WDD 201

Fundamentals of Web Designing

Skill Enhancement Course (SEC)

School of Vocational Studies

उत्तराखण्ड मुक्त विश्वविद्यालय

तीनपानी बाईपास रोड, ट्रांसपोर्ट नगर के पास, हल्द्वानी- 263139
फोन न.- 05946 - 261122, 261123
टॉल फ्री न.- 18001804025
फैक्स न.- 05946-264232, ई-मेल- info@uou.ac.in
वेबसाइट- www.uou.ac.in

WDD 201

Fundamentals of Web Designing

Skill Enhancement Course (SEC)

Note: Edited Version awaited. Wrap up material for study.

Formatting & Typesetting
Dr. Gopal Datt, Assistant Professor
Ms. Riya Giri, Assistant Professor
School of Vocational Studies, Uttarakhand Open University, Haldwani.

Edition- 2024, Pre-Publication copy for Limited Circulation, ISBN:
Publication- Uttarakhand Open University, Haldwani, Nainital.
E- Mail: studies@uou.ac.in
Printer:

WDD 201- Fundamental of Web Designing

INDEX

UNIT 1

INTRODUCTION TO COMPUTER NETWORKS

1.1 INTRODUCTION

This unit presents a broad survey of concepts and terminology related to today's digital computers. Here, the idea of computer literacy is introduced. You will find what a computer is and what it does. You will learn about computer generations, applications of computer, components and categories of a computer (including personal computers, minicomputers, mainframe computers, and supercomputers) and etc. This unit also covers the usefulness of computers, different functional units of a computer, etc.

1.2 OBJECTIVES

After successful completion of this unit, you will be able to-

- Understand computer networks.
- Types of computer networks and Network topology
- Know various devices used in Networking
- Know applications of computer networks

1.3 COMPUTER NETWORKS

Networking is a process of connecting two or more devices and allows the sharing of data/information. A computer network is a collection of computers or network devices; connected to one another for the purpose of sharing information among them. If you want to understand network in real life, so Internet is an excellent example of the network.

Networking is used to share resources (software and hardware both) such as- file, documents, e-mail, printer, storage, etc. The computer network may be connected using wires/cables (telephone lines, cable TV, fiber optics, etc) or wireless (radio waves, satellite, infrared beams, etc).

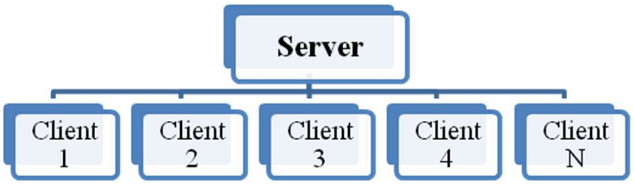


Figure 3.1- Client Server Architecture

Broadly there are two basic forms of computer network design/architecture. As-

- Client server architecture
- Peer-to-peer architecture

In client server network architecture, every computer has a distinct role: that of either a client or a server. A server is featured as a centralized storage, that stores- email, Web pages, files, applications; and designed to share its resources among the client computers on the network. A client computer can communicate only with server computers. A server (or Server Computer) is a single computer or group of servers that are interconnected. A server runs a special network operating system- such as: Windows Server, Linux, UNIX, etc; which is designed to facilitate sharing of its resources among its clients. Figure 3.1 shows client server architecture with single server and 'N' clients.

A peer-to-peer (P2P) network is created when two or more computers are connected and share resources without using a separate server computer. Peers are both suppliers and consumers of resources, in contrast to the client-server model in which the consumption and supply of resources is divided in to server and clients respectively.

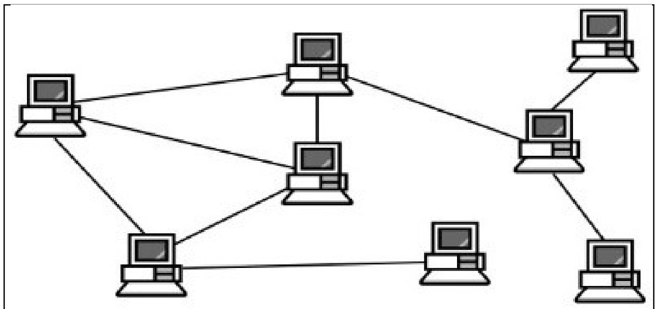


Figure 3.2- Peer to Peer Architecture

Advantages of Computer Network-

- A network provides the means to exchange data/information among the computers.
- It permits the sharing of the resources of the machine.
- A network provides multiple users to access same data at the same time from same or different locations.
- Video Conferencing is possible through a network

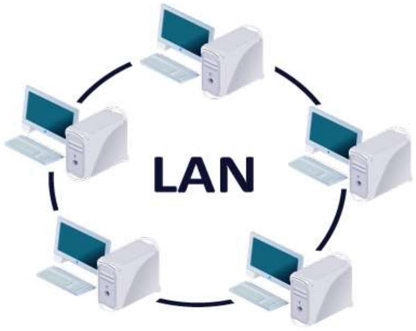
Disadvantages of Computer Network-

- Risk of security issues.
- It encourages people to become dependent on computers.
- A network fault can cause loss of data.
- Complex structure and costlier.

Types of Computer Network

Local Area Network (LAN)

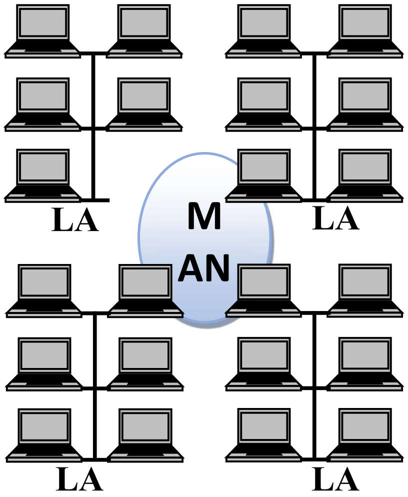
A LAN is a network that is used for communicating among computer devices, usually within an office building or home. It typically consists of several devices (such as-



computers and printers) connected to one another in a network. It enables the sharing of resources such as files or hardware devices that may be needed by multiple users. LANs are the most common form of networks found in most small businesses, offices and schools. A LAN can be connected using wired or wireless, when it is connected as wireless then it is named WLAN (Wireless).

Metropolitan Area Network (MAN)

It consists of two or more LANs connected together using high-speed (1.544 Mbps to 155 Mbps) transmission lines. It connects smaller networks (LAN) within a city or town. It is similar to a LAN in a sense that it connects computers but larger in size because it connects two or more LANs in a certain geographic area. A MAN might be owned and operated by a single organization and typically covers an area of 5 to 50 kilometer in diameter. Examples: Telephone company network, Cable TV network, etc.



Wide Area Network (WAN)

It covers a large geographic area such as country, continent or even whole of the world. A WAN is consisting of two or more MANs connected together; as same the MAN is consisting of two or more LANs are connected together. WAN overcome the distance limitations by transmitting data over leased high-speed phone lines or wireless links such as satellites. WANs can use either analog (telephone lines) or digital (such as satellite transmission) signals, or a combination of both.

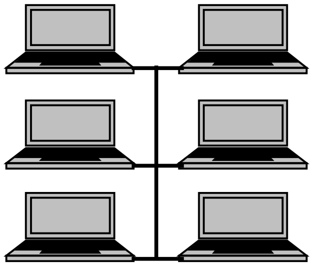
Applications of Computer Networks-

- It provides a reliable communication medium.
- It provides resource sharing.
- It is used for serving information in remote locations.
- It is used for social media and communication such as e-mail, chatting, video conferencing, messaging, etc.
- It is used for electronic banking.

1.4 COMPUTER NETWORK TOPOLOGIES

The topology of a network is the geometric representation of all the links and connected devices in a network to one another. In other words, network topology is an arrangement of network devices those are used to transmit data in a network and it also presents the structure of a network. The layout of the network connection is based on the fact that how the data is being transmitted and how are the devices linked to each other. There are the following types of network topology-

Bus Topology-



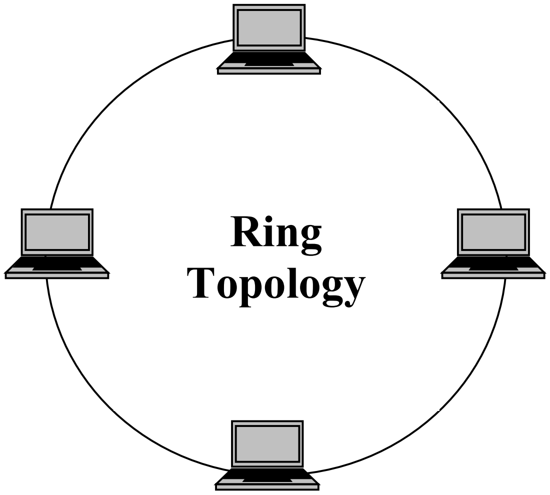
A bus topology is a multipoint connection, one long cable acts as a backbone to connect all the devices in a network. Nodes are connected to the bus cable by drop lines and taps. A drop line is a connection between the device and the main cable and a tap is a connector that either splices into the main cable or punctures the sheathing of a cable to create a connection with main bus. Number of taps and distance between them is limited. Basically, bus topology is popular in LAN (Local Area Network) where a greater number of nodes are connected using a single cable.

Advantages-

- It is easy to install/configure.
- It uses less cabling than mesh or star topology.

Disadvantages-

- Difficult to reconnection and fault isolation.
- Signal reflection at the taps can cause degradation in quality.
- Adding new devices sometimes may cause modification or replacement of the backbone.
- A fault or break in the bus/backbone cable stops all transmission in a network.



Ring Topology

In a ring topology, each device has a dedicated point-to-point connection with only the two devices on either side of it. A signal is passed along the ring in one direction, from device to device, until it reaches its destination.

Advantages-

- It is easy to install and reconfigure.
- Each node has equal access to resources.
- It reduces chances of collision of data packets.

Disadvantages-

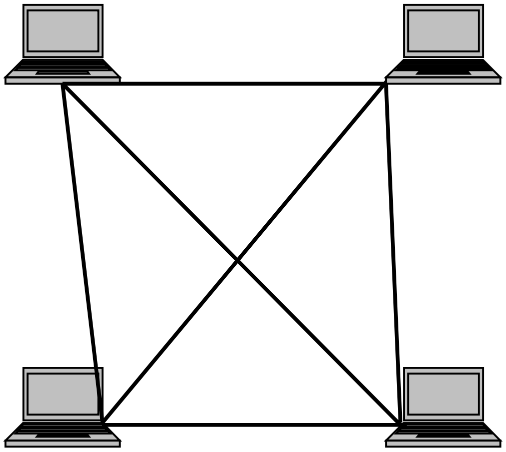
- If one node or port goes down, the entire network gets affected.

- Network is highly dependent on the wire which connects different components.
- Slower in data transmission in comparison to other networks.

Mesh Topology-

In a mesh topology, every device (Computer or node) has a dedicated point-to-point link to every other device. The term dedicated means that the link carries data traffic only between the two devices it connects.

Advantages-



- A mesh topology uses dedicated links which guarantees each connection can carry its own data load.
- It eliminates the data traffic problems because of dedicated links.
- A mesh topology is robust because if one link becomes unusable, it does not incapacitate the entire system.
- It provides secure communication among stations, when each message travels along a dedicated line, only the intended recipient sees it.
- A point-to-point links make fault identification and fault isolation easy.

Disadvantages-

- Amount of cabling and input/output ports used is too much.
- Installation and reconnection is difficult.
- It is too expensive; because hardware (cable and I/O ports) required connecting each link is much larger.
- It is usually implemented in a limited fashion.

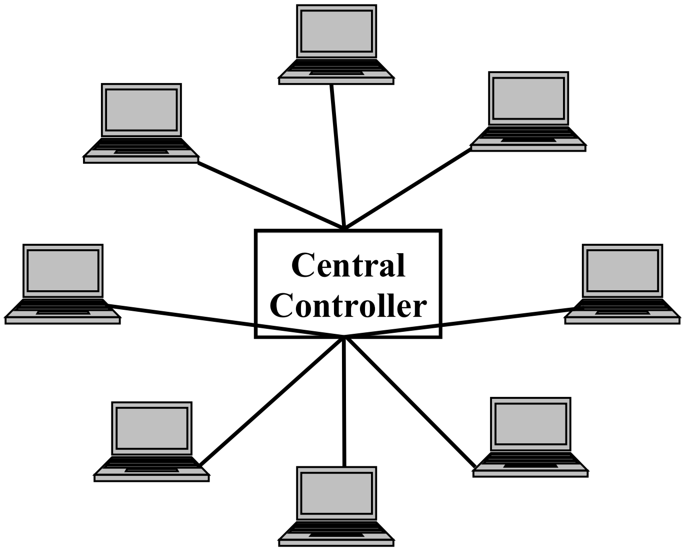
Star Topology-

In a star topology, each device has a dedicated point-to-point link only to a central controller, usually called a hub or switch. The devices are not directly linked to one another. It does not allow direct data traffic/transmission between devices. The central controller acts as an exchange: If one device wants to send data to another, it sends the data to the controller, which then relays the data to the other connected device.

Advantages-

- A star topology is less expensive than a mesh topology.

- In a star topology, each device needs only one link and one I/O port to connect it to any number of other devices.
- Easy to install and reconfigure, if one link fails, only that link is affected.
- It needs less cabling and easy to fault identification & isolation.



Disadvantages-

- Fully dependent on one single point, called the central hub/switch.
- If the central hub goes down, the whole system is unable to work.

1.5 HARDWARE DEVICES USED IN COMPUTER NETWORKS

Hub-

Figure: Sample Hub (A Network connecting device)

A hub connects multiple wires in a Local Area Network (LAN) coming from different nodes. For example- the hub connects different stations in star topology. A Hub cannot filter data packets, received from different stations, so that the data packets are sent to all connected devices; which leads increasing data traffic and wastage of resources. Also, a hub cannot find best path for data packets to deliver it into destination. Hub only supports half duplex, which means at a same point of time only one way communication possible. In addition to that, a hub operates at the physical layer of the OSI/RM (Open System Interconnection / Reference Model).

Hubs are comes in two categories- Active Hub and Passive hub. Most hubs are active hubs, which are also called multiport repeaters. It functions in the same way as a repeater, except that it must also broadcast the regenerated signals to all connected nodes/stations rather than forwarding them to just one node/station. A passive hub provides no signal regeneration, it simply serves as a conduit for data and broadcast incoming data to every other port.

Figure: Sample Switch

Switch-

A switch is a connecting device which is somehow works in the place of a hub. Unlike hub, switches work on the layer 2 (data link layer) of the OSI model and also follows full duplex (two-way communication at the same point of time) mode. Switches works on local area network to connect different nodes and also network types (such as Ethernet and Fast Ethernet) or networks of the same type. It filters each packet and processes it accordingly.

Repeater-

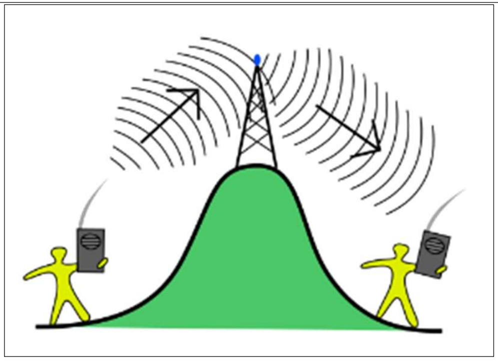


Figure: How Repeater works?

Figure: Sample Repeater

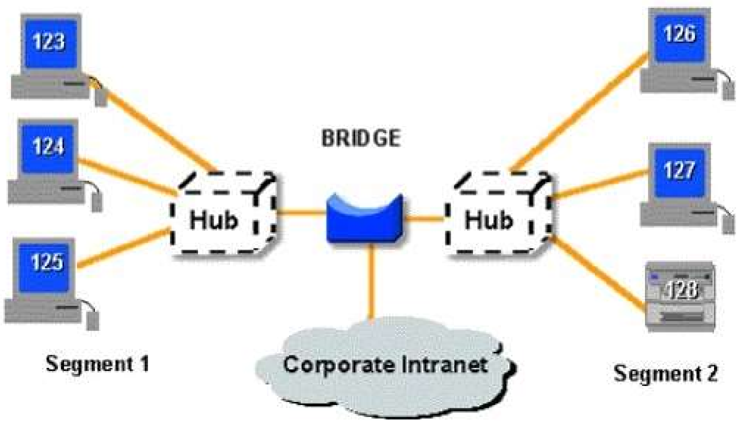


Figure- Example Bridge

A repeater is an electronic device that receives the signal and regenerates it, so that the signal can cover longer distances. It operates at the physical layer of the OSI/RM. When the signal becomes weak, it (repeater) copies the signal bit by bit and regenerates it at the original strength.

Bridge-

A bridge connects a local area network (LAN) to another local area network and also controls data flow between them. It works at the data-link layer of the OSI model. Bridges can operate only in half duplex mode where as a switch operates in both half or full duplex mode. Bridges are important to connect different LANs in one network, so that they can become part of the whole network. A bridge sometimes also known as 'brouter', it is used as a combination of router and bridge both. Brouters are the earlier implementation of the routers.

Router-

Router is a network device (works on network layer of OSI model) that routes data packets around the network. A router has ability to connect dissimilar LANs with same protocol. It processes logical addressing information (IP address) in the Network header of a data packet.

Figure- Example Router

A router manages its routing table which can be built statically or dynamically. A router keeps record of the path that a data packets can use as it moves across the network, this type of records are maintained in a table known as routing table. A router can determine the destination address for the data packet and then, by using tables of defined routes (routing table), the router determines the optimal way for the data packet to forward it next or to its destination.

Gateway-

A gateway is used to connect two or more networks using different protocols together; where as a bridge is used to connect two similar types of networks. The term gateway is applied to any device or software application that can perform the function of translating data from one format to another without changing the data itself. For example- you can understand the working of a router that connects a home/local area network to WAN/Internet.

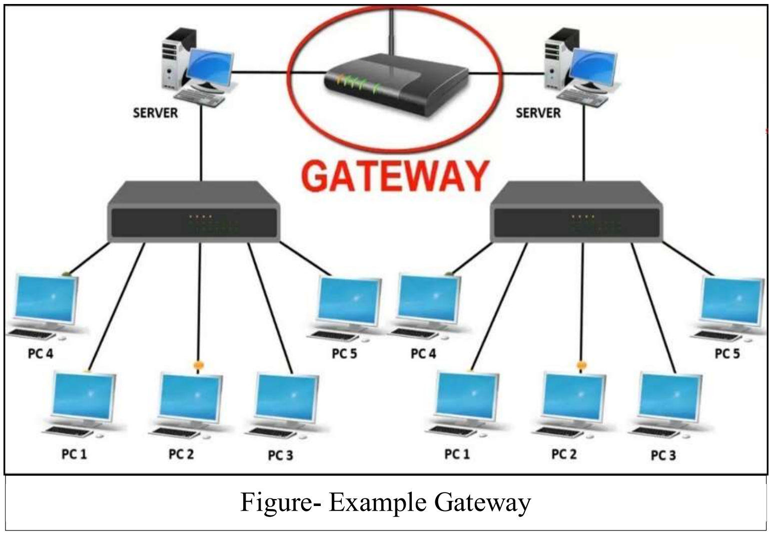


Figure- Example Gateway

1.6 POINTS TO REMEMBER

- The definition of Local Area Network, Peer to Peer and Client/Server network.
- Relationships among Local area network, Metropolitan Area Network and Wide Area Network.
- The graphical representation of LANs, MANs, and WANs.
- The common resources shared within a network.

1.7 GLOSSARY

- **CCITT**: Consultative Committee for International Telegraphy and Telephony
- **ASCII**: American Standard Code for Information Interchange
- **IEEE**: Institute of Electrical and Electronics Engineers

- **ISO**: International Organization for Standardization
- **ITU-T**: International Telecommunication Union-Telecommunication Standards Sector
- **ISP**: Internet service provider
- **TCP/IP**: Transmission Control Protocol/ Internetworking Protocol

1.8 CHECK YOUR PROGRESS

Part- A (Multiple choice type questions)

a) Multipoint topology is-
  A. Bus topology
  B. Star topology
  C. Mesh topology
  D. Ring topology

b) A communication path way that transfers data from one point to another is called-
  A. Node
  B. Medium
  C. Topology
  D. Link

c) Synonymous of rule is-
  A. Standard
  B. Protocol
  C. Forum
  D. Agency

d) Nodes are another name of-
  A. Devices
  B. Links
  C. Medium
  D. Modes

e) Bus, ring and star topologies are mostly used in the-
  A. LAN
  B. MAN
  C. WAN
  D. Internetwork

f) OSI stands for-
  A. Open Systems Interconnection
  B. Online Systems Interconnection
  C. Open Systems Internet
  D. Online systems Interconnection

g) Parameter that refers to set of rules that govern data communications are called-
  A. Forum
  B. Standard
  C. Agency
  D. Protocol

h) Repeater is used in-
   A. Mesh Topology
   B. Ring Topology
   C. Bus Topology
   D. Star Topology

i) ISP stands for-
   A. Internet Service Provider
   B. Internet System Provider
   C. International Service Provider
   D. International System Program

j) Which transmission media has the highest transmission speed in a network?
   A. Coaxial cable
   B. Twisted pair cable
   C. Optical fiber
   D. Electrical cable

Part- B (True/False type questions)

a) A network interface card (NIC) is an expansion card or a PC Card that connects your computer to a network. [T/F]

b) A switch is a device that connects multiple computers into a network in which multiple communications links can be in operation simultaneously. [T/F]

c) A communications protocol (protocol) is a set of rules that every computer follows to transfer information. [T/F]

d) A MAN covers a limited geographic distance, such as an office, building, or a group of buildings. [T/F]

e) Coaxial cable has one central wire surrounded by insulation, a metallic shield, and a final case of insulating material. [T/F]

f) Bandwidth refers to the capacity of the communications line. [T/F]

g) In a peer-to-peer network, all data and software are stored on a centralized server. [T/F]

h) A local area network using wireless technology is termed as WLAN [T/F]

i) Bandwidth is measured in kbps, mbps, and gbps. [T/F]

j) Network Hub is a device that connects multiple computers into a network. [T/F]

Part C- (Descriptive type questions)

(1) Identify the five components of a data communications system.

(2) What are the advantages of a multipoint connection over a point-to-point connection?

(3) Define the network topology.

(4) What is the difference between half-duplex and full-duplex transmission modes?

(5) Name the four basic network topologies with their advantages and disadvantages.

(6) List at least 5 common resources shared over a network.

Answers- (Part A- Multiple choice type questions)		
(a) A	(b) D	(c) B
(d) A	(e) A	(f) A
(g) D	(h) B	(i) A
(j) C		
Answers- (Part B- True/False type questions)		
(a) True	(b) True	(c) True
(d) False	(e) True	(f) True
(g) False	(h) True	(i) True
(j) True		

1.9 BIBLIOGRAPHY/REFERENCES

- https://www.geeksforgeeks.org/computer-network-tutorials/, Nov 2017
- http://www.e-tutes.com/, Nov 2017
- Forouzan, Behrouz A., Data Communications and Networking, McGraw-Hill, Inc. New York

1.10 SUGGESTED READINGS

- Mir, N.F. (2006) Computer and Communication Networks, Prentice Hall.
- Forouzan, Behrouz A., Data Communications and Networking, McGraw-Hill, Inc. New York.
- Tanenbaum, A.S. (2002) Computer Networks, Prentice Hall.

UNIT 2

INTRODUCTION TO INTERNET- I

2.1 INTRODUCTION

There are various things you can do on the Internet. There are many ways to keep up

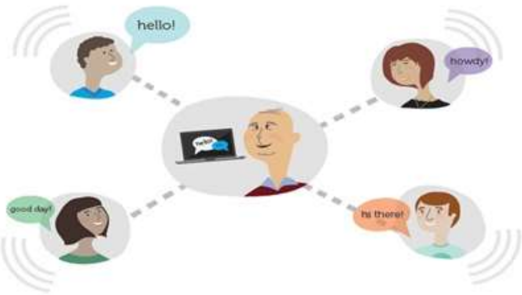


Figure 4.1: connection of people through Internet

with news or buy anything online. You can pay your bills, manage your bank accounts, meet new people, watch TV, or learn new skills. You can gain knowledge from the internet or perform almost anything online.

Current use of Internet is becoming quite popular for accessing information on any topic of your interest. Nowadays, most of the users who have computers around themselves use Internet to access information from the World Wide Web, exchange messages & documents and e-services.

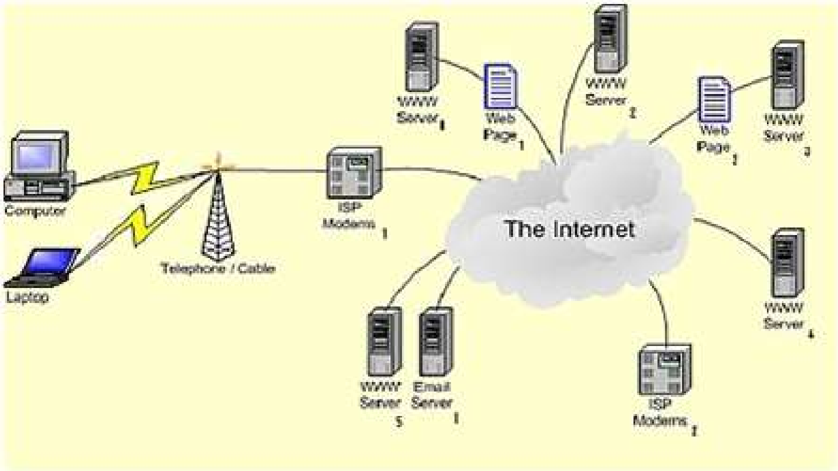


Figure 4.2: Simple diagram of Internet

The internet is a collection of many small networks through which one host can be connected to another host anywhere in the world to share all kinds of information such as text, images, audio, video etc. The internet is the largest computer network of many small private, public, academic, business and government networks of computers. In the present age of information Technology, use of Internet is becoming quite popular for accessing information on any topic of your interest. It also provides tremendous opportunities to students; researchers and professionals for getting information on matters related to academic and professional topics and lot more. The origins of the Internet date back to research commissioned by the United States Federal Government in the 1960s to build robust, fault-tolerant communication via computer networks. The linking of commercial networks and enterprises in the early 1990s marked the beginning of the transition to the modern Internet, and generated rapid growth as institutional, personal, and mobile computers were connected to the network. By the late 2000s, its services and technologies had been incorporated into virtually every aspect of human lives.

The Internet is aimed at enabling the interconnection and integration of the physical world and the cyberspace. It represents the trend of future networking, and leads the third wave of the IT industry revolution. Today everywhere we can see internet.

2.2 OBJECTIVES

After successful completion of this unit, you will be able to-

- Define Internet and search desired information over the internet.

- List different types of Internet connections
- Explain various terminology used in internet
- Use various services provided by internet

2.3 WHAT IS INTERNET?

“The network of networks is called an internet”.

“The Internet or simply the Net is a worldwide network of computer networks. It is an interconnection of large and small networks around the world”. We will discuss common internet terms to help you understand the relationship of various Internet technologies.

Important terms used in Internet-

World Wide Web (WWW)

WWW is the abbreviation of World Wide Web. The World Wide Web is a set of electronic documents (called web pages) that are attached together like a spider web. These web pages are saved on computer and those computers are called server which are located around all over the world.

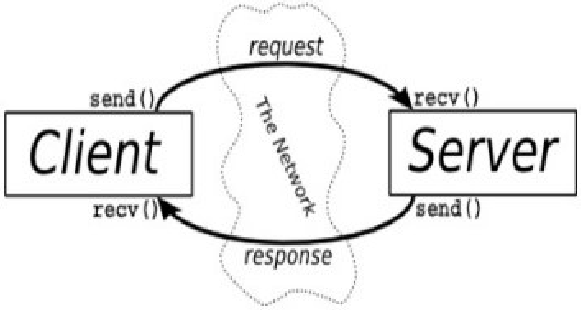


Figure 4.3: Client Server architecture

Web Server

A Web Server is responsible for stores web pages and it is also responsible for accepting request(s) from host computer and serves them with web pages. Two main web server application are: IIS (Internet Information server) and Apache, etc. Web servers are connected to the Internet all the time.

Hyperlink

Hyperlink is an icon, graphic, or text in a document that links to another file or object. It is an electronic document that connected to another place in the same document or to an entirely different document. Hyperlinks usually display as underlined text and in an altered colour, but they may also appear as graphics, such as buttons to click. Hyperlinks may be used to connect another place in the same document, or another page, to play an audio or video file, to download a file, to set up a message to an e-mail address, and to connect the other Internet resources.

HTML (Hypertext Markup Language)

It is a language that consists of certain key words called ‘Tags’, used for writing the documents on the web. We will discuss in Unit 6 of this book.

Web Page

A web page (such as the one you are looking at now) is an electronic document written in a computer language called HTML (Hypertext Markup Language).Web pages can contain text, graphics, video, animation, and sound, as well as interactive features, such as data entry forms. Each page has a unique address known as a URL (Uniform Resource Locator) that identifies its location on the server. Web pages usually contain hyperlinks to other web pages.

Website

A website (often shortened to just site) is one or more web pages, belonging to a particular company, institute, government or an individual. The first page is called the home page, which acts like an index, indicating the content on the site. By default the home page is named as index.htm. From the home page, you can click hyperlinks to access other web pages.

Evolution of the Internet-

The process of development of internet was initiated in 1969, when U.S. Department of Defence initiated a project named Advance Research Project Network (ARPANET). The major purpose of the ARPANET model was to develop a continuous communication channel for researchers of different universities by creating computer network which can link computers at different university and U.S. defence. ARPANET was initiated with a small number of computers but it expanded rapidly as the huge number of researchers from hundreds of colleges and universities started sharing information's and data through its networks.

Then in mid 80's another central organization, the National Science Foundation, created a new, more superior and high power network called NSFnet. Later, large number of personal companies started their own networks, which were later interrelated along with ARPANET and NSFnet to form internet and its protocols to ensure transferring of information over the world wide network as unread.

Purpose of the Internet-

Internet has been the mainly helpful technology of the modern times which helps us not only in our day by day life, but also our private and professional lives developments. The internet gives us achieve this in various different ways. For the students and institutional purposes the internet is broadly used to gather information so as to do the research or add to the information of various subjects. Even in the business fields and the professionals like doctors, access the internet to filter the compulsory knowledge for their use. The internet is therefore the major encyclopaedia for every person, in all age categories. The internet has offers to be more helpful in maintaining contacts with relatives who live abroad permanently. Some of the typical tasks that Internet can perform. As-

- Transfer files as well as software.

Figure 4.4: Benefits of the Internet of

- Browse through information on any topic on web.
- Communicate in real time (chat) with others connected to the Internet.
- Search databases of government, individuals and organizations.
- Read news available from leading news groups.
- Send or receive animation and picture files from distant places.
- Set up a site with information about your company's products and services.

Benefits of the Internet-

- Exchange messages using e-mail (Electronic mail).
- 24 hours a day and 7 days a week internet is available.
- Transfer files as well as software.
- Browse through information on any topic on web.
- Communicate in real time (chat) with others connected to the Internet.
- Search databases of government, individuals and organizations.
- Read news available from leading news groups.
- Send or receive animation and picture files from distant places.
- Set up a site with information about your company's products and services.

Boundaries of Internet-

- **Theft of personal information**

If you use internet, your personal information such as your name, address, credit card, bank details and other information can be accessed by unauthorized person.

- **Negative effects on family communication**

It is generally observed that due to more time spent on Internet, there is a decrease in communication and feeling of togetherness among the family members.

- **Internet addiction**

There is some controversy over whether it is possible to actually be addicted to the Internet or not. Some researchers claim that it is simply people trying to escape their problems in an online world.

- **Children using the Internet**

It has become a big concern. Most parents do not realize the dangers involved when their children log onto the Internet. When children talk to others online, they do not realize they could actually be talking to a harmful person. Moreover, pornography is also a very serious issue concerning the Internet, especially when it comes to young children. There are many pornographic sites on the Internet that can be easily found and can be a detriment to letting children uses the Internet.

- **Virus threat**

Today, not only are humans getting viruses, but computers are also. Computers are mainly getting these viruses from the Internet. Virus is a program which disrupts the normal functioning of your computer systems. Computers attached to internet are more prone to virus attacks and they can end up into crashing your whole hard disk.

2.4 INTERNET PROTOCOL (IP) ADDRESS

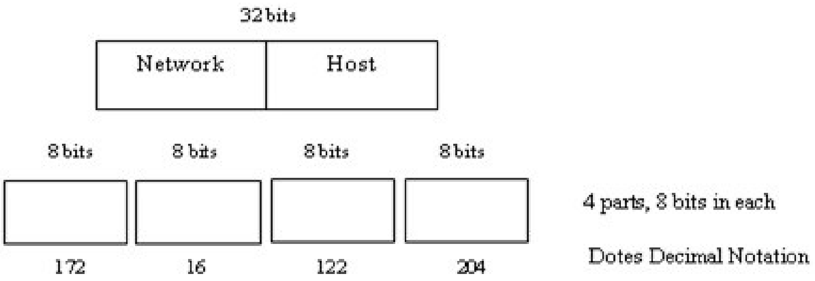
A protocol means set of rules and guidelines for some particular task. In computing (Computer science), a protocol or communication protocol is a set of rules in which computers communicate with each other. The protocol says what part of the conversation comes at which time. Some examples of internet protocols are-

- HTTP (Hypertext Transfer Protocol)
- HTTPS (Hypertext Transfer Protocol Secured)
- TCP/IP (Transmission Control Protocol/Internet Protocol)
- FTP (File Transfer Protocol)
- IMAP (Internet Message Access Protocol)
- POP (Post Office Protocol)
- SMTP (Simple Mail Transfer Protocol)
- TELNET (Telecommunications Network)
- UDP (User Datagram Protocol)
- NNTP (Network News Transfer Protocol)
- MAC (Media Access Control protocol)
- DNS (Domain Name System protocol)
- DHCP (Dynamic Host Configuration Protocol)

In widely installed phase of the Internet Protocol (IP) today, an IP address is a 32-bit number that recognize each sender or receiver of information that is sent in packets across the Internet. When you request an HTML page or send e-mail, the Internet Protocol part of TCP/IP consist your IP address in the message (actually, in each of the packets if more than one is required) and sends it to the IP address that is obtained by looking up the domain name in the url you can requested or in the e-mail address you're sending a message. At the other end, host can see the IP address of the Web page requestor or the e-mail sending host and can reply by sending another message using the IP address it fetched. 32 bits IP address categories into parts network id and host id.

The 32-bit IP address is clubbed eight bits at a time, separated by dots and represented in decimal format. This is known as dotted decimal technique as shown in figure below. The 32-bit number is separated in two parts as Network part and Host part.

The Network part of the Internet Protocol Address-



IP Address Format

The Internet is particularly the interconnection of many individual networks (it's sometimes referred to as an internetwork). So, the Internet Protocol (IP) is primarily the collection of protocols for one network communicating with any other. Each network must know its own address on the Internet and that of any other networks with which it communicates. To be component of the Internet, a group needs an Internet network number. This expulsion network number is in any packet combined out of the network onto the Internet.

The Local or Host Part of the IP Address-

In addition to the network address or number, information is need about which particular host in a network is sending or receiving an information. So, the IP address requires both the unique network number and a host number (which is unique within the network). The host number is most of the times called a local address. Element of the local address can be identified a sub network or subnet address, which makes it easier for a network that is categorized into mainly physical subnetworks (for examples, several different local area networks) to handle many devices. This IP addresses divided into four classes.

IP Address Classes and Their Formats-

Since networks vary in size, there are four different address formats or classes to consider when applying to NIC for a network number as

- **Class A** addresses are for large networks with many devices.
- **Class B** addresses are for medium-sized networks.
- **Class C** addresses are for small networks (fewer than 256 devices).
- **Class D** addresses are multicast addresses

The first few bits of each IP address indicate which of the address class formats it is using. The address structures look like this:

Class A

0	Network (7 bits)	Local address (24 bits)

Note-

Ques1: What is the total number of IP addresses present in class A?

Ans1: IP address is 32 bit number in which first bit is fixed so we have only 31 bits. Using 31 bit we can generate 2^{31} numbers. So class A has 2^{31} IP addresses.

Ques2: How many networks present in class A?

Ans2: Class A has 128 networks.

Ques3: Range of class A?

Ans3: Range of class A (1-126).

Class B

10	Network (14 bits)	Local address (16 bits)

Note-

Ques1: What is the total number of IP address present in class B?

Ans1: IP address is 32 bit number in which first two bit is fixed and then we have only 30 bits. Using 30 bit we can generate 2^{30} numbers. So class B has 2^{30} IP addresses.

Ques2: How many networks present in class B?

Ans2: Class B has 2^{14} networks.

Ques3: Range of class B?

Ans3: Range of class B (128-191).

Class C

110	Network (21 bits)	Local address (8bits)

Note-

Ques1: What is the total number of IP address present in class C?

Ans1: IP address is 32 bit number in which first three bit is fixed and then we have only 29 bits. Using 29 bit we can generate 2^{29} numbers. So class C has 2^{29} IP addresses.

Ques2: How many networks present in class C?

Ans2: Class C has 2^{21} networks.

Ques3: Range of class C?

Ans3: Range of class C (192-223).

Class D

1110	Multicast Address (28 bits)

Note-

Ques1: What is the total number of IP address present in class D?

Ans1: IP address is 32 bit number in which first four bit is fixed and then we have only 28 bits. Using 28 bit we can generate 2^{28} numbers. So, class D has 2^{28} IP addresses.

Ques2: How many networks present in class D?

Ans2: Class D has 2^{28} networks.

Ques3: Range of class D?

Ans3: Range of class D (224-239).

The IP address is usually expressed as four decimal numbers, each representing eight bits, separated by periods. This is sometimes known as the dot address and, more technically, as dotted quad notation. For Class A IP addresses, the numbers would represent "network.local.local.local" for a Class C IP address, they would represent "network.network.network.local". The number version of the IP address can (and usually is) represented by a name or series of names called the domain name.

Static versus Dynamic IP Addresses-

The discussion above assumes that IP addresses are assigned on a static basis. In fact, many IP addresses are assigned dynamically from a pool. Many corporate networks and online services economize on the number of IP addresses they use by sharing a pool of IP addresses among a large number of users. If you're an America Online user, for example, your IP address will vary from one logon session to the next because AOL is assigning it to you from a pool that is much smaller than AOL's base of subscribers.

2.5 TYPES OF INTERNET CONNECTION

Any terminal to activate an internet service on it require an internet connection from an Internet Service Provider. To have a connection one needs to contact the Internet Service Providers. There are several Internet Service Providers (ISP) in each locality. There are some nominal charges that you need to pay to the ISP for the installation of the connection and for the rent (either monthly or annually). Depending on the requirement you choose any one of the available options. If you need a high speed dedicated network then you can opt for a high bandwidth broadband or leased line connection.

