

INTERNET AND WEB TECHNOLOGY

4th Semester Theory-4

Prepared By –Sonalisa Behera
Lecturer in CSE
BOSE, Cuttack



INTERNET AND WEB TECHNOLOGY (IWT)

Chapter 1: Internet Basics

- ✓ Computer network
- ✓ Concept of Internet, Intranet, Modem
- ✓ IP Address, Internet Domains, CIDR Notation, ISP, TCP/IP

Chapter 1 - Internet Basics

Computer network

Introduction to Computer Network

- Computer networks are required for sharing information when we communicate, we are sharing information. This sharing can be local or remote. Between individuals, local communication usually occurs face to face, while remote communication takes place over distance.
- The term **Data communications** means the exchange of data between two devices via some form of transmission medium such as a wire cable whereas the term telecommunication, which includes telephony, telegraphy and television means communication at a distance.
- A data communications system has five components:
 - **1. Message:** The message is the information (data) to be communicated. Popular forms of information include text, numbers, pictures, audio, and video.
 - **2. Sender:** The sender is the device that sends the data message. It can be a computer, workstation, telephone handset, video camera, and so on.
 - **3. Receiver:** The receiver is the device that receives the message. It can be a computer, workstation, telephone handset, television, and so on.
 - **4. Transmission medium:** The transmission medium is the physical path by which a message travels from sender to receiver. Some examples of transmission media include twisted-pair wire, coaxial cable, fiber-optic cable, and radio waves.
 - **5. Protocol:** A protocol is a set of rules that govern data communications. It represents an agreement between the communicating devices. Without a protocol, two devices may be connected but not communicating, just as a person speaking French cannot be understood by a person who speaks only Japanese.

What is a Computer Network?

- A computer network is a set of connected computers. Computers on a network are called nodes. The connection between computers can be done via cabling, most commonly the Ethernet cable, or wirelessly through radio waves. Connected computers can share resources, like access to the Internet, printers, file servers, and others. A network is a multipurpose connection, which allows a single computer to do more.

or

- "A computer network is defined to be an interconnected collection of autonomous computers"

- Interconnected means capable of exchanging information, and autonomous means, that is independent or self governing. The connections allow users to exchange data, and pool and share resources, including Printer, Data, and Computing Power.
- In contrast to Computer Networks, where the existence of several separate machines is usually hidden from the user, distributed systems are computer networks with special software that makes the existence of several computers visible to the user.

Types of Computer Networks

Computer Networks can be of following types:

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

Personal Area Network (PAN)

- PAN is a computer network formed around a person. It generally consists of a computer, mobile, or personal digital assistant. It is mostly personal devices network equipped within a limited area. Allows us to handle the interconnection of IT devices at the surrounding of a single user. It can be wirelessly connected to the internet called WPAN.

Advantages of PAN

- PAN networks are relatively secure and safe
- It offers only short-range solution up to ten meters

Disadvantages of PAN

- It may establish a bad connection to other networks at the same radio bands.
- Distance limits.

Local Area Network (LAN)

- It is also called LAN and designed for small physical areas such as an office, group of buildings or a factory.
- LAN can be a simple network like connecting two computers, to share files and network among each other while it can also be as complex as interconnecting an entire building.

- LAN networks are also widely used to share resources like printers, shared hard –drive etc.

Advantages of LAN

- **Resource Sharing:** Computer resources like printers, modems, DVD–ROM drives and hard disks can be shared with the help of local area networks. This reduces cost and hardware purchases.
- **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.
- **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.
- **Internet Sharing:** 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.

Disadvantages of LAN

- **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.
- **Data Security Threat:** Unauthorized users can access important data of an organization if centralized data repository is not secured properly by the LAN administrator.
- **Covers Limited Area:** Local Area Network covers a small area like one office, one building or a group of nearby buildings.

Metropolitan Area Network (MAN)

- It is basically a bigger version of LAN. It is also called MAN and uses the similar technology as LAN. It is designed to extend over the entire city. It can be means to connecting a number of LANs into a larger network or it can be a single cable. It is mainly hold and operated by single private company or a public company.

Advantages of MAN

- Extremely efficient and provide fast communication via high–speed carriers, such as fibre optic cables.
- It provides a good back bone for large network and provides greater access to WANs.
- A MAN usually encompasses several blocks of a city or an entire city.

Disadvantages of MAN

- More cable required for a MAN connection from one place to another.
- It is difficult to make the system secure from hackers and industrial espionage(spying) graphical regions.

Wide Area Network (WAN)

- It is also called WAN. WAN can be private or it can be public leased network. It is used for the network that covers large distance such as cover states of a country. It is not easy to design and maintain. Communication medium used are satellite, public telephone networks which are connected by routers.

Advantages of WAN

- Covers a large geographical area so long distance business can connect on the one network.
- Expensive things (such as printers or phone lines to the internet) can be shared by all the computers on the network without having to buy a different peripheral for each computer.
- Everyone on the network can use the same data. This avoids problems where some users may have older information than others.

Disadvantages of WAN

- Need a good firewall to restrict outsiders from entering and disrupting the network.
- Setting up a network can be an expensive, slow and complicated. The bigger the network the more expensive it is.
- Security is a real issue when many different people have the ability to use information from other computers. Protection against hackers and viruses adds more complexity and expense.

Types of Network Topology

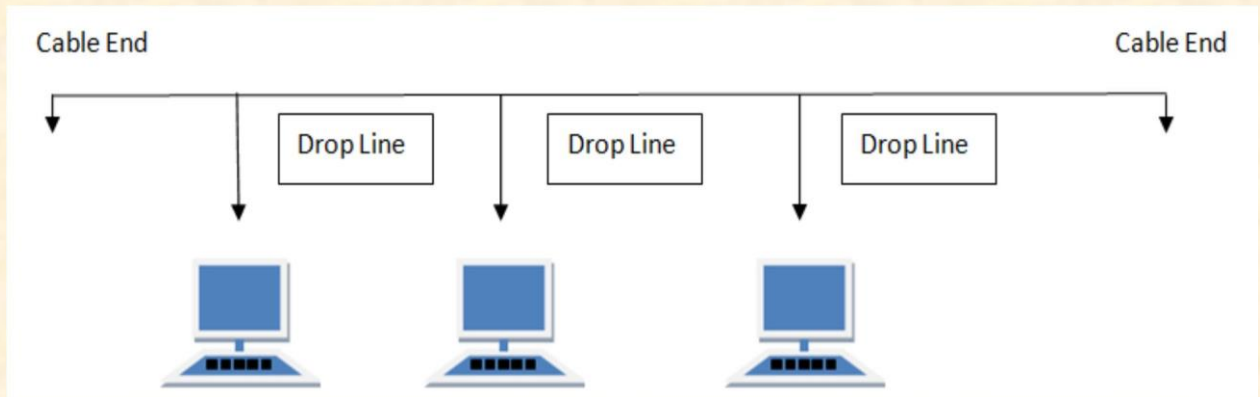
BUS Topology

- Bus topology is a network type in which every computer and network device is connected to single cable. When it has exactly two endpoints, then it is called Linear Bus topology.

Features of Bus Topology

- It transmits data only in one direction.

- Every device is connected to a single cable.



Advantages of Bus Topology

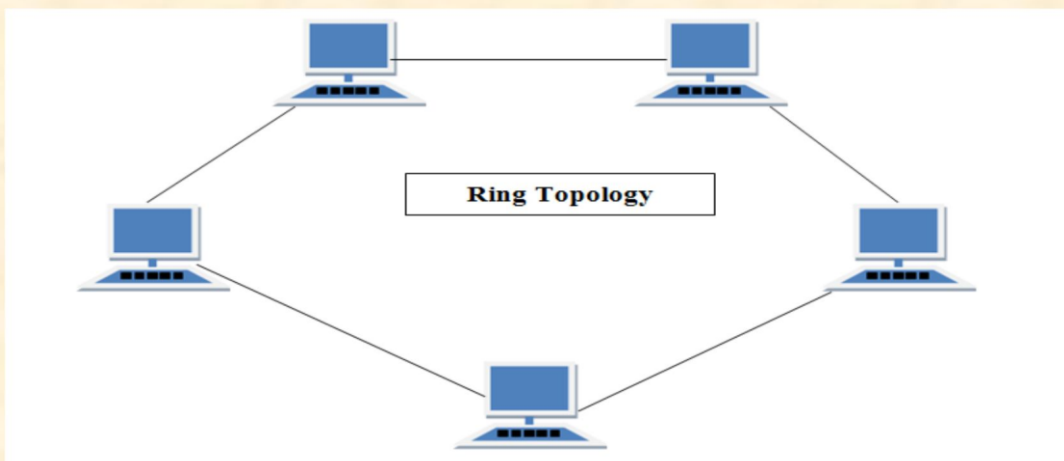
- It is cost effective.
- Cable required is least compared to other network topology.
- Used in small networks.
- It is easy to understand.
- Easy to expand joining two cables together.

Disadvantages of Bus Topology

- Cables fails then whole network fails.
- If network traffic is heavy or nodes are more the performance of the network decreases.
- Cable has a limited length.
- It is slower than the ring topology.

RING Topology

- It is called ring topology because it forms a ring as each computer is connected to another computer, with the last one connected to the first. Exactly two neighbors for



each device.

Features of Ring Topology

- A number of repeaters are used for Ring topology with large number of nodes, because if someone wants to send some data to the last node in the ring topology with 100 nodes, then the data will have to pass through 99 nodes to reach the 100th node. Hence to prevent data loss repeaters are used in the network.
- The transmission is unidirectional, but it can be made bidirectional by having 2 connections between each Network Node, it is called Dual Ring Topology.
- In Dual Ring Topology, two ring networks are formed, and data flow is in opposite direction in them. Also, if one ring fails, the second ring can act as a backup, to keep the network up.
- Data is transferred in a sequential manner that is bit by bit. Data transmitted, has to pass through each node of the network, till the destination node.

Advantages of Ring Topology

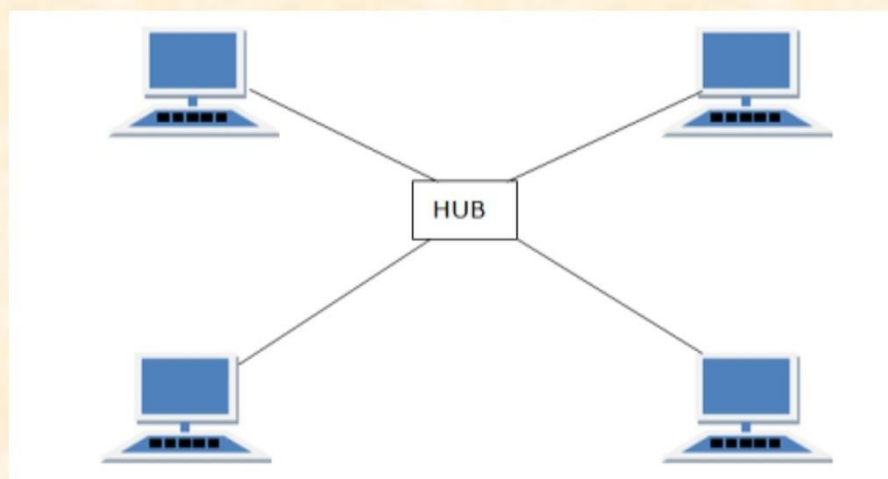
- Transmitting network is not affected by high traffic or by adding more nodes, as only the nodes having tokens can transmit data.
- Cheap to install and expand

Disadvantages of Ring Topology

- Troubleshooting is difficult in ring topology.
- Adding or deleting the computers disturbs the network activity.
- Failure of one computer disturbs the whole network.

STAR Topology

- In this type of topology all the computers are connected to a single hub through a cable. This hub is the central node and all others nodes are connected to the central node.



Features of Star Topology

- Every node has its own dedicated connection to the hub.
- Hub acts as a repeater for data flow.
- Can be used with twisted pair, Optical Fibre or coaxial cable.

Advantages of Star Topology

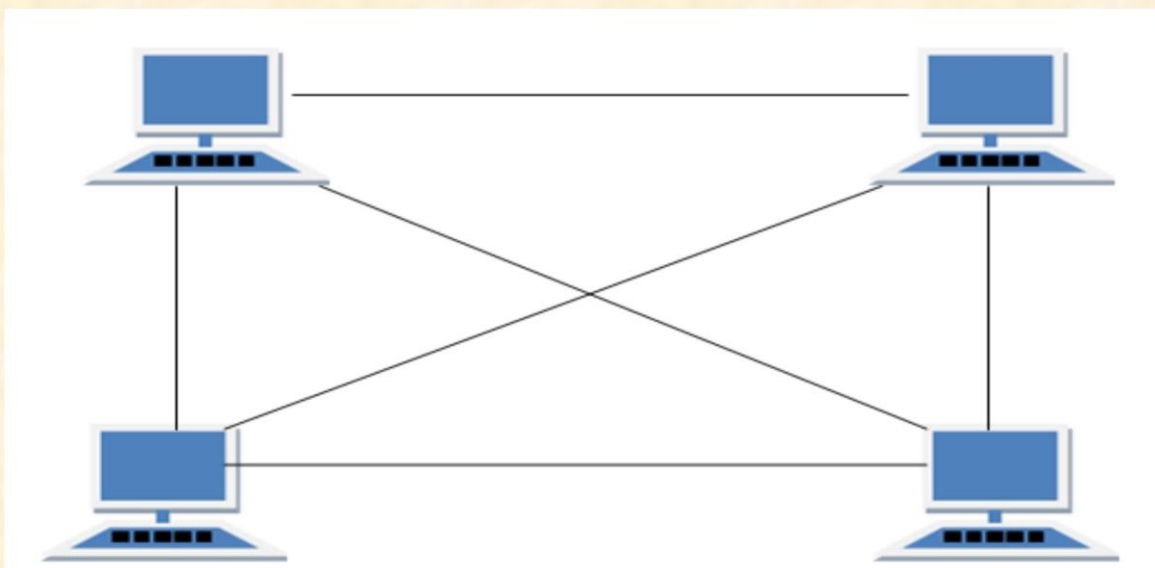
- Fast performance with few nodes and low network traffic.
- Hub can be upgraded easily.
- Easy to troubleshoot.
- Easy to setup and modify.
- Only that node is affected which has failed, rest of the nodes can work smoothly.

Disadvantages of Star Topology

- Cost of installation is high.
- Expensive to use.
- If the hub fails then the whole network is stopped because all the nodes depend on the hub.
- Performance is based on the hub that is it depends on its capacity

MESH Topology

- It is a point-to-point connection to other nodes or devices. All the network nodes are connected to each other. Mesh has $\frac{n(n-1)}{2}$ physical channels to link n devices.



Types of Mesh Topology

- **1. Partial Mesh Topology:** In this topology some of the systems are connected in the same fashion as mesh topology but some devices are only connected to two or three devices.
- **2. Full Mesh Topology:** Each and every nodes or devices are connected to each other.

Features of Mesh Topology

- Fully connected.
- Robust.
- Not flexible.

Advantages of Mesh Topology

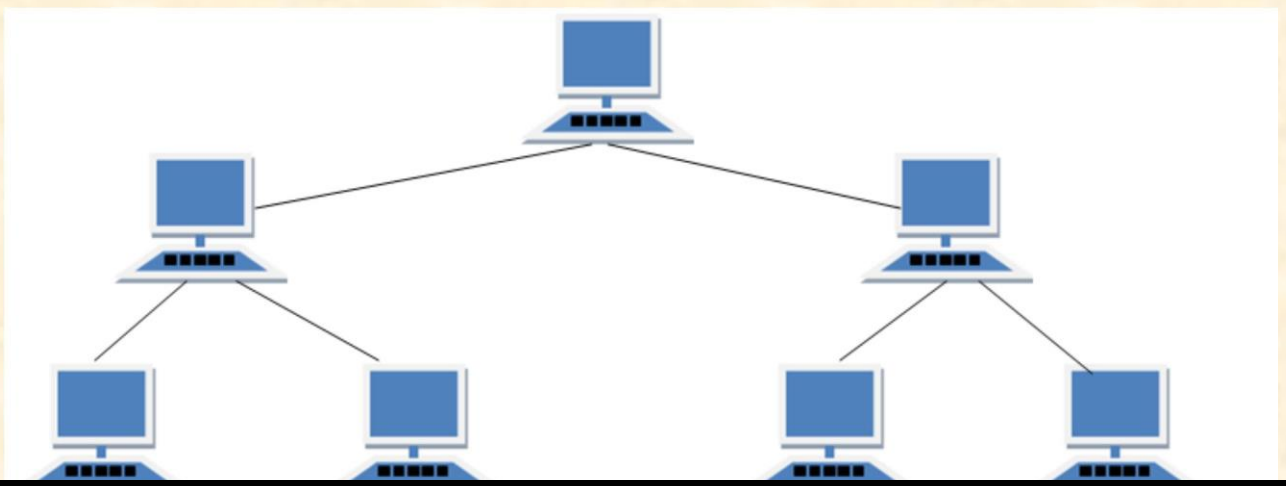
- Each connection can carry its own data load.
- It is robust.
- Fault is diagnosed easily.
- Provides security and privacy.

Disadvantages of Mesh Topology

- Installation and configuration is difficult.
- Cabling cost is more.
- Bulk wiring is required.

TREE Topology

- It has a root node and all other nodes are connected to it forming a hierarchy. It is also called hierarchical topology. It should at least have three levels to the hierarchy.



Features of Tree Topology

- Ideal if workstations are located in groups.
- Used in Wide Area Network.

Advantages of Tree Topology

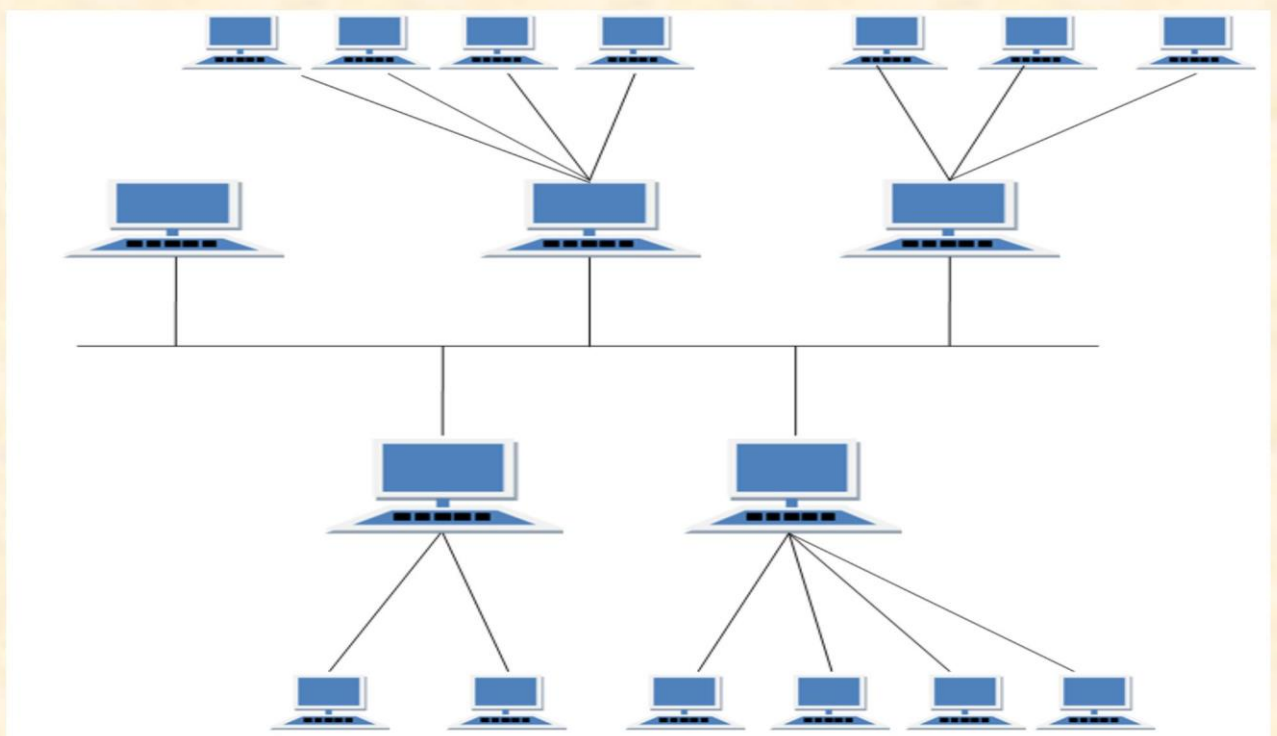
- Extension of bus and star topologies.
- Expansion of nodes is possible and easy.
- Easily managed and maintained.
- Error detection is easily done.

Disadvantages of Tree Topology

- Heavily cabled.
- Costly.
- If more nodes are added maintenance is difficult.
- Central hub fails, network fails.

HYBRID Topology

- It is two different types of topologies which is a mixture of two or more topologies. For example if in an office in one department ring topology is used and in another star topology is used, connecting these topologies will result in Hybrid Topology (ring topology and star topology).



Features of Hybrid Topology

- It is a combination of two or topologies
- Inherits the advantages and disadvantages of the topologies included

Advantages of Hybrid Topology

- Reliable as Error detecting and trouble shooting is easy.
- Effective.
- Scalable as size can be increased easily.
- Flexible.

Disadvantages of Hybrid Topology

- Complex in design.
- Costly

Concept of Internet, Intranet, Modem

Internet

What is Internet?

- The internet is a global system that uses TCP/IP protocol suite to link various types of electric devices worldwide.
- The internet is a collection of interconnected devices that are spread across the globe.
- The internet is a network of networks which consist of public, private, sales, finance, academic, business, and government networks.
- The internet is a type of network and called a network of networks.

Evolution

- The concept of Internet was originated in 1969 and has undergone several technological & Infrastructural changes as discussed below:
 - The origin of Internet devised from the concept of Advanced Research Project Agency Network (ARPANET).
 - ARPANET was developed by United States Department of Defense.
 - Basic purpose of ARPANET was to provide communication among the various bodies of government.
 - Initially, there were only four nodes, formally called Hosts.
 - In 1972, the ARPANET spread over the globe with 23 nodes located at different countries and thus became known as Internet.

- By the time, with invention of new technologies such as TCP/IP protocols, DNS, WWW, browsers, scripting languages etc., Internet provided a medium to publish and access information over the web.

Why Internet?

- The internet can be used by all, not just by scientists and engineers. As entrepreneurs, factory workers, doctors, teachers, federal employees, and citizens, people can use its technology to :
 - Create jobs, and lead towards growth.
 - Reduce health care costs while increasing the quality of services in underserved areas;
 - Deliver higher-quality, lower-cost, government services.
 - Prepare the children for the fast-paced workplace of the 21st century.
 - Build a more open and participatory democracy at all levels of government.
 - Internet enables you to know about many things just with a click, because of the internet we are able to communicate with people far away from us. It makes us feel like we are sitting next to each other.
 - It has information about everything that is known to man, and can serve various purposes based on one's requirement.

Advantages

- The Internet is a network of computers at different locations around the world.
- Allows you to send an email message from every location
- Helps you to send or receive files between different computers
- Using the Internet, you can participate in discussion groups, such as mailing lists and newsgroups.
- It allows all small, medium, and large size businesses to sell their products with small investments.
- It makes information available worldwide
- It helps you stay updated with the latest news and technologies.
- It helps us meet people with the same interests as communities, forums, chats, websites, etc.

Disadvantages

- It allows everybody to speak about everything without any limitations or censorship. That could be a bad influence on impressionable minds.
- The search engines may display some fake news results.
- Internet could replace face to face collaborations and make us lose the human touch.
- Working or on the internet is surely tiring.
- The Internet makes us lazier – as for common things like search the nearest restaurant or finding the best hotel.

Applications of Internet

Now-a-days internet is being used for multiple applications. Some of them are as follows:

1. **E-mail:** E-mail is the best service offered by internet to its users. E-mail enables us to send or receive messages over internet anywhere in the world.
2. **Entertainment:** Internet is widely used for various kinds of entertainment activities like watching movies, listening music, video on demand and playing games.
3. **E-Commerce:** Many stock exchanges and banks offer almost all of their services through internet. On-line payments, accessing accounts, online shopping are very popular these days.
4. **Publicity and Advertisement:** Internet is most common medium of communication that is why internet is used by many companies for promotion and publicity of their products.
5. **Video Conferencing:** Video conferencing uses telecommunication of audio and video to bring people of different sites together for meeting. Video conferencing may be between two individuals persons or may involve people sitting at different sites.
6. **Product Promotion:** Internet is the cheapest means to promote own's product.
7. **Group Discussion:** A number of Newsgroups are available on the internet which allows exchanging views on topics of common interest.
8. **Software Sharing:** Many software developing organizations provides their software versions on the internet,
9. **Feedback:** Commercial organizations are using Internet together customer's satisfaction of existing products, market opportunities of new products and ideas for new produces.
10. **Online Registration:** Many universities and institutions provide training and also online enrolment forms.
11. **On-line Shopping:** On-line shopping is also becoming very popular. Many people instead of going round and wasting their precious time in shopping just, sit on the system and place the order and required stuff.

Internet Issues

Emerging technologies and especially this communications revolution we are witnessing also bring with them new issues relevant to safety, privacy, security, decency and netiquette. The issues are as follows:

1. **Safety:** Safety of information is an important issue in internet. We should be very careful to upload the information on internet.
2. **Privacy:** Don't post or write anything you wouldn't mind anyone else seeing. Remember your information is not secure nor is your e-mail necessarily private. It's supposed to be, but that isn't always the case. Also, never EVER send your credit card number via email.
3. **Security:** To enhance security, many networks have encrypted, secure transmission methods that include site certificates. Be sure you are using one of these methods if you are entering secure information, such as a credit card number at a mail-order catalog's website.
4. **Decency, Filters, Censorship:** Unfortunately this revolution in accessing information means that not all information on the Internet is suitable for everyone. While many servers maintain standards of decency and refuse to allow any indecent material on their computer network, still many others do not, or worse, some exist solely for this purpose of allowing such material. Computerized search engines that enhance our ability to find information also unfortunately make it all the more easy for children and adolescents to find information you may not want them to find.
5. **Netiquette:** All cultures and societies have standards of conduct and customs. There are many good resources for you to read online. The important thing is to take some time and read up on network etiquette so you can be a responsible, well-informed member of cyber-society.

Intranet

What is Intranet?

- An intranet is a private network which gives employees in a company the ability to organize information, manage documents, shares calendars and to-do list. It normally runs in a client/server environment in a local area network.
- In Intranet, every computer is connected via the LAN and has something known as an MAC address. It is a number that allows you to identify the place where the computer is located.

Features of Intranet

- Intranet is fast and accurate.
- Most website with large graphical images, videos, and sound, process fast on Intranet
- Your firewall protects it from external threats.
- It is easy to monitor with your organization
- Easy communication across the company from all levels
- Helps to share ideas and discussions

Elements of Corporate Intranet

- **Group collaboration:** By combining several tools on an intranet, groups can share calendars, take part in "virtual workspaces" that contain public messages and files, and log into private chat areas dedicated to specific business projects. These group

collaboration features are ideal if you have employees in several branch offices that need to work together on a project.

- **Internet access:** It's not mandatory, but many companies also use their intranets to provide Web access for employees. Since the base technologies are the same, it's generally quite easy. Some all-in-one intranet servers come with built-in support for sharing a connection to an Internet service provider.
- **Security:** An intranet is generally intended for employees only – you don't want the rest of the world reading everything on it. So security is a big issue, especially if you also use your intranet to connect to the Internet. A firewall is software or hardware, or both, designed to prevent unauthorized access to your intranet by blocking outside connections.

Advantages of Intranet

- Fast, easy, low-cost to implement
- Based on open standards
- Allows connectivity with other systems
- Access to internal and external information
- Improves communication

Disadvantages of Intranet

- Threat of sharing information and the loss of control
- Unauthorized access
- Limited bandwidth for the business
- Information overload lowers productivity
- Hidden or unknown complexity and cost

Internet vs. Intranet

Apart from similarities there are some differences between the two. Following are the differences between Internet and Intranet:

Intranet	Internet
The Internet is a wide network of computers and is available to all.	Intranet is a network of computers designed for a certain group of users. Internet contains a large number of intranets. Intranet can be accessed from the Internet with specific restrictions.
Number of internet users are very high.	Number of users is limited. Internet contains various source of information. Intranet only contains group-specific information.
Anyone can access the internet	Accessible only by the organization employees or admin who have login details. It is not as safe as compared to intranet Safe and secure

	network.
It is a public network.	It is a private network.
The Internet is a wide network of computers and is available to all.	Intranet is a network of computers designed for a certain group of users. Internet contains a large number of intranets. Intranet can be accessed from the Internet with specific restrictions.

Modem

Introduction to Modem

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

The first modems were "dial-up," meaning they had to dial a phone number to connect to an ISP. These modems operated over standard analog phone lines and used the same frequencies as telephone calls, which limited their maximum data transfer rate to 56 Kbps. Dial-up modems also required full use of the local telephone line, meaning voice calls would interrupt the Internet connection.

Modern modems are typically DSL or cable modems, which are considered "broadband" devices. DSL modems operate over standard telephone lines, but use a wider frequency range. This allows for higher data transfer rates than dial-up modems and enables them to not interfere with phone calls. Cable modems send and receive data over standard cable television lines, which are typically coaxial cables. Most modern cable modems support DOCSIS (Data Over Cable Service Interface Specification), which provides an efficient way of transmitting TV, cable Internet, and digital phone signals over the same cable line.

What is Modem ?

The process of modulation and demodulation, that is, the conversion of digital data to analog form and vice-versa, is carried out by a special device called modem (modulator/demodulator).

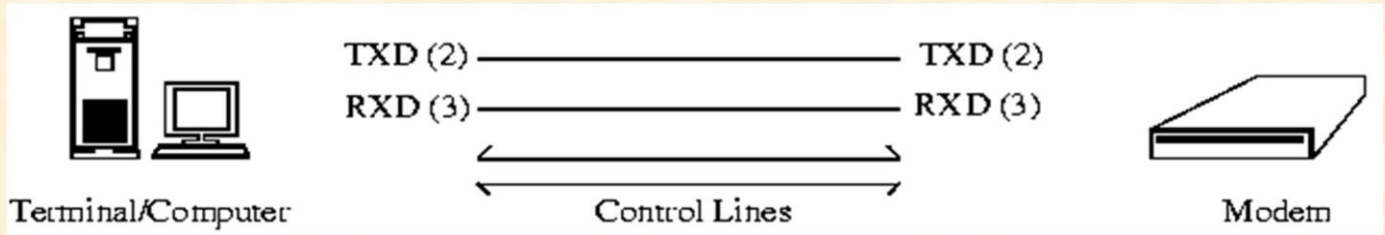
Modulation is the process of changing the form of the signal carrying the information. The demodulation process does the task of extracting information from the signals that are modulated. Analog signals can be transmitted by using devices such as a radio or a diode. Modems are classified on the basis of two criteria:

- Data sent per unit time.
- Change in the state of the signal per unit time.

Modems were first used in the 60s decade, for connecting. Computers over a network of telephone lines. The period is also known as the age of time-share computers, since the computers had to buy time in order to connect to the network by means of a modem which had a speed of 300 bits/second.

Digital Connection

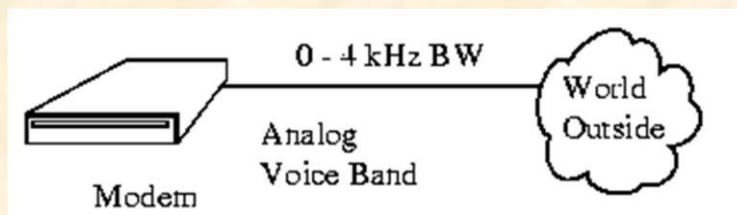
The connection between the modem and terminal/computer is a digital connection. A basic connection consists of a Transmit Data (TXD) line, a Receive Data (RXD) line and many hardware hand-shaking control lines.



The control lines determine : whose turn it is to talk (modem or terminal), if the terminal /computer is turned on, if the modem is turned on, if there is a connection to another modem, etc.

Analog Connection

The connection between the modem and outside world (phone line) is an analog connection. The Voice Channel has a bandwidth of 0–4 kHz but only 300–3400 Hz is usable for data communications. The modem converts the digital information into tones (frequencies) for transmitting through the phone lines. The tones are in the 300–3400 Hz Voice Band.



Baud

Baud is the speed at which the analog data is changing on the voice channel and bps (bits per second) is the speed that the decoded digital data is being transferred.

Features of Modem

1. **Speed** : The speed at which the modem can send data in bps (bits per second). Typically modem speeds are : 300, 600, 1200, 2400, 4800, 9600, 14.4K, 19.2K, 28.8K bps.
2. **Auto Dial /Redial** : Smart Modems can dial the phone number and auto redial if a busy signal is received.
3. **Auto Answer** : Most modems can automatically answer the phone when an incoming call comes in. They have Ring Detect capability.
4. **Self-Testing** : New modems have self-testing features. They can test the digital connection to the terminal /computer and the analog connection to a remote modem. They can also check the modem's internal electronics
5. **Voice over Data** : Voice over Data modems allow a voice conversation to take place while data is being transmitted, This requires both the source and destination modems to have this feature.
6. **Synchronous or Asynchronous Transmission** : Newer modems allow a choice of synchronous or asynchronous transmission of data. Normally, modem transmission is asynchronous. We send individual characters with just start and stop bits. Synchronous transmission or packet transmission is used in specific applications,

Functions of Modem

The essential function of a modem is to create an easily transmitted and decoded signal that allows digital data to be sent from place to place without the loss of information. The most familiar use of modems is to send information over a telephone channel, but modems can be used to relay data over any system that provides a means of transmitting analog signals, including radio and optical networks.

Data Compression

To reduce the amount of time it takes to send data and to cut down on the amount of error in the signal, modems need to employ data compression. This was especially necessary in the early days of modem technology, since data had to be sent via conventional phone lines. Not being designed for digital information, phone lines placed heavy limitations on the size and speed of signals sent over them. Data compression techniques reduce the size of the signal needed to send the required data.

Error Correction

When information is transmitted between modems, it can sometimes be damaged -- meaning that parts of the data are altered or lost. To get around this, modems use error correction. Information is grouped into batches, called frames. Each frame is tagged with a checksum, a small piece of data derived from the information in the frame. A checksum can be thought of as a kind of fingerprint, unique to the data in a particular frame. The modem that receives the information derives its own checksum from the frame it has been sent, then compares its checksum data with the checksum sent by the transmitting modem. If the checksums match, the information is undamaged. If they don't match, the data has been corrupted in transmission; the receiving modem sends it back and waits for the transmitting modem to re-send that frame.

Flow Control

Individual modems send information at different speeds. It's necessary for faster modems to slow down so that slower modems can catch up, otherwise the slower modem will receive more data than it can process. If this starts to happen, the slower modem transmits a character to the faster one. This character is a signal for the fast modem to pause in sending information until the slow modem gets caught up. When the slow modem is ready for more data, it sends a different character that signals to the fast modem that it can start transmitting again. In this way, the two modems can match their speeds.

IP Address, Internet Domains, CIDR Notation, ISP, TCP/IP

Introduction of Internet Address

Every machine on a network has a unique identifier. Just as you would address a letter to send in the mail, computers use the unique identifier to send data to specific computers on a network. Most networks today, including all computers on the internet, use the TCP/IP protocol as the standard for how to communicate on the network. In the TCP/IP protocol, the unique identifier for a computer is called its IP address.

There are two standards for IP addresses: IP Version 4 (IPv4) and IP Version 6 (IPv6). All computers with IP addresses have an IPv4 address, and most use the new IPv6 address system as well. Here are the differences between the two address types:

IPv4 uses 32 binary bits to create a single unique address on the network. An IPv4 address is expressed by four numbers separated by dots. Each number is the decimal (base-10) representation for an eight-digit binary (base-2) number, also called an octet. For example: 216.27.61.137

IPv6 uses 128 binary bits to create a single unique address on the network. An IPv6 address is expressed by eight groups of hexadecimal (base-16) numbers separated by colons, as in 2001:cdba:0000:0000:0000:0000:3257:9652. Groups of numbers that contain all zeros are often omitted to save space, leaving a colon separator to mark the gap (as in 2001:cdba::3257:9652).

DEFINITION :

"Every machine on internet has a unique number, called an IP address. IP stands for Internet Protocol, which are the rules that computer use to communicate over the internet".

127.0.0.1
172.16.0.9
192.0.0.7

A client system communicates with an internet server. An internet server computer communicates with a client Computer. Two server computers communicate with one another. Two client computer communicating via one or more servers. The computer that originates a transaction must identify its intended destination with a unique address. Every computer on the internet has a four part number address called internet protocol address, which contains routine — information that identifies its location.

Each of the four part is a number between 0 and 255, so an IP address look like as under:

205.46.117.104

Each host on TCP/IP network is assigned or unique 32 bit logical address which represents four octets of 8 bit each separated by dots, identifies the computer system on the network. The logical address is divided into two parts:

(1) Network ID (also known as network address) identifies a network and must be assigned by the Internet Network information center. The network ID must be unique to the internetwork.

(2) Host ID (also known as host address) identifies a host (workstation server, router or other TCP/IP) on a network and is assigned by the local network administrator. The address for each host must be unique to the network ID.

Uses of IP Address in Internet

An IP address is a logical address for a network adapter. The IP address uniquely identifies computers on a TCP/IP network. An IP address can be private for use on a local area network (LAN) or public for use on the internet or other wide area network (WAN). IP addresses can be determined statically (assigned to a computer by a system administrator) or dynamically (assigned by another device on the network on demand).

Two IP addressing standards are in use today. The IPv4 standard is most familiar to people and supported everywhere on the internet, but the newer IPv6 standard is planned to replace it and starting to be deployed.

IPv4 addresses consist of four bytes (32 bits). Each byte of an IP address is known as an octet. Octets can take any value between 0 and 255. Various conventions exist for the numbering and use of IP addresses.

Classes of IP Addressing

An IP address can be private for use on LAN, or public, for use on the internet. For example, 255.128.122.25. IP addresses have different five classes:

(i) Class A: Class A is for very large organizations and networks with a large number of hosts or routers like international companies might have class A begin with 0xxx. It has number of blocks approximately 128 and block size is around 16,772,216. First octet from 1 to 126 are part of class A and other three octet are also used to identify the each host, class A is used for unicast.

(ii) Class B: Class B is used for midsize networks or organization. It begins with 10xx or 128 to 191 decimal. It has number of blocks 16,384 and block size 65,536. The block size of class B is also very large like class A. Most of addresses in class B are not used. Class B networks make up a quarter of total IP addresses.

(iii) Class C: Class C is designed for small networks or organizations with a small number of hosts and routers. It is for mid-size businesses. Its addresses begin with 110x or 192 to 223 decimal. In class C numbers of blocks are 2,097,152 and block size is 256. The block size of class C is too small for many organizations. It is used for unicost.

(iv) Class D: Class D is used for multicasting. Its addresses begin with 1110, or 224 to 239 decimal. It is used to define one group of hosts on the internet. It is different from first three classes. In class D number of blocks is one and block size D is 268,435,456.

(v) Class E: Class E is used for experiment purpose only and reserved for future use. Its addresses start with 1111 or 240 to 254 decimal. Only a few organizations are using class E. In class E number of block is one and its block is 268,435,456. It is reserved for future.

INTERNET DOMAINS

The collection of networks making up the internet is divided into groups which are called domains. The domain name system assigns name and members to identify the computers. The names assigned by the DNS are called domain names and the numbers assigned by the DNS are called IP address. Domain names and IP addresses are assigned to all computers on the Internet. IP addresses are difficult to remember and type correctly. Therefore, people prefer to use host names. Domains divide WWW sites into categories based on the nature of their owner. DNS are divided into two parts i.e., domain and subdomains. Example of Internet domains: .com, .edu, .org, etc.

Domain Names

A name that identifies one or more IP address is called domain name. For example microsoft.com represents about a dozen IP addresses.

Domain names are used in URLs to identify particular Web pages. For example, in the URL <http://www.kiet.edu/index.html>, the domain name is kiet.edu. Every domain name has a suffix that indicates which top level domain. For example:

gov – Government agencies

edu – Educational institutions

org – Organizations (nonprofit)

mil – Military

com – Commercial business

net – Network organizations

Ca – Canada

th – Thailand

in – India

Domain Name System

DNS stands for two things: Domain Name Service and Domain Name Servers. One acronym defines the protocol, the other defines the machines that provide the service. The job that DNS performs is very simple: it takes the IP addresses that computers connected to the Internet use to communicate with each other and it maps them to host names.

DNS translates IP addresses into hostnames and back again. The hostnames are for the benefit human end users. The IP addresses are the only essential thing, as far as the computers are concerned.

CIDR NOTATION

WHAT IS CIDR NOTATION?

Classless inter-domain routing (CIDR) is a set of Internet protocol (IP) standards that is used to create unique identifiers for networks and individual devices. The IP addresses allow particular information packets to be sent to specific computers. Shortly after the introduction of CIDR, technicians found it difficult to track and label IP addresses, so a notation system was developed to make the process more standardized. That system is known as CIDR notation. efficient and

CIDR IP addresses consist of two groups of numbers, which are also referred as groups of bits. The most important of these groups is the network address, and it is used to identify a network or a sub network (subnet). The lesser of the bit groups is the host identifier. The host identifier is used determine which host or device on the network should receive incoming information packets. In contrast to classful routing, which categorizes addresses into one of three blocks, CIDR allows for blocks of IP addresses to be allocated to Internet service providers. The blocks are then split up and assigned to the provider customers. Until recently, IP addresses used the IPv4 CIDR standard, but because IPv4 addresses are nearly exhausted a new standard known as IPv6 has been developed and will soon be implemented.

used for IP addresses, but early Internet developers soon discovered that it included a serious flaw in that lacked scalability. To solve this problem, the Internet Engineering Task Force created the IPv4 standard 1993. In addition, CIDR was created as a system of routing the new IPv4 addresses. These

Development of CIDR

When the Internet domain name system (DNS) was first established, the classful routing system was standards were originally published under the names RFC 1518 and RFC 1519. In 2006, a new version of the standard was published as RFC 4632.

According to the CIDR standard, the first part of an IP address is a prefix, which identifies the network. The prefix is followed by the host identifier so that information packets can be sent to particular computers within the network. With the classful routing system, individual networks were either limited to 256 host identifiers or overburdened with 65,536 identifiers. For many network enterprises, 256 identifiers were not enough and 65,536 were too burdensome to be used efficiently. In the 1980s, as TCP/IP grew into the modern Internet, the need for a more flexible routing system was recognized. This need prompted the development of CIDR and subnets. CIDR and the process of variable-length subnet masking (VLSM) allow network administrators to divide individual networks into subnets of various sizes. In addition, addresses for related operations can be grouped together to create a simple system of categorization. Internet providers are also able to allocate a scalable number of addresses, in blocks, to organizations based on how many addresses are needed. These new routing and categorization systems solved most of the problems with IP addresses, and the only remaining problem was deciding how to identify them efficiently. Eventually, CIDR notation was established and accepted as the standard. In CIDR notation, IP addresses are written as a prefix, and a suffix is attached to indicate how many bits are in the entire address. The suffix is set apart from the prefix with a slash mark. For instance, in the CIDR notation 192.0.1.0/24, the prefix is 192.0.1.0, and the total number of bits in the address is 24.

CIDR Blocks

The ability to group blocks of addresses into a single routing network is the hallmark of CIDR, and the prefix standard used for interpreting IP addresses makes this possible. CIDR blocks share the first part of the bit sequence that comprises the binary representation of the IP address, and blocks are identified using the same decimal-dot CIDR notation system that is used for IPv4 addresses. For example, 1010116 /32 is an address prefix with 32 bits, which is the highest number of bits allowed in IPv4. Addresses with identical prefixes and the same number of bits always belong to the same block. In addition, larger blocks can be easily distinguished from smaller blocks by the length of the prefix. Short prefixes allow for more addresses while large prefixes identify small blocks. CIDR notation is also used for the newer IPv6 standard, and the syntax is the same. The only difference is that IPv6 addresses may contain up to 128 bits instead of the 32-bit maximum of IPv4. Even though IPv6 addresses may be up to 128 bits in length, it is important to note that subnets on MAC layer networks always use 64-bit host identifiers.

The assignment of CIDR blocks is handled by the Internet Assigned Numbers Authority (IANA). One of the duties of the IANA is to issue large blocks of IP addresses to regional Internet registries (RIRs). These blocks are used for large geographical areas, such as Europe, North America, Africa and Australia. It is then the duty of each RIR to create smaller, but still quite large, blocks of IP addresses to be assigned to local Internet registries (RIRS). Depending on the organization of regional and local registries, blocks may be subdivided further until they are assigned to end users. The size of blocks assigned to end users is dependent on how many individual addresses will be required by each user. Most end users receive their blocks from a single Internet service provider (ISP), but organizations that make use of multiple ISPs must obtain provider – independent blocks directly from an LIR or RIR.

Subnet Masks

Once blocks of IP addresses within a private network, which is a process known as subnetting. Computers and other connected devices within a particular subnet can be identified because they all use the same IP address prefix. The subnet identifier then becomes the most significant portion of the host identifier. Finally, the last part of the host identifier is used to distinguish individual computers on a subnet. The subnet identifiers within network are assigned according to the network's subnet mask, which is a binary pattern that is used to determine how many subnets are available in a network. In its binary form, a subnet mask begins with a series of ones and ends with a series of zeros. However, subnet masks are usually expressed using the familiar dot-decimal notation used for IP addresses and network prefixes.

In this notation, the series of ones become the number 255. For example, the most common subnet mask expressed using this notation is 255.255.255.0. This and it is used when only one subnet is required or as the first subnet mask is known as subnet zero, of multiple subnets. A specific subnet mask is created by designating a portion of the host identifier, and larger subnets are created by moving more bits from the host identifier to the subnet mask. The final subnet of a network is designated in binary with all ones. When using the CIDR dot-decimal notation, the final 255.255.255.255 with all ones subnet is expressed as

Before CIDR, subnet masks with all zeros 0.0.0.0 and subnet masks 255.255.255.255 could not be used because they could become confused with network identifiers, but CIDR-compliant equipment use the prefixes and suffixes of CIDR notation distinguish between the two.

ISP (INTERNET SERVICE PROVIDERS)

ISP is an organization or business offering public access to the Internet. It is our gateway, to the Net.

It offers access to the Internet, allowing users to travel and explore using tools like Internet explorer. These days a lots of them are available including BSNL, MTNL, etc. check with your local area ISP since this would be easier to work with. We usually ring up ISP for any internet connection.

This is why you are charged for a single call even if you are trying to access a website which is based in USA, some ISPs like AOL or MSN are Online Service Providers. They include their own content such as bulletin boards, financial information etc, as well as offering access to public Internet sites. Other ISPs provide strictly access service users must get their content from public web sites.

Types of ISP

There are many types of Internet providers. We can, for instance, choose one of the big commercial on-line service providers. The primary business of an ISP is looking people to the Internet by giving an Internet account to subscribers, and providing them with two different kinds of access:

1. Shell access and

2. SLIP/PPP access.

Most ISP offers both kinds of access, some offer both with a single account and others require that you choose one or the other. Once you register, your provider will give you a user name (called a user ID), a password, and a phone number to dial. To establish the Internet connection, you have your communications program dial the number. You then log in using your particular user ID and password. At present it is VSNL (Videsh Sanchar Nigam Limited) which is dominating the Internet scene in India through its GIAS (Gateway Internet Access Service). The other service providers in India are MTNL (Mahanagar Telephone Nigam Limited). Mantra –online, Airtel, Reliance Communications and Satyam–online.

TCP/IP

TCP/IP is a name given to the collection of networking protocols that have been used to construct the global Internet. It is a family of more than 100 data communications protocols used to organize computers into networks. The computers that make up the internet, talk to each other in the language of TCP/IP protocol. Any computer that can talk to each other in the language of TCP/IP protocol. Any computer that can talk the language of TCP/IP can be a direct part of the internet that is a part of the reason. Why there is such a wide variety of computers. TCP/IP specifies an addressing scheme for computers on Internet. TCP/IP sets the rules for how data should move between computers must follow in order to move different types of information from place to place.

Advantages of TCP/IP There are following advantages of TCP/IP

- It enables dissimilar systems to be connected.
- It is a protocol standard for the internet.
- It is the most widely used protocol suite by all modern operating system.
- It has many utilities available for troubleshooting.
- It is routable
- It is a scalable client/server framework. (vii) It has been modified and tested for many years and this is a proven protocol.

TCP/IP Layers

TCP/IP stands for transmission control protocol: Internet protocol TCP/IP is a to allow co-operating computers to share resources across a network TCP/IP has

- i. **Application layer**
- ii. **Transport layer**
- iii. **Internet layer**
- iv. **Network access layer**, the network access layer is split into physical and datalink components. There are above four protocol layers that data goes through before it gets sent off to another computer. If the message to be sent is long, it will be broken into smaller

(i) Application Layer: This layer is broadly equivalent to the application, presentation and session layer of OSI model. It gives an application access to the communication environment. The protocols used in application layer are: FTP, SMTP, HTTP, DNS, etc.

(ii) Transport Layer: The transport layer just above the internetwork layer is the host-to-host layer. It is responsible for end-to-end data integrity. The two most commonly protocols used at the layer are TCP (Transmission Control Protocol) and UDP (User Datagram Protocol). Both protocol deliver data between the application layer and the internetwork layer. The only difference between TCP and UDP is that TCP provides the guaranteed delivery of data and controls the high

traffic in network because it is connection oriented protocol whereas UDP is connection less and does not guarantee the delivery of data.

(iii) Internet Layer: This layer is responsible for the routing and delivery of data across network. It allows communication across networks of the same and different types and carries out translation to deal with dissimilar data addressing scheme. The IP addresses are used by internetwork and higher layers to identify devices and to perform internetwork routing. The address Resolution Protocol (ARP) enables IP to identify the physical address that matches given IP address. The internet layer of TCP/IP support four supporting protocols.

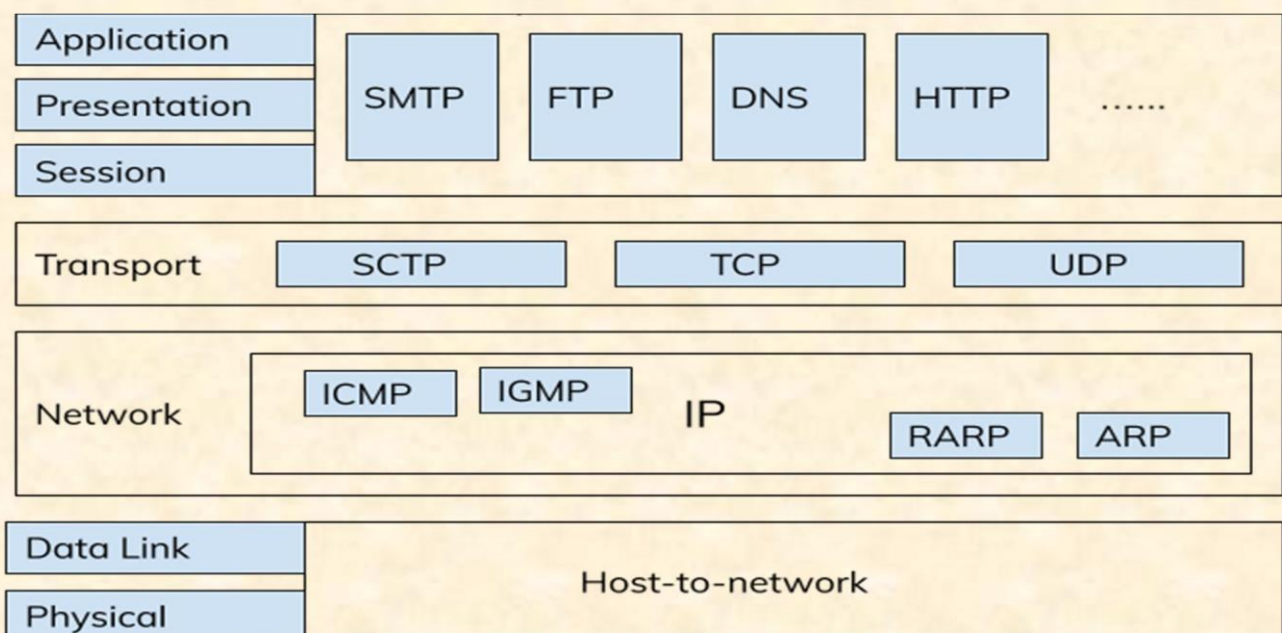
(a) ARP (b) RARP (c) ICMP (d) IGMP

The internet protocol is the transmission mechanism used by the TCP/IP protocol. It is an unreliable and connectionless protocol a best effort delivery service. IP assumes the unreliability of the underlying layers and does its best to get transmission through to its destination, but with no guarantees.

IP transports data in packets called datagrams, each of which is transported separately. Datagrams can travel along different routes and can arrive out of a sequence or be duplicated. IP does not keep track of the routes and has no facility for reordering datagrams once they arrive at their destination. ARP (Address Resolution Protocol) is used to associate a logical address with a physical address. RARP (Reverse Address Resolution Protocol) allows a host to discover its internet address when it knows only its physical address. It is used when a computer is connected to a network for the first time.

ICMP (Internet Control Message Protocol) is a mechanism used by hosts and gateways to send notification of datagram problems back to the sender. IGMP (Internet Group Message Protocol) is used to facilitate the simultaneous transmission of a message to a group of recipients. with pure hardware and access methods such as

(iv) Network Access Layer: The design of TCP/IP hides the function of this layer from users. The combination of data link and physical layer deals CSMA/CD (Carrier Sense Multiple Access with Collision detection). This layer encapsulates the IP datagrams into frames that are transmitted by the network. It also maps the IP addresses to the physical addresses used by the network. The following diagram shows the layers and the protocols in each of the layers:



INTERNET AND WEB TECHNOLOGY (IWT)



Chapter 2: Internet Connectivity & WW W

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

Chapter 2 - Internet Connectivity & WWW

Introduction to connectivity

- **Internet connectivity** or connections support a company's business in a number of ways. Company employees use the Internet for a number of reasons, including the following:
 - Exchange e-mail with other employees at different branch offices, and with business partners and suppliers.
 - Access the LAN when working from home.
 - Find valuable information, or conduct research using the Web
 - Mobile users utilize the Internet to remotely access the LAN.
 - The Internet also provides the means for other organizations to connect to the company to perform business transactions.

Types of Internet Account

To connect to the net, we run a communication programs to dial the number of an Internet service provider. When the remote system answers the phone, your modem talks to the other system and a connection is established.

There are two possible types of connections:

- Shell Account
- PPP Account

PPP Account

The PPP Account under Computer Networking actually represents the Point -to-Point Protocol, which is used to establish Direct Connection between two Nodes (Devices). This protocol performs all the necessary tasks related to Authentication of Connection, Encrypt and Compress Data during Transmission between the two devices that establish Connection.

This PPP Protocol is used in various types of Physical Networks like Serial Cable, Phone Line, Trunk Line, Cellular Telephone, Specialized Radio Links and Fiber Optic Links. In addition, this protocol is also used by various ISPs (Internet Service Providers) to provide Internet Service through Dialup Connections to customers. That is, when the user wants to connect to the Internet through Dialup Connections, then his ISP provides an account to that user to establish Dialup Connection, it is called PPP Account.

Under PPP Account, this PPP Protocol is basically used as two other Protocols Point -to-Point Protocol over Ethernet (PPPoE) and Point -to-Point Protocol over ATM (PPPoA), which is actually PPP Protocol Inherit only the different functionalities.

When we want to get Fast Internet Connection, then register for Fast Internet Connection with an Internet Service Provider, as a result of that Service Provider accepts our registration and creates a

new account for us, so that it can be used by us. We can keep information about the Internet services to be done, how much Internet we have used and can charge us according to the amount of Internet we have used. This account is actually called PPP Account.

Shell account

With this type of account we actually do your work on the remote computer. Your computer is not at the net level; rather your computer emulates a terminal, which allows you to work directly with the remote host.

When we use shell account, the only program that runs on your computer is the terminal emulator. All the Internet clients actually run on the remote computer. The nature of a terminal emulator is that it only supports a text-based interface, not a graphical interface.

The client programs run on the remote host not on your computer. All your computer does is emulate a terminal. This means you do not run programs on your computer, so you can't take advantage of the characteristics of your computer; you are limited to the capabilities of the terminal your computer is emulating.

The main three disadvantages to a shell account:

1. The interface is text based not graphical
2. Your computer is not on the Internet.
3. You are usually limited to running one client at a time.

