MCA S1 04

Computer Network & Web Technology

SEMESTER I

MASTER OF COMPUTER APPLICATION

BLOCK - 2



KRISHNA KANTA HANDIQUI STATE OPEN UNIVERSITY

Subject Experts

Prof. Anjana Kakati Mahanta,	Gauhati University
Prof. (Retd.) Pranhari Talukdar,	Gauhati University
Prof. Sikhar Kumar Sarma,	Cotton University
Dr. Jyotiprokash Goswami,	Assam Engineering College

Course Coordinator

Dr. Sanjib Kr. Kalita, Associate Professor, Dept. of Computer Science, KKHSOU

Dr. Tapashi Kashyap Das, Associate Professor, Dept. of Computer Science, KKHSOU

Sruti Sruba Bharali, Associate Professor, Dept. of Computer Science, KKHSOU

SLM Preparation Team

Units	Contributors
8,9	Arabinda Saikia, KKHSOU
10,11,12,13,14,15	Sankhapani Bharali, Girls' Polytechnique

Editorial Team

Content	Dr. Jyotiprokash Goswami, Assam Engineering College
Language	Prof. (Retd.) Robin Goswami, Cotton College

Structure, Format & Graphics: Dr. Sanjib Kalita, KKHSOU

May, 2018

This Self Learning Material (SLM) of the Krishna Kanta Handiqui State Open University is made available under a Creative Commons Attribution-Non Commercial-Share Alike 4.0 License (international): http://creativecommons.org/licenses/by-nc-sa/4.0/

Printed and published by Registrar on behalf of the Krishna Kanta Handiqui State Open University.

Headquarters : Patgaon, Rani Gate, Guwahati - 781017
Housefed Complex, Dispur, Guwahati-781006; Web: www.kkhsou.in

The University acknowledges with thanks the financial support provided by the **Distance Education Bureau**, **UGC** for the preparation of this study material.

BLOCK INTRODUCTION

This is the Second Block of the course ‘‘Fundamentals of Networking and wes Technology’’. This Block contains 8 units. These units discuss the basic concept of Internet and Web Technology. The Unit 8 deals with the basic concept of Internet. Architecture like Internet Architecture, Client Server model, IP address URL etc are discussed in this unit. Unit 9 introduces principle of World Wide Web like browser, search engine etc..The basic of HTML and HTML tags are discussed in Unit 10. Advanced HTML tags, Forms are discussed in Unit 11. Unit 12 discusses the HTML links, styles, hypertext, hyperlink,hypermediadata etc. The concept of XML and DTD are discussed in Unit 13. Unit 14 and 15 discusses the concept of java script . Different control statements array, java script objects are also discussed in these two units.

MASTER OF COMPUTER APPLICATIONS
Fundamentals of Networking and Web Technology

DETAILED SYLLABUS

BLOCK-2
Semester I

UNIT 8: Internet [8 marks]

Internet - A Definition, Internet Architecture, Peer-to-Peer, Client-Server, Accessing Internet, Dial-Up Connections, ADSL Connections, Cable Connections, Internet Service Providers, Organization of Internet, Internet Protocol Suite, IP Address, Domain Name System, Uniform Resource Locators (URL), Application of Internet

UNIT 9: World Wide Web [7 marks]

Web Page, Web Browsers, Web Server, Web Search Engines, Web Cookies, Technologies in WWW

UNIT 10: Introduction to HTML [6 marks]

Basics of HTML, HTML Tags, HTML Editor, Document Structure Tags , Formatting Tags, List Tags, Hyperlink and Image Tags, Table Tags, Frame Tags, Form Tags

UNIT 11: Advanced HTML Tags [6 marks]

HTML FORM, Additional Advanced HTML Tags

UNIT 12: HTML Links and Styles [5 marks]

Hypertext, Hyperlink and Hypermedia, Anchor and URL, Creating tables, Frames, Forms, E-mailing

UNIT 13: Introduction to XML [5 marks]

Introduction, Comparison of HTML and XML, XML Basics, Introduction to DTD

UNIT 14: Java Script [5 marks]

Introduction, Client Side Programming, Introduction to JavaScript, Advantages of JavaScript, Basic Programming Concepts, Variables, Data-Types, Operators, Control Structures in JavaScript, Branching, Looping, Array, Dense Array, Array Within Array, Methods of Array, Functions: User-Defined Function, Built-in Function, Dialog Box

UNIT 15: Forms and Objects [8 marks]

Introduction, JavaScript Form Object, Elements of Form Object, Properties of Form Object, Events of Form Elements, Methods of Form and Form Elements, Built-in Objects in JavaScript, String Object, Math Object, Date Objects, User Defined Objects in JavaScript, Creating User Defined Objects, Creating Objects Within Objects

*****

UNIT 8 : INTERNET

UNIT STRUCTURE

8.1 Learning Objectives

8.2 Introduction

8.3 Internet - A Definition

8.4 Internet Architecture

- 8.4.1 Peer-to-Peer
- 8.4.2 Client-Server

8.5 Accessing Internet

- 8.5.1 Dial-Up Connections
- 8.5.2 ADSL Connections
- 8.5.3 Cable Connections

8.6 Internet Service Providers

8.7 Organization of Internet

- 8.7.1 Internet Protocol Suite
- 8.7.2 IP Address
- 8.7.3 Domain Name System
- 8.7.4 Uniform Resource Locators

8.8 Application of Internet

8.9 Let Us Sum Up

8.10 Answers to Check Your Progress

8.11 Further Reading

8.12 Model Questions

8.1 LEARNING OBJECTIVES

After going through this unit, you will be able to :

- define Internet and describe the growth of Internet
- describe the organization of Internet
- describe web servers, web browsers and search engines
- learn about server and client side technology

8.2 INTRODUCTION

Ever since the advent of computers, researchers have indulged in a continuous quest to bring the world closer through computers. This effort gave rise to the Internet. It is through the Internet that today millions of people communicate and share information, regardless of their location. This unit will focus on the basics of the Internet and also shed light upon how the Internet works along with its applications.

8.3 INTERNET – A DEFINITION

The Internet can be defined as a network of globally connected computers that is decentralized by design. This definition can be broken down into three parts. Let us understand each part of the definition in isolation.

It is a network. A network is a collection of computers. The Internet can also be referred to as a network because it is a collection of millions of computers.

Globally connected computers. This means that you can be connected to the Internet, regardless of your location. The Internet has brought people in the world closer by connecting computers located in the remotest of locations.

Decentralized design. The Internet has a decentralized design. That is, there is no centralized body that controls the way in which the Internet functions. The Internet does provide online services that are centrally administered, but as a whole, it would not be incorrect to say that the Internet has a decentralized design. Each computer connected to the Internet is called a *host*. The operator/ user of a particular host can choose from the millions of available Internet services and can also make services available through the Internet.

You can consider Internet to have the following characteristics :

- *A complex network* - with simplified definition as a 'network of network'

- *Disorganized* - Internet can be cumbersome and confusing, even for experienced users
- *A decentralized system* - millions of individual networks and over 200 million individual computers connected through the world
- *Composed of many billions of files*(web pages).
- *Dynamic* - changing every minute of every day. On an average, a new network is connected to the Internet every 30 minutes.
- *Expanding exponentially* - the Internet is growing at the rate not less than 15% per month.

8.4 INTERNET ARCHITECTURE

It is important to understand what the term “architecture” means. The notion of network architecture was introduced during the Internet research phase by the research community that had developed the ARPAnet protocols. Network architecture is a set of high-level design principles that guides the technical design of the network, especially the engineering of its protocols and algorithms. There are two most commonly used architecture in Internet technology : *peer-to-peer* and *client server architecture*.

8.4.1 Peer-to-Peer

Peer-to-peer is a communication model in which each and every node is capable of sharing information and can initiate a communication session. On the Internet, peer-to-peer (referred to as P2P) is a type of transient Internet network that allows a group of computer users with the same networking program to connect with each other and directly access files from one another's hard drives. **Napster** and **Gnutella** are examples of this kind of peer-to-peer software.

8.4.2 Client-Server

The Client-Server Architecture is based on the principle where

the client computer requests for some data and the data are sent by the server computer through the network. The concept of *client/ server* computing has particular importance on the Internet because most of the programmes are built using this design. Servers are powerful computers or processes dedicated to managing disk drives (file servers), printers (print servers), or network traffic (network servers). Clients are PCs or workstations on which users run applications. Clients rely on servers for resources, such as files, devices, and even processing power.

The following figure shows the two architectures :

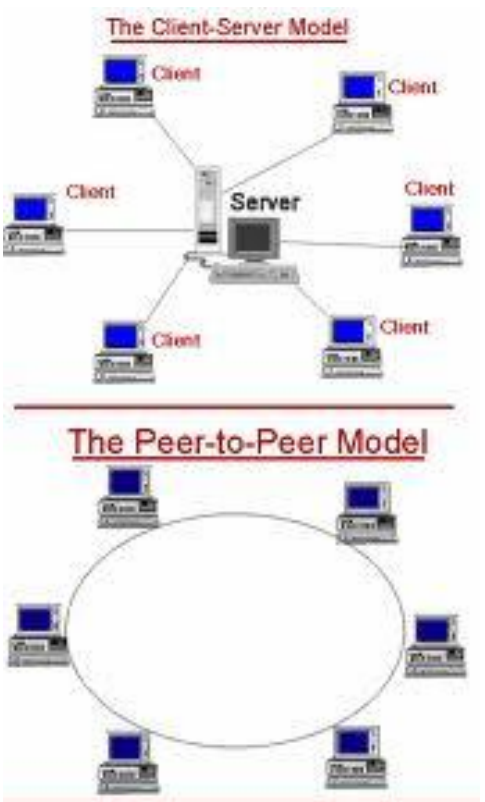


Fig. 8.1 : Client Server and peer-to-peer architecture

8.5 ACCESSING INTERNET

In order to access the Internet, one requires the following:

- A client computer with necessary hardware and client software
- The connectivity through Internet Service Provider (ISP) for flow of data
- Host computer(s) which is also called Web server hosting the

desired data.

The necessary hardware devices that must be had on the client computer are :

- Processor of Pentium (Intel based) or AMD or Macintosh
- Network Interface Card (NIC)
- Modem, it can be an external or internal (fitted inside the computer)

The necessary softwares that must be found on the client computer are :

- Operaring system like WIndows, UNIX, LINUX or OS/2 etc.
- Web browser like Netscape, Internet explorer, FireFox etc. (it must be compitable with the operating system)

Common methods of Internet access in homes include **dial-up, landline broadband(ADSL)** (over coaxial cable, fibre optic or copper wires), **Wi-Fi, satellite and 3G technology cell phones**. We will concentrate basically on the following methods

- dial-up connections
- ADSL connections and
- Cable Connections

8.5.1 Dial-Up Connections

Dial-up connections is a type of Internet connectivity that operates through a standard telephone line. Before a person can subscribe to a dial-up service, he or she must have a computer and dial-up modem. A telephone line feeds into the modem. The modem is controlled by software in the computer; for example, the Network Connections utility that comes with Microsoft Windows operating systems. Here you can setup a profile for the ISP (Internet Service Provider, like BSNL), which will tell the modem what phone number to call and how to communicate with the dial-up service. The ISP itself provides this information.

Upon joining a dial-up service, the subscriber chooses a username and password. Once the modem calls the phone number

and makes a connection, a "*handshake*" takes place in which information is exchanged between the computer modem and the remote server. The username and password are supplied by the modem. This grants the user an access through the dial-up gateway to the Internet. Dial-up connections can be very economic and are widely available, the cost per minute is comparable to that of a local phone call, or priced as a monthly plan which will include a certain amount of time. As these connections use a standard modem the hardware costs are minimal. Dial-up connections are very slow compared to other connection types. When connected to the internet the same phone line cannot be used for phone calls, so if anyone calls you when you are connected, they will get the busy signal.

Dial-up connections transfer data over an analogue line, so before the data is sent it has to be converted from digital to analogue, Likewise, when data is received it has to be converted from analogue to digital (this is what the modem does), This adds a performance overhead which affects the speed of the connection.

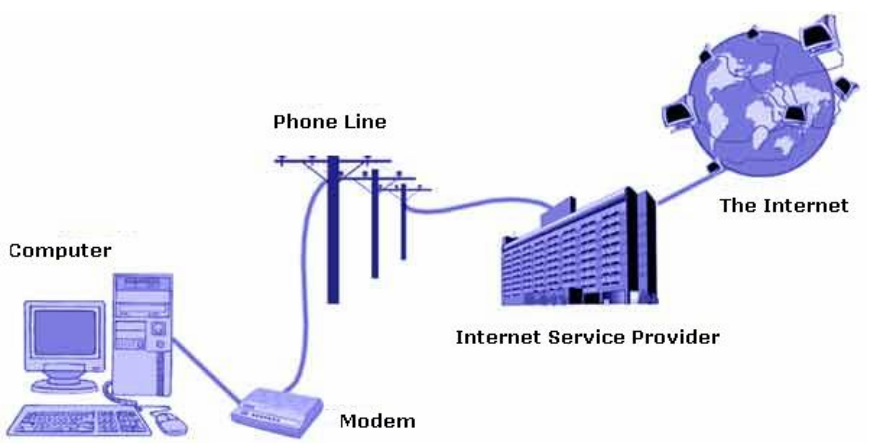


Fig. 8.2 : Dial-up Connection

8.5.2 ADSL Connections

ADSL (Asymmetric Digital Subscribers Line) connections are becoming more and more widely available and can provide an excellent Internet connection. The connections work by splitting your phone line into two separate channels, one for data (Internet) and other voice (phone calls), which means you can talk on the phone

and be connected to the Internet at the same time.

You will often see ADSL connection services advertised as having different speed specifications. Given below are some common configurations:

256Kbps/128Kbps or 512Kbps/128Kbps

1Mbps/256Kbps or 2Mbps/512Kbps

8Mbps/1024Kbps

Notice that there are two values to each configuration. The first figure states the download speed and the second figure is the maximum upload speed. As an example let us take the second configuration 512Kbps/128Kbps. This means that you can potentially download data at a speed of 512Kbps and upload data at 128Kbps.

Advantage of ADSL connection : ADSL technology eliminates the need for a second phone line by allowing voice and data transfer at the same time (you can use the phone as normal while connected to the Internet). Because ADSL transfers data digitally, it eliminates the usual performance overhead associated with standard dial-up connections. In other words, ADSL doesn't need to convert the data from digital to analogue and back again.

Disadvantage of ADSL connections : ADSL connections are not available to everyone. You should always ensure that you have ADSL coverage in your area. The hardware costs can be quite significant as you will need a special ADSL modem and ADSL filters to use the service. Most ISPs allow you to hire these items which can reduce the initial cost. Because ADSL connections are always on you will need a firewall to protect your PC.

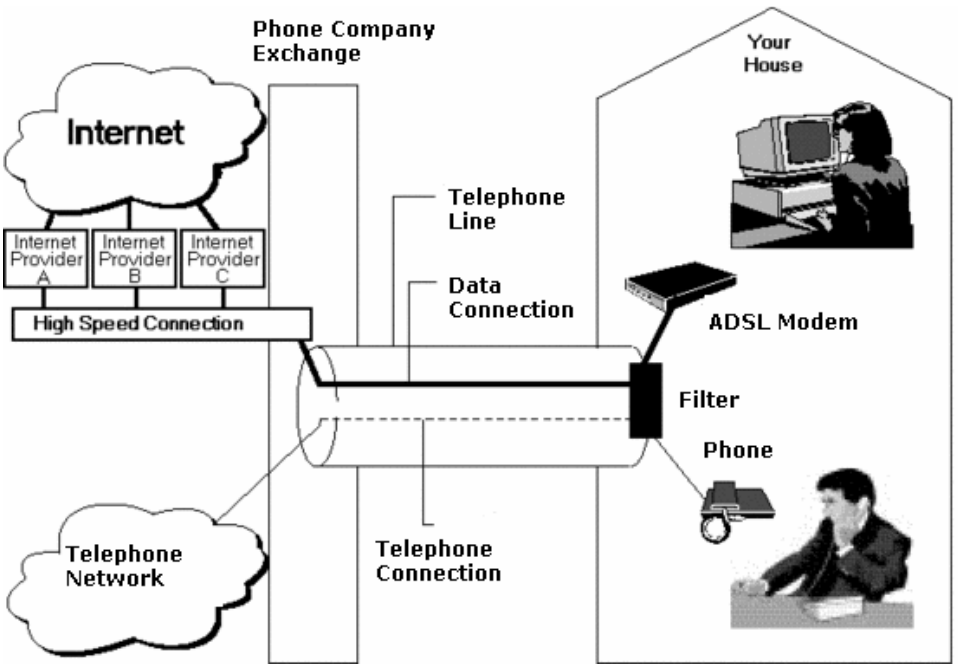


Fig. 8.3 : ADSL Connection

8.5.3 Cable Connections

Cable connections are considered one of the best types of Internet connection available to the home user because they offer very fast and reliable connections with a fixed monthly fee. Cable companies usually offer different packages to suit different Internet subscribers.

The different packages will offer different speed specifications and bandwidth limits. Because a cable connection uses a totally separate medium to transfer data it does not affect your ability to make/receive phone calls.

Advantage of Cable connection : Speed is a major reason for having a cable connection. Like ADSL connections, cable connections transfer data digitally, eliminating any digital/analogue conversion overhead. Cable connections are always on, eliminating long waits to make a connection.

Disadvantage of Cable connection : Cable connections are not available in every area. You will need to contact the cable company of your choice to ensure that you have coverage.

8.6 INTERNET SERVICE PROVIDER (ISP)

Internet service providers (ISPs) are companies that help the users to connect to the Internet for a monthly fee. In return, they provide a username, a password, and telephone number. The username and the password are used to authenticate the user on the Internet. The telephone number is used to establish connection with the dial-up server of the ISP. However, ISPs also support other forms of Internet access like DSL and cable connections and other dedicated technologies. In addition to providing Internet services to individual users, ISPs also provide Internet services to large companies in return for an annual fee. When such a company subscribes to an ISP, the company provides Internet access to all the computers on the corporate network. Examples of ISP in india are Bharti Airtel Ltd, Reliance Communication, BSNL etc.