If you need to have connection on a single PC then you choose a low bandwidth or dialup connection. For small services at your hand, you can even have mobile phones connected to your Internet. There are many ways to get connected to the Internet. You can get internet connection in any of the following ways-

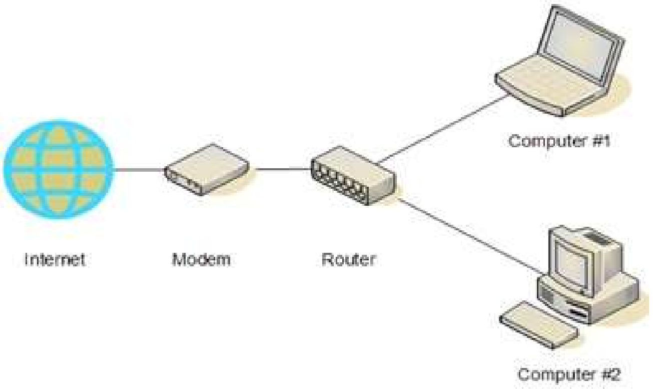


Figure 4.6: Internet and its peripherals

Through dial-up connection-

User is supposed to get connected to the internet after dialling up the number used for connection. This is useful if your network is either confined to small group of computers or for a single PC.

Through leased lines-

In this connection a dedicated line is laid specifically for connection.

Though broadband-

In this connection you can get a broadband connection which provides a high bandwidth for the internet connection. This also provides a good data speed.

These days even wireless connections are available. For this you need to have a Wi-Fi card attached to your computer which can be useful if you do not have proper place to lay down the wires.

To connect to the Internet you need a PC (personal computer) with requisite software including a browser, a telephone connection or a leased line, and a modem, which allows the PC to communicate with other computers. There exist several ways to connect to the internet. Following are these connection types available:

- Dial-up Connection
- ISDN
- DSL
- Cable TV Internet connections
- Satellite Internet connections
- Wireless Internet Connections

Dial-up Connection-

Dial-up connection uses telephone line to connect PC to the internet. It requires a modem to setup dial-up connection. This modem works as an interface between PC and the telephone line. There is also a communication program that instructs the modem to make a call to specific number provided by an ISP.

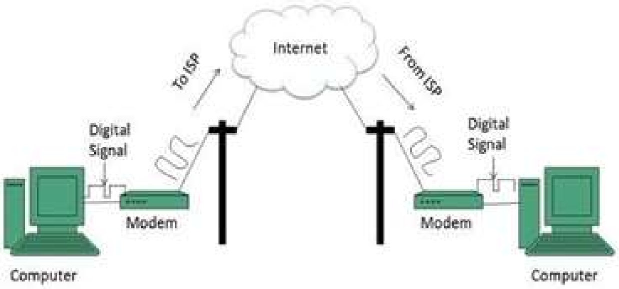


Figure 4.7: Internet Access Dial-up Connection Modem

Dial-up connection uses either of the following protocols: Serial Line Internet Protocol (SLIP), Point to Point Protocol (PPP)

ISDN-

ISDN is acronym of Integrated Services Digital Network**.** It establishes the connection using the phone lines which carry digital signals instead of analog signals. There are two techniques to deliver ISDN services: Basic Rate Interface (BRI) and Primary Rate Interface (PRI).

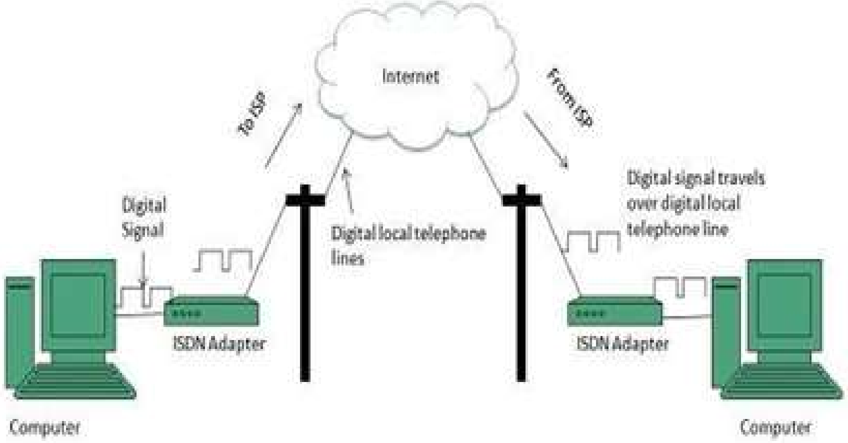

Internet
To ISP
From ISP
Digital Signal
Digital local telephone lines
Digital signal travels over digital local telephone line
ISDN Adapter
ISDN Adapter
Computer
Computer

Figure 4.8: Internet Access ISDN Modem

DSL (Digital Subscriber Line)-

DSL is acronym of Digital Subscriber Line**.** It is a form of broadband connection as it provides connection over ordinary telephone lines. Following are the several versions of DSL technique available today:

- Asymmetric DSL (ADSL)

- Symmetric DSL (SDSL)
- High bit-rate DSL (HDSL)
- Rate adaptive DSL (RDSL)
- Very high bit-rate DSL (VDSL)
- ISDN DSL (IDSL)

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

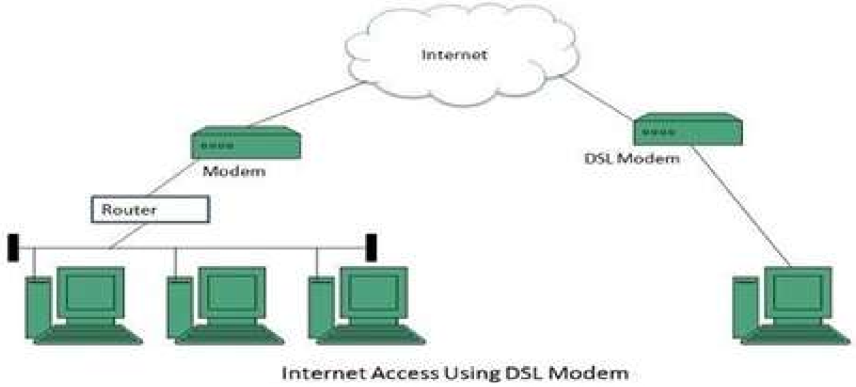


Figure 4.9: Cable TV Internet Connection

Cable TVThe Internet connection is provided through Cable TV lines. It uses coaxial cable which is capable of transferring data at much higher speed than common telephone line. The following diagram shows that how internet is accessed using Cable TV connection:

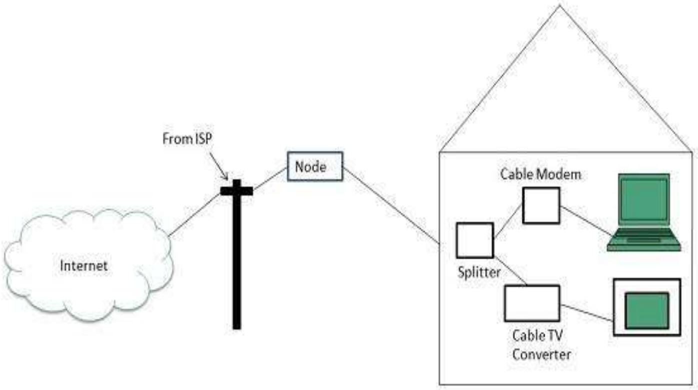


Figure 4.10: Cable TV Internet connections

Satellite Internet Connection-

Satellite Internet connection offers high speed connection to the internet. There are two types of satellite internet connection: one way connection or two-way connection. In one way connection, we can only download data but if we want to upload, we need a dialup access through ISP over telephone line. In two-way connection, we can download and upload the data by the satellite. It does not require any dialup connection. The following diagram shows how internet is accessed using satellite internet connection:

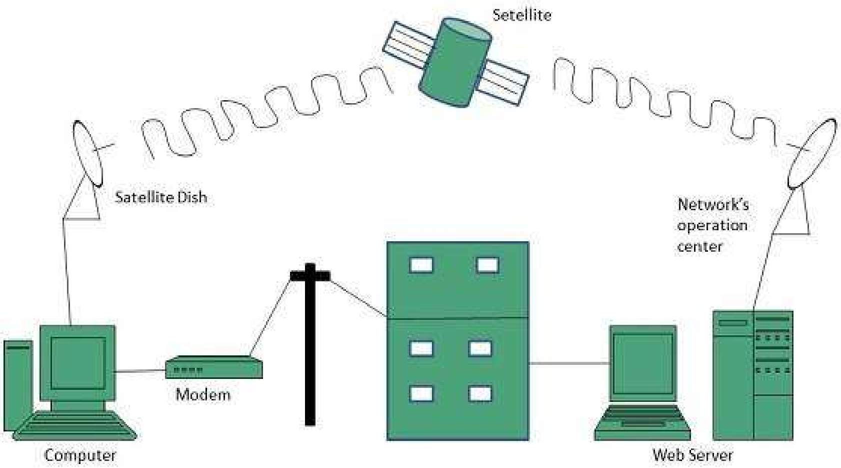


Figure 4.11: Satellite Internet Connection

Wireless Internet Connection-

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

2.6 WORLD WIDE WEB

WWW, this is abbreviation of World Wide Web and also known as web. It came into existence 1991. The father of www is Tim Berners Lee in the European laboratory. In June 1993, the web boasted just 130 sites and in the next year, this was 3000 sites and at the last of April 1998, this became 2.2 million.

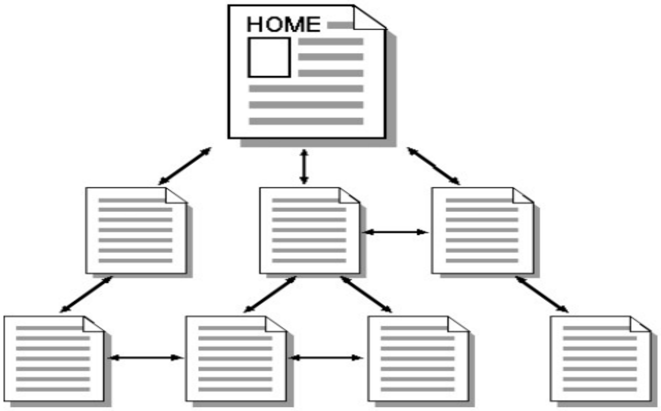


Figure 4.12

Just as each household in the world has unique address, same as each web page in the world has a unique Internet address, sometimes called a URL. For example, the internet address of windows home page is http://www.microsoft.com//windows. Some important terms related to Internet are

- **Download-** To copy data from remote computer to a local computer.
- **Upload-** To send data from a local computer to a remote computer.
- **Filter-** Software that allows targeted sites to blocked from view.
- **Home Page-** This is beginning page of any site.
- **Email (Electronic Mail)-**
- **HTML (Hyper Text Markup Language)-** This is used to create documents for use on the World Wide Web.
- **Search Engine-** A web server that collects data from other web servers, it provides links to pages that contain the object of your search.
- **TCP/IP-** This is also known transmission Control Protocol/Internet Protocol. TCP/IP basically used for communication language or protocol for Internet. When you make up connection to Internet, your computer provides a copy of TCP/IP program to all other computer who wants to communicate.
- **URL (Uniform Resource Locator)-** Every server on the Internet has an IP number that IP number divided into four parts separated by dots. IP number is the server address.

  165.113.246.2

  128.243.233.5

- However, it is difficult to remember than to alphabet address. So address make in form of alphabet called URL. A URL indicates where the web page is stored on the Internet. A URL look like
- https://www.computerhope.com/jargon/t/tcpip.htm
- http://www.tutorialspoint.com/design_and_analysis_of_algorithms/index.htm
- telnet://well.sf.ca.us

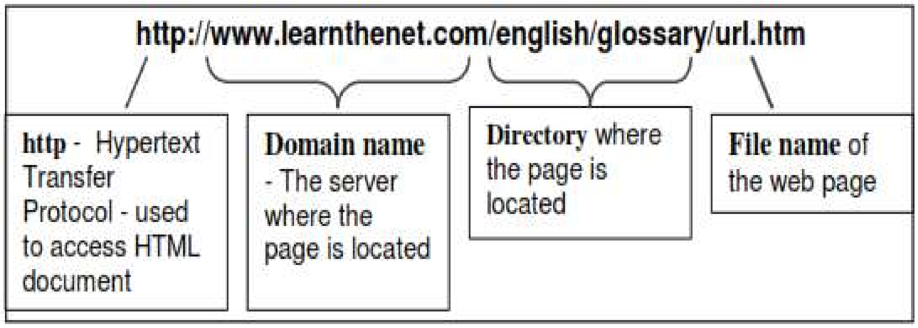

http://www.learnthenet.com/english/glossary/url.htm
http - Hypertext Transfer Protocol - used to access HTML document
Domain name - The server where the page is located
Directory where the page is located
File name of the web page

Figure 4.13: A Sample of URL

Domain-

Domains divide a World Wide Web site into category based on the nature (geographical or location) of their owner and they form a part of sites address. Some important domains are

Table 4.1: Some important domain names

Domains	Purpose / country
ac	academic institutions

com	Commercial entities
edu	Educational institutions
gov	Government institutions
net	Network resources
org	Non-profit organizations
biz	Commercial and personal
mil	Military
au	Australia
jp	Japan
in	India
uk	United Kingdom
us	United States

For instance, www.nios.ac.in is the domain name of a host computer named nios in the academic area (.ac) belongs to geographical domain India (.in).

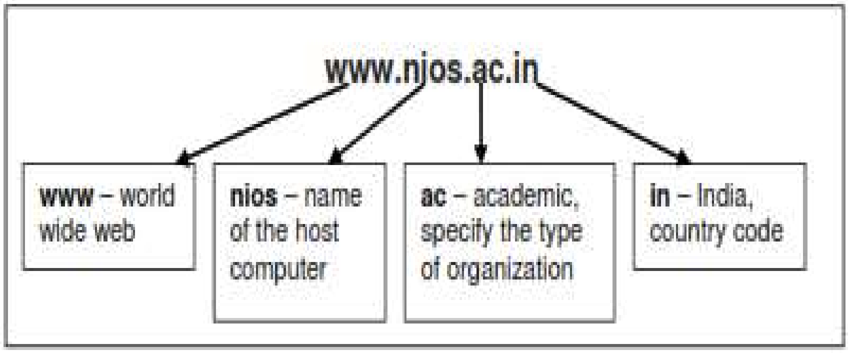

www.nios.ac.in
www – world wide web
nios – name of the host computer
ac – academic, specify the type of organization
in – India, country code

Figure 4.14

2.7 E-MAIL

E-mail or Electronic mail is a paperless method of sending messages, notes or letters from one person to another or even many people at the same time via Internet. E-mail is very fast compared to the normal post. E-mail messages usually take only few seconds to arrive at their destination. One can send messages anytime of the day or night, and, it will get delivered immediately. You need not to wait for the post office to open and you don't have to get worried about holidays. It works 24 hours a day and seven days a week. What's more, the copy of the message you have sent will be available whenever you want to look at it even in the middle of the night. You have the privilege of sending something extra such as a file, graphics, images etc. along with your e-mail. The biggest advantage of using e- mail is that it is cheap, especially when sending messages to other states or countries and at the same time it can be delivered to a number of people around the world.

It allows you to compose note, get the address of the recipient and send it. Once the mail is received and read, it can be forwarded or replied. One can even store it for later use, or delete. In e-mail even the sender can request for delivery receipt and read receipt from the recipient.

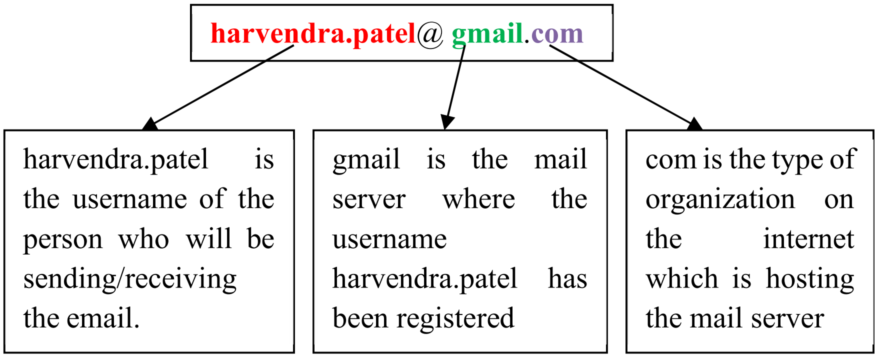
Features of E-mail

- One-to-one or one-to-many communications
- Instant communications
- Physical presence of recipient is not required
- Most inexpensive mail services, 24-hours a day and seven days a week

Components of an E-mail Address

As in the case of normal mail system, e-mail is also based upon the concept of a recipient address. The email address provides all of the information required to get a message to the recipient from anywhere in the world. Consider the e-mail ID.

harvendra.patel@gmail.com



harvendra.patel@ gmail.com

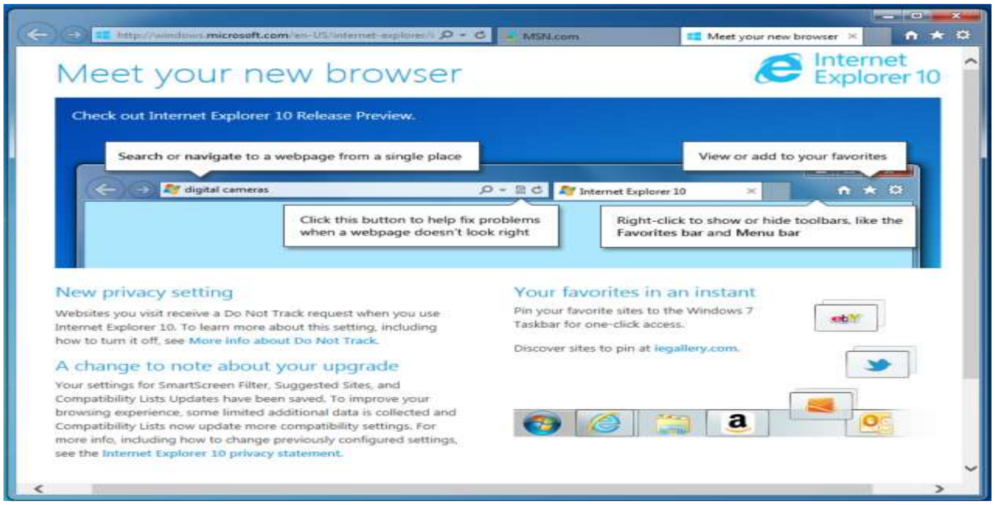
harvendra.patel is the username of the person who will be sending/receiving the email.

gmail is the mail server where the username harvendra.patel has been registered

com is the type of organization on the internet which is hosting the mail server

In the above example “harvendr.patel” is the username of the person who will be sending/receiving the email, “gmail” is the mail server where the username “harvendr.patel” has been registered and com is the type of organization on the internet which is hosting the mail server.

2.8 WEB BROWSER



Meet your new browser

Check out Internet Explorer 10 Release Preview.

Search or navigate to a webpage from a single place

View or add to your favorites

Click this button to help fix problems when a webpage doesn't look right

Right-click to show or hide toolbars, like the Favorites bar and Menu bar

New privacy setting

Websites you visit receive a Do Not Track request when you use Internet Explorer 10. To learn more about this setting, including how to turn it off, see More info about Do Not Track.

A change to note about your upgrade

Your settings for SmartScreen Filter, Suggested Sites, and Compatibility Lists Updates have been saved. To improve your browsing experience, some limited additional data is collected and Compatibility Lists now update more compatibility settings. For more info, including how to change previously configured settings, see the Internet Explorer 10 privacy statement.

Your favorites in an instant

Pin your favorite sites to the Windows 7 Taskbar for one-click access.

Discover sites to pin at iegallery.com.

Figure 4.15: Microsoft Internet Explorer

Contain the basic software you need in order to find, retrieve, view and send information over the Internet. Most popular browsers were Microsoft Internet Explorer, Netscape Navigator. To open browser Microsoft Internet Explorer: click on start→All Programs →Internet Explorer

A web browser defines as "a computer program used for accessing sites or information on a network (such as the World Wide Web)." This is a simple, yet accurate description. A web browser "talks" to a server and asks it for the pages you want to see.

The steps for connecting to a website are shown in Fig. 10.7 and explained further.

- Types a URL for a website in browser say "www.nios.ac.in".
- Your browser attempts to make a connection and sends the request to Web Server.
- The Web Server receives and processes the request.
- The Web Server responds to the request with the home page of the website.
- The webpage is displayed by your browser and the connection between the server and your browser is closed.

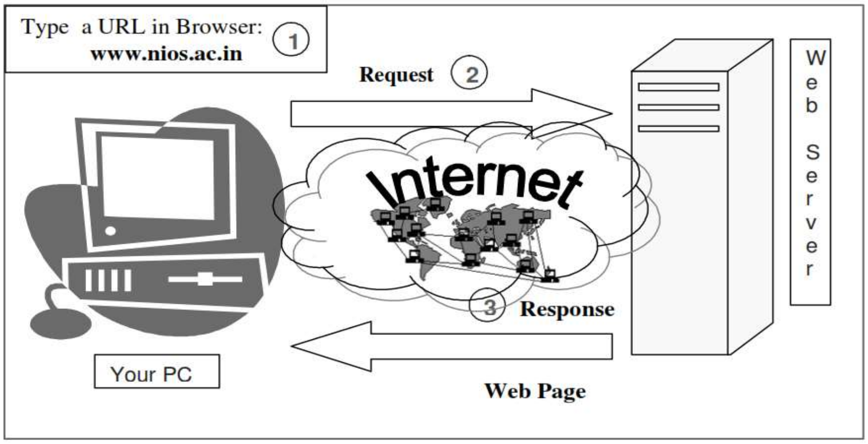


Figure 4.16: Interaction between a Web browser and a Web Server

How a Browser Retrieves a Web Page: - The browser application retrieves (or fetches) code, usually written in HTML (Hypertext Mark-up Language) and other computer languages, from a web server. Then, it interprets this code and displays it as a web page for you to view. In most cases, user interaction is needed to tell the browser what website or specific web page you want to see. Using the browser's address bar is one way to do this. The web address or URL (Uniform Resource Locator), that you type into the address bar tells the browser where to obtain a page or pages from. For example, let's say that you typed the following URL into the address bar: http://www.google.com. That's the home page of google.com. The browser looks at this particular URL in two main sections. The first is the protocol—the "http://" part. HTTP, which stands for Hypertext Transfer Protocol, is the standard protocol used to request and transmit files on the Internet, mostly web pages and their respective components. Because the browser now knows that the protocol is HTTP, it knows how to interpret everything located to the right of the forward slashes.

The browser looks at "www.google.com"—then name—which tells the browser the location of the web server it needs to retrieve the page from. Many browsers no longer require the protocol to be specified when accessing a web page. This means that typing "www. google.com" or even just “google.com" is usually sufficient.

You will often see additional parameters at the end, which helps to further pinpoint the location—typically, particular pages within a website.

Once the browser reaches this web server, it retrieves, interprets, and renders the page in the main window for you to view. The process happens behind the scenes, typically in a matter of seconds.

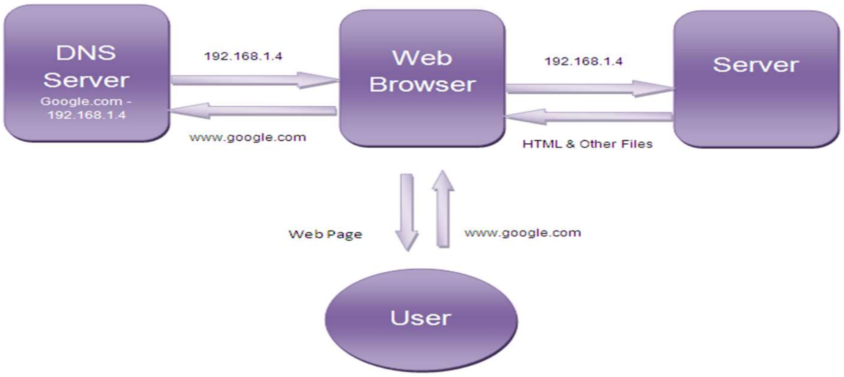


Figure 4.17

Popular Web Browsers: - Web browsers come in many different flavours, each with their own touches. All the best-known ones are free, and each has its own particular set of options governing privacy, security, interface, shortcuts, and other variables. The main reason a person uses any browser is the same, however: to view web pages on the Internet, similar to the way you are viewing this article right now. You probably have heard of the most popular web browsers:

- Google Chrome
- Mozilla Firefox
- Microsoft Edge
- Opera

Many others exist, however. In addition to the big players, try these out to see if any fits your browsing style:

- Maxthon
- Vivaldi
- Brave

Microsoft's Internet Explorer, once the go-to in browsers, has been discontinued, but the developers still maintain the most recent version.

Basic components of a Web-

The primary components of the Web are shown in the following illustration. As:

- **Web servers-**

  Web servers are computers that grasp data for supply over the Internet. In the example application in the diagram, one Web server might grasp the text and graphics of the online magazine what's on in Bath, and another server might hold information on which seats are available for a particular concert. The magazine would be modified using the Web's own publishing language, HTML (Hyper Text Mark-up Language). The data on given seats and their value would be held in a database with connects to particular forms that are published using HTML.

- **Servers**

  Servers can be computers, Macintosh systems or UNIX system; it is the server application that makes them special, rather than the computer itself. That said, servers require being fairly up-market machines. Servers do require to be left on all the time, so that people can access the data on them whenever they want. Another prime point about servers: they are relatively hard to set up.

- **Web clients**

  Web clients can be PCs, Macintoshes and other computers that are connected to the Internet and which can retrieve information from Web servers. A Web client is the computer on your desk. PCs, Macintoshes, UNIX workstations and even simple terminals can run client software. Different client software is marketed (or is given away free) for different platforms. Thus, Mosaic has both a Macintosh and a PC implementation.

- **HTTP protocol**

  Http protocol is used to transmit files between servers and clients. When you click on a hypertext link or fill out a form in a Web document, the results need to be sent across the Internet as quickly as possible, and then to be understood by a server at the other end. Instructions such as `send me this file' or `get me that image' are carried by the Web communications protocol, HTTP. This protocol is the `messenger' that fetches files to and from servers, and then delivers results to your computer every time you click with a request. HTTP has its counterparts in other Internet services: FTP, file transfer protocol, and Gopher are protocols that obtain different sorts of information from across the Internet.

- **Browser**

  Browsers are software, which is needed by a Web client for displaying text, images, video clips and so on. This is provided under the umbrella name `browser', of which Mosaic, Microsoft Corp.'s Internet Explorer and Netscape Communications Corp.'s Navigator and

Communicator browsers are probably the best-known examples. Browser application offers you the skill to scan data fetch from Web servers, as you would browse through a book. It also provides you services for saving and printing information obtained on the Web

2.9 POINTS TO REMEMBER

- Internet is Easy tool to communicate globally.
- Internet is a ocean of information. It is also used for informed decisions.
- Internet is useful for business promotion and innovation.
- Internet is used for research and development

2.10 GLOSSARY

- Internet- the Internet is a network of computer networks. The users can share resources and communicate with each other all over the world. The Internet is also called a meta network.
- The Client/Server Model- The most popular Internet tools operate as client/server systems. You're running a program called a Web client. This piece of software displays documents for you and carries out your requests. The Web server is a computer running another type of Web software which provides data, or "serves up" an information resource to your Web client.
- Electronic mail- E-mail is a fast and efficient way to communicate with friends or colleagues. You can communicate with one person at a time or thousands; you can receive and send files and other information.
- Network news (Usenet News)- is a service comprised of several thousand electronic discussions providing users an effective way to share information with others on just about any topic.
- Freeware- When you download freeware software, the author continues to carry the copyright to the software, but permits you to use the program for free. You can share freeware with others, as long as you don't sell it.
- Public Domain- When you download public domain software, you can use it freely. The creator carries no copyright, and has released it for anyone to use. There are no limits on distribution or sale and anyone can modify the program.
- Shareware- When you download shareware, the author continues to carry the copyright to the software, but you're permitted short-term use of the program for evaluation purposes. At the end of evaluation period, you must either pay the copyright holder for the program or destroy all copies you've made of it.
- Uniform Resource Locators (URL)- A URL is the address of a document you will find on the world wide web (WWW). Your WWW browser interprets the information in the

URL in order to connect to the proper Internet server and to retrieve your desired document.

2.11 CHECK YOUR PROGRESS

Part- A (Fill in the blanks and True/False type questions)

(a) The World Wide Web is a collection of electronic documents called web pages.

(b) Web server is known as web page that connects to another place in the same document or to an entirely different document.

(c) Electronic Mail is a paperless method of sending messages.

(d) File Transfer Procedure is a commonly used protocol for exchanging files over any network.

(e) ..... is a network protocol used on the Internet.

(f) ......refers to the mail server.

(g) A search engine is a program designed to ......stored on a computer.

(h) .... is the standard for resolving names to addresses.

Part- B (Descriptive type questions)

(1) Explain the different types of connections available to get connect to Internet.

(2) Define the following terminologies used in Internet; WWW, Hyperlink, Web page, Website and URL.

(3) Yahoo and Hotmail offer free-email registration. Go to any of these sites and open an e-mail account in your name. Send an e-mail to a friend whose address you know.

(4) What is search engine? Mention two important search engines available on the Internet? What are essential orders to use to get the finest results while searching?

Answers- (Part A- Fill in the blanks and True/False type questions)		
(a) True	(b) False	(c) True
(d) False	(e) Telnet	(f) Domain name
(g) Find information	(h) DNS	

2.12 BIBLIOGRAPHY/REFERENCES

- http://unsweb.aucegypt.edu/UNSWEB2/NetIntro.htm, March, 2018
- http://download.nos.org/coa631/ch3.pdf, January, 2018
- http://download.nos.org/srsec336new/Lesson%2010.pdf, January, 2018
- https://eclass.teicrete.gr/modules/document/file.php/YN103/AC-Windows-Internet/Internet%20Introduction.pdf, January, 2018
- Sluyters, Richard C. Van, Introduction to the Internet and World Wide Web, ILAR Journal, May 2018

- https://www.sandiegocounty.gov/library/PDF/gateway_introduction-to-internet.pdf, March, 2018

2.13 SUGGESTED READINGS

- James, Scott D., Introduction to the Internet, 1998

UNIT 3
INTRODUCTION TO INTERNET- II

3.1 INTRODUCTION

Inter-networking is a platform where we may meet anyone in the world. We may do any kind of project processing over the internet viz. an online book store, an online shopping of products, online universities, sending and receiving payments along with authentication etc. It involves a lot of technologies to come together to process all these tasks. Here we will describe some of the related terminologies which may use frequently while using Internet.

3.2 OBJECTIVES

After successful completion of this unit, you will be able to-

- Hardware software requirement of Internet.
- Understand domain Name system.
- Define URL and search engine.
- Define Internet and E-governance.

3.3 WHAT IS INTERNET?

Internet is a global system of interconnected computer networks. It is also called a network of networks that consists of millions of networks. Internet uses telephone lines, cables, satellites and wireless connections to connect computers and other connecting devices to the World Wide Web. Internet can be used to access a huge 'library' of information from the millions of websites, send and receive email messages, buy/sale goods and services, online banking, play games with other people online and etc.

3.4 TERMINOLOGIES USED IN INTERNET

Domain Name-

Every computer has its unique address. Every computer user using the internet has unique addresses through which we can easily identify them, more specifically, with the help of their IP addresses we may connect to their location very easily. In practice, it is used to recognize IP addresses. The key role of domain names comes in existence when we need to reach to some particular web pages. In the web URL “www.railway.com” the domain name is railway.com. Few standard top-level domains are created to segregate the related contents. Few domains’ names are given below:

com – Commercial business

edu – Educational institution

gov – Government agencies

mil – Military

org – Organizations (non profit)

There are also some additional domains on internet. They include:

aero – Air-transport industry

biz – Businesses

coop – Cooperatives

tv – Television

info – Unrestricted use

Web Browser-

Web browser is software which enables user to access documents, images, videos, audio, and video from the World Wide Web (www). Example of a web browser are- Microsoft Internet Explorer (IE), Mozilla Firefox, Opera Mini, Google Chrome and etc.

E-Mail-

Email is an electronic mail that electronically transmitted through the computer. It has the capability to send message to anyone and at any time. All the emails are unique in itself that is the reason why email addresses already in use can not allowed to new users.

FTP-

FTP stands for file transfer protocol and provides a file transfer over the internet. It is the standard method to upload or download file over internet from one location to another.

Homepage-

Homepage is the first page of the browser or any website that give the details about the web site and browser.

Hyper Text Transfer Protocol (HTTP)-

HTTP is an acronym for hyper text transfer protocol. It is the standard protocol which is applied to view the web pages in web browsers. It is the set of rules by which web pages are transferred across the internet.

Internet Protocol (IP)-

IP stands for Internet Protocol. Each computer on internet can be identified by the set of dot separated four numbers and is called as Internet Protocol. One of the example of IP address is which is composed of four different numbers separated by periods is 205.134.120.60.

Link or Hypertext Link-

Link or hyperlink is an underline words, phrases or graphics on a web page that transfer control from one page to another page of information. That information is linked together with the help of hyperlinks and collectively gives rise to World Wide Web.

3.5 HARDWARE AND SOFTWARE REQUIREMENT TO ACCESS INTERNET

If you want your computer to connect to the huge network of other computers called Internet, it involves the adequate set of equipments and proper procedures to get connected. The steps involved are not difficult to implement. General awareness and knowledge of the hardware and provider is necessary to operate the hardware devices.

As per as hardware requirements concerns, first you need a computer system to connect to the Internet. Further requirements depend what kind of connectivity you are looking for. Such as if you want to use broadband connection for internet access then you need a DSL cable to connect to Digital Subscriber Line. Internet service provider (ISP) usually makes available this service to the internet users. For using a dial-up connectivity dial-up modem is required. Now-a-days these modems are already included in your computer system which are also known as in-built modem, otherwise you may purchase them easily too.

On the other hand, if you want to use Wi-Fi internet connectivity on your computer, then your need to purchase a Wi-Fi router and install it in your machine. Then connect its port to your modem. You may use “Ethernet Cable” for this purpose too. As per as the software requirements concern, these days all software are pre-loaded in the device you purchase. Also, you may download the required files from their websites by entering the exact model number of your device. Once you are connected with the internet, this smart software will not only get updated but also will suggest you some other relevant software that matches your preferences. In a nut shell, these days all the hardware and software requirements are fulfilled and pre-installed by the computer manufacturers. Also, they provide a complete set of drivers either in DVD or both in DVD and their websites.

Minimum Hardware and software requirement to access Internet-

- A working computer system with all necessary peripherals.
- A modem and telephone line (if you are using dial up access) or any other Internet Service Provider (ISP).
- An Internet browser (such as- Internet Explorer, Google Chrome, Safari, Mozilla Firefox, Microsoft Edge, Opera, etc) and software to connect you to the ISP.

Types of Internet Connection-

Dial-Up Connection-

A modem connects to the Internet after the computer dials a phone number. This is a slower analog signal is converted to digital via the modem and sent over a land-line serviced by a public telephone network.

Digital Subscriber Line (DSL) Connection-

It is an internet connection that is always “on”. This uses 2 lines so your phone is not tied up when your computer is connected. There is also no need to dial a phone number to connect to the internet. DSL uses a device named router to transport data over the network.

Cable Connection- Cable provides an internet connection through a cable modem and operates over cable TV lines.

Wireless Connection- Wireless, or Wi-Fi, Instead of cable, it uses radio frequency. Wireless is also an always on connection and it can be accessed from just about anywhere.

Satellite Connection- Satellite accesses the internet via a satellite in Earth's orbit. The enormous distance that a signal travels from earth to satellite and back again, provides a delayed connection compared to cable and DSL.

Cellular Connection- Cellular technology provides wireless Internet access through cell phones. The speeds vary depending on the Internet Service Provider (ISP), but the most common are 3G and 4G speeds. A 3G/4G is a term that describes a 3rd/4th generation cellular network.

Very Small Aperture Terminal (VSAT) Connection- An earthbound station used in satellite communications of data, excluding broadcast television. A VSAT consists of two parts, a transceiver and a device that is placed indoors to interface the transceiver with the end user's communications device, such as a PC. The transceiver receives or sends a signal to a satellite transponder in the sky. The satellite sends and receives signals from a ground station computer that acts as a hub for the system.

3.6 UNIFORM RESOURCE LOCATOR (URL)

Uniform Resource Locator or URL in short is a reference to a resource available on the internet. This term is very common in the internet terminology. URL is used to reach to a specific HTML page or some similar resources from the web. Further categories of the URL are absolute and relative URL; both of them are well used while developing the web pages.

In other words, URL is a accessible name given to some web resource based on where it resides. Most of the URLs refer to a file stored either on the same computer system of on some different machine. This external URL may access from any kind of network. However, remember that URLs also can point to other resources on the network, such as query from a database or accessing some other resource such as audio, video or some other document.

Components of URL are necessary to understand. A standard URL has two components which are protocol identifiers and name of the resource to where URL implies to. Protocol identifier is the name of the protocol which is generally http and is popularly used in our browsers.

Consider the following URL:

"http://mydomainname.com"

In the above domain name http is a protocol identifier while "mydomainname" is a name of the resource. Noteworthy here in the given URL above is, http and the mydomainname.com both are separated by a colon sign (:) followed by two forward slashes (//). Here, protocol identifier (http) indicates the name of the protocol to be used to get one instance of the resource.

Http usually serves the http documents on the web. There are also some other types of resources on the internet used as protocol identifier such as ftp, gopher etc. On the other hand, a resource name is the complete address of the resource which is to be accessed using http. The format of the resource name depends entirely on which protocol is being used. Further, for most of the protocols the resource name contains one or more of the following components:

- Host name
- File name
- Port name
- Reference

Where host name indicates name of the computer system in which the web content resource is residing. Further, file name is the pathname to the file residing on some machine, port name is the port number to build connection and finally reference which is the optional component is referred to a named anchor which usually identifies a specific location within a file.

Generally, for most of the protocols hostname and the file name are needed, whether on the other hand the port number and reference are not forced. For instance the resource name for an HTTP url must specify a server on the network (Host Name) and the path to the document on that machine (Filename). This can also specify a port number and a reference too but are not the forced routines, as said above.

Characters not allowed in a URL-

There could be a simple question that if spaces are allowed in between the URL characters. It is better to know that in RFC 1738 it is documented that the URL string can only contain alphanumeric characters and the !$-_ +*’(), characters. In other words, if some characters other than the given are needed in the URL then they must be encoded. For example, encoding refers to change the characters into equivalent programming code.

3.7 HOSTING OR WEB HOSTING

Web hosting or simply hosting is a facility for allowing organizations and/or other individuals for website posting or webpage posting onto the Internet. A web hosting is a kind of popular business that facilitates the technologies and relevant services that are needed for the website governance which are to be viewed in the internet. A special purpose computer system is called as server where webpages which are coupled and referenced together are hosted and sorted accordingly if required.

In a case when and intended internet users whishes to surf your website, all what they require is to type the url of your website to any web browser which they are using. As a result, their browser will connect to your server based on URL and your web pages will be delivered to their machine too. A number of hosting companies elaborate their services as virtual hosting. Virtual hosting simply means that the services provided will be transparent and all the web sites will have their own domain name and corresponding set of e-mail addresses. In most usages, hosting and virtual hosting are same. Some hosting companies let you have your own virtual server, the appearance that is dedicated entirely to your site. Dedicated hosting is the provision of a dedicated server machine that is dedicated to the traffic to your Website. Only very busy sites require dedicated hosting. Many companies purchase their own servers and place them on a site that provides fast access to the Internet. This practice is called collocation.

