IITM GROUP OF INSTITUITIONS MURTHAL, SONIPAT

BCA-206B: INTRODUCTION TO INTERNET &WEB DESIGNING

BACHELOR OF COMPUTER APPLICATION (BCA)

SEMESTER: IV

UNIT:1(INTERNET BASICS)

INTRODUCTION TO INTERNET

The Internet is a worldwide network of computer networks that connects university, government, commercial, and other computers in over 150 countries. There are thousands of networks, tens of thousands of computers, and millions of users on the Internet, with the numbers expanding daily. Using the Internet, you can send electronic mail, chat with colleagues around the world, and obtain information on a wide variety of subjects.

Three principal uses of the Internet are:

- **Electronic mail**. Electronic mail, or e-mail, lets you electronically "mail" messages to users who have Internet E-mail addresses. Delivery time varies, but it's possible to send mail across the globe and get a response in minutes. LISTSERVs are special interest mailing lists which allow for the exchange of information between large numbers of people.
- **USENET newsgroups**. USENET is a system of special interest discussion groups, called newsgroups, to which readers can send, or "post" messages which are then distributed to other computers in the network. (Think of it as a giant set of electronic bulletin boards.) Newsgroups are organized around specific topics, for example, **alt. education.research, alt.education.distance, and misc.education.science**.
- **Information files**. Government agencies, schools, and universities, commercial firms, interest groups, and private individuals place a variety of information on-line. The files were originally text only, but increasingly contain pictures and sound.

How Do I Explore the Internet?

To access the Internet, you'll need a personal computer, a modem (or direct link to a network), telecommunications software, a telephone line, and an Internet account. Don't worry--this is easier than it sounds, but it still helps when you're getting started to have a few good books on the subject, or better yet, a friend who's an experienced "cyber surfer." Many universities provide Internet accounts to their faculty and students at little or no cost. Commercial vendors will provide Internet service for a fee. Make sure that you access your Internet provider with a local telephone call--otherwise, long distance charges will apply.

The easiest way for new users to navigate the Internet may be through the "gopher," a navigational system that uses a series of menus to organize and provide access to information. Unfortunately, "gopher," while easy to use, provides text-only information. It is much more rewarding to take full advantage of the multimedia opportunities available on the World-Wide Web (WWW). This system organizes information to provide for linkages to related documents (hypertext links), which allow users to move quickly and easily to related documents.

Software such as Mosaic and Netscape give users a graphical interface and (theoretically) allow for effortless "point and click" travel through cyberspace. If you want to use programs such as Mosaic and Netscape, you will need an up-to-date personal computer and a fast modem--the faster the better--but most users find the rewards worth the extra investment.

What Information Is Available?

A world of information awaits potential users, everything from job listings to travel guides to exotic locations to movie reviews to full text versions of many of the classics. On a single session on the Internet, you can

- Take a pictorial tour of Alexandria, the main port of Egypt, and learn the history of the city;
- Visit science museums to learn about the human heart, view the earth from space, or dissect a frog;
- Listen to Bach played on the 10th largest pipe organ in the world; and then
- Visit Paris, where you can check out the Metro, visit the Louvre (yes, you can view the Mona Lisa), and drop into a sidewalk cafe--but you'll need to provide your own espresso!

As these examples show, the Internet contains a wide variety of information, much of it free. Because of copyright issues, the free material is most likely to be material in the public domain, material for which the copyright has expired, or government documents. You won't find a free copy (legally, anyway) of the latest best seller on the Internet, but you will find on-line newspapers, catalogs for mail-order companies, movie review databases, and a wide variety of government publications. Some sites will have both WWW sites and gopher sites. If you have the choice, start with the WWW site--it's usually much more interesting.

Examples of a few of the governmental resources available on the Internet are:

- The **White House** home page, which provides information on the First Family, White House press releases, Presidential addresses, and links to other branches of government. At the White House home page, you can even take a virtual tour of the White House and check out the oval office at http://www.whitehouse.gov!
- The **Library of Congress World Wide Web** site at http://lcweb.loc.gov/ or http://www.loc.gov/ provides information about its collections as well as on-line materials. You can view Civil War photographs, see an image of a draft of the Gettysburg Address, or read the transcripts of interviews from the WPA Federal Writer's Folklore project. The Library of Congress also supports THOMAS, a service which provides full text versions of legislation, House and Senate bills, the

Congressional Record, and daily proceedings of the House and Senate at http://thomas.loc.gov/home/thomas2.html.

- If you don't have WWW access, the **Library of Congress MARVEL** is a gopher-based system that provides links to a large collection of information from the Federal government, including full-text copies of Federal legislation and Congressional publications. LC MARVEL also provides links to State and foreign government gopher sites at gopher://marvel.loc.gov:70/.
- The **U.S. Department of Education WWW Home Page** at http://www.ed.gov, which contains links to key federal education legislation, departmental reports and materials, and information on grant opportunities. While you're at the ED site, be sure to visit the Home Page for the **National Institute on the Education of At-Risk Students**, where you will find information on research activities, grant opportunities, and special projects sponsored by the Institute. Much of the material on the ED WWW is mirrored in a gopher site, so if you don't have WWW access, just gopher to gopher.ed.gov.
- The **Educational Resources Information Center (ERIC)**, a federally funded national information system that provides access to an extensive body of education-related resources.

  The **ERIC Clearinghouse on Information and Technology** (ERIC/IT), sponsor of the AskERIC Project, is one of 16 ERIC Clearinghouses nationwide which provide a variety of services, products, and resources at all education levels. The AskERIC Virtual Library contains selected resources for education and general interest, including lesson plans, ERIC Digests, ERIC publications, Internet guides and directories, and AskERIC InfoGuides at http://ericir.sunsite.syr.edu/.

  The **ERIC Clearinghouse on Urban Education** (ERIC/CUE) provides information, including full-text reports, of special interest to urban educators at http://eric-web.tc.columbia.edu/.

- The **Partnerships Against Violence Network** (PAVNET) is a multi-agency coalition made up of the U.S. Departments of Agriculture, Education, Health and Human Services, Housing and Urban Development, Justice, and Labor. The PAVNET agencies have joined together to improve access to ideas and resources throughout the country.

  The online components of PAVNET include descriptions of promising programs to prevent or combat violence; lists of organizations that can provide assistance such as training, onsite technical assistance, or information over the telephone on violence-related problems; and the names and addresses of government and private agencies which provide funding for violence prevention programs at http://www.usdoj.gov/pavnet.html.

- The **Children Youth and Family Education and Research Network** (CYFERNet) gopher server, which is maintained by U.S. Department of Agriculture's Office of Research, Education, and Extension and the National Agricultural Library's Youth Development Information Center, provides information on child, youth, and family issues at gopher://cyfer.esusda.gov:70/ or gopher://ra.esusda.gov:70:/1/.

The strange looking text we've listed after each site, for example **http://www.ed.gov/** or **gopher.ed.gov**, is the site's "address" on the Internet. With this address, you can connect directly to the location. The address also provides a clue to the location of the site. For instance, sites ending in "gov" are government sites, those ending in "edu" tend to be universities, and ones with "com" at the end are commercial sites.

What Is the Catch?

If the idea of vast quantities of information on-line for free sounds a bit too good to be true--well, you're partly right. While we think that the benefits of using the Internet far outweigh any problems, you should realize that there are a number of traffic jams, roadblocks, potholes, and accidents just waiting to happen on the information superhighway. For example:

- The number of Internet users is increasing faster than the system can absorb them, and during peak periods you may encounter delays due to the volume of calls. If you're having trouble connecting to a site, try again very early in the morning, or very late at night.

- Using the Internet, particularly if you want to install and use the new graphical-interface WWW programs, isn't always easy. However, most commercial Internet providers supply easy-to-use installation software.

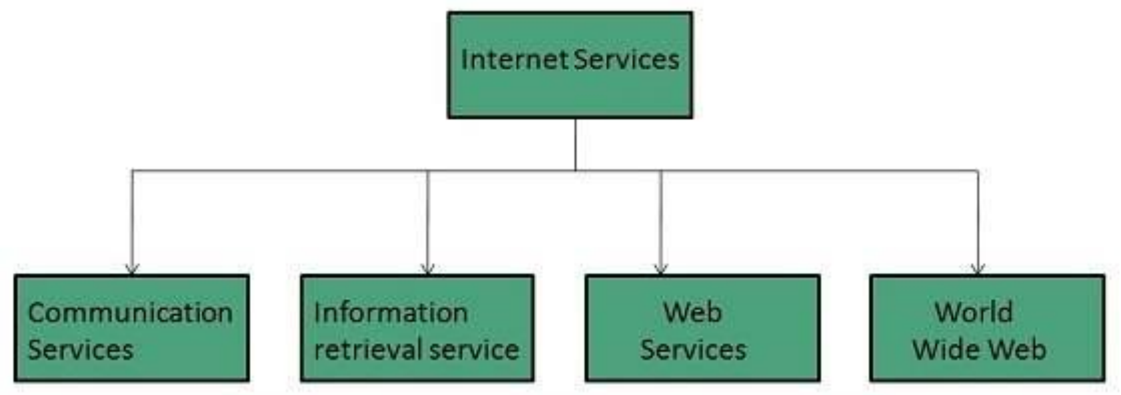
- Navigation through the Internet is not always straightforward. Trails may lead in circles and it can sometimes be frustrating trying to find the site you want. However, a variety of search methods are available on the Internet, you can mark your favorite sites with "bookmarks," and you may find great sites by accident, so don't despair if you get lost!

- While there is a great deal of accurate, useful information available through the Internet, there is also a great deal of inaccurate and totally useless information. Remember, anybody can post anything somewhere on the Internet. Be sure to check the source of the information you acquire.

- If you are providing students with access to the Internet, be aware that there is a great deal of information on the Internet that is totally unsuitable for children. Monitor their access to the system.

  Internet Services

- **Internet Services** allows us to access huge amount of information such as text, graphics, sound and software over the internet. Following diagram shows the four different categories of Internet Services.



- 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:

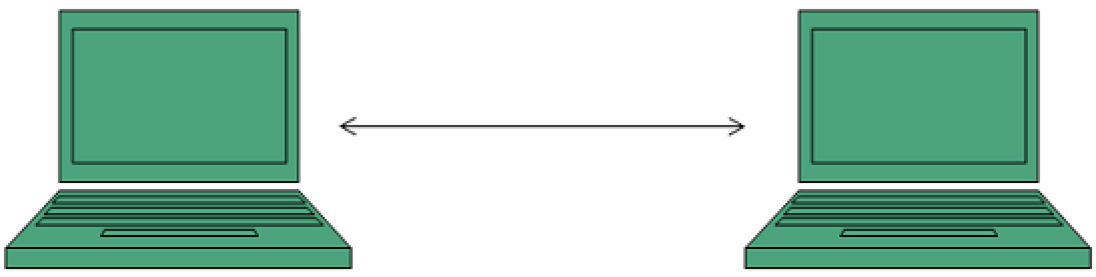
S.N.	Service Description
1	**Electronic Mail** Used to send electronic message over the internet.
2	**Telnet** Used to log on to a remote computer that is attached to internet.
3	**Newsgroup** Offers a forum for people to discuss topics of common interests.
4	**Internet Relay Chat (IRC)** Allows the 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

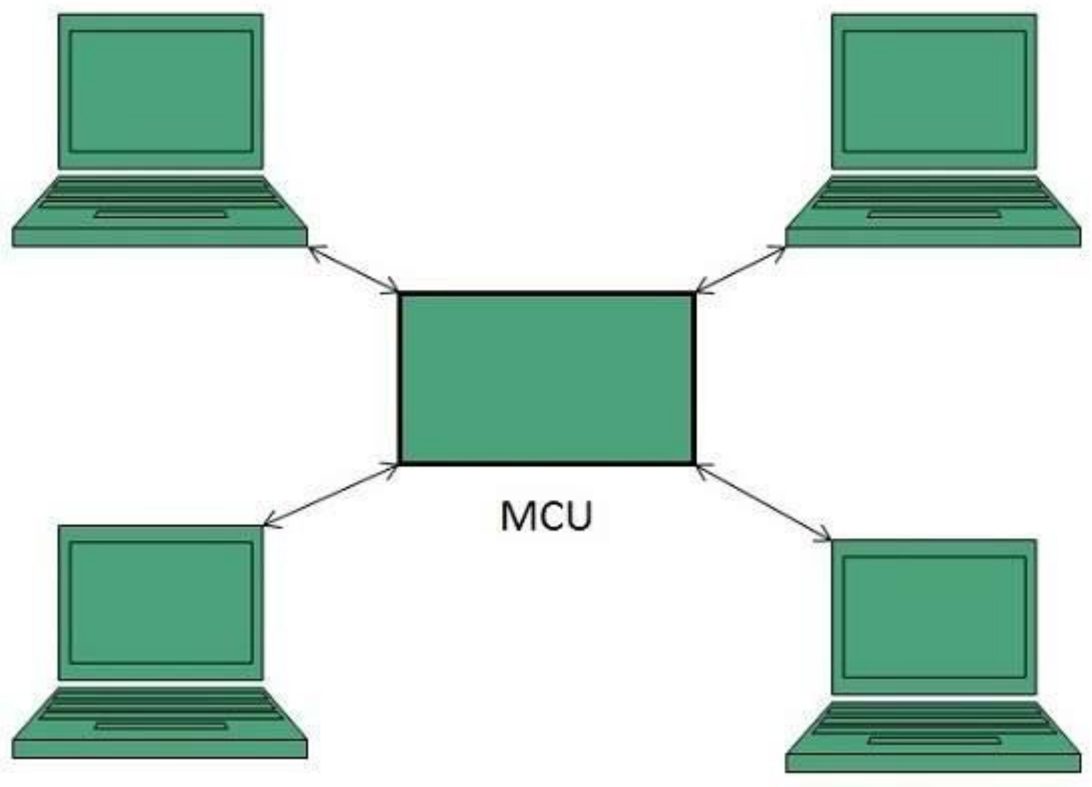
- There exist several Information retrieval services offering easy access to information present on the internet. The following table gives a brief introduction to these services:

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).**



World Wide Web (WWW)

- **World Wide Web (WWW)**, byname **the Web**, the leading information retrieval service of the Internet (the worldwide computer network). The Web gives users access to a vast array of documents that are connected to each other by means of hypertext or hypermedia links—i.e., hyperlinks, electronic connections that link related pieces of information in order to allow a user easy access to them. Hypertext allows the user to select a word or phrase from text and thereby access other documents that contain additional information pertaining to that word or phrase. Hypermedia documents feature links to images, sounds, animations, and movies. The Web operates within the Internet's basic client-server format; servers are computer programs that store and transmit documents to other computers on the network when asked to, while clients are programs that request documents from a server as the user asks for them. Browser software allows users to view the retrieved documents.

- A hypertext document with its corresponding text and hyperlinks is written in HyperText Markup Language (HTML) and is assigned an online address called a Uniform Resource Locator (URL).
- The development of the World Wide Web was begun in 1989 by Tim Berners-Lee and his colleagues at CERN, an international scientific organization based in Geneva, Switzerland. They created a protocol, HyperText Transfer Protocol (HTTP), which standardized communication between servers and clients. Their text-based Web browser was made available for general release in January 1992.

- The World Wide Web gained rapid acceptance with the creation of a Web browser called Mosaic, which was developed in the United States by Marc Andreessen and others at the National Center for Supercomputing Applications at the University of Illinois and was released in September 1993. Mosaic allowed people using the Web to use the same sort of "point-and-click" graphical manipulations that had been available in personal computers for some years. In April 1994 Andreessen cofounded Netscape Communications Corporation, whose Netscape Navigator became the dominant Web browser soon after its release in December 1994. BookLink Technologies' InternetWorks, the first browser with tabs, in which a user could visit another Web site without opening an entirely new window, debuted that same year. By the mid-1990s the World Wide Web had millions of active users.

Apple's Safari was released in 2003 as the default browser on Macintosh personal computers and later on iPhones (2007) and iPads (2010). Safari 2.0 (2005) was the first browser with a privacy mode, Private Browsing, in which the application would not save Web sites in its history, downloaded files in its cache, or personal information entered on Web pages.
The first serious challenger to IE's dominance was Mozilla's Firefox, released in 2004 and designed to address issues with speed and security that had plagued IE. In 2008 Google launched Chrome, the first browser with isolated tabs, which meant that when one tab crashed, other tabs and the whole browser would still function. By 2013 Chrome had become the dominant browser, surpassing IE and Firefox in popularity. Microsoft discontinued IE and replaced it with Edge in 2015.
In the early 21st century, smartphones became more computer-like, and more-advanced services, such as Internet access, became possible. Web usage on smartphones steadily increased, and in 2016 it accounted for more than half of Web browsing.

WORKING OF INTERNET: -

How does the Internet Work?

The Internet works through a packet routing network in accordance with the *Internet Protocol (IP)*, the *Transport Control Protocol* (TCP) and other protocols.

What's a protocol?

A protocol is a set of rules specifying how computers should communicate with each other over a network. For example, the *Transport Control Protocol* has a rule that if one computer sends data to another computer, the destination computer should let the source computer know if any data was missing so the source computer can re-send it. Or the *Internet Protocol* which specifies how computers should route information to other computers by attaching addresses onto the data it sends.

What's a packet?

Data sent across the Internet is called a *message*. Before a *message* is sent, it is first split in many fragments called *packets*. These *packets* are sent independently of each other. The typical maximum *packet* size is between 1000 and 3000 characters. The *Internet Protocol* specifies how messages should be packetized.

What's a packet routing network?

It is a network that routes *packets* from a source computer to a destination computer. The Internet is made up of a massive network of specialized computers called *routers*. Each *router's* job is to know how to move *packets* along from their source to their destination. A *packet* will have moved through multiple *routers* during its journey.

When a *packet* moves from one *router* to the next, it's called a *hop*. You can use the command line-tool traceroute to see the list of hops packets take between you and a host.

The *Internet Protocol* specifies how network *addresses* should be attached to the *packet's headers*, a designated space in the *packet* containing its meta-data. The *Internet Protocol* also specifies how the *routers* should forward the *packets* based on the *address* in the *header*.

Where did these Internet routers come from? Who owns them?

These *routers* originated in the 1960s as *ARPANET*, a military project whose goal was a computer network that was decentralized so the government could access and distribute information in the case of a catastrophic event. Since then, a number of *Internet Service Providers* (ISP) corporations have added *routers* onto these *ARPANET routers*.

There is no single owner of these Internet *routers*, but rather multiple owners: The government agencies and universities associated with *ARPANET* in the early days and *ISP* corporations like AT&T and Verizon later on.

Asking who owns the Internet is like asking who owns all the telephone lines. No one entity owns them all; many different entities own parts of them.

Do the packets always arrive in order? If not, how is the message re-assembled?

The *packets* may arrive at their destination out of order. This happens when a later *packet* finds a quicker path to the destination than an earlier one. But *packet's header* contains information about the *packet's* order relative to the entire *message*. The *Transport Control Protocol* uses this info for reconstructing the message at the destination.

Do packets always make it to their destination?

The *Internet Protocol* makes no guarantee that *packets* will always arrive at their destinations. When that happens, it's called called a *packet loss*. This typically happens when a *router* receives more *packets* it can process. It has no option other than to drop some *packets*.

However, the *Transport Control Protocol* handles *packet loss* by performing re-transmissions. It does this by having the destination computer periodically send acknowledgement *packets* back to the source computer indicating how much of the message it has received and reconstructed. If the destination computer finds there are missing *packets*, it sends a request to the source computer asking it to resend the missing *packets*.

When two computers are communicating through the *Transport Control Protocol*, we say there is a *TCP connection* between them.

What do these Internet addresses look like?

These *addresses* are called *IP addresses* and there are two standards.

The first address standard is called *IPv4* and it looks like 212.78.1.25 . But because *IPv4* supports only 2^{32} (about 4 billion) possible addresses, the *Internet Task Force* proposed a new address standard called *IPv6*, which look like 3ffe:1893:3452:4:345:f345:f345:42fc . *IPv6* supports 2^{128} possible addresses, allowing for much more networked devices, which will be plenty more than the as of 2017 current 8+ billion networked devices on the Internet.

As such, there is a one-to-one mapping between *IPv4* and *IPv6* addresses. Note the switch from *IPv4* to *IPv6* is still in progress and will take a long time. As of 2014, Google revealed their *IPv6* traffic was only at 3%.

How can there be over 8 billion networked devices on the Internet if there are only about 4 billion IPv4 addresses?

It's because there are *public* and *private IP addresses*. Multiple devices on a local network connected to the Internet will share the same *public IP address*. Within the local network, these devices are differentiated from each other by *private IP addresses*, typically of the form 192.168.xx or 172.16.x.x or 10.x.x.x where x is a number between 1 and 255. These *private IP addresses* are assigned by *Dynamic Host Configuration Protocol (DHCP)*.

For example, if a laptop and a smart phone on the same local network both make a request to www.google.com, before the *packets* leave the modem, it modifies the *packet headers* and assigns one of its ports to that *packet*. When the google server responds to the requests, it sends data back to the modem at this specific port, so the modem will know whether to route the *packets* to the laptop or the smart phone.

In this sense, *IP addresses* aren't specific to a computer, but more the connection which the computer connects to the Internet with. The address that is unique to your computer is the *MAC address*, which never changes throughout the life of the computer.

This protocol of mapping *private IP addresses* to *public IP addresses* is called the *Network Address Translation* (NAT) protocol. It’s what makes it possible to support 8+ billion networked devices with only 4 billion possible *IPv4* addresses.

How does the router know where to send a packet? Does it need to know where all the IP addresses are on the Internet?

Every *router* does not need to know where every *IP address* is. It only needs to know which one of its neighbors, called an *outbound link,* to route each packet to. Note that *IP Addresses* can be broken down into two parts, a *network prefix* and a *host identifier*. For example, 129.42.13.69 can be broken down into
Network Prefix: 129.42
Host Identifier: 13.69

All networked devices that connect to the Internet through a single connection (ie. college campus, a business, or ISP in metro area) will all share the same *network prefix*.

Routers will send all packets of the form 129.42.*.* to the same location. So instead of keeping track of billions of *IP addresses, routers* only need to keep track of less than a million *network prefix*.

But a router still needs to know a lot of network prefixes . If a new router is added to the Internet how does it know how to handle packets for all these network prefixes?

A new *router* may come with a few preconfigured routes. But if it encounters a *packet* it does not know how to route, it queries one of its neighboring *routers*. If the neighbor knows how to route the *packet*, it sends that info back to the requesting *router*. The requesting *router* will save this info for future use. In this way, a new router builds up its own *routing table*, a database of *network prefixes* to *outbound links*. If the neighboring *router* does not know, it queries its neighbors and so on.

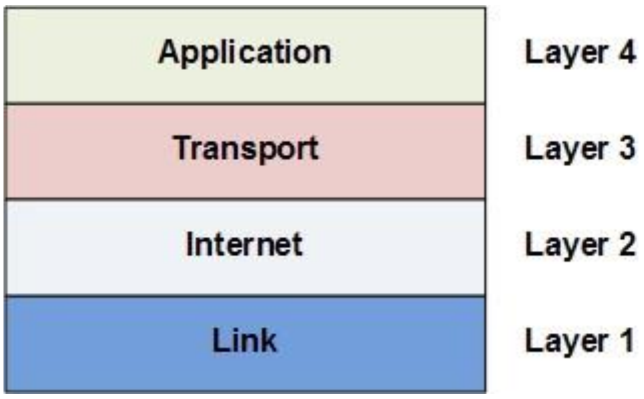
How do networked computers figure out ip addresses based on domain names?

We call looking up the *IP address* of a human-readable domain name like www.google.com “resolving the IP address”. Computers resolve IP addresses through the *Domain Name System* (*DNS*), a decentralized database of mappings from *domain names* to *IP addresses*.

To resolve an IP address, the computer first checks its local *DNS* cache, which stores the *IP address* of web sites it has visited recently. If it can’t find the *IP address* there or that *IP address* record has expired, it queries the *ISP’s DNS* servers which are dedicated to resolving IP addresses. If the *ISP’s DNS* servers can’t find resolve the *IP address*, they query the *root name servers*, which can resolve every domain name for a given *top-level domain* . *Top-level domains* are the words to the right of the right-most period in a domain name. .com .net .org are some examples of *top-level domains*.

How do applications communicate over the Internet?

Like many other complex engineering projects, the Internet is broken down into smaller independent components, which work together through well-defined interfaces. These components are called the *Internet Network Layers* and they consist of *Link Layer*, *Internet Layer*, *Transport Layer*, and *Application Layer*. These are called layers because they are built on top of each other; each layer uses the capabilities of the layers beneath it without worrying about its implementation details.



Internet applications work at the *Application Layer* and don't need to worry about the details in the underlying layers. For example, an application connects to another application on the network via TCP using a construct called a *socket*, which abstracts away the gritty details of routing *packets* and re-assembling *packets* into *messages*.

What do each of these Internet layers do?

At the lowest level is the *Link Layer* which is the "physical layer" of the Internet. The *Link Layer* is concerned with transmitting data bits through some physical medium like fiber-optic cables or wifi radio signals.

On top of the *Link Layer* is the *Internet Layer*. The *Internet Layer* is concerned with routing packets to their destinations. The *Internet Protocol* mentioned earlier lives in this layer (hence the same name). The *Internet Protocol* dynamically adjusts and reroutes *packets* based on network load or outages. Note it does not guarantee *packets* always make it to their destination, it just tries the best it can.

On top of the *Internet Layer* is the *Transport Layer*. This layer is to compensate for the fact that data can be loss in the *Internet* and *Link* layers below. The *Transport Control Protocol* mentioned earlier lives at this layer, and it works primarily to re-assembly packets into their original *messages* and also re-transmit *packets* that were loss.

The *Application Layer* sits on top. This layer uses all the layers below to handle the complex details of moving the packets across the Internet. It lets applications easily make connections with other applications on the Internet with simple abstractions like sockets. The HTTP protocol which specifies how web browsers and web servers should interact lives in the *Application Layer*. The IMAP protocol which specifies how email clients should retrieve email lives in the *Application Layer*. The FTP protocol which specifies a file-transferring protocol between file-downloading clients and file-hosting servers lives in the *Application Layer*.

INTERNET CONNECTION CONCEPTS

Here in this tutorial, we will discuss how to connect to internet i.e. internet service providers, software and hardware requirements, configuring internet connection etc.

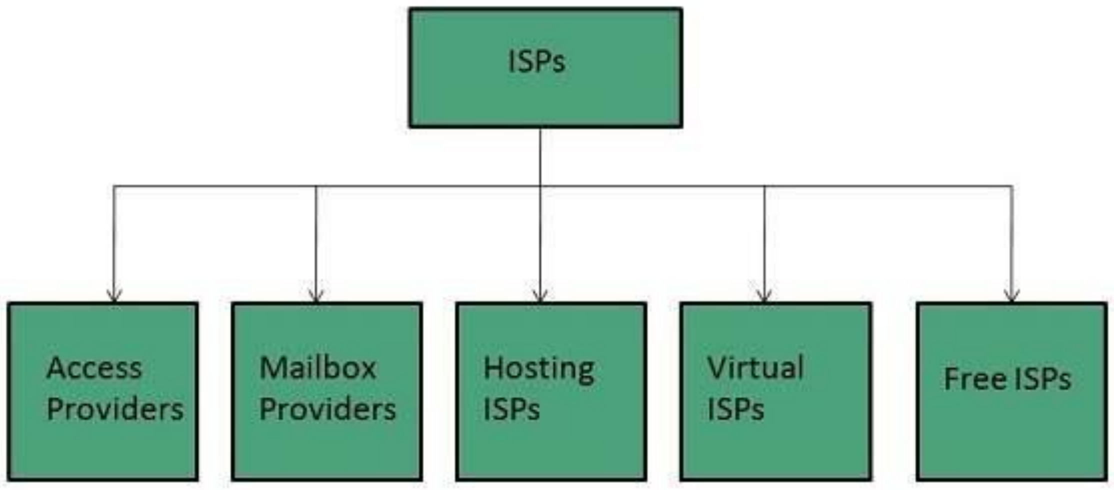
Internet Service Providers (ISP)

Internet Service Provider (ISP) is a company offering access to internet. They offer various services:

- Internet Access
- Domain name registration
- Dial-up access
- Leased line access

ISP Types

ISPs can broadly be classified into six categories as shown in the following diagram:



Access providers

They provide access to internet through telephone lines, cable wi-fi or fiber optics.

Mailbox Provider

Such providers offer mailbox hosting services.

Hosting ISPs

Hosting ISPs offers e-mail, and other web hosting services such as virtual machines, clouds etc.

Virtual ISPs

Such ISPs offer internet access via other ISP services.

Free ISPs

Free ISPs do not charge for internet services.

Connection Types

There exist several ways to connect to the internet. Following are these connection types available:

1. Dial-up Connection
2. ISDN
3. DSL
4. Cable TV Internet connections
5. Satellite Internet connections
6. Wireless Internet Connections

Dial-up Connection

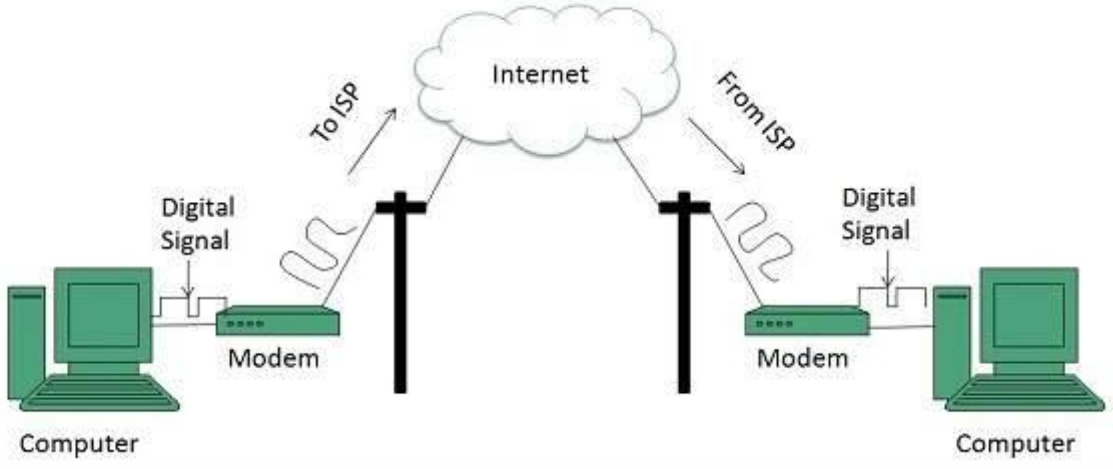
Dial-up connection uses telephone line to connect PC to the internet. It requires a modem to setup dial-up connection. This modem works as an interface between PC and the telephone line.

There is also a communication program that instructs the modem to make a call to specific number provided by an ISP.

Dial-up connection uses either of the following protocols:

1. Serial Line Internet Protocol (SLIP)
2. Point to Point Protocol (PPP)

The following diagram shows the accessing internet using modem:



ISDN:

ISDN is acronym of **Integrated Services Digital Network.** It establishes the connection using the phone lines which carry digital signals instead of analog signals.

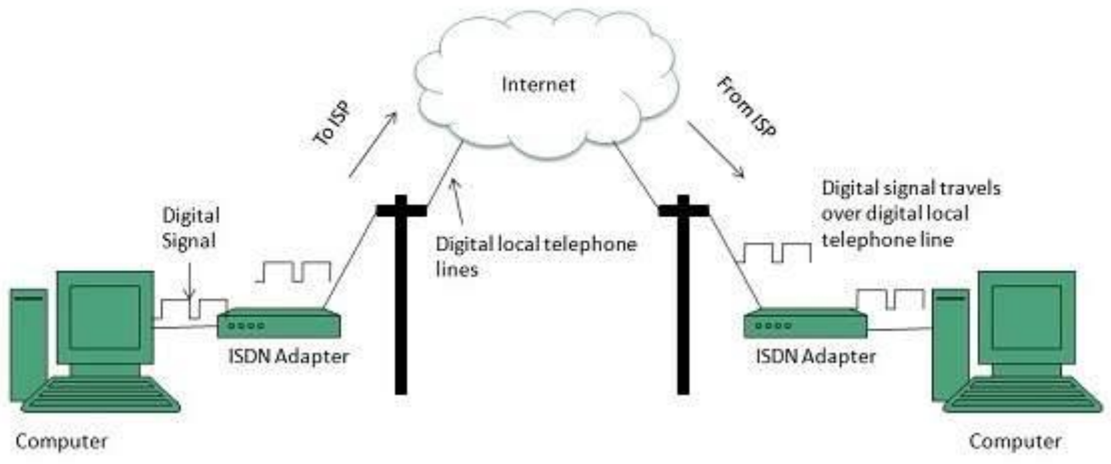
There are two techniques to deliver ISDN services:

1. Basic Rate Interface (BRI)
2. Primary Rate Interface (PRI)

Key points:

- The BRI ISDN consists of three distinct channels on a single ISDN line: t1o 64kbps B (Bearer) channel and one 16kbps D (Delta or Data) channels.
- The PRI ISDN consists of 23 B channels and one D channels with both have operating capacity of 64kbps individually making a total transmission rate of 1.54Mbps.