CHECK YOUR PROGRESS

Q.1. Choose the correct answer

a) In a peer-to-peer communication model a group of computers use

i) different network program

ii) same network program

iii) none of the above

b) Internet is

i) complex system ii) decentralized system

iii) dynamic system iv) all of the above

c) Dial-up connections are

i) very fast ii) fast

iii) very slow iv) none of these

d) Dial-up connections transfer data over

i) analogue line ii) digital line

iii) both analoogue and digital

iv) none of these

e) In ADSL connection data and voice

i) can be sent at the same time

ii) can the send one at a time

iii) cannot be sent iv) all the above

8.7 ORGANIZATION OF INTERNET

The Internet system consists of a number of interconnected packet networks supporting communication among host computers using the ***Internet protocols (IP)***. A computer linked on the Internet is known as the host computer. The data to be moved from one host to another host is broken into small pieces called ***packets***. Each packet has a header with the address of destination host. The packets of different sizes move on in various networks before reaching the destination. Various packets of one file may take different routes to reach a destination. The different networks on the Internet are connected with special purpose devices called **routers,** a term which is already familiar to us. These routers look for destination address given on each packet and direct the packet to take the best route to the destination. Routers take their decisions based on information that is constantly reaching them from all over the Internet. They also hear from other routers about the links that are down or congested/slow, or about routers that are no longer accepting packets for certain destinations. Each packet's destination and proposed route are evaluated individually, in the blink of an eye, and sent off along the best route for that particular packet at that particular moment. The following diagram(8.4) shows the function of routers.

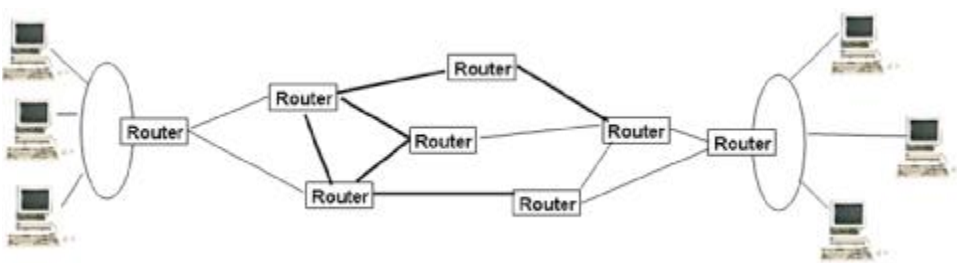


Fig. 8.4 : Routers connected on the Internet

8.7.1 Internet Protocol Suite

The Internet protocol suite is the set of communications protocols that implement the *protocol stack* on which the Internet runs. A protocol stack is a complete set of protocol layers that work together to provide networking capabilities.

A protocol is a mutually agreed-upon format for doing something. With regard to computers, it most commonly refers to a set of rules (i.e., a standard) that enables computers to connect and transmit data to one another; this is also called a ***communications protocol***. The Internet protocol suite is sometimes called the ***TCP/IP protocol suite***, which refers to the two most important protocols in it : the *Transmission Control Protocol (TCP)* and the *Internet Protocol (IP)*. A protocol can be implemented by hardware, software, or a combination of the two. It is called a protocol stack because it is typically designed as a hierarchy of layers as shown in the figure below, each supporting the one above it and using those below it. Each layer solves a specific set of problems involving the transmission of data and provides well-defined services to the layers above it. The application layer is logically closer to the the application programs and deal with more abstract data, and it relies on lower

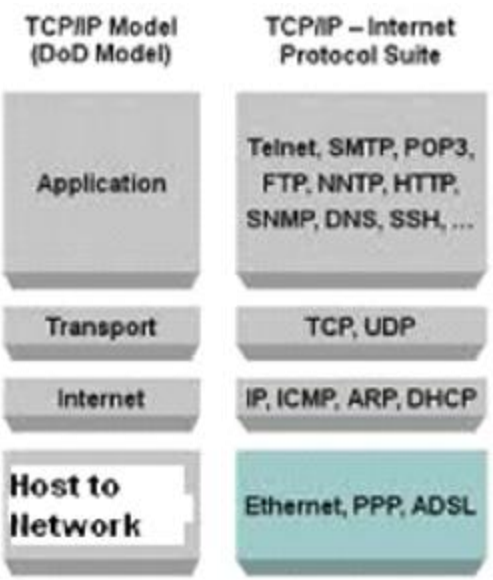


Fig. 8.5 : TCP/IP protocol suite

layers to convert their data into forms that can be physically manipulated for transmission.

Let us see the functions of a few protocols of application layer :

Telnet : Telnet is a popular client-server application program used for terminal services. Telnet is an acronym for Terminal Network. Telnet protocol is used to establish an on-line connection to a remote machine, so that the client can access the utility or application program from the remote machine.

FTP : The FTP (File Transfer Protocol) is a standard network protocol used to copy a file from one host to another over a TCP/IP-based network, such as the Internet. Using FTP, you can also update (delete, rename, move, and copy) files at a server. You need to **log on** to an FTP server.

SMTP : The SMTP (Simple Mail Transfer Protocol) protocol is used by the Mail Transfer Agent (MTA) to deliver your eMail to the recipient's mail server in the Internet. The SMTP protocol can only be used to send emails, not to receive them.

POP 3 : The POP (Post Office Protocol 3) protocol provides a simple, standardized way for users to access mailboxes and download messages to their computers. When using the POP protocol all your eMail messages will be downloaded from the mail server to your local computer.

IMAP : The IMAP (Internet Message Access Protocol) is one of the most prevalent Internet standard protocols for e-mail retrieval.

HTTP : The HTTP (Hypertext Transfer Protocol) is the set of rules for exchanging files (text, graphic images, sound, video, and other multimedia files) on the **World Wide Web.** Relative to the **TCP/IP** suite of protocols (which are the basis for information exchange on the Internet), HTTP is an application **protocol.**

DNS : The DNS (Domain Name System or Service or Server), is an Internet service that translates domain names like www.google.com or www.kkhsou.org etc. into IP addresses.

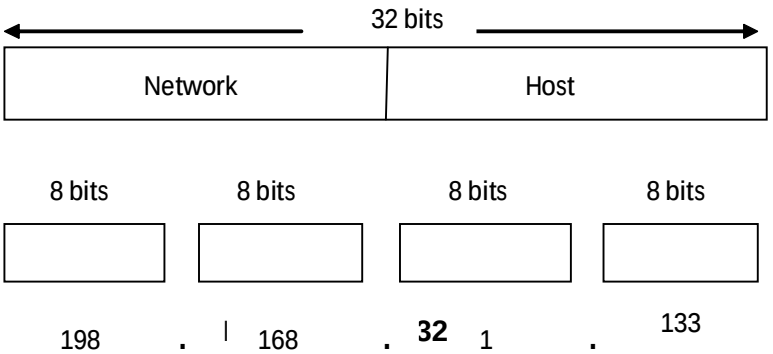
8.7.2 IP Address

So far we have discussed the fundamental concepts of Internet. Our discussion will be incomplete if we donot mention about the Internet address system. Let us see what is Internet address briefly.

The number of nodes on the Internet are increasing day-by-day. Always remember that each host computer or node on the Internet has its own unique address through which communication is possible with every one. To identify a host on the Internet, three addressing systems have been evolved : (a) A numerical system called I*P addressing*, (b) a hierarchical naming system called the *Domain Name System*, and (c) an addressing system called URLs, which are used for identifying sites on the web.

An ***Internet Protocol address (IP address)*** is a numerical label that is assigned to any device connecting to the Internet and that uses the Internet Protocol for communication between its nodes. Each IP address has specific components and follows a basic format. Each IP address is 32-bit logical address that is divided into two main parts: the *network number* and the *host number*. The network number identifies a network and must be assigned by the Internet Network Information Center (InterNIC) if the network is to be part of the Internet. The host number identifies a host on a network and is assigned by the local network administrator.

The 32-bit IP address is grouped eight bits at a time, separated by dots, and represented in decimal format (known as *dotted decimal notation*). Each bit in the octet has a binary weight (128, 64, 32, 16, 8, 4, 2, 1). The minimum value for an octet is 0, and the maximum value for an octet is 255. The following figure illustrates the basic format of an IP address 198.168.1.133 (written in decimal).



We will not discuss IP addressing in more details now but you should know that IP addressing is divided into five different classes which are A, B,C, D, and E. The classes A, B, and C are available for commercial use only. In the following classes A, B and C are shown :

	1st octet	2nd octet	3rd octet	4th octet
Class A	**Network**	**Host**	**Host**	**Host**
Class B	**Network**	**Network**	**Host**	**Host**
Class C	**Network**	**Network**	**Network**	**Host**

Fig. 8.7 : IP Address Class

You have seen that IP addresses are numeric and it may be quite difficult to remember such long numeric data. So there is a system to translate it into simple English names. The Domain Name System (DNS) serves as a directory programme for IP addresses and it takes care of translating IP addresses to simple English names.

8.7.3 Domain Name System

DNS refers to Domain Name System and represents a powerful Internet technology for converting domain names to IP addresses. Behind every site, there is an IP address. But, while it is easy to remember the name of a website, it is quite hard to remember the exact IP address. For example, everybody knows about Google.com, but if you had to remember "74.125.45.100", things would have been much harder.

How does a DNS program works? - Every time a domain name is typed in a browser it is automatically passed on to a DNS server, which translates the name into its corresponding IP address.

This system offers :

- an **hierarchical namespace** where each name is unique
- a system of **distribution servers** enabling namespace to be made available.
- a **client system** which helps to find out the IP address corresponding to a name.

DNS Hierarchy : The structure of the DNS system relies on a tree structure. The top-most node of the tree is the DNS root domain (**.**), under which there are subdomains, such as .com, .edu, .gov, and .mil, which are called ***top level domains (TLDs)***. Each node of the tree is called a **domain name**. The end of a branch is called the **host**, and corresponds to a machine or entity on the network. The

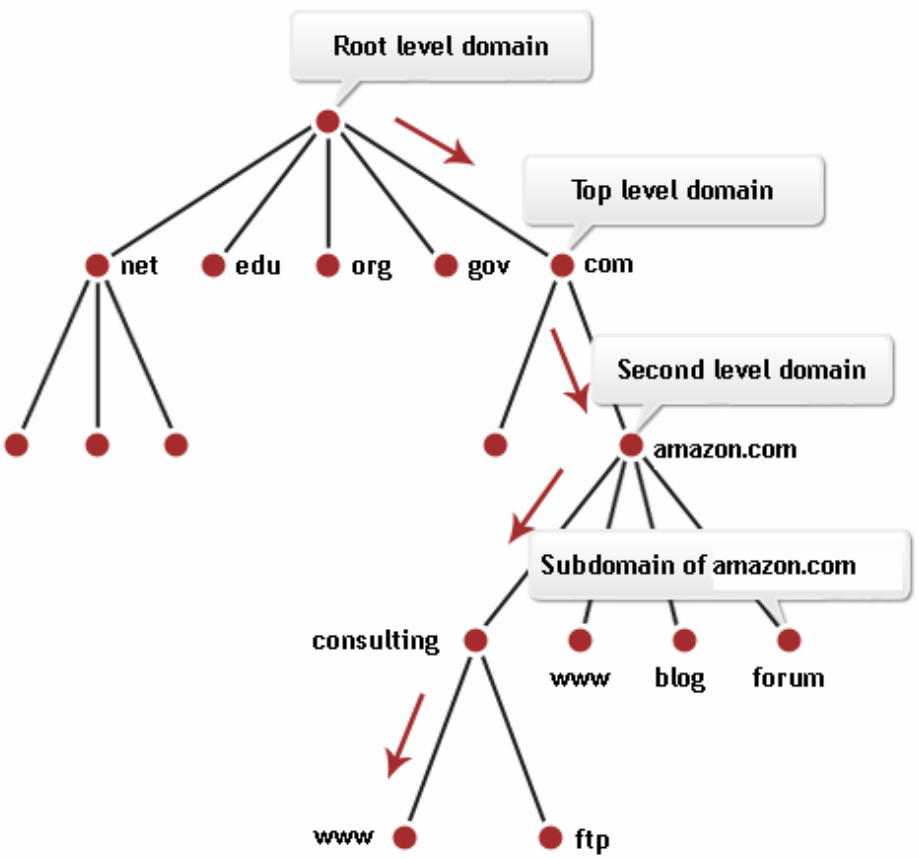


Fig. 8.8 : DNS Hierarchy

host name given to it must be unique in the respective domain. In the following figure DNS hierarchy is shown.

The root DNS servers (root name servers) keep track of all the Top Level Domain (TLD) name servers. The client queries the root name servers to resolve a request for the given domain name. In response the root name server provides the address of the TLD name server for the given query in which the domain name ends with. For example, if a client requests for google.com, the root name server will address the client to the com DNS server so as to solve his query.

The top level name server holds the list of authoritative name server in their respective domain. For example, the ***com*** domain holds the address of yahoo.com, google.com etc. Every TLD includes many second-level domains; for example, amazon.com as shown. Every second level TLD may include number of third level domains. In the figure you have noticed that TLD divides the Internet domain name space into several domains. Most commonly used domains are :

com – Usually used by commercial organization. Eg. Yahoo (yahoo.com)

edu – Usually used by educational institutes. Eg. San Jose State University (sjsu.edu)

org – Used by non profit organizations. Eg. IEEE (ieee.org)

mil – used by military organizations. Eg. US army (army.mil)

net – In earlier days it was used to represent the network infrastructure. Now a days it is open public for any commercial organization.

gov – used to represents government organization. Eg. assam. gov.in

The top level domain names also use two-letter country designations which is known as *geographic domain*. For example,

.us is used for the United States,

.ca for Canada (not California),

.uk or .gb for the United Kingdom or Great Britain,

.il for Israel and

.in for India etc.

8.7.4 Uniform Resource Locators (URL)

A URL (Uniform Resource Locator) is the the global address of documents and other resources on the World Wide Web. The type of resource depends on the Internet application protocol. Using the World Wide Web's protocol (HTTP), the resource can be an HTML page, an image file, a Java applet or any other file supported by HTTP. The URL contains the the following parts

- name of the protocol required to access the resource
- a domain name that identifies a specific computer on the Internet where the resource(file that we want to access) is located,
- path name of the file on the host computer.

The protocol identifier and the resource name are separated by a colon and two forward slashes. In the following an example is shown :

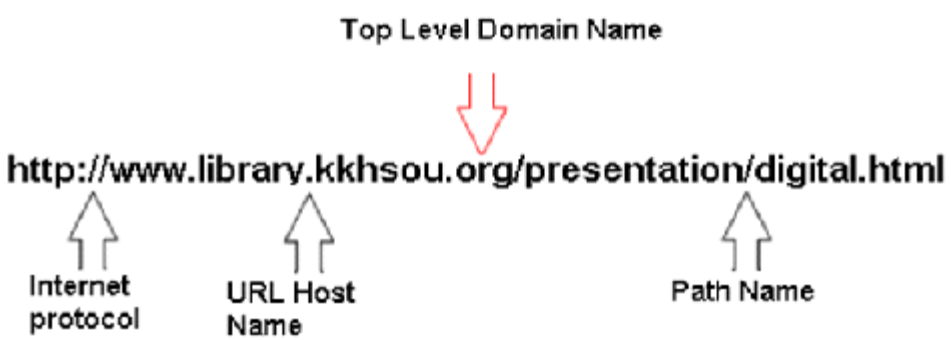


Fig. 8.9 : URL example



CHECK YOUR PROGRESS

Q.2. Choose the correct answer

a) For connection of the Internet, you will need

i.an IP address ii. a TCP/IP protocol

iii.an ISP iv. All the above

b) A host on the Internet finds another host by its

i. postal address ii. IP address

iii. electronic address iv. none of above

c) User can get files from another computer on the Internet by using

i. FTP ii. HTTP iii. UTP iv. i & ii both

d) IP addresses are translated into simple English names by the following system -

i. HTTP ii. SMTP iii. DNS iv. FTP

e) An IP address contains

i. 64 bits ii.16 bits iii. 8 bits iv. 32 bits

Q.3. Define IP address, Domain name and Host.

.....

.....

.....

.....

.....

.....

8.8 APPLICATION OF INTERNET

The Internet is a global system of interconnected computer networks that use the standard Internet Protocol Suite (TCP/IP) to serve billions of users worldwide. It is a network of networks that consists of millions of private, public, academic, business, and government networks, of local to

global scope, that are linked by a broad array of electronic and optical networking technologies. Now-a-days the application of Internet touches the every field of modern society. Let us briefly focus some of them.

Email : Email is now an essential communication tool in business. It is also excellent for keeping in touch with family and friends. The advantages to email is that it is free (no charge per use) when compared to telephone, fax and postal services.

Information : There is a huge amount of information available on the Internet for just about every subject known to man, ranging from government law and services, trade fairs and conferences, market information, new ideas and technical support.

Services : Many services are now provided on the Internet such as online banking, job seeking and applications, and hotel reservations. Often these services are not available off-line or cost more.

E-Commerce : The Internet is a very effective way to buy and sell products all over the world. Business houses use the Internet to provide product information, online support service, etc. Companies carry out online trading including advertising, selling, buying, distributing products, and providing after-sales services through the Internet.

Communities : Communities of all types have sprung up on the Internet. Institutions use the Internet for voice and video conferencing and other forms of communication that allow people to telecommute, or work from a distance. It is a great way to meet up with people of similar interest and discuss common issues. Scientists and scholars use the Internet to communicate with colleagues, to perform research, to distribute lecture notes and course materials to students, and to publish papers and articles.

Entertainment : Entertainment is another popular reason why many people prefer to surf the Internet. In fact, media of Internet has become quite successful in trapping multifaceted entertainment factor. Downloading games, visiting chat rooms, online gaming or just surfing the Web are some of the uses people have discovered. There are numerous games that may be downloaded from the Internet for free.

a very large amount of space if they all had to be stored back on a web site's server.

Security : If a cookie was stored on a web site, then it can be accessed by anyone with access to that web site. However, when the cookie is stored on your computer, then it can't be accessed by hackers that break into the web site.



CHECK YOUR PROGRESS

Q.4. a) Which of the following protocols is used by the Internet mail?

i. HTTP ii. TCP/IP iii. SMTP iv. none of these

b) Find the odd item from the following -

i. Yahoo! ii. HOtBot iii. LycosS iv. Win2000

c) Computers on the Internet owned and operated by education institution form part of the

i. com domain ii. edu domain

iii. mil domaiin iv. gov domain

d) For a small website, one needs to buy space from the

i. Network administrator ii. Telephone exchange

iii. ISP iv. None of these

e) Find the odd item from the following :

i. FTP ii. SMTP iii. JAVA iv. HTTP

Q.5. Fill in the blanks :

i. A _____ is a program that is executed by the web server in response to a request.

ii. The client-side scripts are executed by the _____.

iii. Web _____ allow a Web site to store information about you as well as to track your visits to it over a prolonged period of time.