Options for hosting-

You must be very clear about the related terms and their application before you start hosting your website. Following services are provided as options for hosting a website

Website Builders-

This service is a type of hosting services that builds the technical skills and knowledge. In this service an online browser-based interface is provided and then the website is built upon that. The same setup hosts your website too without requiring any additional setup.

Shared Hosting-

It is another type of hosting service. In this environment, you and some other website hosting interested owner shares one server. It includes sharing of the physical server and the software application within the server. This server is affordable too as the cost being charged for the server setup will be shared between you and these other peer owners. Also, it must be considered that there are also a number of disadvantages of this shared hosting such as being slower due to sharing facility.

Dedicated hosting-

The term dedicated is given as the entire hosting environment the is for yourself. The advantage of being dedicated hosting results in faster performance. It is because of having all the control on server resources without sharing with website owners. This also means that you will be responsible for the cost of server entirely.

Collocated Hosting-

In the last, collocated hosting you will purchase your own server. Further you need to get it housed at facilities given by the web host. In this hosting, you will be responsible for the server itself. An advantage of this type of hosting service is that you have full control of the web server. Further, you as an administrator may install any required script or applications for satisfying your customized needs.

3.8 DOMAIN NAME SYSTEM

DNS is an acronym of domain name system. It is simply a mechanic to convert an alphabetic name to a number-based IP address. In a normal practice of using any website what we do is that we simply key-in the address of that website in the address bar of the web browser. But that address is associated with some unique IP address. Here comes the role of a server which results back the IP address of the requested resource such as web page. Any of such server is the one known as DNS server. The synonym used for DNS server is hierarchical decentralized naming system. It includes computer systems, their associated services and various other resources. This system also has a DNS database which holds various DNS names which are associated with some IP addresses.

Protocol-

A protocol is simply a set of rules and guidelines for making the communication among two or more communicating devices. These rules are defined for every process during communication. The communicating computers or devices have to follow the set of rules defined in the form of protocols for the purpose of communication which results in successful transmission of data.

Internet protocol (IP) is the principal set of digital message formats and rules which helps in exchanging messages between computers across a single network or a series of interconnected networks. It takes the help of internet protocol suite known as TCP/IP. In this process, messages are exchanged in the form of datagrams which are also known as data packets or packets simply. The role of IP is the delivery of data from the source host or source computer to the destination computer which is also known as receiving host.

3.9 OPEN SYSTEM INTERCONNECTION (OSI) MODEL

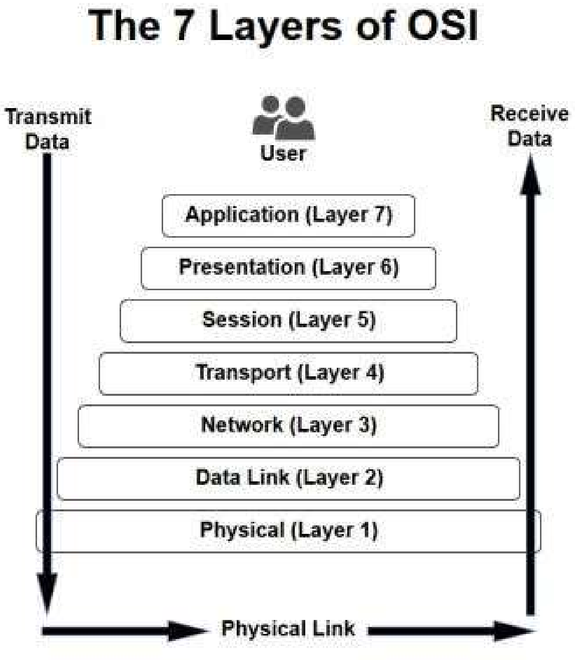
It is a standard model that defines networking framework. It implements various protocols which are simply a set of rules in layers where control goes through one layer to the next. In OSI, it is the responsibility of the lower layer to provide data or services to its adjacent upper layer. Before providing data to its upper layer the provider layer must go for a valid check, so that any error may be trapped easily before providing data to its upper layer. The OSI model defines networking which is a stack of seven layers in vertical order. The upper layers of the OSI model represent software that implements network services like encryption, decryption, connection management etc. The lower layer of the OSI stack model implements various hardware-oriented functions such as routing addresses and flow control of datagrams or ip-packets. The OSI model segregates the architecture of the network into connected seven layers in a progression which is vividly logical.

In the model, lower layer deals with electrical signals and various chunks of data in the form of 0 and 1 i.e. binary data and the routing of these data across the network. In this flow, the higher levels deal with requests and responses across the network, representation of data and network protocols as seen from the point of view of the user.

Layer 1- Physical layer

At this layer of the model, the physical layer of the OSI model is responsible for ultimate transmission of data bits. Ethernet cables, token ring networks are the examples of the physical layer.

At the physical layer, data are transmitted using the type of signaling supported by the physical medium. The various medium used are electric voltages, pulse of infrared, radio frequencies, ordinary light etc.



Layer 2- Data link layer

As per data received from physical layer, the data link layer checks for physical transmission errors and packages bits into data frames. Physical addressing is also managed by data link layer schemes. Such example of physical addressing schemes are MAC addressing for Ethernet networks used for controlling access of any various network devices to physical medium. Due to high load of various responsibilities it is the most complex layer of the OSI model. Further this layer is divided into two sub layers known as media access control layer and logical link control layer which further takes the responsibilities separately.

Layer 3- Network layer

The third layer in the model holds the concept of routing above the data link layer. As the data arrives to the network layer, the source address and destination address contained inside each frame are examined to determine whether the data has reached its final destination. Further, this layer formats the data into datagram known as data packets which are delivered in the upward direction to the next layer known as transport layer for providing further services.

It is the duty of the network layer to maintain the logical addresses for devices on the network to support routing across the network. The network layer also manages the mapping between the logical and physical addresses too. In the IP networking, this mapping is accomplished through the Address Resolution Protocol.

Layer 4- Transport layer

It is the duty of the transport layer for delivering data across network connections. The TCP protocol is the most common example of a transport layer. Various data transfer related protocols also supports a range of optional capabilities such as recovery from errors and flow control. It also supports for re-transmission in case of need.

Layer 5- Session layer

This fifth layer has a duty to manage the sequences as well as flow of events that initiate and destroy down network connections in case of need for ending a session. At this fifth layer, support for multiple types of connections that can be created dynamically and run over individual networks.

Layer 6- Presentation layer

The presentation layer is generally a simplest layer of all the seven layers present in the OSI reference model. Presentation layer handles syntax processing of message data such as format conversion, encryption and decryption whenever needed to support the layer above it known as application layer.

Layer 7- Application layer

This last layer in the stack has the responsibility of that this layer provides network services to end-user applications. Network services are typically the protocols that works with users data. For example, the application layer protocol HTTP packages the data needed to send and receive web page content in a web browser application. Finally, this layer provides data to the presentation layer.

Cross-layer functions

Cross-layer functions are the services that are independent but are not tied to any given layer. On the other hand, it do also affect one or more layers. The following examples are the cases of cross-layer functions:

Management Service- Functions that permits to configure, monitor and terminate the communications among two or more networks entities are management systems. There is a specific protocol that is known as common management information protocol, CMIP in short. It serves this layer. Further, this service needs to interact with every layer in order to deal with their instances.

- **Multiprotocol Label Switching**- The OSI model further subdivides the third layer i.e. network layers in three roles. This division is done to provide a unified data-carrying service. It can also be used to carry various different kinds of network traffics.
- **Domain Name Service**- Domain name service or DNS in short is a service available at application layer service. DNS service is used to look up the IP address of a given domain

name. In this level, when DNS server sends reply, it is then possible to form a four-layer connection or flow to the desired host.

The following table presents all the seven layers of OSI reference model along with basic services and protocols associated with each layer.

Comparison between OSI and TCP/IP Model-

In the TCP/IP model, design of protocols in each layer of the internet does not concern with strict layering of the protocols in these layers. TCP/IP follows a five layer scheme where OSI follows a seven layers scheme. The five layers of TCP/IP are here compared equally with its equivalent OSI layering scheme. The comparison of TCP/IP five layers along with the OSI seven layers is below:

- The internet application layer includes the OSI application layer, and most of the session layer.
- Its end-to-end transport layer includes the elegant close function of the OSI session layer as well as the OSI transport layer.
- The link layer includes the OSI data link layer and sometimes not only the physical layers but also some protocols of the OSI network layer.

The comparisons given above are based on the original seven-layer protocol model as defined in ISO 7498.

3.10 WEB SEARCH ENGINES

The term web search engine is defined as a software system that is developed to search the information from the internet especially from the World Wide Web. Search Engine Result Pages also known as SERPs are the search results that are presented in a line of results generally. The information of the search varies and maybe a collection of various web pages, images, video links to some websites and some other extensions files. The main part of the search engine is that it also maintains real-time information by running various algorithms on web crawler. It is a bot that browses the World Wide Web very systematically. It has various purposes in the internet. Web indexing or web spidering is also implemented by the help of the bots.

Search engines history-

Web search engines were started in the year 1990. The first well documented search engine that was used to search files and contents. More specifically, FTP files Archie was debuted on 10 September 1990. Also, the world wide web was entirely indexed by hand before September 1993. That time, the count of web servers grows at a faster rate as more and more web servers existed on the web. Consequently, the central list management was getting complex.

Archie was the first tool that is used for the purpose of searching content on the Internet. It was created by Alan Emtage, Bill Heelan and J. Peter Deutsch who were students at McGill University Montreal in the department of computer science. Further, Gopher was created in the year 1991 by a computer scientist Mark McCahill at the University of Minnesota. He also led the two common search programs which were called as Veronica and Jughead. They do search like Archie used to applied. They searched the file names and titles which are stored in Gopher index systems. Specifically, veronica provided a keyword search for most Gopher menu titles. Jughead was also a tool which was used to obtain the menu information from some specific Gopher servers. Matthew Gray at MIT developed the first web robot, later in the year 1993. Wanderer was a perl language-based robot to create index. The index created by wanderer was known as wandex. The main purpose of wanderer was to actually measure the size of the World Wide Web. The second search engine of the world wide web was known as Aliweb that was developed in the November 1993. Aliweb was not dependent on any robot. It was depended only to get notified by the various website administrators.

Search Engine Technology-

Normally, a search engine consists of four main components namely ‘Search Interface’, ‘Crawler’, ‘Indexer’, and ‘Database’.

Search Engine Interface

Search engine provides us an interface that groups the search items and enables you to select and specify criteria of your interests and have the engine which finds the matching items as result.

Web Crawler

A web crawler, sometimes called a spider, is a bot (as explained earlier) that browses the World Wide Web.

Indexer

An indexer indexes the web. Web indexing are various methods of indexing the contents of a website or of the internet as a whole.

Database

A database is used to manage the data online.

Working of a Search Engine-

All the above-mentioned units combined work as your search engine. Following two features are important to know about search engines:

Crawling

When a web crawler visits a web page, it starts collecting all the links on the page. Further, it adds these links in a list for visiting the further pages to visit. This process is repeated for further traversing. This process here means that any website that is linked from an indexed website will be crawled. Usually, some sites get crawled more frequently and some to greater depths.

Indexing

The data from a crawl is processed and placed in the database. This is known as indexing.

Retrieval and Ranking

The search engine processes your search query and returns the most relevant pages. Most search engines differentiate themselves by their retrieval methods. Ranking algorithms check your search against billions of pages to check their relevance.

Effective Use of Search Engines-

Nowadays where everything and all the knowledge is available on the Internet, online research has become an essential skill. Previously the knowledge was only available at libraries and any more would need a visit to the experts but the Internet has changed that. Experts can be contacted through E-mails or video conferences and search engines have become extremely important tools, hence there is a need to learn the effective use of the search engines is important.

Search skills can be developed through practice or by reading the help pages provided by the search engines themselves. Over time one realizes what search engines are good to search what kind of information.

The Basics of Searching:

Generally high number of keywords used in your search will get you more accurate results. Try to use additional keywords in your search to eliminate any unwanted results. For example, if you want to search something about Virat Kohli the Cricketer, simply searching “Virat Kohli” might get you some results of websites that mention other Virat Kohli’s that might not be the person that you’re looking for while searching for “Virat Kohli Indian Cricket Team” will get you more accurate results and will eliminate any unwanted data. Although this method may remove any unwanted data but it might miss out the results that mention Virat Kohli the cricketer but not the Indian Cricket team, therefore fine tuning your search as much as possible will give you better results.

Concluding remarks on effective searching

There is no perfect search engine or search algorithm that will give you exactly what you need, however certain search engines do give you better results than others. You should take some time getting used to the different search engines and see what they have to offer. Reading the help section works very well especially when we did not get any relevant filtered content from the search engines. Further, we may combine the help contents with the advance search options that are provided by each search engines for better filtering of the results.

Surfing the web

Web surfing means to search for random various topics and search and get into information about. That is surfing randomly on various topics.

Surfing process is considered like going over through different mediums of web contents and leave the details to be filtered by the user after reading the contents and researching further FOR better and specific results. Generally, surfing is a wider term than simply browsing because in case of dubiousness on filtered material, browsing may get shaped into surfing to search for the reliable web content.

The internet media is a big and ill-managed informational juncture for those who are not tech savvy. There is a lot of material that surfer can search to make their quest for knowledge better and easier too. When looking for a specific item in this cyber active library it is easier to have a preplanning before search, and then gradually fetch the particular subject out of the result given by the internet. Most likely, using searching tools will be your first means of finding what it is that you need. When using a search tool for the first time it is suggested to have a general understanding of the tool first. You must know the proper functioning on how it works, and the type of language supported when processing with such tools. For e.g. a general search tool usually has their own and unique criteria, thus making the search for information that much more difficult matters too. Responses to a particular statement query can vary accordingly from search tool to search tool as their search algorithms are different too. Also, the same query sent by the same inspection tool may come up with various responses depending upon time, place etc, because web pages are dynamically added, removed and updated. You will experience that if you consistently use the internet, then the things which you surf most of the time that things will become less difficult for you to search in future as your experience broadens in the hand of search tools. To be more technical, this idea is called machine learning.

Today life style must include surfing the internet for petty things. Such as finding out nearby landmarks, ATMs, Hospitals, booking a taxi etc. and this has become our daily routine practice. Using internet we do several things viz. online shopping for various things, doing social networking, disseminating various information on the internet etc. The aim of the internet is to provide solutions for our problems related to any sector or area. To better segregate the problems various sections and sub-section-wise listing is usually made in each sector. Such an instance, there exists various categories of all the items such as clothes have sub categories as men-wear, women-wear, kids-wear etc. This is being choosen during the browsing the specific websites. I don’t have to go outsides to buy them, which saves me a lot of time.it is especially convenient when the weather is not good. Second, we can make many kinds of friends on the internet. We can know foreign friends who can tell us about their experiences, cultures work, and so on through the internet. On the other hand, we don't have to go long distance to meet them; we can just sit in front of the computer and talk to them in English. Therefore, making friends on the internet can not only open our mind to the world but also improve our language skills. Third, we can get lots of information from the internet. If we didn't have the internet, we would have to spend much time searching information. Nowadays, we only need to key words into Google and we will find a lot of relevant information. It is very convenient and can save us much time.in addition we can get the news of news of current events instantly on the internet. We can also choose what we want to know on the web. We can not only save time in searching news but also save the money for buying newspapers.

Finally, the internet plays a vital role in making human life convenient and variably problem solvable by providing lots of options.

Web surfing vs. web browsing-

Web surfing and web browsing are two different terms of the internetworking. The main difference is that going through or searching for a specific information on a specific website is web surfing while whereas going through information aimlessly on different subjects on the different websites is what is termed as web browsing. Usually, the internet users just keep sitting in front of the computer and go through the different websites on the internet for many hours without any aim. And often, this habit of simply clicking through the pages of the various site is termed as web browsing.

The terms surfing and browsing are generally referred to the task of searching on the internet. Most of the times, both the browsing and the surfing looks as much as the same. But, the actual difference between the two terms depends on the assigned task based on what a user might browse the internet just might just surf it. So, as said before, if an internet user is using a particular browser, going through websites in search for a particular thing, it is termed as web browsing or simply browsing, whereas if the user is randomly searching or going through different websites for information gathering then this process is termed as surfing.

On the other hand, in general there is no difference between the term's web browsing and web surfing as both of these two processes are quite similar. Due to various varieties of the available web browsers, the type of information and their process to extract the information in these web browsers is also quite different technically.

Web searches may also be used for any kind of query such as doing the research on some topics, searching the latest news, searching the old news, looking for the current openings in any organization, paying the utility bill payment services etc.

So, based on above facts, it doesn't take any notable difference in the acting of browsing and surfing. Both the terms are commonly used by people to define the action of redirecting or switching or moving among websites. Even downloading or streaming of the audio, video or both contents from the internet is also possible by means of web browsing and web surfing. As a result, any intended interned user may watch, read, summarize, listen to music, view and download videos by simply wandering among websites by means of web surfing or web browsing.

Comparison between web surfing and web browsing

	Web surfing	**Web browsing**
About	It is defined as going through information without any aim on different subjects on the different websites.	It is defined as going through or searching for specific information on a relevant and specific website.
Consideration	It is considered as random searches made on the internet.	It is considered as specific searches made on the internet.
Target	It is simply an aimless search on the internet.	In this a user can browse instantly on the internet for some specific content.

Surfing process is time consuming and may create confusions. Sometimes it may lead to unwanted and unnecessary bifurcation. While browsing a specific and known website may lead in getting the desired result without generating any unwanted and unnecessary noise in the results.

The Pros and Cons of surfing-

In the today trend of internet, no person, no enterprise, any tradition or anything else is far from internet. Now-a-days, children would rather use internet for entertainment, infotainment and for other purposes too. For this purpose, he or she uses the devices such as smart phones, laptops, desktops, ipads etc. These devices provide efficient and wireless facility for internetworking. Moreover, online live and interactive game are available in abundance too.

The internet also plays a vital role in our daily lives because of the lots of advantages it brings. Internet has made our work a lot easier than earlier. Employees can find information about new recruitments without physically going to other organizations, customers get update on latest products or new arrivals on their product of interest such as clothing, music, loots on the mall etc. Students may get full knowledge on any topic without touching the pen-paper for his exercise. The list is quite long that gives advantages of using internet. In a nut shell, the things are converting from paper-based to less-paper-based. And the target of the internet community will soon touch the height by governing all the processes paper-less i.e. no paper requirement, as everything will be stored on the internet.

Now-a-days, all the information is transparent and the users are able to find out updated information about what is happening in the world. Because internet user spread all over the world are using the internet and making pressure groups is much easier for them. Using intern one can easily be updated on news of every area. Moreover, the user may subscribe himself or herself to get specific news updates on his phone, or emails by subscribing themselves with the newsgroups. Some examples include political news, competitive exams news, Jobs opening for freshers or experienced employees, weather forecasting updates, updates on product promotion offers by the manufacturers or retailers such as online shopping portals etc.

3.11 E-GOVERNANCE

E-governance is a mechanism that puts the paper-based work to be transformed and governed by the convergent media with the utilization of ICT. E-governance can be considered as a mediator between government and citizens in providing and receiving services respectively. It includes the use of modern technologies by government to improve efficiency, service, etc.

Types of E-Governance-

- Government 2 Business (G2B)- G2B type of e-governance includes the transaction between government and Business firms. The main objective of G2B is to manage the hurdles for business, provide robust and immediate information and enable digital communication by e-business.
- Government 2 Citizen (G2C)- It is transaction between government and Citizens. The goal of G2C e-governance includes the utilization of ICT (information and communication technology) based services to citizens by maintaining the efficiency and accuracy, and to strengthen the relationship between government and citizens who are using the technology.
- Government 2 Employee (G2E)- This type relates the transaction between the government and their employees for providing the better governance in their services.
- Government 2 Government (G2G)- This type of e-governance includes the transaction of services between Government to Citizens and vice-a-versa. G2G is also a broad subject matter, but all relate to how the services and representation are now delivered and how they are now being implemented.

E-Governance Best Practices-

The following are some the best practices which have been implemented in India for developing the good governance. Few of the practices include-

- Fast & Easy Services.
- Increased accountability.
- Increased transparency.
- Reduced corruption.
- Increased efficiency due to connectivity.

- Tickets & Fines
- E-Voting.
- Vehicle registration.
- Transaction.

Benefits of E-Governance

If governance is transformed to electronic mode, then it comprises of lots of benefits. Few benefits include-

- The first benefit is that e-governance is a way towards good governance.
- It will provide better access to information and quality services for citizens.
- It will also expand the reach of governance to remote citizens.
- E-governance will make the governance as SMART governance by adding simplicity, moral, accountable, responsive and transparent traits in the traditional governance.

National E-Governance Plan (NeGP)-

It is an initiative taken by the government of India to make all government services available to the citizens of India via electronic media. NeGP was formulated by the Department of Electronics and Information Technology (DeitY) and Department of Administrative Reforms and Public Grievances (DARPG) government of India. The Government approved the National e-governance Plan, Consisting of various mission mode projects. Further these mission mode projects are categorized into three. First category is national, second category is state level and third is integrated mission mod project.

National Informatics Centre (NIC)

It was established in 1976. It is the premier science and technology organization of government of India in Informatics services and information and communication technology applications. It plays a very important role in handling e-governance websites and applications. All Indian government websites are developed and handled by NIC.

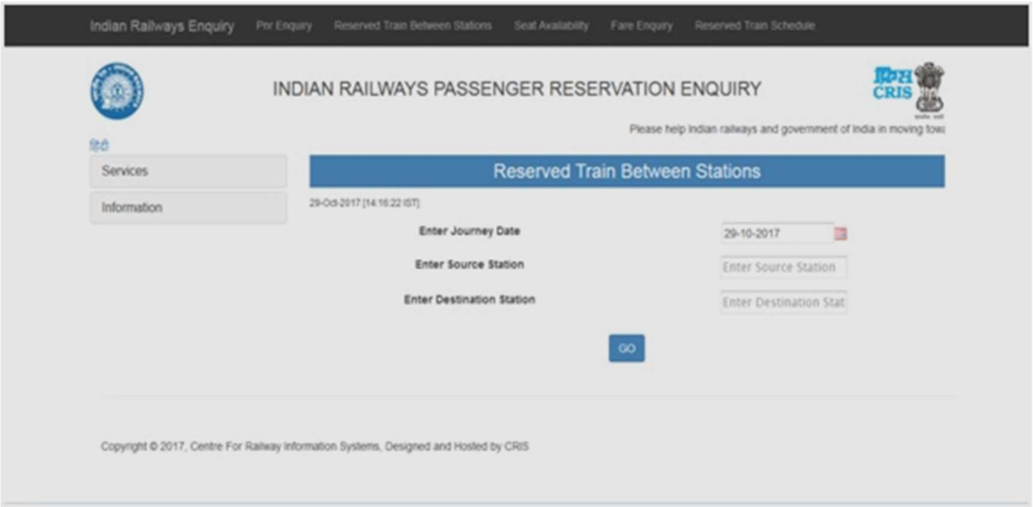
Some E-Governance portals and websites-

- Indian Railways- The official site with information on Trains, Fares and Availability is indianrail.gov.in. You may visit it for getting advantages on it.
- Ministry of Home affairs- It deals with internal security, demographics, promotes the official languages. The URL of the ministry of home affairs website is mha.nic.in.
- Educations and Research Network- This portal provides the information about state-of-the-art communication infrastructure and services to academic and research institutions, government organizations, NGO’s, private sector research and development organizations, and various other non-commercial organizations. The website url is els.ernet.in.

- Reserve Bank of India- Monetary authority, regulator and supervisor of the financial system, manager of exchange control and issuer of the national currency. The website URL is rbi.org.in.
- Ministry of Company Affairs: Regulates companies, stocks and shares. Details of protection for investors. Policy and administration details. The website url is mca.gov.in.
- Income Tax Department- Tax law, rules, and notifications, Tax return and other forms available for download. The website URL is incometaxindia.gov.in.
- Union Public service Commission- Recruitment and examination schedules conducted by the organizations. The website URL is upsc.gov.in.
- National Portal of India- It is an official website which provides access to information and their services which are provided by the Government. The website URL is india.gov.in.

All the sources are taken live from the internet just for showing the case study of the specific area. Here's some screen shots taken for e-governance websites-

For Railway Reservation-



INDIAN RAILWAYS PASSENGER RESERVATION ENQUIRY

Please help Indian railways and government of India in moving tow

Reserved Train Between Stations

29-Oct-2017 [14:16:22 IST]

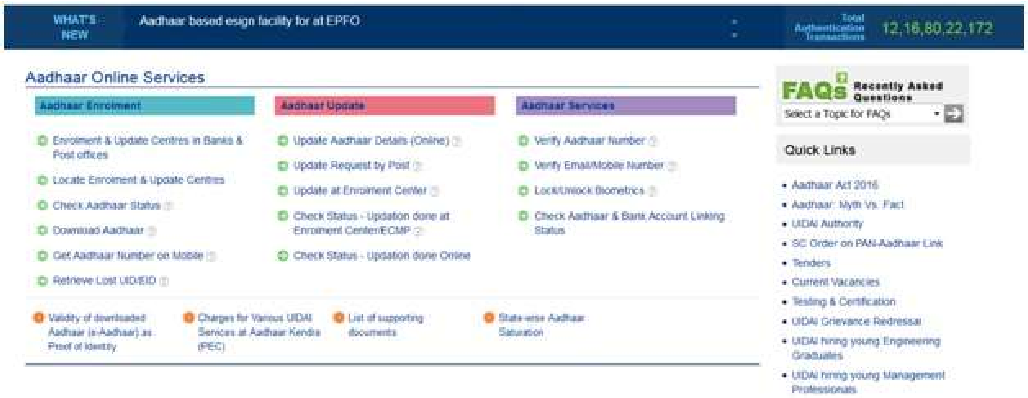
Enter Journey Date 29-10-2017

Enter Source Station Enter Source Station

Enter Destination Station Enter Destination Stat

Copyright © 2017, Centre For Railway Information Systems, Designed and Hosted by CRIS

For Aadhar card Information-



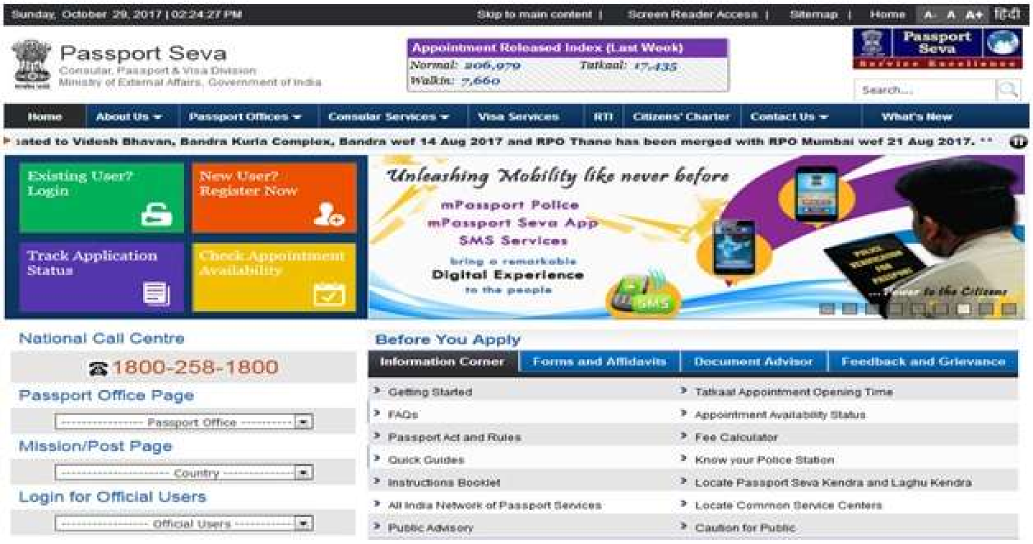
Aadhaar based esign facility for at EPFO

Total Authentication Transactions 12,16,80,22,172

Aadhaar Online Services

Aadhaar Enrolment
Enrolment & Update Centres in Banks & Post offices
Locate Enrolment & Update Centres
Check Aadhaar Status
Download Aadhaar
Get Aadhaar Number on Mobile
Retrieve Lost UID/EID
Aadhaar Update
Update Aadhaar Details (Online)
Update Request by Post
Update at Enrolment Center
Check Status - Updation done at Enrolment Center/ECMP
Check Status - Updation done Online
Aadhaar Services
Verify Aadhaar Number
Verify Email/Mobile Number
Lock/Unlock Biometrics
Check Aadhaar & Bank Account Linking Status
Validity of downloaded Aadhaar (e-Aadhaar) as Proof of Identity
Charges for Various UIDAI Services at Aadhaar Kendra (PEC)
List of supporting documents
State-wise Aadhaar Saturation
FAQs Recently Asked Questions
Select a Topic for FAQs
Quick Links
Aadhaar Act 2016
Aadhaar: Myth Vs. Fact
UIDAI Authority
SC Order on PAN-Aadhaar Link
Tenders
Current Vacancies
Testing & Certification
UIDAI Grievance Redressal
UIDAI hiring young Engineering Graduates
UIDAI hiring young Management Professionals

For Passport Information –

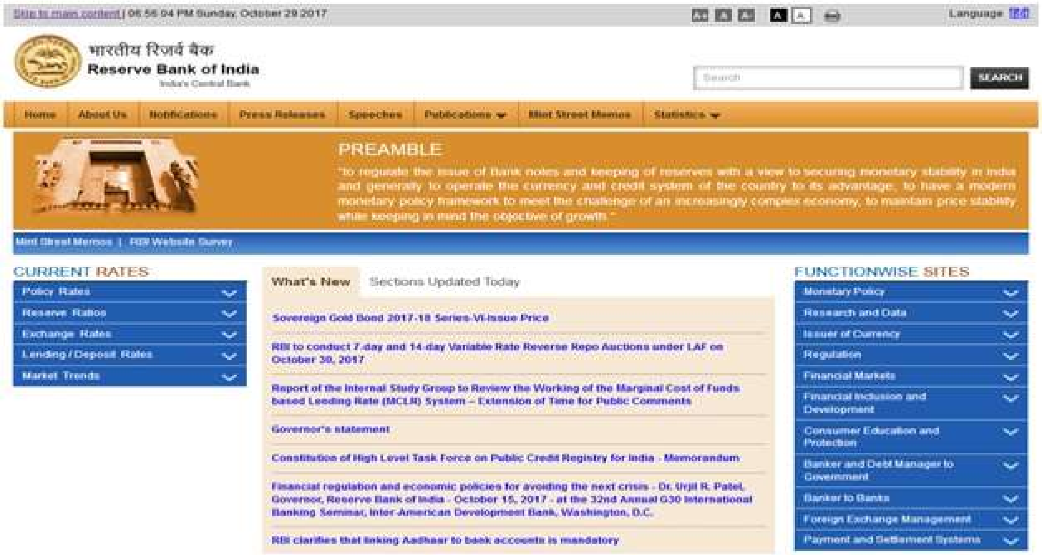


Appointment Released Index (Last Week)
Normal: 206,979 Tatkaal: 17,435
Walkin: 7,660

ated to Videsh Bhavan, Bandra Kurla Complex, Bandra wef 14 Aug 2017 and RPO Thane has been merged with RPO Mumbai wef 21 Aug 2017. **

For Banking-

There are a lot of government services which are being provided to the citizens of India. The mode of service is now available in electronic form with the help of their websites. This is called as e-governance model of the service. Moreover, these services are also being offered by using small mobile based apps such as Android operating system supported apps, Apple I Phone supported apps etc. The concept of providing the service using mobile apps is now known as m-Governance where m stands for mobile.



PREAMBLE

"to regulate the issue of Bank notes and keeping of reserves with a view to securing monetary stability in India and generally to operate the currency and credit system of the country to its advantage; to have a modern monetary policy framework to meet the challenge of an increasingly complex economy, to maintain price stability while keeping in mind the objective of growth."

Sovereign Gold Bond 2017-18 Series-VI-Issue Price

RBI to conduct 7-day and 14-day Variable Rate Reverse Repo Auctions under LAF on October 30, 2017

Report of the Internal Study Group to Review the Working of the Marginal Cost of Funds based Lending Rate (MCLR) System – Extension of Time for Public Comments

Governor's statement

Constitution of High Level Task Force on Public Credit Registry for India - Memorandum

Financial regulation and economic policies for avoiding the next crisis - Dr. Urjit R. Patel, Governor, Reserve Bank of India - October 15, 2017 - at the 32nd Annual G30 International Banking Seminar, Inter-American Development Bank, Washington, D.C.

RBI clarifies that linking Aadhaar to bank accounts is mandatory

3.12 POINTS TO REMEMBER

- Uniform Resource Locators (URL) - URLs are the web browser addresses of internet pages and files. With a URL, you can locate and bookmark specific pages and files for your web browser. The format of a URL resembles this- http://www.examplewebsite.com/mypage

- Hyper Text Transfer Protocol (HTTP), is the data communication standard of web pages. When a web page has this prefix, the links, text, and pictures should work properly in your web browser. Hypertext Transfer Protocol Secure (HTTPS) indicates that the webpage has a special layer of encryption added to hide your personal information and passwords from others.
- IP address- Your computer and every device that connects to the internet uses an Internet Protocol (IP) address for unique identification. An IP address (IP Version 4) can look something like- 202.3.104.55
- Electronic Governance (E-Governance) is the application of information and communication technology (ICT) for delivering government services to the citizens. It is the integration of various government services between governments to citizen (G2C), government to business (G2B), government to government (G2G), government to employees (G2E).

3.13 GLOSSARY

- Archie- A tool (software) for finding files stored on anonymous FTP sites. You need to know the exact file name or a substring of it.
- ARPANet- (Advanced Research Projects Agency Network) - Earlier name of internet. Developed in the late 60's and early 70's by the US Department of Defense as an experiment in wide area networking.
- Bandwidth- How much data traffic you can send through a connection. Usually measured in bits per second (bps). Higher bandwidth means more data transferred.
- Domain Name- The unique name that identifies an Internet site. Domain Names always have two or more parts, separated by dots. The part on the left side of the dot is the most specific, and the part on the right is the most general. For example- dictionary.com, education.com, thehindu.com and etc.
- Intranet- A private network inside a company or organization that uses the same kinds of software that you would find on the public Internet, but that is only for internal use.
- Modem- (Modulator Demodulator) - A device that you connect to your computer and to a phone line that allows the computer to talk to other computers through the phone system. Basically, modems do for computers what a telephone does for humans.

3.14 CHECK YOUR PROGRESS

Part- A (Multiple choice type questions)

(1) What is the software program that lets you search and see material on the Internet?

(A) Browser
(B) Website
(C) Webpage
(D) Homepage

(2) The World Wide Web address of a certain site is the...?

(A) Browser
(B) HTML
(C) ISP
(D) URL

(3) A network that links computers around the world is the...?

(A) Webpage
(B) Homepage
(C) Internet
(D) Browser

(4) What is the term given to the combination of two or more types of information such as text, video, audio, and pictures (graphics)?

(A) Multimedia
(B) HTML
(C) Browser
(D) Modem

(5) The server on the Internet is also known as a...

(A) Gateway
(B) Hub
(C) Host
(D) Repeater

(6) Voice mail, E-mail, Online service, the Internet and the WWW are all example of...

(A) Telecommunication
(B) Connectivity
(C) Computer categories
(D) None of the above

Part- B (Descriptive type questions)

(1) List five applications that use the Internet.

(2) What are data packets?

(3) What do you understand by ID Address? How are IP addresses assigned?

(4) Define URL with an example.

(5) What do you mean by E-governance? List any four advantages of E-governance.

(6) Define the following- Web Hosting, Domain name, Search Engine.

Answers- Part A- (Multiple choice type questions)

(1) A, (2) D, (3) C, (4) A, (5) C, (6) A

3.15 BIBLIOGRAPHY/REFERENCES

- Sluyters, Richard C. Van, Introduction to the Internet and World Wide Web, ILAR Journal, May 2018
- https://www.sandiegocounty.gov/library/PDF/gateway_introduction-to-internet.pdf, March, 2018
- https://www.encyclopedia.com/computing/news-wires-white-papers-and-books/internet-applications, December, 2017

3.16 SUGGESTED READINGS

- James, Scott D., Introduction to the Internet, 1998
- Forouzan, Behrouz A., Data Communications and Networking, McGraw-Hill, Inc. New York.
- Tanenbaum, A.S. (2002) Computer Networks, Prentice Hall.

UNIT 4

INTRODUCTION TO HTML

4.1 INTRODUCTION

HTML is short for Hyper Text Markup Language. HTML is a language of writing web pages. HTML document contains plain text, it can be written in even the simple text editor. Such as- Notepad, Brackets and etc. A more popular choice is to use a HTML editor is WYSIWYG ("What You See Is What You Get"). Such as- FrontPage, Dreamweaver, etc. At the time of saving HTML document, you will ensure to put an .html file extension or an .htm file extension.

4.2 OBJECTIVES

After successful completion of this unit you will be able to-

- Create simple web page using html.
- Use frequently used html tags.
- Create web form using various form elements.

4.3 SAMPLE HTML DOCUMENT

Note: Save the file with .html or .htm extension. As “myfirstfile .html”

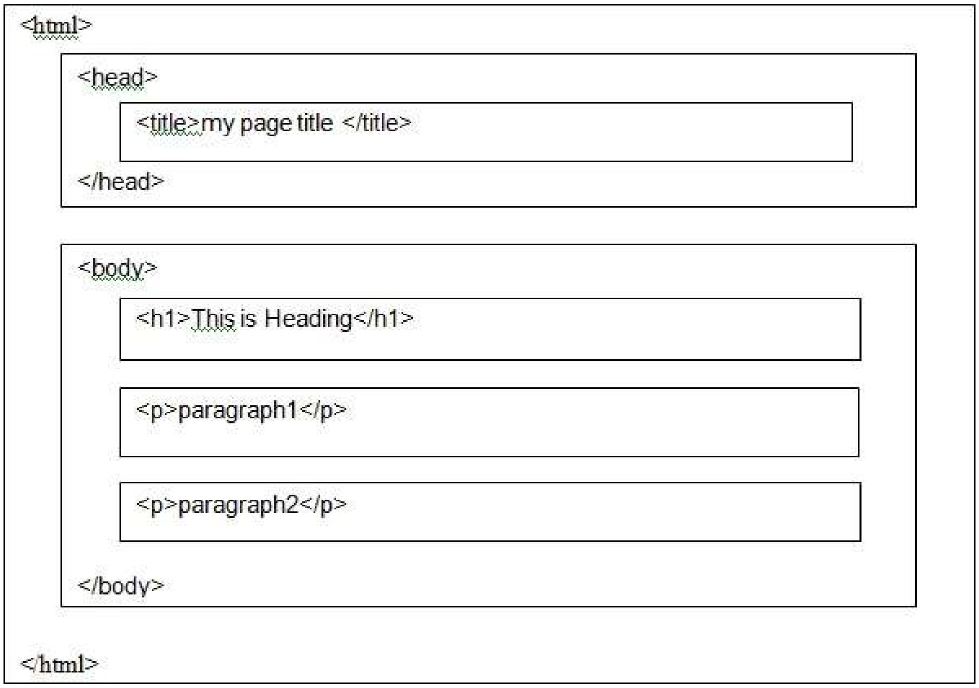
```
<!DOCTYPE html>
<html>
        <head>
                <title>Page Title</title>
        </head>
        <body>
                <h1>My First Heading</h1>
                <p>My first paragraph.</p>
        </body>
</html>
```

- The <!DOCTYPE html> declaration defines this document to be HTML5
- The <html> element is the root element of an HTML page
- The <head> element contains meta information about the document
- The <title> element specifies a title for the document
- The <body> element contains the visible page content
- The <h1> element defines a large heading
- The <p> element defines a paragraph

Web Browsers

The purpose of a web browser (“Chrome, IE, Firefox, Safari, etc”) is to read HTML documents and display them. The browser does not display the HTML tags, but uses them to determine how to display the document:

HTML Sample Page Structure-



```
<html>
    <head>
        <title>my page title </title>
    </head>

    <body>
        <h1>This is Heading</h1>

        <p>paragraph1</p>

        <p>paragraph2</p>
    </body>

</html>
```

4.4 HTML VERSIONS

Since the early days of the web development, there have been many versions of HTML: As-

HTML or HTML 1.0 (1991)-

HTML 1.0 was the first release of HTML to the world and written by Tim Berners-Lee.