Medium and methods of connectivity, ISDN, VSAT, RF Link

Communication Mediums used for Internet Connectivity

The internet is a worldwide system of computers exchanging various types of information among themselves. The following are the communication mediums used for internet connectivity:

1. Modem:

Modem stands for modulation and demodulation. It is an electronic device. It is used to share information between users and computers. It converts analog signals to digital signals and vice-versa. It is the key that unlocks the world of internet and its world wide web, commercial online service, electronic mail etc. When you connect to the internet your computer tells your modem to make this connection. When you send a request to retrieve data from a particular web server, your modem takes care of "translating" this request so that it can be understood by the other end.

2. Dial-Up Account:

The most basic type of Internet connection is called a dial-up connection. This connection is made through a modem, the communication mechanism in computers that uses a telephonic line to connect to the Internet. The modem must dial the telephone every time it wants to connect to the Internet, hence the name dial-up connection. Most people connect to the Internet by using a modem and phone line to dial in to a PPP account on an Internet provider's computer. Most ISPs support modems at speeds 14.4Kbps, 28.2Kbps, and 56Kbps.

3. ISP:

Internet service provider is a business organization that provides the internet facility to consumers and related services. It is a gateway to the Net. You have to subscribe to a provider for your internet. You use your computer and modem to access the providers system and the provider handles the rest of the details of connecting you to the internet. For example: MTNL, BSNL, Reliance, Airtel etc.

4. ISDN —Dial Up Connection:

The second type of dial-up connection is through an ISDN (Integrated Services Digital Network). This connection is a high-speed dial-up and requires a special type of dedicated digital telephone line. The type of connection improves speed because the signals received are already in a digital format, but it is costly. For a dial-up connections were the only way to access the Internet. Now, however broadband connections are available in many parts of the country as telephone and cable companies install new lines to accommodate Internet connections.

5. Broadband Connections:

Broadband is a high speed Internet connection that makes surfing the Web more enjoyable. It also easily accommodates the video, audio, or complex graphics that are becoming commonplace on the Internet. Broadband is a type of data transmission in which a single medium or wire can carry several channels or communications paths at once. So broadband cable connections are able to transmit both television signals and Internet data at the same time. Broadband telephone connections like DSL are able to transmit both voice and data over the same line at the same time.

Broadband connections are always on that means if your computer is on, you will be connected to the internet with no dial-up involved. For most average users, a broadband connection will mean accessing the Internet by either a cable modem provided by their local cable company or a DSL modem and DSL telephone line provided by their local telephone service provider.

6. Internet through Cable:

Internet over cable is a technology and service that uses the existing cable TV networks (made up of coaxial cable lines that bring television signals to TV to bring internet data to PC or TV at a very high speed. The television and the Internet transmission take place simultaneously on the same cable but at different frequencies. This allows the user to view TV and access Internet at the same time. There are only two drawbacks to cable:

First, it is a shared connection, meaning you share the “pipeline” with your neighbours. That doesn't mean that your neighbours will know what you are doing on the Internet.

The second drawback to cable is also minor. Because cable modem connections are always on, they, like DSL connections make you more vulnerable to hacking and security breaches. For this reason, many cable companies are now providing their customers firewall software to help protect their security.

7. Internet Connectivity through Leased Line :

Leased lines are dedicated circuits provided by Basic Service Providers (BSPs), which provide permanent connectivity to the internet. Leased lines provide the last mile access from the user premises to the ISP. They provide permanent connection as compared to the temporary

connectivity through dialup access. The quality of the connection is far superior to what is normally available through dialup, thanks to digital signaling, less noise, fewer exchanges, etc.

Advantages

It provides permanent, reliable, high speed connectivity as compared to the temporary connectivity of dial up access. The quality of the connection is far superior to what is normally available through dialup, because of the digital signaling, less noise, fewer exchanges etc.

Disadvantages

Leased bandwidth prices are quite high, compared to dialup bandwidth of comparable size. Permanent connectivity to the Net exposes the organization to a variety of threats including hacking, malicious code including active vandals, viruses, Trojan Horses, macros, denial of service attacks etc.

8. Communication Software:

To access an internet you need software and that software is called communication software. This software can display text, video, sound etc. over the internet. Example is internet explorer, opera etc. This type of communication software is also known as web browsers.

9. Wi-Max

Wi-max is a standard based technology enabling the delivery of last mile wireless broadband access as an alternative be wired technology to DSL or cable modem. WiMa x stands for worldwide interoperability for Microwave access WiMax support both fixed and mobile wireless broadband.

Methods of Internet Connectivity

To connect internet with computer various devices needed like a modem, communication software, a dialup telephone line, an interent service provider, a computer with either a serial port for an external modem or an expansion slot for an internal modem. There are several ways of internet connectivity. These are as follows:

(1) Connection Through LAN: If a LAN uses TCP/IP protocols for communication within the network. It is a simple matter to connect to the internet through a router, another computer that stores and towards set of data to another computer on the internet.

(2) Direct Connection: In this connection, internet programs run on the local computer, which uses the TCP/IP protocols to exchange data with computer through the internet. An isolated computer can connect to the internet through a serial data communication port using either serial line interface protocol or point to point protocol

(3) Remote Terminal Connection: In this connection internet exchanges commands and data in ASCII text format with a host computer the uses UNIX. The TCP/IP application programs and protocols all run on the host because the command set is UNIX is called a shell, this kind of internet access is called shell account.

(4) Gateway Connection: Many networks don't use TCP/IP commands and protocols may still provide some services like e-mail, file transfer etc. These networks are gateway that converts commands and data to and from TCP/IP format.

(5) Connecting through a Modem: An isolated computer can connect to the internet through a serial data communication port and a modem, using either a shell account and a terminal emulation program or a direct TCP/IP connection with SLIP or PPP account if there is no LAN on site. TCP/IP connections let users to run application programs with graphical user interface.

(6) High Speed Data Link: Modem connections are convenient, but their capacity is limited to the relatively low data transfer speed of a telephone line. A 56 kbps modem is fine for text, but it is not really practical for huge digital audio and video files. Now a day, dedicated high speed data circuits are provided by telephone companies, Cable TV services and other suppliers ISDN (Integrated Service Digital Network) is a new digital telephone service that combines voice data and control signaling through a single circuit.

ISDN

Integrated services digital network is a new set of protocols that combine digital telephone and data transfer services. The main idea of an ISDN network is to digitize the telephone network to permit the transmission of audio, video and text over the existing telephone network. ISDN provides many features like telephone lines with multiple buttons for instant call setup to any telephone all over the world. It allows telephone with called identifications, video phones and the phones connected to computer databases. The main goal of ISDN is to create a wide area network that provides universal end to end connectivity over digital media.

The ISDN only allows voice communication through network . Presently, the demands of customers have been increased so far and they demand transfer of textual, audio and video data in addition to traditional voice telephone services. To satisfy the demands of customers the digital technologies and devices are being used in the networks. Several technologies are used to provide such services.

Broadband ISDN is the name given to entire ISDN service. It includes low endolder ISDN services (only voice services) and modern high end services. The low end ISDN provides traffic at a rate of 64 kbps. This rate is enough to carry voice data. To carry audio, video and other signals, BIsDN offers much higher data rates.

Services of ISDN

The services provided by ISDN can be divided into three main classes. These are:

(i) Bearer Services

(ii) Tele Services

(iii) Supplementary Services

- (i) **Bearer Services:** Bearer services provide means to transfer information (audio, video and text) between the users of ISDN in such a manner that network does not change or process the information being transmitted. The examples of bearer service are circuit switching, packet switching, frame switching and cell switching. These services belong to the first three layers of OSI models are well defined in ISDN archicecture.
- (ii) **Tele Services:** Tele services provide means to transfer information between users in such a manner that the network may change or process the contents of information. The examples of tele services are telephone, telefax and teleconferencing. Tele services correspond to fourth to seventh layer of OSI model and rely on the facilities of bearer

services. Tele services have been designed to accommodate complex user needs while encapsulation, actual working from the user.

- (iii) **Supplementary Services:** Supplementary services are the services that provide additional functionality to tele services and bearer services. Examples of Supplementary services are called waiting, call diverting, revenue charging and message handling etc. Most of these services are provided on telephone connects of modern private companies.

ISDN Structure

ISDN structure includes a central ISDN office. All the users are linked to the office through a digital pipe. The digital pipe may be of different capacities and any may have different data transfer rates. These digital pipes between the customers and central office are organized into multiple channels of different size. ISDN standard defines following three channel types:

- (i) **B Channel:** A bearer channel is defined at a data rate 64 kbps. It is a basic user channel and it can carry any type of digital information in full duplex mode, as long as required transmission rate does not exceed 64 kbps. It can be used to carry digital data, digital voice and any other low data rate information
- (ii) **D Channel:** Data Channels carry control signal for bearer channels. The name of these channels is a misnomer as they don't carry data, It may be of 16 kbps or 64 kbps depending upon the user needs, if we are using in band signaling then same cable carries data as well as control signals, but if we are using out of band signaling (as in ISDN) two different cables carry data and control information together.
- (iii) **H Channel:** Hybrid channels are available with data rates of 384 kbps, 1536 kbps or 1920 kbps. Hybrid channels are used for high data rate applications such as video and teleconferencing etc.

Network termination devices are present to control the electrical and physical termination of ISDN. These ISDN channels are delivered to the user one of two pre-defined configurations:

(1) Basic Rate Interface (BRI): BRI is the ISDN service most people use to connect to the Internet. An ISDN connection supports two 64 kbps B-channels and one 16 kbps. D-channel over a standard phone line, BRI is often called "2B+D" referring to its two B-channels and one D-channel. The D-channel on a BRI line can even support low-speed (9.6 kbps) X.25 data, however, this is not a very popular application in the United States.

(2) Primary Rate Interface (PRI): ISDN PRI services are used primarily by large organizations with intensive communications needs. An ISDN PRI connection supports 23.64 kbps. B-channels and one 64 kbps, D-channel (or 23B + D) over a high speed DSI circuit.

Application of ISDN

- Popular ISDN applications are:
- Internet access
- Telecommunicating/remote access to corporate computing
- Video conferencing

- Small and home office data networking

Advantages of ISDN

1. ISDN is a mature technology, people are using, ISDN from year 1980. It is time tested and trusted technology.
2. ISDN is governed by a world-wide set of standards.
3. ISDN provides symmetrical transfer rates: the transmit rate is the same as the receiver rate.
4. It has consistent transfer rates, If you have a 64 kbps bearer channel then you can transfer data of 64 kbps.
5. It is competitively priced compared to other technologies.

Disadvantages of ISDN

1. An external power supply is required, If the power fails, the phones won't work.
2. Special digital phones are required or a terminal adapter required to talk to the existing PSDN devices.

VERY SMALL APERTURE TERMINAL (VSAT)

The term Very Small Aperture Terminal (VSAT) refers to a small fixed earth station. VSATs provide the vital communication link required to set up a satellite based communication network. VSATs can support any communication requirement be it voice, data or video conferencing.

The VSAT comprises of two modules:

1. An outdoor unit
2. An indoor unit.

The outdoor unit consists of an Antenna and Radio Frequency Transceiver (RFT). The antenna size is typically 1.8 metre or 2.4 metre in diameter, although smaller antennas are also in use. The indoor unit functions as a modem and also interfaces with the end user equipment like stand alone PCs, LANs, Telephones or an EPABX,

The outdoor unit is generally ground or even wall mounted and the indoor unit which is the size of a desktop computer is normally located near existing computer equipment in your office.

Advantages of VSATs

1. **Last Mile Problem:** Let us begin with the situation where you have reliable high-speed links between city exchanges for meeting your communication requirements. But before you begin to feel comfortable, connections from the nearest exchange to your company's office often fail. Consequently, stretching what is technically called the last mile problem into much longer distances. VSATs located at your premises guarantee seamless communication even across the last mile.

2. Reach: You must be well aware of the limitations faced by terrestrial lines in reaching remote and other difficult locations. VSATs, on the other hand, offer you unrestricted and unlimited reach.

3. Reliability: Uptime of upto 99.5 percent is achievable on a VSAT network. This is significantly higher than the typical leased line uptime of approximately 80 to 85 percent.

4. Time: VSAT deployment takes no more than 4–6 weeks as compared to 4 to 6 months for leased lines.

5. Network Management: Network monitoring and control of the entire VSAT network is much simpler than a network of leased lines, involving multiple carriers at multiple locations.

6. Maintenance: A single point contact for operation, maintenance, rapid fault isolation and troubleshooting makes things very simple for a client, using VSAT services. VSATs also enjoy a low mean time to repair of a few hours, which extends upto few days in the case of leased lines. Essentially, lesser elements imply lower MTTR.

7. Flexibility: VSAT networks offer enormous expansion capabilities. This features in changes in the business environment and traffic loads that can be easily accommodated on a technology migration path. Additional VSATs can be rapidly installed to support the network expansion to any site no matter how remote.

8. Cost: A comparison of costs between a VSAT network and a leased line network reveals that a VSAT network offers significant savings over a two to three years time frame.

RF LINK

- RF deploys state of the art devices to provide high bandwidth with latency almost equal to that of fiber ensuring applications like Video conferencing, voice applications, SAP, ERP, etc. work seamlessly.
- RF devices are constantly upgraded to support long distance connectivity with the best in class encryption standards.
- The RF devices are so designed that the radio signals will not be impacted even when the climatic conditions vary from heavy rains to extreme heat.
- 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 Wi-Fi or Bluetooth.
- Wi-Fi means wireless, which means communication by RF, or Radio Frequency. WiFi is a registered name for RF communication, mostly between computers, since 1997.

Working of Internet

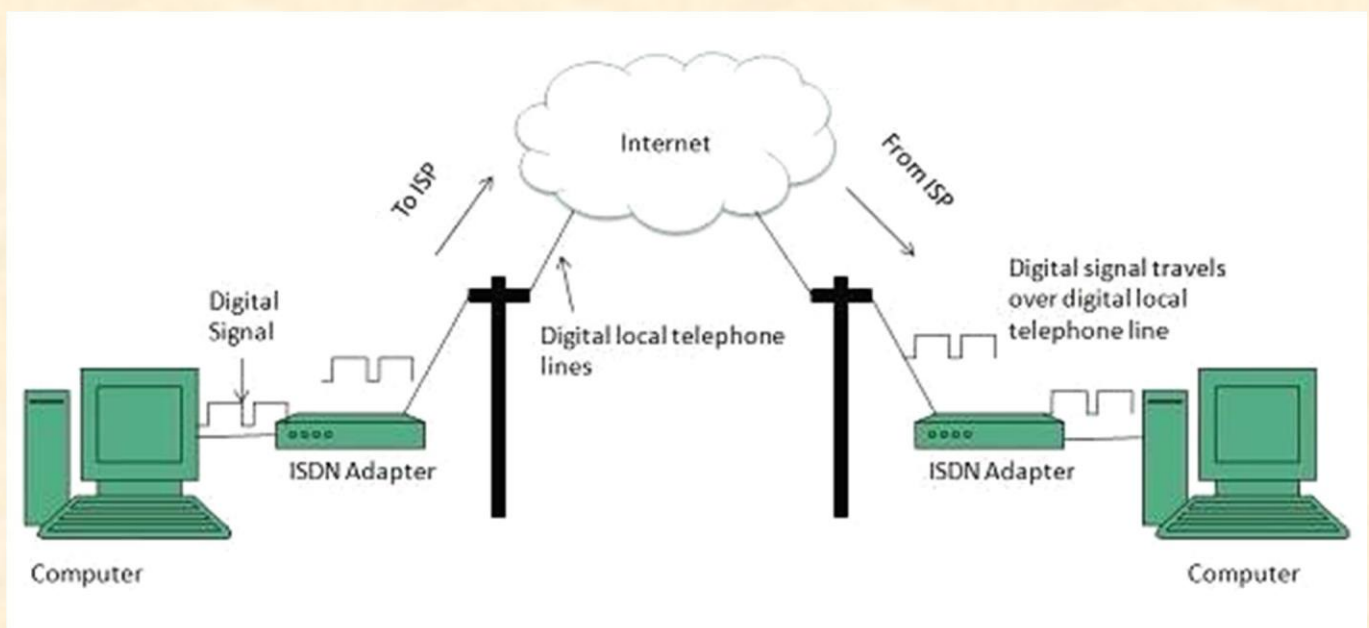
The Internet's growth has become explosive and it seems impossible to escape the bombardment of www.com's seen constantly on television, heard on radio, and seen in magazines. Because the internet has become such a large part of our lives, a good understanding is

needed to use this new tool most effectively. It is important for the students to understand the working of internet. The working of internet can be seen in following steps:

1. Connection to internet.
2. Communication with other network.
3. Networking infrastructure.
4. Internet infrastructure.
5. Internet routing hierarchy.
6. Domain name resolution.

(1) Connection to internet

The picture below illustrates two computers connection to the internet; your computer with IP address 1.2.3.4 and another computer with IP address 5.6.7.8 The Internet is represented as an abstract Object in-between.



If you connect to the Internet through an Internet Service Provider (ISP), you are usually assigned a temporary IP address for the duration of your dial-in session. If you connect to the internet from a local area network (LAN) your computer might have a permanent IP address or it might obtain a temporary one from a DHCP (Dynamic Host Configuration Protocol) server. In any case, if you are connected to the Internet, your computer has a unique IP address.

(2) Communication With Other Computers

So your Computer is connected to the Internet and has a unique address. How does it 'talk' to other computers connected to the Internet? An example should serve here: Let's say your IP address is 1.2.3.4 and you want to send a message to the computer 5.6.7.8 The message you want to send is "Hello Computer 5.6.7.8!", Obviously the message must be transmitted over whatever Kind of wire connects your computer to the Internet. Let's say you've dialled into your ISP from home and the message must be transmitted over the phone line. Therefore the message must be translated from alphabetic text into electronic signals, transmitted over the Internet, then translated back into alphabetic text. How is this accomplished? Through the use of a protocol Stack. Every Computer needs one to communicate on the Internet and it is usually built into the

Internet is referred to as the TCP/IP protocol stack because of the two major communication protocol used. The TCP/IP stack looks like this :

S.NO.	Protocol Layer	Comments
1.	Application Protocol Layer	Protocol specific to application such as WWW, email, FTP, etc.
2.	Transmission Control Protocol Layer	TCP direct packets to a specific application on a computer using a port number.
3.	Internet Protocol Layer	IP directs packets to a specific computer using an IP address.
4.	Hardware Layer(Physical Layer)	Convert binary packets data to network signals and back. (E.g. Ethernet network card, modem for Phone lines, etc.)

If we were to follow the path that the message “Hello computer 5.6.7.8!” took from our computer to the with IP address 5.6.7.8, it would happen something like this:

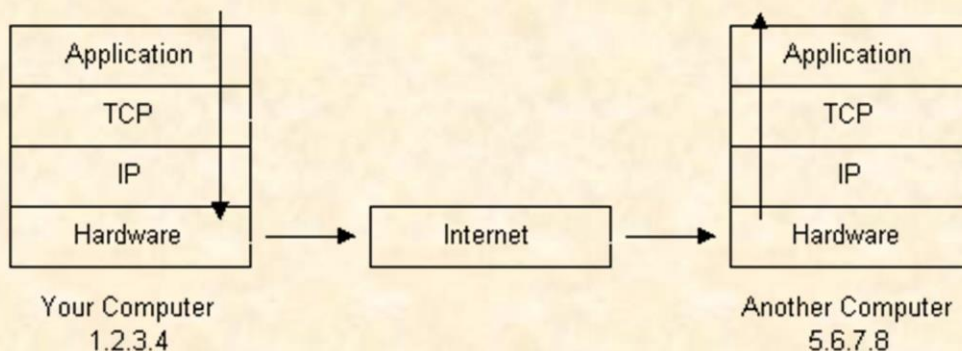


Fig. Flow of Message on Internet.

1. The message would start at the top of the protocol stack on your computer and work its way downward.
2. If the message to be sent is long, each stack layer that the message passes through may break the message up into smaller chunks of data. This is because data sent over the Internet (and most computer networks) are sent in manageable chunks. On the Internet, these chunks of data are known as packets.
3. The packets would go through the application Layer and continue to the TCP layer. Each packets is assigned a port number. Ports will be explained later, but suffice to say that many programmes may be using the TCP/IP stack and sending message. We need to know which program on the destination computer needs to receive the message because it will be listening on a specific port.
4. After going through the TCP layer, the packets proceeds to the IP layer. This is where each packets receives its destination address, 5.6.7.8.

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

(3) Networking Infrastructure

So now you know how packets travel from one computer to another over the Internet. But what's in-between? What actually makes up the Internet? Let's look at another diagram:

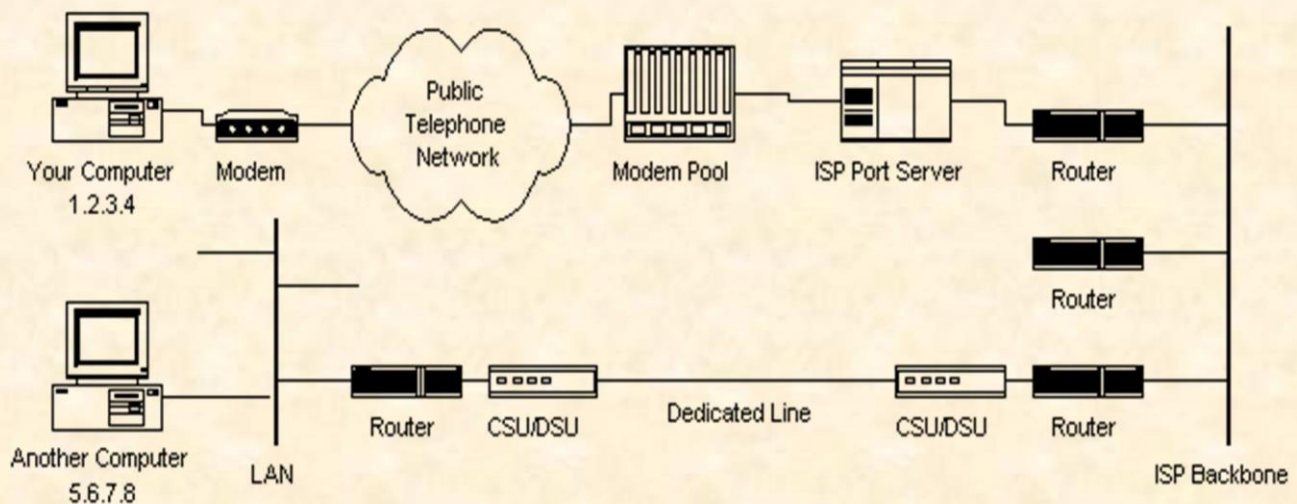


Fig. Packet Travelling Through Internet.

The ISP maintains a pool of modems for their dial-in customers. This is managed by phone form of computer (usually a dedicated one) which controls data flow from the modem pool to a backbone or dedicated line router. This setup may be referred to as a port server, as it's server's access to the network. Billing and usage information is usually collected here as well.

After your packets traverse the phone network and your ISP's local equipment, they are route onto the ISP's backbone or a backbone the ISP buys bandwidth from. From here the packets will usually journey through several backbones, dedicated lines, and other network until they find destination, the computer with address 5.6.7.8.

(4) Internet Infrastructure

The Internet backbone is made up of many large networks which interconnection with each other. These large network are known as Network Service Provides or NSPs.

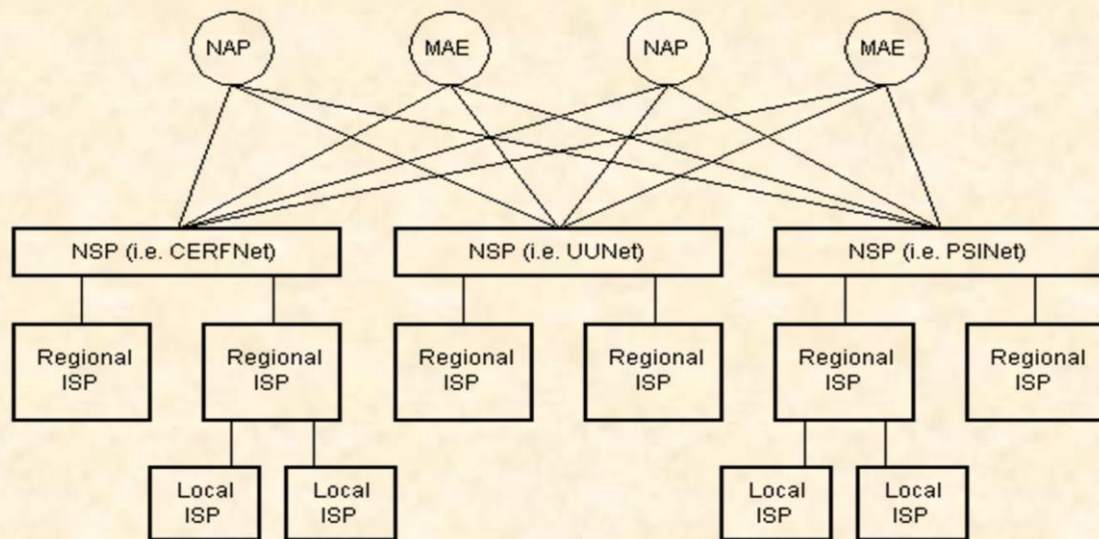


Fig. Internet Infrastructure

Some of the large NSPs are UUNet, CerfNet, IBM, BBN Planet, SprintNet, PSINet, as well as other. These network peer with each other to exchange packets traffic. Each NSP is required to connect to three Network Access Points or NAPs, At the NAPs, packet traffic may jump from one NSP's backbone to another NSP's backbone. NSPs also interconnect at Metropolitan Area Exchange or MAEs. Mae's server the same purpose as the NAPs but are privately owned. NAPs were the original Internet interconnection points, Both NAPs and MAEs are referred to as Internet Exchange Points or IXs. NSPs also sell bandwidth to smaller networks, such as ISPs and smaller bandwidth providers. Below is a picture showing this hierarchical infrastructure.

(5) The Internet Routing Hierarchy

So how do packets find their way across the Internet? Does every computer connected to the Internet know where the other computers are? Do packets simply get 'broadcast' to every computer on the Internet? The answer to both the preceding request is 'no', No computer knows where any of the other computers are, and packets do not get sent to every computer. The information used to get packets to their destinations are contained in routing tables kept by each router connected to the Internet.

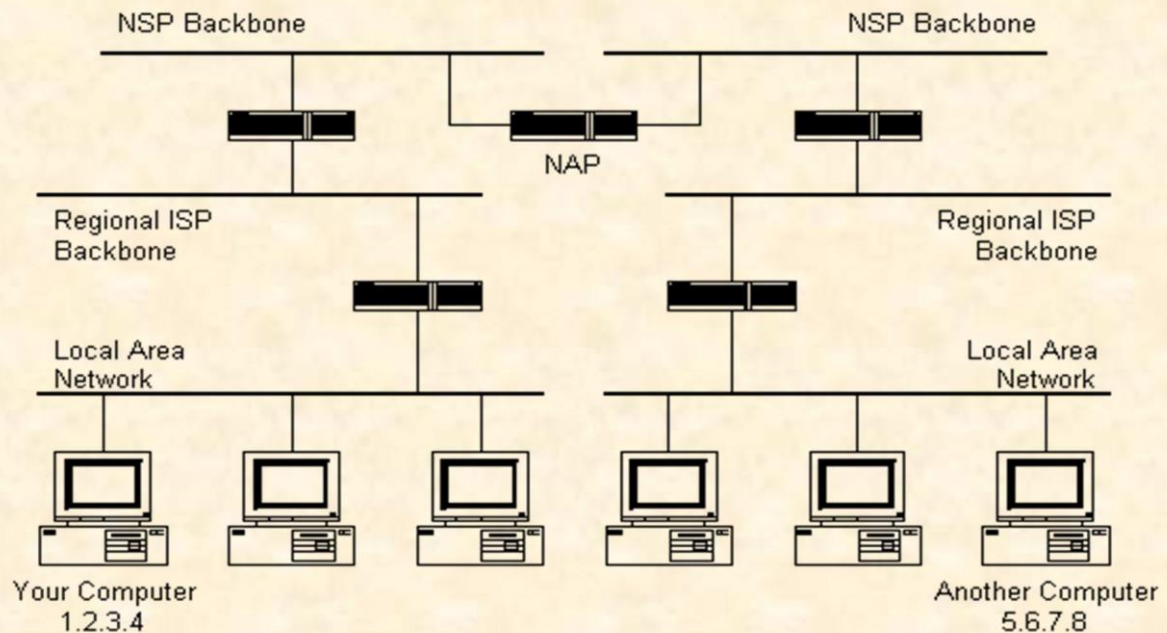


Fig. Internet Routing Hierarchy.

Routers are packet switches. A router is usually connected between networks to route packets between them. Each router knows about its sub-networks and which IP address they use. The router usually doesn't know what IP addresses are 'above' it. Examine Fig. The black boxes connecting the backbones are routers. The larger NSP backbones at the top are connected at a NAP (Network Access Point). Under them are several sub-networks, and under them, more sub-networks. At the bottom are two local area networks with computers attached.

When a packet arrives at a router, the router examines the IP address put there by the IP protocol layer on the originating computer. The router checks its routing table. If the network containing the IP address is found, the packet is sent to that network. If the network containing the IP address is not found, the packet is sent to that network. Then the router sends the packet on a default route, usually up the backbone hierarchy to the next router. Hopefully the next router will know where to send the packet. If it does not, again the packet is routed upwards until it reaches a NSP backbone. The router connected to the NSP backbones holds the largest routing tables and here the packet will be routed to the correct backbone, where it will begin its journey 'downward' through smaller and smaller networks until it finds its destination.

(6) Domains Names and Address Resolution

But what if you don't know the IP address of the computer you want to connect to? What if you need to access a web server referred to as www.anothercomputer.com? How does your web browser know where on the Internet this computer lives? The answer to all these questions is the Domain Name Server or DNS. The DNS is a distributed database which keeps track of computer's names and their corresponding IP addresses on the Internet.

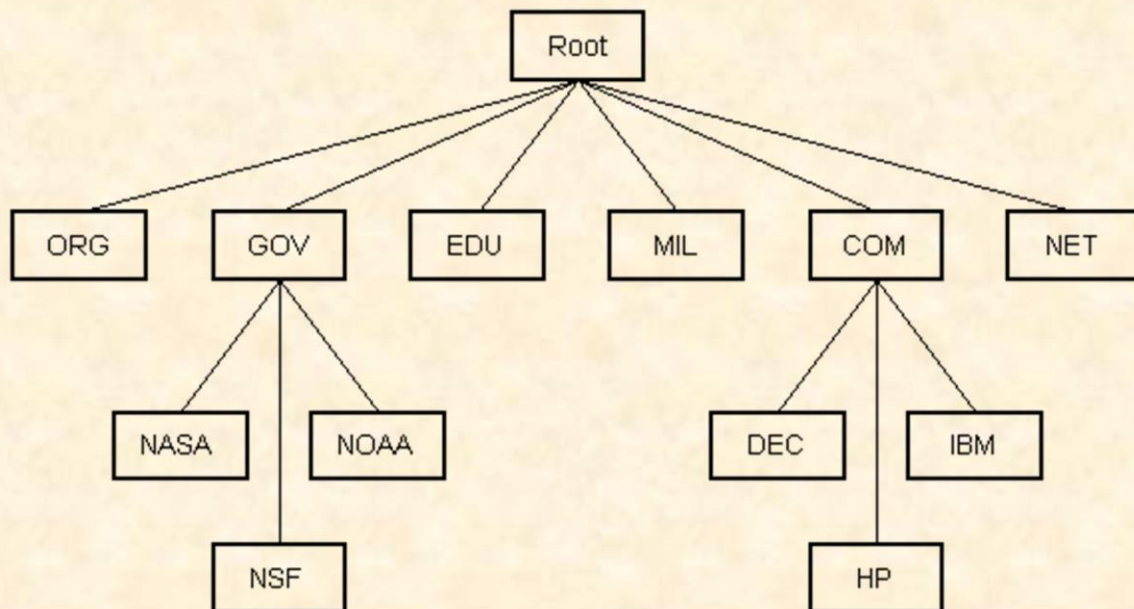


Fig. Domain Names and Address Resolution.

Many computing connected to the Internet host part of the DNS database and the software that others to access it. These computers are known as DNS servers. No DNS server contains the entire database; they only contain a subset of it. If a DNS server does not contain the domain name requested by another computer, the DNS server redirects the requesting computer to another DNS server.

The Domain Name Server is structured as a hierarchy similar to the IP routing hierarchy. The computer requesting a name resolution will be re-directed 'up' the hierarchy until a DNS server is found that can resolve the domain name in the request. At the top of the Tree are the domain roots. Some of the older, more common domains are seen near top. What is not shown are the multitude of DNS servers around the world which form the rest of the hierarchy.

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

Introduction to WWW, Application Level Protocol

Introduction to WWW

The world wide web was developed in 1989 by Tim-Berners-Lee in Switzerland. WWW is broad term, which is collection of distributed documents referred to as "Pages" located on computer (or server) all over the world. The purpose to the web was to use networked hypertext to facilitate communications among its members, who were located in several countries.

Servers store hypertext markup language (HTML) files and respond to requests. The WWW is probably the most popular place to be on the internet. The WWW is an easy to use method for

storing and retrieving the information that resides on systems in the internet. Through the use of a browser, PC users can find and view server-based documents. Browsers like Netscape ensure easy access to server based documents and display multimedia data. When you use a web browser, the web appears as a collection of text, pictures, sounds and digital movies.

Reason for the amazing success of the web include:

- Easy of navigation and use.
- Ease of publishing content,
- New distribution models,
- Enabling a network-centric computing paradigm.
- Enabling new intra-business applications.

How the Information Stored in WWW

WWW is an internet-based system that enables an individual or a company to publish itself to the entire world, except to countries or location that prohibit the free interchange of information. The major service on the internet, the world wide web is the world's largest online shopping mall and the world's largest source of information, news and commentary.

In other words, Interconnected collection of information sources within the internet that allow user to view images, look at film clips, hear sound recording and find valuable and interesting information about a wide variety of subjects. Although no one actually knows the size of WWW, it is reported to be growing at approximately a 50% increase per year. As of early 1998, over 500,000 computers around the world provided information the WWW in an estimated 100 million web pages. It is an effective marketing tool in that it can activity be used to set up a marketing presence. It provides businesses the opportunity to create interactive brochure and virtual store fronts as well as offering consumers information clearing house and efficient customer service.

Evolution of WWW

The usual creation myth of the Web traces its birthplace to CERN in Geneva, and its early adoption the High Energy Physics and NeXTstep developer community ("As a high energy physicists I can only 10 u bat we bad been smarter, and, instead of having people type WWW when they want to surf the met, we had them type HEP perhaps we would have big budgets now."– Burton Richter, SLAC Internet, Director).

Other institutions played a key role in sparking its explosive adoption, though. Notably, NCSA, which had laid the groundwork for a first international Web confer ence of its own days after CERN (the immediate outcome was the creation of an international conference committee which sponsored the first in Geneva and the second in Chicago).

Nevertheless, at the same time other institutions were shifting the center of gravity away from CERN, it was pushing the Web out to make room in its limited budget for new physics experiments. An international bidding war of sorts broke out behind the scenes, and the winner of inventor Berners-Lee's affections was MIT. The Laboratory for Computer Science (LCS) offered to replicate its widely model of academic-industrial IT R&D innovation, the X Consortium. LCS began by adopting its membership future, policies, even some of its original administrative staff.

US Government funding sponsored the US bid, but the European Community was similarly committed to its continental aspirations. At the same time, W3C announced a second agreement with host site INRIA, at three locations in France. A year later, Japanese membership growth and local research funding catalyzed a third site at Keio University in Tokyo (which came on line just last Fall). Future expansion may aim at Southeast Asia, Oceania, and upgrading of a secondary center at Rutherford Appleton Laboratories in the UK. The stated criteria are a sufficient number of local Full Members and a research organization to act as host site; state support should be considered an implicit one.

Web Architecture

To use the Web, in addition to an Internet connection, a user need a special piece of software called a Web browser. The browser act as a graphical interface between the user and the Internet — it sends the necessary command to request data from other Computers and then formats them for the user's screen, Documents that are formatted using hypertext markup language (HTML) contain taglines that inform the browser how to format them.

The web is based on a three-part architecture :

HTML : Hypertext markup language, the format for web pages, provide both formatting and hyperlinking.

HTTP: Hypertext transfer protocol, the protocol for communications between Web servers and browsers.

CGI : The common gateway interface, the interface for invoking programs from web servers.

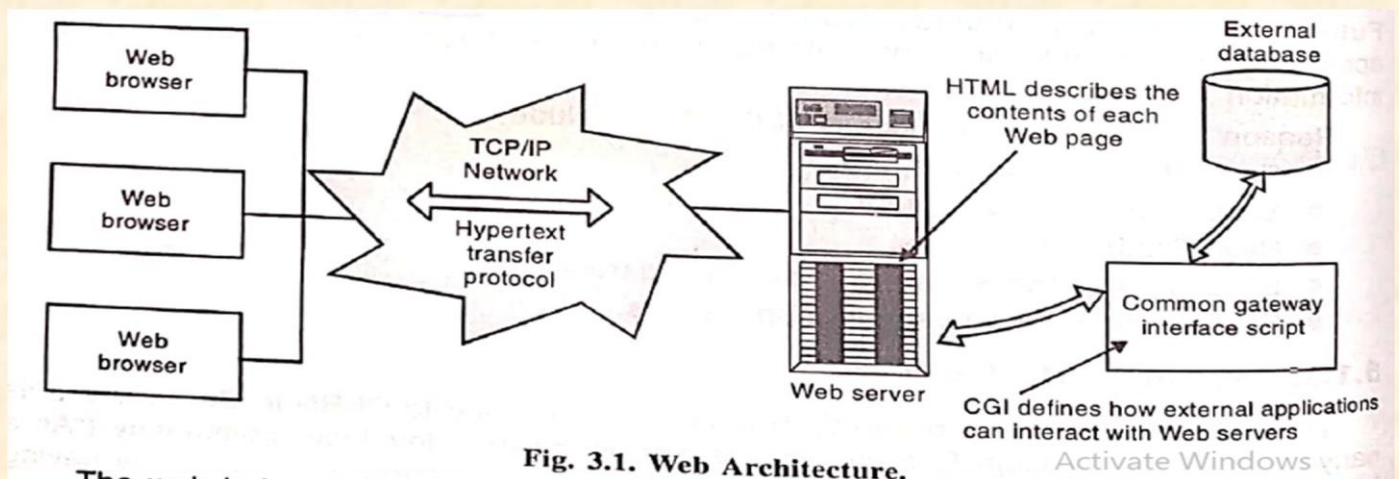


Figure 3.1 shows the components of web architecture, including the web clients, web servers, HTTP protocol language for Web client/server communication and applications. Within the confines of this architecture, hundreds of entrepreneurs have produced tidal wave of software programs and tools.

Web Page

Web pages are what make up the World Wide Web. These documents are written in HTML (hypertext markup language) and are translated by our Web browser.

Web pages can either be static or dynamic. Static pages show the same content each time they are viewed. Dynamic pages have content that can change each time they are accessed. These pages are typically written in scripting languages such as PHP, Perl, ASP, or JSP. The scripts in the pages run functions on the server that return things like the date and time, and database information. All the information is returned as HTML code, so when the page gets to your browser, all the browser has to do is translate the HTML.

Web Servers

Web server is the program resides on server computer to server the request of the client software.

Web server is a combination of software and hardware that store documents and other content. The server content is accessed using Web browsers. Web servers are available for a wide range of platforms including IBM PCs, Apple Macintosh computers, UNIX platform s and a large number of proprietary hosts.

Some the example of the web servers are as follows:

1. Tomcat 2. Weblogic
3. Apache 4. IIS

Application Level Protocol (HTTP, FTP, TCP/IP)

An application layer protocol defines how application processes (clients and servers), running on different end systems, pass messages to each other. In particular, an application layer protocol defines:

- ❖ The types of messages, e.g., request messages and response messages.
- ❖ The syntax of the various message types, i.e., the fields in the message and how the fields are delineated.
- ❖ The semantics of the fields, i.e., the meaning of the information that the field is supposed to contain;

- ❖ Rules for determining when and how a process sends messages and responds to messages.

SMTP (Simple Mail Transfer Protocol):

- ❖ One of the most popular network services is electronic mail (e-mail).
- ❖ The TCP/IP protocol that supports electronic mail on the Internet is called Simple Mail Transfer Protocol (SMTP).
- ❖ SMTP transfers messages from senders' mail servers to the recipients' mail servers using TCP connections.
- ❖ Users based on e-mail addresses.
- ❖ SMTP provides services for mail exchange between users on the same or different computers.
- ❖ Following the client/server model:
- ❖ SMTP has two sides: a client side which executes on a sender's mail server, and server side which executes on recipient's mail server. Both the client and server sides of SMTP run on every mail server. When a mail server sends mail (to other mail servers), it acts as an SMTP client. When a mail server receives mail (from other mail servers) it acts as an SMTP server.

TELNET (Terminal Network /Telecommunication Network):

- ❖ TELNET is client –server application that allows a user to log onto remote machine and lets the user to access any application program on a remote computer.
- ❖ TELNET uses the NVT (Network Virtual Terminal) system to encode characters on the local system. On the server (remote) machine, NVT decodes the characters to a form acceptable to the remote machine.
- ❖ TELNET is a protocol that provides a general, bi –directional, eight-bit byte oriented communications facility.
- ❖ Many application protocols are built upon the TELNET protocol .
- ❖ It is one of the oldest protocol in the TCP/IP.
- ❖ Telnet services are used on PORT 23.

FTP (File Transfer Protocol):

- ❖ FTP is the standard mechanism provided by TCP/IP for copying a file from one host to another.

- ❖ FTP differs from other client-server applications because it establishes 2 connections between hosts.
- ❖ Two connections are: Data Connection and Control Connection.
- ❖ Data Connection uses PORT 20 for the purpose and control connection uses PORT 21 for the purpose.
- ❖ FTP is built on a client-server architecture and uses separate control and data connections between the client and the server.
- ❖ One connection is used for data transfer, the other for control information (commands and responses).
- ❖ It transfer data reliably and efficiently.

Multipurpose Internet Mail Extensions (MIME):

- ❖ It is an extension of SMTP that allows the transfer of multimedia messages.
- ❖ If binary data is included in a message MIME headers are used to inform the receiving mail agent:
- ❖ Content-Transfer-Encoding: Header alerts the receiving user agent that the message body has been ASCII encoded and the type of encoding used.
- ❖ Content-Type: Header informs the receiving mail agent about the type of data included in the message.

POP (Post Office Protocol):

- ❖ POP is also called as POP3 protocol.
- ❖ This is a protocol used by a mail server in conjunction with SMTP to receive and holds mail for hosts.
- ❖ POP3 mail server receives e-mails and filters them into the appropriate user folders. When a user connects to the mail server to retrieve his mail, the messages are downloaded from mail server to the user's hard disk.

HTTP (Hypertext Transfer Protocol):

The HTTP is an application protocol for distributed collaborative, hypermedia information systems. This is a protocol used mainly to access data on the World Wide Web (www). The Hypertext Transfer Protocol (HTTP) the Web's main application -layer protocol although current browsers can access other types of servers. The HTTP protocol transfer data in the form of plain text, hyper text, audio, video and so on. HTTP utilizes TCP connections to send client requests and server replies. It is a synchronous protocol which works by making both persistent and non persistent connections.

Clients (web browsers) send requests to web servers for web elements such as web pages and images. After the request is serviced by a server, the connection between client and server across the Internet is disconnected. A new connection must be made for each request. This means that HTTP is not connection-oriented. On the practical side, it means that after a requested HTML file gets downloaded, your computer is disconnected from the server, then your browser discovers that it needs a .gif image for the page, makes another request to the server, disconnects again, and so on.

✓ Each HTTP interaction works in following ways –

1. The browser established a connection with the server.
2. The browser send a request to server.
3. The server sends response to the client (browser).
4. Both browser and server disconnect and transaction ends.

NNTP:

NNTP stands for Network News Transfer Protocol. NN is a TCP/IP protocol used to post, distribute and retrieve USENET messages. It is a client Server Protocol just like the SMTP or FTP. The global network of collected usenet newsgroups is managed by the servers and NNTP clients is a part of Internet Explorer, Opera, Netscape or any other browsers, It replaces the original USENET protocol i.e., UUCP (Unix –to–Unix copy Proto) NNTP is specially designed to store news articles on one server host and other hosts subscriber can read news articles using stream connections to the news host, while NNTP, hosts exchanging news articles can easily decide which articles are to be transmitted.

Web Browser, URL, Hyper text, Hyperlinks, Hypermedia

Web Browser

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

- Google Chrome
- Mozilla Firefox
- Microsoft Edge,
- Opera browser
- Microsoft Internet Explorer

- Apple Safari

UNIFORM RESOURCE LOCATORS (URLs)

The WWW follows a uniform naming scheme to access resources on the web called URL. The addressing system is used to identify pages and resources on the web. An internet or web address typically is composed of four parts.

1. A protocol name
2. The location of the site
3. The name of the organization that maintains the site.
4. A suffix that identifies the kind of organization it is.

Structure of URL

The following is an outline of the most common form of a URL:

http://www.address.edu: 1234/path/subdir/file.ext

http is : **Service type (since first part is service specifier).**

www.address.edu : 1234 is : Internet address (here internet address is with port number 1234, the internet address with default port number will be www.address.edu).

path/subdir/file.ext is : Resource location (basically here we are giving the path of file.ext file on the server computer).

Service

The above figure gives a general outline of URL structure. This outline is followed by most, but not all, URLs. Specific cases are discussed in subsequent sections. The first part is the service specifier (here HTTP service), which specifies the access method. Specifically this is the part before the colon. Some examples of services are: http:, gopher:, wais: and ftp;.

Address and Port Number

The second part is usually the internet address of the server, indicated by the double forward slash (//). This address can also contain the (optional) port number the service listens at. The full name is specified by a string like //www.address.edu:1234/, where :1234 means Port number 1234. If you want to use the default port number you can leave out both the colon and the number, i.e. //www.address.edu/

Resource Location

The forward slash after the host and port specifications indicates the end of the address and the beginning of the specification for the file/resource to be accessed. This field varies depending on

the service being accessed. A few simple examples follow: more detailed ones are given in the documents listed below.

Some Examples of URL

`http://info.cern.ch/hypertext/WWW/Addressing/URL/Overview.html`

Retrieve the Overview named HTML document from the CERN http server running on the default port.

`gopher://gumby.brain.headache.edu:151/7fonebook.txt`

Access the searchable index 'fonebook.txt' from the named gopher server, running on port number 151.

`news:alt.rec.motorcycle`

Access the newsgroup alt.rec.motorcycle.

List of Schemes

List of schemes of URL used within URLs:

Scheme	Examples
ftp	<code>ftp://ftp.uu.net/usenet/news.answers/alt_tech/pointers.z</code>
Gopher	<code>gopher://gopher.loc.gov/11/ congress</code>
http	Hypertext resource
mail to	Mail a message to the specified address
News	Usenet newsgroup
telnet	<code>telnet://mantra.internet.com:1701 /</code>
WAIS	Access a WAIS database

URLs provide a standard way to specify the exact location and name of just about any Internet resource; In general, most URLs have one of two common formats:

`scheme://hostname/description`

`scheme:description`

Example 1: `http://www.alan.com/alan`

This example describes a particular web page on a particular computer. The URL begins with a name, indicating a specific type of resource. (The name comes from the protocol used to send and receive such information: hypertext Transfer Protocol.)

Example 2: news.rec.human

This example describes a more general resource. The scheme is news, which indicates a Usenet discussion group. URL are used by web browsers (clients) to indicate the name of the web server from which to request information and what information to retrieve that server.

HYPERTEXT

Hypertext most often refers to text on a computer that will lead the user to other related information on demand. Hypertext represents a relatively recent innovation to user interfacing which overcomes some of the limitations of written text. Rather than remaining static like traditional text, hypertext makes possible a dynamic organization of information through links and connections (called hyperlinks). Hypertext refers to the idea of linking different documents together using hyperlinks. It is a text link to other text. Whenever we click on this text, then next page or some other text displays. This is called hypertext. By default the colour of Hypertext is blue and this text is underlined

Characteristics of Good Hypertext

The flexibility of hypertext gives free range to the author's creativity, but good hypertext appears to have some common characteristics:

- **Lots of documents:** Much of the hypertext's power comes from its ability to make large quantities of information accessible. If all the text in your system can be printed on ten pages, it would be just as simple to read through it from beginning to end and forget all this hypertext silliness. On the other hand, if there are ten million pages of text in your system, then someone could follow a link on atomic energy and ultimately hope to find a description of the U-238 decay process.
- **Lots of links:** If each document has just one link, then it is little more than normal sequential text. A hypertext document should present the reader with several links, offering a choice about where to go next. Ideally, a document should present as many relevant links as the reader can easily comprehend and select among.
- **Range of detail:** The great advantage of hypertext is that it permits readers to explore to a breadth and depth that is simply not feasible in print. To make this accessible, available this hypertext documents should range from the broadest possible overview of a subject, or to its gritty details.
- **Correct links:** This may seem trivial, but it's amazing how many Web links point nowhere. In general, be careful linking to any hypertext document not under your direct control.

HYPERLINKS

A hyperlink (often referred to as simple a link) is a reference or navigation element in a document to another section of the same document, another document, or a specified section of another document that automatically brings the referred information to the user. When the navigational element is selected by the user as such it is similar to a citation in literature, but with the distinction of automatic instant access combined with a data network and suitable access protocol, a computer can be inserted to fetch the resource reference.

Hyperlinks are part of the foundation of the world wide web created by Tim Berners—Lee, but are not limited to HTML or the web Hyperlinks may be used in almost any electronic media.

HREF is an acronym for Hypertext REference, as used in HTML in a graphical user interface; the usage of mouse cursor may also change into a hand motif to indicate a link. In most graphical web browsers, links are displayed in underlined blue text when not cached, but underline purple text when cached.

A hyperlink is made up of four main parts:

1. A monitor that identifies the targets location.
2. A string for the location within the target.
3. A friendly or displayable, name for the target.
4. A string that can contain additional parameters.

Hyperlink is a connection that enables an user to jump from one element in an HTML document to another document or website. A hyperlink is associated with normally invisible tags, coded in markup language such as HTML that enables a web browser to find and display the linked document.

HYPER MEDIA

The combination of text, video, graphic images, sound, hyperlink and other elements in the form of typical web documents. Essentially hypermedia is the modern extension of hypertext, the hyperlinked text-based documents of the original internet.

Search Engine, Proxy sever, CGI, URI, Dreamweaver

Search Engine

A program that searches documents for specified keywords and returns a list of documents, where the keywords were found.

A search engine is needed for the same reason you need a card catalogue in a library. There is lots of great and useful information in a library, but it is physically impossible to examine all the books personally. Not even the web-sufer could hyperlink to all the documents in the world wide

web. There are millions of pages and billions of words on the web. A every minute of the day, more is being added. The search engines help you search through all those billions of the information you need. An internet search engine is a tool that web surfers use to search the internet.

Some of the examples of search engines are as follows:

1. Google.com, 2. bing.com, 3. yahoo.com

Working of Search Engines

Typically, a search engine works by sending out a spider to fetch as many documents as possible. Another Program, called an indexer, then reads these documents and creates an index based on the words contained in each document. Each search engine uses a proprietary algorithm to create its indices such that, ideally, only meaningful results are returned for each query.

These are tools that keep track of many websites around the world and let you search for particular items whenever you want. The result of a search is a custom list of links, pointing to whatever items the search engine found that met your criteria.

Proxy Servers

A proxy server is an application that manages control traffic between a protected instead of router for traffic controls to prevent traffic from passing directly between networks. Many proxie s, contains extra login or support for user authentication.

Proxy servers mediates IP traffic between the protected internal network and the internet. Proxy servers runs on a machine that is part of the firewall. To use a proxy server, a user, first logs on to proxy server. A second connection, to the internet, is then made by the proxy server, Since proxy server makes the internet connection, the internet only sees the application gateway IP address. The firewall should be configured such that only, the application gateway is allowed to access the internet and only for services for which there is a proxy. The proxy server may require user authorization and may also write to a log file that is using the service. The proxy mechanism is used mainly for telnet, ftp and web browsing. There are several advantages in using proxy servers as part of your firewall. As only the proxy server connects to the internet, only the corresponding IP address is exposed. No-one needs to know about your company's addresses or domain names. Most Proxy Servers require a user login and may limit who is authorization violation are other advantages of proxy servers.

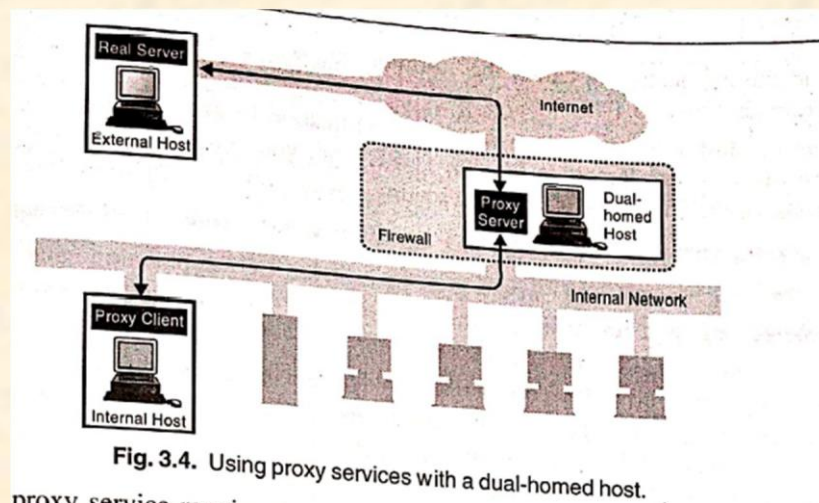
Proxy Services

Some of the following proxy services:

- (i) Proxy services are specialized application or server programs that run on a firewall host either a dual-homed host with an interface on the internal network and one on the external network, or some other bastion host that has access to the Internet and is accessible from the internal machines. These programs take users' requests for

Internet services (such FTP and Telnet) and forward them, as appropriate according to the site's security policy, to the actual services. The proxies provide replacement connections and act as gateways to be services. **For this reason, proxies are sometimes known as application-level gateway.**

- (ii) Proxy services sit, more or less transparently, between a user on the inside on the internal network) and a service on the outside (on the Internet). Instead of talking to each other directly, each talks to a proxy. Proxies handle all the communication between users and Internet services behind the scenes.
- (iii) Transparency is the major benefit of proxy services. It's essentially smoke and mirrors. To the user, a proxy server presents the illusion that the user is dealing directly with real server. To the real server, the oxy server presents the illusion that the real servers is dealing directly with a user on the proxy host (as opposed to the user's real host).
- (iv) Proxy services are effective only when they are used in conjunction with a mechanism that restricts direct communications between the internal and external hosts. Dual-homed hosts and packet filtering are two such mechanisms. If internal hosts are able to communicate directly with external hosts, there's no need for users to use proxy services, and so (in general) they won't. Such a bypass probably isn't in accordance with your security policy.



In Fig. 3.4, a proxy service requires two components: a proxy server and a proxy client. In this situation, the proxy server runs on the dual-homed host. A proxy client is a special version of a normal client program (i.e., a Telnet or FTP client) that talks to the proxy server rather than to the "real" server out on the Internet, in addition, if users are taught special procedures to follow, normal client programs can often be used as proxy clients. The proxy server evaluates requests from the proxy client, and decides which to approve and which to deny. If a request is approved, the proxy server contacts the real server on behalf of the client (thus the term "proxy"), and proceeds to relay requests from the proxy client to the real server, and responses from the real server to the proxy client.

In some proxy systems, instead of installing custom client proxy software, you'll use standard software, but set up custom user procedures for using it. A proxy service is a software solution, not a firewall architecture per se. You can use proxy services in conjunction with any of the firewall architectures described in the section called "Firewall Architectures" below.

The proxy server doesn't always just forward users' requests on to the real Internet services. The proxy server can control what users do, because it can make decisions about the requests it processes. Depending on your site's security policy, requests might be allowed or refused. For example, the FTP proxy might refuse to let users export files, or it might allow users to import files only from certain sites. More sophisticated proxy services might allow different capabilities to different hosts, rather than enforcing the same restrictions on all hosts.

COMMON GATEWAY INTERFACE (CGI)

CGI is a standard method used to generate dynamic content on web pages and web applications. CGI, when implemented on a web server, provides an interface between the web server and programs that generate the web content. These programs are known as CGI scripts or Simply (GIs) they are usually written in scripting language.