The following diagram shows accessing internet using ISDN connection:


Internet
To ISP
From ISP
Digital Signal
Digital local telephone lines
Digital signal travels over digital local telephone line
ISDN Adapter
ISDN Adapter
Computer
Computer

DSL:

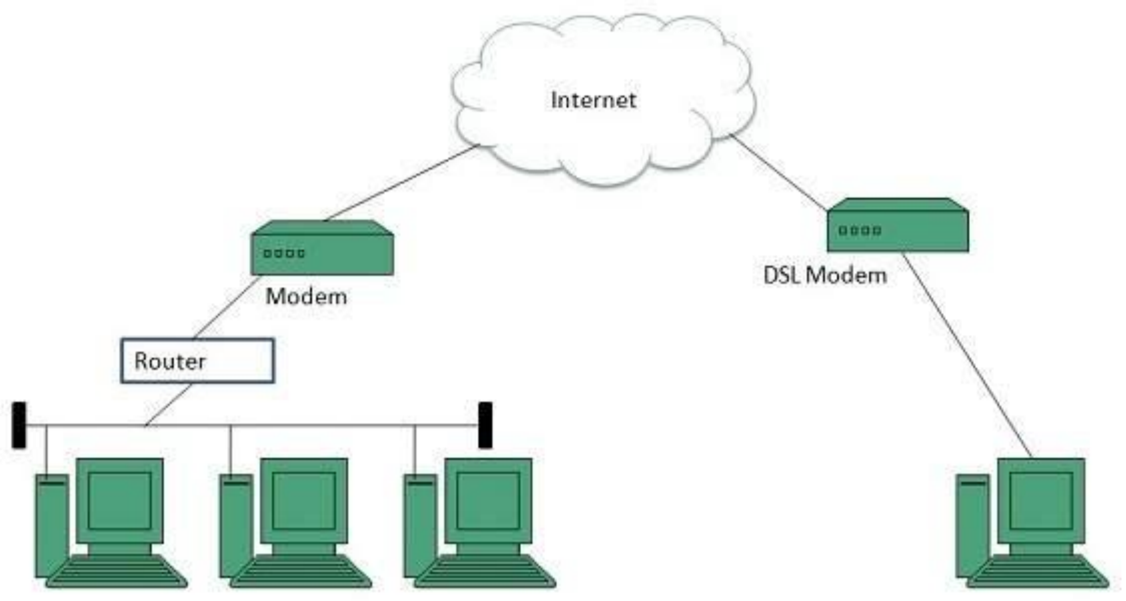
DSL is acronym of **Digital Subscriber Line.** It is a form of broadband connection as it provides connection over ordinary telephone lines.

Following are the several versions of DSL technique available today:

1. Asymmetric DSL (ADSL)
2. Symmetric DSL (SDSL)
3. High bit-rate DSL (HDSL)
4. Rate adaptive DSL (RDSL)
5. Very high bit-rate DSL (VDSL)
6. ISDN DSL (IDSL)

All of the above mentioned technologies differ in their upload and download speed, bit transfer rate and level of service.

The following diagram shows that how we can connect to internet using DSL technology:



Internet Access Using DSL Modem

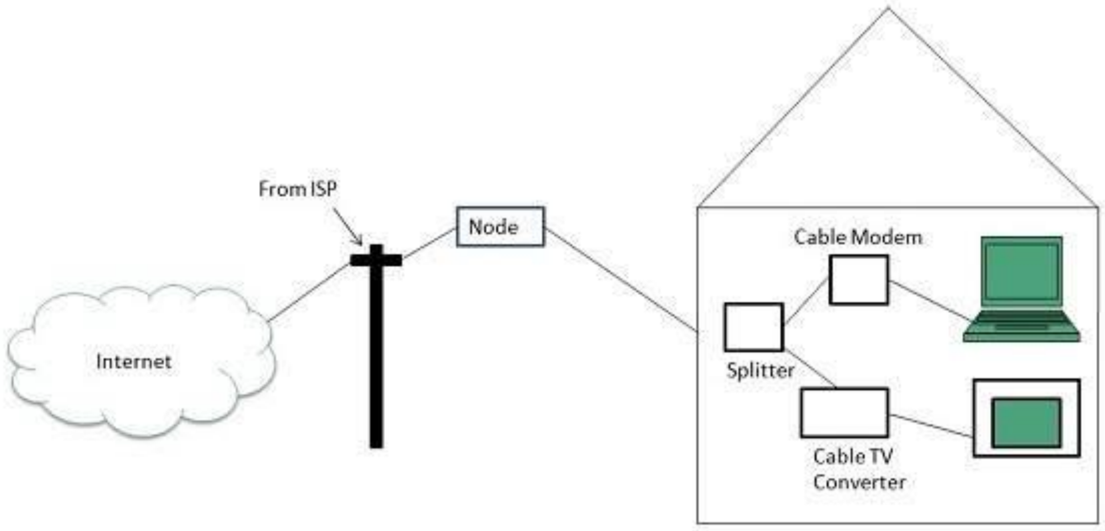
Cable TV Internet Connection

Cable TV Internet connection is provided through Cable TV lines. It uses coaxial cable which is capable of transferring data at much higher speed than common telephone line.

Key Points:

- A cable modem is used to access this service, provided by the cable operator.
- The Cable modem comprises of two connections: one for internet service and other for Cable TV signals.
- Since Cable TV internet connections share a set amount of bandwidth with a group of customers, therefore, data transfer rate also depends on number of customers using the internet at the same time.

The following diagram shows that how internet is accessed using Cable TV connection:



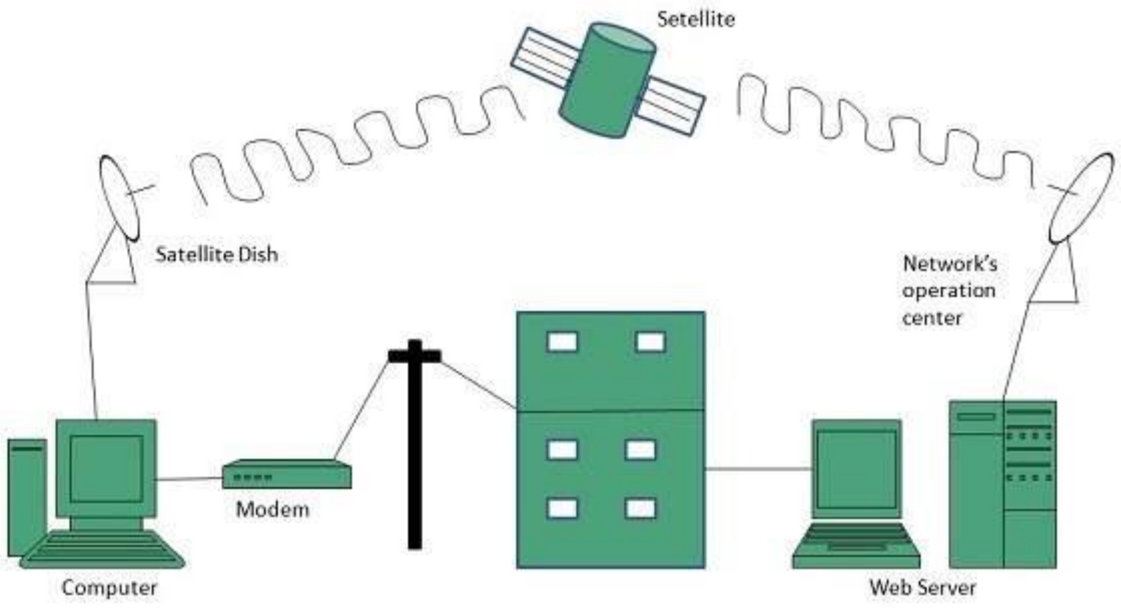
Satellite Internet Connection

Satellite Internet connection offers high speed connection to the internet. There are two types of satellite internet connection: one way connection or two way connection.

In one way connection, we can only download data but if we want to upload, we need a dialup access through ISP over telephone line.

In two way connection, we can download and upload the data by the satellite. It does not require any dialup connection.

The following diagram shows how internet is accessed using satellite internet connection:



Wireless Internet Connection

Wireless Internet Connection makes use of radio frequency bands to connect to the internet and offers a very high speed. The wireless internet connection can be obtained by either WiFi or Bluetooth.

Key Points:

- Wi Fi wireless technology is based on IEEE 802.11 standards which allow the electronic device to connect to the internet.
- Bluetooth wireless technology makes use of short-wavelength radio waves and helps to create personal area network (PAN).

DNS WORKING

The Domain Name System (DNS) is the phonebook of the Internet. Humans access information online through domain names, like nytimes.com or espn.com. Web browsers interact through Internet Protocol (IP) addresses. DNS translates domain names to IP addresses so browsers can load Internet resources.

Each device connected to the Internet has a unique IP address which other machines use to find the device. DNS servers eliminate the need for humans to memorize IP addresses such as 192.168.1.1 (in IPv4), or more complex newer alphanumeric IP addresses such as 2400: cb00:2048:1::c629:d7a2 (in IPv6).

How does DNS work?

The process of DNS resolution involves converting a hostname (such as www.example.com) into a computer-friendly IP address (such as 192.168.1.1). An IP address is given to each device on the Internet, and that address is necessary to find the appropriate Internet device - like a street address is used to find a particular home. When a user wants to load a webpage, a translation must occur between what a user types into their web browser (example.com) and the machine-friendly address necessary to locate the example.com webpage.

In order to understand the process behind the DNS resolution, it’s important to learn about the different hardware components a DNS query must pass between. For the web browser, the DNS lookup occurs “behind the scenes” and requires no interaction from the user’s computer apart from the initial request.

There are 4 DNS servers involved in loading a webpage:

- DNS recursor - The recursor can be thought of as a librarian who is asked to go find a particular book somewhere in a library. The DNS recursor is a server designed to receive queries from client machines through applications such as web browsers. Typically, the recursor is then responsible for making additional requests in order to satisfy the client’s DNS query.
- **Root nameserver** - The root server is the first step in translating (resolving) human readable host names into IP addresses. It can be thought of like an index in a library that points to different racks of books - typically it serves as a reference to other more specific locations.

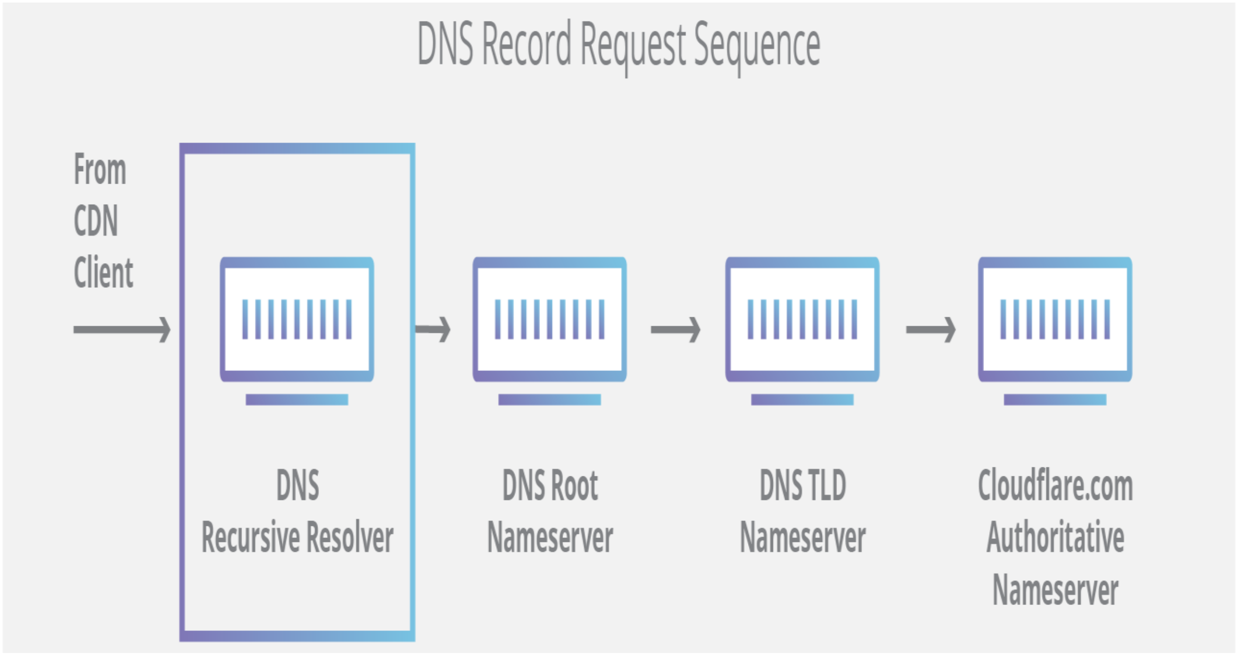
- TLD nameserver - The top-level domain server (TLD) can be thought of as a specific rack of books in a library. This nameserver is the next step in the search for a specific IP address, and it hosts the last portion of a hostname (In example.com, the TLD server is “com”).
- Authoritative nameserver - This final nameserver can be thought of as a dictionary on a rack of books, in which a specific name can be translated into its definition. The authoritative nameserver is the last stop in the nameserver query. If the authoritative name server has access to the requested record, it will return the IP address for the requested hostname back to the DNS Recursor (the librarian) that made the initial request.

What's the difference between an authoritative DNS server and a recursive DNS resolver?

Both concepts refer to servers (groups of servers) that are integral to the DNS infrastructure, but each performs a different role and lives in different locations inside the pipeline of a DNS query. One way to think about the difference is the recursive resolver is at the beginning of the DNS query and the authoritative nameserver is at the end.

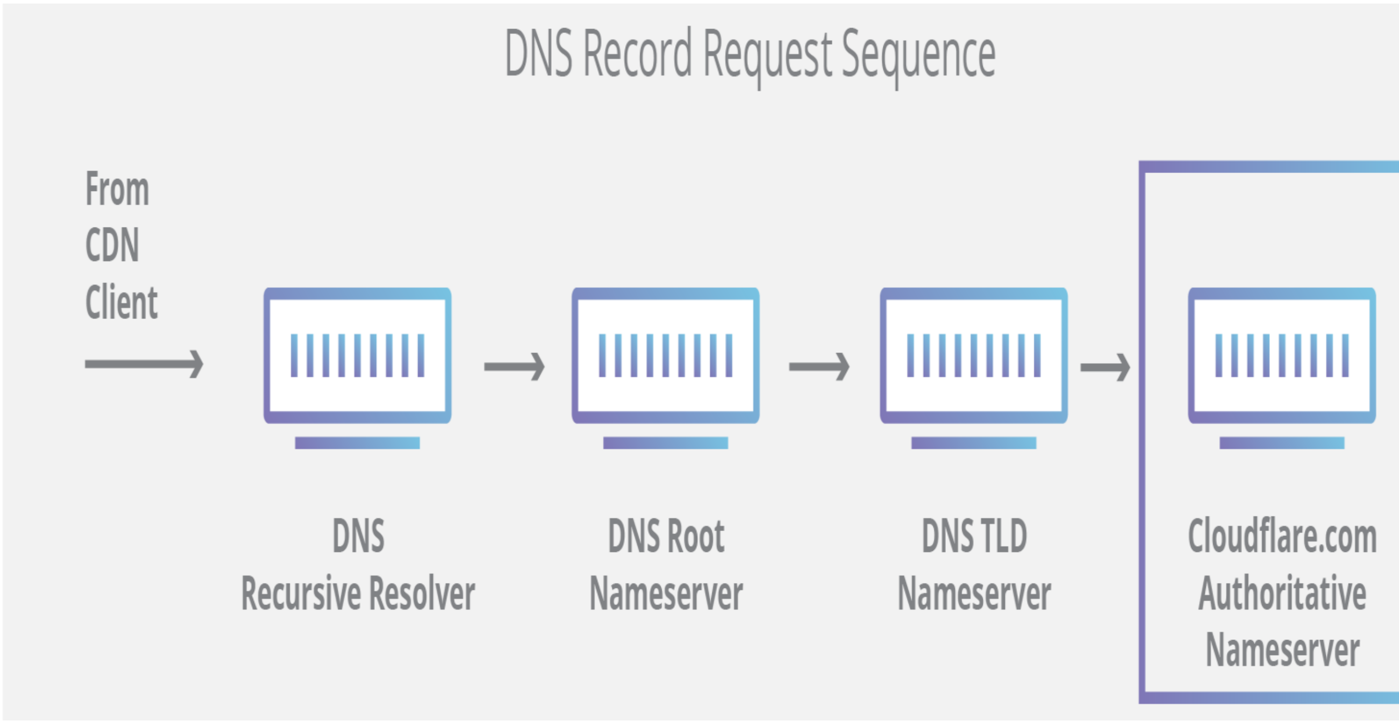
Recursive DNS resolver

The recursive resolver is the computer that responds to a recursive request from a client and takes the time to track down the DNS record. It does this by making a series of requests until it reaches the authoritative DNS nameserver for the requested record (or times out or returns an error if no record is found). Luckily, recursive DNS resolvers do not always need to make multiple requests in order to track down the records needed to respond to a client; caching is a data persistence process that helps short-circuit the necessary requests by serving the requested resource record earlier in the DNS lookup.

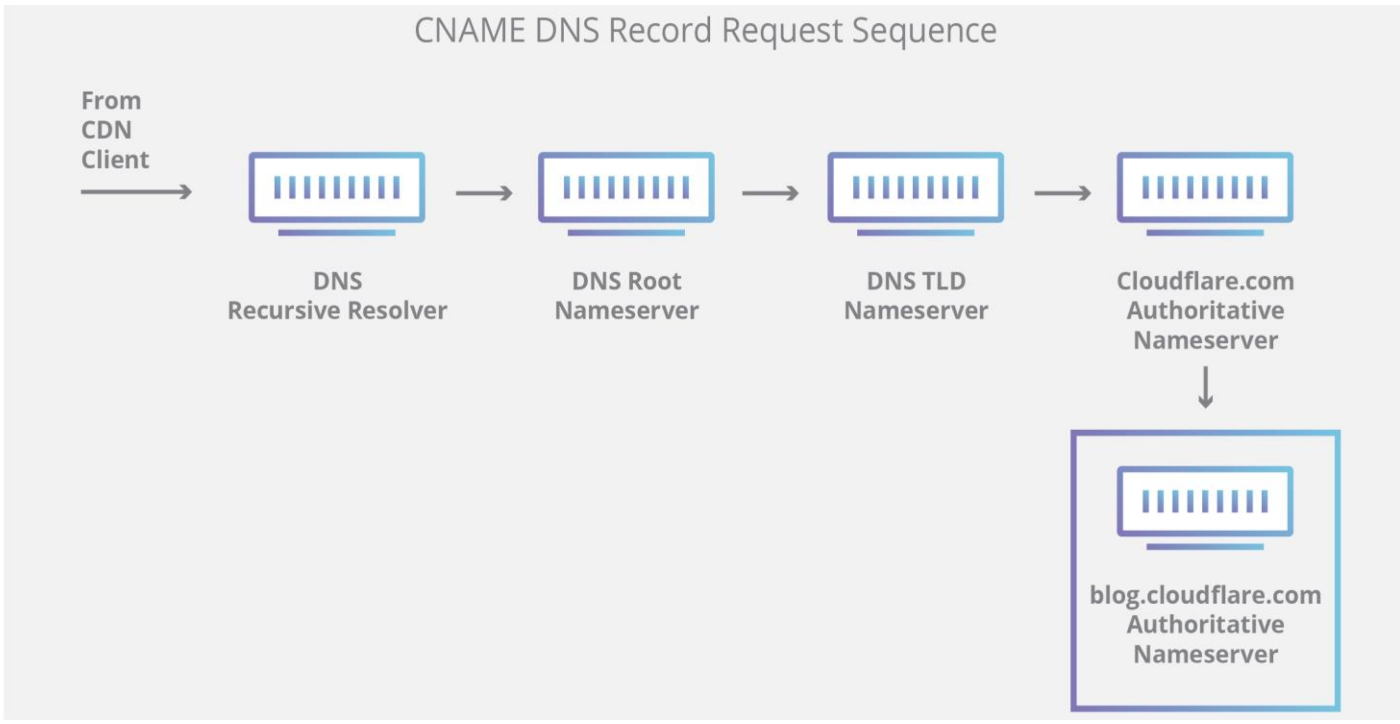


Authoritative DNS server

Put simply, an authoritative DNS server is a server that actually holds, and is responsible for, DNS resource records. This is the server at the bottom of the DNS lookup chain that will respond with the queried resource record, ultimately allowing the web browser making the request to reach the IP address needed to access a website or other web resources. An authoritative nameserver can satisfy queries from its own data without needing to query another source, as it is the final source of truth for certain DNS records.



It's worth mentioning that in instances where the query is for a subdomain such as foo.example.com or blog.cloudflare.com, an additional nameserver will be added to the sequence after the authoritative nameserver, which is responsible for storing the subdomain's CNAME record.


CNAME DNS Record Request Sequence
From CDN Client
DNS Recursive Resolver
DNS Root Nameserver
DNS TLD Nameserver
Cloudflare.com Authoritative Nameserver
blog.cloudflare.com Authoritative Nameserver

There is a key difference between many DNS services and the one that Cloudflare provides. Different DNS recursive resolvers such as Google DNS, OpenDNS, and providers like Comcast all maintain data center installations of DNS recursive resolvers. These resolvers allow for quick and easy queries through optimized clusters of DNS-optimized computer systems, but they are fundamentally different than the nameservers hosted by Cloudflare.

Cloudflare maintains infrastructure-level nameservers that are integral to the functioning of the Internet. One key example is the f-root server network which Cloudflare is partially responsible for hosting. The F-root is one of the root level DNS nameserver infrastructure components responsible for the billions of Internet requests per day. Our Anycast network puts us in a unique position to handle large volumes of DNS traffic without service interruption.

SINGLE USER Vs MULTI USER

What is Single User Operating System?

A single user operating system becomes a mode where the user has a multipurpose computer screen to run the program and the operating system boots into a single super user that controls all the activities. The primary usage for such a system comes whenever the maintenance for several users takes place at the same time on the network servers. Single client mode is a mode in which a multi-user PC is working framework boots into a single super user. It, for the most part, gets utilized for support of multi-client situations, for example, arrange servers. A few assignments may require elite access to shared assets, for instance running program on a system share. This mode can likewise utilize for security purposes – organize administrations are not run, dispose of the likelihood of outside obstruction. On a few frameworks, a lost super user secret key can get changed by changing to single client mode, however not requesting the watchword in such conditions is a security powerlessness. A practical framework can become utilized for an assortment of undertakings, and it's a critical program on a PC. It is intended to sort out memory utilization, equipment network, and serves to legitimately execute applications. Single-undertaking working frameworks can work on electronic gadgets, like a

PC, and will run the just application at once. It can work on remote telephones and two-way informing frameworks.

What is Multi-User Operating System?

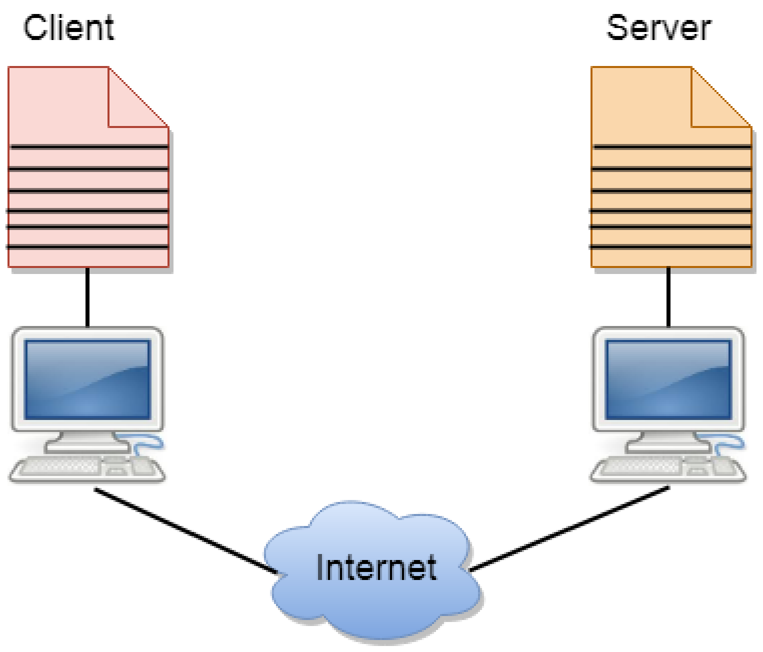
A multi-user operating system becomes a mode where the user has a screen where only one program runs and different people within the environment have access to various devices, and they perform tasks to complete the system. All the programs that have time sharing capacities become such, and most computer systems also have the same name. An illustration is a Unix server where different remote clients approach the Unix shell incite in the meantime. Another case utilizes various X Window sessions spread over multiple terminals controlled by a separate machine – this is an example of the utilization of thin customer. Comparable capacities were additionally accessible under MP/M, Concurrent DOS, Multi-user DOS and FlexOS. Some multi-client working frameworks, for example, Windows renditions from the Windows NT family bolster concurrent access by different clients. For instance, using Remote Desktop Connection and also the capacity for a customer to separate from a neighbourhood session while leaving forms running does take a shot at their sake while another client sign into and utilizes the framework. When alluding to a PC working structure, a multi-client framework is a PC with a practical framework that backings various clients without a moment's delay or peculiar circumstances. These projects are regularly very convoluted and should have the capacity to legitimately deal with the major assignments required by the diverse clients associated with it. The clients will commonly be at terminals or PCs that give them access to the framework through a system, and different machines on the frame, for example, printers aidn't get employed with PCs and gadgets that need various projects running.

	Single User Operating System	Multi-User Operating Syste
Definition	A mode where the user has a multipurpose computer screen to run the program and the operating system boots into a single super user that controls all the activities.	A mode where the user has a sc program runs and different peo environment have access to var
Super User	A super user gets all the powers of maintaining the system and making changes to ensure the system runs smoothly.	Each entity has control over the requirent of super user.
Performance	Only one task at one time gets performed	schedules different tasks for pe rate.

CLIENT-SERVER MODEL:

Client and Server model

- A client and server networking model is a model in which computers such as servers provide the network services to the other computers such as clients to perform a user based tasks. This model is known as client-server networking model.
- The application programs using the client-server model should follow the given below strategies:



- An application program is known as a client program, running on the local machine that requests for a service from an application program known as a server program, running on the remote machine.
- A client program runs only when it requests for a service from the server while the server program runs all time as it does not know when its service is required.
- A server provides a service for many clients not just for a single client. Therefore, we can say that client-server follows the many-to-one relationship. Many clients can use the service of one server.
- Services are required frequently, and many users have a specific client-server application program. For example, the client-server application program allows the user to access the files, send e-mail, and so on. If the services are more customized, then we should have one generic application program that allows the user to access the services available on the remote computer.

Client

A client is a program that runs on the local machine requesting service from the server. A client program is a finite program means that the service started by the user and terminates when the service is completed.

Server

A server is a program that runs on the remote machine providing services to the clients. When the client requests for a service, then the server opens the door for the incoming requests, but it never initiates the service.

A server program is an infinite program means that when it starts, it runs infinitely unless the problem arises. The server waits for the incoming requests from the clients. When the request arrives at the server, then it responds to the request.

Advantages of Client-server networks:

- **Centralized:** Centralized back-up is possible in client-server networks, i.e., all the data is stored in a server.
- **Security:** These networks are more secure as all the shared resources are centrally administered.
- **Performance:** The use of the dedicated server increases the speed of sharing resources. This increases the performance of the overall system.
- **Scalability:** We can increase the number of clients and servers separately, i.e., the new element can be added, or we can add a new node in a network at any time.

Disadvantages of Client-Server network:

- **Traffic Congestion** is a big problem in Client/Server networks. When a large number of clients send requests to the same server may cause the problem of Traffic congestion.
- It does not have a robustness of a network, i.e., when the server is down, then the client requests cannot be met.
- A client/server network is very decisive. Sometimes, regular computer hardware does not serve a certain number of clients. In such situations, specific hardware is required at the server side to complete the work.
- Sometimes the resources exist in the server but may not exist in the client. For example, If the application is web, then we cannot take the print out directly on printers without taking out the print view window on the web.

COMPUTER NETWORK:

A computer network is a group of computer systems and other computing hardware devices that are linked together through communication channels to facilitate communication and resource-sharing among a wide range of users.

Networks are commonly categorized based on their characteristics. One of the earliest examples of a computer network was a network of communicating computers that functioned as part of the U.S. military's Semi-Automatic Ground Environment (SAGE) radar system. In 1969, the University of California at Los Angeles, the Stanford Research Institute, the University of California at Santa Barbara and the University of Utah were connected as part of the Advanced Research Projects Agency Network (ARPANET) project.

It is this network that evolved to become what we now call the internet.

Networks are used to:

- Facilitate communication via email, video conferencing, instant messaging, etc.
- Enable multiple users to share a single hardware device like a printer or scanner.
- Enable file sharing across the network.
- Allow for the sharing of software or operating programs on remote systems.
- Make information easier to access and maintain among network users.

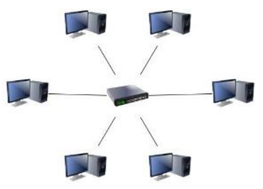
There are many types of networks, including:

- Local Area Networks (LAN).
- Global Area Networks (GAN).
- Personal Area Networks (PAN).
- Home Area Networks (HAN).
- Wide Area Networks (WAN).
- Campus Networks.
- Metropolitan Area Networks (MAN).
- Enterprise Private Networks.
- Internetworks.
- Backbone Networks (BBN).
- Global Area Networks (GAN).
- The internet.

COMPUTER NETWORK (LAN, WAN, MAN):

A ***local-area network (LAN)*** is a computer network that spans a relatively small area. Most often, a LAN is confined to a single room, building or group of buildings, however, one LAN can be connected to other LANs over any distance via telephone lines and radio waves.
A system of LANs connected in this way is called a wide-area network (WAN). The difference between a LAN and WAN is that the wide-area network spans a relatively large geographical area. Typically, a WAN consists of two or more local-area networks (LANs) and are often connected through public networks.

Nodes on a LAN



Most LANs connect workstations and personal computers. Each node (individual computer) in a LAN has its own CPU with which it executes programs, but it also is able to access data and devices anywhere on the LAN. This means that many users can share expensive devices, such as laser printers, as well as data. Users can also use the LAN to communicate with each other, by sending email or engaging in chat sessions.
LANs are capable of transmitting data at very fast rates, much faster than data can be transmitted over a telephone line; but the distances are limited and there is also a limit on the number of computers that can be attached to a single LAN.

Types of Local-Area Networks (LANs)

There are many different types of LANs, with Ethernets being the most common for PCs. Most Apple Macintosh networks are based on Apple's AppleTalk network system, which is built into Macintosh computers. The following characteristics differentiate one LAN from another:

- **Topology:** The geometric arrangement of devices on the network. For example, devices can be arranged in a ring or in a straight line.
- **Protocols:** The rules and encoding specifications for sending data. The protocols also determine whether the network uses a peer-to-peer or client/server architecture.
- **Media:** Devices can be connected by twisted-pair wire, coaxial cables, or fiber optic cables. Some networks do without connecting media altogether, communicating instead via radio waves.

Deploying a Wireless LAN

Wireless networks are relatively easy to implement these days, especially when compared to the prospect of having to route wires when deploying a new wired network or overhauling an existing one. The first step in planning a wireless LAN deployment should be to decide on your wireless networking technology standard. Keep in mind that the standard you need to accommodate your network access points and routers as well as the entire collection of wireless network interface cards (NICs) for your computers and other network resources.

ADVANTAGES OF LAN :

1.RESOURCE SHARING:
Computer hardware resources like printers, modems, DVD-Rom drives and hard disks can be shared with the help of local area networks. This will reduce cost of hardware purchases. For example, a business organization using a Local Area Network for an office can use a single network printer for the employees of this office.
2. Software Applications Sharing
It is cheaper to use same software over network instead of purchasing separate licensed software for each client in a network. It will cost more to purchase a separate licensed software for each computer in a network.
3. Easy and Cheap Communication
Data and messages can easily be transferred over networked computers. It saves a lot of time and money.
4. Centralized Data
The data of all network users can be saved on hard disk of the server computer. This will help users to use any workstation in a network to access their data. Because data is not stored on workstations locally. But it is stored on a server computer. User will access their own data by logging into their accounts from any client computer in the network.
5. Data Security
Since, data is stored on server computer centrally, it will be easy to manage data at only one place and the data will be more secure too, because of more security for the server computer.