HTML 2.0 (1995)-

HTML 2.0 was an expansion of the HTML language (HTML 1.0). HTML 2.0 was created to be a Web standard Even though the original version of HTML was never referred to as HTML 1.0.

HTML 3.2 (1997)-

HTML 3.0 was proposed a set of standards, released in 1995, after HTML 2.0. It would have added new features, including the ability to code tables, display mathematical equations, and allowing text flow around other objects, etc.

However, it was also designed to be backwards compatible[1] with HTML version 2.0. HTML version 3.0 was superseded by HTML 3.2 in 1997. It was the first version of HTML to be released as a W3C (World Wide Web Consortium) recommendation and implemented most of the features that were proposed in HTML 3.0.

HTML 4.01 (1999)-

HTML 4.0/4.1 was developed on the framework of HTML 3.2 and introduces several new and drastic elements that were not implemented in previous versions. Such new elements are- style

[1] Backward compatible refers to a software system that can successfully use interfaces and data from previous versions of the system.

sheets, scripting, frames, embedded objects, improved support for internationalization, richer tables, and enhancements to forms, etc. HTML 4 becomes a standard in April, 1998. Its updated version HTML 4.1 was published in December, 1999.

XHTML (2000)-

XHTML was a rewrite of HTML as an XML (eXtensible Markup Language). XML is a standard markup language that is used to create other markup languages. XML content can easily be shared across applications. This feature of the language makes XML potentially very powerful.

HTML5 (2014)-

HTML5 is the latest version of HTML to be standardized by the W3C. WHATWG (Web Hypertext Application Technology Working Group) began working group decided to develop HTML as a "Living Standard". A living standard is always updated and improved where new features can be added, but old functionality cannot be removed. The HTML5 was developed through collaborative efforts by W3C and WHATWG.

Initially, HTML was limited to static documents and accessed primarily by computers, but in recent years that HTML 4.0 was demanding by users to behave as “dynamic pages”. The dynamic pages behave more like applications than static documents, to use the web on different devices and platforms with high quality audio and video. The goals of HTML 5 is to replace proprietary media plug ins with open standards that allows web applications to behave like live and interactive applications.

The latest versions of browsers (such- Apple Safari, Google Chrome, Mozilla Firefox, Internet Explorer 9.0, Opera, etc.) all support many HTML5. The mobile web browsers that come pre-installed on iPhones, iPads, and Android phones all are also have an excellent support for HTML5.

4.5 FREQUENTLY USED HTML TAGS

HTML tags are element names surrounded by angle brackets-

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

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

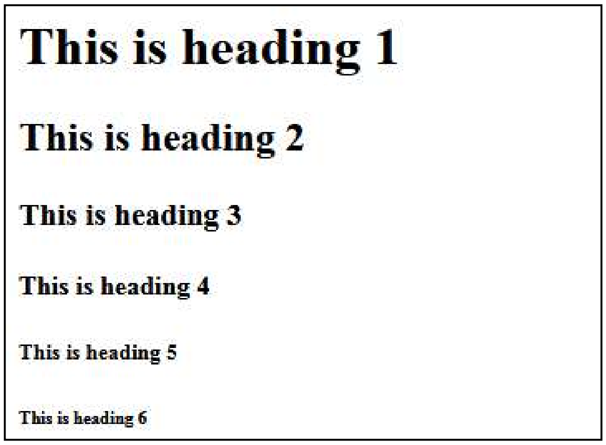
Heading tag-

Basically, there are six heading tags in HTML from h1 (largest) to h6 (smallest). As-

```
<h1>This is heading 1</h1>
<h2>This is heading 2</h2>
<h3>This is heading 3</h3>
<h4>This is heading 4</h4>
<h5>This is heading 5</h5>
```

```
<h6>This is heading 6</h6>
```

Output-



This is heading 1

This is heading 2

This is heading 3

This is heading 4

This is heading 5

This is heading 6

Paragraph tag-

Paragraph tag is used to define paragraphs in a web page.

Example-

```
<p>Paragraph text 1</p>

<p>Paragraph text 2</p>

<p>Paragraph text 3</p>
```

This tag adds some space (margin) before and after each <p> tag automatically.

Font Tag-

The <font> tag contains the "color", "face" and "size" attributes. It is an inline element and also useful for changing font, font size and font colour.

Example-

```
<!DOCTYPE html>

<html>
        <head>
                <title>HTML font Tag</title>
        </head>
        <body>
                <font face = "arial" color = "green" size = "4">
                Uttarakhand Open University, Haldwani, Nainital
                </font>
        </body>
</html>
```

Note: The <font> tag is not supported in HTML5, for the same purpose CSS is used.

Images Embedding-

The image tag (<img>) is used to define the images in to a web page. The <img> tag is non paired, which contains the following attributes. As- "alt", "height", "width".

The alt (alt=”This is Image 1”) attribute provides an alternate text for an image, if the user for some reason cannot view the image, the value of the alt attribute should describe the image into a web page. As-

<img src="img_rose.jpg" alt="Rose flower- My favourite">

You can decide the height and width of your image. As-

<img src="img_rose.jpg" alt="Rose flower- My favourite" height=”100” width=”125”>

Example 1-

```
<!DOCTYPE html>
<html>
        <head>
                <title>My Image Gallery</title>
        </head>
        <body>
                <p>Image Gallery</p>
                <P>Fruit</p>
                <img src="fruit.jpg" width="200" height="150" alt="My fruit Image">
                <P>Music</p>
                <img src="music.jpg" width="200" height="150" alt="My Music Image">
        </body>
</html>
```

Hyper linking-

A link or a hyperlink is an icon, graphic, or text in a document that links to another file or object. All the web pages in the World Wide Web are comprised of hyperlinks to each other.

Example 1-

<a href="www.google.com">Click Here to open Google.com</a>

Example 2-

You have two files as- “Home.html” and “address.html”, and we want to create a hyper link between them.

Note: Both the files (Home.html, address.html) must be saved in a same folder, otherwise need to mention the complete address of the file where it locates.

File name- “Home.html”

```
<!DOCTYPE html>
<html>
        <head>
                <title>My Home Page</title>
        </head>
        <body>
```

```
                <p>This is my Home Page</p>
                <a href=”address.html”>Click here to see my Address</a>
        </body>
</html>
```

File name- “address.html”

```
<!DOCTYPE html>
<html>
        <head>
        <title>My Address Page</title>
        </head>
        <body>
                <p><b> My Address-</b></p>
                <p>Uttarakhand Open University, University Marg, Haldwani- 263139</p>
                <a href=”home.html”> <p><b> Click here to go back to Home page</b></p></a>
        </body>
</html>
```

Hyperlinks-"target" Attribute

The attribute "target" specifies where to open the linked page. This attribute can have one of the following values. As- "_blank", "_self", "_parent", "_top", "frame_name".

Example-

<a href=" www.google.com" target="_blank"> Click Here to open Google.com in a Blank Tab</a>

Note: the target="_blank" opens the linked document in a new window or tab.

Example-

<a href=" www.google.com" target="_self"> Click Here to open Google.com in a same window/tab</a>

Note: the target="_self" opens the linked document in the same window/tab as it was clicked (this is by default).

Example-

<a href=" www.google.com" target="_parent"> Click Here to open Google.com in parent frame/tab</a>

Note: the target="_ parent" opens the linked document in the parent frame.

Example-

<a href=" www.google.com" target="_top"> Click Here to open Google.com in the full body of the window</a>

Note: the target="_ top " opens the linked document in the full body of the window.

Example-

```
<a href=" www.google.com" target="name_of_the_frame"> Click Here to open Google.com in the frame name provided by you</a>
```

Note: the target="_ top" opens the linked document in the frame name provided by you.

Hyperlinks- Linking images-

Example-

```
<!DOCTYPE html>
<html>
        <head>
        <title>Image Liking</title>
        </head>
        <body>
                <p><b>Click on the below Images to follow the link</b></p>
                <a href="Myfile1.html" target="_blank"><img src="img1.jpg" width="30" height="25" alt="My Image 1"></a>
                <a href="Myfile2.html" target="_blank"><img src="img2.jpg" width="30" height="25" alt="My Image 2"></a>
        </body>
</html>
```

Table tag-

When you want to present data in a web page in the form of row and column, then we required HTML table tag (<table>..... <table/>).

Example- we want to create the following table in HTML.

Roll No.	Name	Class	Mobile No.
101	Ankit Singh	B.Sc. II Sem.	1234567890
102	Javed Aalam	BCA III Sem.	9876543210

```
<!DOCTYPE html>
<html>
        <head>
                <title>Table</title>
        </head>
        <body>
                <p>Table</p>
                <table border="1">
                        <tr>
                <td>Roll No.</td>
                <td>Name</td>
                <td>Class</td>
                <td>Mobile No.</td>
            </tr>
            <tr>
                <td>101</td>
                <td>Ankit Singh</td>
                <td>B.Sc. II Sem.</td>
                <td>1234567890</td>
            </tr>
            <tr>
                <td>102</td>
                <td>Javed Aalam</td>
                <td>BCA III Sem.</td>
                <td>9876543210</td>
            </tr>
        </table>
    </body>
</html>
```

Table Attributes-

Align (left, center, right)

Bgcolor (rgb(x,x,x), #xxxxxx, colorname)

Border (0,1)

Cellpadding (pixels)

Cellspacing (pixels)

Width (pixels, %)

Example 1-

```
<table border="1" align="center" width="500" cellpadding="2" cellspacing="0" bgcolor="“”">
    <tr>
        <td>Roll No.</td>
        <td>Name</td>
        <td>Class</td>
        <td>Mobile No.</td>
    </tr>
    <tr>
        <td>11</td>
        <td>ABC</td>
        <td>XII</td>
        <td>2233441234</td>
    </tr>
</table>
```

List tag-

Sometimes we require bullets or numbering in a web page to display any information point wise. We used two types of listing in HTML. As-

- Ordered List (<ol>....</ol>)
- UnOrdered List (<ul>....</ul>)

Example**-** We wants to display the following information into a web page-

Types of Computers-

Super Computer

Mainframe Computer

Mini Computer

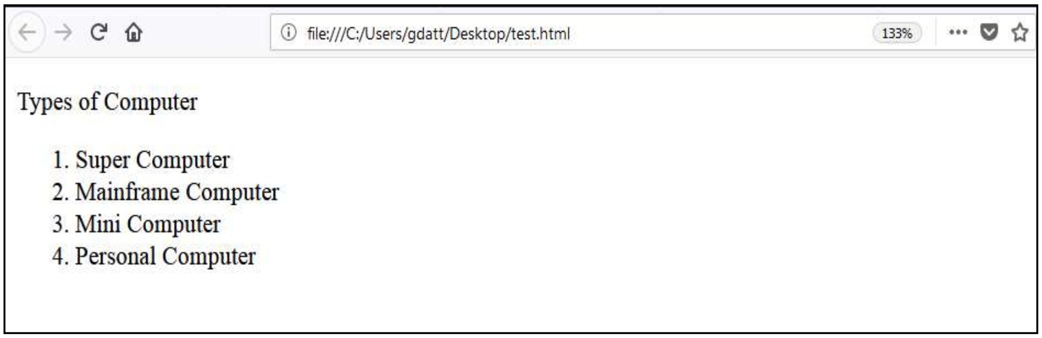
Personal Computer

Example 1- (Ordered List)

```
<!DOCTYPE html>

<html>
    <head>
        <title>Ordered List</title>
    </head>
    <body>
        <p> Types of Computer </p>
        <ol>
            <li>Super Computer</li>
            <li>Mainframe Computer</li>
            <li>Mini Computer</li>
            <li>Personal Computer</li>
        </ol>
    </body>
</html>
```

Output-



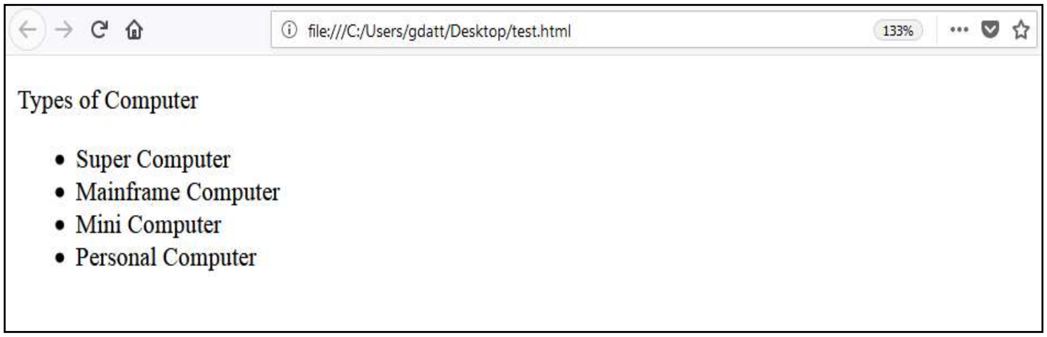
Types of Computer

1. Super Computer
2. Mainframe Computer
3. Mini Computer
4. Personal Computer

Example 2- (Unordered List)

```
<!DOCTYPE html>
<html>
        <head>
                <title>Unordered List</title>
        </head>
        <body>
                <p> Types of Computer </p>
                <ul>
                        <li>Super Computer</li>
                        <li>Mainframe Computer</li>
                        <li>Mini Computer</li>
                        <li>Personal Computer</li>
                </ul>
        </body>
</html>
```

Output-



Types of Computer

- Super Computer
- Mainframe Computer
- Mini Computer
- Personal Computer

Ordered List- Attributes

The following attributes are used with list tag. As- “Type” and “start”.

The “type” attribute and its respective values-

Attribute	Value	Description
Type=	“1”	The list will be numbered with numbers (default)

	“A”	The list will be numbered with uppercase letters
	“a”	The list will be numbered with lowercase letters
	“I”	The list will be numbered with uppercase roman numbers
	“i”	The list will be numbered with lowercase roman numbers

The “start” attribute and its respective values-

Attribute	Value	Description
Type=	“3”	The list will start from 3rd place. (by default list starts from first place)

Example-

```
<ol type="A" start=”4”>
        <li>Green</li>
        <li>Red</li>
</ol>
```

UnOrdered List- Attributes

The following attribute is used with list tag. As- “Type”

The “type” attribute and its respective values-

Attribute	Value	Description
Type=	“disc”	The list will be displayed with discs (default)
	“circle”	The list will be displayed with circles
	“square”	The list will be displayed with square
	“none”	The list item will not be marked

Example-

```
<ul type="square">
        <li>Banana</li>
        <li>Apple</li>
        <li>Mango</li>
</ul>
```

Inline frame-

The inline frame (<iframe>) tag can appear anywhere in the web page, it defines a rectangular region within the document in which the browser can display another web page, including scrollbars and borders (if needed). As-

Example-

```
<iframe src=" Name_of_Web_page.html"></iframe>
```

Iframe- Attributes

The following attributes are used with <iframe>. As:

Attribute	Value	Description

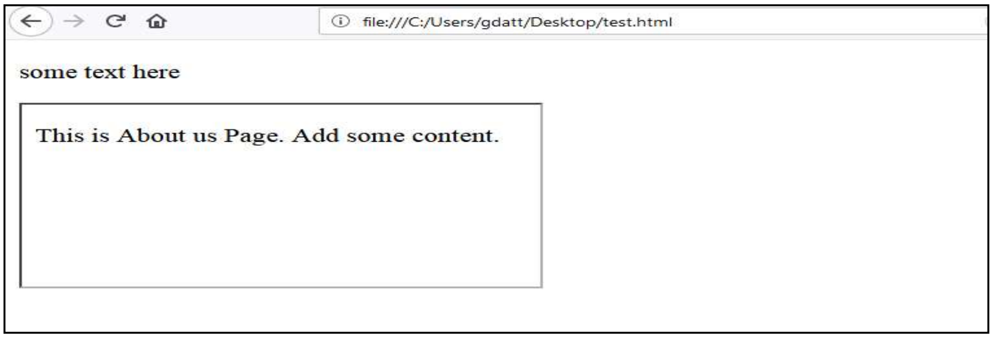
align=	“left” “right” “top” “bottom”	Specifies the alignment of an <iframe> according to the web page.
frameborder=	“1” “0”	Defines border of the <iframe>.
height=	In pixels. As: “200”	Defines height of the <iframe>.
width=	In pixels	Defines width of the <iframe>.
name=	“Any_Valid_Name”	Defines name of the <iframe>, so that this can be identified.
src=	“Valid_file_name”	Defines source/URL to open a page inside the <iframe>.
scrolling=	“yes” “no” “auto”	Defines scrolling into <iframe>. If “yes”, scroll bar will available.
marginwidth=	In pixels	Defines the left and right margins of the content of an <iframe>.
marginheight=	In pixels	Defines the top and bottom margins of the content of an <iframe>.

Example-

Note- The below code is written in “home.html” and “about.html” is already created.

```
<!DOCTYPE html>
<html>
        <head>
                <title>Inline Frame</title>
        </head>
        <body>
                <p> some text here </p>
                <iframe name=”my_iframe1” src="about.html" height=”300”
width=”250” scrolling=”auto” frameborder=”1”>
                </iframe>
        </body>
</html>
```

Output-



file:///C:/Users/gdatt/Desktop/test.html

some text here

This is About us Page. Add some content.

4.6 HTML FORM ELEMENTS

The form element or the form tag (<form>.... </form>) contains several elements. As- input, select, text area, button, etc. We can use several attributes to form element but two are special as- “action” and “method”.

The "action" attribute indicates the location of the script that will process the information sent by the browser, when the form is submitted. This script is written in any server-side language and is supposed to be prepared to receive and process the form's data and the "method" attribute indicates the messenger through which we can send the information stored by the form.

Note:

The form attribute 'method' can contain the value either 'get' or 'post', which specifies the HTTP method to use when the form is submitted. As-

Get- With 'GET' the form data is encoded into a URL by the browser. The form data is visible in the URL allowing it to be bookmarked and stored in web history. The URL length is limited, so that there can be limitations on how much form data can be sent.

Post- With 'POST' all the name value pairs are submitted in the message body of the HTTP request which has no restrictions on the length of the string. The name value pairs cannot be seen in the web browser's address bar.

Form Controls-

- Text Input Controls
- Checkboxes Controls
- Radio Box Controls
- Select Box Controls
- Buttons (Clickable Buttons, Submit and Reset Button)

An HTML form element is any element that may be used inside of a <form> tag. These elements, along with the unique and standard attributes are-

Text Input Controls-

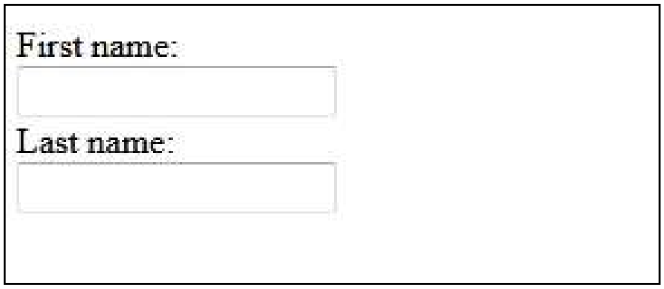
The <input> element can be displayed in as-

```
<input type="text" name=”anyNameHere”>
```

Example-

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

Output-



First name:

Last name:

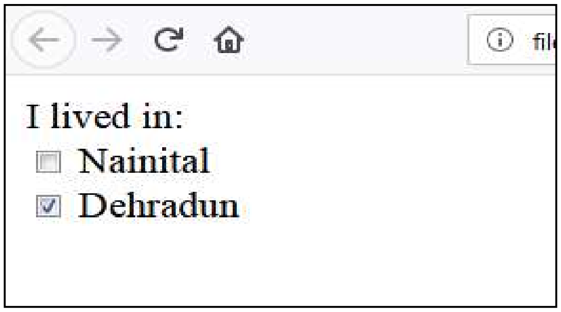
Checkboxes Controls-

Checkboxes Control specifies that an <input> element should be pre-selected (checked) when the page loads. The checked attribute is a boolean attribute.

Example-

```
<!DOCTYPE html>
<html>
        <head>
                <title>Chexk Box Example</title>
        </head>
        <body> I lived in:<br>
                <form action="name_of_file" method="">
                        <input type="checkbox" name="me1" value="City1"> Nainital<br>
                        <input type="checkbox" name="me2" value="City2" checked> Dehradun<br>
                </form>
        </body>
</html>
```

Output-



I lived in:

□ Nainital

☑ Dehradun

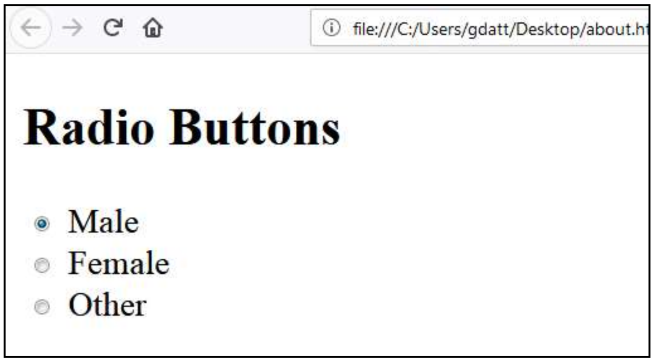
Radio Box Controls-

A radio button is a form element that allows the user to select one option from a range of options. As-

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

```
        <head>
                <title> Radio Buttons</title>
        </head>
        <body>
                <h2>Example- Radio Buttons</h2>
                <form>
                        <input type="radio" name="gender" value="male" checked> Male<br>
                        <input type="radio" name="gender" value="female"> Female<br>
                        <input type="radio" name="gender" value="other"> Other
                </form>
        </body>
</html>
```

Output-



file:///C:/Users/gdatt/Desktop/about.ht

Radio Buttons

● Male
○ Female
○ Other

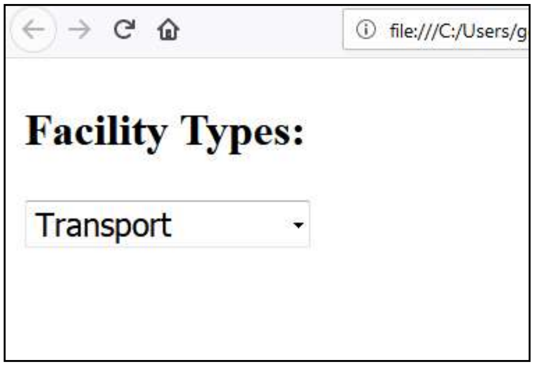
Select Box Controls-

The <select> element is used to create a drop-down list. The <option> tag inside the <select> element is used. As-

Example-

```
<!DOCTYPE html>
<html>
        <head>
                <title> Select Box</title>
        </head>
        <body>
                <form>
                        <h3>Facility Types:</h3>
                        <select>
                                <option value="Library">Library</option>
                                <option value="CompLab">Computer Center</option>
                                <option value="Hostel">Hostel</option>
                                <option value="Transport">Transport</option>
                        </select>
                </form>
        </body>
</html>
```

Output-



file:///C:/Users/g

Facility Types:

Transport

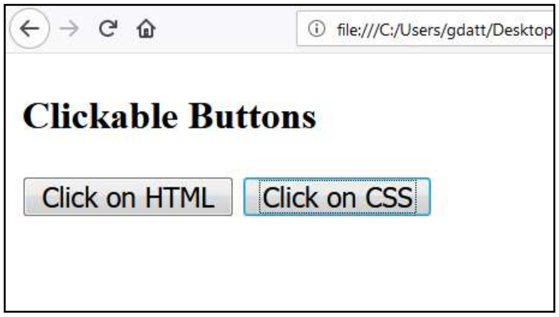
Buttons (Clickable Buttons, Submit and Reset Button)-

Generally, three types of buttons are used in html, as- Button, submit button and Reset button. Submit button is used to send form data, reset button is used to reset form fields and Button is simply a clickable button used as per situation. As-

Example- (Clickable Buttons)

```
<!DOCTYPE html>
<html>
        <head>
                <title> Clickable Buttons</title>
        </head>
        <body>
                <form>
                        <h3>Clickable Buttons</h3>
                        <button name="subject" type="button" value="HTML"> Click on HTML</button>
                        <button name="subject" type="button" value="CSS">Click on CSS</button>
                </form>
        </body>
</html>
```

Output-



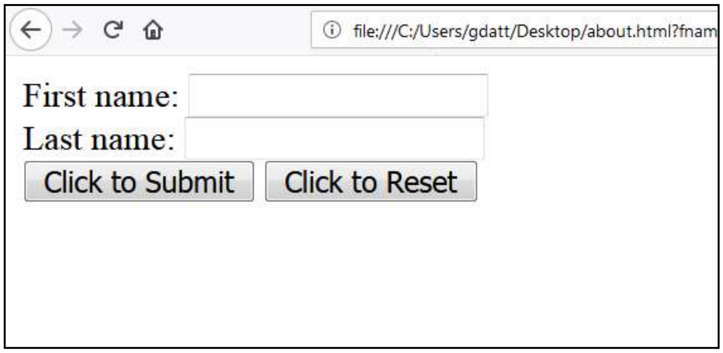
file:///C:/Users/gdatt/Desktop

Clickable Buttons

Example- (Submit and Reset Buttons)

```
<!DOCTYPE html>
<html>
        <head>
                <title>Submit and Reset Buttons</title>
        </head>
        <body>
                <form action="name_of_file" method="get" id="form1">
                        First name: <input type="text" name="fname"><br>
                        Last name: <input type="text" name="lname"><br>
                        <button type="submit" form="form1" value="Submit">Click to Submit</button>
                        <button type="reset" form="form1" value="reset">Click to Reset</button>
                </form>
        </body>
</html>
```

Output-



Note- When you click on ‘Submit Button’ it sends the form data further for necessary action and when you click on ‘Reset Button’ it reset all the form fields.

4.7 POINTS TO REMEMBER

- HTML is the universal markup language for the Web.
- HTML lets you format text, add graphics, create links, input forms, frames and tables, etc., and save it all in a text file that any browser can read and display.
- You can also style individual words in HTML to make them, bold, italic, underline etc.
- Meta Description Tag allows you to suggest how your page should be described in search listings. Meta descriptions can help your website stand out from the others, resulting in additional clicks. Incorporating specific keywords from your research is also a tip.

4.8 GLOSSARY

- Div- A block level container (or 'division' of the web page) for content with no semantic meaning.

- Head- Tag that surrounds important content that is invisible to the user, but is important to the browser. Elements within this tag contain metadata about the page and links to style sheets, scripts, etc.
- id- An HTML element can have an id attribute to identify it. id elements should always be unique to that single element, and each element should never have more than one id. As- <div id="Detail_Sch">School Details</div>
- Body- The body is the container for all of the page's content, Comes after the <head> tag, within the overall <html> tag.
- Comments- HTML comments are sometimes used in code to explain parts of the markup. They are similar to comments in other languages. Users do not see comments in their browser.
- Horizontal rules- This tag creates a black line one pixel thick that runs the all the way across its container. As- <hr>

4.9 CHECK YOUR PROGRESS

Descriptive type questions-

- What is the difference between form's get and post method?
- What is the importance of the HTML DOCTYPE?
- What is web application?
- Name some ways to define a color in html?
- Define with suitable example HTML heading levels.
- Define any 5 html paired tags.
- How do you insert a comment in HTML? Example.
- Define ordered list and unordered list in html.

4.10 BIBLIOGRAPHY/REFERENCES

- https://www.austincc.edu/hr/profdev/eworkshops/docs/HTML_Basics.pdf, December, 2017
- https://www.cs.drexel.edu/~spiros/teaching/SE101/slides/html.pdfDecember, 2017
- https://www.w3schools.com/html/html_intro.asp, January, 2018
- https://www.tc.columbia.edu/idesign/resources/past-workshop-archive/HTML-and-CSS-Workshop.pdf, November, 2017
- http://network-communications.blogspot.com/2011/05/introduction-to-html-basic-html-tags.html, January, 2018

4.11 SUGGESTED READINGS

- Powell. Thomas A., HTML & CSS: The Complete Reference, 5E
- Holzner. Steven, PHP: The Complete Reference.
- Powell. Thomas A., JavaScript: The Complete Reference

UNIT 5
INTRODUCTION TO CASCADING STYLE SHEET

5.1 INTRODUCTION

CSS was first proposed by Hakon Wium Lie On October 10, 1994. Lie is known as the father of HTML at CERN, who has worked with Tim Berners. Becaopt of his contribution in CSS Hakon Wium Lie is also called as father of CSS. When CSS was first proposed in the year 1994, it was well thought-out as a web styling language, so as to solve some HTML 4 problems. Other than CSS there was some styling languages proposed at that time. Some of them are Style Sheets for HTML and JSS. But CSS was better than them all.

W3C developed CSS level 2 (also called CSS 2) specifications and also published it as a recommendation in May 1998. A number of new capabilities were also included in CSS 2 which is as follows- Absolute, Bidirectional text, Concept of media type, fixed, new font properties such as shadows, positioning, z-index

Although, in 1998 CSS3 was started but some of its parts are in development stage while some of its components work on some of the browsers. On June 1999, it got published. Several separate documents in which CSS3 is divided are called *modules*. All the new capabilities or extended features added by each module are distinct in CSS2.

5.2 OBJECTIVES

After successful completion of this unit, you will be able to-

- Use CSS in your web page.
- Embed styles into your html file itself.
- Set margin and borders for your web page.
- Design effective web page using CSS.

5.3 SELECTOR AND DECLARATION

CSS is an acronym of Cascading Style Sheets. It's a rule-based language and opted to pertain styling to HTML elements. The Properties of element such as color, background color, width, border thickness, font size, etc. can be modified by opting or writing CSS rules in HTML elements such as <p>, <img> and many more.

CSS rules

The rules of CSS are divided into two parts- Selector and Declaration

Selector and Declaration-

In CSS, selectors are the patterns which are opted to opt for the element(s) which we want to style. This part of CSS comprises of properties and the values correspondingly.

Selector-

Selectors are opted to locate or opt for the contents we want to style. That are integrated in the CSS rules set. Selectors in CSS necessitate id, class, type, attribute etc. to employ the style. Compound selectors moreover are distinct at a time, while in this case every selector is separated from other selectors by opting comma separator. Example of opting multiple selectors in a single block is mentioned below-

Example-

```
h1,h2,h3

{

        color:blue;

        font-size:12px;

}
```

Property and Value

The property is the style attribute we require to change and value is the value of particular attribute. For example, we would like to change the size of our text 28px and the color is blue then it declares as-

Example-

font-size:28px; color:blue.

Here font-size is the property and 28px is their value and color is also a property and the blue is there value.

Declaration-

Declaration part of the CSS comprises the property and value. For example, if we crave for the size of our text 10px then it declares as font-size:1px. In this font-size is the property and 10px is their value, and all this statement is called declaration.

Declaration Block

In the Declaration Block (given in the diagram above) several lines are integrated which also includes the curly braces. This block is worn for declaring the properties of the selector. The declaration part is also alienated into two parts which are detached by a ':' symbol called as colon. The example is given below:

```
font-size:12px;
```

As shown in the above code property (font-size) and value (12px) are separated by a colon symbol.

Curly Braces

In the declaration section if one property is opted then there is no need to end it or opting semicolon at the end of the line. If declaration part consists numerous properties and their values, then it must be ended with a separator, called semicolon.

Class and ID in CSS-

HTML elements are applied in CSS with selectors Class and ID. When applying CSS on HTML elements at that time ID and class selectors are mainly opted. CSS would be functional in an HTML element by opting any of these two-selector id and class. The benefit of id and class is that there is no need to write code again and again. That is the feature of CSS called, reusability.

ID Selector

With the opts of ID selector we can indicate a style for a single, unique element. So that the id attribute of the CSS on HTML element is opted and is distinct with a hash sign. The syntax of opting ID is given below:

```
#para1
{
        text-align:left;
        color:blue;
}
```

The example of opting ID is given below-

```
<HTML>
        <body>
```

```
                <style>
                        #left{
                                text-align:left;
                        }
                        <h1 id="left">This is heading</h1>
                        <p id="left">This is paragraph</p1>
                </style>
        </body>
</HTML>
```

Result of the above code is given below-

This is heading

This is paragraph

Note: Do not start an ID name with a number. It is not supported in Mozilla Firefox browser.

Class Selector-

To state style for a set of elements the class selector is opted. An id selector, class selector permits us to set a particular style for many HTML elements with the same class. HTML attributes, are opted by class selector which are distinct with a '.' (dot) sign.

In the below example we declare a class with name "left" (.left) which is opted many HTML attribute for display that particular element in left. The example of opting class selector is given below:

```
<HTML>
        <body>
                <style>
                        .left{
                                text-align:left;
                        }
                </style>
                <h1 class="left">This is heading</h1>
                <p class="left">This is paragraph</p>
        </body>
</HTML>
```

The output of the above code is:

This is heading

This is paragraph

We can also state only specific HTML elements should be affected by a class. In below example, all <p> elements will be displayed in centre. For example-

```
p.left{
        text-align:center;
}
```

It is advised not to start a class name with a number. Because it would only support by Microsoft Internet Explorer browser.

5.4 STYLING BACKGROUNDS IN CSS

The background property of CSS is opted to distinct the background effects on HTML element. It is opted to set one or more background images for an element. The background of an element is the total size of an element, including padding and border (but not the margin). There are 5 CSS background properties that affect the HTML elements.

- background-color
- background-image
- background-repeat
- background-position
- background-attachment

CSS Background Color-

The background-color property is opted to set the background color of the HTML element. Example-

```
<HTML>
        <head>
                <style>
                        h2{
                                background: yellow;
                        }
                        p{
                                background: green;
                        }
                </style>
        </head>
        <body>
                <h2>This is h2 Heading</h2>
                <p>This is paragraph</p>
        </body>
</HTML>
```

This is h2 Heading

This is paragraph.

The output of the above code is-

In above example we set the background color green for <p> and yellow for <h3>.

Background Image-

The background-image property is opted to put an image in the background of the HTML element. By default, the image is repeated so it covers the entire element. Example is given below:

```
<HTML>
    <head>
        <style>
            p{
                background-image:
url(http://www.google.com/CSS/sublogo.png);
            }
        </style>
    </head>
    <body>
        <p>This is paragraph</p>
    </body>
</HTML>
```

Output of the above code is given below-

This is paragraph.

CSS Background Repeat-

The background- repeat property is opted to repeat an image in the background. By default, the background-image property repeats the background image horizontally and vertically. Some images are repeated only horizontally or vertically. If we do not want to repeat background image then set no-repeat.

The following code represents the opt of repeat in x-axis-

```
<HTML>
    <head>
        <style>
            p{
                background-image
url(http://www.google.com/CSS/files/sublogo.png);
                background-repeat: repeat-x;
            }
        </style>
    </head>
    <body>
        <p>This is paragraph</p>
    </body>
</HTML>
```

The result of the above code is given below:

This is paragraph.

The example of repeat in y-axis is given below-

```
<HTML>
        <head>
                <style>
                        p{
                                background-image url
(http://www.google.com/CSS/files/sublogo.png);
                                background-repeat: repeat-y;
                        }
                </style>
        </head>
        <body>
                <p>This is paragraph</p>
        </body>
</HTML>
```

The result of the above code is given below:

This is paragraph.

CSS Background Position-

Background-position is opted only when we have opted image in background of any HTML elements. It is opted to distinct the initial position of the background image. By default, the background image is placed on the top-left corner of the webpage. We can set position of the background and all background image positions are given below- top, bottom, center, left, right

5.5 FONT AND FONT STYLING

CSS font is opted for design text or font for display on web page. CSS font properties distinct the font family, boldness, size, and the style of a text.

Font Style-

Font style property is opted to set the font style for a text. Font style property has three values, they are:

- Normal: When normal is opted, the text is shown normally.

- Italic: When italic is opted, the text is shown in italics.
- Oblique: When oblique is opted, the text is shown "leaning" (oblique is very similar to italic, but less supported).

Example of opting font style is depicted in the example given below-

```
<HTML>
        <head>
                <style>
                        h1{
                                font-style: normal;
                        }
                        h2{
                                font-style: italic;
                        }
                        p{
                                font-style: oblique;
                        }
                </style>
        </head>
        <body>
                <h1>This is h1 Heading, normal.</h1>
                <h2>This is h2 Heading, italic.</h2>
                <p>This is Paragraph, oblique.</p>
        </body>
</HTML>
```

Output of the program is given below-

```
This is h1 Heading, normal.
This is h2 Heading, italic.
This is Paragraph, oblique.
```

Font Size-

The font-size property is opted to set the size of a font. We can distinctly size of font in following way-

- Em
- Percent
- Pixels

Font size in Em-

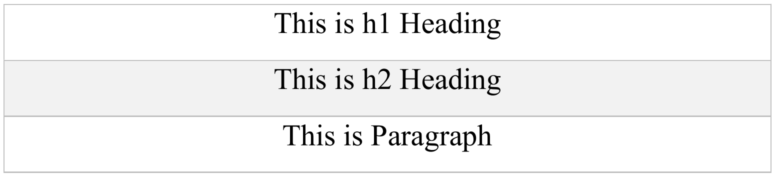
The em size unit is suggested by the W3C. 1em is equal to the default text size in browsers (16px). It permits optrs to resize the text (in the browser menu), many developers opt em instead of pixels. The formula which is opted to convert pixels to em is given below-

pixels/16=em

In the following example HTML code, we have opted em in font size-

```
<HTML>
        <head>
                <style>
                        h1{
                                font-size: 2.5em; /* 40px/16=2.5em */
                        }
                        h2{
                                font-size: 1.875em; /* 30px/16=1.875em */
                        }
                        p{
                                font-size: 0.875em; /* 14px/16=0.875em */
                        }
                </style>
        </head>
        <body>
                <h1>This is h1 Heading</h1>
                <h2>This is h2 Heading</h2>
                <p>This is Paragraph</p>
        </body>
</HTML>
```

The above code gives the following output-



This is h1 Heading

This is h2 Heading

This is Paragraph

Font size in percent-

One em is equal to 16px which is also equal to 100% in normal browser view. Text size works on all browsers, we can opt default font-size in percent for the <body> element. The following HTML code depicts the font size in percent-

```
<HTML>
        <head>
                <style>
                        body {
```

```
                font-size: 100%;
            }
            h1{
                font-size: 2.5em;
            }
            h2{
                font-size: 1.875em;
            }
            p{
                font-size: 0.875em;
            }
        </style>
    </head>
    <body>
        <h1>This is h1 Heading</h1>
        <h2>This is h2 Heading</h2>
        <p>This is Paragraph</p>
    </body>
</HTML>
```

The above code gives the following output-

```
This is h1 Heading
This is h2 Heading
This is Paragraph
```

Font size in pixels-

We may also opt pixels units to distinct the size of the text. If we are setting the text size with pixels then we can provide full control over the text size. The key point to be noted here is that if we opt pixels, we can still opt the zoom tool to resize the entire page. The syntax of opting pixels to set text size is similar to as opted in percent. In the following code the size of font is set opting pixel-

```
<HTML>
    <head>
        <style>
            h1{
                font-size: 40px;
            }
            h2{
                font-size: 30px;
            }
            p{
```

```
                        font-size: 14px;
                    }
            </style>
        </head>
        <body>
            <h1>This is h1 Heading</h1>
            <h2>This is h2 Heading</h2>
            <p>This is Paragraph</p>
        </body>
</HTML>
```

The above code gives the following output-

This is h1 Heading
This is h2 Heading
This is Paragraph

5.6 PSEUDO CLASSES

A pseudo-class is a keyword which we can add to CSS selectors in order to distinct a special state of the belonging HTML element. For example, :hover can be opted to change a link's color when the mouse hovers over it.