The common gateway interface (CGI) is a standard way for web server to pass a web user's request to an application program and to receive data back to forward to the user. When the user requests a web page (for example, by clicking on a highlighted word or entering website address), the server sends back the requested page. However, when a user fills out a form on a web page and sends in, it usually needs to be processed by an application program. The web server typically passes the form information to a small application program that processes the data and may send back a confirmation message. This method or convention for passing data back and forth between the server and the application is called the common gateway interface (CGI). It is part of the web's Hypertext Transfer Protocol (HTTP).

If you are creating a website and want a CGI application to get control, you specify the name of application in the Uniform Resource Locator (URL) that you code in HTML file. The URL can be specified as part of the form tags if you are creating a form. For example, you might code:

<FORM METHOD = POST ACTION = [http://www.mybiz.com /cgi -bin/formprog.pl](http://www.mybiz.com/cgi-bin/formprog.pl)

and the server at "mybiz.com" would pass control the CGI application called "formprog.pl" record the entered data and return a confirmation message (The "pl" indicates a program written in PERL but other languages could have been used.)

The common gateway provides a consistent way for data to be passed from the user's request to the application program and back to the user. This means that the person who writes the application program can makes sure it gets used no matter which operating system the server uses (PC macintosh UNIX, OS/390 or others). It's simply a basic way for information to be passed from a web server abwe your request to the application program and back again.

Because the interface is consistent a programmer can be write a CGI application in a number as different languages. The most popular language for CGI applications are C, C++ and PERL

UNIFORM RESOURCE IDENTIFIER (URI)

Uniform Resource Identifier is a string of characters used to identify a name of a web resource. Such identification enables interaction with representations of the web resource over a network (typically the WWW) using specific protocols, schemes specifying a concrete syntax and associated protocols define each URI.

URI is the way you identify any of the content, whether it be a page of text, video or sound clip a still or animated image or a program. The most common form of URI is the web page address, which is a particular form or subset of URI called URL (Uniform Resource Locator). A URI typically describes:

- The mechanism used to access the resource.
- The specific computer that the resource is housed in.
- The specific name of the resource (a file name) on the computer.

For example this URI:

`http://www.w3.org/icons/www/w3c_mein.gif`

identifies a file that can be accessed using the web protocol application. Hypertext transfer protocol ("http://") that is housed on a computer named "www.w3.org" (which can be mapped to a unique internet address). In the computer's directory structure, the file is located at the pathname of "/icons/www/w3c_main.gif". Character string that identify file transfer protocol FTP addresses and e- addresses are also URIS.

Another kind of URI is Uniform Resource Name (URN). A URN is a form of URI U o "Institutional Persistence", which means that its exact location may change, from time to time, be agency will be able to find it.

DREAMWEAVER

Adobe Dreamweaver is a software application that allows you to create and develop websites. Dreamweaver is considered WYSIWYG (what you see is what you get) meaning that when you format your web page, you see the results of formatting instead of the mark-ups that are used for formatting.

Html is not wysiwyg, whereas Microsoft is wysiwyg. However Dreamweaver allows you to hand code html as well. Dreamweaver also support css and java script as well as other languages including asp and PHP.

Dreamweaver makes it easy to upload your entire website to a webserver. You can also preview your entire website to a webserver. You can also preview your website locally. Dreamweaver also lets you create templates for your website that you can use again and again by modifying certain unrestricted areas within the templates. Then if you want to change one

particular part of your website (Logo changes, a main link changes), you only have to modify the template for the change to propagate throughout your website. Dreamweaver is fairly complex software.

Features

Adobe Dreamweaver is a web design and development application that provides a visual wysiwyg editor and a code editor with standard features such as syntax highlighting, code completion and code collapsing as well as more sophisticated features such as real time syntax checking and code introspection for generating code html to assist the user in writing code. The design view facilitates rapid layout design and code generation as it allows user to quickly create and manipulate the layout of the html elements. It provides transfer and synchronization features, the ability to find and replace lines of text or code by search terms or regular expression across the entire site, a templating features that allows single-source update of shared code.

Adobe Dreamweaver is a software application that allows you to create and develop websites. Dreamweaver is considered WYSIWYG (what you see is what you get) meaning that when you format your web page, you see the results of formatting instead of the mark-ups that are used for formatting. Html is not wysiwyg, whereas Microsoft is wysiwyg. However Dreamweaver allows you to hand code html as well. Dreamweaver also support css and javascript as well as other languages including asp and PHP.

Dreamweaver makes it easy to upload your entire website to a webserver. You can also preview your entire website to a webserver. You can also preview your website locally. Dreamweaver also lets you create templates for your website that you can use again and again by modifying certain unrestricted areas within the templates. Then if you want to change one particular part of your website (logo changes, a main link changes), you only have to modify the template for the change to propagate throughout your website. Dreamweaver is fairly complex software.

Dreamweaver can use third-party "Extensions" to extend core functionality of the application which any web developer can write (largely with html and javascript. Dreamweaver is supported by large community of extension developers who make extensions available (both commercial and free) for most web development tasks from simple roller effects to full-featured shopping carts.

Dreamweaver like other html editors edit files locally then uploads them to the remote webserver using FTP, SFTP or WebDAV.

INTERNET AND WEB TECHNOLOGY (IWT)



Chapter 3: Internet Security

- ✓ 3.1 Introduction to security
- ✓ 3.2 Types of security, Authentication & Authorization
- ✓ 3.3 Firewalls, Encryption & Decryption, SSL

Chapter 3 - Internet Security

Introduction to security

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

Types of security, Authentication & Authorization

Types of security

Network Layer Security

TCP/IP can be made secure with the help of cryptographic methods and protocols that have been developed for securing communications on the internet. These Protocols include SSL and TLS for web traffic, PGP for e-mail and IPsec for network layer security.

IPsec Protocol

This protocol is designed to protect communication in a secure manner using TCP/IP. It is a set of security extensions developed by IETF, and it provides security and authentication at the IP layer by using cryptography. To protect the content the data is transformed using encryption techniques. There are two main type of transformation that form the basis of IPsec: the Authentication Header (AH) and Encapsulating Security Payload (ESP). These two protocol provides data integrity, data origin authentication, and anti replay service.

The basic components of the IPsec security architecture are described in terms of following functionalities:

- ❖ Security protocol for AH and ESP.
- ❖ Security association for policy management and traffic processing.
- ❖ Manual and automatic key management for Internet Key Exchange (IKE).

The set of security services provided at the IP layer includes access control, data origin, integrity, protection against replays and confidentiality.

Electronic Mail Security (E -mail)

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

Pretty Good Privacy (PGP)

PGP provides confidentiality by encrypting messages to be transmitted or data files to be stored using an encryption also such as Triple DES. Email messages can be protected by using cryptography in various ways, such as the following:

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

The first two methods, message signing and message body encrypted are often used together, however, encrypting the transmission between mail server is typically used only when two organizations want to protect e-mails regularly sent between each other. For example, the organization could establish Virtual Private Network (VPN) to encrypt the communication between their mail servers over the internet. Unlike methods that only encrypt a message body, a VPN can encrypt entire messages, including email header information such as sender, recipients and subjects. In some cases, organizations may need to protect header information. However, a VPN solution alone cannot provide a message signing mechanism, nor can it provide protection for email messages along the entire route from sender to recipient.

Multipurpose Internet Mail Extension (MIME)

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

Message Authentication Code

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

AUTHENTICATION AND AUTHORIZATION

- ❖ **Authentication is a process for identifying and verifying who is sending a request.**
- ❖ **Authorization is a process to provide access rights to the individuals.**

The process of identifying an individual usually based on a username and password. In security systems, authentication is distinct from authorization, which is the process of giving individuals access to system objects based on their identity. Authentication merely ensures that the individual is who he or she claims to be, but says nothing about the access rights of the individual.

The Figure 4.1 shows a simplified version of an authentication process.

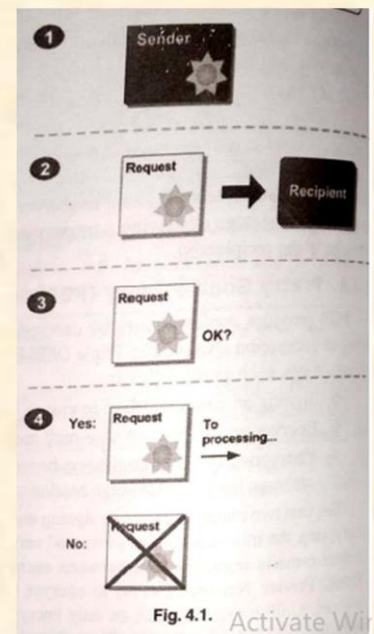
General Process of Authentication

1. The sender obtains the necessary credential.
2. The sender sends a request with the credential to the recipient.
3. The recipient uses the credential to verify the sender truly sent the request.
4. If yes, the recipient processes the request. If no, the recipient rejects the request and responds accordingly.

Normally authentication requires an entity to provide an identifier and then prove the identity by providing something you know (a password), something you have, or something you are.

Multifactor authentication combines multiple factors of information to improve security. Multifactor authentication normally combines something you know with something you are

The process of authorization occurs once the user is authenticated. Authorization is the process which system uses to determine what a user is allowed to do once, he/she is authenticated. Whenever the authenticated user needs to access some files or response, the system verifies that operation against an Access control entries that define which users can or cannot perform certain operations. These operations include read a file, modify a file's contents, update a file properties, perform a backup, shut down a system etc.



Types of Authentication

As stated earlier, encryption is the process of taking all of the data that one computer is sending to another and encoding it into a form that only the other computer will be able to decode. Another process, authentication, is used to verify that the information comes from a trusted source. Basically, if information is “authentic,” we know who created it and we know that it has not been altered in any way since that person created it. These two processes, encryption and authentication, work hand-in-hand to create a secure environment.

There are several ways to authenticate a person or information on a computer:

- 1. Password:** The use of a user name and password provides the most common form of authentication. You enter your name and password when prompted by the computer. It checks the pair against a secure file to confirm. If either the name or the password does not match, then you are not allowed further access.
- 2. Pass Cards:** These cards can range from a simple card with a magnetic strip, similar to a credit card, to sophisticated smart cards that have an embedded computer chip.
- 3. Digital Signatures:** A digital signature is basically a way to ensure that an electronic document (e-mail, spreadsheet, text file) is authentic.

Firewalls, Encryption & Decryption, SSL

FIREWALLS

A Firewall is computer program that monitors the flow of information from the Internet to your computer and prevent unauthorized access to or from a private network.

There are two different types of firewall available for you to use: Hardware firewalls and Software firewalls.

Hardware Firewall

A Hardware Firewall is a physical piece of equipment that sits between the Internet and your computer. An example of a hardware firewall is a broadband router, a common form of Internet connection . The benefit of using a hardware firewall , is that it has the ability to protect multiple computer systems that are connected to it at the same time. This makes it an effective firewall for use in businesses that have multiple computers connected to the Internet, as well as in homes that have more than one computer system.

Advantages

- ❖ Operating system independent
- ❖ Not vulnerable to malicious attacks.

- ❖ Better performance
- ❖ Focuses on only firewall-related duties

Disadvantages

- ❖ Can be single point of failure
- ❖ Higher administrative overhead
- ❖ Higher cost to implement and maintain

Software Firewall

Software Firewalls work in the same way as a hardware firewall, by monitoring and blocking information that comes to your computer via the Internet, however software firewalls must be installed as a program on your computer. These software firewalls can either be installed from a computer disk that you have purchased, or downloaded over the Internet. Software firewalls are the most common type of firewall. **Programs such as Norton 360, Norton Internet Security, ESET Smart Security, and Kaspersky Internet Security all have a firewall bundled within them.**

Advantages

- ❖ Less expensive to implement and maintain
- ❖ Lower administrative overhead

Disadvantages

- ❖ Dependent upon host operating system
- ❖ Requires additional host hardware
- ❖ Vulnerable to malicious attacks
- ❖ Lower performance

Regardless of what type of firewall you choose to use, having one on your computer is a really good idea. A firewall helps to prevent computer hackers from accessing your computer through the Internet and stealing sensitive information or infecting your computer system with some form of computer virus. A hacker can gain access to your system by "back doors or open ports that connect your computer to the Internet. With a firewall in place these ports are blocked from inbound traffic, effectively closing the door in the hackers face.

Components of Firewall

Fundamentally, all firewalls consist of the following kinds of components

1. **Chokes:** Computer or communications devices that restrict the free flow of packets between networks. Chokes are often implemented with router, but they do not have to be. The use of the word "choke" is taken from the field of electronic: **choke is a device that exhibits great resistance to certain types of Single, but not to others.**

2. **Gates:** Specially designated programs, devices, or computers within the firewall's perimeter that receive connection from external networks and handle them appropriately. Other texts on firewalls sometimes that handle all gate functions as bastion hosts.

Ideally, users should not have accounts on a gate computer. This restriction helps improve the computer's reliability and users' security.

3. **Network Client Software:** Client software includes programs such as telnet, ftp and mosaic. One of the simplest ways to give user limited access to the Internet is to allow them to log onto the gate machine and allow them to run network client software directly. This technique has the disadvantage that you must either create user Accounts on the gate computer, or you must have users share a single account.

4. **Proxy Server:** A proxy is a program that poses as another. In the case of a firewall, a proxy is a program that forwards a request through your firewall, from the internal network to the external one.

5. **Network Servers:** Many network servers can also function as proxies. They can do so because they implement simple store-and-forward models, allowing them to forward queries or messages that they cannot handle themselves. Some servers that can operate easily as proxies include SMTP (because e-mail messages are automatically forwarded), NNTP (news is cached locally). NTP (time is maintained locally), and DNS (host addresses are locally cached). The following sections explore a variety of different kinds of firewall configurations in use today.

Firewall Features

The features of firewall are:

1. **Inbound and Outbound Filtering:** Filtering is when a firewall examines information passing through it and determines if that information is allowed to be transmitted and received or should be discarded based on rules or filters that have been created.

2. **Stealth Mode:** When you are connected to the Internet end your computer can be not be detected via probes to your computer, you are in what is called Stealth mode. When you are in Stealth mode the firewall does not send this information back making it seem like you are not even connected. Due to this hackers will not continue targeting your computer they will think you are not online.

3. **Privacy Protection :** Many firewalls now have the ability to block spyware, hijackers, and adware from reaching your computer. This allows you to protect your computer from being infected with software that is known to reveal private information about

what you do on the Internet or other computing habits. These features are usually bundled into the commercial versions of the firewall software packages.

4. **Application Integrity:** Application integrity is when the firewall monitors the files on your computer for modification in the file or how they are launched. When it detects such a change it will notify the user of this and not allow that application to run or transmit data to the Internet. Many times these modifications may have been part of an upgrade, but if it was modified by a malicious program you will now be made aware of it.
5. **Intrusion Detection :** Intruders use various methods to penetrate the security of your computer. Intrusion detection scans incoming data for signatures of known methods and notifies you when such attacks are recognized. This allows you to see what means a hacker is trying to use to hack your computer.
6. **Notifications:** Notifications allow you to see the activity of what is happening on your firewall and for the firewall to notify you in various ways about possible penetration attempts on your computer.

Importance of Firewall

- ❖ A firewall's purpose is to keep the jerks out of your network while still letting us get your job done.
- ❖ Defend resources, validate access
- ❖ Manage and control network traffic
- ❖ Record and report on events
- ❖ It act as an intermediary
- ❖ It provides not only real security it often plays an important role as a security blanket for management.
- ❖ It can act as your corporate "ambassador" to the Internet. Many corporations use their systems as a place to store public information about corporate products and services, files to download, bug-fixes, and so forth.

Firewall Protection

Some firewalls permit only email traffic through them, thereby protecting the network against any attacks than attacks against the email service. Other firewalls provide less strict protections, and block services that are known to be problems.

Generally, firewalls are configured to protect against unauthenticated interactive logins from the outside world. This, more than anything, helps prevent intruders from logging into machines on your network. More elaborate firewalls block traffic from the outside to the inside,

but permit users on the le to communicate freely with the outside. The firewall can protect us against any type of network borne attack if we unplug it.

Because of this, firewall logs are critically important data. They can be used as evidence in a court of law in most countries. We should safeguard, analyze and protect our firewall logs accordingly.

Limitations of Firewall

- ❖ **Firewalls can't protect against attacks that don't go through the firewall.** Many corporations connect to the Internet are very concerned about proprietary data leaking out of the company through that route. For A firewall to work, it must be a part of a consistent overall organizational security architecture. Firewall policies must be realistic and reflect the level of security in the entire network. For example, a site with top secret or classified data doesn't need a firewall at all: they shouldn't be hooking up to the Internet in the first place, or the systems with the real secret data should be volatile from the rest of the corporate network.
- ❖ **A firewall can't really protect us against is traitors or idiots inside our network.** While an industrial spy might export information through our firewall, he's just as likely to export it through a telephone, FAX machine, or Compact Disc. CDs are a far more likely means for information to leak from your organization than a firewall.
- ❖ **Firewalls also cannot protect us against stupidity.** Users who reveal sensitive information over the telephone are good target for social engineering; an attacker may be able to break into our network by completely by. passing our firewall, if he can find a "helpful" employee inside who can be fooled into giving access to a modem pool.
- ❖ **Firewalls cannot protect against bad things being allowed through them.** For instance, many Trojan Horses use the Internet Relay Chat (IRC) protocol to allow an attacker to control compromised internal host from a public IRC server. **If we allow any internal system to any external system, then your firewall will provide no protection from this vector connect.**

Technique of Firewall

Common types of firewalls are as follows:

- i. Application Gateways of Proxy Gateways.
- ii. Packet Filtering
- iii. Circuit Gateways.
- iv. Stateful Packet Inspection
- v. Internet Connection Firewall.

vi. Hybrid Firewall

Packet filters: – This firewall technique is observed the entering and discharging the network of each packet. Packets filters admit or discard it depend on user-defined rules. Packet filtering is quite efficient and transparent to users, but it is hard to build up. Packets filter used to liable to IP spoofing.

Application gateway: – Application gateway used to apply safety method to definite applications, such as Telnet and FTP servers. Application gateways are very effective, but can inf lict presentation poverty.

Circuit-level gateway: – Circuit-level gateway is concerned to security device especially when UDP or TCP connection is going to establish. When connection comes visible then packets can run among the hosts without additional checking.

Proxy server:-Proxy server interrupts all messages incoming and outgoing the network. The proxy server successfully hides the true network addresses.

Stateful Packet Inspection: - A fourth method that can be utilized by firewalls is called "Stateful Packet Inspection". It is called "Stateful" because it examines the contents of the packet to determine what the state of the communication is. It ensures that the stated destination computer has previously acknowledged the Communication from the source computer. In this way all communications are initiated by the "deceiving" computer and are taking place only with sources that are known or trusted from previous communication connections.

Encryption and Decryption

When we think of encryption, we probably think of some sort of s ecret code that prevents others from reading our messages. While this privacy aspect of cryptography is important, it is only one of four aspects that are of particular importance in electronic communication.

- Authentication allows customers to be sure that the merchant they are sending their credit card details to is who they say they are. It can also allow merchant's to verify that the customer is the real owner of the credit card being used.
- Integrity ensures that the messages have not been tampered with by a third party during transmission.
- Non-repudiation prevents customers or merchants denying they ever received or sent a particular message.
- Privacy prevents third parties from reading intercepted Messages.

The translation of data into a secret code is known as **encryption**. It is the most effective way to data security. To read an encrypted file, you must have access to a secret key or password that enables you to decrypt it. Unencrypted data is called plain text; encrypted data is referred to an cipher text. There are two main types of encryption: asymmetric encryption also called public key encryption and symmetric encryption.

Elements of an Encryption System

The main elements of an encryption system are the plaintext, the cryptographic algorithm, the key and the ciphertext.

- i. The plaintext is the raw message or data that is to be encrypted.
- ii. A cryptographic algorithm or cipher is a mathematical set of rules that defines how the plain text is to be combined with a key.
- iii. The key is a string of digits.
- iv. The ciphertext is the encrypted message.

These terms are probably best illustrated through a very simple example. If we take the phrase "Web store" and add 2 characters to each letter the phrase becomes "ygd uvqtg". Here :

"Web store" is the plaintext

"add x characters to each letter" is the cryptographic algorithm

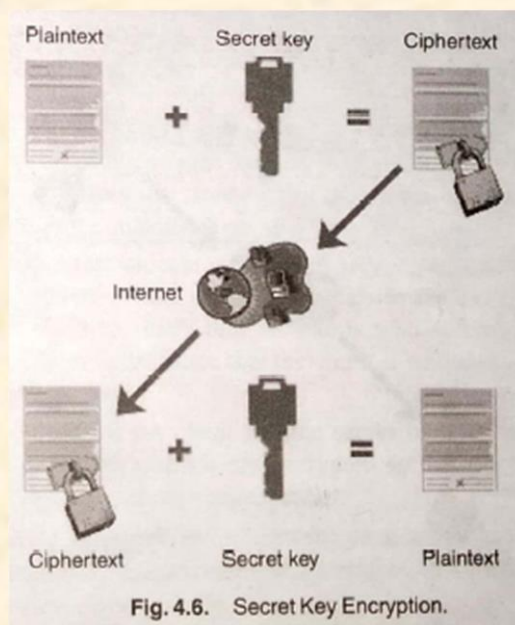
"2" is the key

"ygd uvqtg" is the ciphertext

There are two main types of encryption in common use today: **secret-key** and **public-key**.

Secret-Key Encryption Methods OR Symmetric Encryption

Secret key encryption, also known as single -key or symmetric encryption, involves the use of a single key that is shared by both the sender and the receiver of the message.

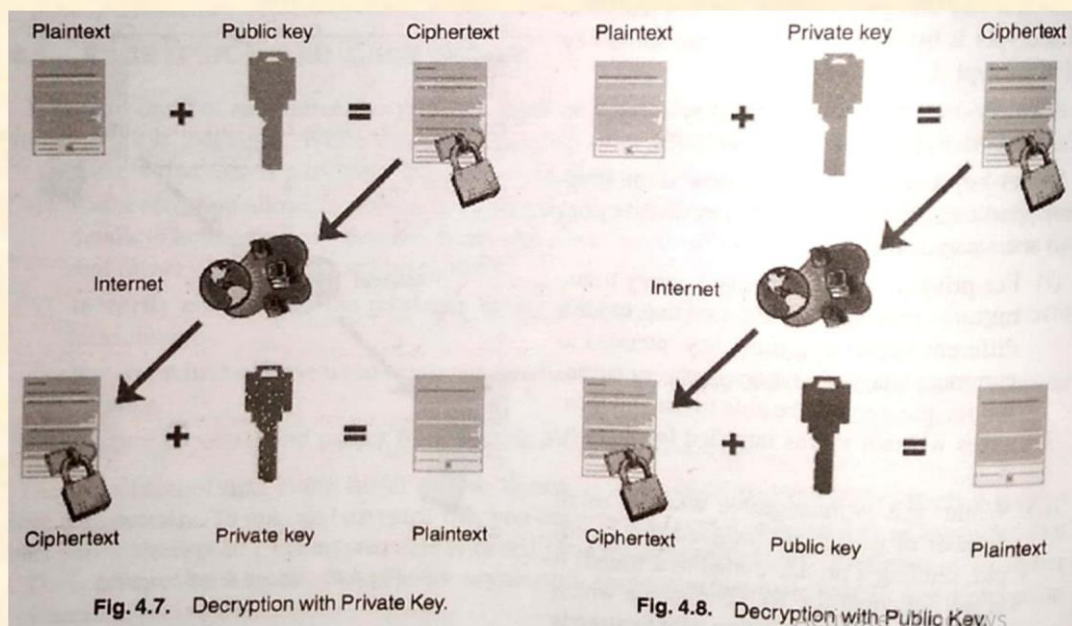


After creating the message, the sender encrypts with their key and passes it to the recipient who then decrypts it by using a copy of the same key used to encrypt it. A widely used method of secret-key encryption is the data encryption standard or DES. Secret-key encryption does have some limitations, particularly with regard to key distribution. Let us see some of them as follows:

- (i) For privacy to be maintained, every transmitter of messages would need to provide a different key to everyone they intended to communicate with otherwise every potential recipient would be able to read all messages whether it was intended for them or not.
- (ii) While this is manageable where a small number of parties are involved (for example, sending a private e-mail to a friend) it is not practical for Web commerce which can involve communicating with thousands of customers.
- (iii) Another limitation with secret-key encryption is its inability to support non-repudiation. As both parties share the same key it is possible for one party to create a message with the shared secret and falsely claim the other party had sent it.
- (iv) Secret-key encryption on its own is not suitable for Web commerce—instead a system known as public-key encryption is used.

Public-Encryption Methods OR Asymmetric Encryption

Public-key encryption, or asymmetric encryption involves the use of two keys, one that can be used to encrypt messages (the public key) and one that can be used to either encrypt them or decrypt them (the private key). These key pairs can be used in two different ways, to provide privacy or authentication. Privacy is ensured by encoding a message with the public key as it can only be decoded by the holder of the private key.



As the public-key can be made widely available from a server or third party, public-key cryptography does not suffer from the same key distribution and management problems as the secret key system.

One disadvantage of the public-key system is that it is relatively slow, so when it is being used only for authentication it is not desirable to encrypt the whole message particularly if it is a long one. To get round this a digital signature is used.

Symmetric Encryption vs. Asymmetric Encryption

Symmetric Key Encryption	Asymmetric Key Encryption
It only requires a single key for both encryption and decryption.	It requires two key one to encrypt and the other one to decrypt.
The size of cipher text is same or smaller than the original plain text.	The size of cipher text is same or larger than the original plain text.
The encryption process is very fast.	The encryption process is slow.
It is used when a large amount of data is required to transfer.	It is used to transfer small amount of data.
It only provides confidentiality.	It provides confidentiality, authenticity and non-repudiation.
Examples: 3DES, AES, DES and RC4	Examples: Diffie-Hellman, ECC, El Gamal, DSA and RSA
In symmetric key encryption, resource utilization is low as compared to asymmetric key encryption.	In asymmetric key encryption, resource utilization is high.

SECURE SOCKET LAYER (SSL)

DEFINITION: Secure Socket Layer (SSL) is a encryption-based Internet security protocol designed and implemented by Netscape Communications. Netscape claims it is designed to work, as the name implies. At the socket layer, to protect any higher level protocol build on sockets, such as Telnet, FTP and HTTP.

Secure Socket Layer (SSL) technology allows web browsers and web servers to communicate over a secure connection. In this secure connection, the data that is being sent is encrypted before being sent and then is decrypted upon receipt and before processing. Both the browser and the server encrypt all traffic before sending any data. SSL addresses the following important security considerations.

(i) Authentication: During your initial attempt to communicate with a web server over a secure connection, that server will present our web browser with a set of credentials in the form of a server certificate. The purpose of the certificate is to verify that the site is who and what it claims to be. In some cases, the server may request a certificate that the client is who and what it claims to be (which is known as client authentication).

(ii) Confidentiality: When data is being passed between the client and the server on a network, third parties can view and intercept this data. SSL responses are encrypted so that the data cannot be deciphered by the third party and the data remains confidential.

(iii) Integrity: When data is being passed between the client and the server on a network, third parties can view and intercept this data. SSL helps guarantee that the data will not be modified in transit by that third party.

Use of SSL

To use SSL, an application server must have an associated certificate for each external interface, or IP address that accepts secure connections. It may be useful to think of a certificate as a "digital driver's license for an internet address. It states with which company the site is associated, along with some basic contact information about the site owner or administrator

The digital certificate is cryptographically signed by its owner and is difficult for anyone else to forge.

SSL uses public key cryptography, which is based on key pairs. Key pairs contain one public key and one private key. If data is encrypted with one key, it can be decrypted only with the other key of the pair. This property is fundamental to establishing trust and privacy in transactions. For example, using SSL, the server computes a value and encrypts the value using its private key. The encrypted value is called a digital signature. The client decrypts the encrypted value using the server's public key and compares the value to its own computed value. If the two values match, the client can trust that the signature is authentic, because only the private key could have been used to produce such a signature

Digital certificates are used with the HTTP protocol to authenticate web clients. The HTTPS service of most web servers will not run unless a digital certificate has been installed. Use the procedure outlined later to set up a digital certificate that can be used by your web server to enable SSL.

SSL is used:

- ❖ To secure online credit card transactions.
- ❖ To secure system logins and any sensitive information exchanged online.
- ❖ To secure webmail and applications like Outlook Web Access, Exchange and Office Communications Server.
- ❖ To secure workflow and virtualization applications like Citrix Delivery Platforms or cloud-based computing platforms.
- ❖ To secure the connection between an email client such as Microsoft Outlook and an email server such as Microsoft Exchange.

- ❖ To secure the transfer of files over https and FTP(s) services, such as website owners updating new pages to their websites or transferring large files.
- ❖ To secure hosting control panel logins and activity like Parallels, cPanel, and others.
- ❖ To secure intranet based traffic such as internal networks, file sharing, extranets, and database connections.
- ❖ To secure network logins and other network traffic with SSL VPNs such as VPN Access Servers or applications like the Citrix Access Gateway.

INTERNET AND WEB TECHNOLOGY (IWT)



Chapter 4: Internet Application

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

Chapter 4 - Internet Application

E-Mail, Email protocols

Introduction to e-mail

Electronic mail is a system in which .computer user can exchange message with other computer users using a communication network. To use email:

- a) You should have access to a computer that is linked to the outside world i.e., Internet.
- b) A mailbox address to which messages are sent. Messages sent through e-mail can arrive within a matter of seconds

A powerful aspect of e-mail is the option to send electronic files to a person's e-mail address. Non ASCII files, known as binary files may be attached to a e-mail messages. These files are referred to a MIME attachments. Many e-mail programs including Eudora, Netscape Messenger offer the ability to read files written in HTML.

Advantages of E-mail

Electronic mail, or e-mail, is the most frequently used service on the Internet for many reasons:

1. Send a message anytime, anywhere and the recipient can read it at his or her convenience.
2. Send the same message to multiple recipients.
3. Easy to use e-mail frees us from the tedious task of managing data of daily use. It helps us to manage our contacts, send mails quickly maintain our mail history store the required information, etc.
4. Forward information to people without retyping it.
5. E-mail is fast usually taking no more than a few minutes to be received.
6. Attach digital files to your messages, including electronic documents, video clips, music and photos.
7. Send messages around the world easily as to someone down the block.
8. No other person can read your letters as you only know the password of your mail box.
9. Since the message lies in your computer, you can always access and see it when replying to it.
10. Your mail can cut right across the globe in just few seconds, leaving the ordinary mail much behind.

11. Unlike the telephone and fax. It does not wait for the user line to be free to reach him. It will be in his box and he can open it at his convenience.
12. E-mail is cost effective At the cost of a single local call. You can send a long letter to a user irrespective of distance he is at.
13. Most of the e-mail programs give you the option of sending the message to more than party, saving your time and money.

Disadvantages of E-mail

Like other good thing e-mail also has some disadvantages as follows:

1. E-mail reaches the recipient most of the time, but delivery is not guaranteed. Sometimes spam filtering software may block a message or direct it to the recipient's Junk Mail folder
2. There is no guarantee of e-mail privacy. A message can be intercepted and read as it makes its way over the Internet to its final destination.
3. Unencrypted mails are unsafe. So the safest way to send sensitive or confidential information via e-mail is by using encryption software.
4. Unsolicited e-mails that are sent huge numbers, may slow mail servers as well as e-mail programs.
5. With hundreds of these junk mails in our inbox, the occasional good e-mail can be hard to spot.
6. The e recipient needs to scan the mails, as viruses are transmitted through them and have the potential to harm computer systems.
7. Checking and deleting these unwanted mails can unnecessarily consume a lot of time, and it has become necessary to block or filter the unwanted e-mails by means of spam filters. Spamming includes, sending hoax e-mails.
8. It is true that e-mail habit forming once you get on to it, you can never pick up a pen for writing a letter to somebody.
9. It is very difficult to find out the e-mail address of somebody unless he tells you. There is no way to find out the e-mail of someone whose is known to you.
10. E-mail restricts you to type out the matter on the computer key instead of writing on the paper in long hand. So for this you must know typing if not at a speed, you should at least know where what word is. This is not always possible.

User Id and Password

User id is the unique e-mail address through which particular user can use the e-mail system on which he/she created e-mail id. For example, arav@kiet.edu is user id through which I can use the e-mail system of KIET.

A password is a secret word or string of characters that is used for authentication, to prove identity or gain access to a e-mail system. The password must be kept secret from those not allowed access.

If a user is the registered user on e-mail system then he/she can use his e-mail by using user id and password.

E-mail Addresses

An e-mail address is required to send and receive e-mail. They usually consist of three parts: a user name, an "at" symbol (@), and a domain name.

arav@kiet.edu

1. User name: arav

Located to the left of the symbol, a user name identifies your account on the e-mail server that handles the e-mail.

2. @

Means "at" and separates your account name and the name of the mail server name. For example, adesh is located at" kiet.edu.

3. Domain Name: kiet.edu

The domain name usually consists of two pieces of identifying information. The first piece is the name of the e-mail server and is located to the right of the @ symbol. For example, kiet is the name of the e-mail server. The second piece, usually a three-letter extension (edu), indicates the top-level domain. They are separated by periods, called "dots."

E-mail Message Components

The software program we use to send, receive and manage electronic messages is called an e-mail client. To send a message, we enter information similar to the heading of a typical interoffice memo. The message components are as follows:

1. **To** contains the e-mail address of the recipient. This is a mandatory entry.
2. **CC**, short for Carbon Copy contains the e-mail addresses for people other than the primary recipients. This is optional.

3. **BCC**, short for Blind Carbon Copy, contains the e-mail addresses of other recipients who receive copies, but their names and addresses are hidden from the other recipients. This is optional.
4. **Subject** contains the main topic of the message. Recipients see this in their summary of incoming messages.
5. **Attachment** contains the names of files that you may be sending, for example, a word-processing the document or a photo.
6. **Body** contains the message itself, which can be of any length.

E-MAIL PROTOCOLS

Protocol is about a standard method used at each end of a communication channel, in order to properly transmit information.

In order to deal with our e-mail we must use a mail client to access a mail server. The mail client and mail server can exchange information with each other using a variety of protocols like:

1. **IMAP (Internet Message Access Protocol)** is a standard protocol for accessing email from our local server. IMAP is a client/server protocol in which e-mail is received and held for us by our Internet server.
2. **The POP (Post Office Protocol 3)** protocol provides a simple, standardized way for users to access mailboxes and download messages to their computers.
3. **The SMTP (Simple Mail Transfer Protocol)** protocol is used by the Mail Transfer Agent (MTA) to deliver our e-mail to the recipient's mail server. The SMTP protocol can only be used to send e-mails, not to receive them.
4. **The HTTP** protocol is not a protocol dedicated for e-mail communications, but it can be used for accessing our mailbox. Also called web based e-mail, this protocol can be used to compose or retrieve e-mails from our account. Hotmail is a good example of using HTTP as an e-mail protocol.

IMAP (Internet Message Access Protocol)

IMAP stands for Internet Message Access Protocol. It is a method of accessing electronic mail or bulletin board messages that are kept on a (possibly shared) mail server.

In other words, it permits a "client" e-mail program to access remote message stores as if they were local. For example, e-mail stored on an IMAP server can be manipulated from a desktop computer at home, a workstation at the office, and a notebook computer while travelling, without the need to transfer messages or files back and forth between these computers.

IMAP's ability to access messages (both new and saved from more than one computer has become extremely important as reliance on electronic messaging and use of multiple computers increase, But this functionality

cannot be taken for granted: the widely used Post Office Protocol (POP) works best when one has only a single computer, since it was designed to support "offline" message access, wherein messages are downloaded and then deleted from the mail server.

Key goals for IMAP include :

- Be fully compatible with internet messaging standards, e.g., MIME.
- Allow message access and management from more than one computer.
- Allow access without reliance on less efficient file access protocols.
- Provide support for "online" offline", and disconnected access modes.
- Support for concurrent access to shared mailboxes.
- Client software needs no knowledge about the server's file store format.

POP (Post Office Protocol)

Protocol provides a simple, standardized way for users to access mailboxes and download messages to their computer. When using the POP protocol all your e-mail messages will be downloaded the mail server to your local computer

The advantage is that once your messages are downloaded you can cut the internet connection in read your e-mail at your leisure without incurring further communication costs. On the other hand you might have transferred a lot of message (including spam or viruses) in which you are not at all interested at this point.

Like it seems everything on the internet, mail retrieval is a client-server application. The Post Office Protocol defines how your e-mail client should talk to the POP server. The POP is a very simple protocol. This makes it easy to implement, has earned the Post Office Protocol widespread adoption and make it very robust, but it also means the Post Office Protocol provides only basic functionality.

Things that can be done via the POP include:

- Retrieve mail from an ISP and delete it from the server.
- Retrieve mail from an ISP but not delete it from the server.
- Ask whether new mail has arrived but not retrieve it.
- Peek at a few lines of a message to see whether it is worth retrieving

SMTP

SMTP the acronym for Simple Mail Transfer Protocol, which is the standard technology used by people around the globe when sending email.

Working of SMTP

SMTP is a simple ASCII protocol. After establishing the TCP connection to port 25, the sending machine operating as the client, waits for the receiving machine operating as the server, to talk first. The server starts by sending a line of text giving its identity and telling whether or not it is prepared to Receive mail. If it is not the client release the connection and tries again later. If the server is willing to accept e-mail, the client announces whom the email is coming from and whom it is pointing to. If a recipient exists at the destination, the server gives the client the go-ahead message. Then the client sends the message and the server acknowledges it

Telnet

TELNET

It is the internet service used for remote login to internet computer. It stands for telecommunication network of the oldest protocol in the TCP/IP . Before accessing to someone else computer. It should be kept in mind that they have given you permission. The computer which initiates the connection in the local computer and the computer which comes the connection is the host computer.

Use of Telnet

- If we are a gamer, there are a number of text based games still available through telnet. See Yahoo's page on MUDs (Multi -User Dungeons).
- Outside of games, unless we already know of a particular use, such as connecting to a particular government database, we probably have no use for it.
- For years, many library catalogs were only reachable through telnet. Many early community resources, called free-nets, were reachable through telnet. You would have trouble finding any of these now.

Connect to a Telnet Address

Your Internet provider supports telnet not all do), we can connect to a telnet address in one of two ways:

- Recent versions of Windows include a command line telnet. Begin at Start/Run and type command. On the next screen, type your telnet command, From a SLIP or PPP or ethernet connected account: type a telnet URL into the location window of our browser: To do this, our browser must be configured with a supporting version of telnet software. If we are using Windows95, make sure we get a version that supports Win95's 32-bit stack.

- From a unix shell account or other account that lets you type telnet commands, we can issue a command like this: telnet some address port-number.

F T P

FILE TRANSFER PROTOCOL

It allows files to be transferred between different kinds of computers, without regard to operating system used by the computers or even how they are connected. FTP stands for file transfer protocol. This is both a program and the method used to transfer files between computers the internet. Anonymous FTP is an option that allows user to transfer files from thousands of host computers on the District to their personal computer account. By using FTP the internet becomes as if a huge disk drive attached to your computer. FTP sites contain books, articles, software, games, images, sounds, multimedia, course work, data sets and more.

FTP transfers, can be performed on the WWW without the need for special software.

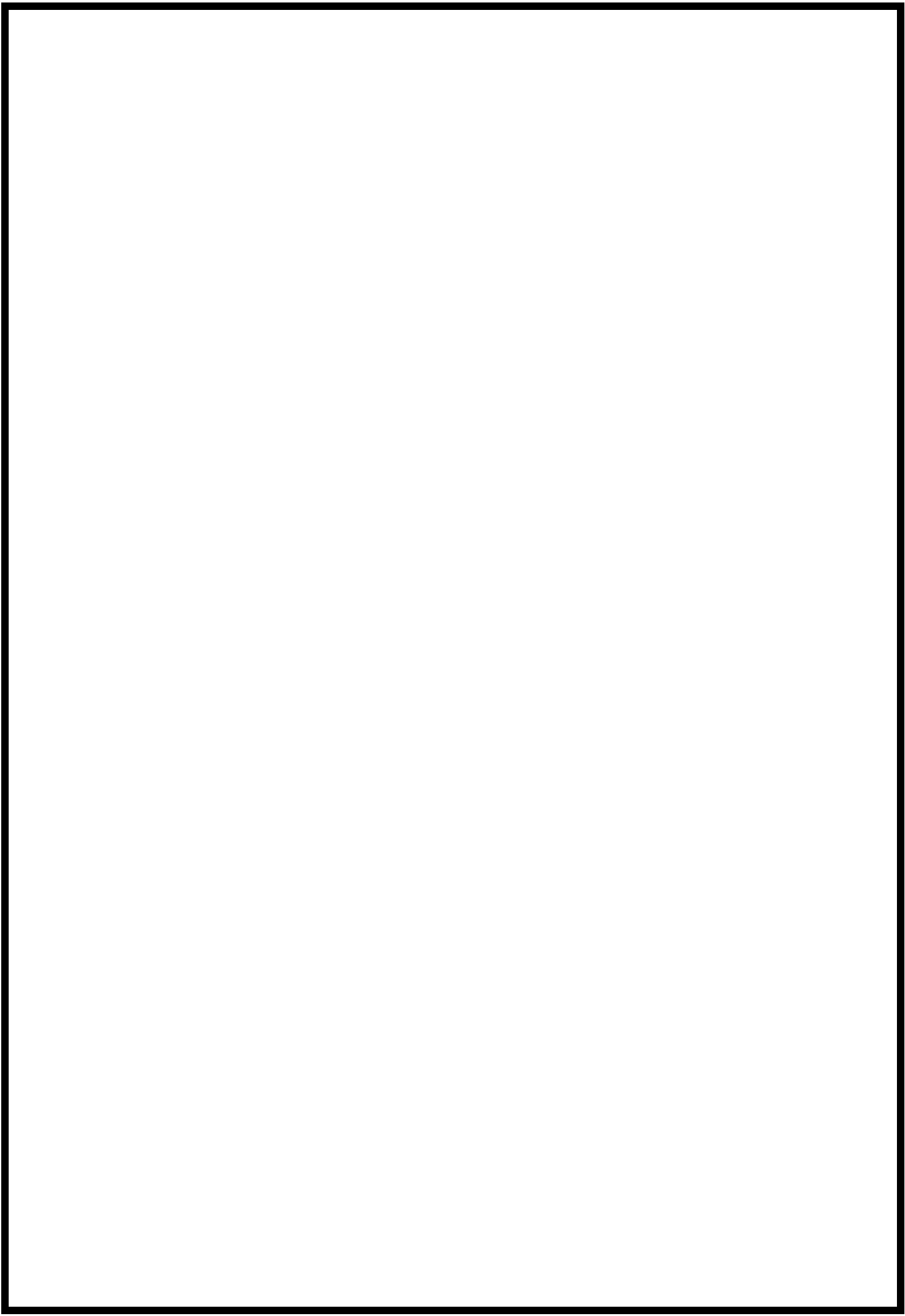
FTP Used for Download (Transfer from another Machine to Your Machine)

1. **Connect to the Server (and Login):** Servers with restricted access requires a username and password. Other servers (such as ftp.netscape.com) support "anonymous login" where the standard is to use the username anonymous and our full email address as your password.
2. **Find the File(s)/Folders We Want to Download:** Navigating the folder hierarchy is just like it is on Macs, PCs, etc. Double-click a folder to open it.
3. **Choose the Appropriate Download Mode:** There are two transfer modes in FTP. The first is called either "ascii" or "text", and the other is called "binary" or "image". The reason for the two modes is that different machines represent End-Of-Line characters in different ways. "text" mode automatically corrects for these differences so that the text we download appears correct on our machine, so for text files we should use text mode (otherwise, we may find, for that all our text appears on one line after it is downloaded). But if we use text mode on a text file, it will corrupt the file. It will try to interpret non-text data as text and therefore cha the file. And for something non-text like a graphic or a microsoft word document (why contains formatting), you will find very unpleasant results if you use text mode! In short, isn't text, use binary mode (which leaves the file untouched).
4. Begin the transfer and get the file (make sure we know where the file is being put on our local machine).

FTP Used for Upload (Transfer from Your Machine to Another Machine)

1. **Connect to the Server (and Login):** Servers with restricted access require a username and a password. Other servers (such as ftp.netscape.com) support "anonymous login" where the standard is to use the username "anonymous" and your full email address as our password.

2. If we are using Fetch, choose to "Put" files/folders, and we will be prompted to specify the files



3. Find the files/folder(s) we want to upload on your computer. Navigating the folder hierarchy is just like it is on Macs, PCs, etc., double-click a folder to open it.

4. Choose the Appropriate Upload Mode! See the notes on download for details. Note the Fetch chooses different names for the file types when uploading. "Raw Data" is the sun as binary/image. "Text" is text.

5. Begin the transfer

Newsgroup

NEWS GROUPS

A Usenet newsgroup is repository usually within the usenet system, for messages posted from many users in different locations. The term may be confusing to some, because it is in fact a discussion group. Newsgroups are technically distinct from, but functionally similar to discussion forums on the world wide web. Newsreader software is used to read newsgroups. In recent years, this form of open discussion the internet has lost considerable ground to browser-accessible forums and social networks such as Facebook or Twitter, and are considered obsolete.

How Newsgroups Work

Newsgroup servers are hosted by various organizations and institutions. Most internet service providers host their own news servers, or send access to one, for their subscribers. There are also a number of companies who sell Access to premium news server.

Every host of a news server maintains agreements with other news servers to regularly synchronize. In this way news servers form a network. When a user posts to one news server, the message is stored locally. That server then shares the message with the servers that are connected to it if both carry this newsgroup, and from those server to servers that they are connected to, and so on. For newsgroups that are not widely carried, sometimes a carrier group is used for cross posting to aid distribution. This is typically only useful for groups that have been removed or newer groups.

Types of Newsgroups

Newsgroups generally come in either of two types, binary or text. There is no technical difference between the two but the naming differentiation allows user and servers with limited network bandwidth usage.

Newsgroups vary in popularity with some newsgroups only getting a few posts a month while others get several hundred (in few cases a couple of thousand) messages a day.

Binary Newsgroups

While newsgroups were not created with the intention of distributing binary files, they have proven to be quite effective for this. Because of the way they work a file uploaded once will

be spread and can then be downloaded by an unlimited number of users. More useful is that every user is drawing on the bandwidth of his or her own news server .

There were originally a number of obstacles to the transmission of binary files over usenet. First, usenet was designed, with the transmission of text in mind. Consequently, for a long period of time, it was impossible to send binary data as it was. So a workaround, U vencode was developed which mapped binary data from the files to be transmitted (e.g.. sound or video files) to text characters which would survive transmission over usenet. At the receiver's end, the data needed to be decoded by the user's news client. Additionally, there was a limit the size of individual posts such that large files could not be sent as single posts. To get around this newsspreaders were developed which were able to split long files into several posts. Intelligent newsspreaders at the other end could then automatically group such split files into single files, allowing the user to easily retrieve the file. These advances meant that usenet is used to send and receive many terabytes of files per day.

There are two main issues that pose problems for transmitting binary files over newsgroups. The first completion rates and other is retention rates. Completion rates are significant when users wish to download large files that are split into pieces, if any one piece is missing, it is impossible to successfully download and reassemble the desired file. To work around the problem, a redundancy scheme known as PAR is commonly used.

Moderated Newsgroups

A moderated newsgroup has one or more individuals who must approve article before they are posted at large. A separate address is used for the submission of posts and the moderators then propagate posts which are approved for the readership.

Chartroom

CHAT ROOM

Chat rooms are created to hold discussions about any topic. Some of the most common topics include: sports, politics, entertainment and education. However, there are Chat Rooms in existence for practically any topic any person would want to talk about.

Chat rooms began as a text based environment for people to communicate with each other and has since expanded to include voice and video as well as three dimensional avatars which represent you in a computer based world. Although basic text based Chat Rooms are largely becoming extinct, there are still quite a few still in existence. They are quickly becoming replaced by more exciting text based Chat Rooms which feature picture avatars or various games you can play while you chat.

A chat room is a virtual place on the Internet where people from all walks of life from around the world can get together in one place and textually chat with one another. The only two things a person needs to connect to a chat room are a computer and an internet connection of any speed. There are, however, Chat Rooms that require client programs, as with IRC Internet Relay Chat).

One of the newest types of Chat Rooms allow user to enroll in an entire virtual world. Owing their history to expensive massive multiplayer online games, these new virtual Chat Rooms (graphical multi-user environments) feature a 3D or 2D environment where people are free to make their own characters or avatars which represent them in the virtual world. Many of these virtual Chat Rooms, such as Second Life, are free of charge. Many members choose to spend real world money to purchase costumes and other virtual items, With graphical multi-user environments, you are free to develop your character however you choose and you can even create your own home away from home in the virtual world provided you are willing to buy the necessary pieces.

Chat Room Etiquette

Because Chat Rooms are place where so many different types of people come together to com with one another, certain rules need to be in place to keep the peace. Nearly all of the larger Chat Room constantly moderated for any offensive activity, but quite a few smaller Chat Rooms lack such mode and it is therefore up to the users to be sure that no offensive activities take place.

Many Chat Rooms restrict offensive language including obscenities, racial slurs, and see terms that may be offensive to certain users. Any type of advertising, flooding the same world words over and over again, and spamming are also frowned upon. Most users will be banned from the chat room for doing such activities.

If you are visiting an online chat room that is not moderated, report any of the above activities to whoever runs the site you are using. Being active in a chat community, you can greatly enhance the chatting environment for everyone in the chat room, so it is important to be proactive with policing the rules – even if you are not a chat room moderator yourself.

Significance: Chatting online with different people using chat rooms exciting because it allows you to meet different people who share common interests. There are those who use chur rooms to find love while others use them to share the latest game chat Whatever your reason for using chat rooms, it is important to understand the advantages and disadvantages that come along with the so that you do not fall victim to negative situations.

Types: Single and romance chat rooms are some of the most common ones on the Internet. They are mainly established by dating websites and allow people to meet each other in the cyber world. Other types of chat rooms include webcam/video chat rooms. Alt hough this is the riskiest form of chatting, they have become popular especially on social websites because they allow seeing your chatting buddy as you chat. Other types of chat rooms include business and gaming hobby chat rooms where the users chat about common interests.

Features: Chat rooms offered by different websites range in features and security. Besides the standard "type" and "end" feature, some allow you to share pictures and share emotions that are used to portray your emotion or sudden expression. Some chat rooms also include multi-chat/group chats that allow two or more people to isolate from the chatting population and begin their own conversation. Some chat room websites include saving options that allow you to save your conversations so you can view them again whenever you want to .

Warning: While chat rooms may be exciting cyber locations to meet new people, they can also be a dangerous ground to run into stalkers, pedophiles and more. In simple terms, be careful. You should also be careful about chat room websites that do not have rules. The “anything goes” philosophy is a dangerous one because it means anyone can do or say anything without being held accountable. If you have children, monitor what chatting websites they use and the people they chat with.

Misconceptions: Just because a chat room is labeled "Christian or "Children's" chat room does not necessarily mean that it cannot be a danger zone. Unfortunately, many sex offenders and pedophiles roam through chat rooms looking for underage children. As shown in TV series “ To Catch a Predator.” there is no guarantee that any chat room website is completely safe, regardless of the rules, regulations and security tools.

Internet Relay Chat

INTERNET RELAY CHAT (IRC)

IRC is an Internet wide talk facility developed in 1988 by Jarkko Oikarinen in Finland and is used in over 60 countries around the world. The Internet Relay Chat (IRC) service provides a way for many users to communicate about a given topic. Each communication occurs on a separate "channel". A user who creates a channel chooses specifies whether the channel is open to anyone or restricted to the set of people specified by the channel creator. A user can request a list of the IRC channels currently in progress, and can choose to join one of the channel. When a user joins an IRC channel, the user enters a nickname that IRC uses to identify the user to others (eg. dancer or big shot). Users often choose nicknames that disguise their identity. For example, a woman might choose computer man., it can be used by anyone on the Net at any time.

A participant receives each line of text that another participant enters along with the participant's nickname. Thus, a participant who enters a line of text knows that all other participants will receive a copy of the text on their screens.

Web Chat

Web sites that enable you to converse directly with other Web users in a manner similar to IRC are providing interactive Web chat. Usually, these sites are referred to as chat rooms, a concept popularized by America Online. However, America Online chat rooms can be used only by AOL users, whereas anyone on the Internet with the right software can access a Web based chatting system.

You can find two main categories of interactive chat systems. Some are basically textual in nature—you type something, and what you type appears on the screens of people around the world who are connected to the same site. Others use graphics and even 3-D animation to provide you with a visual Avatar – your online representation in the chat room, which interacts with other users' avatars as you send and receive messages. Most chat rooms are written in Java, a computer language that enables programs to be distributed and run over the Internet.

Text based chat rooms are very much like IRC. You use your Web browser to connect to the Website, choose a handle, and see the messages in the chat room, when you type messages, anyone using the attic chat room can read what you type.

Video Conferencing

VIDEO CONFERENCING

Video conferencing uses telecommunications of audio and video to bring people at different sites together for a meeting. This can be as simple as a conversation between two people in private offices (point-to-point) or involve several sites (multi -point) with more than one person in large rooms at different sites."

Besides the audio and visual transmission of people, video conferencing can be used to share documents, computer-displayed information, and whiteboards. Improvements are being made in collaborative tools that allow people at different sites to electronically manipulate a common document of computer application.

The quality of a video conference primarily depends on the characteristics of the circuit between the conferencing sites. In the H.323 world, a high-quality conference (excellent audio and videos) needs about 768Kbs (KiloBits/Second) of bandwidth. On campus, this is usually possible since most data networks are at least 10Mbps (MegaBits/Second). It's also possible when the other sites have Internet 2 capabilities (the next generation Internet which is in use at virtually all R1 research universities and some business and government sites).

When a conversation includes a site off-campus, then bandwidth and type of connection between campus and the other site must be considered. If this path travels across the commodity Internet, then you are at the mercy of the activity on this public utility at the time of the conversation. Here are some general rules of thumb for bandwidth:

Need of Video Conferencing

Sometimes it's just not possible or practical to have a face-to-face meeting with two or more people. Sometimes a telephone conversation or conference call is adequate. Other times, an email exchange is adequate. Video conferencing adds another possible alternative. Consider video conferencing when:

- (a) a live conversation is needed.
- (b) visual information is an important component of the conversation.
- (c) the parties of the conversation can't physically come to the same location.
- (d) the expense or time of travel is a consideration.

Here are a few examples of how video conferencing can benefit people around campus :

- (e) faculty member keeps in touch with class while gone for a week at a conference.
- (f) guest lecturer brought into a class from another institution
- (g) researcher collaborates with colleagues at other institutions on a regular basis without time due to travel.
- (h) faculty member participates in a thesis defense at another institution
- (i) administrators on tight schedules collaborate on a budget preparation from different parts of campus
- (j) faculty committee auditions a scholarship candidate
- (k) researcher answers questions about a grant proposal from an agency or review committee.
- (l) student interviews with an employer in another city

E-Commerce

E-COMMERCE

E-Commerce is associated with buying and selling of information, products and services via computer networks. It is a new way of conducting, managing and executing business transactions using computer and tele network.

Electronic commerce has expanded rapidly over the past five years and this growth is forecast to continue or even accelerate. So, electronic commerce a general term for any type of business and commercial transaction, that involves the transfer of information across the internet. It includes e-marketing electronic funds transfer, online marketing, online transaction processing etc. Ecommerce technology is very beneficial to consumer. Online ordering helps consumer to buy online products without any physical involvement and save some time and money.

We can define electronic commerce in following three prospects

From a communication perspective, electronic commerce is the delivery of information, products/services, or payment via telephone lines, computer networks or any other means.

From a business process perspective, e-commerce is the application of technology toward the automation of business transactions and workflows.

From a service perspective, e-commerce is the tool that addresses the desire of firms, consumers and management to cut service costs while improving the quality of goods and increasing the speed of service delivery

Types of E-Commerce

1. Business to Business: B2B refers to electronic commerce between businesses rather than between a business and a consumer. Businesses can often deal with hundreds or thousands of other businesses, either as customers or supplier.

2. Business to Customer: B2C e-commerce means E-commerce involving an individual and a shop selling goods. Electronic mails, virtual storefronts allow individual consumers to browse for products and shop using credit cards. It is more like an extension of catalogue shopping through mail order and telephone ordering using credit cards for making payments.

3. Customer to Customer: It takes place on the internet, without any business middleman. You put an announcement at any of the consumer exchange website that you are selling say second hand laptops. Others with similar Internet then bid against each other for the object, so you get a price you normally could not dream.

