we can share the software and other resources like a hard drive, RAM.
- **Global business:** We can do the business over the internet globally.
- **High bandwidth:** If we use the leased lines for our company then this gives the high bandwidth. The high bandwidth increases the data transfer rate which in turn increases the productivity of our company.

Disadvantages of Wide Area Network:

The following are the disadvantages of the Wide Area Network:

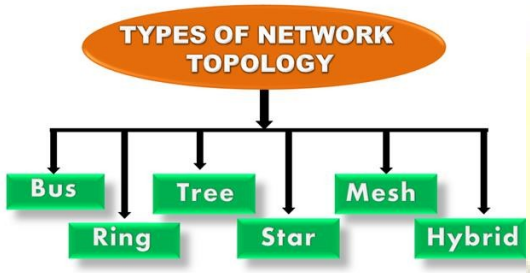
- **Security issue:** A WAN network has more security issues as compared to LAN and MAN network as all the technologies are combined together that creates the security problem.
- **Needs Firewall & antivirus software:** The data is transferred on the internet which can be changed or hacked by the hackers, so the firewall needs to be used. Some people can inject the virus in our system so antivirus is needed to protect from such a virus.

- **High Setup cost:** An installation cost of the WAN network is high as it involves the purchasing of routers, switches.
- **Troubleshooting problems:** It covers a large area so fixing the problem is difficult.

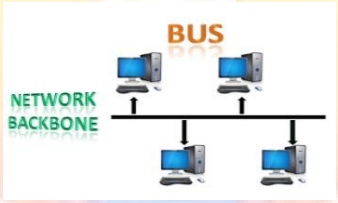
What is Topology?

Topology defines the structure of the network of how all the components are interconnected to each other. There are two types of topology: physical and logical topology.

Physical topology is the geometric representation of all the nodes in a network.



Bus Topology



- The bus topology is designed in such a way that all the stations are connected through a single cable known as a backbone cable.
- Each node is either connected to the backbone cable by drop cable or directly connected to the backbone cable.
- When a node wants to send a message over the network, it puts a message over the network. All the stations available in the network will receive the message whether it has been addressed or not.
- The bus topology is mainly used in 802.3 (ethernet) and 802.4 standard networks.
- The configuration of a bus topology is quite simpler as compared to other topologies.
- The backbone cable is considered as a **"single lane"** through which the message is broadcast to all the stations.

- The most common access method of the bus topologies is **CSMA** (Carrier Sense Multiple Access).

CSMA: It is a media access control used to control the data flow so that data integrity is maintained, i.e., the packets do not get lost. There are two alternative ways of handling the problems that occur when two nodes send the messages simultaneously.

- **CSMA CD:** CSMA CD (**Collision detection**) is an access method used to detect the collision. Once the collision is detected, the sender will stop transmitting the data. Therefore, it works on "**recovery after the collision**".
- **CSMA CA: CSMA CA (Collision Avoidance)** is an access method used to avoid the collision by checking whether the transmission media is busy or not. If busy, then the sender waits until the media becomes idle. This technique effectively reduces the possibility of the collision. It does not work on "recovery after the collision".

Advantages of Bus topology:

- **Low-cost cable:** In bus topology, nodes are directly connected to the cable without passing through a hub. Therefore, the initial cost of installation is low.
- **Moderate data speeds:** Coaxial or twisted pair cables are mainly used in bus-based networks that support upto 10 Mbps.
- **Familiar technology:** Bus topology is a familiar technology as the installation and troubleshooting techniques are well known, and hardware components are easily available.
- **Limited failure:** A failure in one node will not have any effect on other nodes.

Disadvantages of Bus topology:

- **Extensive cabling:** A bus topology is quite simpler, but still it requires a lot of cabling.
- **Difficult troubleshooting:** It requires specialized test equipment to determine the cable faults. If any fault occurs in the cable, then it would disrupt the communication for all the nodes.
- **Signal interference:** If two nodes send the messages simultaneously, then the signals of both the nodes collide with each other.
- **Reconfiguration difficult:** Adding new devices to the network would slow down the network.

- **Attenuation:** Attenuation is a loss of signal leads to communication issues. Repeaters are used to regenerate the signal.

Ring Topology

- Ring topology is like a bus topology, but with connected ends.
- The node that receives the message from the previous computer will retransmit to the next node.
- The data flows in one direction, i.e., it is unidirectional.
- The data flows in a single loop continuously known as an endless loop.
- It has no terminated ends, i.e., each node is connected to other node and having no termination point.
- The data in a ring topology flow in a clockwise direction.
- The most common access method of the ring topology is **token passing**.
    - **Token passing:** It is a network access method in which token is passed from one node to another node.
    - **Token:** It is a frame that circulates around the network.

Working of Token passing

- A token moves around the network, and it is passed from computer to computer until it reaches the destination.
- The sender modifies the token by putting the address along with the data.
- The data is passed from one device to another device until the destination address matches. Once the token received by the destination device, then it sends the acknowledgment to the sender.
- In a ring topology, a token is used as a carrier.

Advantages of Ring topology:

- **Network Management:** Faulty devices can be removed from the network without bringing the network down.
- **Product availability:** Many hardware and software tools for network operation and monitoring are available.
- **Cost:** Twisted pair cabling is inexpensive and easily available. Therefore, the installation cost is very low.
- **Reliable:** It is a more reliable network because the communication system is not dependent on the single host computer.

Disadvantages of Ring topology:

- **Difficult troubleshooting:** It requires specialized test equipment to determine the cable faults. If any fault occurs in the cable, then it would disrupt the communication for all the nodes.
- **Failure:** The breakdown in one station leads to the failure of the overall network.
- **Reconfiguration difficult:** Adding new devices to the network would slow down the network.
- **Delay:** Communication delay is directly proportional to the number of nodes. Adding new devices increases the communication delay.

Star Topology

- Star topology is an arrangement of the network in which every node is connected to the central hub, switch or a central computer.
- The central computer is known as a **server**, and the peripheral devices attached to the server are known as **clients**.
- Coaxial cable or RJ-45 cables are used to connect the computers.
- Hubs or Switches are mainly used as connection devices in a **physical star topology**.
- Star topology is the most popular topology in network implementation.

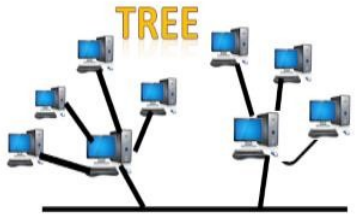
Advantages of Star topology

- **Efficient troubleshooting:** Troubleshooting is quite efficient in a star topology as compared to bus topology. In a bus topology, the manager has to inspect the kilometers of cable. In a star topology, all the stations are connected to the centralized network. Therefore, the network administrator has to go to the single station to troubleshoot the problem.
- **Network control:** Complex network control features can be easily implemented in the star topology. Any changes made in the star topology are automatically accommodated.
- **Limited failure:** As each station is connected to the central hub with its own cable, therefore failure in one cable will not affect the entire network.
- **Familiar technology:** Star topology is a familiar technology as its tools are cost-effective.
- **Easily expandable:** It is easily expandable as new stations can be added to the open ports on the hub.
- **Cost effective:** Star topology networks are cost-effective as it uses inexpensive coaxial cable.
- **High data speeds:** It supports a bandwidth of approx 100Mbps. Ethernet 100BaseT is one of the most popular Star topology networks.

Disadvantages of Star topology

- **A Central point of failure:** If the central hub or switch goes down, then all the connected nodes will not be able to communicate with each other.
- **Cable:** Sometimes cable routing becomes difficult when a significant amount of routing is required.

Tree topology



- Tree topology combines the characteristics of bus topology and star topology.
- A tree topology is a type of structure in which all the computers are connected with each other in hierarchical fashion.
- The top-most node in tree topology is known as a root node, and all other nodes are the descendants of the root node.
- There is only one path exists between two nodes for the data transmission. Thus, it forms a parent-child hierarchy.

Advantages of Tree topology

- **Support for broadband transmission:** Tree topology is mainly used to provide broadband transmission, i.e., signals are sent over long distances without being attenuated.
- **Easily expandable:** We can add the new device to the existing network. Therefore, we can say that tree topology is easily expandable.
- **Easily manageable:** In tree topology, the whole network is divided into segments known as star networks which can be easily managed and maintained.
- **Error detection:** Error detection and error correction are very easy in a tree topology.
- **Limited failure:** The breakdown in one station does not affect the entire network.
- **Point-to-point wiring:** It has point-to-point wiring for individual segments.

Disadvantages of Tree topology

- **Difficult troubleshooting:** If any fault occurs in the node, then it becomes difficult to troubleshoot the problem.
- **High cost:** Devices required for broadband transmission are very costly.
- **Failure:** A tree topology mainly relies on main bus cable and failure in main bus cable will damage the overall network.

- **Reconfiguration difficult:** If new devices are added, then it becomes difficult to reconfigure.

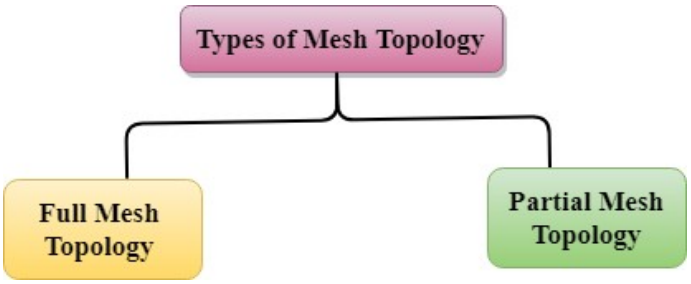
Mesh topology

- Mesh technology is an arrangement of the network in which computers are interconnected with each other through various redundant connections.
- There are multiple paths from one computer to another computer.
- It does not contain the switch, hub or any central computer which acts as a central point of communication.
- The Internet is an example of the mesh topology.
- Mesh topology is mainly used for WAN implementations where communication failures are a critical concern.
- Mesh topology is mainly used for wireless networks.
- Mesh topology can be formed by using the formula: **Number of cables = (n*(n-1))/2;**

Where n is the number of nodes that represents the network.

Mesh topology is divided into two categories:

- Fully connected mesh topology
- Partially connected mesh topology



- **Full Mesh Topology:** In a full mesh topology, each computer is connected to all the computers available in the network.
- **Partial Mesh Topology:** In a partial mesh topology, not all but certain computers are connected to those computers with which they communicate frequently.

Advantages of Mesh topology:

Reliable: The mesh topology networks are very reliable as if any link breakdown will not affect the communication between connected computers.

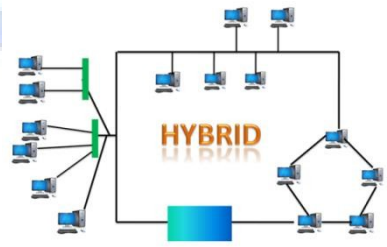
Fast Communication: Communication is very fast between the nodes.

Easier Reconfiguration: Adding new devices would not disrupt the communication between other devices.

Disadvantages of Mesh topology

- **Cost:** A mesh topology contains a large number of connected devices such as a router and more transmission media than other topologies.
- **Management:** Mesh topology networks are very large and very difficult to maintain and manage. If the network is not monitored carefully, then the communication link failure goes undetected.
- **Efficiency:** In this topology, redundant connections are high that reduces the efficiency of the network.

Hybrid Topology



- The combination of various different topologies is known as **Hybrid topology**.
- A Hybrid topology is a connection between different links and nodes to transfer the data.
- When two or more different topologies are combined together is termed as Hybrid topology and if similar topologies are connected with each other will not result in Hybrid

topology. For example, if there exist a ring topology in one branch of ICICI bank and bus topology in another branch of ICICI bank, connecting these two topologies will result in Hybrid topology.

Advantages of Hybrid Topology

- **Reliable:** If a fault occurs in any part of the network will not affect the functioning of the rest of the network.
- **Scalable:** Size of the network can be easily expanded by adding new devices without affecting the functionality of the existing network.
- **Flexible:** This topology is very flexible as it can be designed according to the requirements of the organization.
- **Effective:** Hybrid topology is very effective as it can be designed in such a way that the strength of the network is maximized and weakness of the network is minimized.

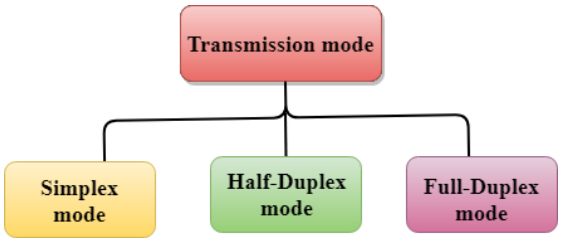
Disadvantages of Hybrid topology

- **Complex design:** The major drawback of the Hybrid topology is the design of the Hybrid network. It is very difficult to design the architecture of the Hybrid network.
- **Costly Hub:** The Hubs used in the Hybrid topology are very expensive as these hubs are different from usual Hubs used in other topologies.
- **Costly infrastructure:** The infrastructure cost is very high as a hybrid network requires a lot of cabling, network devices, etc.

Transmission modes

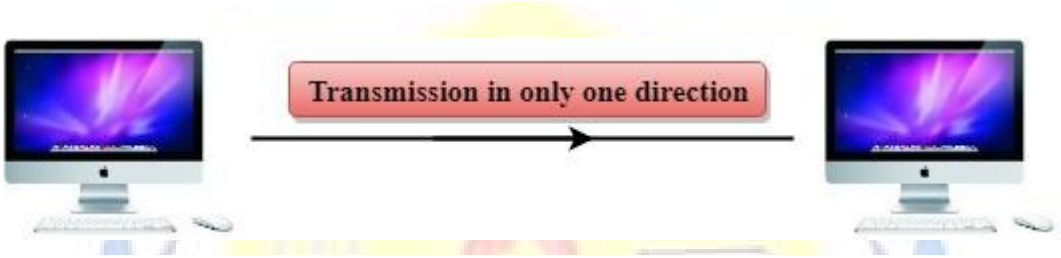
- The way in which data is transmitted from one device to another device is known as **transmission mode**.
- The transmission mode is also known as the communication mode.
- Each communication channel has a direction associated with it, and transmission media provide the direction. Therefore, the transmission mode is also known as a directional mode.
- The transmission mode is defined in the physical layer.

The Transmission mode is divided into three categories:



- Simplex mode
- Half-duplex mode
- Full-duplex mode

Simplex mode



- In Simplex mode, the communication is unidirectional, i.e., the data flow in one direction.
- A device can only send the data but cannot receive it or it can receive the data but cannot send the data.
- This transmission mode is not very popular as mainly communications require the two-way exchange of data. The simplex mode is used in the business field as in sales that do not require any corresponding reply.
- The radio station is a simplex channel as it transmits the signal to the listeners but never allows them to transmit back.
- Keyboard and Monitor are the examples of the simplex mode as a keyboard can only accept the data from the user and monitor can only be used to display the data on the screen.
- The main advantage of the simplex mode is that the full capacity of the communication channel can be utilized during transmission.

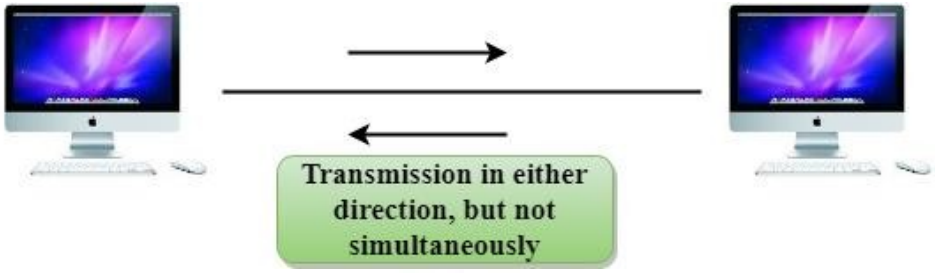
Advantage of Simplex mode:

- In simplex mode, the station can utilize the entire bandwidth of the communication channel, so that more data can be transmitted at a time.

Disadvantage of Simplex mode:

- Communication is unidirectional, so it has no inter-communication between devices.

Half-Duplex mode



- In a Half-duplex channel, direction can be reversed, i.e., the station can transmit and receive the data as well.
- Messages flow in both the directions, but not at the same time.
- The entire bandwidth of the communication channel is utilized in one direction at a time.
- In half-duplex mode, it is possible to perform the error detection, and if any error occurs, then the receiver requests the sender to retransmit the data.
- A **Walkie-talkie** is an example of the Half-duplex mode. In Walkie-talkie, one party speaks, and another party listens. After a pause, the other speaks and first party listens. Speaking simultaneously will create the distorted sound which cannot be understood.

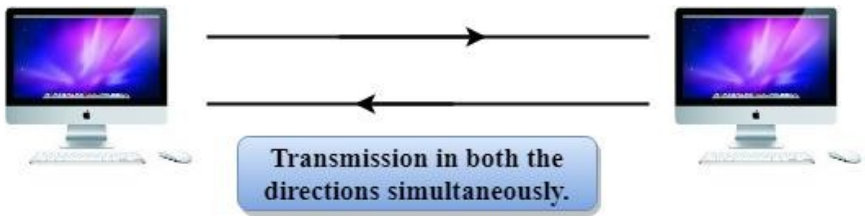
Advantage of Half-duplex mode:

- In half-duplex mode, both the devices can send and receive the data and also can utilize the entire bandwidth of the communication channel during the transmission of data.

Disadvantage of Half-Duplex mode:

- In half-duplex mode, when one device is sending the data, then another has to wait, this causes the delay in sending the data at the right time.

Full-duplex mode



- In Full duplex mode, the communication is bi-directional, i.e., the data flow in both the directions.
- Both the stations can send and receive the message simultaneously.
- Full-duplex mode has two simplex channels. One channel has traffic moving in one direction, and another channel has traffic flowing in the opposite direction.
- The Full-duplex mode is the fastest mode of communication between devices.
- The most common example of the full-duplex mode is a telephone network. When two people are communicating with each other by a telephone line, both can talk and listen at the same time.

Advantage of Full-duplex mode:

- Both the stations can send and receive the data at the same time.

Disadvantage of Full-duplex mode:

- If there is no dedicated path exists between the devices, then the capacity of the communication channel is divided into two parts.

Differences b/w Simplex, Half-duplex and Full-duplex mode

Basis for comparison	Simplex mode	Half-duplex mode	Full-duplex mode
Direction of communication	In simplex mode, the communication is unidirectional.	In half-duplex mode, the communication is bidirectional, but one at a time.	In full-duplex mode, the communication is bidirectional.
Send/Receive	A device can only send the data but cannot receive it or it can only	Both the devices can send and receive the data, but one at a time.	Both the devices can send and

	receive the data but cannot send it.		receive the data simultaneously.
Performance	The performance of half-duplex mode is better than the simplex mode.	The performance of full-duplex mode is better than the half-duplex mode.	The Full-duplex mode has better performance among simplex and half-duplex mode as it doubles the utilization of the capacity of the communication channel.
Example	Examples of Simplex mode are radio, keyboard, and monitor.	Example of half-duplex is Walkie-Talkies.	Example of the Full-duplex mode is a telephone network.

Computer Network Models

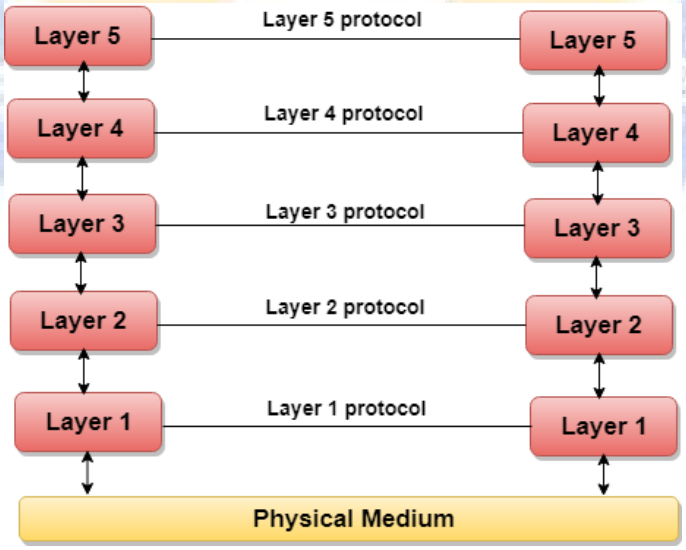
A communication subsystem is a complex piece of Hardware and software. Early attempts for implementing the software for such subsystems were based on a single, complex, unstructured program with many interacting components. The resultant software was very difficult to test and modify. To overcome such problem, the ISO has developed a layered approach. In a layered approach, networking concept is divided into several layers, and each layer is assigned a particular task. Therefore, we can say that networking tasks depend upon the layers.

Layered Architecture

- The main aim of the layered architecture is to divide the design into small pieces.
- Each lower layer adds its services to the higher layer to provide a full set of services to manage communications and run the applications.
- It provides modularity and clear interfaces, i.e., provides interaction between subsystems.

- It ensures the independence between layers by providing the services from lower to higher layer without defining how the services are implemented. Therefore, any modification in a layer will not affect the other layers.
- The number of layers, functions, contents of each layer will vary from network to network. However, the purpose of each layer is to provide the service from lower to a higher layer and hiding the details from the layers of how the services are implemented.
- The basic elements of layered architecture are services, protocols, and interfaces.
    - **Service:** It is a set of actions that a layer provides to the higher layer.
    - **Protocol:** It defines a set of rules that a layer uses to exchange the information with peer entity. These rules mainly concern about both the contents and order of the messages used.
    - **Interface:** It is a way through which the message is transferred from one layer to another layer.
- In a layer n architecture, layer n on one machine will have a communication with the layer n on another machine and the rules used in a conversation are known as a layer-n protocol.

Let's take an example of the five-layered architecture.



- In case of layered architecture, no data is transferred from layer n of one machine to layer n of another machine. Instead, each layer passes the data to the layer immediately just below it, until the lowest layer is reached.
- Below layer 1 is the physical medium through which the actual communication takes place.

- In a layered architecture, unmanageable tasks are divided into several small and manageable tasks.
- The data is passed from the upper layer to lower layer through an interface. A Layered architecture provides a clean-cut interface so that minimum information is shared among different layers. It also ensures that the implementation of one layer can be easily replaced by another implementation.
- A set of layers and protocols is known as network architecture.

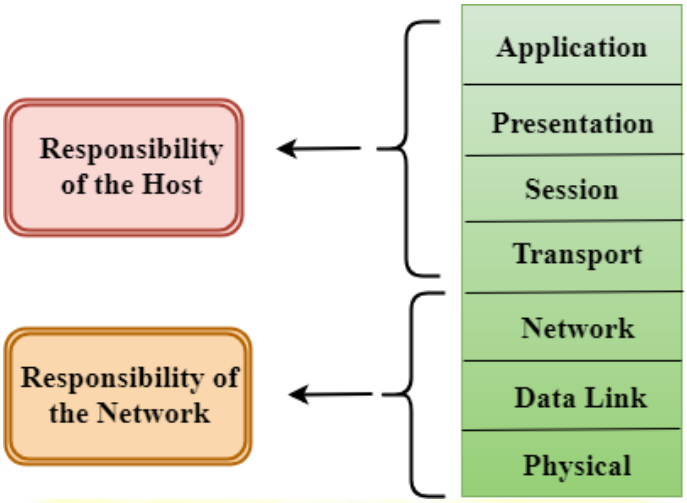
Why do we require Layered architecture?

- **Divide-and-conquer approach:** Divide-and-conquer approach makes a design process in such a way that the unmanageable tasks are divided into small and manageable tasks. In short, we can say that this approach reduces the complexity of the design.
- **Modularity:** Layered architecture is more modular. Modularity provides the independence of layers, which is easier to understand and implement.
- **Easy to modify:** It ensures the independence of layers so that implementation in one layer can be changed without affecting other layers.
- **Easy to test:** Each layer of the layered architecture can be analyzed and tested individually.

OSI Model

- OSI stands for **Open System Interconnection** is a reference model that describes how information from a software application in one computer moves through a physical medium to the software application in another computer.
- OSI consists of seven layers, and each layer performs a particular network function.
- OSI model was developed by the International Organization for Standardization (ISO) in 1984, and it is now considered as an architectural model for the inter-computer communications.
- OSI model divides the whole task into seven smaller and manageable tasks. Each layer is assigned a particular task.
- Each layer is self-contained, so that task assigned to each layer can be performed independently.

Characteristics of OSI Model:

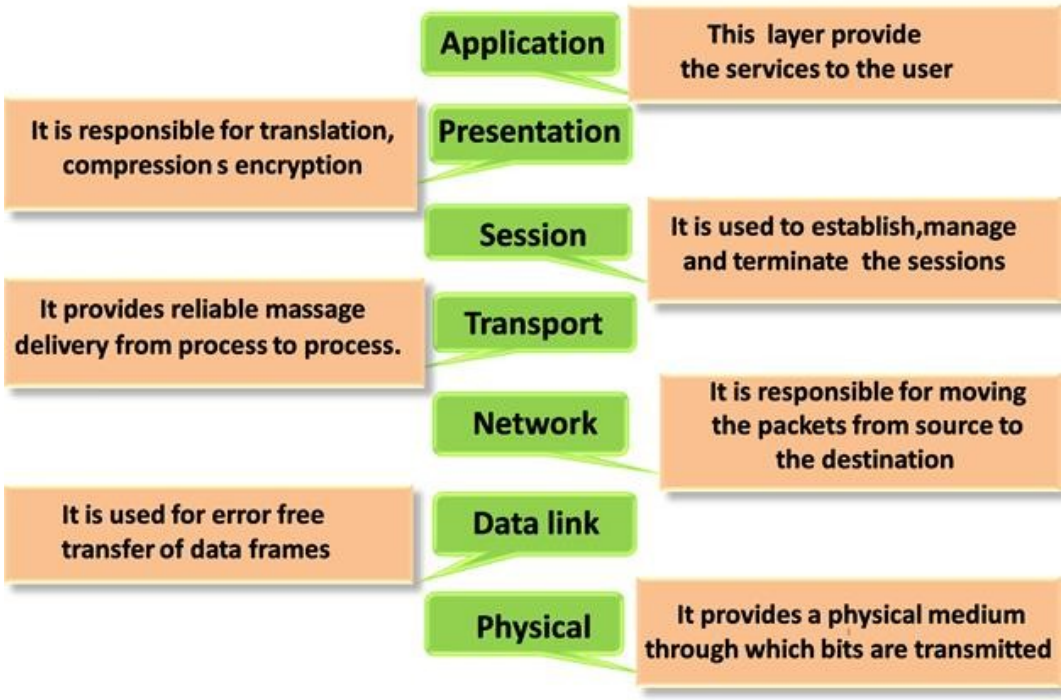


- The OSI model is divided into two layers: upper layers and lower layers.
- The upper layer of the OSI model mainly deals with the application related issues, and they are implemented only in the software. The application layer is closest to the end user. Both the end user and the application layer interact with the software applications. An upper layer refers to the layer just above another layer.
- The lower layer of the OSI model deals with the data transport issues. The data link layer and the physical layer are implemented in hardware and software. The physical layer is the lowest layer of the OSI model and is closest to the physical medium. The physical layer is mainly responsible for placing the information on the physical medium.

Functions of the OSI Layers

There are the seven OSI layers. Each layer has different functions. A list of seven layers are given below:

1. Physical Layer
2. Data-Link Layer
3. Network Layer
4. Transport Layer
5. Session Layer
6. Presentation Layer
7. Application Layer

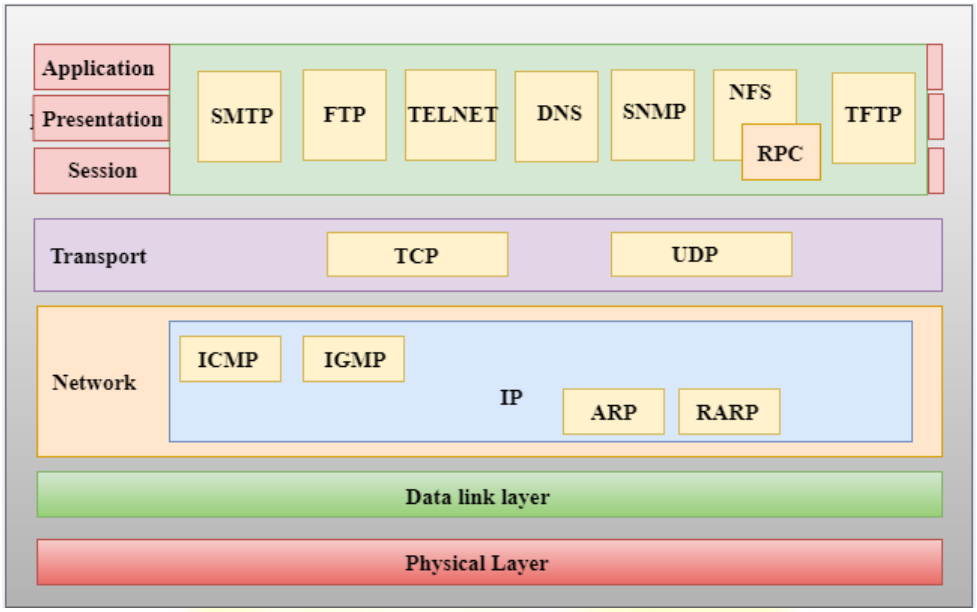


TCP/IP model

- The TCP/IP model was developed prior to the OSI model.
- The TCP/IP model is not exactly similar to the OSI model.
- The TCP/IP model consists of five layers: the application layer, transport layer, network layer, data link layer and physical layer.
- The first four layers provide physical standards, network interface, internetworking, and transport functions that correspond to the first four layers of the OSI model and these four layers are represented in TCP/IP model by a single layer called the application layer.
- TCP/IP is a hierarchical protocol made up of interactive modules, and each of them provides specific functionality.

Here, hierarchical means that each upper-layer protocol is supported by two or more lower-level protocols.

Functions of TCP/IP layers:



Network Access Layer

- A network layer is the lowest layer of the TCP/IP model.
- A network layer is the combination of the Physical layer and Data Link layer defined in the OSI reference model.
- It defines how the data should be sent physically through the network.
- This layer is mainly responsible for the transmission of the data between two devices on the same network.
- The functions carried out by this layer are encapsulating the IP datagram into frames transmitted by the network and mapping of IP addresses into physical addresses.
- The protocols used by this layer are ethernet, token ring, FDDI, X.25, frame relay.

Internet Layer

- An internet layer is the second layer of the TCP/IP model.
- An internet layer is also known as the network layer.
- The main responsibility of the internet layer is to send the packets from any network, and they arrive at the destination irrespective of the route they take.

Following are the protocols used in this layer are:

IP Protocol: IP protocol is used in this layer, and it is the most significant part of the entire TCP/IP suite.

Following are the responsibilities of this protocol:

- **IP Addressing:** This protocol implements logical host addresses known as IP addresses. The IP addresses are used by the internet and higher layers to identify the device and to provide internetwork routing.
- **Host-to-host communication:** It determines the path through which the data is to be transmitted.
- **Data Encapsulation and Formatting:** An IP protocol accepts the data from the transport layer protocol. An IP protocol ensures that the data is sent and received securely, it encapsulates the data into message known as IP datagram.
- **Fragmentation and Reassembly:** The limit imposed on the size of the IP datagram by data link layer protocol is known as Maximum Transmission unit (MTU). If the size of IP datagram is greater than the MTU unit, then the IP protocol splits the datagram into smaller units so that they can travel over the local network. Fragmentation can be done by the sender or intermediate router. At the receiver side, all the fragments are reassembled to form an original message.
- **Routing:** When IP datagram is sent over the same local network such as LAN, MAN, WAN, it is known as direct delivery. When source and destination are on the distant network, then the IP datagram is sent indirectly. This can be accomplished by routing the IP datagram through various devices such as routers.

ARP Protocol

- ARP stands for **Address Resolution Protocol**.
- ARP is a network layer protocol which is used to find the physical address from the IP address.
- **The two terms are mainly associated with the ARP Protocol:**
    - **ARP request:** When a sender wants to know the physical address of the device, it broadcasts the ARP request to the network.
    - **ARP reply:** Every device attached to the network will accept the ARP request and process the request, but only recipient recognize the IP address and sends back its physical address in the form of ARP reply. The recipient adds the physical address both to its cache memory and to the datagram header

ICMP Protocol

- **ICMP** stands for Internet Control Message Protocol.
- It is a mechanism used by the hosts or routers to send notifications regarding datagram problems back to the sender.
- A datagram travels from router-to-router until it reaches its destination. If a router is unable to route the data because of some unusual conditions such as disabled links, a device is on fire or network congestion, then the ICMP protocol is used to inform the sender that the datagram is undeliverable.
- An ICMP protocol mainly uses two terms:
    - **ICMP Test:** ICMP Test is used to test whether the destination is reachable or not.
    - **ICMP Reply:** ICMP Reply is used to check whether the destination device is responding or not.
- The core responsibility of the ICMP protocol is to report the problems, not correct them. The responsibility of the correction lies with the sender.
- ICMP can send the messages only to the source, but not to the intermediate routers because the IP datagram carries the addresses of the source and destination but not of the router that it is passed to.

Transport Layer

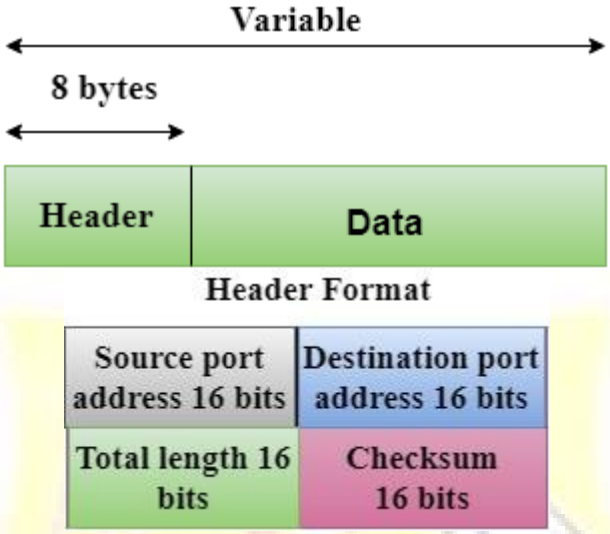
The transport layer is responsible for the reliability, flow control, and correction of data which is being sent over the network.

The two protocols used in the transport layer are **User Datagram protocol and Transmission control protocol**.

- **User Datagram Protocol (UDP)**
    - It provides connectionless service and end-to-end delivery of transmission.
    - It is an unreliable protocol as it discovers the errors but not specify the error.
    - User Datagram Protocol discovers the error, and ICMP protocol reports the error to the sender that user datagram has been damaged.
    - **UDP consists of the following fields:**
      Source port address: The source port address is the address of the application program that has created the message.
      Destination port address: The destination port address is the address of the application program that receives the message.

Total length: It defines the total number of bytes of the user datagram in bytes.
Checksum: The checksum is a 16-bit field used in error detection.

- UDP does not specify which packet is lost. UDP contains only checksum; it does not contain any ID of a data segment.



- **Transmission Control Protocol (TCP)**
  - It provides a full transport layer services to applications.
  - It creates a virtual circuit between the sender and receiver, and it is active for the duration of the transmission.
  - TCP is a reliable protocol as it detects the error and retransmits the damaged frames. Therefore, it ensures all the segments must be received and acknowledged before the transmission is considered to be completed and a virtual circuit is discarded.
  - At the sending end, TCP divides the whole message into smaller units known as segment, and each segment contains a sequence number which is required for reordering the frames to form an original message.
  - At the receiving end, TCP collects all the segments and reorders them based on sequence numbers.

Application Layer

- An application layer is the topmost layer in the TCP/IP model.
- It is responsible for handling high-level protocols, issues of representation.
- This layer allows the user to interact with the application.

- When one application layer protocol wants to communicate with another application layer, it forwards its data to the transport layer.
- There is an ambiguity occurs in the application layer. Every application cannot be placed inside the application layer except those who interact with the communication system. For example: text editor cannot be considered in application layer while web browser using **HTTP** protocol to interact with the network where **HTTP** protocol is an application layer protocol.

Following are the main protocols used in the application layer:

- **HTTP:** HTTP stands for Hypertext transfer protocol. This protocol allows us to access the data over the world wide web. It transfers the data in the form of plain text, audio, video. It is known as a Hypertext transfer protocol as it has the efficiency to use in a hypertext environment where there are rapid jumps from one document to another.
- **SNMP:** SNMP stands for Simple Network Management Protocol. It is a framework used for managing the devices on the internet by using the TCP/IP protocol suite.
- **SMTP:** SMTP stands for Simple mail transfer protocol. The TCP/IP protocol that supports the e-mail is known as a Simple mail transfer protocol. This protocol is used to send the data to another e-mail address.
- **DNS:** DNS stands for Domain Name System. An IP address is used to identify the connection of a host to the internet uniquely. But, people prefer to use the names instead of addresses. Therefore, the system that maps the name to the address is known as Domain Name System.
- **TELNET:** It is an abbreviation for Terminal Network. It establishes the connection between the local computer and remote computer in such a way that the local terminal appears to be a terminal at the remote system.
- **FTP:** FTP stands for File Transfer Protocol. FTP is a standard internet protocol used for transmitting the files from one computer to another computer.

Internetwork

- An internetwork is defined as two or more computer network LANs or WAN or computer network segments are connected using devices, and they are configured by a local addressing scheme. This process is known as **internetworking**.

- An interconnection between public, private, commercial, industrial, or government computer networks can also be defined as **internetworking**.
- An internetworking uses the **internet protocol**.
- The reference model used for internetworking is **Open System Interconnection(OSI)**.

Types of Internetwork:

❖ Extranet:

- An extranet is a communication network based on the internet protocol such as **Transmission Control protocol** and **internet protocol**.
- It is used for information sharing.
- The access to the extranet is restricted to only those users who have login credentials.
- An extranet is the lowest level of internetworking. It can be categorized as **MAN**, **WAN** or other computer networks.
- An extranet cannot have a single **LAN**, at least it must have one connection to the external network.

❖ Intranet:

- An intranet is a private network based on the internet protocol such as **Transmission Control protocol** and **internet protocol**.
- An intranet belongs to an organization which is only accessible by the **organization's employee** or members.
- The main aim of the intranet is to share the information and resources among the organization employees.
- An intranet provides the facility to work in groups and for teleconferences.

❖ Intranet advantages:

- **Communication:** It provides a cheap and easy communication. An employee of the organization can communicate with another employee through email, chat.
- **Time-saving:** Information on the intranet is shared in real time, so it is time-saving.
- **Collaboration:** Collaboration is one of the most important advantage of the intranet. The information is distributed among the employees of the organization and can only be accessed by the authorized user.
- **Platform independency:** It is a neutral architecture as the computer can be connected to another device with different architecture.

- **Cost effective:** People can see the data and documents by using the browser and distributes the duplicate copies over the intranet. This leads to a reduction in the cost.

Internet

Internet is a global computer network providing a variety of information and communication facilities, consisting of interconnected networks using standardized communication protocols.