Q.6. Choose the appropriate option:

i) Which of the following protocols is responsible for Internet mail?

a) TCP/IP b) HTTP c) FTP

d) None of these

ii) A user can get files from another computer on the Internet by using

a) FTP b) HTTP c) UTP

d) None of these

iii) Take the odd item out

a) Internet Explorer b) Mozilla Firefox

c) Google d) Netscape navigator

iv) Take the odd item out

a) PHP b) JSP c) HTML d) XML

v) FTP is used for

a) Uploading files only b) Downloading files only

c) Both (a) and (b) d) None of these



LET US KNOW

An ***Intranet*** is a Web-based architecture used for managing internal information of an office, company, university or college or even a large library that runs on IP protocols. It means that an intranet is a private network that is contained within an enterprise.. It uses Internet protocols such as **TCP/IP** (Transmission Control Protocol/Internet Protocol), **HTTP** (Hypertext Transfer Protocol), **HTML** (Hyper Text Markup Language) **SMTP** (Small Message Transfer Protocol), **FTP** (File Transfer Protocol), and **AAA** (Access, Authorization, and Authentication). It may consist of many interlinked Local Area Networks and also use leased lines in the Wide Area Network. The main purpose of an intranet is to share information within the organization and computing

resources among employees. An intranet can also be used to facilitate working in groups and for teleconferences.

An ***Extranet***, or extended Intranet, can be defined as a private network of linking branch offices or several cooperating organizations located outside the walls of any organisation. An Extranet service uses existing Intranet interactive infrastructure, including standard servers, e-mail clients and Web browsers. This makes Extranet far more economical than the creation and maintenance of a proprietary network. It enables trading partners, suppliers and customers with common interests to form a tight business relationship and a strong communication bond.



8.9 LET US SUM UP

The Internet is a network of networks. It connects millions of computers and thousands of computer networks throughout the world. The Internet has revolutionised our society, our economy and our technological systems. This unit has discussed specifically the technology associated with the Internet such as Internet architecture, organization of Internet (Internet protocol and Internet addressing). It is important to know the various information resources and services available in Internet or World Wide Web (www). It is equally important to know about search engines, selecting and searching from a particular search engine. Client side technologies includes HTML, DHTML, VB Script, Java Script,ECMAScript, JScript, Java applets and Flash Animations etc. Server side technologies includes PHP, JSP, CGI/ Perl, XML/XSL, MYSQL, PostgreSQL etc.

8.10 ANSWERS TO CHECK YOUR PROGRESS

Ans. to Q. No. 1 : a) ii, b) iv, c) iii, d) i, e) i.

Ans. to Q. No. 2 : a) iv, b) ii, c) i, d) iii, e) iv.

Ans. to Q. No. 3 : IP address is the Internet Protocol (IP) address given to every computer connected to the Internet. An IP address is needed to route information much like a street address or PO box is needed to receive regular mail. Example: 177.198.48.157

Domain name is a text name which a computer network registers. The domain name is used to give computers text names rather than using the numeric IP addresses. This is like getting a vanity phone number that spells out a word to make it easy to remember.

Computer (host) names are names given to individual computers. Each host name corresponds to an IP address. Host names and domain names are optional and everything will work fine using just IP addresses.

Ans. to Q. No. 4 : a) iii, b) iv, c) ii, d) iii, e) iii

Ans. to Q. No. 5 : i) script, ii) browser, iii) cookies

Ans. to Q. No. 6 : i) a, ii) a, iii) c, iv) c, v) c

8.11 FURTHER READING

- Data and Computer Communications by William Stallings, Pearson Education.
- Data Communications and networking by Behrouz A. Forouzan, Tata McGrawHill.
- *"*Web technologies: a computer science perspective*", Jeffrey C. Jack son*, Pearson Prentice Hall.

- *"*Web Technology : A Developer S Perspective*", Gopalan, Gopalan/ akilandeswari*, Pearson Prentice Hall.



8.12 MODEL QUESTIONS

Q.1. What is Internet? Explain why Internet is called a 'network of networks'?

Q.2. What do you mean by the Internet architecture? Describe the most commonly used Internet architecture.

Q.3. Differentiate between the dia-up and ADSL Internet connection.

Q.4. Differentiate between the ADSL Internet connection and cablle Internet connection.

Q.5. What is Internet protocol suite? List out the functions of five Internet protocols.

Q.6. What is an IP address? Discuss the function of domain name system.

Q.7. What is an URL? Using an example identify the components of an URL.

Q.8. Write short notes on the following :

a) Dial-up connections b) ADSL connections

c) Internet Service Providers d) IP Address

e) Domain name system f) Uniform resource locator

****

UNIT 9 : WORLD WIDE WEB

UNIT STRUCTURE

9.1 Learning Objectives

9.2 Introduction

9.3 Web Pages

9.4 Web Browsers

9.5 Web Server

9.6 Web Search Engines

9.7 Web Cookies

9.8 Technologies in WWW

9.9 Let Us Sum Up

9.10 Answers to Check Your Progress

9.11 Further Reading

9.12 Model Questions

9.1 LEARNING OBJECTIVES

After going through this unit, you will be able to :

- define Internet and elaborate on its growth
- describe the organization of Internet
- illustrate the world wide web
- describe web servers, web browsers and search engines
- know the server and client side technology

9.2 INTRODUCTION

The World Wide Web- known as WWW or W3 or simply, the Web - is one of several Internet resource discovery tools developed to help people publish, organize and provide access to information on the Internet. The Web was first developed by Berners Lec in 1989 while working at CERN, European Particle Physics Laboratory in Switzerland, and has now become the most powerful, and popular, resource discovery toll on the Internet. The

WWW can be defined as a hypertext, multimedia, distributed information system that provides links to hypertext documents, as well as to many other Internet tools and databases.

The most important things required to understand the underlying mechanism of the Web are the client-server architecture, the Hypertext transfer protocol (HTTP), Universal Resource Locators (URLs) which we have already seen very briefly and the another important resource is Hypertext Markup Language (HTML). The most common method of creating a web document is through the use of Markup Languages. Most of these are created by adding a set of formatting code to ASCII text to show fonts, justifi-cations, links, etc. on the web. A markup language is a language that uses tags to indicate a change in presentation style or a change in content type.

By now, you can imagine that the web ***documents*** are referring to each other by ***links***. This simple view is known as the ***hypertext*** paradigm. The reader sees on the screen a document with sensitive parts of text representing the links. A link is followed by mere pointing and clicking.

9.3 WEB PAGES

A web page is nothing but a hypertext document which is suitable for the World Wide Web and can be accessed through a web browser and displayed on a monitor or mobile device. Web pages are usually written in HTML or XHTML and may contain static contents or dynamic contents. Web pages provide navigation to other web pages via ***hypertext links***. Hypertext is a system of organizing, navigating, distributing and publishing information electronically.

Contents of a Web Page :

A web page may contain various types of information, which can be divided into two main groups - *perceived information* (visible to the website visitor) and *hidden information* (hidden from the visitor's eye).

Depending on the purpose and target audience of a website, its perceived information could be textual, non-textual and interactive.

The non-textual information includes static images (e.g. GIF, JPEG, PNG or TIFF), animated images (e.g. animated GIF, Flash, Shockwave, Java Applet), vector formats (e.g. Flash, SVG), audio file formats (MIDI, WAV, MP3, Java Applets), video files (WMV, RM , FLV, MPG, MOV) etc. Interactive content on web pages could be displayed via DHTML, interactive illustrations, DHTML based buttons. *Hyperlink* or *link* is a navigational element in a web page through which navigation between the content on separate pages is possible.

The hidden information on web pages includes comments, metadata, charset details, CSS visual specifications, scripts (e.g. the interactivity focused JavaScript) etc.

Depending on the type of information contained in a web page, web pages can be divided into three categories :

- ***dynamic web page***
- ***static web page***
- ***active web page***

A ***dynamic web page*** is a kind of web page that has been prepared with fresh information (content and/or layout), for each individual viewing. It is not static because it changes with the time (ex. a news content), the user (ex. preferences in a login session), the user interaction (ex. web page game), the context (parametric customization), or any combination of the foregoing.

A web page is **static**, if it does not change its behaviour in response to external actions. A static web page remains the same, i.e. static, for all its life, unless and until someone manually changes its contents. Examples of some static web pages are some home pages, page specifying the contact details, etc., which do not change that often.

An **active web page** is a web page that executes a program on the client machine, i.e. web browser. It means the web page shown by a web browser is the program currently executed by the client and hence that page is called an active web page.

Creating a Web Page :

A web page can be created in many different ways. There are a lot of web pages created by simply using HTML code which are simple and not very interactive. Advanced web pages can be created by using a programming language which is also known as scripting languages, such as ***JSP, PHP, Python*** or ***Perl***.

Working of Web Pages:

The information on a web page is displayed online with the help of a web browser, which connects with the server where the website's contents are hosted through the Hypertext Transfer Protocol (HTTP). For instance, if you look at the URL of the web page you are on at the moment, you could notice the prefix 'http://', which tells the browser what protocol to use to execute the particular URL request.

Each web page's contents are usually presented in HTML or XHTML format, which allows for the information to be easily structured and then quickly read by the client's web browser. With the help of CSS (Cascading Style Sheets), designers can precisely control the web page's look and feel as far as the layout typographic elements, color scheme and navigation are concerned. CSS instructions can be either embedded within the HTML web page (valid for that particular page) or can be included in a separate external file (valid for the whole site).

9.4 WEB BROWSERS

In a network such as the World Wide Web, all the computers are divided into two categories : *client* and *server*. A ***web client*** is a software that accesses a web server by sending requests and processing the resulting response.

The Figure 9.1 shows how a web server and a web client is connected to a network and what components are actually held by them for communicating each other.

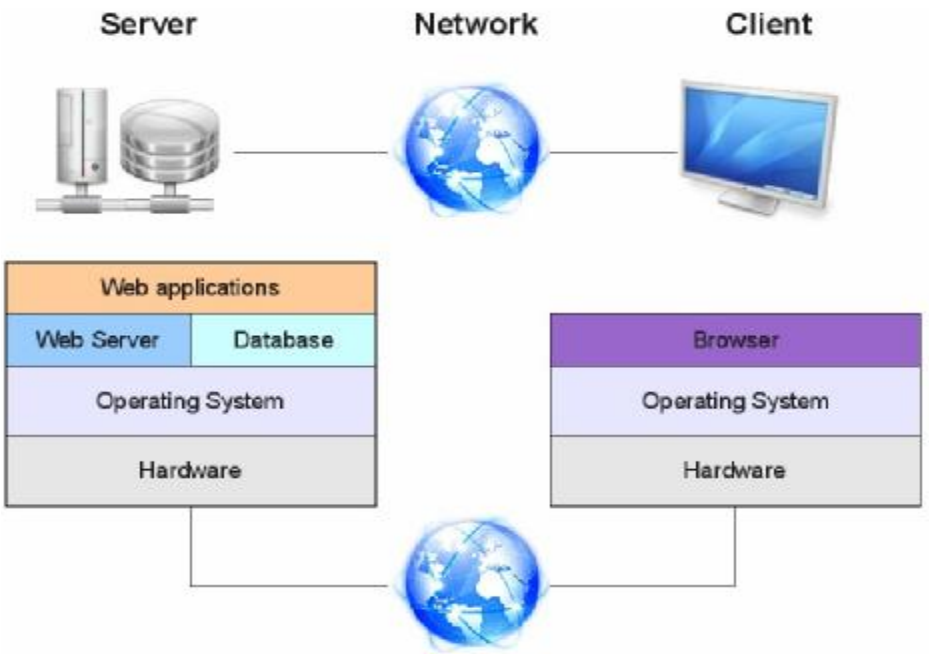


Figure 9.1 : A Web Server and a Client

The web server is the computer on the Internet that stores the web pages, images and other web resources. When we click a hyperlink or type a URL into the web browser, a request for the web page is made by the web client to the web server. The server then receives the request and responds by sending the HTML page to the browser. The browser then interprets and displays the page.

The client-side processing is carried out by the **Web browser**. The browser consists of an interface and processing software. ***Web browsers*** are programs running on the client. They are called browsers because they allow the users to browse the resources available on the servers. We can search for and view different kinds of information on the Web with browsers. The main purpose of a web browser is to make the information resources in the Web available to the users. A browser allows you to view the documents written in HTML (Hyper Text Markup Language). These documents are called *web pages* which we have alrady discussed.

9.5 WEB SERVERS

Server software runs exclusively on server machines, handling the storage and transmission of documents. In contrast, client software such as Netscape, Internet Explorer, etc. runs on client computer that access, translates and displays documents.

A web server is a software package that processes HTML documents for viewing by Web browsers that runs on client computers. Web servers can be run from any hardware platform. There are servers that are specifically designed for Macintosh computers, PCs, Silicon Graphics, and various other platforms, including the Amiga. It can also run under several operating systems, including MS Windows, Windows NT, Unix, Linux etc.

The web server is responsible for document storage and retrieval. It sends the document requested (or an error message) back to the requesting client. The client interprets and presents the document. The client is responsible for document presentation. The language that web clients and servers use to communicate with each other is called the Hypertext Transfer Protocol (HTTP). All web clients and servers must be able to speak HTTP in order to send and receive hypermedia documents. For this reason, web servers are often called HTTP servers, or HTTP Daemons (HTTPD).

9.6 SEARCH ENGINES

The Web is quickly becoming a vast information space that needs search tools to find information efficiently. The search engines can be defined as the tool created for finding, classifying and storing information about various Websites on the Internet. These can help in locating information of relevance on a particular subject by using various search methods. A search engine is a service that indexes, organizes, and often rates and reviews Web sites. Search engines

are online utilities that quickly search thousands of Web documents for an entered word or phrase. Search engines are usually accessed through Web browser software. Each search engine provides different searching options and has its own look.



Figure 9.2 : A list of Search Engines

You may wonder as to how the search engines find the answers to a query so quickly. Actually it is a four-step process and the steps are :

- ***Crawling*** the Web, following links to find pages.
- ***Indexing*** the pages to create an index from every word to every place it occurs.
- ***Ranking*** the pages so that the best ones show up first.
- ***Displaying*** the results in a way that is easy for the user to understand.

Given below is a list of important search engines which you can use for searching the web.

Name	**URL :**
Altavista	http://www.altavista.com
Excite	http://www.excite.com
Hotbot	http://www.hotbot.com
Infoseek	http://www.infoseek.com

Lycos	http://www.lycos.com
Northern Light	http://www.northernlight.com
Webcrawler	http://www.webcrawler.com
Yahoo	http://www.yahoo.com
Google	http://www.google.com

9.7 WEB COOKIES

Basically, the cookies stores tracking data about your browsing. Cookies is a message given to a Web browser by a Web server. The browser stores the message in a text file in the client machine and uses each time the browser requests a page from the server. Web cookies were also called "magic cookies" when they were first introduced. When you enter a Web site using cookies, you may be asked to fill out a form providing such information as your name and interest.

If you use Microsoft Internet Explorer, your cookies are kept as individual files in a folder named "Cookies", often found in the "Documents and Settings" folder. If you use Mozilla Firefox, your cookies are stored in a text file named "cookies", often found in the "Firefox/Profiles" folder.

Some of the reasons why web sites use cookies are described below:

Customization : Some sites use cookies to record your surfing patterns, and then optimize the information the site subsequently presents. For example, a search site may present you with advertising that reflects your interests based on the keywords you search for.

Distribution : One of the key reasons for which web sites use cookies is to distribute the information storage. A cookie takes up a small amount of space on an individual computer, but would take up a very large amount of space if they all had to be stored back on a web site's server.

Security : If a cookie was stored on a web site, then it can be accessed by anyone with access to that web site. However, when the cookie is stored on your computer, then it can't be accessed by hackers that break into the web site.



CHECK YOUR PROGRESS

Q.1. Distinguish between web server and web browser.

.....

.....

.....

.....

Q.2. Distinguish between Internet and WWW.

.....

.....

9.8 TECHNOLOGIES IN WWW

The Web uses the HTTP protocol, which governs the dialog between the client and the server. Now, if we consider the technologies used in WWW then we can divide it into two categories like : *technologies in client side* and *technologies in the server side*. Client-side technologies are the programs that operate in the browser. There is no need to interact with the server. These programs are created by using the languages known as ***client side scripting language*** which are generally very easy to use, and you can try them on your own computer. **HTML** is a basic mark-up language that you will use to create the structure of your web page so HTML is one of the client side language. Remember that HTML is not a scripting language which is a basic building block of static web page.

A ***script*** is a program that is executed by the web server in response to a request. The SCRIPT element includes a script in the HTML document. Scripts allow greater interactivity in a document by responding to the user events. A scripting language is a lightweight programming language. *VBScript, Java Script* are examples of scripting languages.

The client-side scripts are executed by the browser or the client. Their work is confined only to the browser. They donot have to wait for the server to process and return data. This makes the web pages more responsive. The browser should also support the scripts that the web page wants to display.

The following applications are used to form server-side technology.

PHP : PHP is a fast, server-side scripting language that is used to create interactive, dynamic web sites.

JSP : Java Server Pages or JSP for short is Sun's solution for developing dynamic web sites. JSP provides excellent server side scripting support for creating database driven web applications.

CGI/Perl : Perl is a programming language that can handle input and output from a Web Server, usually through the Common Gateway Interface.

XML/XSL : XML is a software and hardware independent markup language designed for describing and transmitting information. XSL is a language for defining, transforming and formatting XML documents.

MySQL : MySQL is a fast, open-source Relational Database Management System that uses the popular Structured Query Language (SQL).

PostgreSQL : PostgreSQL is a more sophisticated open-source Relational Database Management System. It is well-suited to sites that require robust database functionality, such as e-commerce sites.

Linux/Apache : Linux is a popular open-source operating system, and Apache is the most widely-used web server on the Internet. Together they provide a fast, extremely reliable, secure platform for web sites.



CHECK YOUR PROGRESS

Q.3. Fill in the blanks :

i.A _____ is a program that is executed by the web server in response to a request.

ii. The client-side scripts are executed by the _____.

iii. Web _____ allow a Web site to store information about you as well as to track your visits to it over a prolonged period of time.

Q.4. Choose the appropriate option:

i) Which of the following protocols is responsible for Internet mail?

a) TCP/IP b) HTTP c) FTP

d) None of these

ii) A user can get files from another computer on the Internet by using

a) FTP b) HTTP c) UTP

d) None of these

iii) Take the odd item out

a) Internet Explorer b) Mozilla Firefox

c) Google d) Netscape navigator

iv) Take the odd item out

a) PHP b) JSP c) HTML d) XML

v) FTP is used for

a) Uploading files only b) Downloading files only

c) Both (a) and (b) d) None of these



9.9 LET US SUM UP

It is important to know the various information resources and services available in Internet or World Wide Web (www). It is equally important to know about search engines, selecting and searching from a particular search engine. Client side technologies includes HTML, DHTML, VB Script, Java Script,ECMAScript, JScript, Java applets and Flash Animations etc. Server side technologies includes PHP, JSP, CGI/Perl, XML/XSL, MYSQL, PostgreSQL etc.



9.10ANSWERS TO CHECK YOUR PROGRESS

Ans. to Q. No. 1 : A *Web server is a computer* that delivers (serves up) *Web pages.* Every Web server has an *IP address* and possibly a *domain name.* For example, if you enter the *URL http:/www.pcwebopedia.com/index.html* in your browser, this sends a request to the server whose domain name is *pcwebopedia.com.* The server then fetches the page named index.html and sends it to your browser. *Web browser,* a *software application* used to locate and display Web pages which is found on a client computer. The main function of the Web browser is to make the information resource available to the users. The two most popular browsers are *Netscape Navigator* and Microsoft *Internet Explorer.* Both of these are graphical browsers, which means that they can display graphics as well as text.