Advantages of E-Commerce

1. It increases sales and decreases costs of goods.
2. It enables people to work from home and has an added benefit of reduction in traffic and pollution caused by employees who have to commute in office.
3. It provides the buyers a wide range of choices than traditional commerce.
4. Advertising on the web can make a big/small firm's promotional message reach out to potential customer all over the world.
5. Help protect against frauds and theft losses because electronic payments can be easier to monitor than payments made by cheques.
6. It increases sales and decreases costs of goods.
7. It enables people to work from home and has an added benefit of reduction in traffic.

Disadvantages of E-Commerce

1. Frauds are not completely eliminated in an E-commerce transaction.
2. Cyber laws are not properly enacted and the existing ones are not clearly defined.
3. Things like food, Jewellery, antique etc., can never turn to E-commerce because it is not possible to inspect them from remote location.
4. Many firms have had trouble recruiting and retaining employees with the technological, and business process skills need to create an effective electronic commerce presence.

INTERNET AND WEB TECHNOLOGY (IWT)



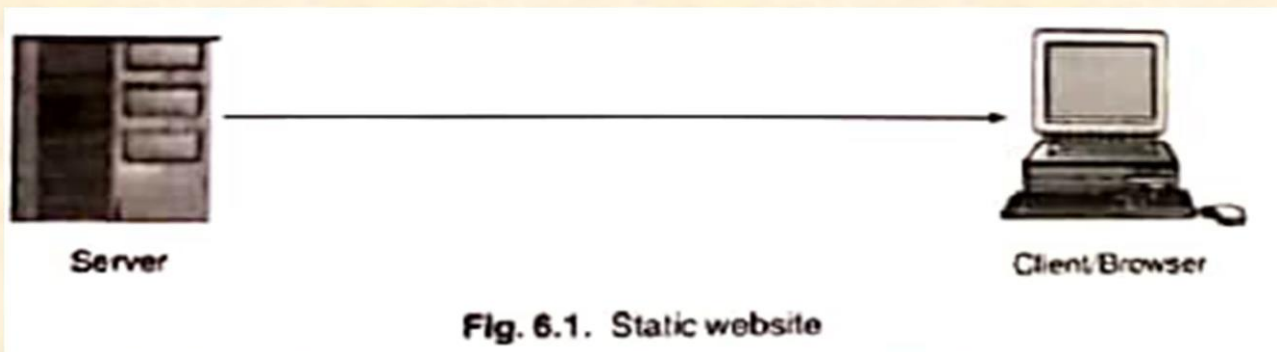
Chapter 5: Website Classifications

- ✓ 5.1 Static Websites
- ✓ 5.2 Dynamic websites
- ✓ 5.3 Web portals
- ✓ 5.4 Social Networking Sites
- ✓ 5.5 RSS Feed, Blog, Netiquette

Chapter 5 - Internet Connectivity & WWW

Static Websites

- A static website contains Web pages with fixed content. Each page is coded in HTML and displays the same information to every visitor. Static sites are the most basic type of website and are the easiest to create. It does not require any web programming or database design. A static site can be built by simply creating a few HTML pages and publishing them to a web server.
- Since static web pages contain fixed code, the content of each page does not change unless it is manually updated by the webmaster. This works well for small websites, but it can make large sites with hundreds or thousands of pages difficult to maintain. Static sites that contain a lot of pages are often designed using templates. This makes it possible to update several pages at once, and also helps provide a consistent layout throughout the site.



- The static website is simple website design which is cost effective and beneficial for the small enterprises or individual to expand their business through web. Through static websites individual to expand their business through web. Through static website individuals or small business houses can place simple information regarding their company and products in simple manner and at low cost. This type of website is very useful for expanding market of company with its information and appearance on internet.
- Static websites follow “what you see is what you have” concept. thus, i deal for those companies or individuals who just need website to establish their web presence or use it as a contact platform for their clients.

Examples of Static Websites

- www.yashamanhospital.com
- www.mukundamhospitalandkindeycentre.com
- www.difgardencitygurgaon.net
- www.rajputanasecurityguard.com

Advantages of Static Websites

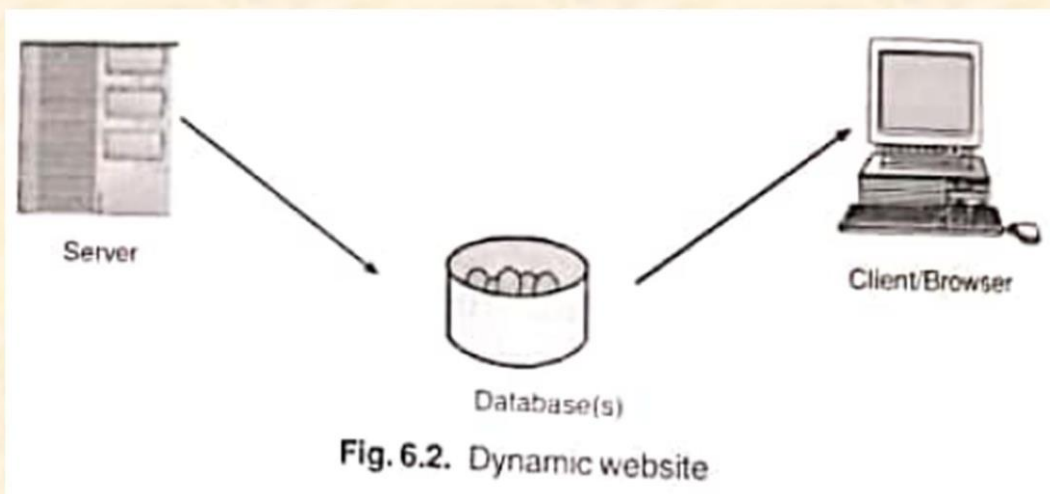
- Quick to develop
- Cheap to develop
- Cheap to host

Disadvantages of Static Websites

- Requires web development expertise to update site.
- Site not as useful for the user,
- Content can get stagnant.

Dynamic websites

- A dynamic website contains information that changes, depending on the viewer of the site, the time of the day, the time zone, the native language of the country the viewer is in or many other factors.
- Dynamic websites contain web pages that are generated in real-time. These pages include web scripting code, such as PHP or ASP. When a dynamic page is accessed, the code within the page is passed on the web server and the resulting HTML is sent to the client web browser,



- Most large websites are dynamic. Since they are easier to maintain than static websites. This is because static pages each contain unique content meaning must be manually opened, edited and published whenever a change is made. Dynamic Pages, on the other hand, access information from a database. Therefore, to alter the content of a dynamic page, the web master may only need to update a database record. This is especially helpful for large sites that contain hundred or thousands of pages. It also makes it possible for multiple users to update the content of a website without editing the layout of the pages.

- Dynamic websites that access information from a database are also called database-driven websites.
- Dynamic sites can be more expensive to develop initially, but the advantages are numerous. At a basic level, a dynamic website can give the website owner the ability to simply update and add new content to the site.

Example of Dynamic Website

- Blogs, e-commerce sites, calendar or to-do sites, and other sites that requires updating often.
- Here is a simple way to find to find the whether the site is dynamic or not. If you interact with it, apparently, then it is a dynamic site, which is complicated and expensive to create since they stores information in data base interacted by using PHP like languages.
- Generally speaking, very few people choose the simplicity and ease of the static website, in the other hand the other few are fascinated with functionality and intuitiveness of dynamic websites. As a matter of fact, both have their own advantages and limitations.

Advantages of Dynamic Website

- Much more functional website.
- Much easier to update,
- New content brings people back to the site and helps in search engines
- Can work as a system to allow staff or users to collaborate.

Disadvantages of Dynamic Websites :

- Slower/more expansive to develop.
- Hosting costs a little more.

Web portals

- A web portal is most often one specially designed webpage at a website which brings information together from diverse sources in a uniform way, Usually, each information source gets its dedicated area on the page for displaying information (a portlet) often the user can configure which ones to display. The extent to which content is displayed in a “uniform way” may depend on intended user and the intended purpose, as Well as the diversity of the content.

- A portal may use a search engine API to permit users to search intranet content as opposed to extranet content by restricting which domains may be searched, Apart from this common search engines feature, web portals may offer other services such as e-mail, news, stock quotes, information from databases and even entertainment content. Portals provide a way for enterprises and organizations to provide a consistent look and feel with access control and procedures for multiple applications and databases, which otherwise would have been different web entities at various URLs. The features available may be restricted by whether access is by an authorized and authenticated user (employee, member) or an anonymous site Visitor,
- Examples of Early Public webportal were AOL, Excite, iGoogle, MSN, Rediff and Yahoo,

Classification of Web Portals

- Web portals are classified as horizontal or vertical.
- A horizontal portal is used as a platform to several companies in same economic sector or to the same type of manufacturers or distributors.
- A vertical portal (also known as portal) is a specialized entry point to a specific market or industry niche, subject area or interest. Some vertical portals are known as “Vertical information portals” (VIPs). VIPs provide news, editorial content, digital publications and e-commerce capabilities.

Social Networking Sites

- Social networking platforms may allow organizations to improve communication and productivity by disseminating information among different groups of employees in a more efficient manner.

Advantages of Social Networking Sites

- Facilitates open communication, leading to enhanced information discovery and delivery.
- Allows employees to discuss ideas, post news, ask questions and share links.
- Provides an Opportunity to widen business contacts.
- Targets a wide audience, making it a useful and effective recruitment tool.
- Improves business reputation and client base with minimal use of advertising.
- Expands market research, implements campaigns, delivers communications and directs integrated people to specific websites.

Disadvantages of Social Networking Sites

- Opens up the possibility for hackers to commit fraud and launch spam and virus attacks.
- Increases the risk of people falling prey to online scams that seem genuine, resulting in data or identity theft,
- Potentially results in lost productivity, especially if employees are busy updating profiles etc.
- Potentially results in negative comments from employees about the company or potential legal consequences if employees use these sites to view objectionable, illicit or offensive material.

A Social Networking Use Policy Generally

- Define what social networking is particular to your organization, so employees know exactly what is meant by the term.
 - Establishes a clear and defined purpose for the policy.
 - Communicates benefits of social networking and of having a policy.
 - Provides a clear platform for educating employees,
 - Refers to confidentiality of employer trade secrets and private or confidential information. Talks about productivity in terms of social networking.
 - Provides examples of policy violations.
 - Outlines disciplinary measures to be taken for policy violations.
- What may be the most concerning aspect of social networking platforms is that they encourage people to share personal information, Even the most cautious and well-meaning individuals can give away information they should not, the same applies to what is posted on company approved social networking platforms.

Examples of Social Networking Sites

- Facebook, Flickr, Google+, Hi5, ibibo, MyOpera, Orkut

RSS Feed, Blog, Netiquette

RSS Feed

- RSS is about getting live web feeds directly to your computer. RSS takes the latest headlines from different websites, and pushes those headlines down to your computer for quick scanning.

- The acronym RSS stands for many versions of the same thing:
 - Really simply syndication,
 - Rich site summary (RS 0.91)
 - RDF site summary (RS0.9 and 1.0)
 - Real-time simple syndication (RSS 2.0),
- In each of the above meaning, the purpose is the same to have websites of your choice deliver their latest news directly to your monitor. So instead of having to visit 14 different places to get your weather, sports, favourite photos, latest gossip or latest political debates, you just go to one screen and see it combined “aggregated” into a single window. The RSS headlines and stories are effectively immediate. Once published at the source server, RSS headlines take only moments to get to your screen.

➤ Why You Would Use RSS

- **Hurricane Watch:** If you live in “hurricanently” of the USA, you will want to pay close attention to hurricane warning and evacuation tips. RSS is absolutely a good tool here.
- **News:** To get the freshest news on your favourite celebrity, the country you are about to visit, or your favourite sports team.
- **Hobby Interests:** If you are motor cyclist, a skier, or perhaps, a dog trainer, hundreds of conversations and bits of hobby advice can be fed directly to your screen.
- **Photos:** If you like to change your computer wallpaper daily, the RSS feeds are an excellent way to get the latest from photographers on the web.
- **Reading Your Friends Blogs:** If you have loved ones around the globe who do blogging, then you can have all their latest entries fed directly to your screen. This is very helpful for families when one of their own is in the military serving in Iraq or Afghanistan or Africa. A good way to read about how they are doing during their military assignment.
- **Politics:** If you are helping a political candidate get elected, then RSS is an invaluable tool for watching popular opinion and blog postings.
- **Jokes and Inspirational Quotes:** Add a clean laugh to your morning, or a pick-me-up quote from a famous person_RSS can do that for you.
- **Currency Exchange Rates:** If you are planning a trip to another country, you can watch for when the best time is to buy the that currency,

How it Works:

1. **Behind the Scenes:** RSS headlines are really simple text files that the publishing web master submits to a special feed server. That RSS feed server, in turn, pushes the text file to the screens of its subscribers. Time lag is usually 30 seconds to 30 minutes before the subscribers see the updates. In most cases, the lag is not noticeable.
2. **To Get Started:** You choose an RSS reader tool for yourself. Most RSS readers are free to use and easy to learn.
3. **Setup Your Screen:** You load the RSS feeds into your reader tool. This is achieved through multiple different ways, You can visit the web feed site directly, you can copy – paste the special code from an e-mail, or you can load copies from your friend’s RSS reader screen.
4. **Then You Start Reading Your Web Feed News:** You simply log in to your RSS reader page or start your RSS software, and you can scan all your web feeds instantly. You can arrange RSS feeds into folders, just like e-mail, and you can even set alert and sounds for when particular web feed is updated.

Examples of RSS Readers:

- Newsgator.com
- Bloglines.com
- My Yahoo Reader
- Goodle Reader

BLOG

- A blog (a truncation of the expression web logs) is a discussion or informational site published on the world wide web and consisting of discrete entities(“Posts”) typically displayed in reverse chronological order the most recent post appears first). Until, 2009 blogs were usually the work of a single individual occasionally of a small group, and often caused a single subject. More recently “multi-author blogs”(MABs) have developed, with posts written by large numbers of authors and professionally edited. The rise of Twitter and other “Microblogging” systems helps integrated MABs and single –author blogs into societal newstreams. Blog can also be used as a verb, meaning to maintain or add content to a blog.
- Blogging can be seen as a form of social networking service. Indeed, bloggers do not only produce content to post on their blogs, but also build social relations with their readers and other bloggers. There are high-readership blogs, which do not allow comments, such as Daring Fireball.
- Many blogs provide commentary on a particular subject, other functions as more personal online diaries. Most blogs are primarily textual, although some focus on act (act blogs),

photographs (photoblogs), of blogging, featuring very short posts. In education, blogs can be used as instructional resources. These blogs are referred to as edublogs. According to critics and other bloggers. Blogger is the most popular blogging service used today.

NETIQUETTE

- The word netiquette derives from new and etiquette and as such, is a set of established conventions that have evolved over time on the internet and on the usenet news. Netiquette is most of all based on:
 - What has been found useful and proper in the electronic form of communication made possible by the Internet ?
 - What is appropriate in any form of communication between civilized human beings?
 - and what is dictated by the common sense ?

What Good might following the netiquette do you ?

What you write and how you behave makes you internet personality, which then directly influences and alters your “real-world personality”. Think carefully what kind of picture you wish to give of yourself, One day you will most probably have dealings with a few of the persons who have formed their picture of you on the internet in real life. Displaying a good familiarity of netiquette in one’s postings can sometimes have a “secret handshake effect” on the internet.

If you pay attention to netiquette:

- Your communication will be better you will gain more credibility.
- You will in all probability get helpful answers more readily when you have questions to ask or need help on the internet.

If, however, you repeatedly and deliberately ignore the netiquette, you will soon develop a reputation of a troublemaker and at least the more serious users will start avoiding you altogether.

7 Chapter

Developing Portals Using HTML

INSIDE THIS CHAPTER

7.1. Design a Web Page 7.2. Good Web Design 7.3. HTML Introduction 7.4. HTML Tags/Elements
7.5. HTML Anchor Tag 7.6. HTML Table Tag 7.7. HTML Frames 7.8. HTML Forms 7.9. Disadvantages
of HTML 7.10. Separating Style from Structure with Style Sheets

7.1. DESIGN A WEB PAGE

We have following steps to understanding the design a good web page :

7.1.1. Gathering Information

1. What kind of information are they presenting (homepage, informational article, etc.) ?
2. How did they present the information (page layout) ?
3. Does their webpage download quickly ?
4. Can I navigate easily throughout their activity ?
5. Does this site engage the visitor ?
6. Is the content rich enough that I will want to return again ?
7. Do all of the links work ?
8. Have they included a contact resource person ?
9. Does it tell when the page was last updated ?
10. How could you improve this webpage ?

7.1.2. Determine Your Audience

Your audience will effect the design and content of your project. If your target is educators, your presentation will be different than if you were creating a page intended for children to use. It is better to offer a separate page for educators and another one for children if you want to offer both kinds of resource or activity in your project.

7.1.3. Create a Storyboard

103

As you begin designing your web page, remember that the top structure should be a menu format pointing to available resources. Include folders near the top of your structure so that new resources can be added at a later date. Create a storyboard of ideas that you want to use in your activity. It's helpful to write each concept or activity on a different piece of paper, then physically arrange them until you have the structure you want.

When naming your files, avoid using uppercase letters in the filename. Try to use short, but descriptive names for each file that you are creating.

7.1.4. Plan Your Navigational Tools

Plan how people will navigate within your activity. Every page should have a link to either the main menu page of the activity or a link to your school or class homepage. Your class homepage should have a link to your school's homepage and the school's homepage should include a link to the corporation's homepage.

If your activity contains more than one file, you need to plan how visitors will advance to the next page or return to the previous page.

7.1.5. Create an Aesthetically Appealing Web Page

We've become a very visual society. Although content is more important than a "glitzy" page layout, your visual presentation plays an important role to the appeal of your lesson. Include the following items in your webpage :

1. Graphics

- ❖ If your graphic is larger than a two inch square or a one by three inch rectangle, use a thumbnail image of the picture.
- ❖ Reduce the color palette to the lowest possible palette size to ensure quick download of your webpage.
- ❖ Use the same graphics as often as possible. An image that is used repetitively, only had to be downloaded once. Using a large number of different images will slow the download time of your webpage. Visitors may decide not to visit your webpage if it takes a long time to download.

2. Add background color or a background image.

3. **Vary your font color.** *Light backgrounds should have dark text and dark backgrounds should have light text. Be sure that your text is readable if you are using a background color or image. Be careful when choosing your colors. Not all computers see the same color hues.*
4. **Vary your font size.** *Use the font size tags to vary the body text and the header tags to vary the size of your headings and sub-headings.*
5. **Give credit to the graphic's artist.** *Read the terms artists have given when using an image from their website. Many appreciate a link back to the website where you got the graphic, while others state that no link is necessary.*

7.1.6. Establish Credibility

Each webpage should include pieces of information that will establish credibility to your activity. This can be a standard closing at the end of each webpage that you simply copy and paste from on webpage to the other. You should include the following information:

1. The name of the author of this activity.
2. E-mail address of the contact person for this activity.
3. When was this webpage last updated and links were checked.
4. The name of the host school or organization.

7.2. GOOD WEB DESIGN

The design of your site is the most important part of building a website. Make a bad choice here and it won't matter how great your content is or how much advertising you do. If your site looks bad no one will visit and those that do won't stay long or buy anything. Therefore, the right design for your site is the one that's right for you. How do you know what's right for you when there are so many different styles of web design out there? There are so many different ways you can design your site that it's easy for someone to get so lost when designing their site that they may give up all together.

You have to remember that the choices are all yours, because it's your site, and you have to decide what you like. Choices like which colors, styles, fonts, graphics and writing styles to use are all very personal choices and not to be made lightly. What you say and do on your website says a lot about who you are?

TIPS

A good website is a critical component of any new business launch. It does not mean creating a few pages of text. It has an importance attached to it. Therefore, website design is a unique blend of artistic skill, technical prowess. It also involves the knowledge of constantly changing technology and the implementation of a complete Internet marketing campaign.

Here are some of the basics you need to consider when designing your site.

1. Layout

How do you want your page to look? Do you want all the text to the right and graphics on the left, or vice-versa? Do you want an advertisement at the top of the page or do you want all your ads on the sides?

Layout is an interesting beast. There is no definitive right or wrong ways of approaching it. What you need to do is make sure that the layout *works* for the site that you are designing. The layout of all the pages may have the same looks or it may be different depending upon what suits to the reader. Layout primarily consists of the following aspects:

- (a) **Whitespace:** Whitespace is space between elements on a page. It is often referred to as *negative space*. Some websites have a policy of no white space between text or other graphics. But this is not recommended. A good use of whitespace can make a good design great.
- (b) **Flow:** Flow in web design is the act of guiding the user through page. All sites makes use of some form of flow but some are more effective than others. Effective flow takes in all aspects of web design including properties that are not directly related to layout such as color, size of element, typography and images.
- (c) **Alignment:** Alignment helps create a good flow. An alignment plays a key part in the overall flow. Alignment also contributes to a natural order that makes content easier to scan and read.

2. Navigation

Navigation is one of the most important things to consider when designing a website. After all, it is one of the main ways that will determine how a user interacts with a site. It is also an important thing to consider as it has an effect on usability.

Easy to use navigation is essential to a good website. If people can't find what's on your site, well, they just won't stick around, will they? Hence, to make navigation easy, the site's navigation buttons should be grouped together. Make sure that links on all your pages are working. Whether they are internal or external links they must work.

3. Typography

Effective use of typography helps content become more readable and the easier the content is to read for the site's users. It also ties in well with whitespace as that can give text enough room to breathe, making it easier to read. Text is the most common element of design, so it's not surprising that a lot of thought has gone into it. It's important to consider things like:

- (a) **Font Choices:** Different types of fonts say different things about a design. Some look modern, some look retro. Make sure you are using the right tool for the job.
- (b) **Font sizes:** Years ago it was trendy to have really small text. Happily, these days' people have started to realize that text is meant to be read, not just looked at. Make sure your text sizes are consistent, large enough to be read, and proportioned so that headings and sub-headings stand out appropriately.
- (c) **Spacing:** As discussed above, spacing between lines and away from other objects is important to consider. You should also be thinking about spacing between letters, though on the Web this is of less importance, as you don't have that much control.
- (d) **Line Length:** There is no hard and fast rule, but generally your lines of text shouldn't be too long. The longer they are, the harder they are to read. Small columns of text work much better (think about how a newspaper lays out text).
- (e) **Color:** is an often underrated aspect of web design but can play a very important role in usability as well as convey the overall meaning of a brand as well as the overall mood of the website. Different color combinations can evoke different emotions and reactions. Color design is often very subjective but it is important to keep usability in mind. Changing color for subjective reasons may ultimately have a negative affect on usability.
- (f) **Paragraphing:** Unfortunately, justified text tends to create weird gaps between words where they have been auto-spaced. This isn't nice for your eye when reading, so stick to left-aligned unless you happen to have a magic body of text that happens to space out perfectly.

4. Clarity (Sharpness)

Keeping your design crisp and sharp is super important in Web design. And when it comes to clarity, it's all about the pixels. In your CSS, everything will be pixel perfect so there's nothing to worry about, but in Photoshop it is not so. To achieve a sharp design you have to :

- ❖ Keep shape edges snapped to pixels. This might involve manually cleaning up shapes, lines, and boxes if you're creating them in Photoshop.
- ❖ Make sure any text is created using the appropriate anti-aliasing setting.

- ❖ Ensuring that contrast is high so that borders are clearly defined.
- ❖ Over-emphasizing borders just slightly to exaggerate the contrast.

5. Consistency

Consistency means making everything match. Heading sizes, font choices, coloring, button styles, spacing, design elements, illustration styles, photo choices, etc. Everything should be in theme to make your design coherent between pages and on the same page.

6. Usability

Web design isn't just about pretty pictures. With so much information and interaction to be effected on a Website, it's important that you, the designer, provide for it all. That means making your Website design usable. We've already discussed some aspects of usability—navigation, precedence, and text.

7. Ads

Too many advertisements on your site make for a bad user experience. Try for well placed, highly noticeable ads instead.

8. Things That Blink

Blinking can be annoying and you don't want to annoy your readers.

9. Noise

A lot of people like to add music to their Websites. That's all well and good, but if it's a loud song and your reader jumps out of their seat when they open your Web page they will leave very quickly and never return.

10. Graphics

Graphics is an important part of a Website. They add color, attitude, and theme. They're necessary, right? Right, as long as you don't over do it. Too many graphics not only make your site upload really slow but it also makes it look like a kid scribbled on your page. Here are few points to be kept in mind while adding graphics to your website.

- ❖ Buttons are not big and dorky
- ❖ Every graphic has an *alt* label
- ❖ Every graphic link has a matching text link
- ❖ Graphics and backgrounds use browser-safe colors
- ❖ Animated graphics turn off by themselves

11. Content

What is your site going to be about? What kinds of things are you going to write about? Of course, this is completely up to you. Whenever you write something make sure you spell check it and, if you have time, re-read it to make sure there are not grammatical errors.

12. Maintenance

After creating a Website you have to maintain it. If you have links to other Web sites you need to check them periodically to make sure the links still take you somewhere because sometimes Web sites disappear.

7.3. HTML INTRODUCTION

DEFINITION

The standard language the web uses for creating and recognizing hypermedia documents is the Hypertext Markup Language (HTML). It is loosely related to, but technically not a subset of the Standard Generalized Markup Language (SGML), a method of representing document formatting languages.

Languages such as HTML that follow the SGML format allow document writers to separate information from document presentation—that is, documents containing the same information can be presented in a number of different ways. Users have the option of controlling visual elements such as fonts, font size and paragraph spacing without changing the original information.

HTML is widely praised for its ease of use. Web documents are typically written in HTML and are usually named with the suffix “.html”. HTML documents are nothing more than standard 7-bit ASCII files with formatting codes that contain information about layout (text styles, document titles, paragraphs, lists) and hyperlinks.

7.3.1. Features of HTML Language

This language gained popularity because of its following advantages:

1. It is easy to understand and can be easily modified.
2. It provides a flexible way to design the web pages along with the text.
3. Graphics, video, sound can also be used and imported to give attractive look to the web pages.
4. Effective presentations can be made with all formatting effects.
5. HTML documents can be displayed on any platform such as: Macintosh, Windows and UNIX.

7.3.2. HTML Document Structure

Html documents are structured into two parts, the *head*, and the *body*. The *head* contains information about the document that is not generally displayed with the document, such as its title. The *body* contains the body of the text, *i.e.*, where you place the document material to be displayed. Elements allowed inside the *head*, such as *title*, are not allowed inside the *body*, and vice versa.

Structure of a HTML document:

```
<html>
  <head>
    <title>
      .....
    </title>
  </head>
  <body>
    .....
    .....
  </body>
</html>
```

Example:

```
<html>
  <head>
    <title>
      Introduction to HTML
    <\title>
  <\head>
  <body>
    This is the first HTML Page
  <\body>
<\html>
```

7.3.3. Creating a HTML Program

To create a HTML document, type *text* in any word processor like Notepad, WordPad and then create the page. The following are the steps through which we can create a HTML document using Notepad.

- (1) Open Notepad by clicking on Start => Programs => Accessories => Notepad.
- (2) The Notepad Window will appear. Type the HTML code in the Notepad.
- (3) Click on File => Save menu option. The **Save As** dialog box will appear. Select the folder to save the file. Select **All Files** in the **Save As** type option. Give the name in the file name text box with extension .html or .htm.
- (4) Click on the **Save** button.

7.3.4. Viewing a HTML Program

To view any HTML file, we need only the web browser. To view the web page, follow these steps :

- (1) Click on Start => Programs menu and choose **Internet Explorer** option. The Internet Explorer Window will appear.
- (2) Click on File => Open menu option. The Open dialog box will appear. Click on **Browse** button and dialog box will appear.
- (3) Choose the path where you have stored the HTML document and after selecting your file. Click on **Open** button and Click **OK** on Open dialog box.
- (4) Resultant web page will be displayed on the Web Browser.

Alternative way of viewing a HTML Program

Choose the path where you have stored the HTML document and after selecting your file, double click on your HTML document. Your web page will be displayed on the **Web Browser**.

7.3.5. Naming Conventions for HTML Documents

When the HTML browser (Netscape Navigator, Internet Explorer, Opera, lynx etc.....) retrieves a file, it must know what type of data it has received in order to know what to do with it. Hypertext transfer protocol (that is, HTTP) servers explicitly tell the browser about the type of the data being sent. In other cases, such as when the browser is using FTP to access a remote file, or when the browser is reading a file from the local disk (such as when you are editing pages prior to publishing them to a Web server), the browsers guesses the data type from the filename extension — that is the part after the dot in the filename.

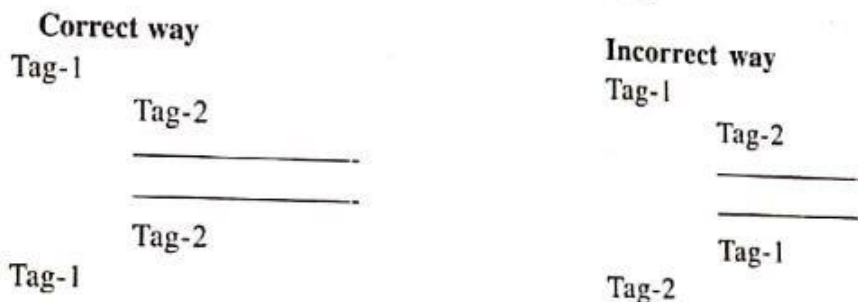
For example, HTML files are identified by names such as *name.html*, where the *.html* extension indicates an HTML document. Four letter extensions are common. Generally the extension is truncated to three letters i.e., *.html* becomes *.htm*. Some of the standard extensions useful while using HTML are stated below along with their meanings:

- ❖ **.html (also specified as .htm)** : HTML document, containing text and HTML mark-up instructions.
- ❖ **.txt** : A plain text file. The browser presents the file as a block of text and does not process it for mark-up instructions. Browsers generally treat unknown types of data as a text file.
- ❖ **.gif** : A GIF format image file.
- ❖ **.xbm** : A X-Bitmap (black&white) image file.
- ❖ **.xpm** : A X-Pixmap (color) image file.
- ❖ **.jpeg (also .jpg)** : A jpeg-encoded image file.
- ❖ **.mpeg (also .mpg or .mpe)** : A mpeg-encoded video file.

7.4. HTML TAGS/ELEMENTS

HTML is a tag-based language. Tag is an element that instructs the web browser what to show and how to show. A tag comprises of text enclosed in angular brackets $\langle \rangle$. These tags are not case sensitive i.e., $\langle \text{tag_name} \rangle$ and $\langle \text{TAG_NAME} \rangle$ has the same effect. The tags are of two types:

- (a) **Container Tags:** These tags include both the ON and OFF tag. The ON is denoted as $\langle \rangle$ and OFF is denoted as $\langle / \rangle$ (i.e., $/$) sign marks the difference between starting and ending tag). If tags are nested, always keep these tag sets balanced.



- (b) **Empty Tags:** These tags include only the ON tag. These elements do not enclose any data.

7.4.1. Description of Various Tags

1. **ADDRESS tag:** The ADDRESS tag should be used to enclose contact information, addresses and the likes. It is often rendered with a slightly indented left margin and italics.

Appearance:	$\langle \text{ADDRESS} \rangle \dots \dots \dots \langle / \text{ADDRESS} \rangle$
Attributes:	None.

2. **APPLET tag:** The APPLET tag is used to include Java applets. The CODE attribute indicates the location of the class of the applet itself. CODEBASE can be used to specify an absolute URL for the applet.

Appearance:	$\langle \text{APPLET CODE=string HEIGHT=n WIDTH=n} \rangle \langle / \text{APPLET} \rangle$
Attributes:	CODEBASE=URL, CODE=string, NAME=string, ALT=string, ALIGN=left right top middle bottom, HEIGHT=n, WIDTH=n, HSPACE=n, VSPACE=n

The NAME attribute gives the name of the applet. Just as IMG, WIDTH and HEIGHT are used to specify the width and height of the applet's window, and HSPACE and VSPACE control horizontal and vertical spacing around the applet. ALIGN sets the horizontal or vertical alignment for the applet. The ALT text may contain text that should be displayed if the applet cannot be run, but you should use the contents of APPLET instead. In here you may use markup, so you can provide a better alternative than with the ALT text.

Not all browsers support Java applets, and those that do often allow the user to disable it.

3. AREA tag: Inside the MAP tag, each "hotzone" in the client-side imagemap is defined with an AREA tag. The HREF attribute specifies the URL for the destination that should be chosen if this area was selected. If you specify NOHREF instead, this area won't do anything.

Appearance:	<AREA SHAPE=x HREF=URL COORDS=string ALT=string>	
Attributes:	SHAPE=rect circle poly default, NOHREF HREF=URL, ALT=string	COORDS=string,

SHAPE and COORDS define the actual region. SHAPE can be a rectangle, circle, or polygon, and COORDS should contain a set of coordinates describing that shape. This is done with a comma-separated list of numbers, enclosed in quotes. If SHAPE is set to DEFAULT, no coordinates need to be specified. The default area is what will be chosen if no others match. The syntax for COORDS depends on what shape you choose. The ALT text is used by text browsers to present the URLs in the imagemap in a more readable fashion.

4. ANCHOR tag: The anchor tag is the "glue" for hypertext documents. The enclosed text and/or image(s) will be selectable by the user, and doing so will take the user to the location specified in the HREF attribute. The TITLE attribute can be used to provide a description of that location, which is displayed by some browsers when the mouse moves over the URL. The NAME attribute is used to set up "named anchors." The enclosed text will be marked as a "target" to which a browser can jump directly.

Appearance:	
Attributes:	HREF=URL, NAME=string, TITLE=string

5. BASEFONT tag: The BASEFONT tag is used to suggest a default font size. FONT tags using a relative font size will use the value set here. The SIZE attribute is an integer value ranging from 1 to 7. The base font size applies to the normal and preformatted text but not to headings, except where these are modified using the FONT element with a relative font size.

Appearance:	<BASEFONT SIZE=n>
Attributes:	SIZE=n

6. BASE tag: The BASE tag is used to indicate the correct location of the document. Normally, the browser already knows this location. But in cases such as a mirrored site, the URL used to get the document is not the one that should be used when resolving relative URLs. That's when you use the BASE tag. The required HREF attribute must contain a full URL which lists the real location of the document.

Appearance:	<BASE HREF=URL>
Attributes:	HREF=URL

7. BIG tag: A browser should display the enclosed text in a larger font if available, and ignore the tag otherwise.

Appearance:	<code><BIG> </BIG></code>
Attributes:	None.

8. BLOCKQUOTE tag: If you are quoting more than a few lines from a document, use a BLOCKQUOTE to indicate this. Block quotations are often rendered with indented margins, and possibly in italics, although rendering with the standard quotation symbol for E-mail, ">", is ofcourse also possible. Do not use BLOCKQUOTE simply to create indented text. This is not the required rendering, so you will not achieve the effect you want on all browsers.

Appearance:	<code><BLOCKQUOTE> </BLOCKQUOTE></code>
Attributes:	None.

9. BODY tag: The BODY tag contains the actual contents of the document. Those contents should consist of block elements only. The BACKGROUND attribute should point to the location of an image, which is used as the (tiled) background of the document. The other attributes set the colors for the background, text, links, visited links and active links.

Appearance:	<code><BODY> </BODY></code>
Attributes:	BACKGROUND=URL, BGCOLOR=#RRGGBB, TEXT=#RRGGBB, LINK=#RRGGBB, VLINK=#RRGGBB, ALINK=#RRGGBB

TEXT sets color of the text on the page. LINK sets color of links that haven't been followed yet. VLINK sets color of links that have been followed. ALINK sets color of links while you are clicking on them.

10. BR tag: The BR tag is used to force line breaks within text. Normally, linebreaks are treated as a space by browsers (except inside the PRE tag). The optional CLEAR attribute is used when you have an IMG image in your text. If that image uses ALIGN=LEFT or ALIGN=RIGHT, the text will flow around it. If you have text you want below the image, you can do this with `<BR CLEAR=LEFT>` or `<BR CLEAR=RIGHT>` to force scrolling down to a clear left or right margin, respectively. Using CLEAR=ALL will scroll down until both margins are clear. CLEAR=NONE is the default, and does nothing.

Appearance:	<code>
</code>
Attributes:	CLEAR=left all right none

11. B tag: B is used to indicate that the enclosed text must be rendered in a bold typeface. If you want to indicate strong emphasis, use the STRONG element instead.

Appearance:	<code> </code>
Attributes:	None.

12. CAPTION tag: The CAPTION tag is used to provide a caption for a TABLE. This caption can either appear above or below the table. This can be indicated with the ALIGN attribute. It is usually centered with respect to the table itself, and usually appears in bold or is emphasized in some other way.

Appearance:	<CAPTION> </CAPTION>
Attributes:	ALIGN=top bottom

13. **CENTER tag:** It is used to indicate that large blocks of text should appear in center.

Appearance:	<CENTER> </CENTER>
Attributes:	None.

14. **CITE:** The CITE element marks up the title of a cited work. For example, a text discussing HTML could say <CITE>RFC 1866</CITE>

Appearance:	<CITE> </CITE>
Attributes:	None.

15. **CODE:** CODE is used for snippets of code that appear inside a paragraph of text. It is usually rendered in a mono-spaced font. For larger blocks of code, use PRE instead.

Appearance:	<CODE> </CODE>
Attributes:	None.

16. **DD tag:** The DD tag is used inside a DL definition list to provide the definition of the text in the DT tag. It may not only contain block elements but also plain text and markup. The end tag is optional, as it's always clear from the context where the tag's contents ends.

Appearance:	<DD> </DD>
Attributes:	None.

17. **DFN:** DFN is used to mark up terms that are used for the first time.

Appearance:	<DFN> </DFN>
Attributes:	None.

18. **DIR:** The DIR element is similar to the UL element. It represents a list of short items, typically up to 20 characters each. Items in a directory list may be arranged in columns, typically 24 characters wide. It is not permitted to use a block element, list element or TABLE in a LI inside a MENU or DIR.

Appearance:	<DIR> </DIR>
Attributes:	COMPACT

19. **DIV tag:** The DIV tag is used to mark up divisions in a document. It can enclose paragraphs, headers and other block elements.

Appearance:	<DIV> </DIV>
Attributes:	ALIGN=left right center

20. **DL:** DL is used to provide a list of items with associated definitions. Every item should be put in a DT, and its definition goes in the DD directly following it.

Appearance:	<DL> </DL>
Attributes:	COMPACT

21. **DT:** The DT tag is used inside DL. It marks up a term whose definition is provided by the next DD. The DT tag may only contain text-level markup.

Appearance:	<DT> </DT>
Attributes:	None.

22. **EM:** EM is used to indicate emphasized text.

Appearance:	
Attributes:	None.

23. **FONT:** The FONT tag can be used to change the appearance of the current block, in terms of size and color.

Appearance:	
Attributes:	SIZE=string, COLOR=#RRGGBB

24. **FORM:** Forms allow a person to send data to the WWW server. You can use the INPUT, TEXTAREA and SELECT tags to add individual elements, such as checkboxes, input fields or "drop down" lists to your form. A form may contain all markups (both text and body level tags), but it may not have a nested form.

FORM has one required attribute, ACTION, specifying the URL of a CGI script that processes the form and sends back feedback. There are two methods to send form data to a server. GET, the default, will send the form input in an URL, whereas POST sends it in the body of the submission. The latter method means you can send larger amounts of data, and that the URL of the form results doesn't show the encoded form.

Appearance:	<FORM ACTION=URL> </FORM>
Attributes:	ACTION=URL, METHOD=get post, ENCTYPE=string

An encoding type is specified with ENCTYPE, the default of "application/x-www-form-urlencoded" is most widely supported. An alternative is "text/plain", which is typically used in combination when the ACTION attribute points to a mailto: URL. If a browser supports both, the contents of the form are sent in plain text to the indicated recipient.

25. **FRAMESET:** FRAMESET starts any frame page. It alerts the browser that FRAMES are going here. Attributes to the FRAMESET are: COLS, ROWS. They break the frame into columns and rows.

Appearance:	<FRAMESET> </FRAMESET>
Attributes:	ROWS=",...", COLS=",..."
Appearance:	<FRAME FRAMEBORDER="0"> </FRAME>
Attributes:	FRAMEBORDER="0", MARGINHEIGHT="x%", MARGINWIDTH="x%">

26. H1: Headers should be used in hierarchical order. The level 1 heading is the most important header in the document. It is usually used to indicate the title of the document. The optional **ALIGN** attribute controls the horizontal alignment of the header. It can be **LEFT**, **CENTER** or **RIGHT**.

Appearance:	<code><H1> </H1></code>
Attributes:	<code>ALIGN=left right center</code>

There are six header tags available *i.e.*, from H1 to H6, where H1 displays the text in largest font supported, H2 displays the text less larger than H1 and so on. H6 displays the smallest text.

For other headers the attributes are same.

27. HEAD : The HEAD part of the document provides information about the document. It should not contain text or normal markup. If a browser encounters such markup, it will assume it has arrived in the BODY section of the document already.

Appearance:	<code><HEAD> </HEAD></code>
Attributes:	None.

28. HR: HR is used to draw horizontal rules across the browser window. The thickness of the rule is specified with the **SIZE** attribute, which takes an integer number of pixels. The width of the rule can be specified in number of pixels or as a percentage of the currently available window width, using the **WIDTH** attribute. Don't forget that percentage values must be quoted! The **NOSHADE** attribute is used to indicate that the rule should not get its usual shaded appearance, but instead should be drawn as a thick line.

Appearance:	<code><HR></code>
Attributes:	<code>ALIGN=left right center, NOSHADE, SIZE=n, WIDTH=n p%</code>

29. HTML: The HTML tag is the outermost tag.

Appearance:	<code><HTML> </HTML></code>
Attributes:	<code>VERSION=string</code>

30. IMG: The IMG tag is used to insert images within text. These are often called "inline" images. The location of the image file should be specified in the **SRC** attribute. It can be a relative or an absolute URL. When the image cannot be displayed for whatever reason, the browser should display the **ALT** text, if specified, instead. The **WIDTH** and **HEIGHT** attributes should contain the image's dimensions.

Appearance:	<code></code>
Attributes:	<code>SRC=URL, ALT=string, ALIGN=left right top middle bottom, HEIGHT=n, WIDTH=n, BORDER=n, HSPACE=n, VSPACE=n, USEMAP=URL, ISMAP</code>

ALIGN controls the alignment of the image with respect to the text. **VSPACE** and **HSPACE** get a numeric value, indicating the number of pixels that should be left free around the image. **VSPACE** covers vertical spacing and **HSPACE** the horizontal spacing.

The **BORDER** attribute is used when the image is a link. It indicates that the browser should draw a border of the indicated size around the image to show that it is a link. It's most often used as **BORDER=0** to turn it off. This has the disadvantage that the image must make it very clear that it's a hyperlink.

ISMAP and USEMAP are used for imagemaps. The ISMAP attribute specifies that the link that this image is in, goes to an imagemap program on the server, so the browser can send the coordinates of the selected location to the server. USEMAP is used for a so-called client-side imagemap. It specifies the URL of the imagemap information.

31. INPUT: The INPUT tag is probably the most useful tag inside forms. It can generate buttons, input fields and checkboxes. In all cases, the NAME attribute must be set.

Appearance:	<INPUT TYPE=x NAME=y>
Attributes:	TYPE=text password checkbox radio submit reset file hidden image, NAME=string, VALUE=string, CHECKED, SIZE=n, MAXLENGTH=n, SRC=URL, ALIGN=top middle bottom left right

TYPE=text

This generates an input field, where the user can enter up to MAXLENGTH characters. The SIZE attribute lists the length of the input field (if the user enters more characters, the text will scroll). The VALUE attribute specifies the initial value for the input field.

TYPE=password

Same as TYPE=text, but the text will be hidden by "*" or similar characters.

TYPE=checkbox

Produces a checkbox. It has two states, on and off. If it is on when the form is submitted, it will be sent as "name=on", otherwise it is ignored altogether. If you use CHECKED, it will come up checked (selected) initially.

TYPE=radio

Produces a radio button. A radio button always exists in a group. All members of this group should have the same NAME attribute, and different VALUES. The VALUE of the selected radio button will be sent to the server. You must specify CHECKED on exactly one radio button, which then will come up selected initially.

TYPE=submit

Produces a button, which when pressed sends the contents of the form to the server. You can have more than one submit button in the form. Each should have a different NAME. The name and value of the pressed button will be sent to the server as well. The value of the VALUE attribute is typically used as text on the submit button.

TYPE=reset

Produces a button, which will restore the form to its original state if pressed. The value of the VALUE attribute is typically used as text on the reset button.

TYPE=file

Allows the user to upload a file. It is still very new, so it is not very widely supported. It is typically presented as an input box with a button to start browsing the local hard disk. This way, a user can specify one or more filename(s) to upload.

TYPE=hidden

Allows to embed information in the form which you do not want to change. This can be useful if the document is generated by a script and you need to store state information. NAME and VALUE of this input field will be sent to the server without modifications.

TYPE=image

Functions similar to a submit button, but uses an image instead. The ALIGN attribute controls the alignment of the image. The coordinates of the selected region will also be sent to the server, in the form of "NAME.x=n&NAME.y=n". A text browser will treat it identical to a normal submit button.

32. ISINDEX: The ISINDEX tag was used before FORMs became more popular. When inserted in a document, it will allow the user to enter keywords that are then sent to the server. The server then executes a search and returns the results. The PROMPT attribute can be used to override the default text in the dialog box.

Appearance:	<ISINDEX>
Attributes:	PROMPT=string.

33. I: I is used to indicate that the enclosed text must be rendered in a italic (slanted) typeface.

Appearance:	<I> </I>
Attributes:	None.

34. KBD: KBD is used to indicate text that should be entered by the user. It differs from CODE in that CODE indicates *code fragments* and KBD indicates *input*.

Appearance:	<KBD> </KBD>
Attributes:	None.

35. LINK: LINK is used to indicate relationships between documents. There are two possible relationships: REL indicates a normal relationship to the document specified in the URL. REV indicates a reverse relationship. The TITLE attribute can be used to suggest a title for the referenced URL or relation.

Appearance:	<LINK REL=string HREF=URL>
Attributes:	REL=string, REV=string, HREF=URL, TITLE=string

36. LI: The LI element is used to mark list items within a list. When the list used is OL, ordered list, the LI element will be rendered with a number. The appearance of that number can be controlled with the TYPE attribute. Similarly, inside an unordered list UL the type of bullet that is displayed can be controlled with the TYPE attribute. For ordered lists, you can also reset the sequence with the VALUE attribute.

Appearance:	 []
Attributes:	TYPE=disclsquare circle when in UL, TYPE=1 a A i I when in OL, VALUE=n

37. MAP: When you use a client-side imagemap, the information on the "hot spots" (clickable areas) in the image is defined here. Every selectable area should be mentioned in an AREA tag inside the MAP tag. The NAME attribute on the MAP tag assigns a name to the imagemap. The URL for the client-side imagemap should point to this.

Appearance:	<code><MAP NAME=string> </MAP></code>
Attributes:	<code>NAME=string</code>

38. **MENU:** The MENU item produces a list like UL, but it should be rendered more compact. Not all browsers make this distinction, and some render it without a bullet at all.

Appearance:	<code><MENU> </MENU></code>
Attributes:	<code>COMPACT</code>

39. **META:** The META tag is used to convey meta-information about the document, but can also be used to specify headers for the document. You can use either HTTP-EQUIV or NAME to name the meta-information, but CONTENT must be used in both cases. By using HTTP-EQUIV, a server *should* use the name indicated as a header, with the specified CONTENT as its value.

Appearance:	<code><META NAME=string CONTENT=string></code>
Attributes:	<code>HTTP-EQUIV=string NAME=string, CONTENT=string</code>

Popular uses for META include:

`<META NAME="generator" CONTENT="Some program">`

This indicates that same program is used to generate this document. It is often the name of the HTML editor used.

`<META NAME="author" CONTENT="Name">`

This indicates the name of the author.

`<META NAME="keywords" CONTENT="keyword keyword keyword">`

Provides keywords for search engines such as Infoseek or Alta Vista. These are added to the keywords found in the document itself. If you insert a keyword more than seven times here, the whole tag will be ignored!

`<META NAME="description" CONTENT="This is a site">`

Search engines which support the above tag will now display the text you specify here, rather than the first few lines of text from the actual document when the document shows up in a search result. You have about 1,000 characters for your description, but not all these will be used.

`<META HTTP-EQUIV="refresh" CONTENT="n; URL=http://foo.bar/">`

This is a so-called "meta refresh", which on certain browsers causes the document mentioned in the URL to be loaded after *n* seconds. This can be used for slide shows or for often-changing information, but has some drawbacks. In particular, if you use a value of zero seconds, the user can no longer go "Back" with his back button. He will be transferred to the specified URL, and when he presses "back" there, he will go back to the document with the refresh, which immediately redirects him to the document he tried to get away from.

`<META HTTP-EQUIV="expires" CONTENT="Tue, 20 Aug 1996 14:25:27 GMT">`

This indicates that the document containing this META tag will expire at this date. If the document is requested after this date, the browser should load a new copy from the server, instead of using the copy in its cache.

40. OL: The OL tag marks up an ordered list of items. Each item should be marked up with a LI, and it will be displayed with a number in front of it. The appearance of the number can be controlled with the TYPE attribute:

- ❖ 1 - Arabic numbers (default) (1, 2, 3, 4, ...)
- ❖ a - Alphanumeric, lowercase (a, b, c, d, ...)
- ❖ A - Alphanumeric, uppercase (A, B, C, D, ...)
- ❖ i - Roman numbers, lowercase (i, ii, iii, iv, ...)
- ❖ I - Roman numbers, uppercase (I, II, III, IV, ...)

Appearance:	
Attributes:	TYPE=1 a A i I, START=n, COMPACT

The START attribute indicates where the list should start. The COMPACT attribute indicates that the list contains only short list items, so it may be rendered in a more compact way.

41. OPTION: The OPTION tag is used inside a SELECT selection list to indicate an option. You may not use markup in an option. All options listed will be displayed in a list or drop-down box, and the user can select one or more options from the list. The VALUE of each option should be unique. If you specify the SELECTED attribute, this option will appear selected when the form comes up initially.

Appearance:	<OPTION> </OPTION>
Attributes:	VALUE=string, SELECTED

42. PARAM: The PARAM element is used to provide "command-line" arguments to a Java applet, which is embedded in a document with the APPLET element. It has two attributes: NAME specifies the name of the argument, and VALUE specifies the value for this argument.

Appearance:	<PARAM NAME=x VALUE=y>
Attributes:	NAME=string VALUE=string

43. PRE - Preformatted text: PRE is used to include sections of text in which formatting is critical. Unlike in the other HTML containers, text in a PRE pair will only be wrapped at the linebreaks in the source, and spaces will not be collapsed. You can even use tabs, although it is better to use multiple spaces since those will always be the right number. Text inside this tag will be displayed in a mono-spaced font to retain the formatting. This is the reason you cannot include font-changing tags inside PRE text. The optional WIDTH attribute can be used to indicate how wide the text is.

Appearance:	<PRE> </PRE>
Attributes:	WIDTH=n

44. P: The P tag is used to indicate paragraphs. The optional attribute ALIGN indicates the preferred alignment for the contents of this paragraph. ALIGN=LEFT is the default.

Appearance:	<P> </P>
Attributes:	ALIGN=left center right

45. SAMP – Sample text: SAMP is used to indicate a sample of text which should be used literally. It differs from KBD text in that KBD text indicates the text that the user must *enter*, whereas SAMP text can also be output.

Appearance:	<SAMP> </SAMP>
Attributes:	None.

46. SCRIPT: The SCRIPT tag is included only to ensure upward compatibility. Newer versions of HTML will include support for inline scripts, which should be contained in this tag. The tag should contain a valid script.

Appearance:	<SCRIPT> </SCRIPT>
Attributes:	None.

47. SELECT: The SELECT tag is used inside forms to generate a list of items from which the user can select one or more item. Each item is listed in an OPTION tag. The value of the selected OPTION tag is assigned to the NAME of the SELECT tag, and both are sent to the server when the form is submitted.

Appearance:	<SELECT NAME=string> </SELECT>
Attributes:	NAME=string, SIZE=n, MULTIPLE

The SIZE attribute indicates how many items are visible at once.

48. SMALL: A browser should draw the enclosed text in a smaller font if available, and ignore the tag otherwise.

Appearance:	<SMALL> </SMALL>
Attributes:	None.

49. STRIKE: The STRIKE tag specifies that the enclosed text should be rendered in a strike-through appearance. Usually this is done with a line through the middle of the text.

Appearance:	<STRIKE> </STRIKE>
Attributes:	None.

50. STRONG: STRONG is used to indicate strongly emphasized text.

Appearance:	
Attributes:	None.

51. STYLE: The STYLE tag is included only to ensure upward compatibility.

Appearance:	<STYLE> </STYLE>
Attributes:	TYPE=string

52. SUB: SUB specifies that the enclosed text should be rendered in subscript, with the enclosed text slightly lower than the surrounding text. This is mainly used for chemical formulas.

Appearance:	<code><SUB> </SUB></code>
Attributes:	None.

53. SUP: SUP specifies that the enclosed text should be rendered in superscript, with the enclosed text slightly higher than the surrounding text. This is also used for mathematical formulas.

Appearance:	<code><SUP> </SUP></code>
Attributes:	None.

54. TABLE: Table start with an optional caption followed by one or more rows. Each row consists of one or more cells, which can be either header or data cells. Cells can overlap across rows and columns.

Appearance:	<code><TABLE> </TABLE></code>
Attributes:	<code>ALIGN=left center right, WIDTH=n p%, BORDER=n, CELLSPACING=n, CELLPADDING=n</code>

The ALIGN attribute controls the alignment of the table itself, but not of the individual cells. This can be set either in the TR element for an entire row, or in the TD and TH elements for individual cells. The WIDTH attribute can be a pixel width or a percentage. The width of the border around the table is set with the BORDER attribute.

Whitespaces inside a table can also be increased. The CELLPADDING attribute indicates how many pixels should be there between a cell's contents and the border. CELLSPACING indicates how much whitespace (in pixels) should be there between individual cells.

55. TD: The TD tag is used to mark up individual cells inside a table row. It may contain almost all tags, including nested tables. If the cell is a label of some sort, use TH instead of TD. The NOWRAP attribute indicates the contents of the current cell should not be wrapped. You must use BR in the cell to force line breaks to prevent the entire cell from showing up as just one line.

Appearance:	<code><TD> </TD></code>
Attributes:	<code>ROWSPAN=n, COLSPAN=n, NOWRAP, ALIGN=left right center, VALIGN=top middle bottom, WIDTH=n, HEIGHT=n</code>

The ROWSPAN and COLSPAN attributes indicate how many rows or columns this cell overlaps. The ALIGN and VALIGN attributes control the horizontal and vertical alignment of the current cell. ALIGN can be set for left, right or centered cells. VALIGN indicates that the table cell's contents should appear at the top, the middle or the bottom of the row. The WIDTH and HEIGHT attributes can be used to suggest a width and height for this cell. This value should be in pixels.

56. TEXTAREA: The TEXTAREA tag, used inside FORMs, sets up an area in which the user can type text. This text will be sent to the server when the form is submitted. The NAME attribute assigns the text area a name, used by the script that processes the form. ROWS and COLS are used to specify the height and width of the text area, in number of characters.

Appearance:	<code><TEXTAREA NAME=string, ROWS=n, COLS=n> <TEXTAREA></code>
Attributes:	<code>NAME=string, ROWS=n, COLS=n</code>

57. **TH:** Like the TD tag, a TH is used for a table cell. However, TH should be used when the cell's content is a heading of some sort, for example, when the text is a label for a row of column.

Appearance:	<code><TH> </TH></code>
Attributes:	<code>ROWSPAN=n, COLSPAN=n, NOWRAP, ALIGN=left right center, VALIGN=top middle bottom, WIDTH=n, HEIGHT=n</code>

58. **TITLE:** Each document must have exactly one TITLE element. This element provides the title of the document. It is usually displayed at the top of the browser's window, but also used to label a bookmark entry for the document and as a caption in search engine results. It may only contain text and entities, but no markup.

Appearance:	<code><TITLE> </TITLE></code>
Attributes:	None.

59. **TR:** HTML tables are constructed as a sequence of rows. Each row of table cells should be enclosed in a TR tag. The end tag is optional, since it is usually obvious to see where a row ends - where the new row begins, or where the entire table ends.

Appearance:	<code><TR> </TR></code>
Attributes:	<code>ALIGN=left right center, VALIGN=top middle bottom</code>

The ALIGN and VALIGN attributes control the horizontal and vertical alignment of the entire row. ALIGN can be set for left, right or centered cells. VALIGN indicates that the table cell's contents should appear at the top, the middle or the bottom of the row.

