

Internet & Web Page Designing

REVISION 1.0
1 DECEMBER, 2016

ER. Gaurav Khandelwal
Professor
Department of Computer Science



KCRI COLLEGE

ALWAR 301001 RAJASTHAN

2017-2018

CONTENTS

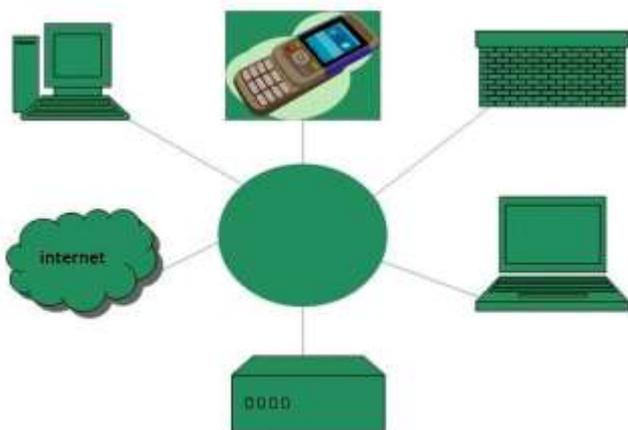
UNIT – 1	2-8
Overview of Internet :	
What is Internet, Internet Evolution, Advantage & Disadvantage of Internet, Extra Net, Reference Models, Layers, TCP/ IP Model, Domain name system Architecture, DNS Working.	
UNIT – 2	9-17
Internet Services & Protocols :	
Communication Services, Web Services, World Wide Web, Video Conferencing, Internet Protocols, TCP, IP, UDP, FTP, HTTP, Email, IMAP, POP, Email Working, E-mail Security.	
UNIT – 3	18-33
Web Designing Concepts & Hosting :	
Wireframe, Web Designing Tools, Web Development Process, Websites Hosting, Types of Hosting, Web sites Security, World Wide Web Evolution & Architecture, Web Browsers, Web Server, Proxy Server, Search Engines, Usenet, Digital Signature, Firewall Security.	
UNIT – 4	34-42
HTML – Overview :	
HTML Document Structure, Basic Tags- Heading Paragraph, Line Break, Center Etc, HTML Elements, HTML Attributes.	
UNIT – 5	42-56
HTML Formatting :	
Type Style, Meta Tags, Comments, Images, Tables, Text Links, HTML Fonts etc.	

UNIT-1

Overview of Internet

What is Internet:

- Internet is a world-wide global system of interconnected computer networks.
- Internet uses the standard Internet Protocol (TCP/IP).
- Every computer in internet is identified by a unique IP address.
- IP Address is a unique set of numbers (such as 110.22.33.114) which identifies a computer location.
- A special computer DNS (Domain Name Server) is used to give name to the IP Address so that user can locate a computer by a name.
- For example, a DNS server will resolve a name **http://www.kricollege.com** to a particular IP address to uniquely identify the computer on which this website is hosted.
- Internet is accessible to every user all over the world.



Internet Evolution

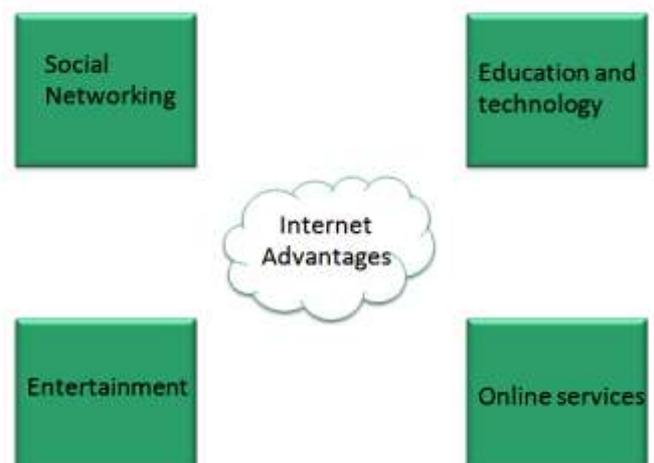
The concept of Internet was originated in 1969 and has undergone several

technological & Infrastructural changes as discussed below:

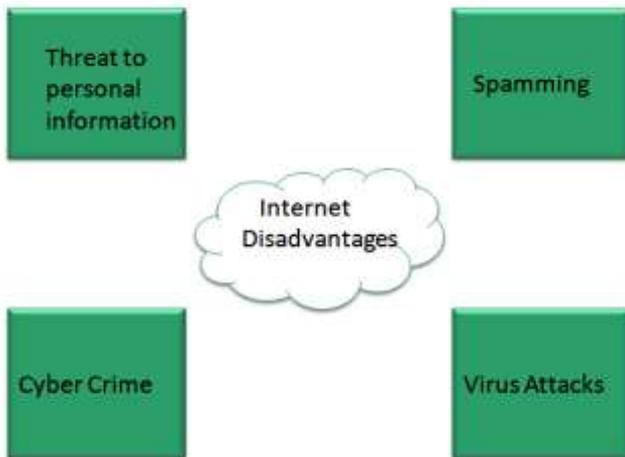
- The origin of Internet devised from the concept of **Advanced Research Project Agency Network (ARPANET)**.
- **ARPANET** was developed by United States Department of Defense.
- Basic purpose of ARPANET was to provide communication among the various bodies of government.
- Initially, there were only four nodes, formally called **Hosts**.
- In 1972, the **ARPANET** spread over the globe with 23 nodes located at different countries and thus became known as **Internet**.
- By the time, with invention of new technologies such as TCP/IP protocols, DNS, WWW, browsers, scripting languages etc., Internet provided a medium to publish and access information over the web.

Internet Advantages

Internet covers almost every aspect of life, one can think of. Here, we will discuss some of the advantages of Internet:

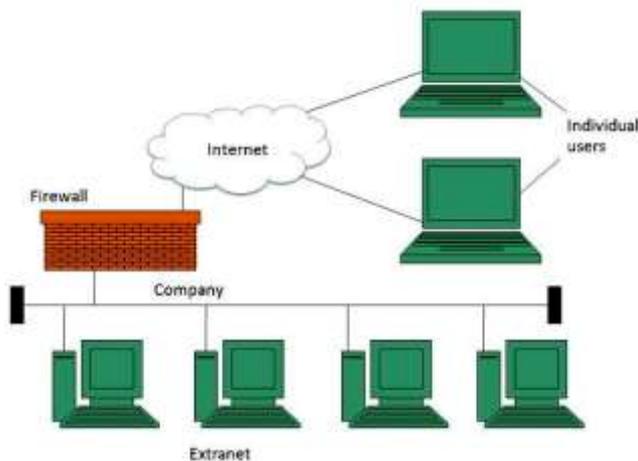


Internet Disadvantages



Extranet

Extranet refers to network within an organization, using internet to connect to the outsiders in controlled manner. It helps to connect businesses with their customers and suppliers and therefore allows working in a collaborative manner.



Extranet Benefits

Extranet proves to be a successful model for all kind of businesses whether small or big. Here are some of the advantages of extranet for employees, suppliers, business partners, and customers:



Extranet Issues

Apart for advantages there are also some issues associated with extranet. These issues are discussed below:

Hosting

Where the extranet pages will be held i.e. who will host the extranet pages. In this context there are two choices:

- Host it on your own server.
- Host it with an Internet Service Provider (ISP) in the same way as web pages.

But hosting extranet pages on your own server requires high bandwidth internet connection which is very costly.

Security

Additional firewall security is required if you host extranet pages on your own server which result in a complex security mechanism and increase work load.

Accessing Issues

Information can not be accessed without internet connection. However, information can be accessed in Intranet without internet connection.

Decreased Interaction

It decreases the face to face interaction in the business which results in lack of communication

among customers, business partners and suppliers.

Extranet vs. Intranet

The following table shows differences between Extranet and Intranet:

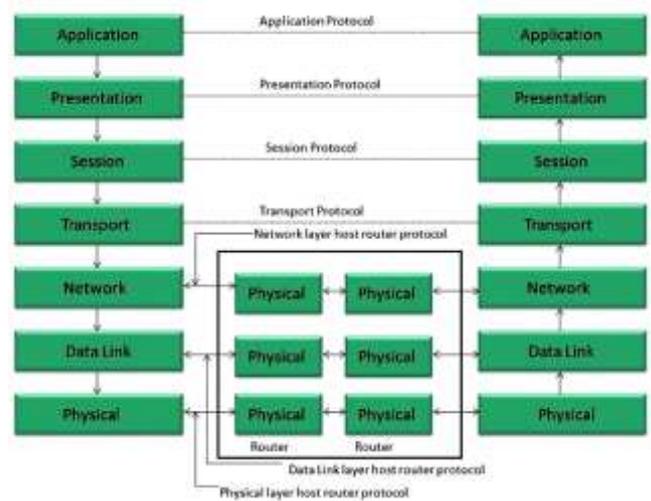
Extranet	Intranet
Internal network that can be accessed externally.	Internal network that can not be accessed externally.
Extranet is extension of company's Intranet.	Only limited users of a company.
For limited external communication between customers, suppliers and business partners.	Only for communication within a company.

Internet Reference Models

OSI Reference Model

OSI is acronym of **Open System Interface**. This model is developed by the **International organization of Standardization (ISO)** and therefore also referred as **ISO-OSI** Model.

The OSI model consists of seven layers as shown in the following diagram. Each layer has a specific function, however each layer provide services to the layer above.



PHYSICAL LAYER

The Physical layer is responsible for the following activities:

- Activating, maintaining and deactivating the physical connection.
- Defining voltages and data rates needed for transmission.
- Converting digital bits into electrical signal.
- Deciding whether the connection is simplex, half duplex or full duplex.

DATA LINK LAYER

The data link layer performs the following functions:

- Performs synchronization and error control for the information which is to be transmitted over the physical link.
- Enables error detection, and adds error detection bits to the data which are to be transmitted.

NETWORK LAYER

Following are the functions of Network Layer:

- To route the signals through various channels to the other end.
- To act as the network controller by deciding which route data should take.

- To divide the outgoing messages into packets and to assemble incoming packets into messages for higher levels.

TRANSPORT LAYER

The Transport layer performs the following functions:

- It decides if the data transmission should take place on parallel paths or single path.
- It performs multiplexing, splitting on the data.
- It breaks the data groups into smaller units so that they are handled more efficiently by the network layer.

The Transport Layer guarantees transmission of data from one end to other end.

SESSION LAYER

The Session layer performs the following functions:

- Manages the messages and synchronizes conversations between two different applications.
- It controls logging on and off, user identification, billing and session management.

PRESENTATION LAYER

The Presentation layer performs the following functions:

```
<="" ul="" style="box-sizing: border-box; color: rgb(49, 49, 49); font-family: Verdana, Geneva, Tahoma, Arial, Helvetica, sans-serif; font-size: 14px; font-style: normal; font-variant-ligatures: normal; font-
```

```
variant-caps: normal; font-weight: 400; letter-spacing: normal; orphans: 2; text-align: start; text-indent: 0px; text-transform: none; white-space: normal; widows: 2; word-spacing: 0px; -webkit-text-stroke-width: 0px; text-decoration-style: initial; text-decoration-color: initial;">
```

- This layer makes it sure that the information is delivered in such a form that the receiving system will understand and use it.

APPLICATION LAYER

The Application layer performs the following functions:

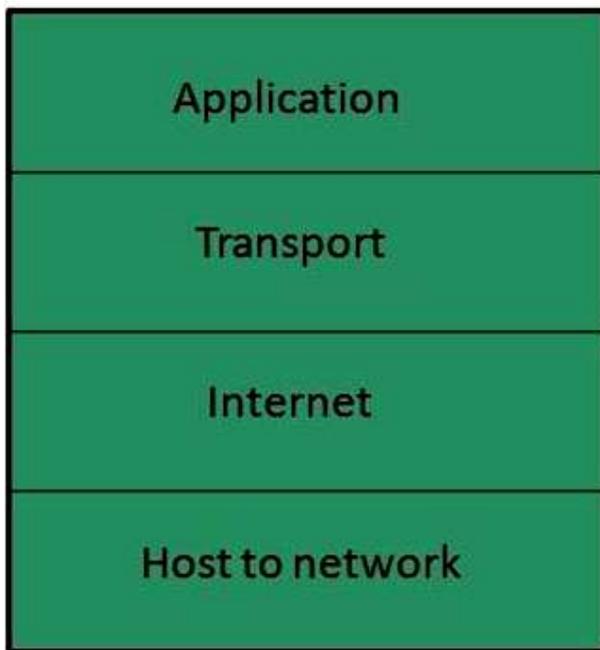
- It provides different services such as manipulation of information in several ways, retransferring the files of information, distributing the results etc.
- The functions such as LOGIN or password checking are also performed by the application layer.

TCP/IP Reference Model

TCP/IP model is practical model and is used in the Internet. TCP/IP is acronym of Transmission Control Protocol and Internet Protocol.

The **TCP/IP** model combines the two layers (Physical and Data link layer) into one layer i.e. **Host-to-Network** layer. The following diagram shows the various layers of TCP/IP model:

TCP/IP Model



APPLICATION LAYER

This layer is same as that of the OSI model and performs the following functions:

- It provides different services such as manipulation of information in several ways, retransferring the files of information, distributing the results etc.
- The functions such as LOGIN or password checking are also performed by the application layer.

Protocols used: TELNET, FTP, SMTP, DN, HTTP, NNTP are the protocols employed in this layer.

TRANSPORT LAYER

It does the same functions as that of transport layer in OSI model. Here are the key points regarding transport layer:

- It uses **TCP** and **UDP** protocol for end to end transmission.
- TCP is reliable and **connection oriented protocol**.

- TCP also handles flow control.
- The UDP is not reliable and a **connection less protocol** also does not perform flow control.