Ans. to Q. No. 2 : The terms Internet and World Wide Web are often used in everyday speech without much distinction. However, the Internet and the World Wide Web are not one and the same. The Internet is a global data communications system. It is a hardware and software infrastructure that provides connectivity between computers. In contrast, the Web is one of the services communicated

via the Internet. It is a collection of interconnected documents and other resources, linked by hyperlinks and URLs.

Ans. to Q. No. 3 : i) script, ii) browser, iii) cookies

Ans. to Q. No. 4 : i) a, ii) a, iii) c, iv) c, v) c



9.11 FURTHER READING

- Data and Computer Communications by William Stallings, Pearson Education.
- Data Communications and networking by Behrouz A. Forouzan, Tata McGrawHill.
- *"*Web technologies: a computer science perspective*", Jeffrey C. Jack son*, Pearson Prentice Hall.
- *"*Web Technology : A Developer S Perspective*", Gopalan, Gopalan/ akilandeswari*, Pearson Prentice Hall.



9.12 MODEL QUESTIONS

Q.1. What is WWW? How does it function? How is it different from Internet?

Q.2. Define the term WWW ? What are the various features of WWW ?

Q.3. Define the terms Web site, Web page, Web server, URL and home age.

Q.4. Discuss the functioning of a web server and web browser. Also give two examples of a web server and web browser.

Q.5. What is Web server application ? How is it different from a Web rowser?

Q.6. Describe how the Web browser communicates with the web server.

Q.15. Write short notes on the following :

a) WWW b) Web servers

c) Web browsers d) Search Engines

****

UNIT 10 : INTRODUCTION TO HTML

UNIT STRUCTURE

10.1 Learning Objectives

10.2 Introduction

10.3 Basics of HTML

- 10.3.1 HTML Tags
- 10.3.2 HTML Editor

5.4 Document Structure Tags

5.5 Formatting Tags

5.6 List Tags

5.7 Hyperlink and Image Tags

5.8 Table Tags

5.9 Frame Tags

5.10 Form Tags

5.11 Let Us Sum Up

5.12 Answers to Check Your Progress

5.13 Further Reading

5.14 Model Questions

10.1 LEARNING OBJECTIVES

After going through this unit, you will able to :

- describe the basic concept of HTML
- explain the use of document structure tags
- identify the formatting tags
- deal with different list tags
- illustrate the use of hyperlink tags
- discuss how to add graphics using image tags
- explain the use of the table tags
- design framed output using frame tags
- describe the use the form tags.

10.2 INTRODUCTION

In the previous unit we have learnt about the introduction to Internet and the World Wide Web. This unit deals with the Static web page design. In this unit we will learn about the basic concept of Hyper Text Markup Language. We will also gain knowledge of Document structure tags and formatting tags. Besides, this unit also provides a description of list tags, hyperlink tags and image tags. In addition, the unit also describes the table tags and frame tags. At the end, the form tags are explained.

10.3 BASICS OF HTML

HTML stands for **H**yper **T**ext **M**arkup **L**anguage. It is the language on which the Internet documents are written. With HTML you can create your own Web site. When you see a web page on the Internet and if you click at 'view source' by right clicking on the page, you will visualize the HTML code behind the page. You will find that it is nothing but a collection of commands written within some angular brackets, which are known as HTML tags. The HTML language is based only on the tags. It is a very simple language. A web browser like Internet Explorer can understand a page written in HTML. Though this language provides its own syntax for giving different tags, it does not give you error if you make mistake in writing codes. The only possibility is that you will not get your desired output. It displays outputs only for the correct parts of what you have written.

10.3.1 HTML Tags

All the HTML tags are written in angular brackets. HTML tags are divided into two types:

- Container tags
- Standalone tags or empty tags

CONTAINER TAGS : Container tags are appearing in pairs i.e., they have a starting tag and a closing tag. They contain stuff between the start and the end tags. Both the starting and closing tags have

the same name but the closing tag has a slash in front of the tag name. e.g. <HTML> tag. It is the most important tag and is written as follows:

<HTML> Body of HTML tag. </HTML>

STANDALONE TAGS : It is also known as empty tag. It appears only once. It does not have any closing counterpart. So it does not contain anything. It affects at the position where it is applied.

e.g.
 tag. It breaks a line in the output at the position where it appeares.



Standard English Color: The 16 standard colors are: Black, White, Aqua, Silver, Gray, Maroon, Red, Purple, Fuschia, Green, Lime, Olive, Yellow, Navy, Blue and Teal.

RGB hexadeci-mal triplet : It is a combination of three colors: Red, Green and Blue. It has six digits, every continuous two digit represent one color. Each digit is represented by a hexadecimal number. So, it takes value between "#000000" and "#FFFFFF".

10.3.2 HTML Editor

The HTML codes can be written in any text editor such as Notepad, WordPad, Edit command of MS-DOS, MS-WORD, and ***Dreamweaver*** etc. We must use .html or .htm as the file extension of the HTML file. Once we create the HTML file in any of the text editors we can view it through any web browser.

10.4 DOCUMENT STRUCTURE TAGS

Document Structure tags are used to construct the web page as a whole. A web page created in HTML consists of three distinct sections:

- Comment section
- Header section
- Body section.

The **comment section** contains the comment or remark lines for better understanding about the document. They are non-executable and ignored by the web browser when they are executed. The comment tag is denoted as follows :

<!— This is a comment line in a web page document —>

Header section consists of tags like: <HEAD>, <TITLE>, <META> etc. Using these tags we can construct the header section of a web page.

Body section contains the major portion of the tags and they are included within the <BODY> tags. This section consists of the entire

information about the web document.

<HTML> tags : It is a container tag. The entire web page is enclosed within this tag.

Syntax :

```
<HTML>
    … Web page documents
</HTML>
```

<HEAD> tags : This tag is used to create the header section of a web page document. In this section we normally put the title of the document and other related information like information about the author and their profile etc. It is also a container tag. It may contain other tags like: <TITLE>, <LINK>, <META> etc.

Syntax :

```
<HEAD>
    Header section documents and tags only
</HEAD>
```

<TITLE> tags : This tag is used to create the title of the web page document. It is a container tag.

Syntax :

```
<TITLE> Title of the web page document. </ TITLE >
```

<BODY> tags : It is used to create the body section of the HTML document. This section contains the actual web page document, which is visible to the user through a web browser. Major portion of the tags are belonging to this section.

Syntax :

```
<BODY Bgcolor=”color” Text=”text-color” Link =”unvisited-link-color” Alink=”active-link-color” Vlink=”visited-link-color”>
    … Body section documents and tags.
</BODY>
```

Attributes : Bgcolor: Here we can set the background Color, which is either 16 **Standard English Color** or an **RGB hexadecimal triplet** e.g. Color=”#FF44AA”

Text : Here we can set the color of the entire document.



Dreamweaver: It is a software, which enables us to develop a web page with a very little effort. Here you need to specify what your requirement is. The software will design the HTML page against that page.

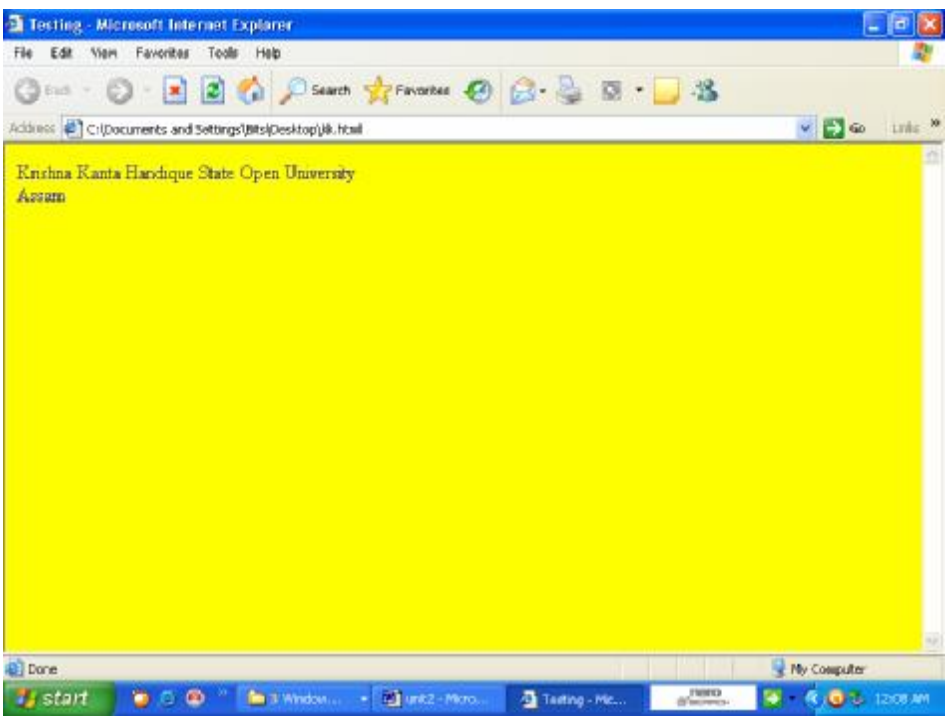
Link : It sets the color of unvisited hyperlinks.

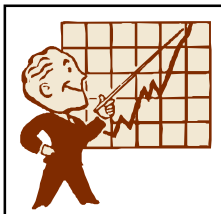
Alink : It sets the color of active hyperlinks.

Vlink : It sets the color of visited hyperlinks.

Example :

```
<HTML>
    <BODY Bgcolor="Yellow" Text="Blue">
    Krishna Kanta Handique State Open University<BR>
    Assam<BR>
</BODY>
</HTML>
```





CHECK YOUR PROGRESS

Q.1. State true or false:

a) Standalone tag contains a body.

b) Any text editor can be used to create a HTML document.

c) Web-browser gives you errors if you make mistakes in writing a HTML code.

d) Each color code in hexadecimal triplex varies from 00 to

FF and it is equivalent to the range 0 to 255 in decimal.

Q.2. Fill in the gaps:

a) Entire _____ can be included within <HTML> tag.

b) Text attribute of <BODY>tag represent the _____ of the entire HTML document.

c) Author's information can be included in the _____ section.

d) <BODY Bgcolor="#00FF00"> will set the _____ with _____ color.

10.5 FORMATING TAGS

Formatting tags can be used to change the typographic formatting of a page. For example, to change font name and size, to change font style like bold, underline, italic, strikeout, superscript, subscript etc.

<BASEFONT> tags : It is a standalone tag. Using this tag we can set the typeface, size and color of the entire body text.

Syntax :

< BASEFONT Face= "list of font names" Size="size" Color="color">

Attributes :

Face : Here we can put several font names separated by comma. Browser will try to apply the first font; if it is not available then the browser will go for the second and so on.

Size : It may vary within the range of 1 to 7.

Color : It may be any one of 16 Standard English color names or an RGB hexadecimal triplet.

<FONT> tag : It is a container tag. This tag can be used to modify font-face or font-size of a particular text portion.

Syntax :

< FONT Face= "font names" Size="size" Color="color">

Text where font properties are applied

</FONT>

Attributes :

Here, explanation of Face, Size and Color are same as BASEFONT.

<Big> tag : It is a container tag. Using it we can display the enclosed text to be bigger than the default font size.

Syntax :

<Big> … Text in bigger size …</Big>

<Small> tag : It is a container tag. Using it we can display the enclosed text to be smaller than the default font size.

Syntax :

<Small> … Text in smaller size … </ Small >

<B> tag: It is a container tag. Using this tag we can change the enclosed text to be bold.

Syntax :

<B> …Boldface text goes here... </B>

<I> tag : It is a container tag. Using this tag we can change the enclosed text to be italics.

Syntax :

<I> …Italic text goes here... </I>

<U> tag : It is a container tag. Using this tag we can change the enclosed text to be underlined.

Syntax :

<U> …Underlined text goes here... </U>

<S> or <Strike>tag : It is a container tag. Using this tag we can put a strikethrough character over the enclosed text.

Syntax :

<S> … Strikethrough text goes here... </S> (or)

< Strike > … Strikethrough text goes here... </ Strike >

<Sub> tag : It is a container tag. Using this tag we can make the enclosed text as subscript one.

Syntax :

_{… subscript text goes here...}

<Sup> tag: It is a container tag. Using this tag we can make the enclosed text as superscript one.

Syntax :

^{… superscript text goes here...}

<TT> tag : It is a container tag. Using this tag we can obtain a fixed width font, which is equivalent to typewriter font.

Syntax :

<TT> … Text with typewriter font goes here... </TT>

<DFN> tag : It is a container tag. Using this tag we can define the instance of a term. Some browsers display it in italic text.

Syntax :

<DFN> … Term to be defined goes here... </DFN>

<EM> tag : It is a container tag. Using this tag we can obtain emphasized text against the enclosed text. Most of the browsers display it in italic text.

Syntax :

<EM> … Emphasized text goes here... </EM>

<STRONG> tag : It is a container tag. Using this tag we can obtain strongly emphasized text against the enclosed text. Most of the browsers display it in boldface text.

Syntax :

<STRONG> … Strongly emphasized text goes here...
</STRONG>

**
 tag :** It is a standalone tag. It is used to insert a line break in the document. Even if we write the text in different line while designing the documents, browser will not display them in different lines. i.e. Browser do not understand carriage return.

Syntax :

<CENTER> tag : It is a container tag. Using this tag we can align the page contents into centre position.

Syntax :

<CENTER> … Page content with center alignment...
</CENTER>

<DIV> tag : It is a container tag. It contains division of document, which requires special alignment.

Syntax :

<DIV Align="left| right| center| justify">

… Text to be aligned...

</DIV>

<HR> tag : It is a standalone tag. It is used to insert a horizontal line on the page.

Syntax :

<HR Align="left|right|center" Noshade Size="thickness" Width="percentage of screen or no. of pixels">

Attributes :

Align: default is center.

Noshade: It suppresses the shading effect.

Size: It indicates thickness of the line in pixels.

Width: Here we can set the width of the line in two ways- percentage of the screen or number of pixels.

<H1>- <H6> tag : These are container tags. Using these six tags we can set six different levels of heading. <H1> creates the largest possible heading whereas <H6> creates the smallest possible heading. Each heading will be of boldface text with little extra line spacing.

Syntax :

<Hn Align="left|right|center|justify">

Level n heading

</Hn>

Where n may be any digit between 1 and 6.

<P> tag : It is a container tag. Using this tag we can denote a paragraph. Multiple uses of <P> tags make no difference. i.e. Browser ignores the use of multiple tags.

Syntax :

<P Align="left|right|center|justify">

Paragraph text

</P>

<PRE> tag : It is a container tag. It is used to display pre-formatted text. i.e. The enclosed text will be displayed as it is.

Syntax :

<PRE Width="Width of the widest line"> Preformatted text
</PRE>

<SPAN> tag : It is a container tag. It is used to apply some of the style information.

Syntax :

<SPAN Style="style information" Align="left|right| center| justify">
Text on which style is to be applied.
</SPAN>



CHECK YOUR PROGRESS

Q.2. What is the output of the following two HTML codes:

a) <P><PRE>

Name	City	PO	Pin
Mantu	Ghy	Chand	781003
Nantu	Ghy	Ulu	781007
Badan	Nag	Pani	782410

</PRE></P>

b) <P>

Name	City	PO	Pin
Mantu	Ghy	Chand	781003
Nantu	Ghy	Ulu	781007
Badan	Nag	Pani	782410

</P>

Q.3. Select the correct answer :

a) Using <H1>/<H6> tag smallest heading can be produced.

b) Both <BASEFONT> and <FONT> tags can set font; they have differences/no differences.

c) Multiple <BR> tag produces multiple/single blank line.

d) Multiple <P> tag produces more gaps/ same gap between paragraphs.

10.6 LIST TAGS

HTML has several different lists such as:

- Ordered List
- Unordered List
- Menu List
- Directory List
- Definition List

Most of the list uses <LI> tag.

<LI> tag : It is a container tag. Using this tag we can denote an item in a list.

Syntax :

<LI Type="List-type" Start="Start-value" > List Item </LI>

Example :

<LI Type="A" Start="2" > Elephant</LI>

è Output :

A. Elephant

<OL> tag : It is a container tag. Using this tag we can create an ordered list.

Syntax :

<OL Type="1|A|a|I|I" Start="Start-value" Compact>

<LI> List Item 1 </LI>

<LI> List Item 2 </LI>

…

</OL>

Example :

<OL Type="a" >

<LI> COW</LI>

<LI> CAT</LI>

<LI> DOG</LI>

</OL>

è Output :

a. COW

b. CAT

c. DOG

<UL> tag : It is a container tag. Using this tag we can create an unordered list. It is normally known as bulleted list.

Syntax :

```
<UL Type="disc| square| circle" Compact>
    <LI> List Item 1 </LI>
    <LI> List Item 2 </LI>
    …
</UL>
```

Example :

```
<UL Type="circle" >
    <LI> COW</LI>
    <LI> CAT</LI>
    <LI> DOG</LI>
</UL>
```

è Output :

- COW
- CAT
- DOG

<MENU> tag : It is a container tag. Using this tag we can create a menu list. It is a short list and is arranged in a single column.

Syntax :

```
<MENU Compact>
    <LI> List Item 1 </LI>
    <LI> List Item 2 </LI>
    …
</MENU>
```

Example :

```
<MENU Compact >
    <LI> CUT</LI>
    <LI> COPY</LI>
    <LI> PASTE</LI>
</MENU>
```

è Output :

- CUT
- COPY
- PASTE

<DIR> tag : It is a container tag. Using this tag we can create a directory list. It is also a short list and is arranged in a single column.

Syntax :

```
<DIR Compact>
        <LI> List Item 1 </LI>
        <LI> List Item 2 </LI>
        …
</DIR>
```

Example :

```
<DIR Compact >
        <LI> Ramen, Chandmari</LI>
        <LI> Gakul, Jorhat</LI>
        <LI> Bandita, Nalbari</LI>
</DIR>
```

è Output :

- Ramen, Chandmari
- Gakul, Jorhat
- Bandita, Nalbari

<DL> tag : It is a container tag. Using this tag we can create a definition list.

Syntax :

```
<DL Compact>
        Term to be defined and definition of the term.
        …
</DL>
```

Example :

```
<DL>
        <DT>Folder</DT>
        <DD>It is a container of files and subfolders. </DD>
```

```
<DT>Icon</DT>
<DD>It is a graphical representation of a program or
file.  </DD>
</DL>
```

è Output :

Folder :

It is a container of files and subfolders.

Icon :

It is a graphical representation of a program or file.

<DT> tag : It is a container tag. It is used in a definition list to denote the term to be defined.