Characteristics of Internet:

- **Interoperable:** Interoperable means that the standards allow communication across networks. This does not limit the access of information to a proprietary site, location, machine or band name.
- **Packet switched:** Connection are not fixed from point to point for the duration of the transmission. A telephone call is circuit switched-which means a dedicated path is established to transmit your entire conversation. When data is sent packet switched over the internet-it transmits a small part of the data, verifies it is correct then sends more information toward the destination. Packet switched networks do not require all of the information to be delivered through the same path. By not dedicating the path for the duration of the connection, this method allows more connections to be sending information across the same space or allows for sharing resources.
- **Data network:** A network that carries data information (digital- computer) instead of voice information (analog-telephone). There are many instances where these "definition"of data and voice are starting to overlap. Computers connecting to regular phone lines are technically carrying data over a voice line and in some progressive parts of the country digital phone lines are starting to make appearances.

History of Internet

❖ 1960's

- **1969-** The department of defence advanced research projects organization (ARPA) creates an experiment network called ARPANET.
- This network provides a test-bed for emerging network technologies.
- ARPANET continued to expand, connecting many more sites throughout the 1970's and 1980's.

❖ 1970's

- Networking tools are developed in the 1970's such as :
- **1972-** The national centre for supercomputing applications (NCSA) develops the telnet application for remote login, making it easier to connect to a remote computer.
- **1973-** FTP (file transfer protocol) is introduced, standardizing the transfer of files between networked computers.

❖ 1980's

- TCP/IP suite of networking protocols, or rules, becomes the only set of protocols used on the ARPANET.
- To keep military and non-military network sites separates, the ARPANET splits into two networks: ARPANET and MILNET.
- 1982-1983:- **The first desktop computers begin to appear. Many are equipped with an operating system called Berkeley UNIX which includes networking software.**
- 1985-86:- **The national science foundation (NSF) connects the nation's six supercomputing centers together. This network is called the NSFNET, or NSFNET backbone.**
- 1987:- **The NSF awards a grant to merit network, inc to operate and manage future development of the NSFNET backbone.**
- **1989-** The backbone network is upgraded to "TI" which means that it is able to transmit data at speeds of 1.5 millions bits of data per second, or about 50 pages of text per second.

❖ 1990's

- The ARPANET is dissolved.
- **Gopher** is developed at the university of Minnesota. Gopher provides a hierarchical, menu-based method for providing and locating information on the internet.
- European laboratory for particle physics in Switzerland (CERN) releases the **world wide web** (WWW), developed by Tim burners-lee.
- The WWW uses hypertext transfer protocol (HTTP) and hypertext links, changing the way information can be organized.

- **1993-** The NSFNET backbone network is upgraded to "T3" which means that it is able to transmit data at speeds of 45 million bits of data per second, of about 1400 pages of text per second.
- **1993-1994-** The graphical web browser mosaic and Netscape navigator are introduced and spread through the internet community.
- **1995-** The NSFNET back bone is replaced by a new network architecture, called VBNS (very high speed backbone network system) that utilizes network service providers, regional networks and network access points (NAPs).

How Internet Works

To visit any website.

- First you enter the address of URL of the website in your web browser.
- Then your browser requests the web page from the web server that hosts the site.
- Then server sends the data over the internet to your computer.
- Then your web browser interprets the data, displaying it on your computer screen.

To access the web we need a web browser, such as Netscape navigator or Microsoft internet explorer. Web pages are written in a computer language called HTML

World Wide Web (WWW)

- WWW stands for World Wide Web.
- A technical definition of the World Wide Web is − All the resources and users on the Internet that are using the Hypertext Transfer Protocol HTTP.
- The World Wide Web is the universe of network-accessible information, an embodiment of human knowledge.
- The World Wide Web is a way of exchanging information between computers on the Internet, tying them together into a vast collection of interactive multimedia resources.
- It is a hypertext interface to internet information resources.
- **Internet** and **Web** is not the same thing: Web uses internet to pass over the information.

Basic WWW Concepts

The World Wide Web is a way of exchanging information between computers on the Internet, tying them together into a vast collection of interactive multimedia resources. It is a hypertext interface to internet information resources.

HTTP

- HTTP is the short form for Hypertext Transfer Protocol.
- An example of the standard URL is **http://www.google.com**.
- The prefix http in the URL designates the protocol being used for communication.
- The HTTP protocol is used to access hypertext documents on the World Wide Web.

URL

- URL refers to Uniform Resource Locator.
- URLs are used as the address of the documents available on the World Wide Web.
- A URL is the fundamental identifier of any resource available on the web e. g., hypertext pages, images, and sound files.
- The standard format of a URL has been specified as –

  protocol://hostname/other_information
- A protocol is, basically, a specification of the flow of information in a network.
- The protocol required for accessing the resources available on the web is called **Hyper Text Transfer Protocol** (HTTP). Other protocols supported by the web browsers include **telnet, FTP, Gopher**, etc.
- The name of the protocol is followed by a colon, two forward slashes, and then the hostname.
- The hostname refers to the computer having the requested resource. Hostname is followed by a single forward slash and links to subdirectories or any particular file.
- The path to the directory holding the requested file or resource may contain single forward slashes.

Website

- Website refers to a collection of web pages usually written in markup language.
- For example: www.google.com refers to the website of Google having 'com' as the domain name.

- Websites can be understood as a location on the World Wide Web that holds the resources which people can request when required.
- The first or the initial page of any website is known as the home page of that particular website.

Web Server

- Websites are usually hosted on a computer which is known as a Web server.
- These servers should require constant connectivity with the internet so that the resources stored by them can be accessed from anywhere in the world.
- A web server is provided with a unique address which is required to access the contents of the server.
- The unique address is the identifier for the server and is termed as IP Address. These IP Addresses comprise of a series of four numbers separated by dots or periods.
- Each number lies between the range of 0 – 255, for example: 192.168.1.1.
- IP address of the web server is also required when the URL of the website hosted on that server is registered on the World Wide Web.

Web Browser

- Web Browser is a software required to access the resources available on the World Wide Web.
- Hence, in order to access any website web browser is required.
- Examples of web browsers are: Microsoft Internet Explorer, Google Chrome, Mozilla Firefox, Netscape Navigator, etc. Navigating through the pages of a website is termed as web browsing or surfing.

ISP

- ISP is the short form for Internet Service Provider.
- They are basically the organizations responsible for providing internet connections to the subscribers.
- Several ISPs also provide space on their web servers for hosting website of the subscribers.

HTML

- HTML refers to Hyper Text Markup Language.
- It is the development language of the websites available on the web.
- HTML can be understood as a subset of SGML (Standard Generalized Mark-Up Language).

Hyperlink

- Hyperlinks can be understood as the access points to other resources of the same or different websites.
- A hyperlink, sometimes referred to as a link, is a clickable element in an electronic document used to navigate through the linked resources.
- Typically, you click the hyperlink to access the linked resource.
- Hyperlinks can be in the form of clickable textual links, icons, buttons, etc.

DNS

- DNS is the short form for Domain Name System.
- It is used to provide a mapping between the name of the requested website and its IP address.
- Usually, user remembers the name or URL of a website.
- When user enters this URL in the web browser, a request is sent to DNS for resolving the entered URL in terms of its IP address.
- It is the IP address which will be used to locate the requested resource on the web.
- The mapping of the URL and the IP address a website is usually built during the registration of the domain name.

W3C

- W3C (World Wide Web Consortium) is an international organization comprising of full-time staff, member organizations, and the public for the development of Web standards.
- The inventor of W3C is Tim Berners-Lee while CEO Jeffrey Jaffe is the CEO of W3C.
- The main aim of the W3C community is to streamline the development of the open standards for the web.
- The mission of W3C's is to shape the Web to its full potential.
- W3C is also one of the main standards body for HTTP and HTML.

CLIENTS AND SERVERS

- A host machine that requests for a resource on the Internet is treated as client.
- The machine and/or software that manages and replies these requests are called as servers.
- There are different types of clients like email client, web client, FTP client the corresponding servers are Email server, web server and FTP server.

Internet Domains

- Machines on the internet are uniquely identified using their corresponding IP addresses. Similarly, the names assigned to the computers must be unique to avoid ambiguity.
- A **Namespace** refers to the mapping between the names of the machine and their corresponding IP address. Namespaces can be organized in two ways:
    - **Flat Name Space:** This indicates the direct mapping of the IP address to the complete name of the machine. Names are simply a series of characters having no specific structure.
    - **Hierarchical Name Space:** In this name space, every name is composed of several parts. The first component may refer to the nature of organization, second may indicate name of organization, third may denote the department in the organization, and henceforth.
    - Hierarchical name space requires **domain name space** which is an inverted tree structure with each level corresponding to a component of the name space. The complete tree shall have only 128 levels, where level 0 refers to the root. Figure 1.2 shows the structure of DNS.

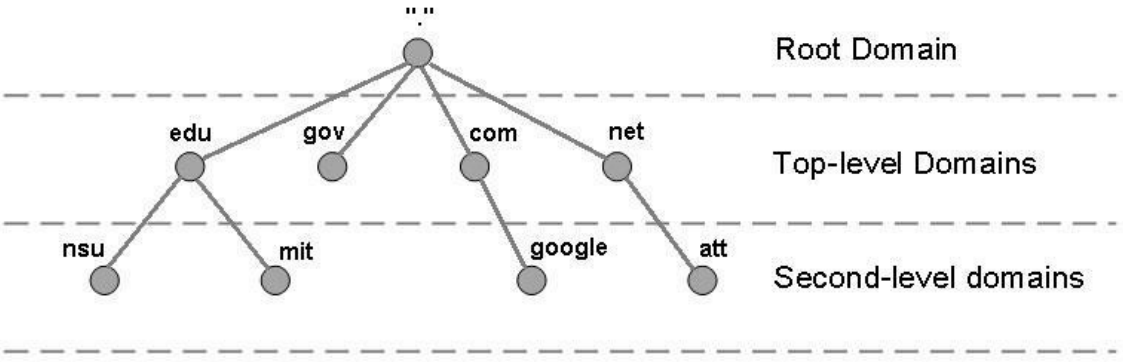


Figure 1.2: Domain Name Space

A domain refers to a sub tree of the domain name space, as shown in below Figure . The domain name of the root of the sub tree will be the name of the domain. A domain can be further subdivided into sub domains.

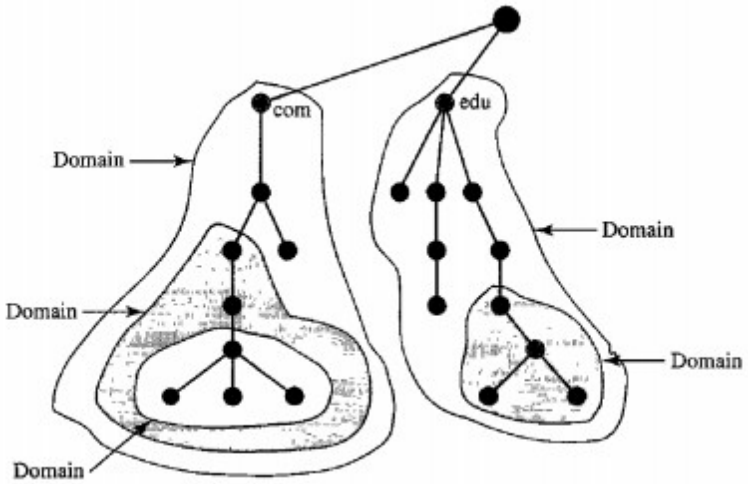


Figure 1.3: Domains

In the context of the Internet, the domain name space (tree) can be classified as:

- Generic domains
- Country domains
- Inverse domains

Generic Domains: Generic domains identify the registered hosts as per their general capabilities. Each node of a such a tree refers to a domain which serves as the index for the domain name space database. Figure below shows structure of Generic Domains

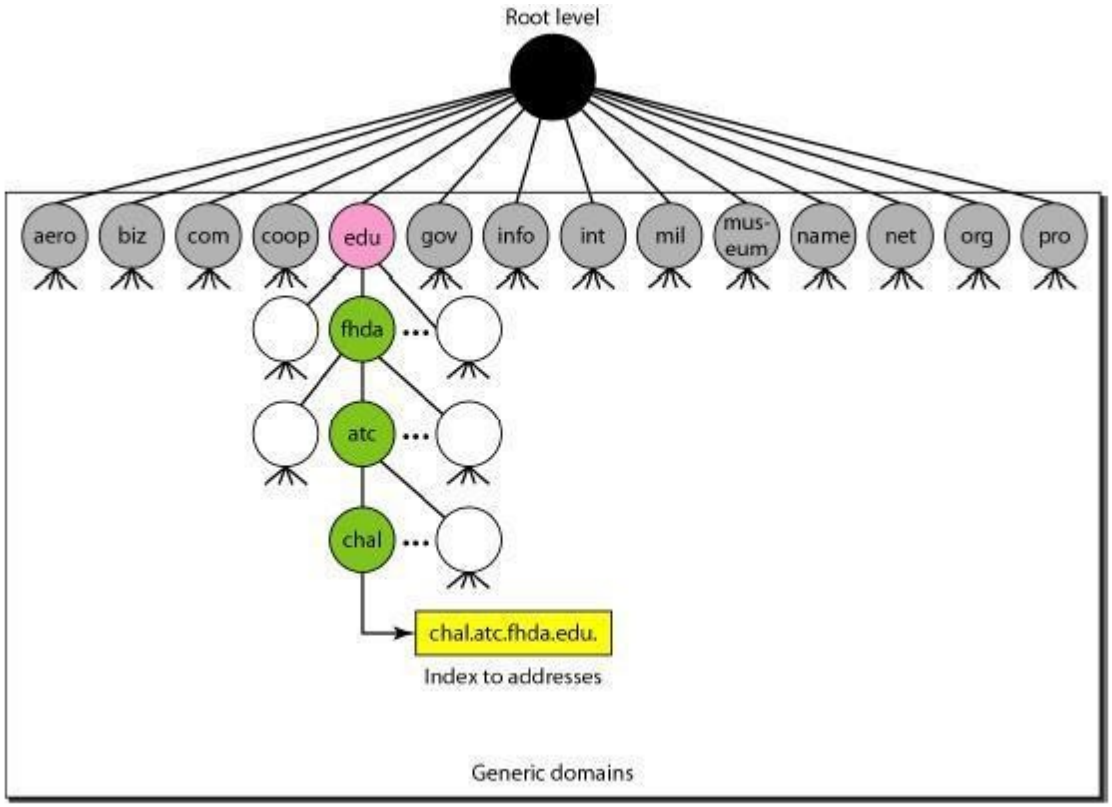


Figure 1.4: Generic Domains

Country Domains: Country Domains employ abbreviations of the country. The abbreviations are limited to two characters, e.g. *In* for India. Further labels may refer to the specific organization or other national designations. Consider the following country domain: **anza.cup.ca.us.** This refers to De Anza College in Cupertino, California, United States.

Inverse Domains: The inverse domain maps the IP address of the machine to its corresponding name. Consider, a client requesting some resource from a server. The server may have a list of authenticated clients. However, the request received from the client has only the IP address of the client. In such cases, it is required to resolve the IP address to the name of the client so that the server can verify it with the list of authenticated clients it has.

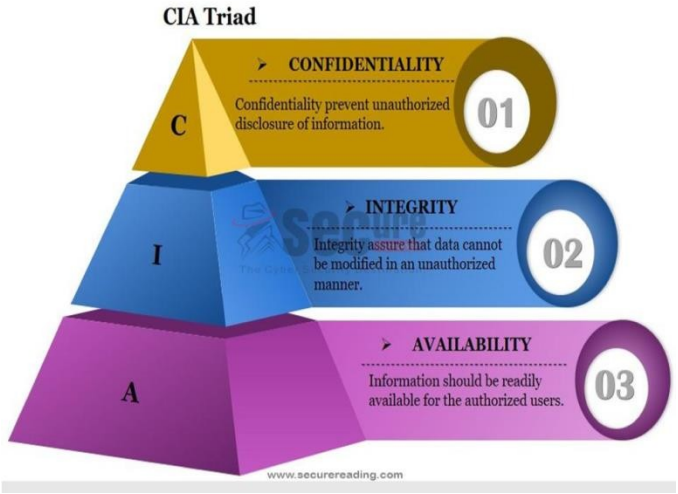
UNIT – 3 INTERNET SECURITY

Security

Security is **freedom from, or resilience against, potential harm** (or other unwanted coercive change) caused by others. Beneficiaries (technically referents) of security may be of persons and social groups, objects and institutions, ecosystems or any other entity or phenomenon vulnerable to unwanted change.

Basic Security Principle Concepts

- Three basic information security concepts important to information are **Confidentiality, Integrity, and Availability.** If we relate these concepts with the people who use that information, then it will be **authentication, authorization, and non-repudiation.**
- Information security is the confidentiality, integrity, and availability of information also referred as C-I-A triad or information security triad.
- In brief, **confidentiality** is a set of rules that limits access to information, **Integrity** is the assurance that the information is trustworthy and accurate, and **Availability** is a guarantee of reliable access to the information by authorized people.



Confidentiality

- When information is read or copied by someone not authorized to do so, then it will be **"loss of confidentiality".**
- For sensitive information, confidentiality is a very important criterion. Bank account statements, personal information, credit card numbers, trade secrets, government

documents are some examples of sensitive information. This goal of the CIA triad emphasizes the need for information protection.

- For example, confidentiality is maintained for a computer file, if authorized users are able to view it, while unauthorized persons are blocked from seeing it.

Integrity

- Information can be corrupted or manipulated if it's available on an insecure network, and is referred to as "**loss of integrity."** This means that unauthorized changes are made to information, whether by human error or intentional tampering.
- Integrity is particularly important for critical safety and financial data used for activities such as electronic funds transfers, air traffic control, and financial accounting.
- For example, banks are more concerned about the integrity of financial records, with confidentiality having only second priority. Some bank account holders or depositors leave ATM receipts unchecked and hanging around after withdrawing cash. This shows that confidentiality does not have the highest priority. In the CIA triad, integrity is maintained when the information remains unchanged during storage, transmission, and usage not involving modification to the information.

Availability

- Information can be erased or become inaccessible, resulting in **"loss of availability."** This means that people who are authorized to get information are restricted from accessing. Availability is often the most important attribute in service-oriented businesses that depend on information. Denying access to information has become a very common attack nowadays.
- Availability of the network itself is important to anyone whose business or education relies on a network connection. When users cannot access the network or specific services provided on the network, they experience a denial of service.

Now let's take a look at other key terms in Information Security – Authorization, Authentication, and Non-repudiation processes and methods, which are some of the main controls aimed at protecting the C-I-A triad

To make information available or accessible/modifiable to those who need it and who can be trusted with it (for accessing and modification), organizations use **authentication and authorization.**

Authentication

- **Authentication** is proving that a user is the person he or she claims to be. That proof may involve something the user knows (such as a password), something the user has (such as a “smartcard”), or something about the user that proves the person’s identity (such as a fingerprint).

Authorization

- **Authorization** is the act of determining whether a particular user (or computer system) has the right to carry out a certain activity, such as reading a file or running a program.

Non-repudiation

- Users must be authenticated before carrying out the activity they are authorized to perform. Security is strong when the means of authentication cannot later be refuted—the user cannot later deny that he or she performed the activity. This is known as **non-repudiation.**

Internet Security

Internet security is a tree branch of computer security specifically related to the Internet, often involving browser security but also network security on a more general level as it applies to other applications or operating systems on a whole. Its objective is to establish rules and measures to use against attacks over the Internet. The Internet represents an insecure channel for exchanging information leading to a high risk of intrusion or fraud, such as phishing. Different methods have been used to protect the transfer of data, including encryption.

Types of security

1. Network layer security

TCP/IP which stands for Transmission Control Protocol (TCP) and Internet Protocol (IP) aka Internet protocol suite can be made secure with the help of cryptographic methods and protocols. These protocols include Secure Sockets Layer (SSL), succeeded by Transport

Layer Security (TLS) for web traffic, Pretty Good Privacy (PGP) for email, and IPsec for the network layer security.

Internet Protocol Security (IPsec)

- This protocol is designed to protect communication in a secure manner using TCP/IP aka Internet protocol suite. It is a set of security extensions developed by the Internet Task force IETF, and it provides security and authentication at the IP layer by transforming data using encryption.
- Two main types of transformation that form the basis of IPsec: the **Authentication Header** (AH) and **Encapsulating Security Payload** (ESP). These two protocols provide data integrity, data origin authentication, and anti-replay service. These protocols can be used alone or in combination to provide the desired set of security services for the Internet Protocol (IP) layer.
- The basic components of the IPsec security architecture are described in terms of the following functionalities:
    - Security protocols for AH and ESP
    - Security association for policy management and traffic processing
    - Manual and automatic key management for the Internet key exchange (IKE)
    - Algorithms for authentication and encryption

Electronic mail security (E-mail)

- Email messages are composed, delivered, and stored in a multiple step process, which starts with the message's composition. When the user finishes composing the message and sends it, the message is transformed into a standard format: an RFC 2822 formatted message. Afterwards, the message can be transmitted.
- Using a network connection, the mail client, referred to as a mail user agent (MUA), connects to a mail transfer agent (MTA) operating on the mail server. The mail client then provides the sender's identity to the server.
- Next, using the mail server commands, the client sends the recipient list to the mail server. The client then supplies the message. Once the mail server receives and processes the message, several events occur: recipient server identification, connection establishment, and message transmission.
- Using Domain Name System (DNS) services, the sender's mail server determines the mail server(s) for the recipient(s). Then, the server opens up a connection(s) to the

recipient mail server(s) and sends the message employing a process similar to that used by the originating client, delivering the message to the recipient(s).

Pretty Good Privacy (PGP)

Pretty Good Privacy provides confidentiality by encrypting messages to be transmitted or data files to be stored using an encryption algorithm such Triple DES or CAST-128. Email messages can be protected by using cryptography in various ways, such as the following:

- Signing an email message to ensure its integrity and confirm the identity of its sender.
- Encrypting the body of an email message to ensure its confidentiality.
- Encrypting the communications between mail servers to protect the confidentiality of both message body and message header.

The first two methods, message signing and message body encryption, are often used together; however, encrypting the transmissions between mail servers is typically used only when two organizations want to protect emails regularly sent between each other.

Multipurpose Internet Mail Extensions (MIME)

MIME transforms non-ASCII data at the sender's site to Network Virtual Terminal (NVT) ASCII data and delivers it to client's Simple Mail Transfer Protocol (SMTP) to be sent through the Internet. The server SMTP at the receiver's side receives the NVT ASCII data and delivers it to MIME to be transformed back to the original non-ASCII data.

Message Authentication Code

A Message authentication code (MAC) is a cryptography method that uses a secret key to encrypt a message. This method outputs a MAC value that can be decrypted by the receiver, using the same secret key used by the sender. The Message Authentication Code protects both a message's data integrity as well as its authenticity.

Firewalls

A computer firewall controls access between networks. It generally consists of gateways and filters which vary from one firewall to another. Firewalls also screen network traffic and are able to block traffic that is dangerous. Firewalls act as the intermediate server between SMTP and Hypertext Transfer Protocol (HTTP) connections.

Malicious software

A computer user can be tricked or forced into downloading software onto a computer that is of malicious intent. Such software comes in many forms, such as viruses, Trojan horses, spyware, and worms.

- **Malware,** short for malicious software, is any software used to disrupt computer operation, gather sensitive information, or gain access to private computer systems. Malware is defined by its malicious intent, acting against the requirements of the computer user, and does not include software that causes unintentional harm due to some deficiency. The term badware is sometimes used, and applied to both true (malicious) malware and unintentionally harmful software.
- A **botnet** is a network of zombie computers that have been taken over by a robot or bot that performs large-scale malicious acts for the creator of the botnet.
- **Computer Viruses** are programs that can replicate their structures or effects by infecting other files or structures on a computer. The common use of a virus is to take over a computer to steal data.
- **Computer worms** are programs that can replicate themselves throughout a computer network, performing malicious tasks throughout.
- **Ransomware** is a type of malware which restricts access to the computer system that it infects, and demands a ransom paid to the creator(s) of the malware in order for the restriction to be removed.
- **Scareware** is scam software with malicious payloads, usually of limited or no benefit that are sold to consumers via certain unethical marketing practices. The selling approach uses social engineering to cause shock, anxiety, or the perception of a threat, generally directed at an unsuspecting user.

- **Spyware** refers to programs that surreptitiously monitor activity on a computer system and report that information to others without the user's consent.
- A **Trojan horse,** commonly known as a *Trojan*, is a general term for malicious software that pretends to be harmless, so that a user willingly allows it to be downloaded onto the computer.

Denial-of-service attack

A denial-of-service attack (DoS attack) or distributed denial-of-service attack (DDoS attack) is an attempt to make a computer resource unavailable to its intended users. Although the means to carry out, motives for, and targets of a DoS attack may vary, it generally consists of the concerted efforts to prevent an Internet site or service from functioning efficiently or at all, temporarily or indefinitely.

Suppose you wish to visit an e-commerce site in order to shop for a gift. Your computer sends a small packet of information to the website. The packet works as a "hello" – basically, your computer says, "Hi, I'd like to visit you, please let me in."

When the server receives your computer's message, it sends a short one back, saying in a sense, "OK, are you real?" Your computer responds — "Yes!" — and communication is established.

The website's homepage then pops up on your screen, and you can explore the site. Your computer and the server continue communicating as you click links, place orders, and carry out other business.

In a DoS attack, a computer is rigged to send not just one "introduction" to a server, but hundreds or thousands. The server — which cannot tell that the introductions are fake — sends back its usual response, waiting up to a minute in each case to hear a reply. When it gets no reply, the server shuts down the connection, and the computer executing the attack repeats, sending a new batch of fake requests.

DoS attacks mostly affect organizations and how they run in a connected world. For consumers, the attacks hinder their ability to access services and information.

Phishing

Phishing is a cyber-attack involving disguised emails. Hackers try to trick email recipients into believing that a message is genuine and relevant – a request from their bank or a note from a co-worker, for example – so that they click on a link or open an attachment. The goal is to deceive people into handing over their personal information or downloading malware.

Phishing is one of the oldest internet security threats, dating back to the 1990s. It has remained popular to this day since it is one of the cheapest and easiest ways for criminals to steal information. In recent years, phishing techniques and messages have become increasingly sophisticated.

Internet security products

1. Antivirus

   Antivirus and Internet security programs can protect a programmable device from malware by detecting and eliminating viruses; Antivirus software was mainly shareware in the early years of the Internet, but there are now several free security applications on the Internet to choose from for all platforms.

2. Security suites

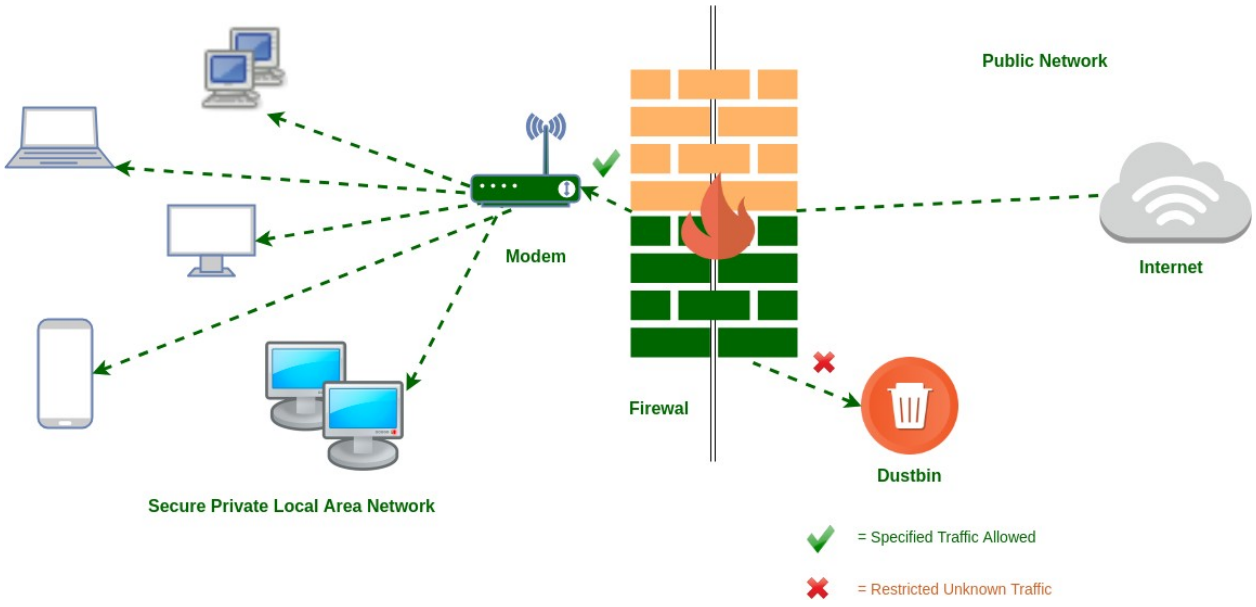
   So called “security suites” were first offered for sale in 2003 (McAfee) and contain a suite of firewalls, anti-virus, anti-spyware and more. They may now offer theft protection, portable storage device safety check, private Internet browsing, cloud anti-spam, a file shredder or make security-related decisions (answering popup windows) and several were free of charge as of at least 2012.

Firewall

A firewall can be defined as a special type of network security device or a software program that monitors and filters incoming and outgoing network traffic based on a defined set of security rules. It acts as a barrier between internal private networks and external sources (such as the public Internet).

The primary purpose of a firewall is to allow non-threatening traffic and prevent malicious or unwanted data traffic for protecting the computer from viruses and attacks. A firewall is a cyber

security tool that filters network traffic and helps users block malicious software from accessing the Internet in infected computers.



Firewall: Hardware or Software

This is one of the most problematic questions whether a firewall is a hardware or software. As stated above, a firewall can be a network security device or a software program on a computer. This means that the firewall comes at both levels, i.e., hardware and software, though it's best to have both.

Each format (a firewall implemented as hardware or software) has different functionality but the same purpose.

A hardware firewall is a physical device that attaches between a computer network and a gateway. For example, a broadband router.

On the other hand, a software firewall is a simple program installed on a computer that works through port numbers and other installed software.

Apart from that, there are cloud-based firewalls. They are commonly referred to as FaaS (firewall as a service). A primary advantage of using cloud-based firewalls is that they can be

managed centrally. Like hardware firewalls, cloud-based firewalls are best known for providing perimeter security.

Why Firewall

Firewalls are primarily used to prevent malware and network-based attacks. Additionally, they can help in blocking application-layer attacks. These firewalls act as a gatekeeper or a barrier. They monitor every attempt between our computer and another network. They do not allow data packets to be transferred through them unless the data is coming or going from a user-specified trusted source.

Firewalls are designed in such a way that they can react quickly to detect and counter-attacks throughout the network. They can work with rules configured to protect the network and perform quick assessments to find any suspicious activity. In short, we can point to the firewall as a traffic controller.

Some of the important risks of not having a firewall are:

Open Access

If a computer is running without a firewall, it is giving open access to other networks. This means that it is accepting every kind of connection that comes through someone. In this case, it is not possible to detect threats or attacks coming through our network. Without a firewall, we make our devices vulnerable to malicious users and other unwanted sources.

Lost or Comprised Data

Without a firewall, we are leaving our devices accessible to everyone. This means that anyone can access our device and have complete control over it, including the network. In this case, cybercriminals can easily delete our data or use our personal information for their benefit.

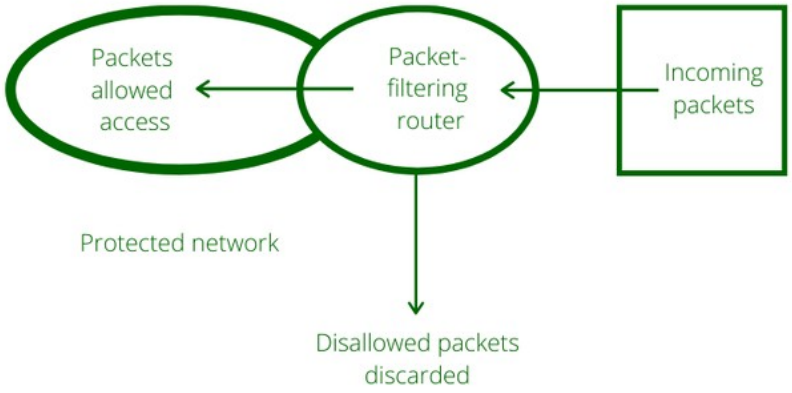
Network Crashes

In the absence of a firewall, anyone could access our network and shut it down. It may lead us to invest our valuable time and money to get our network working again.

Therefore, it is essential to use firewalls and keep our network, computer, and data safe and secure from unwanted sources.

Types of firewall

Packet Filters –



Packet filter firewall

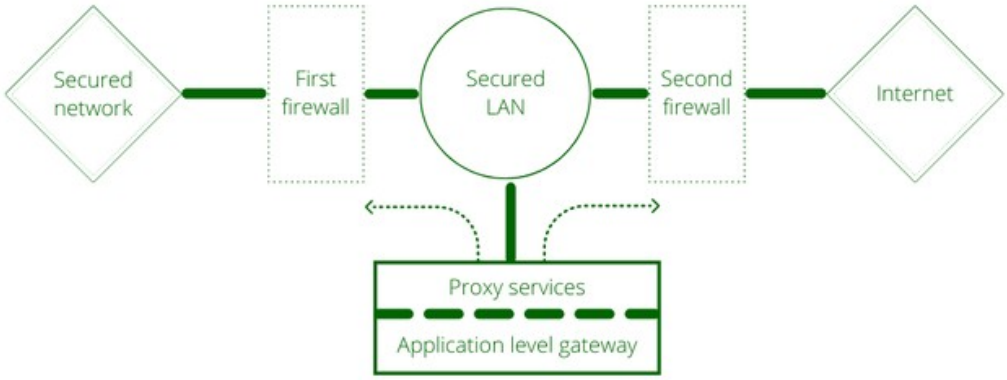
- It works in the network layer of the OSI Model. It applies a set of rules (based on the contents of IP and transport header fields) on each packet and based on the outcome, decides to either forward or discard the packet.
- Packet filter firewall controls access to packets on the basis of packet source and destination address or specific transport protocol type. It is done at the OSI (Open Systems Interconnection) data link, network, and transport layers. Packet filter firewall works on the network layer of the OSI model.
- Packet filters consider only the most basic attributes of each packet, and they don't need to remember anything about the traffic since each packet is examined in isolation. For this reason, they can decide packet flow very quickly.
- Example: Filter can be set to block all UDP segments and all Telnet connections. This type of configuration prevents outsiders from logging onto internal hosts using Telnet and insider from logging onto external hosts using Telnet connections.

Circuit-level Gateways

Circuit-level gateways are another simplified type of firewall that can be easily configured to allow or block traffic without consuming significant computing resources. These types of firewalls typically operate at the session-level of the OSI model by verifying **TCP (Transmission Control Protocol)** connections and sessions. Circuit-level gateways are designed to ensure that the established sessions are protected.

Typically, circuit-level firewalls are implemented as security software or pre-existing firewalls. Like packet-filtering firewalls, these firewalls do not check for actual data, although they inspect information about transactions. Therefore, if a data contains malware, but follows the correct TCP connection, it will pass through the gateway. That is why circuit-level gateways are not considered safe enough to protect our systems.

Application Gateways –



Application level gateway

- Application-level gateway is also called a bastion host. It operates at the application level. Multiple application gateways can run on the same host but each gateway is a separate server with its own processes.
- These firewalls, also known as application proxies, provide the most secure type of data connection because they can examine every layer of the communication, including the application data.

- Example: Consider FTP service. The FTP commands like getting the file, putting the file, listing files, and positioning the process at a particular point in a directory tree. Some system admin blocks put command but permits get command, list only certain files, or prohibit changing out of a particular directory. The proxy server would simulate both sides of this protocol exchange. For example, the proxy might accept get commands and reject put commands.

It works as follows:

Step-1: User contacts the application gateway using a TCP/IP application such as HTTP.

Step-2: The application gateway asks about the remote host with which the user wants to establish a connection. It also asks for the user id and password that is required to access the services of the application gateway.

Step-3: After verifying the authenticity of the user, the application gateway accesses the remote host on behalf of the user to deliver the packets.

Stateful Multi-layer Inspection (SMLI)

Stateful multi-layer inspection firewalls include both packet inspection technology and TCP handshake verification, making SMLI firewalls superior to packet-filtering firewalls or circuit-level gateways. Additionally, these types of firewalls keep track of the status of established connections.

In simple words, when a user establishes a connection and requests data, the SMLI firewall creates a database (state table). The database is used to store session information such as source IP address, port number, destination IP address, destination port number, etc. Connection information is stored for each session in the state table. Using stateful inspection technology, these firewalls create security rules to allow anticipated traffic.

In most cases, SMLI firewalls are implemented as additional security levels. These types of firewalls implement more checks and are considered more secure than stateless firewalls. This is why stateful packet inspection is implemented along with many other firewalls to track statistics for all internal traffic. Doing so increases the load and puts more pressure on

computing resources. This can give rise to a slower transfer rate for data packets than other solutions.

USES OF A FIREWALL

- The useful thing about a Firewall is that it prevents unauthorized remote access and remote control of your computer by a hacker for evil purposes.
- Data security is ensured based on IP address and protocol.
- It ensures continuity of operations and availability of information.
- It blocks destructive or unsuitable or pornographic content.
- It prevents ransomware, malware, and phishing attacks.
- It shields older PCs with earlier versions of OS.
- It is safer for online gamers.

Advantages of Firewall

- A Firewall prevents hackers and remote access.
- It protects data.
- It ensures better privacy and security.
- It protects from Trojans.
- A network-based Firewall, like a router, can offer protection to multiple systems, while an OS-based Firewall can protect individual systems.

Disadvantages of Firewall

- **Cost:** Installation of a Firewall can be costly depending on the sophistication required.
- **Performance:** This is affected as each packet has to be verified for authenticity before it is allowed into the network.
- **Virus and Malware:** There are a few limitations in a Firewall like its inability to prevent virus and malware attacks for which separate applications would be required, at the individual system level.
- A network-level Firewall might bring in a false sense of security in employees and make them slacken on securing individual systems. Companies need to make all employees understand the concept of a Firewall and the importance of a Firewall for information security and their responsibility.
- Firewall maintenance and up-gradation require extra manpower and resources.

How to use firewall protection

Proper setup and maintenance of your firewall are essential to keep your network and devices protected.

1. **Always update your firewalls as soon as possible:** Firmware patches keep your firewall updated against any newly discovered vulnerabilities. Personal and home firewall users can usually safely update immediately. Larger organizations may need to check configuration and compatibility across their network first. However, everyone should have processes in place to update promptly.
2. **Use antivirus protection:** Firewalls alone are not designed to stop viruses and other infections. These may get past firewall protections, and you'll need a security solution that's designed to disable and remove them. Kaspersky Total Security can protect you across your personal devices, and our many business security solutions can safeguard any network hosts you'll seek to keep clean.
3. **Limit accessible ports and hosts with a whitelist:** Default to connection denial for inbound traffic. Limit inbound and outbound connections to a strict whitelist of trusted IP addresses. Reduce user access privileges to necessities. It is easier to stay secure by enabling access when needed than to revoke and mitigate damage after an incident.
4. **Segmented network:** Lateral movement by malicious actors is a clear danger that can be slowed by limiting cross-communication internally.
5. **Have active network redundancies to avoid downtime:** Data backups for network hosts and other essential systems can prevent data loss and productivity during an incident.