A pseudo-class can't exist on its own. It must be close to a selector. The pseudo-class will only distinct a picky state of that selector. It has the following syntax-

selector:pseudo-class {property: value}

CSS classes can also be opted with pseudo-classes as given below-

selector.class:pseudo-class {property: value}

The most commonly opted pseudo-classes are given in the table below-

Value	Description
:link	Opt this class to add special style to an unvisited link.
:visited	Opt this class to add special style to a visited link.
:hover	Opt this class to add special style to an element when we move over it.
:active	Opt this class to add special style to an active element.
:focus	Opt this class to add special style to an element while the element has focus.

:first-child	Opt this class to add special style to an element that is the first child of some other element.
:lang	Opt this class to state a language to opt in a specified element.

While defining pseudo-classes in a <style>...</style> block, following points should be followed by us-

- a:hover must come after a:link and a:visited in the CSS definition in order to be effective.
- a:active must come after a:hover in the CSS definition in order to be effective.
- Pseudo-class names are not case-sensitive.
- Pseudo-class is different from CSS classes but they can be combined.

The :link pseudo class-

The following example demonstrates how to opt the :link class to set the link color. Possible values could be any color name in any valid format.

```
<HTML>
        <head>
                <style type="text/CSS">
                        a:link {color:#000000}
                </style>
        </head>
        <body>
                <a href="">Black Link</a>
        </body>
</HTML>
```

It will produce the following black link-

The :visited pseudo class-

This pseudo-class targets links that have been visited by the user. By default the links are blue and turn purple when we’ve visited them.

```
<HTML>
        <head>
                <style type="text/CSS">
                        a:visited {color: #006600}
                </style>
        </head>
        <body>
                <a href="">Click this link</a>
        </body>
</HTML>
```

The output of this code will produce following link. Once we will click this link, it will change its color to green.

The :hover pseudo class-

This :hover pseudo class will apply CSS when the targeted element is hovered.

```
<HTML>
        <head>
                <style type="text/CSS">
                        a:hover {color: #FFCC00}
                </style>
        </head>
        <body>
                <a href="">Bring Moopt Here</a>
        </body>
</HTML>
```

Now, if we bring mouse over the link then we will see that it changes its color to yellow as set in the code above.

The :active pseudo class-

This is opted to apply CSS to active elements. This is commonly opted with <a> and <button> elements.

```
<HTML>
        <head>
                <style type="text/CSS">
                        a:active {color: #FF00CC}
                </style>
        </head>
        <body>
                <a href="">Click This Link</a>
        </body>
</HTML>
```

When a user clicks on the link, the color changes to the pink.

The :focus pseudo class-

This pseudo-class apply CSS when an HTML element is in focus.

```
<HTML>
        <head>
                <style type="text/CSS">
                        a:focus {color: #0000FF}
                </style>
```

```
    </head>
    <body>
        <a href="">Click this Link</a>
    </body>
</HTML>
```

It will produce the following link. When this link gets focused, its color changes to orange. The color changes back when it loses focus.

The :first-child pseudo-class

The :first-child CSS pseudo-class represents the first element along with a set of sibling elements. The selected element must have a parent and to make: first-child work in IE <!DOCTYPE> must be declared at the top of the document.

For example, to indent the first paragraph of all <div> elements, we could opt this definition:

```
<HTML>
    <head>
        <style type="text/CSS">
            div p:first-child
            {
                text-indent: 25px;
            }
        </style>
    </head>
    <body>
        <div>
            <p>First paragraph in div. This paragraph will be indented</p>
            <p>Second paragraph in div. This paragraph will not be indented</p>
        </div>
        <p>But it will not match the paragraph in this HTML:</p>
        <div>
            <h3>Heading</h3>
            <p>The first paragraph inside the div. This paragraph will not be effected.</p>
        </div>
    </body>
</HTML>
```

It will produce the following result −
First paragraph in div. This paragraph will be indented
Second paragraph in div. This paragraph will not be indented

But it will not match the paragraph in this HTML:
Heading
The first paragraph inside the div. This paragraph will not be effected.

The :lang pseudo-class

The language pseudo-class :lang, permits constructing selectors based on the language setting for specific tags.

This class is optful in documents that must appeal to multiple languages that have different conventions for certain language constructs. For example, the French language typically opts angle brackets (< and >) for quoting purposes, while the English language opts quote marks (' and ').

In a document that needs to address this difference, we can opt the :lang pseudo-class to change the quote marks appropriately. The following code changes the <blockquote> tag appropriately for the language being opted:

Syntax of lang pseudo class selector is-

selector : lang(ISO language code) { declaration list; }

It helps us to opt for any HTML element targeted by the selector, if it has the lang attribute value set to the specified language code. The example of optingit is given below-

```
p:lang(en){
        border: 2px dotted blue;
}
```

It selects any p element, if its lang attribute value is set to en.ISO: International Organization for Standardization. This is depicted below-

```
<HTML>
        <head>
                <style type="text/CSS">
                        :lang(en) { quotes: '"' '"'  "'"  "'"; }
                        :lang(fr) { quotes: "<<" ">>" "<" ">"; }
                </style>
        </head>
        <body>
                <p>...<q lang="fr">A quote in a paragraph</q>...</p>
        </body>
</HTML>
```

The :lang selectors will apply to all the elements in the document. However, not all elements make opt of the quotes property, so the effect will be transparent for most elements. It will produce the following result:

...<<A quote in a paragraph>>...

5.7 BORDER STYLING

The CSS border properties permit us to state the style, size, and color of an HTML element border. Following CSS border properties are opted for HTML elements-

- border-style
- border-width
- border-color
- border-radius

Border Style-

The border-style property specifies the kind of border to display. The likely values which may be given to border-style are listed below-

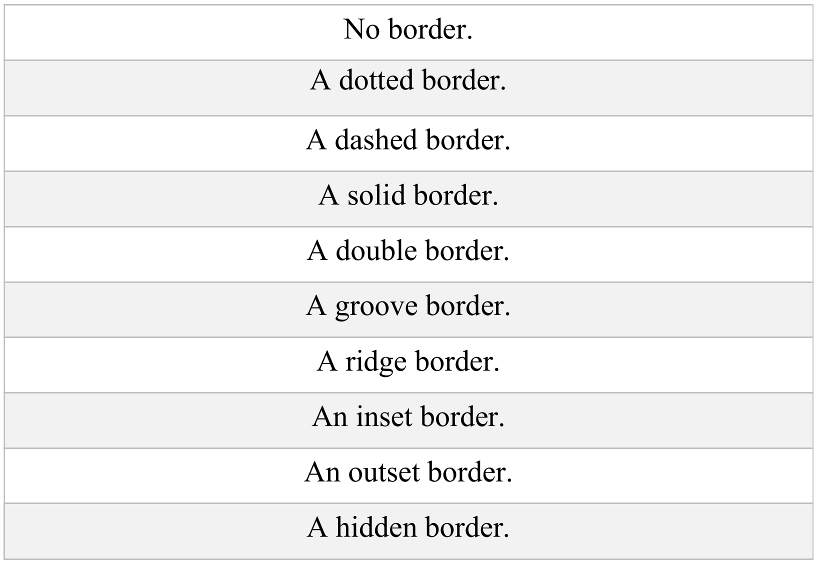
Value	Description
none	It is opted for does not distinct any border.
dotted	It is opted to distinct a dotted border.
dashed	It is opted to distinct a dashed border.
solid	It is opted to distinct a solid border.
double	It distinct two borders with the same border-width value.
groove	It distinct a 3D grooved border. Effect is generated according to border-color value.
ridge	It distinct a 3D ridged border. Effect is generated according to border-color value.
inset	It distinct a 3D inset border. Effect is generated according to border-color value.
outset	It distinct a 3D outset border. Effect is generated according to border-color value.

Below is the example in which the CSS has opted various types of styles-

```
<!DOCTYPE HTML>
<HTML>
        <head>
                <style>
                        p.none {border-style: none;}
                        p.dotted {border-style: dotted;}
                        p.dashed {border-style: dashed;}
                        p.solid {border-style: solid;}
                        p.double {border-style: double;}
                        p.groove {border-style: groove;}
                p.ridge {border-style: ridge;}
                p.inset {border-style: inset;}
                p.outset {border-style: outset;}
                p.hidden {border-style: hidden;}
            </style>
        </head>
        <body>
            <p class="none">No border.</p>
            <p class="dotted">A dotted border.</p>
            <p class="dashed">A dashed border.</p>
            <p class="solid">A solid border.</p>
            <p class="double">A double border.</p>
            <p class="groove">A groove border.</p>
            <p class="ridge">A ridge border.</p>
            <p class="inset">An inset border.</p>
            <p class="outset">An outset border.</p>
            <p class="hidden">A hidden border.</p>
        </body>
</HTML>
```

This will provide the following output. We may run the code in any browser to get better understanding.

The output is-



```
No border.
A dotted border.
A dashed border.
A solid border.
A double border.
A groove border.
A ridge border.
An inset border.
An outset border.
A hidden border.
```

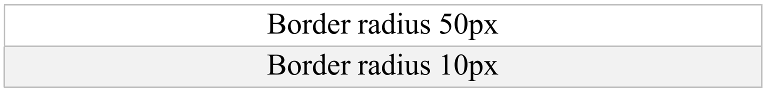
Border Width-

The border-width property is opted to set the border width. It is set in pixels. We can also set the width of the border by opting pre- distinct values, thin, medium or thick.

Example- CSS border-width

```
<!DOCTYPE HTML>
<HTML>
        <head>
                <style>
                        p.para1w {
                                border-style: solid;
                                width:200px;
                                border-color: green;
                        }
                        p.para2w {
                                border-style: solid;
                                width:100px;
                                border-color: red;
                        }
                </style>
        </head>
        <body>
                <p class="para1w">This is a solid red border</p>
                <p class="para2w">This is a solid green border</p>
        </body>
</HTML>
```

Output**-**



```
Border radius 50px
Border radius 10px
```

Border Color-

This property is opted to set the color of HTML elements border. There are three methods to set the color of border.

- Name: It specifies the color name. For example: "red".
- RGB: It specifies the RGB value of the color. For example: "rgb(255,0,0)".
- Hex: It specifies the hex value of the color. For example: "#ff0000".

Following code depicts how to opt border-color.

```
<!DOCTYPE HTML>
<HTML>
        <head>
                <style>
                        p.para1{
                                border-style: solid;
                border-color: green;
            }
            p.para2 {
                border-style: solid;
                border-color: red;
            }
        </style>
    </head>
    <body>
        <p class="para1">This is a solid red border</p>
        <p class="para2">This is a solid green border</p>
    </body>
</HTML>
```

Output for above code is-

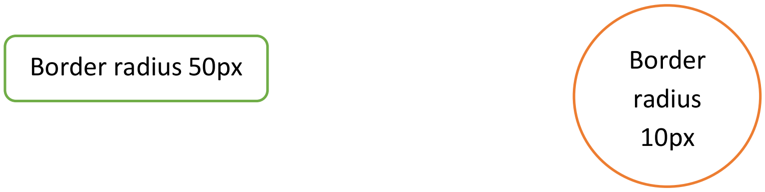
This is a solid red border
This is a solid green border

Border Radius-

Border-radius is opted to give the radius for border, opting this we can make a circle in a document. Following code depicts the opting of CSS border-radius-

```
<!DOCTYPE HTML>
<HTML>
    <head>
        <style>
            p.parar1{
                border-style: solid;
                border-radius: 100%;
                border-color: green;
            }
            p.parar2{
                border-style: solid;
                border-radius: 50%;
                height:100px;
                width:100px;
                border-color: orange;
            }
        </style>
    </head>
    <body>
        <p class="parar1">This is a solid orange border</p>
        <p class="parar2">This is a solid green border</p>
    </body>
</HTML>
```

Output of the program is given below-



Border radius 50px

Border radius 10px

5.8 MARGINS AND PADDING

Margin-

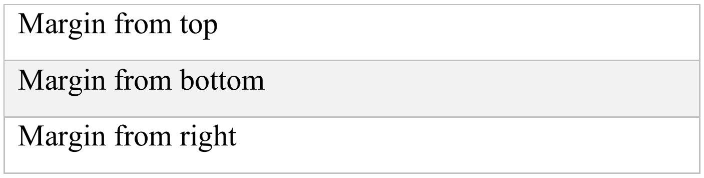
Margin is opted to give plenty space around an element. The margin does not have a background color, and it is completely transparent. Margin has four properties which are given below-

- top
- bottom
- left
- right

The following HTML illustrates how to opt margin in style-

```
<HTML>
    <body>
        <p style="margin-top:100px;"> Margin from top</p>
        <p style="margin-bottom:100px;">Margin from bottom</p>
        <p style="margin-right:50px;">Margin from right</p>
        <p style="margin-left:50px;">Margin from left</p>
    </body>
</HTML>
```

Output of the program is given below-



```
Margin from top
Margin from bottom
Margin from right
```

```
Margin from left
```

Padding-

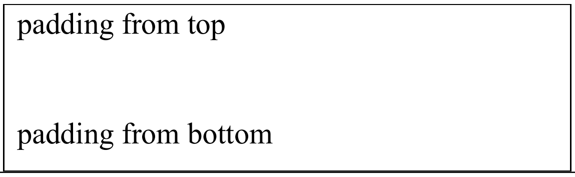
The padding clears an area near by the content (inside the border) of an element. The padding is exaggerated by the background color of the element. Padding also has four properties which are given below-

- top
- bottom
- left
- right

The following HTML code illustrates how to opt padding-

```
<HTML>
        <body>
                <div style="border:1px solid black;">
                <p style="padding-top:100px;">padding from top</p>
                <p style="padding-bottom:100px;">padding from bottom</p>
                <div>
                        <p style="padding-right:50px;">padding from right</p>
                        <p style="padding-left:50px;">padding from left</p>
                </body>
</HTML>
```

Output of the program is given below:



```
padding from top


padding from bottom
```

5.9 POINTS TO REMEMBER

- CSS describes how HTML elements are to be displayed on screen.
- CSS can specify fonts other than the default for the browser.
- CSS can specify color and size of text and links in a web page.
- CSS can apply colours to backgrounds
- CSS saves a lot of work; it also can control the layout of multiple web pages all at once.

5.10 GLOSSARY

- attribute- A value associated with an element, consisting of a name, and an associated (textual) value.
- Style sheet- A set of statements that specify presentation of a document. The interaction of these sources is described in the section on cascading and inheritance.

- Universal selector- The universal selector (*) may be used to select all the elements in a particular range. Be aware that the universal selector is the most performance taxing selector, and should be used in restricted manner. Example-

```
*{
Background-color: blue;
}
```

5.11 CHECK YOUR PROGRESS

Descriptive type questions-

a) What is the use of CSS? Also, list find the difference between CSS and HTML?

b) Explain the three main ways to apply CSS styles to a web page.

c) Define different kinds of selectors used in CSS.

d) What are pseudo classes and what are they used for?

e) What are attributes and how are they used?

f) How do margin, border and padding fit together into a web page?

g) Describe about class selector in CSS?

h) Explain external Style Sheet? How would you link to it in your web page? give example.

5.12 BIBLIOGRAPHY/REFERENCES

- https://www.austincc.edu/hr/profdev/eworkshops/docs/CSS_intro.pdf, November, 2017
- http://cglab.ca/~morin/teaching/2405/notes/css.pdf, November, 2017
- https://www.w3.org/TR/CSS2/css2.pdf, December, 2017
- http://www.inf.ufsc.br/~bosco.sobral/downloads/Livro-Java-Como-Programar-Deitel-Ed6/additional/addnlRefs/CSS/CSS.pdf, December, 2017
- https://www.w3schools.com/css/css_intro.asp, November, 2017

5.13 SUGGESTED READINGS

- Powell. Thomas A., HTML & CSS: The Complete Reference, 5E
- Holzner. Steven, PHP: The Complete Reference.
- Powell. Thomas A., JavaScript: The Complete Reference.

UNIT 6
INTRODUCTION TO CSS 3

6.1 INTRODUCTION

The CSS means Cascading Style Sheets. It indicates that the HTML elements displayed on screen, paper, or in other type of media. It saves a lot of work. By using Cascading Style Sheets programmer can control the layout of many web pages all at once. These CSS are stored in the file of .css extensions file.

Cascading Style Sheet is opted to define styles of the web pages. It also defines the web page design, layout and variations in display for different types of the devices and their screen size. This way CSS solved a big problem. However, HTML is not more focused on formatting a web page and it does not contain more formatting tags.

6.2 OBJECTIVES

After successful completion of this unit, you will be able to-

- Differentiate CSS and CSS3.
- Define CSS3 and implement in your web page.
- Use two dimensional transformations in your web page.
- Use CSS3 animation.

6.3 CSS BASICS

The HTML was developed to **describe the content** of the web pages. As-

<h1>This is first heading</h1>

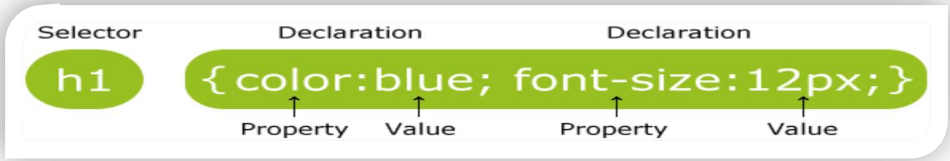
<p>This is first paragraph. </p>

Primarily the number of tags is available in HTML. Lately tags like <font>, color etc. attributes were added in the HTML version. It was a giant progress for web designers. The Font and color tags were typically added to every single page for the Development of bulky websites but it was very long, expensive and boring process. Therefore to solve this problem, the World Wide Web Consortium (W3C) has created Cascading Style Sheets.

This way the CSS is opted for the style formatting and HTML is no longer opted for style formatting of any web page and CSS Saves a Lot of Works in recent time. Normally all the style settings are saved in external .css files. By using that file, we can change the formatting and look of an entire site by changing just one file.

CSS Syntax-

Following is the syntax of Cascading Style Sheets. It consists of a selector and a declaration



block-

Fig 1: Syntax of Cascading Style Sheets

The selector is a HTML element which we want to style. In the declaration block there are many declarations statement which are separated by semicolons. Every declaration has property name and a value, and which is separated by a full colon. Each CSS declaration ends by semicolon, and the whole declaration blocks are surrounded by “---”.

As the following example where all <p> elements will be left-aligned, with a blue text color:

Example-

```
p{
        color: blue;
```

```
    text-align: left;
}
```

CSS Selectors-

Cascading Style Sheets selectors are opted to "find" HTML elements on the source of their element name, id, class, attribute, and many more.

The element Selector

The element selector selects elements on the bases element name. In the following example all <p> elements in a web page look like-

<p> elements will be left-aligned, with a blue text color:

Example-

```
p{
    color: blue;
    text-align: left;
}
```

The id Selector-

For selecting a particular element the, id selector opted the id attribute of the HTML element. This id of the element should be exclusive in entire page. The id selector is opted to select one exclusive element. For selecting an element with a specific id, user should write a hash (#) character, followed by the id of the element.

The style rule as per the following will be applied to the HTML element where id="para1"(id could not start with a number):

Example-

```
#para1{
    text-align: right;
    color: green;
}
```

The class Selector

The class selector also selects elements with a specific class attribute. We write or start with a period (.) when we select elements with a specific class, character, and then followed by the name of the class.

In the following example, all HTML elements are having class="center", colored with red and center-aligned:

Example-

```
.center{
    text-align: center;
    color: red;
}
```

Hear we can also specify that only particular HTML elements should be affected by a class. In the following example <p> is element and class name is "center" which is center-aligned:

Example-

```
p.center{
    text-align: center;
    color: red;
}
```

With HTML elements we can also define more than one class at a time. Class name should not start with number. As in the following example, the element <p> will be styled according to class="center" and to class="large" both:

Example-

```
<p class="center large">
    In this paragraph we have mentioned two classes at a time.
</p>
```

Grouping Selectors

Consider the following code where different elements with the same style definitions, as:

```
h1{
    text-align: center;
    color: red;
}

h2{
    text-align: center;
    color: red;
}
p{
    text-align: center;
    color: red;
}
```

The above code is too long. It is better to group all the selectors, so that the code could be minimized. To group selectors, separate each selector with a comma. In the following example we have grouped the selectors so that the above code could be minimized:

Example-

```
h1,h2,p{
    text-align: center;
```

```
        color: red;
}
```

CSS Comments-

We can use the comments to explain the code. It will help we at the time when we edit the source code at a later date. The Comments are always ignored by the browsers. The CSS comments are starts with /* and ends with */. We can span comments in multiple lines:

Example-

```
p{
  color: red;  /*single-linecomment*/
  text-align: center;
}/*multi-line comment */
```

6.4 TYPES OF IMPLEMENTATIONS OF CSS INTO A WEB PAGE

A browser reads a cascading style sheet, and then it formats the HTML document as per the declaration specified in the style sheet. There are three ways to insert a style sheet into a web page-

- External style sheet
- Internal style sheet
- Inline style
- External Style Sheet-

External style sheet is an external file with .css file extension. When we make changes in this file, then we can modify the appearance of an entire website. Every page of website has to include a reference of the external style sheet within the <link> element. The <link> element is specified in the <head> section. As-

Example-

```
<head>
        <link rel="stylesheet" type="text/css" href="style_exp.css">
</head>
```

We can opt any text editor to create an external style sheet. This type of file is not required any html tags. The CSS file must be saved with a .css extension.

Example of "style_exp.css":

```
body{
        background-color: lightgreen;
}
h1{
        color: green;
        margin-left: 30px;
}
```

In the above code, we have mentioned that there should not be any space left blank between the property value and the unit (e.g. margin-left: 20px; is correct, while margin-left: 20 px; is incorrect)

Internal Style Sheet-

If a web developer has a single page than the internal style sheet may be opted. It is defined within the <style> element, of the <head> section of an HTML page-

Example-

```
<head>
        <style>
                body{
                        background-color: red;
                }

                h1{
                        color: green;
                        margin-left: 40px;
                }
        </style>
</head>
```

Inline Styles-

An inline style may be opted to apply a exclusive style for a single element. To use inline styles, we added the style attribute to the relevant element. The style attribute can contain any CSS property. Following example shows the change of the color and the left margin of a <h1> element:

Example-

```
<h1 style="color: red; margin-left:30px;">It’s heading</h1>
```

However the inline style has many drawbacks (where we are mixing content with presentation). This method should not be opted frequently.

Managing Multiple Style Sheets-

If any properties have been defined for the same selector/element in many style sheets, the value from the last read style sheet will be opted.

Example-

Suppose that an external style sheet has the style below for the <h1> element-

```
h1{
        color: navy;
}
```

Further, suppose that an internal style sheet also has the following style also for the <h1> element-

```
h1{
        color: orange;
}
```

Here the internal style is defined after the external style sheet, the <h1> elements will be "orange" it means that the value from the last read style sheet will be opted.

Example-

```
<head>
        <link rel="stylesheet" type="text/css" href=”style_exp.css">
        <style>
                h1{
                        color: orange;
                }
        </style>
</head>
```

Its opposite when the internal style is defined before the external style sheet, the <h1> elements will be "navy".

Example-

```
<head>
        <style>
                h1{
                        color: orange;
                }
        </style>
        <link rel="stylesheet" type="text/css" href="style_exp.css">
</head>
```

Cascading Order-

In HTML elements we can opt more than one style in a sheet. They can be opted by “Cascading” a new style sheet called “Virtual Style Sheet”. The rule is followed to provide the highest priority.

- Inline style
- External/Internal style sheets
- Browser default

In this the inline style which is inside the HTML elements has the top precedence; it means that this will override the style mentioned inside the tag<head> which is internal style sheet, or in an external style sheet, or a browser default style value.

6.5 BASICS OF CSS 3

In the upbringings of the CSS, CSS3 appears. It contains the new features of CSS and the previous also. The new style sheets called CSS3 have numerous advance features and properties like rounded corners, shadows, gradients, transitions or animations, also new layouts like multi-columns, flexible box or grid layouts. It’s completely backwards-compatible with previous CSS versions. The new CSS3 properties are implemented in modern browsers.

CSS3 Modules-

CSS3 has special "modules", which also have the old CSS specifications. They have been split into smaller pieces. Some new modules are added in it. Following are the some frequently used CSS3 modules. As-

- Selectors
- Box Model
- Backgrounds and Borders
- Image Values and Replaced Content
- Text Effects
- 2D/3D Transformations
- Animations
- Multiple Column Layout
- User Interface

Difference between CSS and CSS3-

- CSS3 is an enhanced version of CSS2. It maintains compatibility with all of the previous CSS's features. It doesn't censure any of the CSS code. The CSS3 code was opted to develop a Web page with the feature of better appearance and faster loading. It also reduces the development time to prepare web pages in a user's browser.
- CSS3 also makes Web design less contingent on image files for page blueprint elements. It reduces the file transfer requests and the download time by using less number of images.
- The CSS1 mainly focused on appearance formatting.
- The CSS2 further positioning capabilities for text and objects.
- CSS3's and its added features are not supported by the older browser e.g. IE 9 or prior version.

Include rounded corners & gradients using CSS3-

Earlier Web developers desired to design image files to add things like rounded corners for structural borders and background image gradients. But now CSS3 has same feature with a few lines of code. In CSS, a developer has to design/create a border or gradient, upload it into the image server, place the same image on the web page and use Cascading Style Sheet to correctly position the border. In CSS3, the developer can do same with the following code as-

Example-

".roundBorder {border-radius:10px;}".

More over the Background gradients are slight bit complicated and they involve different code for different browsers. E.g. a white-to-black gradient for Chrome and IE opted the code as: ".gradBG {background: liner-gradient (white, black);}".

Include animation and text effects using CSS3-

CSS3 has number of advance features to improve the web page like developers can include a text-shadow to the text also to compose it easy to read add visual stylishness. It can force the line breaks within the longer words to make them fit inside the columns with the feature of using word wrap.

Before CSS3, Web developers opted animation code in scripting languages like JavaScript, vb Script and jQuery. CSS3 provides many animation features and they can be implemented in the design layer. The W3C is continuously working for Web animation standards.W3C is also working on compatibility issues between the different types of animation techniques.

Example- (Animation)

```
<!DOCTYPE html>
<html>
        <head>
                <style>
                        div{
                                width: 100px;
                                height: 100px;
                                background-color: red;
                                -webkit-animation-name: example;
                                -webkit-animation-duration: 4s;
                                animation-name: example;
                                animation-duration: 4s;
                        }
                        /* Safari 4.0 - 8.0 */
                        @-webkit-keyframes example{
                                from {background-color: red;}
                                to{background-color: yellow;}
                        }
                        /* Standard syntax */
                        @keyframes example{
                                from {background-color: red;}
                                to{background-color: yellow;}
```

```
                    }
                </style>
        </head>
        <body>
                <p><b>Note:</b> This example does not work in Internet Explorer 9 and earlier versions.</p>
                <div></div>
                <p><b>Note:</b> When an animation is finished, it changes back to its original style.</p>
        </body>
</html>
```

CSS3 Defines Text Columns-

CSS3 also has the potential to split text sections into multiple columns like a newspaper. However in the CSS2 it’s tricky to construct multi-column text sections and it does not involuntarily split text. This feature of CSS3 is beneficial especially with the sites whose feature is responsive design. A development technique rearranges and resizes content based on the viewing device's screen.

CSS3 border-radius Property-

Using CSS3, we can give "rounded corners" for any element, using the property `border-radius`. Following are three examples of this property-

Example-

```
#rcorners1{
        border-radius: 30px;
        background: #73AD21;
        padding: 40px;
        width: 400px;
        height: 200px;
}
#rcorners2{
        border-radius: 30px;
        border: 4pxsolid#73AD21;
        padding: 40px;
        width: 400px;
        height: 200px;
}
#rcorners3{
        border-radius: 30px;
        background: url(cloud.gif);
        background-position: lefttop;
        background-repeat: repeat;
        padding: 40px;
        width: 400px;
        height: 200px;
}
```

CSS3 Gradients-

The CSS3 gradients allow us to exhibit smooth transitions between two or more precise colors. Previously we had to use images for these effects. However, by using CSS3 gradients developer can reduce the download time and neglect the bandwidth usage. The elements with gradients look better when zoomed.

There are two types of gradients of CSS3-

- **Linear Gradients (it goes down/up/left/right/diagonally)**
- **Radial Gradients (it is defined by their center)**

We have to define at least two-color stops for creating a linear gradient. Hear the Color stops are the colors which render smooth transitions among. Beside with the gradient effect, we can also set a starting point and a direction (angle). Following is the syntax of Linear Gradient-

Syntax-

background: linear-gradient (*direction, color-stop1, color-stop2, ...);*

Linear Gradient - Top to Bottom (default)

Following is the example of linear gradient which starts from the top. It starts from green, transitioning to yellow-

Example-

```
#grad{
        background: green; /* This is for the browsers that do not support gradients */
        background: -webkit-linear-gradient(green, yellow); /* For browser Safari 5.1 to 6.0*/
        background: -o-linear-gradient(green, yellow); /* For browser Opera 11.1 to 12.0 */
        background: -moz-linear-gradient(green, yellow); /* For browser Firefox 3.6 to 15*/
        background: linear-gradient(green, yellow);  /*Standard syntax */
}
```

Linear Gradient - Left to Right

Following is the example where a linear gradient starts from the left. Hear it starts from green, and transitioning to yellow-

Example-

```
#grad{
        background: green; /* For all browsers that do not support gradients */
        background: -webkit-linear-gradient(left, green , yellow); /* For browser Safari 5.1 to
6.0*/
        background: -o-linear-gradient(right, green, yellow); /* For browser Opera 11.1 to
12.0 */
        background: -moz-linear-gradient(right, green, yellow); /* For browser Firefox 3.6 to
15 */
        background: linear-gradient(to right, green , yellow); /* Standard syntax */
}
```

Linear Gradient - Diagonal

We have prepared a gradient diagonally by mentioning both the vertical and horizontal starting positions. In the following example a linear gradient which starts from top left (and then goes to bottom right). It starts from green, transitioning to yellow-

Example-

```
#grad{
        background: green; /* For browsers that do not support gradients */
        background: -webkit-linear-gradient(left top, green, yellow); /* For browsers Safari
5.1to6.0*/
        background: -o-linear-gradient(bottom right, green, yellow);        /* For browsers
Opera 11.1to12.0*/
        background: -moz-linear-gradient(bottom right, green, yellow);      /* For browsers
Firefox3.6to15*/
        background: linear-gradient(to bottom right, green, yellow); /* Standard syntax */
}
```

Text-shadow Property of CSS3-

Example- basic text-shadow Property of CSS3

```
h1{
        text-shadow: 2px2px#ff0000;
}
```

Definition and Usage of the text-shadow property

This property adds shadow to the text. Text-shadow property includes a comma-separated list of shadows which would be applied to the text. Its default value is none. It is animatable and inherited. Its version is CSS3. Its JavaScript syntax is as follows-

JavaScript syntax:	object.style.textShadow="2px 5px 5px red"

Following is the CSS Syntax-

text-shadow: *h-shadow v-shadow blur-radius color*/none/initial/inherit;

If we are adding many shadows to the text, add them as comma-separated list of shadows. Following are the Property Values-

Value	Description

h-shadow	Fundamentally it's mandatory. It's opted for the position of the horizontal shadow. In this we can also opted negative values.
v-shadow	Fundamentally it's mandatory. It's opted for the position of the vertical shadow. In this we can also opted negative values.
blur-radius	The blur radius is Optional and the default value of it's 0
Color	It provides the color to the shadow. It's optional.
None	It's by default value that means no shadow.
Initial	This property sets to its default value.
Inherit	It inherits the all properties from its parent element.

Example**-**

The following example displays a text-shadow with a blur effect-

```
h1{
        text-shadow: 2px2px8px#FF0000;
}
```

The following example display text-shadow on a white text:

```
h2{
        color: white;
        text-shadow: 2px2px4px#000000;
}
```

Example-

Following example display text-shadow showing with a red glow-

```
h2{
        text-shadow: 003px#FF0000;
}
```

Example-

Following example display text-shadow showing with a red and blue glow-

```
h2{
        text-shadow: 003px#FF0000,005px#0000FF;
}
```

6.6 CSS3- TWO DIMENSIONAL TRANSFORMATIONS

CSS3 opted transforms to translate, rotate, scale, and skew the elements. A transformation is a procedure where an element changes shape, size and position. CSS3 helps in 2D and 3D transformations. Following are the method which is opted for 2D transformations-

- translate()
- rotate()
- scale()
- skewX()
- skewY()
- matrix()

Translate () Method-

The method translate () in CSS3 moves an element from its existing position to specified position. The element moves as per the parameters specified for the X-axis and the Y-axis. In the following example here the <div> element moves 25 pixels to the right and 50 pixels down from its existing position-

Example-

```
div{
        -ms-transform: translate(25px, 50px); /* statement supported by IE 9 Browser */
        -webkit-transform: translate(25px, 50px); /* statement supported by Safari Browser*/
        transform: translate(25px,50px);
}
```

Rotate() Method-

The rotate() method rotates an element, according to a specified degree in clockwise or counter-clockwise. Here the following <div> element rotates clockwise with 30 degrees-

Example-

```
div{
        -ms-transform: rotate(30deg);           /* statement supported by IE 9 Browser */
        -webkit-transform: rotate(30deg);       /* statement supported by Safari Browser*/
        transform: rotate(30deg);
}
```

When we will use negative values it rotate the element counter-clockwise. The <div> element in the following example will rotates counter-clockwise with 30 degrees-

Example-

```
div{
        -ms-transform: rotate(-30deg);           /* statement supported by IE 9 Browser */
        -webkit-transform: rotate(-30deg);      /* statement supported by Safari Browser*/
        transform: rotate(-30deg);
}
```

Scale() Method-

The method scale() increases or decreases the size of an element. The size of element is changed as per the specified parameters of width and height. In the following example the element <div> increases three times of its original width, and two times of its original height:

Example-

```
div{
        -ms-transform: scale(3, 2);     /* statement supported by IE 9 Browser */
        -webkit-transform: scale(3, 2); /* statement supported by Safari Browser*/
        transform: scale(3,2);
}
```

The <div> element decreases to the half of its original width and height in the following example-

```
div{
        -ms-transform: scale(0.5, 0.5);         /* statement supported by IE 9 Browser */
        -webkit-transform: scale(0.5, 0.5);     /* statement supported by Safari Browser*/
        transform: scale(0.5,0.5);
}
```

SkewX() Method-

The method skewX() skews an element beside the X-axis as per the specified angle. The example below skews the <div> element by 30 degrees beside the X-axis-

Example-

```
div{
        -ms-transform: skewX(30deg);            /* statement supported by IE 9 Browser */
        -webkit-transform: skewX(30deg);        /* statement supported by Safari Browser*/
        transform: skewX(30deg);
}
```

SkewY() Method-

The method skewY() skews an element beside the Y-axis as per the specified angle. The example specified below skews the <div> element 30 degrees beside the Y-axis-

Example-

```
div{
        -ms-transform: skewY(30deg);            /* statement supported by IE 9 Browser */
        -webkit-transform: skewY(30deg);        /* statement supported by Safari Browser*/
        transform: skewY(30deg);
}
```

Skew() Method-

The method skew() skews the element beside the X and Y-axis as per the specified angles. The example below skews the <div> element by 30 degrees in X-axis, and 20 degrees in Y-axis-

Example-

```
div{
        -ms-transform: skew(30deg, 20deg); /* statement supported by IE 9 Browser */
        -webkit-transform: skew(30deg, 20deg); /* statement supported by Safari Browser*/
        transform: skew(30deg,20deg);
}
```

If we didn’t provide the second parameter than its value is zero by default. The example below skews the <div> element 20 degrees in the X-axis-

Example-

```
div{
        -ms-transform: skew(20deg); /* statement supported by IE 9 Browser */
        -webkit-transform: skew(20deg); /* statement supported by Safari Browser*/
        transform: skew(20deg);
}
```

Matrix() Method-

The method matrix() concatenates all the 2D transform methods under one method. The method takes 6 parameters, which contains mathematical functions, and consent us to rotate, scale, move (translate), and skew elements. Following are the parameters of matrix function-

matrix(scaleX(),skewY(),skewX(),scaleY(),translateX(),translateY())

Example-

```
div{
        -ms-transform: matrix(0, -0.2, 0, 1, 0, 1); /* statement supported by IE 9 Browser */
        -webkit-transform: matrix(0, -0.2, 0, 1, 0, 1); /* statement supported by Safari
Browser*/
        transform: matrix(0,-0.2,0,1,0,1);
}
```

Transform Properties of CSS3-

In this table we listed diverse 2D properties of transform where transform and transform-origin are 2 different properties which is described-

Property	Description
transform	It's opted for the transformation upon an element in a 2D or 3D style.
transform-origin	It's opted to modify the position upon the transformed elements

6.7 CSS3- THREE DIMENSIONAL TRANSFORMATIONS

In CSS3 We can format the elements using 3D transformations. Hear we will learn the following methods of 3D transformation also we will learn about their properties.

rotateX()

rotateY()

rotateZ()

RotateX() Method-

The method rotateX() rotates an element around its X-axis as per the specified degree-

Example-

```
div{
        -webkit-transform: rotateX(90deg); /* statement supported by Safari Browser*/
        transform: rotateX(90deg);
}
```

RotateY() Method-

The method rotateY() rotates an element around its Y-axis as per the specified degree.