60. **TT:** The TT tag specifies that the enclosed text should be rendered in a teletype (mono-spaced) font.

Appearance:	<code><TT> </TT></code>
Attributes:	None.

61. **UL:** The UL element indicates an unordered list. Every item in a list is marked with LI, and usually appears with a bullet of some sort in front of it. The type of bullet can be suggested with the TYPE attribute. It specifies the type of bullet that can be used with the UL element. There are *three* values that can be defined with UL tag. They are:

- ❖ DISC: Specifies a solid round bullet.
- ❖ CIRCLE: Specifies a hollow round bullet.
- ❖ SQUARE: Specifies a filled square bullet.

Appearance:	<code> </code>
Attributes:	<code>TYPE=disc square circle, COMPACT</code>

The default value for UL tag is a filled disc. The COMPACT attribute is used to indicate that the list items are short, so the browser can render the list more compact. For example, it could put more than one item on a line.

62. **U:** U is used to indicate the enclosed text should be underlined.

Appearance:	<U> </U>
Attributes:	None.

63. **VAR:** VAR is used to mark up variables, for example in discussions of computer programs. Using this tag allows programs to automatically generate lists of the used variables. Example: "The variable <VAR>c</VAR> is used as a counter in this program."

Appearance:	<VAR> </VAR>
Attributes:	None.

7.4.2. Special Characters

Special characters such as <, > etc. can be included in a web page using escape codes which begin with the ampersand (&) symbol. The ampersand symbol must be followed by the mnemonic keyword for the symbol. Instead of the mnemonic code, the ASCII code of the symbol may also be given in the HTML code.

Table 7.1. Special Characters with their Symbol, Description and Decimal Value.

S.No.	Mnemonic	Symbol	Description	Decimal
1	Lt	<	Less than	#60
2	Gt	>	Greater than	#62
3	Amp	&	Ampersand	#38
4	AE lig	Æ	Capital AE diphthong	#198
5	O Slash	ô	Capital oh slash	#216
6		£	Pound sign	#163
7		©	Copyright sign	#169
8		±	Plus or minus	#177
9		¼	Quarter	#188
10		½	Half	#189
11		¾	Three-quarters	#190

7.4.3. Example HTML Code

Example 1 :

```
<html>
<head>
  <title>
    specifying the use of tags
  </title>
<body bgcolor="77aabd" text="blue">
  <br><br>
  <hr size="5"width="500"align="center"color="red"noshade></hr>
  <center>
  <br>
```

Save the code as tags.htm

Run the file tags.htm and the following output is displayed.

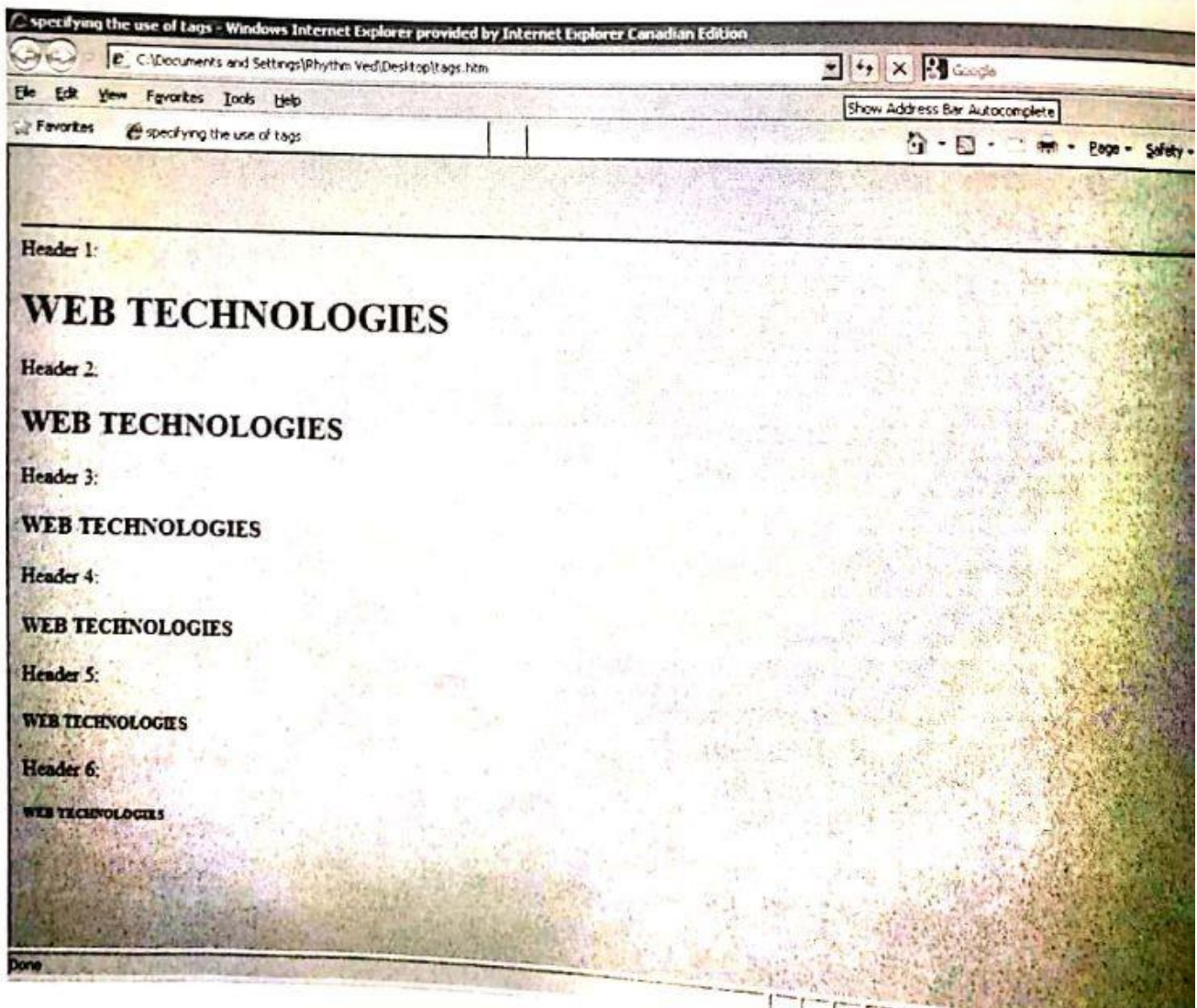


Example 2 :

```

<html>
  <head>
    <title>
      specifying the use of tags
    </title>
  <body bgcolor="orange">
    <br><br>
    <hr color="blue" ></hr>
    Header 1:<h1>WEB TECHNOLOGIES</h1>
    Header 2:<h2>WEB TECHNOLOGIES</h2>
    Header 3:<h3>WEB TECHNOLOGIES</h3>
    Header 4:<h4>WEB TECHNOLOGIES</h4>
    Header 5:<h5>WEB TECHNOLOGIES</h5>
    Header 6:<h6>WEB TECHNOLOGIES</h6>
  </body>
</html>

```

Output

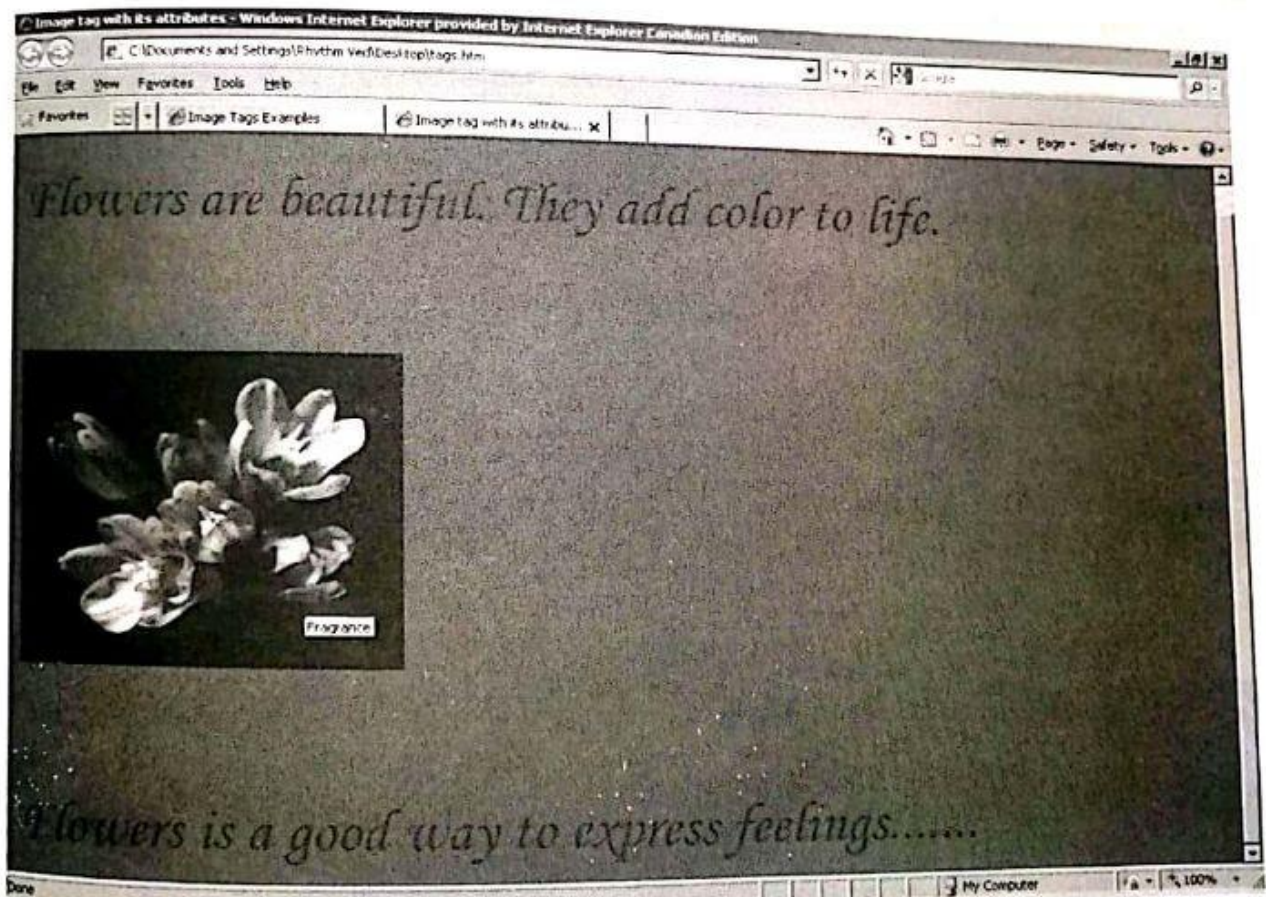
Example 3 :

```

<html>
  <head>
    <title>
      Image tag with its attributes
    </title>
  </head>
  <body bgcolor="ab4faf">
    <br><br>
    <font face="monotype corsiva" color="maroon" size="18">Flowers are beautiful. They add
color to life.</font><br>
    <IMG border="4" src="flower.jpg" height="250" width="300" alt="Fragrance"
vspace="100"><br>
    <font face="monotype corsiva" color="maroon" size="18">Flowers is a good way to express
feelings.....</font>
  </body>
</html>

```

Output



Example 4 :

```

<html>
  <head>
    <title>
      Example_of_Unordered List
    </title>

```

```

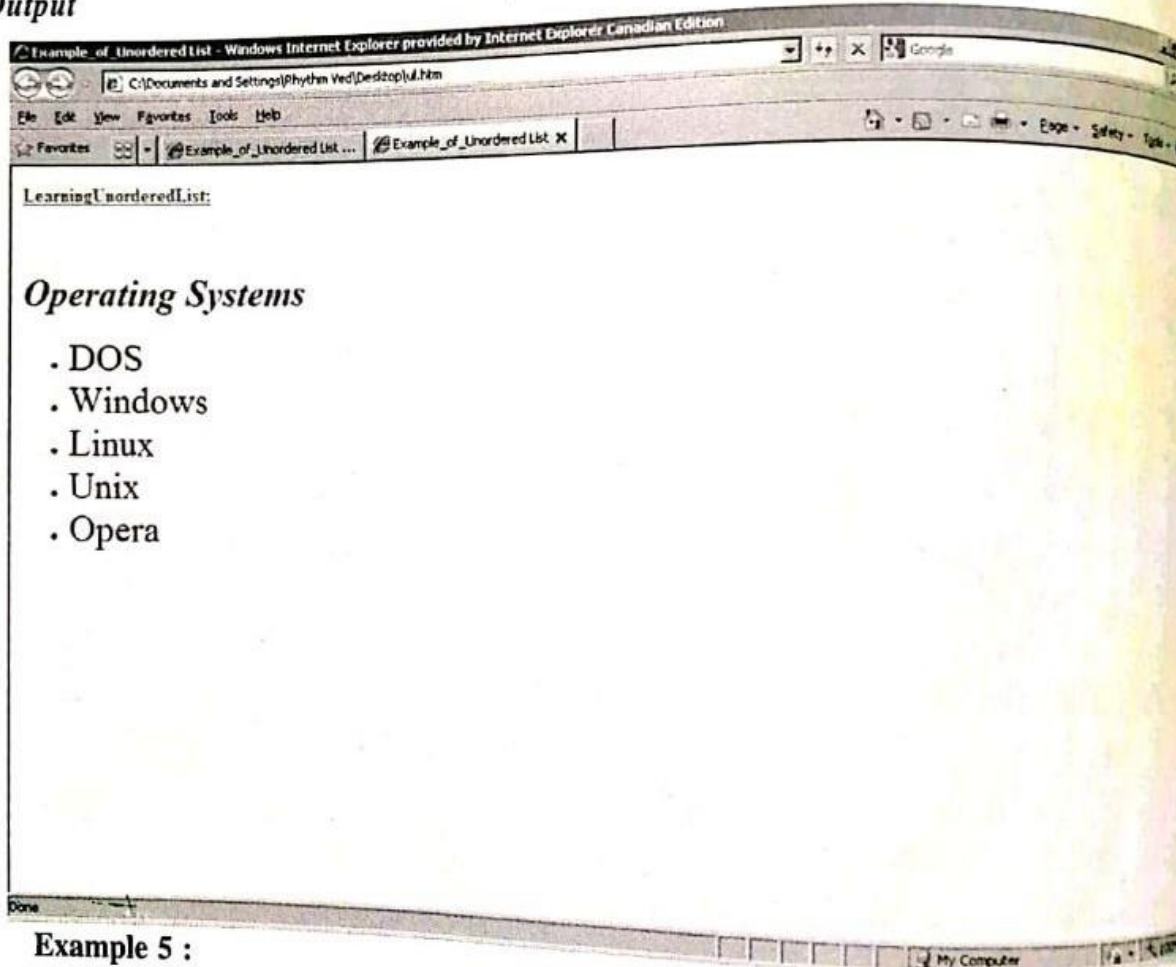
</head>
<body>
  <font size=6>
    <h4><b><u><font
      color="red">LearningUnorderedList: </font></u></b></h4>
    <br>
    <b><i><font color="blue">Operating Systems</font></i></b>

    <ul>
      <li>DOS
      <li>Windows
      <li>Linux
      <li>Unix
      <li>Opera

    </ul>
  </font> </body>
</html>

```

Output



Example 5 :

```

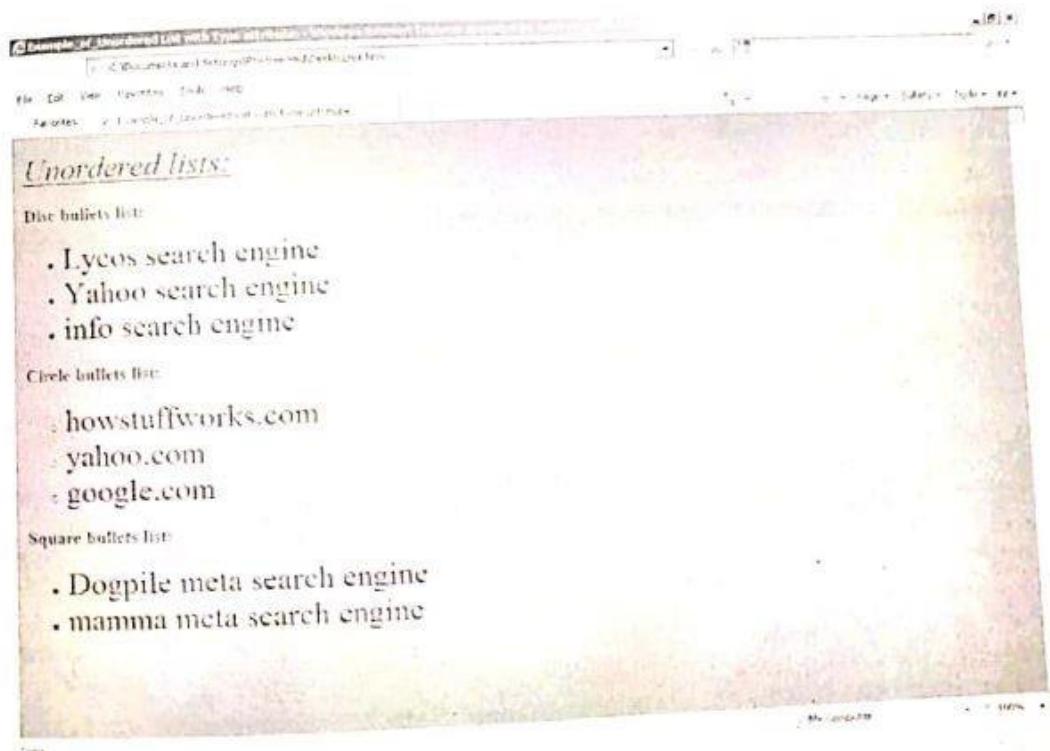
<html>
  <head>
    <title>
      Example_of_Unordered List with Type attribute:
    </title>

```

```

</head>
<body bgcolor="green">
<font size=6>
<font color="blue"> <u><i>Unordered lists:</i></u>
<h3>Disc bullets list:</h3></font>
<ul>
<li>Lycos search engine
<li>Yahoo search engine
<li>info search engine
</ul>
<h3><font color="blue">Circle bullets list:</font></h3>
<ul type="circle">
<li>howstuffworks.com
<li>yahoo.com
<li>google.com
</ul>
<h3><font color="blue">Square bullets list:</font></h3>
<ul type="square">
<li>Dogpile meta search engine
<li>mamma meta search engine
</ul>
</font>
</body>
</html>

```

Output**Example 6 :**

```

<html>
<head>
<title>
Example_of_Ordered List with the use of Type attribute
</title>

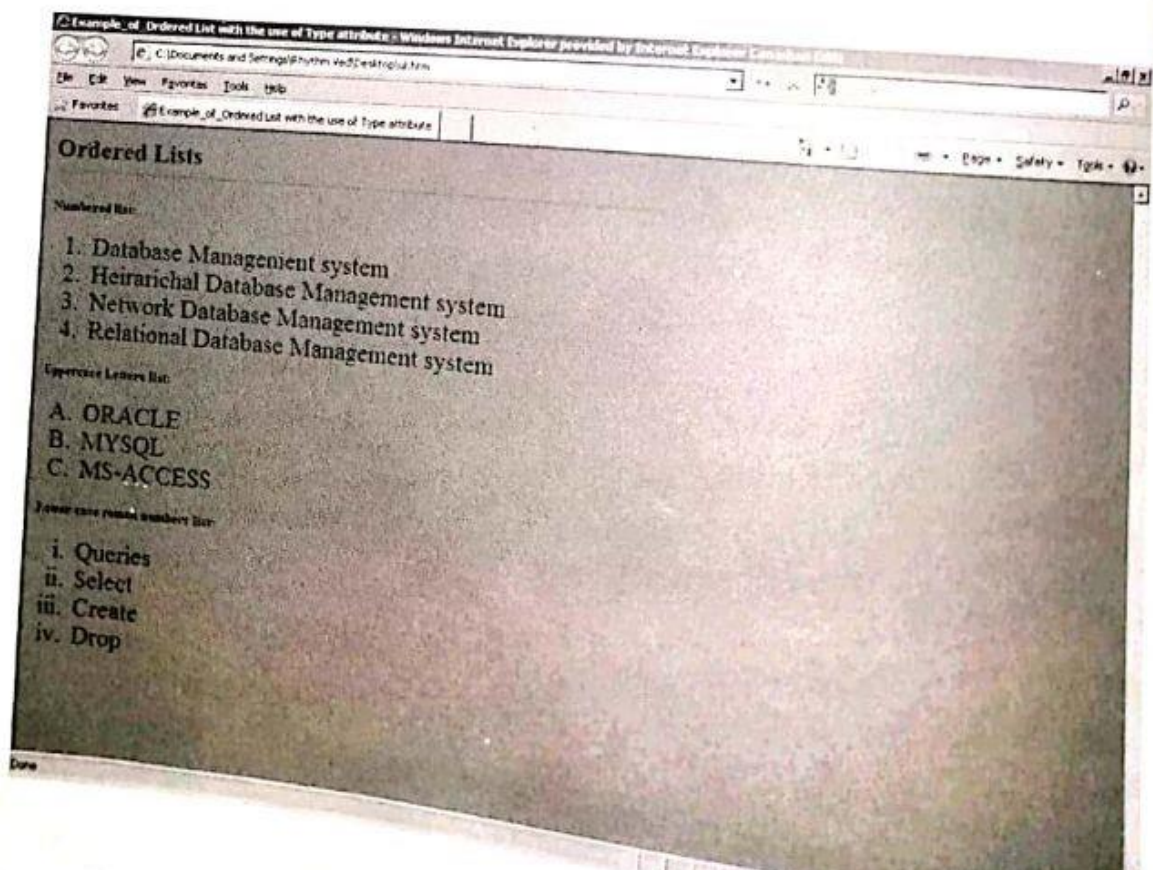
```

```

</head>
<body bgcolor="ab9fed">
<font size=5>
  <b> Ordered Lists</b>
  <HR ALIGN="Left" SIZE="2" WIDTH="60%" NOSHADE>
<h5>Numbered list:</h5>
  <ol>
    <li>Database Management system
    <li>Heirarichal Database Management system
    <li>Network Database Management system
    <li>Relational Database Management system
  </ol>
<h5>Uppercase Letters list:</h5>
  <ol type="A" >
    <li>ORACLE
    <li>MYSQL
    <li>MS-ACCESS
  </ol>
<h5>Lower case roman numbers list:</h5>
  <ol type="i">
    <li>Queries
    <li>Select
    <li>Create
    <li>Drop
  </ol>
</font>
</body>
</html>

```

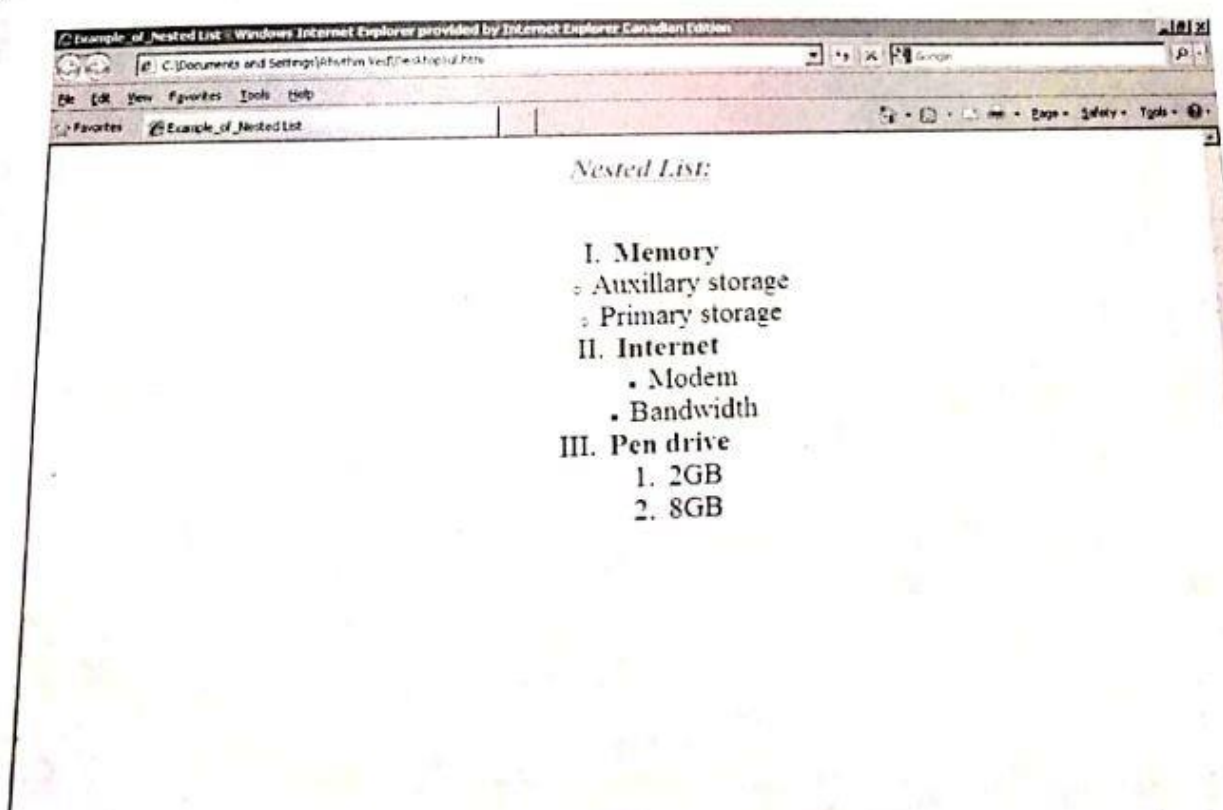
Output



Example 7 :

```
<html>
<head>
  <title>
    Example_of_Nested List
  </title>
</head>
<body>
  <font size=5>
    <center>
      <h2><u><b><i><font color="red">Nested List: </font></i></b></u>
      </h2> <br>
      <ol type="I">
        <li><b><font color="blue">Memory</font></b>
          <ul>
            <li>Auxillary storage
            <li>Primary storage
          </ul>
        <li><b><font color="blue">Internet</font></b>
          <ul type="square">
            <li>Modem
            <li>Bandwidth
          </ul>
        <li><b><font color="blue">Pen drive</font></b>
          <ol>
            <li>2GB
            <li>8GB
          </ol>
        </li>
      </ol>
    </center>
  </font>
</body>
</html>
```

Output

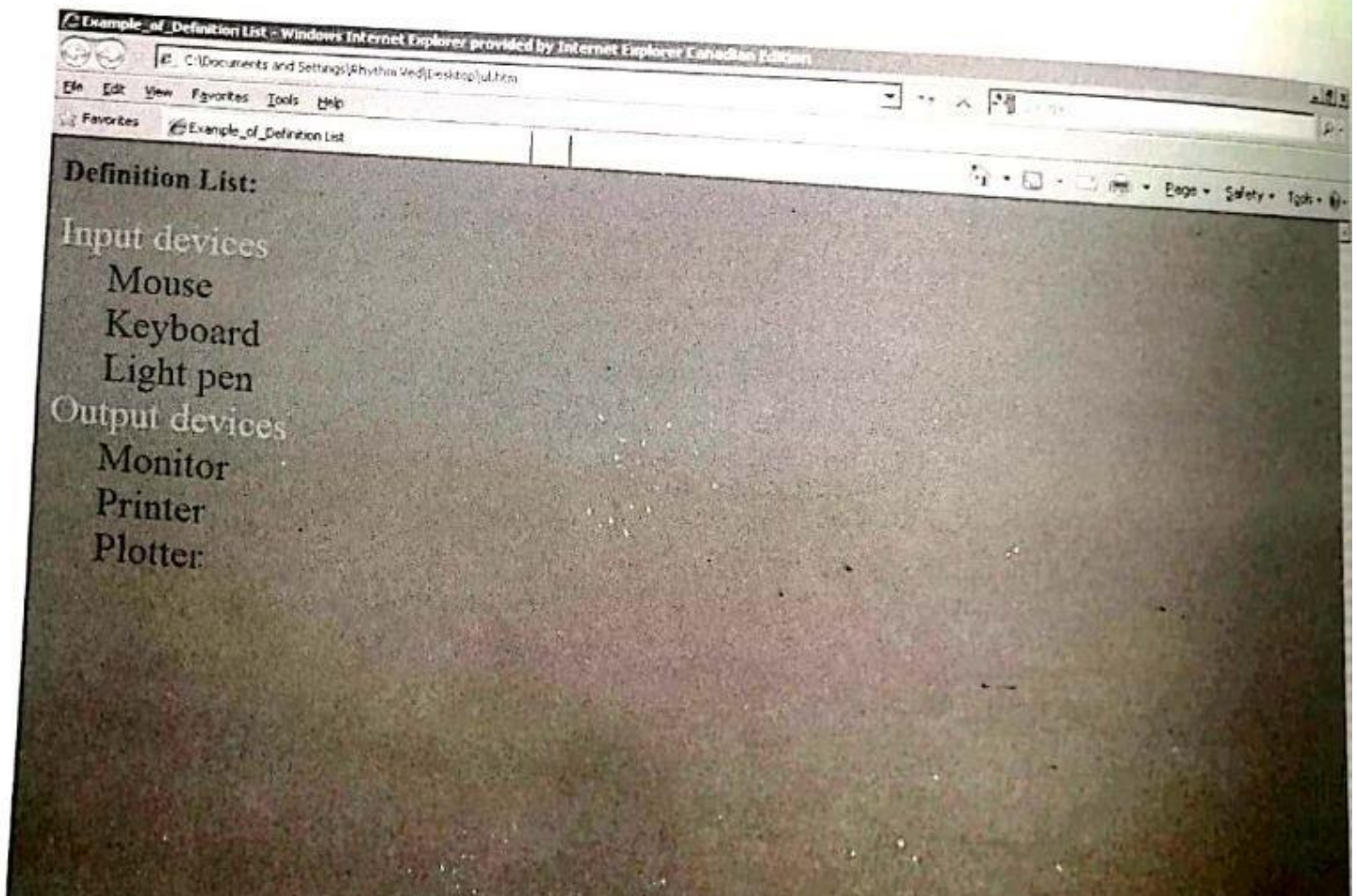


Example 8 :

```

<html>
  <head>
    <title>
      Example_of_Definition List
    </title>
  </head>
  <body bgcolor="0088ab">
    <h2>Definition List:</h2>
    <font size=6>
      <dl>
        <dt><font color="orange">Input devices</font></dt>
        <dd>Mouse</dd>
        <dd>Keyboard</dd>
        <dd>Light pen</dd>
      <br>
        <dt><font color="orange">Output devices</font></dt>
        <dd>Monitor</dd>
        <dd>Printer</dd>
        <dd>Plotter</dd>
      </dl>
    </font>
  </body>
</html>

```

Output

7.5. HTML ANCHOR TAG

A website consists of various pages that give access to various pages that are linked to each other. This is the most powerful feature of WWW known as **hyperlink**. In HTML, **anchor tag** (`<a>....</a>`) is used to mark the text as a hypertext link that a user can click to display the document which is anchored to it. It is a container tag. This tag can be used in two ways:

- ❖ The text or image that is clicked to activate the link.
- ❖ The address that will be opened using the defined link.

These links are used in the following ways:

- ❖ To jump from one section of the page to another within the same web page.
- ❖ To link to another page within the same website.
- ❖ To link to another page or website anywhere in the world.

7.5.1. Creating Hyperlink to a Document

The hyperlink to a document is created using HREF attribute of anchor tag which marks the beginning of the link to another document, resource or to a particular place in another document. The general syntax is:

```
<A HREF="path of the filename">name of the link</A>
```

where HREF attribute indicates the hypertext reference—path of the file name. This path can be an absolute address or relative address. HREF is also responsible for displaying "name of the link" to be underlined and colored.

Example 9 :

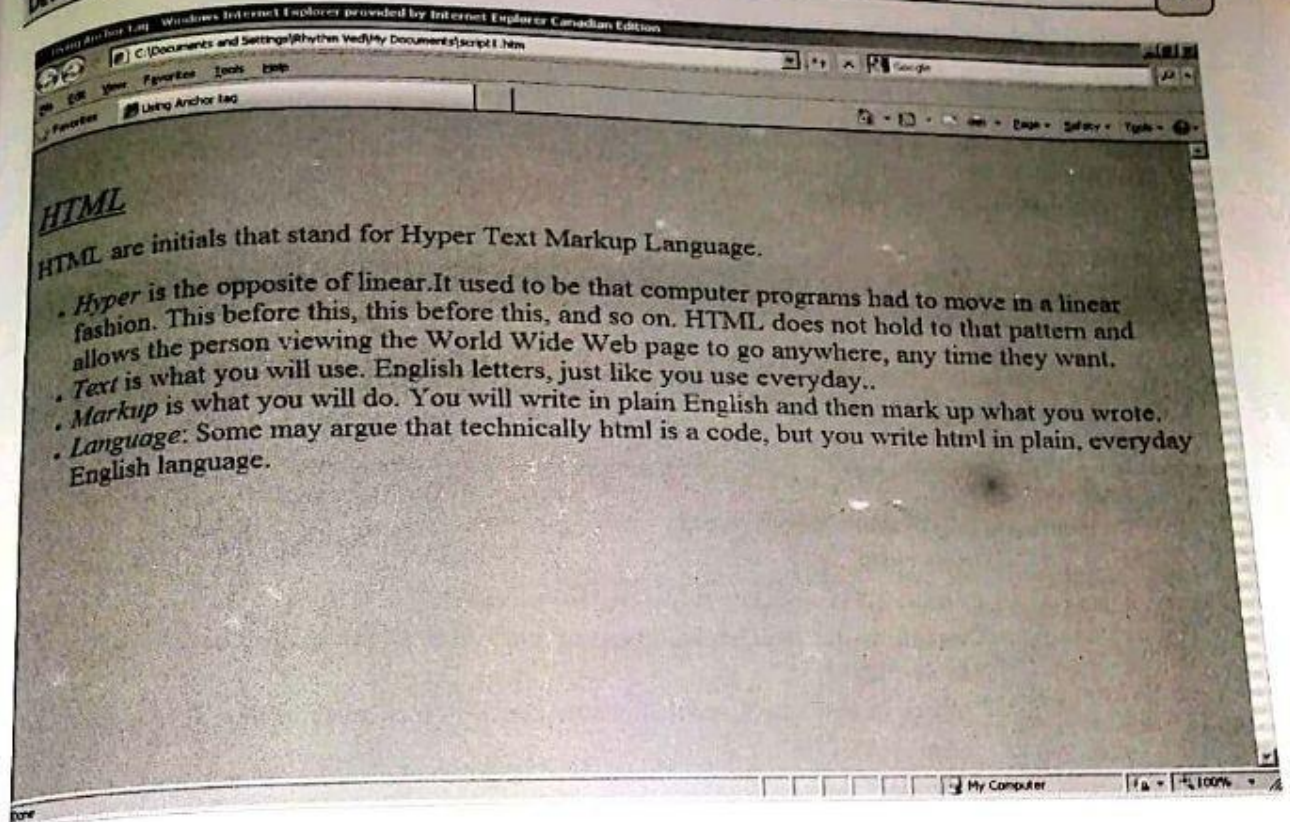
```
<html>
  <head>
    <title>
      Using Anchor tag
    </title>
  </head>
  <body bgcolor="orange">
    <br><br>
    <u><i><font size="6">HTML</font></i></u><br><br> <font color="blue"
    size="5"> HTML are initials that stand for Hyper Text Markup Language.<br>
    <ul>

    <li><i>Hyper</i> is the opposite of linear. It used to be that computer programs
    had to move in a linear fashion. This before this, this before this, and so on.
    HTML does not hold to that pattern and allows the person viewing the World
    Wide Web page to go anywhere, any time they want.

    <li><i>Text</i> is what you will use. English letters, just like you use everyday..

    <li><i>Markup</i> is what you will do. You will write in plain English and then
    mark up what you wrote.

    <li><i>Language</i>: Some may argue that technically html is a code, but you
    write html in plain, everyday English language.
    </body>
</html>
```

7.5.2. Creating a Link within a Web Page

It is possible to create the links to the other parts of the same document. To accomplish this, anchor tag requires two attributes:

- HREF attribute:** It serves as a hyperlink *i.e.*, the command that sends the browser to another section on your web page. The general syntax is:

name of the link n the same document

Where *Linkname* is the name of the section you are linking or jumping to and # sign tells the anchor tag that it is looking for the section within the same document.

- NAME attribute:** It acts as a reference *i.e.*, the place on the page where you want the browser to jump to. The value of the name attribute is an identifier for the anchor and these should be unique. The text that is defined with the name attribute is not highlighted when displayed by the browser. The general syntax is:

Definition

Always remember that: (i) the linkname specified with HREF attribute should be same as the one used with NAME attribute, (ii) The value of the HREF attribute and the value of the NAME attribute should be enclosed with quotation marks otherwise the link will not work, (iii) Anchor tag cannot be nested.

7.5.3. Creating Link to Another Website

If the HTML document to which you want to link, is located on the Internet, you need to specify absolute URL or name of the website.

The general syntax is:

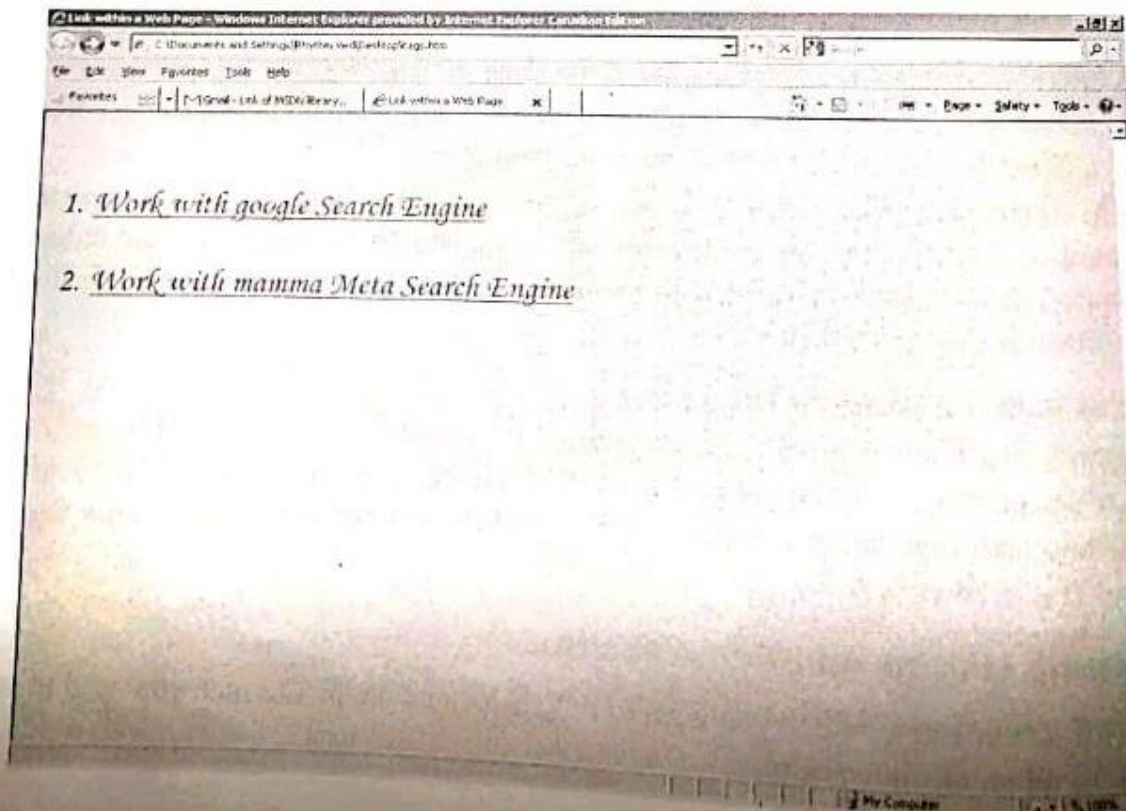
```
<A HREF="http://www.google.com">Search Engine</A>
```

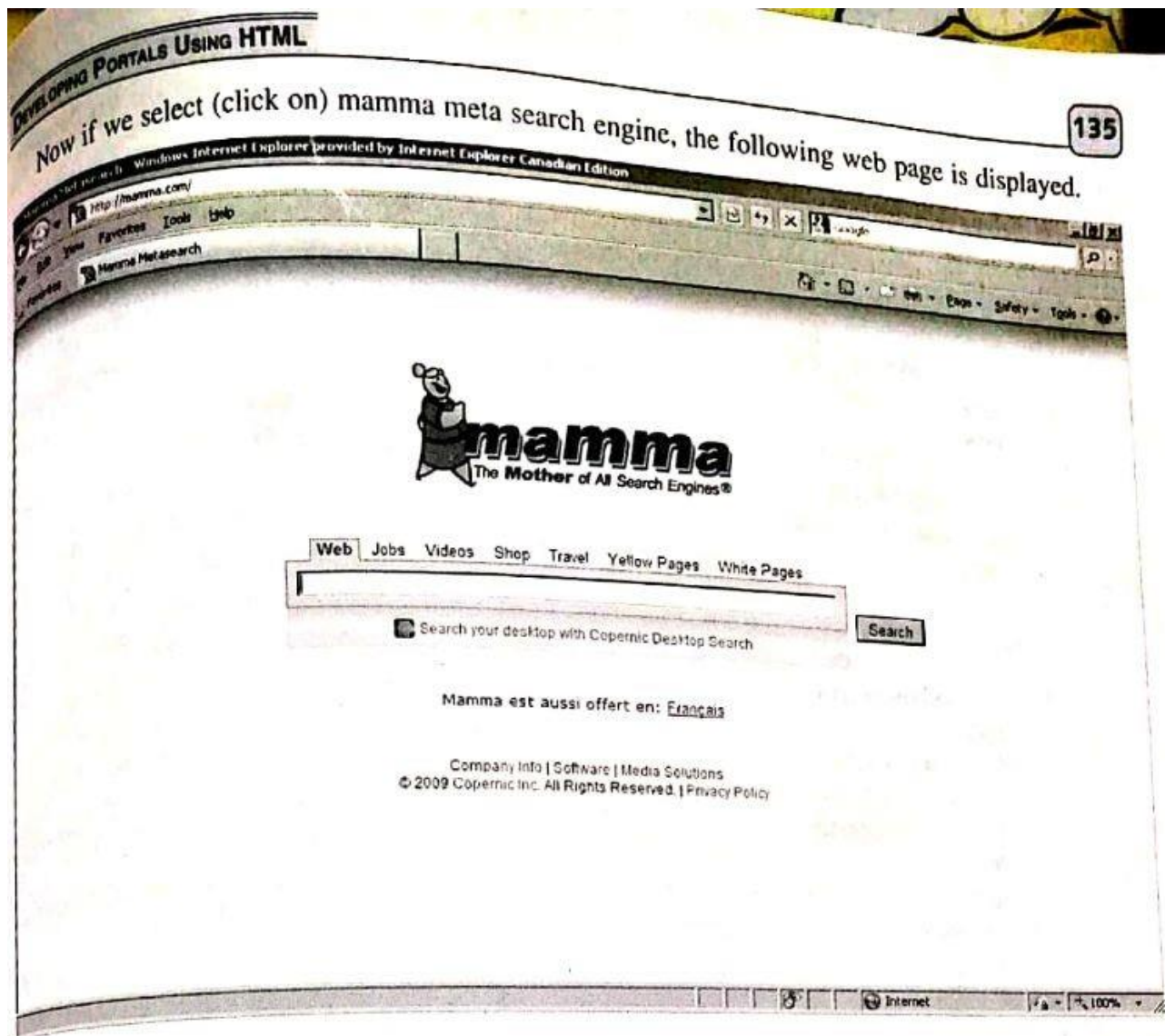
It displays the text "Search Engine" colored and underlined. When this highlighted text is clicked, the control moves to the website "google.com" specified by the value of the attribute HREF, only if the computer is connected to the Internet.

Example 10 :

```
<html>
  <head>
    <title>
      Link within a Web Page
    </title>
  </head>
  <body bgcolor="pink"><br><br><br>
    <ol type="A">
      <font size="6">
        <li><a href="http://www.google.com/">Work with google Search Engine</a></li>
        <li><a href="http://www.mamma.com/">Work with mamma Meta Search Engine</a></li>
      </font>
    </ol>
  </body>
</html>
```

Output





7.6. HTML TABLE TAG

Tables are an effective means of displaying information in a concise and precise form. Instead of writing several pages of explanation, a table can effectively give all necessary information. Tables are defined with the `<table>` tag. A table is divided into rows (with the `<tr>` tag), and each row is divided into data cells (with the `<td>` tag). The letters `td` stands for "table data," which is the content of a data cell. A *data cell* can contain text, images, lists, paragraphs, forms, horizontal rules, tables, etc.

Example 11 :

```
<html>
  <head>
    <title>
      Illustration of Table with caption
    </title>
  </head>
  <body>
    <center>
      <table border="4" bgcolor="yellow">
        <caption><h1><font color="red">Quantity-Cost table</font></h1></caption>
        <tr>
          <th>Quantity</th>
```

```

<th>Cost(in Rs.)</th>
<th>Quality</th>
</tr>
<tr>
<td>10</td>
<td>2500</td>
<td>Best</td>
</tr>
<tr>
<td>20</td>
<td>200</td>
<td>Good</td>
</tr>
<tr>
<td>10</td>
<td>25</td>
<td>Poor</td>
</tr>
<tr>
<td>100</td>
<td>3000</td>
<td>Good</td>
</tr>
</table>
</center>
</body>
<html>

```

Output

Illustration of Table with caption - Windows Internet Explorer provided by Internet Explorer Canadian Edition

C:\Documents and Settings\Rhythm Ved\Desktop\ul.htm

File Edit View Favorites Tools Help

Illustration of Table with caption

Quantity-Cost table

Quantity	Cost(in Rs.)	Quality
10	2500	Best
20	200	Good
10	25	Poor
100	3000	Good

7.7. HTML FRAMES

A framed page consists of two or more HTML documents. Since each document is a separate entity with its own URL, it behaves independently of the others. Thus, scrolling or reloading of one document will not result in equivalent behaviour in the others.

Advantages

1. Frames prove to be an excellent navigational tool if used properly. Thus, one frame can be reserved for the navigational links, while the other may be dedicated for displaying the actual content.
2. Site maintenance is relatively easy with frames. Suppose, you want to add another major section to your website, you need to change only the navigational frame that contains the links.
3. With frames, you can display contents of other websites without letting your users leave your site. For example, you may display your logo and navigation links in one frame and load the pages of other websites in another frame.
4. With `<NOFRAME>` tag you can add alternative content for browsers which do not support the `<FRAME>` tag.
5. In the context of a two framed document with one page containing high quality (and thus large) navigation buttons and the other, the main content, display of content may be faster after the initial downloading. This is because only the frame containing the main content is changed; the navigational frame with its large graphics is not loaded again.

Disadvantages

1. Framed sites are not indexed well by search engines. Furthermore, results of search engine might have your document displayed out of its frameset. You shall thus lose the navigational or other aspects which you so meticulously developed.
2. Users cannot bookmark individual pages. This issue can be solve by first checking the properties of the page in question (right clicking on that frame) and opening it in another browser window. This is all good for the visitor but a nightmare for the developer who loses his navigational frame since the visitor bookmarks only the frame containing the main content.
3. Frames are not supported by older browsers.
4. Designing good-looking sites with continuity amongst various frames of a page is challenging.

Example 12 :

```
<html>
<head>
  <title> HTML Frames Examples </title>
</head>
<body>
  <h1>1 <sup>st</sup> FRAME</h1>
</body>
</html>
```

Save the code as *frame1.htm*

```
<html>
<head>
  <title> HTML Frames Examples </title>
</head>
```

```

<body>
  <h1>2<sup>nd</sup>FRAME</h1>
</body>
</html>

```

Save the code as *frame2.htm*

```

<html>
<head>
  <title> HTML Frames Examples </title>
</head>
<body>
  <h1>3<sup>rd</sup>FRAME</h1>
</body>
</html>

```

Save the code as *frame3.htm*

```

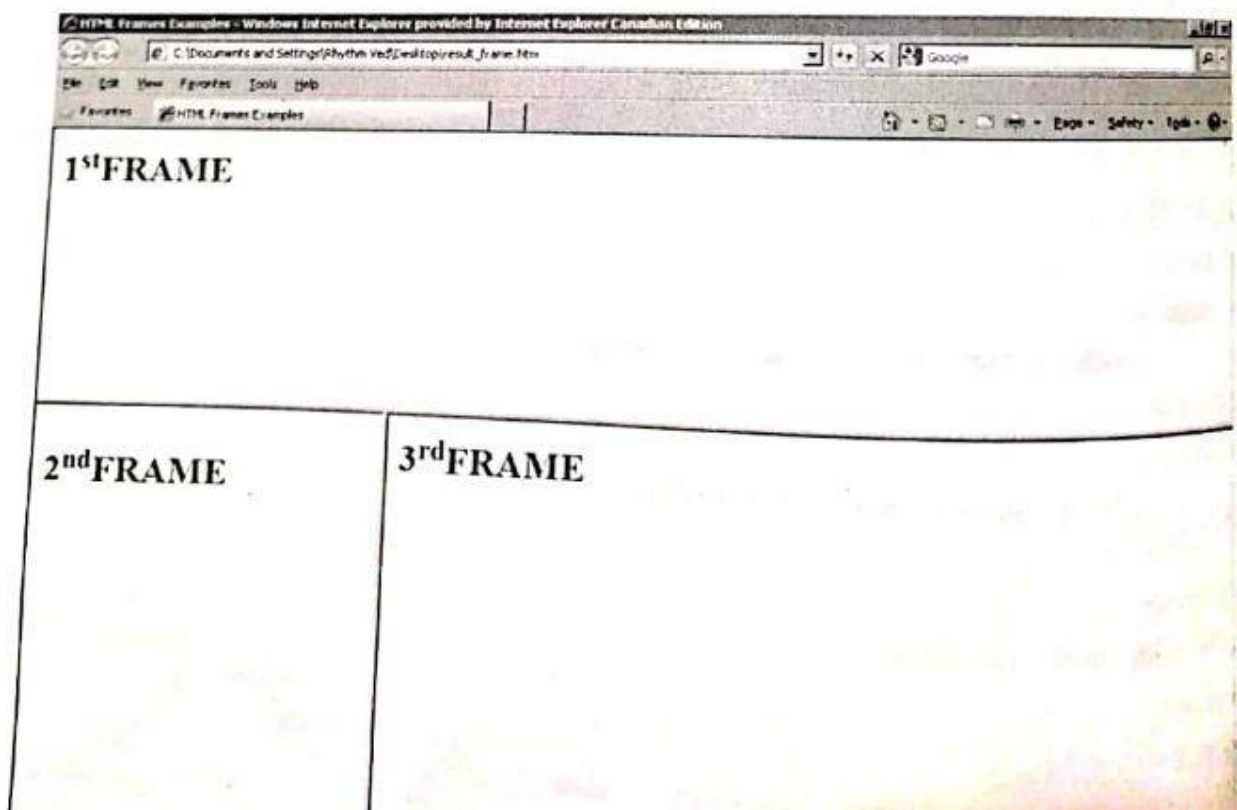
<html>
  <head>
    <title> HTML Frames Examples </title>
  </head>
  <frameset rows="40%,60%">
    <frame src="frame1.htm">
    <frameset cols="30%,*">
      <frame src="frame2.htm">
      <frame src="frame3.htm">
    </frameset>
  </frameset>
</html>

```

Save the code as *result_frame.htm*

Output

When the file *result_frame.htm* is run, it breaks the page into two rows then breaks the second into two columns:



Frames with No border

To make frame border invisible, we simply need to add the following parameters to the frameset cols line.

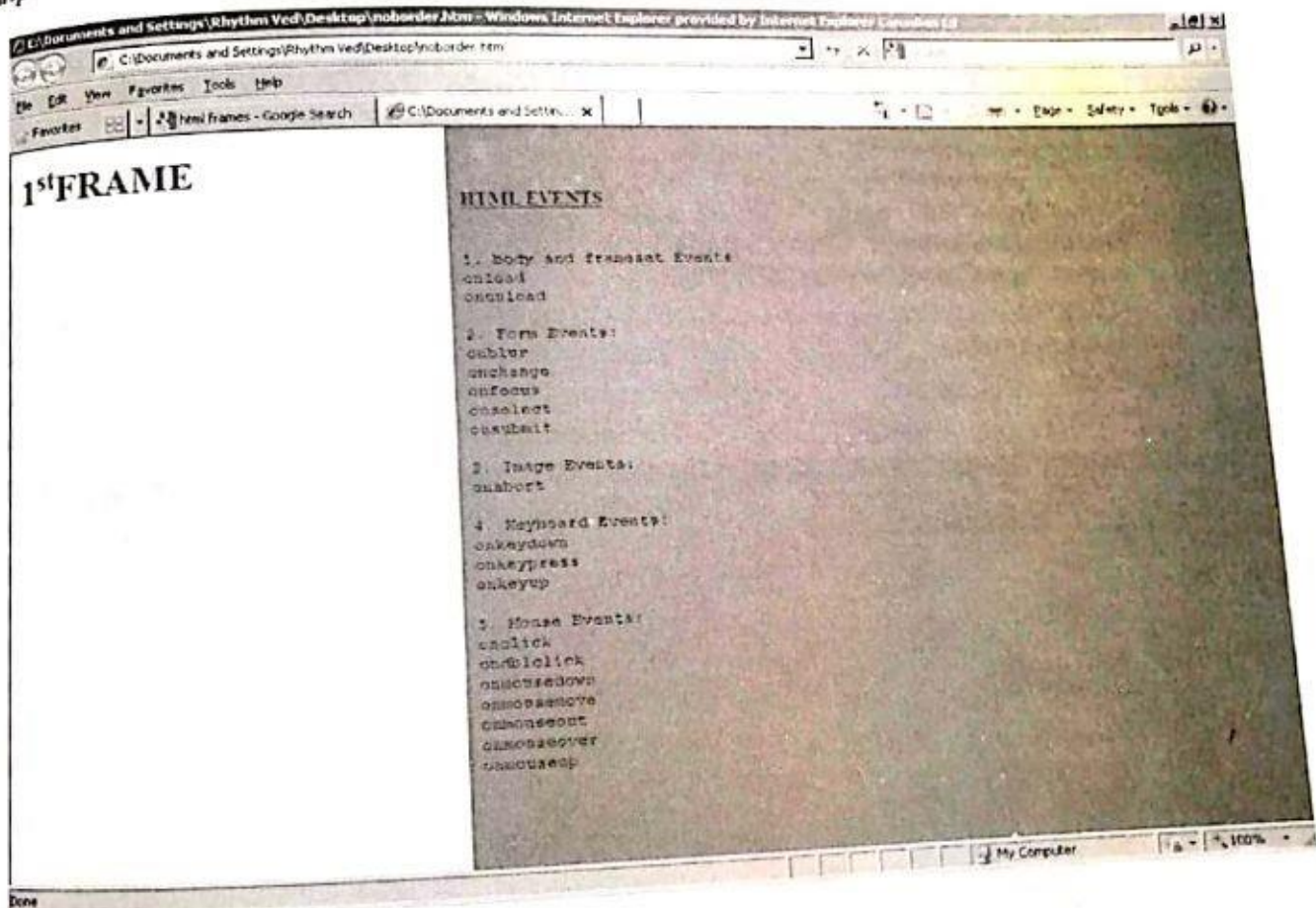
```
frameborder="0" border="0" framespacing="0"
```

Considering the files created earlier i.e., frame1.htm and html_events.htm.

Example 13 :

```
<html>
<frameset cols="35%,*" frameborder="0" border="0" framespacing="0">
<frame src="frame1.htm" name="A" >
<frame src="html_events.htm" name="B" >
</frameset>
</html>
```

Output



7.8. HTML FORMS

To make your website more interactive you can use **HTML forms**. Forms allow your website visitors to enter information into fields which are then submitted in the way you decide. There are many possible ways to handle HTML form; you can have an e-mail sent to you through a form, logins, CGI scripts (Common Gateway Interface), Perl Scripts, sent to an e-mail address, and more. The possibilities are very wide.