Local Area Network provides the facility to share a single internet connection among all the LAN users. In Net Cafes, single internet connection sharing system keeps the internet expenses cheaper. Because only one high speed internet connection is purchased by a net cafe from any ISP - Internet Service Provider company.
This single high-speed internet connection is managed on the server computer of the Net Cafe and available for all network clients with the help of Internet Connection Sharing facilities of the operating system.

DISADVANTAGES OF LAN:

1.High Setup Cost:

Although the LAN will save cost over time due to shared computer resources but the initial setup costs of installing Local Area Networks is high. This is because any organization that will setup a network, will have to purchase necessary hardware equipment for networking. It may require a sophisticated server computer - a Mini Computer, Network LAN cards, Network Routers, HUBS / Switches, Networking Cables (for wired networks only) and connectors etc.

If an organization has a large network, it must hire a network administrator for smooth running of network and solving any problems.

2. Privacy Violations

The LAN administrator has the rights to check personal data files of each and every LAN user. Moreover, he can check the internet history and computer use history of the LAN users.

3. Data Security Threat

Unauthorized users can access important data of an organization if centralized data repository is not secured properly by the LAN administrator. LAN Administer is responsible for the security of the whole data resource in an organization.

LAN administrator will implement a fully functional security policy for database safety.

4. LAN Maintenance Job

Local Area Network requires a LAN Administrator. Because, there are problems of software installations or hardware failures or cable disturbances in Local Area Network. A LAN Administrator is needed at this full-time job. A LAN Administrator may be with an M.C.S. or B.S.C.S. degree holder person optionally with a course / diploma in networking field or with relative experience in the field.

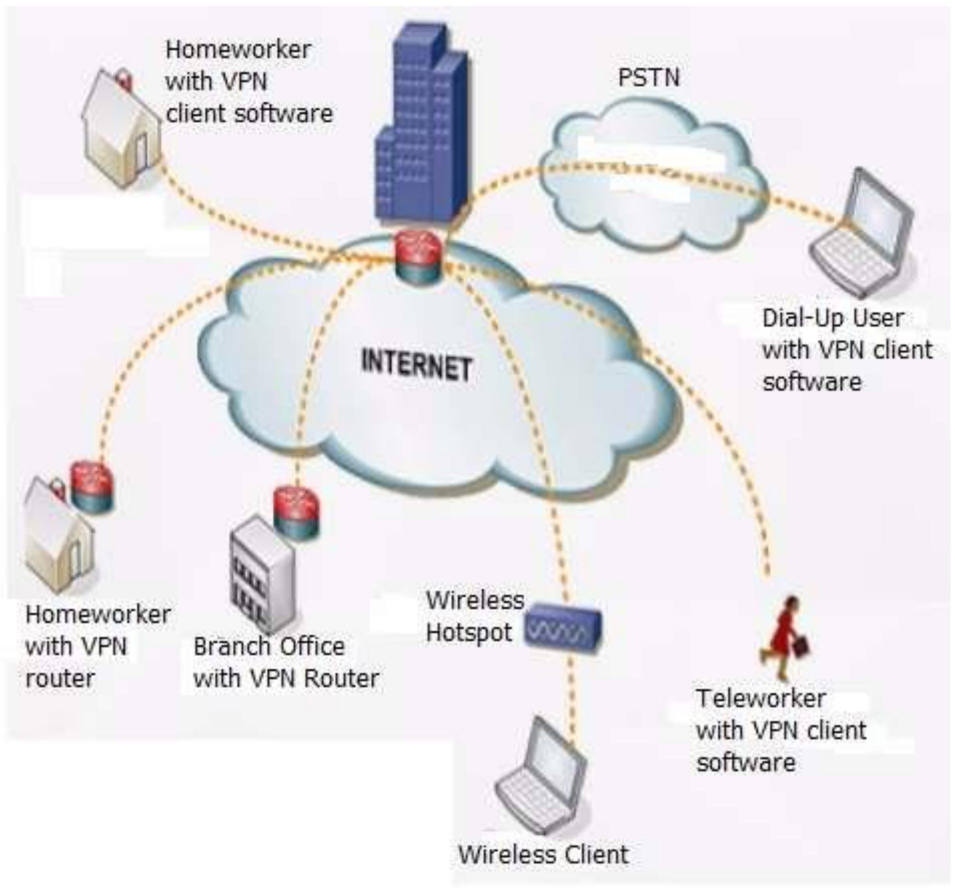
5. Covers Limited Area

Local Area Network covers a small area like one office, one building or a group of nearby buildings.

What is WAN?

Introduction:

The WAN is a network which is a collection of LANs and other network types connected using router. It covers large geographical distance compare to LAN and MAN types.


Homeworker with VPN client software
PSTN
INTERNET
Dial-Up User with VPN client software
Homeworker with VPN router
Branch Office with VPN Router
Wireless Hotspot
Teleworker with VPN client software
Wireless Client

There many types of WAN (Wide Area Network) viz. private network, value added service, PSDN, PSTN, ISDN etc. Many different techniques are used in WAN viz. wireless WAN, cell relay, circuit switching, packet switching, leased line etc. Based on distance covered, the computer networks are divided into three main parts viz. LAN, MAN and WAN. Among these types, WAN has largest coverage of about 1000Km or unlimited based on WAN technologies. The data rate depends on wireless technologies or standards viz. wifi, zigbee, LoRa, Satellite etc. Refer difference between LAN vs WAN vs MAN>>.

Benefits or advantages of WAN

Following are the benefits or **advantages of WAN**:

➡WAN covers larger geographical area. Hence business offices situated at longer distances can easily communicate.

➡Like LAN, it allows sharing of resources and application softwares among distributed workstations or users.

➡The software files are shared among all the users. Hence all will have access to latest files. This avoids use of previous versions by them.

➡Organizations can form their global integrated network through WAN. Moreover, it supports global markets and global businesses.

➡The emergence of IoT (Internet of Things) and advanced wireless technologies such as LAN or LAN-Advanced have made it easy for the growth of WAN based devices. Messages can be sent very quickly across the globe with the help of applications such as whatsApp, facebook messenger etc.

Drawbacks or disadvantages of WAN

Following are the **disadvantages of WAN**:

➡Initial investment costs are higher.

➡It is difficult to maintain the network. It requires skilled technicians and network administrators.

➡There are more errors and issues due to wide coverage and use of different technologies. Often it requires more time to resolve issues due to involvement of multiple wired and wireless technologies.

➡It has lower security compare to LAN and MAN due to wider coverage and use of more technologies.

➡Security is big concern and requires use of firewall and security softwares/protocols at multiple points across the entire system. This will avoid chances of hacking by intruders.

What is a metropolitan area network (MAN)?

A metropolitan area network (MAN) is a network which covers a city or a large university campus. MAN connect users within an area larger than local area network (LAN) but smaller than a wide area network (WAN). MAN is made with a combination of several LAN through point to point connections. MAN, usually involves connection of fiber optics wires to boost data transfer speed.

MAN depends upon different technologies to establish a network. Some of these technologies are mentioned below: -

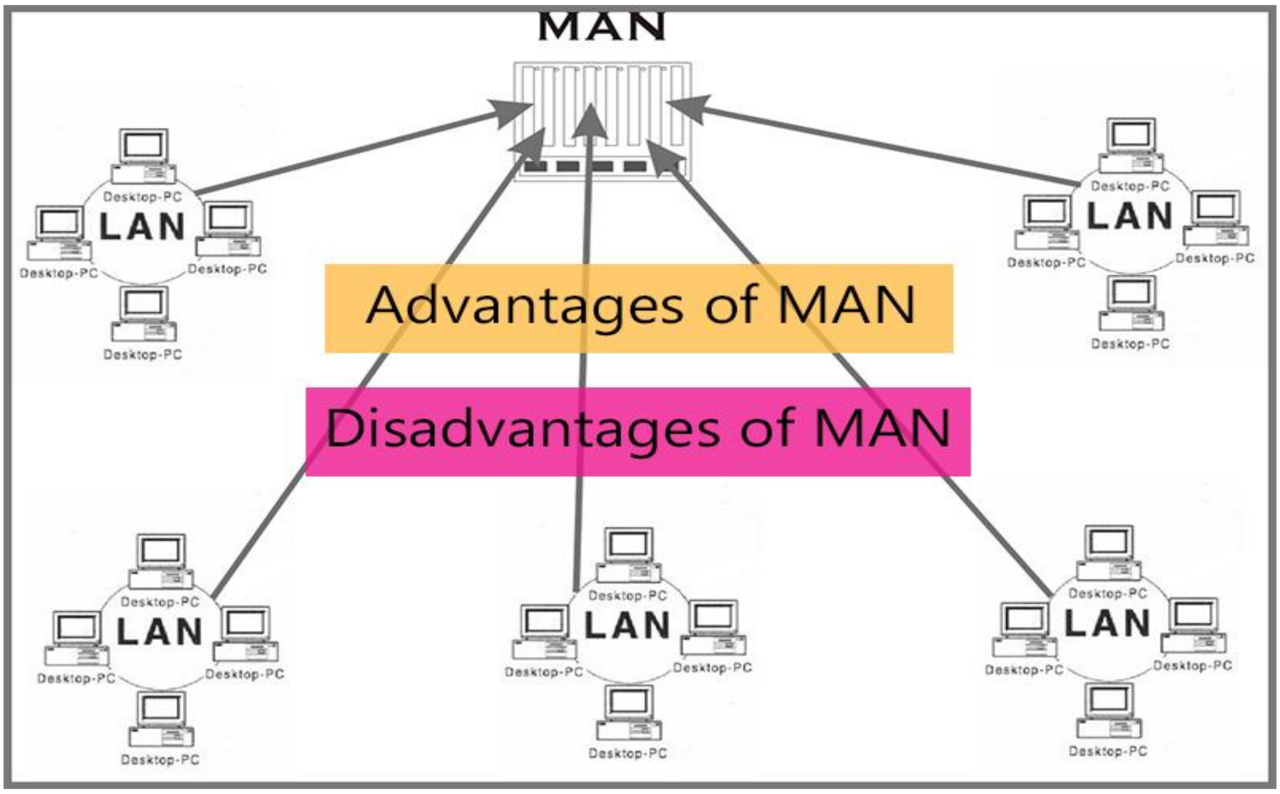
- FDDI (Fiber distribution data interface)
- ATM (Asynchronous transfer mode)
- SMDS (Switched multi-megabit data service)

FDDI is used for transferring data in LAN. Its range is within 200 kms. As FDDI can handle thousands of users so this standard is also included in MAN.

ATM (Asynchronous transfer mode) was developed in 1980 to transfer digital data fast. Mostly ATM is a widely used method in MAN. In ATM data is distributed online in the form of small fixed size packets. This method is best for doing audio and video conference online. ATM used packet switching and circuit switching to transfer real-time data.

In SMDS, data is transferred in packets over the large geographical area. SMDS is a connectionless service meaning data info is stored in the header and then it is transferred through any network.

The main purpose of MAN is to interconnect two LAN together. MAN also uses routers and switches to transfer data. MAN has a range of over 50kms. The **speed of metropolitan area network** is about 1000 Mbps.



Metropolitan Area Network (MAN) Diagram

Advantages of a metropolitan area network (MAN)

Below are some of the benefits of MAN: -

Less expensive:

It is less expensive to attach MAN with WAN. MAN gives the good efficiency of data. In MAN data is easily managed in a centralized way.

Sending local emails:

On MAN you can send local emails fast and free.

High speed than WAN:

MAN uses fiber optics so the speed of data can easily reach upon 1000 Mbps. Files and databases can be transferred fast.

Sharing of the internet:

In some installation of MANs, users can share their internet connection. So multiple users can get the same high-speed internet.

Conversion from LAN to MAN is easy:

MAN is a faster way to connect two fast LANs together. This is due to the fast configuration of links.

High Security:

MAN has a high-security level than WAN.

Disadvantages of metropolitan area network (MAN)

Difficult to manage:

If MAN becomes bigger then it becomes difficult to manage it. This is due to a security problem and other extra configuration.

Internet speed difference:

MAN cannot work on traditional phone copper wires. If MAN is installed on copper wires then there will be very low speed. So, it required the high cost to set up fiber optics for the first time.

Hackers attack:

In MAN there are high chances of attacking hackers on the network compared to LAN. So, data may be leaked. Data can be secured but it needs high trained staff and security tools.

Technical people required to set up:

To setup MAN it requires technical people that can correctly setup MAN. The technical people are network administrators and troubleshooters.

More wires required:

In MAN additional cables are required to connect two LAN which is another problem.

Examples of metropolitan area network (MAN)

Some of the examples of MAN are: -

- Digital cable television
- Used in government agencies

- University campuses
- Cable broadband
- Used to connect several branches of the local school
- In hospital (for communication between doctors, research offices, labs)
- A network of fire stations
- In airports
- Networking between community colleges within the country
- Used in public libraries

What is network topology?

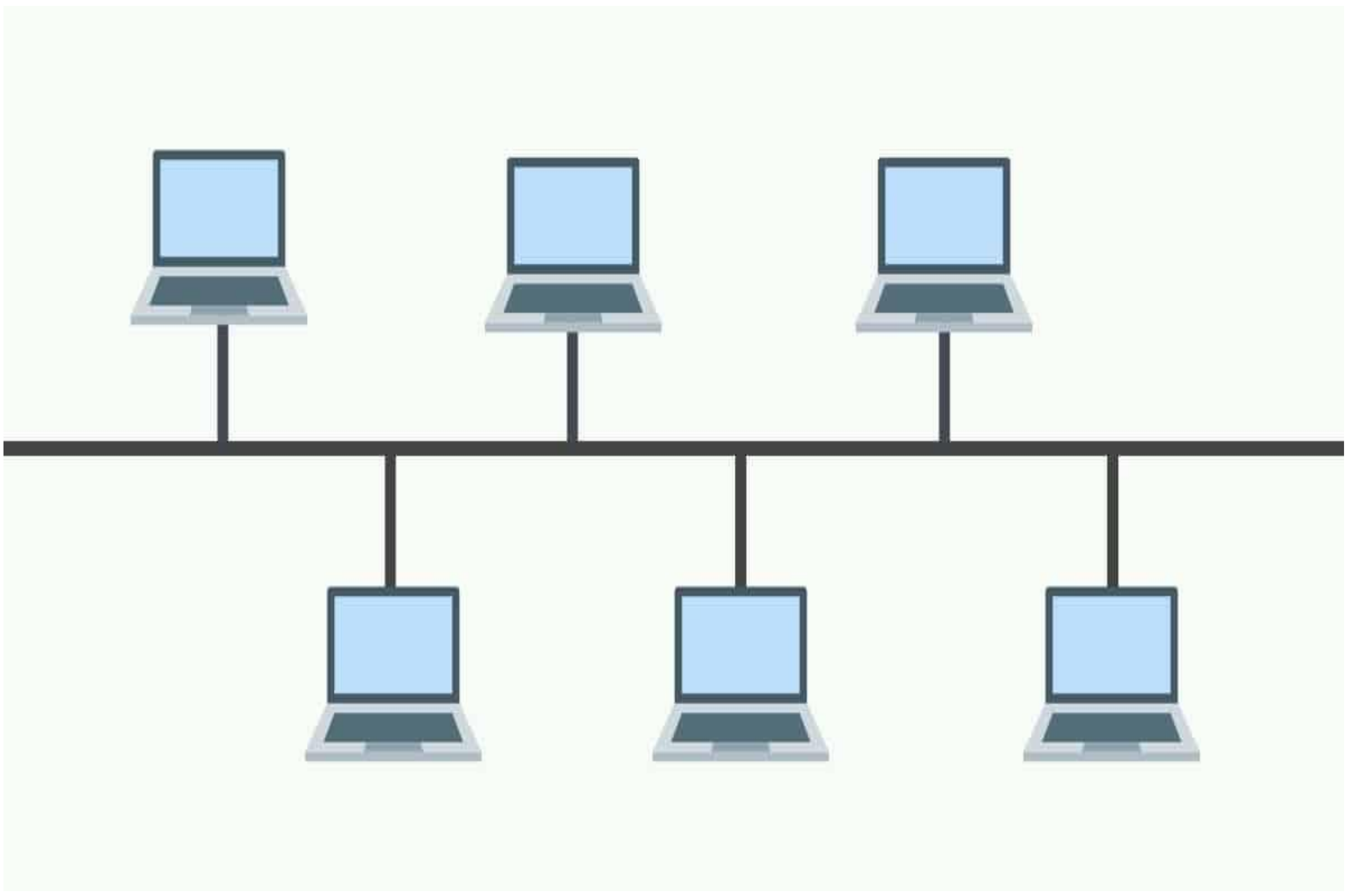
- Network topology is the description of the arrangement of nodes (e.g. switches and routers) and connections in a network, often represented as a graph.
- No matter how identical two organizations are, no two networks are exactly alike. However, many organizations are relying on well-established network topology models. Network topologies outline how devices are connected together and how data is transmitted from one node to another.

A **logical network topology** is a conceptual representation of how devices operate at particular layers of abstraction. A **physical topology** details how devices are physically connected. Logical and physical topologies can both be represented as visual diagrams.

A **network topology map** is a **map that allows an administrator to see the physical layout of connected devices**. Having the map of a network's topology on hand is very useful for understanding how devices connect to each other and the best techniques for troubleshooting.

There are many different types of topologies that enterprise networks have built on today and in the past. Some of the network topologies we're going to look at include **bus topology**, **ring topology**, **star topology**, **mesh topology**, and **hybrid topology**.

Bus Topology



Bus topology is a type of network where every device is connected to a single cable which runs from one end of the network to the other. This type of type of topology is often referred to as **line topology**. In a bus topology, data is transmitted in one direction only. If the bus topology has two endpoints then it is referred to as a **linear bus topology**.

Smaller networks with this type of topology use a coaxial or RJ45 cable to link devices together. However, the bus topology layout is outdated and you're unlikely to encounter a company using a bus topology today.

Advantages

Bus topologies were often used in smaller networks. One of the main reasons is that they **keep the layout simple**. All devices are connected to a single cable so you don't need to manage a complex topological setup.

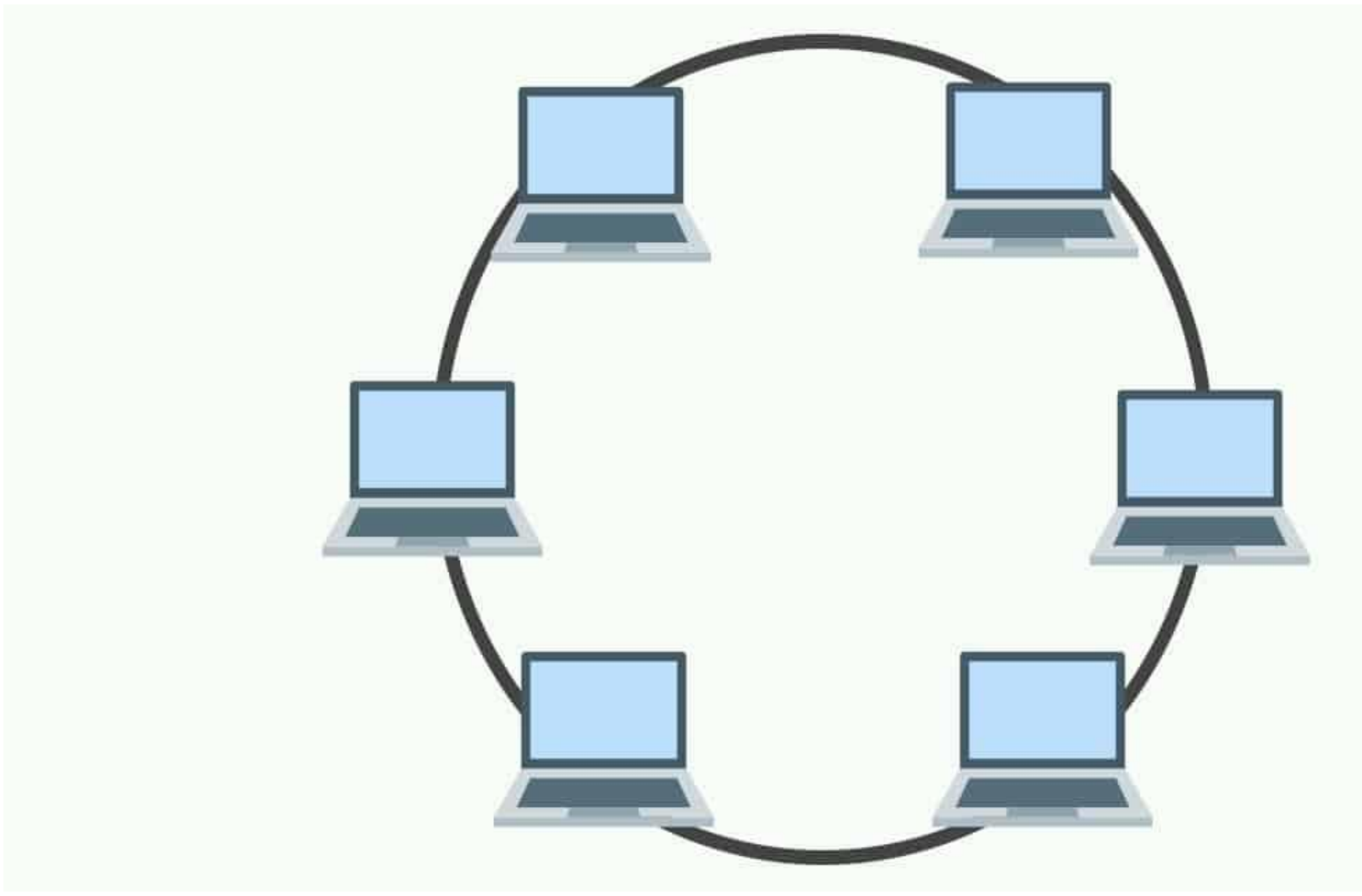
The layout also helped make bus topologies cost effective because they **can be run with a single cable**. In the event that more devices need to be added then you could simply join your cable to another cable.

Disadvantages

However, relying on one cable does mean that bus topologies have a single point of failure. If the cable fails then the entire network will go down. A cable failure would cost organizations a lot of time while they attempt to resume service. Further to this, high network traffic would decrease network performance because all the data travels through one cable.

This limitation makes bus topologies suitable only for smaller networks. The primary reason is that the more nodes you have, the slower your transmission speeds are going to be. It is also worth noting that bus topologies are limited in the sense that they are half-duplex, which means that data can't be transmitted in two opposite directions simultaneously.

Ring Topology

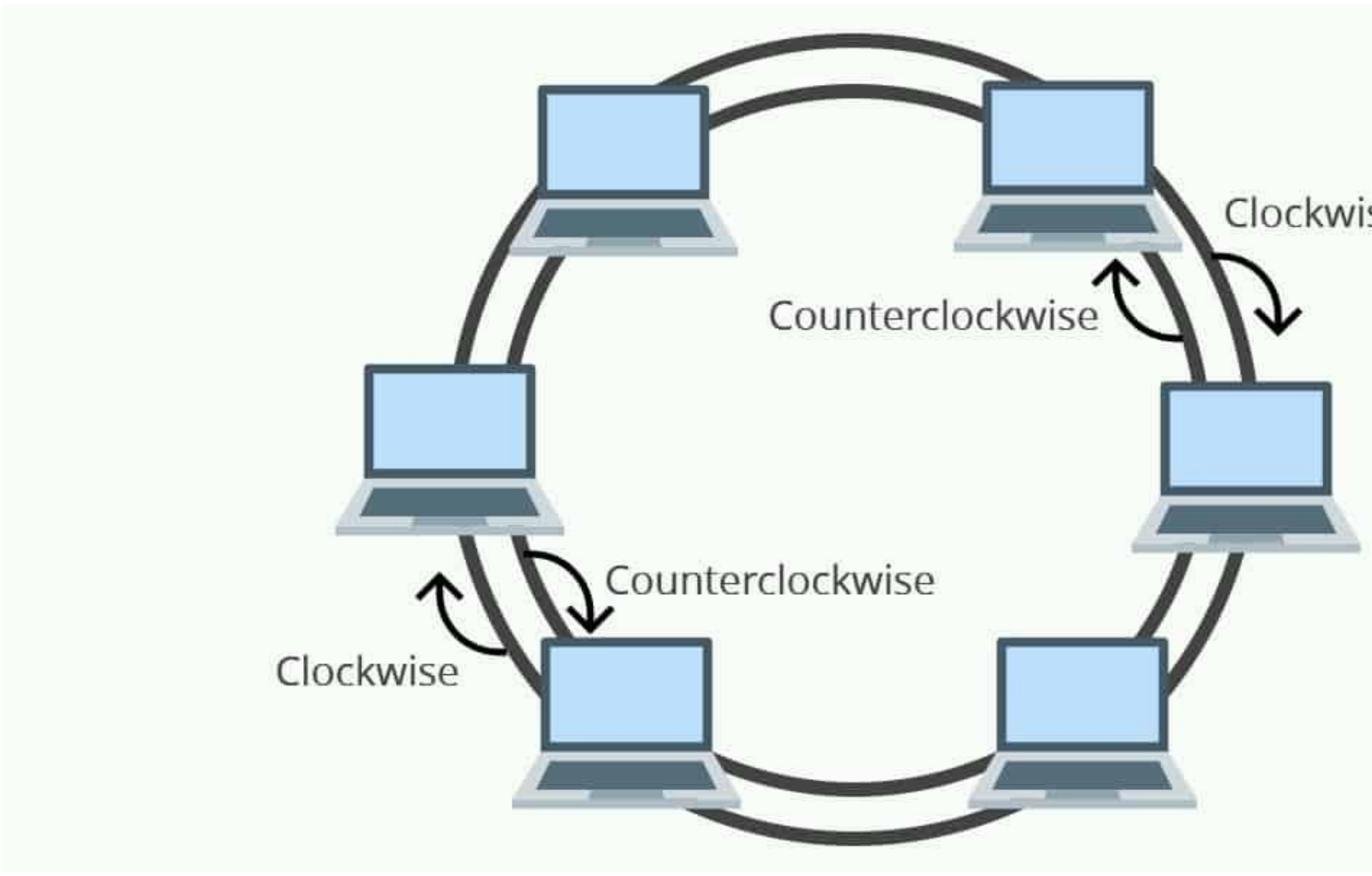


In networks with a ring topology, computers are connected to each other in a circular format. **Every device in the network will have two neighbours** and no more or no less. Ring topologies were commonly used in the past but you would be hard pressed to find an enterprise still using them today.

The first node is connected to the last node to link the loop together. As a consequence of being laid out in this format packets need to travel through all nodes on the way to their destination.

Within this topology, one node is chosen to configure the network and monitor other devices. Ring topologies are **half-duplex but can also be made full-duplex**. To make ring topologies full-duplex you would need to have two connections between network nodes to form a **Dual Ring Topology**.

Dual Ring Topology



As mentioned above, if ring topologies are configured to be bidirectional then they are referred to as dual ring topologies. Dual ring topologies provide each node with two connections, one in each direction. Thus, data can flow in a **clockwise** or **counter clockwise** direction.

Advantages:

In ring topologies the risk of packet collisions is very low due to the use of token-based protocols, which only allow one station to transmit data at a given time. This is compounded by the fact that data can move through **nodes** at high speeds which can be expanded on when more nodes are added.

Dual ring topologies provided an extra layer of protection because they were more resistant to failures. For instance, if a ring goes down within a node then the other ring can step up and back it up. Ring topologies were also low cost to install**.**

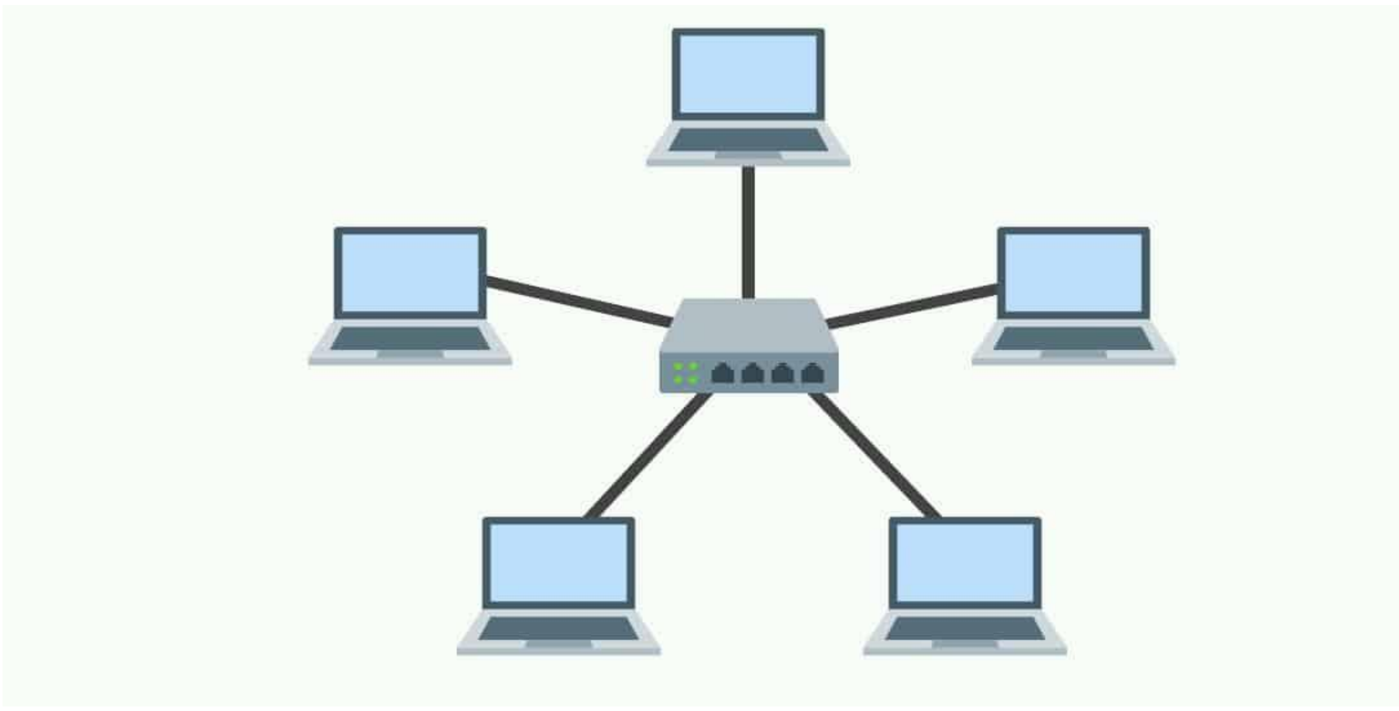
Disadvantages:

One of the reasons why ring topologies were replaced is because they are very vulnerable to failure. The **failure of one node can take the entire network out of operation**. This means that ring topology networks needed to be constantly managed to ensure that all nodes are in good health. However, even if the nodes were in good health your network **could still be knocked offline by a transmission line failure!**

Ring topologies also **raised scalability concerns**. For instance, bandwidth is shared by all devices within the network. In addition, **the more devices that are added** to a network **the more communication delay** the network experiences. This means that the number of devices added to a network topology needed to be monitored carefully to make sure that the network resources weren't stretched beyond their limit.

Making changes to a ring topology was also complicated because you **need to shut down the network to make changes** to existing nodes **or add new nodes**. This is far from ideal as you'll need to factor in downtime every time you want to make a change to the topological structure!

Star Topology



A star topology is a topology where every node in the network is connected to one central switch. Every device in the network is directly connected to the switch and indirectly connected to every other node. The relationship between these elements is that the central network device is a server and other devices are treated as clients. The central node has the responsibility of managing data transmissions across the network and acts as a repeater. In star topologies, computers are connected with a coaxial cable, twisted pair, or optical fibre cable.