Cryptography

Cryptography is used to secure and protect data during communication. It is helpful to prevent unauthorized person or group of users from accessing any confidential data. Encryption and decryption are the two essential functionalities of cryptography.

A message sent over the network is transformed into an unrecognizable encrypted message known as data encryption. At the receiving end, the received message is converted to its original form known as decryption.

Encryption

Encryption is the process in which a sender converts the original information to another form and transmits the resulting unintelligible message over the network. The sender requires an **encryption algorithm** and a **key** to transform the **plaintext** (original message) into a **cipher text** (encrypted message). Encryption is also known as **enciphering**.

Plaintext is the data that needs to be protected during transmission. The **cipher text** is the scrambled text produced as an outcome of the encryption algorithm for which a specific key is used. The cipher text is not shielded and it flows on the transmission channel. The **encryption algorithm** is a cryptographic algorithm that inputs plain text and an encryption key and produces a cipher text.

Decryption

Decryption inverts the encryption process in order to convert the message back to its original form. The receiver uses a decryption algorithm and a key to transform the cipher text back to the original plaintext. Decryption is also known as deciphering.

A decryption algorithm is a mathematical process utilized for decryption that generates original plaintext as an outcome of any given cipher text and decryption key. This is the reverse process of the encryption algorithm.

Types of Keys

Symmetric Key:

Symmetric-key encryptions are algorithms which use the same cryptographic keys for both encryption of plaintext and decryption of cipher text.

Asymmetric Key:

Asymmetric Key refers to those algorithms that make use of two pairs of the key for the purpose of encryption. The public key is made available to any random person, whereas the secret, a private key is made available only to the receiver of the message

Public Key:

Public key cryptography is an encryption system which is based on two pairs of keys. Public keys are used to encrypt messages for a receiver.

Private Key:

Private key may be part of a public/ private asymmetric key pair. It can be used in asymmetric encryption as you can use the same key to encrypt and decrypt data.

Pre-Shared Key:

In cryptography, a pre-shared key (PSK) is a shared secret which was earlier shared between the two parties using a secure channel before it is used.

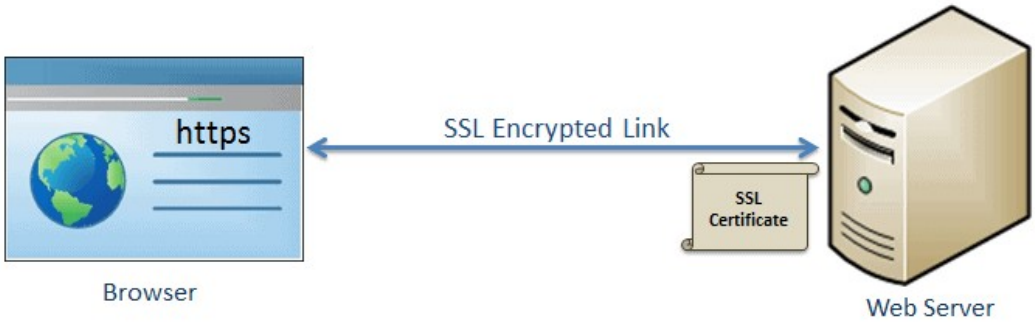
Difference between Encryption and Decryption

Parameter	Encryption	Decryption
What is	It is a process of converting normal data into an unreadable form. It helps you to avoid any unauthorized access to data	It is a method of converting the unreadable/coded data into its original form.
Process	Whenever the data is sent between two separate machines, it is encrypted automatically using a secret key.	The receiver of the data automatically allows you to convert the data from the codes into its original form.
Location of Conversion	The person who is sending the data to the destination.	The receiver receives the data and converts it.
Example	An employee is sending essential documents to his/her manager.	The manager is receiving the essential documents from his/her employee.
Use of Algorithm	The same algorithm with the same key is used for the encryption-decryption process.	The only single algorithm is used for encryption and decryption with a pair of keys where each use for encryption and decryption.

Secure Socket Layer (SSL)

SSL is the standard security technology for establishing an encrypted link between the two systems. These can be browser to server, server to server or client to server. Basically, SSL ensures that the data transfer between the two systems remains encrypted and private.

The https is essentially http over SSL. SSL establishes an encrypted link using an SSL certificate which is also known as a digital certificate.



SSL(secure socket layer)

How SSL works?

Https uses SSL protocol to secure the communication by transferring encrypted data. Before discussed how SSL works first we know:

SSL fundamentally works with the following concepts:

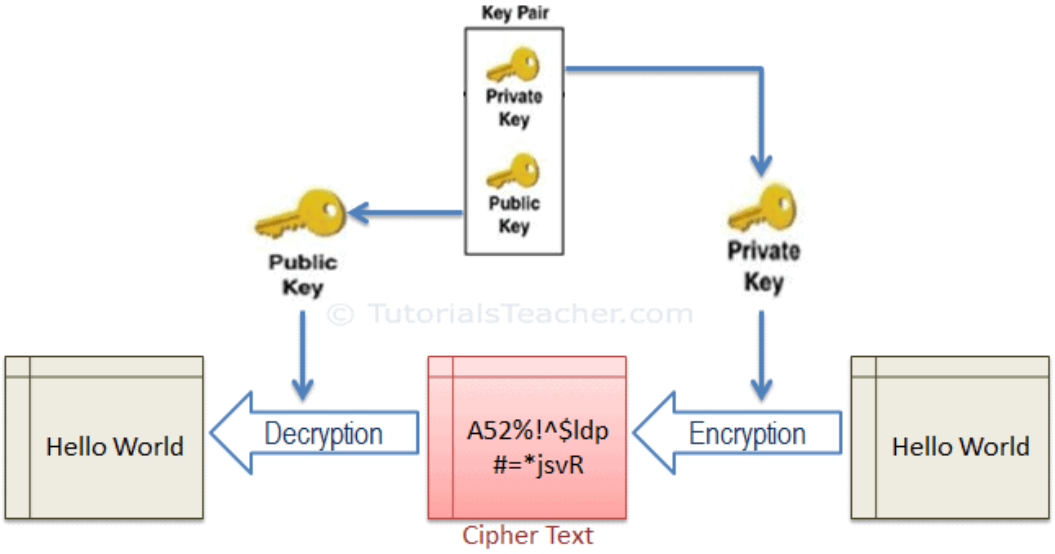
1. Asymmetric Cryptography
2. Symmetric Cryptography

Asymmetric Cryptography

Asymmetric cryptography (also known as Asymmetric Encryption or Public Key Cryptography) uses a mathematically-related key pair to encrypt and decrypt data. In a key pair, one key is shared with anyone who is interested in a communication. This is called **Public Key**. The other key in the key pair is kept secret and is called **Private Key**.

Here, the keys referred to a mathematical value and were created using a mathematical algorithm which encrypts or decrypts the data.

In the asymmetric cryptography, the data can be signed with a private key, which can only be decrypted using the related public key in a pair.



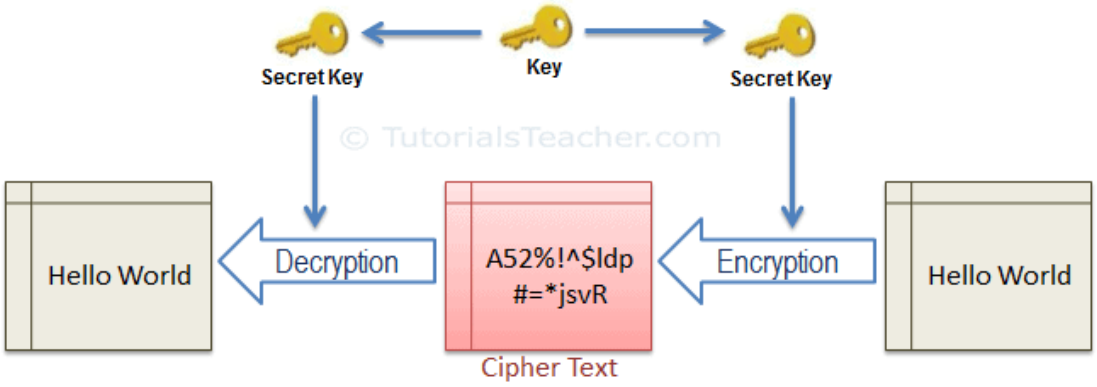
Asymmetric Cryptography

SSL uses asymmetric cryptography to initiate the communication which is known as SSL handshake.

Most commonly used asymmetric key encryption algorithms include EIGamal, RSA, DSA, Elliptic curve techniques and PKCS.

Symmetric Cryptography

In the symmetric cryptography, there is only one key which encrypts and decrypts the data. Both sender and receiver should have this key, which is only known to them.

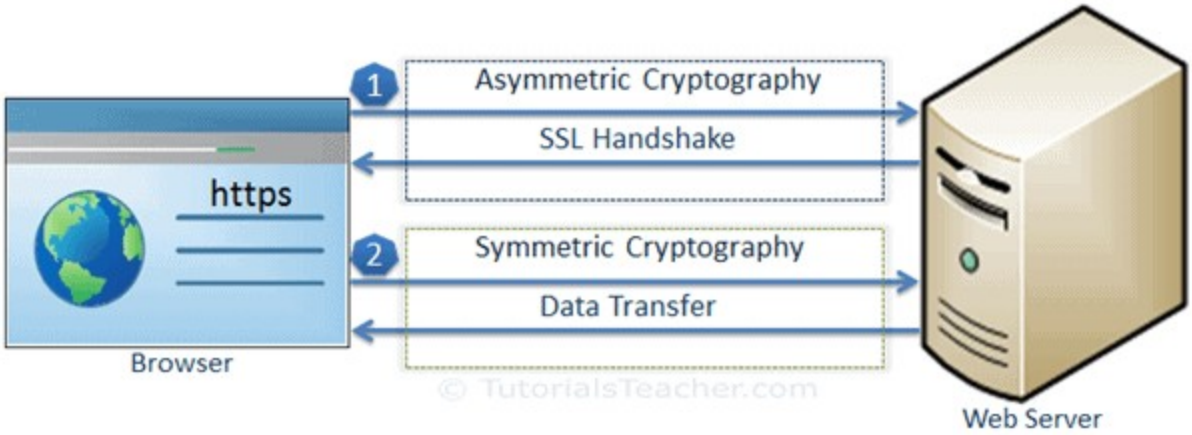


Symmetric Cryptography

SSL uses symmetric cryptography using the session key after the initial handshake is done. The most widely used symmetric algorithms are AES-128, AES-192 and AES-256.

Data Transfer over SSL

SSL protocol uses asymmetric and symmetric cryptography to transfer data securely. The following figure illustrates the steps of SSL communication:



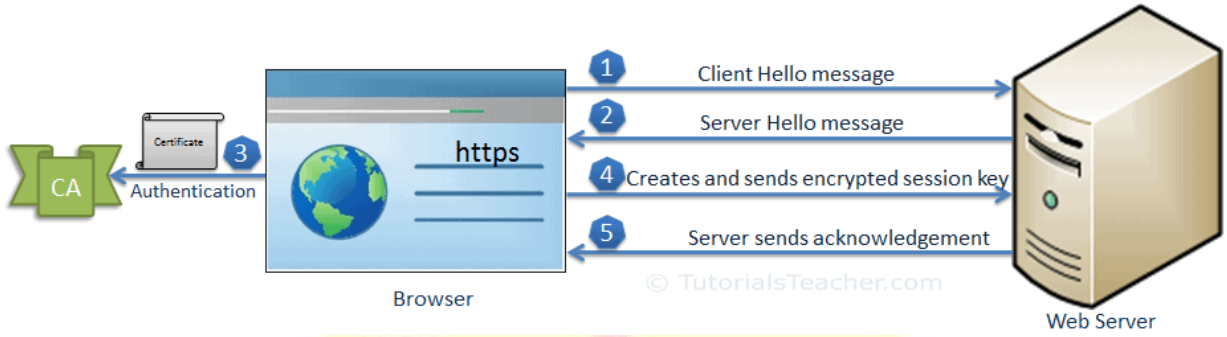
SSL Communication

In the above figure, SSL communication between the browser and the web server (or any other two systems) is mainly divided into two steps: the **SSL handshake and the actual data transfer**.

SSL Handshake

The communication over SSL always begins with the SSL handshake. The SSL handshake is an asymmetric cryptography which allows the browser to verify the web server, get the public key and establish a secure connection before the beginning of the actual data transfer.

The following figure illustrates the steps involved in the SSL handshake:



SSL Handshake

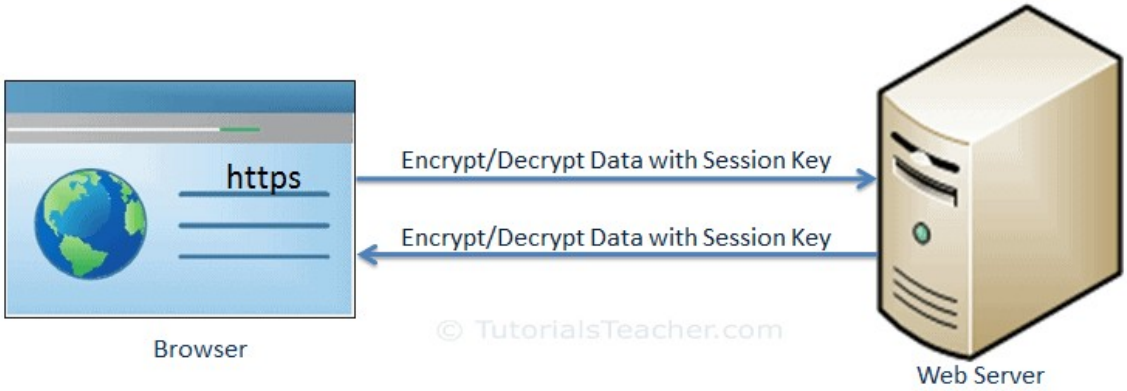
Let's understand the above steps:

1. The client sends a "client hello" message. This includes the client's SSL version number, cipher settings, session-specific data and other information that the server needs to communicate with the client using SSL.
2. The server responds with a "server hello" message. This includes the server's SSL version number, cipher settings, session-specific data, an SSL certificate with a public key and other information that the client needs to communicate with the server over SSL.
3. The client verifies the server's SSL certificate from CA (Certificate Authority) and authenticates the server. If the authentication fails, then the client refuses the SSL connection and throws an exception. If the authentication succeeds, then proceed to step 4.
4. The client creates a session key, encrypts it with the server's public key and sends it to the server. If the server has requested client authentication (mostly in server to server communication), then the client sends his own certificate to the server.
5. The server decrypts the session key with its private key and sends the acknowledgement to the client encrypted with the session key.

Thus, at the end of the SSL handshake, both the client and the server have a valid session key which they will use to encrypt or decrypt actual data. The public key and the private key will not be used any more after this.

Actual Data Transfer

The client and the server now use a shared session key to encrypt and decrypt actual data and transfer it. This is done using the same session key at both ends and so, it is a symmetric cryptography. The actual SSL data transfer uses symmetric cryptography because it is easy and takes less CUP consumption compared with the asymmetric cryptography.



SSL Data Transfer

Thus, SSL fundamentally works using asymmetric cryptography and symmetric cryptography. There are certain infrastructures involved in achieving SSL communication in real life, which are called Public Key Infrastructure .

Public Key Infrastructure

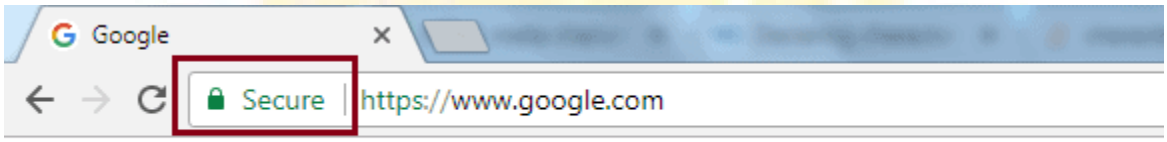
The public key infrastructure (PKI) is a set of roles, policies, and procedures needed to create, manage, distribute, use, store, and revoke digital certificates and manage public key encryption.

SSL Certificate

The SSL certificate (also known as digital certificate) plays an important role in securing the communication between two systems.

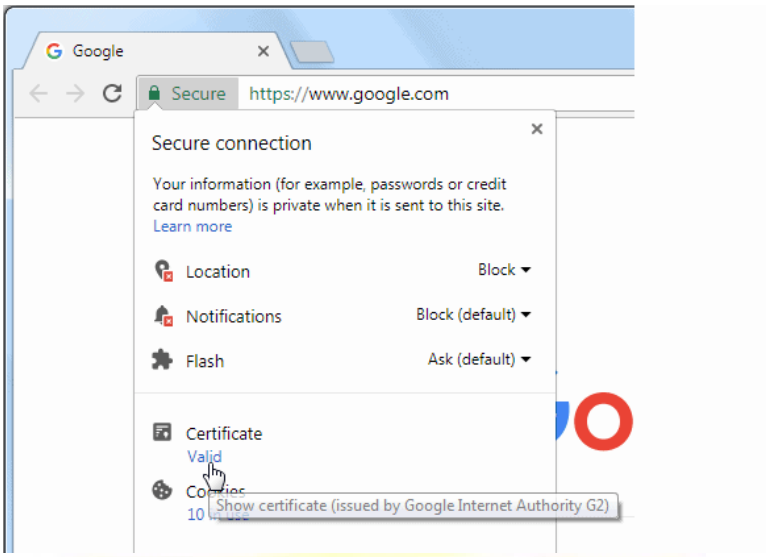
The SSL certificate is a data file issued by the authorised Certificate Authority (CA). As we know SSL uses asymmetric cryptography to establish an encrypted link between the two systems using a key pair (public key and private key). The SSL certificate contains the owner's public key and other details. The web server sends a public key to the browser through an SSL certificate and the browser verifies it and authenticates the web server using the SSL certificate.

You can open the certificate of any https website. For example, enter the url https://www.google.com in Google Chrome browser to check the SSL certificate of google.com. Any https website will have a padlock Secure in the addressbar, as shown below.



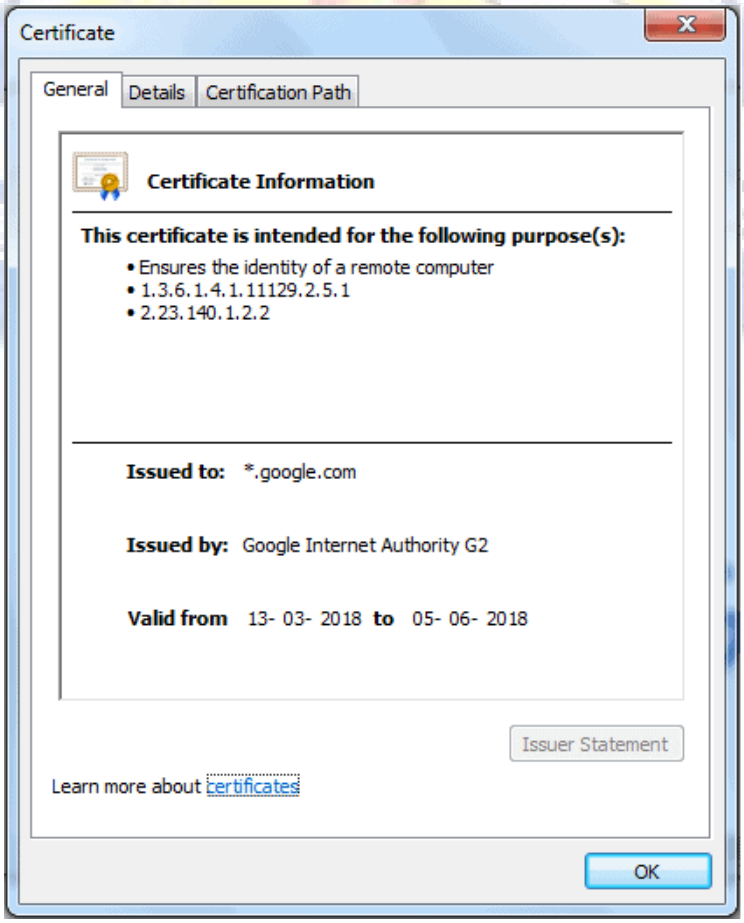
SSL in Browser

Click on the padlock symbol and click on Certificate, as shown below.



SSL Certificate

This will open the certificate as shown below.



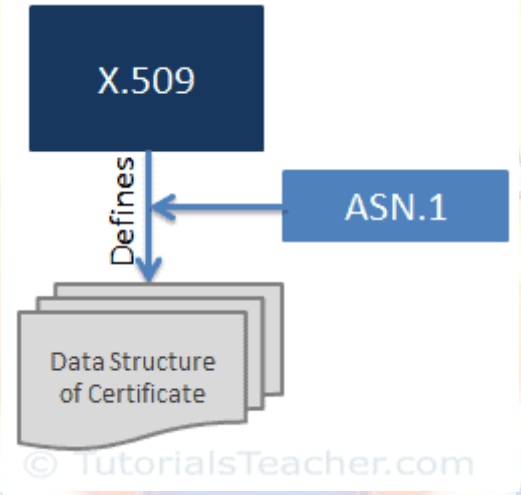
SSL Certificate

As you can see, in the General tab, the certificate displays Issued to, Issued by and Valid from and to dates. The Details tab includes other information. The Certificate Path tab includes information about all the intermediate certificates and the root CA certificate.

X.509

X.509 is a standard that defines the format of the digital certificate. SSL uses the X.509 format. In other words, SSL certificates are actually X.509 certificates.

X.509 uses a formal language called Abstract Syntax Notation One (ASN.1) to express the certificate's data structure.



SSL Certificate

The SSL certificate in X.509 format includes the following information:

- **Version:** The version number of the certificate data format as per X.509.
- **Serial number:** Unique identifier of the certificate assigned by CA
- **Public Key:** The owner's public key
- **Subject:** Owner's name, address, country and domain name
- **Issuer:** Name of the CA who issued the certificate
- **Valid-From:** The date the certificate is valid from
- **Valid-To:** The expiration date
- **Signature Algorithm:** The algorithm used to create the signature

- **Thumbprint:** The hash of the certificate
- **Thumbprint Algorithm:** The algorithm used to create a hash of the certificate

UNIT- 4
INTERNET APPLICATION

Email

Email is a service which allows us to send the message in electronic mode over the internet. It offers an efficient, inexpensive and real time mean of distributing information among people.

E-Mail Address

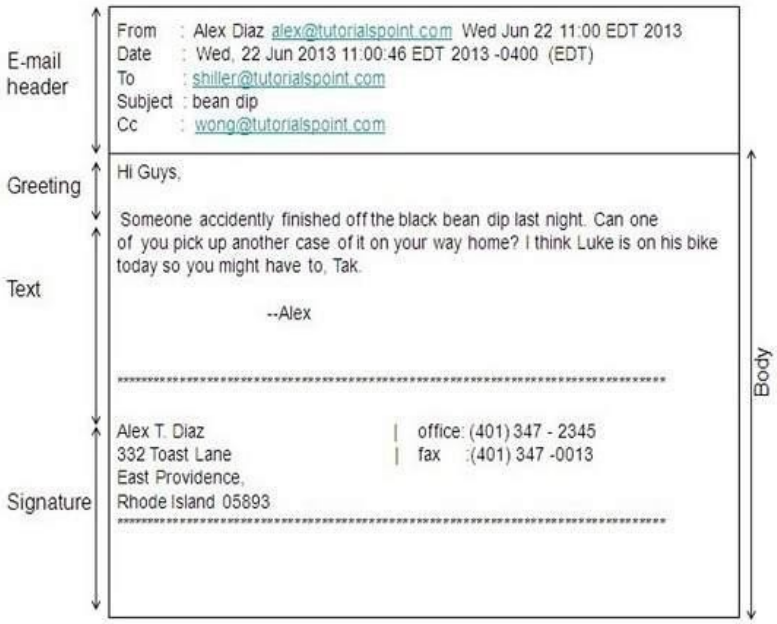
Each user of email is assigned a unique name for his email account. This name is known as E-mail address. Different users can send and receive messages according to the e-mail address.

E-mail is generally of the form username@domainname. For example, webmaster@tutorialspoint.com is an e-mail address where webmaster is username and tutorialspoint.com is domain name.

- The username and the domain name are separated by **@ (at)** symbol.
- E-mail addresses are not case sensitive.
- Spaces are not allowed in e-mail address

E-mail Message Components

E-mail message comprises of different components: E-mail Header, Greeting, Text, and Signature. These components are described in the following diagram:



E-mail Header

The first five lines of an E-mail message is called E-mail header. The header part comprises of following fields:

- From
- Date
- To
- Subject
- CC
- BCC

From :- The **From** field indicates the sender's address i.e. who sent the e-mail.

Date :- The **Date** field indicates the date when the e-mail was sent.

To :- The **To** field indicates the recipient's address i.e. to whom the e-mail is sent.

Subject :- The **Subject** field indicates the purpose of e-mail. It should be precise and to the point.

CC :- CC stands for Carbon copy. It includes those recipient addresses whom we want to keep informed but not exactly the intended recipient.

BCC :- BCC stands for Black Carbon Copy. It is used when we do not want one or more of the recipients to know that someone else was copied on the message.

Greeting :- Greeting is the opening of the actual message. Eg. Hi Sir or Hi Guys etc.

Text :- It represents the actual content of the message.

Signature :- This is the final part of an e-mail message. It includes Name of Sender, Address, and Contact Number.

Advantages

E-mail has prooved to be powerful and reliable medium of communication. The benefits of **E-mail** are:

- Reliable
- Convenience
- Speed
- Inexpensive
- Printable
- Global
- Generality

Reliable :-Many of the mail systems notify the sender if e-mail message was undeliverable.

Convenience :-There is no requirement of stationary and stamps. One does not have to go to post office. But all these things are not required for sending or receiving an mail.

Speed:-E-mail is very fast. However, the speed also depends upon the underlying network.

Inexpensive :-The cost of sending e-mail is very low.

Printable :-It is easy to obtain a hardcopy of an e-mail. Also an electronic copy of an e-mail can also be saved for records.

Global :-E-mail can be sent and received by a person sitting across the globe.

Generality:- It is also possible to send graphics, programs and sounds with an e-mail.

Disadvantages

- Forgery
- Overload
- Misdirection
- Junk
- No response

Forgery :- E-mail doesn't prevent from forgery, that is, someone impersonating the sender, since sender is usually not authenticated in any way.

Overload :-Convenience of E-mail may result in a flood of mail.

Misdirection :-It is possible that you may send e-mail to an unintended recipient.

Junk :-Junk emails are undesirable and inappropriate emails. Junk emails are sometimes referred to as spam.

No Response :-It may be frustrating when the recipient does not read the e-mail and respond on a regular basis.

E-mail Protocols

E-mail Protocols are set of rules that help the client to properly transmit the information to or from the mail server. Here in this tutorial, we will discuss various protocols such as **SMTP, POP,** and **IMAP.**

SMTP

SMTP stands for **Simple Mail Transfer Protocol**. It was first proposed in 1982. It is a standard protocol used for sending e-mail efficiently and reliably over the internet.

- SMTP is application level protocol.
- SMTP is connection oriented protocol.
- SMTP is text based protocol.

- It handles exchange of messages between e-mail servers over TCP/IP network.
- Apart from transferring e-mail, SMPT also provides notification regarding incoming mail.
- When you send e-mail, your e-mail client sends it to your e-mail server which further contacts the recipient mail server using SMTP client.
- These SMTP commands specify the sender's and receiver's e-mail address, along with the message to be send.
- The exchange of commands between servers is carried out without intervention of any user.
- In case, message cannot be delivered, an error report is sent to the sender which makes SMTP a reliable protocol.

SMTP Commands

The following table describes some of the SMTP commands:

S.N.	Command Description
1	**HELLO** This command initiates the SMTP conversation.
2	**EHELLO** This is an alternative command to initiate the conversation. ESMTP indicates that the sender server wants to use extended SMTP protocol.
3	**MAIL** **FROM** This indicates the sender's address.
4	**RCPT** **TO** It identifies the recipient of the mail. In order to deliver similar message to multiple users this command can be repeated multiple times.
5	**SIZE** This command let the server know the size of attached message in bytes.
6	**DATA** The **DATA** command signifies that a stream of data will follow. Here stream of data refers to the body of the message.

7	**QUIT** This commands is used to terminate the SMTP connection.
8	**VERFY** This command is used by the receiving server in order to verify whether the given username is valid or not.
9	**EXPN** It is same as VRFY, except it will list all the users name when it used with a distribution list.

IMAP

IMAP stands for **Internet Message Access Protocol.** It was first proposed in 1986. There exist five versions of IMAP as follows:

1. Original IMAP
2. IMAP2
3. IMAP3
4. IMAP2bis
5. IMAP4

- IMAP allows the client program to manipulate the e-mail message on the server without downloading them on the local computer.
- The e-mail is hold and maintained by the remote server.
- It enables us to take any action such as downloading, delete the mail without reading the mail.It enables us to create, manipulate and delete remote message folders called mail boxes.
- IMAP enables the users to search the e-mails.
- It allows concurrent access to multiple mailboxes on multiple mail servers.

IMAP Commands

The following table describes some of the IMAP commands:

S.N.	Command Description

1	**IMAP_LOGIN** This command opens the connection.
2	**CAPABILITY** This command requests for listing the capabilities that the server supports.
3	**NOOP** This command is used as a periodic poll for new messages or message status updates during a period of inactivity.
4	**SELECT** This command helps to select a mailbox to access the messages.
5	**EXAMINE** It is same as SELECT command except no change to the mailbox is permitted.
6	**CREATE** It is used to create mailbox with a specified name.
7	**DELETE** It is used to permanently delete a mailbox with a given name.
8	**RENAME** It is used to change the name of a mailbox.
9	**LOGOUT** This command informs the server that client is done with the session. The server must send BYE untagged response before the OK response and then close the network connection.

POP

POP stands for Post Office Protocol. It is generally used to support a single client. There are several versions of POP but the POP 3 is the current standard.

- POP is an application layer internet standard protocol.
- Since POP supports offline access to the messages, thus requires less internet usage time.
- POP does not allow search facility.

- In order to access the messaged, it is necessary to download them.
- It allows only one mailbox to be created on server.
- It is not suitable for accessing non mail data.
- POP commands are generally abbreviated into codes of three or four letters. Eg. STAT.

POP Commands

The following table describes some of the POP commands:

S.N.	Command Description
1	**LOGIN** This command opens the connection.
2	**STAT** It is used to display number of messages currently in the mailbox.
3	**LIST** It is used to get the summary of messages where each message summary is shown.
4	**RETR** This command helps to select a mailbox to access the messages.
5	**DELE** It is used to delete a message.
6	**RSET** It is used to reset the session to its initial state.
7	**QUIT** It is used to log off the session.

Comparison between POP and IMAP

S.N.	POP	IMAP
1	Generally used to support single client.	Designed to handle multiple clients.

2	Messages are accessed offline.	Messages are accessed online although it also supports offline mode.
3	POP does not allow search facility.	It offers ability to search emails.
4	All the messages have to be downloaded.	It allows selective transfer of messages to the client.
5	Only one mailbox can be created on the server.	Multiple mailboxes can be created on the server.
6	Not suitable for accessing non-mail data.	Suitable for accessing non-mail data i.e. attachment.
7	POP commands are generally abbreviated into codes of three or four letters. Eg. STAT.	IMAP commands are not abbreviated, they are full. Eg. STATUS.
8	It requires minimum use of server resources.	Clients are totally dependent on server.
9	Mails once downloaded cannot be accessed from some other location.	Allows mails to be accessed from multiple locations.
10	The e-mails are not downloaded automatically.	Users can view the headings and sender of e-mails and then decide to download.
10	POP requires less internet usage time.	IMAP requires more internet usage time.

Telnet

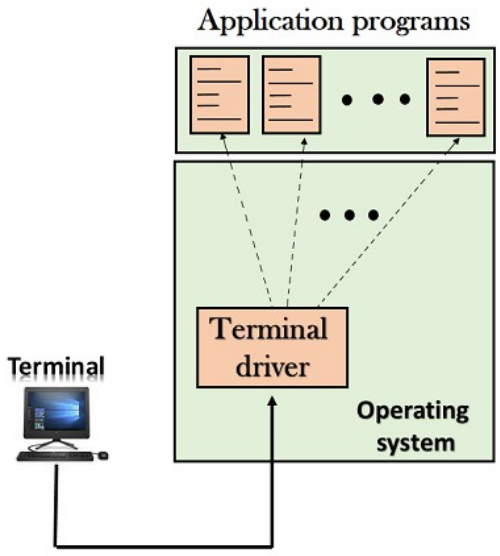
- The main task of the internet is to provide services to users. For example, users want to run different application programs at the remote site and transfers a result to the local site. This requires a client-server program such as FTP, SMTP. But this would not allow us to create a specific program for each demand.
- The better solution is to provide a general client-server program that lets the user access any application program on a remote computer. Therefore, a program that allows a user to log on

to a remote computer. A popular client-server program Telnet is used to meet such demands. Telnet is an abbreviation for **Terminal Network**.

- Telnet provides a connection to the remote computer in such a way that a local terminal appears to be at the remote side.

Types of Login:

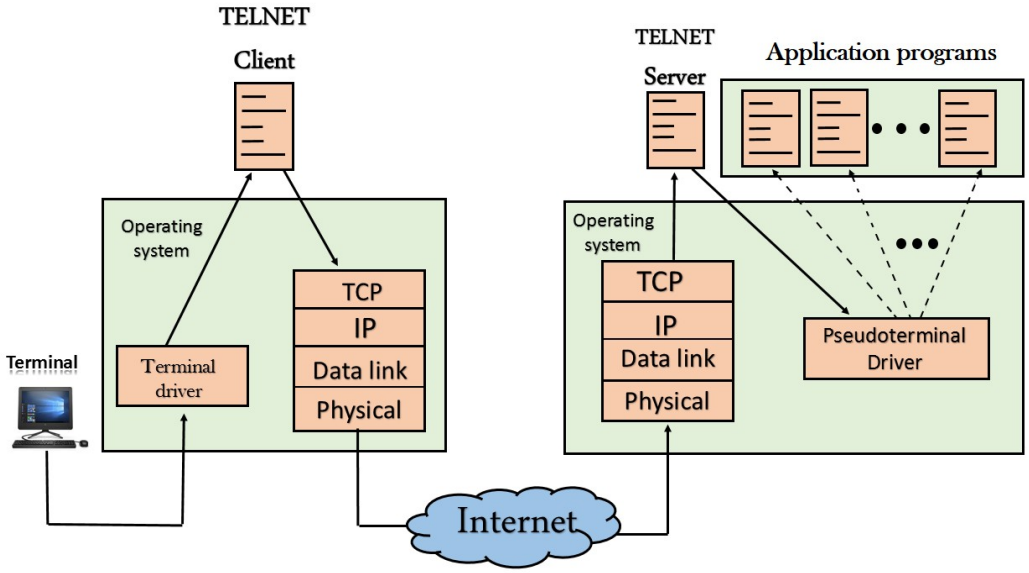
Local Login



- **When a user logs into a local computer, then it is known as local login.**
- **When the workstation running terminal emulator, the keystrokes entered by the user are accepted by the terminal driver. The terminal driver then passes these characters to the operating system which in turn, invokes the desired application program.**
- **However, the operating system has special meaning to special characters. For example, in UNIX some combination of characters have special meanings such**

as control character with "z" means suspend. Such situations do not create any problem as the terminal driver knows the meaning of such characters. But, it can cause the problems in remote login.

Remote login



- When the user wants to access an application program on a remote computer, then the user must perform remote login.

How remote login occurs

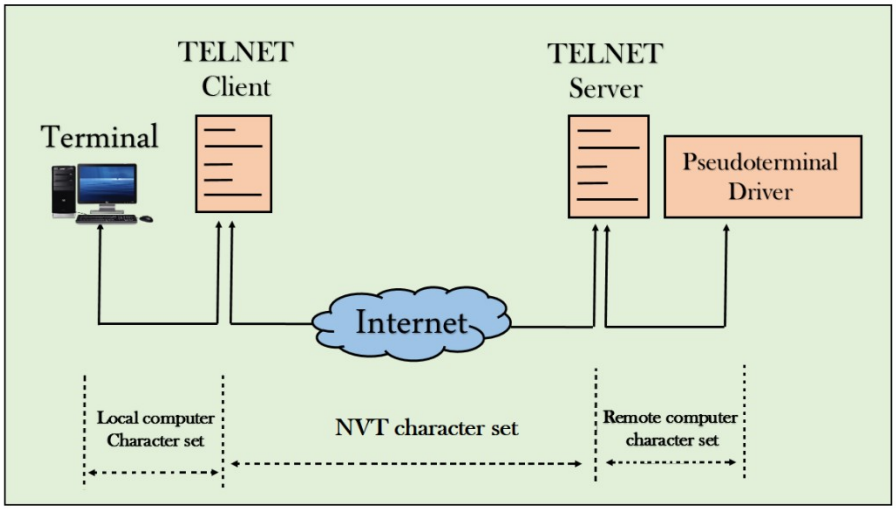
At the local site

The user sends the keystrokes to the terminal driver, the characters are then sent to the TELNET client. The TELNET client which in turn, transforms the characters to a universal character set known as network virtual terminal characters and delivers them to the local TCP/IP stack

At the remote site

The commands in NVT forms are transmitted to the TCP/IP at the remote machine. Here, the characters are delivered to the operating system and then pass to the TELNET server. The TELNET server transforms the characters which can be understandable by a remote computer. However, the characters cannot be directly passed to the operating system as a remote operating system does not receive the characters from the TELNET server. Therefore it requires some piece of software that can accept the characters from the TELNET server. The operating system then passes these characters to the appropriate application program.

Network Virtual Terminal (NVT)



- The network virtual terminal is an interface that defines how data and commands are sent across the network.
- In today's world, systems are heterogeneous. For example, the operating system accepts a special combination of characters such as end-of-file token running a DOS operating system *ctrl+z* while the token running a UNIX operating system is *ctrl+d*.
- TELNET solves this issue by defining a universal interface known as network virtual interface.
- The TELNET client translates the characters that come from the local terminal into NVT form and then delivers them to the network. The Telnet server then translates the data from NVT form into a form which can be understandable by a remote computer.

FTP

- FTP stands for File transfer protocol.
- FTP is a standard internet protocol provided by TCP/IP used for transmitting the files from one host to another.
- It is mainly used for transferring the web page files from their creator to the computer that acts as a server for other computers on the internet.
- It is also used for downloading the files to computer from other servers.