Protocols used:

TCP/IP and **UDP** protocols are employed in this layer.

INTERNET LAYER

The function of this layer is to allow the host to insert packets into network and then make them travel independently to the destination. However, the order of receiving the packet can be different from the sequence they were sent.

Protocols used: Internet Protocol (IP) is employed in Internet layer.

HOST-TO-NETWORK LAYER

This is the lowest layer in TCP/IP model. The host has to connect to network using some protocol, so that it can send IP packets over it. This protocol varies from host to host and network to network.

Protocols used: ARPANET, SATNET, LAN, packet radio are the protocols which are used in this layer.

Domain Name System Architecture

The Domain name system comprises of **Domain Names, Domain Name Space, Name Server** that have been described below:

Domain Names

Domain Name is a symbolic string associated with an IP address. There are several domain names available; some of them are generic such as **com, edu, gov, net** etc, while

some country level domain names such as **au, in, za, us** etc. The following table shows the **Generic** Top-Level Domain names:

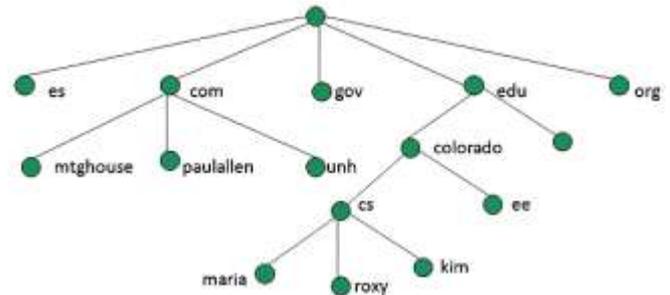
Domain Name	Meaning
Com	Commercial business
Edu	Education
Gov	U.S. government agency
Int	International entity
Mil	U.S. military
Net	Networking organization
Org	Non profit organization

The following table shows the **Country** **top-level** domain names:

Domain Name	Meaning
au	Australia
in	India
cl	Chile
fr	France
us	United States
za	South Africa
uk	United Kingdom
jp	Japan
es	Spain
de	Germany
ca	Canada
ee	Estonia
hk	Hong Kong

Domain Name Space

The domain name space refers a hierarchy in the internet naming structure. This hierarchy has multiple levels (from 0 to 127), with a root at the top. The following diagram shows the domain name space hierarchy:



In the above diagram each subtree represents a domain. Each domain can be partitioned into sub domains and these can be further partitioned and so on.

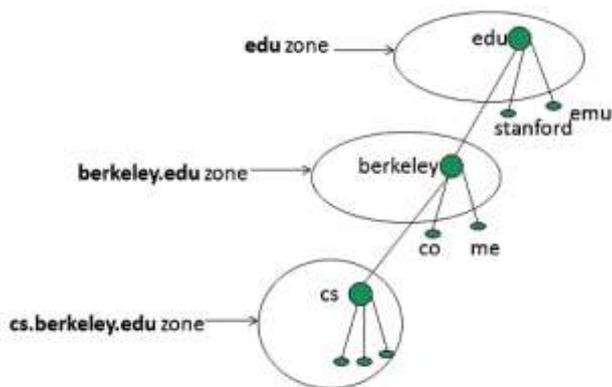
Name Server

Name server contains the DNS database. This database comprises of various names and their corresponding IP addresses. Since it is not possible for a single server to maintain entire DNS database, therefore, the information is distributed among many DNS servers.

- Hierarchy of server is same as hierarchy of names.
- The entire name space is divided into the zones

Zones

Zone is collection of nodes (sub domains) under the main domain. The server maintains a database called zone file for every zone.



If the domain is not further divided into sub domains then domain and zone refers to the same thing.

The information about the nodes in the sub domain is stored in the servers at the lower levels however; the original server keeps reference to these lower levels of servers.

TYPES OF NAME SERVERS

Following are the three categories of Name Servers that manages the entire Domain Name System:

1. Root Server
2. Primary Server
3. Secondary Server

ROOT SERVER

Root Server is the top level server which consists of the entire DNS tree. It does not contain the information about domains but delegates the authority to the other server.

PRIMARY SERVERS

Primary Server stores a file about its zone. It has authority to create, maintain, and update the zone file.

SECONDARY SERVER

Secondary Server transfers complete information about a zone from another server which may be primary or secondary server. The secondary server does not have authority to create or update a zone file.

DNS Working

DNS translates the domain name into IP address automatically. Following steps will take you through the steps included in domain resolution process:

1. When we type **www.tutorialspoint.com** into the browser, it asks the local DNS Server for its IP address.
Here the local DNS is at ISP end.
2. When the local DNS does not find the IP address of requested domain name, it forwards the request to the root DNS server and again enquires about IP address of it.
3. The root DNS server replies with delegation that **I do not know the IP address of www.tutorialspoint.com but know the IP address of DNS Server.**
4. The local DNS server then asks the com DNS Server the same question.
5. The **com** DNS Server replies the same that it does not know the IP address of **www.tutorialspoint.com** but knows the address of **tutorialspoint.com**.
6. Then the local DNS asks the **tutorialspoint.com** DNS server the same question.
7. Then **tutorialspoint.com** DNS server replies with IP address of **www.tutorialspoint.com**.
8. Now, the local DNS sends the IP address of **www.tutorialspoint.com** to the computer that sends the request.

UNIT-2

Internet Services & Protocols

Communication Services

There are various Communication Services available that offer exchange of information with individuals or groups. The following table gives a brief introduction to these services:

S. N.	Service Description
1	Electronic Mail Used to send electronic message over the internet.
2	Telnet Used to log on to a remote computer that is attached to internet.
3	Newsgroup Offers a forum for people to discuss topics of common interests.
4	Internet Relay Chat (IRC) Allows the people from all over the world to communicate in real time.
5	Mailing Lists Used to organize group of internet users to share common information through e-mail.
6	Internet Telephony (VoIP) Allows the internet users to talk across internet to any PC equipped to receive the call.
7	Instant Messaging Offers real time chat between individuals and group of people. Eg. Yahoo messenger, MSN messenger.

Information Retrieval Services

There exist several Information retrieval services offering easy access to information present on the internet. The following table gives a brief introduction to these services:

S. N.	Service Description
1	File Transfer Protocol (FTP) Enable the users to transfer files.
2	Archie It's updated database of public FTP sites and their content. It helps to search a file by its name.
3	Gopher Used to search, retrieve, and display documents on remote sites.
4	Very Easy Rodent Oriented Netwide Index to Computer Achieved (VERONICA) VERONICA is gopher based resource. It allows access to the information resource stored on gopher's servers.

Web Services

Web services allow exchange of information between applications on the web. Using web services, applications can easily interact with each other.

The web services are offered using concept of **Utility Computing**.

World Wide Web (WWW)

WWW is also known as W3. It offers a way to access documents spread over the several servers over the internet. These documents may contain texts, graphics, audio, video, hyperlinks. The hyperlinks allow the users to navigate between the documents.

Video Conferencing

Video conferencing or Video teleconferencing is a method of communicating by two-way video and audio transmission with help of telecommunication technologies.

MODES OF VIDEO CONFERENCING

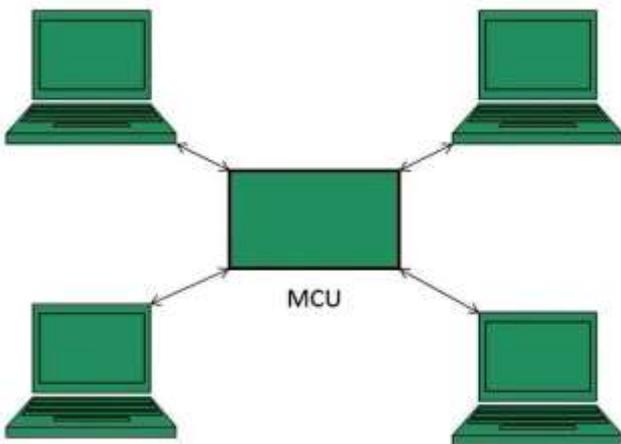
POINT-TO-POINT

This mode of conferencing connects two locations only.



MULTI-POINT

This mode of conferencing connects more than two locations through **Multi-point Control Unit (MCU)**.



Internet Protocols

Transmission Control Protocol (TCP)

- Transmission Control Protocol (TCP) corresponds to the Transport Layer of OSI Model.
- TCP is a reliable and connection oriented protocol.
- TCP offers:
 - Stream Data Transfer.
 - Reliability.

- Efficient Flow Control
- Full-duplex operation.
- Multiplexing.

- TCP offers connection oriented end-to-end packet delivery.
- TCP ensures reliability by sequencing bytes with a forwarding acknowledgement number that indicates to the destination the next byte the source expect to receive.
- It retransmits the bytes not acknowledged with in specified time period.

Internet Protocol (IP)

Internet Protocol is **connectionless** and **unreliable** protocol. It ensures no guarantee of successfully transmission of data. In order to make it reliable, it must be paired with reliable protocol such as TCP at the transport layer. Internet protocol transmits the data in form of a datagram as shown in the following diagram:

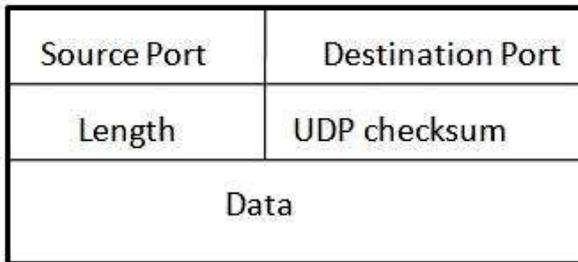
4	8	16	32 bits
VER	HLEN	D.S. type of service	Total length of 16 bits
Identification of 16 bits		Flags 3 bits	Fragmentation Offset (13 bits)
Time to live	Protocol	Header checksum (16 bits)	
Source IP address			
Destination IP address			
Option + Padding			

User Datagram Protocol (UDP)

Like IP, UDP is connectionless and unreliable protocol. It doesn't require making a connection with the host to exchange data. Since UDP is unreliable protocol, there is no

mechanism for ensuring that data sent is received.

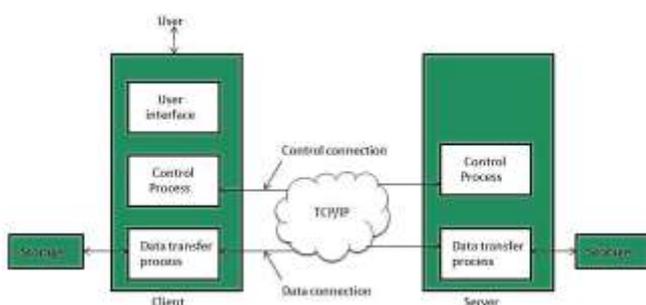
UDP transmits the data in form of a datagram. The UDP datagram consists of five parts as shown in the following diagram:



File Transfer Protocol (FTP)

FTP is used to copy files from one host to another. FTP offers the mechanism for the same in following manner:

- FTP creates two processes such as Control Process and Data Transfer Process at both ends i.e. at client as well as at server.
- FTP establishes two different connections: one is for data transfer and other is for control information.
- **Control connection** is made between **control processes** while **Data Connection** is made between **Data transfer process**
- FTP uses **port 21** for the control connection and **Port 20** for the data connection.



Trivial File Transfer Protocol (TFTP)

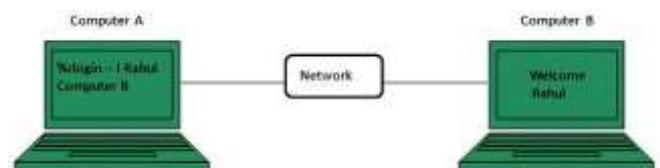
Trivial File Transfer Protocol is also used to transfer the files but it transfers the files without authentication. Unlike FTP, TFTP does not separate control and data information. Since there is no authentication exists, TFTP lacks in security features therefore it is not recommended to use TFTP.

Key points

- TFTP makes use of UDP for data transport. Each TFTP message is carried in separate UDP datagram.
- The first two bytes of a TFTP message specify the type of message.
- The TFTP session is initiated when a TFTP client sends a request to upload or download a file.
- The request is sent from an ephemeral UDP port to the **UDP port 69** of an TFTP server.

Telnet

Telnet is a protocol used to log in to remote computer on the internet. There are a number of Telnet clients having user friendly user interface. The following diagram shows a person is logged in to computer A, and from there, he remote logged into computer B.



Hyper Text Transfer Protocol (HTTP)

HTTP is a communication protocol. It defines mechanism for communication between browser and the web server. It is also called request and response protocol because the communication between browser and server takes place in request and response pairs.

HTTP REQUEST

HTTP request comprises of lines which contains:

- Request line
- Header Fields
- Message body

Key Points

- The first line i.e. the **Request line** specifies the request method i.e. **Get** or **Post**.
- The second line specifies the header which indicates the domain name of the server from where index.htm is retrieved.

HTTP RESPONSE

Like HTTP request, HTTP response also has certain structure. HTTP response contains:

- Status line
- Headers
- Message body

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 Protocols

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.

Key Points:

- 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, SMTP 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.

IMAP

IMAP stands for **Internet Mail 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

Key Points:

- 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.

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.

Key Points

- 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.

E-mail Working

Email working follows the client server approach. In this client is the mailer i.e. the mail application or mail

program and server is a device that manages emails.