Advantages:

Star topologies are most commonly-used because you **can manage the entire network from one location**: the central switch. As a consequence, if a node that isn't the central node goes down then the network will remain up. This gives star topologies a layer of protection against failures that aren't always present with other topology setups. Likewise, you **can add new**

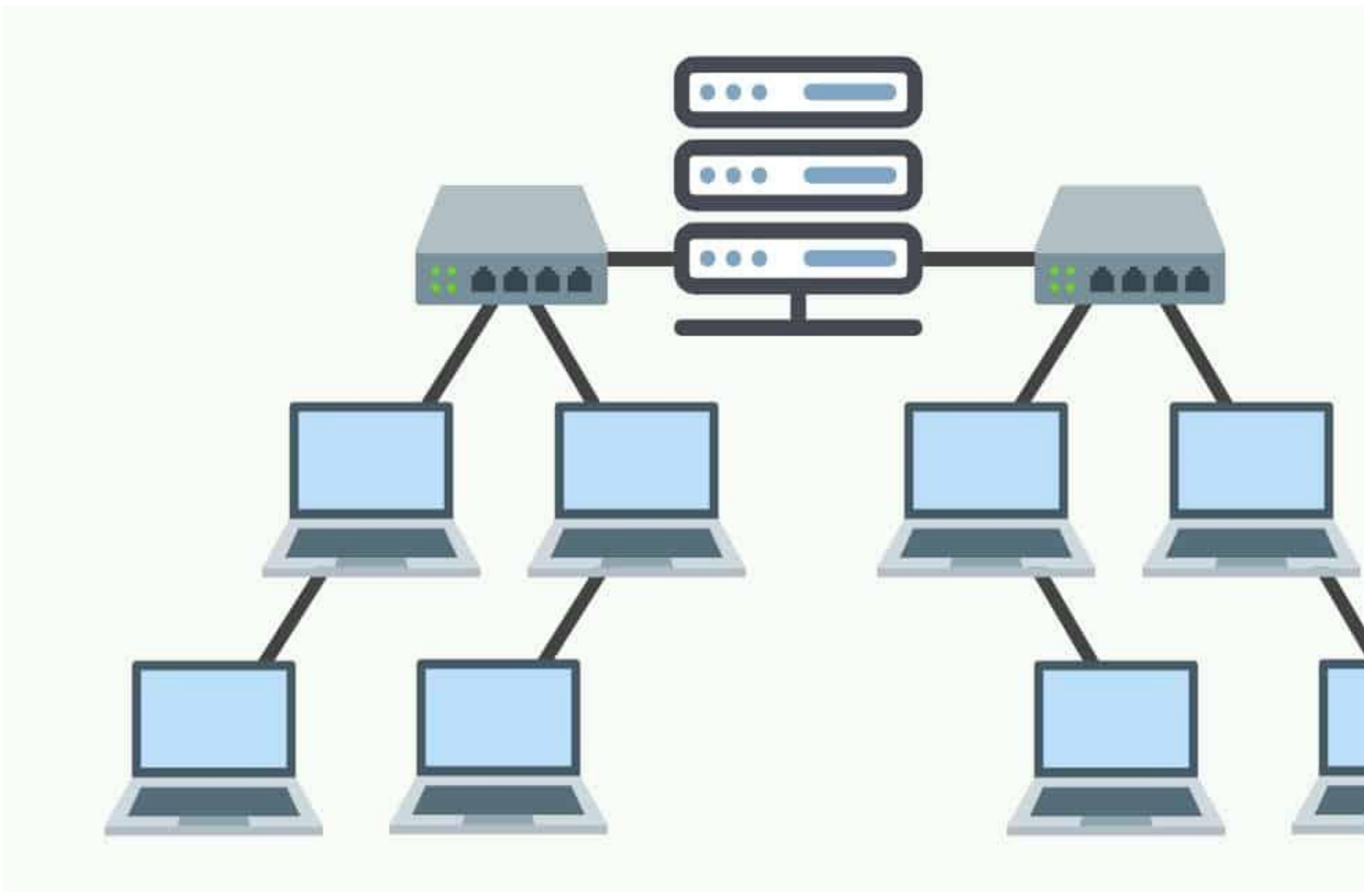
computers without having to take the network offline like you would have to do with a ring topology.

In terms of physical structure, star topologies require fewer cables than other topology types. This makes them **simple to set up and manage** over the long-term. The simplicity of the overall design makes it much easier for administrators to run troubleshooting when dealing with performance faults.

Disadvantages:

Though star topologies may be relatively safe from failure, **if the central switch goes down then the entire network will go down**. As such, the administrator needs to manage the health of the central node closely to make sure that it doesn't go down. The performance of the network is also **tied to the central node's configurations and performance**. Star topologies are easy to manage in most ways but they are far from cheap to set up and use.

Tree Topology



As the name suggests, a tree topology is a network structure that is shaped like a tree with its many branches. Tree topologies **have a root node** which is connected to other node hierarchy. The **hierarchy is parent-child** where there is only one mutual connection between two connected nodes. As a general rule, a tree topology needs to have three levels to the hierarchy in order to be classified this way. This form of topology is **used within Wide Area Networks** to sustain lots of spread-out devices.

Advantages:

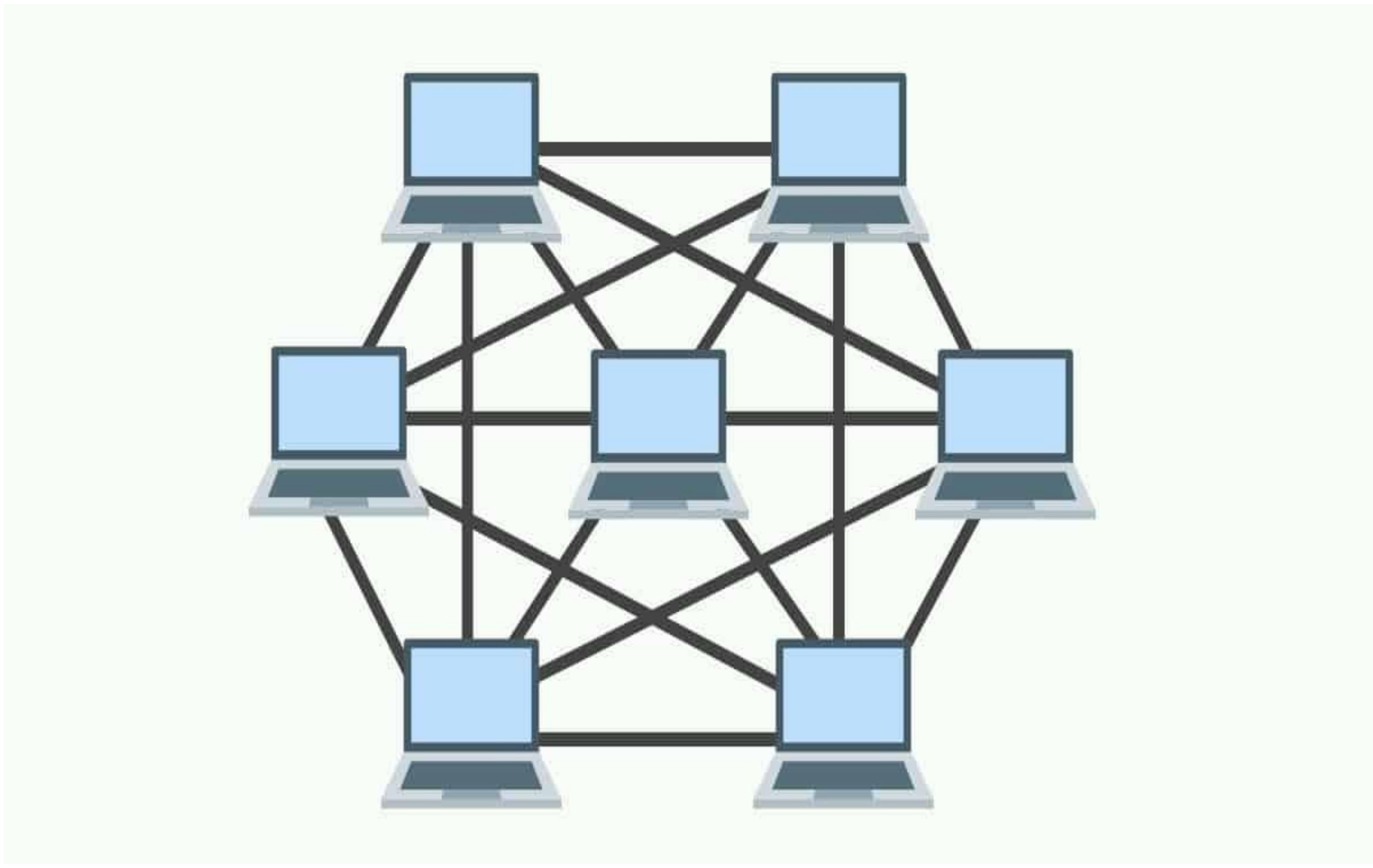
The main reason why tree topologies are **used is to extend bus and star topologies**. Under this hierarchical format, it is easy to add more nodes to the network when your organization grows in size. This format also **lends itself well to finding errors and troubleshooting** because you can check for performance issues systematically throughout the tree.

Disadvantages:

The most significant weakness of tree topology is the root node. **If the root node fails then all of its subtrees become partitioned**. There will still be partial connectivity within the network amongst other devices such as the failed node's parent.

Maintaining the network is not simple either because **the more nodes you add, the more difficult it becomes to manage** the network. Another disadvantage of a tree topology is the number of cables you need. Cables are required to connect every device throughout the hierarchy which makes the layout more complex when compared to a simpler topology.

Mesh Topology



A mesh topology is a point-to-point connection where nodes are interconnected. In this form of topology, **data is transmitted via two methods**: **routing** and **flooding**. Routing is where nodes use routing logic to work out the shortest distance to the packet's destination. In contrast, flooding is where data is sent to all nodes within the network. Flooding doesn't require any form of routing logic to work.

There are **two forms of mesh topology**: **partial mesh topology** and **full mesh topology**. With partial mesh topology, most nodes are interconnected but there are a few which are only connected to two or three other nodes. A full mesh topology is where every node is interconnected.

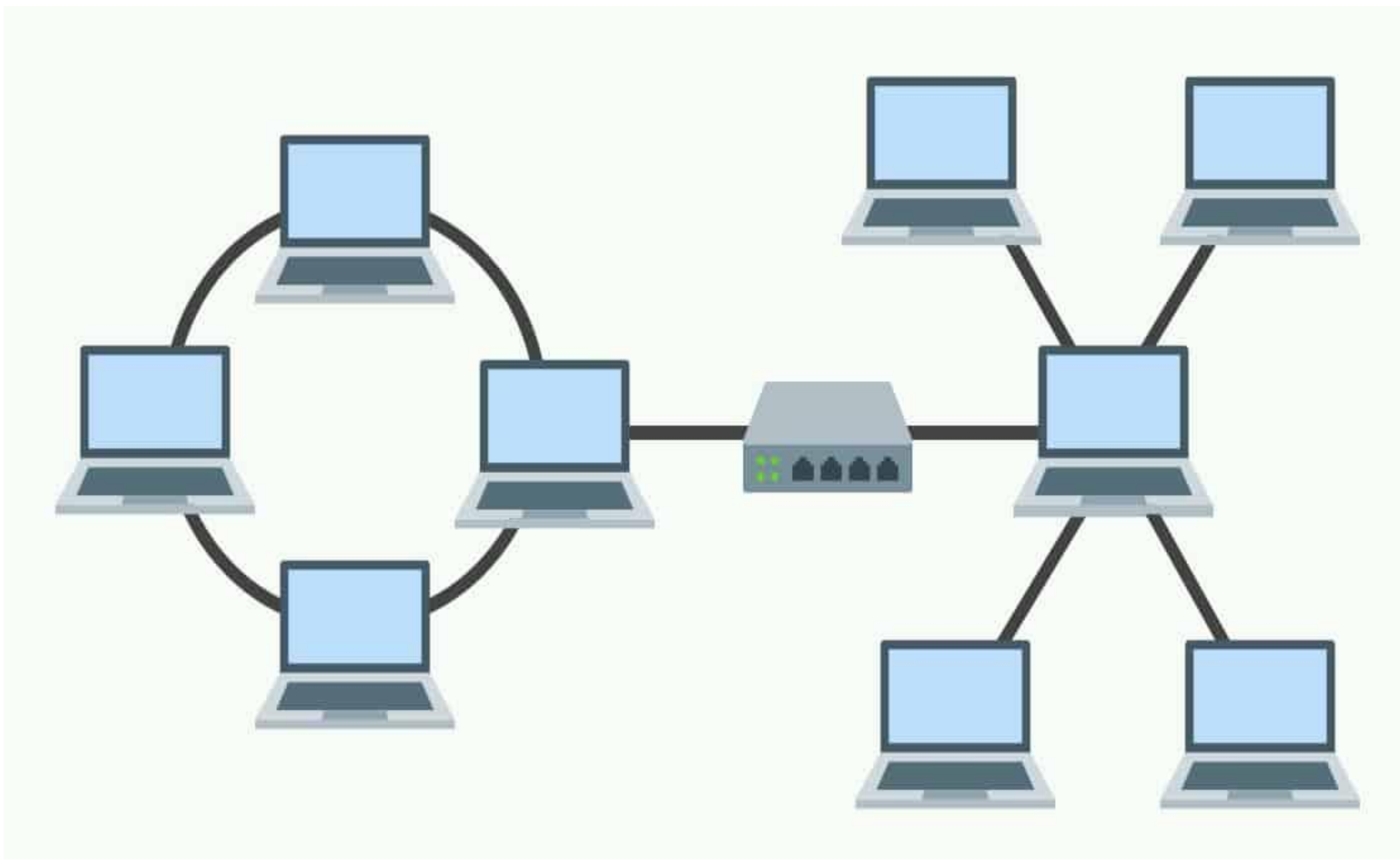
Advantages:

Mesh topologies are used first and foremost because they are reliable. The **interconnectivity of nodes makes them extremely resistant to failures**. There is no single machine failure that could bring down the entire network. The absence of a single point of failure is one of the reasons why this is a popular topology choice. This setup is also secure from being compromised.

Disadvantages:

However, mesh topologies are far from perfect. They **require an immense amount of configuration** once they are deployed. The topological layout is more complex than many other topologies and this is reflected by how long it takes to set up. You'll need to accommodate a whole host of new wiring which can add up to be quite expensive.

Hybrid Topology



When a topology is comprised of two or more different topologies it is referred to as a hybrid topology. Hybrid topologies are **most-commonly encountered in larger enterprises** where individual departments have network topologies that different from another topology in the organization. Connecting these topologies together will result in a hybrid topology. As a consequence, the capabilities and vulnerabilities depend on the types of topology that are tied together.

Advantages:

There are many reasons why hybrid topologies are used but they all have one thing in common: **flexibility**. There are few constraints on the structure that a hybrid topology cannot accommodate, and you **can incorporate multiple topologies into one hybrid setup**. As a consequence, hybrid topologies are very scalable. The scalability of hybrid setups makes them well-suited to larger networks.

Disadvantages:

Unfortunately, hybrid topologies **can be quite complex**, depending on the topologies that you decide to use. Each topology that is part of your hybrid topology will have to be managed according to its unique requirements. This makes administrators' jobs more difficult because they are going to have to attempt to manage multiple topologies rather than a single one. In addition, setting up a hybrid topology **can end up being quite costly**.

Which Topology Should I Choose?

There is a range of factors that you need to take into account when choosing which topology to use. Before choosing a topology, you'll want to closely consider the following:

- Length of cable needed
- Cable type
- Cost
- Scalability

First, you need to **take into account the length of the cable you need** to provide service to all your network devices. A bus topology is the most lightweight in terms of cable needs. In this sense, this would be the simplest topology to install and buy cable for. This ties into the second factor, you need to **consider the type of cable you're going to use**. Cable types range from twister pairs to coaxial cables and optical fibre cables.

The cost of installing the topology is also very important. The more complex the topology you choose is, the more you'll need to pay in terms of resources and time to create that setup.

The final factor you'll want to take into account is scalability. **If you're planning to upscale** your network infrastructure in the future you want to make sure that you **use a network that is easy to add devices to**. A star topology network is ideal for this because you can add nodes with minimal disruption. This isn't as simple within a ring network because you will incur downtime if you add any nodes.

Internet connection concepts: -

Using the Internet, computers connect and communicate with one another, primarily using the TCP/IP (Transmission Control Protocol / Internet Protocol). Think of TCP/IP as a book of rules, a step-by-step guide that each computer uses to know how to talk to another computer. This book of rules dictates what each computer must do to transmit data, when to transmit data, how to transmit that data. It also states how to receive data in the same manner. If the rules are not followed, the computer can't connect to another computer, nor send and receive data between other computers.

To connect to the Internet and other computers on a network, a computer must have a NIC (network interface card) installed. A network cable plugged into the NIC on one end and plugged into a cable modem, DSL modem, router, or switch can allow a computer to access the Internet and connect to other computers.

ISPs (Internet service providers)

ISPs (Internet service providers), the companies that provide Internet service and connectivity, also follow these rules. The ISP provides a bridge between your computer and all the other computers in the world on the Internet. The ISP uses the TCP/IP protocols to make computer-to-computer connections possible and transmit data between them. When connected to an ISP, you're assigned an IP address, which is a unique address given to your computer or network to communicate on the Internet.

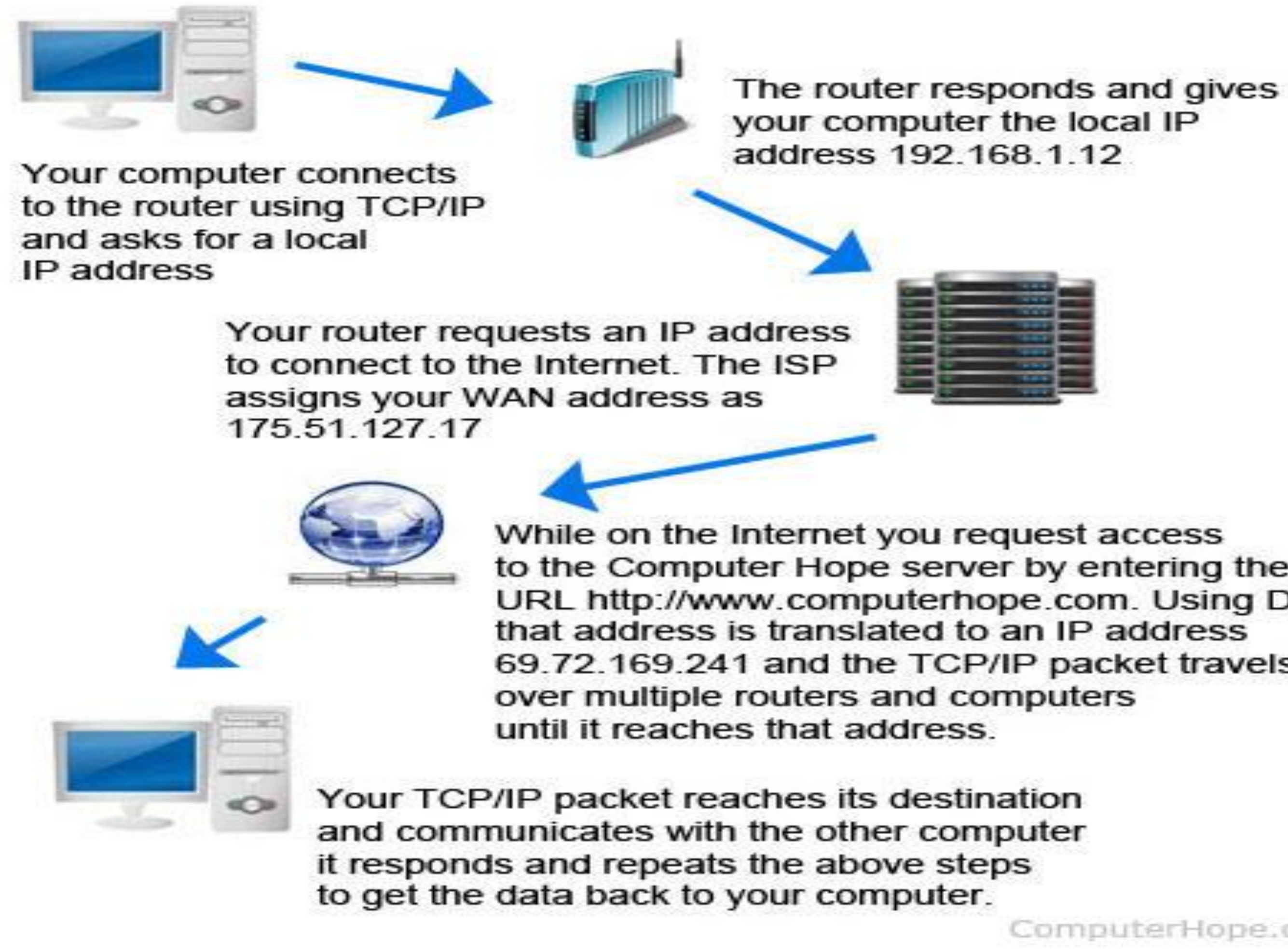
Home network

If you have a home computer network, the computers are also using TCP/IP to connect. The TCP/IP protocol allows each computer to "see" the other computers on the network, as well as share files and printers.

When computers connect on the same network, it is called a local area network, or LAN. When multiple networks are connected, it is called a wide area network, or WAN. With this type of network, your home has a network router that connects to your ISP. The router is given the IP address for your connection to the Internet and then assigns local IP addresses to each device in your network. These local addresses are often 192.168.1.2-255. When accessing a local computer in your network, your router sends your TCP/IP packets between the local IP addresses. However, when you want to connect to the Internet, your router uses the IP address assigned by the ISP. Your IP address is not a 192.168.x.x address because the ISP assigns that IP address and not your router.

When requesting information from a web page, such as Computer Hope, you enter a URL that is easy to understand and remember. For your computer to access the computer containing the pages, that URL must be converted into an IP address, which is done with DNS. Once DNS has converted the URL into an IP address, the routers on the Internet will know how to route your TCP/IP packet.

The illustration below helps explain the information in the previous sections about your computer communicates with others on the Internet.



The router responds and gives your computer the local IP address 192.168.1.12
Your computer connects to the router using TCP/IP and asks for a local IP address
Your router requests an IP address to connect to the Internet. The ISP assigns your WAN address as 175.51.127.17
While on the Internet you request access to the Computer Hope server by entering the URL http://www.computerhope.com. Using D that address is translated to an IP address 69.72.169.241 and the TCP/IP packet travels over multiple routers and computers until it reaches that address.
Your TCP/IP packet reaches its destination and communicates with the other computer it responds and repeats the above steps to get the data back to your computer.
ComputerHope.

Windows, macOS, and Linux computers use the TCP/IP protocol to connect to other computers on a LAN or WAN. Connecting to a LAN or WAN requires either a wired connection or a wireless connection. A wired connection is usually done using a network cable (Cat 5 or Cat 6 network cable). A wireless connection (Wi-Fi) uses an 802.11b, 802.11g or 802.11n wireless network card. With both connection types, a network router is usually required to connect to other computers. Connecting to the Internet at your home also requires either a cable modem or a DSL modem, depending on which ISP you use.

Unit: -2

Web browser

Definition

A browser, short for web browser, is the software application (a program) that you're using right now to search for, reach and explore websites. Whereas Excel® is a program for spreadsheets and Word® a program for writing documents, a browser is a program for Internet exploring (which is where that name came from).

Browsers don't get talked about much. A lot of people simply click on the "icon" on our computers that take us to the Internet—and that's as far as it goes. And in a way, that's enough. Most of us simply get in a car and turn the key...we don't know what kind of engine we have or what features it has...it takes us where we want to go. That's why when it comes to computers:

- There are some computer users that can't name more than one or two browsers
- Many of them don't know they can switch to another browser for free
- There are some who go to Google's webpage to "google" a topic and think that Google is their browser.

So for some basic browser education sake, let's cover a few points:

- **Know your browser.** Look at the very far-upper-left corner of your screen. You'll see the name of your browser.
- **Get the latest version.** Browsers get updates and updated regularly, usually because computers and technology change fast also. You can check what version of your browser you're currently using by going to whatbrowser.org.
- **Try a different browser.** You can switch to another browser at any time. It won't affect your computer and it will give you an idea of how they are different.

- **Read browser reviews.** You can compare features of the different browsers on websites like http://internet-browser-review.toptenreviews.com. You'll learn what kind of features browsers offer and what to look for.

FUNCTION:

The purpose of a web browser is to fetch information resources from the Web and display them on a user's device.

This process begins when the user inputs a Uniform Resource Locator (URL), such as *https://en.wikipedia.org/*, into the browser. Virtually all URLs on the Web start with either *http:* or *https:* which means the browser will retrieve them with the Hypertext Transfer Protocol (HTTP). In the case of *https:*, the communication between the browser and the web server is encrypted for the purposes of security and privacy.

Once a web page has been retrieved, the browser's rendering engine displays it on the user's device. This includes image and video formats supported by the browser.

Web pages usually contain hyperlinks to other pages and resources. Each link contains a URL, and when it is clicked or tapped, the browser navigates to the new resource. Thus the process of bringing content to the user begins again.

Most browsers use an internal cache of web page resources to improve loading times for subsequent visits to the same page. The cache can store many items, such as large images, so they do not need to be downloaded from the server again.[15] Cached items are usually only stored for as long as the web server stipulates in its HTTP response messages.

Categories of search engines:

What is a Search Engine?

The purpose of a search engine is to extract requested information from the huge database of resources available on the internet. Search engines become an important day to day tool for finding the required information without knowing where exactly it is stored. Internet usage has been tremendously increased in recent days with the easy to use search engines like Google, Bing and Yahoo! There are different types of search engines to get the information you are looking for. In this article, we will explain different types of search engines and purpose of them.

Popular Search Engines

Why Search Engines are Important?

Search engines are part of daily life for two types of people.

- Users who search and get information
- Site owners who try to optimize their websites for getting top rank in the search results.
  User do more than billions of searches only on Google to find relevant information. This opens out a huge scope for businesses and online content publishers to attract people to their website for free. Search engines follow guidelines and have their own algorithm to decide the ranking of websites in search results. Optimizing websites for Google and other search engines is an essential part of any website owner for reaching out the large audience. The visitors can generate revenue for site owners either through advertisements displayed on the site or though purchasing products.

Different Types of Search Engines

Search engines are classified into the following three categories based on how it works.

1. Crawler based search engines
2. Human powered directories
3. Hybrid search engines
4. Other special search engines
   Let us discuss all types of search engines in detail in the following sections.

1. Crawler Based Search Engines

All crawler based search engines use a crawler or bot or spider for crawling and indexing new content to the search database. There are four basic steps, every crawler based search engines follow before displaying any sites in the search results.

- Crawling
- Indexing
- Calculating Relevancy
- Retrieving the Result

1.1. Crawling

Search engines **crawl** the whole web to fetch the web pages available. A piece of software called *crawler* or *bot* or *spider*, performs the crawling of the entire web. The crawling frequency depends on the search engine and it may take few days between crawls. This is the reason sometimes you can see your old or deleted page content is showing in the search results. The search results will show the new updated content, once the search engines crawl your site again.

1.2. Indexing

Indexing is next step after crawling which is a process of identifying the words and expressions that best describe the page. The identified words are referred as keywords and the page is assigned to the identified keywords. Sometimes when the crawler does not understand the meaning of your page, your site may rank lower on the search results. Here you need to optimize your pages for search engine crawlers to make sure the content is easily understandable. Once the crawler's pickup correct keywords your page will be assigned to those keywords and rank high on search results.

1.3. Calculating Relevancy

Search engine compares the search string in the search request with the indexed pages from the database. Since it is likely that more than one page contains the search string, search engine starts **calculating the relevancy** of each of the pages in its index with the search string.
There are various algorithms to calculate relevancy. Each of these algorithms has different relative weights for common factors like keyword density, links, or meta tags. That is why different search engines give different search results pages for the same search string. It is a known fact that all major search engines periodically change their algorithms. If you want to keep your site at the top, you also need to adapt your pages to the latest changes. This is one reason to devote permanent efforts to SEO, if you like to be at the top.

1.4. Retrieving Results

The last step in search engines' activity is **retrieving** the results. Basically, it is simply displaying them in the browser in an order. Search engines sort the endless pages of search results in the order of most relevant to the least relevant sites.

Examples of Crawler Based Search Engines

Most of the popular search engines are crawler-based search engines and use the above technology to display search results. Example of crawler-based search engines:

- Google
- Bing
- Yahoo!
- Baidu
- Yandex

Besides these popular search engines there are many other crawler-based search engines available like DuckDuckGo, AOL and Ask.

2. Human Powered Directories

Human powered directories also referred as open directory system depends on human based activities for listings. Below is how the indexing in human powered directories work:

- Site owner submits a short description of the site to the directory along with category it is to be listed.
- Submitted site is then manually reviewed and added in the appropriate category or rejected for listing.
- Keywords entered in a search box will be matched with the description of the sites. This means the changes made to the content of a web pages are not taken into consideration as it is only the description that matters.
- A good site with good content is more likely to be reviewed for free compared to a site with poor content.

Yahoo! Directory and DMOZ were perfect examples of human powered directories. Unfortunately, automated search engines like Google, wiped out all those human powered directory style search engines out of the web.

3. Hybrid Search Engines

Hybrid Search Engines use both crawler based and manual indexing for listing the sites in search results. Most of the crawler-based search engines like Google basically uses crawlers as a primary mechanism and human powered directories as secondary mechanism. For example, Google may take the description of a webpage from human powered directories and show in the search results. As human powered directories are disappearing, hybrid types are becoming more and more crawler-based search engines.

But still there are manual filtering of search result happening to remove the copied and spammy sites. When a site is being identified for spammy activities, the website owner needs to take corrective action and resubmit the site to search engines. The experts do manual review of the submitted site before including it again in the search results. In this manner though the crawlers control the processes, the control is manual to monitor and show the search results naturally.

4. Other Types of Search Engines

Besides the above three major types, search engines can be classified into many other categories depending upon the usage. Below are some of the examples:

- Search engines have different types of bots for exclusively displaying images, videos, news, products and local listings. For example, Google News page can be used to search only news from different newspapers.
- Some of the search engines like Dogpile collects meta information of the pages from other search engines and directories to display in the search results. This type of search engines are called metasearch engines.

Surfing the net:

Surfing the Internet is a term typically used to describe an undirected type of web of browsing where users whimsically follow one interesting link to another without a planned search strategy or definite objective. Surfing the net has become a popular pastime, for many Internet users.

Surfing the Internet' is not to be confused with the phrase 'browsing the Internet' which refers to exploring the web with a clear-cut objective but without any planned search strategies. Searching the web refers to exploring the Internet with a definite in both strategy and objective.

Surfing the Internet has been likened to the ironic term 'channel surfing, which is used to describe randomly changing TV channels. Its only relation to actual surfboarding has to do with the notion of 'going with flow' when surfing.

Jean Armour Polly is credited with the first published use of the phrase. She used it in an article titled 'surfing the Internet' that was published, in the June 1992 issue of the monthly magazine, Wilson Library Bulletin.

Polly was also key in popularizing the phrase; she maintains that she purposefully wanted it to have the exact connotation it currently has. Coining the phrase has since been attributed to Internet pioneer Mark McCahill.

• At the other end of the spectrum is the so-called Internet appliance, the very low-cost device for just **surfing the Net**.

• It was the year of spinoffs, surging financial markets and **surfing the Net**.

• We give them quizzes on Britain and allow them to **surf the net**

.• Who spends an inordinate number of work hours **surfing the Internet**?

• That means you can **surf the Net** and talk on the phone at the same time over one line

.• So a user could be **surfing the Net** at warp speed while talking on the phone

.• A recent survey shows that about half of all users **surf the Net** from their homes

.• Surveys show millions of workers use their office computers to play games, **surf the Net** or worse.

HTTP (HYPERTEXT TRANSFER PROTOCOL)

The **Hypertext Transfer Protocol** (**HTTP**) is an application protocol for distributed, collaborative, hypermedia information systems. HTTP is the foundation of data communication for the World Wide Web, where hypertext documents include hyperlinks to other resources that the user can easily access, for example by a mouse click or by tapping the screen in a web browser.

Development of HTTP was initiated by Tim Berners-Lee at CERN in 1989. Development of early HTTP Requests for Comments (RFCs) was a coordinated effort by the Internet Engineering Task Force (IETF) and the World Wide Web Consortium (W3C), with work later moving to the IETF.

HTTP/1.1 was first documented in RFC 2068 in 1997. That specification was obsoleted by RFC 2616 in 1999, which was likewise replaced by the RFC 7230 family of RFCs in 2014.

HTTP/2 is a more efficient expression of HTTP's semantics "on the wire", and was published in 2015; it is now supported by major web servers and browsers over Transport Layer Security (TLS) using an Application-Layer Protocol Negotiation (ALPN) extension[2] where TLS 1.2 or newer is required.[3]

HTTP/3 is the proposed successor to HTTP/2, which is already in use on the web, using UDP instead of TCP for the underlying transport protocol. Like HTTP/2, it does not obsolete previous major versions of the protocol. Support for HTTP/3 was added to Cloudflare and Google Chrome in September 2019, and can be enabled in the stable versions of Chrome and Firefox.