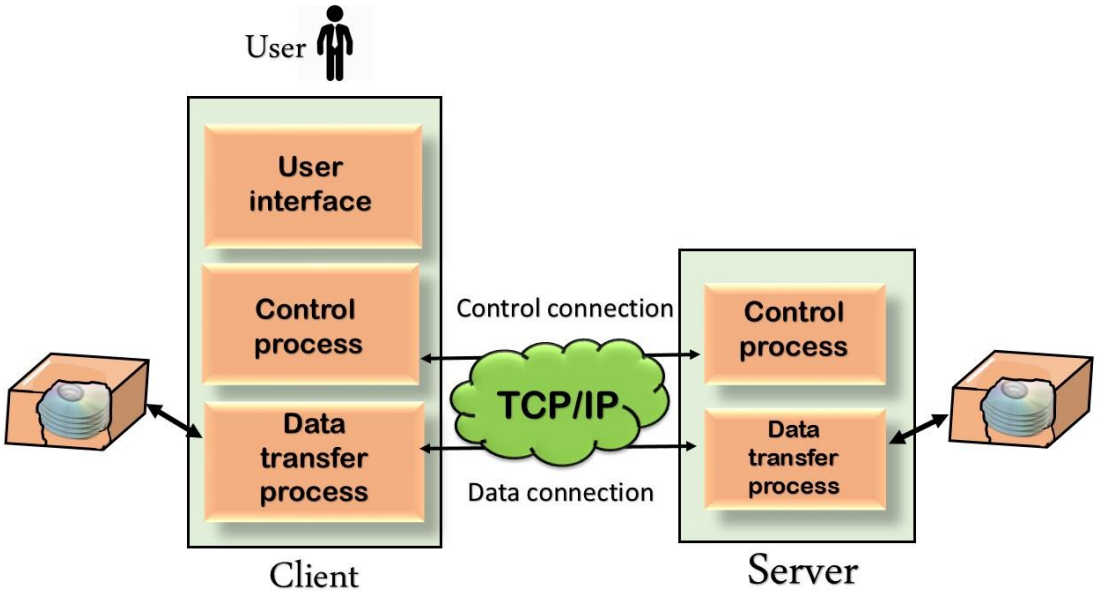
Objectives of FTP

- It provides the sharing of files.
- It is used to encourage the use of remote computers.
- It transfers the data more reliably and efficiently.

Why FTP?

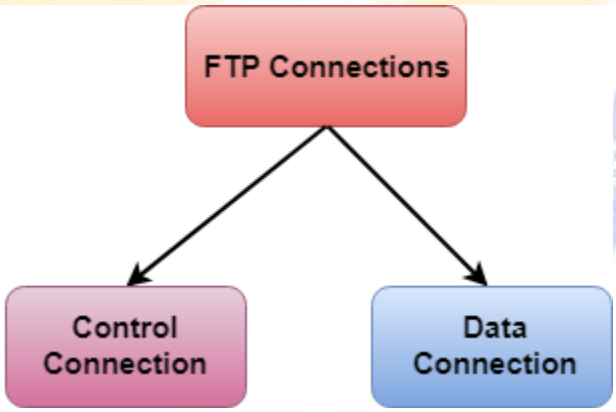
Although transferring files from one system to another is very simple and straightforward, but sometimes it can cause problems. For example, two systems may have different file conventions. Two systems may have different ways to represent text and data. Two systems may have different directory structures. FTP protocol overcomes these problems by establishing two connections between hosts. One connection is used for data transfer, and another connection is used for the control connection.

Mechanism of FTP



The above figure shows the basic model of the FTP. The FTP client has three components: the user interface, control process, and data transfer process. The server has two components: the server control process and the server data transfer process.

There are two types of connections in FTP:



- **Control Connection:** The control connection uses very simple rules for communication. Through control connection, we can transfer a line of command or line of response at a time. The control connection is made between the control processes. The control connection remains connected during the entire interactive FTP session.
- **Data Connection:** The Data Connection uses very complex rules as data types may vary. The data connection is made between data transfer processes. The data connection opens when a command comes for transferring the files and closes when the file is transferred.

FTP Clients

- FTP client is a program that implements a file transfer protocol which allows you to transfer files between two hosts on the internet.
- It allows a user to connect to a remote host and upload or download the files.
- It has a set of commands that we can use to connect to a host, transfer the files between you and your host and close the connection.
- The FTP program is also available as a built-in component in a Web browser. This GUI based FTP client makes the file transfer very easy and also does not require to remember the FTP commands.

Advantages of FTP:

1. **Speed:** One of the biggest advantages of FTP is speed. The FTP is one of the fastest way to transfer the files from one computer to another computer.
2. **Efficient:** It is more efficient as we do not need to complete all the operations to get the entire file.
3. **Security:** To access the FTP server, we need to login with the username and password. Therefore, we can say that FTP is more secure.
4. **Back & forth movement:** FTP allows us to transfer the files back and forth. Suppose you are a manager of the company, you send some information to all the employees, and they all send information back on the same server.

Disadvantages of FTP:

- The standard requirement of the industry is that all the FTP transmissions should be encrypted. However, not all the FTP providers are equal and not all the providers offer encryption. So, we will have to look out for the FTP providers that provides encryption.
- FTP serves two operations, i.e., to send and receive large files on a network. However, the size limit of the file is 2GB that can be sent. It also doesn't allow you to run simultaneous transfers to multiple receivers.
- Passwords and file contents are sent in clear text that allows unwanted eavesdropping. So, it is quite possible that attackers can carry out the brute force attack by trying to guess the FTP password.
- It is not compatible with every system.

Newsgroup

A newsgroup is an online discussion forum accessible through Usenet. Each newsgroup contains discussions about a specific topic, indicated in the newsgroup name. You can browse newsgroups and post or reply to topics using a newsreader program. Access to newsgroups also requires a Usenet subscription. Most Usenet providers offer monthly access for around $10 USD per month.

Newsgroups may be either moderated or unmoderated. In a moderated newsgroup, a moderator must approve posts in order for them to become part of the discussion. In an unmoderated group, everything posted is included in the discussion. Some newsgroups may also use bots to moderate the content, automatically eliminating posts that are deemed offensive or off topic.

While many people now use web forums and online chat instead of newsgroups, the service is still popular around the world. In fact, there are estimated to be over 100,000 newsgroups in existence. While many newsgroups host traditional text-based discussions, a large number of newsgroups are now used for file sharing. These newsgroups, which primarily provide links to files, often have the term "binaries" in their name.

Newsgroup Examples

Below are some examples of active newsgroups. The first part of the name (before the first dot) is the primary category (or hierarchy) or the newsgroup. For example, sci. is used for science-related discussions.

- alt.politics
- talk.religion
- sci.physics
- comp.software.testing
- alt.binaries.documentaries
- alt.binaries.multimedia.comedy

You can browse discussions and post to newsgroups using a newsreader.

Chat room

A **chat room** is a location where multiple people can all talk at the same time. When in a chat room anything said is seen by all people participating in the chat room. To send a message to only one person in a chat room a PM (private message) or DM (direct message) can be sent.
When a registered chat room user types a message into a text box, it will be visible to other users almost immediately.
The term can mean online chatting, instant messaging and online forums using either synchronous or asynchronous conferencing. Some chat rooms require a username and password combination in order to log in or join a conversation, allowing for privacy among the users.

Internet Relay Chat (IRC)

Internet Relay Chat (IRC) is Internet application that was developed by **Jakko Oikarinen** in Finland.
Chat is most convenient immediate way to communicate with others via Internet.
There are number of topics called “channels” through which you can chat with many people all over world.

Model used for IRC :

IRC follows client-server model. It means that both client and server software is required in order to use it. Many IT (Information Technology) clients are available for different kinds of computers, so whether you have PC, Macintosh, or UNIX work-section, you will be able to use IRC.

Chatting on IRC :

IRC client connects/communicates with IRC server on Internet. First, you have to log on to server using client and then pick channel on which you want to chat. They are sent to your server when you type words on your keyboard. Now your server is part of global IRC server network. Your server sends your messages to other servers, which in turn, sends your messages to people who are part of your channel. They can then read and respond to your messages. Many websites use proprietary chat software that does not use IRC protocol but enables you to chat when you are on site. There is another kind of chat, called **Instant Messaging**. In this kind of chatting, you communicate privately, one-to-one, with another person. You can create special lists so that you are informed when your “buddies” come online, ready to chat, and they are informed when you come online.

Working on IRC :

When you want to chat first you have to make connection to Internet and then start your client software. After that, you need to log on to IRC server which is located on Internet. There are many IRC servers are located all over world. Those IRC servers are connected together in network so that they can communicate with each other.

Servers are connected in spanning tree fashion architecture. In this, each server is connected to several others, but these servers are not directly connected to one another. When you connect to server, first you have to choose specific channel to join and choose user name to identify yourself when you at chat. Many channels are available that cover different topics.

Your message is sent from client software on your PC to IRC server to which you are connected. Then message is sent from one server to other servers where all users on this channel are logged on.

In this network, message is transferred from one server to another server. Under spanning-tree server architecture, message always takes shortest route through network to reach its final destination. Each server sends messages to client software of their respected users who are connected to channel/network. Then people/users can read and respond to your message at their computers.

Client Software for Chat :

CHAT TYPE	SOFTWARE REQUIRED
Internet Relay Chat (IRC)	Chat program such as mIRC or Ircle
Web-based chat	Web browsers like Netscape or Internet Explorer
AOL (America Online) Chat	AOL access program for America online
Direct Chat Programs	ICQ, AOL Instant Messenger, or other program
Online Conferencing	Conferencing program (CU-SeeMe, Netscape conference, Netmeeting)

Chat Servers :

Communication servers permit you to give your information to large number of users in environment that is just like Internet newsgroups. The most advanced servers have recently started augmenting text-based medium of conversation with dynamic voice and video support.
There are three major types of communication servers :

1. EFnet servers
2. UnderNet Servers
3. DALnet servers

Each server has its hostname, which mostly consists of name of server and Internet that it accesses. As servers already might have maximum number of user connections, you may not be allowed to connect server of your choice.

Smileys :

When we talk to people face-to-face, tone of your voice and our facial expressions impart great meaning to what we say. You can personalise your written messages by using smileys you create with your keyboard. Maximum time main use of smileys is to indicate joke. When text might not be clear.
There are different types of smileys which are as follows :

1. Basic Smileys
2. Widely Used Smileys
3. Midget Smileys
4. Mega Smileys
5. Usenet Smileys
6. Emotional Smileys

Video Conferencing

Video conferencing refers to conducting a video conference or video teleconference in which two or more sets of hardware and software interact while simultaneously transmitting and receiving video and audio signals from two or more geographic locations. Video conferencing may also involve sharing documents, various presentation materials, whiteboards, flip charts and similar group presentation visual aids. A telepresence system is most often used at the corporate or enterprise level and

represents high-end video conferencing systems. Video conferencing is differentiated from video phone calls, which serve individuals as opposed to a conference. Video conferencing is also known as online video conferencing and PC video conferencing.

Ecommerce

Ecommerce, also known as electronic commerce or internet commerce, refers to the buying and selling of goods or services using the internet, and the transfer of money and data to execute these transactions. Ecommerce is often used to refer to the sale of physical products online, but it can also describe any kind of commercial transaction that is facilitated through the internet.However, eCommerce can also entail other types of activities, such as online auctions, payment gateways, online ticketing, and internet banking.

Whereas e-business refers to all aspects of operating an online business, ecommerce refers specifically to the transaction of goods and services.

Types of Ecommerce Models

There are four main types of ecommerce models that can describe almost every transaction that takes place between consumers and businesses.

1. **Business to Consumer (B2C):**
   When a business sells a good or service to an individual consumer (e.g. You buy a pair of shoes from an online retailer).

2. Business to Business (B2B):

When a business organization sells a good or service to another business organization (e.g. A business sells software-as-a-service for other businesses to use)

3. Consumer to Consumer (C2C):

When a consumer sells a good or service to another consumer (e.g. You sell your old furniture on eBay to another consumer).

4. Consumer to Business (C2B):

When a consumer sells their own products or services to a business or organization (e.g. An influencer offers exposure to their online audience in exchange for a fee, or a photographer licenses their photo for a business to use).

Examples of Ecommerce

Ecommerce can take on a variety of forms involving different transactional relationships between businesses and consumers, as well as different objects being exchanged as part of these transactions.

1. **Retail:**

The sale of a product by a business directly to a customer without any intermediary.

2. Wholesale:

The sale of products in bulk, often to a retailer that then sells them directly to consumers.

3. Dropshipping:

The sale of a product, which is manufactured and shipped to the consumer by a third party.

4. Crowdfunding:

The collection of money from consumers in advance of a product being available in order to raise the startup capital necessary to bring it to market.

5. Subscription:

The automatic recurring purchase of a product or service on a regular basis until the subscriber chooses to cancel.

6. **Physical products:**

Any tangible good that requires inventory to be replenished and orders to be physically shipped to customers as sales are made.

7. Digital products:

Downloadable digital goods, templates, and courses, or media that must be purchased for consumption or licensed for use.

8. Services:

A skill or set of skills provided in exchange for compensation. The service provider's time can be purchased for a fee.

Advantages and Disadvantages of Ecommerce

E-commerce offers consumers the following advantages:

- **Convenience**: E-commerce can occur 24 hours a day, seven days a week.
- **Increased selection**: Many stores offer a wider array of products online than they carry in their brick-and-mortar counterparts. And many stores that solely exist online may offer consumers exclusive inventory that is unavailable elsewhere.

But there are certain drawbacks that come with ecommerce sites, too. The disadvantages include:

- **Limited customer service**: If you shop online for a computer, you cannot simply ask an employee to demonstrate a particular model's features in person. And although some websites let you chat online with a staff member, this is not a typical practice.
- **Lack of instant gratification**: When you buy an item online, you must wait for it to be shipped to your home or office. However, e-tailers like Amazon make the waiting game a little bit less painful by offering same-day delivery as a premium option for select products.
- **Inability to touch products**: Online images do not necessarily convey the whole story about an item, and so e-commerce purchases can be unsatisfying when the products received do not match consumer expectations. Case in point: an item of clothing may be made from shoddier fabric than its online image indicates.

UNIT – 6
WEBPAGE

A webpage is a document written in HTML and can be viewed on any web browser. It is contained within the web server, which can be accessed by entering the URL for that web page, and once it is loaded, it appears on the user's web browser. Each webpage is linked with a unique URL; hence two pages cannot have the same URL.

A webpage may contain ***text, links for other pages, graphics, videos, etc***. Moreover, it is mainly used to provide information to the user in text, images, etc.

A webpage is a part of a website; it means a website contains different web pages. Such as **javaTpoint.com** is a website, and the page currently you are accessing is the webpage. It can be understood as an example of a book. So, a Website is like a complete book, and a webpage is like a page of that book.

The WWW or Internet contains millions of web pages, and daily, a lot is being added. Tim Berners-Lee developed **the first webpage.**

Characteristics of a Web Page

- A simple webpage can be created very quickly.
- It takes very little time to create a webpage compared to a Website.
- A web page and a website should be compatible with any device, such as Mobile, Desktop, Laptop, etc.
- The search engine provides a web page through a link, and when a user clicks on that link, it is redirected to the webpage of a website.
- A webpage can have any type of information including videos, and audios.
- It can be made up of only HTML(Hypertext Markup Language), or CSS, or JavaScript for dynamic and attractive behavior.

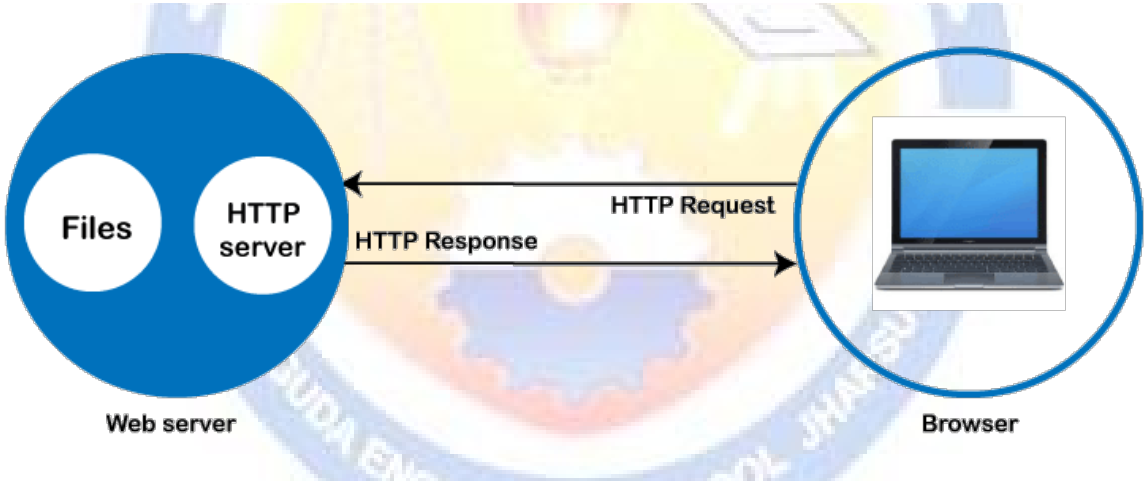
How does a Web Page Work?

A simple web page is created using HTML, which is a markup language. However, we can also use CSS and JavaScript to add more functionalities and make it more attractive.

It is created using HTML, hence containing different markup tags that specify how the data should be formatted on screen.

The webpage is contained within the webserver. To load this webpage, a client sends the request to the server, and generally, the browser is known as the client, which can request the page on the internet.

The web browser requests the page on the internet. Once it is responded to by the server, the browser interprets the markup tags and displays them on the user's screen in the correct format.



The browser sends the request for a page or a file via **an HTTP request.** The HTTP is the **Hypertext Transfer Protocol**, a network protocol that allows transferring hypermedia documents over the internet between a browser and server.

Once the request reaches the server, the HTTP server accepts the request, finds the requested page, and sends it back to the browser through **the HTTP response**. If a server is unable to find the requested page, it returns a **404 response**.

Elements of a Webpage

The main element of the webpage is a text file composed of HTML. Apart from this, a webpage can also have the following elements:

- **CSS:** The CSS code is used to make the page more interactive and control its look and feel.
- **Scripts:** The JavaScript code is included in a webpage to add interactivity to the page and add more functionalities to it.
- **Media:** It is used to include media components such as audio, video, and images.

How to Create a Simple Webpage?

Creating a simple webpage is very easy; anyone with basic knowledge of computers and HTML can create it. But before creating a webpage, you should be aware of the below points:

- A simple webpage can be created using HTML code only. Such pages are simple but not interactive and have very few functionalities.
- To make your webpage interactive and add functionality, you need to learn and use scripting languages, such as **PHP, Python**, etc.
- A web page can also be created using Notepad, but it is recommended to use IDEs for advanced uses such as **Atom, Sublime Text editor, PyCharm,**

Follow the below steps to create your webpage:

1. Open the Notepad application on your computer.
2. Write Below code on it.

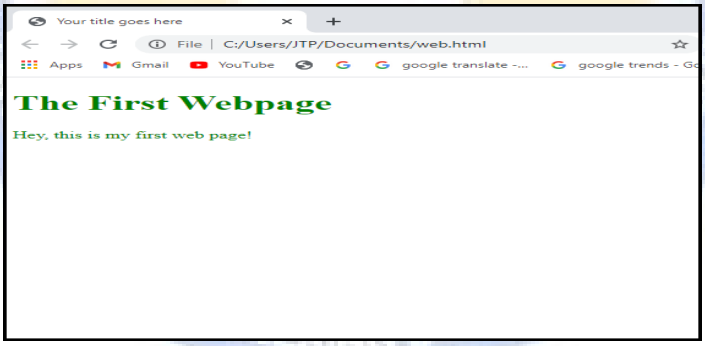
```
<!DOCTYPE html>
<html>
    <head>
        <title>Your title goes here</title>
    </head>
    <body bgcolor="white" text="red">
        <h1>first Web page</h1>
        <p>This is my first web page!</p>
```

```
    </body>
</html>
```

In the above code, the following tags are used:

- **<!DOCTYPE html>:** It is used for document type declaration, which means which version of HTML you are using. It indicates the browser that which language it is supposed to interpret.
- **<html>:** it indicates the start and of the HTML code.
- **<head>:** It can have different types of information such as title, meta tag, etc.; this information will not appear on the webpage.
- **<body>:** This tag contains other tags on the webpage, and users can see them. In above code, we have included <h1> heading tag and <p> paragraph tags.

3. Save the file with any name and **.html extension**. For example, save it with **html** name.
4. Go to the saved file (web.html), double click or right-click and run it.
5. It will open on your default browser and will display the below output:



You can also add more tags for different elements, such as add images, background images, border, table, table, etc., using HTML. You can learn all these from here.

After creating the page, you can also make changes in your file through the editor. Just make the change, save the file again, and reload the page; those changes will appear on the screen.

HTML

HTML is an acronym which stands for **Hyper Text Markup Language** which is used for creating web pages and web applications. Let's see what is meant by Hypertext Markup Language, and Web page.

Hyper Text: HyperText simply means "Text within Text." A text has a link within it, is a hypertext. Whenever you click on a link which brings you to a new webpage, you have clicked on a hypertext. HyperText is a way to link two or more web pages (HTML documents) with each other.

Markup language: A markup language is a computer language that is used to apply layout and formatting conventions to a text document. Markup language makes text more interactive and dynamic. It can turn text into images, tables, links, etc.

Web Page: A web page is a document which is commonly written in HTML and translated by a web browser. A web page can be identified by entering an URL. A Web page can be of the static or dynamic type. **With the help of HTML only, we can create static web pages**.

Hence, HTML is a markup language which is used for creating attractive web pages with the help of styling, and which looks in a nice format on a web browser. An HTML document is made of many HTML tags and each HTML tag contains different content.

Let's see a simple example of HTML.

```
<!DOCTYPE>
<html>
    <head>
        <title>Web page title</title>
    </head>
    <body>
        <h1>Write Your First Heading</h1>
        <p>Write Your First Paragraph.</p>
    </body>
</html>
```

Description of HTML Example

<!DOCTYPE>: It defines the document type or it instruct the browser about the version of HTML.

<html > :This tag informs the browser that it is an HTML document. Text between html tag describes the web document. It is a container for all other elements of HTML except <!DOCTYPE>

<head>: It should be the first element inside the <html> element, which contains the metadata(information about the document). It must be closed before the body tag opens.

<title>: As its name suggested, it is used to add title of that HTML page which appears at the top of the browser window. It must be placed inside the head tag and should close immediately. (Optional)

<body> : Text between body tag describes the body content of the page that is visible to the end user. This tag contains the main content of the HTML document.

<h1> : Text between <h1> tag describes the first level heading of the webpage.

<p> : Text between <p> tag describes the paragraph of the webpage.

Building blocks of HTML

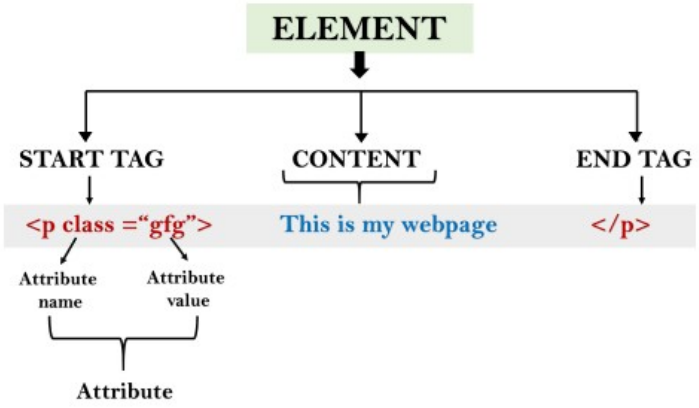
An HTML document consist of its basic building blocks which are:

- **Tags:** An HTML tag surrounds the content and apply meaning to it. It is written between < and > brackets.
- **Attribute:** An attribute in HTML provides extra information about the element, and it is applied within the start tag. An HTML attribute contains two fields: name & value.

Syntax

1. <tag name attribute_name= " attr_value"> content </ tag name>

- **Elements:** An HTML element is an individual component of an HTML file. In an HTML file, everything written within tags are termed as HTML elements.



HTML Tags

HTML tags are like keywords which defines that how web browser will format and display the content. With the help of tags, a web browser can distinguish between an HTML content and a simple content. HTML tags contain three main parts: opening tag, content and closing tag. But some HTML tags are unclosed tags.

When a web browser reads an HTML document, browser reads it from top to bottom and left to right. HTML tags are used to create HTML documents and render their properties. Each HTML tags have different properties.

An HTML file must have some essential tags so that web browser can differentiate between a simple text and HTML text. You can use as many tags you want as per your code requirement.

- All HTML tags must enclosed within < > these brackets.
- Every tag in HTML perform different tasks.
- If you have used an open tag <tag>, then you must use a close tag </tag> (except some tags)

Syntax

<tag> content </tag>

Tag name	Description
<!-- -->	This tag is used to apply comment in an HTML document.
<!DOCTYPE>	This tag is used to specify the version of HTML
A	
<a>	It is termed as anchor tag and it creates a hyperlink or link.
<abbr>	It defines an abbreviation for a phrase or longer word.
<acronym>	It defines acronym for a word. **(Not supported in HTML5)**
<address>	It defines the author's contact information of the HTML article
<applet>	It defines an embedded Java applet. **(Not supported in HTML5)**
<area>	It defines the area of an image map.
<article>	It defines the self-contained content.
<aside>	It defines content aside from main content. Mainly represented as sidebar.
<audio>	It is used to embed sound content in HTML document.
B	
<b>	It is used to make a text bold.
<base>	This tag defines the base URL for all relative URL within the document.
<basefont>	This tag is used to set default font, size and color for all elements of document. **(Not supported in HTML5)**
<bdi>	This tag is used to provide isolation for that part of text which may be formatted in different directions from its surrounding text.

<bdo>	It is used to override the current text direction.
<big>	This tag is used to make font size one level larger than its surrounding content. **(Not supported in HTML5)**
<blockquote>	It is used to define a content which is taken from another source.
<body>	It is used to define the body section of an HTML document.
 	It is used to apply single line break.
<button>	It is used to represent a clickable button
C	
<canvas>	It is used to provide a graphics space within a web document.
<caption>	It is used to define a caption for a table.
	It is used to align the content in center. **(Not supported in HTML5)**
<cite>	It is used to define the title of the work, book, website, etc.
<code>	It is used to display a part of programming code in an HTML document.
<col>	It defines a column within a table which represent common properties of columns and used with the <colgroup> element.
<colgroup>	It is used to define group of columns in a table.
D	
<data>	It is used to link the content with the machine-readable translation.
<datalist>	It is used to provide a predefined list for input option.
<dd>	It is used to provide definition/description of a term in description list.
<del>	It defines a text which has been deleted from the document.
<details>	It defines additional details which user can either view or hide.

<dfn>	It is used to indicate a term which is defined within a sentence/phrase.
<dialog>	It defines a dialog box or other interactive components.
<dir>	It is used as container for directory list of files. **(Not supported in HTML5)**
	It defines a division or section within HTML document.
<dl>	It is sued to define a description list.
<dt>	It is used to define a term in description list.
E	
	It is used to emphasis the content applied within this element.
<embed>	It is used as embedded container for external file/application/media, etc.
F	
<fieldset>	It is used to group related elements/labels within a web form.
<figcaption>	It is used to add a caption or explanation for the <figure> element.
<figure>	It is used to define the self-contained content, and s mostly refer as single unit.
<font>	It defines the font, size, color, and face for the content. **(Not supported in HTML5)**
<footer>	It defines the footer section of a webpage.
<form>	It is used to define an HTML form.
<frame>	It defines a particular area of webpage which can contain another HTML file. **(Not supported in HTML5)**
<frameset>	It defines group of Frames. **(Not supported in HTML5)**
H	

<h1> to <h6>	It defines headings for an HTML document from level 1 to level 6.
<head>	It defines the head section of an HTML document.
<header>	It defines the header of a section or webpage.
<hr>	It is used to apply thematic break between paragraph-level elements.
<html>	It represents root of an HTML document.
I	
<i>	It is used to represent a text in some different voice.
<iframe>	It defines an inline frame which can embed other content.
<img>	It is used to insert an image within an HTML document.
<input>	It defines an input field within an HTML form.
<ins>	It represent text that has been inserted within an HTML document.
<isindex>	It is used to display search string for current document. **(Not supported in HTML5)**
K	
<kbd>	It is used to define keyboard input.
L	
<label>	It defines a text label for the input field of form.
<legend>	It defines a caption for content of <fieldset>
	It is used to represent items in list.
<link>	It represents a relationship between current document and an external resource.
M	

<main>	It represents the main content of an HTML document.
<map>	It defines an image map with active areas.
<mark>	It represents a highlighted text.
<marquee>	It is used to insert the scrolling text or an image either horizontally or vertically. **(Not supported in HTML5)**
<menu>	It is used for creating a menu list of commands.
<meta>	It defines metadata of an HTML document.
<meter>	It defines scalar measurement with known range or fractional value.
N	
<nav>	It represents section of page to represent navigation links.
<noframes>	It provides alternate content to represent in browser which does not support the <frame> elements. **(Not supported in HTML5)**
<noscript>	It provides an alternative content if a script type is not supported in browser.
O	
<object>	It is used to embed an object in HTML file.
	It defines an ordered list of items.
<optgroup>	It is used to group the options of a drop-down list.
<option>	It is used to define options or items in a drop-down list.
<output>	It is used as container element which can show result of a calculation.
P	
<p>	It represents a paragraph in an HTML document.
<param>	It defines parameter for an <object> element

<picture>	It defines more than one source element and one image element.
<pre>	It defines preformatted text in an HTML document.
<progress>	It defines the progress of a task within HTML document.
Q	
<q>	It defines short inline quotation.
R	
<rp>	It defines an alternative content if browser does not supports ruby annotations.
<rt>	It defines explanations and pronunciations in ruby annotations.
<ruby>	It is used to represent ruby annotations.
S	
<s>	It render text which is no longer correct or relevant.
<samp>	It is used to represent sample output of a computer program.
<script>	It is used to declare the JavaScript within HTML document.
<section>	It defines a generic section for a document.
<select>	It represents a control which provides a menu of options.
<small>	It is used to make text font one size smaller than document?s base font size.
<source>>	It defines multiple media recourses for different media element such as <picture>, <video>, and <audio> element.
	It is used for styling and grouping inline.
<strike>	It is used to render strike through the text. **(Not supported in HTML5)**
	It is used to define important text.
<style>	It is used to contain style information for an HTML document.

<sub>	It defines a text which displays as a subscript text.
<summary>	It defines summary which can be used with <details> tag.
<sup>	It defines a text which represent as superscript text.
<svg>	It is used as container of SVG (Scalable Vector Graphics).
T	
<table>	It is used to present data in tabular form or to create a table within HTML document.
<tbody>	It represents the body content of an HTML table and used along with <thead> and <tfoot>.
<td>	It is used to define cells of an HTML table which contains table data
<template>	It is used to contain the client side content which will not display at time of page load and may render later using JavaScript.
<textarea>	It is used to define multiple line input, such as comment, feedback, and review, etc.
<tfoot>	It defines the footer content of an HTML table.
<th>	It defines the head cell of an HTML table.
<thead>	It defines the header of an HTML table. It is used along with <tbody> and <tfoot> tags.
<time>	It is used to define data/time within an HTML document.
<title>	It defines the title or name of an HTML document.
<tr>	It defines the row cells in an HTML table
<track>	It is used to define text tracks for <audio> and <video> elements.
<tt>	It is used to define teletype text. **(Not supported in HTML5)**
U	

<u>	It is used to render enclosed text with an underline.
	It defines unordered list of items.
V	
<var>	It defines variable name used in mathematical or programming context.
<video>	It is used to embed a video content with an HTML document
W	
<wbr>	It defines a position within text where break line is possible.

HTML Anchor

The **HTML anchor tag** defines *a hyperlink that links one page to another page*. It can create hyperlink to other web page as well as files, location, or any URL. The "href" attribute is the most important attribute of the HTML a tag. and which links to destination page or URL.

href attribute of HTML anchor tag

The href attribute is used to define the address of the file to be linked. In other words, it points out the destination page.

The syntax of HTML anchor tag is given below.

```
<a href = "....."> Link Text </a>
```

Let's see an example of HTML anchor tag.

```
<a href="second.html">Click for Second Page</a>
```

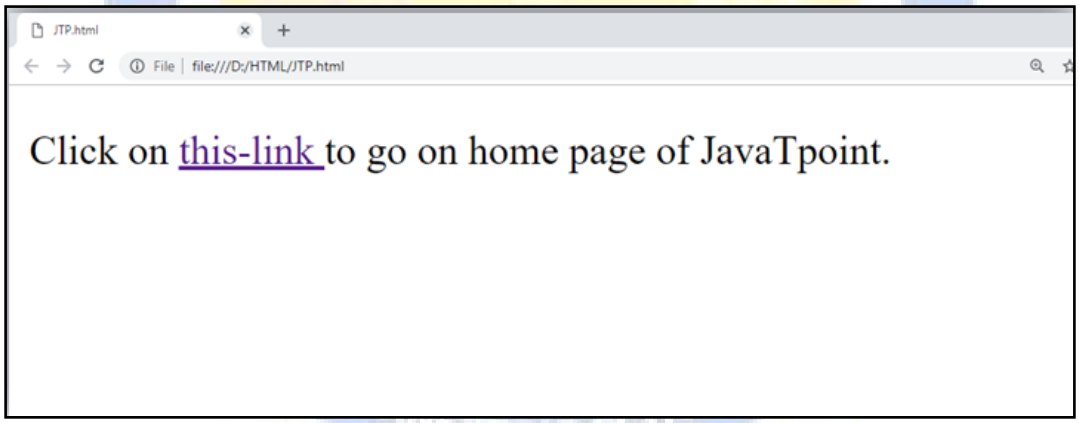
Specify a location for Link using target attribute

If we want to open that link to another page then we can use target attribute of <a> tag. With the help of this link will be open in next page.

Example:

```
<!DOCTYPE html>
<html>
<head>
  <title></title>
</head>
<body>
    <p>Click on <a href="https://www.javatpoint.com/" target="_blank"> this-
    link </a>to go on home page of JavaTpoint.</p>
</body>
</html>
```

Output:



Note:

- The **target** attribute can only use with href attribute in anchor tag.
- If we will not use target attribute then link will open in same page.

Appearance of HTML anchor tag

An **unvisited link** is displayed underlined and blue.

A **visited link** displayed underlined and purple.

An **active link** is underlined and red.

HTML Table

An HTML table is defined with the **<table>** tag. Each table row is defined with the **<tr>** tag. A table header is defined with the **<th>** tag. By default, table headings are bold and centered. A table data/cell is defined with the **<td>** tag.

Example

```
<table style="width:100%">
        <tr>
                <th>Firstname</th>
                <th>Lastname</th>
                <th>Age</th>
         </tr>
        <tr>
                 <td>Jill</td>
                <td>Smith</td>
                <td>50</td>
        </tr>
        <tr>
                <td>Eve</td>
                <td>Jackson</td>
                <td>94</td>
        </tr>
</table>
```

Note: The <td> elements are the data containers of the table. They can contain all sorts of HTML elements; text, images, lists, other tables, etc.

Tag	Description
<TABLE>	It defines a table.
<TR>	It defines a row in a table.
<TH>	It defines a header cell in a table.
<TD>	It defines a cell in a table.
<CAPTION>	It defines the table caption. <caption> STUDENT TABLE </caption>
BORDER	Border attribute of table tag in HTML is used to specify border. EX:- <table border="1">
CELL PADDING	Cell padding represents the distance between cell borders and the content within a cell.

	EX:- <table cellpadding = "5" >
CELLSPACING	Cellspacing attribute defines space between table cells. EX:- <table cellspacing = "5">
COLSPAN	Colspan attribute is used if you want to merge two or more columns into a single column. EX:- <td COLspan = "3">
ROWSPAN	Rowspan attribute is used if you want to merge two or more rows. EX:- <td rowspan = "2">
BGCOLOR	bgcolor attribute is used to set background color for whole table or just for one cell. <table bgcolor = "yellow">
BORDERCOLOR	Border color attribute is used to set color of the table border . <table border = "1" bordercolor = "green">
BACKGROUND	background attribute is used to set background image for whole table or just for one cell. <table border = "1" bordercolor = "green" background = "/images/test.png">

HTML Frameset

The <frameset> tag defines a frameset. The <frameset> element holds one or more <frame> elements. Each <frame> element can hold a separate document. The <frameset> element specifies how many columns or rows there will be in the frameset, and how much percentage/pixels of space will occupy each of them.

Attribute	Value	Description
cols	*pixels* *%* *	Specifies the number and size of columns in a frameset
rows	*pixels* *%* *	Specifies the number and size of rows in a frameset

Example

A simple three-framed page:

```
<frameset cols="25%,*, 25%">
  <frame src="frame_a.htm">
  <frame src="frame_b.htm">
  <frame src="frame_c.htm">
</frameset>
```

HTML Form

An **HTML form** is *a section of a document* which contains controls such as text fields, password fields, checkboxes, radio buttons, submit button, menus etc.

An HTML form facilitates the user to enter data that is to be sent to the server for processing such as name, email address, password, phone number, etc.

Why use HTML Form

HTML forms are required if you want to collect some data from of the site visitor.

For example: If a user want to purchase some items on internet, he/she must fill the form such as shipping address and credit/debit card details so that item can be sent to the given address.

HTML Form Syntax

```
<form action="server url" method="get|post">
        //input controls e.g. textfield, textarea, radiobutton, button
</form>
```

HTML Form Tags

Tag	Description
<form>	It defines an HTML form to enter inputs by the used side.

<input>	It defines an input control.
<textarea>	It defines a multi-line input control.
<label>	It defines a label for an input element.
<fieldset>	It groups the related element in a form.
<legend>	It defines a caption for a <fieldset> element.
<select>	It defines a drop-down list.
<optgroup>	It defines a group of related options in a drop-down list.
<option>	It defines an option in a drop-down list.
<button>	It defines a clickable button.

HTML 5 Form Tags

Tag	Description
<datalist>	It specifies a list of pre-defined options for input control.
<keygen>	It defines a key-pair generator field for forms.
<output>	It defines the result of a calculation.

HTML <form> element

The HTML <form> element provide a document section to take input from user. It provides various interactive controls for submitting information to web server such as text field, text area, password field, etc.

> Note: The <form> element does not itself create a form but it is container to contain all required form elements, such as <input>, <label>, etc.

Syntax:

```
<form>
        //Form elements
</form>
```

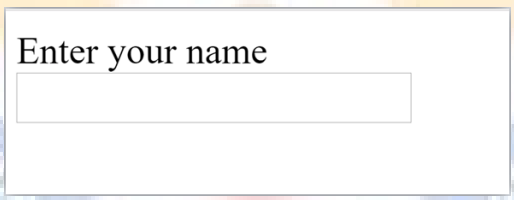
HTML <input> element

The HTML <input> element is fundamental form element. It is used to create form fields, to take input from user. We can apply different input filed to gather different information form user. Following is the example to show the simple text input.