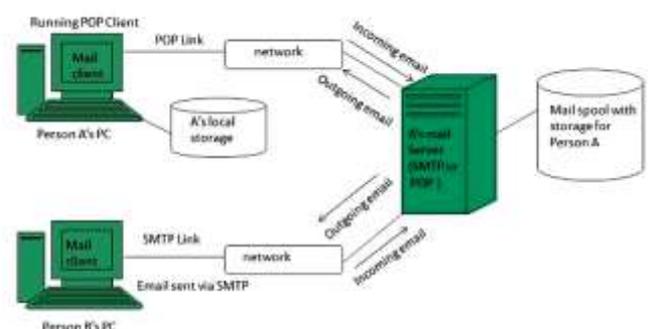
Following example will take you through the basic steps involved in sending and receiving emails and will give you a better understanding of working of email system:

- Suppose person A wants to send an email message to person B.
- Person A composes the messages using a mailer program i.e. mail client and then select Send option.
- The message is routed to **Simple Mail Transfer Protocol** to person B's mail server.
- The mail server stores the email message on disk in an area designated for person B.

The disk space area on mail server is called mail spool.

- Now, suppose person B is running a POP client and knows how to communicate with B's mail server.
- It will periodically poll the POP server to check if any new email has arrived for B. As in this case, person B has sent an email for person A, so email is forwarded over the network to B's PC. This is message is now stored on person B's PC.

The following diagram gives pictorial representation of the steps discussed above:



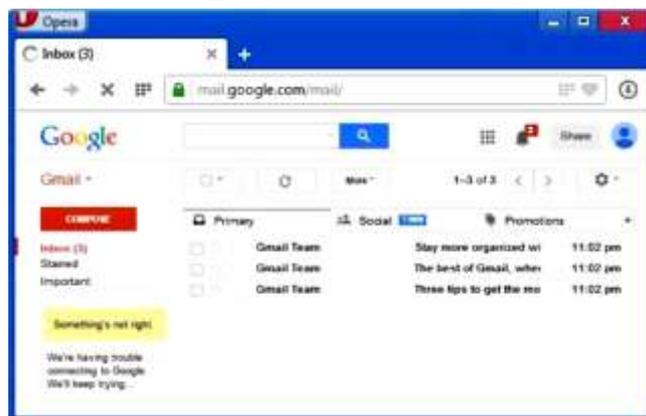
E-mail Operations

CREATING EMAIL ACCOUNT

There are various email service provider available such as **Gmail, hotmail, ymail, rediff mail** etc. Here we will learn how to create an account using Gmail.

- Open gmail.com and click **create an account**.
- Now a form will appear. Fill your details here and click **Next Step**.

- This step allows you to add your picture. If you don't want to upload now, you can do it later. Click **Next Step**.
- Now a welcome window appears. Click **Continue to Gmail**.
- Wow!! You are done with creating your email account with Gmail. It's that easy. Isn't it?
- Now you will see your Gmail account as shown in the following image:



Key Points:

- Gmail manages the mail into three categories namely **Primary, Social** and **Promotions**.
- **Compose** option is given at the right to compose an email message.
- **Inbox, Starred, Sent mail, Drafts** options are available on the left pane which allows you to keep track of your emails.

COMPOSING AND SENDING EMAIL

Before sending an email, we need to compose a message. When we are composing an email message, we specify the following things:

- Sender's address in To field
- Cc (if required)
- Bcc (if required)
- Subject of email message
- Text
- Signature

You should specify the correct email address; otherwise it will send an error back to the sender.

Once you have specified all the above parameters, It's time to send the email. The mailer program provides a Send button to send email, when you click Send, it is sent to the mail server and a message **mail sent successfully** is shown at the above.

READING EMAIL

Every email program offers you an interface to access email messages. Like in Gmail, emails are stored under different tabs such as primary, social, and promotion. When you click one of tab, it displays a list of emails under that tab.

In order to read an email, you just have to click on that email. Once you click a particular email, it gets opened. The opened email may have some file attached with it. The attachments are shown at the bottom of the opened email with an option called **download attachment**.

REPLYING EMAIL

After reading an email, you may have to reply that email. To reply an email, click **Reply** option shown at the bottom of the opened email.

Once you click on Reply, it will automatically copy the sender's address in to the To field. Below the To field, there is a text box where you can type the message.

Once you are done with entering message, click Send button. It's that easy. Your email is sent.

FORWARDING EMAIL

It is also possible to send a copy of the message that you have received along with your own comments if you want. This can be done using **forward** button available in mail client software.

The difference between replying and forwarding an email is that when you reply a message to a person who has send the mail but while forwarding you can send it to anyone.

When you receive a forwarded message, the message is marked with a > character in front of each line

and **Subject:** field is prefixed with **Fw.**

DELETING EMAIL

If you don't want to keep email into your inbox, you can delete it by simply selecting the message from the message list and clicking **delete** or pressing the appropriate command.

Some mail clients offers the deleted mails to be stored in a folder called deleted items or trash from where you can recover a deleted email.

E-mail Security

E-MAIL HACKING

Email hacking can be done in any of the following ways:

- Spam
- Virus
- Phishing

SPAM

E-mail spamming is an act of sending **Unsolicited Bulk E-mails (UBI)** which one has not asked for. Email spams are the junk mails sent by commercial companies as an advertisement of their products and services.

VIRUS

Some emails may incorporate with files containing malicious script which when run on your computer may lead to destroy your important data.

PHISHING

Email phishing is an activity of sending emails to a user claiming to be a legitimate enterprise. Its main purpose is to steal sensitive information such as usernames, passwords, and credit card details.

Such emails contains link to websites that are infected with malware and direct the user to enter details at a fake website whose look and feels are same to legitimate one.

E-MAIL SPAMMING AND JUNK MAILS

Email spamming is an act of sending Unsolicited Bulk E-mails (UBI) which one has not asked for. Email spams are the junk mails sent by commercial companies as an advertisement of their products and services. Spams may cause the following problems:

- It floods your e-mail account with unwanted e-mails, which may result in loss of important e-mails if inbox is full.
- Time and energy is wasted in reviewing and deleting junk emails or spams.
- It consumes the bandwidth that slows the speed with which mails are delivered.
- Some unsolicited email may contain virus that can cause harm to your computer.

BLOCKING SPAMS

Following ways will help you to reduce spams:

- While posting letters to newsgroups or mailing list, use a separate e-mail address than the one you used for your personal e-mails.
- Don't give your email address on the websites as it can easily be spammed.

- Avoid replying to emails which you have received from unknown persons.
- Never buy anything in response to a spam that advertises a product.

E-MAIL CLEANUP AND ARCHIVING

In order to have light weighted Inbox, it's good to archive your inbox from time to time. Here I will discuss the steps to clean up and archive your Outlook inbox.

- Select **File** tab on the mail pane.
- Select **Cleanup Tools** button on account information screen.
- Select **Archive** from cleanup tools drop down menu.
- Select **Archive this folder and all subfolders** option and then click on the folder that you want to archive. Select the date from the **Archive items older than:** list. Click **Browse** to create new **.pst** file name and location. Click **OK**.

E-mail Providers

There are several email service providers available in the market with their enabled features such as sending, receiving, drafting, storing an email and much more.

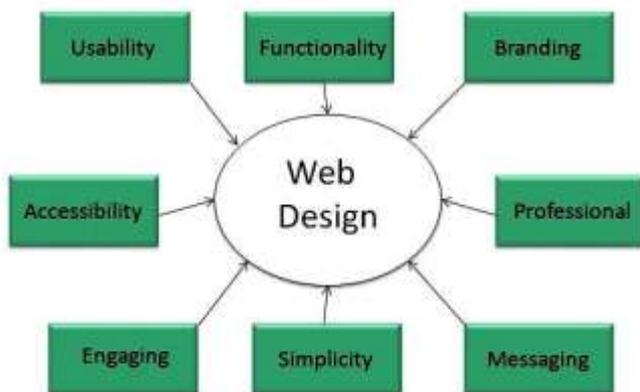
The following table shows the popular email service providers:

S. N.	Service and Description
1.	<p>Gmail Gmail is an email service that allows users to collect all the messages. It also offers approx 7 GB of free storage.</p>
2.	<p>Hotmail Hotmail offers free email and practically unlimited storage accessible on web.</p>
3.	<p>Yahoo Mail Yahoo Mail offers unlimited storage, SMS texting, social networking and instant messaging to boot.</p>
4.	<p>iCloud Mail iCloud Mail offers ample storage, IMAP access, and an elegantly functional web application.</p>
5.	<p>ATM Mail ATM Mail is a free email service with good spam protection.</p>
6.	<p>Mail.com and GMX Mail Mail.com and GMX Mail offers reliable mail service with unlimited online storage.</p>
7.	<p>Shortmail Shortmail offers easy and fast email service but with limited 500 characters per message.</p>
8.	<p>Inbox.com Inbox.com offers 5 GB of free online storage. IMAP is not supported by Inbox.com</p>
9.	<p>Facebook Messages Facebook Messages includes the message conversation.</p>
10.	<p>My Way Mail My Way Mail offers clean and fast free email service but lacks in secure messaging.</p>

UNIT-3

Website Designing Concepts & Hosting

Web designing has direct link to visual aspect of a web site. Effective web design is necessary to communicate ideas effectively.



Web designing is subset of web development. However these terms are used interchangeably.

Key Points

Design Plan should include the following:

- Details about information architecture.
- Planned structure of site.
- A site map of pages

Wireframe

Wireframe refers to a visual guide to appearance of web pages. It helps to define structure of web site, linking between web pages and layout of visual elements.

Following things are included in a wireframe:

- Boxes of primary graphical elements
- Placement of headlines and sub headings
- Simple layout structure

- Calls to action
- Text blocks

Wireframe can be created using program like Visio but you can also use a pen and paper.

Web Designing Tools

Here is the list of tools that can be used to make effective web designs:

- Photoshop CC
- Illustrator CC
- Coda 2
- Omni Graffle
- Sublime Text
- GitHub
- Pen and Parer
- Vim
- Image optim
- Sketch 3
- Heroku
- Axure
- Hype 2
- Slicy
- Framer.js
- Image Alpha
- Emmet Live Style
- Hammer
- Icon Slate
- JPEG mini Lite
- Bug Herd

Web Page Anatomy

A web site includes the following components:

Containing Block

Container can be in the form of page's body tag, an all containing div tag. Without container there would be no place to put the contents of a web page.

Logo

Logo refers to the identity of a website and is used across a company's various forms of marketing such as business cards, letterhead, brochures and so on.

Naviagation

The site's **navigation system** should be easy to find and use. Oftenly the anvigation is placed righth at the top of the page.

Content

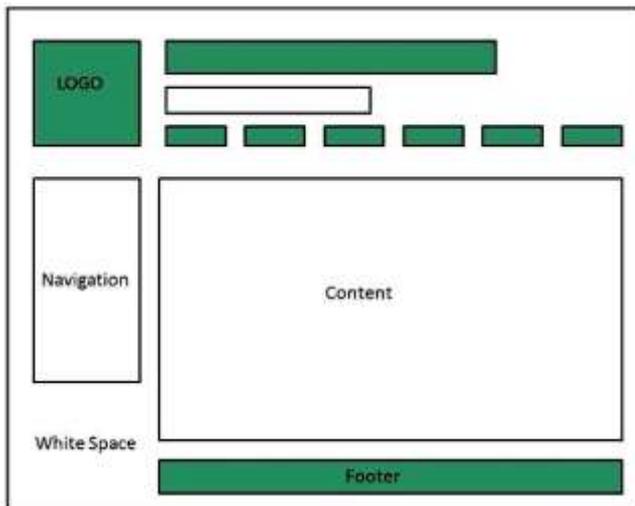
The content on a web site should be relevant to the purpose of the web site.

Footer

Footer is located at the bottom of the page. It usually contains copyright, contract and legal information as well as few links to the main sections of the site.

Whitespace

It is also called as **negative space** and refers to any area of page that is not covered by type or illustrations.



Web design Mistakes

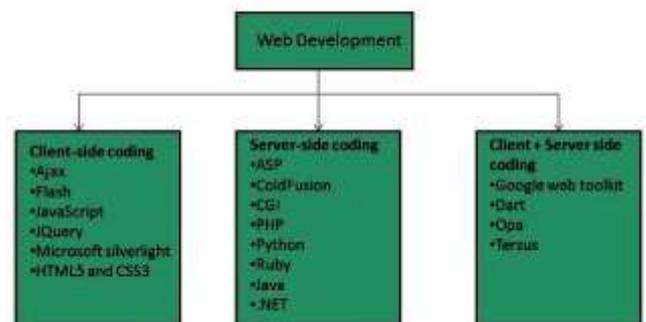
One should be aware of the following common mistakes should always keep in mind:

- Website not working in any other browser other internet explorer.
- Using cutting edge technology for no good reason

- Sound or video that starts automatically
- Hidden or disguised navigation
- 100% flash content.

Web development

Web development refers to building website and deploying on the web. Web development requires use of scripting languages both at the server end as well as at client end.

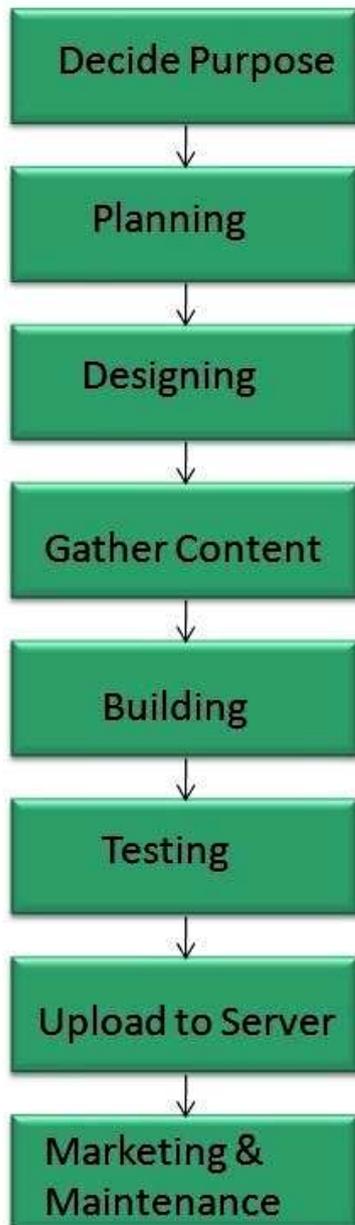


Before developing a web site once should keep several aspects in mind like:

- What to put on the web site?
- Who will host it?
- How to make it interactive?
- How to code it?
- How to create search engine friendly web site?
- How to secure the source code frequently?
- Will the web site design display well in different browsers?
- Will the navigation menus be easy to use?
- Will the web site loads quickly?
- How easily will the site pages print?
- How easily will visitors find important details specific to the web site?
- How effectively the style sheets be used on your web sites?