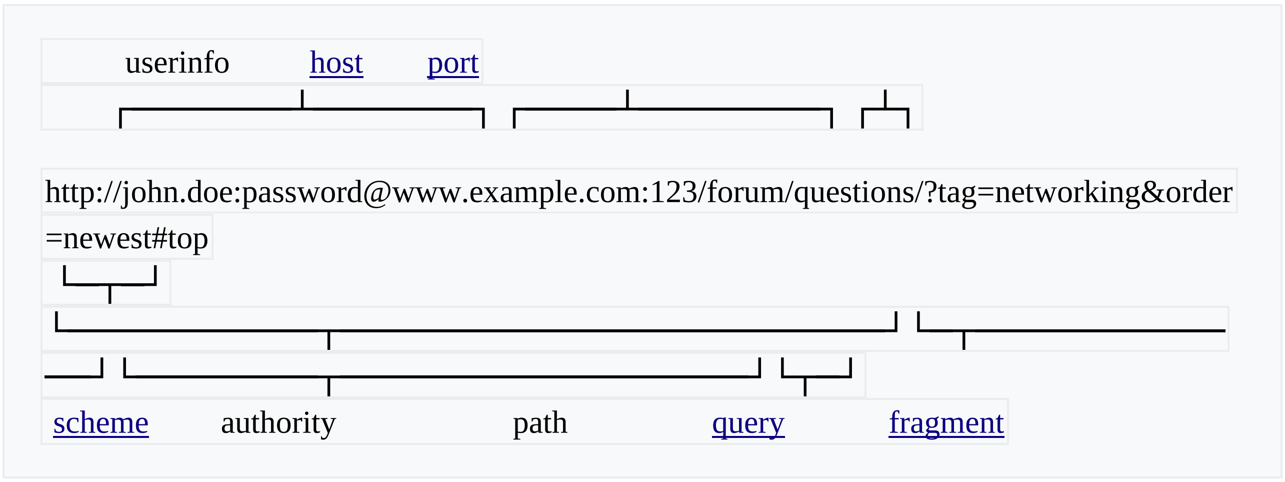
HTTP functions as a request–response protocol in the client–server computing model. A web browser, for example, may be the *client* and an application running on a computer hosting a website may be the *server*. The client submits an HTTP *request* message to the server. The server, which provides *resources* such as HTML files and other content, or performs other functions on behalf of the client, returns a *response* message to the client. The response contains completion status information about the request and may also contain requested content in its message body.

A web browser is an example of a *user agent* (UA). Other types of user agent include the indexing software used by search providers (web crawlers), voice browsers, mobile apps, and other software that accesses, consumes, or displays web content.

HTTP is designed to permit intermediate network elements to improve or enable communications between clients and servers. High-traffic websites often benefit from web cache servers that deliver content on behalf of upstream servers to improve response time. Web browsers cache previously accessed web resources and reuse them, when possible, to reduce network traffic. HTTP proxy servers at private network boundaries can facilitate communication for clients without a globally routable address, by relaying messages with external servers.

HTTP is an application layer protocol designed within the framework of the Internet protocol suite. Its definition presumes an underlying and reliable transport layer protocol,[9] and Transmission Control Protocol (TCP) is commonly used. However, HTTP can be adapted to use unreliable protocols such as the User Datagram Protocol (UDP), for example in HTTPU and Simple Service Discovery Protocol (SSDP).

HTTP resources are identified and located on the network by Uniform Resource Locators (URLs), using the Uniform Resource Identifiers (URI's) schemes *http* and *https*. For example, including all optional components:


userinfo
host
port
http://john.doe:password@www.example.com:123/forum/questions/?tag=networking&order
=newest#top
scheme
authority
path
query
fragment

As defined in RFC 3986, URIs are encoded as hyperlinks in HTML documents, so as to form interlinked hypertext documents.

HTTP/1.1 is a revision of the original HTTP (HTTP/1.0). In HTTP/1.0 a separate connection to the same server is made for every resource request. HTTP/1.1 can reuse a connection multiple times to download images, scripts, stylesheets, *etc* after the page has been delivered. HTTP/1.1 communications therefore experience less latency as the establishment of TCP connections presents considerable overhead.

URL(UNIFORM RESOURCE LOCATOR):

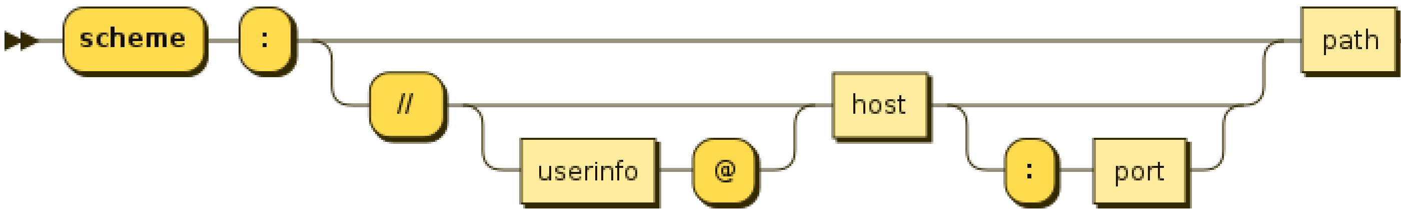
A **Uniform Resource Locator** (**URL**), colloquially termed a **web address**, is a reference to a web resource that specifies its location on a computer network and a mechanism for retrieving it. A URL is a specific type of Uniform Resource Identifier (URI), although many people use the two terms interchangeably. URLs occur most commonly to reference web pages (http), but are also used for file transfer (ftp), email (mailto), database access (JDBC), and many other application .Every HTTP URL conforms to the syntax of a generic URI. The *URI generic syntax* consists of a hierarchical sequence of five *components*:[14]

```
URI = scheme: [//authority] path [? query] [#fragment]
```

where the authority component divides into three *subcomponents*:

```
authority = [userinfo@] host [: port]
```

This is represented in a syntax diagram as:



The URI comprises:

- A non-empty **scheme** component followed by a colon (`:`), consisting of a sequence of characters beginning with a letter and followed by any combination of letters, digits, plus (`+`), period (`.`), or hyphen (`-`). Although schemes are case-insensitive, the canonical form is lowercase and documents that specify schemes must do so with lowercase letters. Examples of popular schemes include `http`, `https`, `ftp`, `mailto`, `file`, `data`, and `irc`. URI schemes should be registered with the Internet Assigned Numbers Authority (IANA), although non-registered schemes are used in practice.
- An optional **authority** component preceded by two slashes (`//`), comprising:
    - An optional **userinfo** subcomponent that may consist of a user name and an optional password preceded by a colon (`:`), followed by an at symbol (`@`). Use of the format `username:password` in the userinfo subcomponent is deprecated for security reasons. Applications should not render as clear text any data after the first colon (`:`) found within a userinfo subcomponent unless the data after the colon is the empty string (indicating no password).
    - A **host** subcomponent, consisting of either a registered name (including but not limited to a hostname), or an IP address. IPv4 addresses must be in dot-decimal notation, and IPv6 addresses must be enclosed in brackets (`[]`).
    - An optional **port** subcomponent preceded by a colon (`:`).
- A **path** component, consisting of a sequence of path segments separated by a slash (`/`). A path is always defined for a URI, though the defined path may be empty (zero length). A segment may also be empty, resulting in two consecutive slashes (`//`) in the path component. A path component may resemble or map exactly to a file system path, but does not always imply a relation to one. If an authority component is present, then the path component must either be empty or begin with a slash (`/`). If an authority component is absent, then the path cannot begin with an empty segment, that is with two slashes (`//`), as the following characters would be interpreted as an authority component.[18] The final segment of the path may be referred to as a 'slug'.

Query delimiter	Example
Ampersand (`&`)	`key1=value1&key2=value2`
Semicolon (`;`)[d][*incomplete short citation*]	`key1=value1;key2=value2`

- An optional **query** component preceded by a question mark (`?`), containing a query string of non-hierarchical data. Its syntax is not well defined, but by convention is most often a sequence of attribute–value pairs separated by a delimiter.
- An optional **fragment** component preceded by a hash (`#`). The fragment contains a fragment identifier providing direction to a secondary resource, such as a section heading in an article identified by the remainder of the URI. When the primary resource

is an HTML document, the fragment is often an id attribute of a specific element, and web browsers will scroll this element into view.

A web browser will usually dereference a URL by performing an HTTP request to the specified host, by default on port number 80. URLs using the https scheme require that requests and responses be made over a secure connection to the website.

ONLINE CHATTING

Online chat may refer to any kind of communication over the Internet that offers a real-time transmission of text messages from sender to receiver. Chat messages are generally short in order to enable other participants to respond quickly. Thereby, a feeling similar to a spoken conversation is created, which distinguishes chatting from other text-based online communication forms such as Internet forums and email. Online chat may address point-to-point communications as well as multicast communications from one sender to many receivers and voice and video chat, or may be a feature of a web conferencing service.

Online chat in a less stringent definition may be primarily any direct text-based or video-based (webcams), one-on-one chat or one-to-many group chat (formally also known as synchronous conferencing), using tools such as instant messengers, Internet Relay Chat (IRC), talkers and possibly MUDs. The expression *online chat* comes from the word *chat* which means "informal conversation". Online chat includes web-based applications that allow communication – often directly addressed, but anonymous between users in a multi-user environment. Web conferencing is a more specific online service, that is often sold as a service, hosted on a web server controlled by the vendor.

The first online chat system was called Talkomatic, created by Doug Brown and David R. Woolley in 1973 on the PLATO System at the University of Illinois. It offered several channels, each of which could accommodate up to five people, with messages appearing on all users' screens character-by-character as they were typed. Talkomatic was very popular among PLATO users into the mid-1980s. In 2014, Brown and Woolley released a web-based version of Talkomatic.

The first online system to use the actual command "chat" was created for The Source in 1979 by Tom Walker and Fritz Thane of Dialcom, Inc.

Other chat platforms flourished during the 1980s. Among the earliest with a GUI was Broadcast, a Macintosh extension that became especially popular on university campuses in America and Germany.

The first transatlantic Internet chat took place between Oulu, Finland and Corvallis, Oregon in February 1989.

The first dedicated online chat service that was widely available to the public was the CompuServe CB Simulator in 1980, created by CompuServe executive Alexander "Sandy" Trevor in Columbus, Ohio. Ancestors include network chat software such as UNIX "talk" used in the 1970s.

SOFTWRAE AND PROTOCOLS:

The following are common chat programs and protocols:

- AOL Instant Messenger (AIM)

- Apple Messages
- BroadCast
- Camfrog
- Campfire
- Discord
- Gadu-Gadu
- Google Talk
- I2P-Messenger (anonymous, end-to-end encrypted in for the I2Pnetwork)
- Internet Citizen's Band (ICB)
- ICQ (OSCAR)
- Internet Relay Chat (IRC)
- Mattermost

VIDEO CONFERENCING CONCEPTS

Videoconferencing (or video conference) means to conduct a conference between two or more participants at different sites by using computer networks to transmit audio and video data. For example, a *point-to-point* (two-person) video conferencing system works much like a video telephone. Each participant has a video camera, microphone, and speakers mounted on his or her computer. As the two participants speak to one another, their voices are carried over the network and delivered to the other's speakers, and whatever images appear in front of the video camera appear in a window on the other participant's monitor. Multipoint videoconferencing allows three or more participants to sit in a virtual conference room and communicate as if they were sitting right next to each other. Until the mid-90s, the hardware costs made videoconferencing prohibitively expensive for most organizations, but that situation is changing rapidly. Many analysts believe that videoconferencing will be one of the fastest-growing segments of the computer industry in the latter half of the decade.

Video Conference Calls:

If your organization often sets internal or external meetings between groups of people in multiple locations, video can greatly enhance the experience. With low-latency live video, you can more closely replicate the experience of meeting in person than an audio conference call. You can read emotions and discuss ideas with more clarity.

Live video meeting tools provide an easy web link to share video between conference rooms. You can set them up on laptop webcams or install video systems in your conference rooms.

One-on-One Video Conversations

Meeting tools are also great for one-on-one meetings. If you have two people or small teams in different locations who need to meet virtually, many services offer web links that allow for quick face-to-face conversations.

Interactive Video Meetings

If you have a small group of individuals (under a few dozen) in many different locations (*think a fully remote team*), video meeting tools can be a great way to host interactive meetings. These tools will allow for discussion among individuals invited to the meeting.

You May Have Heard Of...

There are countless live video meeting tools on the market. You may have come across any number of these names:

- **WebEx**
- **join.me**
- **Skype for Business**
- **GoToMeeting**
- **Zoom**
- **Google Hangouts**

What Live Video Meeting Tools Are Bad For?

Live video meeting tools cut lag time at the expense of quality.

You shouldn't be using these tools for:

- Sending a stream to more than a few dozen people
- Hosting big or recurring events
- Creating high quality live video

E-MAIL MAILING LISTS

An **electronic mailing list** or **email list** is a special use of email that allows for widespread distribution of information to many Internet users. It is similar to a traditional mailing list – a list of names and addresses – as might be kept by an organization for sending publications to its members or customers, but typically refers to four things:

- a list of email addresses,
- the people ("subscribers") receiving mail at those addresses, thus defining a community gathered around a topic of interest.
- the publications (email messages) sent to those addresses, and
- a *reflector*, which is a single email address that, when designated as the recipient of a message, will send a copy of that message to all of the subscribers.

- HOW AUTOMATED ELECTRONIC MAIL LIST WORK
- Electronic mailing lists usually are fully or partially automated through the use of special mailing list software and a reflector address set up on a server capable of receiving email. Incoming messages sent to the reflector address are processed by the software, and, depending on their content, are acted upon internally (in the case of messages containing commands directed at the software itself) or are distributed to all email addresses subscribed to the mailing list.
- A web-based interface is often available to allow people to subscribe, unsubscribe, and change their preferences. However, mailing list servers existed long before the World Wide Web, so most also accept commands over email to a special email address. This allows subscribers (or those who want to be subscribers) to perform such tasks as subscribing and unsubscribing, temporarily halting the sending of messages to them, or changing available preferences - all via email. The common format for sending these commands is to send an email that contains simply the command followed by the name of the electronic mailing list the command pertains to. Examples: *subscribe anylist* or *subscribe anylist John Doe*.
- Electronic mailing list servers may be set to forward messages to subscribers of a particular mailing list either individually as they are received by the list server, or in digest form in which all messages received on a particular day by the list server are combined into one email that is sent once per day to subscribers. Some mailing lists allow individual subscribers to decide how they prefer to receive messages from the list server (individual or digest).

TYPES:

Announcement list

- One type of electronic mailing list is an *announcement list*, which is used primarily as a one-way conduit of information and may only be "posted to" by selected people. This may also be referred to by the term *newsletter*. Newsletter and promotional emailing lists are employed in various sectors as parts of direct marketing campaigns.
- **Discussion list**
- Another type of electronic mailing list is a *discussion list*, in which any subscriber may post. On a discussion list, a subscriber uses the mailing list to send messages to all the other subscribers, who may answer in similar fashion. Thus, actual discussion and information exchanges can happen. Mailing lists of this type are usually topic-oriented (for example, politics, scientific discussion, health problems, joke contests), and the topic may range from extremely narrow to "whatever you think could interest us". In this they are similar to Usenet newsgroups, another form of discussion group that may have an aversion to off-topic messages.

USENET NEWSGROUP CONCEPTS

A **Usenet newsgroup** is a repository usually within the Usenet system, for messages posted from many users in different locations using Internet. They are discussion groups and are not devoted to publishing news. Newsgroups are technically distinct from, but functionally similar to, discussion forums on the World Wide Web. Newsreader software is used to read the content of newsgroups.

Before the adoption of the World Wide Web, Usenet newsgroups were among the most popular Internet services, and have retained their non-commercial nature in contrast to the increasingly ad-laden web. In recent years, this form of open discussion on the Internet has lost considerable ground to individually-operated browser-accessible forums and big media social networks such as Facebook and Twitter.

Communication is facilitated by the Network News Transfer Protocol (NNTP) which allows connection to Usenet servers and data transfer over the internet. Similar to another early (yet still used) protocol SMTP which is used for email messages. NNTP allows both server-server and client-server communication - this means that newsgroups can be replicated from server to server which gives the Usenet network the ability to maintain a level of robust data persistence as a result of built-in data redundancy. However, most users will access using only the client-server commands of NNTP and in almost all cases will use a GUI for browsing as opposed to command line-based client-server communication specified in the NNT protocol.

TYPES:

Newsgroups generally come in either of two types, binary or text. There is no technical difference between the two, but the naming differentiation allows users and servers with limited facilities to minimize network bandwidth usage. Generally, Usenet conventions and rules are enacted with the primary intention of minimizing the overall amount of network traffic and resource usage. Typically, the newsgroup is focused on a particular topic of interest. A message sent for publication on a newsgroup is called a "post". Some newsgroups allow posts on a wide variety of themes, regarding anything a member chooses to discuss as on -topic, while others keep more strictly to their particular subject, frowning on off-topic posts. The news admin (the administrator of a news server) decides how long posts are kept on their server before being expired (deleted). Different servers will have different retention times for the same newsgroup; some may keep posts for as little as one or two weeks, others may hold them for many months. Some admins keep posts in local or technical newsgroups around longer than posts in other newsgroups.

Back when the early community was the pioneering computer society, the common habit seen with many posts was a notice at the end that disclosed whether the author had (or was free of) a personal interest (financial, political or otherwise) in making the post. This is rarer now, and the posts must be read more sceptically, as with other media. Privacy and phishing issues have also risen in importance.

The number of newsgroups grew from more than 100 as of 1983 to more than 110,000, but only 20,000 or so of those are active. Newsgroups vary in popularity; some newsgroups receive under a dozen posts per year while the most popular can get several thousand in under an hour.

Newsgroups are often arranged into *hierarchies*, theoretically making it simpler to find related groups. The term *top-level hierarchy* refers to the hierarchy defined by the prefix before the first dot.

The most commonly known hierarchies are the *Usenet hierarchies*. So for instance newsgroup *rec.arts.sf.starwars.games* would be in the *rec.** top-level Usenet hierarchy, where the asterisk (*) is defined as a wildcard character. There were seven original major hierarchies of Usenet newsgroups, known as the "Big 7":

- *comp.** — Discussion of computer-related topics
- *news.** — Discussion of Usenet itself
- *sci.** — Discussion of scientific subjects
- *rec.** — Discussion of recreational activities (e.g. games and hobbies)
- *soc.** — Socialising and discussion of social issues.
- *talk.** — Discussion of contentious issues such as religion and politics.
- *misc.** — Miscellaneous discussion—anything which does not fit in the other hierarchies.

These were all created in the Great Renaming of 1986–1987, before which all of these newsgroups were in the net.* hierarchy. At that time there was a great controversy over what newsgroups should be allowed

IRC

(INTERNET RELAY CHAT)

Internet Relay Chat (**IRC**) is an application layer protocol that facilitates communication in the form of text. The chat process works on a client/server networking model. IRC clients are computer programs that users can install on their system or web-based applications running either locally in the browser or on a 3rd party server. These clients communicate with chat servers to transfer messages to other clients. IRC is mainly designed for group communication in discussion forums, called channels, but also allows one-on-one communication via private messages as well as chat and data transfer, including file sharing.

Client software is available for every major operating system that supports Internet access. As of April 2011, the top 100 IRC networks served more than half a million users at a time, with hundreds of thousands of channels operating on a total of roughly 1,500 servers out of roughly 3,200 servers worldwide. IRC usage has been declining steadily since 2003, losing 60% of its users (from 1 million to about 400,000 in 2012) and half of its channels (from half a million in 2003).

IRC was created by Jarkko Ouaknine in August 1988 to replace a program called MUT (MultiUser Talk) on a BBS called OuluBox at the University of Oulu in Finland, where he was working at the Department of Information Processing Science. Jarkko intended to extend the BBS software he administered, to allow news in the Usenet style, real time discussions and similar BBS features. The first part he implemented was the chat part, which he did with borrowed parts written by his friends Jyrki Kuoppala and Jukka Pihl. The first IRC network was running on a single server named tolsun.oulu.fi. Oikarinen found inspiration in a chat system known as Bitnet Relay, which operated on the BITNET.

Jyrki Kuoppala pushed Oikarinen to ask Oulu University to free the IRC code so that it also could be run outside of Oulu, and after they finally got it released, Jyrki Kuoppala immediately installed another server. This was the first "irc network". Oikarinen got some friends at the Helsinki University and Tampere University to start running IRC servers when his number of users increased and other universities soon followed. At this time Oikarinen realized that the rest of the BBS features probably wouldn't fit in his program.

Oikarinen got in touch with people at the University of Denver and Oregon State University. They had their own IRC network running and wanted to connect to the Finnish network. They had obtained the program from one of Oikarinen's friends, Vijay Subramaniam—the first non-Finnish person to use IRC. IRC then grew larger and got used on the entire Finnish national network—Funet—and then connected to Nordunet, the Scandinavian branch of the Internet. In November 1988, IRC had spread across the Internet and in the middle of 1989, there were some 40 servers worldwide

IRCnet fork

In July 1996, after months of flame wars and discussions on the mailing list, there was yet another split due to disagreement in how the development of the ircd should evolve. Most notably, the "european" (most of those servers were in Europe) side that later named itself IRCnet argued for nick and channel delays where the EFnet side argued for timestamps. There were also disagreements about policies: the European side had started to establish a set of rules directing what IRCops could and could not do, a point of view opposed by the US side.

Most (not all) of the IRCnet servers were in Europe, while most of the EFnet servers were in the US. This event is also known as "The Great Split" in many IRC societies. EFnet has since (as of August 1998) grown and passed the number of users it had then. In the (northern) autumn of the year 2000, EFnet had some 50,000 users and IRCnet 70,000.

Modern IRC

IRC has changed much over its life on the Internet. New server software has added a multitude of new features.

- Services: Network-operated bots to facilitate registration of nicknames and channels, sending messages for offline users and network operator functions.

- Extra modes: While the original IRC system used a set of standard user and channel modes, new servers add many new modes for features such as removing color codes from text,[17] or obscuring a user's hostmask ("cloaking") to protect from denial-of-service attacks.
- Proxy detection: Most modern servers support detection of users attempting to connect through an insecure (misconfigured or exploited) proxy server, which can then be denied a connection. This proxy detection software is used by several networks, although that real time list of proxies is defunct since early 2006.

INSTANTMESSAGING

The Internet has revolutionized the way we communicate. E-mail has been the most rapidly adopted form of communication ever known. Less than two decades ago, not many people had heard of it. Now, many of us e-mail instead of writing letters or even calling people on the phone. People around the world send out billions of e-mail messages every day.

But sometimes even e-mail isn't fast enough. You might not know if a person you want to e-mail is online at that moment. Also, if you're e-mailing back and forth with someone, you usually have to click through a few steps. This is why **instant messaging** (IM) has become so popular.

With IM, you can keep a list of people you interact with. You can IM with anyone on your **buddy list** or **contact list** as long as that person is online. You type messages to each other into a small window that shows up on both of your screens.

Most IM programs provide these features:

- **Instant messages** - Send notes back and forth with a friend who is online
- **Chat** - Create a chat room with friends or co-workers
- **Web links** - Share links to your favourite Web sites
- **Video** - Send and view videos, and chat face to face with friends
- **Images** - Look at an image stored on your friend's computer
- **Sounds** - Play sounds for your friends
- **Files** - Share files by sending them directly to your friends
- **Talk** - Use the Internet instead of a phone to actually talk with friends
- **Streaming content** - Real-time or near-real-time stock quotes and news
- **Mobile capabilities** - Send instant messages from your cell phone

In this article, you will learn about the history of instant messaging and how it works. You will also learn what the major IM programs are, what makes them different from each other and what the future holds for IM.

Before the Internet became popular, a lot of people were already online. The major online services, such as **America Online** (AOL), **Prodigy** and **CompuServe**, were the main way that ordinary people could connect and communicate with each other online. Online services provide the actual interface that you use when you're connected to the service, which creates a targeted experience for users.

In the early 1990s, people began to spend more and more time on the Internet. Creative software developers designed **chat-room** software and set up chat rooms on Web servers. In a chat room, a group of people can type in messages that are seen by everyone in the "room." Instant messages are basically a chat room for just two people.

Instant messaging really exploded on the Internet scene in November 1996. That's when **Mirabilis** introduced **ICQ**, a free instant-messaging utility that anyone could use. ICQ, shorthand for "I seek you," uses a software application, called a **client**, that resides on your computer. The client communicates with an ICQ server whenever you are online and the client is running.

In 1997, AOL, considered the pioneer of the online community, gave its users the ability to talk in real time with each other through chat rooms and instant messages. In June 1998, AOL acquired Mirabilis and ICQ.

The ICQ model is the basis for most instant-messaging utilities on the market today. In the next section we'll examine how these services work.

Unit: -3(world wide web)

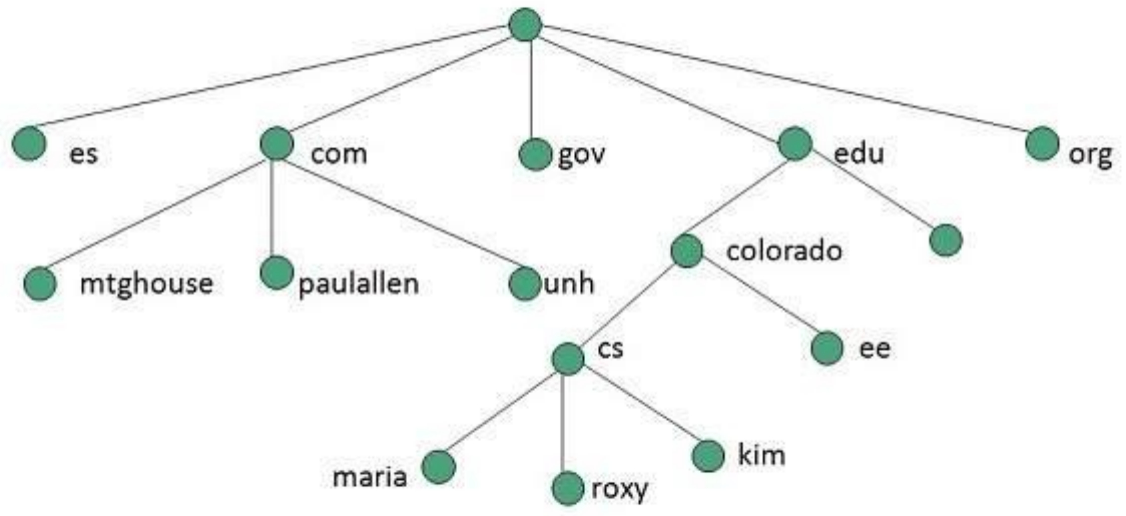
WWW stands for **World Wide Web.** A technical definition of the World Wide Web is: all the resources and users on the Internet that are using the Hypertext Transfer Protocol (HTTP).

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.

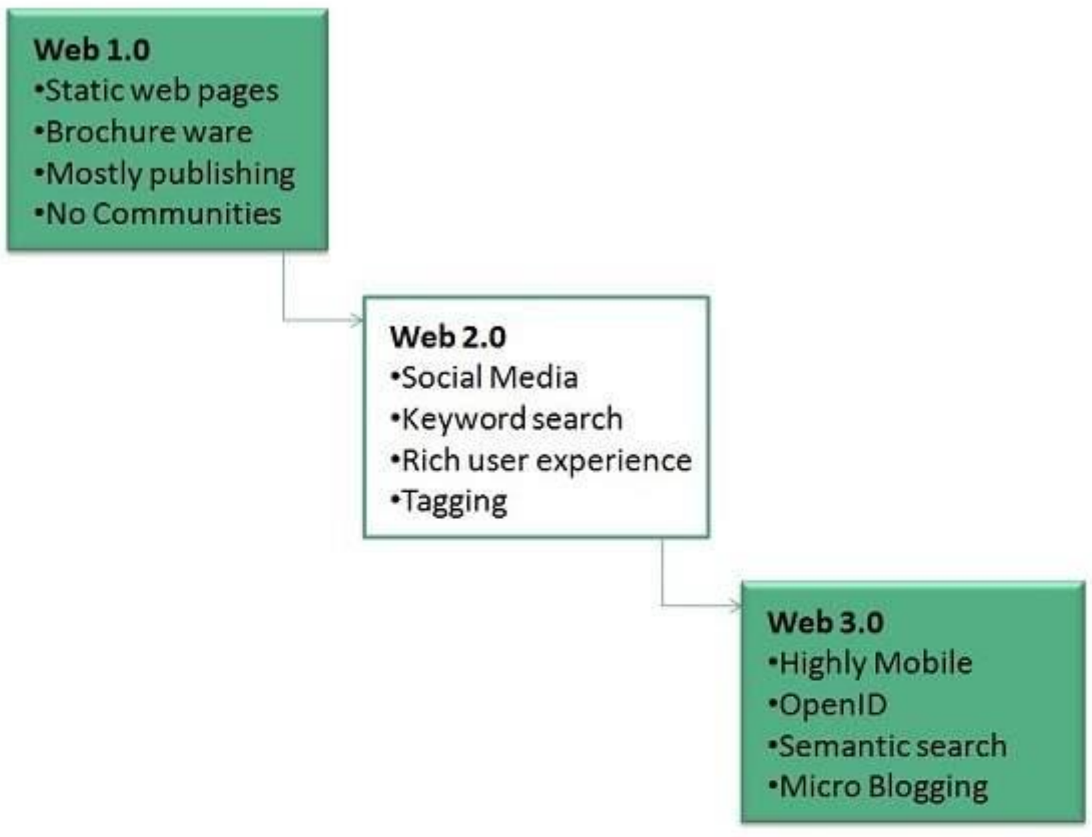
Internet and **Web** is not the same thing: Web uses internet to pass over the information.



Evolution

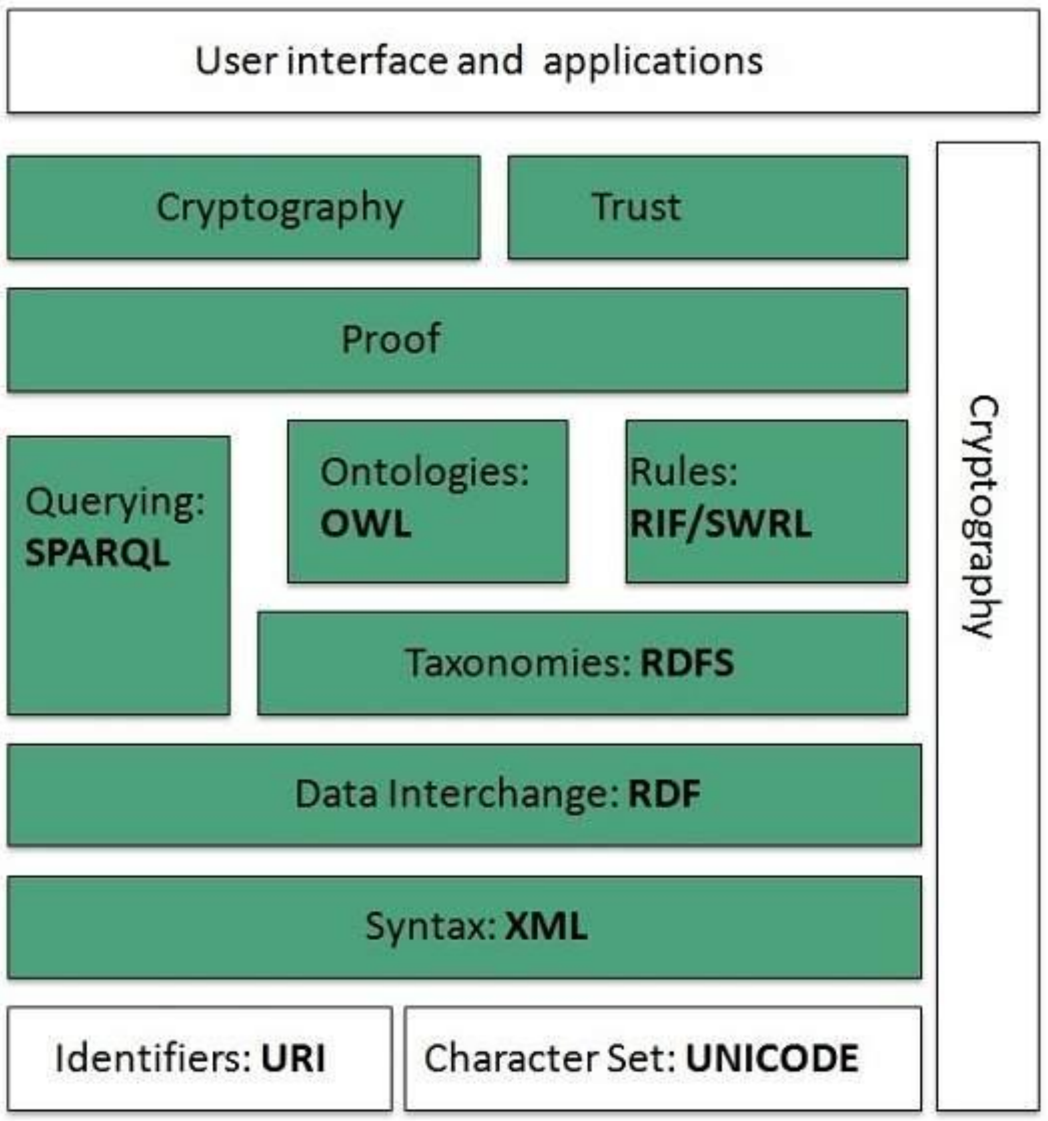
World Wide Web was created by **Timothy Berners Lee** in 1989 at **CERN** in **Geneva.** World Wide Web came into existence as a proposal by him, to allow researchers to work together effectively and efficiently at **CERN.** Eventually it became **World Wide Web.**

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 build web pages that can be read and write in human languages.