Example 14 :

```

<html>
  <head>
    <title>
      Form tag with its attributes
    </title>
  </head>

  <body bgcolor="pink">
    <br>
    <h2><center>Application Form</center></h2>
    <hr size="3" width="200" color="green" noshade>
    <br>
    <br>
    <form>
      First Name<input type="text" name="first"
      size="15" maxlength="20" value="Your first name"><br><br>
      Password<input type="password"
      name="secret" size="15" maxlength="6"
      value=""><br><br>
      <input type="radio" name="gender" value="yes" checked>Male<br>
      <input type="radio" name="gender" value="no">Female<br><br>
      <input type="checkbox" name="percent" >Check if percentage is above 70 <br>
      <br>
      Additional Information<br><textarea name="info" cols="50" rows="10">
      </textarea><br><br>
      Upload your latest resume<br>
      <input type="file" name="Upload" value=""><br><br>
      <input type="submit" value="Submit Form">
      <input type="reset" value="Clear">
    </form>
  </body>
</html>

```

Output

The screenshot shows a web browser window with the title 'Form tag with its attributes - Windows Internet Explorer provided by Internet Explorer Canadian Edition'. The address bar shows a local file path. The form is titled 'Application Form' and contains the following elements:

- First Name:** A text input field with the placeholder text 'Your first name'.
- Password:** A password input field.
- Gender:** Two radio buttons labeled 'Male' (which is selected) and 'Female'.
- Check if percentage is above 70:** A checkbox that is currently unchecked.
- Additional Information:** A large text area for additional information.
- Upload your latest resume:** A file upload section with a 'Browse...' button.
- Form Controls:** 'Submit Form' and 'Clear' buttons at the bottom.

7.9. DISADVANTAGES OF HTML

141

There are many potential problems associated with marking up your data using HTML. Three particularly serious problems come to mind:

1. **The GUI is embedded in the data.** What happens if you decide that you like a table-based presentation better than a list-based presentation? In order to change to a table-based presentation, you must recode your entire HTML! This could mean editing many of pages.
2. **Searching for information in the data is tough.** How would you get a quick list of only the clients in California? Certainly, some type of script would be necessary. But how would that script work? It would probably have to search through the file word for word looking for the string "California". And even if it found matches, it would have no way of knowing that California might have a relationship to "New York"—that they are both states. Forget about the relationships between pieces of data which are crucial to power searching.
3. **The data is tied to the logic and language of HTML.** What happens if you want to present your data in a Java applet? Well, unfortunately, your Java applet would have to parse through the HTML document stripping out tags and reformat the data. Non-HTML processing applications should not be burdened with extraneous work.

7.10. SEPARATING STYLE FROM STRUCTURE WITH STYLE SHEETS

Style sheets are often treated as a means to separate style from content. That's not entirely true depending on your definitions of *style* and *content*. Style sheets allow you to control the rendering e.g., fonts, colors, leading, margins, typefaces, and other aspects of style of a web document without compromising its structure. Style sheets maintain the separation between structural and presentation information, and provide for a graceful fall-back if the specific presentation requested by the style sheet is not available. *Cascading Style Sheets (CSS)* are simple style sheet mechanism that allows authors and readers to attach style to web pages. It uses common desktop publishing terminology that should make it easy for professional as well as untrained designers to make use of its features. Visual design issues, such as page layout, can thus be addressed separately from the web page logical structure.

TIPS

One of the main purposes of CSS is to separate the content from the document's structure. CSS is used to style the content of the document while HTML/XHTML/XML is used to build the structure. That way, once the document is built, any changes to the look of the page can be made in the style sheets. The changes are made once and then take effect throughout the entire site. This makes the site much easier to maintain and, if necessary, change a small part of it or its entire format.

All tags in HTML have a *default* style. That is, the browser has a certain way of rendering each tag type based on some basic rules. This can be overridden via an inline style using the `STYLE` attribute on the tag.

```
<p style="float:right;margin-left:1em;width:40%">Some text.</p>
```

However, this method only affects that particular instance of a tag. Style sheets provide ways to apply styles to all instances of a given tag, or to specific subsets. They allow you to group style settings together rather than embedding them throughout the HTML code. That's where the concept of separation comes in. This also allows styles to be shared by multiple documents.

7.10.1. Including Style Sheets

A style sheet is simply a set of style rules. You can define a style sheet *within* an HTML page using a `<style>` tag in the head of the document (between the `<head>` and `</head>` tags).

```
<style type="text/css">
```

```
...style rules...
```

```
</style>
```

Or, like JavaScript, style sheets can be defined in a separate file and referenced by a URL. An *external style sheet* is included in a page using the LINK tag, again in the head of the document.

```
<link href="url" rel="stylesheet" type="text/css">
```

This is useful for sharing a common style sheet among several pages in a site.

You can combine any number of `<style>` and `<link>` tags within a single page. As there may be conflicting styles, the order in which they are placed can be important.

7.10.2. Advantages of CSS

Some of the benefits of using CSS:

1. Style sheets allow you to separate the style and layout of your HTML files from their informational content. You can define the look of a site in one file, and change the whole site by changing just the one file.
2. Using *relative measurements* in your style sheet, you can present your documents to look good on any monitor at any resolution.
3. You have finer and more predictable control over presentation. CSS has been designed to deal with issues that HTML was not.
4. Older browsers will simply ignore your style sheets so they will not break existing pages.
5. The CSS style sheet mechanism allows readers to influence the presentation of HTML documents.
6. Style sheets will allow companies to easily adopt a house look and feel, and this will help give readers a sense of where they are and what they are reading.
7. The definition of a common style sheet format for HTML documents offers significant advantages over the many proprietary solutions which currently exist.
8. Style sheets will also improve the printing of web documents.
9. People with disabilities have better access to your pages. Visually impaired web users may need increased font sizes and will be among the first to benefit from style sheets.

7.10.3. CSS Rules

A CSS *rule set* consists of one or more statements in the following forms:

```
selector { property : value }
```

```
selector
```

```
{
  property : value;
  property : value;
  property : value;
  ...
}
```

The *selector* defines what elements the rule will be applied to. Following it is a *declaration block*, denoted by the curly braces ('{', '}'). Within this block are zero or more declarations. Multiple declarations are separated by a semicolon (;).

Each declaration consists of a *property* and a *value*, separated by a colon (':'). Properties are identifiers, such as font-size, background-color, etc. The values permitted depend on the individual property.

Whitespace may be used around any of these tokens, so you can format them for easy reading. You can also insert comments anywhere whitespace is permitted.

```
selector { /* a comment for this rule */
  property : value;
  property : value;
  property : value;
```

Selectors

In their simplest form, selectors consist of a tag name. Multiple tag names can be specified by separating them with a comma (','). So the following rules:

```
h1 { background-color: #ffff00 }
h2 { background-color: #ffff00 }
h3 { background-color: #ffff00 }
```

could also be expressed as:

```
h1, h2, h3 { background-color: #ffff00 }
```

This is known as *grouping*. Also note that a tag name can appear in any number of style rules.

```
h1, h2, h3 { background-color: #ffff00 }
h1 { color: red }
```

The styles of all rules that match a given element will be applied.

The simplest style rules associate a style declaration with an HTML tag. The following rule assigns the value green to the color property of first-level headers, i.e., h1's are green.

```
h1 {color: green;} or h1 {color:#00FF00;}
```

Multiple properties set in the same rule are separated by semicolons. This style might be used by a well-known high-tech culture magazine:

```
h1
{
```

```

color: white; background-color: red; font-family: Arial, sans-serif;
}

or

h1
{
color: #fff; (short cut for #ffffff) background-color: #f00; (short cut for #ff0000) font-family: Arial,
sans-serif;
}

```

7.10.4. Types of CSS

There are *three* types of CSS styles:

1. **Inline styles:** Inline styles are styles that are written directly in the tag on the document. Inline styles affect only the tag they are applied to.

```
<a href="" style="text-decoration: none;">
```

2. **Internal/Embedded styles:** Embedded styles are styles that are embedded in the head of the document. Embedded styles affect only the tags on the page they are embedded in.

```

<style type="text/css">
p { color: #00f; }
</style>

```

3. **External styles:** External styles are styles that are written in a separate document and then attached to various Web documents. External style sheets can affect any document they are attached to.

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

CSS best practises recommends that you use primarily external style sheets for styling web pages so that you get the most benefit of the cascade and inheritance.

Internal Style Sheets

Internal style sheets are styles that are placed in the <head> of the HTML document. These style sheets affect only the document they are in, and cannot be referenced by any other web document. Internal style sheets allow a designer to define styles for an HTML page. A complete internal style sheet would look like this:

```

<style type="text/css" media="all">
<!--
p { font: 1em Times serif; color: #c00; }
h1 { font: 2em Times serif; color: #f00; }
-->
</style>

```

Advantages of Internal Style Sheets

1. **Internal style sheets affect only the page they are on:** If you are working on a large site and need to test styles before you load them on the site as a whole, internal style sheets can be a great tool. They allow you to test the styles in the context of the entire site without breaking any page but the one you are testing.
2. **Internal style sheets can use classes and IDs:** Unlike inline styles, internal style sheets can still take advantage of classes, IDs, siblings, and other element relationships.
3. **Internal style sheets don't require that you upload multiple files:** This is especially useful when you're working with things like *email* where you have only one HTML file available to edit.
4. **Internal styles may have higher precedence than external style sheets.** This is determined by the order that the external style sheets are loaded. The Web developer of the page has control over where the internal styles will be placed in the head of the document. If they are placed after the link to external styles, they will have a higher precedence in the cascade, and over-ride the external style sheet.

Disadvantages of Internal Style Sheets

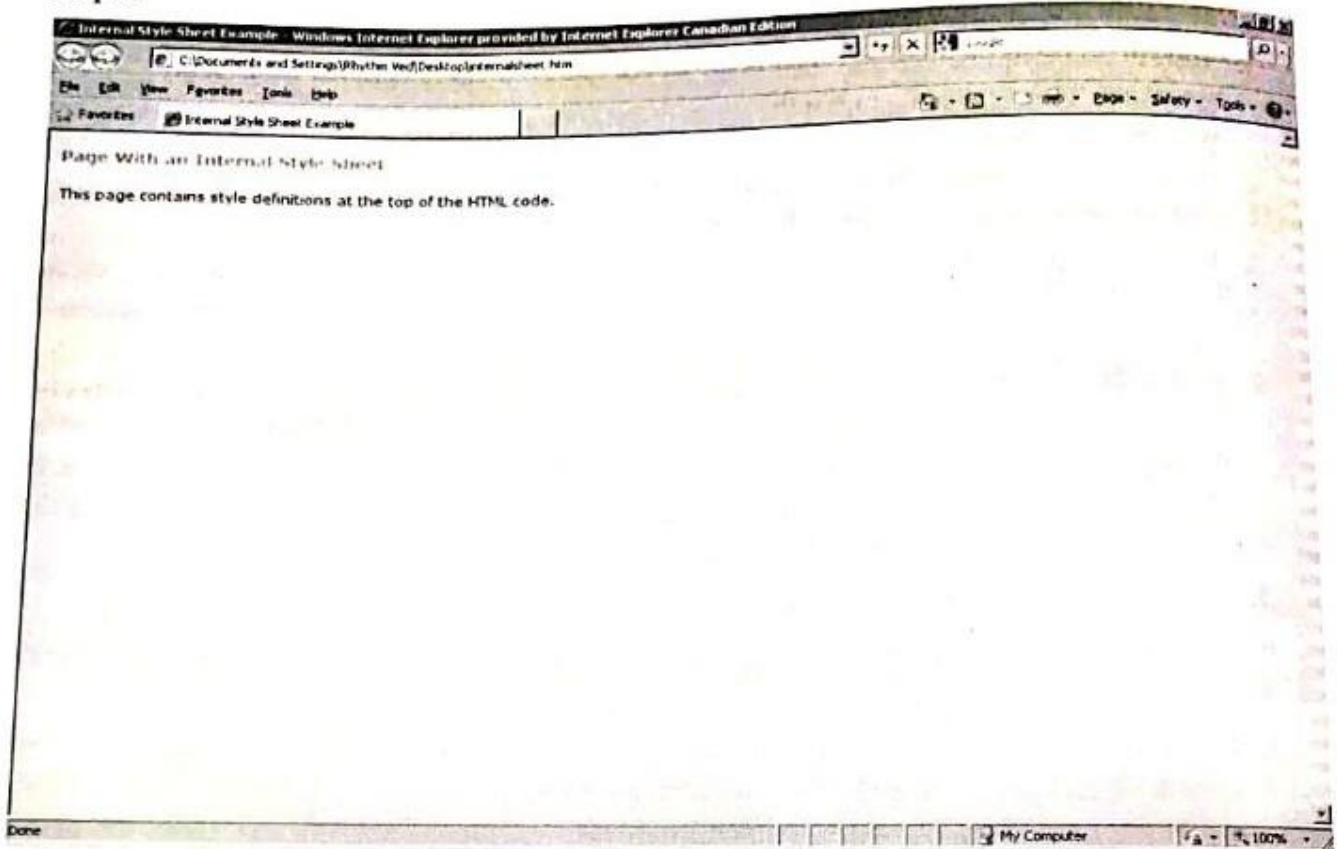
1. **They affect only the page they are on:** If you want to use the same styles in several documents, you need to repeat them for every page (or link to an external style sheet).
2. **Internal style sheets increase page load times:** Every page that has an internal style sheet must load and parse the style sheet information every time the page is loaded. External style sheets are cached by browsers—which improves load times for every page after the first is loaded.

Example 15 :

```
<html>
  <head>
    <title>Internal Style Sheet Example</title>
    <style>
      <!--
      body { background: #C9F1C5 }
      h1 { color: #54B24B; font: bold 14px Verdana, Arial, Helvetica }
      p { font: 12px Verdana, Arial, Helvetica }
      -->
    </style>
  </head>
  <body>
    <h1>Page With an Internal Style Sheet</h1>
    <p>This page contains style definitions at the top of the HTML code.</p>
  </body>
</html>
```

Save the code as *internalsheet.htm*

Output



External Style Sheet

An *external style sheet* is a separate file where you can declare all the styles that you want to use on your website. You then link to the external style sheet from all your HTML pages. This means you only need to set the styles for each element once. If you want to update the style of your website, you only need to do it in one place.

One of the best things about Cascading Style Sheets is that you can use them to keep your site consistent. The easiest way to do this is to link or import an external style sheet. If you use the same external style sheet for every page of your site, you can be sure that all the pages will have the same style. An external style sheet is a file with a .css extension that contains style definitions. External style sheets can be linked to multiple pages for a consistent look across a site. It might look something like this:

```
body { background: #C2DAED }
h1 { color: #3270A4; font: bold 14px Verdana, Arial, Helvetica}
p { font: 12px Verdana, Arial, Helvetica }
```

TIPS

If the link to the external style sheet is placed after the internal style sheet in HTML <head>, the external style sheet will override the internal style sheet!

Advantages of External Style Sheets

1. You can control the look and feel of several documents at once. This is especially useful if you work with a team of people to create your website. Many style rules can be difficult to remember, and while you might have a printed style guide, it is inefficient and tedious to be continuously flipping through it.
2. You can create classes of styles that can then be used on many different HTML elements. If you often use a special Wingdings font to give emphasis to various things on your page, you can use the Wingdings class you set up in your style sheet to create them, rather than defining a specific style for each instance of the emphasis.
3. You can easily group your styles to be more efficient. All the grouping methods that are available to CSS can be used in external style sheets, and this provides you with more control and flexibility on your pages.

Disadvantages of External Style Sheets

1. External style sheets can increase the download time, if they are extremely large.
2. If you only have a small number of styles, they can increase the complexity of your site.
3. Like with table rendering, you have to wait until the entire style sheet is loaded before the page can display.
4. External style sheets get big very quickly as it's hard to tell when a style is no longer in use, because it's not deleted when the page is removed.

Example 16 :

```
<html>
  <head>
    <title>
      Document Title
    </title>
    <link REL="StyleSheet" TYPE="text/css"
      HREF="extern.css">
  </head>
  <body>
    <h1 ALIGN="center">Welcome to my home page!</h1>
    <hr>
    <br><br>
    <h2 > About <em>CSS</em> </h2>
```

<p>Cascading style sheet languages such as CSS allow style information from several sources to be blended together. However, not all style sheet languages support cascading. To define a cascade, authors specify a sequence of LINK and/or STYLE elements.

The style information is cascaded in the order the elements appear in the HEAD.</p>

```
<br><br>
```

Types of CSS

```
<table border="2">
```

```
<tr>
```

```
<td>Inline sheet</td>
```

```
</tr>
```

```
<tr>
```

```
<td>Internal style sheet</td>
```

```
</tr>
```

```
<tr>
```

```
<td>External style sheet</td>
```

```
</tr>
```

```
</table>
```

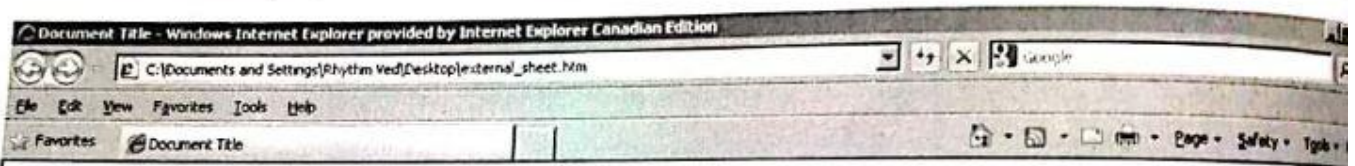
```
</body>
```

```
</html>
```

Save the code as *external_sheet.htm*

Output

To view the output, run the *external_sheet.htm* file



Welcome to my home page!

About CSS

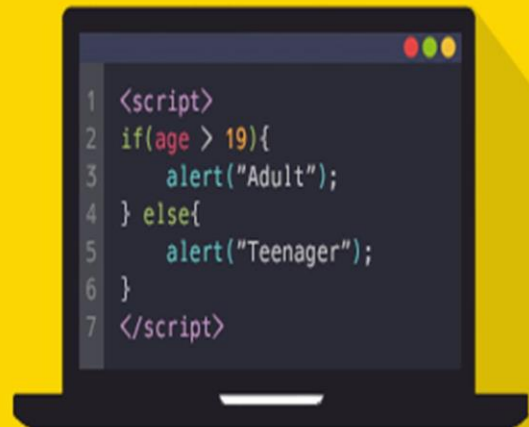
Cascading style sheet languages such as CSS allow style information from several sources to be blended together. However, not all style sheet languages support cascading. To define a cascade, authors specify a sequence of LINK and STYLE elements. The style information is cascaded in the order the elements appear in the HEAD.

Types of CSS

Inline sheet
Internal style sheet
External style sheet

Internet and Web Technology

Chapter 7 - Client side Scripting with JavaScript



Chapter Contents

- ❖ Introduction to script, Client side Scripting, Types of Scripting
- ❖ Variables in JavaScript, Built -in Function
- ❖ Arrays in JavaScript, Conditional statements, Loops
- ❖ Document Object Model
- ❖ Creating Functions, objects in JavaScript
- ❖ Event handling in JavaScript
- ❖ Embedding JavaScript with HTML
- ❖ Working with Cookies
- ❖ Connecting database using JavaScript in HTML Page
- ❖ Working with Browser, validating and submitting Forms

Client side Scripting with JavaScript - Introduction to script, Clientside Scripting, Types of Scripting

Introduction to Script

HTML is a language used to create hyperlinked text documents, which as a total comprise what is called the World Wide Web. CSS is a language used to control the presentation of these documents. Sometimes, these documents are not static, unchanging texts, but are created by computers on-the-fly. To facilitate this, technologies were developed several years ago that create web pages dynamically, Programs that do this run on web servers, and usually use data from some source (e.g., a database) and some user input (e.g., the data submitted via an HTML form), process these, and create output in the form of a web page

These kinds of programs differ from other kinds of computer programs in that their output is not sent to a screen printer file, but is a web page. The output of these programs is no different from a Static HTML document in any way. In order to write such programs, you have to learn HTML just as if you were designing a static set of documents. The only difference is that these programs automate the procedure.

However, after a while, someone came up with the idea of sticking programs straight into web pages, so that these web pages would turn into small applications that run on user's computers. These programs are called side scripts, because they run on the client (i.e., the browser displaying the document) as opposed to the server i.e., the computer offering the document).

What is a scripting language?

- A script or scripting language is a computer language with a series of commands within a file capable of being executed without being compiled. Good examples of server-side scripting languages include Perl, PHP, and Python. The best example of a client side scripting language is JavaScript. A full list of scripting languages and other programming languages are found in our programming language definition.
- A scripting language is used to write scripts. These contain a series of commands that are interpreted one by one at runtime unlike programming languages that are compiled first before running.

Advantages of scripts

- Open source, allowing users to view and edit the script if needed.
- Does not require the file to be compiled, but may be when necessary.
- Easy to learn and write.
- Easy to port between different operating systems.
- Much faster to develop than an actual program – some individuals and companies write scripts as a prototype for actual programs.

Disadvantages of scripts

- Open source, allows others to view source code, which may be prohibited by some companies.
- Requires the user to install an interpreter or separate program before the script can run.
- In some situations, they may be slower than a compiled program.

Nowadays, scripts are generally associated with web development where they are widely used to make dynamic web applications.

Types of scripts

Scripting languages can be divided into two categories:

- ❖ Server Side Scripting Languages
- ❖ Client Side Scripting Languages

Server-side scripting languages create the scripts that run on the server and hence minimize the workload of a browser. The functionality of your website is written in those scripting languages. The most commonly used server-side scripting languages are Perl, Ruby, Python, PHP, etc.

Client-side scripting languages create the scripts that run on the client side (i.e. your browser). These are sent from the server by server-side scripts. Some good examples are JavaScript, jQuery, CSS etc

Scripting languages

- HTML is the way to create the static web pages. In order to validate the forms on the client side adding dynamically to the static pages etc., one have to use client side scripting language. The main scripting languages available are:
 - ✓ **VB Script**
 - ✓ **JavaScript**
- All the scripting is placed in the `<script>` . `</script>` tags. Since, internet explorer support both visual basic and JavaScript, one will have to set language attribute of the `<script>` tag,

```
<script language = "JavaScript">
```

or

```
<script language = "VB Script">
```

- The scripting language performs somewhat task of the server side languages but resides on the client side. Hence, the word load on the server reduces and the job speeds up. The scripting languages contain all the procedural fundamentals i.e.,
 1. **Conditional executions**
 2. **Looping constructs**
 3. **Subroutines**
 4. **Data validations etc.**

- Before the implementation of client side scripting, the whole task was the responsibility of the server side scripting languages. The client side was supposed to send the whole data to the server. The server validates the input data, checks for any corrections and then sends the response back to the client. This process slows up the whole phenomenon. Then, the task like validation of input forms, adding little bit of dynamically to the web pages were handed over to the scripting language.

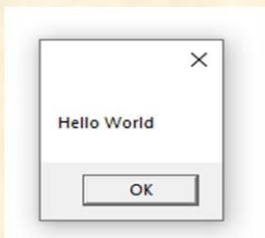
VB Script:

The VBScript stands for Visual Basics Script language. The Microsoft has developed its way to the dynamic web programming in VB script. It is used to develop dynamic web pages. It is much lighter compared to Visual Basic programming language but works as a scripting language like JavaScript. The way to program the VB script is to divide the entire task in the subroutines or functions, and then attach the appropriate functions to the different events.

Example:

```
MsgBox "Hello World"
```

Output:



The above snippet of code, displays a message whenever a button (having caption "Show Message") is clicked. The text and the HTML tags can be sent to the browsers by using document write method.

Example:

```
<html>
  <body>
    <script language = "vbscript" type = "text/vbscript">
      document.write("Hello World")
    </script>
  </body>
</html>
```

JavaScript:

Java script is a dynamic computer programming language. JavaScript was first known as Live Script, but Netscape changed its name to JavaScript, possibly because of the excitement being generated by Java. JavaScript made its first appearance in Netscape 2.0 in 1995 with the name Live Script. The general-purpose core of the language has been embedded in Netscape, Internet Explorer, and other web browsers. The ECMA -262 Specification defined a standard version of the core JavaScript language. It is a cross-platform, object based scripting language. JavaScript is not a

object-oriented language but it is a object based language. Object based languages are those which have some of the capabilities of the object but not all of them.

Features of JavaScript

- JavaScript is a lightweight, interpreted programming language.
- Designed for creating network-centric applications.
- Complementary to and integrated with Java.
- Complementary to and integrated with HTML.
- Open and cross-platform.

Advantages of JavaScript

The merits of using JavaScript are:

- **Less server interaction:** You can validate user input before sending the page off to the server. This saves server traffic, which means less load on your server.
- **An interpreted Language:** It is an interpreted language, which requires no compilation steps. This provides an easy development process.
- **Embedded within HTML:** Within HTML code, It does not require any special or separate editor, it is embedded
- **Quick Development:** As it does not require any time consuming compilation scripts can be developed in less time.
- **Easy to Learn:** It implements simple rules of syntax and very few commands which are easy
- **Performance:** It also minimizes storage requirements on the web server and down-loading time for the client.
- **Procedural Capabilities:** JavaScript supports almost all the procedural capabilities such as looping, branching etc.
- **Immediate feedback to the visitors:** They don't have to wait for a page reload to see if they have forgotten to enter something.
- **Increased interactivity:** You can create interfaces that react when the user hovers over them with a mouse or activates them via the keyboard.
- **Richer interfaces:** You can use JavaScript to include such items as drag and-drop components and sliders to give a Rich Interface to your site visitors.

Limitations of JavaScript

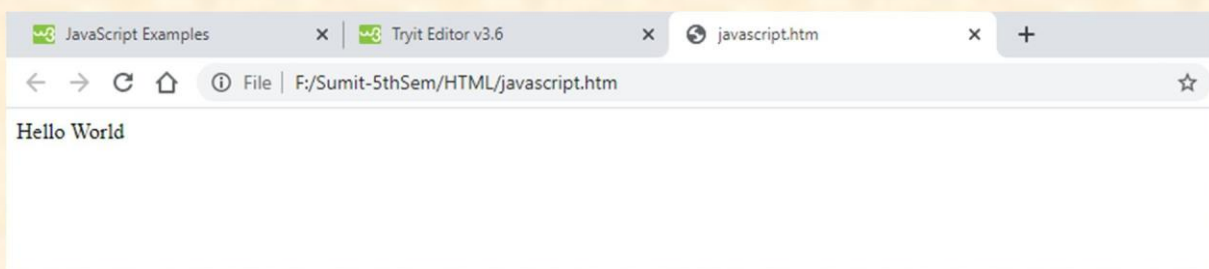
We cannot treat JavaScript as a full-fledged programming language. It lacks the following important features:

- Client-side JavaScript does not allow the reading or writing of files. This has been kept for security reason.
- JavaScript cannot be used for networking applications because there is no such support available.
- JavaScript doesn't have any multithreading or multiprocessor capabilities. Once again, JavaScript is a lightweight, interpreted programming language that allows you to build interactivity into otherwise static HTML pages.

Example: The following code script shows us how to embed JavaScript into the HTML:

```
<html>
  <body>
    <script>
      document.write("Hello World");
    </script>
  </body>
</html>
```

Output:



Role of Scripting

There is no definitive definition of a scripting language. Sometimes the term is used to make distinction from compiled languages. However, some languages like C or C++ can be used for scripting as well as full applications, the term scripting is also used because a language will react to control "script" a series of events. Even macro languages built into PC applications like spreadsheets, databases, word processors, and multimedia applications are now often called scripting languages.

The purpose of most scripting languages is to extend the capabilities of applications. Just as the authors of this book cannot imagine every creative use you will make of JavaScript 4/31 cannot imagine every possible use of their applications, to make their products more a scripting language. With JavaScript you have a scripting language to use your imagination on the web.

Current uses of scripting languages may give you an insight of the potential for JavaScript. You probably know that macros are built into many PC applications. Apple's HyperCard contains a very powerful scripting feature. Perl is a scripting language used in many CGI scripts you use on the web.

Client-Side Scripting

Client side scripting is scripting which is done in Web Pages resides at client side (Deals with web browsers). The Client side scripts are not all special effects however some can be functional, providing a way to check form input for validity before sending the data to the server.

Ideas of when to use client side scripts

- Complimentary form preprocessing (should not be relied upon!)
- To get data about the user's screen or browser.
- Online games.
- Customizing the display (without reloading the page)

Scripts in Web Applications

Scripting differs from full-scale application programming in that full-blown programs run from the operating system and manipulate data and system resources to carry out their processing. On the other hand, scripts are designed to be embedded in a container application, where they manipulate data and application resources to carry out their processing, while being precluded from directly accessing the underlying system. While they will not be discussed in detail in this article. Web applications often contain scripts that run on the web server-called server-side scripts-as well as client-side scripts. Client-side scripts in a web application are run by a software application-the scripting engine that is part of the web browser. The scripting engine parses and runs the script code, allowing it "manipulate resources of the browser and web document.

The Role of Client-Side Scripting

There are following two important role of client-side scripting:

1. Using a client-side script for interactivity in a web application usually provides the appearance of better performance than using a server-side script to implement the same functionality because the client-side script does not incur the delay of a round trip between the client and the server. Making the web application more responsive. This is crucial for a dynamic user experience in a web application.
2. Client-side scripting offers the benefit of off-loading part of the computational demands of an application from the server to the client. The client-side computer is usually wasting most of its processing potential, idly waiting for the user to do something. Web servers, on the other hand, are usually taxed to their limit, serving hundreds or thousands of concurrent users.

Capabilities of Client-Side Scripts

1. Client-side scripts execute in the web browser under strictly enforced rules designed to protect the user's system from malicious code. Client-side scripts can control most browser behavior and manipulate most HTML document elements.
2. Client-side scripts are often used for navigation and to enrich and customize the user experience. Web developers extend the look and feel of the standard application by

associating a script with a document element in such a way that the script executes in response to user interaction with the element.

Limitations of Client-Side Scripts

Like any technology, the value of client-side scripting is a trade-off between its pros and cons. The following are some limitations of client side scripts

1. The safety is major concern for client side scripts. Client-side scripts are prevented from implementing certain features, which may cause threat to the security. For example, client-side scripts cannot carry out file I/O or access a database.
2. Anyone who can load the document can readily examine the client-side script code.
3. It is very difficult to hide the programming logic from the user, which raises serious concerns about protecting intellectual property.

Types of scripts

There are several types of scripts. Let us see some of them as follows:

Immediate Scripts

Immediate script is the term to indicate lines of JavaScript that not only run when the browser loads the document, but also influence the layout of the page. Such Scripts are placed in <body> sections

```
<html>
  <body>
    <script>
      // Script that produces content for the body
    </script>
  </body>
</html>
```

Example:

```
<!DOCTYPE html>
<html>

<head>
  <title>
    How to run a function when the
    page is loaded in javascript ?
  </title>
</head>

<body onload="console.log('The Script will load now.')">

  <h1 style="color: green">TECHSTUDY</h1>

  <b>
    How to run a function when the
    page is loaded in javascript ?
  </b>
```

```
<p>
  The script has been executed. Check
  the console for the output.
</p>
</body>
</html>
```

Deferred Scripts

A deferred script is one that the browser sees when the document loads, but the wording of the script tells the browser not to do anything with the code other than to be aware that it exists. Such script sections usually consist of small groups of script lines. Adding a deferred script to the Head Block prepares the browser to respond to user interaction later.

```
<html>
  <body>
    <script>
      // Script that Initializes items for user-driven actions
    </script>
  </body>
</html>
```

Example:

```
<html>
  <head>
    <script>
      function sayHello()
      {
        document.write ("Hello there!");
      }
    </script>
  </head>
  <body>
    <p>Click the following button to call the function</p>
    <button type="button" onclick="sayHello()" id="sh"> Say Hello</button>
  </body>
</html>
```

Hybrid Scripts

These scripts are used in designing pages, which require both immediate and deferred scripts. The immediate script lines help to create the contents of the page while the deferred script lines react to the user's actions once the page has been fully loaded.

```
[1] <script language="JavaScript" src="filename.js"></script>
[2] <script language="JavaScript">
      // Script goes here </SCRIPT>
[3] <html><head>
      <script language="JavaScript"> var hello="Welcome to JavaScript";
      </script> <noscript>
```

```
        Sorry! Your browsers does not support
</noscript>
</head> <body>
<script language="javaScript"> document.write(hello);
</script>
</body> </html>
```

Variables in JavaScript, Built-in Function

Variables in JavaScript

- Variables are name given to some memory location to store values temporarily.
- Variable names in JavaScript, can begin with alphabet (A through Z and a through z) or underscore) character or (\$) sign. The rest of the characters may be letters, digits (0 through 9) or underscore.
- E.g. name, total_amount , etc.

Variable Types

- JavaScript accepts the following types of variables:
- **Numeric:** Any numeric value, whether a whole number (an integer) or a number that includes a fractional part (a real),
 - e.g., 12, 3.1 4159.. etc.
- **String:** A group of text characters,
 - e.g., Ian, Macintosh G4, etc
- **Boolean:** A value which can only be either True or False,

We create variables and assign values to them in the following way:

```
var christianName = "Fred" (string)
var surname = "Jones" (string)
var age = 37 (numeric)
var married = false (Boolean)
```

Note that:

- When a new variable is created (or declared) its name must be preceded by the word var
- The type of the variable is determined by the way it is declared:
 - ✚ if it is enclosed within quotes, it's a string
 - ✚ if it is set to true or false (without quotes) it's a boolean
 - ✚ if it is a number (without quotes) it's numeric
- We refer to the equals sign as the assignment operator because we use it to assign values to variables;

- Variable names must begin with a letter or an underscore
- Variable names must not include spaces
- JavaScript is case-sensitive

Reserved words (i.e., words which indicate an action or operation in JavaScript) cannot be used as variable names.

Type Conversion

JavaScript has the capability to convert values automatically from one type to another, when used in expression JavaScript provides some pre-defined functions for explicit type conversion like `eval()`, `parseInt()` and `parseFloat()`.

- **`eval()` function** can be used to convert string value or expression to a numeric value.

Example:

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

<p>Click the button to evaluate/execute JavaScript code/expressions.</p>
<button onclick="myFunction()">Try it</button>

<p id="demo"></p>

<script>
function myFunction() {
  var x = 10;
  var y = 20;
  var a = eval("x * y") + "<br>";
  var b = eval("2 + 2") + "<br>";
  var c = eval("x + 17") + "<br>";

  var res = a + b + c;
  document.getElementById("demo").innerHTML = res;
}
</script>

</body>
</html>
```

- **`parseInt()` function** converts a string to an Integer value. It returns 0 if the string does not begin with an integer or else, returns the first integer contained in the string.

Example:

```
<!DOCTYPE html>
<html>
<body>
  <script>
```

```

a = parseInt("100");
document.write('parseInt("100") = ' +
a + "<br>");

// It returns a Integer until
// it encounters Not a Number character
b = parseInt("2018@helloandhi");
document.write('parseInt("2018@helloandhi") = ' +
b + "<br>");

// It returns NaN on Non numeral character
c = parseInt("helloandhi@2018");
document.write('parseInt("helloandhi@2018") = ' +
c + "<br>");

// It returns Integer value of a Floating point Number
d = parseInt("3.14");
document.write('parseInt("3.14") = ' + d + "<br>");

// It returns only first Number it encounters
e = parseInt("21 7 2018");
document.write('parseInt("21 7 2018") = ' +
e + "<br>");

</script>
</body>
</html>

```

- **parseFloat()** function is very similar to the **parseInt()** function. It returns the first floating point number contained in as string or else, returns 0 if the string does not begin with a valid floating point number.

JavaScript Comments

JavaScript uses similar syntax for comments as C/C++ and Java. You can use `//` or `*` and `*/` combination can be used for comments.

JavaScript Built In Functions

Like other languages JavaScript also provides in built function like:

alert(string or variable): To give a message dialog box

prompt("message","default value"): To give a message and get value in the form of string and pass is to some variable.

confirm("message"): Used to confirm something from user by giving a confirmation dialog box with command buttons OK and Cancel. It returns boolean value, true or false which can be used in conditional statements.

Example:

```

<html>
  <head>
    <title>Using Built-in Functions</title>
  </head>
  <body>
    <h2>JavaScript Built-in Functions</h2>

    <button onclick="myFunction()">Try it</button>

    <p id="demo"></p>

    <script>
      function myFunction() {
        var txt;
        alert("Welcome to JavaScript");
        var name = prompt("Please Enter your name", "");
        if (confirm("Press a button!"))
        {
          alert("Dear "+name+ " Thanks for Using JavaScript");
        }
        else
        {
          alert("Try next Time");
        }
        document.getElementById("demo").innerHTML = txt;
      }
    </script>
  </body>
</html>

```

Common Built-in Functions

Functions	Description
isNaN()	Returns true, if the object is Not a Number. Returns false, if the object is a number.
parseFloat (string)	If the string begins with a number, the function reads through the string until it finds the end of the number; cuts off the remainder of the string and returns the result. If the string does not begin with a number, the function returns NaN.
parseInt (string)	If the string begins with an integer, the function reads through the string until it finds the end of the integer, cuts off the remainder of the string and returns the result. If the string does not begin with an

	integer, the function returns NaN (Not a Number).
String (object)	Converts the object into a string.
eval()	Returns the result of evaluating an arithmetic expression.

User-defined Functions

User-defined function means you can create a function for your own use. You can create yourself according to your need.

In JavaScript, these functions are written in between the <HEAD> tag of the HTML page.

Syntax:

```
function function_name()
{
    //Code;
}
```

Example : Function Declaration

```
function add()
{
    var a, b;
    var sum = 0;
    sum = a + b;
    document.write("Addition : "+sum);
}
```

How is a function executed on an event in JavaScript?

```
<input type= "button" onClick= "add()" value= "buttonvalue">
```

onClick – Event Handler

add() – Function name

Arrays in JavaScript, Conditional statements, Loops

Arrays In Javascript

Arrays

The Array object lets you store multiple values in a single variable. It stores a fixed-size sequential collection of elements of the same type. An array is used to store a collection of data, but it is often more useful to think of an array as a collection of variables of the same type.

These values are stored in indexed locations within the array. The length of an array is the number of elements that an array contains. The individual elements of an array not accessed without using the name of array followed by the index value of the array element enclosed in square brackets. The array element index starts with 0. Hence the last element is one less than the length of the array Array must be declared before it is used. The array can be declared as:

Syntax :

```
arrayname = new array (Array length)
           or
arrayname = new array, (Array length)
           or
var arrayname = [ "newarray", " newarray ", " newarray " ];
```

Array indexes are zero-based: The first element in the array is 0, the second is 1, and so on. The Array parameter is a list of strings or integers. When you specify a single numeric parameter with the Array constructor, you specify the initial length of the array. We will use ordinal numbers to access and to set values inside an array as follows.

Example:

```
var fruits = [ "apple", "orange", "mango" ];
```

fruits[0] is the first element

fruits[1] is the second element

fruits[2] is the third element

Array Methods

Method Description

concat()	Joins two or more arrays, and returns a copy of the joined arrays
copyWithin()	Copies array elements within the array, to and from specified positions
find()	Returns the value of the first element in an array that pass a test
findIndex()	Returns the index of the first element in an array that pass a test
forEach()	Calls a function for each array element
indexOf()	Search the array for an element and returns its position
isArray()	Checks whether an object is an array
join()	Joins all elements of an array into a string

lastIndexOf()	Search the array for an element, starting at the end, and returns its position
pop()	Removes the last element of an array, and returns that element
push()	Adds new elements to the end of an array, and returns the new length
reverse()	Reverses the order of the elements in an array
sort()	Sorts the elements of an array
splice()	Adds/Removes elements from an array
toString()	Converts an array to a string, and returns the result
valueOf()	Returns the primitive value of an array

Example:

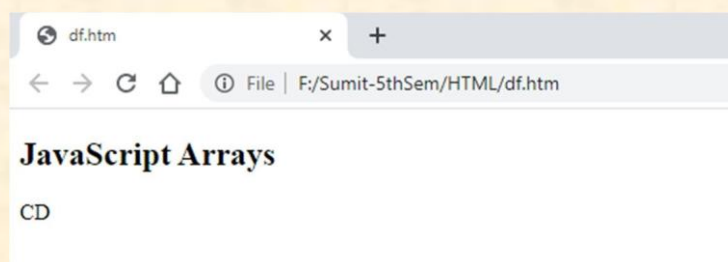
```
<html>
  <body>
    <h2>JavaScript Arrays</h2>

    <p id="demo"></p>

    <script>
      var A = ["AB", "CD", "EF"];
      document.getElementById("demo").innerHTML = A[1];
    </script>

  </body>
</html>
```

Output :



Conditional statements, Loops

If Statement

Used to test the given condition, if condition is true the execute the statement written within a block otherwise execute the statement written in else block. Let us see some of the Syntax variations as follows:

```

(a)  if (condition)
      {
          statements;
      }
(b)  if(condition)
      {
          statements;
      }
      else
      {
          statements;
      }
(c)  if(condition)
      {
          if (condition)
          {
              statements;
          }
      }
(d)  if (condition 1)
      {
          statements;
      }
      else if(condition2)
      {
          statements;
      }
      else
      {
          statements;
      }

```

Example: Get two numbers and show greatest of that numbers.

```

<html>
  <head>
    <title>Show greatest of that numbers.</title>
  </head>
  <body>
    <h2>Show greatest of that numbers</h2>

    <script language="JavaScript">
      var n1 = prompt("Please Enter 1st Number","");
      var n2 = prompt("Please Enter 2nd Number","");
      n1 = parseFloat(n1);
      n2 = parseFloat(n2);
      if(n1>n2)
      {
        document.write(n1+" is greater");
      }
      else

```

```

        {
            document.write(n2+" is greater");
        }
    </script>
</body>
</html>

```

Output:



Example: Get three numbers and find smallest of three numbers.

```

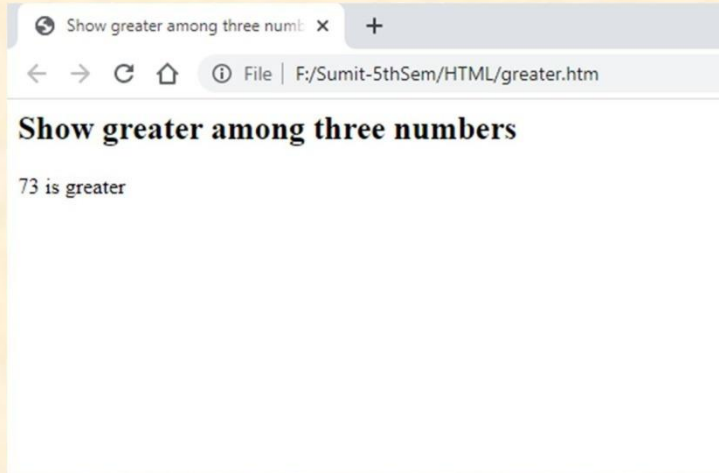
<html>
  <head>
    <title>Show greater among three numbers</title>
  </head>
  <body>
    <h2>Show greater among three numbers</h2>

    <script language="JavaScript">
      var n1 = prompt("Please Enter 1st Number","");
      var n2 = prompt("Please Enter 2nd Number","");
      var n3 = prompt("Please Enter 3rd Number","");
      n1 = parseFloat(n1);
      n2 = parseFloat(n2);
      n3 = parseFloat(n3);
      if(n1>n2)
      {
          if(n1>n3)
          {
              document.write(n2+" is greater");
          }
          else
          {
              document.write(n3+" is greater");
          }
      }
      else
      {
          if(n2>n3)
          {
              document.write(n2+" is greater");
          }
          else
          {
              document.write(n3+" is greater");
          }
      }
    </script>
  </body>
</html>

```

```
    }  
    </script>  
</body>  
</html>
```

Output :



Switch Case:

The objective of a switch statement is to give an expression to evaluate and several different statements to execute based on the value of the expression. The interpreter checks each case against the value of the expression until a match is found. If nothing matches, a default condition will be used.

Syntax :

```
switch (expression)  
{  
  case condition 1:  
    statement(s)  
    break;  
  case condition 2:  
    statement(s)  
    break;  
  ...  
  case condition n:  
    statement(s)  
    break;  
  default:  
    statement(s)  
}
```

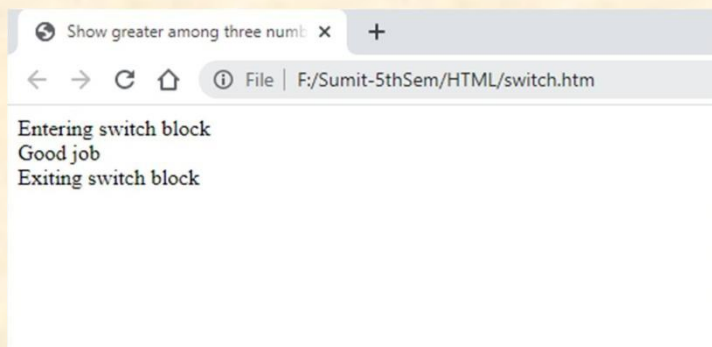
The break statements indicate the end of a particular case. If they were omitted, the interpreter would continue executing each statement in each of the following cases.

Example:

```
<html>  
  <head>
```

```
<title>Show greater among three numbers</title>
</head>
<body>
  <script language="JavaScript">
    var grade='A';
    document.write("Entering switch block<br />");
    switch (grade)
    {
      case 'A':
        document.write("Good job<br />");
        break;
      case 'B':
        document.write("Pretty good<br />");
        break;
      case 'C':
        document.write("Passed<br />");
        break;
      case 'D':
        document.write("Not so good<br />");
        break;
      case 'F':
        document.write("Failed<br />");
        break;
      default:
        document.write("Unknown grade<br />")
    }
    document.write("Exiting switch block");
  </script>
</body>
</html>
```

Output :



Ternary Operator (?:):

It is shortcut of if-else

condition ? value for true condition : value for false condition;

Example: Get two numbers and show greatest of that numbers.

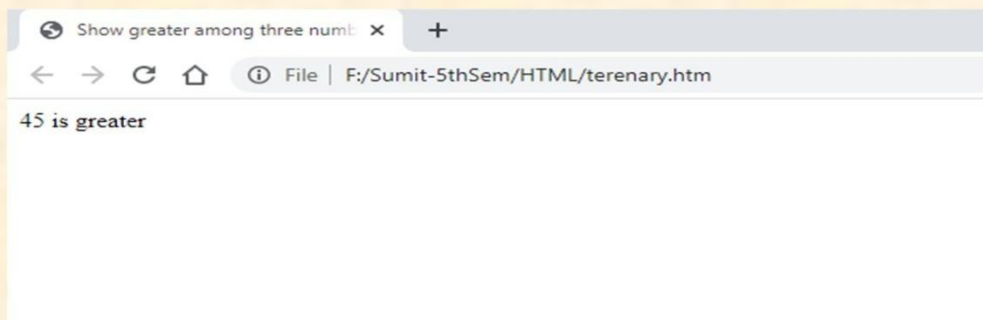
```

<html>
  <head>
    <title>Show greater among three numbers</title>
  </head>
  <body>
    <script language="JavaScript">
      var n1 = prompt("Please Enter 1st Number","");
      var n2 = prompt("Please Enter 2nd Number","");
      n1 = parseFloat(n1);
      n2 = parseFloat(n2);
      var g=(n1>n2)? n1:n2;
      document.write(g+" is greater");

    </script>
  </body>
</html>

```

Output :



Iterations Or Loops

The while Loop :

The most basic loop in JavaScript is the while loop. The purpose of a while loop is to execute a statement or code block repeatedly as long as an expression is true. Once the expression becomes false, the loop terminates.

Syntax:

```

while (expression)
{
  Statement(s) to be executed if expression is true
}

```

Example:

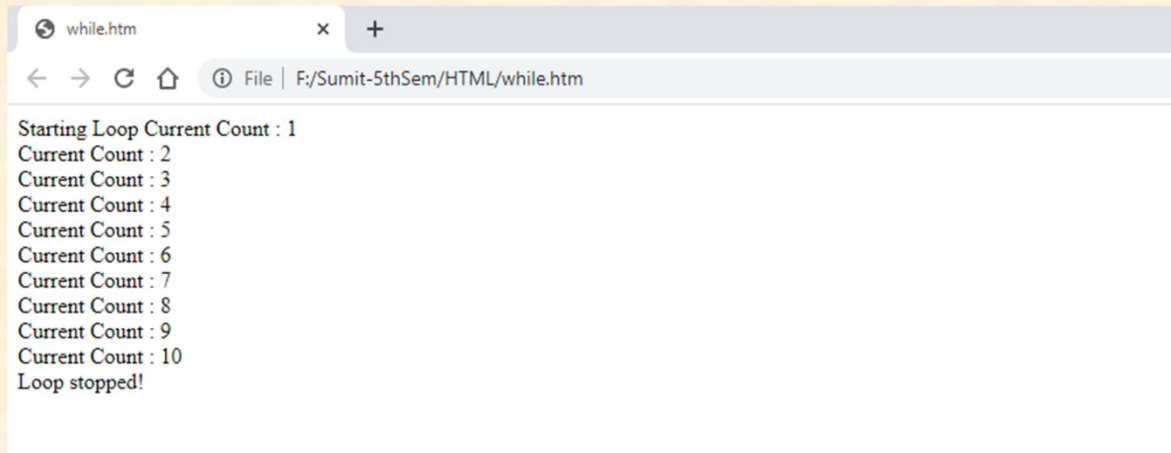
```

<html>
  <body>
    <script>
      var count = 1;
      document.write("Starting Loop ");
      while (count <= 10)
      {
        document.write("Current Count : " + count + "<br />");
      }
    </script>
  </body>
</html>

```

```
        count++;
    }
    document.write("Loop stopped!");
</script>
</body>
</html>
```

Output :



The do...while Loop

The do...while loop is similar to the while loop except that the condition check happens at the end of the loop. This means that the loop will always be executed at least once, even if the condition is false.

Syntax:

```
do
{
    Statement(s) to be executed;
}
while (expression);
```

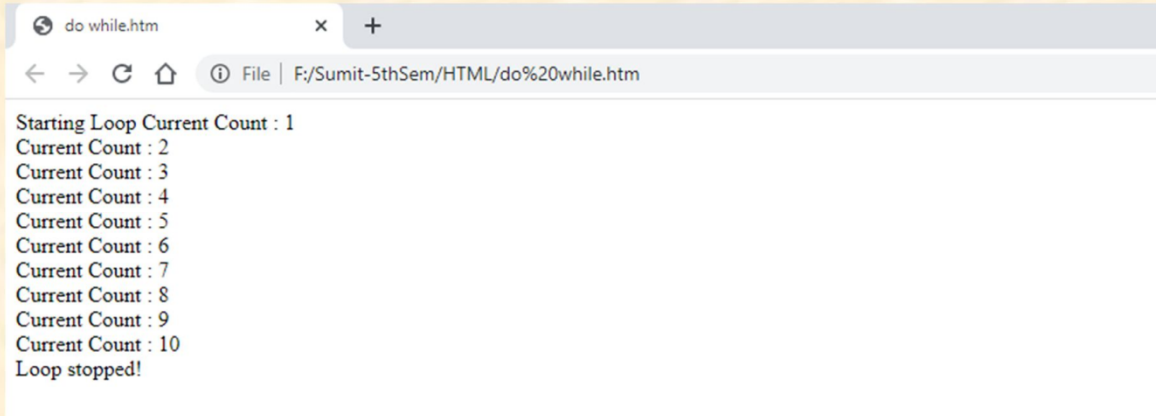
Note: Don't miss the semicolon used at the end of the do...while loop.

Example:

```
<html>
<body>
<script>
    var count = 1;
    document.write("Starting Loop ");
    do
    {
        document.write("Current Count : " + count + "<br />");
        count++;
    }
    while (count <= 10);
    document.write("Loop stopped!");
</script>
```

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

Output :



```
Starting Loop Current Count : 1
Current Count : 2
Current Count : 3
Current Count : 4
Current Count : 5
Current Count : 6
Current Count : 7
Current Count : 8
Current Count : 9
Current Count : 10
Loop stopped!
```

The for Loop :

The 'for' loop is the most compact form of looping. It includes the following three important parts:

- The loop initialization where we initialize our counter to a starting value. The initialization statement is executed before the loop begins.
- The test statement which will test if a given condition is true or not. If the condition is true, then the code given inside the loop will be executed, otherwise the control will come out of the loop.
- The iteration statement where you can increase or decrease your counter. You can put all the three parts in a single line separated by semicolons.

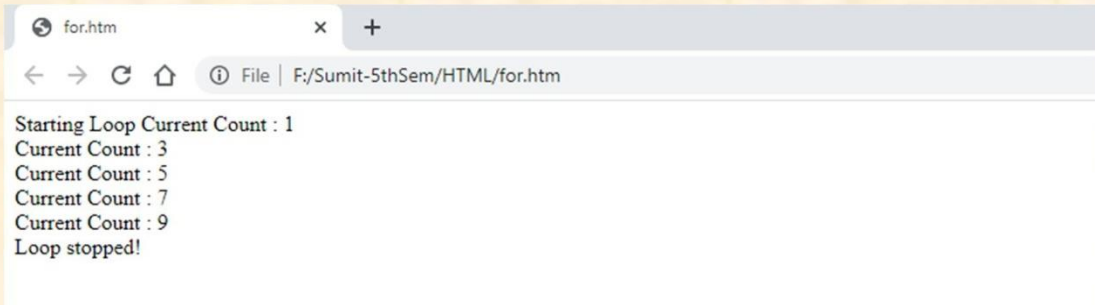
Syntax:

```
for (initialization; test condition; iteration statement)
{
    statement(s) to be executed if test condition is true
}
```

Example:

```
<html>
  <body>
    <script>
      var count;
      document.write("Starting Loop ");
      for(count=1;count<=10;count++)
      {
        document.write("Current Count : " + count + "<br />");
        count++;
      }
      document.write("Loop stopped!");
    </script>
  </body>
</html>
```

Output :



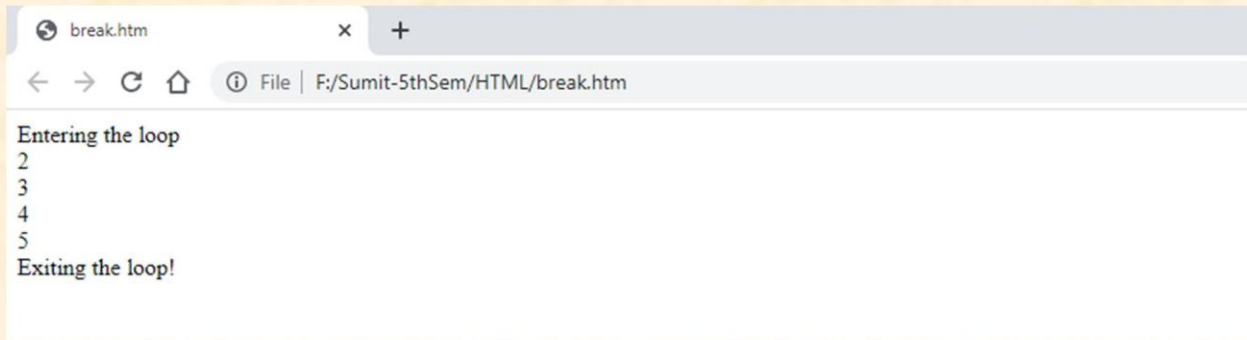
The break Statement :

JavaScript provides full control to handle loops and switch statements. There may be a situation when you need to come out of a loop without reaching at its bottom. There may also be a situation when you want to skip a part of your code block and start the next iteration of the loop. To handle all such situations, JavaScript provides break and continue statements. These statements are used to immediately come out of any loop or to start the next iteration of any loop respectively. The break statement, which was briefly introduced with the switch statement, is used to exit a loop early, breaking out of the enclosing curly braces.

Example:

```
<html>
  <body>
    <script>
      var x = 1;
      document.write("Entering the loop<br /> ");
      while (x < 20)
      {
        if (x == 5)
        {
          break; // breaks out of loop completely
        }
        x = x + 1;
        document.write( x + "<br />");
      }
      document.write("Exiting the loop!<br /> ");
    </script>
  </body>
</html>
```

Output :



```
break.htm
x +
← → ↻ 🏠 ⓘ File | F:/Sumit-5thSem/HTML/break.htm
Entering the loop
2
3
4
5
Exiting the loop!
```

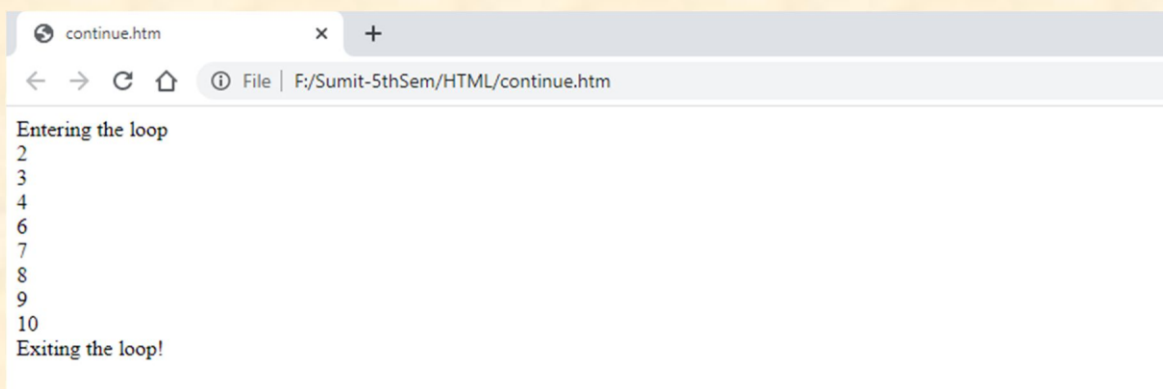
The continue Statement :

The continue statement tells the interpreter to immediately start the next iteration of the loop and skip the remaining code block. When a continue statement is encountered, the program flow moves to the loop check expression immediately and if the condition remains true, then it starts the next iteration, otherwise the control comes out of the loop.

Example:

```
<html>
  <body>
    <script>
      var x = 1;
      document.write("Entering the loop<br /> ");
      while (x < 10)
      {
        x = x + 1;
        if (x == 5)
        {
          continue; // skip rest of the loop body
        }
        document.write( x + "<br />");
      }
      document.write("Exiting the loop!<br /> ");
    </script>
  </body>
</html>
```

Output :



```
continue.htm
x +
← → ↻ 🏠 ⓘ File | F:/Sumit-5thSem/HTML/continue.htm
Entering the loop
2
3
4
6
7
8
9
10
Exiting the loop!
```

Operators Used In Javascript

Name of Operators	Operators	Description
Arithmetic Operators	+	Addition
	-	Subtraction
	*	Multiplication
	/	Division
	%	Modulus
	++	Increment
	--	Decrement
Logical Operators	&&	Logical And
		Logical Or
	!	Logical Not
Comparison Operators	==	Equal to
	!=	Not equal to
	===	Strictly equal to
	!==	Strictly Not Equal to
	>	Greater than
	>=	Greater than or equal to
	<	Less than
	<=	Less than or equal to
String Operators	+	String concatenation
Bit Manipulation Operators	&	And
		Or
	^	Exclusive Or
	<<	Left Shift
	>>	Right Shift
	>>>	Zero-fill right shift
Assignment Operators:	=	Equal to
Ternary Operators:	?:condition ? value1 : value2	This operator is ternary operator since it takes three operators—a condition to be evaluated and two alternative values to be returned based on the truth and falsity of the condition.

Special Operators	,	The comma operator
	The delete operator	The delete operator–used to delete property of an object or an element at an array index.
	The new operator	It is used to create instance of an object type
	The typeof operator–	It returns a string value that identifies the type of an operand.
	The void operator–	It does not return a value. It is typically used with the JavaScript.

Operator Precedence

- The precedence of the operator determines which operations are evaluated before others during the parsing and execution of complex expression.

Precedence	Operator
1	parenthesis (), function call or array subscript[]
2	!, ~, - (unary), ++, --, typeof, new, void, delete
3	*, /, %
4	+, -(binary)
5	<<, >>, >>>
6	<, <=, >, >=
7	==, !=, ===, !==
8	&
9	^
10	
11	&&
12	
13	?:
14	=, +=, -=, *=, /=, %=, <<=, >>=, >>>=, &=, ^=, =
15	comma (,) operator

- For a precedence level, precedence is Left to Right.

Document Object Model

A Document object represents the HTML document that is displayed in that window. The Document object has various properties that refer to other objects which allow access to and modification of document content.

The way document content is accessed and modified is called the Document Object Model, or DOM. The Objects are organized in a hierarchy. This hierarchical structure applies to the organization of objects in a Web document.

- **Window object** – Top of the hierarchy. It is the outmost element of the object hierarchy.
- **Document object** – Each HTML document that gets loaded into a window becomes a document object. The document contains the contents of the page.

- **Form object** – Everything enclosed in the <form>...</form> tags sets the form object.
- **Form control elements** – The form object contains all the elements defined for that object such as text fields, buttons, radio buttons, and checkboxes.

Here is a simple hierarchy of a few important objects –

HTML DOM

There are several DOMs in existence. The following sections explain each of these DOMs in detail and describe how you can use them to access and modify document content.

The Legacy DOM – This is the model which was introduced in early versions of JavaScript language. It is well supported by all browsers, but allows access only to certain key portions of documents, such as forms, form elements, and images.

The W3C DOM – This document object model allows access and modification of all document content and is standardized by the World Wide Web Consortium (W3C). This model is supported by almost all the modern browsers.

The IE4 DOM – This document object model was introduced in Version 4 of Microsoft's Internet Explorer browser. IE 5 and later versions include support for most basic W3C DOM features.

DOM compatibility

If you want to write a script with the flexibility to use either W3C DOM or IE 4 DOM depending on their availability, then you can use a capability-testing approach that first checks for the existence of a method or property to determine whether the browser has the capability you desire. For example –

```
if (document.getElementById) {  
    // If the W3C method exists, use it  
} else if (document.all) {  
    // If the all[] array exists, use it  
} else {  
    // Otherwise use the legacy DOM  
}
```

Object Types and Instances

An object type is a template from which specific objects of that type are created. It defines properties and methods that are common to all objects of that type. New instances are created using the object template. Each instance has its own set of properties and methods to be applied.

Creating Instances of Objects

Instances of objects of specific type are created using new operator.