Web Development Process

Web development process includes all the steps that are good to take to build an attractive, effective and responsive website. These steps are shown in the following diagram:



Web development tools

Web development tools helps the developer to test and debug the web sites. Now a days the web development tool come with the web browsers as add-ons. All web browsers have built in tools for this purpose.

These tools allow the web developer to use HTML, CSS and JavaScript etc.. These are accessed by hovering over an item on a web page and selecting the "Inspect Element" from the context menu.

FEATUES

Following are the common featurers that every web development tool exhibits:

HTML AND THE DOM

HTML and DOM viewer allows you to see the DOM as it was rendered. It also allows to make changes to HTML and DOM and see the changes reflected in the page after the change is made.

WEB PAGE ASSESTS, RESOURCES, AND NETWORK INFORMATION

Web development tools also helps to inspect the resources that are loaded and available on the web page.

PROFIING AND AUDITING

Profiling refers to get information about the performance of a web page or web application and **Auditing** provides developers suggestions, after analyzing a page, for optimizations to decrease page load time and increase responsiveness.

Skills Required

For being a successful web developer, one should possess the following skills:

- Understanding of client and server side scripting.
- Creating, editing and modifying templates for a CMS or web development framework.

- Testing cross browser inconsistencies.
- Conducting observational user testing.
- Testing for compliance to specified standards such as accessibility standards in the client region.
- Programming interaction with javaScript, PHP, and JQuery etc.

Websites Hosting

Web hosting is a service of providing online space for storage of web pages. These web pages are made available via **World Wide Web**. The companies which offer website hosting are known as **Web hosts**.

The servers on which web site is hosted remain switched on 24 x7. These servers are run by web hosting companies. Each server has its own IP address. Since IP addresses are difficult to remember therefore, webmaster points their domain name to the IP address of the server their website is stored on.

It is not possible to host your website on your local computer, to do so you would have to leave your computer on 24 hours a day. This is not practical and cheaper as well. This is where web hosting companies comes in.

Types of Hosting

The following table describes different types of hosting that can be availed as per the need:

S.N.	Hosting Description
1.	Shared Hosting In shared hosting, the hosting

	company puts thousand of website on the same physical server. Each customer has their own allocation of physical web space and a set of bandwidth limit. As all websites share same physical memory, MYSQL server and Apache server, one website on the server experiencing high traffic load will affect performance of all websites on the server.
2.	Virtual Private Server (VPS) It is also known as Virtual Dedicated Server. It is a server which is partitioned into smaller servers. In this customer is given their own partition, which is installed with its own operating system. Unlike shared hosting, VPS doesn't share memory or processor time rather it allocates certain amount of memory and CPU to use which means that any problem on a VPS partition on the same drive will not affect other VPS customers.
3.	Dedicated Server In this kind of hosting, single dedicated server is setup for just one customer. It is commonly used by the businesses that need the power, control and security that a dedicated server offers.
4.	Reseller Hosting A reseller acts as a middle man and sells hosting space of someone else's server.
5.	Grid Hosting Instead of utilizing one server, Grid Hosting spreads resources over a large number of servers. It is quite stable and flexible. The servers can be added or taken away from the grid without crashing the system.

Web Hosting Companies

Following are the several companies offering web hosting service:

S.N.	Hosting Company
1.	Blue Host
2.	Go Daddy
3.	Host Gator
4.	just Host
5.	Laughing Squid
6.	Hivelocity
7.	liquid Web
8.	Media TempleServInt
9.	Wired Tree
10.	Wild West Domains
11.	Wix
12.	WIPL
13.	Big Rock

Website Security Considerations

Websites are always to prone to security risks. **Cyber crime** impacts your business by hacking your website. Your website is then used for hacking assaults that install malicious software or malware on your visitor's computer.

Hackers may also steal important customer data such as credit card information, destroy your business and propagate illegal content to your users.

Security Considerations

UPDATED SOFTWARE

It is mandatory to keep you software updated. It plays vital role in keeping your website secure.

SQL INJECTION

It is an attempt by the hackers to manipulate your database. It is easy to insert rogue code into your query that can be used to manipulate your database such as change tables, get information or delete data.

CROSS SITE SCRIPTING (XSS)

It allows the attackers to inject client side script into web pages. Therefore, while creating a form It is good to endure that you check the data being submitted and encode or strip out any HTML.

ERROR MESSAGES

You need to be careful about how much information to be given in the error messages. For example, if the user fails to log in the error message should not let the user know which field is incorrect: username or password.

VALIDATION OF DATA

The validation should be performed on both server side and client side.

PASSWORDS

It is good to enforce password requirements such as of minimum of eight characters, including upper case, lower case and special character. It will help to protect user's information in long run.

UPLOAD FILES

The file uploaded by the user may contain a script that when executed on the server opens up your website.

SSL

It is good practice to use SSL protocol while passing personal information between website and web server or database.

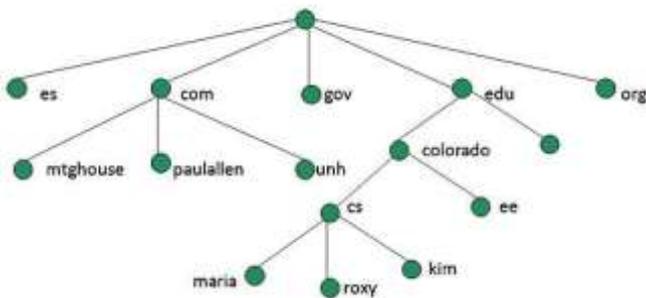
World Wide Web

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).

A broader definition comes from the organization that Web inventor **Tim Berners-Lee** helped found, the **World Wide Web Consortium (W3C)**.

The World Wide Web is the universe of network-accessible information, an embodiment of human knowledge. In simple terms, 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.

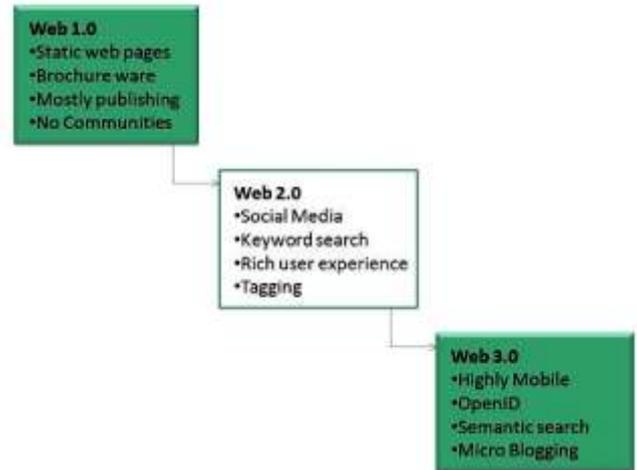
Internet and **Web** is not the same thing: Web uses internet to pass over the information.



WWW Evolution

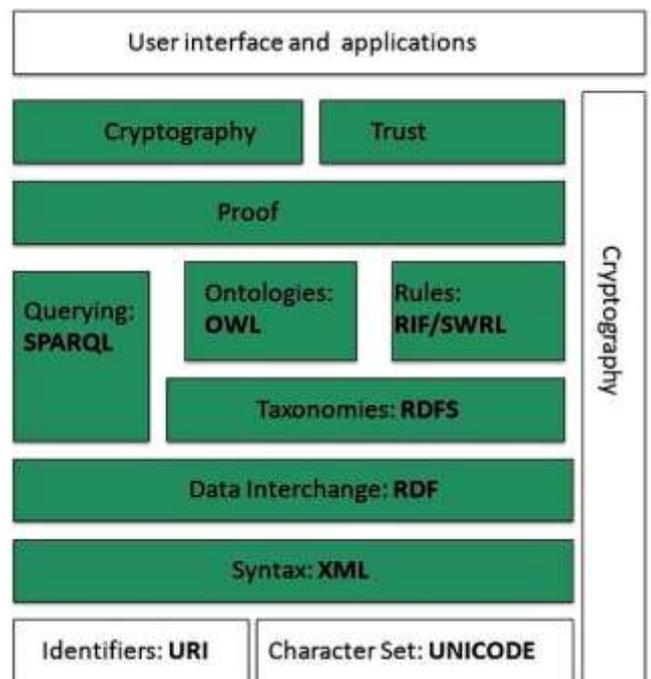
World Wide Web was created by **Timothy Berners Lee** in 1989 at **CERN** in **Geneva**. World Wide Web came into existence as a proposal by him, to allow researchers to work together effectively and efficiently at **CERN**. Eventually it became **World Wide Web**.

The following diagram briefly defines evolution of World Wide Web:



WWW Architecture

WWW architecture is divided into several layers as shown in the following diagram:



IDENTIFIERS AND CHARACTER SET
Uniform Resource Identifier (URI) is used to uniquely identify resources on the web and **UNICODE** makes it possible to built web pages that can be read and write in human languages.

SYNTAX**XML(Extensible Markup**

Language) helps to define common syntax in semantic web.

DATA INTERCHANGE

Resource Description Framework (RDF) framework helps in defining core representation of data for web. RDF represents data about resource in graph form.

TAXONOMIES

RDF Schema (RDFS) allows more standardized description of **taxonomies** and other **ontological** constructs.

ONTOLOGIES

Web Ontology Language (OWL) offers more constructs over RDFS. It comes in following three versions:

- OWL Lite for taxonomies and simple constraints.
- OWL DL for full description logic support.
- OWL for more syntactic freedom of RDF

RULES

RIF and **SWRL** offers rules beyond the constructs that are available from **RDFs** and **OWL**. **Simple Protocol and RDF Query Language (SPARQL)** is SQL like language used for querying RDF data and OWL Ontologies.

PROOF

All semantic and rules that are executed at layers below Proof and their result will be used to prove deductions.

CRYPTOGRAPHY

Cryptography means such as digital signature for verification of the origin of sources is used.

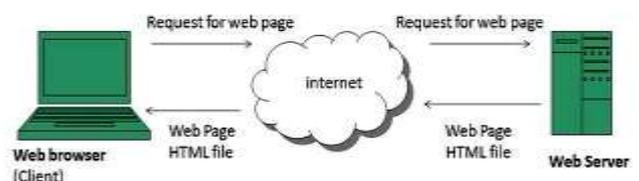
USER INTERFACE AND APPLICATIONS

On the top of layer **User interface and Applications** layer is built for user interaction.

WWW Operation

WWW works on client- server approach. Following steps explains how the web works:

1. User enters the URL (say, **http://www.tutorialspoint.com**) of the web page in the address bar of web browser.
2. Then browser requests the Domain Name Server for the IP address corresponding to **www.tutorialspoint.com**.
3. After receiving IP address, browser sends the request for web page to the web server using HTTP protocol which specifies the way the browser and web server communicates.
4. Then web server receives request using HTTP protocol and checks its search for the requested web page. If found it returns it back to the web browser and close the HTTP connection.
5. Now the web browser receives the web page, It interprets it and display the contents of web page in web browser's window.



WWW Future

There had been a rapid development in field of web. It has its impact in almost every area such as education, research, technology, commerce, marketing etc. So the future of web is almost unpredictable.

Apart from huge development in field of WWW, there are also some technical issues that W3 consortium has to cope up with.

USER INTERFACE

Work on higher quality presentation of 3-D information is under deveopment. The W3 Consortium is also looking forward to enhance the web to full fill requirements of global communities which would include all regional languages and writing systems.

TECHNOLOGY

Work on privacy and security is under way. This would include hiding information, accounting, access control, integrity and risk management.

ARCHITECTURE

There has been huge growth in field of web which may lead to overload the internet and degrade its performance. Hence more better protocol are required to be developed.

Web Browsers

web Browser is an application software that allows us to view and explore information on the web. User can request for any web page by just entering a URL into address bar.

Web browser can show text, audio, video, animation and more. It is the responsibility of a web browser to interpret text and commands contained in the web page.

Earlier the web browsers were text-based while now a days graphical-based or voice-based web browsers

are also available. Following are the most common web browser available today:

Browser	Vendor
Internet Explorer	Microsoft
Google Chrome	Google
Mozilla Firefox	Mozilla
Netscape Navigator	Netscape Communications Corp.
Opera	Opera Software
Safari	Apple
Sea Monkey	Mozilla Foundation
K-meleon	K-meleon

Architecture

There are a lot of web browser available in the market. All of them interpret and display information on the screen however their capabilities and structure varies depending upon implementation. But the most basic component that all web browser must exhibit are listed below:

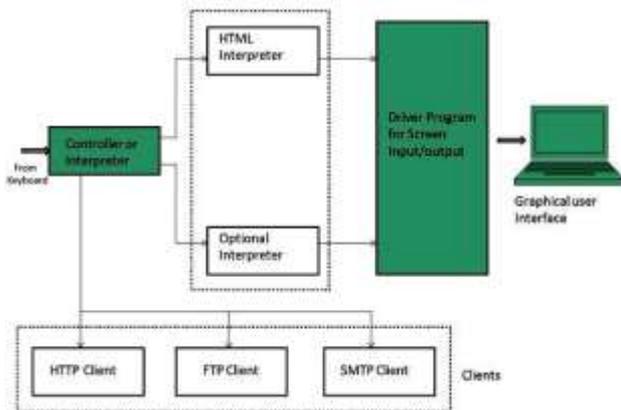
- Controller/Dispatcher
- Interpreter
- Client Programs

Controller works as a control unit in CPU. It takes input from the keyboard or mouse, interpret it and make other services to work on the basis of input it receives.