XML (Extensible Mark-up 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

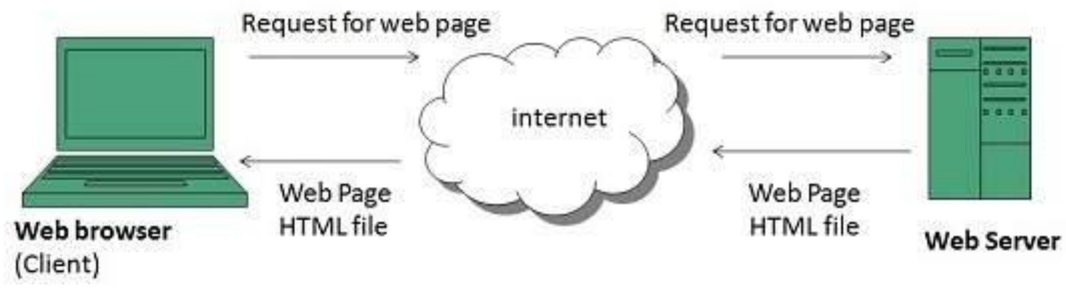
On the top of layer **User interface and Applications** layer is built for user interaction.

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.



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 development. 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 better protocol are required to be developed.

What are static and dynamic Web pages?

Web pages can be either static or dynamic. "Static" means unchanged or constant, while "dynamic" means changing or lively. Therefore, static Web pages contain the same prebuilt content each time the page is loaded, while the content of dynamic Web pages can be generated on-the-fly.
Standard HTML pages are static Web pages. They contain HTML code, which defines the structure and content of the Web page. Each time an HTML page is loaded, it looks the same. The only way the content of an HTML page will change is if the Web developer updates and publishes the file.
Other types of Web pages, such as PHP, ASP, and JSP pages are dynamic Web pages. These pages contain "server-side" code, which allows the server to generate unique content each time

the page is loaded. For example, the server may display the current time and date on the Web page. It may also output a unique response based on a Web form the user filled out. Many dynamic pages use server-side code to access database information, which enables the page's content to be generated from information stored in the database. Websites that generate Web pages from database information are often called database-driven websites.
You can often tell if a page is static or dynamic simply by looking at the page's file extension in the URL, located in the address field of the Web browser. If it is ".htm" or ".html," the page is probably static. If the extension is ".php," ".asp," or ".jsp," the page is most likely dynamic. While not all dynamic Web pages contain dynamic content, most have at least some content that is generated on-the-fly.

WEB PAGE :-A **web page** or **webpage** is a document commonly written in HTML(Hypertext Markup Language) that is accessible through the Internet or other networks using an Internet browser. A web page is accessed by entering a URL address and may contain text, graphics, and hyperlinks to other web pages and files

How to open a web page:

Viewing a web page requires a browser, like Internet Explorer, Edge, Safari, Firefox, or Chrome. For example, you are reading this web page using a browser. Once in a browser, you can open a web page by entering the URL in the address bar. For example, typing "https://www.computerhope.com/esd.htm" (or copying and pasting) opens the Computer Hope ESD page. If you don't know the URL of the site you'd like to visit, you can use a search engine to find it.

When was the first web page created?

The first web page was created at CERN by Tim Berners-Lee on August 6, 1991. You can visit and browse the first website and the first web page at the info.cern.ch address.

- The history of the Internet.
- Who invented the Internet?

Examples of a web page

The page you are reading now is an example of a web page. It comprises several web technologies, including HTML, CSS, and JavAlthough the body of a web page is created using HTML, that HTML code can be created using an HTML editor and written by a human or generated using server-side scripts or other scripts. A web page created by a human ends with a .htm or .html file extension. For example, this page has the file name "webpage.htm." Pages generated by a script can end in .cgi, .php, .pl, and other extensions.aScript

Introduction to Scripting Languages

All scripting languages are programming languages. The scripting language is basically a language where instructions are written for a run time environment. They do not require the compilation step and are rather interpreted. It brings new functions to applications and glue complex system together. A scripting language is a programming language designed for integrating and communicating with other programming languages.

There are many scripting languages some of them are discussed below:

- **bash:** It is a scripting language to work in the Linux interface. It is a lot easier to use bash to create scripts than other programming languages. It describes the tools to use and code in the command line and create useful reusable scripts and conserve documentation for other people to work with.
- **Node js:** It is a framework to write network applications using **JavaScript**. Corporate users of Node.js include IBM, LinkedIn, Microsoft, Netflix, PayPal, Yahoo for real-time web applications.
- **Ruby:** There are a lot of reasons to learn Ruby programming language. Ruby’s flexibility has allowed developers to create innovative software. It is a scripting language which is great for web development.
- **Python:** It is easy, free and open source. It supports procedure-oriented programming and object-oriented programming. Python is an interpreted language with dynamic semantics and huge lines of code are scripted and is currently the most hyped language among developers.
- **Perl:** A scripting language with innovative features to make it different and popular. Found on all windows and Linux servers. It helps in text manipulation tasks. High traffic websites that use Perl extensively include priceline.com, IMDB.

Advantages of scripting languages:

- **Easy learning:** The user can learn to code in scripting languages quickly, not much knowledge of web technology is required.
- **Fast editing:** It is highly efficient with the limited number of data structures and variables to use.

- **Interactivity:** It helps in adding visualization interfaces and combinations in web pages. Modern web pages demand the use of scripting languages. To create enhanced web pages, fascinated visual description which includes background and foreground colors and so on.
- **Functionality:** There are different libraries which are part of different scripting languages. They help in creating new applications in web browsers and are different from normal programming languages.

Application of Scripting Languages: Scripting languages are used in many areas:

- Scripting languages are used in web applications. It is used in server side as well as client side. Server-side scripting languages are: JavaScript, PHP, Perl etc. and client side scripting languages are: JavaScript, AJAX, jQuery etc.
- Scripting languages are used in system administration for example: Shell, Perl, Python scripts etc.
- It is used in Games application and Multimedia.
- It is used to create plugins and extensions for existing applications.

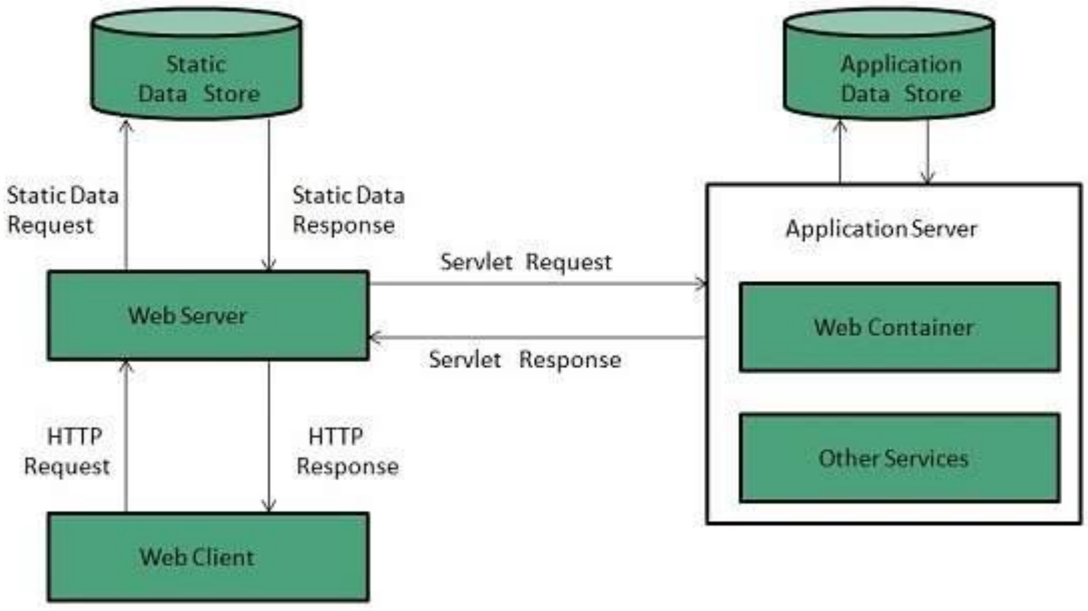
Web server is a computer where the web content is stored. Basically, web server is used to host the web sites but there exist other web servers also such as gaming, storage, FTP, email etc.

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 Description
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.	**Lighted** 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

Monitoring and Filtering

Proxy servers allow us to do several kinds of filtering such as:

- Content Filtering
- Filtering encrypted data
- Bypass filters
- Logging and eavesdropping

Improving performance

It fastens the service by process of retrieving content from the cache which was saved when previous request was made by the client.

Translation

It helps to customize the source site for local users by excluding source content or substituting source content with original local content. In this the traffic from the global users is routed to the source website through Translation proxy.

Accessing services anonymously

In this the destination server receives the request from the anonymizing proxy server and thus does not receive information about the end user.

Security

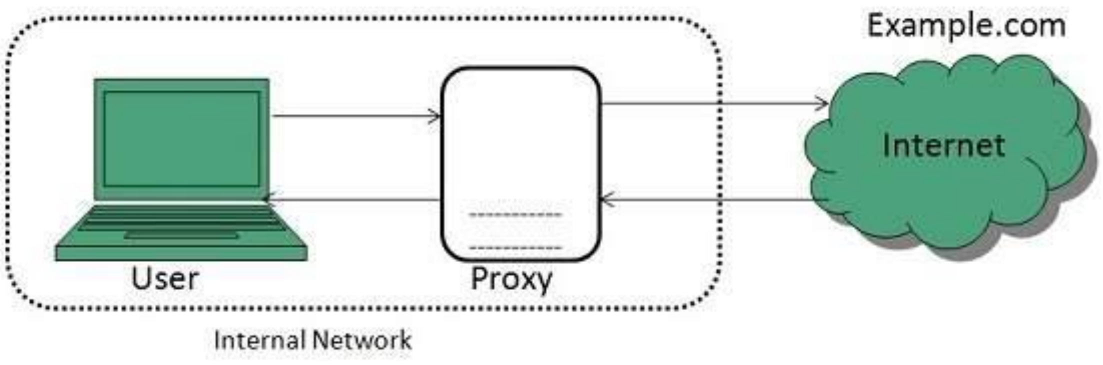
Since the proxy server hides the identity of the user hence it protects from spam and the hacker attacks.

Type of Proxies

Following table briefly describes the type of proxies:

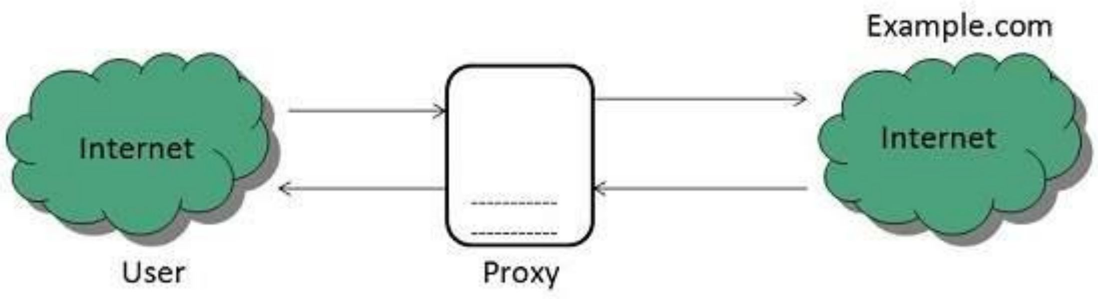
Forward Proxies

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



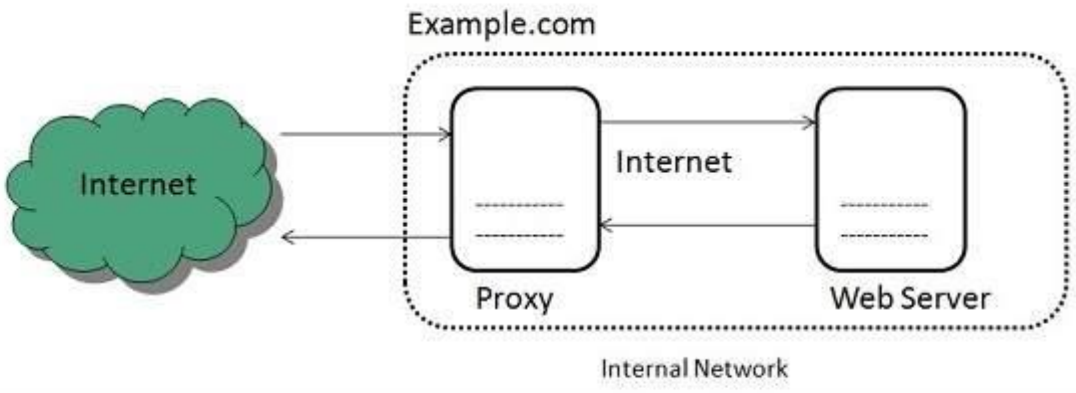
Open Proxies

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



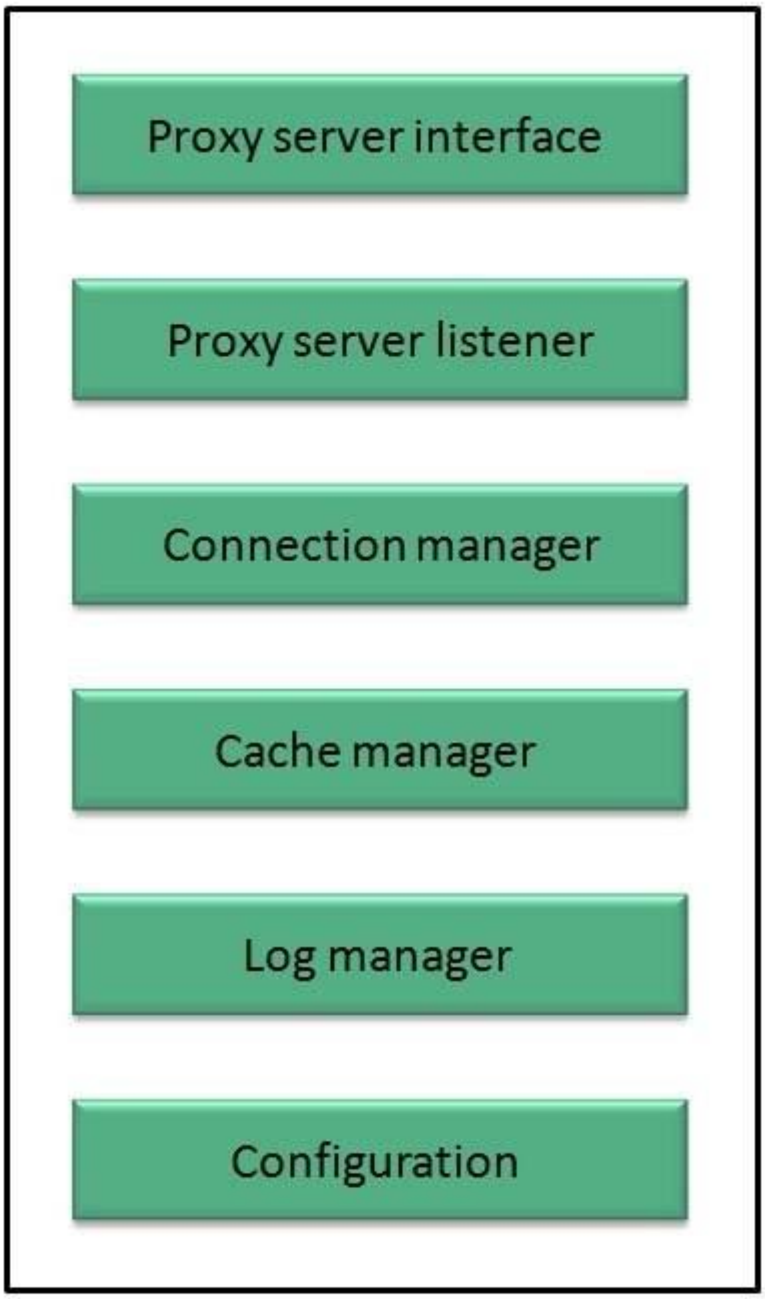
Reverse Proxies

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.

WEB DESIGNING

Web design is the process of creating websites. It encompasses several different aspects, including webpage layout, content production, and graphic design. While the terms web design and web development are often used interchangeably, web design is technically a subset of the broader category of web development.

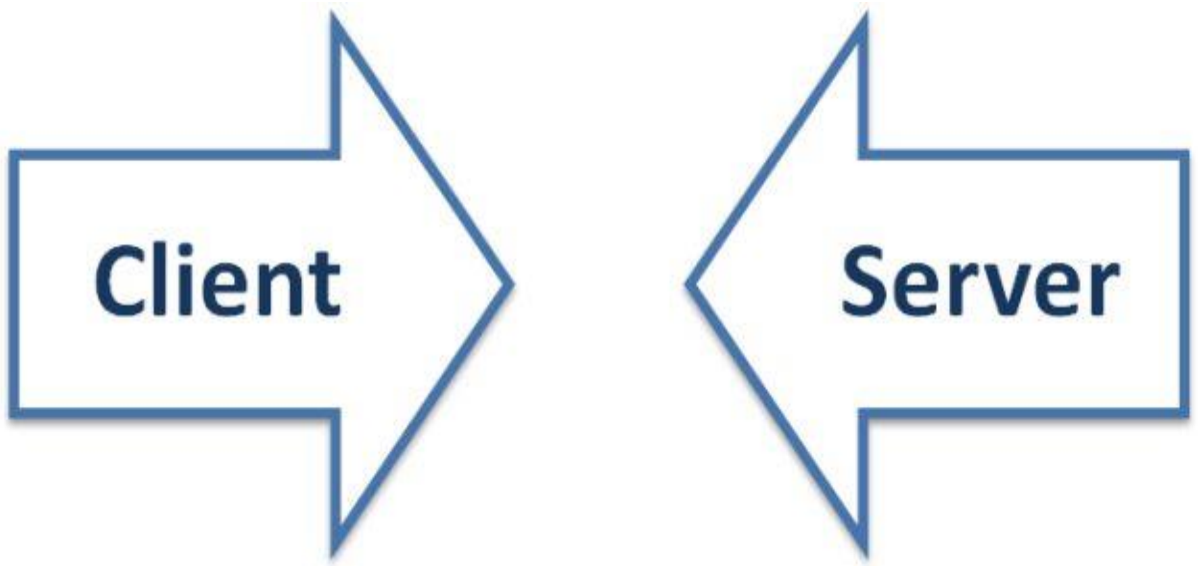
Websites are created using a markup language called HTML. Web designers build webpages using HTML tags that define the content and metadata of each page. The layout and appearance of the elements within a webpage are typically defined using CSS, or cascading style sheets. Therefore, most websites include a combination of HTML and CSS that defines how each page will appear in a browser.

Some web designers prefer to hand code pages (typing HTML and CSS from scratch), while others use a "WYSIWYG" editor like Adobe Dreamweaver. This type of editor provides a visual interface for designing the webpage layout and the software automatically generates the corresponding HTML and CSS code. Another popular way to design websites is with a content management system like WordPress or Joomla. These services provide different website templates that can be used as a starting point for a new website. Web masters can then add content and customize the layout using a web-based interface.

While HTML and CSS are used to design the look and feel of a website, images must be created separately. Therefore, graphic design may overlap with web design, since graphic designers often create images for use on the Web. Some graphics programs like Adobe Photoshop even include a "Save for Web…" option that provides an easy way to export images in a format optimized for web publishing.

CLIENT SIDE AND SERVER-SIDE LANGUAGE

Web development is all about communication and data exchange. This communication takes place via two parties over the HTTP protocol.
These parties are:



Server

The Server is responsible for serving the web pages depending on the client/end user requirement. It can be either static or dynamic.

Client

A client is a party that requests pages from the server and displays them to the end user. In general a client program is a web browser.

Example | Working

We can explain this entire mechanism using the following:

- The user opens his web browser (client)
- The user starts browsing

  (for example: http://c-sharpcorner.com)

- The client forwards this request to the server, for accessing their web page.
- The server then acknowledges the request and replies back to the client program.

  (An access link to that web page)

- The client then receives the page source and renders it.

  (Into a viewable/under a stable website)

- Now the user types into search bar
- The client then submits data to the server
- The server processes the data and replies back with a related search result
- The client again renders it back for the user's view
- The user gets access to the requested link.

Server-side Programming

Server-side programming can be explained as:

It is the general name for the kind of program that runs directly on the server.

Or we can say that server-side programming must deal with dynamic content. It runs on the server. Most web pages are not static since they deal with searching databases.

Server-side Uses

- It processes the user input
- Displays the requested pages
- Structure web applications
- Interaction with servers/storages
- Interaction with databases
- Querying the database
- Encoding of data into HTML

  Operations over databases like delete, update.

Server-side Languages Example

There are several languages that can be used for server-side programming:

- PHP
- ASP.NET (C# OR Visual Basic)
- C++
- Java and JSP
- Python
- Ruby on Rails and so on.

Server-side Example

```
// This is a sample C# code.
using System;
// namespace
class ServerSide
{
    public static void Main()
    {
        System.Console.WriteLine("Hello C# Corner");
        // printing a line
    }
}
```

Client-side Programming

Similarly, to server-side programming, client-side programming is also the name of the entire program that runs on the client.

Or we can say that client-side programming mostly deals with the user interface with which the user interacts in the web. It is mostly a browser, in the user's machine, that runs the code and is mainly done in any scripting language like JavaScript (or we can use Flash instead of JavaScript or VNScript).

Client-side Uses

- Makes interactive web pages
- Make stuffs work dynamically
- Interact with temporary storage
- Works as an interface between user and server
- Sends requests to the server
- Retrieval of data from Server
- Interact with local storage
- Provides remote access for client server program

Client-side Languages Example

There are many client-side scripting languages too.

- JavaScript
- VBScript
- HTML (Structure)
- CSS (Designing)

- AJAX
- jQuery etc.

(Some other languages also can be used on the basis of the modeling/designing /graphics/animations and for extra functionalities.)

Client-side Example

```
// sample HTML code
<html>
<head>
    <title>Client Side </title>
</head>
<body>
    <h1>
        Hello C# Corner
    </h1>
</body>
</html>
```

Website development phases:

There are numerous steps in the web site design and development process. From gathering initial information, to the creation of your web site, and finally to maintenance to keep your web site up to date and current.

The exact process will vary slightly from designer to designer, but the basics are the same.

1. Information Gathering
2. Planning
3. Design
4. Development
5. Testing and Delivery
6. Maintenance

PHASE 1

INFORMATION GATHERING

The first step in designing a successful web site is to gather information. Many things need to be taken into consideration when the look and feel of your site is created.

This first step is actually the most important one, as it involves a solid understanding of the company it is created for. **It involves a good understanding of *you*** – what your business goals and dreams are, and how the web can be utilized to help you achieve those goals.

It is important that your web designer start off by asking a lot of questions to help them understand your business and your needs in a web site. Certain things to consider are:

- **Purpose**
  What is the purpose of the site? Do you want to provide information, promote a service, sell a product… ?
- **Goals**
  What do you hope to accomplish by building this web site? Two of the more common goals are either to make money or share information.

- **Target Audience**
  Is there a specific group of people that will help you reach your goals? It is helpful to picture the "ideal" person you want to visit your web site. Consider their age,interests – this will later help determine THE content .
  What kind of information will the target audience be looking for on your site? Are they looking for specific information, a particular product or service, online ordering…?

Phase Two: Planning

Using the information gathered from phase one, it is time to put together a plan for your web site. This is the point where a site map is developed.

The site map is a list of all main topic areas of the site, as well as sub-topics, if applicable. This serves as a guide as to what content will be on the site, and is essential to developing a consistent, easy to understand navigational system. The end-user of the web site – aka your customer – must be kept in mind when designing your site. These are, after all, the people who will be learning about your service or buying your product. A good user interface creates an easy to navigate web site, and is the basis for this.

During the planning phase, your web designer will also help you decide what technologies should be implemented. Elements such as what CMS (content management system) such as WordPress to incorporate, will any contact forms be needed, etc. are discussed when planning your web site.

Phase Three: Design

Drawing from the information gathered up to this point, it's time to determine the look and feel of your site.

Target audience is one of the key factors taken into consideration. A site aimed at teenagers, for example, will look much different than one meant for a financial institution. As part of the design phase, it is also important to incorporate elements such as the company logo or colors to help strengthen the identity of your company on the web site.

Your web designer will create one or more prototype designs for your web site. This is typically a .jpg image of what the final design will look like. Often times you will be sent an email with the mock-ups for your web site, while other designers take it a step further by giving you access to a secure area of their web site meant for customers to view work in progress.

Either way, your designer should allow you to view your project throughout the design and development stages. The most important reason for this is that it gives you the opportunity to express your likes and dislikes on the site design.

In this phase, communication between both you and your designer is crucial to ensure that the final web site will match your needs and taste. It is important that you work closely with your designer, exchanging ideas, until you arrive at the final design for your web site.

Then development can begin…

Phase Four: Development

The developmental stage is the point where the web site itself is created. At this time, your web designer will take all of the individual graphic elements from the prototype and use them to create the actual, functional site.

This is typically done by first developing the home page, followed by a "shell" for the interior pages. The shell serves as a template for the content pages of your site, as it contains the main navigational structure for the web site. Once the shell has been created, your designer will take your content and distribute it throughout the site, in the appropriate areas.

Elements such as the CMS (content management system) like Word Press, interactive contact forms, or ecommerce shopping carts are implemented and made functionThis entire time, your designer should continue to make your in-progress web site available to you for viewing, so that you can suggest any additional changes or corrections you would like to have done.

On the technical front, a successful web site requires an understanding of front-end web development. This involves writing valid HTML / CSS code that complies to current web standards, maximizing functionality, as well as accessibility for as large an audience as possible.

This is tested in the next phase…

Phase Five: Testing and Delivery

At this point, your web designer will attend to the final details and test your web site. They will test things such as the complete functionality of forms or other scripts, as well last testing for last minute compatibility issues (viewing differences between different web browsers), ensuring that your web site is optimized to be viewed properly in the most recent browser versions.

A good web designer is one who is well versed in current standards for web site design and development. The basic technologies currently used are HTML and CSS (Cascading Style Sheets). As part of testing, your designer should check to be sure that all of the code written for your web site validates. Valid code means that your site meets the current web development standards – this is helpful when checking for issues such as cross-browser compatibility as mentioned above.

Unit: - 4(HTML)

Html (Hypertext mark-up language)

HTML BASIC CONCEPTS: -

What is HTML?

HTML is the standard markup language for creating Web pages.

- HTML stands for Hyper Text Markup Language
- HTML describes the structure of a Web page
- HTML consists of a series of elements
- HTML elements tell the browser how to display the content
- HTML elements are represented by tags
- HTML tags label pieces of content such as "heading", "paragraph", "table", and so on
- Browsers do not display the HTML tags, but use them to render the content of the page

Simple HTML Document/structure of html document

Example

```
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
</head>
<body>

<h1>My First Heading</h1>
<p>My first paragraph.</p>

</body>
</html>
```

DEFININIG TERMS:-

- The <!DOCTYPE html> declaration defines this document to be HTML5
- The <html> element is the root element of an HTML page
- The <head> element contains meta information about the document
- The <title> element specifies a title for the document

- The <body> element contains the visible page content
- The <h1> element defines a large heading
- The <p> element defines a paragraph

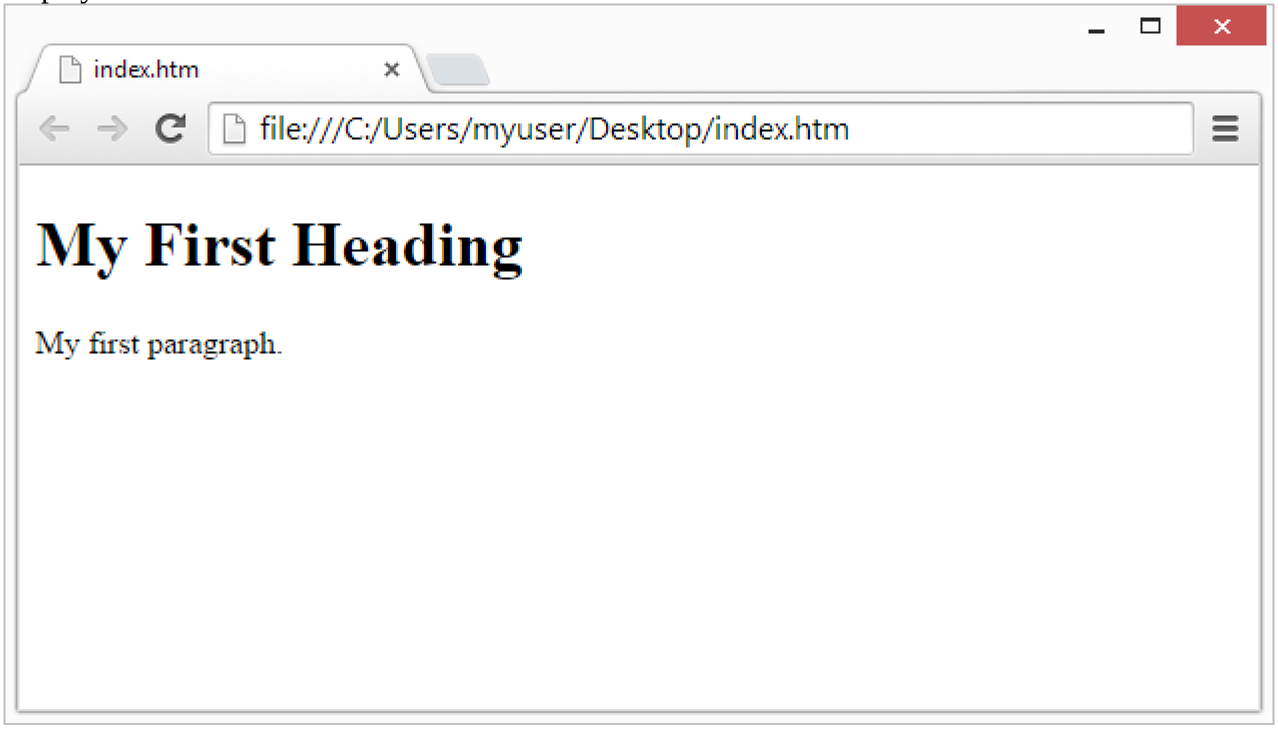
HTML Tags

HTML tags are element names surrounded by angle brackets:

<tagname>content goes here...</tagname>

- HTML tags normally come **in pairs** like <p> and </p>
- The first tag in a pair is the **start tag,** the second tag is the **end tag**
- The end tag is written like the start tag, but with a **forward slash** inserted before the tag name

- Web Browsers
- The purpose of a web browser (Chrome, Edge, Firefox, Safari) is to read HTML documents and display them.
- The browser does not display the HTML tags, but uses them to determine how to display the document:



My First Heading

My first paragraph.

- HTML Page Structure
- Below is a visualization of an HTML page structure:
- <html>
- <head>
- <title>Page title</title>
- </head>
- <body>
- <h1>This is a heading</h1>
- <p>This is a paragraph.</p>
- <p>This is another paragraph.</p>

- </body>
- </html>
- **Note:** Only the content inside the <body> section (the white area above) is displayed in a browser.