Syntax :

```
<DT>  term to be defined </DT>
```

Example :

```
<DL>
    <DT>Cow: </DT>
    <DD>A cow is a four footed animal. </DD>
</DL>
```

è Output:

Cow :

A cow is a four footed animal.

<DD> tag : It is a container tag. It is used in a definition list to denote the definition of the term.

Syntax :

```
<DD> Definition of the term. </DD>
```

Example :

```
<DL>
    <DT>Internet: </DT>
    <DD>Internet is network of computer networks.
    </DD>
</DL>
```

è Output :

Internet :

Internet is network of computer networks.



CHECK YOUR PROGRESS

Q.4. Fill in the gaps :

a) The _____ tag is used to denote list items.

b) The _____ tag can also be termed as bulleted tag.

c) For defining a scientific formula _____ tag is best suitable.

d) You can use _____ tag for designing a list of files and directories.

Q.5. State true or false :

a) The ordered list is used to denote bulleted list.

b) The menu list is best suitable for a lengthy list of many columns.

c) The <DT> tag is used in a <DIR> list to denote term to be defined.

d) The circle option in <UL> tag produces a solid circle as bullet.

10.7 HYPERLINK AND IMAGE TAGS

<A> tag : Using this tag we can create hyperlink. Hyperlink is created for the text found between <A>and </A> pair. The URL given against the HREF attribute specifies the target document.

Syntax :

<A HREF=”URL” Target=”Frame-Name” Accesskey=”Key-letter”>

Hyperlinked Term.

</A>

Attributes :

HREF=here URL of target document is to be specified.

Target=It is the name of frame to which target document is to be loaded.

Accesskey= It is the shortcut key, using which also target document

can be accessed from keyboard by pressing accesskey and ALT key together.

Example :

```
<A HREF="c:\my documents\test.html" Accesskey="T">Test
</A>
```

<IMG> tag : It is a standalone tag. Using this tag we can place an image into a document.

Syntax :

```
<IMG SRC="URL" Width="width in pixel" Height="Height in
pixel" Border="Thickness in pixel" Align="Alignment"
Hspace="Horizontal spacing" Vspace="Vertical spacing"
ISMAP Usemap="Map-name">
```

Attributes :

SRC=here URL of the Image file is to be specified.

Width=Width of the image in pixel

Height=Height of the image in pixel

Border=Thickness of the border in pixel around the image.

Align=It controls how text flows around the image such as: Top, Middle, Bottom, Left or Right.

Hspace=Horizontal spacing in pixel around left and right side of the image.

Vspace=Vertical spacing in pixel around top and bottom side of the image.

ISMAP= Identifies the image to be used as imagemap.

Usemap=Set the name of imagemap.

<MAP> tag : It is a container tag. Using this tag we can create a clickable region, which can be termed as hot region.

Syntax :

```
<MAP Name="Map name">
        Definition of hot region
</MAP>
```

Attributes :

Name=here Map name should be specified and it should be unique within the Web page so that this can be correctly referred by USEMAP attribute of <IMG> tag.

<AREA> tag : It is a standalone tag. Using this tag we can define a hot region in a imagemap.

Syntax :

<AREA Shape="RECT| CIRCLE| POLY| DEFAULT" Coords="co-ordinate lists" HREF="URL" Target="frame name" Accesskey="key letter" NOHREF>

Attributes :

Shape= It specifes the shape of the hot region possible values may be Rect, Circle, Poly Or Default.

Coords=list of co-ordinate which specifies the hot region.

HREF=URL of linked document.

Target=Name of frame to which linked document is to be loaded.

Accesskey=defines the shortcut key

NOHREF=It deactivates the hot regions.

10.8 TABLE TAGS

<TABLE> tag : It is a container tag. Using this tag we can denote a table. All the related tags for creating a table are included within this tag.

Syntax :

<Table Align="alignment" Border="in pixel" Bgcolor="color" Width="pixel or percentage of browser" Cols="no. of columns" Cellspacing="pixel" Cellpadding="pixel" Frame="outer border" Rules="Inner border">

…

</Table>

Attributes :

Align=alignment such as left, right, center

Border=Thickness in pixels

Bgcolor=Background color of the table.

Width=Pixel or percentage of browser width

Cols=Number of columns.

Cellspacing=Number of pixels between cells.

Cellpadding= Number of pixels between cell border and the content.

Frame=It is used for outer border rendering and it can be any one of the following :

ABOVE: Displays border on top of table.

BELOW: Displays border at the bottom of table.

BORDER: Displays border on all the four sides of table.

BOX: Displays border on all the four sides of table.

HSIDES: Displays border on left and right sides of table.

LHS: Displays border on the left side of table.

RHS: Displays border on the right side of table.

VSIDES: Displays border on top and bottom side of table.

VOID: Off the borders.

Rules= It is used for inner border rendering and it can be any one of the following:

ALL: Displays border between all rows and columns.

COLS: Displays border between all columns.

GROUPS: Displays border between all logical groups like:

THEAD, TBODY TFOOT COLGROUP etc.

NONE: Off all inner borders.

ROWS: Displays border between all rows.

<CAPTION> tag : It is a container tag. Using this tag we can denote caption of a table.

Syntax :

```
<CAPTION Align="left|right|top|bottom">
        Caption Text
</CAPTION >
```

<THEAD> tag : It is a container tag. Using this tag we can denote header section of a table.

Syntax :

```
<THEAD Align=" left| right| justify| center"
Valign="top|middle|bottom|baseline">
        Rows of the header.
</THEAD>
```

Attribute :

Align: Horizontal alignment.

Valign: Vertical alignment.

<TFOOT> tag : It is a container tag. Using this tag we can denote footer section of a table.

Syntax :

```
<TFOOT Align="left| right| justify| center"
Valign="top|middle|bottom|baseline">
        Footnote.
</TFOOT>
```

Attribute :

Align: Horizontal alignment.

Valign: Vertical alignment.

<TBODY> tag : It is a container tag. Using this tag we can denote the body section of a table.

Syntax :

```
<TBODY Align="left| right| justify| center"
Valign="top|middle|bottom|baseline">
        Footnote.
</TBODY>
```

Attribute :

Align: Horizontal alignment.

Valign: Vertical alignment.

<COLGROUP> tag : It is a container tag. Using this tag we can denote a grouping of columns, which shares the same properties. i.e. we can set properties to that group of columns as a whole rather than setting properties for each column individually.

Syntax :

```
<COLGROUP Span=”no. of columns” Width=”width of
column group” Align=”left| right| justify| center”
Valign=”top|middle|bottom|baseline”>
    <COL>
    …
</COLGROUP>
```

Attribute :

Span: It specifies no. of columns in a column group.

Align: Horizontal alignment.

Valign: Vertical alignment.

<COL> tag : It is a standalone tag. Using this tag we can set properties of a column.

Syntax :

```
<COL Span=”no. of columns” Width=”width of column sub-
group” Align=”left| right| justify| center”
Valign=”top|middle|bottom|baseline”>
```

Attribute :

Span: It specifies no. of columns in a column group.

Align: Horizontal alignment.

Valign: Vertical alignment.

<TR> tag : It is a container tag. Using this tag we can denote a row of a table.

Syntax :

```
<TR Align=”left| right| justify| center”
Valign=”top|middle|bottom|baseline”>
    <TD>…</TD>
    …
</TR>
```

Attribute :

Align: Horizontal alignment.

Valign: Vertical alignment.

<TD>/<TH> tag : It is a container tag. Using these tags we can

create cells of a table. <TH> is used to create cells of header section and <TD> is used to create cells of <TBODY> section. The cells created with <TH> become bold with center alignment.

Syntax :

```
<TH Align="left| right| justify| center"
Valign="top|middle|bottom| baseline" NOWRAP
Colspan="no. of columns" Scope="row|col|
rowgroup|colgroup">
        Header item
</TH>
```

or

```
<TD Align="left| right| justify| center"
Valign="top|middle|bottom| baseline" NOWRAP
Colspan="no. of columns" Scope="row|col|rowgroup|
colgroup">
        Table item
</TD>
```

Attribute :

Align: Horizontal alignment.

Valign: Vertical alignment.

NOWRAP: Suppress text wrapping within a cell.

Scope: It lets the users to create groups of row, columns, row-groups and column-groups.

109 FRAME TAGS

In a framed layout browser window can be divided into smaller regions called **frames**. Framed layout gives a better look to a web page. A few frame tags are used to create a framed layout. When we use the framed layout the <BODY> tag should not be used.

<FRAMESET> tag : It is a container tag. Using this tag we can divide the browser window into frames. It does the job in two ways: divides the window either *horizontally* or *vertically*.

Syntax :

<FRAMESET Rows=”list of row sizes” Cols=”list of column sizes”>

… Frame tags

</FRAMESET>

Attribute :

Though it has two attributes, we can use only one at a time.

Rows: It divides the browser window horizontally. Here we need to specify the size of individual fragments in pixel, in percentage of browser’s window or an asterisk mark (i.e. *). Asterisk mark is used to indicate the remaining horizontal portion.

Column: It divides the browser window vertically. Here also we need to specify the size of individual fragments in pixel, in percentage of browser’s window or an asterisk mark (i.e. *). Asterisk mark is used to indicate the remaining vertical portion.

<FRAME> tag : It is a standalone tag. Using this tag we can place contents into a frame.

Syntax :

<FRAME Src=”URL of document” Name=”frame name” Frameborder=”0|1" Marginwidth=”width in pixel” Marginheight=”height in pixel” NORESIZE Scrolling=”yes|no|auto”>

Attribute :

Src=Here we need to specify the URL of the document to be placed.

Name=Name of frame can be specified here.

Frameborder=It is either 0 to indicate no border or 1 to turn on the frame border.

Marginwidth=It specifies the size of left margin in pixel.

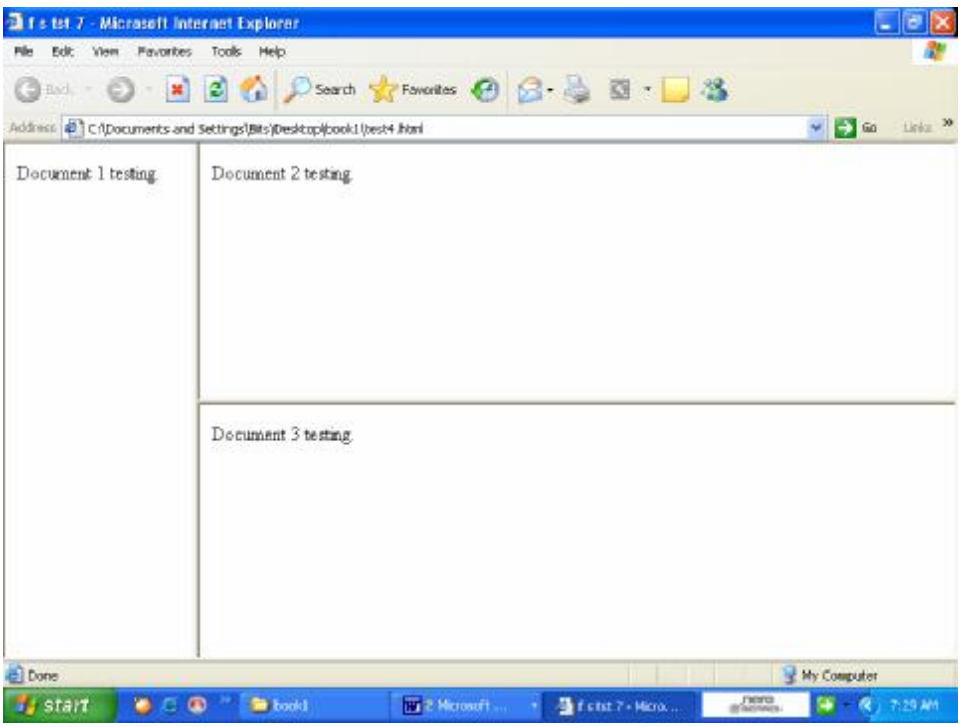
Marginheight= It specifies the size of top margin in pixel.

NORESIZE: If we use this option the user will not be able to drag and drop the frame border.

Scrolling=It can be yes, no or auto

Example :

```
1.  <FRAMESET Cols="20%, *">
        <Frame Src="doc1.html">
        <FRAMESET Cols="50%, *">
            <Frame Src="doc2.html">
            <Frame Src="doc3.html">
        </FRAMESET>
    </FRAMESET>
```



<IFRAME> tag : It is a container tag. Using this tag we can place a floating frame into a page. Float frame is one which you can place anywhere else within the page.

Syntax :

```
<IFRAME Src="URL of document" Name="frame name" Frameborder="0|1" Width="width in pixel or percentage" Height="height in pixel or percentage" Margin width=" width in pixel" Marginheight="height in pixel" NORESIZE Scrolling="yes|no|auto" Align="alignment">
    … Alternative text. …
</IFRAME>
```

Attribute :

Src=Here we need to specify the URL of the document to be placed.

Name=Name of frame can be specified here.

Frameborder=It is either 0 to indicate no border or 1 to turn on the frame border.

Width= Width of the frame in pixel or percentage of the browser window.

Height= Height of the frame in pixel or percentage of the browser window.

Marginwidth=It specifies the size of left margin in pixel.

Marginheight= It specifies the size of top margin in pixel.

NORESIZE: If we use this option user will not able to drag and drop frame border.

Scrolling=It can be yes, no or auto

Align= Top, middle, bottom, left or right alignment.

10.10 FORM TAG

HTML Forms are used to select different kinds of user input and to pass data to a server.

A form can contain input elements like text fields, checkboxes, radio-buttons, submit buttons and more. A form can also contain select lists, text area, field set, legend, and label elements.

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

The Input Element : The most important form element is the input element. The input element is used to select user information. An input element can vary in many ways, depending on the type attribute. An input element can be of type text field, checkbox, password, radio button, submit button, and more.

The most used input types are described below.

Text Fields :

<input type="text" /> defines a one-line input field that a user can enter text into:

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

How the HTML code above looks in a browser :

First name:

Last name:

Note : The form itself is not visible. Also note that the default width of a text field is 20 characters.

Password Field :

<input type="password" /> defines a password field:

```
<form>
Password: <input type="password" name="pwd" />
</form>
```

How the HTML code above looks in a browser:

Password :

Note : The characters in a password field are masked (shown as asterisks or circles).

Radio Buttons :

<input type="radio" /> defines a radio button. Radio buttons let a user select ONLY ONE one of a limited number of choices:

```
<form>
<input type="radio" name="sex" value="male" /> Male<br />
<input type="radio" name="sex" value="female" /> Female
</form>
```

How the HTML code above looks in a browser:

- Male
- Female

Checkboxes :

<input type="checkbox" /> defines a checkbox. Checkboxes let a user select ONE or MORE options of a limited number of choices.

```
<form>
<input type="checkbox" name="vehicle" value="Bike" /> I have a bike<br />
<input type="checkbox" name="vehicle" value="Car" /> I have a car
</form>
```

How the HTML code above looks in a browser:

- I have a bike
- I have a car

Submit Button :

<input type="submit" /> defines a submit button.

A submit button is used to send form data to a server. The data is sent to the page specified in the form's action attribute. The file defined in the action attribute usually does something with the received input:

```
<form name="input" action="html_form_action.asp" method="get">
Username: <input type="text" name="user" />
<input type="submit" value="Submit" />
</form>
```

How the HTML code above looks in a browser:

Username : Submit



10.11 LET US SUM UP

- HTML, which stands for Hyper Text Markup Language, is the Universal language on which the web documents(pages) are written (eg. especially pages on the World Wide Web)
- In HTML, a tag tells the browser what to do. When you write an HTML page, you enter tags for many reasons — to change the appearance of text, to show a graphic, or to make a link to another page. HTML tags are divided into two types. Container tags and Standalone tags or empty tags.
- The extension of the HTML file is written as **.html** or **.htm**
- A web page created in HTML consists of three distinct sections- Comment section, Header section, and Body section.
- Formatting tags are used to change the typographic formatting of a page.
- HTML has several different lists such as- Ordered List, Unordered List, Menu List, Directory List and Definition List.
- Hyperlink tags are used to create hyperlink for the text found between <A>and </A> pair.
- Table tag is a container tag which is uses to denote a table.
- Frame tags are used to create a framed layout. When we use the framed layout the <BODY> tag should not be used.
- The form tag is used to create an HTML form.



10.12 ANSWERS TO CHECK YOUR PROGRESS

Ans. to Q. No. 1 : a) False, b) True, c) False, d) True

Ans. to Q. No. 2 : a) HTML document, b) Text color, c) Head, d) Background, green

Ans. to Q. No. 3 : Output of a) Name City PO Pin

	Mantu	Ghy	Chand	781003
	Nantu	Ghy	Ulu	781007
	Badani	Nag	Pani	782410
Output of b)	Name	City	PO	Pin
	Mantu	Ghy	Chand	781003
	Nantu	Ghy	Ulu	781007
	Badani	Nag	Pani	782410

Ans. to Q. No. 4 : a) H6, b) Differences, c) Multiple, d) Same gap

Ans. to Q. No. 5 : a) <LI>, b) <UL>, c) <DL>, d) <DIR>

Ans. to Q. No. 6 : a) False, b) False, c) False, d) False



10.13 FURTHER READING

- **Data Communications and Networking** – Behrouza A Forouzan (Tata McGraw-Hill)
- **Computer Networks** *-A. S. Tanenbaum*
- **Web Technologies** - *A.S. Godbole and Atul Kahate.*
- Link for online tutorial : www.w3schools.com
- Using HTML, XML and Java by Eric Ladd and Jim O'Donnell, PHI.



10.14 MODEL QUESTIONS

Q.1.What are meta tags and why are they is used?

Q.2.What is the difference between width="100" and width="100%".

Q.3.What is DocType?

Q.4.How do you create frames? What is a frameset?

Q.5.What is a Hypertext link?

Q.6.Is it possible in HTML to add more than one submit button having different actions in a single form? Explain.

****

UNIT 11 : ADVANCED HTML TAGS

UNIT STRUCTURE

11.1 Learning Objectives

11.2 Introduction

11.3 HTML Form

11.4 Additional Advanced HTML Tags

11.5 Let Us Sum Up

11.6 Answers to Check Your Progress

11.7 Further Reading

11.8 Model Questions

11.1 LEARNING OBJECTIVES

After going through this unit, you will able to :

- identify the different HTML FORM elements like Text Box, Combo Box, Checkbox, Radio Button etc.
- explain about the additional advanced HTML Tags

11.2 INTRODUCTION

In the previous unit you have learnt about the Static Web page design. This unit will deal with the Advanced HTML tags and an introduction to XML. This unit provides a description of the various HTML Form elements. The unit also describes some additional advanced HTML tags such as Executable Content tags, Cascading Style Sheet (CSS), Class, External style sheet (ESS) and SPAN tag. Finally this unit will focus on the basic concept of extensible Markup Language (XML).