Interpreter receives the information from the controller and execute the instruction line by line. Some interpreter are mandatory while some are optional For example, HTML interpreter program is mandatory and java interpreter is optional.

Client Program describes the specific protocol that will be used to access a particular service. Following are the client programs that are commonly used:

- HTTP
- SMTP
- FTP
- NNTP
- POP



Web Server

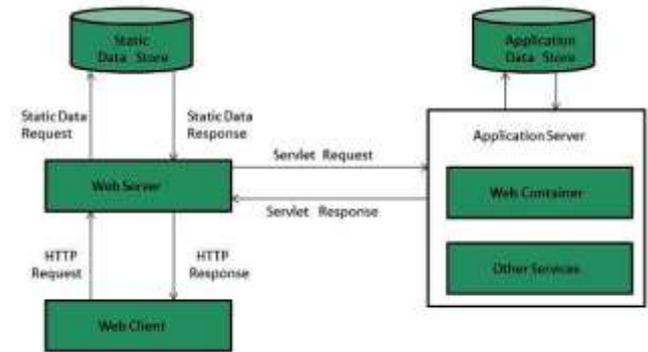
Web server is a computer where the web content is stored. Basically web server is used to host the web sites but there exists other web servers also such as gaming, storage, FTP, email etc.

Web site is collection of web pages while web server is a software that respond to the request for web resources.

Web Server Working

Web server respond to the client request in either of the following two ways:

- Sending the file to the client associated with the requested URL.
- Generating response by invoking a script and communicating with database



Key Points

- When client sends request for a web page, the web server search for the requested page if requested page is found then it will send it to client with an HTTP response.
- If the requested web page is not found, web server will the send an **HTTP response: Error 404 Not found**.
- If client has requested for some other resources then the web server will contact to the application server and data store to construct the HTTP response.

Architecture

Web Server Architecture follows the following two approaches:

1. Concurrent Approach
2. Single-Process-Event-Driven Approach.

CONCURRENT APPROACH

Concurrent approach allows the web server to handle multiple client requests at the same time. It can be achieved by following methods:

- Multi-process
- Multi-threaded
- Hybrid method.

MULTI-PROCESSING

In this a single process (parent process) initiates several single-threaded child processes and distribute incoming requests to these child processes. Each of the child processes are responsible for handling single request.

It is the responsibility of parent process to monitor the load and decide if processes should be killed or forked.

MULTI-THREADED

Unlike Multi-process, it creates multiple single-threaded process.

HYBRID

It is combination of above two approaches. In this approach multiple process are created and each process initiates multiple threads. Each of the threads handles one connection. Using multiple threads in single process results in less load on system resources.

Examples

Following table describes the most leading web servers available today:

S.N.	Web Server Descriptino
1	Apache HTTP Server This is the most popular web server in the world developed by the Apache Software Foundation. Apache web server is an open source software and can be installed on almost all operating systems including Linux, UNIX, Windows, FreeBSD, Mac OS X and more. About 60% of the web server machines run the Apache Web Server.
2.	Internet Information Services

	(IIS) The Internet Information Server (IIS) is a high performance Web Server from Microsoft. This web server runs on Windows NT/2000 and 2003 platforms (and may be on upcoming new Windows version also). IIS comes bundled with Windows NT/2000 and 2003; Because IIS is tightly integrated with the operating system so it is relatively easy to administer it.
3.	Lighttpd The lighttpd, pronounced lighty is also a free web server that is distributed with the FreeBSD operating system. This open source web server is fast, secure and consumes much less CPU power. Lighttpd can also run on Windows, Mac OS X, Linux and Solaris operating systems.
4.	Sun Java System Web Server This web server from Sun Microsystems is suited for medium and large web sites. Though the server is free it is not open source. It however, runs on Windows, Linux and UNIX platforms. The Sun Java System web server supports various languages, scripts and technologies required for Web 2.0 such as JSP, Java Servlets, PHP, Perl, Python, and Ruby on Rails, ASP and Coldfusion etc.
5.	Jigsaw Server Jigsaw (W3C's Server) comes from the World Wide Web Consortium. It is open source and free and can run on various platforms like Linux, UNIX, Windows, and Mac OS X Free BSD etc. Jigsaw has been written in Java and can run CGI scripts and PHP programs.

Proxy Server

Proxy server is an intermediary server between client and the internet. Proxy servers offers the following basic functionalities:

- Firewall and network data filtering.
- Network connection sharing
- Data caching

Proxy servers allow to hide, conceal and make your network id anonymous by hiding your IP address.

Purpose of Proxy Servers

Following are the reasons to use proxy servers:

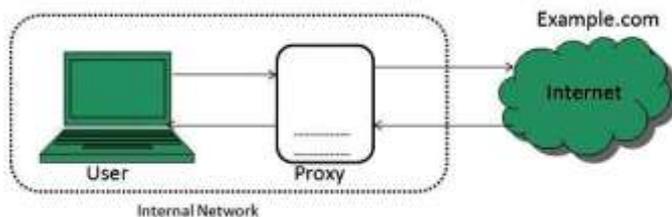
- Monitoring and Filtering
- Improving performance
- Translation
- Accessing services anonymously
- Security

Type of Proxies

Following table briefly describes the type of proxies:

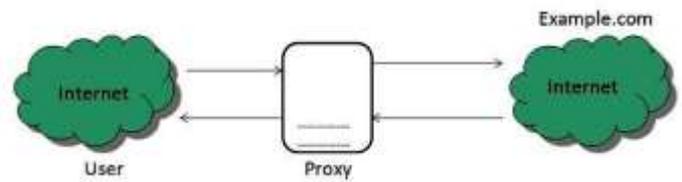
FORWARD PROXIES

In this the client requests its internal network server to forward to the internet.



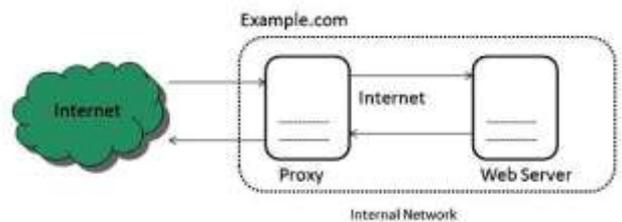
OPEN PROXIES

Open Proxies helps the clients to conceal their IP address while browsing the web.



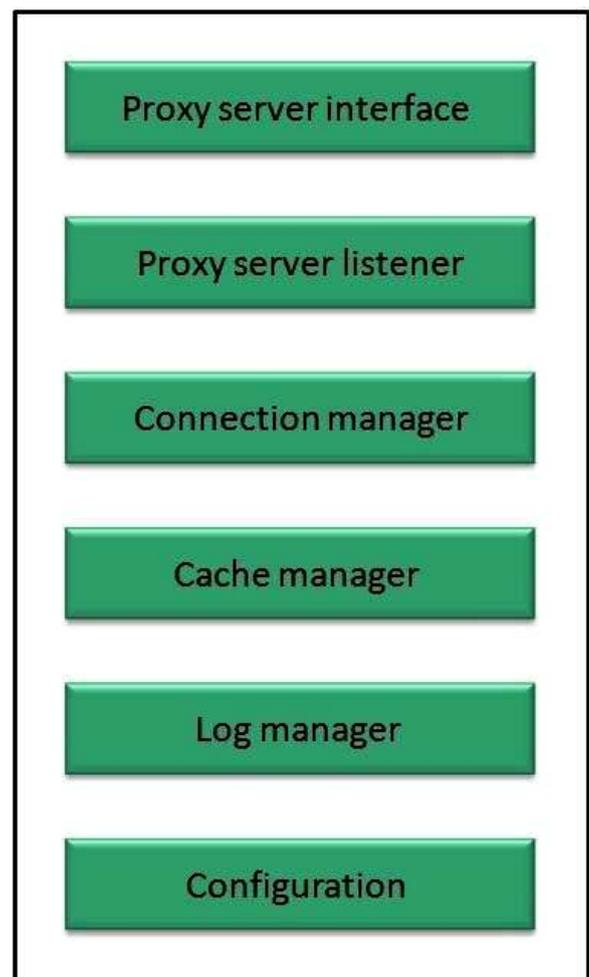
REVERSE PROXIES

In this the requests are forwarded to one or more proxy servers and the response from the proxy server is retrieved as if it came directly from the original Server.



Architecture

The proxy server architecture is divided into several modules as shown in the following diagram:



PROXY USER INTERFACE

This module controls and manages the user interface and provides an easy to use graphical interface, window and a menu to the end user. This menu offers the following functionalities:

- Start proxy
- Stop proxy
- Exit
- Blocking URL
- Blocking client
- Manage log
- Manage cache
- Modify configuration

PROXY SERVER LISTENER

It is the port where new request from the client browser is listened. This module also performs blocking of clients from the list given by the user.

CONNECTION MANAGER

It contains the main functionality of the proxy server. It performs the following functions:

- It contains the main functionality of the proxy server. It performs the following functions:
- Read request from header of the client.
- Parse the URL and determine whether the URL is blocked or not.
- Generate connection to the web server.
- Read the reply from the web server.
- If no copy of page is found in the cache then download the page from web server else will check its last modified date from the reply header and accordingly will read from the cache or server from the web.

- Then it will also check whether caching is allowed or not and accordingly will cache the page.

CACHE MANAGER

This module is responsible for storing, deleting, clearing and searching of web pages in the cache.

LOG MANAGER

This module is responsible for viewing, clearing and updating the logs.

CONFIGURATION

This module helps to create configuration settings which in turn let other modules to perform desired configurations such as caching.

Search Engines

Search Engine refers to a huge database of internet resources such as web pages, newsgroups, programs, images etc. It helps to locate information on World Wide Web.

User can search for any information by passing query in form of keywords or phrase. It then searches for relevant information in its database and return to the user.



Search Engine Components

Generally there are three basic components of a search engine as listed below:

1. Web Crawler
2. Database
3. Search Interfaces

WEB CRAWLER

It is also known as **spider** or **bots**. It is a software component that traverses the web to gather information.

DATABASE

All the information on the web is stored in database. It consists of huge web resources.

SEARCH INTERFACES

This component is an interface between user and the database. It helps the user to search through the database.

Search Engine Working

Web crawler, database and the search interface are the major component of a search engine that actually makes search engine to work. Search engines make use of Boolean expression AND, OR, NOT to restrict and widen the results of a search. Following are the steps that are performed by the search engine:

- The search engine looks for the keyword in the index for predefined database instead of going directly to the web to search for the keyword.
- It then uses software to search for the information in the database. This software component is known as web crawler.
- Once web crawler finds the pages, the search engine then

shows the relevant web pages as a result. These retrieved web pages generally include title of page, size of text portion, first several sentences etc.

These search criteria may vary from one search engine to the other. The retrieved information is ranked according to various factors such as frequency of keywords, relevancy of information, links etc.

- User can click on any of the search results to open it.

Usenet (USER NETWORK)

Like mailing lists Usenet is also a way of sharing information. It was started by Tom Truscott and Jim Ellis in 1979. Initially it was limited to two sites but today there are thousands of Usenet sites involving millions of people.

Usenet is a kind of discussion group where people can share views on topic of their interest. The article posted to a newsgroup becomes available to all readers of the newsgroup.

Mailing list vs. Newsgroup

S. N.	Mailing List	Newsgroup
1.	Messages are delivered to individual mailboxes of subscribed member of group.	Messages are not posted to individual mailboxes but can be viewed by anyone who has subscribed to that newsgroup.
2.	Working with mailing list is easier than newsgroup. It is easy to compose and receive emails.	Working with a particular newsgroup requires proper knowledge of that newsgroup.

3.	In order to send or receive mails, you required an email program.	It requires a newsgroup reader.
4.	Messages are delivered to certain group of people.	Messages are available to public.
5.	Mailing list does not support threaded discussion.	Newsgroup supports threaded discussion.
6.	Messages delivered to listed subscribers can not be cancelled.	Article posted on a newsgroup can be cancelled.

Online Education

There are several forms of online education available as discussed below:

Online Training

Online Training is a form of distance learning in which educational information is delivered through internet. There are many online applications. These applications vary from simple downloadable content to structured programs.

Social Networking

Social Networking refers to grouping of individuals and organizations together via some medium, in order to share thoughts, interests, and activities.

There are several web based social network services are available such as facebook, twitter, linkedin, Google+ etc. which offer easy to use and interactive interface to connect with people with in the country an overseas as well. There are also several mobile

based social networking services in for of apps such as Whatsapp, hike, Line etc.

Available Social networking Services

The following table describes some of the famous social networking services provided over web and mobile:

S. N.	Service Description
1.	Facebook Allows to share text, photos, video etc. It also offers interesting online games.
2.	Google+ It is pronounced as Google Plus. It is owned and operated by Google.
3.	Twitter Twitter allows the user to send and reply messages in form of tweets. These tweets are the small messages, generally include 140+ characters.
4.	Faceparty Faceparty is a UK based social networking site. It allows the users to create profiles and interact with each other using forums messages.
5.	LinkedIn Linkedin is a business and professional networking site.
6.	Flicker Flicker offers image hosting and video hosting.
7.	Ibibio Ibibo is a talent based social networking site. It allows the users to promote one's self and also discover new talent.
8.	Whatsapp It is a mobile based messaging app. It allows to send text, video, and

	audio messages
9.	Line It is same as whatsapp. Allows to make free calls and messages.
10.	Hike It is also mobile based messenger allows to send messages and exciting emoticons.