- The <!DOCTYPE> Declaration
- The <!DOCTYPE> declaration represents the document type, and helps browsers to display web pages correctly.
- It must only appear once, at the top of the page (before any HTML tags).
- The <!DOCTYPE> declaration is not case sensitive.
- The <!DOCTYPE> declaration for HTML5 is:
- <!DOCTYPE html>

- HTML History
  Since the early days of the World Wide Web, there have been many versions of HTML:

Year	**Version**
1989	Tim Berners-Lee invented www
1991	Tim Berners-Lee invented HTML
1993	Dave Raggett drafted HTML+
1995	HTML Working Group defined HTML 2.0
1997	W3C Recommendation: HTML 3.2
1999	W3C Recommendation: HTML 4.01
2000	W3C Recommendation: XHTML 1.0

2008	WHATWG HTML5 First Public Draft
2012	WHATWG HTML5 Living Standard
2014	W3C Recommendation: HTML5
2016	W3C Candidate Recommendation: HTML 5.1
2017	W3C Recommendation: HTML5.1 2nd Edition
2017	W3C Recommendation: HTML5.2

- HERE we use the latest HTML 5 standard.

HTML Elements

An HTML element usually consists of a **start** tag and an **end** tag, with the content inserted in between:

<tag name>Content goes here...</tag name>

The HTML **element** is everything from the start tag to the end tag:

<p>My first paragraph. </p>

Start tag	Element content	En
<h1>	My First Heading	</h

<p>	My first paragraph.	</p
<br>		

HTML elements with no content are called empty elements. Empty elements do not have an end tag, such as the <br> element (which indicates a line break).

Nested HTML Elements

HTML elements can be nested (elements can contain elements).

All HTML documents consist of nested HTML elements.

This example contains four HTML elements:

EXAMPLE:- <!DOCTYPE html>
```
<html>
<body>

<h1>My First Heading</h1>
<p>My first paragraph.</p>

</body>
</html>
```

Example Explained

The <html> element defines the **whole document**.

It has a **start** tag <html> and an **end** tag </html>.

Inside the <html> element is the <body> element.

```
<html>
<body>

<h1>My First Heading</h1>
<p>My first paragraph.</p>

</body>
</html>
```

The <body> element defines the **document body**.

It has a **start** tag <body> and an **end** tag </body>.

Inside<body>

<h1>My First Heading</h1>
<p>My first paragraph.</p>

</The <h1> element defines a **heading**.

It has a **start** tag <h1> and an **end** tag </h1>.

The element **content** is: My First Heading.

<body>

the <body> element is two other HTML elements:

<h1> and <p>.<h1>My First Heading</h1>

The <p> element defines a **paragraph**.

It has a **start** tag <p> and an **end** tag </p>.

The element **content** is: My first paragraph.

<p>My first paragraph. </p>

Empty elements can be "closed" in the opening tag like this: <br />.

HTML5 does not require empty elements to be closed. But if you want stricter validation, or if you need to make your document readable by XML parsers, you must close all HTML elements properly.

H

TML Is Not Case Sensitive

H

TML tags are not case sensitive: <P> means the same as <p>.

The HTML5 standard does not require lowercase tags, but W3C **recommends** lowercase in HTML, and **demands** lowercase for stricter document types like XHTML.

HTML Attributes

- All HTML elements can have **attributes**
- Attributes provide **additional information** about an element
- Attributes are always specified in **the start tag**
- Attributes usually come in name/value pairs like: **name="value"**

The href Attribute

HTML links are defined with the <a> tag. The link address is specified in the href attribute:

Example

```
<a href="https://www.w3schools.com">This is a link</a>
```

The src Attribute

HTML images are defined with the <img> tag.

The filename of the image source is specified in the src attribute:

Example

```
<img src="img_girl
```

The width and height Attributes

HTML images also have width and height attributes, which specifies the width and height of the image:

Example

```
<img src="img_girl.jpg" width="500" height="600">
```

The width and height are specified in pixels by default; so width="500" means 500 pixels wide

The alt Attribute

The alt attribute specifies an alternative text to be used, if an image cannot be displayed.

The value of the alt attribute can be read by screen readers. This way, someone "listening" to the webpage, e.g. a vision impaired person, can "hear" the element.

Example

```
<img src="img_girl.jpg" alt="Girl with a jacket">
```

The alt attribute is also useful if the image cannot be displayed (e.g. if it does not exist):

Example

See what happens if we try to display an image that does not exist:

```
<img src="img_typo.jpg" alt="Girl with a jacket">
```

Global Event Attributes

HTML has the ability to let events trigger actions in a browser, like starting a JavaScript when a user clicks on an element.

To learn more about programming events, please visit our JavaScript tutorial.

Below are the global event attributes that can be added to HTML elements to define event actions.

CORE EVENTS

Window Event Attributes

Events triggered for the window object (applies to the <body> tag):

Attribute	Value	Description
onafterprint	*script*	Script to be run after the document is printed
onbeforeprint	*script*	Script to be run before the document is printed
onbeforeunload	*script*	Script to be run when the document is about to be unloaded
onerror	*script*	Script to be run when an error occurs
onhashchange	*script*	Script to be run when there has been changes to the anchor part of the a
onload	*script*	Fires after the page is finished loading
Onmessage	*script*	Script to be run when the message is triggered

onoffline	*script*	Script to be run when the browser starts to work offline
ononline	*script*	Script to be run when the browser starts to work online
Onpagehide	*script*	Script to be run when a user navigates away from a page
onpageshow	*script*	Script to be run when a user navigates to a page
Onpopstate	*script*	Script to be run when the window's history changes
onresize	*script*	Fires when the browser window is resized
Onstorage	*script*	Script to be run when a Web Storage area is updated
onunload	*script*	Fires once a page has unloaded (or the browser window has been close

Form Events

Events triggered by actions inside a HTML form (applies to almost all HTML elements, but is most used in form elements):

Attribute	**Value**	**Description**

onblur	*script*	Fires the moment that the element loses focus
onchange	*script*	Fires the moment when the value of the element is changed
oncontextmenu	*script*	Script to be run when a context menu is triggered
onfocus	*script*	Fires the moment when the element gets focus
oninput	*script*	Script to be run when an element gets user input
oninvalid	*script*	Script to be run when an element is invalid
onreset	*script*	Fires when the Reset button in a form is clicked
onsearch	*script*	Fires when the user writes something in a search field (for <input="sea
onselect	*script*	Fires after some text has been selected in an element
onsubmit	*script*	Fires when a form is submitted

Keyboard Events

Attribute	Value	Description

onkeydown	*script*	Fires when a user is pressing a key
onkeypress	*script*	Fires when a user presses a key
onkeyup	*script*	Fires when a user releases a key

Mouse Events

Attribute	**Value**	**Description**
onclick	*script*	Fires on a mouse click on the element
ondblclick	*script*	Fires on a mouse double-click on the element
onmousedown	*script*	Fires when a mouse button is pressed down on an element
onmousemove	*script*	Fires when the mouse pointer is moving while it is over an element
onmouseout	*script*	Fires when the mouse pointer moves out of an element
onmouseover	*script*	Fires when the mouse pointer moves over an element
onmouseup	*script*	Fires when a mouse button is released over an element

onmousewheel	*script*	Deprecated. Use the onwheel attribute instead
onwheel	*script*	Fires when the mouse wheel rolls up or down over an element

Drag Events

Attribute	Value	Description
ondrag	*script*	Script to be run when an element is dragged
ondragend	*script*	Script to be run at the end of a drag operation
ondragenter	*script*	Script to be run when an element has been dragged to a valid drop targ
ondragleave	*script*	Script to be run when an element leaves a valid drop target
ondragover	*script*	Script to be run when an element is being dragged over a valid drop tar
ondragstart	*script*	Script to be run at the start of a drag operation
ondrop	*script*	Script to be run when dragged element is being dropped
onscroll	*script*	Script to be run when an element's scrollbar is being scrolled

Clipboard Events

Attribute	**Value**	**Description**
oncopy	*script*	Fires when the user copies the content of an element
oncut	*script*	Fires when the user cuts the content of an element
onpaste	*script*	Fires when the user pastes some content in an element

Media Events

Events triggered by medias like videos, images and audio (applies to all HTML elements, but is most common in media elements, like <audio>, <embed>, <img>, <object>, and <video>).

BLOCK LEVEL EVENTS

All the HTML elements can be categorized into two categories **(a)** Block Level Elements **(b)**Inline Elements.

Block Elements

Block elements appear on the screen as if they have a line break before and after them. For example, the <p>, <h1>, <h2>, <h3>, <h4>, <h5>, <h6>, <ul>, <ol>, <dl>, <pre>, <hr />, <blockquote>, and <address> elements are all block level elements. They all start on their own new line, and anything that follows them appears on its own new line.

Inline Elements

Inline elements, on the other hand, can appear within sentences and do not have to appear on a new line of their own. The <b>, <i>, <u>, <em>, <strong>, <sup>, <sub>, <big>, <small>, <li>, <ins>, <del>, <code>, <cite>, <dfn>, <kbd>, and <var> elements are all inline elements.

Grouping HTML Elements

There are two important tags which we use very frequently to group various other HTML tags (i) <div> tag and (ii) <span> tag

The <div> tag

This is the very important block level tag which plays a big role in grouping various other HTML tags and applying CSS on group of elements. Even now <div> tag can be used to create webpage layout where we define different parts (Left, Right, Top etc.) of the page using <div> tag. This tag does not provide any visual change on the block but this has more meaning when it is used with CSS.

Example

Following is a simple example of <div> tag. We will learn Cascading Style Sheet (CSS) in a separate chapter but we used it here to show the usage of <div> tag –

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

  <head>
    <title>HTML div Tag</title>
  </head>

  <body>
    <!-- First group of tags -->
    <div style = "color:red">
      <h4>This is first group</h4>
      <p>Following is a list of vegetables</p>

      <ul>
        <li>Beetroot</li>
        <li>Ginger</li>
        <li>Potato</li>
        <li>Radish</li>
      </ul>
    </div>

    <!-- Second group of tags -->
    <div style = "color:green">
      <h4>This is second group</h4>
      <p>Following is a list of fruits</p>

      <ul>
        <li>Apple</li>
        <li>Banana</li>
        <li>Mango</li>
        <li>Strawberry</li>
      </ul>
    </div>
  </body>

</html>
```

This will produce the following result –

The <span> tag

The HTML <span> is an inline element and it can be used to group inline-elements in an HTML document. This tag also does not provide any visual change on the block but has more meaning when it is used with CSS.

The difference between the <span> tag and the <div> tag is that the <span> tag is used with inline elements whereas the <div> tag is used with block-level elements.

Example

Following is a simple example of <span> tag. We will learn Cascading Style Sheet (CSS) in a separate chapter but we used it here to show the usage of <span> tag −

```
<!DOCTYPE html>
<html>

   <head>
      <title>HTML span Tag</title>
   </head>

   <body>
      <p>This is <span style = "color:red">red</span> and this is
         <span style = "color:green">green</span></p>
   </body>

</html>
```

This will produce the following result −

This is red and this is green

Html text level events

These don't cause paragraph breaks. Text level elements that define character styles can generally be nested. They can contain other text level elements but not block level elements.

Font style elements

These all require start and end tags, e.g.

This has some <B>bold text</B>.

TT teletype or monospaced text

The quick red fox jumped over the lazy brown dog.

I italic text style

The quick red fox jumped over the lazy brown dog.

B bold text style

The quick red fox jumped over the lazy brown dog.

U underlined text style

The quick red fox jumped over the lazy brown dog.

STRIKE strike-through text style

~~The quick red fox jumped over the lazy brown dog.~~

BIG places text in a large font

The quick red fox jumped over the lazy brown dog.

SMALL places text in a small font

The quick red fox jumped over the lazy brown dog.

SUB places text in subscript style

The fox$_{\text{red}}$ jumped over the dog$_{\text{brown}}$.

SUP places text in superscript style

The fox$^{\text{quick}}$ jumped over the dog$^{\text{lazy}}$.

Phrase Elements

These all require start and end tags, e.g.

This has some <EM>emphasized text</EM>.

EM basic emphasis typically rendered in an italic font

The quick red fox jumped over the lazy brown dog.

STRONG strong emphasis typically rendered in a bold font

The quick red fox jumped over the lazy brown dog.

DFN defining instance of the enclosed term

The quick red fox jumped over the lazy brown dog.

CODE used for extracts from program code

The quick red fox jumped over the lazy brown dog.

SAMP used for sample output from programs, and scripts etc.

The quick red fox jumped over the lazy brown dog.

KBD used for text to be typed by the user

The quick red fox jumped over the lazy brown dog.

VAR used for variables or arguments to commands

The quick red fox jumped over the lazy brown dog.

CITE used for citations or references to other sources

The quick red fox jumped over the lazy brown dog.

Form fields

INPUT, SELECT and TEXTAREA

INPUT elements are not containers and so the end tag is forbidden. INPUT, SELECT and TEXTAREA are only allowed within FORM elements. INPUT can be used for a variety of form fields including

- single line text fields,
- password fields,
- checkboxes,
- radio buttons,
- submit and reset buttons,
- hidden fields,
- file upload, and
- image buttons.

SELECT elements require start and end tags and contain one or more OPTION elements. SELECT elements are used for single or multi-selection menus. TEXTAREA elements require start and end tags, and are used to define multi-line text fields. The content of the element is used to initialize the field.

Special Text level Elements

Anchors, IMG, APPLET, FONT, BR and MAP.

The A (anchor) element

Used to define hypertext links and also to define named locations for use as targets for hypertext links, e.g.

```
  The way to <a href="kamasutra.html">happiness</a>.
```

The attributes are: NAME, HREF, REL, REV and TITLE. HREF is used to supply a URL identifying the linked document or image etc. NAME is used to associate a name with this part of a document for use with URLs that target a named section of a document. Anchors can't be nested.

IMG

e.g. <IMG SRC="canyon.gif" ALT="Grand Canyon">

Used to insert images. This is an empty element and so the end tag is forbidden. The attributes are: SRC, ALT, ALIGN, WIDTH, HEIGHT, BORDER, HSPACE, VSPACE, USEMAP and ISMAP. Images can be positioned vertically relative to the current textline or floated to the left or right. See BR with the CLEAR attribute for control over textflow.

APPLET

Requires start and end tags. This element is supported by all Java enabled browsers. It allows you to embed a Java applet into HTML documents, e.g. to include an animation. The contents of the element are used as a fallback if the applet can't be loaded. The attributes are: CODE, CODEBASE, NAME, ALT, ALIGN, WIDTH, HEIGHT, HSPACE and VSPACE. APPLET uses associated PARAM elements to pass parameters to the applet.

FONT

Requires start and end tags. This allows you to change the font size and/or color for the enclosed text. The attributes are: SIZE, COLOR. Colors are given as RGB in hexadecimal notation or as one of 16 widely understood color names.

BR

Used to force a line break. This is an empty element and so the end tag is forbidden. The CLEAR attribute can be used to move down past floating images on either margin, e.g. <BR CLEAR=LEFT>.

MAP

Requires start and end tags. This allows you to define client-side image maps. MAP elements contain one or more AREA elements that specify hotzones on the associated image and bind these hotzones to URLs.

Html linking basics :-

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 </a> 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>
```

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 *targetframe*.

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.

Use of Base Path

When you link HTML documents related to the same website, it is not required to give a complete URL for every link. You can get rid of it if you use **<base>** tag in your HTML document header. This tag is used to give a base path for all the links. So your browser will concatenate given relative path to this base path and will make a complete URL.

Example

Following example makes use of <base> tag to specify base URL and later we can use relative path to all the links instead of giving complete URL for every link.

```
<!DOCTYPE html>
<html>

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

  <body>
    <p>Click following link</p>
    <a href = "/html/index.htm" target = "_blank">HTML Tutorial</a>
  </body>

</html>
```

This will produce the following result, where you can click on the link generated **HTML Tutorial** to reach to the HTML tutorial.

Now given URL <a href = "/html/index.htm" is being considered as <ahref = "http://www.tutorialspoint.com/html/index.htm"

Linking to a Page Section

You can create a link to a particular section of a given webpage by using **name** attribute. This is a two-step process.

Note − The *name* attribute deprecated in HTML5. Do not use this attribute. Use *id* and *title* attribute instead.

First create a link to the place where you want to reach with-in a webpage and name it using <a...> tag as follows −

<h1>HTML Text Links <a name = "top"></a></h1>

Second step is to create a hyperlink to link the document and place where you want to reach −

<a href = "/html/html_text_links.htm#top">Go to the Top</a>

This will produce following link, where you can click on the link generated **Go to the Top** to reach to the top of the HTML Text Link tutorial.

Go to the Top

Setting Link Colors

You can set colors of your links, active links and visited links using **link**, **alink** and **vlink** attributes of <body> tag.

Example

Save the following in test.htm and open it in any web browser to see how **link**, **alink** and **vlink** attributes work.

```
<!DOCTYPE html>
<html>

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

  <body alink = "#54A250" link = "#040404" vlink = "#F40633">
    <p>Click following link</p>
    <a href = "/html/index.htm" target = "_blank" >HTML Tutorial</a>
  </body>

</html>
```

This will produce the following result. Just check color of the link before clicking on it, next check its color when you activate it and when the link has been visited.

Download Links

You can create text link to make your PDF, or DOC or ZIP files downloadable. This is very simple; you just need to give complete URL of the downloadable file as follows −

```
<!DOCTYPE html>
<html>

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

  <body>
    <a href = "https://www.tutorialspoint.com/page.pdf">Download PDF File</a>
  </body>

</html>
```

This will produce following link and will be used to download a file.

File Download Dialog Box

Sometimes it is desired that you want to give an option where a user will click a link and it will pop up a "File Download" box to the user instead of displaying actual content. This is very easy and can be achieved using an HTTP header in your HTTP response.

For example, if you want make a **Filename** file downloadable from a given link then its syntax will be as follows.

```
#!/usr/bin/perl

# Additional HTTP Header
print "Content-Type:application/octet-stream; name = \"FileName\"\r\n";
print "Content-Disposition:attachment; filename = \"FileName\"\r\n\n";

# Open the target file and list down its content as follows
open( FILE, "<FileName" );

while(read(FILE, $buffer, 100)){
  print("$buffer");
}
```

Note − For more detail on PERL CGI programs, go through tutorial PERL and CGI

Image and anchor tag

HTML <img> Tag

How to insert an image:

<img src="smiley.gif" alt="Smiley face" height="42" width="42">

Definition and Usage

The <img> tag defines an image in an HTML page.

The <img> tag has two required attributes: src and alt.

Note: Images are not technically inserted into an HTML page, images are linked to HTML pages. The <img> tag creates a holding space for the referenced image.

HTML <a> Tag

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

Definition and Usage

The <a> tag defines a hyperlink, which is used to link from one page to another.

The most important attribute of the <a> element is the href attribute, which indicates the link's destination.

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

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

Anchor attribute

The **HTML <a> element** (or *anchor* element), with its href attribute, creates a hyperlink to web pages, files, email addresses, locations in the same page, or anything else a URL can address. Content within each <a> **should** indicate the link's destination.

```
<p>You can reach Michael at:</p>

<ul>

  <li><a href="https://example.com">Website</a></li>

  <li><a href="mailto:m.bluth@example.com">Email</a></li>

  <li><a href="tel:+123456789">Phone</a></li>

</ul>
```

HTML Image Maps

Image Maps

The <map> tag defines an image-map. An image-map is an image with clickable areas.

Click on the computer, the phone, or the cup of coffee in the image below:

```
<img src="workplace.jpg" alt="Workplace" usemap="#workmap">

<map name="workmap">
  <area shape="rect" coords="34,44,270,350" alt="Computer" href="computer.htm">
  <area shape="rect" coords="290,172,333,250" alt="Phone" href="phone.htm">
  <area shape="circle" coords="337,300,44" alt="Coffee" href="coffee.htm">
</map>
```

How Does it Work?

The idea behind an image map is that you should be able to perform different actions depending on where in the image you click.

To create an image map you need an image, and a map containing some rules that describe the clickable areas.

The Image

The image is inserted using the <img> tag. The only difference from other images is that you must add a usemap attribute:

```
<img src="workplace.jpg" alt="Workplace" usemap="#workmap">
```

The usemap value starts with a hash tag # followed by the name of the image map, and is used to create a relationship between the image and the image map.

Note: You can use any image as an image map.

HTML Semantic Elements

What are Semantic Elements?

A semantic element clearly describes its meaning to both the browser and the developer.

Examples of **non-semantic** elements: <div> and <span> - Tells nothing about its content.

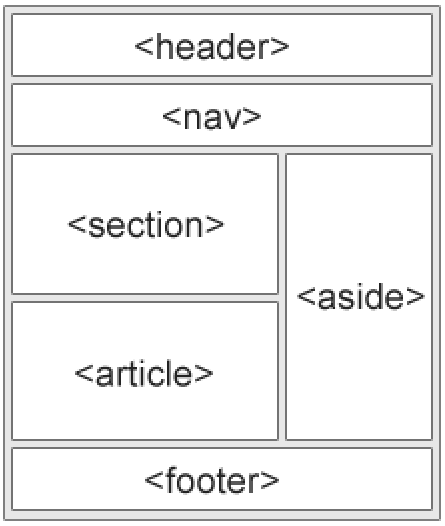
Examples of **semantic** elements: <form>, <table>, and <article> - Clearly defines its content

Semantic Elements in HTML

Many web sites contain HTML code like: <div id="nav"> <div class="header"> <div id="footer"> to indicate navigation, header, and footer.

In HTML there are some semantic elements that can be used to define different parts of a web page:

- <article>
- <aside>
- <details>
- <figcaption>
- <figure>
- <footer>
- <header>
- <main>
- <mark>
- <nav>
- <section>
- <summary>
- <time>



HTML <section> Element

The <section> element defines a section in a document.

According to W3C's HTML documentation: "A section is a thematic grouping of content, typically with a heading."

A home page could normally be split into sections for introduction, content, and contact information.

Example

```
<section>
  <h1>WWF</h1>
  <p>The World Wide Fund for Nature (WWF) is....</p>
</section>
```

Image button in html

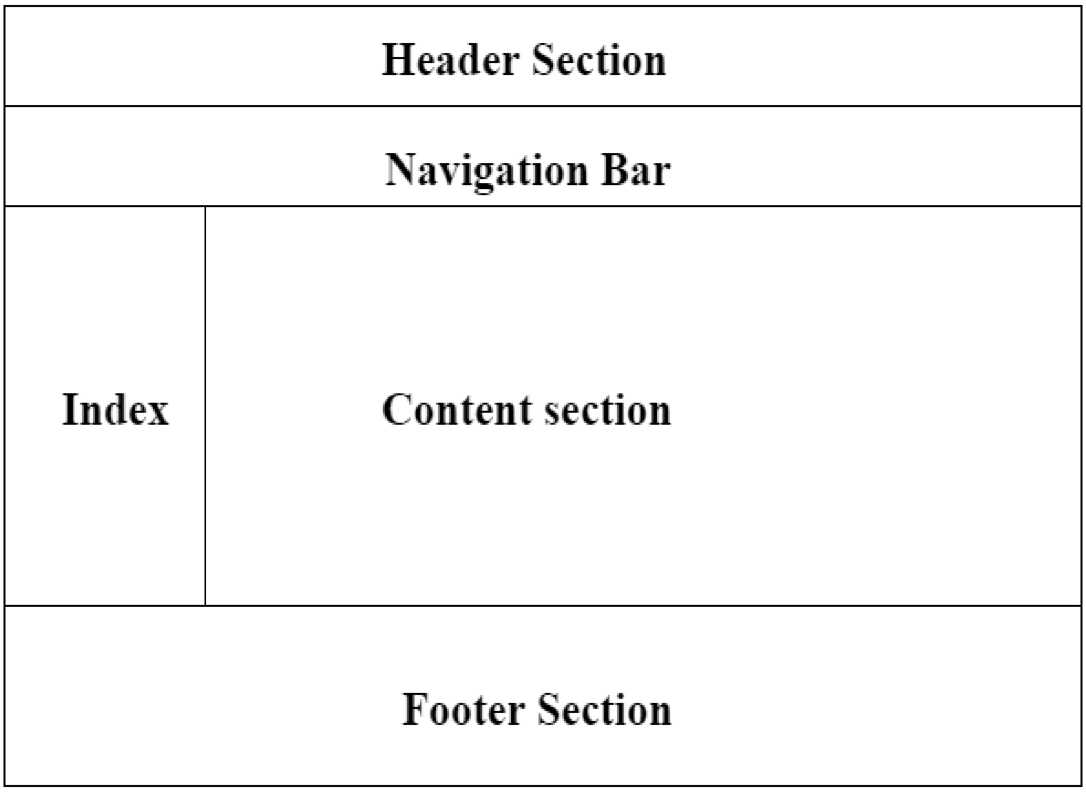
To add image button with HTML5, use the <button> element and set the image inside it before it is closed </button>:

```
<button type = "submit" name = "learn" value = "myimage">
   <p>Tutorials for all</p>
   <img src="https://www.tutorialspoint.com/latest/inter-process-communication.png " />
</button>
```

Html layout

HTML | Layout

Page layout is the part of graphic design that deals with the arrangement of visual elements on a page. Page layout is used to make the web pages look better. It establishes the overall appearance, relative importance, and relationships between the graphic elements to achieve a smooth flow of information and eye movement for maximum effectiveness or impact.



Page Layout Information:
Header: The part of a front end which is used at the top of the page. <header> tag is used to add header section in web pages.

- **Navigation bar:** The navigation bar is same as menu list. It is used to display the content information using hyperlink.
- **Index / Sidebar:** It holds additional information or advertisements and is not always necessary to be added into the page.
- **Content Section:** The content section is the main part where content is displayed.
- **Footer:** The footer section contains the contact information and other query related to web pages. The footer section always put on the bottom of the web pages. The <footer> tag is used to set the footer in web pages.

Example:
filter_none
edit
play_arrow
brightness_4

```
<!DOCTYPE html>
<html>
<head>
   <title>Page Layout</title>
   <style>
      .head1 {
         font-size:40px;
         color:#009900;
         font-weight:bold;
      }
      .head2 {
         font-size:17px;
```

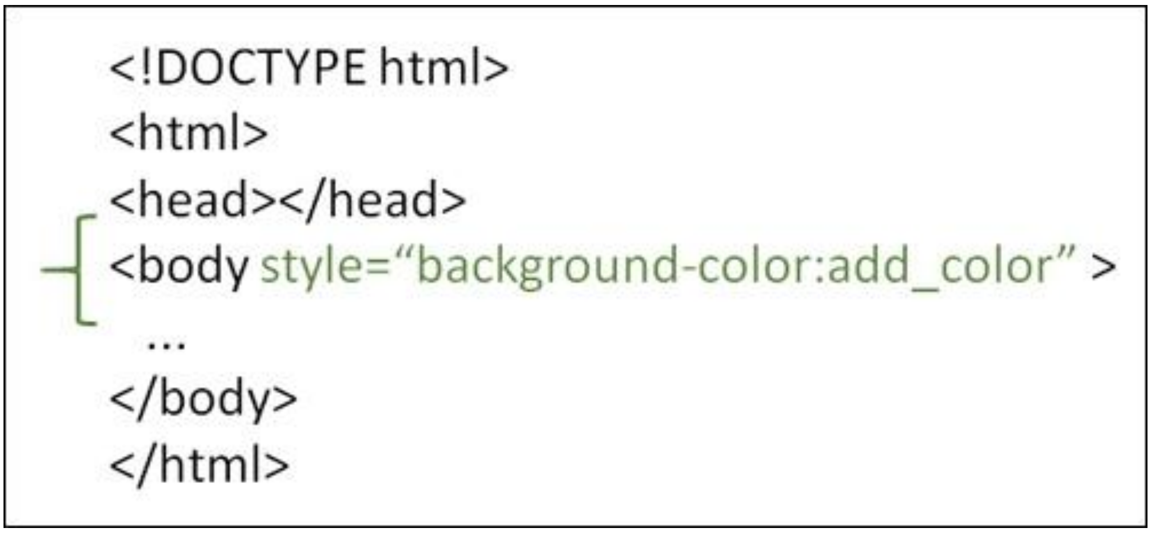
```
            margin-left:10px;
            margin-bottom:15px;
        }
        body {
            margin: 0 auto;
            background-position:center;
            background-size: contain;
        }

        .menu {
            position: sticky;
            top: 0;
            background-color: #009900;
            padding:10px 0px 10px 0px;
            color:white;
            margin: 0 auto;
            overflow: hidden;
        }
        .menu a {
            float: left;
            color: white;
            text-align: center;
            padding: 14px 16px;
            text-decoration: none;
            font-size: 20px;
        }
        .menu-log {
            right: auto;
            float: right;
        }
        footer {
            width: 100%;
            bottom: 0px;
            background-color: #000;
            color: #fff;
            position: absolute;
            padding-top:20px;
            padding-bottom:50px;
            text-align:center;
            font-size:30px;
            font-weight:bold;
        }
        .body_sec {
            margin-left:20px;
        }
    </style>
</head>

<body>
```

```
  <!-- Header Section -->
  <header>
    <div class="head1">GeeksforGeeks</div>
    <div class="head2">A computer science portal for geeks</div>
  </header>

  <!-- Menu Navigation Bar -->
  <div class="menu">
    <a href="#home">HOME</a>
    <a href="#news">NEWS</a>
    <a href="#notification">NOTIFICATIONS</a>
    <div class="menu-log">
      <a href="#login">LOGIN</a>
    </div>
  </div>

  <!-- Body section -->
  <div class = "body_sec">
    <section id="Content">
      <h3>Content section</h3>
    </section>
  </div>

  <!-- Footer Section -->
  <footer>Footer Section</footer>
</body>
</html>
```

HOW TO CHANGE TEXT AND BACKGROUND COLOR :

To set the background color in HTML, use the style attribute. The style attribute specifies an inline style for an element. The attribute is used with the HTML <body> tag, with the CSS property background-color. HTML5 do not support the <body> tag bgcolor attribute, so the CSS style is used to add background color. The bgcolor attribute deprecated in HTML5.

Just keep in mind, the usage of style attribute overrides any style set globally. It will override any style set in the HTML <style> tag or external style sheet.



```
<!DOCTYPE html>
<html>
<head></head>
<body style="background-color:add_color" >
  ...
</body>
</html>
```

Example

You can try to run the following code to set the background color in HTML –

```
<!DOCTYPE html>
<html>
  <head>
    <title>HTML Backgorund Color</title>
  </head>
  <body style="background-color:grey;">
    <h1>Products</h1>
    <p>We have developed more than 10 products till now.</p>
  </body>
</html>
```

HTML Forms

The <form> Element

The HTML <form> element defines a form that is used to collect user input:

```
<form>
.
form elements
.
</form>
```

An HTML form contains **form elements**.

Form elements are different types of input elements, like: text fields, checkboxes, radio buttons, submit buttons, and more.

T

he <input> Element

T

he <input> element is the most important form element.

The <input> element is displayed in several ways, depending on the **type** attribute.

Here are some examples:

Type	**Description**
<input type="text">	Defines a single-line text input field
<input type="radio">	Defines a radio button (for selecting one of many choices)
<input type="submit">	Defines a submit button (for submitting the form)

Using HTML Tables for Page Layout

Do You Know HTML Tables?

If you haven't looked at our Introduction to HTML Tables then head over there and then come back!

Using tables, as with using HTML Frames, to create a page layout is an old-school method of creating page layouts. That being said, if you know your target visitors are on tablets or larger screens then you could still use this method.

Another restriction of tables is that they are not innately responsive (not mobile-friendly). Using CSS you can re-defined tables in such a way as to make them mobile-friendly but it is a pain in the back side!

With those two caveats in mind let's look at how you can create a page layout with HTML Tables.

Plan Your Layout

The most important task you can do before starting any coding is plan! This applies even when we later add mobile-friendly (responsive) tutorials later. Planning your site, the website structure and the layout can save you headaches and heartaches later.

For the purposes of this tutorial, I want to keep the layout simple as a demonstration. You can then use your imagination and build whatever you want.

So here's a 'standard' layout including header, sidebar, content and footer. I have used background colours here to make the different segments stand out.