11.3 HTML FORM

FORM is a powerful tool of HTML for developing interactive web documents. It has several elements for developing different buttons or boxes.

<FORM> tag is a container tag.

Syntax :

```
<FORM Name="form name" Method="Get or Post">
    Form elements
</FORM>
```

The method can be GET when small amount data are to be sent to the server. The maximum limit is 1024 byte. The POST method is used for large data and in this case data is sent as a separate bit-stream.

The form elements may be text box, text area, password, radio button, checkbox, combo box hidden field, image, submit button, reset button etc.

TEXT BOX : It is used to capture one line of textual data from user in a web page. The <INPUT> tag is used for the purpose. <INPUT> is a stand alone tag.

Syntax :

```
<INPUT Type="Text" Name="Textbox Name" Size="size of box" Maxlength="maximum length of text to be received" Tabindex="number for controlling tab order"/>
```

TEXT AREA : It is also used to capture text from the user. It is a container tag. Here we can obtain multiple lines of texts instead of a single line of the text as in the case of Text Box.

Syntax :

```
<TEXTAREA Name="Textarea Name" Rows="number of rows" Cols="number of columns">
</TEXTAREA>
```

Password : It is used to capture password from user in a web page. Like Text Box Password also uses <INPUT> tag. Whenever a text typed in this box the letters are hidden.

Syntax :

```
<INPUT Type="Password" Name="password name" Size="size of box" Maxlength="maximum length of password text to be received"/>
```

RADIO BUTTON : It is also termed as option button and it is used to capture the user's selection from several option buttons.

Syntax :

<INPUT Type="Radio" Name="radio button name" Value="caption" Checked="true/false" Tabindex="number for controlling tab order"/>

CHECKBOX : It is similar to Radio Button but here we can select multiple options simultaneously.

Syntax :

<INPUT Type="Checkbox" Name="checkbox name" Value= "caption" Checked="true/false" Tabindex="number for controlling tab order"/>

Combo Box : It holds a list of options. It initially displays only one option which is declared by using <OPTION SELECTED>tag. Once the user clicks it a dropped down list will appear and the user can select any one option from that list. It uses a container tag <SELECT> for the purpose. We can use <OPTION> or <OPTION SELECTED> tags within <SELECT> tag, which are again container tags.

Syntax :

<SELECT Name="selectbox name" Size="number; to specify no. of items to display at a time; default 1" Tabindex="number for controlling tab order"/>

<OPTION [SELECTED]> Opt1 </OPTION>

<OPTION [SELECTED]> Opt2 </OPTION>

<OPTION [SELECTED]> Opt3 </OPTION>

.....

</SELECT>

HIDDEN FIELD : Sometimes it is required to isolate some data from other form data. Or sometimes it is needed to hide some data from the user and send directly to the server. In that case we can use hidden field.

Syntax :

<INPUT Type="Hidden" Name="hidden field name"

Value="hidden data: may be I or D or U"/>

Where "I" means Input, "D" means Delete and "U" means Update.

IMAGE : It can be used as a clickable element in a form. It gives a graphical look of the clickable area.

Syntax :

<INPUT Type="Image" Name="image name" Src="path to image file" Width ="width in pixel" Height="height in pixel" Border="border size in pixel"/>

Submit and Reset Button : After placing all the other elements in the <FORM> tag next requirement is to send them to web server. The Submit and Reset Buttons are used for that purpose. Submit Button and Reset Button are discussed below:

SUBMIT BUTTON : It is used to submit the data captured by the form using different form elements.

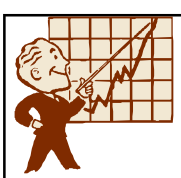
Syntax :

<INPUT Type="Submit" Name="submit button name" Value= "caption"/>

RESET BUTTON : It is used to clear the values of all the buttons set by different form elements. Sometimes it becomes necessary to reset the buttons before capturing data.

Syntax :

<INPUT Type="Reset" Name="reset button name" Value= "caption"/>



CHECK YOUR PROGRESS

Q.1. State true or false :

a) Textbox can receive multiple lines of text as input data.

b) Using checkbox we can select any one option from multiple options.

c) The submit button can send captured data to web server.

d) POST method of FORM tag has a limitation of sending maximum 1024 bytes of data.

11.4 ADDITIONAL ADVANCED HTML TAGS

EXECUTABLE CONTENT TAGS : The executable content tags can add more dynamism to a web page. Dynamic content can be developed by using Java applet and ActiveX control. The executable content tags are: <APPLET> tag, <OBJECT> tag and <PARAM> tag.

<APPLET> tag : It is a container tag using which we can place a Java applet into a web page.

Syntax :

<APPLET Width=”in-pixel” Height=”width_in_pixel” Codbase=“base_URL_for_applet” Code=”applet_class_file” Name=”applet _name” Align=”top/middle/bottom/left/right” Hspace=”horizontal white space” Vspace=”vertical white space” >

….

</APPLET>

<PARAM> tag : It is a stand-alone tag using which we can pass parameter to a Java applet or other executable object.

Syntax :

<PARAM Id=”unique identifier” Name=”parameter_name” Value=”Parameter_value” Valuetype=”type of value: Data/Ref/Object” Type=”executable content type” >

<OBJECT> tag : It is a container tag using which we can place an executable object in a web page.

Syntax :

< OBJECT Width=”in-pixel” Height=”width_in_pixel” Codbase= “URL_of_ object” Codetype=”type of code” Data=”URL of data” Type=”type of data” Usemap=”map_name” Name=”object _name” Align=”top/ middle/bottom/left/right” Hspace=”horizontal white space” Vspace=”vertical white space” Border=”in_pixel”

```
Tabindex=”tab_position” >
….
</OBJECT>
```

CASCADING STYLE SHEET (CSS) : CSS is a powerful tool for adding style to a web page and to ensure uniformity throughout the page. It also adds dynamism to a web page with the help of its wide range of attributes. **Dynamic HTML** is possible due to CSS and scripting language. Mainly CSS helps in formatting a web page in an efficient way.

Syntax :

```
<STYLE Type=”text/css”>
    Tag {attribute:value; attribute:value …}
    …
</STYLE>
```

Description of different attributes which can be used in a <STYLE> tag.

Attributes	Values
1. Font Attributes	
font-family	font names (Arial, Helvetica etc.)
font-style	normal/italic/oblique
font-weight	normal/bold/bolder/lighter/100/200 /300/400/500/600/700/800/900
font-size	absolute size(xx-small/x-small/small/ medium/large/x-large), relative size (larger /smaller), number of pixel, percentage of parent element’s size
2. Color and Background Attribute	
color	a color name or color code
background-color	a color name or color code
background-image	a URL of a image/none
background-repeat	repeat-x (means horizontal)/repeat-y (means vertical)/ repeat (both direction)/no-repeat

3. Text Attribute

text-decoration	none/underline/overline/line-through/ blink
vertical-align	baseline/sub/super/top/text-top/ middle/bottom/text-bottom/% of element's height
text-transform	capitalize/uppercase/lowercase/none
text-align	left/right/center/justify
text-indent	a length / % of element's width

4. Border Attribute

border-style	solid/double/groove/ridge/inset/outset
border-color	a color name or color code
border-width	thin/medium/thick/a length
border -top-width	thin/medium/thick/a length
border-bottom-width	thin/medium/thick/a length
border-left-width	thin/medium/thick/a length
border-right-width	thin/medium/thick/a length
border-top	specify width/color/style
border-bottom	specify width/color/style
border-left	specify width/color/style
border-right	specify width/color/style
border	set all properties

5. Margin Attribute

margin-top	percent/length/auto
margin-bottom	percent/length/auto
margin-left	percent/length/auto
margin-right	percent/length/auto
margin	percent/length/auto

6. List Attribute

list-style	disc/cilcle/square/decimal/lower-roman/upper-roman/lower-alpha/ upper-alpha/none

Example :

```
<HTML>
    <HEAD>
        <TITLE> Use of Style Sheets with Font Attributes</TITLE>
        <STYLE Type = “text/css”>
            H1 {font-family:arial; font-size:26pt;text-
            decoration:overline; color:white; background-image:
            url(spb2.jpg);background-repeat:repeat-x;
            text-align = center}
            H2{font-family:helvetica;background-color:yellow}
            P{font-size:14pt; font-style:italic; border-color:blue;
            border-style:groove;border-width:thick;
            margin-left:20%; margin-right:10% ;
            background-image:url(spb2.jpg);
            background-repeat:no-repeat}
            UL {list-style-type:lower-roman}
        </STYLE>
    </HEAD>
    <BODY >
        <H1>HTML Tags</H1>
        <H2>HTML Tags are of two types :</H2>
        <UL><LI>Stand-alone</LI> <LI>Container</LI>
        </UL>
        <P>The Stand-alone tag appears only once. It affects the
        position where it is used. But container tags appear in pairs.
        i.e., this tag has a starting tag and an ending tag</P>
    </BODY>
    </HTML>
```

CLASS : CSS is applicable to the whole page. Using class we can control a part of the page such as a paragraph or a few paragraphs. In a style sheet class is defined by a dot followed by the name of the class.

```
<HTML>
```

```
<HEAD>
<TITLE> example of class </TITLE>
<STYLE Type = “text/css”>
    P{font-size:14pt;  border-color:blue;
    border-width:thick; margin-left:20%;
    margin-right:10%}
    .term {color:red; font-style:italic}
    .defn {color:brown}
    </STYLE>
</HEAD>
<BODY>
    <P class=’term’> Cascading Style Sheet (CSS)</P>
    <P class=’defn’> CSS is a powerful tool for adding style to a
    web page and ensure uniformity thorough out the page. It
    also adds dynamism to a web page with the help of its wide
    range of attributes.</P>
</BODY>
</HTML>
```

EXTERNAL STYLE SHEET (ESS) : The external style sheets are stored in a separate file with extension .css. We can include all the style facilities whatever we have used in case of CSS. This .css file is linked to the web page with the help of <LINK> tag where it is required. The <LINK> is a standalone tag and it is applied in the header section of a page. Its syntax is shown below:

```
<LINK  Rel=stylesheet Href=”the .css file name”>
```

eg.

Say **test.css** is the external file which contains the following contents:

```
P{font-size:14pt;  border-color:blue;
border-width:thick; margin-left:20%;
margin-right:10%}
.term {color:red; font-style:italic}
.defn {color:brown}
```

This test.css is applied in a HTML file as follows:

```
<HTML>
    <HEAD>
    <TITLE> example of ESS </TITLE>
        <LINK Rel=stylesheet Href=”test.css” >
    </HEAD>
    <BODY>
        <P class=’term’> Cascading Style Sheet (CSS)</P>
        <P class=’defn’> CSS is a powerful tool for adding
        style to a web page and ensure uniformity thorough
        out the page. It also adds dynamism to a web page
        with the help of its wide range of attributes.</P>
    </BODY>
</HTML>
```

SPAN TAG : We can use this tag to modify part of a style sheet.

e.g.

```
<HTML>
    <HEAD>
    <TITLE> example of class </TITLE>
    <STYLE Type = “text/css”>
        P{font-size:14pt;  border-color:blue;
        border-width:thick; margin-left:20%;
        margin-right:10%}
        .term {color:red; font-style:italic}
        .defn {color:brown}
        .hi {font-size:16pt;color:blue}
        </STYLE>
    </HEAD>
    <BODY>
<P class=’term’> Cascading Style Sheet <SPAN
class=”hi”>(CSS)</SPAN></P>
        <P class=’defn’> <SPAN class=”hi”> CSS </SPAN>
        is a powerful tool for adding style to a web page and ensure
        uniformity thorough out the page. <SPAN class=”hi”> CSS
```

```
</SPAN> also adds dynamism to a web page with the help
of its wide range of attributes.</P>
</BODY>
</HTML>
```



CHECK YOUR PROGRESS

Q.2. Fill up the gaps:

a) _____ is similar to CSS, but it has an external file with extension _____ to include style.

b) Stylesheet is included in the _____ section using _____ tag.

c) Applet and Param are the executable _____ tag.

d) Class is used to modify _____ of a web-page, but _____ is applicable to the whole page.



11.5 LET US SUM UP

- FORM is a powerful tool of HTML for developing interactive web documents.
- Text box is used to capture one line of textual data from user in a web page.
- The executable content tags can add more dynamism to a web page. Dynamic content can be developed by using Java applet and ActivX control. The executable content tags are: <APPLET> tag, <OBJECT> tag and <PARAM> tag.
- CSS is a powerful tool for adding style to a web page and ensure uniformity thorough out the page.

11.6 ANSWERS TO CHECK YOUR PROGRESS

Ans. to Q. No. 1 : a) False, b) False, c) True, d) False

Ans. to Q. No. 2 : a) ESS, .css, b) Head, STYLE, c) Content
d) Part, CSS

11.7 FURTHER READINGS

- **Web enabled commercial Application Development using HTML, Javascript, DHTML and PHP** *–Ivan Bayross, BPB*
- **Web Technologies** *–A.S. Godbole and Atul Kahate, Tata McGraw-Hill.*
- Using HTML, XML and Java *–Eric Ladd and Jim O'Donnell, PHI*
- **Fundamental of Database Systems** *–Ramez Elmasri and Shamkant B. Navathe, Pearson.*

11.8 MODEL QUESTIONS

Q.1. Give an introduction to forms.

Q.2. What is the difference between radio button and checkbox.

Q.3. Design the following form :

Application for Admission

1. Name :
2. Address :
3. Roll No.

Submit

****

UNIT 12 : HTML LINKS AND STYLES

UNIT STRUCTURE

12.1 Learning Objectives

12.2 Introduction

12.3 Hypertext, Hyperlink and Hypermedia

12.4 Anchor and URL

12.5 Creating tables

12.6 Frames

12.7 Forms

12.8 E-mailing

12.9 Let Us Sum Up

12.10 Answers To Check Your Progress

12.11 Further Reading

12.12 Model Questions

12.1 LEARNING OBJECTIVES

After going through this unit, you will be able to:

- define hypertext, hyperlink and hypermedia
- learn the use of anchors and URLs
- create links to external documents
- use tables in a page
- create frames inside a page
- learn about forms and their uses.

12.2 INTRODUCTION

In the previous unit, we learned about HTML, which is the basic markup language used for building the primary structure of any web-page. We also got familiarized to the various HTML tags and their attributes that were used to mark and display web-pages in a variety of ways. Manipulating text on a page which is a way of editing text features and appearance was

also discussed. Lists, which are used to organize textual data in a well conceivable way was also introduced along with the different types of lists that are used.

In this unit we will be introduced to some more advanced concepts of creating pages that deal with the terms of Hypertext, Hyperlinks and Hypermedia. Tables will also be introduced that are used to display data in a tabular format. Form handling which is an integral part of any site dealing with using and processing user related data will also be discussed.

12.3 HYPERTEXT, HYPERLINK AND HYPERMEDIA

Hypertext : Hypertext means a body of written or pictorial material interconnected in such a complex way that it could not conveniently be presented or represented. Hypertext is text which is not constrained to be linear. It is text displayed on a computer or other electronic device with references (hyperlinks) to other text that the reader can immediately access, usually by a mouse click or keypress sequence. Apart from running text, hypertext may contain tables, images and other presentational devices. Hypertext is the underlying concept defining the structure of the World Wide Web, making it an easy-to-use and flexible format to share information over the Internet. The prefix 'hyper-' signifies the overcoming of the old linear constraints of written text. Hypertext documents can either be static (prepared and stored in advance) or dynamic (continually changing in response to user input). Static hypertext can be used to cross-reference collections of data in documents, software applications, or books on CDs. A well-constructed system can also incorporate other user-interface conventions, such as menus and command lines. Hypertext can develop very complex and dynamic systems of linking and cross-referencing. The most famous implementation of hypertext is the World Wide Web. Machine-readable text that is not sequential but is organized so that related items of information are connected is hypertext.

Hyperlinks : Hypertext may at times present information where instead of reading or learning about things in the order in a predefined way,

readers may follow their own path, create their own order. This is accomplished by creating "**links**" between texts. These links are provided so that the readers may "jump" to further information about a specific topic being discussed (which may have more links, leading each reader off into a different direction). Hyperlink helps to create links between various web pages.

Example :

Go

The **start-tag** contains attributes about the link. The element content "Go" defines the part to be displayed. The **href** attribute defines the link "address". The **target attribute** defines **where** the linked document will be opened. The code above will open the document in a new browser window.

Hypermedia : Hypermedia is a superset of hypertext that simply combines hypertext and multimedia. Hypermedia documents contain links not only to other pieces of text, but also to other forms of media - sounds, images, video etc. Images themselves can be selected to link to sounds or documents. This means that browsers might not display a text file, but might display images or sound or animations.

Hypermedia is used as a logical extension of the term hypertext in which graphics, audio, video, plain text and hyperlinks intertwine to create a generally non-linear medium of information. This contrasts with the broader term multimedia, which may be used to describe non-interactive linear presentations as well as hypermedia. It is also related to the field of Electronic literature. The World Wide Web is a classic example of hypermedia, whereas a non-interactive cinema presentation is an example of standard multimedia due to the absence of hyperlinks.

Some examples of Hypermedia might be :

- While reading text in the Japanese language we select a Japanese phrase, and then hear the phrase as spoken in the native tongue.
- While viewing a company's floor plan, we select an office by clicking on a room. The employee's name and picture appears with a list of their current projects.

12.4 ANCHOR AND URL

Anchors : The **anchor** tag allows a link to another document or location, or furnishes a way for other documents or locations to link to the tag itself.

An anchor element is called an anchor because we can use it to anchor a URL (Uniform Resource Locator) to some text on a web page. When users view the web page in a browser, they can click the text to activate the link and visit the page whose URL is in the link.

In HTML, an anchor can be either the origin or the target (destination) end of a hyperlink. With the attribute href (hypertext reference), the anchor becomes a hyperlink to either another part of the document or another resource (e.g. a webpage) using an external URL.

An anchor can be used in two ways :

1. To create a link to another document, by using the **href** attribute.
2. To create a bookmark inside a document, by using the **name** attribute.

The '**a'** element is usually referred to as a link or a hyperlink.

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

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

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

The **<a>** tag is supported in all major browsers like Internet Explorer, Mozilla Firefox, Google Chrome, Opera and Safari.