Internet Security

Internet security refers to securing communication over the internet. It includes specific security protocols such as:

- Internet Security Protocol (IPSec)
- Secure Socket Layer (SSL)

Threats

Internet security threats impact the network, data security and other internet connected systems. Cyber criminals have evolved several techniques to threat privacy and integrity of bank accounts, businesses, and organizations. Following are some of the internet security threats:

- Mobile worms
- Malware
- PC and Mobile ransomware
- Large scale attacks like Stuxnet that attempts to destroy infrastructure.
- Hacking as a Service
- Spam
- Phishing

Email Phishing

Email phishing is an activity of sending emails to a user claiming to be a legitimate enterprise. Its main purpose is to steal sensitive information such as usernames, passwords, and credit card details. Such emails contains link to websites that are infected with malware and direct the user to enter details at a fake website whose look and feels are same to legitimate one.

WHAT A PHISHING EMAIL MAY CONTAIN?

Following are the symptoms of a phishing email:

SPELLING AND BAD GRAMMAR

Most often such emails contain grammatically incorrect text. Ignore such emails, since it can be a spam.

BEWARE OF LINKS IN EMAIL

Don't click on any links in suspicious emails.

THREATS

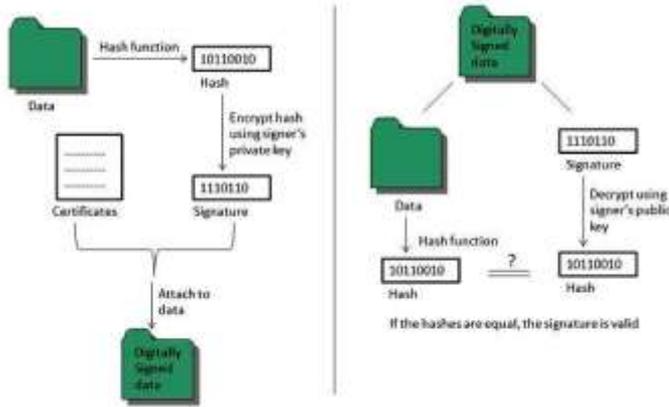
Such emails contain threat like "your account will be closed if you didn't respond to an email message".

SPOOFING POPULAR WEBSITES OR COMPANIES

These emails contain graphics that appear to be connected to legitimate website but they actually are connected to fake websites.

Digital Signature

Digital signatures allow us to verify the author, date and time of signatures, authenticate the message contents. It also includes authentication function for additional capabilities.



A digital signature should not only be tied to the signing user, but also to the message.

Applications

There are several reasons to implement digital signatures to communications:

AUTHENTICATION

Digital signatures help to authenticate the sources of messages. For example, if a bank's branch office sends a message to central office, requesting for change in balance of an account. If the central office could not authenticate that message is sent from an authorized source, acting of such request could be a grave mistake.

INTEGRITY

Once the message is signed, any change in the message would invalidate the signature.

NON-REPUDIATION

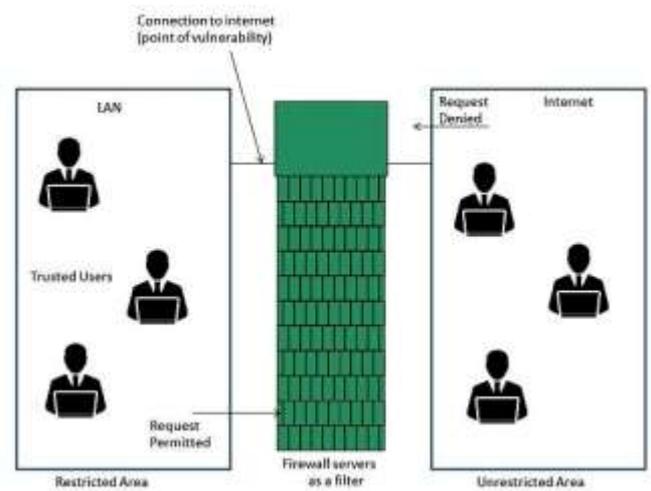
By this property, any entity that has signed some information cannot at a later time deny having signed it.

Firewall Security

Firewall is a barrier between Local Area Network (LAN) and the Internet. It allows keeping private resources confidential and minimizes the

security risks. It controls network traffic, in both directions.

The following diagram depicts a sample firewall between LAN and the internet. The connection between the two is the point of vulnerability. Both hardware and the software can be used at this point to filter network traffic.



There are two types of Firewall system: One works by using filters at the network layer and the other works by using proxy servers at the user, application, or network layer.

Key Points

- Firewall management must be addressed by both system managers and the network managers.
- The amount of filtering a firewall varies. For the same firewall, the amount of filtering may be different in different directions.

UNIT - 4

HTML – Overview

HTML stands for **H**ypertext **M**arkup **L**anguage, and it is the most widely used language to write Web Pages.

- **Hypertext** refers to the way in which Web pages (HTML documents) are linked together. Thus, the link available on a webpage is called Hypertext.
- As its name suggests, HTML is a **Markup Language** which means you use HTML to simply "mark-up" a text document with tags that tell a Web browser how to structure it to display.

Originally, HTML was developed with the intent of defining the structure of documents like headings, paragraphs, lists, and so forth to facilitate the sharing of scientific information between researchers.

Now, HTML is being widely used to format web pages with the help of different tags available in HTML language.

Basic HTML Document

In its simplest form, following is an example of an HTML document –

```
<!DOCTYPE html>
<html>

  <head>
    <title>This is document
title</title>
  </head>

  <body>
    <h1>This is a heading</h1>
    <p>Document content goes
here.....</p>
  </body>

</html>
```

HTML Tags

As told earlier, HTML is a markup language and makes use of various tags to format the content. These tags are enclosed within angle braces **<Tag Name>**. Except few tags, most of the tags have their corresponding closing tags. For example, **<html>** has its closing tag **</html>** and **<body>** tag has its closing tag **</body>** tag etc. Above example of HTML document uses the following tags –

Sr. No	Tag & Description
1	<!DOCTYPE...> This tag defines the document type and HTML version.
2	<html> This tag encloses the complete HTML document and mainly comprises of document header which is represented by <head>...</head> and document body which is represented by <body>...</body> tags.
3	<head> This tag represents the document's header which can keep other HTML tags like <title> , <link> etc.
4	<title> The <title> tag is used inside the <head> tag to mention the document title.
5	<body> This tag represents the document's body which keeps other HTML tags like <h1> , <div> , <p> etc.

6	<h1> This tag represents the heading.
7	<p> This tag represents a paragraph.

To learn HTML, you will need to study various tags and understand how they behave, while formatting a textual document. Learning HTML is simple as users have to learn the usage of different tags in order to format the text or images to make a beautiful webpage.

World Wide Web Consortium (W3C) recommends to use lowercase tags starting from HTML 4.

HTML Document Structure

A typical HTML document will have the following structure –

```
<html>
  <head>
    Document header related tags
  </head>
  <body>
    Document body related tags
  </body>
</html>
```

We will study all the header and body tags in subsequent chapters, but for now let's see what is document declaration tag.

The <!DOCTYPE> Declaration

The <!DOCTYPE> declaration tag is used by the web browser to understand the version of the HTML used in the document. Current version of HTML is 5 and it makes use of the following declaration –

```
<!DOCTYPE html>
```

There are many other declaration types which can be used in HTML document depending on what version of HTML is being used. We will see more details on this while discussing <!DOCTYPE...> tag along with other HTML tags.

HTML - Basic Tags

Heading Tags

Any document starts with a heading. You can use different sizes for your headings. HTML also has six levels of headings, which use the elements **<h1>**, **<h2>**, **<h3>**, **<h4>**, **<h5>**, and **<h6>**. While displaying any heading, browser adds one line before and one line after that heading.

Example

```
<!DOCTYPE html>
<html>
  <head>
    <title>Heading Example</title>
  </head>
  <body>
    <h1>This is heading 1</h1>
```

```

<h2>This is heading 2</h2>
<h3>This is heading 3</h3>
<h4>This is heading 4</h4>
<h5>This is heading 5</h5>
<h6>This is heading 6</h6>
</body>

```

```
</html>
```

This will produce the following result –

Paragraph Tag

The **<p>** tag offers a way to structure your text into different paragraphs. Each paragraph of text should go in between an opening **<p>** and a closing **</p>** tag as shown below in the example –

Example

```

<!DOCTYPE html>
<html>

  <head>
    <title>Paragraph
Example</title>
  </head>

  <body>
    <p>Here is a first paragraph
of text.</p>
    <p>Here is a second paragraph
of text.</p>
    <p>Here is a third paragraph
of text.</p>
  </body>

</html>

```

This will produce the following result –

Line Break Tag

Whenever you use the **
** element, anything following it starts from the next line. This tag is an example of an **empty** element, where you do not need opening and

closing tags, as there is nothing to go in between them.

The **
** tag has a space between the characters **br** and the forward slash. If you omit this space, older browsers will have trouble rendering the line break, while if you miss the forward slash character and just use **
** it is not valid in XHTML.

Example

```

<!DOCTYPE html>
<html>

  <head>
    <title>Line Break
Example</title>
  </head>

  <body>
    <p>Hello<br />
      You delivered your
assignment ontime.<br />
      Thanks<br />
      Mahnaz</p>
  </body>

</html>

```

This will produce the following result –

Centering Content

You can use **<center>** tag to put any content in the center of the page or any table cell.

Example

```

<!DOCTYPE html>
<html>

  <head>
    <title>Centring Content
Example</title>
  </head>

  <body>

```

```

    <p>This text is not in the
center.</p>

    <center>
    <p>This text is in the
center.</p>
    </center>
</body>
</html>

```

This will produce following result –

Horizontal Lines

Horizontal lines are used to visually break-up sections of a document. The **<hr>** tag creates a line from the current position in the document to the right margin and breaks the line accordingly.

For example, you may want to give a line between two paragraphs as in the given example below –

Example

```

<!DOCTYPE html>
<html>

    <head>
        <title>Horizontal Line
Example</title>
    </head>

    <body>
        <p>This is paragraph one and
should be on top</p>
        <hr />
        <p>This is paragraph two and
should be at bottom</p>
    </body>

</html>

```

This will produce the following result – Again **<hr />** tag is an example of the **empty** element, where you do not need opening and closing tags, as

there is nothing to go in between them.

The **<hr />** element has a space between the characters **hr** and the forward slash. If you omit this space, older browsers will have trouble rendering the horizontal line, while if you miss the forward slash character and just use **<hr>** it is not valid in XHTML

Preserve Formatting

Sometimes, you want your text to follow the exact format of how it is written in the HTML document. In these cases, you can use the preformatted tag **<pre>**.

Any text between the opening **<pre>** tag and the closing **</pre>** tag will preserve the formatting of the source document.

Example

```

<!DOCTYPE html>
<html>

    <head>
        <title>Preserve Formatting
Example</title>
    </head>

    <body>
        <pre>
            function testFunction(
strText ){
                alert (strText)
            }
        </pre>
    </body>

</html>

```

This will produce the following result – Try using the same code without keeping it inside **<pre>...</pre>** tags

Nonbreaking Spaces

Suppose you want to use the phrase "12 Angry Men." Here, you would not want a browser to split the "12, Angry" and "Men" across two lines –

An example of this technique appears in the movie "12 Angry Men."

In cases, where you do not want the client browser to break text, you should use a nonbreaking space entity ** ** instead of a normal space. For example, when coding the "12 Angry Men" in a paragraph, you should use something similar to the following code –

Example

```
<!DOCTYPE html>
<html>

  <head>
    <title>Nonbreaking Spaces
    Example</title>
  </head>

  <body>
    <p>An example of this
    technique appears in the movie
    "12&nbsp;Angry&nbsp;Men."</p>
  </body>

</html>
```

This will produce the following result –

HTML - Elements

An **HTML element** is defined by a starting tag. If the element contains other content, it ends with a closing tag, where the element name is preceded by a forward slash as shown below with few tags –

Start Tag	Content	End Tag
<p>	This is paragraph content.	</p>
<h1>	This is heading content.	</h1>
<div>	This is division content.	</div>

So here **<p>....</p>** is an HTML element, **<h1>...</h1>** is another HTML element. There are some HTML elements which don't need to be closed, such as **<img.../>**, **<hr />** and **
** elements. These are known as **void elements**.

HTML documents consists of a tree of these elements and they specify how HTML documents should be built, and what kind of content should be placed in what part of an HTML document.

HTML Tag vs. Element

An HTML element is defined by a *starting tag*. If the element contains other content, it ends with a *closing tag*.

For example, **<p>** is starting tag of a paragraph and **</p>** is closing tag of the same paragraph but **<p>This is paragraph</p>** is a paragraph element.

Nested HTML Elements

It is very much allowed to keep one HTML element inside another HTML element –

Example

```
<!DOCTYPE html>
<html>

  <head>
    <title>Nested Elements
Example</title>
  </head>

  <body>
    <h1>This is <i>italic</i>
heading</h1>
    <p>This is <u>underlined</u>
paragraph</p>
  </body>

</html>
```

This will display the following result –

HTML – Attributes

We have seen few HTML tags and their usage like heading tags **<h1>**, **<h2>**, paragraph tag **<p>** and other tags. We used them so far in their simplest form, but most of the HTML tags can also have attributes, which are extra bits of information.

An attribute is used to define the characteristics of an HTML element and is placed inside the element's opening tag. All attributes are made up of two parts – a **name** and a **value**

- The **name** is the property you want to set. For example, the paragraph **<p>** element in the example carries an attribute whose name is **align**, which you can use to indicate the alignment of paragraph on the page.
- The **value** is what you want the value of the property to be set and always put within quotations. The below example shows three possible values of

align attribute: **left**, **center** and **right**.

Attribute names and attribute values are case-insensitive. However, the World Wide Web Consortium (W3C) recommends lowercase attributes/attribute values in their HTML 4 recommendation.