```
variable = new objectType(parameters)
```

Example:

```
currentDate=new Date()
```

```
myBday= new Date(1973,3,2)
```

Other Predefined Objects

- Math Object
- String Object
- Array Object
- Date Object
- Function Object

Each object can have Properties, Methods and Events attached with it. Properties defines behavior of the object like color, size etc., methods defines actions taken by the object and events are particular Instance to perform a task like click, dblClick, mouseMove, mouseDown etc.

Using Properties

```
objectName.propertyName
```

Using Methods

```
objectName.methodName(arguments)
```

Example:

```
document.write(document.forms(1).element[4].name);  
document.write("Welcome to JavaScript");
```

With Statement

It is provided as a convenience to eliminate retyping the name of an object that is to be referenced in a series of property references and method invocation. Syntax is

```
with (variableName or object Name)  
{  
statements;  
}
```

Example:

```
with (document)  
{  
write("<h1>Welcome to World of JavaScript</h1>");  
write("<hr><font color='red'>We want to learn its</font>");  
}
```

Creating Functions, objects in JavaScript

Functions in JavaScript

Function is a set of statement defined under a single name. Functions are created using function command.

Syntax:

```
function name()
{
statement;
statement;
statement;
}
```

Example:

```
function myfun()
{
a=10;
b=20;
c=a+b;
document.write("the sum is "+c);
}
```

Functions can be called from within other functions.

Example:

```
function sayGoodbye()
{
alert("Goodbye!")
}
function sayHello()
{
alert("Hi, there!");
sayGoodbye()
}
```

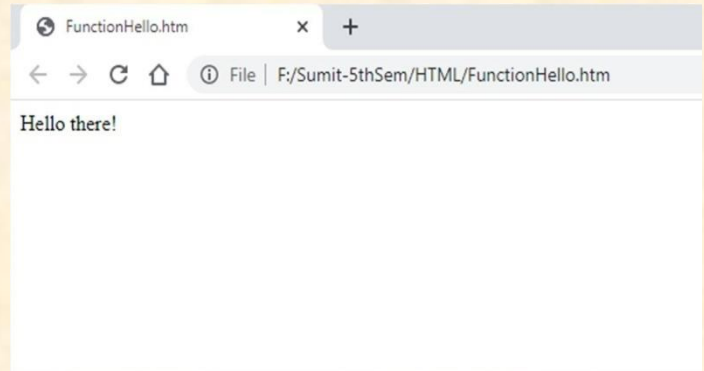
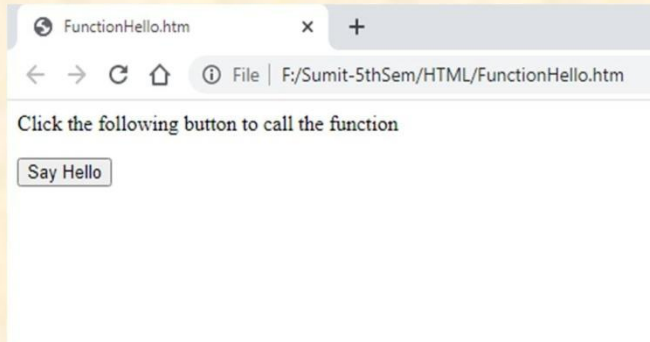
Calling Function :

Example:

```
<html>
  <head>
    <script>
      function sayHello()
      {
        document.write ("Hello there!");
      }
    </script>
```

```
</head>
<body>
<p>Click the following button to call the function</p>
<Button type="button" onclick="sayHello()" id="sh"> Say Hello</button>
</body>
</html>
```

Output :



Passing Parameters to Functions

Some functions perform a simple task for which no extra information is needed. However, it is often necessary to supply information to a function so that it can carry out its task. To do this we would pass the price into the function as a parameter. Parameters are listed in between the brackets that follow the function name.

```
function name(parameter_1, parameter_2)
{
statement(s);
}
```

In this case two parameters are used, but it's possible to use more than this if necessary. The additional parameter names would simply be added on to the list of parameters inside the brackets, separated from one another by commas.

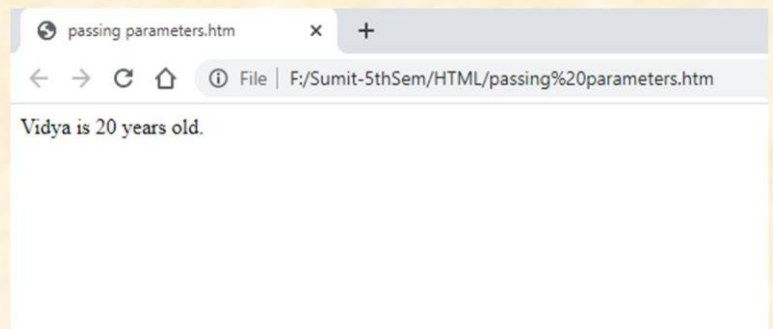
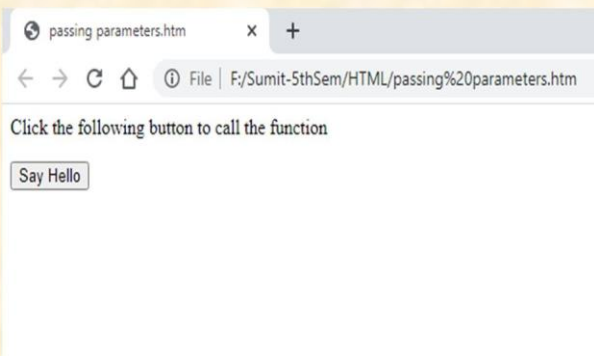
```
function addVAT(price)
{
price *= 1.21;
alert(price)
}
```

Example:

```
<html>
<head>
<script>
function sayHello(name, age)
{
    document.write (name + " is " + age + " years old.");
}
```

```
</script>
</head>
<body>
<p>Click the following button to call the function</p>
<button type="button" onclick="sayHello('Vidhya', 20)" id="sh">
Say Hello
</button>
</body>
</html>
```

Output :



JavaScript Object

Creating Objects In Javascript

An empty object, we can also say the empty associative array is create following two different syntax's,

```
var myObj = new Object();
var myObj = { }           // both are same
```

Assigning objects In Javascript

You can store data by keys and value pairs. We can use "dot" property notation for assigning, accessing, and deleting Object values.

```
var myObj = {};           // indicates an object is created
myObj.commercial = ".com";
myObj.network = ".net";
myObj.organization = ".org";
myObj.government = ".gov";
```

You can also assign values to multiple properties at the time of creating an object { key1: value1, key2: value2, ... }.

```
var myObj = {
  "commercial": ".com",
  "network": ".net",
```

```
"organization": ".org",  
"government": ".gov"  
};
```

Accessing Objects In Javascript

You can use either square brackets or dot notation to access object properties.

```
<!DOCTYPE html>  
<html lang="en">  
<head>  
  <meta charset="UTF-8">  
  <title>JavaScript object creating accessing</title>  
</head>  
<body>  
<pre>  <!-- Use pre element for work document.writeln() method -->  
<script type="text/javascript">  
  
  var myObj = {};          // indicates an Object is created  
  myObj.commercial = ".com";  
  myObj.network= ".net";  
  myObj.organization= ".org";  
  myObj.government= ".gov";  
  
  document.writeln("commercial domain: " + myObj['commercial']);  
                                     // Outputs .com  
  document.writeln("commercial domain: " + myObj.commercial);  
                                     // Outputs .com  
  
</script>  
</pre>  
</body>  
</html>
```

Output :

```
commercial domain: .com  
commercial domain: .com
```

Deleting object s In Javascript

delete statement is preferable for removing properties,.

```
<!DOCTYPE html>  
<html lang="en">  
<head>  
  <meta charset="UTF-8">  
  <title>JavaScript object deleting</title>  
</head>  
<body>  
<pre>  <!-- Use pre element for work document.writeln() method -->
```

```

<script type="text/javascript">
  // Press F12 - View result on Console Panel

  var myObj = {}; // indicates an Object is created
  myObj.commercial = ".com";
  myObj.network = ".net";
  myObj.organization = ".org";
  myObj.government = ".gov";

  document.writeln("commercial domain: " + myObj['commercial']);
  // Outputs .com
  document.writeln("commercial domain: " + myObj.commercial);
  // Outputs .com

  delete myObj.commercial;

  console.log("After removing : ", myObj);
</script>
</pre>
</body>
</html>

```

Output :

```

commercial domain: .com
commercial domain: .com

```

Object Properties and methods In Javascript

Object can be stored not only variables but also store data structures (associative array, set, union), and functions.

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>JavaScript object properties and method</title>
</head>
<body>
<pre>  <!-- Use pre element for work document.writeln() method -->
<script type="text/javascript">
  var employee= {
    name: "Rohan Mishra",
    salary: function() {
      this.salary= 15000;
      this.allowance = 1300;
    },
    information: function() {
      document.writeln("Employee name: " + employee.name);
      document.writeln("Employee Salary: " + this.salary);
      document.writeln("Employee Allowance: " + this.allowance );
    }
  }
}

```

```
        employee.salary()  
        employee.information()  
</script>  
</pre>  
</body>  
</html>
```

Output :

Employee name: Rohan Mishra
Employee Salary: 15000
Employee Allowance: 1300

New keyword to create user-define object In Javascript

JavaScript new keyword creates an instance of a user-defined object type as well as built-in object types that has a constructor function of the object.

Syntax:

```
new constructor([([arguments])])
```

Example:

```
<!DOCTYPE html>  
<html lang="en">  
<head>  
    <meta charset="UTF-8">  
    <title>JavaScript new keyword to create user-define object</title>  
</head>  
<body>  
<pre>    <!-- Use pre element for work document.writeln() method -->  
<script type="text/javascript">  
    function Employee(name, salary) {  
        this.name = name  
        this.salary= 15000;  
    }  
  
    var employee = new Employee("Opal Kole", "15000")  
  
    document.writeln("Employee Name: " + employee.name);  
    document.writeln("Employee Salary: " + employee.salary);  
</script>  
</pre>  
</body>  
</html>
```

Output :

Employee Name: Opal Kole

Employee Salary: 15000

Event handling in JavaScript

Events describe actions that occur as the result of user interaction with a web page or other browser the form those events. The processing is performed in response to the occurrence of an event is known as **event handling**. The code that performs this processing is called an **event handler**.

Functions of Event Handling

- Event Handling identifies where an event should be forwarded.
- It makes the forward event.
- It receives the forwarded event.
- It takes some kind of appropriate action in response, such as writing to a log, sending an error or recovery routine or sending a message.
- The event handler may ultimately forward the event to an event consumer.

How JavaScript Handles Event

JavaScript's approach to event handling is a two-step process:

1. Defining the events that can be handled by scripts.
2. Providing a standard method of connecting these events to user-supplied JavaScript code.

JavaScript defines events for most of the major objects found in web pages including links, images, image maps, form elements and windows. The JavaScript defines special attributes for the tags corresponding to use HTML elements that permit the script to identify event-handling JavaScript code instead of the default HTML event handlers. The events defined by JavaScript that are common to Navigator and Internet Explorer is given below:

HTML Elements	HTML Tags	JavaScript Event
All elements	various tags	mouseMove
Link	<a>...	click, double Click, mouseDown. mouseUp, mouseOver, mouseOut, keyDown, keyUp, keyPress
Image		abort, error. load, keyDown, keyUp, keyPress
Area	<area>	mouseOver, mouseOut, db Click

document body	<code><body>.</body></code>	load, unLoad
window, frameset, frame	<code><body>,<frame>, <frameset></code>	blur, cursor, focus, load, unload, movie, resize, drag Drop
Form	<code><form></code>	submit, reset
text field	<code><input type="text"></code>	blur, focus, change, select
password field	<code><input type="password"></code>	blur, focus
text area	<code><textarea></code>	blur, focus, change, select, keyDown, key Up, keyPress
Button	<code><input type="button"></code>	click, blur, focus, mouseDown, mouseUp
Submit	<code><input type="submit"></code>	click, blur, focus
Reset	<code><input type="reset"></code>	click, blur, focus
Radio button	<code><input type="radio"></code>	click, blur, focus
checkbox	<code><input type="checkbox"></code>	click, blur, focus
selection	<code><select>..</select></code>	blur, focus, change

Some new attributes are added with HTML tags to use events. On is added before the event name to * air event, e.g. onClick, onLoad, onDbIClick, onMouseOver etc.

Event Handling Attributes

onAbort	The loading of an image is aborted as the result of a user action
onBlur	A document, window, frame set or form element loses the current focus
onChange	A text field, text area r selection is modified and loses the current focus
onClick	A link, image map or form element is clicked map or form
onDbLClick	A link, image map or form element is double clicked
onError	An error occurs during the loading of an image, window or frame
onFocus	A document, window, frame set or form element receives the current input focus

onKeyDown	The user presses a key
onKeyPress	The user presses a key and releases a key
onKeyUp	The user releases a key
onLoad	An image, document or frame set is loaded
onMouseDown	The user presses a mouse button
onMouseMove	The user moves the mouse
onMouseOut	The mouse is moved out of a link or an area of a client-side image map
onMouseOver	The mouse is moved over a link or an area of a client-side image map
onMouseUp	The user releases a mouse button
onMove	The user moves a window or frame
onReset	A user resets a form by clicking on form reset button
onResize	The user resizes a window or frame
onSelect	Text is selected in a text field or text area
onSubmit	A form is submitted
onUnload	The user exits a document or frameset

Embedding JavaScript with HTML

JavaScript Meets the HTML Page

JavaScript works with browsers by embedding code directly into an HTML page. Netscape added a new generic tag called SCRIPT to recognize scripting languages. To inform the browser that your code is JavaScript, you must add the attribute of LANGUAGE="JavaScript" to the SCRIPT tag. Much of your JavaScript coding is enclosed within these tags, as you can see in the following example:

```
<script language="JavaScript">
a = "Hello!"
//...set a variable called 'a' to a value of "Hello!"</script>
```

HTML Objects in JavaScript

Buttons, Checkboxes and Radio Buttons We will examine how JavaScript builds special objects as it reads your HTML code and explore how to use these objects to create an interactive form. The form Object Whenever you declare a `<form>...</form>` in your web page, JavaScript takes the information contained within that form and builds a form object.

Example: form Attributes

```
< form
  name = "formname"
  target = "window name"
  action = "serverURL"
  method = GET|POST
  enctype = "encodingType"
  [onSubmit="handler Text" ]>
</form>
```

JavaScript places each form it finds in a page in an array called forms, we can access each form by either its index in this array (with the first form at 0, the next at 1, and so on) or by its name. For example,

```
<form name="this form">
<input type=text name="input1" value="hi there!" >
</form>
```

By doing this you create an object called this form which also exists as `document.forms[0]`. We can access individual fields in this form—where you can either read that field directly or send any value to that field—which might (depending on the type of input field) dynamically change on the screen. To access information about this form, use the following naming conventions

- `formName.action`— From the ACTION argument
- `formName.elements`— To get to the input (etc.) fields
- `formName.encoding`— From the ENCTYPE argument
- `formName.method`— From the METHOD argument
- `formName.target`— From the TARGET argument

Example : Accessing a form Element in JavaScript

```
<html>
  <head>
    <script language="javascript">
<form name="thisForm"
  action="foo.html"
  ENCTYPE="text/ascii"
  method="GET">
</form>
<!-- hide this code
document.forms[0].method = "POST"
document.write(document.thisForm.method)
-->
```

```
</script>
</head>
</html>
```

Output :

POST

The button Object: The button object is a new type of INPUT tag that allows you to create general purpose buttons in a form. You can use these buttons to activate any function, or to open a new URL, or perform any other action that JavaScript can initiate. To define a button object use the following

Syntax :

```
<input
type="button"
name= "nameInside"
value = "nameOnButton"
[onClick="handler Text"]>
```

We access a button's properties and methods in the following way:

- `buttonName.property Name`
- `buttonName.methodName(parameters)`
- `formName.elements[index].property Name`
- `formName.elements[index].methodName(parameters)`

Example: Using a Button form Element in JavaScript

```
<html>
  <head>
    <script language="JavaScript">
      <!-- hide me from other browsers
      function hey()
      {
        document.alert("Hey there! You pushed the button!")
      }
      // end hiding from other browsers ->
    </script>
  </head>
  <body>
    <form>
      <h1>Click below for an alert</h1>
      <input type="button" name="alert button" value="Click Me!"
      onClick="hey() ">
    </form>
  </body>
</html>
```

The checkbox Object : The checkbox object is created inside a form and appears as a small box with or without a mark (usually an x) inside it. Think of a checkbox as an On/Off switch.

The radio Object: This object allows a user to make a choice of a single selection from many. Usually this is a list of related items from which you wish the user to pick only one choice. The radio object is very similar to the checkbox object, except that a series of radio objects with the same NAME attribute toggle all of the radio buttons off except for the one that was picked. In other words, when you click one radio button in a group of related buttons, all of them will be off except for the one you clicked.

Example : radio Object Syntax

```
<input  
type="radio"  
name="radioName"  
[CHECKED]  
(onClick = "handlerText")>  
textToDisplay
```

Working with Cookies

What are Cookies?

Web Browsers and Servers use HTTP protocol to communicate and HTTP is a stateless protocol. But for a commercial website, it is required to maintain session information among different pages. For example, one user registration ends after completing many pages. But how to maintain users' session information across all the web pages. In many situations, using cookies is the most efficient method of remembering and tracking preferences, purchases, commissions, and other information required for better visitor experience or site statistics.

How It Works?

Your server sends some data to the visitor's browser in the form of a cookie. The browser may accept the cookie. If it does, it is stored as a **plain text record** on the visitor's hard drive. Now, when the visitor arrives at another page on your site, the browser sends the same cookie to the server for retrieval. Once retrieved, your server knows/remembers what was stored earlier.

```
<html>  
  <head>  
    <script>  
function clicked()  
  {  
    console.log("Function Clicked");  
  }  
  
    </script>  
  </head>  
  <body>  
    <form name="myform" action="">
```

```
<p> Enter Cookie Info:</p>
<input type="text" name= "name"/>
<input type="button" value="submit" onclick="clicked()"/>
</form>
</body>
</html>
```

Cookies are a plain text data record of 5 variable -length fields

- **Expires** The date the cookie will expire. If this is blank, the cookie will expire when the visitor quits the browser.
- **Domain** The domain name of your site.
- **Path** The path to the directory or web page that set the cookie. This may be blank if you want to retrieve the cookie from any directory or page.
- **Secure** If this field contains the word "secure", then the cookie may only be retrieved with a secure server. If this field is blank, no such restriction exists.
- **Name=Value** Cookies are set and retrieved in the form of key-value pairs

Cookies were originally designed for CGI programming. The data contained in a cookie is automatically transmitted between the web browser and the web server, so CGI scripts on the server can read and write cookie values that are stored on the client. JavaScript can also manipulate cookies using the cookie property of the Document object. JavaScript can read, create, modify, and delete the cookies that apply to the current web page.

Storing Cookies : The simplest way to create a cookie is to assign a string value to the document.cookie object, which looks like this.

```
document.cookie = "key1=value1; key2=value2;expires=date";
```

Here the expires attribute is optional. If you provide this attribute with a valid date or time, then the cookie will expire on a given date or time and thereafter, the cookies' value will not be accessible.

Note: Cookie values may not include semicolons, commas, or whitespace. For this reason, you may want to use the JavaScript escape() function to encode the value before storing it in the cookie. If you do this, you will also have to use the corresponding unescape() function when you read the cookie value.

Example:

```
<html>
  <head>
    <script>
function clicked()
{
  console.log("Function Clicked");
  var value =escape(document.myForm.name.value+";");
}
```

```

    console.log(value);
    document.cookie="name="+ value;
  }

</script>
</head>
<body>
<form name="myForm" action="">
<p> Enter Cookie Info:</p>
<input type="text" name="name"/>
<input type="button" value="submit" onclick="clicked()"/>
</form>
</body>
</html>

```

Output:

Now your machine has a cookie called name. You can set multiple cookies using multiple key=value pairs separated by comma.

Reading Cookies : Reading a cookie is just as simple as writing one, because the value of the document.cookie object is the cookie. So you can use this string whenever you want to access the cookie. The document.cookie string will keep a list of name=value pairs separated by semicolons, where name is the name of a cookie and value is its string value. We can use strings split() function to break a string into key and values as follows

Retrieving Cookies :

Example:

```

<html>
  <head>
    <script>
function clicked()
{
  console.log("Function Clicked");
  var value =escape(document.myForm.name.value+";");
  console.log(value);
  document.cookie="name="+ value;
  var allcookies = unescape(document.cookie);
  console.log(allcookies);
}

    </script>
  </head>
  <body>
    <form name="myForm" action="">
      <p> Enter Cookie Info:</p>
      <input type="text" name="name"/>
      <input type="button" value="submit" onclick="clicked()"/>
    </form>
  </body>
</html>

```

Setting Cookies Expiry Date

You can extend the life of a cookie beyond the current browser session by setting an expiration date and saving the expiry date within the cookie. This can be done by setting the 'expires' attribute to a date and time.

Example: It illustrates how to extend the expiry date of a cookie by 1 Month.

```
<html>
  <head>
    <script>
function clicked()
{
  console.log("Function Clicked");
  var value =escape(document.myForm.name.value+");");
  console.log(value);
  //document.cookie="name="+ value;

  // var allcookies = unescape(document.cookie);
  // console.log(allcookies);

  var now =new Date();
  now.setMonth(now.getMonth()+1);
  document.cookie="name=" + value;
  document.cookie="expire=" +now.toUTCString()+";";
  var allcookies = unescape(document.cookie);
  console.log(allcookies);
}

    </script>
  </head>
  <body>
    <form name="myForm" action="">
    <p> Enter Cookie Info:</p>
    <input type="text" name= "name"/>
    <input type="button" value="submit" onclick="clicked()"/>
    </form>
  </body>
</html>
```

Deleting a Cookie

Sometimes you will want to delete a cookie so that subsequent attempts to read the cookie return nothing. To do this, you just need to set the expiry date to a time in the past.

Example:

It illustrates how to delete a cookie by setting its expiry date to one month behind the current date.

```
<html>
  <head>
    <script>
function clicked()
```

```

{
console.log("Function Clicked");
var value =escape(document.myForm.name.value+";");
console.log(value);
//document.cookie="name="+ value;

// var allcookies = unescape(document.cookie);
// console.log(allcookies);

    var now =new Date();
    now.setMonth(now.getMonth()+1);
    document.cookie="name=" + value;
    document.cookie="expire=" +now.toUTCString()+"";

    var allcookies = unescape(document.cookie);
    console.log(allcookies);

    var now =new Date();
    now.setMonth(now.getMonth()-1);
    document.cookie="name=" + value;
    document.cookie="expire=" +now.toUTCString()+"";

    var allcookies = unescape(document.cookie);
    console.log(allcookies);
}

</script>
</head>
<body>
<form name="myForm" action="">
<p> Enter Cookie Info:</p>
<input type="text" name= "name"/>
<input type="button" value="submit" onclick="clicked()"/>
</form>
</body>
</html>

```

The server writes this code to your machine and will reread it when you visit again. Although this feature is very useful, there are still debates as to its security and validity of use.

The hidden Object: Often when you create an interactive form, you want to keep some information hidden, yet still pass this information on to the server when the form is submitted. This is often information about the user—perhaps when he last accessed your page, or some preference that he had set in the previous form that generated this one. You can keep track of this information with hidden fields. This is often used in place of Netscape "cookies" for compatibility to browsers that do not support the This specification. Hidden fields contain text information and are not displayed on the screen with the rest of the form. To create a hidden object, use the following syntax:

```
<input
```

```
type="hidden"
name="hiddenName"
(value = "text Value")>
```

We can access the properties of this object by using the following:

- hidden Name.propertyName
- formName.elements[index].propertyName

For example, the following returns "hi there!":

```
<form>
<input type=hidden name=hideme value="hi there!">
</form>
document.write(hideme.value);
```

The password Object: The password input field in a form is very useful for those times when you need to create a logon screen and keep the password of that logon hidden from view or from being printed by some malicious user who sees the login screen. Any time you want the information hidden from sight as a user types that information on the screen, use the password input field. To create this as an object in JavaScript, use the following syntax:

```
<input
type="password"
name = "passwordName"
size=integer
[value = "text Value"]>
```

Accessing the properties and methods of this object once it is created is identical to previous examples and is as follows:

- passwordName.propertyName
- password Name.methodName(parameters)
- formname.elements[index].propertyname
- formname.elements[index].methodname(parameters)

The password object is the focus, blur, and select event handlers as methods. Let's say we want to check the validity of a password before a user actually sends it back to the server for a login. We can create a function that checks the password and notifies the user if he entered an invalid password.

Example : Form Validation Example

```
<script language="JavaScript">
function checkPass(mypass) {
if not pass) {
alert ("You have entered an invalid password. Try again.")
mypass.focus()
```

```

mypass.select()
}
function not
<form onSubmit="checkPass(this)">
<input type="password" name="mypass">
</form>

```

The text Object: The text input field in an HTML document is your workhorse for inputting many types of information. Most other types of input in forms are derivations of this kind of input

Connecting database using JavaScript in HTML Page

You cannot connect directly from a browser to the database as the database does not have access. Even if you could, you should not: if your java script sends SQL statements to the database, there is nothing stopping a random visitor from having fun by changing your SQL to "drop table", and your entire data base is gone.

What you do is setting up a server side web service (using PHP, Java servlets or whatever) that accepts a URL, converts it to a DB query and returns the result as JSON. The primary thing from a security standpoint is that you never insert a string from the web into the SQL without validating it first.

Example:

```

--browser javascript (using jQuery)
-----
$.getJSON ("/product Detail.json ?
product Id = 123", function (data)
{
//do something with the data });
---- server (using java servlet) --
void
do productDetail (HttpServletRequest req, HttpServletResponse res) {
int productId = Integer.parseInt(req.getParameter("productId"));
//get product ID from query param string sql = "select * from product where productId = " + productId;
//generate the SQL
string json = generateSQL(sql);
//execute query
convert result to json
res.write (json); //send to browser
}

```

Example:

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title></title>
<script type="text/javascript" >
function loadDB()
{
var connection = new ActiveXObject("ADODB.Connection");

```

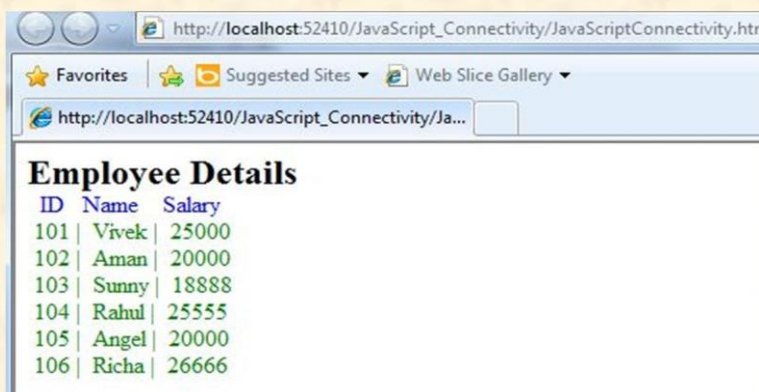
```

        var connectionString = "Data Source=.;Initial Catalog=EmpDetail;Persist Security Info=True;User ID=sa;Password=Micr0s0ft;Provider=SQLOLEDB";
        connection.Open(connectionString);
        var rs = new ActiveXObject("ADODB.Recordset");
        rs.Open("select * from emp", connection);
        rs.MoveFirst();
        var span = document.createElement("span");
        span.style.color = "Blue";
        span.innerHTML = "ID "+" Name "+" Salary";
        document.body.appendChild(span);
        while (!rs.eof)
        {
            var span = document.createElement("span");
            span.style.color = "green";
            span.innerHTML = "\n " + rs.fields(0) + " | " + rs.fields(1) + "
| " + rs.fields(2);
            document.body.appendChild(span);
            rs.MoveNext();
        }

        rs.close();
        connection.close();
    }
</script>
<style type="text/css">
    #main {
        height: 264px;
    }
</style>
</head>
<body onload="loadDB()">
    <div id="show" style="font-size: x-large; font-weight: bold">
        Employee Details</div>
</body>
</html>

```

Output:



ID	Name	Salary
101	Vivek	25000
102	Aman	20000
103	Sunny	18888
104	Rahul	25555
105	Angel	20000
106	Richa	26666

Working with Browser, validating and submitting Forms

Working with Browser Window

This linkage is provided by JavaScript's extremely rich set of browser and HTML objects. The browser objects are a reflection of the browser environment, and include objects that can be used to reference the current page, the history list, and the current URL. There are also methods for opening new windows, putting up dialog boxes, and writing HTML directly. We have already been leaning heavily on one such method, the write method of the document object. The browser (or navigator) objects are at the top of JavaScript object hierarchy, since they represent overall information and actions that are not necessarily associated with a particular web page. Within a given web page, however, each HTML element has a corresponding object, an HTML objects, within the object hierarchy. In particular, every HTML form, and every HTML element within every form, has a corresponding object.

Browser Objects

The primary browser objects, in rough order of significance, are as follows: location

- ❖ window
- ❖ location
- ❖ document
- ❖ history

The window Object

The window object is the top object in the JavaScript object hierarchy. Every browser window that is currently open will have a corresponding window object. All the other objects are children of one of the window objects. In particular, every window is associated with a particular web page, and the HTML structure of this page is reflected in the window document object. Every window corresponds to some URL: that URL is reflected in the location object. Every window has a history of the previous pages that have been displayed in that window, which are represented by the various properties of the history object.

JavaScript maintains an idea of the current window, so that almost all references to sub-objects of the current window do not need to refer to it explicitly. This is why all of our output has been done using document.write() rather than window.document.write(). Window objects have the following interesting methods (among others):

- ❖ alert(msgstr)
- ❖ confirm (msgstr)
- ❖ prompt(msgstr)
- ❖ close()
- ❖ open(urlstr, wname)

All these methods are used to manipulate the window state of the browser itself. The alert and confirm methods are used to display their msgstr argument in a dialog box. The alert method is used to alert the user to something about which the user can do nothing. An alert dialog box contains a single OK button. The confirm dialog box is more flexible, and displays its message with both an OK and a Cancel button. If the user selects OK then the confirm method returns true, otherwise it returns false.

The prompt method is used to solicit user input, in the form of a string. It displays a dialog box with the msgstr and an editable text field. This method also accepts a second optional argument that can be used to set a default value in the input field. This method returns whatever the user typed as a string. You use the open method of the window object when you wish to open a new browser window. The urlstr argument is a string representing the URL that will be loaded into that window, The wname argument is a string that gives the new window its name. This method returns an instance of the window object representing the

new window created. This method also accepts a third argument that can be used to specify a wide variety of display options for the new window (such as whether or not it should blink its toolbar). When the `close` method is invoked from a window instance the underlying window is closed and the URL in it is unloaded.

The document Object

Every window is associated with a document object. The document object contains properties for every anchor, link, and form on that page, as well as all of the sub-elements of those elements. It also contains properties for its title (the content of the `<title>` field of the page), its foreground color (the `fgColor` property), its background color (the `bgColor` property), its various link colors, and other attributes of the page itself. The document object has the following methods:

- ❖ `clear()`
- ❖ `close()`
- ❖ `open()`
- ❖ `write (str)`
- ❖ `writeln (str)`

The `clear` method is used to completely erase a document window. The entire contents are wiped out, regardless of how they got there. The `clear` method is particularly useful if you are constructing a web page entirely within JavaScript, and want to make sure it is empty before you start. The `open` and `close` methods are used to start and stop buffered output. If you call the `open` method, perform a series of `writes` and/or `writelines`, and then call the `close` method, the results of your write operations are used out and appear on the page.

The history and location Objects

The history object is used to refer to the history list of previously visited URLs. The history object has a property known as `length`, which indicates how many URLs are stored on the history list at present. It also has the following three methods:

- ❖ `back()`
- ❖ `forward()`
- ❖ `go (where)`

The `go` method is used to navigate the history list. The `where` argument can be a number or a string. If the `where` argument is a number then it indicates how far we wish to move in the history list. Positive number means that we wish to move that many documents forward in this history list, while a negative number is used to move backward. Thus, `go(5)` has the same effect as using the Forward button five times, while `go(-1)` would be the same as clicking the Back button once. If `where` is a string representing a URL then that URL is loaded, and becomes the current document.

The location object describes the URL of a document. It has properties representing the various components of the URL, including its protocol part, its hostname part, its pathname part, and its port number part, among other properties. Unfortunately, these properties are often null, at least in the UNIX version of Netscape Navigator 2.0. It also has a `toString` method which can be used to convert to a string. We can use the following code to display a formatted message giving the current URL:

```
var loc = document.location;  
document.write("<br />URL is" + loc.toString());
```

```
document.write("<br />");
```

Validating and Submitting a Form

This section covers the final pieces of information you need to complete your exploration of JavaScript and forms. The last two form-based objects `submit` and `reset`, are accompanied with an example of a simple mail-in form that checks the input before it is sent back to you.

The Submit Object: The submit button was originally intended in HTML to be the final button a JavaScript, you can now use this button to also send all of the information collected in a form to another window on your browser, or to the same window itself, which causes the contents of the window to would click to send a form back to the server. It would submit information, send feedback, or present a structured request for new information (you see this in search engines like Yahoo!). With change in some way. An example of this is changing the background color of the window based on the user's preference on a form. You create a submit object by using the following syntax

```
<input  
type="submit"  
name="submitName"  
value="button Text"  
[onClick="handler Text"]>
```

You access this object's properties and methods by the following syntax:

- ❖ `submitName.property Name`
- ❖ `submitName.methodName(parameters)`
- ❖ `formName.elements[index].propertyName`
- ❖ `formName.elements[index].methodName(parameters)`

When you click a submit button, it always loads a new page—even if that page is the same page you were already on. This is useful in that you can use a form to change attributes of the current page and see them change when you submit the form. The submit object uses the `onClick` event handler and can be clicked by using the `submitName.click` method.

The reset Object: The reset button allows a user to completely reset a form's input fields to their defaults. You create a reset object in JavaScript by using the following syntax:

```
<input  
type="reset"  
name="resetName"  
value="button Text"  
[onClick ="handler Text"]>
```

To access its methods and properties, you use the same familiar syntax, as follows:

- ❖ `resetName.property Name`
- ❖ `resetName.methodName(parameters)`
- ❖ `formName.elements[index].property Name`
- ❖ `formName.elements[index].methodName(parameters)`

The reset button uses the same `onClick` event handler and click method as the submit object.

Example : Form Input Validation Example

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<title>JavaScript Form validation</title>
<style>
body {
    font-size: 16px;
    background: #f9f9f9;
    font-family: "Segoe UI", "Helvetica Neue", Arial, sans-serif;
}
h2 {
    text-align: center;
    text-decoration: underline;
}
form {
    width: 300px;
    background: #fff;
    padding: 15px 40px 40px;
    border: 1px solid #ccc;
    margin: 50px auto 0;
    border-radius: 5px;
}
label {
    display: block;
    margin-bottom: 5px
}
label i {
    color: #999;
    font-size: 80%;
}
input, select {
    border: 1px solid #ccc;
    padding: 10px;
    display: block;
    width: 100%;
    box-sizing: border-box;
    border-radius: 2px;
}
.row {
    padding-bottom: 10px;
}
.form-inline {
    border: 1px solid #ccc;
    padding: 8px 10px 4px;
    border-radius: 2px;
}
.form-inline label, .form-inline input {
    display: inline-block;
    width: auto;
    padding-right: 15px;
}
.error {
    color: red;
    font-size: 90%;
}
input[type="submit"] {
    font-size: 110%;
    font-weight: 100;
    background: #006dcc;
```

```

border-color: #016BC1;
box-shadow: 0 3px 0 #0165b6;
color: #fff;
margin-top: 10px;
cursor: pointer;
}
input[type="submit"]:hover {
    background: #0165b6;
}
</style>
<script>
// Defining a function to display error message
function printError(elemId, hintMsg) {
    document.getElementById(elemId).innerHTML = hintMsg;
}

// Defining a function to validate form
function validateForm() {
    // Retrieving the values of form elements
    var name = document.contactForm.name.value;
    var email = document.contactForm.email.value;
    var mobile = document.contactForm.mobile.value;
    var country = document.contactForm.country.value;
    var gender = document.contactForm.gender.value;
    var hobbies = [];
    var checkboxes = document.getElementsByName("hobbies[]");
    for(var i=0; i < checkboxes.length; i++) {
        if(checkboxes[i].checked) {
            // Populate hobbies array with selected values
            hobbies.push(checkboxes[i].value);
        }
    }

    // Defining error variables with a default value
    var nameErr = emailErr = mobileErr = countryErr = genderErr = true;

    // Validate name
    if(name == "") {
        printError("nameErr", "Please enter your name");
    } else {
        var regex = /^[a-zA-Z\s]+$/;
        if(regex.test(name) === false) {
            printError("nameErr", "Please enter a valid name");
        } else {
            printError("nameErr", "");
            nameErr = false;
        }
    }

    // Validate email address
    if(email == "") {
        printError("emailErr", "Please enter your email address");
    } else {
        // Regular expression for basic email validation
        var regex = /^\S+@\S+\.\S+$/;
        if(regex.test(email) === false) {
            printError("emailErr", "Please enter a valid email address");
        } else {
            printError("emailErr", "");
            emailErr = false;
        }
    }
}

```

```

    }
}

// Validate mobile number
if(mobile == "") {
    printError("mobileErr", "Please enter your mobile number");
} else {
    var regex = /^[1-9]\d{9}$/;
    if(regex.test(mobile) === false) {
        printError("mobileErr", "Please enter a valid 10 digit mobile number");
    } else {
        printError("mobileErr", "");
        mobileErr = false;
    }
}

// Validate country
if(country == "Select") {
    printError("countryErr", "Please select your country");
} else {
    printError("countryErr", "");
    countryErr = false;
}

// Validate gender
if(gender == "") {
    printError("genderErr", "Please select your gender");
} else {
    printError("genderErr", "");
    genderErr = false;
}

// Prevent the form from being submitted if there are any errors
if((nameErr || emailErr || mobileErr || countryErr || genderErr) == true) {
    return false;
} else {
    // Creating a string from input data for preview
    var dataPreview = "You've entered the following details: \n" +
        "Full Name: " + name + "\n" +
        "Email Address: " + email + "\n" +
        "Mobile Number: " + mobile + "\n" +
        "Country: " + country + "\n" +
        "Gender: " + gender + "\n";

    if(hobbies.length) {
        dataPreview += "Hobbies: " + hobbies.join(", ");
    }
    // Display input data in a dialog box before submitting the form
    alert(dataPreview);
}
};
</script>
</head>
<body>
<form name="contactForm" onsubmit="return validateForm()" action="/examples/actions/confirmatio
n.php" method="post">
    <h2>Application Form</h2>
    <div class="row">
        <label>Full Name</label>
        <input type="text" name="name">
        <div class="error" id="nameErr"></div>

```

```

</div>
<div class="row">
  <label>Email Address</label>
  <input type="text" name="email">
  <div class="error" id="emailErr"></div>
</div>
<div class="row">
  <label>Mobile Number</label>
  <input type="text" name="mobile" maxlength="10">
  <div class="error" id="mobileErr"></div>
</div>
<div class="row">
  <label>Country</label>
  <select name="country">
    <option>Select</option>
    <option>Australia</option>
    <option>India</option>
    <option>United States</option>
    <option>United Kingdom</option>
  </select>
  <div class="error" id="countryErr"></div>
</div>
<div class="row">
  <label>Gender</label>
  <div class="form-inline">
    <label><input type="radio" name="gender" value="male"> Male</label>
    <label><input type="radio" name="gender" value="female"> Female</label>
  </div>
  <div class="error" id="genderErr"></div>
</div>
<div class="row">
  <label>Hobbies <i>(Optional)</i></label>
  <div class="form-inline">
    <label><input type="checkbox" name="hobbies[]" value="sports"> Sports</label>
    <label><input type="checkbox" name="hobbies[]" value="movies"> Movies</label>
    <label><input type="checkbox" name="hobbies[]" value="music"> Music</label>
  </div>
</div>
<div class="row">
  <input type="submit" value="Submit">
</div>
</form>
</body>
</html>

```

Output :

Application Form

Full Name

Email Address

Mobile Number

Country

 ▼

Gender

☐ Male ☐ Female

Hobbies *(Optional)*

☐ Sports ☐ Movies ☐ Music

Submit

Activate Windows

Go to Settings to activate Windows.

EXAMPLES

Program 1: Basic Program of JS in html

```
<html>
  <body>
    <script>
      document.write("Hello World");
    </script>
  </body>
</html>
```

Program 2: Immediate Scripts

```
<html>
  <head> </head>
  <body>
    <script>
      // Script that produces content for the body
      document.write("Lets start JavaScript")
    </script>
  </body>
</html>
```

Program 3: Deferred Scripts

```
<html>
  <head>
    <script>
      // Script that Initializes items for user-driven actions
    </script>
  </head>
  <body>

  </body>
</html>
```

Program 4: Hybrid Scripts

```
<html>
<head>
  <script language="JavaScript">
    var hello = "Welcome to JavaScript";
  </script> <noscript>
    Sorry! Your browsers does not support
  </noscript>
</head>

<body>
  <script language="javaScript"> document.write(hello);
  </script>
</body>
</html>
```

Program 5: Variables

```
<!DOCTYPE html>
<html lang="en">
```

```

<head>
  <script>
    var christianName = "Fred";
    var surname = "Jones";
    var age = 37;
    var married = false;
    document.write(christianName);
    // document.write(surname);
    // document.write(age);
    // document.write(married);
  </script>
  <title>Document</title>
</head>
<body>

</body>
</html>

```

Program 6: eval() function

```

<!DOCTYPE html>
<html>
<body>
<p>Click the button to evaluate/execute JavaScript code/expressions.</p>
<button onclick="myFunction()">Try it</button>
<p id="demo"></p>
<script>
function myFunction() {
  var x = 10;
  var y = 20;
  var a = eval("x * y") + "<br>";
  var b = eval("2 + 2") + "<br>";
  var c = eval("x + 17") + "<br>";

  var res = a + b + c;
  document.getElementById("demo").innerHTML = res;
}
</script>
</body>
</html>

```

Program 7: parseInt()function

```

<!DOCTYPE html>
<html>
<body>
  <script>

    a = parseInt("100");
    document.write('parseInt("100") = ' +
    a + "<br>");

    // It returns a Integer until
    // it encounters Not a Number character
    b = parseInt("2018@helloandhi");
    document.write('parseInt("2018@helloandhi") = ' +
    b + "<br>");

    // It returns NaN on Non numeral character
    c = parseInt("helloandhi@2018");

```

```

document.write('parseInt("helloandhi@2018") = ' +
c + "<br>");

// It returns Integer value of a Floating point Number
d = parseInt("3.14");
document.write('parseInt("3.14") = ' + d + "<br>");

// It returns only first Number it encounters
e = parseInt("21 7 2018");
document.write('parseInt("21 7 2018") = ' +
e + "<br>");

</script>
</body>
</html>

```

Program 8: Built-in Functions

```

<html>
<head>
<title>Using Built-in Functions</title>
</head>
<body>
<h2>JavaScript Built-in Functions</h2>

<button onclick="myFunction()">Try it</button>

<p id="demo"></p>

<script>
function myFunction() {
var txt;
alert("Welcome to JavaScript");
var name = prompt("Please Enter your name", "");
if (confirm("Press a button!"))
{
alert("Dear "+name+ " Thanks for Using JavaScript");
}
else
{
alert("Try next Time");
}
document.getElementById("demo").innerHTML = txt;
}
</script>

</body>
</html>

```

Program 9: User – Defined Function

```

<html>
<head>
<title>Add program</title>
<script language="javascript">
function addNumbers()
{
var val1 = parseInt(document.getElementById("value1").value);
var val2 = parseInt(document.getElementById("value2").value);
var ansD = document.getElementById("answer");
ansD.value = val1 + val2;
}

```

```

        </script>
</head>
<body>
    value1 = <input type="text" id="value1" name="value1" value=""/>
    value2 = <input type="text" id="value2" name="value2" value=""/>
    <input type="button" name="Sumbit" value="Click here" onclick="javascript:addNumbers()"
/>
    Answer = <input type="text" id="answer" name="answer" value=""/>
</body>
</html>

```

Program 10: Arrays

```

<html>
<body>
<h2>JavaScript Arrays</h2>

<p id="demo"></p>

<script>
    var A = ["AB", "CD", "EF"];
    document.getElementById("demo").innerHTML = A[1];
</script>

</body>
</html>

```

Program 11: If statement

```

<html>
<body>
<script>

    var age = 13;

    if( age > 18 ) {
        document.write("<b>Qualifies for driving</b>");
    }
    //-->
</script>

</body>
</html>

```

Program 12: If-else statement

```

<html>
<body>
<script>

    var age = 13;

    if( age > 18 ) {
        document.write("<b>Qualifies for driving</b>");
    }
    else {
        document.write("<b>Does not qualify for driving</b>");
    }
    //-->

```

```
    </script>

</body>
</html>
```

Program 13: if...else if... statement

```
<html>
  <body>
    <script>

      var book = "maths";
      if( book == "history" ) {
        document.write("<b>History Book</b>");
      } else if( book == "maths" ) {
        document.write("<b>Maths Book</b>");
      } else if( book == "economics" ) {
        document.write("<b>Economics Book</b>");
      } else {
        document.write("<b>Unknown Book</b>");
      }

    </script>

  </body>
</html>
```

Program 14: Switch

```
<html>
  <head>
    <title>Show greater among three numbers</title>
  </head>
  <body>
    <script>
      var grade='A';
      document.write("Entering switch block<br />");
      switch (grade)
      {
        case 'A':
          document.write("Good job<br />");
          break;
        case 'B':
          document.write("Pretty good<br />");
          break;
        case 'C':
          document.write("Passed<br />");
          break;
        case 'D':
          document.write("Not so good<br />");
          break;
        case 'F':
          document.write("Failed<br />");
          break;
        default:
          document.write("Unknown grade<br />")
      }
      document.write("Exiting switch block");
    </script>
  </body>
</html>
```

Program 15: Events in JavaScript : onclick

```

<html>
  <head>
    <script type = "text/javascript">
      <!--
        function sayHello() {
          alert("Hello World")
        }
      //-->
    </script>
  </head>

  <body>
    <p>Click the following button and see result</p>
    <form>
      <input type = "button" onclick = "sayHello()" value = "Say Hello" />
    </form>
  </body>
</html>

```

Program 16: Events in JavaScript : **onmouseover**

```

<html>
  <head>
    <script type = "text/javascript">
      <!--
        function over() {
          document.write ("Mouse Over");
        }
        function out() {
          document.write ("Mouse Out");
        }
      //-->
    </script>
  </head>

  <body>
    <p>Bring your mouse inside the division to see the result:</p>
    <div onmouseover = "over()" onmouseout = "out()">
      <h2> This is inside the division </h2>
    </div>
  </body>
</html>

```

Program 17: Events in JavaScript : **onchange**

```

<!DOCTYPE html>
<html>
<body>

<p>Select a new car from the list.</p>

<select id="mySelect" onchange="myFunction()">
  <option value="Audi">Audi
  <option value="BMW">BMW
  <option value="Mercedes">Mercedes
  <option value="Volvo">Volvo
</select>

<p>When you select a new car, a function is triggered which outputs the value of the selected car.</p>

```

```

<p id="demo"></p>

<script>
function myFunction() {
    var x = document.getElementById("mySelect").value;
    document.getElementById("demo").innerHTML = "You selected: " + x;
}
</script>

</body>
</html>

```

Program 18: Events in JavaScript : **onkeypress**

```

<!DOCTYPE html>
<html>
<body>

<p>A function is triggered when the user is pressing a key in the input field.</p>

<input type="text" onkeypress="myFunction()">

<script>
function myFunction() {
    alert("You pressed a key inside the input field");
}
</script>

</body>
</html>

```

Program 19: Cookie in JavaScript :

```

<html>
    <head>
        <script>
function clicked()
{
    console.log("Function Clicked");
}

        </script>
    </head>
    <body>
        <form name="myform" action="">
            <p> Enter Cookie Info:</p>
            <input type="text" name= "name"/>
            <input type="button" value="submit" onclick="clicked()"/>
        </form>
    </body>
</html>

```

Program 20: Storing Cookie in JavaScript :

```

<html>
    <head>
        <script>
function clicked()
{
    console.log("Function Clicked");
    var value =escape(document.myForm.name.value+"");
    console.log(value);
    document.cookie="name="+ value;
}

```

```

}

</script>
</head>
<body>
<form name="myForm" action="">
<p> Enter Cookie Info:</p>
<input type="text" name= "name"/>
<input type="button" value="submit" onclick="clicked()"/>
</form>
</body>
</html>

```

Program 21: Reteriving Cookie in JavaScript :

```

<html>
  <head>
    <script>
function clicked()
{
  console.log("Function Clicked");
  var value =escape(document.myForm.name.value+";");
  console.log(value);
  document.cookie="name="+ value;
  var allcookies = unescape(document.cookie);
  console.log(allcookies);
}

    </script>
  </head>
  <body>
    <form name="myForm" action="">
    <p> Enter Cookie Info:</p>
    <input type="text" name= "name"/>
    <input type="button" value="submit" onclick="clicked()"/>
    </form>
  </body>
</html>

```

Program 21: Expire date Cookie in JavaScript :

```

<html>
  <head>
    <script>
function clicked()
{
  console.log("Function Clicked");
  var value =escape(document.myForm.name.value+";");
  console.log(value);
  //document.cookie="name="+ value;

  //  var allcookies = unescape(document.cookie);
  //  console.log(allcookies);

  var now =new Date();
  now.setMonth(now.getMonth()+1);
  document.cookie="name=" + value;
  document.cookie="expire=" +now.toUTCString()+";";
  var allcookies = unescape(document.cookie);
  console.log(allcookies);
}

```

```

</script>
</head>
<body>
<form name="myForm" action="">
<p> Enter Cookie Info:</p>
<input type="text" name= "name"/>
<input type="button" value="submit" onclick="clicked()"/>
</form>
</body>
</html>

```

Program 22: Delete Cookie in JavaScript :

```

<html>
  <head>
    <script>
function clicked()
{
  console.log("Function Clicked");
  var value =escape(document.myForm.name.value+";");
  console.log(value);
  //document.cookie="name="+ value;

  // var allcookies = unescape(document.cookie);
  // console.log(allcookies);

  var now =new Date();
  now.setMonth(now.getMonth()+1);
  document.cookie="name=" + value;
  document.cookie="expire=" +now.toUTCString()+";";

  var allcookies = unescape(document.cookie);
  console.log(allcookies);

  var now =new Date();
  now.setMonth(now.getMonth()-1);
  document.cookie="name=" + value;
  document.cookie="expire=" +now.toUTCString()+";";

  var allcookies = unescape(document.cookie);
  console.log(allcookies);
}

</script>
</head>
<body>
<form name="myForm" action="">
<p> Enter Cookie Info:</p>
<input type="text" name= "name"/>
<input type="button" value="submit" onclick="clicked()"/>
</form>
</body>
</html>

```

Program 23: Math Objects in JavaScript :

Round

How to round a specified number to the nearest whole number

```
<html>
<body>
<script type="text/javascript">
document.write(Math.round(7.25))
</script>
</body>
</html>
```

Random number

The random method returns a random number between 0 and 1

```
<html>
<body>
<script type="text/javascript">
document.write(Math.random())
</script>
</body>
</html>
```

Random number from 0-10

How to write a random number from 0 to 10, using the round and the random method.

```
<html>
<body>
<script type="text/javascript">
no=Math.random()*10
document.write(Math.floor(no))
</script>
</body>
</html>
```

Max number

How to test which of two numbers, has the highest value.

```
<html>
<body>
<script type="text/javascript">
document.write(Math.max(2,4))
</script>
</body>
</html>
```

Min number

How to test which of two numbers, has the lowest value.

```
<html>
<body>
<script type="text/javascript">
document.write(Math.min(2,4))
</script>
</body>
</html>
```

Program 24: String Objects in JavaScript :

The length method

The length method returns the number of characters in a string.