Example-

```
div{
        -webkit-transform: rotateY(120deg); /* statement supported by Safari Browser*/
        transform: rotateY(120deg);
}
```

RotateZ() Method-

The method rotateZ() rotates an element around its Z-axis at a 90 degree as specified in the following example-

Transform Properties and there description of CSS3	
Property	Description
transform	It's opted for 2D or 3D transformation to an element
transform-origin	It's opted for changing the position on transformed elements
transform-style	It's opted for, how nested elements are rendered in 3D space
perspective	It's opted for the perspective on how 3D elements are viewed
perspective-origin	It's opted for the bottom position of 3D elements
backface-visibility	It's opted for, whether an element should be visible or not when not facing the screen.

```
div{
        -webkit-transform: rotateZ(90deg); /* statement supported by Safari Browser*/
        transform: rotateZ(90deg);
}
```

3D Transform Methods	
Function	Description of Function
matrix3d (n,n,n,n,n,n,n,n,n,n,n,n,n,n,n,n)	It's is opted to describe a 3D transformation, using a 4x4 matrix of 16 values
translate3d(x,y,z)	It's is opted to defines a 3D translation
translateX(x)	It's is opted to defines a 3D translation, using the value for the axis-X only
translateY(y)	It's is opted to defines a 3D translation, using the value for the axis-Y only.
translateZ(z)	It's is opted to defines a 3D translation, using the value for the axis-Z only.
scale3d(x,y,z)	It's is opted to defines a 3D scale transformation

scaleX(x)	It’s is opted to defines a 3D scale transformation by giving a value for the axis-X
scaleY(y)	It’s is opted to defines 3D scale transformation by giving a value for the axis-Y
scaleZ(z)	It’s is opted to defines 3D scale transformation by giving a value for the axis-Z
rotate3d(x,y,z,angle)	It’s is opted to define 3D rotation
rotateX(angle)	It’s is opted to define 3D rotation in the axis-X
rotateY(angle)	It’s is opted to define 3D rotation in the axis-Y
rotateZ(angle)	It’s is opted to define 3D rotation in the axis-Z
perspective(n)	It’s is opted to define perspective view for a 3D transformed element

6.8 CSS3 TRANSITIONS

It allows us to change one property values with another property values very smoothly. It happens over a specified duration. For transition effect, we have to consider following two points. As- (i) The CSS property that we want to add an effect to. (ii) The duration of the effect.

If duration part is not mentioned, the transition will have no effect, because it’s by default value is 0. The example below shows a 50px * 50px blue <div> element. The <div> element has also mentioned in a transition effect for the width property, with duration of 3 seconds:

Example-

```
div{
        width: 50px;
        height: 50px;
        background: blue;
        -webkit-transition: width3s; /* statement supported by Safari Browser*/
        transition: width3s;
}
```

The transition effect will start when the specified or mentioned width CSS property changes its value. In the following example a new value 200 specified as width property when user mouses over the element <div> element:

Example-

```
div:hover{
        width: 200px;
}
```

When cursor moves out of the element, it will steadily come back to its original style. The example below adds a transition effect for the width and height both property, in duration of 3 seconds for the width & 5 seconds for the height-

Example-

```
div{
        -webkit-transition: width 3s, height 5s; /* statement supported by Safari Browser*/
transition: width3s,height5s;
}
```

To Delay the Transition Effect-

It is mentioned a delay in seconds for the transition effect. The example below has a 2 second delay before starting.

Example-

```
div{
        -webkit-transition-delay: 2s; /* statement supported by Safari Browser*/
transition-delay: 2s;
}
```

Transition and Transformation-

The example below combines or adds a transformation to the transition effect.

Example-

```
div{
 -webkit-transition: width 2s, height 2s, -webkit-transform 2s; /* statement supported by
Safari   Browser*/
transition: width2s,    height2s,        transform2s;
}
```

Following are the some examples of Transition and the property of CSS3 transition can be specified one by one, as-

Example-

```
div{
        transition-property: width;
        transition-duration: 5s;
        transition-timing-function: linear;
        transition-delay: 2s;
}
```

We can also use following shorthand property too for transition-

Example

```
div{
        transition: width5slinear2s;
}
```

Properties of CSS3 Transition-

The table below displays different properties of transition- As.

Property	Description
Transition	It's a shorthand property for setting the four transition properties by one single property
transition-delay	It's specifies a delay in seconds for the transition effect
transition-duration	It's specifies that the transition effect takes how many seconds or milliseconds to complete.
transition-property	It's specifies the name of the CSS property for the transition effect.
transition-timing-function	It's specifies speed curve of the transition effect

6.9 CSS3 ANIMATION

CSS3 allows us to opting animations upon the HTML elements. Here the JavaScript /Flash is not opted for the animation. Animation means to gradually change one style of element to another style. There are many CSS properties can be changed in CSS3.A user can change the property many times as he crave for. User must specify some key frames for the animation. These Key frames hold the styles of the element of specific times.

The rule of @keyframes-

If a user mention styles of CSS inside the @keyframes rule then the animation will progressively change from the existing style to the new style in the interval of fix times. We should have to combine the animation with an element. The example below combines the animation "example" with the element <div>. The animation will last for 6 seconds, and it will progressively convert the background-color of the element <div> from "yellow" to "blue":

Example-

```
        @keyframesexample{
        from {background-color: yellow;}
        to {background-color: blue;}
}
/* Following are the element whom animation is to be applied */
div{
        width: 200px;
        height: 200px;
        background-color: yellow;
        animation-name: example;
        animation-duration: 6s;
}
```

In above example style will be changed using the keywords "from" and "to".

Delay an Animation-

In the below example animation-delay property indicates a delay for the start of an animation. It has a 4 seconds delayed before starting the animation.

Example-

```
div{
        width: 200px;
        height: 200px;
        position: relative;
        background-color: red;
        animation-name: example;
        animation-duration: 6s;
        animation-delay: 4s;
}
```

The animation-iteration-count property-

The animation iteration count property fix the number of times an animation should be run. The example below will execute a program to play animation 4 times before it stops.

Example-

```
div{
        width: 200px;
        height: 200px;
        position: relative;
        background-color: green;
        animation-name: example1;
        animation-duration: 4s;
        animation-iteration-count: 4;
}
```

However, the below example will execute a program to play animation "infinite” times continue forever.

Example-

```
div{
        width: 200px;
        height: 200px;
        position: relative;
        background-color: green;
        animation-name: example2;
        animation-duration: 4s;
        animation-iteration-count: infinite;
}
```

To Run any Animation in Reverse Direction/Alternate Cycles-

It runs an animation either in reverse direction or alternate cycles. The example below will run animation into reverse direction.

Example-

```
div{
        width: 200px;
        height: 200px;
        position: relative;
        background-color: green;
        animation-name: example3;
        animation-duration: 4s;
        animation-iteration-count: 4;
        animation-direction: reverse;
}
```

The example below, where we are using "alternate" value, it runs the animation first forward, then backward, then forward and so on.

Example-

```
div{
        width: 200px;
        height: 200px;
        position: relative;
        background-color: green;
        animation-name: example4;
        animation-duration: 4s;
        animation-iteration-count: 4;
        animation-direction: alternate;
}
```

Shorthand Property of Animation-

In the following example there are six animation properties and we can display them by Shorthand Property.

Example-

```
div{
        animation-name: example5;
        animation-duration: 4s;
        animation-timing-function: linear;
        animation-delay: 3s;
        animation-iteration-count: infinite;
        animation-direction: alternate;
}
```

This, above animation effect is achieved by the following code using shorthand property of animation.

Example-

```
div{
        animation: example4slinear3sinfinitealternate;
}
```

CSS3 Animation Properties-

Property	Description
@keyframes	It Specifies the code of animation
Animation	It is a shorthand property for setting all the animation.
animation-delay	It means a delay to start of an animation
animation-direction	It means whether any animation would play in reverse direction or alternate cycles
animation-duration	It means the seconds or milliseconds an animation takes to complete one cycle
animation-fill-mode	It means style for the element in case the animation is not playing
animation-iteration-count	It means number of times any animation would be played
animation-name	It means name of the @keyframes animation
animation-play-state	It means weather animation is running or paused
animation-timing-function	It means the animation speed curve

6.10 POINTS TO REMEMBER

- CSS3 is compatible With Older Versions of CSS.
- CSS3 is made of small modules which makes the application easier and simpler to use.
- CSS3 is compatible with all kinds of platforms and can be viewed with similar ease in mobile as well as PCs and tablets.
- CSS3 loads a lot faster than its previous versions also well compatible with all the browsers available.

- CSS3 individual modules also help in saving time during the development, implementation and end of production.
- CSS3 is platform independent and cross Browser compatible.

6.11 GLOSSARY

- CSS3 modules- CSS3 is collaboration of CSS2 specifications and new specifications, we can call this collaboration is module.
- Gradients- Gradients display the combination of two or more colors.
- Shadow- CSS3 supports to add shadow to text or elements.
- Web fonts- Web fonts are used to allow the fonts in CSS, which are not installed on local system.
- CSS3 Animation- Animation is process of making shape changes and creating motions with elements.

6.12 CHECK YOUR PROGRESS

Descriptive type questions-

a) What are the limitations of CSS?
b) What are the advantages of CSS?
c) Discuss merits of CSS3.
d) How many ways through which you can integrate CSS in your web page?
e) What is the CSS3 animation? Explain.
f) What are the possible values of the “Position” attributes?
g) What are CSS3 Transitions?

6.13 BIBLIOGRAPHY/REFERENCES

- http://www.fba.org.au/wordpress/wp-content/uploads/2013/04/smashing-ebook-19-mastering-css3.pdf, January, 2018
- https://www.austincc.edu/hr/profdev/eworkshops/docs/CSS_intro.pdf, January, 2018
- lemay. Laura, rafe colburn, jennifer kyrnin, Mastering HTML, CSS & JavaScript Web, BPB Publication, 2016
- vishvajeet. Sisodia, Basic of Web Design, HTML, CSS3, Centrum Press, 2014.
- https://mitseu.files.wordpress.com/2014/08/html5-and-css3-all-in-one-for-dummies-3rd-edition-andy-harris.pdf, January, 2018

6.14 SUGGESTED READINGS

- lemay. Laura, rafe colburn, jennifer kyrnin, Mastering HTML, CSS & JavaScript Web, BPB Publication, 2016
- vishvajeet. Sisodia, Basic of Web Design, HTML, CSS3, Centrum Press, 2014.
- Powell. Thomas A., HTML & CSS: The Complete Reference, 5E.
  Holzner. Steven, PHP: The Complete Reference.

UNIT 7

INTRODUCTION TO HTML 5

7.1 INTRODUCTION

HTML5 is an advance version of HTML. This new HTML5 standard is better than HTML 4.01, XHTML 1.0, and XHTML 1.1. We can present and structure the content on the World Wide Web (WWW) through HTML5 standard. It is a previous version HTML 4, and it was modified & standardized in 1997. On 28 October 2014, it was released.

The World Wide Web Consortium (W3C) and the Web Hypertext Application Technology Working Group (WHATWG) has developed HTML5. The World Wide Web Consortium (W3C) is an international community. It consists of full-time staff, Member organizations and the public. They work together to develop international Web standards. This fifth revised version of the HTML is the complete and final standard of the World Wide Web Consortium (W3C). The new standard of HTML5 includes following features-

- video playbacks
- audio playback
- Drag-and-drop which is previously dependent on third-party browser plug-ins (such as Adobe Flash, Microsoft Silverlight, and Google Gears).
- Generally, all latest browsers support HTML5. Most of the HTML5 features and functionality is used by the latest version of following Browsers- Apple Safari, Google Chrome, Mozilla Firefox, Internet Explorer 9.0, Opera, and etc.

As well as the latest mobile web browsers support the features and functionality of HTML5. These HTML5 supported mobile web browsers are excellent and which are pre-installed on iPads, iPhones, and Android phones. HTML5 initiated many new elements and attributes. These features are very useful to develop a modern responsive website.

7.2 OBJECTIVES

After successful completion of this unit, you will be able to-

- Use HTML5 and CSS3 to build responsive websites that render well on a variety of devices.
- Work with images.
- Work with forms and the HTML5 validation features
- Add audio and video to your pages
- Format web pages for printing
- Use the CSS3 features for transitions, animation, transform, and filter in collaboration with HTML5.

7.3 IMPORTANCE AND ADVANTAGES OF HTML5

Mutuality-

As per the modern-day need, the websites should be more interactive. So, the programmer needs to develop highly interactive web page. So, the latest Web pages needs to include- fluid animations, stream video, play music and Social Network sites like Facebook and Twitter into the websites.

Presently above features are integrated with the help of the web tools like Flash or Silverlight, Flex or java script. But these tools are very complex to develop a web page and they also increase the complexity of web page. With help of HTML5, we can embed-

- Video and audio
- High-quality drawings
- Charts
- Animation
- Many rich contents without using any plug-ins or third-party programs

Cleaner markup / Improved Code-

The code used in HTML 5 is very neat and clean.

Improved Semantics-

We can increase more the semantic value of web page by using HTML5 elements because its codes are very standard.

Elegant forms-

The form developed in HTML5 is very user friendly and their graphics is very improved in design. It enhances user interface. It provides different type of text inputs, search and different fields for different purpose.

Consistency-

The websites based on the new HTML5 has more consistency as compare to normal web site. For web programmer and developer, it is very easier to easily and quickly understand how a web page is structured.

Improved Accessibility-

Using HTML5 different technologies can enhance their features and the page designed by it, has improved understanding and look.

Fulfill the need of Web application-

HTML 5 has many emerging new features and standards. The new features of HTML5 is to develop more easier application with effective front-ends, drag and drop tools, discussion boards, wikis and many more useful features.

Offline Application cache-

Different latest internet browsers have their own kind of caching mechanism. When after some time user again browses, he opens his computer, laptop and press the Back button in his browser. He can see his previous page that was opened previously. Although the user is not connected to the internet at that time. All this happen due to HTML 5. The developer can mention or specify the file which is cached by the browser, during development of web site. Therefore, that cached page loaded correctly when user is offline or he refreshes the page. This type of caching is very beneficial it gives many benefits as-

- files load much faster
- offline browsing
- reduced load on server

Client-side database-

As we know that the cookies are used to track unique user data. The cookie data is added with each HTTP request header. By using HTML5 user can reduce cookie size with help of session Storage and local Storage. They are two different storages in HTML5 in place of cookies.

Geo-location support-

Anybody can find your location in the world and also, he can share that location information to other people, with help of Geo-location support. We can also find out our IP address, our wireless network connection through it. User can find out; which cell tower his phone is connected. Dedicated GPS hardware also calculates latitude and longitude from data and information sent by satellites. The HTML5 has geo-location APIs that find the location. Geo-location can be directly accessed by any variable in HTML5-compatible browser-based application.

Overview and the various advantages of HTML5 are provided above. In case of HTML4, the programmers as well as developers use its own class/id names to provide various styles to elements: bottom, navigation, main, footer, menu, container, top nav, header, top, content, article, sidebar etc. After all, in HTML5 the various elements (<header> <footer> <nav> <section> <article>) are very easy to use. There are many more advantages of HTML5. In future the HTML5 will become an online web development tool or language.

7.4 HTML 5 BASICS

The new features and elements of HTML5 are easily integrated with multimedia and graphical content of web. If we are using HTML5 in the web then flash and third-party plug-in are not necessary to use. Following are some most important features of HTML5.

- **Semantic Elements-**

  <header>, <footer>, <section>.
- **Forms 2.0-**

  It is improved version of HTML web forms where new attributes can be introduced with <input> tag.
- **Persistent Local Storage-**

  To achieve without resorting to third-party plug-ins.
- **Web Socket-**

  It is used in web application for new-generation bidirectional communication technology.
- **Server-Sent Events-**

  Events that connect web server to the web browsers.
- **Canvas-**

  It supports a two-dimensional drawing surface, which can be in the program of JavaScript.
- **Audio & Video-**

  Without resorting to third-party plug-in, Audio and video can be embedded in webpage. With the help of audio control, browser gives user to playback, including volume, seeking, and pause/resume playback.

  Example- (Running audio on the webpage)

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

```
  <body>
          <audio src="song.ogg" controls="controls">
                  This particular browser does not support the <audio> tag.
          </audio>
  </body>
</html>
```

Example- (Running video on the webpage)

```
<! DOCTYPE HTML>
<html>
  <body>
          <video src="movie.ogg"  width="320" height="240”>
                  This browser does not support the <video> tag.
          </video>
  </body>
</html>
```

- **Geo location-**

  The user can find and share their physical location using web application.

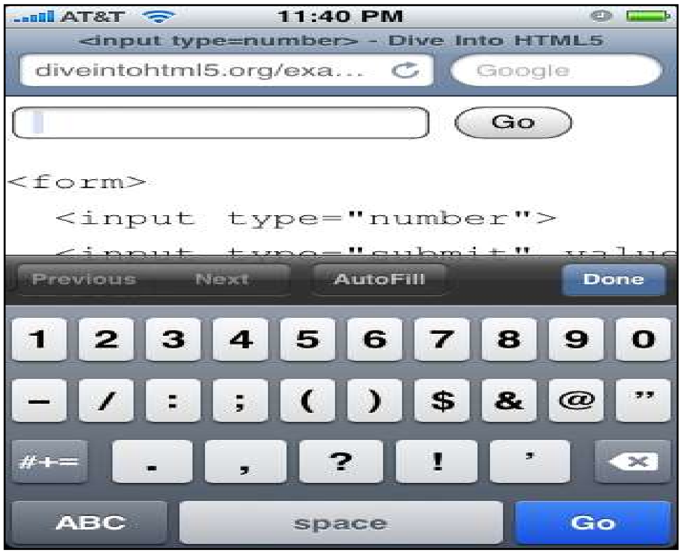
- **Micro data-**

  User can create his own vocabularies and extend his web pages along the custom semantics.

- **Drag and drop-**

  User can Drag and drop the items from one place to another place in any webpage.

- **Placeholder text-**
- **Specific text input: email, URL, number, search-**



```
<form>
  <input type="number">
```

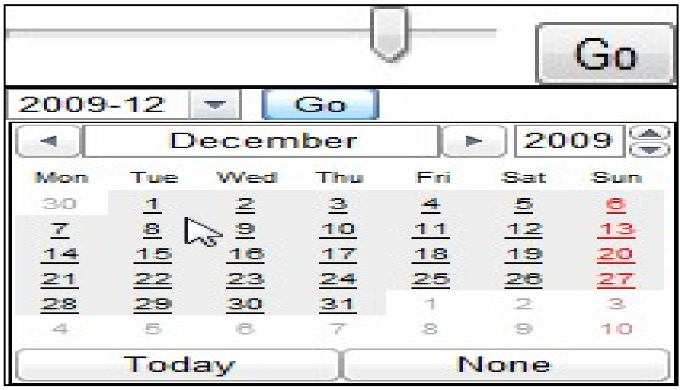
- **Slider-**
- **Input: E-mail-**

User can use INPUT TYPE =”email”. This format of email type is mainly used for input. After submitting the form the E-Mail field value is validated automatically as follows-

E-mail: <input type="email" name="user_email"/>

- **Date picker-**
- **User Agent validation**

7.5 SEMANTIC ELEMENTS IN HTML5



As per W3C, the semantic web is- "Allows data to be shared and reused across applications, enterprises, and communities”. "Semantics provides the meanings of words and phrases in any language. Semantic element describes very clearly their meaning to browser where as non-semantic elements do not tell about its content.

Example of non-semantic elements are- <div> and <span>.

Example of non-semantic elements are- <form>, <table>, and <article>

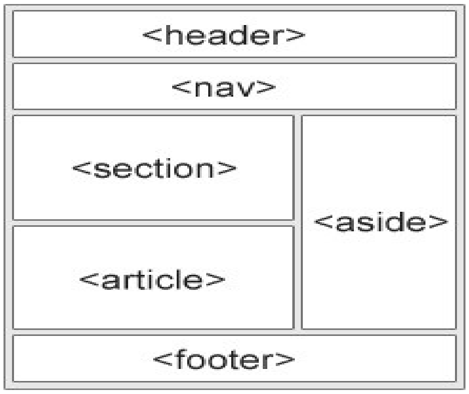
Semantic Elements of HTML5-

HTML5 has following new semantic elements in an alphabetical order for developing a web page. As-

Tag	Description
<article>	Which defines an article
<aside>	Which defines content aside from the page content
<details>	Which defines additional details that the web user can view/hide
<figcaption>	Which defines a caption for a <figure> element

<figure>	Which Specifies self-contained content, like illustrations, diagrams, photos, code listings, etc.
<footer>	Which defines a footer for a document/section
<header>	Which Specifies a header for a document/section
<main>	Which Specifies the main content of a document
<mark>	Which defines marked or highlighted text
<nav>	Which defines navigation links
<section>	Which defines a section in a document
<summary>	Which defines a visible heading for a element <details>
<time>	Which defines a date/time

Block Diagram of semantic elements-



<section> Element of HTML5-

This element defines a section in a document in web page. We can also say that it is used to define a block of related elements. Section is also called as thematic grouping of content with a heading. The home page of User can be divided into various sections such as contact information, introduction, and content.

Example-

```
<section>
        <h1>WWW</h1>
        <p>World    Wide Web    (WWW)     is     a…………....</p>
</section>
```

<article> Element of HTML5-

The above-mentioned element mainly indicates self-contained content and independent contents. This is required when any article needs to be read independently from the rest of the web site. The <article> element is used with the- Forum post, Newspaper article, Blog post

Example-

```
<article>
        <h1>WhatisHTML5?</h1>
        <p> advantages of HTML5:
```

It is cleaner mark-up language and it’s code is very improved. It enables web designers to use cleaner, neater code. User can remove div tags and replace them with HTML5 semantic elements…………...</p>

</article>

<header> Element of HTML5-

The above-mentioned element shows a header of a document/section. A document contains many <header> elements, and is used as a container for introductory content.

Example-

```
<article>
        <header>
                <h1>HTML</h1>
                <p>WhatisHTML5?</p>
        </header>

        <p> advantages of HTML5:

                It is cleaner mark-up language and its code is very improved.

                It enables web designers to use cleaner, neater code.

                User can remove div tags and replace them with HTML5 semantic
        elements…………...

        </p>

</article>
```

<footer> Element of HTML5-

The footer of a document/section can be displayed using above element. There may be many < footer > elements in a document. Basically, footer section contains- Author of the document, Copyright information, Contact information, Page Number etc.

Note- User can mention many <footer> elements in a web page.

Example-

```
<footer>
        <p>Program    written by:Dr   John</p>
        <p>Contact     information:<ahref="mailto:jhon@example.com">
                jhon@example.com</a>.

        </p>
</footer>
```

<nav> Element of HTML5-

This element defines a set of navigation links. As-

Example-

```
<nav>
        <ahref="/html5/">HTML5</a>
        <ahref="/css/">CSS</a>
        <ahref="/xml/">XML</a>
        <ahref="/sql/">SQL</a>
</nav>
```

<aside> Element of HTML5-

The above element defines content such that aside from the content it is placed in e.g. sidebar.

Example-

```
<p>India    is    very  great  country.                       </p>
<aside>
        <h4>Capital  Delhi</h4>
        <p>The      culture of    India  is      very   rich.</p>
</aside>
```

<figure> and <figcaption> Elements of HTML5-

The figure caption is used to add an image visual explanation or description. User can group both image and caption in a <figure> element. As-

Example-

```
<figure>
        <img src="cloud.jpg"     alt="Cloud"     width="400"     height="300">
        <figcaption>Fig1.  -  The  Cloudy  weather  at  Darjeeling.  </figcaption>
</figure>
```

In the above example, the element <img> defines the image; The element <figcaption> defines the caption of the image.

7.6 MIGRATION OF HTML4 ELEMENTS TO HTML5

An HTML5 web page can be obtained by HTML4 without destroying any of the original content or structure of HTML4 web page. All modern browsers support the HTML5 semantic elements. Following are some example of migration of HTML elements from HTML4 to HTML5.

HTML4	HTML5
<div id="header">	<header>
<div id="menu">	<nav>
<div id="content">	<section>
<div class="article">	<article>
<div id="footer">	<footer>

Example- (Following is an example of a sample HTML4 web Page)

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html lang="en">
    <head>
        <metahttp-equiv="Content-Type"content="text/html;charset=utf-8">
        <title>HTML4</title>
        <style>
            body{
                font-family:Verdana,sans-serif;
                font-size:0.9em;
            }
            div#header, div#footer{
                padding:10px;
                color:white;
                background-color:black;
            }
            div#content{
                margin:5px;
                padding:10px;
                background-color:lightgrey;
            }
            div.article{
                margin:5px;
                padding:10px;
                background-color:white;
            }
            div#menu ul{
                padding:0;
            }
            div#menu ul li{
                display:inline;
                margin:5px;
            }
        </style>
    </head>
    <body>
        <divid="header">
            <h1>MondayTimes</h1>
        </div>
        <div id="menu">
            <ul>
                <li>News</li>
                <li>Sports</li>
                <li>Weather</li>
            </ul>
        </div>
        <div id="content">
            <h2>NewsSection</h2>
            <div class="article">
                <h2>NewsArticle</h2>
                <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Pellentesque in porta lorem. Morbi  condimentum est nibh, et consectetur tortor feugiat at.</p>
            </div>
            <div class="article">
                <h2>NewsArticle</h2>
                <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Pellentesque in porta lorem. Morbi condimentum est nibh, et consectetur tortor feugiat at.</p>
            </div>
        </div>
        <div id="footer">
            <p>&amp;copy; 2016 Monday Times. All rights reserved.</p>
        </div>
    </body>
</html>
```

The above example explained step by step-

Change in the HTML5 doc type-

<! DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">

Example**-** <! DOCTYPE html>

Change in the HTML5 encoding information-

<meta http-equiv="Content-Type" content="text/html;charset=utf-8">

Example**-** <meta charset="utf-8">

Some modifications in the Semantic Elements of HTML5-

Following id's and classes for styling the elements are in the existing CSS-

```
body{
    font-family:Verdana,sans-serif;
    font-size:0.9em;
}
div#header, div#footer{
    padding:10px;
    color:white;
```

```
        background-color:black;
}
div#content{
        margin:5px;
        padding:10px;
        background-color:lightgrey;
}
div.article{
        margin:5px;
        padding:10px;
        background-color:white;
}
div#menu ul{
        padding:0;
}
div#menu ul li{
        display:inline;
        margin:5px;
}
```

For above CSS styles HTML5 semantic elements can be written as follows-

```
body{
        font-family:Verdana,sans-serif;
        font-size:0.9em;
}
header, footer{
        padding:10px;
        color:white;
        background-color:black;
}
section{
        margin:5px;
        padding:10px;
        background-color:lightgrey;
}
article{
        margin:5px;
        padding:10px;
        background-color:white;
}
nav ul{
        padding:0;
}
nav ul li{
        display:inline;
        margin:5px;
}
```

7.7 CODING CONVENTIONS IN HTML5

Document Type-

The following statement will be used as the first line of your webpage in HTML5.

```
        <! DOCTYPE html>
```

HTML5 we can write uppercase or lowercase letters in element names. But the uppercase and lowercase names should not be mixed. Developer should use either uppercase or lowercase element names. Generally, the lowercase letters are used to write the HTML elements. As-

```
<section>
        <p>I    Love    my      country.</p>
</section>
```

Image Attributes-

In HTML5 "alt" attribute is always used for images. This attribute is very important whenever image is not properly displaying. The width and height of image can also be defined by user. Since browser reserves some space for the image before its actual loading, it decreases flickering. As-

```
        <img src="sky.gif" alt="HTML5" style="width:70px;height:58px">
```

<html> and <body>

Example -

```
<!DOCTYPEhtml>
<html lang="en-UK">

        <head>
                <title>Title    of      Webpage         </title>
        </head>

        <body>
                <h1>This        is      first   heading</h1>
                <p>This is a paragraph.</p>

        </body>

</html>
```

In HTML5, you can omit <html> and <body> tag, however it omitted. The <html> tag is the root of your web page. User also can specify the page language using <html> tag. Defining a language is very important for accessibility and for search engines.

Meta Data (Data about Data)

Setting of the Viewport-

HTML5 introduced a method to let web designers take control over the viewport, through the <meta> tag. The user's visible area in HTML5 web page is known as viewport. The size of viewport varies as per the requirement. It will be larger on a computer screen and smaller on a mobile phone. So <meta> viewport element in this web page is used as follows-

<meta name="viewport" content="width=device-width, initial-scale=1.0">

Dimensions and scaling of web page can be controlled using, a <meta> viewport element which gives the instructions to the browser.

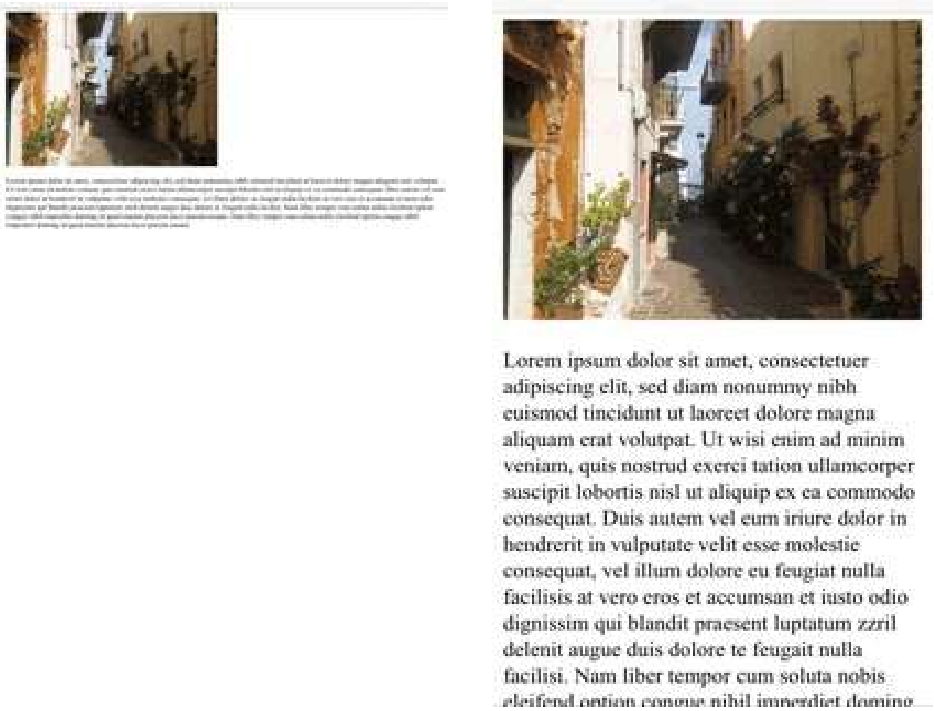
Here the width means the device-width, and hence the width of the page follows the screen-width of the device which will vary, according to the device (computer, phone or a tablet).

The initial-scale=1.0 part means the initial zoom level. Which is the page first that browser loads.

The two examples of web page are given as follows for the same web page. As-

i. Without the viewport meta tag

ii. With the viewport meta tag



Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat. Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue duis dolore te feugait nulla facilisi. Nam liber tempor cum soluta nobis eleifend option congue nihil imperdiet doming

(Left) without the viewport <meta> tag. (Right)With the viewport <meta> tag

HTML Comments-

User can use comments in HTML5 as-

```
<!-- This is a comment in one Line-->
```

Style Sheets-

We can link any cascading style sheet (CSS) with help of following syntax.

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

An example of how to write code in a cascading style sheet(CSS) is given as follows-

```
body{
        background-color:lightgrey;
        font-family:"ArialBlack",Helvetica,sans-serif;
        font-size:16em;
        color:black;
}
```

- After each property-value pair semicolon should be used. If the value contains spaces then it should be kept inside quotes e.g. "Arial Black".

Loading JavaScript into html-

User can use following syntax for loading any other scripts (e.g. Java Script, VB Script or C # Script)-

```
<script src="example.js">
```

7.8 SAMPLE WEBPAGE DESIGNING

The Figure Element-

The following statement displays the image in a web page-

```
<img src="path/of/image" alt="About image" />

<p>Image of House. </p>
```

HTML5 has improved the <figure> element tag. User can use <figcaption> element and this way figure can be associated with its captions.

```
<figure>

                <img src="path/of/image" alt="image Description" />

                <figcaption>

                        <p>This is an image of House. </p>

                </figcaption>

</figure>
```

<small> element-

The small element is used for "small print." Suppose user has to create subheadings or caption of the logo, a copyright statement in the footer of any site then the element <small> is best suited for it.

For Making the Content Editable-

This attribute used in the new browsers named as content editable and it allows the user to edit any of the text contained within the element. Following example show the uses of content editable-

```
<! DOCTYPE html>
```

```
<html lang="en">
        <head>
                <meta charset="utf-8">
                <title> Content Editable </title>
        </head>
        <body>
                <h2> Edit Area</h2>
                <ul content editable="true">
                        <li>This attribute used in the new browsers named as </li>
                        <li>content editable and it allows the user to edit any of the text </li>

                        <li> contained within the element. </li>
                </ul>
        </body>
</html>
```

In the above program we have written <ul content editable=true>

Email as Inputs-

In HTML5 the element "email" is used for the input a valid email address structure. The new browser has this facility however the older browsers are not able to understand the type "email". User simply uses regular textbox for E-Mail input. As-

```
<! DOCTYPE html>
<html lang="en">
        <head>
                <meta charset="utf-8">
                <title>untitled</title>
        </head>
        <body>
                <form action="" method="get">
                        <label for="email">E-Mail:</label>
                        <input id="email" name="email" type="email" />
                        <button type="submit"> New Submit Form </button>
                </form>
        </body>
</html>
```

The Header and Footer element-

In HTML5, simply <header> and <footer> elements are used and are very useful when we have multiple headers and footers in our webpage-

```
<header>
        ...
</header>
<footer>
        ...
```

```
</footer>
```

Autofocus Attribute-

HTML5 have advance autofocus attribute. It automatically focuses a particular text box. As-

```
<input type="text" name="Country" placeholder="India” autofocus>
```

Audio Support-

HTML5 provide new <audio> element to play songs, audio files etc. Some most recent browsers offer support for HTML5 audio. As-

```
<audio autoplay="autoplay" controls="controls">
        <source src="song1.ogg" />
        <source src="song2.mp3" />
        <a>Download above files.</a>
</audio>
```

Video Support-

Like the <audio> element, the HTML5 also have <video> tag. It can also used in the new browsers. Recently, YouTube has announced a new HTML5 video embed for their videos, for browsers which support it. As-

```
<video controls preload>
        <source src="file1.ogv" type="video/ogg; codecs='vorbis, theora'" />
        <source src="File2.mp4" type="video/mp4; codecs='avc1.42E01E,
mp4a.40.2'" />
        <p> Your browser isn’t supporting <a href="file3.mp4">Download
video.</a></p>
</video>
```

Regular Expressions-

HTML has the facility to for writing some regular expression to verify a particular type of textbox. By this new pattern attribute, user can insert a regular expression directly into his mark-up web page.

```
<form action="" method="post">
        <label for="username">Create a User name: </label>
        <input type="text" name="username" id="username" placeholder="name"
pattern="[A-Za-z]{4,10}" autofocus>
        <button type="submit">Move </button>
</form>
```

Hear the pattern: [A-Za-z]{4, 10} means that it will accepts only upper and lowercase letters. Also the string must have 4 minimum characters, and 10 maximum characters.

To Detect Support for Attributes-

Sometime the browser does not recognize HTML5 attributes so the newer changes and advances becomes absolute. Therefore it is important to figure out how to apply these advance attribute in such browser. There are several ways to figure this out. Mainly we have two ways. The first method is to utilize the excellent modernize library. On the other way, user can create and dissect these elements and to determine what the browsers are capable of or support. Suppose, in the previous example, as user want to determine if the browser can implement the pattern attribute, he could include a small JavaScript code in his page:

```
alert( 'pattern' in document.createElement('input') ) // boolean;
```

This is most commonly used method to find out the browser compatibility. Generally the jQuery library uses this type of trick. In the above example, the user is creating a new input element, and also determining whether the pattern attribute is recognized within it or not. If it is recognized then the browser supports this functionality. As-

```
<script>
        if (!'pattern' in document.createElement('input') ){
                // start validation client or server side
        }
</script>
```

Mark Element-

The <mark> element is like a highlighter. Any text wrapped within this tag should be appropriate to the recent activity of the user. e.g. when user has to find "Indi’s best cities" on any blog, he could then utilize small JavaScript code to wrap each occurrence of this string within <mark> tags.

```
<h3> Results of Search </h3>
<p> They came India, just to know about India and ask for, <mark>"Indi’s best
cities"</mark>. </p>
```

Use of <div>

If we have already div tag, then what is the need of header, article, section, and footer tags. These tags are smarter however Div should also be utilized when there is not any better element is available for the job.

For example, suppose user has to find that he needs to wrap a code block within a wrapper element then preferably for the purpose of positioning the content, a <div> makes perfect sense. Sometime a user is wrapping a new blog post, or, perhaps, a list of links in his footer, consider using the <article> and <nav> elements, respectively then the div is more semantic.

The Output Element-

The output element is used to display some part of calculation. Suppose user like to print coordinates of a mouse position, or the addition of a series of numbers, then such information should be inserted into the output element.

In the following example, suppose user has to insert the addition of 2 numbers into an empty output along with JavaScript, after pressing the submit button.

```
<form action="" method="get">
        <p>
                10 + 5 = <output name="sum"></output>
        </p>
        <button type="submit"> Calculate </button>
</form>
<script>
        function(){
                var f = document.forms[0];

                if ( typeof f['sum'] !== 'undefined' ) {
                        f.addEventListener('submit', function(e) {
                                f['sum'].value = 15;
                                e.preventDefault();
                        }, false);
                }else { alert('Your browser is not ready yet.'); }
        })();
</script>
```

Create sliders with the range input-

The HTML5 introduces the new input type range.

```
<input type="range">
```

By using this input user can receive min, max, step, and value attributes, among other value. However Opera browser supports this kind of input type.

For example- we are building a real-world polling solution, and we are reviewing how it is happening quite easily.

Step one: Mark-up

In step one, you will create your own mark-up.