Example:

```
<body>
 <form>
   Enter your name  <br>
   <input type="text" name="username">
 </form>
</body>
```

Output:

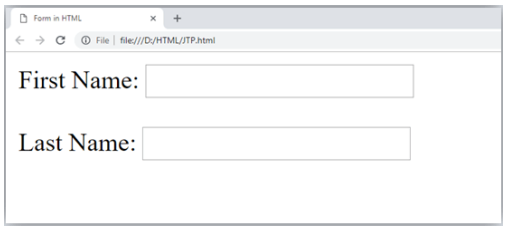


HTML TextField Control

The type="text" attribute of input tag creates textfield control also known as single line textfield control. The name attribute is optional, but it is required for the server side component such as JSP, ASP, PHP etc.

```
<form>
   First Name: <input type="text" name="firstname"/> <br/>
   Last Name:  <input type="text" name="lastname"/> <br/>
 </form>
```

Output:



Note: If you will omit 'name' attribute then the text filed input will not be submitted to server.

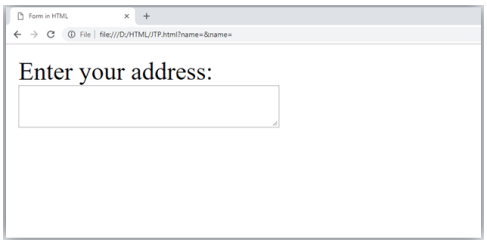
HTML <textarea> tag in form

The <textarea> tag in HTML is used to insert multiple-line text in a form. The size of <textarea> can be specify either using "rows" or "cols" attribute or by CSS.

Example:

```
<!DOCTYPE html>
<html>
<head>
   <title>Form in HTML</title>
</head>
<body>
 <form>
     Enter your address:<br>
    <textarea rows="2" cols="20"></textarea>
 </form>
</body>
</html>
```

Output:



Label Tag in Form

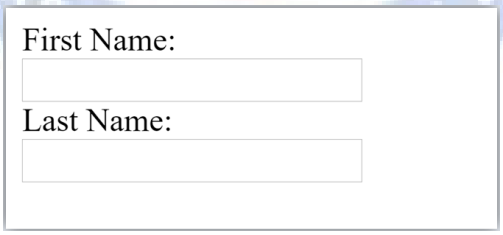
It is considered better to have label in form. As it makes the code parser/browser/user friendly.

If you click on the label tag, it will focus on the text control. To do so, you need to have for attribute in label tag that must be same as id attribute of input tag.

> NOTE: It is good to use <label> tag with form, although it is optional but if you will use it, then it will provide a focus when you tap or click on label tag. It is more worthy with touchscreens.

```
<form>
  <label for="firstname">First Name: </label> <br/>
        <input type="text" id="firstname" name="firstname"/> <br/>
  <label for="lastname">Last Name: </label>
        <input type="text" id="lastname" name="lastname"/> <br/>
</form>
```

Output:



HTML Password Field Control

The password is not visible to the user in password field control.

```
<form>
  <label for="password">Password: </label>
```

```
        <input type="password" id="password" name="password"/> <br/>
    </form>
```

Output:

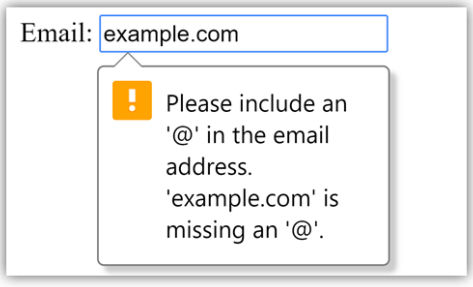
HTML 5 Email Field Control

The email field in new in HTML 5. It validates the text for correct email address. You must use @ and . in this field.

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

It will display in browser like below:

Note: If we will not enter the correct email, it will display error like:



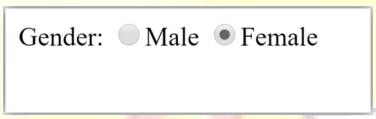
Radio Button Control

The radio button is used to select one option from multiple options. It is used for selection of gender, quiz questions etc.

If you use one name for all the radio buttons, only one radio button can be selected at a time.

Using radio buttons for multiple options, you can only choose a single option at a time.

```
<form>
  <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/>
</form>
```



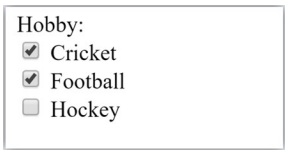
Checkbox Control

The checkbox control is used to check multiple options from given checkboxes.

```
<form>
Hobby:<br>
      <input type="checkbox" id="cricket" name="cricket" value="cricket"/
>
        <label for="cricket">Cricket</label> <br>
      <input type="checkbox" id="football" name="football" value="footba
ll"/>
        <label for="football">Football</label> <br>
      <input type="checkbox" id="hockey" name="hockey" value="hockey"
/>
        <label for="hockey">Hockey</label>
</form>
```

> Note: These are similar to radio button except it can choose multiple options at a time and radio button can select one button at a time, and its display.

Output:



Submit button control

HTML **<input type="submit">** are used to add a submit button on web page. When user clicks on submit button, then form get submit to the server.

Syntax:

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

The type = submit , specifying that it is a submit button

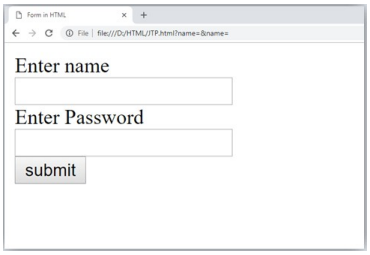
The value attribute can be anything which we write on button on web page.

The name attribute can be omit here.

Example:

```
<form>
   <label for="name">Enter name</label><br>
   <input type="text" id="name" name="name"><br>
   <label for="pass">Enter Password</label><br>
   <input type="Password" id="pass" name="pass"><br>
   <input type="submit" value="submit">
</form>
```

Output:



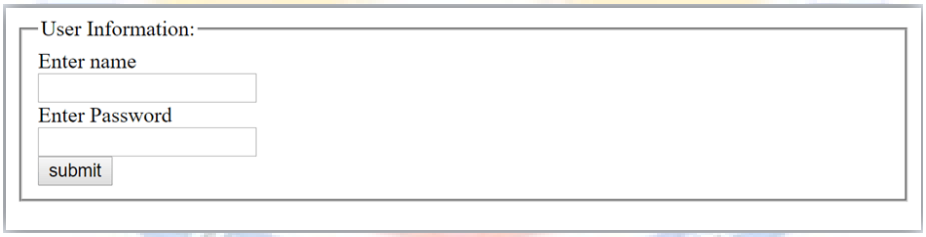
HTML <fieldset> element:

The <fieldset> element in HTML is used to group the related information of a form. This element is used with <legend> element which provide caption for the grouped elements.

Example:

```
<form>
  <fieldset>
   <legend>User Information:</legend>
  <label for="name">Enter name</label><br>
<input type="text" id="name" name="name"><br>
<label for="pass">Enter Password</label><br>
<input type="Password" id="pass" name="pass"><br>
<input type="submit" value="submit">
</fieldset>
</form>
```

Output:



HTML Form Example

```
<!DOCTYPE html>
<html>
<head>
<title>Form in HTML</title>
</head>
```

```
<body>
  <h2>Registration form</h2>
 <form>
  <fieldset>
    <legend>User personal information</legend>
    <label>Enter your full name</label><br>
    <input type="text" name="name"><br>
     <label>Enter your email</label><br>
     <input type="email" name="email"><br>
     <label>Enter your password</label><br>
     <input type="password" name="pass"><br>
     <label>confirm your password</label><br>
     <input type="password" name="pass"><br>
     <br><label>Enter your gender</label><br>
     <input type="radio" id="gender" name="gender" value="male"/>Male  <br>
     <input type="radio" id="gender" name="gender" value="female"/>Female <br
/>
     <input type="radio" id="gender" name="gender" value="others"/>others <br/
>
      <br>Enter your Address:<br>
     <textarea></textarea><br>
     <input type="submit" value="sign-up">
  </fieldset>
 </form>
</body>
</html>
```

Output:

Advantages:

- HTML is widely used.
- Every browser supports HTML Language.
- Easy to learn and use.
- HTML is light weighted and fast to load.
- Do not get to purchase any extra software because it's by default in every window.
- Easy to use
- Loose syntax (although, being too flexible won't suit standards).
- HTML is easy enough to write
- HTML is that it is easy to code even for novice programmers.
- HTML also allows the utilization of templates, which makes designing a webpage easy.
- Very useful for beginners in the web designing field.
- HTML can be supported to each and every browser, if not supported to all the browsers.
- HTML is built on almost every website, if not all websites.
- HTML is increasingly used for data storage as like XML syntax.
- Free – You need not buy any software.
- HTML is present in every window by default so you not need to buy the software which cost too much.
- HTML has many tag and attributes which can short your line of code.

Disadvantages:

- It cannot produce dynamic output alone, since it's a static language.

- Making the structure of HTML documents becomes tough to understand.
- Errors can be costly.
- It is the time consuming as the time it consume to maintain on the colour scheme of a page and to make lists, tables and forms.
- It can create only static and plain pages so if we'd like dynamic pages then HTML isn't useful.
- Required to write a lot of code for just creating a simple webpage.
- We have to check up the deprecated tags and confirm not to use them to appear because another language that works with HTML has replaced the first work of the tag, and hence the opposite language needs to be understood and learned.
- Security features offered by HTML are limited.
- If we need to write down long code for creating a webpage then it produces some complexity.
- HTML can create only static and plain pages so if we'd like dynamic pages then HTML isn't useful.
- I need to write down tons of code for creating an easy webpage.
- Security features are not good at HTML.
- If we'd like to write down long code for creating a webpage then it produces some complexity.

Cascading Style Sheets (CSS) describe how documents are presented on screens, in print, or perhaps how they are pronounced. W3C has actively promoted the use of style sheets on the Web since the consortium was founded in 1994.

Cascading Style Sheets (CSS) provide easy and effective alternatives to specify various attributes for the HTML tags. Using CSS, you can specify a number of style properties for a given HTML element. Each property has a name and a value, separated by a colon (:). Each property declaration is separated by a semi-colon (;).

Example

First let's consider an example of HTML document which makes use of <font> tag and associated attributes to specify text color and font size −

Note − The *font* tag deprecated and it is supposed to be removed in a future version of HTML. So they should not be used rather, it's suggested to use CSS styles to manipulate your fonts. But still for learning purpose, this chapter will work with an example using the font tag.

Live Demo

```
<!DOCTYPE html>
<html>

   <head>
      <title>HTML CSS</title>
   </head>

   <body>
      <p><font color = "green" size = "5">Hello, World!</font></p>
   </body>

</html>
```

We can re-write above example with the help of Style Sheet as follows −

Live Demo

```
<!DOCTYPE html>
<html>

   <head>
      <title>HTML CSS</title>
   </head>

   <body>
      <p style = "color:green; font-size:24px;" >Hello, World!</p>
   </body>

</html>
```

This will produce the following result −

You can use CSS in three ways in your HTML document −

- **External Style Sheet** − Define style sheet rules in a separate .css file and then include that file in your HTML document using HTML <link> tag.
- **Internal Style Sheet** − Define style sheet rules in header section of the HTML document using <style> tag.
- **Inline Style Sheet** − Define style sheet rules directly along-with the HTML elements using **style** attribute.

Let's see all the three cases one by one with the help of suitable examples.

External Style Sheet

If you need to use your style sheet to various pages, then its always recommended to define a common style sheet in a separate file. A cascading style sheet file will have extension as **.css** and it will be included in HTML files using <link> tag.

Example

Consider we define a style sheet file **style.css** which has following rules −

```
.red {
   color: red;
}
.thick {
   font-size:20px;
}
.green {
   color:green;
}
```

Here we defined three CSS rules which will be applicable to three different classes defined for the HTML tags. I suggest you should not bother about how these rules are being defined because you will learn them while studying CSS. Now let's make use of the above external CSS file in our following HTML document −

Live Demo

```
<!DOCTYPE html>
<html>

   <head>
      <title>HTML External CSS</title>
      <link rel = "stylesheet" type = "text/css" href =
"/html/style.css">
   </head>

   <body>
      <p class = "red">This is red</p>
      <p class = "thick">This is thick</p>
      <p class = "green">This is green</p>
      <p class = "thick green">This is thick and green</p>
   </body>

</html>
```

This will produce the following result −

Internal Style Sheet

If you want to apply Style Sheet rules to a single document only, then you can include those rules in header section of the HTML document using <style> tag.

Rules defined in internal style sheet overrides the rules defined in an external CSS file.

Example

Let's re-write above example once again, but here we will write style sheet rules in the same HTML document using <style> tag −

Live Demo

```
<!DOCTYPE html>
<html>

   <head>
      <title>HTML Internal CSS</title>

      <style type = "text/css">
         .red {
            color: red;
         }
         .thick{
            font-size:20px;
         }
         .green {
            color:green;
         }
      </style>
   </head>

   <body>
      <p class = "red">This is red</p>
      <p class = "thick">This is thick</p>
      <p class = "green">This is green</p>
      <p class = "thick green">This is thick and green</p>
   </body>

</html>
```

This will produce the following result −

Inline Style Sheet

You can apply style sheet rules directly to any HTML element using **style** attribute of the relevant tag. This should be done only when you are interested to make a particular change in any HTML element only.

Rules defined inline with the element overrides the rules defined in an external CSS file as well as the rules defined in <style> element.

Example

Let's re-write above example once again, but here we will write style sheet rules along with the HTML elements using **style** attribute of those elements.

Live Demo

```
<!DOCTYPE html>
<html>

   <head>
      <title>HTML Inline CSS</title>
   </head>

   <body>
      <p style = "color:red;">This is red</p>
      <p style = "font-size:20px;">This is thick</p>
      <p style = "color:green;">This is green</p>
      <p style = "color:green;font-size:20px;">This is thick and
green</p>
   </body>

</html>
```

This will produce the following result –

UNIT – 7
CLIENT SIDE SCRIPTING WITH JAVASCRIPT

Introduction to JavaScript

JavaScript is a lightweight, cross-platform, and interpreted scripting language. It is well-known for the development of web pages, many non-browser environments also use it. JavaScript can be used for **Client-side** developments as well as **Server-side** developments. JavaScript contains a standard library of objects, like **Array**, **Date**, and **Math**, and a core set of language elements like **operators**, **control structures**, and **statements**.

- **Client-side:** It supplies objects to control a browser and its Document Object Model (DOM). Like if client-side extensions allow an application to place elements on an HTML form and respond to user events such as **mouse clicks**, **form input**, and **page navigation**. Useful libraries for the client-side are **AngularJS**, **ReactJS**, **VueJS** and so many others.
- **Server-side:** It supplies objects relevant to running JavaScript on a server. Like if the server-side extensions allow an application to communicate with a database, and provide continuity of information from one invocation to another of the application, or perform file manipulations on a server. The useful framework which is the most famous these days is **node.js**.

JavaScript can be added to your HTML file in two ways:

- **Internal JS:** We can add JavaScript directly to our HTML file by writing the code inside the <script> tag. The <script> tag can either be placed inside the <head> or the <body> tag according to the requirement.
- **External JS:** We can write JavaScript code in other file having an extension .js and then link this file inside the <head> tag of the HTML file in which we want to add this code.

Syntax:

```
<script>
    // JavaScript Code
</script>
```

Example

```
<script>
        document.write("Hello JavaScript by JavaScript");
</script>
```

Application of JavaScript

JavaScript is used to create interactive websites. It is mainly used for:

- Client-side validation,
- Dynamic drop-down menus,
- Displaying date and time,
- Displaying pop-up windows and dialog boxes (like an alert dialog box, confirm dialog box and prompt dialog box),
- Displaying clocks etc.

Limitations of JavaScript:

- **Performance:** JavaScript does not provide the same level of performance as offered by many traditional languages as a complex program written in JavaScript would be comparatively slow. But as JavaScript is used to perform simple tasks in a browser, so performance is not considered a big restriction in its use.
- **Complexity:** To master a scripting language, programmers must have a thorough knowledge of all the programming concepts, core language objects, client and server-side objects otherwise it would be difficult for them to write advanced scripts using JavaScript.
- **Weak error handling and type checking facilities:** It is weakly typed language as there is no need to specify the data type of the variable. So wrong type checking is not performed by compile.

Features of JavaScript

1. All popular web browsers support JavaScript as they provide built-in execution environments.
2. JavaScript follows the syntax and structure of the C programming language. Thus, it is a structured programming language.
3. JavaScript is a weakly typed language, where certain types are implicitly cast (depending on the operation).
4. JavaScript is an object-oriented programming language that uses prototypes rather than using classes for inheritance.
5. It is a light-weighted and interpreted language.
6. It is a case-sensitive language.
7. JavaScript is supportable in several operating systems including, Windows, macOS, etc.
8. It provides good control to the users over the web browsers.

JavaScript Variable

A **JavaScript variable** is simply a name of storage location. There are two types of variables in JavaScript : local variable and global variable.

There are some rules while declaring a JavaScript variable (also known as identifiers).

1. Name must start with a letter (a to z or A to Z), underscore(_), or dollar($) sign.
2. After first letter we can use digits (0 to 9), for example value1.
3. JavaScript variables are case sensitive, for example x and X are different variables.

Example:

```
var x = 10;
var _value="sonoo";
```

Let's see a simple example of JavaScript variable.

```
<script>
    var x = 10;
    var y = 20;
```

```
        var z=x+y;
        document.write(z);
    </script>
```

Output of the above example

```
30
```

JavaScript local variable

A JavaScript local variable is declared inside block or function. It is accessible within the function or block only. For example:

```
<script>

    function abc()
        {
        var x=10;//local variable
        }
</script>
```

JavaScript global variable

A **JavaScript global variable** is accessible from any function. A variable i.e. declared outside the function or declared with window object is known as global variable. For example:

```
<script>
        var data=200;//gloabal variable
        function a()
            {
            document.writeln(data);
            }
        function b()
            {
            document.writeln(data);
            }
        a();//calling JavaScript function
        b();
</script>
```

JavaScript Data Types

JavaScript provides different **data types** to hold different types of values. There are two types of data types in JavaScript.

1. Primitive data type
2. Non-primitive (reference) data type

JavaScript is a **dynamic type language**, means you don't need to specify type of the variable because it is dynamically used by JavaScript engine. You need to use **var** here to specify the data type. It can hold any type of values such as numbers, strings etc. For example:

```
var a=40;//holding number
var b="Rahul";//holding string
```

Primitive Data Types

There are five types of primitive data types in JavaScript.

Data Type	Description
String	represents sequence of characters e.g. "hello"
Number	represents numeric values e.g. 100
Boolean	represents boolean value either false or true
Undefined	represents undefined value
Null	represents null i.e. no value at all

Non-Primitive Data Types

The non-primitive data types are as follows:

Data Type	Description
Object	represents instance through which we can access members
Array	represents group of similar values
RegExp	represents regular expression

JavaScript Operators

JavaScript operators are symbols that are used to perform operations on operands. For example:

```
var sum=10+20;
```

Here, + is the arithmetic operator and = is the assignment operator.

There are following types of operators in JavaScript.

1. Arithmetic Operators
2. Comparison (Relational) Operators
3. Bitwise Operators
4. Logical Operators
5. Assignment Operators
6. Special Operators

Arithmetic Operators

Arithmetic operators are used to perform arithmetic operations on the operands. The following operators are known as JavaScript arithmetic operators.

Operator	Description	Example
+	Addition	10+20 = 30

-	Subtraction	20-10 = 10
*	Multiplication	10*20 = 200
/	Division	20/10 = 2
%	Modulus (Remainder)	20%10 = 0
++	Increment	var a=10; a++; Now a = 11
--	Decrement	var a=10; a--; Now a = 9

Comparison Operators

The JavaScript comparison operator compares the two operands. The comparison operators are as follows:

Operator	Description	Example
==	Is equal to	10==20 = false
===	Identical (equal and of same type)	10==20 = false
!=	Not equal to	10!=20 = true
!==	Not Identical	20!==20 = false
>	Greater than	20>10 = true
>=	Greater than or equal to	20>=10 = true
<	Less than	20<10 = false
<=	Less than or equal to	20<=10 = false

Bitwise Operators

The bitwise operators perform bitwise operations on operands. The bitwise operators are as follows:

Operator	Description	Example
&	Bitwise AND	(10==20 & 20==33) = false
|	Bitwise OR	(10==20 | 20==33) = false
^	Bitwise XOR	(10==20 ^ 20==33) = false
~	Bitwise NOT	(~10) = -10
<<	Bitwise Left Shift	(10<<2) = 40
>>	Bitwise Right Shift	(10>>2) = 2
>>>	Bitwise Right Shift with Zero	(10>>>2) = 2

Logical Operators

Operator	Description	Example
&&	Logical AND	(10==20 && 20==33) = false
||	Logical OR	(10==20 || 20==33) = false
!	Logical Not	!(10==20) = true

Assignment Operators

Operator	Description	Example
=	Assign	10+10 = 20

+=	Add and assign	var a=10; a+=20; Now a = 30
-=	Subtract and assign	var a=20; a-=10; Now a = 10
*=	Multiply and assign	var a=10; a*=20; Now a = 200
/=	Divide and assign	var a=10; a/=2; Now a = 5
%=	Modulus and assign	var a=10; a%=2; Now a = 0

Special Operators

Operator	Description
(?:)	Conditional Operator returns value based on the condition. It is like if-else.
,	Comma Operator allows multiple expressions to be evaluated as single statement.
delete	Delete Operator deletes a property from the object.
in	In Operator checks if object has the given property
instanceof	checks if the object is an instance of given type
new	creates an instance (object)
typeof	checks the type of object.
void	it discards the expression's return value.
yield	checks what is returned in a generator by the generator's iterator.

JavaScript Branching statement

The **JavaScript if-else statement** is used *to execute the code whether condition is true or false*. There are three forms of if statement in JavaScript.

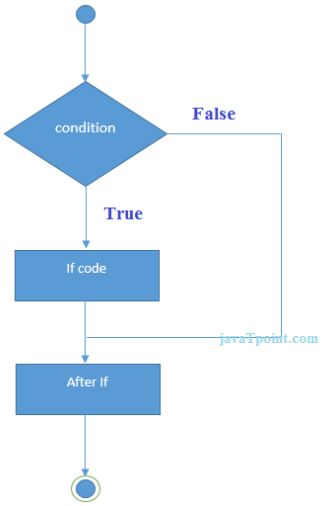
1. If Statement
2. If else statement
3. if else if statement

JavaScript If statement

It evaluates the content only if expression is true. The signature of JavaScript if statement is given below.

```
if(expression){
//content to be evaluated
}
```

Flowchart of JavaScript If statement



Example:

```
<script>
    var a=20;
    if(a>10)
        {
        document.write("value of a is greater than 10");
        }
```

```
</script>
```

Output of the above example

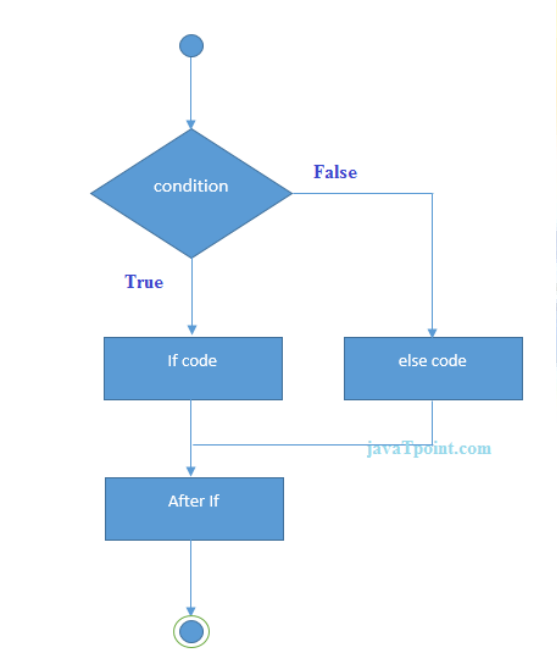
```
value of a is greater than 10
```

JavaScript If...else Statement

It evaluates the content whether condition is true of false. The **syntax** of JavaScript if-else statement is given below.

```
if(expression){
//content to be evaluated if condition is true
}
else{
//content to be evaluated if condition is false
}
```

Flowchart of JavaScript If...else statement



Example:

```
<script>
```

```
var a=20;
if(a%2==0)
        {
        document.write("a is even number");
        }
Else
        {
        document.write("a is odd number");
        }
</script>
```

Output of the above example

```
a is even number
```

JavaScript If...else if statement

It evaluates the content only if expression is true from several expressions. The signature of JavaScript if else if statement is given below.

```
if(expression1){
//content to be evaluated if expression1 is true
}
else if(expression2){
//content to be evaluated if expression2 is true
}
else if(expression3){
//content to be evaluated if expression3 is true
}
else{
//content to be evaluated if no expression is true
}
```

Example:

```
<script>
var a=20;
if(a==10)
        {
        document.write("a is equal to 10");
        }
else if(a==15)
        {
        document.write("a is equal to 15");
        }
else if(a==20)
        {
        document.write("a is equal to 20");
        }
Else
        {
        document.write("a is not equal to 10, 15 or 20");
        }
</script>
```

Output of the above example

```
a is equal to 20
```

JavaScript Switch

The **JavaScript switch statement** is used *to execute one code from multiple expressions*. It is just like else if statement that we have learned in previous page. But it is convenient than *if..else..if* because it can be used with numbers, characters etc.

The signature of JavaScript switch statement is given below.

```
switch(expression)
{
```

```
        case value1:
                code to be executed;
         break;
        case value2:
                code to be executed;
         break;
        ......

        default:
                code to be executed if above values are not matched;
}
```

Example:

```
<script>
var grade='B';
var result;
switch(grade)
{
        case 'A':
                result="A Grade";
        break;
        case 'B':
                result="B Grade";
        break;
        case 'C':
                result="C Grade";
        break;
        default:
                result="No Grade";
}
document.write(result);
</script>
```

Output of the above example

```
B Grade
```

JavaScript Loops

The **JavaScript loops** are used *to iterate the piece of code* using for, while, do while or for-in loops. It makes the code compact. It is mostly used in array.

There are three types of loops in JavaScript.

1. for loop
2. while loop
3. do-while loop

JavaScript For loop

The **JavaScript for loop** *iterates the elements for the fixed number of times*. It should be used if number of iteration is known. The syntax of for loop is given below.

```
for (initialization; condition; increment)
{
   code to be executed
}
```

Example:

```
<script>
for (i=1; i<=5; i++)
        {
        document.write(i + "<br/>")
        }
</script>
```

Output:

```
1
2
3
4
5
```

JavaScript while loop

The **JavaScript while loop** *iterates the elements for the infinite number of times*. It should be used if number of iteration is not known. The syntax of while loop is given below.

```
while (condition)
{
   code to be executed
}
```

Example:

```
<script>
var i=11;
while (i<=15)
        {
        document.write(i + "<br/>");
        i++;
        }
</script>
```

Output:

```
11
12
13
14
15
```

JavaScript do while loop

The **JavaScript do while loop** *iterates the elements for the infinite number of times* like while loop. But, code is *executed at least* once whether condition is true or false. The syntax of do while loop is given below.

```
Do
```

```
{
   code to be executed
} while (condition);
```

Example:

```
<script>
var i=21;
do
{
        document.write(i + "<br/>");
        i++;
}while (i<=25);
</script>
```

Output:

```
21
22
23
24
25
```

JavaScript Array

JavaScript array is an object that represents a collection of similar type of elements.

There are 3 ways to construct array in JavaScript

1. By array literal
2. By creating instance of Array directly (using new keyword)
3. By using an Array constructor (using new keyword)

JavaScript array literal

The syntax of creating array using array literal is given below:

```
var arrayname=[value1,value2.....valueN];
```

As you can see, values are contained inside [ ] and separated by , (comma).

Example:

```
<script>
var emp=["Sonoo","Vimal","Ratan"];
for (i=0;i<emp.length;i++)
        {
        document.write(emp[i] + "<br/>");
        }
</script>
```

The .length property returns the length of an array.

Output of the above example

```
Sonoo

Vimal

Ratan
```

JavaScript Array directly (new keyword)

The syntax of creating array directly is given below:

```
var arrayname=new Array();
```

Here, **new keyword** is used to create instance of array.

Example:

```
<script>
var i;
var emp = new Array();
emp[0] = "Arun";
emp[1] = "Varun";
emp[2] = "John";
```

```
for (i=0;i<emp.length;i++)
        {
        document.write(emp[i] + "<br>");
        }
</script>
```

Output of the above example

```
Arun

Varun

John
```

JavaScript array constructor (new keyword)

Here, you need to create instance of array by passing arguments in constructor so that we don't have to provide value explicitly.

Example:

```
<script>
var emp=new Array("Jai","Vijay","Smith");
for (i=0;i<emp.length;i++)
        {
        document.write(emp[i] + "<br>");
        }
</script>
```

Output of the above example

```
Jai

Vijay

Smith
```

JavaScript Array Methods

Methods	Description
concat()	It returns a new array object that contains two or more merged arrays.
copywithin()	It copies the part of the given array with its own elements and returns the modified array.
entries()	It creates an iterator object and a loop that iterates over each key/value pair.
every()	It determines whether all the elements of an array are satisfying the provided function conditions.
flat()	It creates a new array carrying sub-array elements concatenated recursively till the specified depth.
flatMap()	It maps all array elements via mapping function, then flattens the result into a new array.
fill()	It fills elements into an array with static values.
from()	It creates a new array carrying the exact copy of another array element.
filter()	It returns the new array containing the elements that pass the provided function conditions.

find()	It returns the value of the first element in the given array that satisfies the specified condition.
findIndex()	It returns the index value of the first element in the given array that satisfies the specified condition.
forEach()	It invokes the provided function once for each element of an array.
includes()	It checks whether the given array contains the specified element.
indexOf()	It searches the specified element in the given array and returns the index of the first match.
isArray()	It tests if the passed value ia an array.
join()	It joins the elements of an array as a string.
keys()	It creates an iterator object that contains only the keys of the array, then loops through these keys.
lastIndexOf()	It searches the specified element in the given array and returns the index of the last match.
map()	It calls the specified function for every array element and returns the new array

of()	It creates a new array from a variable number of arguments, holding any type of argument.
pop()	It removes and returns the last element of an array.
push()	It adds one or more elements to the end of an array.
reverse()	It reverses the elements of given array.
reduce(function, initial)	It executes a provided function for each value from left to right and reduces the array to a single value.
reduceRight()	It executes a provided function for each value from right to left and reduces the array to a single value.
some()	It determines if any element of the array passes the test of the implemented function.
shift()	It removes and returns the first element of an array.
slice()	It returns a new array containing the copy of the part of the given array.
sort()	It returns the element of the given array in a sorted order.

splice()	It add/remove elements to/from the given array.
toLocaleString()	It returns a string containing all the elements of a specified array.
toString()	It converts the elements of a specified array into string form, without affecting the original array.
unshift()	It adds one or more elements in the beginning of the given array.
values()	It creates a new iterator object carrying values for each index in the array.

Document Object Model

The **document object** represents the whole html document.

When html document is loaded in the browser, it becomes a document object. It is the **root element** that represents the html document. It has properties and methods. By the help of document object, we can add dynamic content to our web page.

As mentioned earlier, it is the object of window. So

```
window.document
```

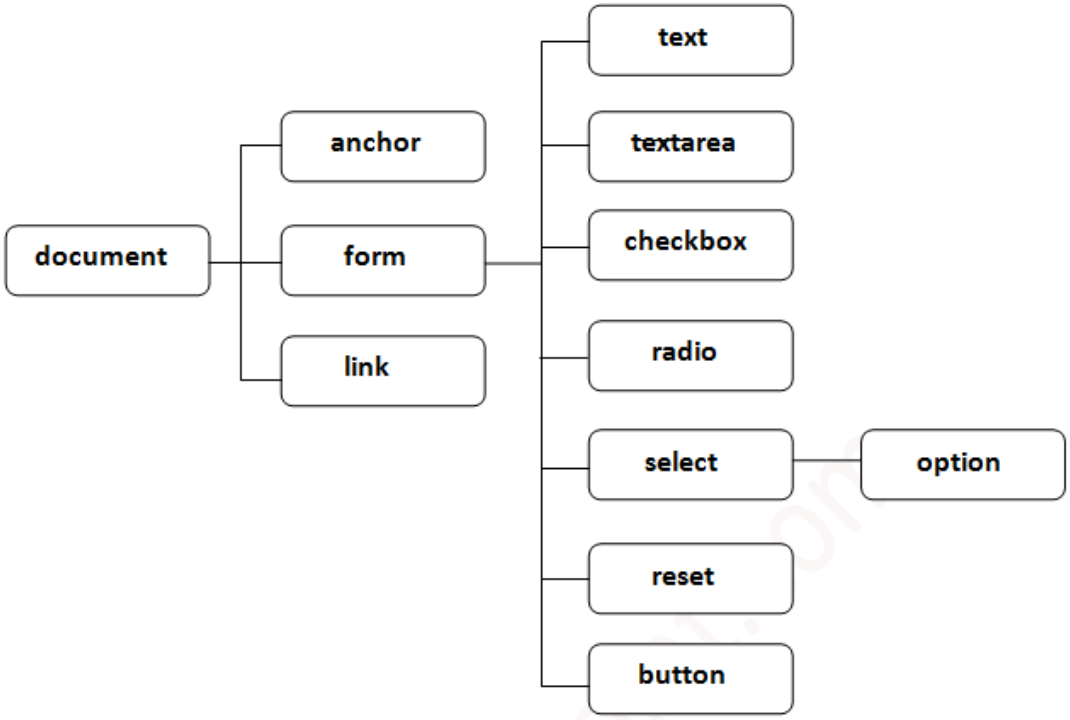
Is same as

```
document
```

According to W3C - *"The W3C Document Object Model (DOM) is a platform and language-neutral interface that allows programs and scripts to dynamically access and update the content, structure, and style of a document."*

Properties of document object

Let's see the properties of document object that can be accessed and modified by the document object.



Methods of document object

We can access and change the contents of document by its methods.

The important methods of document object are as follows:

Method	Description
write("string")	writes the given string on the doucment.
writeln("string")	writes the given string on the doucment with newline character at the end.
getElementById()	returns the element having the given id value.

getElementsByName()	returns all the elements having the given name value.
getElementsByTagName()	returns all the elements having the given tag name.
getElementsByClassName()	returns all the elements having the given class name.

Accessing field value by document object

In this example, we are going to get the value of input text by user. Here, we are using **document.form1.name.value** to get the value of name field. Here,

- **document** is the root element that represents the html document.
- **form1** is the name of the form.
- **name** is the attribute name of the input text.
- **value** is the property, that returns the value of the input text.

Let's see the simple example of document object that prints name with welcome message.

```
<script type="text/javascript">
function printvalue()
{
        var name=document.form1.name.value;
        alert("Welcome: "+name);
}
</script>

<form name="form1">
Enter Name:<input type="text" name="name"/>
<input type="button" onclick="printvalue()" value="print name"/>
</form>
```

Output of the above example

Enter Name:

JavaScript Functions

JavaScript functions are used to perform operations. We can call JavaScript function many times to reuse the code.

Advantage of JavaScript function

There are mainly two advantages of JavaScript functions.

1. **Code reusability**: We can call a function several times so it save coding.
2. **Less coding**: It makes our program compact. We don't need to write many lines of code each time to perform a common task.

JavaScript Function Syntax

The syntax of declaring function is given below.

```
function functionName([arg1, arg2, ...argN])
{
 //code to be executed
}
```

JavaScript Functions can have 0 or more arguments.

JavaScript Function Example

Let's see the simple example of function in JavaScript that does not has arguments.

```
<script>
function msg()
{
        alert("hello! this is message");
}
</script>
<input type="button" onclick="msg()" value="call function"/>
```

JavaScript Function Arguments

We can call function by passing arguments. Let's see the example of function that has one argument.

```
<script>
function getcube(number)
{
        alert(number*number*number);
}
</script>
<form>
<input type="button" value="click" onclick="getcube(4)"/>
</form>
```

Function with Return Value

We can call function that returns a value and use it in our program. Let's see the example of function that returns value.

```
<script>
        function getInfo()
        {
                return "hello javatpoint! How r u?";
        }
</script>
<script>
        document.write(getInfo());
</script>
```

```
hello javatpoint! How r u?
```

JavaScript Function Object

In JavaScript, the purpose of **Function constructor** is to create a new Function object. It executes the code globally. However, if we call the constructor directly, a function is created dynamically but in an unsecured way.

Syntax

```
new Function ([arg1[, arg2[, ....argn]],] functionBody)
```

Parameter

arg1, arg2, ... , argn - It represents the argument used by function.

functionBody - It represents the function definition.

JavaScript Function Methods

Method	Description
apply()	It is used to call a function contains this value and a single array of arguments.
bind()	It is used to create a new function.
call()	It is used to call a function contains this value and an argument list.
toString()	It returns the result in a form of a string.

JavaScript Function Object Examples

Example 1

Let's see an example to display the sum of given numbers.

```
<script>
var add=new Function("num1","num2","return num1+num2");
document.writeln(add(2,5));
</script>
```

Output:

```
7
```

Example 2

Let's see an example to display the power of provided value.

```
<script>
```

```
2. var pow=new Function("num1","num2","return Math.pow(num1,num2)");
3. document.writeln(pow(2,3));
4. </script>
```

Output:

```
8
```

JavaScript Objects

- A javaScript object is an entity having state and behavior (properties and method). For example: car, pen, bike, chair, glass, keyboard, monitor etc.
- JavaScript is an object-based language. Everything is an object in JavaScript.
- JavaScript is template based not class based. Here, we don't create class to get the object. But, we direct create objects.

Creating Objects in JavaScript

There are 3 ways to create objects.

1. By object literal
2. By creating instance of Object directly (using new keyword)
3. By using an object constructor (using new keyword)

JavaScript Object by object literal

The syntax of creating object using object literal is given below:

```
object={property1:value1,property2:value2.....propertyN:valueN}
```

As you can see, property and value is separated by : (colon).

Example : creating object in JavaScript.

```
<script>
    emp={id:102,name:"Shyam Kumar",salary:40000}
    document.write(emp.id+" "+emp.name+" "+emp.salary);
</script>
```

Output of the above example

```
102 Shyam Kumar 40000
```

By creating instance of Object

The syntax of creating object directly is given below:

```
var objectname=new Object();
```

Here, **new keyword** is used to create object.

Example : creating object directly.

```
<script>
var emp=new Object();
emp.id=101;
emp.name="Ravi Malik";
emp.salary=50000;
document.write(emp.id+" "+emp.name+" "+emp.salary);
</script>
```

Output of the above example

```
101 Ravi 50000
```

By using an Object constructor

Here, you need to create function with arguments. Each argument value can be assigned in the current object by using this keyword.The **this keyword** refers to the current object.

Example : creating object by object constructor

```
<script>
function emp(id,name,salary)
```

```
{
        this.id=id;
        this.name=name;
        this.salary=salary;
}
e=new emp(103,"Vimal Jaiswal",30000);

document.write(e.id+" "+e.name+" "+e.salary);
</script>
```

Output of the above example

```
103 Vimal Jaiswal 30000
```

Defining method in JavaScript Object

We can define method in JavaScript object. But before defining method, we need to add property in the function with same name as method.