Example:

```
<html>
<body>
<script type="text/javascript">
var str="Web Enabling Tools is Cool!"
document.write("<p>" + str + "</p>")
document.write(str.length)
</script>
</body>
</html>
```

The toLowerCase() and toUpperCase() methods

Converts a string to lower case and upper case respectively.

Example:

```
<html>
<body>
<script type="text/javascript">
var str=("Hello JavaScripters!")
document.write(str.toLowerCase())
document.write("<br>")
document.write(str.toUpperCase())
</script>
</body>
</html>
```

The Array object

An Array object is used to store a set of values in a single variable name. Each value is an element of the array and has an associated index number.

You can refer to a particular element in the array by using the name of the array and the index number. The index number starts at zero.

You create an instance of the Array object with the "new" keyword.

```
var family_names=new Array(5)
```

Examples

Array

Arrays are used to store a series of related data items. This example demonstrates how you can make an array that stores names.

```
<html>
<body>
<script type="text/javascript">
var famname = new Array(6)
famname[0] = "Jan Egil"
famname[1] = "Tove"
famname[2] = "Hege"
famname[3] = "Stale"
famname[4] = "Kai Jim"
famname[5] = "Borge"
for (i=0; i<6; i++)
{
```

```
document.write(famname[i] + "<br>")
}
</script>
</body>
</html>
```

The Date object

The Date object is used to work with dates and times.

You create an instance of the Date object with the "new" keyword.

To store the current date in a variable called "my_date":

```
var my_date=new Date()
```

After creating an instance of the Date object, you can access all the methods of the object from the "my_date" variable. If, for example, you want to return the date (from 1-31) of a Date object, you should write the following:

```
my_date.getDate()
```

You can also write a date inside the parentheses of the Date() object, like this:

```
new Date("Month dd, yyyy hh:mm:ss")
new Date("Month dd, yyyy")
new Date(yy,mm,dd,hh,mm,ss)
new Date(yy,mm,dd)
new Date(milliseconds)
```

Here is how you can create a Date object for each of the ways above:

```
var my_date=new Date("October 12, 1988 13:14:00")
var my_date=new Date("October 12, 1988")
var my_date=new Date(88,09,12,13,14,00)
var my_date=new Date(88,09,12)
var my_date=new Date(500)
```

Examples

Date

Returns today's date including date, month, and year. Note that the getMonth method returns 0 in January, 1 in February etc. So add 1 to the getMonth method to display the correct date.

```
<html>
<body>
<script type="text/javascript">
var d = new Date()
document.write(d.getDate())
document.write(".")
document.write(d.getMonth() + 1)
document.write(".")
document.write(d.getFullYear())
</script>
</body>
</html>
```

Time

Returns the current local time including hour, minutes, and seconds. To return the GMT time use `getUTCHours`, `getUTCMinutes` etc.

```
<html>
<body>
<script type="text/javascript">
var d = new Date()
document.write(d.getHours())
document.write(".")
document.write(d.getMinutes() + 1)
document.write(".")
document.write(d.getSeconds())
</script>
</body>
</html>
```

UTC time

The `getUTCDate` method returns the Universal Coordinated Time which is the time set by the World Time Standard.

```
<html>
<body>
<script type="text/javascript">
var d = new Date()
document.write(d.getUTCHours())
document.write(".")
document.write(d.getUTCMinutes() + 1)
document.write(".")
document.write(d.getUTCSeconds())
</script>
</body>
</html>
```

Display weekday

A simple script that allows you to write the name of the current day instead of the number. Note that the array object is used to store the names, and that Sunday=0, Monday=1 etc.

```
<html>
<body>
<script type="text/javascript">
var d = new Date()
var weekday=new Array("Sunday","Monday","Tuesday","Wednesday","Thursday","Friday","Saturday")
document.write("Today is " + weekday[d.getDay()])
</script>
</body>
</html>
```

Display full date

How to write a complete date with the name of the day and the name of the month.

```
<html>
<body>
<script type="text/javascript">
var d = new Date()
var weekday=new Array("Sunday","Monday","Tuesday","Wednesday","Thursday","Friday","Saturday")
```

```

var monthname=new Array("Jan","Feb","Mar","Apr","May","Jun","Jul","Aug","Sep","Oct","Nov","Dec"
)
document.write(weekday[d.getDay()] + " ")
document.write(d.getDate() + ". ")
document.write(monthname[d.getMonth()] + " ")
document.write(d.getFullYear())
</script>
</body>
</html>

```

JavaScript Form Controls

Examples

E-mail validation :How you can validate an input-field that contains an e-mail address.

```

<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>JavaScript form validation - checking email</title>
<style>
    li {list-style-type: none;
font-size: 16pt;
}
.mail {
margin: auto;
padding-top: 10px;
padding-bottom: 10px;
width: 400px;
background : #D8F1F8;
border: 1px solid silver;
}
.mail h2 {
margin-left: 38px;
}
input {
font-size: 20pt;
}
input:focus, textarea:focus{
background-color: lightyellow;
}
input submit {
font-size: 12pt;
}
.rq {
color: #FF0000;
font-size: 10pt;
}
</style>

</head>
<body onload='document.form1.text1.focus()>
<div class="mail">
<h2>Input an email and Submit</h2>
<form name="form1" action="#">
<ul>
<li><input type='text' name='text1'/></li>
<li>&nbsp;</li>

```

```

<li class="submit"><input type="submit" name="submit" value="Submit" onclick="ValidateEmail(document.form1.text1)"/></li>
<li>&nbsp;</li>
</ul>
</form>
</div>
<script>
    function ValidateEmail(inputText)
    {
var mailformat = /^[a-zA-Z0-9.!#$%&'*/+=?^_`{|}~-]+@[a-zA-Z0-9-]+(?:\.[a-zA-Z0-9-]+)*$/;
if(inputText.value.match(mailformat))
{
alert("Valid email address!");
document.form1.text1.focus();
return true;
}
else
{
alert("You have entered an invalid email address!");
document.form1.text1.focus();
return false;
}
}
</script>
</body>
</html>

```

Value validation

How you can validate an input-field with min and max values.

Length validation

How you can validate number of letters in an input-field.

Form validation

A form containing all of the validation above.

Focus

How to set focus on an input field.

Selected

How to make the content in an input field selected.

Radio button

How the client can select options from radio buttons.

Checkbox

How the client can select options from checkboxes.

Select from a dropdown list

How the client can select options from a drop down list.

Select more than one option

How the client can select many options from a drop down list.

JavaScript Window Objects

Examples-Alert

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

<h2>JavaScript Alert</h2>

<button onclick="myFunction()">Try it</button>

<script>
function myFunction() {
    alert("I am an alert box!");
}
</script>

</body>
</html>
```

Examples-Confirm

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

<h2>JavaScript Confirm Box</h2>

<button onclick="myFunction()">Try it</button>

<p id="demo"></p>

<script>
function myFunction() {
    var txt;
    if (confirm("Press a button!")) {
        txt = "You pressed OK!";
    } else {
        txt = "You pressed Cancel!";
    }
    document.getElementById("demo").innerHTML = txt;
}
</script>

</body>
</html>
```

Examples-Prompt

```

<!DOCTYPE html>
<html>
<body>

<h2>JavaScript Prompt</h2>

<button onclick="myFunction()">Try it</button>

<p id="demo"></p>

<script>
function myFunction() {
    var txt;
    var person = prompt("Please enter your name:", "Sumit");
    if (person == null || person == "") {
        txt = "User cancelled the prompt.";
    } else {
        txt = "Hello " + person + "! How are you today?";
    }
    document.getElementById("demo").innerHTML = txt;
}
</script>

</body>
</html>

```

Examples-OPEN

```

<!DOCTYPE html>
<html>
<head>
<script>
function openWin() {
    window.open("https://techstudymaterials.blogspot.com");
}
</script>
</head>
<body>

<form>
    <input type="button" value="Open Window" onclick="openWin()">
</form>

</body>
</html>

```

Examples-PRINT

```

<!DOCTYPE html>
<html>
<head>
<script>
function printPage() {
    window.print();
}

```

```

}
</script>
</head>
<body>

<input type="button" value="Print this page" onclick="printPage()" />

</body>
</html>

```

PRATICE QUESTIONS

1. Write a JavaScript program to find the largest of five given integers.
2. Write a JavaScript program to reverse a given string.
3. Write a JavaScript program to calculate multiplication and division of two numbers (input from user).
4. Write a JavaScript program to find the area of a triangle where lengths of the three of its sides are 5, 6, 7.
5. Write a JavaScript Program to Subtract and Multiply Two Numbers.
6. Write a JavaScript program to convert temperatures to and from Celsius, Fahrenheit.
7. Write a JavaScript exercise to create a variable using a user-defined name.
8. Write a JavaScript program to compute the sum of the two given integers. If the two values are same, then returns triple their sum.
9. Write a JavaScript program to check whether a string starts with 'Java' and false otherwise.
10. Write a JavaScript function to convert a decimal number to binary, hexadecimal or octal number.
11. Write a JavaScript function to find the highest value in an array.
12. Write a JavaScript program to construct the following pattern, using a nested for loop.

```

*
* *
* * *
* * * *
* * * * *

```

13. Write a JavaScript function that reverse a number.
14. Write a JavaScript function that returns a passed string with letters in alphabetical order.
15. Write a simple JavaScript program to join all elements of the following array into a string
16. Write a JavaScript program to find the most frequent item of an array.
17. Write a JavaScript function to check whether a string is blank or not.
18. Write a JavaScript function to extract a specified number of characters from a string.
19. Write a JavaScript function to convert a string in abbreviated for m.
20. Write a JavaScript function to capitalize the first letter of a string.
21. Write a JavaScript function to split a string and convert it into an array of words.
22. Write a JavaScript function to get the number of days in a month.
23. Write a JavaScript function to get the last day of a month.
24. Write a JavaScript function to get all prime numbers from 0 to a specified number.
25. Write a JavaScript function that converts hours into seconds.
26. Write a JavaScript function that takes the base and height of a triangle and return its area.
27. Write a JavaScript function that takes an integer minutes and converts it to seconds.
28. Write a JavaScript function that takes the age and return the age in days.
29. Write a JavaScript function that takes two numbers as arguments and return their sum.

30. Write a JavaScript program to display the current day and time in the following format.

Sample Output : Today is : Tuesday.

Current time is : 10 PM : 30 : 38

INTERNET AND WEB TECHNOLOGY (IWT)



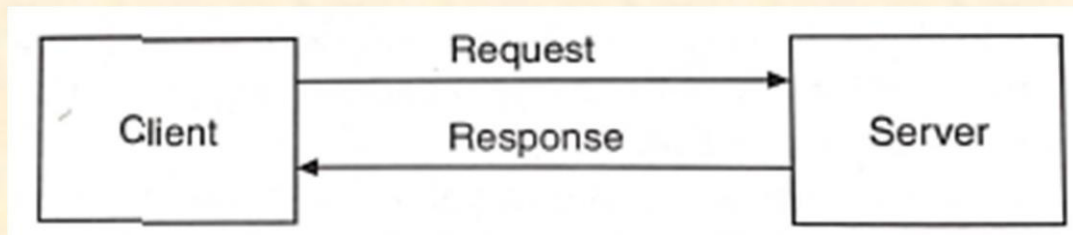
Chapter 8: Server Side Scripting

- ✓ 8.1 Introduction to server side Scripting
- ✓ 8.2 Components of SSS
- ✓ 8.3 Difference between CSS and SSS
- ✓ 8.4 Server side Scripting method
- ✓ 8.5 JavaScript on server
- ✓ 8.6 SQL

Chapter 8 - Server Side Scripting

Introduction to Server-side Scripting

- **Server side scripting** is defined as “ web server technology in which the user’s request is fulfilled by running a script on a web server to generate dynamic web pages.”
- In simple terms, a server-side scripts runs on the server rather than your computer. When you visit a website, the script will create the webpage on dynamically. In case of server side scripting, the client send the request on the machine and then send the result to the client machine.



- Popular Server side scripting languages include Perl PHP, ASP, JSP, Ruby, Cold fusion and Python.

Advantages:

- Load times are generally faster than client side scripting.
- It doesn't require the user to download plugins like JAVA or flash.
- You can create a single website template for the entire website. Each new dynamic page you create will automatically use it.
- Your scripts are hidden from views. Users only see HTML output, even when they view the source.

Components of SSS

- Server side scripting is not only collection of some programs that are executed on request, but it also contains some other below mentioned parts:

Web server

- A web server is a software that continuously monitors for any request that has been made from any client. It reads any request, it trees out the particular file or values from the database and sends the results to the client machine.
- The web servers that available these days are:
- **PWS (Personal Web server):** This web server is available for window 98, 98 SE (Second Edition) and ME (Mellenium Edition).

- **IIS (Internet Information Server):** This is available for window NT (Server and workstation). 2000 (Server and professional), XP, 2003 etc.
- **Apache:** This web server is available for both window and UNIX platforms. Every web server creates a default web directory for strong is files e.g., IIS creates.

{Root Directory}/inetpub/www root

- where (Root Directory) is the name of the drive where the windows is install.
- Web server stores its files in the particular web directory. The web server reads the request from the client (i.e., browser) and sends the output to the client. Also, the particular web-directory is referred a local host from the browser if the server and client are running on the same machine.

Scripting Files

- The second part of the server side scripting is the scripting files that are executed on the request of the client. These files are developed in any of the server side scripting languages that are available with us:
 - ASP(Active Server Pages)
 - JSP(Java Server Pages)
 - PHP(Personal Home Pages)
 - Servlets
- These languages are having its own set of syntax to perform the specific tasks. Each language works under different requirement e.g., ASP needs simply IIS installed and PHP can work with both IIS and Apache whereas ISP and servlets do require another software that work along with the web both server so that JSP and servlets programs can be executed.
- These software are like:
 - Tomcat server
 - Weblogics
 - Apache Jserve etc.

Database Files

- The third part of working of the server side scripting is the database files. These database files can be designed in any of the available database such that:
 - MS Access
 - SQL Server
 - My SQL

▪ ORACLE

- The job of the database is to store the information about various entities (any physically existing object is known as entity).
- A database is the collection of tables which inturn contains the records or rows of any existing object.

emp: Table

<i>emp.no</i>	<i>Name</i>	<i>Sol</i>
1.	ABC	1000
2.	XYZ	2000

Records

The database tables can be queried by using some standard SQL statement.

SQL

- SQL is Structured Query Language that is used to interact with the databases. It is integral part of any Relational Database Management System (RDBMS) package system. SQL uses some predefined syntax statement to perform various database operations.

Scripting Methods

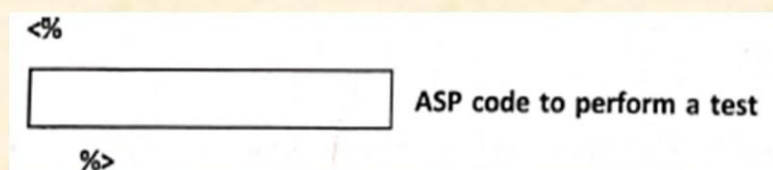
The main scripting methods that are used in server side scripting are mentioned as under:

(a) ASP

(b) JSP

(a) ASP: The active Server pages are the server side scripting language that executes on the windows platform only. When the IIS web server is installed on a machine, the ASP itself starts running on the computer (there is no need of installing the special compiler or parser for the ASP programs).

The ASP code is developed by using the below mentioned syntax in the

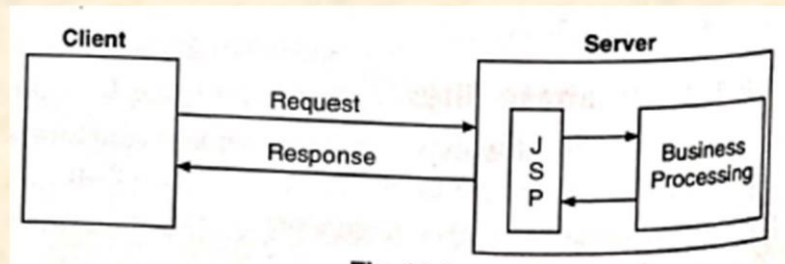


HTML files ASP code is stored in the root directory and have a special extension i.e., asp. In order to execute the files or scripts, the script is stored in the root directory (web server directory) and then referred from the client machine or browser by putting the following address in the address bar:

http: \ \ local host\ test.asp

The above mentioned address can execute the test.asp file stored in the root directory of the computer.

(b) JSP: Java Server Pages is the other scripting language that one is using instead of using the ASP. It is branch of java language. The knowledge of Java language is must



before using the JSP. Without Java, one is not able to use JSP. It works on the below mentioned Architecture.

JSP needs a special type of software package in order to run properly. This software package is always working in conjunction with the web server. One such software is known as Tomcat server.

Difference between CSS and SSS

1. Client side scripting :

Web browsers execute client side scripting. It is used when browsers have all code. Source code is transferred from web server to user's computer over internet and runs directly on browsers. It is also used for validations and functionality for user events.

It allows for more interactivity. It usually performs several actions without going to user. It cannot be basically used to connect to databases on web server. These scripts cannot access file system that resides at web browser. Pages are altered on basis of user's choice. It can also be used to create "cookies" that store data on user's computer.

2. Server side scripting :

Web servers are used to execute server side scripting. They are basically used to create dynamic pages. It can also access the file system residing at web server. Server-side environment that runs on a scripting language is a web-server.

Scripts can be written in any of a number of server-side scripting languages available. It is used to retrieve and generate content for dynamic pages. It is used to require to download plugins. In this load times are generally faster than client-side scripting. When you need to store and retrieve information a database will be used to contain data. It can use huge resources of server. It reduces client-side computation overhead. Server sends pages to request of user/client.

Difference between client side scripting and server side scripting :

CLIENT SIDE SCRIPTING	SERVER SIDE SCRIPTING
Source code is visible to user.	Source code is not visible to user because its output of server side is a HTML page.

It usually depends on browser and it's version.	In this any server side technology can be use and it does not depend on client.
It runs on user's computer.	It runs on web server.
There are many advantages link with this like faster. response times, a more interactive application.	The primary advantage is it's ability to highly customize, response requirements, access rights based on user.
It does not provide security for data.	It provides more security for data.
It is a technique use in web development in which scripts runs on clients browser.	It is a technique that uses scripts on web server to produce a response that is customized for each clients request.
HTML, CSS and javascript are used.	PHP, Python, Java, Ruby are used.

Server side scripting method

- Let us see some of the server side scripting method:

CGI (Common Gateway Interface)

- Simple scripting mechanisms supported by practically all web server. Most commonly written in Perl, C/C++, or Python.

Java Servlets

- Ability to execute Java object on the server. Newer, less common technology but has dramatic performance advantage over CGI.

JavaScript on server

- Java Script extends the capabilities of the server. By providing languages the software can do more without calling an external program. This makes it easier for webmasters to add features to their sites that browsers can take advantages of. And it can reduce the load on the server by keeping the processing within the server software.

Request Object

- Each time a browser wants more information it sends a request to a server. Data about these requests are available on the server from the request object in JavaScript. If you have written CGI scripts, you should be familiar with the properties that are built into the request object.
 - **The Built-In Properties:** The request object is initialized with four properties: agent, ip, information is a line of text, but there is no standard format to this text. The

information to determine its response to the request, The following list provides details about each property:

- i. **Agent:** This property tells the server about the client's software, usually a browser. This information is a line of text, but there is no standard format to this text. The same browser always presents the same information. For example, if you use Netscape Navigator 2.0 and your friend uses Netscape Navigator 2.0, the same information is sent by either of you when you access this server.
 - ii. **IP:** The ip property provides the ip address of the client. It tells the server from where on the Internet the request is sent and where to return a response. IP addresses are composed of a set of four numbers. Each is a value between 0 and 255. A sample IP number is 127.0.0.1.
 - iii. **Method:** This property provides the HTTP method associated with the request. There are three methods currently used: get, post, and head.
 - iv. **Protocol:** This is a standard text indicating the level of HTTP protocol used by the client. Currently this should be HTTP/ 1.0 in every case.
- **Other Request Properties:** A request may include additional information. Your browser's request may have resulted from submitting information from an HTML form. This information becomes a request property. It does not matter if the data is submitted with either the post or get method. The name of each element of the form becomes a property name.
 - **Application of Request Properties:** These built-in properties allow the server to vary the response to the request. The agent property is commonly used to supply different HTML documents to different types of browser. For example, an HTML page with tables might only be served to a limited number of browsers while another receives a different version of the page.
 - Server may limit access to certain information to specific IP addresses. Using the ip property, the server might only provide sensitive company information to a client requesting it from a known company computer. Usually you want additional security, just screening IP address is not enough.

Client Object

- Each time a new client accesses the server application, a new client object is created. However, there are no built-in properties for the client object. If you want information retained about the client, then you create a property.
- **Client Properties:** A very common use for client properties is a client identification number. When the client first accesses the server, he can receive a form. The server can then process this information and assign the client an identification number. This might be randomly generated or looked up from a list of previous customers.
- As an example, a client can send a request. Part of the request can include a user-supplied customer number in the text box named idnumber. You can then create a property called custNo with the following line:

Client.custNo = Request.idnumber

- **Client Object Expire:** The server overflows with client objects unless they are properly deleted. Since client object are automatically created for each and every client to access, then there must be a mechanism for deleting them.
- Live Wire automatically deletes any client object with no property values. So if you don't use a client object, you don't have to worry about deleting it. In other words, you only have to clean up after yourself.
- The default expiration is ten minutes of inactivity. If the client does not send another request to the server within ten minutes of the previous request, the object expires.
- Obviously, in many cases ten minutes is inefficient time and, therefore, you might want to manually control this. You can simply use the expiration statement

client.expiration(seconds)

where second is the number of second before the client expires.

- Another manual control is destroy. If you no longer need the client object, simply use the statement

client.destroy()

- LiveWire looks at the client that sent the request and destroys its client object, This eliminates all of the client object's properties as well.
- **Cookies Store Information between Sessions:** Another technique for retaining client information is cookies. The browser must support the Netscape cookie protocol. If supported, the server sends the information to the client as name/value pairs. Obviously, this increases network traffic, but can offer substantial advantages to large access server.

JavaScript and CGIs

- Undoubtedly with your active interest in building websites, you have dealt with common Gateway Interface (CGI) Scripts. Prior to JavaScript, this was the primary means of creating interactive applications. Libraries of CGI script include counters, e-mailers boards, and many other function.
- LiveWire can replace CGI programming. Instead of calling external programs, the server software runs applications that are closely integrated to it. JavaScript is the language of these applications.
- Applications are developed with three tools. You build these application using LiveWire's Site Manager. The source files for the Applications are developed using the same HTML editors used to build browser JavaScript pages. The applications run in response to request from Netscape Navigator.

The JavaScript Balancing Act

- When interactivity was only done with a CGI, all of the processing was done on the server. Because JavaScript can run on both the server and the browser, writing a successful application requires you to properly allocate the processing between the two.

1. Browser Handle the Work: In general, let the browser code “polish” the information before sending it to the server. The following are several different tasks best handled by the browser :

- Validate user inputs
- Check that the values are within range
- Prompt for confirmations
- Verify inputs that are valid, but not usual
- Perform aggregate calculations (totals, average, means, sales tax)
- Conditionalize HTML
- Perform other functions that require no server data

2. More Work for the Server: Though the browser code can take much of the burden off the server, the server still must perform. Given the increasingly interactive nature of web applications, the demand on the server are increasing. The following types of tasks should be performed by the server :

- Direct the flow from one page to another
- Maintain data about the client from one request to the next
- Maintain the shared data among clients and applications
- Access various databases
- Access various files on the server (for example, multimedia)
- Call C libraries, as needed
- Dynamic customization of Java applets

- Obviously, the server is a busy fellow. And this load is expected to grow dramatically as servers provide multimedia materials and more interactivity.

Exchanging information between the browser and server

- In making your application come alive, the server and browser must exchange information. The client, or browser, typically sends user response. These can first be “polished” by JavaScript routines on the browser side. The server in turn sends data back to the browser as HTML pages.
- **From client to server:** User responses are submitted just as you currently handle forms. The user completes the form and clicks the submit button. The data from the radio buttons,

checkboxes, textboxes, and textarea are sent to the server. The server then places this data into the request object. Each element of the form has a corresponding property.

- **From Server to Client:** Usually a server only returns a static in response to a browser request. In LiveWire, the response is still a page, but the contents of the page vary. User input can result in changes to default form values, new values of hidden form elements, or direct substitutions.
- A server-side JavaScript can dynamically build in the HTML code for a form element that is part of the page. As an example, you can have the following statement in a source document:

<INPUT TYPE="text" NAME="example" VALUE='request.agent'>

- In this case the default value of the text is the browser agent information.
- You use an identical procedure for hidden form elements. The only difference is that the type is hidden instead of text. Your client-side JavaScript can then value as part of any function.
- Another means of making your page come alive is to change part of the JavaScript code. When you send a page to the browser it can contain JavaScript code for the browser to execute as part of the page. There is no reason that this code has to be static.
- The server can modify the page being sent to change the JavaScript code embedded in the page. The page on the server side might include

<SERVER>

Write ("<SCRIPT>var luck =" + client.winnings + "</SCRIPT>")

</SERVER>

- Assuming the values of client.winnings is 1000, the browser sees a line of

<SCRIPT>var luck = 1000 </SCRIPT>

SQL (STRUCTURE QUERY LANGUAGE)

- It is used to interact with the databases. It is integral part of any Relational Database Management System (RDBMS) package system. SQL uses some predefined syntaxed statement to perform various database operations like.

➤ **Retrieving Information from a Table**

- SQL uses the select statement for this purpose.
- **Syntax:**

Select column names /* from tablename;

- Where;

i. Column names is the list of columns (separated by (,) comma) or * for listing all columns.

ii. And Tablenames is the name of the table.

- Hence, the statement can be written as:

Select * from emp;

- And the above statement retrieves all the columns and rows the emp table.

➤ Inserting Records into a Table

- For inserting rows and tables into the database tables, the SQL uses following syntax:

Insert into tablename values (set of values);

- **Example:** To insert any record in a table having name emp and having field empno and sol, we can issue a SQL statement:

Insert into emp values (1, 1000)

➤ Deletion of Records

- This SQL statement is used for the deletion of the records from a table. It uses the following syntax:

Delete from tablename;

- **Example:** If one wants to delete all the records present in the table, the SQL statement would be like

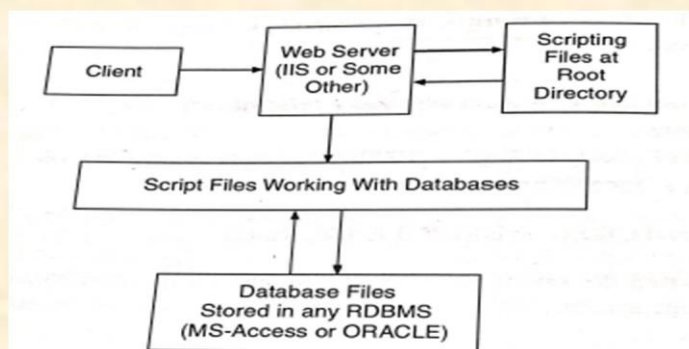
Delete from emp;

➤ Updation of Records

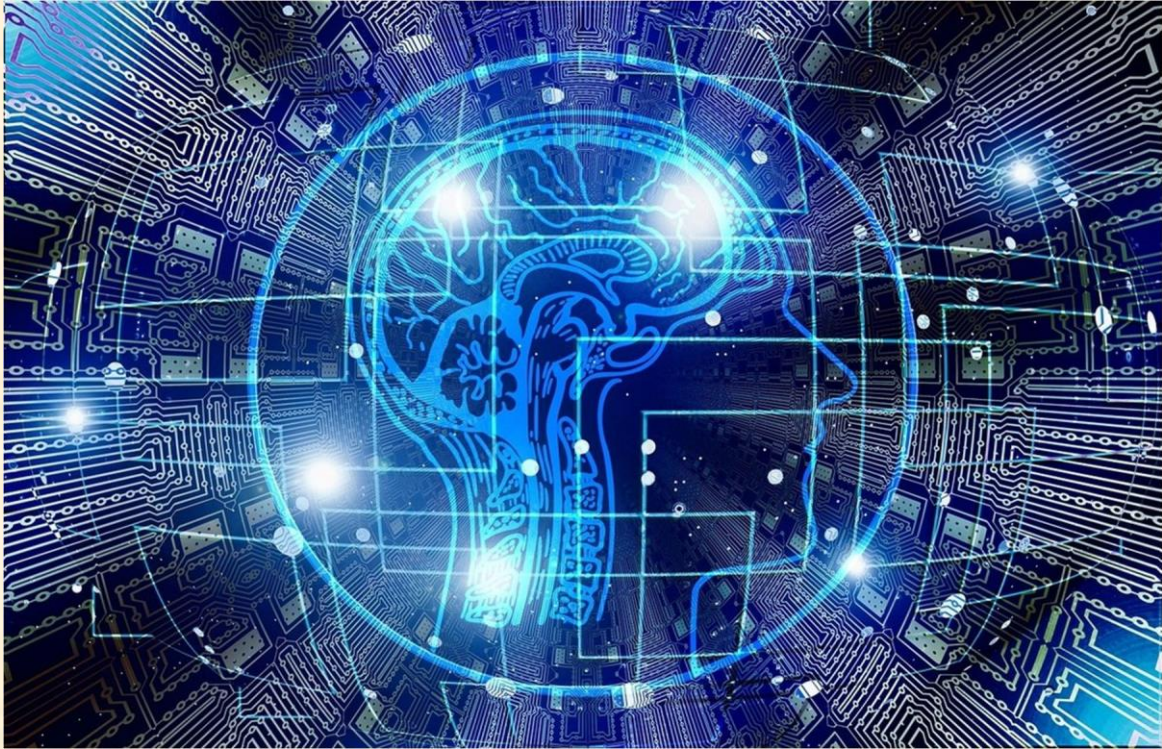
- For updating the already existing records:

Update tablename set column name = new value;

- All of these parts i.e., web server, scripting files and database files work together to perform the server side scripting property. The figure given below shows the working.



INTERNET AND WEB TECHNOLOGY (IWT)



Chapter 9: Server Side Programming Using PHP

- ✓ 9.1 Introduction to PHP
- ✓ 9.2 Variables, string, Operator types
- ✓ 9.3 Conditional Statement. Loops
- ✓ 9.4 Array
- ✓ 9.5 GET and POST Method and Session

Chapter 9 – Server Side Programming Using PHP

Introduction to PHP

- PHP started out as a small source project that evolved as more and more people found out how useful it was. Rasmus Lerdorf unleashed the first version of PHP way back in 1994.
- PHP is a recursive acronym for "PHP: Hypertext Preprocessor".
- PHP is an open-source server-side scripting language. PHP version 5.2.14 is available on web server.
- We can create dynamic web pages with the PHP scripting language. A dynamic web page interacts with the user, so that each user visiting the page sees customized information. PHP can also be used to create dynamic web pages that are generated from information accessed from a MySQL database.
- We can embed PHP commands within a standard HTML page. PHP's syntax is similar to that of C and Perl, making it easy to learn for anyone with basic programming skills another feature that PHP offers is connectivity to most of the common databases.
- PHP also offers integration with various external libraries, which allow the developers to do anything from generating PDF documents to parsing XML.
- PHP is pleasingly zippy in its execution, especially when compiled as an Apache module on the Unix side. The MySQL server, once started, executes even very complex queries with huge result sets in record-setting time.
- PHP supports a large number of major protocols such as POP3, IMAP, and LDAP. PHP4 added support for Java and distributed object architectures (COM and CORBA), making n-tier development a possibility for the first time.
- PHP is forgiving: PHP language tries to be as forgiving as possible.

Common Uses of PHP

PHP performs system functions, i.e. from files on a system it can create, open, read, write, and close them. The other uses of PHP are:

- PHP can handle forms, i.e. gather data from files, save data to a file, thru email you can send data, return data to the user.
- You add, delete, and modify elements within your database thru PHP.
- Access cookies variables and set cookies.
- Using PHP, you can restrict users to access some pages of your website.

- It can encrypt data.

Characteristics of PHP

Five important characteristics make PHP's practical nature possible:

- Simplicity
- Efficiency
- Security
- Flexibility
- Familiarity

PHP echo and print Statements

With PHP, there are two basic ways to get output: echo and print. echo and print are more or less the same. They are both used to output data to the screen.

The differences are small: echo has no return value while print has a return value of 1 so it can be used in expressions. echo can take multiple parameters (although such usage is rare) while print can take one argument. echo is marginally faster than print.

Variables, string, Operator types

Variables

Sometimes it is convenient to be able to have variable names. Variables in PHP are represented by a dollar sign followed by the name of variable. The variable name is case-sensitive. Variable names follow the same rules as other labels in PHP. A valid variable name starts with a letter or underscore, followed by any number of letters, numbers, or underscores. Store values for future reference, use variable name to refer to the value stored in it.

For example:

```
<? php
```

```
$ var 'Bob';
```

```
$var 'joe';
```

```
echo "Svar, $var"; //outputs "Bob, joe"
```

```
$4site 'not yet'; //invalid; starts with numbers
```

```
$_4site = 'not yet'; //valid; starts with an underscore
```

```
$taye 'manikka'; //valid; 'a' is (extended) ASCII 228
```

```
?>
```

By default, variables are always assigned by value. That is to say, when you assign an expression to a variable, the entire value of the original expression is copied into the destination variable. This means, for instance, that after assigning one variable's value to another changing one of those variables will have no effect on the other.

Data types

PHP has a total of eight data types which we use to construct our variables:

- **Integers:** are whole numbers, without a decimal point, like 4195.
- **Doubles:** are floating-point numbers, like 3.14159 or 49.1.
- **Booleans:** have only two possible values either true or false.
- **NULL:** is a special type that only has one value: NULL.
- **Strings:** are sequences of characters, like 'PHP supports string operations.'
- **Arrays:** are named and indexed collections of other values.
- **Objects:** are instances of programmer-defined classes, which can package up both other kinds of values and functions that are specific to the class.
- **Resources:** are special variables that hold references to resources external to PHP (such as database connections).

Strings

In PHP Strings that are delimited by double quotes (as in "this") are preprocessed in both the following two ways by PHP :

- Certain character sequences beginning with backslash (\) are replaced with special characters
- Variable names (starting with \$) are replaced with string representations of their values.

A string is series of characters where a character is the same as a byte. This means that PHP only supports a 256 character set, and hence does not offer native unicode support.

A string literal can be specified in four different ways:

1. Single Quoted

The simplest way to specify a string is to enclose it in single quotes (the character).

For example

```
<? php  
  
echo 'this is a simple string';  
  
?>
```

2. Double Quoted

If the string is enclosed in double quotes (" "), the PHP will interpret more escape sequences for special characters:

Escaped Characters	
Sequence	Meaning
\n	linefeed
\r	Carriage return
\t	horizontal tab
\v	vertical tab
\e	escape sequence
\f	form feed
\\	backslash
\\$	dollar sign
\"	double quote
\[0-1] {1, 3}	The sequence of characters matching the regular expression is a character in octal notation.
\x[0-9A-Fa-f] {1, 2}	The sequence of characters matching the regular expression is a character in hexadecimal notation.

3. Heredoc

A third way to delimit strings is the heredoc syntax : <<<. After this operator, an identifier is provided then a newline. The string itself follows, then the same identifier again to close the quotation.

The closing identifier must begin in the first column of the line. Also, the identifier must follow the same naming rules as any other label in PHP, it must contain only alphanumeric characters and underscores and must start with a non-digit character or underscore.

Heredoc text behaves just like a double quoted string, without the double quotes. This means that quotes in a heredoc don't need to be escaped, but the escape codes listed above can still be used. Variables are expanded, but the same care must be taken when expression complex variables inside a heredoc as with strings.

For example

Heredoc in arguments: It is also possible to use the syntax to pass data to function arguments:

```
<?php
var_dump (array(<<<EDD
footbar!
EDD
));
?>
```

4. NOWdoc

NOWdocs are to single-quoted strings what heredocs are to double-quoted strings. A nowdoc is specified similarly to a heredoc; but no parsing is done inside a nowdoc. The construct is ideal for embedding PHP code or others large blocks of text without the need for escaping.

A NOWdoc is identified with same <<< sequence used for heredocs, but the identifier which follows is enclosed in signal quotes, e.g., <<<'EOT'. All the rules for heredoc identifier also apply to nowdoc identifier, especially those regarding the appearance of the closing identifier.

Example:

```
static data example
<? php
class foo {
public $bar = <<<'EOT'
bar EOT;
}
?>
```

Strlen() is a function: Return the Length of a String

Example:

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

```
<body>

<?php

echo strlen("Hello world!");

?>

</body>

</html>
```

Output: 12

str_word_count() - Count Words in a String

The PHP str_word_count() function counts the number of words in a string.

Example: <!DOCTYPE html>

```
<html>

<body>

<?php

echo str_word_count("Hello world!");

?>

</body>

</html>
```

Output:2

strrev() - Reverse a String

The PHP strrev() function reverses a string.

Example: <!DOCTYPE html>

```
<html>

<body>

<?php

echo strrev("Hello world!");

?>
```

```
</body>
```

```
</html>
```

Output: !dlrow olleH

strpos() - Search For a Text Within a String

The PHP strpos() function searches for a specific text within a string. If a match is found, the function returns the character position of the first match. If no match is found, it will return FALSE.

Example: <!DOCTYPE html>

```
<html>
```

```
<body>
```

```
<?php
```

```
echo strpos("Hello world!", "world");
```

```
?>
```

```
</body>
```

```
</html>
```

Output: 6

str_replace() - Replace Text Within a String

The PHP str_replace() function replaces some characters with some other characters in a string.

Example: <!DOCTYPE html>

```
<html>
```

```
<body>
```

```
<?php
```

```
echo str_replace("world", "Dolly", "Hello world!");
```

```
?>
```

```
</body>
```

```
</html>
```

Output: Hello Dolly!

Concatenation of strings – the . operator

```
$a = `Hello` ;
```

```
$b = `World` ;
```

```
echo $a . ` ` . $b . `!` ; //prints `Hello World!`
```

Operators

Simple answer can be given using expression 4 + 5 is equal to 9. Here 4 and 5 are called operands and + is called operator. PHP language support following type of operator.

1. Arithmetic Operators

There are following arithmetic operators supported by PHP language. Assume variable A and B holds value 5 and 10 respectively then

Show Example		
Operator	Description	Example
+	Adds two operands	A + B will give 15
*	Multiply both operand	A*B will get 50
-	Subtracts second operand from the first	A - B will get -5
/	Divide numerator by denominator	B/A will get 2
%	Modulus operator and remainder of after an integer division	B%A will give 0
++	Increment operator, increases integer value by 1	A++ give 6
--	Decrement operator, decreases: integer value by one	A -- give 4

2. Comparison Operators

There are following comparison operators supported by PHP language. Assume variable A holds 5 and variable B holds 10 then:

Operator	Description	Example
==	Checks if the value of two operands are equal or not, if yes then condition becomes true.	(A == B) is not true.
!=	Checks if the value of two operands are equal or not, if values are not equal then condition becomes true	(A != B) is true.
>	Checks if the value of left operand is greater than the value of right operand, if yes then condition becomes true	(A > B) is not true.
<	Checks if the value of left operand is less than the value of right operand, if yes then condition becomes true.	(A < B) is true.
>=	Checks if the value of left operand is greater than or equal to the value of right operand, if yes then condition becomes true	(A >= B) is not true.
<=	Checks if the value of left operand is less than or equal to the value of right operand, if yes then condition becomes true.	(A <= B) is true.

3. Logical Operator

There are following logical operators supported by PHP language. Assume variable A holds 10 and B holds 20 then:

Show Examples:

Operator	Description	Example
AND	Called logical AND operator. If both the operands are true then condition becomes true.	(A and B) is true.
OR	Called logical OR operator. If any of the two operands are non-zero then condition becomes true.	(A or B) is true.
%%	Called logical AND operator. If both the operands are non-zero then condition becomes true.	(A && B) is true.
	Called logical OR operator. If any of the two operands are non-zero then condition becomes true.	
!	Called logical NOT operator, use to reverse the logical state of its operand. If a condition is true then logical NOT operator will make false.	!(A && B) is false

4. Assignment Operators

There are following assignment operators supported by PHP language:

217

Examples:

Operator	Description	Example
=	Simple assignment operator, assigns values from right side operands to left side operand	C = A + B, will assign value of A + B into C
+=	Add AND assignment operator, it adds right operand to the left operand and assign the result to the left operand	C += A is equivalent to C = C + A
-=	Subtract AND assignment operator, it subtract right operand from the left operand and assign the result to the left operand	C -= A is equivalent to C = C - A
*=	Multiply AND assignment operator, it multiply right operand to the left operand and assign the result to the left operand	C *= A is equivalent to C = C * A
/=	Divide AND assignment operator, it divides left operand with the right operand and assign the result to left operand	C /= A is equivalent to C = C/A
%=	Modulus AND assignment operator, it takes modulus using two operands and assign the result to left operand.	C% = A is equivalent to C = C%A

5. Conditional Operator

There is one more operator called conditional operator. This first evaluates an expression for a true or false value and then execute one of the two given statements depending upon the result of the evaluation. The conditional operator has this syntax.

Show Example:

Operator	Description	Example
?:	Conditional expression	If condition is true ? then value x otherwise value y

Operator Categories

All the operators we have discussed above can be categorised into following categories:

- Unary prefix operators, which precede a single operand.
- Binary operators, which take two operands and perform a variety of arithmetic and logical operations.
- The conditional operator (a ternary operator), which takes three operands and evaluates either the second or third expression, depending on the evaluation of the first expression.
- Assignment operators, which assign a value to a variable.

Precedence of PHP Operators

Operator precedence determines the grouping of terms in an expression. This affects how an expression is evaluated. Certain operators have higher precedence than others. For example, the multiplication operator has higher precedence than the addition operator. So multiplication is performed before the addition operation. Within an expression, higher precedence operators will be evaluated first.

218		
Category	Operator	Associativity
Unary	!++ --	Right to left
Multiplicative	* / %	Left to right
Additive	+ -	Left to right
Relational	<<= >>=	Left to right
Equality	= = !=	Left to right
Logical AND	&&	Left to right
Logical OR		Left to right
Conditional	?:	Right to left
Assignment	= += -= *= /= %=	Right to left

Conditional Statement. Loops

PHP Conditional Statements

- Very often when you write code, you want to perform different actions for different conditions. You can use conditional statements in your code to do this.
- In PHP we have the following conditional statements:

If statement: executes some code if one condition is true.

Example :

```
<!DOCTYPE html>
```

```
<html>

<body>

<?php

$t = date("H");

if ($t < "20") {

    echo "Have a good day!";

}

?>

</body>

</html>
```

Output: Have a good day

If . . .else statement: executes some code if a condition is true and another code if that condition is false.

Example:

```
<!DOCTYPE html>

<html>

<body>

<?php

$t = date("H");

if ($t < "20") {

    echo "Have a good day!";

} else {

    echo "Have a good night!";

}

?>

</body>
```

```
</html>
```

Output: Have a good day

If . . . elseif . . . else statement: executes different codes for more than two conditions.

Example:

```
<!DOCTYPE html>

<html>

<body>

<?php

$t = date("H");

echo "<p>The hour (of the server) is " . $t;

echo ", and will give the following message:</p>";

if ($t < "10") {

    echo "Have a good morning!";

} elseif ($t < "20") {

    echo "Have a good day!";

} else {

    echo "Have a good night!";

}

?>

</body>

</html>
```

Output:

The hour (of the server) is and will give the following message

Have a good day

Switch statement: selected one of many blocks of code to be executed.

Example:

```
<!DOCTYPE html>

<html>

<body>


<?php

$favcolor = "red";


switch ($favcolor) {

    case "red":

        echo "Your favorite color is red!";

        break;

    case "blue":

        echo "Your favorite color is blue!";

        break;

    case "green":

        echo "Your favorite color is green!";

        break;

    default:

        echo "Your favorite color is neither red, blue, nor green!";

}

?>

</body>

</html>
```

Output : Your favorite color is red.

PHP Loops

- Often when you write code, you want the same block of code to run over and over again a certain number of times. So, instead of adding several almost equal code-lines in a script, we can use loops.
- Loops are used to execute the same block of code again and again, as long as a certain condition is true.
- In PHP, we have the following loop types:

While: loops through a block of code as long as the specified condition is true.

Example:

```
<!DOCTYPE html>

<html>

<body>

<?php

$x = 1;

while($x <= 5) {

    echo "The number is: $x <br>";

    $x++;

}

?>

</body>

</html>
```

Output:

```
The number is 1

The number is 2

The number is 3

The number is 4

The number is 5
```

Explanation:

\$x = 1; - Initialize the loop counter (\$x), and set the start value to 1

\$x <= 5 - Continue the loop as long as \$x is less than or equal to 5

\$x++; - Increase the loop counter value by 1 for each iteration

Example 2:

```
<!DOCTYPE html>

<html>

<body>

<?php

$x = 0;

while($x <= 100) {

    echo "The number is: $x <br>";

    $x+=10;

}

?>

</body>

</html>
```

Output:

The number is 10

The number is 20

The number is 30

The number is 40

The number is 50

The number is 60

The number is 70

The number is 80

The number is 90

The number is 100

Explanation:

\$x = 0; - Initialize the loop counter (\$x), and set the start value to 0

\$x <= 100 - Continue the loop as long as \$x is less than or equal to 100

\$x+=10; - Increase the loop counter value by 10 for each iteration

Do . . while: loops through a block of code once, and then repeat the loop as long as the specified condition is true.

Example:

```
<!DOCTYPE html>

<html>

<body>

<?php

$x = 1;

do {

    echo "The number is: $x <br>";

    $x++;

} while ($x <= 5);

?>

</body>

</html>
```

Output:

The number is 1

The number is 2

The number is 3

The number is 4

The number is 5

- **For:** loops through a block of code for each element in an array.

Example:

```
<!DOCTYPE html>

<html>

<body>

<?php

for ($x = 0; $x <= 10; $x++) {

    echo "The number is: $x <br>";

}

?>

</body>

</html>
```

Output:

```
The number is 1
The number is 2
The number is 3
The number is 4
The number is 5
The number is 6
The number is 7
The number is 8
The number is 9
The number is 10
```

- **For-each :** For every loop iteration, the value of the current array element is assigned to \$value and the array pointer is moved by one, until it reaches the last array element.

Example:

```
<!DOCTYPE html>

<html>

<body>

<?php

$colors = array("red", "green", "blue", "yellow");

foreach ($colors as $value) {

    echo "$value <br>";

}

?>

</body>

</html>
```

Output:

```
red
green
blue
yellow
```

Array

- An array is a special variable, which can hold more than one value at a time.
- If you have a list of items (a list of car names, for example), storing the cars in single variables could look like this:

```
$cars1 = "Volvo";

$cars2 = "BMW";

$cars3 = "Toyota";
```

- An array can hold many values under a single name, and you can access the values by referring to an index number.

Create an Array in PHP

- In PHP, the array() function is used to create an array:

`array();`

- In PHP, there are three types of arrays:

1. Indexed arrays: Arrays with a numeric index

Example:

```
<!DOCTYPE html>

<html>

<body>

<?php

$cars = array("Volvo", "BMW", "Toyota");

    $arlength = count($cars);

    for($x = 0; $x < $arlength; $x++) {

    echo $cars[$x];

    echo "<br>";

    }

?>

</body>

</html>
```

Output:

```
Volvo

BMW

Toyota
```

2. Associative arrays: Arrays with named keys

Example:

```
<!DOCTYPE html>

<html>

<body>
```

```

<?php

$age = array("Peter"=>"35", "Ben"=>"37", "Joe"=>"43");

echo "Peter is " . $age['Peter'] . " years old.";

?>

</body>

</html>

```

Output:

Peter is 35 years old.

Multidimensional arrays: Arrays containing one or more arrays.

Example:

```

<!DOCTYPE html>

<html>

<body>

<?php

$cars = array (

    array("Volvo",22,18),

    array("BMW",15,13),

    array("Saab",5,2),

    array("Land Rover",17,15)

);

echo $cars[0][0].": In stock: ".$cars[0][1].", sold: ".$cars[0][2]."<br>";

echo $cars[1][0].": In stock: ".$cars[1][1].", sold: ".$cars[1][2]."<br>";

echo $cars[2][0].": In stock: ".$cars[2][1].", sold: ".$cars[2][2]."<br>";

echo $cars[3][0].": In stock: ".$cars[3][1].", sold: ".$cars[3][2]."<br>";

?>

</body>

```

</html>

Output:

Volvo: In stock: 22, sold: 18.

BMW: In stock: 15, sold: 13.

Saab: In stock: 5, sold: 2.

Land Rover: In stock: 17, sold: 15.

Get The Length of an Array : The count() Function

The count() function is used to return the length (the number of elements) of an array.

GET and POST Method and Session

GET and POST method in PHP

- There are two ways the browser client can send information to the web server.
 - The GET Method
 - The POST Method
- Before the browser sends the information, it encodes it using a scheme called URL encoding. In this scheme, name/value pairs are joined with equal signs and different pairs are separated by the ampersand.

name1=value1&name2 =value2&name3 =value3

- Spaces are removed and replaced with the + character and any other nonalphanumeric characters are replaced with a hexadecimal values. After the information is encoded it is sent to the server.

The GET Method

- The GET method sends the encoded user information appended to the page request. The page and the encoded information are separated by the ? character.
 - The GET method produces a long string that appears in your server logs, in the browser's Location: box.
 - The GET method is restricted to send upto 1024 characters only.
 - Never use GET method if you have password or other sensitive information to be sent to the server.

- GET can't be used to send binary data, like images or word documents, to the server.
- The data sent by GET method can be accessed using QUERY_STRING environment variable.
- The PHP provides \$_GET associative array to access all the sent information using GET method.

Example:

```
<?php
If ($_GET ["Name"] || $_GET ["age"])
{
echo "Welcome". $_GET['name']. "<br/>";
echo "you are". $_GET['age']. "years old";

    exit()
}

?>

<html>

    <body>

        <form action = "<?php $_PHP_SELF ?>" method "GET">

Name: <input type = "text" name = "name"/>

Age: <input type = "text" name = "age"/>

<input type = "submit"/>

</form>

</body>

</html>
```

The POST Method

- The POST method transfers information via HTTP headers. The information is encoded as described in case of GET method and put into a header called QUERY_STRING.
- The POST method does not have any restriction on data size to be sent.

- The POST method can be used to send ASCII as well as binary data.
- The data sent by POST method goes through HTTP header so security depends on HTTP protocol. By using Secure HTTP (https) you can make sure that your information is secure.
- The PHP provides \$_POST associative array to access all the sent information using GET method.
- **Example:**

```
<?php
If ($_POST ["name"] || $_POST["age"]);
{
{
echo "Welcome".$_POST['name'] "<br/>";
echo "you are".$_POST['age']. "years old";
    exit()
}
?>

<html>

<body>

<form action = "<?php $ PHP_SELF?" method = "POST">
Name: <input type = "text" name = "name"/>
Age: <input type = "text" name = "age"/>
<input type = "submit"/>
</form>
</body>
</html>
```

- An alternative way to make data accessible across the various page of an entire website is to use a PHP session.
- A session creates a file in a temporary directory on the server where registered session variables and their values are stored. This data will be available to all pages on the site during that visit. The location of the temporary file is determined by setting in the php.ini file called session.save.path.
- When a session is started following things happen:
 - PHP first creates a unique identifier for that particular session which is a random string of 32 hexadecimal numbers.
 - A cookie called PHPSESSID is automatically sent to the user's computer to store unique session identification string.
 - A file is automatically created on the server in the designated temporary directory and bears the name of the unique identifier prefixed by sess- ie.

When a PHP script wants to retrieve the value from a session variable, PHP automatically gets the unique session identifier string from the PHPSESSID cookie and then looks in its for the file bearing that name and a validation can be temporary directory done by comparing both values.

A session ends when the user loses the browser or after leaving the site, the server will terminate the session after a predetermined period of time, commonly 30 minutes duration.

Starting a PHP Session

- A PHP session is easily started by making a call to the session_start(), function. This function first checks if a session is already started and if none is started then it starts one. It is recommended to put the call to session_start() at the beginning of the page.
- Session variables are stored in associative array called \$_SESSION []. These variables can be accessed during lifetime of a session. The following example starts a session then register a variable called counter that is incremented each the time page is visited during the session. Make use of isset(function to check if session variable is already set or not. Put this code in a test.php file and load this file many times to see the result.

```
<?php
```

```
session_start();
```

```
If (isset($_SESSION['Counter']))
```

```
{
```

```

        $_SESSION['Counter']+=1;

    }

    else

    {

        $_SESSION ['Counter'] 1;

        $msg = "you have visited this page".$_SESSION['Counter'];

        $msg = "In this session";

    ?>

<html>

<head>

<title> Setting up a PHP session </title>

</head>

<body

<?php echo ($msg);?>

</body>

</html>

```

Destroying a PHP Session

- A PHP session can be destroyed by session_destroy () function. This function doesnot need any argument and a single call can destroy all the session variables. If you want to destroy a single session variable then you can use unset () function to unset a session variable.
- Here is the example to unset a single variable

```

<?php

unset ($_SESSION ['Counter']);

?>

```

Here is the call which will destroy all the session variables:

```

<?php

session_destroy( );

```

?>

PHP Cookies

Cookies are text files stored on the client computer and they are kept of use tracking purpose. PHP transparently supports HTTP cookies. There are three steps involved in identifying returning users:

- Server script sends a set of cookies to the browser. For example name, age or identification number etc.
- Browser stores this information on local machine for future use.
- When next time browser sends any request to web server then it sends those cookies information to the server and server uses that information to identify the user.

Setting Cookies with PHP

PHP provided `setcookie ( )` function to set a cookies. This function, requires upto six arguments and should be called before and `<html>` tag. For each cookie this function has to be called separately.

`Setcookie(name, value, expire, path, domain, security);`

Here is the detail of all the arguments:

Name: This sets the name of the cookie and is stored in an environment variable called `HTTP_COOKIE_VARS`. This variable is used while accessing cookies.

Value: This sets the value of the named variable and is the content that you actually want to store.

Expiry: This specify a future time in seconds since 00:00:00 GMT on 1st Jan. 1970. After this time cookie will become inaccessible. If this parameter is not set then cookie will automatically expire when the web browser is closed.

Path: This specifies the directories for which the cookie is valid. A single forward slash character permits the cookie to be valid for all directories.

Domain: This can be used to specify the domain name is very large domains and must contain at least two periods to be valid. All cookies are only valid for the host and domain which created them.

Security: This can be to 1 to specify that the cookie should only be sent by secure transmission using HTTPS otherwise set to 0 which mean cookie can be sent by regular HTTP.

Example: This example will create two cookies name and age these cookies will be expired after one hour.

```

<?php

Setcookie ("name", "John Watkin", time( ) + 3600, "/", " ", 0);

Setcookie ("age", "36", time\ ) + 3600, "/", "", 0);

<html>

<head>

<title> setting cookies with PHP </title>

</head>

<body>

<? php echo "set cookies"?>

</body>

</html>

```

Accessing Cookies with PHP

PHP provides many ways to access cookies. Simplest way is to use either \$_cookie or \$HTTP_COOKIE_VARS variables. Following example will access all the cookies set in above example.

```

<html>

<head>

<title> Accessing cookies with PHP </title>

</head>

<body>

<?php

    echo$_COOKIE ["name"]."<br/>";

/*is equivalent to */

echo $HTTP_COOKIE_VARS ["name"]. "<br/>" ;

echo$_COOKIE ["age"]."<br/>";

/is equivalent to */

echo $HTTP_COOKIE_VARS [" name"]."<br/>";

```

```
?>
```

```
</body>
```

```
</html>
```

Deleting Cookie With PHP

Officially, to delete a cookie you should call `setcookie( )` with the name argument only but this does not always work well, however, and should not be relied on.

It is safest to set the cookie with a date that has already expired:

```
<?php
```

```
setcookie("name", " ", time( ) - 60, "/", " ", 0);
```

```
setcookie ("age", " ", time( ) - 60, "/", " ", 0);
```

```
?>
```

```
<html>
```

```
<head>
```

```
<title> Deleting Cookies with PHP </title>
```

```
</head>
```

```
<body>
```

```
<?php echo "Deleted Cookies"?>
```

```
</body>
```

```
</html>
```