```
<form method="post">
          <h1> Total number of candidate turn up </h1>
          <input type="range" name="range" min="0" max="10" step="1" value="">
          <output name="result"> </output>
</form>
```

You will notice that, the value of min and max are “0” and “10” and the step is initialized as “1”. When the step is set to 1, then there will then be 10 values to choose. Here you can also take advantage of the new output element that you have learned already.

Step two: CSS

In the step two, you will style it. We'll use colon (:) before and after tag and you will inform the user about your specified min and max values. As-

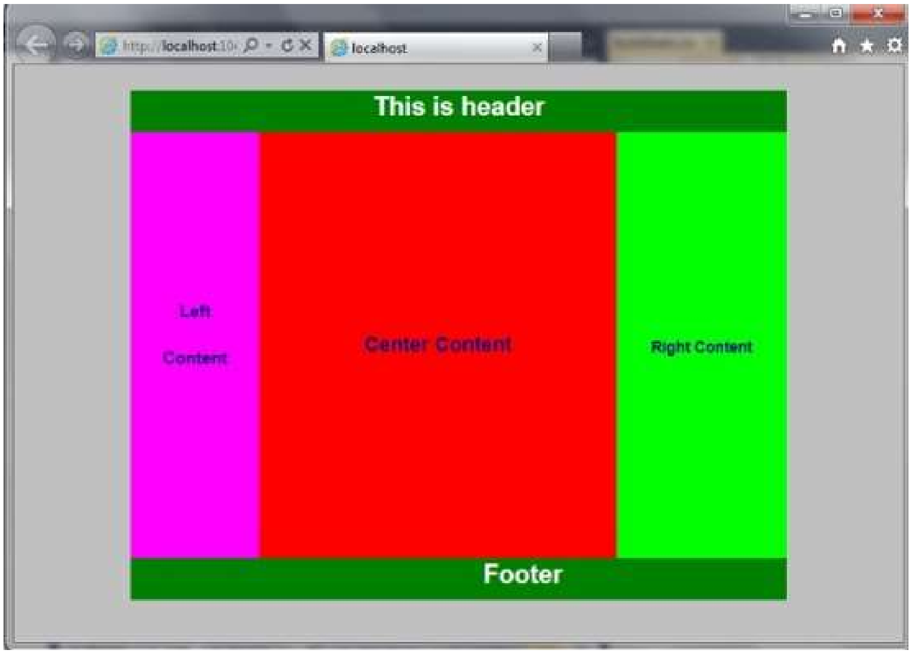
```
body{
        font-family: 'Myriad-Pro', 'myriad', helvetica, arial, sans-serif;
        text-align: center;
}
input{ font-size: 14px; font-weight: bold;  }
input[type=range]:before{ content: attr(min); padding-right: 5px; }
input[type=range]:after { content: attr(max); padding-left: 5px;}
output{
        display: block;
        font-size: 5.5em;
        font-weight: bold;
}
```

In the above example, you are creating content before and after the range input, and making their values equal to the min and max values, respectively.

Step three: The JavaScript

- If the current browser knows what is the input range then it’s ok otherwise alert the user that the demo will not work.
- And dynamically update the output element, as the user moves the slider.
- Listen for when the user mouse off the slider further grab the value, and save it in to the local memory.
- At last when, next time the user refreshes the page, the range and output will automatically be set at the level where they last time selected.

```
function()
{
```



```
var f = document.forms[0],range = f['range'],
        result = f['result'],
        cachedRangeValue = localStorage.rangeValue ?
        localStorage.rangeValue : 5;
        // determine whether the browser is one who
        // recognizes the range input.
var o = document.createElement('input');
o.type = 'range';
if ( o.type === 'text' ) alert('Sorry. Your browser is not
supporting. Try the latest Opera.');
// Set initial values of the input & output elements to
// either what's stored locally, or the number 5.
range.value = cachedRangeValue;
result.value = cachedRangeValue;
// When the user makes a selection, update local storage.
range.addEventListener("mouseup", function()
{
        alert("The selected value was " + range.value + ". I am using local
        storage to remember the value. Refresh and check on a modern
  browser.");
        localStorage ? (localStorage.rangeValue = range.value) :
        alert("Save data to database or something instead.");
}, false);
// Display chosen value when sliding.
```

```
    range.addEventListener ("change", function(){
        result.value = range.value;
    }, false);
})();
```

Use of div tag-

In a HTML page to define section or a division the <div> tag is used. In your document you can group large sections as a block and can be formatted using Style Sheet (CSS) in your document. User can wrap a header block in one div and content block in another div. In the given below code, Many div is used for each block. Basically, it's like a container where an user can create a division in which his documents can reside and also he can alter his own document blueprint by using the class and id names in the div tag. In the position of div tag new HTML5 has begins a number of tags like article, aside, footer, header, section and nav. The example and syntax are given as follows:

Syntax-

```
<div align="It Specifies the alignment of the paragraph or content inside a div tag. It can also
be aligned: center | justify | left | right"
class=" class name of Specification"
dir="   rtl (Right to Left | ltr (Left to Right)"
id="It specifies unique id identifier"
lang=" It Specifies a language code for the content in an element. "
style=" It Specifies style information"
title=" It Specifies advisory text">

</div>
```

Example- (File Name- Html_div_page.htm)

```
<!DOCTYPEhtml>
<html>
    <head>
        <linkhref="HtmlStyleSheet.css"rel="stylesheet"type="text/css"/>
        <title>   </title
    </head>
    <body>
        <div class="MainBody">
            <divclass="header">
                <h3>Its header</h3>
            </div>
            <div class="left">
                <h4>Left</h4>
                <h5>Content</h5>
            </div>
            <div class="center">
```

```
                <h1>CenterContent</h1>
            </div>
            <div class="right">
                <h2>RightContent</h2>
            </div>
            <div class="footer">
                <h3>Footer</h3>
            </div>
        </div>
    </body>
</html>
```

(File Name- HtmlStyleSheet.css)

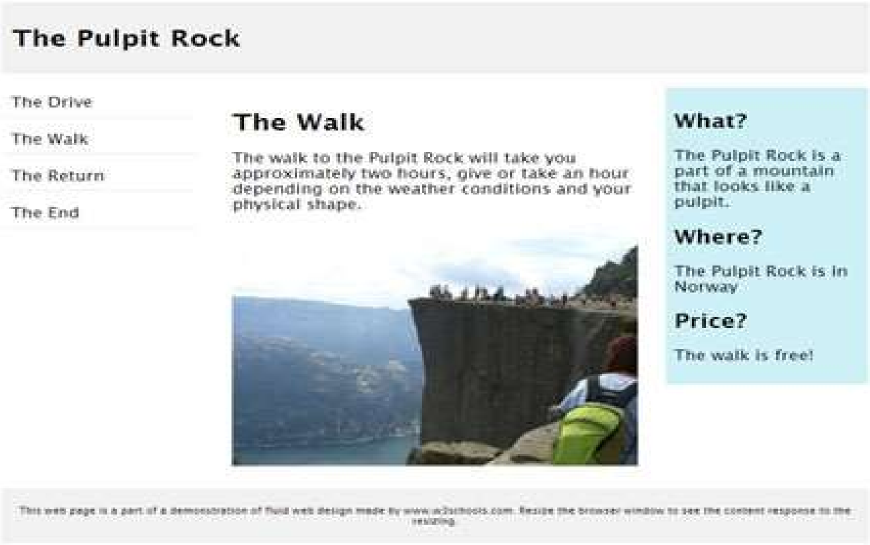
```
body{
    margin:0px;
    padding:0px;
    width:100%;
    color:#959595;
    font:normal12px/1.8emArial,Helvetica,sans-serif;
    background:silver;
}
.MainBody{
    width:770px;
    margin:0auto;
    padding:0px;
}
.header{
    background-color:Green;
    height:50px;
    text-align:center;
    font-size:20px;
    color:White;
    margin:0px;
    padding:0px;
    width:auto;
}
.headerh3{
    font-size:30px;
    line-height:40px;
}
.left{
    width:150px;
    height:500px;
```

```
        margin:0px;
        padding:0px;
        background-color:Fuchsia;
        float:left;
        text-align:center;
}
.lefth4{
        font-size:20px;
        color:Navy;
        margin-top:200px;
}
.lefth5{
        font-size:20px;
        color:Navy;
}
.center{
        width:420px;
        background-color:Red;
        margin-top:-16.1px;
        margin-left:150px;
        padding:0px;
        height:500px;
}
.centerh1{
        color:Navy;
        text-align:center;
        padding:250000;
        vertical-align:bottom;
        line-height:500px;
}
.right{
        width:200px;
        height:500px;
        background-color:Lime;
        float:right;
        margin-top:-500px;
        margin-left:0px;
        padding:0px;
        color:Navy;
        text-align:center;
        line-height:475px;
}
.footer{
        background:yellow;
        background-color:Green;
        height:50px;
        text-align:center;
        padding-left:0px;
        font-size:20px;
        color:White;
        margin:-30px0px0px0px;
        width:auto;
}
.footerh3{
        font-size:30px;
        line-height:40px;
}
```

Output-

7.9 SAMPLE RESPONSIVE WEBPAGE DESIGNING

Responsive web design is basically for look good of web page on different devices as desktops, tablets, and phones etc. By using the Responsive Webpage designing in HTML and CSS we can shrink, enlarge, resize, hide, or move the content of web page and make it looks good on any screen size. The following figure will always look same in any kind of screen width. As-



The Pulpit Rock

The Walk

The walk to the Pulpit Rock will take you approximately two hours, give or take an hour depending on the weather conditions and your physical shape.

What?

The Pulpit Rock is a part of a mountain that looks like a pulpit.

Where?

The Pulpit Rock is in Norway

Price?

The walk is free!

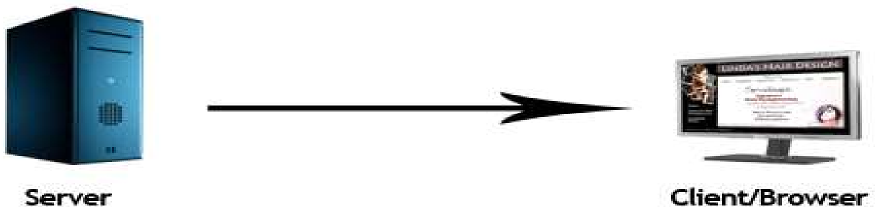
This web page is a part of a demonstration of fluid web design made by www.w3schools.com. Resize the browser window to see the content response to the resizing.

7.10 STATIC V/S DYNAMIC WEBSITE DESIGN

Generally, we categories websites into two types- Static Website and Dynamic Website. The static website is normally written using HTML/CSS (without use of database) and its pages are displayed to the user. On the other hand, a dynamic website is written by means of a server-side scripting language e.g. JSP, PHP, ASP, or ColdFusion (must use database).

Static Website-

This type of web site is very easy to create. Web developer doesn't need any web programming and database design/ database connectivity to create a static website. This type of web page is prepared in HTML. The HTML code is fixed for each page. Therefore, the information contained in the page does not change dynamically.

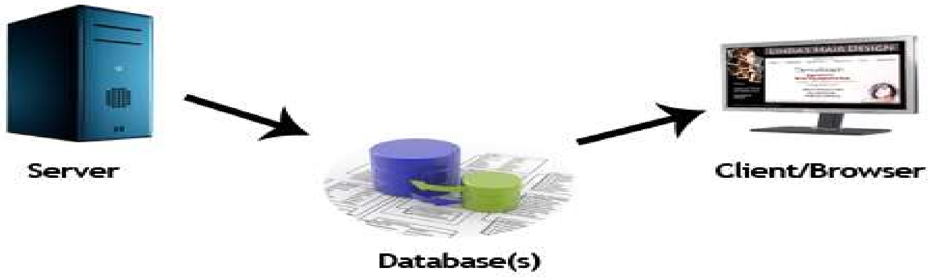


Some points about static website are-

- It's every page can differ.
- Layout of every page can be designed in a different way as per the content.
- The designer is free to mention any special effects which may be demanded by the client in a unique way on different pages.
- It is also called informational website.
- It cannot interact to visitors during web surfing.
- Modification of the content is one of the key problem.
- If you have to change anything on the website then you have to contact to the designer to have any content changes made.
- Static web pages are not scalable. If you are selling products on your site and you have a lot of products then you may have to construct individual pages for each product, then it will take considerable time, cost and effort for web development.

Dynamic Websites-

This type website accesses content from a database or Content Management System (CMS). If you modify or alter the content of the database, simultaneously the content of the website is also altered or modified. It uses client-side scripting or server-side scripting or both and also generates dynamic content. On the basis of user input, Client side scripting generates content at the client computer and Server Side Scripting creates script for the server. In server side



scripting, the software runs on the server. When the processing in the server is completed then plain pages are sent to the user as responses. The client web browser downloads the web page from the server. After that it processes the code within the web page to render information to the user.

Some points about dynamic website are-

- Companies can develop their product pages or categories of related item or products sorted in a variety of different ways depending on how the client wants to view them.
- User can easily pull in information in an organized and structured way.
- We can create a content management system due to the ability to connect to a database.
- The Costs of the dynamic website is higher initially than for a static site.

7.11 POINTS TO REMEMBER

- Multimedia Enabled- HTML5 enables a browser to play video and audio files.
- HMTL5 Input Element Types- In HTML5 a few recently made input types have been included.
- Canvas in HTML5- HTML5 is canvas areas enabled. Canvas is a rectangular area, which allows pixel level work such as a line, circle, box, performing graphics, etc.
- Editable Contents in HTML5- This feature lets the end users edit the HTML control's make up. This type of feature allows developers to build web pages that contain sections, such as- HTML editor notes etc.
- HTML 5 enables web designers to use cleaner, neater code; you can remove most <div> tags and replace them with semantic HTML 5 elements.

7.12 GLOSSARY

- HTML5 stands for Hypertext Markup Language version 5. World Wide Web Consortium (W3C) published it in October 2014.
- HTML5 is the new standard that is expected to take over the Web. Presently, there are three firms that oversee the description of HTML5, they are-
  [i] Web Hypertext Application Technology Working Group (WHATWG)- HTML5 specification was made and developed by WHATWG. The development makes for a working relationship with browser vendors as well as other interested parties.
  [ii] World Wide Web Consortium (W3C)- It takes care of producing HTML5 specifications.
  [iii] Internet Engineering Task Force (IETF)- It develops html5 web Socket known as API

7.13 CHECK YOUR PROGRESS

Descriptive type questions-

a) What is HTML5? How is different from HTML4? What are its characteristics? Explain.

b) What are the different Support Elements of HTML5? Explain.

c) What do you understand by Semantics? Explain the difference between Semantic and non-semantic element? Give example.

d) What are the major benefits of HTML5? How will you migrate from HTML4 to HTML5? Explain.

e) Explain different Style Guide Media? Give proper syntax or example.

f) What are the different tips and tricks of Sample Web Page Designing? Explain.

g) What is the importance of Div tag in HTML? How is it used? Give suitable example.

h) What do understand by the Responsive web design? What is the effect of responsive web design? How it will affect the look and feel of your web page? Explain.

i) What is the difference between Static and Dynamic Websites? How Dynamic website is better than the Static web site? Explain.

j) What are the advantage and disadvantages of static and dynamic web site? Explain with help of some example.

7.14 BIBLIOGRAPHY / REFERENCES

- https://www.w3.org/2009/Talks/2009-07-24-smith-html5/slides.pdf, December, 2017
- http://www.testingbrain.com/pdf-tutorials/html5-tutorials.html, December, 2017
- https://www.alvinisd.net/cms/lib03/TX01001897/Centricity/Domain/1077/beginning_html5_and_css3.pdf, December, 2017
- https://www.w3schools.com/html/html5_intro.asp, December, 2017
- http://codinginparadise.org/presentations/intro_html5.pdf, December, 2017
- http://home.hit.no/~hansha/documents/software/software_development/topics/resources/programming/exercises/Introduction%20to%20Web%20Programming.pdf, December, 2017

7.15 SUGGESTED READINGS

- Powell. Thomas A., HTML & CSS: The Complete Reference, 5E
- Holzner. Steven, PHP: The Complete Reference
- Powell. Thomas A., JavaScript: The Complete Reference
- lemay. Laura, rafe colburn, jennifer kyrnin, Mastering HTML, CSS & JavaScript Web, BPB Publication, 2016.
- vishvajeet. Sisodia, Basic of Web Design, HTML, CSS3, Centrum Press, 2014.

UNIT 8

JAVASCRIPT

8.1 INTRODUCTION

JavaScript is the programming language that facilitates web pages to interact with client. It presents a page more interactive and dynamic. It was formed in 1995 for web-based programming and today it also runs on your phone applications.

This script was invented by Brendan Eich in 1995, and became an ECMA (European Computer Manufacturers Association) standard in 1997. Initially JavaScript was named as “LiveScript”, but Netscape Corporation changed its name to JavaScript, possibly because of the excitement being generated by Java. JavaScript was introduced on Netscape 2.0 in 1995 with the name “LiveScript”.

Features of JavaScript

- JavaScript is a lightweight programming language.
- JavaScript is an Interpreted programming language.

- It is designed to create networking centred applications.
- It is complimentary to and integrated with Java.
- JavaScript is very easy to implement because it is integrated with HTML.
- It is cross-platform and open.

8.2 OBJECTIVE

After successful completion of this unit you will be able to-

- Define JavaScript
- Include JavaScript in your HTML page.
- Create objects and access their properties.
- Create your own interactive web page and develop logical thinking.
- Take input from user and providing relevant response using JavaScript.

8.3 ADD JAVASCRIPT IN YOUR HTML PAGE

There is a flexibility given to include JavaScript code anywhere in an HTML Document. However, the most preferred ways to include JavaScript in an HTML file are as follows:

- Script in <head>...</head> section.
- Script in <body>...</body> section.
- Script in <body>...</body> and <head>...</head> sections.
- Script in an external file and then include in <head>...</head> section.

Script in <head>...</head> section-

Now you want to have a script run on some event, such as when a user clicks somewhere, then you will place that script in the head as follows.

```
<html>
        <head>
        <script type= “text/JavaScript”>
        function thismessage()
        {
                alert(“Hello World”);
        }
        </script>
        </head>
        <body>
        Click here for the result
        <input type= “button” onclick= “thismessage()” value= “This message” />
        </body>
</html>
```

JavaScript in <body>...</body> segment-

If you require a script to run as the page loads so that the script generates content in the page, then the script goes in the <body> portion of the document. In this case, you would not have any function defined using JavaScript. Take a look at the following code.

```
<html>
        <head>
        </head>
        <body>
                <script type= “text/JavaScript”>
                document.write(“Hello World”);
                </script>
        <p>This is web page body </p>
        </body>
</html>
```

Script in <body>...</body> and <head>...</head> segment-

You can put your JavaScript code in <head> and <body> section altogether.

```
<html>
        <head>
                <script type= “text/JavaScript”>
                function thismessage() {
                        alert(“Hello World!”);
                        }
                </script>
        </head>
        <body>
                <script type=”text/JavaScript”>
                        document.write(“Hello World!”);
                </script>
                <input type= “button” onclick= “thismessage()” value= “This Message” />
        </body>
</html>
```

External file script and then include in <head>...</head> section-

As you begin to work more comprehensively with JavaScript, you will be likely to find that there are cases where you are reusing identical JavaScript code on multiple pages of a site. You are not restricted to be maintaining identical code in multiple HTML files.

The script tag provides a mechanism to allow you to store JavaScript in an external file and then include it into your HTML files.

Here is an example to show how you can include an external JavaScript file in your HTML code using script tag and its *src* attribute.

```
<html>
        <head>
                <script type= “text/JavaScript” src= “filename.js” > </script>
        </head>
        <body>
```

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

To use JavaScript from an external file source, you need to write all your JavaScript source code in a simple text file with the extension ".js" and then include that file as shown above. For example, you can keep the following content in *filename.js* file and then you can use message() function in your HTML file after including the *filename.js* file.

```
function message() {
        alert("Hello World");
}
```

8.4 PROGRAMMING VS. SCRIPTING

JavaScript and Java-

JavaScript and java are both Object-Oriented Programming languages, but JavaScript is an object-oriented scripting language. They are completely different programming languages with a small number of similarities. JavaScript is mainly used in Web pages. Almost all Java expression syntax and naming conventions are followed by JavaScript. This is the main reason for calling it JavaScript. Unlike Java, JavaScript does not bother about the methods, variables and classes at all. Java is much more complex than JavaScript. Let us compare Java and JavaScript in detail.

- In Java, the source code is first converted into an intermediate code, known as the byte code. This byte code is non-readable by humans and is independent of any machine. Afterwards, this byte code is interpreted by the interpreter (Java Virtual Machine). Since JavaScript is a scripting language, it is only interpreted. So, making any modification in a Java program are more difficult than making the changes in a JavaScript program.
- Java needs the Java Development Kit. JavaScript mainly requires a text editor.
- Java is a strongly typed language instead JavaScript is a loosely typed language.
- In a Web environment, Java is a back-end language whereas JavaScript is a front-end language.
- JavaScript is considered as a part of HTML file, so it is visible in the source file. Java applets are not a part of HTML file so they are not visible in the source file.
- Java is very complex to learn due to its rules and restrictions whereas JavaScript is very easy to learn, as compared to Java.
- The User Interface of JavaScript is developed in HTML and is very easy to understand, whereas the User Interface of Java is developed in AWT or Swing, which is very complex to understand.
- The client-side is more secured in Java compared to JavaScript.
- In Java and JavaScript, routines are known as the methods and functions respectively.

- Java supports Polymorphism, but JavaScript does not support Polymorphism.
- Java uses the classes and objects to make its code reusable easily, but this service is not available in JavaScript.

JavaScript can be implemented using JavaScript statements that are placed within the <script>... </script> HTML tags in a web page. The <script> tags may be placed anywhere in your web page, but it is a recommended to keep it within <head> tags. The <script> tag alerts the browser program to start interpreting all the text between these tags as a script.

The syntax of our JavaScript should be -

```
<script> JavaScript code </script>
```

The script tag contains two attributes:

- Type: This attribute value should be set to “text/JavaScript”.
- Language: This attribute specifies the scripting language we are using. Here it is JavaScript.

So now our JavaScript syntax is:

```
<script language= “JavaScript” type= “text/JavaScript”>
        JavaScript code;
</script>
```

Example:

Let us take a sample example to print “Hello World”.

we call a function *document.write* which writes a string into our HTML document.

```
<html>
<body>
<script language= “JavaScript” type= “text/JavaScript”>
document.write (“Hello World!”);
</script>
</body>
</html>
```

This code will produce the following result:

```
Hello World!
```

Programming Vs. Scripting-

Basically, all scripting languages are programming languages. The theoretical difference between the two is that scripting languages do not require the compilation step and are rather interpreted. For example, normally, a C program needs to be compiled before running whereas normally, a scripting language like JavaScript or PHP need not be compiled.

Generally, compiled programs run faster than interpreted programs because they are first converted native machine code. Also, compilers read and analyze the code only once, and report the errors collectively that the code might have, but the interpreter will read and analyze the code statements each time it meets them and halts at that very instance if there is some error.

Scripting languages are languages that are not compiled, more like interpreted at run-time. It's like a file containing instructions for a computer to follow to carry out a task. Examples of scripting languages are JavaScript, VB script, Perl, python, php, shell script, etc.

A scripting language generally sits behind some programming language. JavaScript is extremely fast to load since most of the web browser comes with already installed components.

8.5 TOKENS IN JAVASCRIPT

Tokens are the smallest individual words, phrases, or characters that JavaScript can understand. When JavaScript is interpreted, the browser parses the script into these tokens while ignoring comments and white space. JavaScript tokens fit in five categories:

- Identifiers
- Keywords
- Literals
- Operators
- Separators

Identifiers-

Identifiers are simply names that represent variables, methods, or objects. They consist of a combination of characters and digits. Some names are already built into the JavaScript language and are therefore reserved these identifiers are called as "Keywords". Aside from these keywords, you can define your own creative and meaningful identifiers. Of course, you have a couple of rules to follow. As-

- You must begin all identifiers with either a letter or underscore (_).
- You can then use letters, digits, or underscores for all subsequent characters.
- Letters include all uppercase characters, “A” through “Z”, and all lowercase characters, “a” through “z”.
- Digits include the characters “0” through “9”.

The table below shows some examples of valid and invalid identifiers. As-

Valid	Invalid	Reason
current- WebSite	current WebSite	contains a space
NumberOfHits	#ofIslands	pound sign is prefixed

Keywords-

Keywords are predefined identifiers that make up the core of a programming language. In JavaScript, they perform unique functions such as declaring new variables and functions, making decisions based on the present state of the computer, or starting a repetitive loop inside your application. Keywords, which are built into JavaScript, are always available for use by the programmer but must follow the correct syntax.

List of keywords		

break
continue
else
false
this
while
	for
	function
	if
	in
	true
		int
		new
		null
		return
		var

Literals-

Literals are data comprised of numbers or strings used to represent fixed values in JavaScript. They are values that do not change during the execution of your scripts. The following five sections contain descriptions and examples of the different types of literals that you can use.

Integer Literals

Integers can be expressed in either decimal (base 10), octal (base 8), or hexadecimal (base 16) format. An integer literal in decimal format can include any sequence of digits that does not begin with a 0 (zero). A zero in front of an integer literal designates octal form. The integer itself can include a sequence of the digits 0 through 7. To designate hexadecimal, Ox (or OX) is used before the integer. Hexadecimal integers can include digits 0 through 9 along with the letters. a through f or A through F. Some examples include:

Decimal (base 10)	33,2139
Octal (base 8)	071,03664
Hexadecimal (base 16)	Ox7b8, OX395

Floating-Point Literals

Floating-point literals represent decimal numbers with fractional parts. They can be expressed in either standard or scientific notation. With scientific notation, use either e or E to designate the exponent. Both the decimal number and exponent can be either signed or unsigned as shown in the examples:

3405.673

-1.958

8.3200e+ 11

8.3200e11

9.98E-12

Boolean Literals

JavaScript implements Boolean data types and therefore supports the two literals, true and false.

They represent the Boolean values 1 and 0, respectively. The true and false keywords must appear in lowercase. As a result, the capitalized words TRUE and FALSE are left open to define as your own identifiers, but it is not recommended.

String Literals

A string literal is zero or additional characters enclosed in double (“) or single (‘) quotes. JavaScript gives you this option, but you must use the same type of quote to surround each string. The following are examples of string literals enclosed in quotes:

“virtual communities”

‘virtual communities’

“Look, up in the sky!”

Special Characters

When writing scripts, you might sometimes need to tell the computer to use a special character or keystroke such as a tab or carriage return. To do this, use a backslash in front of one of the special characters as shown in the following list:

\b	indicates a backspace
\f	indicates a form feed
\n	indicates a new-line character
\r	indicates a carriage return
\t	indicates a tab character

Operators-

The following tokens are used in JavaScript as operators:

Operator	Description	Example	Result in ‘x’
=	x = y	x = y	x = 5
+=	x += y	x = x + y	x = 15
-=	x -= y	x = x - y	x = 5
*=	x *= y	x = x * y	x = 50
/=	x /= y	x = x / y	x = 2
%=	x %= y	x = x % y	x = 0

Operator	Description	Example	Result in ‘y’	Result in ‘x’
+	Addition	x = y + 2	y = 5	x = 7
-	Subtraction	x = y - 2	y = 5	x = 3
*	Multiplication	x = y * 2	y = 5	x = 10
/	Division	x = y / 2	y = 5	x = 2.5
%	Modulus (division remainder)	x = y % 2	y = 5	x = 1
++	Increment	x = ++y	y = 6	x = 6
		x = y++	y = 6	x = 5
--	Decrement	x = --y	y = 4	x = 4
		x = y--	y = 4	x = 5

Operator	Description	Comparing If x=5	Returns
= =	equal to	x = =5	true
= = =	equal value and equal type	x = = = "5"	false
		x = = = 5	true
!=	not equal	x != 8	true
!= =	not equal value or not equal type	x != = "5"	true
		x != = 5	false
>	greater than	x > 8	false
<	less than	x < 8	true
>=	greater than or equal to	x >= 8	false
<=	less than or equal to	x <= 8	true

Separators-

The following characters are used in JavaScript as separators (punctuators):

Separator: one of :- ( ) { } [ ],

8.6 BASIC PROGRAMMING CONSTRUCTS IN JAVASCRIPT

JavaScript is a programming language that adds interactivity to your website (for example: games, responses when buttons are pressed or data entered in forms, dynamic styling, and animation). For doing that you should know the basics of language, which are discussed below.

Displaying output in JavaScript

JavaScript can “display” data in different ways:

(i) Writing into an HTML element, using innerHTML

(ii) Writing into the HTML output using document.write()

(iii) Writing into an alert box, using window.alert()

Writing into an HTML element, using innerHTML

To access an HTML element, JavaScript can use the document.getElementById(id) method. The id attribute defines the HTML element. The innerHTML property defines the HTML content:

```
        <html>
                <body>
                        <h1>My First ever Web Page</h1>
                        <p>My First ever Paragraph</p>
                        <p id= “demo1”></p>
                        <script>
                                document.getElementById(“demo1”).innerHTML = 7 + 6;
                        </script>
                </body>
</html>
```

Writing into the HTML output using document.write()

Writing into the HTML output using document.write(). For testing purposes, it is convenient to use document.write():

```
<html>
        <body>
                <h1>My First ever Web Page</h1>
                <p>My first ever paragraph.</p>
                <script>
                document.write(7 + 6);
                </script>
        </body>
</html>
```

Using document.write() after an HTML document is completely loaded, will delete all existing HTML.

Writing into an alert box, using window.alert()

You can use an alert box to display data:

```
<html>
        <body>
                <h1>My First ever Web Page</h1>
        <p>My first ever paragraph.</p>
        <script>
            window.alert(7 + 6);
        </script>
    </body>
</html>
```

Example: JavaScript program to calculate multiplication and division of two numbers (input from user).

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset=utf-8 />
        <title>Program to calculate multiplication and division of two numbers
    </title>
        <style type= “text/css”>
            body {margin: 30px;}
        </style>
        <script type = “text/JavaScript” language=”JavaScript”>
            function multiplyBy()
            {
                num1 = document.getElementById(“fstNum”).value;
                num2 = document.getElementById(“scndNum”).value;
                document.getElementById(“answer”).innerHTML = num1 *
num2;
            }
            function divideBy()
            {
                num1 = document.getElementById(“fstNum”).value;
                num2 = document.getElementById(“scndNum”).value;
                document.getElementById(“answer”).innerHTML = num1 /
num2;
            }
        </script>
    </head>
    <body>
        <form>
            1st Number : <input type= “text” id=“ fstNum” /><br>
            2nd Number: <input type= “text” id= “scndNum” /><br>
            <input type= “button” onClick= “multiplyBy()” Value= “Multiply” />
            <input type= “button” onClick= “divideBy()” Value= “Divide” />
        </form>
        <p> Answer is : <br>
            <span id = “answer”></span>
        </p>
    </body>
</html>
```

Some Basics about JavaScript

Statement-

A statement is a line of code.

Example:

```
var hello = “Hello”;

var world = “World”;

var message = hello + “ ” + world;
```

This code can be executed by another program called an interpreter that will read the code, and execute all the statements in the right order.

Case sensitivity-

All JavaScript identifiers are **case sensitive**. The variables **lastName** and **lastname**, are two different variables.

```
var lastname, lastName;
lastName = “Naman”;
lastname = “Malik”;
```

JavaScript does not interpret VAR or Var as the keyword var.

JavaScript is a case-sensitive language. This means that the language keywords, variables, function names, and any other identifiers must always be typed with a consistent capitalization of letters. So the identifiers Time and TIME will convey different meanings in JavaScript.

Semicolon-

Semicolons separate JavaScript statements. Add a semicolon at the end of each executable statement:

```
var x, y, z;
x = 5;
y = 6;
z = x + y;
```

Comments-

Comments are statements that will not be executed by the interpreter, comments are used to mark annotations for other programmers or small descriptions of what your code does, thus making it easier for others to understand what your code does.

In JavaScript, comments can be written in 2 different ways:

A. **Line starting with //:**

```
// This is a comment, it will be ignored by the interpreter
```

B. **Section of code starting with /*and ending with */, this method is used for multi-line comments:**

```
/*

This is a multi-line comment,
```

it will be ignored by the interpreter

*/

Variables-

Variables are containers for changeable values. Variables can hold all kind of values and also the results of computations. Variables have a name and a value separated by an equal sign (=). Use the var keyword only for declaration or initialization, first time you use the variable in a document. You should not re-declare same variable twice.

JavaScript is **untyped** language. This means that a JavaScript variable can hold a value of any data type. Unlike many other languages, you don't have to tell JavaScript during variable declaration what type of value the variable will hold. The value type of a variable can change during the execution of a program and JavaScript takes care of it automatically.

Example:

```
var value;

var result = 2;
```

You can also declare multiple variables with the same var keyword as follows –

```
var value, result;
```

Example:

This example defines two variables, after computing their addition the result is stored in third variable.

```
var x = 5;

var y = 6;

var result = x + y;
```

Types of variable-

Computers are sophisticated and can make use of more complex variables than just numbers. This is where variable types come in. Variables come in several types and different languages support different types.

The most common types are:

a. **Numbers:**

   i. Float: a number, like 1.21323, 4, -33.5, 100004 or 0.123

   ii. Integer: a number like 1, 12, -33, 140 but not 1.233

b. **String**: a line of text like “boat”, “elephant” or “damn, you are tall!” (We will discuss more on strings in later section).

c. **Boolean**: either true or false, but nothing else

d. **Arrays**: a collection of values like: 1, 2, 3, 4, ‘I am bored now’

e. **Objects**: a representation of a more complex object

f. **Null**: a variable that contains null contains no valid Number, String, Boolean, Array, or Object.

g. **undefined**: the undefined value is obtained when you use an object property that does not exist, or a variable that has been declared, but has no value assigned to it.

Strings-

This is a string.

“I love programming”;

You can define strings using var keyword either using single or double quotes.

// ‘Single quotes’

var str = ‘Please Enter your name’;

// “Double quotes”

var print = “Thank you for your input”;

Concatenation of strings-

Concatenation is adding two or more strings together to creating a larger string. This is done using the + operator.

var message = ‘ wow’ + ‘JS is easy to learn’ + ‘and ’ + ‘easy to use’;

var firstName = “Alok”;

var lastName = “Kumar”;

var fullName = firstName + “ ” + lastName;

You can make sure the length of your string in JavaScript (length is the number of characters in string) just use the property *.length*

var sizeofString = ‘I Love India’.length;

Keywords-

There are some reserved words in every language which cannot be used as variables. JavaScript keywords are:

Abstract	arguments	boolean	Break
Byte	Case	catch	Char
Const	Continue	debugger	Default
Delete	Do	double	Else
Eval	False	Final	Finally
Float	For	function	Goto
If	implements	In	instanceof
Int	Interface	Long	Native

New	Null	package	Private
Protected	Public	return	short
Static	Switch	synchronized	this
Throw	Throws	transient	true
Try	Typeof	Var	void
Volatile	While	With	yield

Conditional Logic-

(i) If Statements

(ii) If…else Statements

(iii) Conditional operator

If Statements-

The 'if' statement is the basic control statement that permit JavaScript to make decisions and execute statements conditionally

Syntax

The syntax for a basic if statement is as follows:

```
if (expression){

   Statement(s) to be executed if expression is true

}
```

Here a JavaScript expression is evaluated. If the resulting value is true, the given statement(s) are executed. If the expression is false, then no statement would be not executed. Most of the times, you will use comparison operators while making decisions.

Example:

```
<html>
        <body>
                <script type= "text/JavaScript">
                var age = 20;
                if( age > 18 ){
                        document.write("<b>Qualifies for driving</b>");
                }
                </script>
        </body>

</html>
```

If…else Statements-

The 'if...else' statement is the next form of control statement that allows JavaScript to execute statements in a more controlled way.

Syntax

The syntax of an if-else statement is as follows:

```
if (/* expression */){
```

```
Statement executed if expression is true
} else {
Statement executed if expression is false
}
```

Here JavaScript expression is evaluated. If the resulting value is true, the given statement(s) in the 'if' block, are executed. If the expression is false, then the given statement(s) in the else block are executed.

Example:

```
<html>
        <body>
                <script type= "text/JavaScript">
                var age = 14;
                if( age > 18 ){
                        document.write("<b>Qualifies for driving</b>");
                }else{
                        document.write("<b>Does not qualify for driving</b>");
                }
                </script>
        </body>

</html>
```

Conditional operator-

In order to avoid the if-else hassle, simple logical comparisons can be utilised.

```
var adult = (age > 18) ? "YES" : "NO";
```

In the above example, ? is a logical operator. The code says that if the value of age is greater than 18, then adult = YES; otherwise, adult = NO.

Basically, if the condition is true, the first argument is accessed and if the comparison/condition is false, the second argument is accessed.

Looping-

Loops are used for statements that need to be repeated. One or more variable in the loop changes so that you can check condition to stop. You can run the same code again and again, each time with a different value. There are three types of loops.

(i) For loop

(ii) For … in loop

(iii) While loop

(iv) Do while loop

For loop-

The easiest form of a loop is *for statement*. This one has a syntax that is similar to *if statement*, but with more options:

Syntax-

```
for (statement 1; statement 2; statement 3) {
        code block to be executed
}
```

Statement 1- is executed before the loop starts.

Statement 2- defines the end condition when the loop had to stop.

Statement 3- is executed each time after the loop has been executed.

Example: for loop to print numbers from 1 to 5

```
<html>
        <body>
                <p id="output"></p>
                <script>
                        var text = "";
                        var i;
                        for (i = 0; i < 5; i++) {
                                text += “The number is “ + i + “<br>”;
                        }
                        document.getElementById(“output”).innerHTML = text;
                </script>
        </body>
</html>
```

For … in loop

A *for...in* loop is used to loop through an object’s properties. As we have not discussed Objects yet, you may not feel comfortable with this loop. But once you understand how objects behave in JavaScript, you will find this loop very useful.

Syntax

```
for (variablename in object)
{
    statement or block to execute
}
```

While loop-

While Loops repetitively execute a block of code as long as a specified condition is true.

```
while(condition){
   // do it as long as condition is true
}
```

Example: the loop in this example will repetitively execute its block of code as long as the variable *i* is less than 5:

```
var i = 0, x = “”;

while (i < 5) {

   x = x + “The number is ” + i;

   i++;

}
```

Do while loop-

The Do/While Loop is a variant of the while loop. This loop will execute the code block once before checking if the condition is true. It then repeats the loop as long as the condition is true:

```
do {
   // code block to be executed
} while (condition);
```

Note: Be careful to avoid infinite looping if the condition is always true!

Example: Print numbers less than 10 using do...while loop:

```
var i = 0;

do {

    document.write(i + “ “);

        i++; // incrementing i by 1

} while (i < 10);
```

Note: i = i + 1 can be written i++.

*Example***:** using this code you can access array with loop.

```
var myArray = [“aa”, “bb”];
// for-in loop
for (var i in myArray) {
        document.write(myArray[i]+”, ”); // output is : aa,  bb,
}
```

Functions

A function is a group of reusable code which can be called anywhere in your program. This eliminates the need of writing the same code again and again. It helps programmers in writing modular codes. Functions allow a programmer to divide a big program into a number of small and manageable functions. Like any other advanced programming language, JavaScript also supports all the features necessary to write modular code using functions. You must have seen functions like alert() and write() in the earlier chapters. We were using these functions again and again, but they had been written in core JavaScript only once.

JavaScript allows us to write our own functions as well. This section explains how to write your own functions in JavaScript.

(i) Function definition

(ii) Function calling

(iii) Function parameters/arguments

(iv) Return statement

Function definition-

Before we use a function, we need to define it. The most common way to define a function in JavaScript is by using the function keyword, followed by a unique function name, a list of parameters (that might be empty), and a statement block surrounded by curly braces.

Syntax:

```
<script type= “text/JavaScript”>
function functionname(parameter-list)

{

    statements

}

</script>
```

Example:

Try the following example. It defines a function called sayHello that takes no parameters:

```
<script type= “text/JavaScript”>
function sayHello()

{

    alert(“Hello there”);

}

</script>
```

Function calling-

Calling a Function To invoke a function somewhere later in the script, you would simply need to write the name of that function as shown in the following code.

```
<html>
        <head>
                <script type= “text/JavaScript”>
                        function sayHello()
                        {
                                document.write (“Hello there!”);
                        }
                </script>
        </head>
        <body>
                <p>Click the following button to call the function</p>
                <form>
                        <input type= “button” onclick= “sayHello()” value= “Say Hello”>
                </form>
        </body>

</html>
```

Function Parameter/Arguments-

Till now, we have seen functions without parameters. But there is a facility to pass different parameters while calling a function. These passed parameters can be captured inside the function and any manipulation can be done over those parameters. A function can take multiple parameters separated by comma. Example Try the following example. We have modified our “sayHello” function here. Now it takes two parameters.

```
<html>
        <head>
                <script type= “text/JavaScript”>
                        function message(name, age)
                        {
                                document.write (name + “ is “ + age + “ years old.");
                        }
                </script>
        </head>
        <body>
                <p>Click the following button to call the function</p>
                <form>
                        <input type= “button” onclick= “message(Neha, 7)” value= “some
message”>
                </form>
        </body>
        </html>
```

Return statement-

A JavaScript function can have an optional return statement. This is required if you want to return a value from a function. This statement should be the last statement in a function. For example, you can pass two numbers in a function and then you can expect the function to return their multiplication in your calling program.