The **<a>** tag supports the following standard attributes :

access key	Specifies a keyboard shortcut to access an element
class	Specifies a classname for an element
dir	Specifies the text direction for the content in an element
id	Specifies a unique id for an element
lang	Specifies a language code for the content in an element
style	Specifies an inline style for an element
tabindex	Specifies the taborder of an element
title	Specifies extra information about an element

Example :

```
<html>
<head>
<title>Anchor</title>
</head>
<body>
<a href="http://www.kkhsou.org/">Go</a>
</body>
</html>
```

The above code gives the following output in Mozilla Firefox :

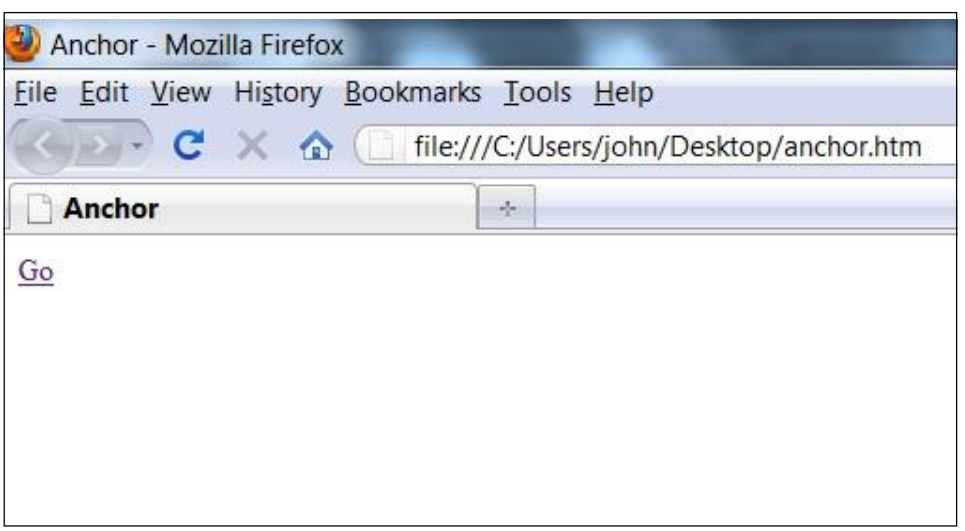


Fig 12.1 : The Anchor Tag

Clicking on the link "**Go**" will direct us to the page which is defined by the **href** attribute of the **<a>** tag. Since the value of the href attribute is set to http://www.kkhsou.org, we will be directed to the home page of KKHSOU.

Uniform Resource Locator (URL) : In computing, a **Uniform Resource Locator** (URL) is a Uniform Resource Identifier (URI) that specifies where an identified resource is available and the mechanism for retrieving it. The best-known example of the use of URLs is for the addresses of web pages on the World Wide Web, such as http://www.example.com/.

Every URL consists of some of the following: the **scheme name** (commonly called protocol), followed by a colon, then, depending on scheme, a **domain name** (alternatively, IP address), a **port number**, the path of the resource to be fetched or the program to be run, then, for programs such as **Common Gateway Interface** (CGI) **scripts**, a **query string**, and an optional **fragment identifier**.

Syntax : scheme://domain:port/path?query_string#fragment_id

- The **scheme** name defines the namespace, purpose, and the syntax of the remaining part of the URL.
- The **domain** name or IP address gives the destination location for the URL. The domain google.com, or its IP address 72.14.207.99, is the address of Google's website.
- The domain name portion of a URL is not case sensitive since DNS ignores case: http://en.example.org/ and HTTP://EN.EXAMPLE.ORG/ both open the same page.
- The **port** number is optional; if omitted, the default for the scheme is used.
- The **path** is used to specify and find the resource requested. It is case-sensitive, though it may be treated as case-insensitive by some servers, especially those based on Microsoft Windows.
- The **query string** contains data to be passed to software running on the server. It may contain name/value pairs separated by ampersands.
- The **fragment identifier**, if present, specifies a part or a position within the overall resource or document. When used with HTTP, it

usually specifies a section or location within the page, and the browser may scroll to display that part of the page.

URLs can be of two types, **Absolute URLs** and **Relative URLs**.

Absolute URLs : Absolute URLs identify a resource *independent of its context.* When we use an absolute URL, we point directly to a file thereby specifying the exact location of a file/directory on the internet. It also follows that each absolute URL is unique, which means that if two URLs are identical, they point to the same file.

For example, http://www.webinfo.com/images/email.gif specifies an image file email.gif located in the images directory, under the domain www.webinfo.com

Relative URLs : A relative URL points to a file/directory in relation to the present file/directory.

In case of absolute URL, we had to specify the entire path to the image email.gif, as

But in case of specifying a Relative URL for the same we simply use–



CHECK YOUR PROGRESS

Q.1. Fill in the blanks :

i) _____ help to create links between various web pages.

ii) The _____ tag allows a link to another document or location.

iii) URLs are of _____ types.

iv) Absolute URLs identify a _____ independent of its context.

v) A _____ URL points to a file/directory in relation to the present file/directory.

Q.2. State whether the following statements are True or False:

i) Hypertext is text which is constrained to be linear.

ii) The href attribute defines the link address.

iii) The WWW is not an example of hypermedia.

iv) An anchor can be used in two ways.

v) A visited link is underlined and purple.

vi) An absolute URL points indirectly to a file.

12.5 CREATING TABLES

HTML tables allow us to organize data in a clear tabular format. Tables are defined with the **<table>** tag. A table is divided into rows (with the **<tr>** tag), and each row is divided into data cells (with the **<td>** tag). The letters td stand for “table data,” which is the content of a data cell. A data cell can contain text, images, lists, paragraphs, forms, horizontal rules, tables, etc.

Example :

```
<table border=”1">
<tr>
<td>row 1, cell 1</td>
<td>row 1, cell 2</td>
</tr>
<tr>
<td>row 2, cell 1</td>
<td>row 2, cell 2</td>
</tr>
</table>
```

- Increasing the value of the **border** attribute thickens the outer border of the table, but not the cells.
- Headings in a table can be defined with the **<th>** tag.
- There can be empty cells in a table also. Table cells with no content are not displayed very well in most browsers.

NOTE: The borders around empty table cells are not displayed. To avoid this, we add a non-breaking space ( ) to empty data cells, to make the borders visible:

Example :

```
<html>
<head>
<title>Table</title>
</head>
<body>
<table border=”5">
<tr>
<th>Heading</th>
<th>Another Heading</th>
</tr>
<tr>
<td>row 1, cell 1</td>
<td>row 1, cell 2</td>
</tr>
<tr>
<td>row 2, cell 1</td>
<td> </td>   <!— An empty cell —>
</tr>
</table>
</body>
</html>
```

The above code gives the following output in Mozilla Firefox:

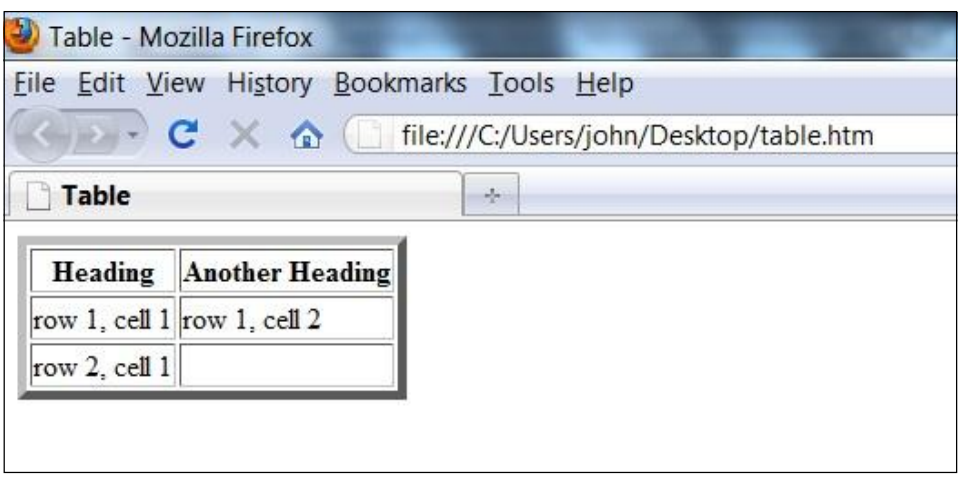


Fig 12.2 : A simple Table with rows and columns

The **<table>** tag supports the following standard attributes:

Attribute	Description
align	Specifies the alignment for the table
bgcolor	Specifies the background color of the table
border	Specifies the thickness of the outermost table border
background	Specifies a background image for the entire table
bordercolor	Specifies the border color of the table
cellspacing	Specifies the amount of space between table cells. (default is 2)
cellpadding	Specifies the amount of space between the contents of the cell and the cell wall. (default is 1)

COLSPAN and ROWSPAN : Once we have a table with many rows and columns, we may want the data in one cell to flow across several rows or columns. This is where the attributes **colspan** and **rowspan** come into play. They are attributes of the <td> and <th> tag.

Colspan : The colspan attribute defines the number of columns a cell should span. It tells the browser to create a cell that spans more than one column. Cells with a multiple colspan will be wider and cross over column boundaries. If the table border is not zero, there will be no separator in the middle of the columns of a cell with multiple colspan.

```
<table border="2" celpadding="4">
<tr><th colspan="2" bgcolor="lime">Production</th></tr>
<tr><td>Raha Mutisya</td><td>1493</td></tr>
<tr><td>Shalom Buraka</td><td>3829</td></tr>
<tr><td>Brandy Davis</td><td>0283</td></tr>
<tr><th colspan="2" bgcolor="lime">Sales</th></tr>
<tr><td>Claire Horne</td><td>4827</td></tr>
<tr><td>Bruce Eckel</td><td>7246</td></tr>
<tr><td>Danny Zeman</td><td>5689</td></tr>
</table>
```

The code above uses the **colspan** attribute to display the following table:

Production	
Raha Mutisya	1493
Shalom Buraka	3829
Brandy Davis	0283
Sales	
Claire Horne	4827
Bruce Eckel	7246
Danny Zeman	5689

Rowspan : Rowspan acts in exactly the same way as colspan, only instead of spanning the columns in a table, the cell will span 2 or more rows in the table. This attribute defines the number of rows a cell should span.

```
<table border="2" celpadding="4">
<tr>
<th rowspan="3" bgcolor="lime">Production</th>
<td>Raha Mutisya</td> <td>1493</td>
</tr>
<tr><td>Shalom Buraka</td> <td>3829</td></tr>
<tr><td>Brandy Davis</td> <td>0283</td></tr>
<tr>
<th rowspan="3" bgcolor="lime">Sales</th>
<td>Claire Horne</td> <td>4827</td>
</tr>
<tr><td>Bruce Eckel</td> <td>7246</td></tr>
<tr><td>Danny Zeman</td> <td>5689</td></tr>
</table>
```

The code above uses the **rowspan** attribute to display the following table:

Production	Raha Mutisya	1493
	Shalom Buraka	3829
	Brandy Davis	0283
Sales	Claire Horne	4827
	Bruce Eckel	7246
	Danny Zeman	5689

Creating a complex table structure : We can combine all the above mention tags and their properties to create table structures that are even more complex. The following example demonstrates such an use of these tags and their attributes.

Example :

```
<html>
<head>
<title>Table</title>
</head>
<body>
<table align="left" bgcolor="yellow" border="5" cellpadding="4" cellspacing="4" bordercolor="blue">
<tr>
```

```
<td colspan="2">Chocolate</td>
<td rowspan="2">Ice cream</td>
<td rowspan="2">Cake</td>
<td rowspan="2">Pastry</td>
</tr>
<tr>
<td>Pie</td>
<td>Wafer</td>
</tr>
<tr>
<td>Cola</td>
<td>Pepsi</td>
<td>Thumbs-up</td>
<td>Sprite</td>
<td>Juice</td>
</tr>
</table>
</body>
</html>
```

The above code gives the following output in Mozilla Firefox:

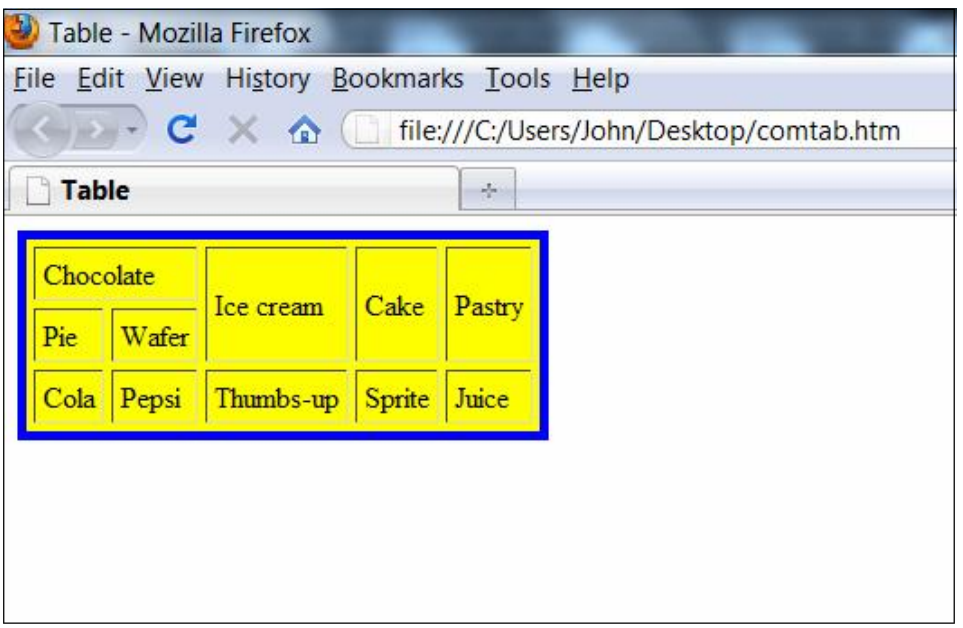


Fig 12.3 : A Table example

3.6 FRAMES

We use frames to display more than one HTML document in the same browser window. Each HTML document is called a **frame**, and each frame is independent of the others. Even though frames may seem to be fairly useful, they have some disadvantages.

The disadvantages of using frames are:

- The web developer must keep track of more HTML documents
- It is difficult to print the entire page

The Frameset Tag :

- The <frameset> tag defines how to divide the window into frames
- Each frameset defines a set of rows or columns
- The values of the rows/columns indicate the amount of screen area each row/column will occupy

The Frame Tag :

- The <frame> tag defines what HTML document to put into each frame

In the example below we have a frameset with two columns. The first column is set to 25% of the width of the browser window. The second column is set to 75% of the width of the browser window. The HTML document “frame_a.htm” is put into the first column, and the HTML document “frame_b.htm” is put into the second column:

Example :

```
<html>
<frameset cols="25%,75%">
<frame src="frame_a.htm">
<frame src="frame_b.htm">
</frameset>
</html>
```

This is frame_a.htm	This is frame_b.htm
<html> *<head>* *<title>Frame_a</title>* *</head>* *<body>* *This is frame_a* *</body>* *</html>*	*<html>* *<head>* *<title>Frame_b</title>* *</head>* *<body>* *This is frame_b* *</body>* *</html>*

The above code gives the following output in Mozilla Firefox:

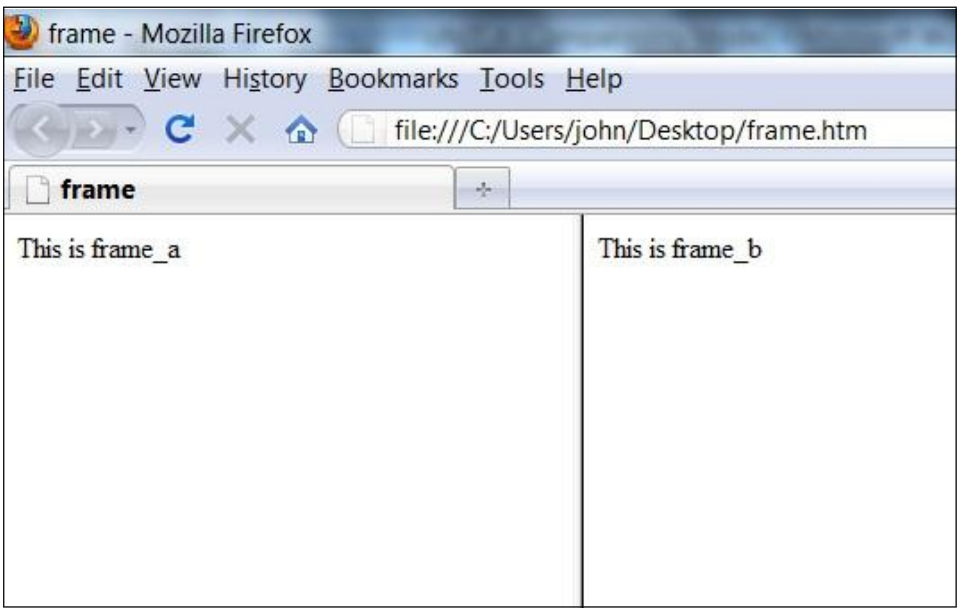


Fig 12.4: A frameset with two columns

The frameset column size value can also be set in pixels (cols=”200,500"), and one of the columns can be set to use the remaining space (cols=”25%,*”).

The IFrame tag : The <iframe> tag defines an inline frame that contains another document. Unlike frames created using the <frameset> and <frame> tags, the <iframe> tag creates a frame that sits in the middle of a regular non-framed web page. <iframe> works like <img>, only instead of putting a picture on the page, it puts another web page. The <iframe> tag is supported by all major browsers

NOTE : To deal with browsers that do not understand iframes, we place the text that we want between the opening <iframe> tag and the closing </iframe> tag.

An **inline** frame is marked up as follows :

```
<iframe src =”html_intro.php” width=”100%” height=”300">
    <p>Your browser does not support iframes.</p>
</iframe>
```

The **<iframe>** tag supports the following standard attributes:

Attribute	Description
class	Specifies a classname for an element
Id	Specifies a unique for an element
style	Specifies an inline style for an element
title	Specifies extra information about an element

Example :

```
<html>
<head>
<title>Iframe</title>
</head>
<body>
<iframe src=”KKHSOU.htm” width=”30%” height=”200"></iframe>
<p>Some older browsers don’t support iframes.</p>
<p>If they don’t, the iframe will not be visible.</p>
</body>
</html>
```

In the above code of the page **iframeEg.htm**, the **src** attribute specifies the source of the document (KKHSOU.htm here) to be displayed inside the iframe. The **width** and **height** attributes specify the dimension of the iframe that will display the document.