Example

```
<!DOCTYPE html>
<html>

  <head>
    <title>Align Attribute
Example</title>
  </head>

  <body>
    <p align = "left">This is left
aligned</p>
    <p align = "center">This is
center aligned</p>
    <p align = "right">This is
right aligned</p>
  </body>

</html>
```

This will display the following result –

Core Attributes

The four core attributes that can be used on the majority of HTML elements (although not all) are –

- Id
- Title
- Class
- Style

The Id Attribute

The **id** attribute of an HTML tag can be used to uniquely identify any element within an HTML page. There are two primary reasons that you might want to use an id attribute on an element –

- If an element carries an id attribute as a unique identifier,

it is possible to identify just that element and its content.

- If you have two elements of the same name within a Web page (or style sheet), you can use the id attribute to distinguish between elements that have the same name.

We will discuss style sheet in separate tutorial. For now, let's use the id attribute to distinguish between two paragraph elements as shown below.

Example

```
<p id = "html">This para explains
what is HTML</p>
<p id = "css">This para explains
what is Cascading Style Sheet</p>
```

The title Attribute

The **title** attribute gives a suggested title for the element. The syntax for the **title** attribute is similar as explained for **id** attribute –

The behavior of this attribute will depend upon the element that carries it, although it is often displayed as a tooltip when cursor comes over the element or while the element is loading.

Example

```
<!DOCTYPE html>
<html>

  <head>
    <title>The title Attribute
Example</title>
  </head>

  <body>
    <h3 title = "Hello
HTML!">Titled Heading Tag
Example</h3>
  </body>
```

```
</html>
```

This will produce the following result – Now try to bring your cursor over "Titled Heading Tag Example" and you will see that whatever title you used in your code is coming out as a tooltip of the cursor.

The class Attribute

The **class** attribute is used to associate an element with a style sheet, and specifies the class of element. You will learn more about the use of the class attribute when you will learn Cascading Style Sheet (CSS). So for now you can avoid it. The value of the attribute may also be a space-separated list of class names. For example –

```
class = "className1 className2
className3"
```

The style Attribute

The style attribute allows you to specify Cascading Style Sheet (CSS) rules within the element.

```
<!DOCTYPE html>
<html>

  <head>
    <title>The style
Attribute</title>
  </head>

  <body>
    <p style = "font-family:arial;
color:#FF0000;">Some text...</p>
  </body>

</html>
```

This will produce the following result – At this point of time, we are not learning CSS, so just let's proceed

without bothering much about CSS. Here, you need to understand what are HTML attributes and how they can be used while formatting content.

Internationalization Attributes

There are three internationalization attributes, which are available for most (although not all) XHTML elements.

- dir
- lang
- xml:lang

The dir Attribute

The **dir** attribute allows you to indicate to the browser about the direction in which the text should flow. The dir attribute can take one of two values, as you can see in the table that follows –

Value	Meaning
ltr	Left to right (the default value)
rtl	Right to left (for languages such as Hebrew or Arabic that are read right to left)

Example

```
<!DOCTYPE html>
<html dir = "rtl">

  <head>
    <title>Display
    Directions</title>
  </head>

  <body>
    This is how IE 5 renders
    right-to-left directed text.
  </body>
```

```
</html>
```

This will produce the following result – When *dir* attribute is used within the <html> tag, it determines how text will be presented within the entire document. When used within another tag, it controls the text's direction for just the content of that tag.

Generic Attributes

Here's a table of some other attributes that are readily usable with many of the HTML tags.

Attribute	Options	Function
align	right, left, center	Horizontally aligns tags
valign	top, middle, bottom	Vertically aligns tags within an HTML element.
bgcolor	numeric, hexadecimal, RGB values	Places a background color behind an element
background	URL	Places a background image behind an element
id	User Defined	Names an element for use with Cascading Style Sheets.
class	User Defined	Classifies an element for use with Cascading Style Sheets.
width	Numeric Value	Specifies the width of tables, images, or table

		cells.
height	Numeric Value	Specifies the height of tables, images, or table cells.
title	User Defined	"Pop-up" title of the elements.

We will see related examples as we will proceed to study other HTML tags. For a complete list of HTML Tags and related attributes please check reference to [HTML Tags List](#).

UNIT-5

HTML – Formatting

If you use a word processor, you must be familiar with the ability to make text bold, italicized, or underlined; these are just three of the ten options

available to indicate how text can appear in HTML and XHTML.

Bold Text

Anything that appears within `...` element, is displayed in bold as shown below –

Example

```
<!DOCTYPE html>
<html>

  <head>
    <title>Bold Text
    Example</title>
  </head>

  <body>
    <p>The following word uses a
    <b>bold</b> typeface.</p>
  </body>

</html>
```

This will produce the following result –

Italic Text

Anything that appears within `<i>...</i>` element is displayed in italicized as shown below –

Example

```
<!DOCTYPE html>
<html>

  <head>
    <title>Italic Text
    Example</title>
  </head>

  <body>
    <p>The following word uses an
    <i>italicized</i> typeface.</p>
  </body>

</html>
```

This will produce the following result –

Underlined Text

Anything that appears within **<u>...</u>** element, is displayed with underline as shown below –

Example

```
<!DOCTYPE html>
<html>

  <head>
    <title>Underlined Text
    Example</title>
  </head>

  <body>
    <p>The following word uses an
    <u>underlined</u> typeface.</p>
  </body>

</html>
```

This will produce the following result –

Strike Text

Anything that appears within **<strike>...</strike>** element is displayed with strikethrough, which is a thin line through the text as shown below –

Example

```
<!DOCTYPE html>
<html>

  <head>
    <title>Strike Text
    Example</title>
  </head>

  <body>
```

```
<p>The following word uses a
<strike>strikethrough</strike>
typeface </n>
</body>

</html>
```

Superscript Text

The content of a **^{...}** element is written in superscript; the font size used is the same size as the characters surrounding it but is displayed half a character's height above the other characters.

Example

```
<!DOCTYPE html>
<html>

  <head>
    <title>Superscript Text
    Example</title>
  </head>

  <body>
    <p>The following word uses a
    <sup>superscript</sup> typeface.</p>
  </body>

</html>
```

This will produce the following result –

Subscript Text

The content of a **_{...}** element is written in subscript; the font size used is the same as the characters surrounding it, but is displayed half a character's height beneath the other characters.

Example

```
<!DOCTYPE html>
```

```
<html>
  <head>
    <title>Subscript Text
Example</title>
  </head>

  <body>
    <p>The following word uses a
<sub>subscript</sub> typeface.</p>
  </body>

</html>
```

This will produce the following result –

Inserted Text

Anything that appears within **<ins>...</ins>** element is displayed as inserted text.

Example

```
<!DOCTYPE html>
<html>

  <head>
    <title>Inserted Text
Example</title>
  </head>

  <body>
    <p>I want to drink
<del>cola</del> <ins>wine</ins></p>
  </body>

</html>
```

This will produce the following result –

Deleted Text

Anything that appears within **...** element, is displayed as deleted text.

Example

```
<!DOCTYPE html>
<html>

  <head>
```

```
  <title>Deleted Text
Example</title>
  </head>

  <body>
    <p>I want to drink
<del>cola</del> <ins>wine</ins></p>
  </body>

</html>
```

This will produce the following result –

Emphasized Text

Anything that appears within **...** element is displayed as emphasized text.

Example

```
<!DOCTYPE html>
<html>

  <head>
    <title>Emphasized Text
Example</title>
  </head>

  <body>
    <p>The following word uses an
<em>emphasized</em> typeface.</p>
  </body>

</html>
```

This will produce the following result –

Marked Text

Anything that appears within **<mark>...</mark>** element, is displayed as marked with yellow ink.

Example

```
<!DOCTYPE html>
<html>

  <head>
    <title>Marked Text
Example</title>
```

```

</head>

<body>
  <p>The following word has been
<mark>marked</mark> with yellow</p>
</body>

</html>

```

This will produce the following result –

Strong Text

Anything that appears within **...** element is displayed as important text.

Example

```

<!DOCTYPE html>
<html>

  <head>
    <title>Strong Text
    Example</title>
  </head>

  <body>
    <p>The following word uses a
<strong>strong</strong>
typeface.</p>
  </body>

</html>

```

This will produce the following result –

Text Abbreviation

You can abbreviate a text by putting it inside opening **<abbr>** and closing **</abbr>** tags. If present, the title attribute must contain this full description and nothing else.

Example

```

<!DOCTYPE html>
<html>

  <head>
    <title>Text
    Abbreviation</title>

```

```

</head>

<body>
  <p>My best friend's name is
<abbr title =
"Abhishek">Abhy</abbr>.</p>
</body>

</html>

```

This will produce the following result –

HTML - Meta Tags

HTML lets you specify metadata - additional important information about a document in a variety of ways. The META elements can be used to include name/value pairs describing properties of the HTML document, such as author, expiry date, a list of keywords, document author etc.

The **<meta>** tag is used to provide such additional information. This tag is an empty element and so does not have a closing tag but it carries information within its attributes.

You can include one or more meta tags in your document based on what information you want to keep in your document but in general, meta tags do not impact physical appearance of the document so from appearance point of view, it does not matter if you include them or not.

Adding Meta Tags to Your Documents

You can add metadata to your web pages by placing **<meta>** tags inside the header of the document which is represented

by **<head>** and **</head>** tags. A meta tag can have following attributes in addition to core attributes –

Sr. No	Attribute & Description
1	Name

	Name for the property. Can be anything. Examples include, keywords, description, author, revised, generator etc.
2	content Specifies the property's value.
3	scheme Specifies a scheme to interpret the property's value (as declared in the content attribute).
4	http-equiv Used for http response message headers. For example, http-equiv can be used to refresh the page or to set a cookie. Values include content-type, expires, refresh and set-cookie.

Specifying Keywords

You can use <meta> tag to specify important keywords related to the document and later these keywords are used by the search engines while indexing your webpage for searching purpose.

Example

Following is an example, where we are adding HTML, Meta Tags, Metadata as important keywords about the document.

```
<!DOCTYPE html>
<html>

  <head>
    <title>Meta Tags
    Example</title>
    <meta name = "keywords"
    content = "HTML, Meta Tags,
    Metadata" />
  </head>

  <body>
    <p>Hello HTML5!</p>
  </body>
```

```
</html>
```

This will produce the following result –

Document Description

You can use <meta> tag to give a short description about the document. This again can be used by various search engines while indexing your webpage for searching purpose.

Example

```
<!DOCTYPE html>
<html>

  <head>
    <title>Meta Tags
    Example</title>
    <meta name = "keywords"
    content = "HTML, Meta Tags,
    Metadata" />
    <meta name = "description"
    content = "Learning about Meta
    Tags." />
  </head>

  <body>
    <p>Hello HTML5!</p>
  </body>

</html>
```

Document Revision Date

You can use <meta> tag to give information about when last time the document was updated. This information can be used by various web browsers while refreshing your webpage.

Example

```
<!DOCTYPE html>
<html>

  <head>
    <title>Meta Tags
    Example</title>
```

```

    <meta name = "keywords"
content = "HTML, Meta Tags,
Metadata" />
    <meta name = "description"
content = "Learning about Meta
Tags." />
    <meta name = "revised" content
= "Tutorialspoint, 3/7/2014" />
</head>

<body>
    <p>Hello HTML5!</p>
</body>

</html>

```

Document Refreshing

A <meta> tag can be used to specify a duration after which your web page will keep refreshing automatically.

Example

If you want your page keep refreshing after every 5 seconds then use the following syntax.

```

<!DOCTYPE html>
<html>

    <head>
        <title>Meta Tags
Example</title>
        <meta name = "keywords"
content = "HTML, Meta Tags,
Metadata" />
        <meta name = "description"
content = "Learning about Meta
Tags." />
        <meta name = "revised" content
= "Tutorialspoint, 3/7/2014" />
        <meta http-equiv = "refresh"
content = "5" />
    </head>

    <body>
        <p>Hello HTML5!</p>
    </body>

</html>

```

Page Redirection

You can use <meta> tag to redirect your page to any other webpage. You can also specify a duration if you want to redirect the page after a certain number of seconds.

Example

Following is an example of redirecting current page to another page after 5 seconds. If you want to redirect page immediately then do not specify *content* attribute.

```

<!DOCTYPE html>
<html>

    <head>
        <title>Meta Tags
Example</title>
        <meta name = "keywords"
content = "HTML, Meta Tags,
Metadata" />
        <meta name = "description"
content = "Learning about Meta
Tags." />
        <meta name = "revised" content
= "Tutorialspoint, 3/7/2014" />
        <meta http-equiv = "refresh"
content = "5; url =
http://www.tutorialspoint.com" />
    </head>

    <body>
        <p>Hello HTML5!</p>
    </body>

</html>

```

HTML – Comments

Comment is a piece of code which is ignored by any web browser. It is a good practice to add comments into your HTML code, especially in complex documents, to indicate sections of a document, and any other notes to

anyone looking at the code. Comments help you and others understand your code and increases code readability.

HTML comments are placed in between `<!-- ... -->` tags. So, any content placed with-in `<!-- ... -->` tags will be treated as comment and will be completely ignored by the browser.

Example

```
<!DOCTYPE html>
<html>

  <head> <!-- Document Header
Starts -->
    <title>This is document
title</title>
  </head> <!-- Document Header Ends
-->

  <body>
    <p>Document content goes
here.....</p>
  </body>

</html>
```

This will produce the following result without displaying the content given as a part of comments –

HTML – Images

Images are very important to beautify as well as to depict many complex concepts in simple way on your web page. This tutorial will take you through simple steps to use images in your web pages.

Insert Image

You can insert any image in your web page by using **** tag. Following is the simple syntax to use this tag.

```
<img src = "Image URL" ...
attributes-list/>
```

The `` tag is an empty tag, which means that, it can contain only list of attributes and it has no closing tag.