Example : defining method in object

```
<script>
function emp(id,name,salary)
{
        this.id=id;
        this.name=name;
        this.salary=salary;
        this.changeSalary=changeSalary;
        function changeSalary(otherSalary)
        {
                this.salary=otherSalary;
        }
}
e=new emp(103,"Sonoo Jaiswal",30000);
document.write(e.id+" "+e.name+" "+e.salary);
```

```
e.changeSalary(45000);
document.write("<br>"+e.id+" "+e.name+" "+e.salary);
</script>
```

Output of the above example

```
103 Sonoo Jaiswal 30000
103 Sonoo Jaiswal 45000
```

JavaScript Object Methods

S.No	Methods	Description
1	Object.assign()	This method is used to copy enumerable and own properties from a source object to a target object
2	Object.create()	This method is used to create a new object with the specified prototype object and properties.
3	Object.defineProperty()	This method is used to describe some behavioral attributes of the property.
4	Object.defineProperties()	This method is used to create or configure multiple object properties.
5	Object.entries()	This method returns an array with arrays of the key, value pairs.
6	Object.freeze()	This method prevents existing properties from being removed.
7	Object.getOwnPropertyDescriptor()	This method returns a property descriptor for the specified property of the specified object.
8	Object.getOwnPropertyDescriptors()	This method returns all own property descriptors of a given object.
9	Object.getOwnPropertyNames()	This method returns an array of all properties (enumerable or not) found.

10	Object.getOwnPropertySymbols()	This method returns an array of all own symbol key properties.
11	Object.getPrototypeOf()	This method returns the prototype of the specified object.
12	Object.is()	This method determines whether two values are the same value.
13	Object.isExtensible()	This method determines if an object is extensible
14	Object.isFrozen()	This method determines if an object was frozen.
15	Object.isSealed()	This method determines if an object is sealed.
16	Object.keys()	This method returns an array of a given object's own property names.
17	Object.preventExtensions()	This method is used to prevent any extensions of an object.
18	Object.seal()	This method prevents new properties from being added and marks all existing properties as non-configurable.
19	Object.setPrototypeOf()	This method sets the prototype of a specified object to another object.
20	Object.values()	This method returns an array of values.

JavaScript Events

The change in the state of an object is known as an **Event**. In html, there are various events which represents that some activity is performed by the user or by the browser. When javascript code is included in HTML, js react over these events and allow the execution. This process of reacting over the events is called **Event Handling**. Thus, js handles the HTML events via **Event Handlers**.

For example, when a user clicks over the browser, add js code, which will execute the task to be performed on the event.

Some of the HTML events and their event handlers are:

Mouse events:

Event Performed	Event Handler	Description
click	onclick	When mouse click on an element
mouseover	onmouseover	When the cursor of the mouse comes over the element
mouseout	onmouseout	When the cursor of the mouse leaves an element
mousedown	onmousedown	When the mouse button is pressed over the element
mouseup	onmouseup	When the mouse button is released over the element
mousemove	onmousemove	When the mouse movement takes place.

Keyboard events:

Event Performed	Event Handler	Description
Keydown & Keyup	onkeydown & onkeyup	When the user press and then release the key

Form events:

Event Performed	Event Handler	Description
focus	onfocus	When the user focuses on an element
submit	onsubmit	When the user submits the form
blur	onblur	When the focus is away from a form element
change	onchange	When the user modifies or changes the value of a form element

Window/Document events

Event Performed	Event Handler	Description
load	onload	When the browser finishes the loading of the page
unload	onunload	When the visitor leaves the current webpage, the browser unloads it
resize	onresize	When the visitor resizes the window of the browser

Example :

Click Event

```
<html>
    <head> Javascript Events </head>
```

```
    <body>
        <script language="Javascript" type="text/Javascript">
            <!--
            function clickevent()
                {
                    document.write("This is JavaTpoint");
                }
            //-->
        </script>
        <form>
            <input type="button" onclick="clickevent()" value="Who's this?"/>
        </form>
    </body>
</html>
```

Embedding JavaScript with html

JavaScript, also known as JS, is one of the scripting (client-side scripting) languages, that is usually used in web development to create modern and interactive web-pages. The term "script" is used to refer to the languages that are not standalone in nature and here it refers to JavaScript which run on the client machine.

In other words, we can say that the term scripting is used for languages that require the support of another language to get executed. For example, JavaScript programs cannot get executed without the help of HTML or without integrated into HTML code.

JavaScript is used in several ways in web pages such as generate warning messages, build image galleries, DOM manipulation, form validation, and more.

Adding JavaScript to HTML Pages

There are following three ways in which users can add JavaScript to HTML pages.

1. Embedding code
2. Inline code

3. External file

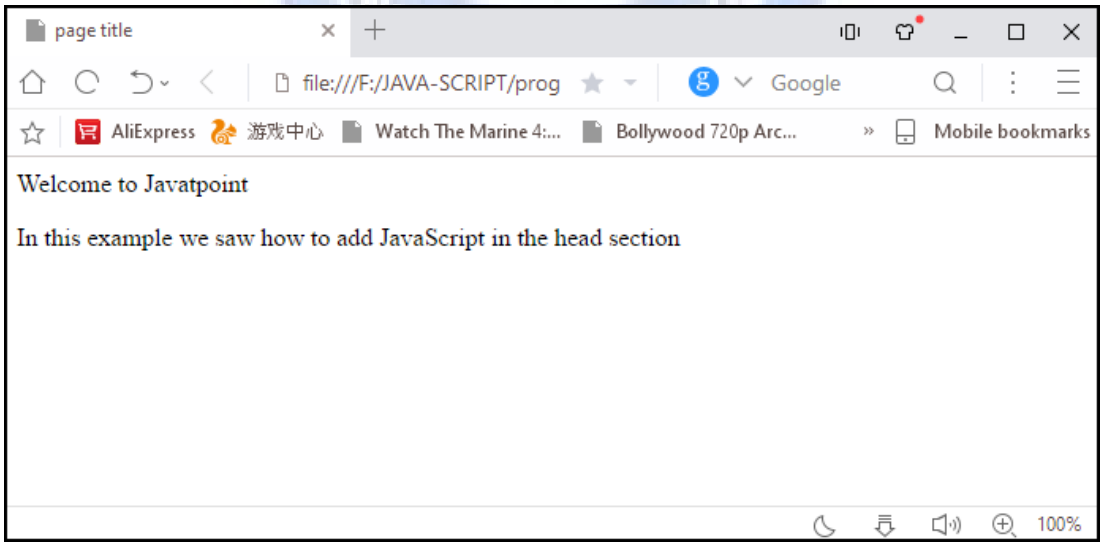
I. Embedding code:-

To add the JavaScript code into the HTML pages, we can use the <script>.....</script> tag of the HTML that wrap around JavaScript code inside the HTML program. Users can also define JavaScript code in the <body> tag (or we can say body section) or <head> tag because it completely depends on the structure of the web page that the users use.

Example

```
<!DOCTYPE html >
<html>
    <head>
        <title> page title</title>
        <script>
            document.write("Welcome to Javatpoint");
        </script>
    </head>
    <body>
        <p>Inthis example we saw how to add JavaScript in the head section </p>
    </body>
</html>
```

Output:

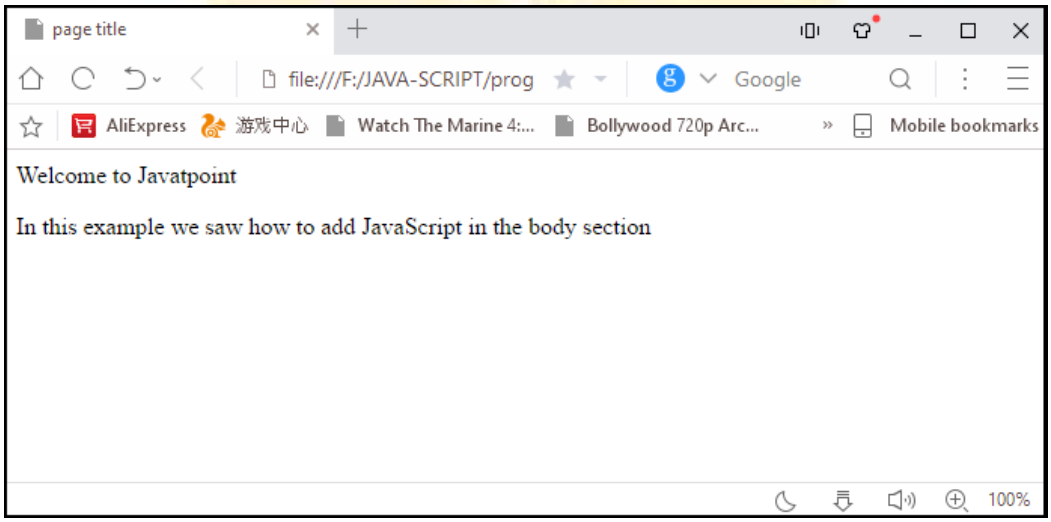


We can also define the JavaScript code in the <body> tags or body section.

Let's understand through an example.

```
<!DOCTYPE html >
<html>
    <head>
        <title> page title</title>
    </head>
    <body>
        <script>
            document.write("Welcome to Javatpoint");
        </script>
        <p> In this example we saw how to add  JavaScript in the body section </p>
    </body>
</html>
```

Output



II. Inline code:-

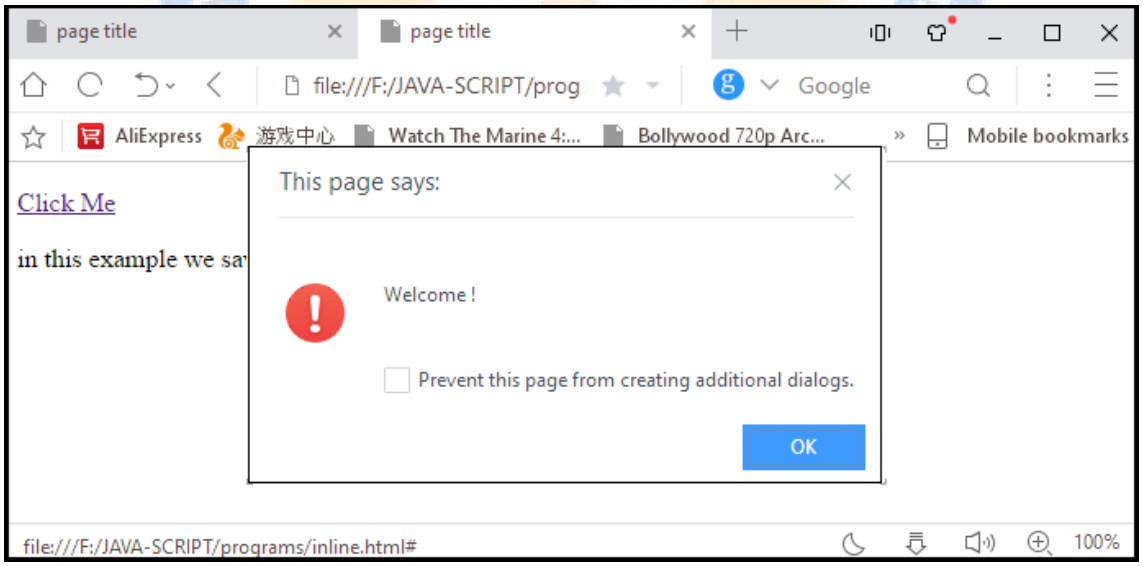
Generally, this method is used when we have to call a function in the HTML event attributes. There are many cases (or events) in which we have to add JavaScript code directly eg., OnMover event, OnClick etc.

Let's see with the help of an example, how we can add JavaScript directly in the html without using the <script>.... </script> tag.

Let's look at the example.

```
<!DOCTYPE html >
<html>
    <head>
        <title> page title</title>
    </head>
    <body>
        <p>
            <a href="#" onClick="alert('Welcome !');">Click Me</a>
        </p>
        <p> in this example we saw how to use inline JavaScript or directly in an HTML
        tag. </p>
    </body>
</html>
```

Output



III. External file:-

We can also create a separate file to hold the code of JavaScript with the (.js) extension and later incorporate/include it into our HTML document using the **src** attribute of the <script> tag.

It becomes very helpful if we want to use the same code in multiple HTML documents. It also saves us from the task of writing the same code over and over again and makes it easier to maintain web pages.

In this example, we will see how we can include an external JavaScript file in an HTML document.

Let's understand through a simple example.

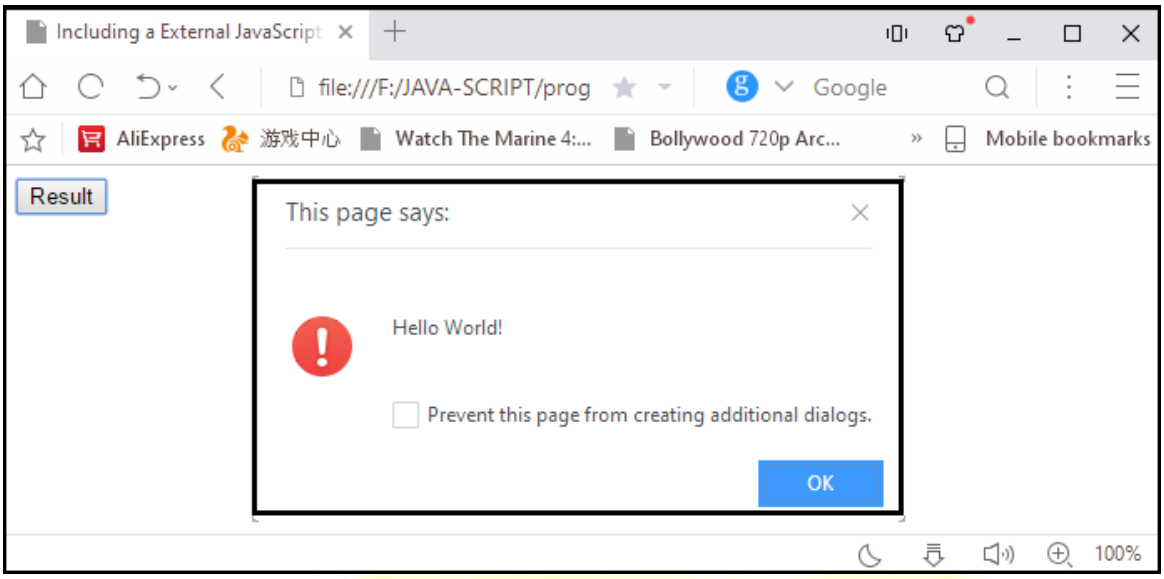
```
<html>
    <head>
        <meta charset="utf-8">
        <title>Including a External JavaScript File</title>
    </head>
    <body>
        <form>
            <input type="button" value="Result" onclick="display()"/>
        </form>
        <script src="hello.js">
        </script>
    </body>
</html>
```

Now let's create separate JavaScript file

Hello.js

```
function display()
{
    alert("Hello World!");
}
```

Output



Both of the above programs are saved in the same folder, but you can also store JavaScript code in a separate folder, all just you need to provide the address/path of the (.js) file in the src attribute of <script> tag.

JavaScript Cookies

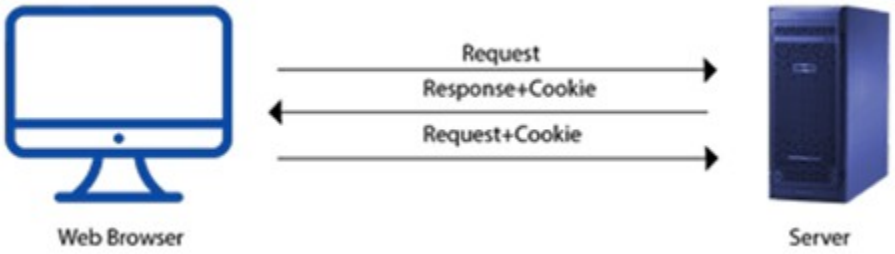
A cookie is an amount of information that persists between a server-side and a client-side. A web browser stores this information at the time of browsing.

A cookie contains the information as a string generally in the form of a name-value pair separated by semi-colons. It maintains the state of a user and remembers the user's information among all the web pages.

How Cookies Works?

- When a user sends a request to the server, then each of that request is treated as a new request sent by the different user.
- So, to recognize the old user, we need to add the cookie with the response from the server.
- Browser at the client-side.

- Now, whenever a user sends a request to the server, the cookie is added with that request automatically. Due to the cookie, the server recognizes the users.



How to create a Cookie in JavaScript?

In JavaScript, we can create, read, update and delete a cookie by using **document.cookie** property.

The following syntax is used to create a cookie:

```
document.cookie="name=value";
```

Example 1

Let's see an example to set and get a cookie.

```
<!DOCTYPE html>
<html>
    <head>
    </head>
    <body>
        <input type="button" value="setCookie" onclick="setCookie()">
        <input type="button" value="getCookie" onclick="getCookie()">
          <script>
            function setCookie()
            {
              document.cookie="username=Duke Martin";
            }
            function getCookie()
            {
              if(document.cookie.length!=0)
```

```
                {
                alert(document.cookie);
                }
                else
                {
                alert("Cookie not available");
                }
            }
        </script>

    </body>
</html>
```

Example 2

Here, we display the cookie's name-value pair separately.

```
<!DOCTYPE html>
<html>
    <head>
    </head>
    <body>
        <input type="button" value="setCookie" onclick="setCookie()">
        <input type="button" value="getCookie" onclick="getCookie()">
          <script>
            function setCookie()
            {
               document.cookie="username=Duke Martin";
            }
            function getCookie()
            {
               if(document.cookie.length!=0)
               {
                  var array=document.cookie.split("=");
               alert("Name="+array[0]+" "+"Value="+array[1]);
               }
               else
```

```
        {
        alert("Cookie not available");
        }
    }
</script>

</body>
</html>
```

How to enable JavaScript in the browser

Nowadays, almost all web-developers use JavaScript in their web-pages. JavaScript is a client-side scripting language that runs on a client machine (such as a web-browser). It helps web-developers to create modern web-pages that allow users to interact directly with the web-pages, without reloading them. If JavaScript is disabled in the web-browser, the web pages may not function properly, or their functionality may be reduced or unavailable.

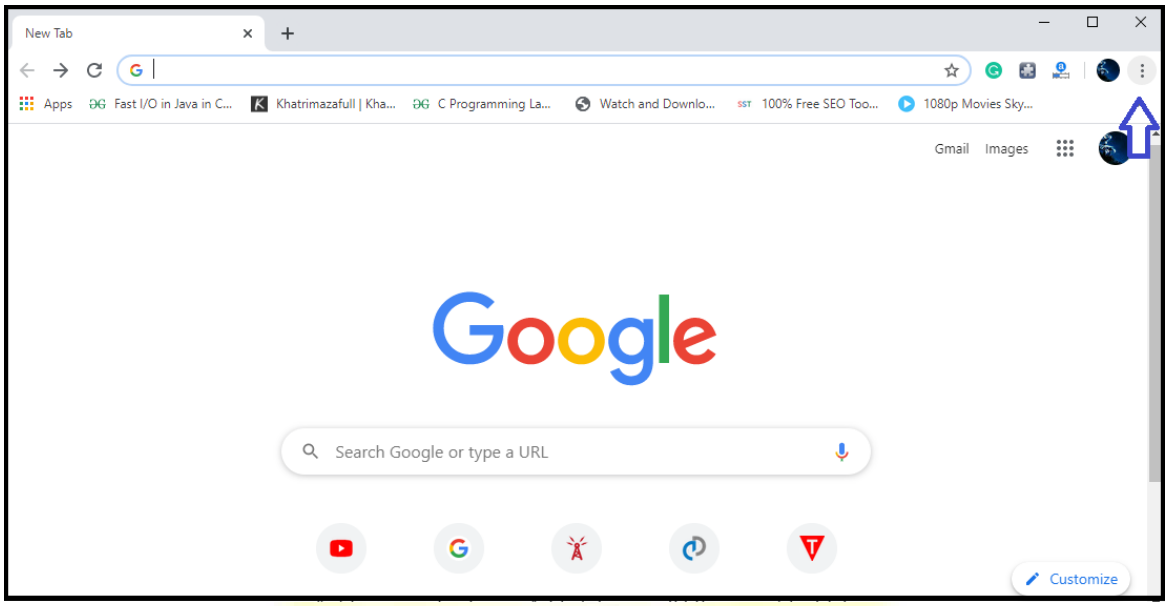
Let's see how we can enable JavaScript in the most commonly used web-browsers by following the given instructions:

1. Chrome Web-browser:

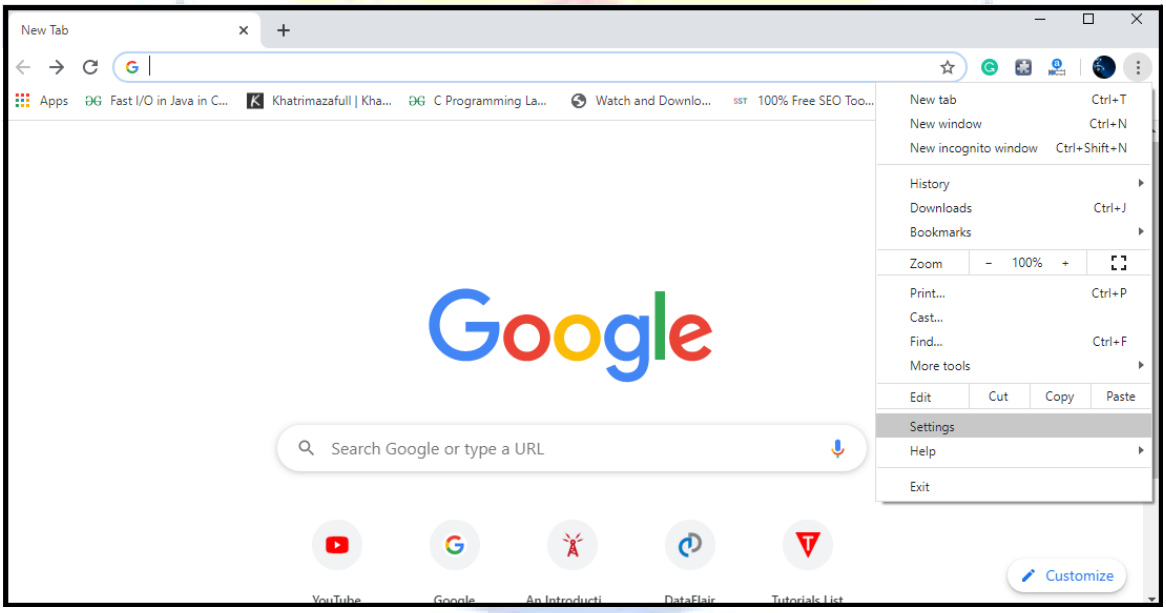
It is a free software owned by GOOGLE. Almost all updates and new source-code of Chrome comes from another Google project "Chromium," which is available for free to download and is also an open-source project.

The steps to enable the JavaScript in the Chrome Browser are as follows:

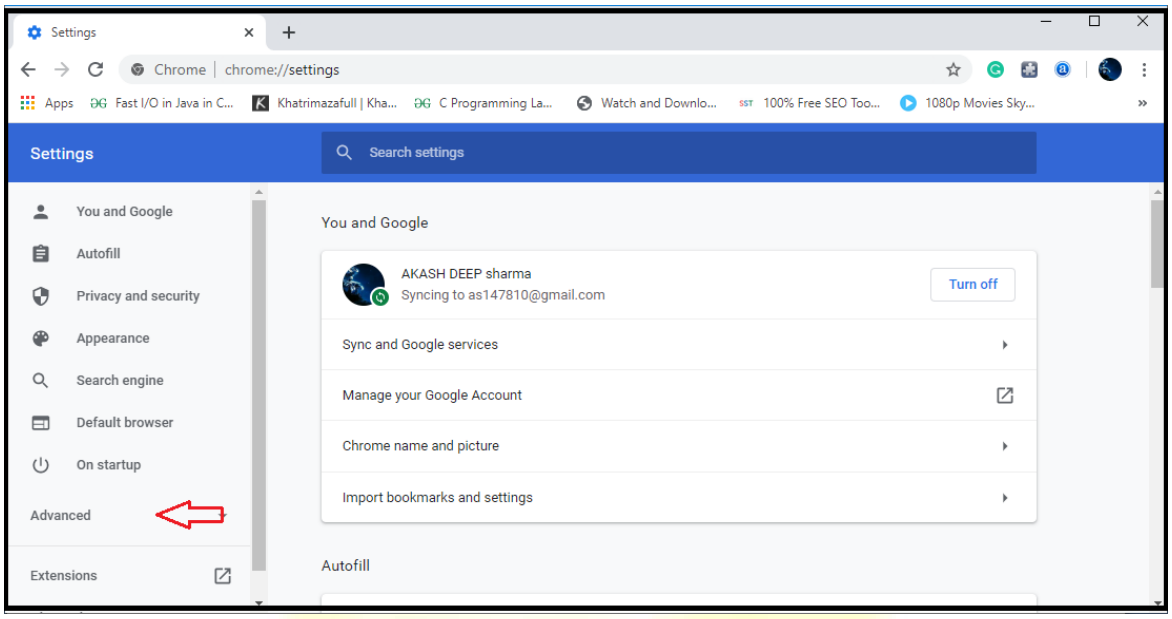
I. First of all, open the chrome browser by double-clicking on it and then click on the three dots sign located at the right- top in the chrome browser, as shown in the below screenshot.



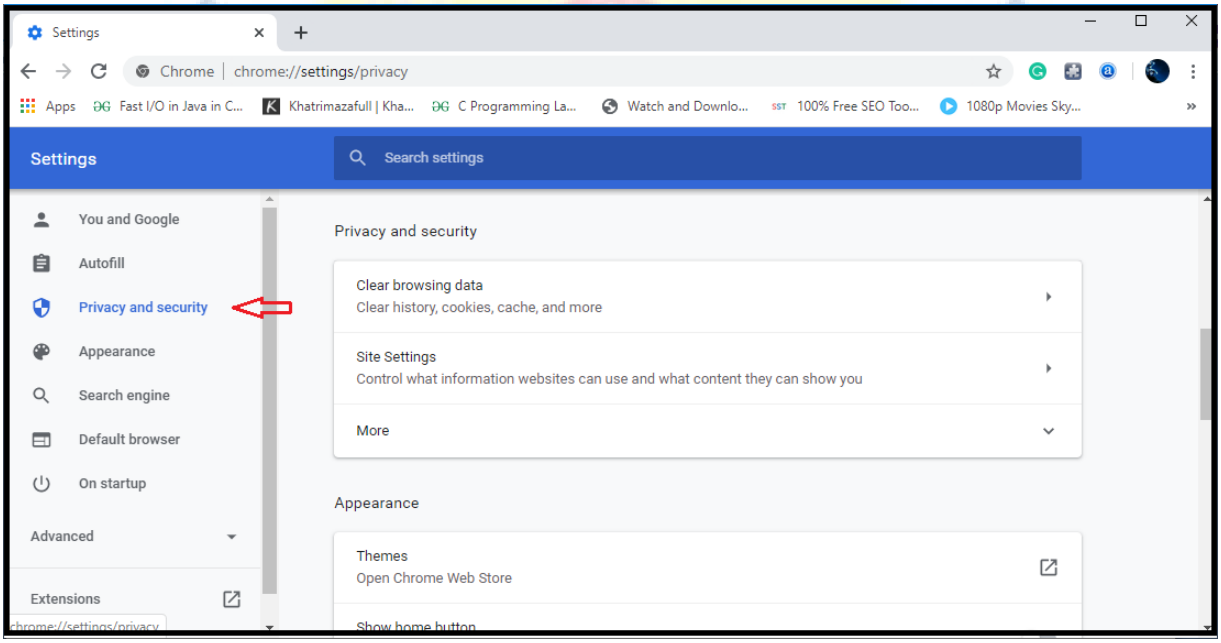
II. Now look for the **"Settings"** option and click on it as shown below:



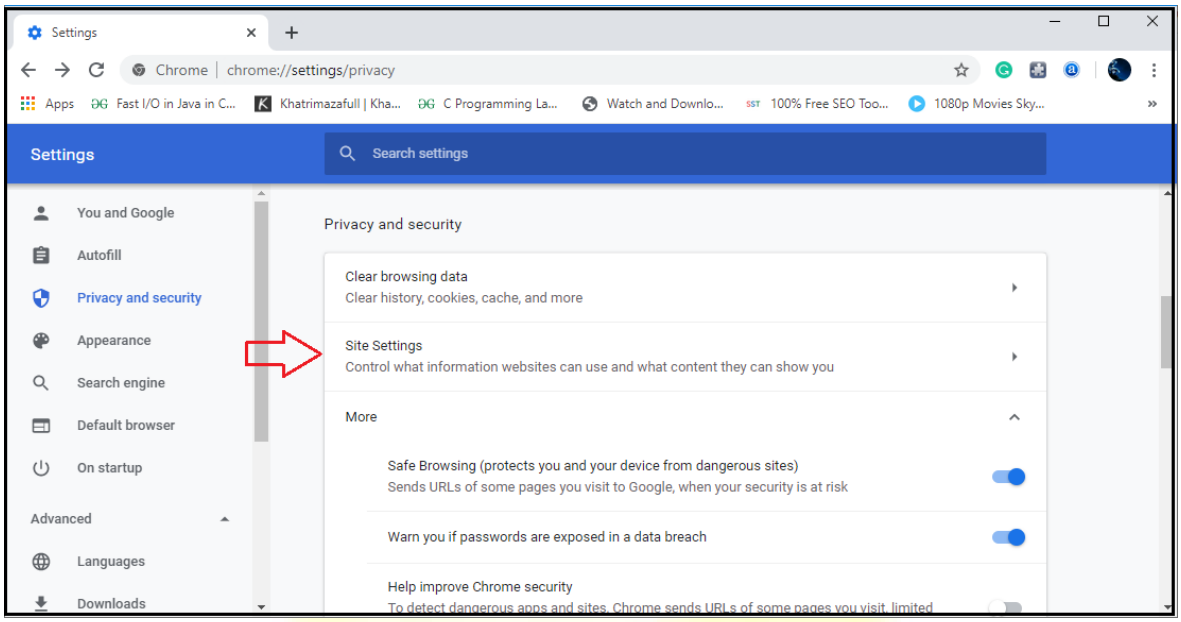
III. When the control panel/Setting of the chrome browser gets open, click on the **"Advanced"** option located at the left side of the setting tab, as shown below.



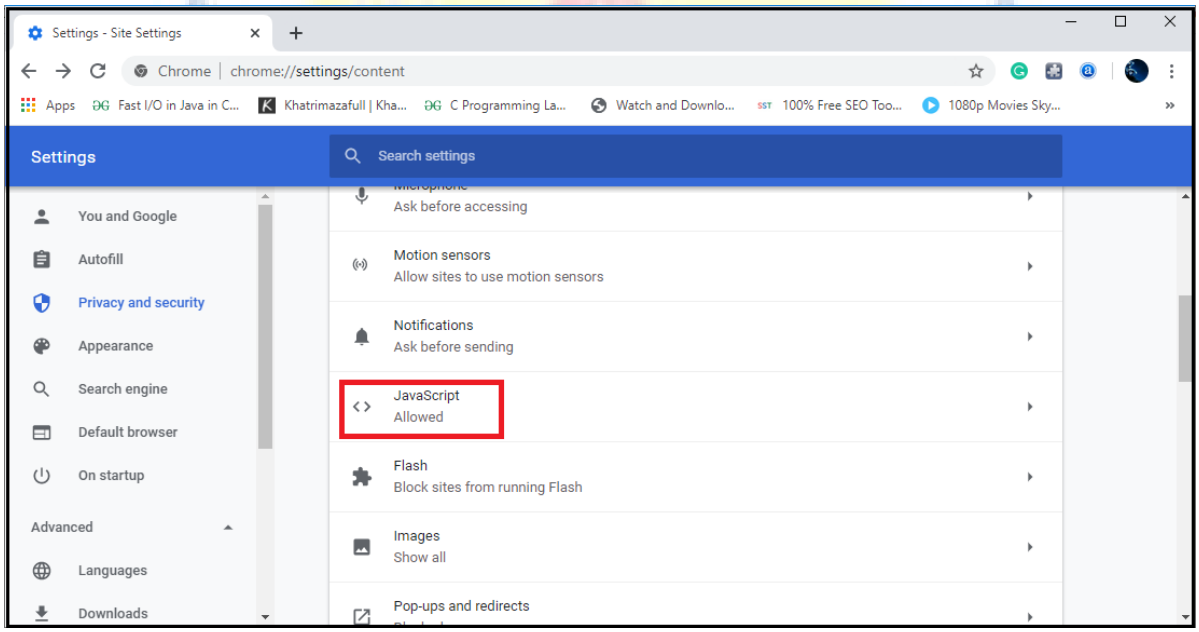
IV. Now click on the **"Privacy and Security"** option.



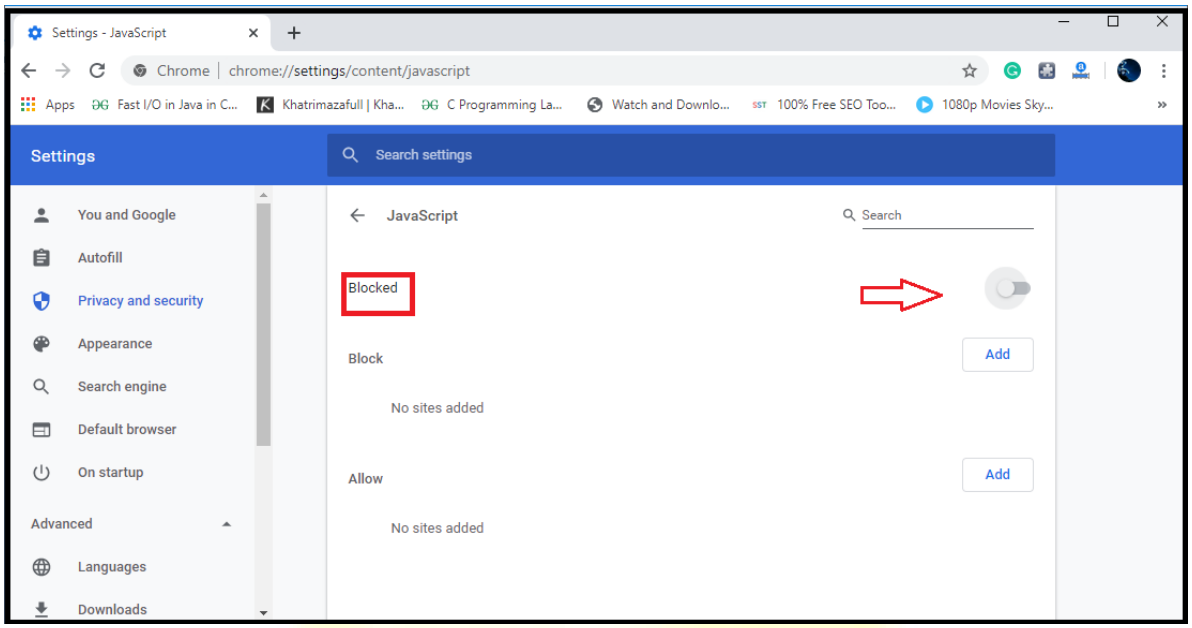
V. Now click on the **"Site Settings"** option.



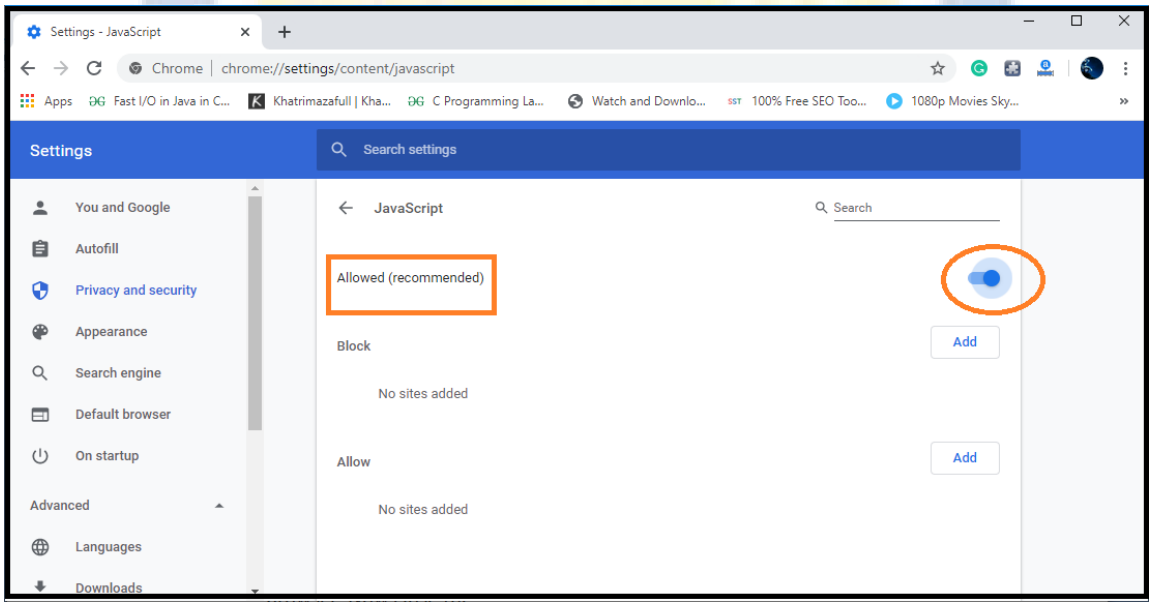
VI. Now search for the "JavaScript" from the following options as shown below:



VII. Now click on the "JavaScript" option and then click on the toggle button given in the right side of the "Blocked" option, as shown in the given screenshot.



VIII. After clicking the toggle button, the "Allowed (recommended)" options will appear in the place of "Blocked," which confirms that JavaScript is enabled successfully in the browser as shown below:



IX. Now you can use the browser for browsing the web-pages after closing the Settings tab.

JavaScript Form Validation

It is important to validate the form submitted by the user because it can have inappropriate values. So, validation is must to authenticate user.

JavaScript provides facility to validate the form on the client-side so data processing will be faster than server-side validation. Most of the web developers prefer JavaScript form validation.

Through JavaScript, we can validate name, password, email, date, mobile numbers and more fields.

JavaScript Form Validation Example

In this example, we are going to validate the name and password. The name can't be empty and password can't be less than 6 characters long.