Example: following function that takes two parameters and concatenates them before returning the resultant in the calling program.

```
<html>
        <head>
                <script type= "text/JavaScript">
                        function concatenate(first, last)
                        {
                                var full;
                        full = first + last;
                        return full;
                        }
                        function secondFunction()
                        {
                        var result;
                        result = concatenate(Neha, Pant);
                        document.write (result);
                        }
                </script>
        </head>
        <body>
                <p>Click the following button to call the function</p>
                <form>
                        <input type= "button" onclick= "secondFunction()" value= "Call
Function">
                </form>
        </body>
</html>
```

8.7 ARRAYS IN JAVASCRIPT

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.

Syntax-

Use the following syntax to create an Array Object. As-

var fruits = new Array(“apple”, “orange”, “mango”);

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. The maximum length allowed for an array is 4,294,967,295.

You can create array by simply assigning values as follows:

var fruits = [“apple”, “orange”, “mango”];

You will use ordinal numbers to access and to set values inside an array as follows.

fruits[0] is the 1st element

fruits[1] is the 2nd element

fruits[2] is the 3rd element

Properties of array-

(i) Length

(ii) Prototype

(iii) Methods (readymade functions) of array

Length-

Array length property returns an unsigned, 32-bit integer that specifies the number of elements in an array.

Syntax-

array.length

Return Value

Returns the length of an array.

Example:

```
<html>
<head>
<title>JavaScript Array length Property</title>
</head>
<body>
<script type= “text/JavaScript”>
        var arr = new Array( 10, 20, 30 );
        document.write(“arr.length is:” + arr.length);
</script>
</body>
</html>
```

Output-

```
arr.length is: 3
```

Prototype-

The property of prototype permits you to add properties and methods to any object *viz.* (Number, Boolean, String, Date etc).

Syntax-

object.prototype.name = value

Example:

```
<html>
        <head>
                <title>User-defined objects</title>
        <script type= “text/JavaScript”>
                function book(title, author)
                {
                this.title = title;
                this.author = author;
                }
        </script>
        </head>
        <body>
                <script type= “text/JavaScript”>
                var myBook = new book(“Programming in C”, “Kanitkar”);
        book.prototype.price = null;
        myBook.price = 100;
        document.write(“Book title is : ”  + myBook.title + “<br>”);
        document.write(“Book author is : ”+ myBook.author + “<br>”);
        document.write(“Book price is : ” + myBook.price + “<br>”);
                </script>
        </body>
</html>
```

Output:

```
Book title is : Programming in C
Book author is : Kanitkar
Book price is : 100
```

Methods (readymade functions) of array-

Method	Description
concat()	Returns a new array comprised of this array joined with other array(s) and/or value(s).
every()	Returns true if every element in this array satisfies the provided testing function.
filter()	Creates a new array with all of the elements of this array for which the provided filtering function returns true.
forEach()	Calls a function for each element in the array.
indexOf()	Returns the first (least) index of an element within the array equal to the specified value, or 1 if none is found.
join()	Joins all elements of an array into a string.
lastIndexOf()	Returns the last (greatest) index of an element within the array equal to the specified value, or 1 if none is found.

map()	Creates a new array with the results of calling a provided function on every element in this array.
pop()	Removes the last element from an array and returns that element
push()	Adds one or more elements to the end of an array and returns the new length of the array.
reduce()	Apply a function simultaneously against two values of array (from left-to-right) as to reduce it to a single value.
reduceRight()	Apply a function simultaneously against two values of array (from right-to-left) as to reduce it to a single value.
reverse()	Reverses the order of the elements of an array the first becomes the last, and the last becomes the first.
shift()	Removes the first element from an array and returns that element.
slice()	Extracts a section of an array and returns a new array.
some()	Returns true if at least one element in this array satisfies the provided testing function.
toSource()	Represents the source code of an object
sort()	Sorts the elements of an array.
splice()	Adds and/or removes elements from an array.
toString()	Returns a string representing the array and its elements.
unshift()	Adds one or more elements to the front of an array and returns the new length of the array.

Example:

Here is an example of array method concat().
JavaScript array concat() method returns a new array comprised of this array joined with two or more arrays.
Syntax-
array.concat(value1, value2, ..., valueN);
Parameter Details
valueN : Arrays and/or values to concatenate to the resulting array.
Return Value
Concatenated values of different arrays in the resulting array
Example:

```
<html>
        <head>
        <title>JavaScript Array concat Method</title>
        </head>
        <body>
                <script type="text/JavaScript">
                var alpha = [“a”, “b”, “c”];
                var numeric = [1, 2, 3];
                var alphaNumeric = alpha.concat(numeric);
                document.write(“alphaNumeric : ” + alphaNumeric );
                </script>
        </body>
</html>
```

Output:

```
alphaNumeric: a,b,c,1,2,3
```

Example:

array pop() method which returns top(last in) element.

```
<html>
        <body>
                <h2>JavaScript Array Methods</h2>
                <h2>pop()</h2>
                <p>The pop() method removes the last element from an array.</p>
                <p id= “prev”></p>
                <p id= “last”></p>
                <script>
                        var fruits = [“banana”, “apple”, “orange”, “mango”];
                        document.getElementById(“prev”).innerHTML = fruits;
                        fruits.pop();
                        document.getElementById(“last”).innerHTML = fruits;
                </script>
        </body>
</html>
```

8.8 OBJECTS IN JAVASCRIPT

It is an Object-Oriented Programming (OOP) language. A programming language can be called object-oriented if it provides four basic capabilities to developers.

(i) Encapsulation
(ii) Aggregation
(iii) Inheritance
(iv) Polymorphism

Encapsulation

The capability to store related information, whether data or methods, together in an object.

Aggregation

The capability to store one object inside another object.

Inheritance

The capability of a class to rely upon another class (or number of classes) for some of its properties and methods.

Polymorphism

The capability to write one function or method that works in a variety of different ways. Objects are composed of attributes. If an attribute contains a function, it is considered to be a method of the object; otherwise, the attribute is considered a property.

A good Example from our real life to understand an object. As- a car

A car has “properties” like weight and color, and “methods (functions)” like start and stop:

Properties- car.name=Alto, car.model = K10, car.color = white, car.type = hatchback

Methods- car.start(), car.accelerate(), car.brake(), car.stop()

Creation of objects

There are two ways to create an object in JavaScript: as-

(i) Literal

var object = {};

(ii) object-oriented

var object = new Object();

Object Properties

Object properties can be any of the three primitive data types, or any of the abstract data types, such as another object. Object properties are usually variables that are used internally in the object's methods, but can also be globally visible variables that are used throughout the page.

The syntax for adding a property to an object is:

objectName.objectProperty = propertyValue;

Example: The following code gets the document title using the "title" property of the document object.

var str = document.title;

Object Methods-

Methods are the functions that let the object do something or let something be done to it. There is a small difference between a function and a method – at a function is a standalone unit of statements and a method is attached to an object and can be referenced by the this keyword.

Methods are useful for everything from displaying the contents of the object to the screen to performing complex mathematical operations on a group of local properties and parameters.

Example: Following is a simple example to show how to use the write() method of document object to write any content on the document.

document.write (“This is test”);

Note: We will discuss objects in detail in next unit

8.9 POINTS TO REMEMBER

- JavaScript is a client-side as well as server-side scripting language.
- JavaScript is an Interpreted programming language.
- JavaScript is also an Object based Programming language.
- JavaScript code is place inside <script></script> tags.
- JavaScript is a *"loosely typed"* language, which means that you don't have to explicitly declare what type of data the variables are. You just need to use the var keyword to indicate that you are declaring a variable, and the interpreter will work out with its associated properties.
- JavaScript is case sensitive.
- You can define strings in JavaScript by enclosing the text in single quotes or double quotes. Ex: var name='Anika'; or var name="Anika";
- For … in statement is used to access object properties.
- You can find number of elements in array using array.length property.
- Functions can be defined using function keyword: function fn_name{}
- Object is an entity having some methods and some properties

8.10 GLOSSARY

- **ECMA Script:** It is a trademarked scripting-language specification standardized by ECMA International in ECMA-262 and ISO/IEC 16262. It was created to standardize JavaScript.
- **Compilation**: the translation of source code into object code by a compiler
- **Interpretation**: a process to translate source code into some efficient intermediate representation which can be immediately executed.

8.11 CHECK YOUR PROGRESS

Part- A (Fill in the blanks and True/False type questions)

a) We can change an html element attribute using …… method.

b) Variables are declared using … keyword.

c) Javascript variables can hold values of ………

d) We can check length of string var message as:

e) Syntax of logical comparison operator is:

f) pop() method of array removes the ….. element of array.

g) The code inside loop block for(i=1;i<5;i++) will be executed …. times.

h) alert() is an inbuilt ……. of javascript.

i) An object can be defined using …. Keyword.

j) Object.color is a (property/method).

k) document.write() is a (property/method)

l) Identifiers are combination of ………. and ………

m) Literals contain ….. values.

n) Break, continue, new are some JavaScript ………..

o) Boolean literals contain value 0 and 1 (True/False).

p) Interpretation of code is done at run time. (True/False)

q) Lowercase and uppercase variables are treated as same variables in javascript. (True/False)

r) By using Array.length method we can also find number of elements in array. (True/False)

s) The condition statement in a do while loop is written at the end of block. (True/False)

t) Object is a collection of property and methods. (True/ False)

Part- B (Descriptive type questions)

a) Inside which HTML element do we put the JavaScript?

b) What is the correct syntax for referring to an external script called "myscript.js"?

c) How can one create an Object in JavaScript?

d) How can you read properties of an Object in JavaScript?

e) How can you build an Array in JavaScript?

f) How to read elements of an array in JavaScript?

g) Which built-in method returns the length of the string?

h) Which built-in method removes the last element from an array and returns that element?

i) What is a method in JavaScript?

j) Define function in JavaScript.

Answers- (Part- A- Fill in the blanks and True/False type)		
(a) innerHTML	(b) var	(c) any type
(d) message.length	(e) var x = (condition)? True : false	(f) Last
(g) 4 times	(h) Function	(i) New
(j) property	(k) method	(l) Character and digits
(m) Fixed	(n) Keywords	(o) True
(p) True	(q) True	(r) True
(s) True	(t) True	

8.12 BIBLIOGRAPHY/REFERENCES

[1] www.w3schools.com/js/default.asp, January 2018

[2] www.tutorialspoint.com/javascript/index.htm, January 2018

[3] en.wikipedia.org/wiki/JavaScript, January 2018

[4] www.html-5-tutorial.com, January 2018

8.13 SUGGESTED READINGS

[1] Daggett Mark E. (2013) Expert Javascript, Apress, XIV, 248

UNIT 9

JAVASCRIPT EVENTS

9.1 INTRODUCTION

In previous unit we have learnt about basic programming constructs of JavaScript. Now we will discuss how to handle user driven events and reply accordingly. We will also discuss objects in detail.

In this unit you will learn how DHTML can be used to develop an interactive web page and cookies can be used to store user preferences.

When users trip your web, they do things like click on text and images and given links, hover over things etc. These are examples of what JavaScript calls events.

When page loads called an event. If the user clicks on a button, that click too is an event. Other examples include events like pressing any key, closing a window, resizing a window, etc.

9.2 OBJECTIVE

After successful completion of this unit, you will be able to-

- Create JavaScript events in a webpage.

- Create and access objects with relevant properties and methods.
- Validate input by user in a web page.
- Use cookies in a web page /project.
- Design interactive web pages using DHTML.
- Develop logical thinking while using logical constructs of JavaScript.

9.3 JAVASCRIPT EVENTS

Interaction of JavaScript with HTML is handled through events that occur when the user or the browser manipulates a page.

(i) onclick event
(ii) onsubmit event
(iii) onmouseover & onmouseout events
(iv) HTML5 standard events

onclick event-

“onclick” event is frequently used event. This event occurs when a user clicks the left button of his/her mouse. You can use/activate your validation or warning or etc., against this event type.

Example

```
<html>

    <head>

        <script type= “text/JavaScript”>

    <!-- function sayWelcome () {

                        alert(“Welcome”)

                }

        //-->   //comments placed to prevent execution of js if browser does not support it.

        </script>

        </head>

        <body>

                <p>Click on the button and get the result</p>

                <form>

                        <input type=”button” onclick= “sayWelcome()” value=”Say
Welcome” />

                </form>
        </body>

</html>
```

onsubmit event-

“onsubmit” event is generated when someone submits a form. You can create form validation action against “onsubmit” event.

Example:

The example given below show how to use “onsubmit” event. Here we are calling a validate() function. The validate() function called before submitting the form data to the webserver. If this function returns true, then the form data will be submitted to the webserver, otherwise it will not be submitted to the web server.
Try the following example.

```
<html>
        <head>
                <script type= “text/JavaScript”>
                <!--
                        function validate() {
                                all validation goes here
                                ........
                                return either true or false
                        }
                //-->
                </script>
        </head>
        <body>
        <form method= “POST” action= “submit.html” onsubmit= “return validate()”>
                 .......
           <input type= “submit” value= “Submit” />
        </form>
        </body>
</html>
```

onmouseover & onmouseout events-

The “onmouseover” and “onmouseout” events will help you to create effects with images as well as with text too. The “onmouseover” event triggers (occurs) when you move or bring mouse pointer over any element.

The “theonmouseout” event triggers (occurs) when you move or bring your mouse pointer out from that element. Example as given below-

```
<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>
```

HTML5 standard events-

The standard HTML 5 events are listed here for your reference. Here script indicates a JavaScript function to be executed against that event.

Attribute	Value	Description
Offline	script	Triggers when the document goes offline
Onabort	script	Triggers on an abort event
onafterprint	script	Triggers after the document is printed
onbeforeonload	script	Triggers before the document loads
onbeforeprint	script	Triggers before the document is printed
onblur	script	Triggers when the window loses focus
oncanplay	script	Triggers when media can start play, but might has to stop for buffering
oncanplaythrough	script	Triggers when media can be played to the end, without stopping for buffering
onchange	script	Triggers when an element change
onclick	script	Triggers on a mouse click

oncontextmenu	script	Triggers when a context menu is triggered
ondblclick	script	Triggers on a mouse double-click
ondrag	script	Triggers when an element is dragged
ondragend	script	Triggers at the end of a drag operation
ondragenter	script	Triggers when an element has been dragged to a valid drop target
ondragleave	script	Triggers when an element is being dragged over a valid drop target
ondragover	script	Triggers at the start of a drag operation
ondragstart	script	Triggers at the start of a drag operation
ondrop	script	Triggers when dragged element is being dropped
ondurationchange	script	Triggers when the length of the media is changed
onemptied	script	Triggers when a media resource element suddenly becomes empty.
onended	script	Triggers when media has reached the end
onerror	script	Triggers when an error occur
onfocus	script	Triggers when the window gets focus
onformchange	script	Triggers when a form change
onforminput	script	Triggers when a form gets user input
onhaschange	script	Triggers when the document has change
oninput	script	Triggers when an element gets user input
oninvalid	script	Triggers when an element is invalid
onkeydown	script	Triggers when a key is pressed
onkeypress	script	Triggers when a key is pressed and released
onkeyup	script	Triggers when a key is released
onload	script	Triggers when the document loads
onloadeddata	script	Triggers when media data is loaded

onloadedmetadata	script	Triggers when the duration and other media data of a media element is loaded
onloadstart	script	Triggers when the browser starts to load the media data
onmessage	script	Triggers when the message is triggered
onmousedown	script	Triggers when a mouse button is pressed
onmousemove	script	Triggers when the mouse pointer moves
onmouseout	script	Triggers when the mouse pointer moves out of an element
onmouseover	script	Triggers when the mouse pointer moves over an element
onmouseup	script	Triggers when a mouse button is released
onmousewheel	script	Triggers when the mouse wheel is being rotated
onoffline	script	Triggers when the document goes offline
onoine	script	Triggers when the document comes online
ononline	script	Triggers when the document comes online
onpagehide	script	Triggers when the window is hidden
onpageshow	script	Triggers when the window becomes visible
onpause	script	Triggers when media data is paused
onplay	script	Triggers when media data is going to start playing
onplaying	script	Triggers when media data has start playing
onpopstate	script	Triggers when the window's history changes
onprogress	script	Triggers when the browser is fetching the media data
onratechange	script	Triggers when the media data's playing rate has changed
onreadystatechange	script	Triggers when the ready-state changes
onredo	script	Triggers when the document performs a redo
onresize	script	Triggers when the window is resized
onscroll	script	Triggers when an element's scrollbar is being scrolled
onseeked	script	Triggers when a media element's seeking attribute is no longer true, and the seeking has ended

onseeking	script	Triggers when a media element's seeking attribute is true, and the seeking has begun
onselect	script	Triggers when an element is selected
onstalled	script	Triggers when there is an error in fetching media data
onstorage	script	Triggers when a document loads
onsubmit	script	Triggers when a form is submitted
onsuspend	script	Triggers when the browser has been fetching media data, but stopped before the entire media file was fetched
ontimeupdate	script	Triggers when media changes its playing position
objectsonundo	script	Triggers when a document performs an undo
onunload	script	Triggers when the user leaves the document
Onvolumechange	script	Triggers when media changes the volume, also when volume is set to "mute"
Onwaiting	script	Triggers when media has stopped playing, but is expected to resume

9.4 JAVASCRIPT OBJECTS AND PROPERTIES

JavaScript is designed on a simple object-based paradigm. An object is a collection of properties, and a property is an association between a name (or *key*) and a value.

Objects are variable that contain several values. JavaScript objects are containers for named values, called properties and methods.

In JavaScript, almost "everything" is an object.

- Booleans can be objects (if defined with the new keyword)
- Numbers can be objects (if defined with the new keyword)
- Strings can be objects (if defined with the new keyword)
- Dates are always objects
- Arrays are always objects
- Functions are always objects
- Objects are always objects
- All JavaScript values, except primitives, are objects.

An empty object can be created using one of two syntaxes:

a. let myuser = new Object(); // "object constructor" syntax, almost like Java
b. let myuser = {}; // "object literal" syntax

Object properties-

We can immediately put some properties into {...} as “key: value” pairs:

```
let user = {                    // an object
name: “Meha”,          // key “name” contains value “Meha”
        age: 23                 // key “age” contains value 23
};
```

A property has a key (also known as “name” or “identifier”) before the colon “:” and a value to the right of it. In the user object, there are two properties:

a. The first property has the name “name” and the value “Meha”.

b. The second one has the name “age” and the value 23.

Example:

```
<!DOCTYPE html>
<html>
    <body>
        <p>building a JavaScript Object.</p>
        <p id=”show”></p>
        <script>
            var employee ={
                firstName: “Meha”,
                age: 23,
                designation: “Programmer”,
                address: “Delhi”
            };
            document.getElementById(“show”).innerHTML =
employee.firstName + “ is a ” + employee.designation + “ and “ + employee.age + “ years
old.”;
        </script>
    </body>
</html>
```

Output: Meha is a Programmer and 23 years old.

Example:

```
<!DOCTYPE html>
<html>
<body>
<p>
There are two different ways to access an object property:
</p>
<p>You can use person.property or person["property"].</p>
<p id="show"></p>

<script>
var employee ={
firstName: “Meha”,
age: 23,
designation: “Programmer”,
address: “Delhi”
};
```

```
document.getElementById("show").innerHTML = employee.firstName + " is a " +
employee.designation + " and " + employee.age + " years old." + "\n";
document.getElementById("show").innerHTML = employee["firstName"] + " is a " +
employee["designation"] + " and " + employee["age"] + " years old." + "\n";
</script>
</body>
</html>
```

Output:

```
Meha is a Programmer and 23 years old.
Meha is a Programmer and 23 years old
```

Object methods-

Methods are the functions that let the object do something or let something be done to it. There is a small difference between a function and a method – at a function is a standalone unit of statements and a method is attached to an object and can be referenced by this keyword.

Methods are useful for everything from displaying the contents of the object to the screen to performing complex mathematical operations on a group of local properties and parameters.

Example:

Following is a simple example to show how to use the write() method of document object to write any content on the document.

```
document.write ("This is text is displayed using JavaScript document.write method");
```

User-defined Objects-

In spite of in built objects you can build your own personalized objects in JavaScript. All user-defined objects and built-in objects are descendants of an object called Object.

The *new* operator is used to create an instance of an object. To create an object, the new operator is followed by the constructor method.

In the following example, the constructor methods are Object(), Array(), and Date(). These constructors are built-in JavaScript functions.

a. var employee = new Object();
c. var cars = new Array("Maruti", "Tata", "Hyundai");
d. var day = new Date("November 17, 2010");

The object() constructor-

A constructor is a function that creates and initializes an object. JavaScript provides a special constructor function called Object() to build the object. The return value of the Object() constructor is assigned to a variable.

The variable contains a reference to the new object. The properties assigned to the object are not variables and are not defined with the var keyword.

Example 1: How to build an Object.

```
<html>
  <head>
    <title>Building User-defined objects</title>
    <script type= "text/JavaScript">
        var book = new Object();   // Create the object
        book.title = "JavaScript"; // Assign properties to the object
        book.publisher  = "Pearson";
    </script>
  </head>
 <body>
    <script type= "text/JavaScript">
```

```
        document.write(“Book Title is : ” + book.title + “<br>”);
        document.write(“Book Publisher is : ” + book.publisher + “<br>”);
    </script>
  </body>
</html>
```

Output:
Book Title is : JavaScript
Book Publisher is : Pearson

Example 2:

Build an object using function and this keyword

This example demonstrates how to create an object with a User-Defined Function. Here this keyword is used to refer to the object that has been passed to a function.

```
<html>
  <head>
    <title>User-defined objects using function and this keyword</title>
      <script type= “text/JavaScript”>
        function book(title, publisher){
                        this.title = title;
                        this.publisher  = author;
        }
      </script>
    </head>
  <body>
    <script type= “text/JavaScript”>
         var myBook = new book(“JavaScript”, “Pearson”);
        document.write(“Book Title is : ” + myBook.title + "<br>");
        document.write(“Book Publisher  is : ” + myBook.publisher  + "<br>");
    </script>
  </body>
</html>
```

Output:
Book Title is : JavaScript
Book Publisher is : Pearson

Example 3:

Accessing objects using for … in loop

```
<html>
<body>
<h2>JavaScript object</h2>
<p>The for .. in statement to access properties of an object.</p>
<p id= “output”></p>
<script>
    var txt = “”;
    var book = {title: “JavaScript”, publisher: “Pearson”, price: 350};
    var info;
    for (info in book) {
                txt = txt + book[info] + “ ”;
    }
    document.getElementById(“output”).innerHTML = txt;
</script>
```

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

Output:

```
JavaScript object
The for .. in statement to access properties of an object.
JavaScript Pearson 350
```

9.5 CLIENT-SIDE VALIDATIONS

Form validation normally used to occur at the server, after the client had entered all the necessary data and then pressed the Submit button. If the data entered by a client was incorrect or was simply missing, the server would have to send all the data back to the client and request that the form be resubmitted with correct information. This was really a lengthy process which used to put a lot of burden on the server. JavaScript provides a way to validate form's data on the client's computer before sending it to the web server. Form validation generally performs two functions.

- Basic Validation - First of all, the form must be checked to make sure all the mandatory fields are filled in. It would require just a loop through each field in the form and check for data.
- Data Format Validation - Secondly, the data that is entered, must be checked for correct form and value. Your code must include appropriate logic to test correctness of data.

Example:

We will take an example to understand the process of validation. Here is a simple form in html format.

```
<html>
<head>
        <title>Form Validation</title>
        <script type= “text/JavaScript”>
        // Form validation code will come here.
        </script>
</head>
<body>
<form action= “/php/test.php” name= “myForm” onsubmit= “return validate();”>
<table cellspacing= “2” cellpadding= “2” border= “1”>
<tr>
<td align=“right”>Name</td>
<td><input type=“text” name=“txtName” /></td>
</tr>
<tr>
<td align=“right”>EMail</td>
<td><input type=“text” name=“txtEmail” /></td>
</tr>
<tr>
<td align=“right”>Zip Code</td>
<td><input type=“text” name=“Zip” /></td>
</tr>
<tr>
<td align=“right”>Country</td>
```

```
<td>
<select name="cmbCountry">
                    <option value="-1" selected>[choose yours]</option>
                    <option value="1">USA</option>
                    <option value="2">UK</option>
                    <option value="3">INDIA</option>
</select>
</td>
</tr>
<tr>
<td align="right"></td>
<td><input type="submit" value="Submit" /></td>
</tr>
</table>
</form>
</body>
</html>
```

Basic Form Validation-

First let us see how to do a basic form validation. In the above form, we are calling validate() to validate data when onsubmit event is occurring. The following code shows the implementation of this validate() function.

Example:

```
<script type="text/JavaScript">
          function validate()
          {
                    if( document.myForm.Name.value == "" )
                    {
                              alert( "your name please!" );
                              document.myForm.Name.focus() ;
                              return false;
                    }
                    if( document.myForm.EMail.value == "" )
                    {
                              alert( "Email please!" );
                              document.myForm.EMail.focus();
                              return false;
                    }
                    if(document.myForm.Zip.value == "" || isNaN( document.myForm.Zip.value
) || document.myForm.Zip.value.length != 5)
                    {
                              alert( "Please provide a zip in the format #####." );
                              document.myForm.Zip.focus() ;
                              return false;
                    }
                    if( document.myForm.Country.value == "-1" )
                    {
                              alert( "Please provide your country!" );
                              return false;
                    }
           return( true );
```

```
	}
</script>
```

Data Format Validation-

So, we will see how you can validate your entered form data before submitting it to the web server. The following example shows how to validate an entered email address. An email address must contain at least a '@' sign and a dot (.). Also, the '@' must not be the first character of the email address, and the last dot must at least be one character after the '@' sign.

Example:

```
<script type="text/JavaScript">
	function validateEmail()
	{
		var emailID = document.myForm.EMail.value;
		atpos = emailID.indexOf("@");
		dotpos = emailID.lastIndexOf(".");
		if (atpos < 1 || ( dotpos - atpos < 2 ))
		{
			alert("Please enter correct email ID")
			document.myForm.EMail.focus() ;
			return false;
		}
		return( true );
	}
</script>
```

9.6 COOKIES

Servers and web-browsers 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 cookies work?

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.

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 attribute named *expires* 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:

It sets a customer name in an input cookie.

```
<html>
    <head>
        <script type=“text/JavaScript”>
        <!--
            function WriteCookie()
            {
                if( document.myform.customer.value == ““ )
                {
                    alert (“Enter some value!”);
                    return;
                }
                cookievalue= escape(document.myform.customer.value) + “;”;
                document.cookie=“name=“ + cookievalue;
                document.write (“Setting Cookies : “ + “name=“ + cookievalue
                );
            }
        //-->
        </script>
    </head>
    <body>
        <form name=“myform” action=“”>
            Enter name: <input type=“text” name=“customer”/>
            <input type=“button” value=“Set Cookie” onclick=“WriteCookie();”/>
        </form>
    </body>
</html>
```

Output:

Enter name:

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.

You can use strings’ split() function to break a string into key and values as follows:

Example:

Following example to gets all the cookies.

```
<html>
    <head>
        <script type=“text/JavaScript”>
        <!--
            function ReadCookie()
            {
                var allcookies = document.cookie;
                document.write (“All Cookies : ” + allcookies );
                // Get all the cookies pairs in an array
                cookiearray = allcookies.split(‘;’);
                // Now take key value pair out of this array
                for(var i=0; i<cookiearray.length; i++){
                    name = cookiearray[i].split(‘=’)[0];
                    value = cookiearray[i].split(‘=’)[1];
                    document.write (“Key is : ” + name + “ and Value is :
” + value);
                }
            }
        //-->
        </script>
    </head>
    <body>
        <form name="myform" action="">
            <p> click the following button and see the result:</p>
            <input type="button" value="Get Cookie"
onclick="ReadCookie()"/>
        </form>
    </body>
</html>
```

Note: Here length is a method of Array class which returns the length of an array. We have discussed Arrays in unit 1.

Output:

click the following button and see the result:

Note: There may be some other cookies already set on your machine. The above code will display all the cookies set on your machine.

Setting expiry of cookies-

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 type="text/JavaScript">
        <!--
            function WriteCookie()
            {
                var now = new Date();
                now.setMonth( now.getMonth() + 1 );
                cookievalue = escape(document.myform.customer.value) + ";"
                document.cookie="name=" + cookievalue;
                document.cookie = "expires=" + now.toUTCString() + ";"
                document.write ("Setting Cookies : " + "name=" + cookievalue );
            }
        //-->
        </script>
        </head>
        <body>
```

```
                <form name="formname" action="">
                        Enter name: <input type="text" name="customer"/>
                        <input type="button" value="Set Cookie"
onclick="WriteCookie()"/>
                </form>
        </body>
</html>
```

Output:

Enter Cookie Name:

Deleting cookies-

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:

Try the following example. It illustrates how to delete a cookie by setting its expiry date to one month behind the current date.

```
<html>
        <head>
                <script type="text/JavaScript">
                <!--
                        function WriteCookie()
                        {
                                var now = new Date();
                                now.setMonth( now.getMonth() - 1 );
                                cookievalue = escape(document.myform.customer.value) +
";"
                                document.cookie="name=" + cookievalue;
                                document.cookie = "expires=" + now.toUTCString() + ";"
                                document.write("Setting Cookies : " + "name=" +
cookievalue );
                        }
                //-->
                </script>
        </head>
        <body>
                <form name="formname" action="">
                        Enter name: <input type="text" name="customer"/>
                        <input type="button" value="Set Cookie"
onclick="WriteCookie()"/>
                </form>
        </body>
</html>
```

Output:

Enter Cookie Name: □

9.7 DHTML

With DHTML, you can easily add effects to your web pages. Basically, DHTML is a combination of HTML, JavaScript, DOM (Document Object Model), and CSS (Cascade Style Sheet). It is not a language it stands for Dynamic HTML.

(i) HTML – it supports JavaScript, CSS and DOM.
(ii) JavaScript - scripting standard for HTML. Manipulates elements using events.
(iii) HTML DOM - The HTML DOM defines a standard set of objects for HTML, and a standard way to access and manipulate them.
(iv) CSS – allows us to style the layout of web pages.

Note: We described HTML and CSS in CWDD 101, Unit 6 and 7 respectively.

The function of a web browser (like Internet Explorer or Firefox, etc) is to read HTML documents and display them as web pages. The browser does not display the HTML tags, but uses the tags to interpret Content of the web page.

For Example:

```
<html>
        <body>
                <h1>A Heading</h1>
                <p>A paragraph.</p>
        </body>
</html>
```

Explanation:

- The text between <html> and </html> describes the web page
- The text between <body> and </body> is the visible page content
- The text between <h1> and </h1> is displayed as a heading
- The text between <p> and </p> is displayed as a paragraph

HTML DOM (Document Object Model)-

It is the Document Object Model for HTML. It is a standard for how to get, change, add, or delete HTML elements.

HTML DOM methods are **actions** you can perform (on HTML Elements).
HTML DOM properties are **values** (of HTML Elements) that you can set or change.

Example:

The following example changes the content (the innerHTML) of the <p> element with id="show":

```
<html>
    <body>
    <p id="show"></p>
        <script>
    document.getElementById(“show”).innerHTML = “Hello World!”;
    </script>
    </body>
</html>
```

In the example above, getElementById is a **method**, while innerHTML is a **property**.

“getElementById” method-

The most common way to access an HTML element is to use the id of the element. In the example above the getElementById method used id=“demo” to find the element

“innerHTML” property-

a. The easiest way to get the content of an element is by using the innerHTML property.

b. The innerHTML property is useful for getting or replacing the content of HTML elements.

c. The innerHTML property can be used to get or change any HTML element, including <html> and <body>.

JavaScript and HTML DOM-

A JavaScript can also be used to change the content or attributes of HTML elements. To modify the content of an HTML element:
document.getElementById(id).innerHTML=new HTML
To change the attribute of an HTML element:
document.getElementById(id).attribute=new value
Example: The Below example changes the content of an h1 element:

```
<html>
        <body>
                <h1 id=“header”>Watch me changing</h1>
                <script type=“text/JavaScript”>
                        document.getElementById(“header”).innerHTML=“I am New Text”;
                </script>
        </body>
</html>
```

- The HTML document above contains an h1 element with id=“header”
- We use the HTML DOM to get the element with id=“header”
- A JavaScript changes the content (innerHTML) of that element

Note- You also have learnt about JavaScript in CWDD 102, Unit 1.

JavaScript and CSS-

A JavaScript can also change the style of HTML elements. To change the style of an HTML element:
document.getElementById(id).style.property=new style
Example: Below example changes color of heading

```
<html>
```

```
        <body>
                <h1 id=“header”
onclick=“document.getElementById(‘header’).style.color=‘red’“>
                Click here to change my color
                </h1>
        </body>
</html>
```

Explanation:

- The HTML document above contains an h1 element with id=“header”
- We use the HTML DOM to get the element with id=“header”
- A JavaScript changes the style.color of that element

Example: Here is an example of DHTML, using JavaScript onclick event and CSS style property.

```
<html>
        <body>
                <p id=“para”>This ia an example of DHTML. You can add effects and
dynamically change your page style using JavaScript events and CSS.<br />
You are now capable to learn methods that can be used to design a dynamic and interactive
web page. Welcome to the world of Magic…
                </p>
                <input type=“button” value=“Hide text”
onclick=“document.getElementById(‘para’).style.visibility=‘hidden’“ />
                <input type=“button” value=“Show text”
onclick=“document.getElementById(‘para’).style.visibility=‘visible’“ />
        </body>
</html>
```

9.8 POINTS TO REMEMBER

- JavaScript handles user responses with the help of events.
- Onclick, onsubmit, onmouseover, onmouseout are some useful events.
- Object is a collection of property and methods.
- You can create object as: object = {key:value, key:value…..};
- The new operator is used to create an instance of an object.
- This operator can be used to initialize object properties.
- JavaScript events are used to validate form and data.
- A JavaScript can be used to store and retrieve information on the visitor’s computer means it can also be used to create cookies.
- Cookies are a simple plain text data record of 5 variable-length fields.
- document.cookie object is used to store cookie values.

- DHTML is a combination of HTML, JavaScript, DOM, and CSS. It is not a language it stands for Dynamic HTML.
- HTML stands for Hyper Text Markup Language
- CSS stands for Cascading Style Sheets
- The Document Object Model of HTML is a standard to get, change, add, or delete HTML elements.
- Dynamically you can change style of HTML by JavaScript.

9.9 GLOSSARY

- **DHTML**: It stands for Dynamic hypertext transfer protocol. It is a Combination of HTML, CSS and JavaScript.
- **HTML 5**: HTML 5 is a revision of the Hypertext Markup Language (HTML), the standard programming language for describing the contents and appearance of Web pages.
- **Constructor**: In class-based object-oriented programming, a constructor is a special type of method called to create an object.
- **HTTP**: The Hypertext Transfer Protocol (HTTP) is an application protocol for World Wide Web.
- **Stateless Protocol**: HTTP is a stateless protocol, which means that the connection between the browser and the server is lost once the transaction ends
- **Cookies:** Cookies are small files which are stored on a user's computer. They are designed to hold a modest amount of data specific to a particular client and website, and can be accessed either by the web server or the client computer.
- **CGI programming:** CGI is the abbreviation of Common Gateway Interface. ... A CGI program is any program designed to accept and return data that conforms to the CGI specification. The program could be written in any programming language, including C, Perl, Java, or Visual Basic.

9.10 CHECK YOUR PROGRESS

Part- A (Fill in the blanks and True/False type)

(1) JavaScript's interaction with HTML is handled through …........

(2) On submit event is called when we …………… form ……… to server.

(3) .... event is triggered when the HTML element is changed.

(4) Name two events that are triggered when keyboard is used.

(5) JavaScript …… are containers for property and methods.

(6) Object …........... are also known as keys.

(7) [function]Methods are ……….. that let the object do something.

(8) .... operator is used to create an instance of object.

(9) ……………… is done to make sure that all mandatory fields are filled.

(10) Final validation is mostly done at ………. Event.

(11) ……………. is done to ensure that valid data is entered.

(12) Cookies are data that ………. has sent to client's computer.

(13) Name the 5 variable length fields of cookies.

(14) JavaScript Cookies are stored inside …….. object.

(15) Cookies are store as ……… pairs separated by semicolon.

(16) Cookies are deleted by setting the expiry date to …………….

(17) JavaScript is a combination of ………..

(18) HTML is a …….. language

(19) CSS stands for ……

(20) HTML DOM methods are ……. we perform on HTML elements.

(21) ..... is the syntax to change the style of an element.

(22) JavaScript is used to validate data at client's computer before sending it to server. (True/False)

(23) Page load is also an event. (True/False).

(24) On click event can be used for page validation. (True/ False).

(25) CSS selector is an HTML element. (True/False)

(26) getElementById is a method. (True/False)

(27) innerHTML property is used to change or get the contents of elements. (True/ False)

(28) Cookies re plain text files. (True/ False)

(29) JavaScript arrays are objects. (True/False)

(30) We can create object as var stud = new object(); (True/False)

Part- B (Descriptive type)

(a) Which event is fired when a document and all of its external resources are fully loaded and displayed to the user?

(b) What are the Event Handlers in JavaScript?

(c) Which syntax is used to describe elements in CSS?

(d) Which of the following can be used to select HTML elements based on the value of their name attributes?

(e) Explain JavaScript objects.

(f) What is JavaScript standard objects?

(g) What is InnerHTML Property in JavaScript?

(h) What are Cookies in JavaScript?

Answers- (Part- A- Fill in the blanks and True/False type)		
1- Events	2- submit, data	3- onchange

4- Onkeypress and onkeyup	5- Objects	6- Properties
7- function	8- new	9- Basic validation
10- Form submit	11- Data Validation	12- Server
13- Expires, domain, path, secure, name=value	14- Document.cookie	15- Name=value
16- A time in past	17- HTML, JavaScript, DOM and CSS	18- markup
19- cascading style sheet	20- action	21- document.getElement ById(id).style.property= new style
22- True	23- True	24- True
25- True	26- True	27- True
28- True	29- True	30- True

9.11 BIBLIOGRAPHY/REFERENCES

- www.w3schools.com/js/js_events.asp, January 2018
- www.tutorialspoint.com/javascript/javascript_events.htm, January 2018
- www.html-5-tutorial.com, January 2018

9.12 SUGGESTED READINGS

- Duckett Jon (2014) Javascript & JQuery: Interactive Front-End Web Development, Wiley, 640.