Example

To try following example, let's keep our HTML file test.htm and image file test.png in the same directory –

```
<!DOCTYPE html>
<html>

  <head>
    <title>Using Image in
Webpage</title>
  </head>

  <body>
    <p>Simple Image Insert</p>
    <img src =
"/html/images/test.png" alt = "Test
Image" />
  </body>

</html>
```

This will produce the following result – You can use PNG, JPEG or GIF image file based on your comfort but make sure you specify correct image file name in **src** attribute. Image name is always case sensitive.

The **alt** attribute is a mandatory attribute which specifies an alternate text for an image, if the image cannot be displayed.

Set Image Location

Usually we keep all the images in a separate directory. So let's keep HTML file test.htm in our home directory and create a subdirectory **images** inside the home directory where we will keep our image test.png.

Example

Assuming our image location is "image/test.png", try the following example –

```
<!DOCTYPE html>
<html>

  <head>
    <title>Using Image in
Webpage</title>
  </head>

  <body>
    <p>Simple Image Insert</p>
    <img src =
"/html/images/test.png" alt = "Test
Image" />
  </body>

</html>
```

This will produce the following result –

Set Image Width/Height

You can set image width and height based on your requirement using **width** and **height** attributes. You can specify width and height of the image in terms of either pixels or percentage of its actual size.

Example

```
<!DOCTYPE html>
<html>

  <head>
    <title>Set Image Width and
Height</title>
  </head>

  <body>
    <p>Setting image width and
height</p>
    <img src =
"/html/images/test.png" alt = "Test
```

```
Image" width = "150" height =
"100"/>
  </body>

</html>
```

This will produce the following result –

Set Image Border

By default, image will have a border around it, you can specify border thickness in terms of pixels using border attribute. A thickness of 0 means, no border around the picture.

Example

```
<!DOCTYPE html>
<html>

  <head>
    <title>Set Image
Border</title>
  </head>

  <body>
    <p>Setting image Border</p>
    <img src =
"/html/images/test.png" alt = "Test
Image" border = "3"/>
  </body>

</html>
```

This will produce the following result –

Set Image Alignment

By default, image will align at the left side of the page, but you can use **align** attribute to set it in the center or right.

Example

```
<!DOCTYPE html>
<html>

  <head>
```

```

<title>Set Image
Alignment</title>
</head>

<body>
  <p>Setting image Alignment</p>
  <img src =
"/html/images/test.png" alt = "Test
Image" border = "3" align =
"right"/>
</body>

</html>

```

HTML – Tables

The HTML tables allow web authors to arrange data like text, images, links, other tables, etc. into rows and columns of cells.

The HTML tables are created using the **<table>** tag in which the **<tr>** tag is used to create table rows and **<td>** tag is used to create data cells. The elements under **<td>** are regular and left aligned by default

Example

```

<!DOCTYPE html>
<html>

  <head>
    <title>HTML Tables</title>
  </head>

  <body>
    <table border = "1">
      <tr>
        <td>Row 1, Column 1</td>
        <td>Row 1, Column 2</td>
      </tr>

      <tr>
        <td>Row 2, Column 1</td>
        <td>Row 2, Column 2</td>
      </tr>
    </table>

```

```

</body>
</html>

```

This will produce the following result – Here, the **border** is an attribute of **<table>** tag and it is used to put a border across all the cells. If you do not need a border, then you can use **border = "0"**.

Table Heading

Table heading can be defined using **<th>** tag. This tag will be put to replace **<td>** tag, which is used to represent actual data cell. Normally you will put your top row as table heading as shown below, otherwise you can use **<th>** element in any row. Headings, which are defined in **<th>** tag are centered and bold by default.

Example

```

<!DOCTYPE html>
<html>

  <head>
    <title>HTML Table
Header</title>
  </head>

  <body>
    <table border = "1">
      <tr>
        <th>Name</th>
        <th>Salary</th>
      </tr>
      <tr>
        <td>Ramesh Raman</td>
        <td>5000</td>
      </tr>

      <tr>
        <td>Shabbir Hussein</td>
        <td>7000</td>
      </tr>
    </table>
  </body>

```

```
</html>
```

Tables Backgrounds

You can set table background using one of the following two ways –

- **bgcolor** attribute – You can set background color for whole table or just for one cell.
- **background** attribute – You can set background image for whole table or just for one cell.

You can also set border color also using **bordercolor** attribute.

Note – The *bgcolor*, *background*, and *bordercolor* attributes deprecated in HTML5. Do not use these attributes.

Example

```
<!DOCTYPE html>
<html>

  <head>
    <title>HTML Table
    Background</title>
  </head>

  <body>
    <table border = "1"
    bordercolor = "green" bgcolor =
    "yellow">
      <tr>
        <th>Column 1</th>
        <th>Column 2</th>
        <th>Column 3</th>
      </tr>
      <tr>
        <td rowspan = "2">Row 1
        Cell 1</td>
        <td>Row 1 Cell 2</td>
        <td>Row 1 Cell 3</td>
      </tr>
      <tr>
        <td>Row 2 Cell 2</td>
        <td>Row 2 Cell 3</td>
      </tr>
    </table>
```

```
        <td colspan = "3">Row 3
        Cell 1</td>
      </tr>
    </table>
  </body>
</html>
```

This will produce the following result – Here is an example of using **background** attribute. Here we will use an image available in /images directory.

```
<!DOCTYPE html>
<html>

  <head>
    <title>HTML Table
    Background</title>
  </head>

  <body>
    <table border = "1"
    bordercolor = "green" background =
    "/images/test.png">
      <tr>
        <th>Column 1</th>
        <th>Column 2</th>
        <th>Column 3</th>
      </tr>
      <tr>
        <td rowspan = "2">Row 1
        Cell 1</td>
        <td>Row 1 Cell
        2</td><td>Row 1 Cell 3</td>
      </tr>
      <tr>
        <td>Row 2 Cell 2</td>
        <td>Row 2 Cell 3</td>
      </tr>
      <tr>
        <td colspan = "3">Row 3
        Cell 1</td>
      </tr>
    </table>
  </body>
</html>
```

This will produce the following result. Here background image did not apply to table's header.

Table Height and Width

You can set a table width and height using **width** and **height** attributes. You can specify table width or height in terms of pixels or in terms of percentage of available screen area.

Example

```
<!DOCTYPE html>
<html>

  <head>
    <title>HTML Table
    Width/Height</title>
  </head>

  <body>
    <table border = "1" width =
    "400" height = "150">
      <tr>
        <td>Row 1, Column 1</td>
        <td>Row 1, Column 2</td>
      </tr>

      <tr>
        <td>Row 2, Column 1</td>
        <td>Row 2, Column 2</td>
      </tr>
    </table>
  </body>
</html>
```

This will produce the following result –

Table Caption

The **caption** tag will serve as a title or explanation for the table and it shows up at the top of the table. This tag is deprecated in newer version of HTML/XHTML.

Example

```
<!DOCTYPE html>
<html>

  <head>
    <title>HTML Table
    Caption</title>
  </head>

  <body>
    <table border = "1" width =
    "100%">
      <caption>This is the
    caption</caption>

      <tr>
        <td>row 1, column
    1</td><td>row 1, columnn 2</td>
      </tr>

      <tr>
        <td>row 2, column
    1</td><td>row 2, columnn 2</td>
      </tr>
    </table>
  </body>
</html>
```

This will produce the following result –

Table Header, Body, and Footer

Tables can be divided into three portions – a header, a body, and a foot. The head and foot are rather similar to headers and footers in a word-processed document that remain the same for every page, while the body is the main content holder of the table.

The three elements for separating the head, body, and foot of a table are –

- **<thead>** – to create a separate table header.

- **<tbody>** – to indicate the main body of the table.
- **<tfoot>** – to create a separate table footer.

A table may contain several `<tbody>` elements to indicate *different pages* or groups of data. But it is notable that `<thead>` and `<tfoot>` tags should appear before `<tbody>`

Example

```
<!DOCTYPE html>
<html>

  <head>
    <title>HTML Table</title>
  </head>

  <body>
    <table border = "1" width =
"100%">
      <thead>
        <tr>
          <td colspan =
"4">This is the head of the
table</td>
        </tr>
      </thead>

      <tfoot>
        <tr>
          <td colspan =
"4">This is the foot of the
table</td>
        </tr>
      </tfoot>

      <tbody>
        <tr>
          <td>Cell 1</td>
          <td>Cell 2</td>
          <td>Cell 3</td>
          <td>Cell 4</td>
        </tr>
      </tbody>

    </table>
  </body>
```

```
</html>
```

This will produce the following result –

HTML - Text Links

A webpage can contain various links that take you directly to other pages and even specific parts of a given page. These links are known as hyperlinks.

Hyperlinks allow visitors to navigate between Web sites by clicking on words, phrases, and images. Thus you can create hyperlinks using text or images available on a webpage.

Note – I recommend you to go through a short tutorial on [Understanding URL](#)

Linking Documents

A link is specified using HTML tag `<a>`. This tag is called **anchor tag** and anything between the opening `<a>` tag and the closing `` tag becomes part of the link and a user can click that part to reach to the linked document. Following is the simple syntax to use `<a>` tag.

```
<a href = "Document URL" ...
attributes-list>Link Text</a>
```

Example

Let's try following example which links <http://www.tutorialspoint.com> at your page –

```
<!DOCTYPE html>
<html>

  <head>
    <title>Hyperlink
Example</title>
  </head>

  <body>
    <p>Click following link</p>
```

```
<a href =
"https://www.tutorialspoint.com"
target = "_self">Tutorials Point</a>
</body>

</html>
```

This will produce the following result, where you can click on the link generated to reach to the home page of Tutorials Point (in this example).

The target Attribute

We have used **target** attribute in our previous example. This attribute is used to specify the location where linked document is opened. Following are the possible options –

Sr. No	Option & Description
1	_blank Opens the linked document in a new window or tab.
2	_self Opens the linked document in the same frame.
3	_parent Opens the linked document in the parent frame.
4	_top Opens the linked document in the full body of the window.
5	targetframe Opens the linked document in a named <i>targetframe</i> .

Example

Try following example to understand basic difference in few options given for target attribute.

```
<!DOCTYPE html>
```

```
<html>

  <head>
    <title>Hyperlink
    Example</title>
    <base href =
"https://www.tutorialspoint.com/">
  </head>

  <body>
    <p>Click any of the following
    links</p>
    <a href = "/html/index.htm"
    target = "_blank">Opens in New</a> |
    <a href = "/html/index.htm"
    target = "_self">Opens in Self</a> |
    <a href = "/html/index.htm"
    target = "_parent">Opens in
    Parent</a> |
    <a href = "/html/index.htm"
    target = "_top">Opens in Body</a>
  </body>

</html>
```

This will produce the following result, where you can click on different links to understand the difference between various options given for target attribute.

HTML – Fonts

Fonts play a very important role in making a website more user friendly and increasing content readability. Font face and color depends entirely on the computer and browser that is being used to view your page but you can use HTML **** tag to add style, size, and color to the text on your website. You can use a **<basefont>** tag to set all of your text to the same size, face, and color. The font tag is having three attributes called **size**, **color**, and **face** to customize your fonts. To change any of the font attributes at any time within your webpage, simply use the **** tag. The text that follows will remain changed until you close with

the `` tag. You can change one or all of the font attributes within one `` tag.

Note –The *font* and *basefont* tags are 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 explain font and basefont tags in detail.

Set Font Size

You can set content font size using **size** attribute. The range of accepted values is from 1(smallest) to 7(largest). The default size of a font is 3.

Example

```
<!DOCTYPE html>
<html>

  <head>
    <title>Setting Font
    Size</title>
  </head>

  <body>
    <font size = "1">Font size =
    "1"</font><br />
    <font size = "2">Font size =
    "2"</font><br />
    <font size = "3">Font size =
    "3"</font><br />
    <font size = "4">Font size =
    "4"</font><br />
    <font size = "5">Font size =
    "5"</font><br />
    <font size = "6">Font size =
    "6"</font><br />
    <font size = "7">Font size =
    "7"</font>
  </body>

</html>
```

This will produce the following result –

Relative Font Size

You can specify how many sizes larger or how many sizes smaller than the preset font size should be. You can specify it like **** or ****

Example

```
<!DOCTYPE html>
<html>

  <head>
    <title>Relative Font
    Size</title>
  </head>

  <body>
    <font size = "-1">Font size =
    "-1"</font><br />
    <font size = "+1">Font size =
    "+1"</font><br />
    <font size = "+2">Font size =
    "+2"</font><br />
    <font size = "+3">Font size =
    "+3"</font><br />
    <font size = "+4">Font size =
    "+4"</font>
  </body>

</html>
```

This will produce the following result –

Setting Font Face

You can set font face using *face* attribute but be aware that if the user viewing the page doesn't have the font installed, they will not be able to see it. Instead user will see the default font face applicable to the user's computer.

Example

```
<!DOCTYPE html>
<html>

  <head>
```

```
<title>Font Face</title>
</head>

<body>
  <font face = "Times New Roman"
size = "5">Times New Roman</font><br
/>
  <font face = "Verdana" size =
"5">Verdana</font><br />
  <font face = "Comic sans MS"
size =" 5">Comic Sans MS</font><br
/>
  <font face = "WildWest" size =
"5">WildWest</font><br />
  <font face = "Bedrock" size =
"5">Bedrock</font><br />
</body>

</html>
```

This will produce the following result –

Specify alternate font faces

A visitor will only be able to see your font if they have that font installed on their computer. So, it is possible to specify two or more font face alternatives by listing the font face names, separated by a comma.

```
<font face = "arial,helvetica">
<font face = "Lucida
Calligraphy,Comic Sans MS,Lucida
Console">
```

When your page is loaded, their browser will display the first font face available. If none of the given fonts are installed, then it will display the default font face *Times New Roman*.

Note – Check a complete list of **HTML Standard Fonts**.