Here, we are validating the form on form submit. The user will not be forwarded to the next page until given values are correct.

```
<script>
    function validateform()
    {
        var name=document.myform.name.value;
        var password=document.myform.password.value;

        if (name==null || name=="")
        {
            alert("Name can't be blank");
            return false;
        }
        else if(password.length<6)
        {
            alert("Password must be at least 6 characters long.");
            return false;
         }
    }
</script>
```

```
<body>
    <form name="myform" method="post" action="abc.jsp" onsubmit="return validatefor
    m()" >
        Name: <input type="text" name="name"><br/>
        Password: <input type="password" name="password"><br/>
        <input type="submit" value="register">
    </form>
  </body>
```

JavaScript Retype Password Validation

```
<script type="text/javascript">
    function matchpass()
    {
        var firstpassword=document.f1.password.value;
        var secondpassword=document.f1.password2.value;

        if(firstpassword==secondpassword)
        {
            return true;
        }
        Else
        {
            alert("password must be same!");
            return false;
        }
    }
</script>

<form name="f1" action="register.jsp" onsubmit="return matchpass()">
    Password:<input type="password" name="password" /><br/>
    Re-enter Password:<input type="password" name="password2"/><br/>
    <input type="submit">
</form>
```

JavaScript Number Validation

Let's validate the textfield for numeric value only. Here, we are using isNaN() function.

```
<script>
    function validate()
    {
        var num=document.myform.num.value;
        if (isNaN(num))
        {
            document.getElementById("numloc").innerHTML="Enter Numeric value only";
            return false;
        }
        Else
        {
            return true;
        }
    }
</script>
<form name="myform" onsubmit="return validate()" >
    Number: <input type="text" name="num"><span id="numloc"></span><br/>
    <input type="submit" value="submit">
</form>
```

JavaScript validation with image

Let’s see an interactive JavaScript form validation example that displays correct and incorrect image if input is correct or incorrect.

```
<script>
    function validate()
    {
        var name=document.f1.name.value;
        var password=document.f1.password.value;
        var status=false;
```

```
                if(name.length<1)
                {
                        document.getElementById("nameloc").innerHTML=
                        " <img src='unchecked.gif'/> Please enter your name";
                        status=false;
                }
                Else
                {
                        document.getElementById("nameloc").innerHTML=" <img src='checked.
                        gif'/>";
                        status=true;
                }
                if(password.length<6)
                {
                        document.getElementById("passwordloc").innerHTML=
                        " <img src='unchecked.gif'/> Password must be at least 6 char long";
                        status=false;
                }
                Else
                {
                        document.getElementById("passwordloc").innerHTML=" <img src='chec
                        ked.gif'/>";
                }
                return status;
        }
</script>

<form name="f1" action="#" onsubmit="return validate()">
        <table>
                <tr><td>Enter Name:</td><td><input type="text" name="name"/>
                <span id="nameloc"></span></td></tr>
                <tr><td>Enter Password:</td><td><input type="password" name="passwo
                rd"/>
                <span id="passwordloc"></span></td></tr>
```

```
<tr><td colspan="2"><input type="submit" value="register"/></td></tr>
</table>
</form>
```

Output:

Enter Name:

Enter Password:

JavaScript email validation

There are many criteria that need to be follow to validate the email id such as:

- email id must contain the @ and . character
- There must be at least one character before and after the @.
- There must be at least two characters after . (dot).

Let's see the simple example to validate the email field.

```
<script>
    function validateemail()
    {
        var x=document.myform.email.value;
        var atposition=x.indexOf("@");
        var dotposition=x.lastIndexOf(".");
        if (atposition<1 || dotposition<atposition+2 || dotposition+2>=x.length)
        {
            alert("Please enter a valid e-
            mail address \n atpostion:"+atposition+"\n dotposition:"+dotposition);
            return false;
        }
    }
</script>
<body>
```

```
        <form name="myform"  method="post" action="#" onsubmit="return validateemail();
        ">
                Email: <input type="text" name="email"><br/>

                <input type="submit" value="register">
        </form>
</body>
```

UNIT – 8
SERVER SIDE SCRIPTING

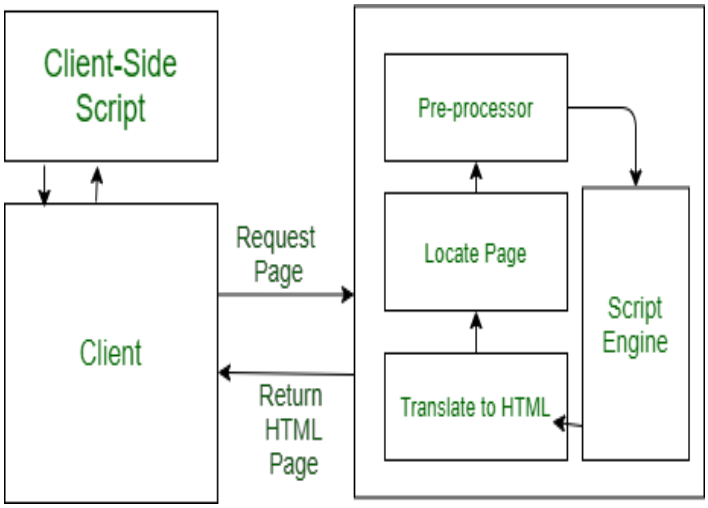
Scripting

The process of creating and embedding scripts in a web page is known as **web-scripting**. A script or a computer-script is a list of commands that are embedded in a web-page normally and are interpreted and executed by a certain program or scripting engine.

- Scripts may be written for a variety of purposes such as for automating processes on a local-computer or to generate web pages.
- The programming languages in which scripts are written are called scripting language, there are many scripting languages available today.
- Common scripting languages are VBScript, JavaScript, ASP, PHP, PERL, JSP etc.

Types of Script :

Scripts are broadly of following two type :



Web Scripting

Client-Side Scripts :

1. Client-side scripting is responsible for interaction within a web page. The client-side scripts are firstly downloaded at the client-end and then interpreted and executed by the browser (default browser of the system).
2. The client-side scripting is browser-dependent. i.e., the client-side browser must be scripting enables in order to run scripts
3. Client-side scripting is used when the client-side interaction is used. Some example uses of client-side scripting may be :
   - To get the data from user's screen or browser.
   - For playing online games.
   - Customizing the display of page in browser without reloading or reopening the page.
4. Here are some popular client-side scripting languages VBScript, JavaScript, Hypertext Processor(PHP).

Server-Side Scripts :

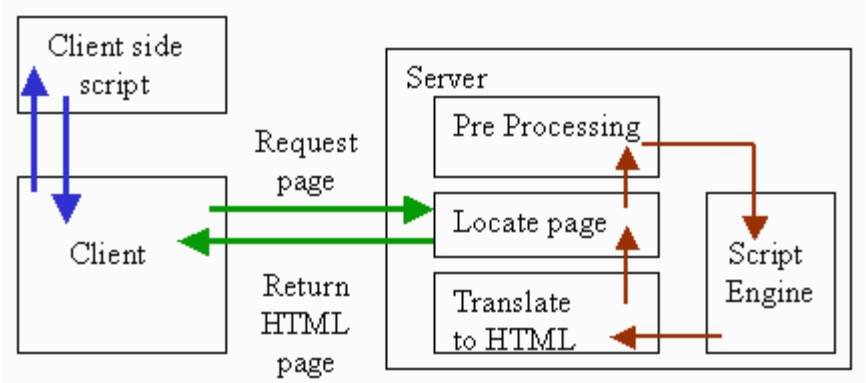
1. Server-side scripting is responsible for the completion or carrying out a task at the server-end and then sending the result to the client-end.
2. In server-side script, it doesn't matter which browser is being used at client-end, because the server does all the work.
3. Server-side scripting is mainly used when the information is sent to a server and to be processed at the server-end. Some sample uses of server-scripting can be :
   - Password Protection.
   - Browser Customization (sending information as per the requirements of client-end browser)
   - Form Processing
   - Building/Creating and displaying pages created from a database.
   - Dynamically editing changing or adding content to a web-page.
4. Here are some popular server-side scripting languages PHP, Perl, ASP (Active Server Pages), JSP (Java Server Pages).

Client side scripting:

web browsers execute client side scripting. It is use when browsers has all code. Source code used to transfer from web server to user's computer over internet and run directly on browsers. It is also used for validations and functionality for user events.

It allows for more interactivity. It usually performs several actions without going to user. It cannot be basically used to connect to databases on web server. These scripts cannot access file system that resides at web browser. Pages are altered on basis of users choice.

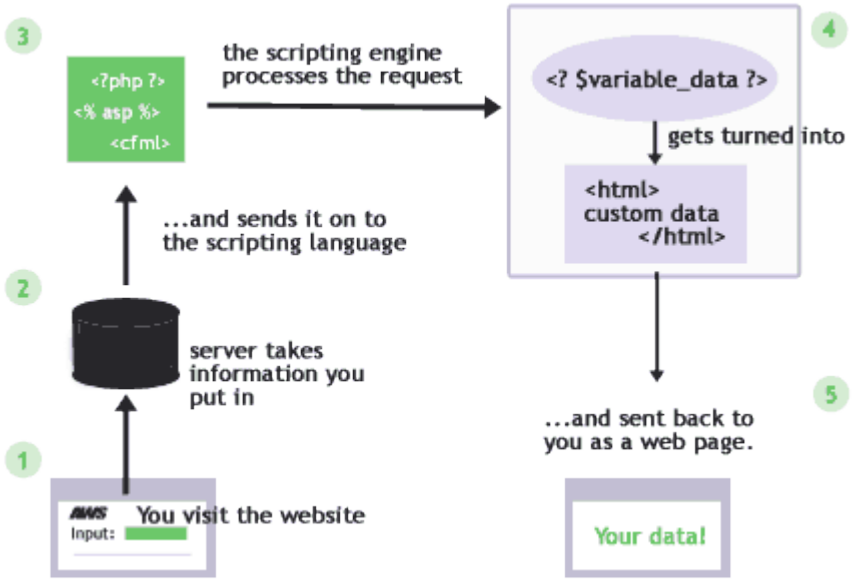
It can also used to create "cookies" that store data on user's computer.



Server side scripting:

Web servers are used to execute server side scripting. They are basically used to create dynamic pages. It can also access the file system residing at web server. Server-side environment that runs on a scripting language is a web-server.

Scripts can be written in any of a number of server-side scripting language available. It is used to retrieve and generate content for dynamic pages. It is used to require to download plugins. In this load times are generally faster than client-side scripting. When you need to store and retrieve information a database will be used to contain data. It can use huge resources of server. It reduces client-side computation overhead. Server sends pages to request of user/client.



Difference Between Server Side Scripting and Client-Side Scripting

Parameters	Server Side Scripting	Client-Side Scripting
Script Running	For server-side scripting, a web server serves as the medium for running it. They create the pages that one would send to the browser.	The script for the client-side scripting runs by using a browser. It is present already in any user's computer.
Uses	We use the server-side scripting at the back-end, where the source code stays hidden from the browser (client-side)- making it non-viewable.	We use the client-side scripting at the front end, and any user can view it using the browser itself.
Occurrence	Server-side scripting occurs whenever a browser initiates a request for it. As a result, many dynamic pages get created on the basis of several conditions.	Client-side scripting occurs when all the codes that a browser possesses in a page later change/ alter according to any user's input.
Operation	Any server is capable of carrying out server-side scripting, but it can't carry out client-side scripting.	A typical browser performs the client-side scripting after it receives a page (that the server sends).

Execution	The server-side scripting occurs on a remote computer. Thus, the response we get is slower as compared to that of the client-side scripting.	The client-side scripting occurs on the local computers. Thus, in this case, we get a comparatively quicker response in comparison with the server-side scripting.
Suitability	The server-side scripting works well for the areas that require the loading of a dynamic type of data and information.	The client-side scripting works well in those cases that require user interaction.
Connection to the Server Database	The server-side scripting assists a user in connecting to the database that already exists in the concerned web server.	The client-side server doesn't connect to those databases that exist primarily on the concerned web server.
Access To Various Files	The server-side scripting has complete access to all the files present in any web server.	The client-side scripting has no access to the files that exist in a web server.
Languages	Languages like Ruby on Rails, Perl, ASP, Python, ColdFusion, PHP, etc., come into play in the case of server-side scripting.	Languages like VB Script, CSS, HTML, Javascript, etc., are very common in the case of client-side scripting.
Security	The server-side scripting is way more secure as compared to the client-side one. It is because the scripts of the server-side stay hidden from any random clients.	The client-side scripts are much less secure as compared to the server-side ones. It is because these scripts don't stay hidden from any random client's end.

Application of Scripting Languages:

Scripting languages are used in many areas:

- Scripting languages are used in web applications. It is used in server side as well as client side. Server side scripting languages are: JavaScript, PHP, Perl etc. and client side scripting languages are: JavaScript, AJAX, jQuery etc.
- Scripting languages are used in system administration. For example: Shell, Perl, Python scripts etc.
- It is used in Games application and Multimedia.
- It is used to create plugins and extensions for existing applications.

Advantages of scripting languages:

- **Easy learning:** The user can learn to code in scripting languages quickly, not much knowledge of web technology is required.
- **Fast editing:** It is highly efficient with the limited number of data structures and variables to use.
- **Interactivity:** It helps in adding visualization interfaces and combinations in web pages. Modern web pages demand the use of scripting languages. To create enhanced web pages, fascinated visual description which includes background and foreground colors and so on.
- **Functionality:** There are different libraries which are part of different scripting languages. They help in creating new applications in web browsers and are different from normal programming languages.

UNIT-9

Server Side Programming using PHP

INTRODUCTION

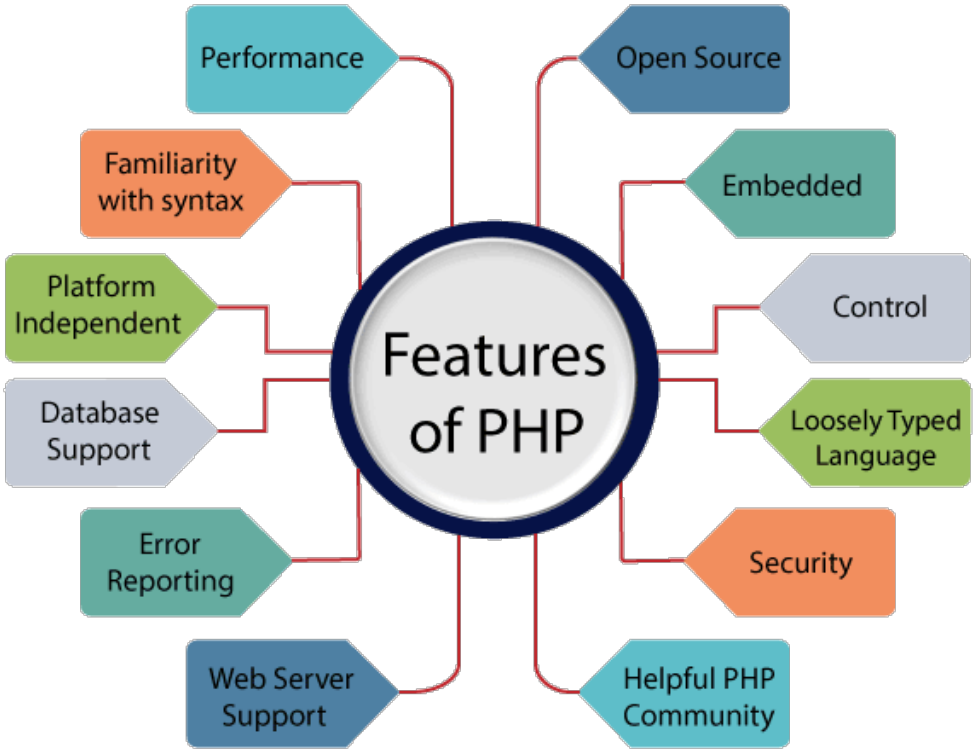
PHP is an open-source, interpreted, and object-oriented scripting language that can be executed at the server-side. PHP is well suited for web development. Therefore, it is used to develop web applications (an application that executes on the server and generates the dynamic page.).

PHP was created by **Rasmus Lerdorf in 1994** but appeared in the market in 1995. **PHP 7.4.0** is the latest version of PHP, which was released on **28 November**. Some important points need to be noticed about PHP are as followed:

- PHP stands for Hypertext Preprocessor.
- PHP is an interpreted language, i.e., there is no need for compilation.
- PHP is faster than other scripting languages, for example, ASP and JSP.
- PHP is a server-side scripting language, which is used to manage the dynamic content of the website.
- PHP can be embedded into HTML.
- PHP is an object-oriented language.
- PHP is an open-source scripting language.
- PHP is simple and easy to learn language.
- PHP runs on various platforms (Windows, Linux, Unix, Mac OS X, etc.)
- PHP is compatible with almost all servers used today (Apache, IIS, etc.)
- PHP supports a wide range of databases
- PHP is free. Download it from the official PHP resource: www.php.net
- PHP is easy to learn and runs efficiently on the server side

FEATURES

PHP is very popular language because of its simplicity and open source. There are some important features of PHP given below:



- Performance
- Open Source
- Familiarity with syntax
- Embedded
- Platform Independent
- Database Support
- Error Reporting
- Loosely Typed Language
- Web servers Support
- Security
- Control
- Helpful PHP Community

HOW PHP WORKS ?

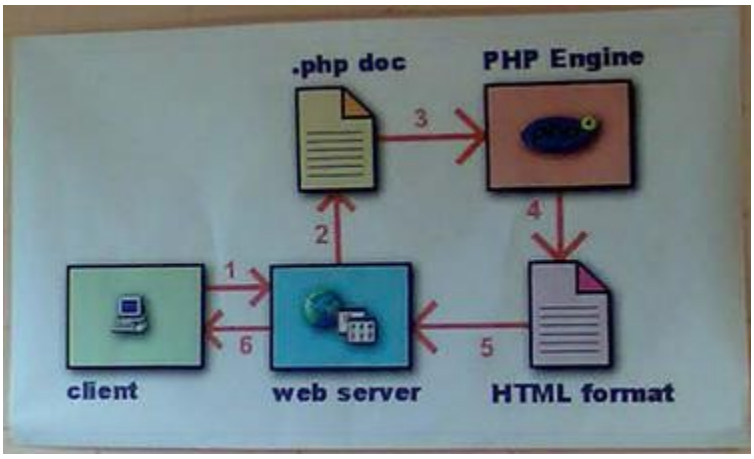
The normal process a Web server goes through to deliver a page to a browser is as follows. It all begins when a browser makes a request for a Web page. Based on the URL,the browser resolves the address of the Web server, identifies the page it would like, and gives any other information the Web server may need. Some of this information is about the browser itself, like its name (Mozilla), its version (4.08), or the operating system (Linux). Other information given the Web server could include text the user typed into form fields.

If the request is for an HTML file, the Web server will simply find the file, tell the browser to expect some HTML text, and then send the contents of the file. The browser gets the contents and begins rendering the page based on the HTML code. If you have been programming HTML for any length of time, this will be clear to you.

Hopefully you have also had some experience with CGI scripts. When a Web server gets a request for a CGI, it can't just send the contents of the file. It must execute the script first. The script will generate some HTML code, which then gets sent to the browser. As far as the browser is concerned, it's just getting HTML. The Web server does a bunch of work that it gets very little recognition for,but Web servers rarely get the respect they deserve. The medium is definitely not the message.

When a PHP page is requested, it is processed exactly like a CGI, at least to the extent that the script is not simply sent to the browser. It is first passed through the PHP engine, which gives the Web server HTML text.

What happens when the user clicks the stop button before the page finishes downloading? The Web server detects this situation and usually terminates the PHP script. It is possible to force a script to finish despite an aborted connection. You may also allow the script to terminate but execute special code first.



PHP | BASIC SYNTAX

PHP or Hypertext Preprocessor is a widely used open-source general purpose scripting language and can be embedded with HTML. PHP files are saved with ".php" extension. PHP scripts can be written anywhere in the document within PHP tags along with normal HTML.

- **Canonical PHP Tags**:

The script starts with **<?php** and ends with **?>** . These tags are also called 'Canonical PHP tags'. Every PHP command ends with a semi-colon (**;**). Let's look at the *hello world* program in PHP:

```
<?php

# Here echo command is used to print

echo "Hello, world!";

?>
```

- **SGML or Short HTML Tags**:

These are the shortest option to initialize a PHP code. The script starts with **<?** and ends with **?>**. This will only work by setting the short_open_tag setting in php.ini file to 'on'.

```
<?
```

```
# Here echo command will only work if

# setting is done as said before

echo "Hello, world!";

?>
```

- **HTML Script Tags**:

  These are implemented using script tags. This syntax is removed in PHP 7.0.0. so its no more used.

```
<script language="php">

echo "hello world!";

</script>
```

- **ASP Style Tags**:

  To use this we need to set the configuration of php.ini file. These are used by Active Server Pages to describe code blocks. These starts with **<%** and ends with **%>**.

```
<%

# Can only be written if setting is turned on

# to allow %

echo "hello world";

%>
```

PHP VARIABLES

Variables are "containers" for storing information. In PHP, a variable starts with the $ sign, followed by the name of the variable:

Example

```php
<?php
$txt = "Hello world!";
$x = 5;
$y = 10.5;
?>
```

When you assign a text value to a variable, put quotes around the value. PHP has no command for declaring a variable. It is created the moment you first assign a value to it.

Rules for PHP variables:

- A variable starts with the $ sign, followed by the name of the variable
- A variable name must start with a letter or the underscore character
- A variable name cannot start with a number
- A variable name can only contain alpha-numeric characters and underscores (A-z, 0-9, and _)
- Variable names are case-sensitive ($age and $AGE are two different variables)

PHP Data Types

Variables can store data of different types, and different data types can do different things.PHP supports the following data types:

- String
- Integer
- Float (floating point numbers - also called double)
- Boolean
- Array
- Object
- NULL
- Resource

❖ **String**

A string is a sequence of characters, like "Hello world!".A string can be any text inside quotes. You can use single or double quotes:

EXAMPLE:-

```
$x = "Hello world!";
```

- **Integer**

An integer data type is a non-decimal number between -2,147,483,648 and 2,147,483,647.

Rules for integers:

- An integer must have at least one digit
- An integer must not have a decimal point
- An integer can be either positive or negative
- Integers can be specified in: decimal (base 10), hexadecimal (base 16), octal (base 8), or binary (base 2) notation

EXAMPLE:-

```
$x = 5985;
```

- **Float**

A float (floating point number) is a number with a decimal point or a number in exponential form.

EXAMPLE:-

```
$x = 10.365;
```

- **Boolean**

A Boolean represents two possible states: TRUE or FALSE.

```
$x = true;
$y = false;
```

Booleans are often used in conditional testing.

- **Array**

  An array stores multiple values in one single variable.

EXAMPLE:-

```
<?php
$cars = array("Volvo","BMW","Toyota");
var_dump($cars);
?>
```

- **PHP Object**
  - Classes and objects are the two main aspects of object-oriented programming.
  - A class is a template for objects, and an object is an instance of a class.
  - When the individual objects are created, they inherit all the properties and behaviors from the class, but each object will have different values for the properties.
  - Let's assume we have a class named Car. A Car can have properties like model, color, etc. We can define variables like $model, $color, and so on, to hold the values of these properties.
  - When the individual objects (Volvo, BMW, Toyota, etc.) are created, they inherit all the properties and behaviors from the class, but each object will have different values for the properties.

- **NULL Value**

Null is a special data type which can have only one value: NULL.A variable of data type NULL is a variable that has no value assigned to it.If a variable is created without a value, it is automatically assigned a value of NULL.Variables can also be emptied by setting the value to NULL:

EXAMPLE:-

```
$x = null;
```

PHP OPERATORS

Operators are used to perform operations on variables and values.PHP divides the operators in the following groups:

- Arithmetic operators
- Assignment operators
- Comparison operators
- Increment/Decrement operators
- Logical operators
- String operators
- Array operators
- Conditional assignment operators

❖ PHP Arithmetic Operators

The PHP arithmetic operators are used with numeric values to perform common arithmetical operations, such as addition, subtraction, multiplication etc.

Operator	Name	Example	Result
+	Addition	$x + $y	Sum of $x and $y

-	Subtraction	$x - $y	Difference of $x and $y
*	Multiplication	$x * $y	Product of $x and $y
/	Division	$x / $y	Quotient of $x and $y
%	Modulus	$x % $y	Remainder of $x divided by $y
**	Exponentiation	$x ** $y	Result of raising $x to the $y'th power

- **PHP Assignment Operators**

The PHP assignment operators are used with numeric values to write a value to a variable.

The basic assignment operator in PHP is "=". It means that the left operand gets set to the value of the assignment expression on the right.

Assignment	Same as...	Description
x = y	x = y	The left operand gets set to the value of the expression on the right
x += y	x = x + y	Addition

x -= y	x = x - y	Subtraction
x *= y	x = x * y	Multiplication
x /= y	x = x / y	Division
x %= y	x = x % y	Modulus

- **PHP Comparison Operators**

The PHP comparison operators are used to compare two values (number or string):

Operator	Name	Example	Result
==	Equal	$x == $y	Returns true if $x is equal to $y
===	Identical	$x === $y	Returns true if $x is equal to $y, and they are of the same type
!=	Not equal	$x != $y	Returns true if $x is not equal to $y
<>	Not equal	$x <> $y	Returns true if $x is not equal to $y

!==	Not identical	$x !== $y	Returns true if $x is not equal to $y, or they are not of the same type
>	Greater than	$x > $y	Returns true if $x is greater than $y
<	Less than	$x < $y	Returns true if $x is less than $y
>=	Greater than or equal to	$x >= $y	Returns true if $x is greater than or equal to $y
<=	Less than or equal to	$x <= $y	Returns true if $x is less than or equal to $y
<=>	Spaceship	$x <=> $y	Returns an integer less than, equal to, or greater than zero, depending on if $x is less than, equal to, or greater than $y. Introduced in PHP 7.

- **PHP Increment / Decrement Operators**

The PHP increment operators are used to increment a variable's value.The PHP decrement operators are used to decrement a variable's value.

Operator	Name	Description

++$x	Pre-increment	Increments $x by one, then returns $x
$x++	Post-increment	Returns $x, then increments $x by one
--$x	Pre-decrement	Decrements $x by one, then returns $x
$x--	Post-decrement	Returns $x, then decrements $x by one

- **PHP Logical Operators**

The PHP logical operators are used to combine conditional statements.

Operator	Name	Example	Result
and	And	$x and $y	True if both $x and $y are true
or	Or	$x or $y	True if either $x or $y is true
xor	Xor	$x xor $y	True if either $x or $y is true, but not both
&&	And	$x && $y	True if both $x and $y are true

||	Or	$x || $y	True if either $x or $y is true
!	Not	!$x	True if $x is not true

- **PHP String Operators**

PHP has two operators that are specially designed for strings.

Operator	Name	Example	Result
.	Concatenation	$txt1 . $txt2	Concatenation of $txt1 and $txt2
.=	Concatenation assignment	$txt1 .= $txt2	Appends $txt2 to $txt1

- **PHP Array Operators**

The PHP array operators are used to compare arrays.

Operator	Name	Example	Result
+	Union	$x + $y	Union of $x and $y

==	Equality	$x == $y	Returns true if $x and $y have the same key/value pairs
===	Identity	$x === $y	Returns true if $x and $y have the same key/value pairs in the same order and of the same types
!=	Inequality	$x != $y	Returns true if $x is not equal to $y
<>	Inequality	$x <> $y	Returns true if $x is not equal to $y
!==	Non-identity	$x !== $y	Returns true if $x is not identical to $y

- **PHP Conditional Assignment Operators**

The PHP conditional assignment operators are used to set a value depending on conditions:

Operator	Name	Example	Result
?:	Ternary	$x = *expr1* ? *expr2* : *expr3*	Returns the value of $x. The value of $x is *expr2* if *expr1* = TRUE. The value of $x is *expr3* if *expr1* = FALSE

??	Null coalescing	$x = *expr1* ?? *expr2*	Returns the value of $x. The value of $x is *expr1* if *expr1* exi sts, and is not NULL. If *expr1* does not exist, or is NULL, the value of $x is *expr2*. Introduced in PHP 7

PHP DECISION MAKING

PHP if else statement is used to test condition. There are various ways to use if statement in PHP.

- if
- if-else
- if-else-if
- nested if

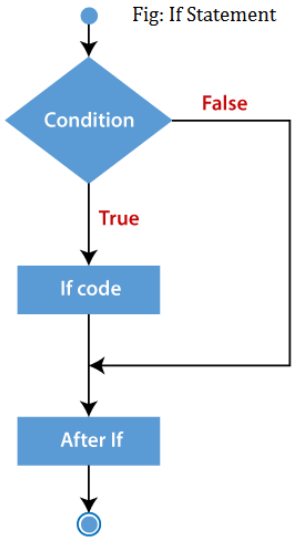
❖ PHP If Statement

PHP if statement allows conditional execution of code. It is executed if condition is true. If statement is used to executes the block of code exist inside the if statement only if the specified condition is true.

Syntax

```
if(condition)
{
//code to be executed
}
```

Flowchart



Example

```
<?php
$num=12;
if($num<100)
{
echo "$num is less than 100";
}
?>
```

Output:

```
12 is less than 100
```

❖ PHP If-else Statement

PHP if-else statement is executed whether condition is true or false. If-else statement is slightly different from if statement. It executes one block of code if the specified condition is **true** and another block of code if the condition is **false**.

Syntax

```
if(condition){
//code to be executed if true
}else{
//code to be executed if false
}
```

Flowchart

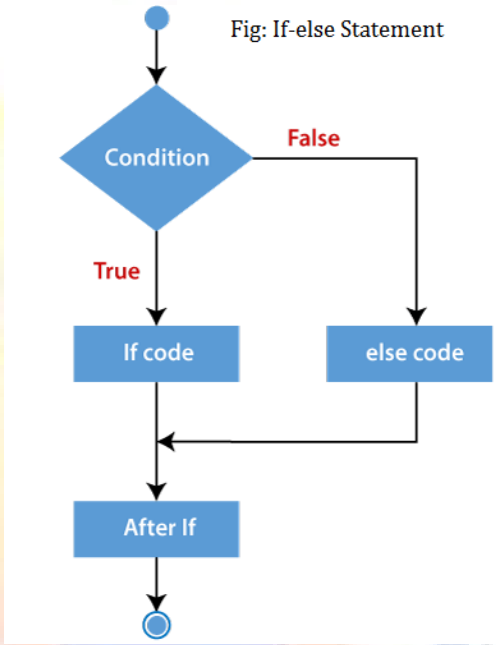


Fig: If-else Statement

Example

```
<?php
$num=12;
if($num%2==0){
echo "$num is even number";
}else{
echo "$num is odd number";
}
```

```
?>
```

Output:

```
12 is even number
```

❖ PHP If-else-if Statement

The PHP if-else-if is a special statement used to combine multiple if-else statements. So, we can check multiple conditions using this statement.

Syntax

```
if (condition1){
//code to be executed if condition1 is true
} elseif (condition2){
//code to be executed if condition2 is true
} elseif (condition3){
//code to be executed if condition3 is true
....
}  else{
//code to be executed if all given conditions are false
}
```

Flowchart

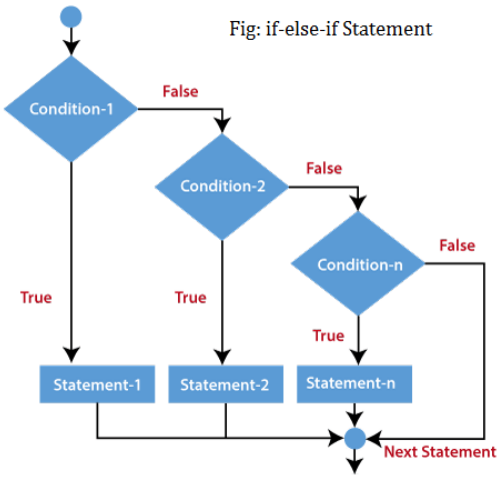


Fig: if-else-if Statement

Example

```
<?php
   $marks=69;
   if ($marks<33){
      echo "fail";
   }
   else if ($marks>=34 && $marks<50) {
      echo "D grade";
   }
   else if ($marks>=50 && $marks<65) {
     echo "C grade";
   }
   else if ($marks>=65 && $marks<80) {
      echo "B grade";
   }
   else if ($marks>=80 && $marks<90) {
      echo "A grade";
   }
```

```
    else if ($marks>=90 && $marks<100) {
        echo "A+ grade";
    }
    else {
        echo "Invalid input";
    }
?>
```

Output:

```
B Grade
```

❖ PHP nested if Statement

The nested if statement contains the if block inside another if block. The inner if statement executes only when specified condition in outer if statement is **true**.

Syntax

```
if (condition) {
//code to be executed if condition is true
if (condition) {
//code to be executed if condition is true
}
}
```

Flowchart

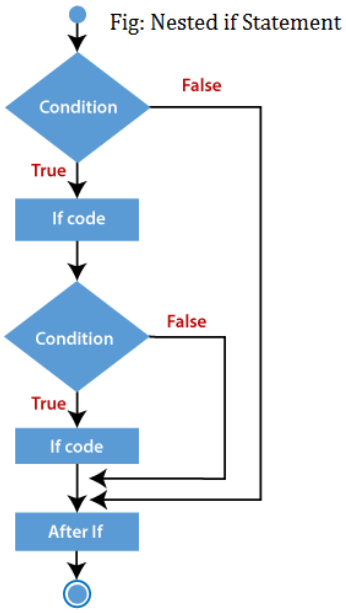


Fig: Nested if Statement

Example

```
<?php
    $age = 23;
$nationality = "Indian";
//applying conditions on nationality and age
if ($nationality == "Indian")
{
if ($age >= 18) {
    echo "Eligible to give vote";
}
else {
    echo "Not eligible to give vote";
}
}
?>
```

Output:

```
Eligible to give vote
```

❖ PHP Switch

PHP switch statement is used to execute one statement from multiple conditions. It works like PHP if-else-if statement.

Syntax

```
switch(expression){
case value1:
 //code to be executed
 break;
case value2:
 //code to be executed
 break;
.....
default:
 code to be executed if all cases are not matched;
}
```

Flowchart

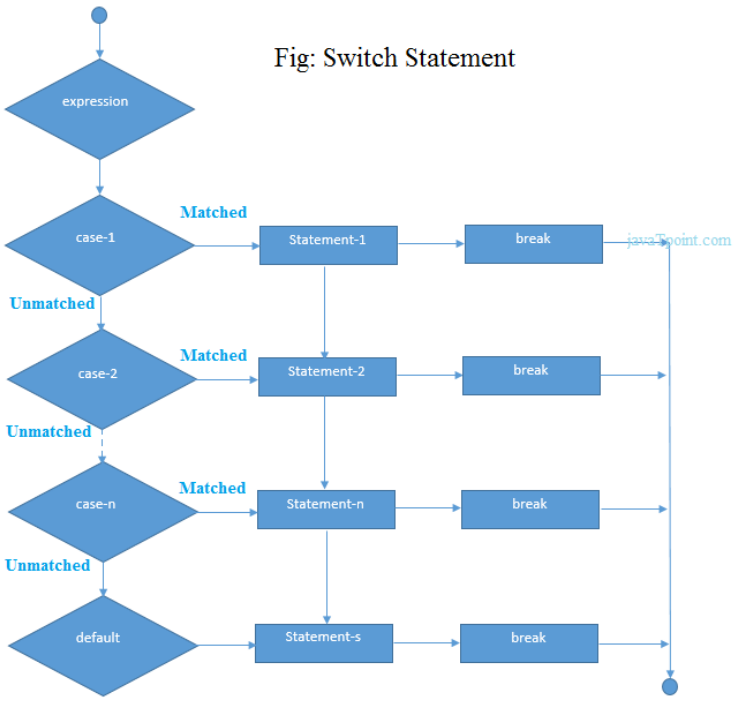


Fig: Switch Statement

Example

```
<?php
   $ch = 'U';
   switch ($ch)
   {
      case 'a':
         echo "Given character is vowel";
         break;
      case 'e':
         echo "Given character is vowel";
         break;
      case 'i':
         echo "Given character is vowel";
         break;
```

```
        case 'o':
            echo "Given character is vowel";
            break;
        case 'u':
            echo "Given character is vowel";
            break;
        case 'A':
            echo "Given character is vowel";
            break;
        case 'E':
            echo "Given character is vowel";
            break;
        case 'I':
            echo "Given character is vowel";
            break;
        case 'O':
            echo "Given character is vowel";
            break;
        case 'U':
            echo "Given character is vowel";
            break;
        default:
            echo "Given character is consonant";
            break;
    }
?>
```

Output:

```
Given character is vowel
```

PHP LOOP

❖ PHP For Loop

PHP for loop can be used to traverse set of code for the specified number of times. It should be used if the number of iterations is known otherwise use while loop. This means for loop is used when you already know how many times you want to execute a block of code. It allows users to put all the loop related statements in one place.

Syntax

```
for(initialization; condition; increment/decrement){
//code to be executed
}
```

Parameters

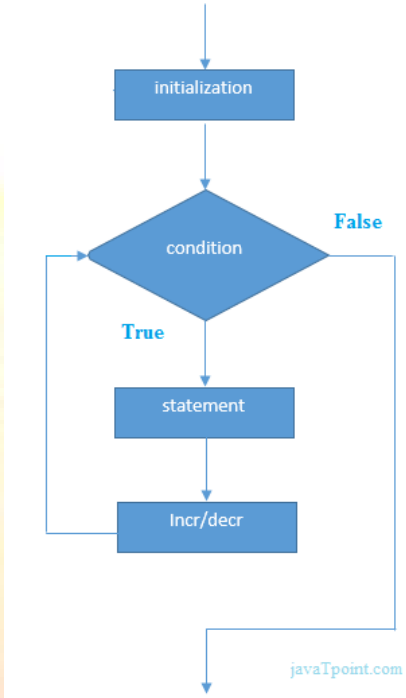
The php for loop is similar to the java/C/C++ for loop. The parameters of for loop have the following meanings:

initialization - Initialize the loop counter value. The initial value of the for loop is done only once. This parameter is optional.

condition - Evaluate each iteration value. The loop continuously executes until the condition is false. If TRUE, the loop execution continues, otherwise the execution of the loop ends.

Increment/decrement - It increments or decrements the value of the variable.

Flowchart



Example

```
<?php
for($n=1;$n<=10;$n++){
echo "$n<br/>";
}
?>
```

Output:

1

```
2
3
4
5
6
7
8
9
10
```

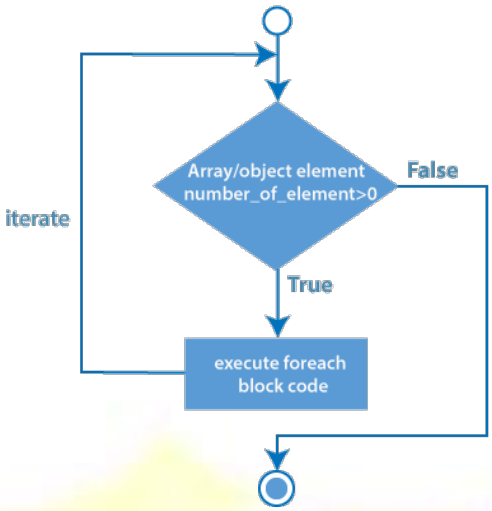
❖ PHP foreach loop

The foreach loop is used to traverse the array elements. It works only on array and object. It will issue an error if you try to use it with the variables of different datatype. The foreach loop works on elements basis rather than index. It provides an easiest way to iterate the elements of an array. In foreach loop, we don't need to increment the value.