Example HTML Table Page Layout	
Home About Us Contact Us	Here we will have the page content. The home page should be an introduction to the purpose of your site. It should also encourage people to go deeper and learn more about you.
© Tutorial Resource Centre	

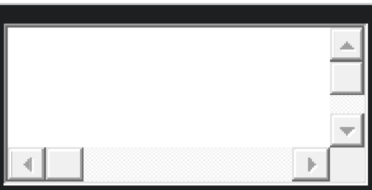
Construct Your HTML Table

When creating a layout using <table> tags, you don't need to panic. The reason is that it is just a simple table. However, you will need to bear in mind the following:

- **You need to use valign**: in this layout you must use the valign="top" on both your sidebar and your content area. The reason is that your content area will vary across pages. If you don't valign the navigation on long pages, your links will appear in the middle. If your navigation has a lot of links, but the content area is short (say, a Contact page) then your page content will end up in the middle (between the top and bottom, that is). So you pretty much always need to valign="top" on your content and navigation area
- **Check your colspans and rowspans:** As your site gets more complicated you need to verify your colspans and rowspans. If you need to put another table inside your content area this is fine, but you will need to be careful. *Always close your TABLE, TD and TR tags before you populate them*.
- **Create a 'Template File'**: Unlike HTML Frames, you don't need to use the target attribute on links. However, each page has its own table layout, its own header/footer/navigation etc. So, create yourself a template file containing all the links you intend to have. If you spot differences in layout across pages, it will be because you have something different in your HTML. And this is one of the downfalls of using HTML Tables for page layout!

The Code

Here's the code to create the above layout (without the background colours).



```
<table width="100%" border="1" cellpadding="5">
 <tr>
    <th colspan="2"><h2>Example HTML Table Page Layout</h2></th>
 </tr>
 <tr>
    <td valign="top" width="20%"><center><a href="index.html">Home</a><br><a href="a
bout.html">About Us</a><br><a href="contact.html">Contact Us</a></center></td>
    <td valign="top"><p>Here we will have the page content.</p><p>The home page should
be an introduction to the purpose of your site.</p><p>It should also encourage people to go d
eeper and learn more about you.</p></td>
 </tr>
 <tr>
    <td colspan="2">&copy; Tutorial Resource Centre</td>
 </tr>
</table>
```

Cantering Your Table

You have two options when deciding how much space your website will take up: percentage of screen size, or absolute size. If you want your site to work on most screens, then **use percentages**. If however you want to specify the width, then you can do that – e.g. `width="800"` rather than `width="100%"`.

But your table will sit on the left-hand side of the browser by default. This can give acres of whitespace to the right of your website – not a good luck. Thankfully this can be easily changed.

`<table border="1" width="100%" align="centre" cellpadding="5">`

The **align** attribute here ensures your table sits in the middle of the screen. This is important for wider screens especially.

Tables and Responsive Design

Earlier I said you shouldn’t use this layout method when making a mobile-friendly website. I still stand by that. However, you *can* get around this. The problem with mobile-friendly designs is that, unlike desktops/laptops, phones are in **portrait orientation**.

As phones are usually interacted with in portrait mode, this makes the use of sidebars pointless. So, you can get around this and make a roughly mobile-friendly design using tables if you **use percentages** and **row-based TRs** rather than column-based. e.g.

Tables and Mobile

Home – **About Us** – **Contact Us**

Your Page Content Goes Here
On mobile the **valign=" top"** is not strictly necessary.

© Tutorial Resource Centre

The main downside to this is that you need to use width="100%" on your table which will make your site hard to read on wide screens. So in theory this is possible, but it isn't ideal. I'll cover responsive design later on in our CSS3 Tutorials.

Tables and Images

The last issue I want to deal with is a difficulty in using images in a table layout. When you create an image, say in Photoshop or GIMP, it has a set width. This means you need to make sure your table is sized appropriately.

For images you are using as part of your layout – say a header banner or graphic – you have to think this through. If your site is of a fixed width – then you want your graphics to be no bigger than your fixed width. If you're using percentages however, you may need to create a graphic much bigger – so that it always fills the space.

To ensure your header graphic does not warp or distort, you can set the width="100%" but not set a height attribute. This is done like this:

```
<img src="images/header-graphic.jpg" alt="Welcome to my website" title="Welcome to my website" width="100%">
```

By not setting the height you **keep the aspect ratio the same** regardless of the size of your table cell.

If you find your images are making your tables wider, then you can reduce the width attribute until your layout returns to normal.

TL; DR HTML Tables for Page Layout

- Using Tables is an old way of creating page layouts, but is possible.
- You have to choose fixed width or percentage base when determining the size of your website.
- Bear in mind that common elements (header, footer, navigation) will need to be copied across all pages of your website.

Html frames

HTML frames are used to divide your browser window into multiple sections where each section can load a separate HTML document. A collection of frames in the browser window is known as a frameset. The window is divided into frames in a similar way the tables are organized: into rows and columns.

Disadvantages of Frames:

There are few drawbacks with using frames, so it's never recommended to use frames in your webpages −

- Some smaller devices cannot cope with frames often because their screen is not big enough to be divided up.
- Sometimes your page will be displayed differently on different computers due to different screen resolution.
- The browser's *back* button might not work as the user hopes.
- There are still few browsers that do not support frame technology.

Creating Frames

To use frames on a page we use <frameset> tag instead of <body> tag. The <frameset> tag defines, how to divide the window into frames. The **rows** attribute of <frameset> tag defines horizontal frames and **cols** attribute defines vertical frames. Each frame is indicated by <frame> tag and it defines which HTML document shall open into the frame.

Note − The <frame> tag deprecated in HTML5. Do not use this element.

Example

Following is the example to create three horizontal frames −

```
<!DOCTYPE html>
<html>

   <head>
      <title>HTML Frames</title>
   </head>

   <frameset rows = "10%,80%,10%">
      <frame name = "top" src = "/html/top_frame.htm" />
      <frame name = "main" src = "/html/main_frame.htm" />
      <frame name = "bottom" src = "/html/bottom_frame.htm" />

      <noframes>
         <body>Your browser does not support frames.</body>
      </noframes>

   </frameset>
```

```
</html>
```

This will produce the following result −

Example

Let's put the above example as follows, here we replaced rows attribute by cols and changed their width. This will create all the three frames vertically −

```
<!DOCTYPE html>
<html>

   <head>
      <title>HTML Frames</title>
   </head>

   <frameset cols = "25%,50%,25%">
      <frame name = "left" src = "/html/top_frame.htm" />
      <frame name = "center" src = "/html/main_frame.htm" />
      <frame name = "right" src = "/html/bottom_frame.htm" />

      <noframes>
         <body>Your browser does not support frames.</body>
      </noframes>
   </frameset>

</html>
```

This will produce the following result −

The <frameset> Tag Attributes

Following are important attributes of the <frameset> tag −

Sr. No	Attribute & Description
1	**cols** Specifies how many columns are contained in the frameset and the size of each column. You can spe each column in one of the four ways − Absolute values in pixels. For example, to create three vertical frames, use *cols = "100, 500, 100"*. A percentage of the browser window. For example, to create three vertical frames, use *cols = "10%, 80%* Using a wildcard symbol. For example, to create three vertical frames, use *cols = "10%, *, 10%"*. In th takes remainder of the window. As relative widths of the browser window. For example, to create three vertical frames, use *cols = "3*, 2* alternative to percentages. You can use relative widths of the browser window. Here the window is divide first column takes up half of the window, the second takes one third, and the third takes one sixth.

2	**rows** This attribute works just like the cols attribute and takes the same values, but it is used to specify the rows For example, to create two horizontal frames, use *rows = "10%, 90%"*. You can specify the height of each way as explained above for columns.
3	**border** This attribute specifies the width of the border of each frame in pixels. For example, border = "5". A valu no border.
4	**frameborder** This attribute specifies whether a three-dimensional border should be displayed between frames. Thi value either 1 (yes) or 0 (no). For example, frameborder = "0" specifies no border.
5	**Frame spacing** This attribute specifies the amount of space between frames in a frameset. This can take any integer val frame spacing = "10" means there should be 10 pixels spacing between each frames.

The <frame> Tag Attributes

Following are the important attributes of <frame> tag −

Sr. No	**Attribute & Description**
1	**src** This attribute is used to give the file name that should be loaded in the frame. Its value can be any URL. F = "/html/top_frame.htm" will load an HTML file available in html directory.
2	**name** This attribute allows you to give a name to a frame. It is used to indicate which frame a document shoul This is especially important when you want to create links in one frame that load pages into an another case the second frame needs a name to identify itself as the target of the link.
3	**frameborder** This attribute specifies whether or not the borders of that frame are shown; it overrides the value given in attribute on the <frameset> tag if one is given, and this can take values either 1 (yes) or 0 (no).
4	**marginwidth**

	This attribute allows you to specify the width of the space between the left and right of the frame's borders content. The value is given in pixels. For example, marginwidth = "10".
5	**marginheight** This attribute allows you to specify the height of the space between the top and bottom of the frame's contents. The value is given in pixels. For example, marginheight = "10".
6	**noresize** By default, you can resize any frame by clicking and dragging on the borders of a frame. The noresize attr user from being able to resize the frame. For example, noresize = "noresize".
7	**scrolling** This attribute controls the appearance of the scrollbars that appear on the frame. This takes values eith "auto". For example, scrolling = "no" means it should not have scroll bars.
8	**longdesc** This attribute allows you to provide a link to another page containing a long description of the contents c example, longdesc = "framedescription.htm"

Browser Support for Frames

If a user is using any old browser or any browser, which does not support frames then <noframes> element should be displayed to the user.

So you must place a <body> element inside the <noframes> element because the <frameset> element is supposed to replace the <body> element, but if a browser does not understand <frameset> element then it should understand what is inside the <body> element which is contained in a <noframes> element.

You can put some nice message for your user having old browsers. For example, *Sorry!! your browser does not support frames.* as shown in the above example.

Frame's name and target attributes

One of the most popular uses of frames is to place navigation bars in one frame and then load main pages into a separate frame.

Let's see following example where a test.htm file has following code −

```
<!DOCTYPE html>
<html>

   <head>
      <title>HTML Target Frames</title>
   </head>
   <frameset cols = "200, *">
      <frame src = "/html/menu.htm" name = "menu_page" />
      <frame src = "/html/main.htm" name = "main_page" />

      <noframes>
         <body>Your browser does not support frames. </body>
      </noframes>
   </frameset>

</html>
```

Here, we have created two columns to fill with two frames. The first frame is 200 pixels wide and will contain the navigation menu bar implemented by **menu.htm** file. The second column fills in remaining space and will contain the main part of the page and it is implemented by **main.htm** file. For all the three links available in menu bar, we have mentioned target frame as **main_page**, so whenever you click any of the links in menu bar, available link will open in main page.

Following is the content of menu.htm file

```
<!DOCTYPE html>
<html>

   <body bgcolor = "#4a7d49">
      <a href = "http://www.google.com" target =
"main_page">Google</a>
      <br />
      <br />

      <a href = "http://www.microsoft.com" target =
"main_page">Microsoft</a>
      <br />
      <br />

      <a href = "http://news.bbc.co.uk" target = "main_page">BBC
News</a>
   </body>

</html>
```

Following is the content of main.htm file −

```
<!DOCTYPE html>
<html>

   <body bgcolor = "#b5dcb3">
      <h3>This is main page and content from any link will be
displayed here.</h3>
      <p>So now click any link and see the result.</p>
   </body>

</html>
```

When we load **test.htm** file, it produces following result −

Now you can try to click links available in the left panel and see the result. The *targetattribute* can also take one of the following values −

Sr.No	Option & Description
1	**_self** Loads the page into the current frame.
2	**_blank** Loads a page into a new browser window. Opening a new window.
3	**_parent** Loads the page into the parent window, which in the case of a single frameset is the main browser wi
4	**_top** Loads the page into the browser window, replacing any current frames.
5	**targetframe** Loads the page into a named targetframe.

Html layers

The HTML <layer> tag is used to position and animate (through scripting) elements in a page. A layer can be thought of as a separate document that resides on top of the main one, all existing within one window.

This tag has support in Netscape 4 and higher versions of it.

Example

This example creates three overlapping layers. The back one is red, the middle one is blue, and the front one is green.

```
<!DOCTYPE html>
<html>

   <head>
      <title>HTML layer Tag</title>
   </head>

   <body>
      <layer id = "layer1" top = "250" left = "50" width = "200"
         height = "200" bgcolor = "red">
         <p>layer 1</p>
      </layer>

      <layer id = "layer2" top = "350" left = "150" width = "200"
         height = "200" bgcolor = "blue">
         <p>layer 2</p>
      </layer>

      <layer id = "layer3" top = "450" left = "250" width = "200"
         height = "200" bgcolor = "green">
         <p>layer 3</p>
      </layer>
   </body>

</html>
```

This will produce the following result; it will work in Netscape 4 and higher versions.

Global Attributes

This tag supports all the global attributes described in − HTML Attribute Reference

Specific Attributes

The HTML <layer> tag also supports the following additional attributes −

Attribute	Value	Description
above	layer name	The name of the inline layer that will be positioned directly above layer in the z-order.
background	URL	A filename or URL for an image upon which the inline layer's text appear.
below	layer name	The name of the inline layer that will be positioned directly below layer in the z-order.
bgcolor	rgb(x,x,x) #xxxxxx colorname	The color to use for the inline layer background.
clip	Number	The coordinates of the inline layer's viewable area.
height	Pixels	The inline layer's height, in pixels.

left	Number	The position of the left side of the inline layer. If the current inline another layer.called the parent layer-then the position is relative layer.
name	layer name	The name of the inline layer.
pagex	Number	The position of the left side of the inline layer relative to the brow
pagey	Number	The position of the top of the inline layer relative to the browser w
src	URL	The URL of a page that will appear inside the inline layer.
top	Number	The position of the top of the inline layer. If the current inline laye another layer--called the parent layer--then the position is relative layer.
visibility	show hide inherit	Determines whether the inline layer is visible.
width	Pixels	The inline layer's width, in pixels.
z-index	Number	The inline layer's position within the z-order. Inline layers with hig values are positioned above inline layers with lower Z-INDEX val

HTML STYLESHEET

Cascading Style Sheets (CSS) describe how documents are presented on screens, in print, or perhaps how they are pronounced. W3C has actively promoted the use of style sheets on the Web since the consortium was founded in 1994.

Cascading Style Sheets (CSS) provide easy and effective alternatives to specify various attributes for the HTML tags. Using CSS, you can specify a number of style properties for a given HTML element. Each property has a name and a value, separated by a colon (:). Each property declaration is separated by a semi-colon (;).

Example

First let's consider an example of HTML document which makes use of <font> tag and associated attributes to specify text color and font size −

Note − The *font* tag deprecated and it is supposed to be removed in a future version of HTML. So, they should not be used rather, it's suggested to use CSS styles to manipulate your fonts. But still for learning purpose, this chapter will work with an example using the font tag.

```
<!DOCTYPE html>
<html>

   <head>
      <title>HTML CSS</title>
   </head>

   <body>
      <p><font color = "green" size = "5">Hello, World!
</font></p>
   </body>

</html>
```

We can re-write above example with the help of Style Sheet as follows −

```
<!DOCTYPE html>
<html>

   <head>
      <title>HTML CSS</title>
   </head>

   <body>
      <p style = "color:green; font-size:24px;" >Hello,
World!</p>
   </body>

</html>
```

This will produce the following result −

You can use CSS in three ways in your HTML document −

- **External Style Sheet** − Define style sheet rules in a separate .css file and then include that file in your HTML document using HTML <link> tag.
- **Internal Style Sheet** − Define style sheet rules in header section of the HTML document using <style> tag.

- **Inline Style Sheet** − Define style sheet rules directly along-with the HTML elements using **style** attribute.

Let's see all the three cases one by one with the help of suitable examples.

External Style Sheet

If you need to use your style sheet to various pages, then its always recommended to define a common style sheet in a separate file. A cascading style sheet file will have extension as **.css** and it will be included in HTML files using <link> tag.

Example

Consider we define a style sheet file **style.css** which has following rules −

```
. red {
   color: red;
}
. thick {
   font-size:20px;
}
. green {
   color:green;
}
```

Here we defined three CSS rules which will be applicable to three different classes defined for the HTML tags. I suggest you should not bother about how these rules are being defined because you will learn them while studying CSS. Now let's make use of the above external CSS file in our following HTML document −

```
<!DOCTYPE html>
<html>

   <head>
      <title>HTML External CSS</title>
      <link rel = "stylesheet" type = "text/css" href =
"/html/style.css">
   </head>

   <body>
      <p class = "red">This is red</p>
      <p class = "thick">This is thick</p>
      <p class = "green">This is green</p>
      <p class = "thick green">This is thick and green</p>
   </body>

</html>
```

This will produce the following result −

Internal Style Sheet

If you want to apply Style Sheet rules to a single document only, then you can include those rules in header section of the HTML document using <style> tag.

Rules defined in internal style sheet overrides the rules defined in an external CSS file.

Example

Let's re-write above example once again, but here we will write style sheet rules in the same HTML document using <style> tag −

```
<!DOCTYPE html>
<html>

   <head>
      <title>HTML Internal CSS</title>

      <style type = "text/css">
         . red {
           color: red;
         }
         . thick {
           font-size:20px;
         }
         . green {
           color:green;
         }
      </style>
   </head>

   <body>
      <p class = "red">This is red</p>
      <p class = "thick">This is thick</p>
      <p class = "green">This is green</p>
      <p class = "thick green">This is thick and green</p>
   </body>

</html>
```

This will produce the following result −

Inline Style Sheet

You can apply style sheet rules directly to any HTML element using **style** attribute of the relevant tag. This should be done only when you are interested to make a particular change in any HTML element only.

Rules defined in line with the element overrides the rules defined in an external CSS file as well as the rules defined in <style> element.

Example

Let's re-write above example once again, but here we will write style sheet rules along with the HTML elements using **style** attribute of those elements.

```
<!DOCTYPE html>
<html>

   <head>
      <title>HTML Inline CSS</title>
   </head>

   <body>
      <p style = "color:red;">This is red</p>
      <p style = "font-size:20px;">This is thick</p>
      <p style = "color:green;">This is green</p>
      <p style = "color:green;font-size:20px;">This is thick and
green</p>
   </body>

</html>
```

This will produce the following result –

This is red

This is thick

This is green

This is thick and green

POSITIONING WITH STYLESHEETS

Position an <h2> element:

```
h2 {
  position: absolute;
  left: 100px;
  top: 150px;
}
```

Definition and Usage

The `position` property specifies the type of positioning method used for an element (static, relative, absolute, fixed, or sticky).

Default value:	Static
Inherited:	No
Animatable:	no. Read about *animatable*
Version:	CSS2
JavaScript syntax:	*object*.style.position="absolute

HTML FORMS

HTML Forms are required, when you want to collect some data from the site visitor. For example, during user registration you would like to collect information such as name, email address, credit card, etc.

A form will take input from the site visitor and then will post it to a back-end application such as CGI, ASP Script or PHP script etc. The back-end application will perform required processing on the passed data based on defined business logic inside the application.

There are various form elements available like text fields, textarea fields, drop-down menus, radio buttons, checkboxes, etc.

The HTML **<form>** tag is used to create an HTML form and it has following syntax −

```
<form action = "Script URL" method = "GET|POST">
   form elements like input, textarea etc.
</form>
```

Form Attributes

Apart from common attributes, following is a list of the most frequently used form attributes −

Sr .No	Attribute & Description

1	**action** Backend script ready to process your passed data.
2	**method** Method to be used to upload data. The most frequently used are GET and POST methods.
3	**target** Specify the target window or frame where the result of the script will be displayed. It takes values like _blar etc.
4	**enctype** You can use the enctype attribute to specify how the browser encodes the data before it sends it to the values are − **application/x-www-form-urlencoded** − This is the standard method most forms use in simple scenarios **mutlipart/form-data** − This is used when you want to upload binary data in the form of files like image, w

Note − You can refer to Perl & CGI for a detail on how form data upload works.

HTML Form Controls

There are different types of form controls that you can use to collect data using HTML form −

- Text Input Controls
- Checkboxes Controls
- Radio Box Controls
- Select Box Controls
- File Select boxes
- Hidden Controls
- Clickable Buttons
- Submit and Reset Button

Text Input Controls

There are three types of text input used on forms −

- **Single-line text input controls** − This control is used for items that require only one line of user input, such as search boxes or names. They are created using HTML **<input>** tag.
- **Password input controls** − This is also a single-line text input but it masks the character as soon as a user enters it. They are also created using HTMl <input> tag.
- **Multi-line text input controls** − This is used when the user is required to give details that may be longer than a single sentence. Multi-line input controls are created using HTML **<textarea>** tag.

Single-line text input controls

This control is used for items that require only one line of user input, such as search boxes or names. They are created using HTML <input> tag.

Example

Here is a basic example of a single-line text input used to take first name and last name −

```
<!DOCTYPE html>
<html>

   <head>
      <title>Text Input Control</title>
   </head>

   <body>
      <form >
         First name: <input type = "text" name = "first_name" />
         <br>
         Last name: <input type = "text" name = "last_name" />
      </form>
   </body>

</html>
```

This will produce the following result −

Attributes

Following is the list of attributes for <input> tag for creating text field.

Sr.No	Attribute & Description
1	**type** Indicates the type of input control and for text input control it will be set to **text**.
2	**name** Used to give a name to the control which is sent to the server to be recognized and get the value.
3	**value** This can be used to provide an initial value inside the control.
4	**size**

	Allows to specify the width of the text-input control in terms of characters.
5	**maxlength** Allows to specify the maximum number of characters a user can enter into the text box.

Password input controls

This is also a single-line text input but it masks the character as soon as a user enters it. They are also created using HTML <input>tag but type attribute is set to **password**.

Example

Here is a basic example of a single-line password input used to take user password −

```
<!DOCTYPE html>
<html>

   <head>
      <title>Password Input Control</title>
   </head>

   <body>
      <form >
         User ID : <input type = "text" name = "user_id" />
         <br>
         Password: <input type = "password" name = "password" />
      </form>
   </body>

</html>
```

This will produce the following result −

Attributes

Following is the list of attributes for <input> tag for creating password field.

Sr.No	Attribute & Description
1	**type** Indicates the type of input control and for password input control it will be set to **password**.
2	**name**

	Used to give a name to the control which is sent to the server to be recognized and get the value.
3	**value** This can be used to provide an initial value inside the control.
4	**size** Allows to specify the width of the text-input control in terms of characters.
5	**maxlength** Allows to specify the maximum number of characters a user can enter into the text box.

Multiple-Line Text Input Controls

This is used when the user is required to give details that may be longer than a single sentence. Multi-line input controls are created using HTML <textarea> tag.

Example

Here is a basic example of a multi-line text input used to take item description −

```
<!DOCTYPE html>
<html>

   <head>
      <title>Multiple-Line Input Control</title>
   </head>

   <body>
      <form>
         Description : <br />
         <textarea rows = "5" cols = "50" name = "description">
            Enter description here...
         </textarea>
      </form>
   </body>

</html>
```

This will produce the following result −

Attributes

Following is the list of attributes for <textarea> tag.

Sr.No	Attribute & Description
1	**name** Used to give a name to the control which is sent to the server to be recognized and get the value.
2	**rows** Indicates the number of rows of text area box.
3	**cols** Indicates the number of columns of text area box

Checkbox Control

Checkboxes are used when more than one option is required to be selected. They are also created using HTML <input> tag but type attribute is set to **checkbox.**.

Example

Here is an example HTML code for a form with two checkboxes −

```
<!DOCTYPE html>
<html>

   <head>
      <title>Checkbox Control</title>
   </head>

   <body>
      <form>
         <input type = "checkbox" name = "maths" value = "on">
Maths
         <input type = "checkbox" name = "physics" value = "on">
Physics
      </form>
   </body>

</html>
```

This will produce the following result −

Attributes

Following is the list of attributes for <checkbox> tag.

Sr. No	Attribute & Description
1	**type** Indicates the type of input control and for checkbox input control it will be set to **checkbox.**.
2	**name** Used to give a name to the control which is sent to the server to be recognized and get the value.
3	**value** The value that will be used if the checkbox is selected.
4	**checked** Set to *checked* if you want to select it by default.

Radio Button Control

Radio buttons are used when out of many options, just one option is required to be selected. They are also created using HTML <input> tag but type attribute is set to **radio**.

Example

Here is example HTML code for a form with two radio buttons −

```
<!DOCTYPE html>
<html>

   <head>
      <title>Radio Box Control</title>
   </head>

   <body>
      <form>
         <input type = "radio" name = "subject" value = "maths">
Maths
         <input type = "radio" name = "subject" value =
"physics"> Physics
      </form>
   </body>

</html>
```

This will produce the following result −

Attributes

Following is the list of attributes for radio button.

Sr. No	Attribute & Description
1	**type** Indicates the type of input control and for checkbox input control it will be set to radio.
2	**name** Used to give a name to the control which is sent to the server to be recognized and get the value.
3	**value** The value that will be used if the radio box is selected.
4	**checked** Set to *checked* if you want to select it by default.

Select Box Control

A select box, also called drop down box which provides option to list down various options in the form of drop down list, from where a user can select one or more options.

Example

Here is example HTML code for a form with one drop down box

```
<!DOCTYPE html>
<html>

   <head>
      <title>Select Box Control</title>
   </head>

   <body>
      <form>
         <select name = "dropdown">
            <option value = "Maths" selected>Maths</option>
            <option value = "Physics">Physics</option>
         </select>
      </form>
   </body>

</html>
```

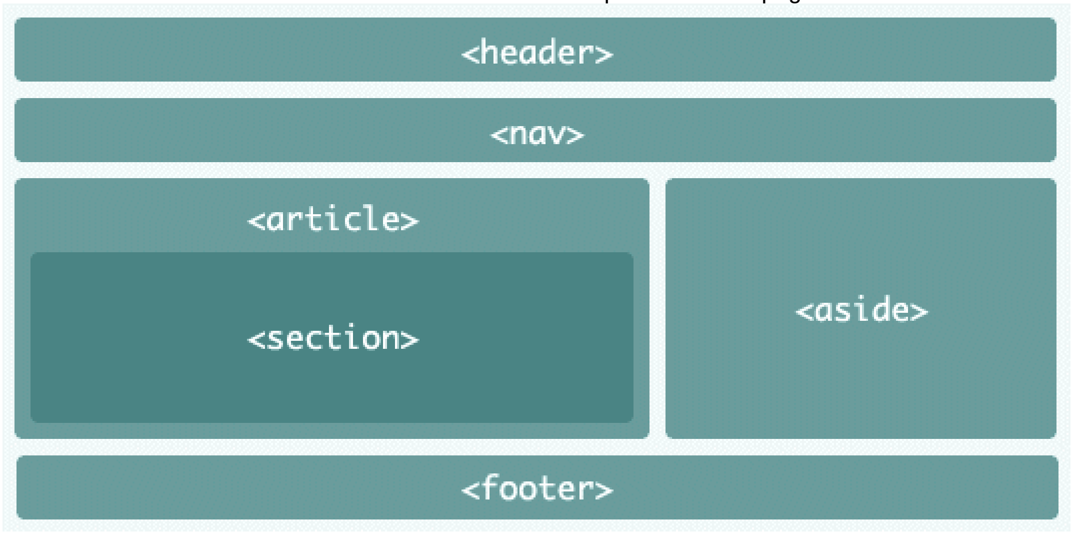
This will produce the following result –A button with two options maths and physics .

New and emerging form elements /new tag in html

The W3C HTML5 specification introduces numerous new tags to define semantic/structural elements, text-formatting instructions, form controls, input types, and more. This page describes all of the new HTML5 tags along with updates to an existing tag.

New Semantic/Structural Elements

HTML5 offers new semantic elements to define different parts of a web page:



A self-contained composition in a document that is independently distributable or reusable, e.g. a forum post, a magazine or newspaper article, or blog entry.
<aside>
Defines content tangentially related to the content surrounding it, such as related reading links and glossaries, which may or not be nested within an article element.
<figcaption>
Defines a caption for a <figure> element
<figure>
Used in conjunction with the <figcaption> element to mark up diagrams, illustrations, photos, and code listings, etc.
<header>
Not be confused with the <head> element, the <header> tag typically contains the section?s heading (an h1?h6 element), as well as other content, such as a navigation links, table of contents, a search form, or any relevant logos.
<footer>
For content located at the very bottom of the web page or nearest section. A footer typically contains information about its section such as who wrote it, links to related documents, copyright data, etc. It in turn may contain entire sections, with appendices, indexes, license agreements, and other similar content.
<main>
Delineates the main content of the body of a document or web app. As such, the main content area holds content that is directly related to or expands upon the central topic of the page. Moreover, it helps screen readers and other assistive technologies understand where the main page content begins.
<mark>
Meant to bring the reader?s attention to a part of the text due to its contextual relevance.

<nav>
Denotes a section with navigation links, either to other pages or to parts within the same page.
<section>
Unlike the <div> tag, which is used for a myriad of purposes, not the least of which is formatting content, the <section> element demarcates a thematic grouping of content. Each section typically includes a heading element and associated content within DIVs and Paragraphs. Examples include introduction, blog entries, and contact information.
<details>
Produces an expandable box to display additional information. The expanding/collapsing behaviour does not depend on scripting, so it should work even if JavaScript is disabled or not supported.
<summary>
An optional element that summarizes the contents of the parent details element. As such, it may contain a description, caption, or legend.
<time>
Contains both human-friendly contents, along with a machine-readable form of those contents in the datetime attribute. The kinds of content range from various kinds of dates, to times, time-zone offsets, and durations.

Text-level Elements

<bdi>
Defines a part of text that might be formatted in a different direction from other text
<wbr>
Defines a possible line-break in text that is written as one long word.

New Form Elements

HTML5 introduces a number of new input types, attributes, and other elements to the HTML language.
<datalist>
Defines pre-defined options for input controls. It works in a similar way to an autocomplete textbox.
<keygen>
Defines a key-pair generator field (for forms). When the control's form is submitted, the private key is stored in the local keystore, and the public key is packaged and sent to the server.
<output>
Defines the result of a calculation
<progress>
Presents a progress bar that tracks the progress of a task.

New Input Types

HTML5 introduced a number of new input types for forms to address specific behavioral and formatting requirements that were lacking for the HTML 4.01 spec. For instance, handling the inputting of dates, numbers, telephone numbers, etc.

New Input Types	New Input Attributes
• color • date • datetime • datetime-local • email • month • number • range • search • tel • time • url • week	• autocomplete • autofocus • form • formaction • formenctype • formmethod • formnovalidate • formtarget • height and width • list • min and max • multiple • pattern (regexp) • placeholder • required • step

More Relaxed Attribute Syntax

In HTML5, attribute values may be delimited by single or double quotes or, in the case of single word entries, without either. Attributes without a value (such as disabled) don't require an equals or quotes (="").

Graphic Elements

Defines graphic drawing using JavaScript Defines graphic drawing using SVG
See the following article for more information on the HTML5 canvas element and this article for more on the HTML5 svg element.

Media Elements

As the Internet becomes a more immersive multi-media experience, browsers are now equipped to handle many media types without the need for additional plugins. A prime example is the <embed> element that has to an explosion in YouTube video sharing.
<audio>
Defines sound or music content
<embed>
Defines containers for external applications (usually a video player)
<source>
Defines sources for <video> and <audio>
<track>
Defines tracks for <video> and <audio>
<video>
Defines video or movie content

QUESTION BANK:

Q.1 what is internet? Explain services provided by internet?

Q.2 Explain types of computer networks?

Q.3 Explain all network topologieswith their drawbacks?

Q.4What is client-server model ? Explain it in detail with example?

Q.5 Write short note on :

a) single user

b) multi user

Q.6 Differentiate between single user and multi user O.S.

Q.7 Explain working of internet?

Q.8 What is DNS? Explain how its works?

Q.9 What are search engines? Explain its all categories?

Q.10 Explain HTTP?

Q.11 Explain URL and its types?

Q.12 Explain video-conferencing concepts?

Q.13 What are internet tools? Explain its all types?

Q.14 Write short note on e-mail mailing list?

Q.15 Write short note on IRC?

Q.16 Explain working of usenet newsgroups?

Q.17 Mailing list Vs Newsgroup

Q.18 Write short note on:

a) proxy server

b) web server

Q.19 Define web page? Explain two types of web page?

Q.20 What is scripting languages?where is it used.Explain it with characteristics?

Q.21 What are client side and server side scripting in web? Explain.

Q.22 Explain features of HTML?

Q.23 What is meant by FORM? What are essential steps while designing the idle form?

Q24 Explain following tags used to construct web form: a. FORM b. INPUT c. SELECT

Q.25What is FORM? Give different object of form with tags

Q.26. What is Web server? Explain the features of WEB Server

Q.27 21. List the applications of Internet

Q.28 Explain in brief the history of internet.

Q.29 What is Search Engine. Write a steps to search using google.com

Q.30 How we design a good web site? List the criterion to define a better web.

Q.31 Write a java script to validate an email address

Q.32 write a java script to validate phone number

Q.33 Write a script to demonstrate the use of Date object?

Q.34 Write a script to display sum of first 10 numbers

Q.35. generate html page for our own biodata(use all formatting tag)

Q.36 generate college admission form using html form tag

Q.37 create html application for demo of Error! Filename not specified.tag