Syntax

```
foreach ($array as $value) {
   //code to be executed
}
```

Flowchart



Example

PHP program to print array elements using foreach loop.

```
<?php
   //declare array
   $season = array ("Summer", "Winter", "Autumn", "Rainy");

   //access array elements using foreach loop
   foreach ($season as $element) {
      echo "$element";
      echo "</br>";
   }
?>
```

Output:

```
Summer
Winter
Autumn
Rainy
```

❖ PHP While Loop

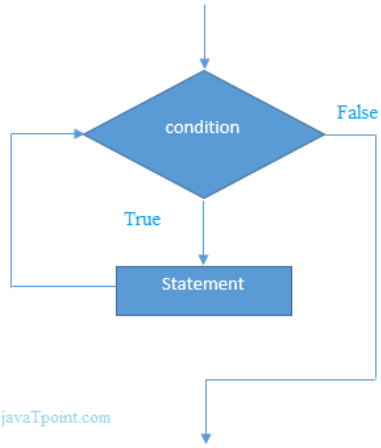
PHP while loop can be used to traverse set of code like for loop. The while loop executes a block of code repeatedly until the condition is FALSE. Once the condition gets FALSE, it exits from the body of loop. It should be used if the number of iterations is not known.

The while loop is also called an **Entry control loop** because the condition is checked before entering the loop body. This means that first the condition is checked. If the condition is true, the block of code will be executed.

Syntax

```
while(condition){
//code to be executed
}
```

Flowchart



Example

```
<?php
$n=1;
while($n<=10){
```

```
echo "$n<br/>";

$n++;

}

?>
```

Output:

```
1
2
3
4
5
6
7
8
9
10
```

❖ PHP do-while loop

PHP do-while loop can be used to traverse set of code like php while loop. The PHP do-while loop is guaranteed to run at least once.

The PHP do-while loop is used to execute a set of code of the program several times. If you have to execute the loop at least once and the number of iterations is not even fixed, it is recommended to use the **do-while** loop.

It executes the code at least one time always because the condition is checked after executing the code.

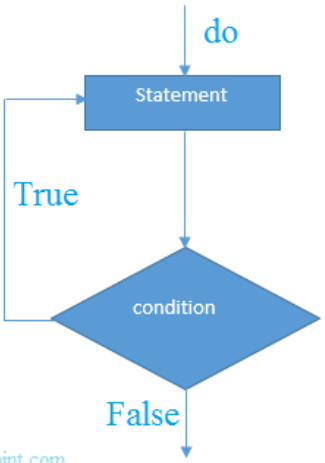
The do-while loop is very much similar to the while loop except the condition check. The main difference between both loops is that while loop checks the condition at the beginning, whereas do-while loop checks the condition at the end of the loop.

Syntax

```
do{

//code to be executed
```

```
}while(condition);
```

Flowchart



Example

```
<?php
$n=1;
do{
echo "$n<br/>";
$n++;
}while($n<=10);
?>
```

Output:

```
1
2
3
4
5
6
7
8
9
10
```

Difference between while and do-while loop

while Loop	do-while loop
The while loop is also named as **entry control loop**.	The do-while loop is also named as **exit control loop**.
The body of the loop does not execute if the condition is false.	The body of the loop executes at least once, even if the condition is false.
Condition checks first, and then block of statements executes.	Block of statements executes first and then condition checks.
This loop does not use a semicolon to terminate the loop.	Do-while loop use semicolon to terminate the loop.

PHP String

PHP string is a sequence of characters i.e., used to store and manipulate text. PHP supports only 256-character set and so that it does not offer native Unicode support. There are 2 ways to specify a string literal in PHP.

1. single quoted
2. double quoted

Single Quoted

We can create a string in PHP by enclosing the text in a single-quote. It is the easiest way to specify string in PHP.

For specifying a literal single quote, escape it with a backslash (\) and to specify a literal backslash (\) use double backslash (\\). All the other instances with backslash such as \r or \n, will be output same as they specified instead of having any special meaning.

Example 1

```
<?php
```

```
$str='Hello text within single quote';

echo $str;

?>
```

Output:

```
Hello text within single quote
```

We can store multiple line text, special characters, and escape sequences in a single-quoted PHP string.

Example 2

```
<?php

$str1='Hello text

multiple line

text within single quoted string';

$str2='Using double "quote" directly inside single quoted string';

$str3='Using escape sequences \n in single quoted string';

echo "$str1 <br/> $str2 <br/> $str3";

?>
```

Output:

```
Hello text multiple line text within single quoted string
Using double "quote" directly inside single quoted string
Using escape sequences \n in single quoted string
```

Example 3

```
<?php

$num1=10;

$str1='trying variable $num1';

$str2='trying backslash n and backslash t inside single quoted string \n \t';
```

```
$str3='Using single quote \'my quote\' and \\backslash';

echo "$str1 <br/> $str2 <br/> $str3";

?>
```

Output:

```
trying variable $num1
trying backslash n and backslash t inside single quoted string \n \t
Using single quote 'my quote' and \backslash
```

Note: ***In single quoted PHP strings, most escape sequences and variables will not be interpreted. But, we can use single quote through \' and backslash through \\ inside single quoted PHP strings.***

Double Quoted

In PHP, we can specify string through enclosing text within double quote also. But escape sequences and variables will be interpreted using double quote PHP strings.

Example 1

```
<?php

$str="Hello text within double quote";

echo $str;

?>
```

Output:

```
Hello text within double quote
```

Now, you **can't use double quote directly** inside double quoted string.

Example 2

```
<?php

$str1="Using double "quote" directly inside double quoted string";

echo $str1;

?>
```

Output:

```
Parse error: syntax error, unexpected 'quote' (T_STRING) in C:\wamp\www\string1.php
on line 2
```

We **can store multiple line text, special characters and escape sequences** in a double quoted PHP string.

Example 3

```
<?php
$str1="Hello text
multiple line
text within double quoted string";
$str2="Using double \"quote\" with backslash inside double quoted string";
$str3="Using escape sequences \n in double quoted string";
echo "$str1 <br/> $str2 <br/> $str3";
?>
```

Output:

```
Hello text multiple line text within double quoted string
Using double "quote" with backslash inside double quoted string
Using escape sequences in double quoted string
```

In double quoted strings, **variable will be interpreted**.

Example 4

```
<?php
$num1=10;
echo "Number is: $num1";
?>
```

Output:

```
Number is: 10
```

PHP String Function Examples

String functions in PHP are used to manipulate string values.

Function	Description	Example	Output
strtolower	Used to convert all string characters to lower case letters	echo strtolower('Benjamin');	outputs benjamin
strtoupper	Used to convert all string characters to upper case letters	echo strtoupper('george w bush');	outputs GEORGE W BUSH
strlen	The string length function is used to count the number of character in a string. Spaces in between characters are also counted	echo strlen('united states of america');	24
explode	Used to convert strings into an array variable	$settings = explode(';', "host=localhost; db=sales; uid=root; pwd=demo"); print_r($settings);	Array ([0] => host=localhost [1] => db=sales [2] => uid=root [3] => pwd=demo)
substr	Used to return part of the string. It accepts three (3) basic parameters. The	$my_var = 'This is a really long sentence that I wish to cut short';	This is a re…

Function	Description	Example	Output
	first one is the string to be shortened, the second parameter is the position of the starting point, and the third parameter is the number of characters to be returned.	echo substr($my_var,0, 12).'…';	
str_replace	Used to locate and replace specified string values in a given string. The function accepts three arguments. The first argument is the text to be replaced, the second argument is the replacement text and the third argument is the text that is analyzed.	echo str_replace ('the', 'that', 'the laptop is very expensive');	that laptop is very expensive
strpos	Used to locate the and return the position of a character(s) within a string. This function accepts two arguments	echo strpos('PHP Programing','Pro');	4

Function	Description	Example	Output
str_word_count	Used to count the number of words in a string.	echo str_word_count ('This is a really long sentence that I wish to cut short');	12
ucfirst	Make the first character of a string value upper case	echo ucfirst('respect');	Outputs Respect
lcfirst	Make the first character of a string value lower case	echo lcfirst('RESPECT');	Outputs rESPECT

PHP - Arrays

An array is a data structure that stores one or more similar type of values in a single value. For example if you want to store 100 numbers then instead of defining 100 variables its easy to define an array of 100 length.

There are three different kind of arrays and each array value is accessed using an ID c which is called array index.

Numeric array − An array with a numeric index. Values are stored and accessed in linear fashion.

Associative array − An array with strings as index. This stores element values in association with key values rather than in a strict linear index order.

Multidimensional array − An array containing one or more arrays and values are accessed using multiple indices

Numeric Array

These arrays can store numbers, strings and any object but their index will be represented by numbers. By default array index starts from zero.

Example

Following is the example showing how to create and access numeric arrays.

Here we have used **array()** function to create array. This function is explained in function reference.

```
<html>
   <body>

      <?php
         /* First method to create array. */
         $numbers = array( 1, 2, 3, 4, 5);

         foreach( $numbers as $value ) {
            echo "Value is $value <br />";
         }

         /* Second method to create array. */
         $numbers[0] = "one";
         $numbers[1] = "two";
         $numbers[2] = "three";
         $numbers[3] = "four";
         $numbers[4] = "five";

         foreach( $numbers as $value ) {
            echo "Value is $value <br />";
         }
      ?>

   </body>
</html>
```

This will produce the following result −

```
Value is 1
Value is 2
Value is 3
Value is 4
Value is 5
Value is one
Value is two
Value is three
```

```
Value is four
Value is five
```

Associative Arrays

The associative arrays are very similar to numeric arrays in term of functionality but they are different in terms of their index. Associative array will have their index as string so that you can establish a strong association between key and values.

To store the salaries of employees in an array, a numerically indexed array would not be the best choice. Instead, we could use the employees names as the keys in our associative array, and the value would be their respective salary.

NOTE − Don't keep associative array inside double quote while printing otherwise it would not return any value.

Example

```
<html>
   <body>

      <?php
         /* First method to associate create array. */
         $salaries = array("mohammad" => 2000, "qadir" => 1000, "zara"
=> 500);

         echo "Salary of mohammad is ". $salaries['mohammad'] . "<br
/>";
         echo "Salary of qadir is ".  $salaries['qadir']. "<br />";
         echo "Salary of zara is ".  $salaries['zara']. "<br />";

         /* Second method to create array. */
         $salaries['mohammad'] = "high";
         $salaries['qadir'] = "medium";
         $salaries['zara'] = "low";

         echo "Salary of mohammad is ". $salaries['mohammad'] . "<br
/>";
         echo "Salary of qadir is ".  $salaries['qadir']. "<br />";
         echo "Salary of zara is ".  $salaries['zara']. "<br />";
      ?>

   </body>
</html>
```

This will produce the following result −

```
Salary of mohammad is 2000
Salary of qadir is 1000
Salary of zara is 500
Salary of mohammad is high
Salary of qadir is medium
Salary of zara is low
```

Multidimensional Arrays

A multi-dimensional array each element in the main array can also be an array. And each element in the sub-array can be an array, and so on. Values in the multi-dimensional array are accessed using multiple index.

Example

In this example we create a two dimensional array to store marks of three students in three subjects −

This example is an associative array, you can create numeric array in the same fashion.

```
<html>
   <body>

      <?php
         $marks = array(
            "mohammad" => array (
               "physics" => 35,
               "maths" => 30,
               "chemistry" => 39
            ),

            "qadir" => array (
               "physics" => 30,
               "maths" => 32,
               "chemistry" => 29
            ),

            "zara" => array (
               "physics" => 31,
               "maths" => 22,
               "chemistry" => 39
            )
         );

         /* Accessing multi-dimensional array values */
         echo "Marks for mohammad in physics : " ;
         echo $marks['mohammad']['physics'] . "<br />";
```

```
         echo "Marks for qadir in maths : ";
         echo $marks['qadir']['maths'] . "<br />";

         echo "Marks for zara in chemistry : " ;
         echo $marks['zara']['chemistry'] . "<br />";
      ?>

   </body>
</html>
```

This will produce the following result −

```
Marks for mohammad in physics : 35
Marks for qadir in maths : 32
Marks for zara in chemistry : 39
```

PHP Array Functions

PHP provides various array functions to access and manipulate the elements of array. The important PHP array functions are given below.

1) PHP array() function

PHP array() function creates and returns an array. It allows you to create indexed, associative and multidimensional arrays.

Syntax

```
array array ([ mixed $... ] )
```

Example

```
<?php
$season=array("summer","winter","spring","autumn");
echo "Season are: $season[0], $season[1], $season[2] and $season[3]";
?>
```

Output:

```
Season are: summer, winter, spring and autumn
```

2) PHP array_change_key_case() function

PHP array_change_key_case() function changes the case of all key of an array.

Note: It changes case of key only.

Syntax

```
array array_change_key_case ( array $array [, int $case = CASE_LOWER ] )
```

Example

```
<?php
$salary=array("Sonoo"=>"550000","Vimal"=>"250000","Ratan"=>"200000");
print_r(array_change_key_case($salary,CASE_UPPER));
?>
```

Output:

```
Array ( [SONOO] => 550000 [VIMAL] => 250000 [RATAN] => 200000 )
```

Example

```
<?php
$salary=array("Sonoo"=>"550000","Vimal"=>"250000","Ratan"=>"200000");
print_r(array_change_key_case($salary,CASE_LOWER));
?>
```

Output:

```
Array ( [sonoo] => 550000 [vimal] => 250000 [ratan] => 200000 )
```

3) PHP array_chunk() function

PHP array_chunk() function splits array into chunks. By using array_chunk() method, you can divide array into many parts.

Syntax

```
array array_chunk ( array $array , int $size [, bool $preserve_keys = false ] )
```

Example

```
<?php
$salary=array("Sonoo"=>"550000","Vimal"=>"250000","Ratan"=>"200000");
print_r(array_chunk($salary,2));
?>
```

Output:

```
Array (
[0] => Array ( [0] => 550000 [1] => 250000 )
[1] => Array ( [0] => 200000 )
)
```

4) PHP count() function

PHP count() function counts all elements in an array.

Syntax

```
int count ( mixed $array_or_countable [, int $mode = COUNT_NORMAL ] )
```

Example

```
<?php
$season=array("summer","winter","spring","autumn");
echo count($season);
?>
```

Output:

```
4
```

5) PHP sort() function

PHP sort() function sorts all the elements in an array.

Syntax

```
bool sort ( array &$array [, int $sort_flags = SORT_REGULAR ] )
```

Example

```
<?php
$season=array("summer","winter","spring","autumn");
sort($season);
foreach( $season as $s )
{
  echo "$s<br />";
}
?>
```

Output:

```
autumn
spring
summer
winter
```

6) PHP array_reverse() function

PHP array_reverse() function returns an array containing elements in reversed order.

Syntax

```
array array_reverse ( array $array [, bool $preserve_keys = false ] )
```

Example

```
<?php
```

```
$season=array("summer","winter","spring","autumn");

$reverseseason=array_reverse($season);

foreach( $reverseseason as $s )

{

  echo "$s<br />";

}

?>
```

Output:

```
autumn
spring
winter
summer
```

7) PHP array_search() function

PHP array_search() function searches the specified value in an array. It returns key if search is successful.

Syntax

```
mixed array_search ( mixed $needle , array $haystack [, bool $strict = false ] )
```

Example

```
<?php

$season=array("summer","winter","spring","autumn");

$key=array_search("spring",$season);

echo $key;

?>
```

Output:

```
2
```

8) PHP array_intersect() function

PHP array_intersect() function returns the intersection of two array. In other words, it returns the matching elements of two array.

Syntax

```
array array_intersect ( array $array1 , array $array2 [, array $... ] )
```

Example

```
<?php
$name1=array("sonoo","john","vivek","smith");
$name2=array("umesh","sonoo","kartik","smith");
$name3=array_intersect($name1,$name2);
foreach( $name3 as $n )
{
  echo "$n<br />";
}
?>
```

Output:

```
sonoo
smith
```

Get and Post Methods in PHP

PHP provides two methods through which a client (browser) can send information to the server. These methods are given below, and discussed in detail:

1. GET method
2. POST method

Get and Post methods are the HTTP request methods used inside the <*form*> tag to send form data to the server.

HTTP: The Hypertext Transfer Protocol (HTTP) is designed to enable communications between clients and servers. HTTP works as a request-response protocol between a client and server. A web browser may be the client, and an application on a computer that hosts a website may be the server. A client (browser) submits an HTTP request to the server; then the server returns a response to the client. The response contains status information about the request and may also contain the requested content.

GET method

The **GET** method is used to submit the HTML form data. This data is collected by the predefined **$_GET variable** for processing.

The information sent from an HTML form using the GET method is visible to everyone in the browser's address bar, which means that all the variable names and their values will be displayed in the URL. Therefore, the get method is not secured to send sensitive information.

For Example

localhost/gettest.php?**username**=*Harry*&**bloodgroup**=*AB+*

The **bold** part in the above URL is the variables name and *italic* part contains the values for their corresponding variable.

Note that only a limited amount of information can be sent using the GET method.

With the help of an example, let's understand how the GET method works-

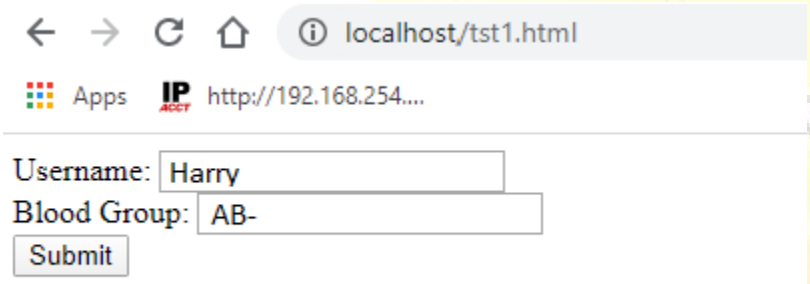
Example

The below code will display an HTML form containing two input fields and a submit button. In this HTML form, we used the method = "get" to submit the form data.

file: test1.html

```
<html>
  <body>
```

```
<form action = "gettest.php" method = "GET">
  Username: <input type = "text" name = "username" /> <br>
  Blood Group: <input type = "text" name = "bloodgroup" /> <br>
  <input type = "submit" />
</form>

</body>
</html>
```



Create gettest.php file, which will accept the data sent by HTML form.

file: gettest.php

```
<html>
  <body>

    Welcome <?php echo $_GET["username"]; ?> </br>
    Your blood group is: <?php echo $_GET["bloodgroup"]; ?>

  </body>
</html>
```

When the user will click on **Submit** button after filling the form, the URL sent to the server could look something like this:

localhost/gettest.php?username=Harry&bloodgroup=AB-

The output will look like the below output:

```
Welcome Harry
Your blood group is: AB-
```

Advantages of GET method (method = "get")

- You can bookmark the page with the specific query string because the data sent by the GET method is displayed in URL.
- GET requests can be cached.
- GET requests are always remained in the browser history.

Disadvantages of GET Method

- The GET method should not be used while sending any sensitive information.
- A limited amount of data can be sent using method = "get". This limit should not exceed 2048 characters.
- For security reasons, never use the GET method to send highly sensitive information like username and password, because it shows them in the URL.
- The GET method cannot be used to send binary data (such as images or word documents) to the server.

POST method

Similar to the GET method, the **POST** method is also used to submit the HTML form data. But the data submitted by this method is collected by the predefined super global variable **$_POST** instead of $_GET.

Unlike the GET method, it does not have a limit on the amount of information to be sent. The information sent from an HTML form using the POST method is not visible to anyone.

For Example

localhost/posttest.php

> ***Note that the "post" method is more secure than the "get" method because the data sent using the POST method is not visible to user.***

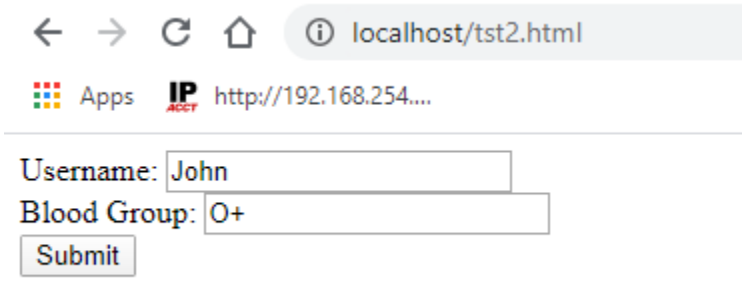
With the help of an example, let's understand how the POST method works-

Example

The below code will display an HTML form containing two input fields and a submit button. In this HTML form, we used the method = "post" to submit the form data.

file: test2.html

```
<html>
  <body>

   <form action = "posttest.php" method = "post">
    Username: <input type = "text" name = "username" /> <br>
    Blood Group: <input type = "text" name = "bloodgroup" /> <br>
    <input type = "submit" />
   </form>

  </body>
</html>
```



Now create **posttest.php** file to accept the data sent by HTML form.

file: posttest.php

```
<html>
  <body>

    Welcome <?php echo $_POST["username"]; ?> </br>
    Your blood group is: <?php echo $_POST["bloodgroup"]; ?>

  </body>
</html>
```

When the user will click on **Submit** button after filling the form, the URL sent to the server could look something like this:

localhost/posttest.php

The output will look like the below output:

```
Welcome Harry
Your blood group is: O+
```

Advantages of POST method (method = "post")

- The POST method is useful for sending any sensitive information because the information sent using the POST method is not visible to anyone.
- There is no limitation on size of data to be sent using the POST Method. You can send a large amount of information using this method.
- Binary and ASCII data can also be sent using the POST method.
- Data security depends on the HTTP protocol because the information sent using the POST method goes through the HTTP header. By using secure HTTP, you can ensure that your data is safe.

Disadvantages of POST Method

- POST requests do not cache.
- POST requests never remain in the browser history.
- It is not possible to bookmark the page because the variables are not displayed in URL.

$_REQUEST variable

The **$_REQUEST** variable is a **superglobal variable**, which can hold the content of both $_GET and $_POST variable. In other words, the PHP $_REQUEST variable is used to collect the form data sent by either GET or POST methods. It can also collect the data for $_COOKIE variable because it is not a method-specific variable.

Form Validation in PHP

An HTML form contains various input fields such as text box, checkbox, radio buttons, submit button, and checklist, etc. These input fields need to be validated, which ensures that the user has entered information in all the required fields and also validates that the information provided by the user is valid and correct.

There is no guarantee that the information provided by the user is always correct. PHP validates the data at the server-side, which is submitted by HTML form. You need to validate a few things:

1. Empty String
2. Validate String
3. Validate Numbers
4. Validate Email
5. Validate URL
6. Input length

Empty String

The code below checks that the field is not empty. If the user leaves the required field empty, it will show an error message. Put these lines of code to validate the required field.

```
if (emptyempty ($_POST["name"])) {
    $errMsg = "Error! You didn't enter the Name.";
        echo $errMsg;
```

```
} else {
    $name = $_POST["name"];
}
```

Validate String

The code below checks that the field will contain only alphabets and whitespace, for example - name. If the name field does not receive valid input from the user, then it will show an error message:

```
$name = $_POST ["Name"];
if (!preg_match ("/^[a-zA-z]*$/", $name) ) {
    $ErrMsg = "Only alphabets and whitespace are allowed.";
        echo $ErrMsg;
} else {
    echo $name;
}
```

Validate Number

The below code validates that the field will only contain a numeric value. **For example -** Mobile no. If the *Mobile no* field does not receive numeric data from the user, the code will display an error message:

```
$mobileno = $_POST ["Mobile_no"];
if (!preg_match ("/^[0-9]*$/", $mobileno) ){
    $ErrMsg = "Only numeric value is allowed.";
    echo $ErrMsg;
} else {
    echo $mobileno;
```

```
}
```

Validate Email

A valid email must contain @ and . symbols. PHP provides various methods to validate the email address. Here, we will use regular expressions to validate the email address.

The below code validates the email address provided by the user through HTML form. If the field does not contain a valid email address, then the code will display an error message:

```
$email = $_POST ["Email"];

$pattern = "^[_a-z0-9-]+(\.[_a-z0-9-]+)*@[a-z0-9-]+(\.[a-z0-9-]+)*(\.[a-z]{2,3})$^";

if (!preg_match ($pattern, $email) ){

   $ErrMsg = "Email is not valid.";

      echo $ErrMsg;

} else {

   echo "Your valid email address is: " .$email;

}
```

Input Length Validation

The input length validation restricts the user to provide the value between the specified range, for Example - Mobile Number. A valid mobile number must have 10 digits.

The given code will help you to apply the length validation on user input:

```
$mobileno = strlen ($_POST ["Mobile"]);

$length = strlen ($mobileno);


if ( $length < 10 && $length > 10) {

   $ErrMsg = "Mobile must have 10 digits.";

      echo $ErrMsg;
```

```
} else {
    echo "Your Mobile number is: " .$mobileno;
}
```

Validate URL

The below code validates the URL of website provided by the user via HTML form. If the field does not contain a valid URL, the code will display an error message, i.e., "URL is not valid".

```
$websiteURL = $_POST["website"];
if (!preg_match("/\b(?:(?:https?|ftp):\/\/|www\.)[-a-z0-9+&@#\/%?=~_|!:,.;]*[-a-z0-9+&@#\/%=~_|]/i",$website)) {
  $websiteErr = "URL is not valid";
  echo $websiteErr;
} else {
    echo "Website URL is: " .$websiteURL;
}
```

Button Click Validate

The below code validates that the user click on submit button and send the form data to the server one of the following method - get or post.

```
if (isset ($_POST['submit']) {
    echo "Submit button is clicked.";
    if ($_SERVER["REQUEST_METHOD"] == "POST") {
        echo "Data is sent using POST method ";
    }
} else {
    echo "Data is not submitted";
```

}

Note: Remember that validation and verification both are different from each other.

Now we will apply all these validations to an HTML form to validate the fields. Thereby you can learn in detail how these codes will be used to validation form.

Create a registration form using HTML and perform server-side validation. Follow the below instructions as given:

Create and validate a Registration form

```
<!DOCTYPE html>

<html>

<head>

<style>

.error {color: #FF0001;}

</style>

</head>

<body>


<?php

// define variables to empty values

$nameErr = $emailErr = $mobilenoErr = $genderErr = $websiteErr = $agreeErr = "";

$name = $email = $mobileno = $gender = $website = $agree = "";


//Input fields validation

if ($_SERVER["REQUEST_METHOD"] == "POST") {
```

```
//String Validation
  if (emptyempty($_POST["name"])) {
      $nameErr = "Name is required";
  } else {
     $name = input_data($_POST["name"]);
        // check if name only contains letters and whitespace
        if (!preg_match("/^[a-zA-Z ]*$/",$name)) {
           $nameErr = "Only alphabets and white space are allowed";
        }
}

//Email Validation
if (emptyempty($_POST["email"])) {
        $emailErr = "Email is required";
} else {
        $email = input_data($_POST["email"]);
        // check that the e-mail address is well-formed
        if (!filter_var($email, FILTER_VALIDATE_EMAIL)) {
           $emailErr = "Invalid email format";
        }
 }


//Number Validation
```

```
if (emptyempty($_POST["mobileno"])) {
        $mobilenoErr = "Mobile no is required";
} else {
        $mobileno = input_data($_POST["mobileno"]);
        // check if mobile no is well-formed
        if (!preg_match ("/^[0-9]*$/", $mobileno) ) {
        $mobilenoErr = "Only numeric value is allowed.";
        }
    //check mobile no length should not be less and greator than 10
    if (strlen ($mobileno) != 10) {
        $mobilenoErr = "Mobile no must contain 10 digits.";
        }
}

//URL Validation
if (emptyempty($_POST["website"])) {
    $website = "";
} else {
        $website = input_data($_POST["website"]);
        // check if URL address syntax is valid
        if (!preg_match("/\b(?:(?:https?|ftp):\/\/|www\.)[-a-z0-9+&@#\/%?=~_|!:,.;]*[-a-
z0-9+&@#\/%=~_|]/i",$website)) {
            $websiteErr = "Invalid URL";
        }
```

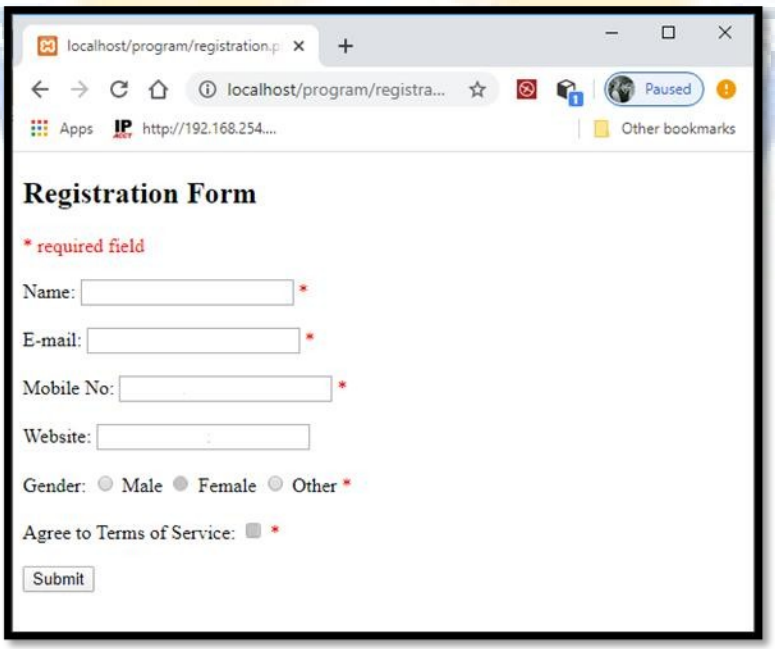
```
    }

    //Empty Field Validation
    if (emptyempty ($_POST["gender"])) {
        $genderErr = "Gender is required";
    } else {
        $gender = input_data($_POST["gender"]);
    }

    //Checkbox Validation
    if (!isset($_POST['agree'])){
        $agreeErr = "Accept terms of services before submit.";
    } else {
        $agree = input_data($_POST["agree"]);
    }
}
function input_data($data) {
  $data = trim($data);
  $data = stripslashes($data);
  $data = htmlspecialchars($data);
  return $data;
}
?>
```

```
<h2>Registration Form</h2>

<span class = "error">* required field </span>

<br><br>

<form method="post" action="<?php echo htmlspecialchars($_SERVER["PHP_SELF"]); ?>" >

  Name:

  <input type="text" name="name">

  <span class="error">* <?php echo $nameErr; ?> </span>

  <br><br>

  E-mail:

  <input type="text" name="email">

  <span class="error">* <?php echo $emailErr; ?> </span>

  <br><br>

  Mobile No:

  <input type="text" name="mobileno">

  <span class="error">* <?php echo $mobilenoErr; ?> </span>

  <br><br>

  Website:

  <input type="text" name="website">

  <span class="error"><?php echo $websiteErr; ?> </span>

  <br><br>

  Gender:

  <input type="radio" name="gender" value="male"> Male
```

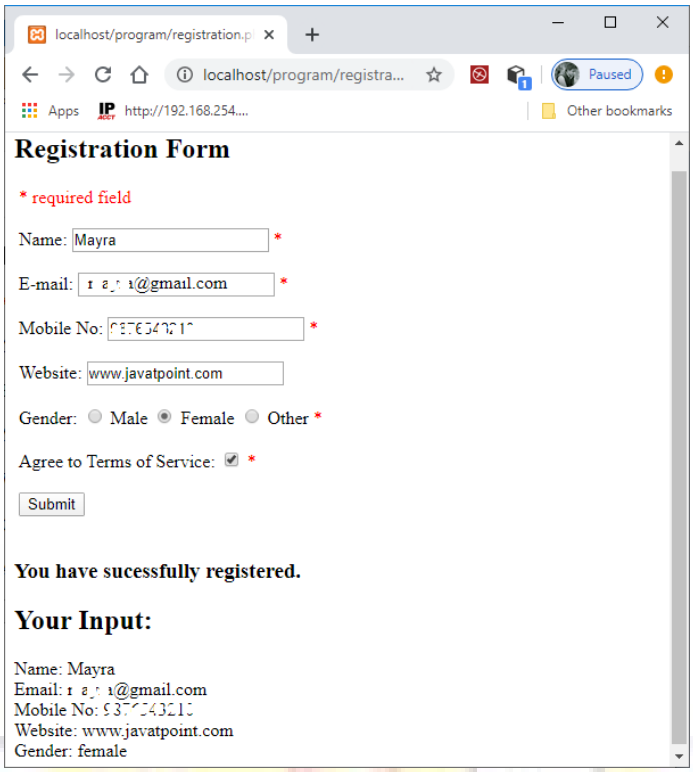
```
<input type="radio" name="gender" value="female"> Female

<input type="radio" name="gender" value="other"> Other

<span class="error">* <?php echo $genderErr; ?> </span>

<br><br>

Agree to Terms of Service:

<input type="checkbox" name="agree">

<span class="error">* <?php echo $agreeErr; ?> </span>

<br><br>

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

<br><br>

</form>


<?php

if(isset($_POST['submit'])) {

if($nameErr == "" && $emailErr == "" && $mobilenoErr == "" && $genderErr == ""
&& $websiteErr == "" && $agreeErr == "") {

    echo "<h3 color = #FF0001> <b>You have sucessfully registered.</b> </h3>";

    echo "<h2>Your Input:</h2>";

    echo "Name: " .$name;

    echo "<br>";

    echo "Email: " .$email;

    echo "<br>";

    echo "Mobile No: " .$mobileno;

    echo "<br>";
```

```
        echo "Website: " .$website;
        echo "<br>";
        echo "Gender: " .$gender;
    } else {
        echo "<h3> <b>You didn't filled up the form correctly.</b> </h3>";
    }
    }
?>

</body>
</html>
```

When the above code runs on a browser, the output will be like the screenshot below:



Fill the registration form and click on the Submit button. If all required information is provided correctly, the output will be displayed on the same page below the submit button. See the screenshot below:



Remember that we have not used a database to store the data for registered users.

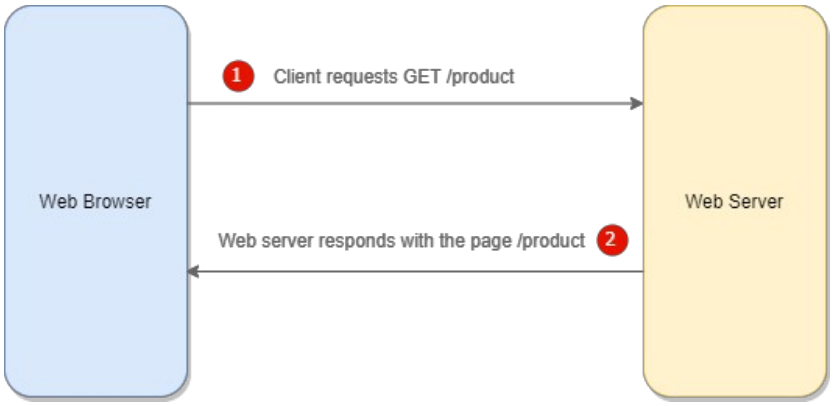
PHP Session

PHP session is used to store and pass information from one page to another temporarily (until user close the website).

PHP session technique is widely used in shopping websites where we need to store and pass cart information e.g. username, product code, product name, product price etc from one page to another.

PHP session creates unique user id for each browser to recognize the user and avoid conflict between multiple browsers.

The HTTP protocol is stateless. For example, when you visit the product page `product.php`, the web server responds with the page:



Suppose, you click the add to cart button on the product.php page and navigate to the cart.php page, the web server won't know that you have added the product to the cart.

To persist the information across the pages, the web server uses sessions. In this example, when you click the add to cart button, the web server will store the product on the server.

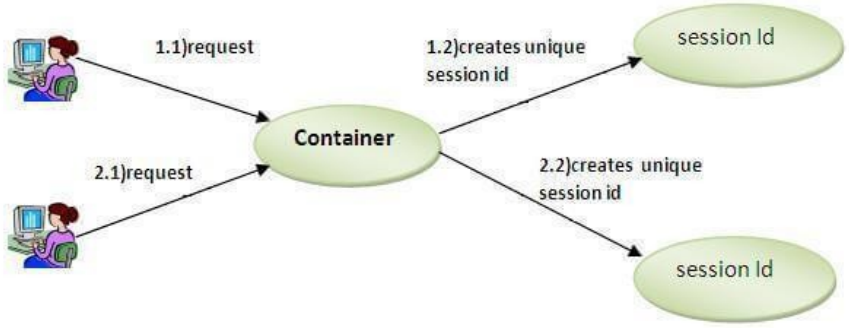
When you view the cart.php page, the web server gets the products from the session and displays them on the cart.php page:

How it works.

- First, the web browser requests for the product.php page.
- Second, the web server responds with the product.php page's content.

- Third, you click the Add To Cart button on the product.php page. The page will send an HTTP request (either POST or GET) to the web server. The web server validates the product and generates a session id. It also creates a new text file on the server to store the information related to the selected product.
- Fourth, the web server responds to the web browser with the PHPSESSID cookie in the response header. If the web browser allows cookies, it will save the PHPSESSID cookie, which stores the session id passed by the web server.
- Fifth, in the subsequent request, for example, when you view the cart.php page, the web browser passes the PHPSESSID back to the web server. When the web server sees the PHPSESSID cookie, it will resume the session with the session id stored in the cookie.
- Finally, the web server returns the cart page with the products that you selected.

Sessions allow you to store data on the web server associated with a session id. Once you create a session, PHP sends a cookie that contains the session id to the web browser. In the subsequent requests, the web browser sends the session id cookie back to the web server so that PHP can retrieve the data based on the session id.



PHP session_start() function

PHP session_start() function is used to start the session. It starts a new or resumes existing session. It returns existing session if session is created already. If session is not available, it creates and returns new session.

Syntax

```
bool session_start ( void )
```

Example

```
session_start();
```

PHP $_SESSION

PHP $_SESSION is an associative array that contains all session variables. It is used to set and get session variable values.

Example: Store information

```
$_SESSION["user"] = "Sachin";
```

Example: Get information

```
echo $_SESSION["user"];
```

PHP Session Example

File: session1.php

```
<?php
session_start();
?>
<html>
<body>
<?php
$_SESSION["user"] = "Sachin";
echo "Session information are set successfully.<br/>";
?>
<a href="session2.php">Visit next page</a>
</body>
```

```
</html>
```

File: session2.php

```
<?php
session_start();
?>
<html>
<body>
<?php
echo "User is: ".$_SESSION["user"];
?>
</body>
</html>
```

PHP Session Counter Example

File: sessioncounter.php

```
<?php
  session_start();


  if (!isset($_SESSION['counter'])) {
    $_SESSION['counter'] = 1;
  } else {
    $_SESSION['counter']++;
  }
  echo ("Page Views: ".$_SESSION['counter']);
```

```
?>
```

PHP Destroying Session

PHP session_destroy() function is used to destroy all session variables completely.

File: session3.php

```
<?php

session_start();

session_destroy();

?>
```