The above code gives the following output in Mozilla Firefox:

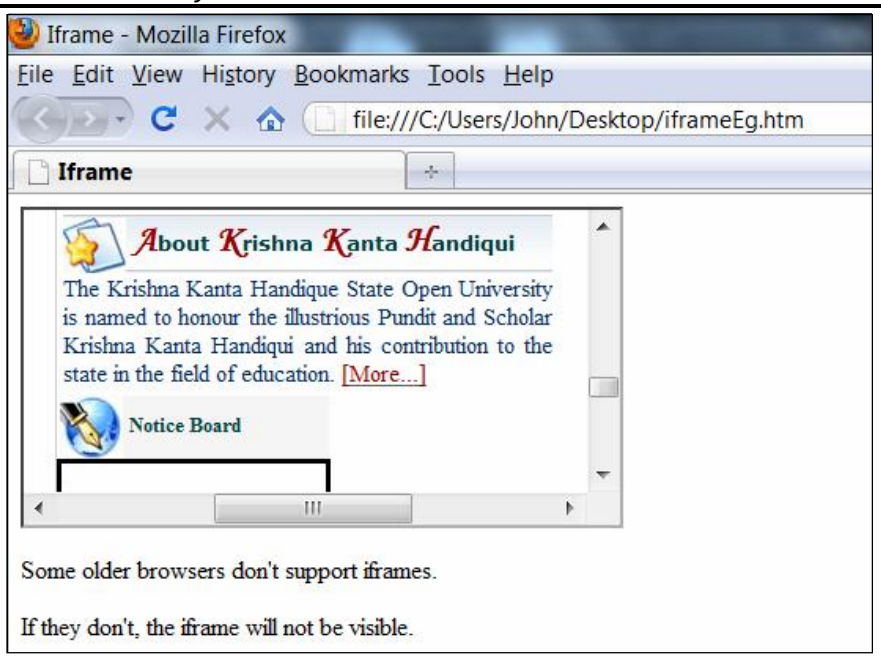


Fig 12.5 : The iframe tag

The Noframe tag : The <noframes> tag is used for browsers that do not handle frames. The noframes element can contain all the elements that are found inside the body element of a normal HTML page. This element is most used to link to a non-frameset version of the web site or to display a message to users that frames are required. The noframes element goes inside the frameset element. This tag is supported by all major browsers

The **<noframes>** tag supports the following standard attributes:

Attribute	Description
class	Specifies a classname for an element
dir	Specifies the text direction for the content in an element
id	Specifies a unique id for an element
lang	Specifies a language code for the content in an element
style	Specifies an inline style for an element
title	Specifies extra information about an element
xml:lang	Specifies a language code for the content in an element, in XHTML documents

Example :

```
<html>
<frameset cols="25%,50%,25%">
<frame src="frame_a.htm">
<frame src="frame_b.htm">
<frame src="frame_c.htm">
<noframes>
Sorry, your browser does not handle frames!
</noframes>
</frameset>
</html>
```



CHECK YOUR PROGRESS

Q.3. Fill in the blanks.

i) HTML _____ allow us to organize data in a tabular format.

ii) A table is divided into _____ and _____.

iii) The _____ attribute defines the number of columns a cell should span.

iv) _____ specifies the amount of space between the cell contents and the cell wall.

v) _____ display more than one HTML document in the same browser window.

vi) The _____ tag is used for browsers that do not handle frames.

vii) The _____ tag defines an inline frame that contains another document.

12.7 FORMS

HTML forms are mainly used to collect information from the user. The HTML used to create forms is similar to any other page creation

techniques that we learned so far. With pages containing forms we add tags that are specific to forms, just like certain tags that are specific to frames, tables etc. These form specific tags allow us to add all types of features to our form page, like buttons, check boxes, drop-down lists, password area, and text boxes to name a few. These input items provide a quick and efficient way for the user to interact with the site.

A form is defined with the <form> tag.

<form>

.

input elements

.

</form>

The Input tag : The most used form tag is the **<input>** tag. The type of input is specified with the **type** attribute. The most commonly used input types are explained below.

Text Fields : Text fields are used when we want the user to type letters, numbers, etc. in a form.

Radio Buttons : Radio Buttons are used when we want the user to select only one of a limited number of choices.

Checkboxes : Checkboxes are used when we want the user to select one or more options of a limited number of choices.

Text Area : When the single line approach of a regular text input field isn't enough for long lines of text, the textarea input field is used which allows ample space to type-in the text. The <textarea> tag must be used with a closing </textarea> tag, and the text placed between them will be the default text inside the text field.

Syntax :

<textarea name="area" cols="10" rows="10">……</textarea>

Where **name** is the element's name; **rows** specifies the number of visible text lines; and **cols** specifies the visible width in average character widths.

Selection list : Selection lists provide us the option of selecting "only one option" or "more than one option" from a selection list. A selection

list starts with an opening **<select>** tag and ends with the closing **</select>** tag. Between them are multiple opening and closing **<option>** tags that include the options to be included in the list.

Example :

```
<select name="selectionlist" size="5" multiple="multiple">
<option>First</option>
<option>Second</option>
<option>Third</option>
<option selected="selected">Fourth</option>
<option>Fifth</option>
<option>Sixth</option>
<option>Seventh</option>
<option>Eighth</option>
<option>Ninth</option>
<option>Tenth</option>
</select>
```

Where **name** is the element's name; **size** specifies the number of list-rows to be displayed at one time, it usually displays the selection as a scrolling list; and **multiple** is a Boolean attribute that allows multiple selections to be made. The **selected** attribute of any option tag allows that option to be selected by default when the page loads for the very first time.

Action Attribute and the Submit Button : When the user clicks on the "**Submit**" button, the content of the form is sent to the server. The form's **action** attribute defines the name of the file to send the content to. The file defined in the **action** attribute usually does something with the received input. A "**Reset**" button at the end allows the user to completely reset/ empty all the input fields, so that the user can re-enter all the values.

Example : The following code demonstrates the use of all the above form elements.

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

```
<form action="process.php" method="post">
Username: <input type="text" name="user"><br><br>
Password: <input type="password" name="password"><br><br>
Name: <input type="text" name="name1"><br><br>
Gender: <input type="radio" name="gender" value="male"> Male
<input type="radio" name="gender" value="female">
Female<br><br>
Age: <select name="age">
<option>18 to 25</option>
<option>25 to 35</option>
<option>35 to 50<option>
</select>
<br><br>I own: A bike <input type="checkbox" name="vehicle"
value="Bike">
A car <input type="checkbox" name="vehicle" value="Car">
An SUV <inputtype="checkbox" name="vehicle"
value="Airplane"><br><br>
Hobbies: <br><br>
<select name="hobby" size="5" multiple="multiple">
<option>Reading</option>
<option>Singing</option>
<option>Writing</option>
<option selected="selected">Cooking</option>
<option>Travelling</option>
<option>None</option>
</select>
<br><br>
<input type="submit" value="Submit">
<input type="reset" value="Reset">
</form>
</body>
</html>
```

The above code gives the following output in Mozilla Firefox:

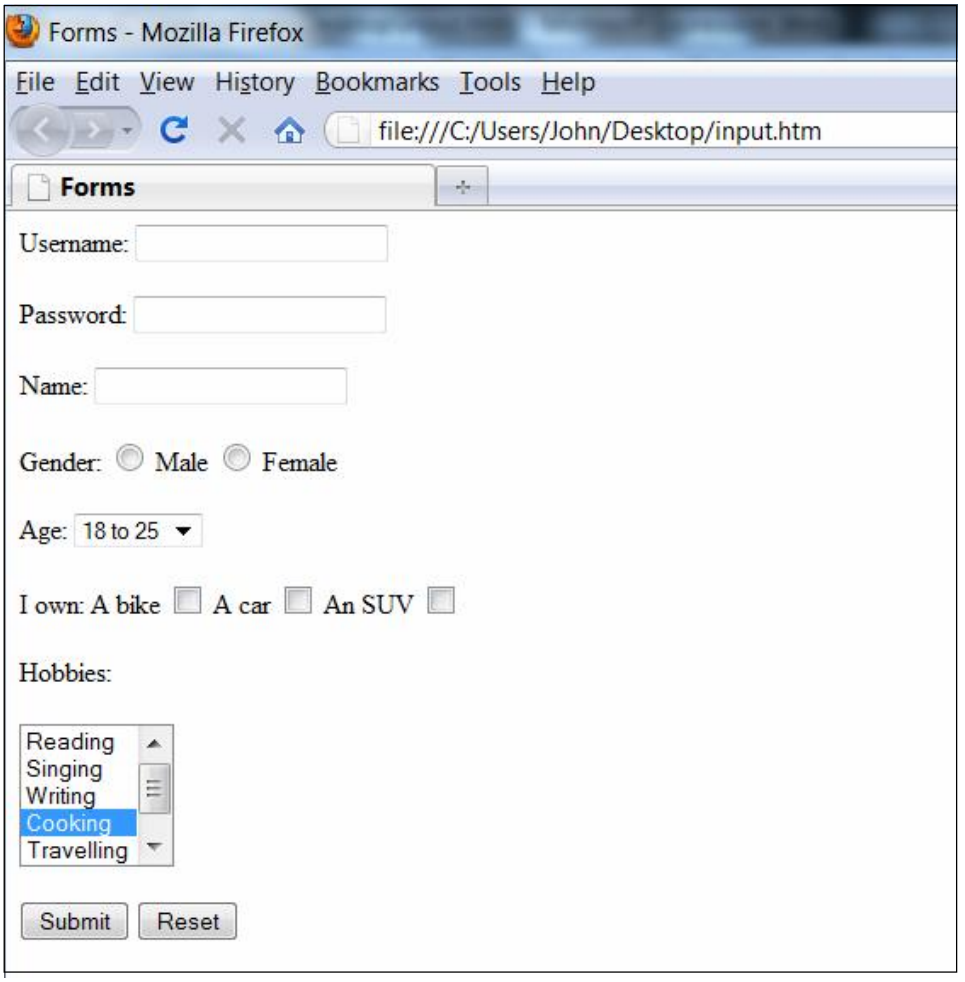


Fig 12.6 : A Form example

If we type some characters in the text fields above, and click the "**Submit**" button, the browser will send the filled-in input to a processing script "**process.php**" which is the value of the **action** attribute. The **method** attribute defines how the input data is to be sent to the server. These two attributes, **action** and **method**, will be explained in the later units.

Form tags :

<form>	Defines a form for user input
<input>	Defines an input field
<textarea>	Defines a text-area (a multi-line text input control)
<label>	Defines a label to a control
<fieldset>	Defines a fieldset
<legend>	Defines a caption for a fieldset
<select>	Defines a selectable item (a drop-down box)
<optgroup>	Defines an option group
<option>	Defines an option in the drop-down box
<button>	Defines a push button

12.8 E-MAILING

We are already familiar with HTML forms that allow users to enter information and allow the page to process the information for carrying out certain web-related tasks. We can also use this feature of HTML forms to send E-mail, which is what will be discussed in this section.

Electronic communication, because of its speed and broadcasting ability, is fundamentally different from paper-based communication. Because the turnaround time can be so fast, email is more conversational than traditional paper-based media. In a paper document, it is absolutely essential to make everything completely clear and unambiguous because our audience may not have a chance to ask for clarification. But with email documents, our recipient can ask questions immediately.

The **MailTo** command is a pretty common one in HTML to perform this task. The basic syntax to create a link on a page that tells the user's mail system to start a message off is:

Mail the Webmaster

We simply put in the email address and the text to read. A new message box will be popped up at the user's machine, letting them use their normal mail package to send a message. The **MailTo** command can

do more than just entering a single e-mail address in the "**Send To**" field while activating the user's e-mail program.

The example below demonstrates how to link to a mail message.

Example:

```
<html>
<body>
<p>
This is an email link:
<a href="mailto:someone@example.com?Subject=Hello%20again">
Send Mail</a>
</p>
</body>
</html>
```

The above code gives the following output in Mozilla Firefox.



Fig 12.7 : A Simple Email link

The next example demonstrates a more complicated **mailto** link.

```
<html>
<body>
<head>
<title>MailTo link</title>
</head>
<p>
```

This is another mailto link :

<a

href="mailto:person1@example.com?cc=person2@example.com&

bcc=person3@example.com&subject=Summer%20Party&body=

You%20are%20invited%20to%20a%20big%20summer%20party!" >Send

mail!

</p>

</body>

</html>

The above code gives the following output in Mozilla Firefox:



Fig 12.8 : MailTo link with cc, bcc, subject and body

The next example demonstrates how to send e-mail from a form.

<html>

<body>

<head>

<title>Form Email</title>

<h3>Send e-mail to john@example.com:</h3>

<form action="mailto:john@example.com" method="post"

enctype="text/plain">

*Name:
*

*<input type="text" name="name" value="your name">
*

```
E-mail:<br>
<input type="text" name="mail" value="your email"><br>
Comment:<br>
<input type="text" name="comment" value="your comment"
size="50">
<br><br>
<input type="submit" value="Send">
<input type="reset" value="Reset">
</form>
</body>
</html>
```

The above code gives the following output in Mozilla Firefox:

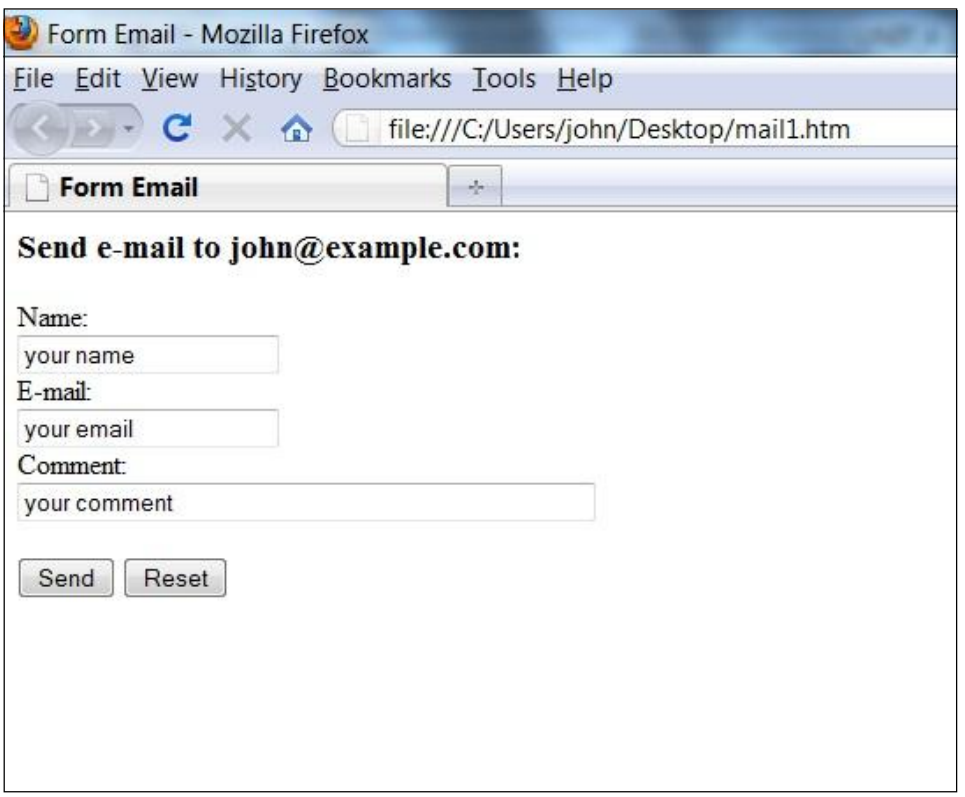


Fig 12.9 : Email sent through a form

Spaces between words should be replaced by %20 to ensure that the browser will display the text properly.



CHECK YOUR PROGRESS

Q.4. Fill in the blanks:

i) A form is defined with the _____ tag.

ii) A selection list starts with an opening _____ tag.

iii) The _____ tag defines an input field.

iv) The _____ button completely resets/empties all the input fields.

v) _____ are used to select one or more options.

vi) The _____ attribute defines how form data is to be sent to the server.



12.9 LET US SUM UP

- ***Hypermedia*** is a superset of hypertext that simply combines hypertext and multimedia.
- In HTML, an ***anchor*** can be either the origin or the target (destination) end of a hyperlink.
- A ***URL*** specifies as to where an identified resource is available and the mechanism for retrieving it.
- The ***<frameset>*** tag defines how to divide the window into frames.
- The ***noframes*** element is most used to link to a non-frameset version of the web site or to display a message to users that frames are required.
- HTML ***forms*** are used to collect information from the user.



12.10 ANSWERS TO CHECK YOUR PROGRESS

Ans. to Q. No. 1 : i) hyperlinks; ii) anchor; iii) two; iv) resource; v) relative

Ans. to Q. No. 2 : i) false; ii) true; iii) false; iv) true; v) true; vi) false

Ans. to Q. No. 3 : i) tables; ii) rows, columns; iii) colspan; iv) cellpadding; v) frames; vi) noframes; vii) iframe

Ans. to Q. No. 4 : i) form; ii) select; iii) input; iv) reset; v) Checkboxes; vi) method



12.11 FURTHER READING

- HTML for beginners – *H. S. Prentice*, Hall India.
- "Web technologies: a computer science perspective*", Jeffrey C. Jackson*, Pearson Prentice Hall.
- *"*Web Technology : A Developer S Perspective*", Gopalan, Gopalan/ akilandeswari*, Pearson Prentice Hall.
- "Learning VBScript", Volume 1, *Paul Lomax,* O'Reilly & Associates.
- WWW.W3schools.com.



12.12 MODEL QUESTIONS

Q.1. Define the terms Hypertext, Hyperlink and Hypermedia withan example each.

Q.2. What are Anchors? Mention some attributes along with anexample?

Q.3. What is an URL? What are its types and working principle?

Q.4. How do we create and use an HTML table?

Q.5. What are frames? Describe all the various tags with an example each.

Q.6. Giving an example describe the working principle of HTML forms.

Q.7. Write a note on HTML's emailing feature.

****

UNIT 13 : INTRODUCTION TO XML

UNIT STRUCTURE

13.1 Learning Objectives
13.2 Introduction
13.3 Comparison of HTML and XML
13.3 XML Basics
13.4 Introduction to DTD
13.5 Let Us Sum Up
13.6 Answers to Check Your Progress
13.7 Further Reading
13.8 Model Questions

13.1 LEARNING OBJECTIVES

After going through this unit, you will able to :

- describe about the XML
- discuss the difference between HTML and XML.

12.2 INTRODUCTION

XML stand for **eXtensible Markup Language**. XML can be used to exchange data between two applications, no matter whether they are compatible or not. The applications may or may not be web applications. As on today XML is the highest demanding data exchange standard in the world. There are many popular DBMS products in the market like Oracle, SQL Server, IMS, IBM, Informix etc. But no one out of them is considered as a data exchange format, because they are not compatible to each other. Prior to the existence of XML also there was the need of exchanging data amongst the applications. At that time they converted their data to textual format, which was a complicated process. But XML makes this communication easier between two applications, regardless of their platform, format or purpose.

13.3 Comparison of HTML and XML

Coding style of XML is almost similar to HTML. i.e. XML also uses tags to structure their documents like HTML. Apart from this there are no similarities between the HTML and XML.

HTML is an information presentation language whereas XML is an information description language. i.e. HTML focu on how to display data to the end user, but XML deals with the representation of data in documents.

HTML is not sufficient for conducting electronic business on the Internet as electronic business requires processing, rearranging, storing, forwarding, exchanging and encrypting of data, which is lacking in HTML. XML solves this problem of HTML.

13.4 XML Basics

A XML document may have the following parts:

- XML tag
- Root element
- Start and end element indicator
- Start and end of element declaration
- Element name and value
- Attribute name and value

An example of a XML document is shown below to discuss the different parts of it.

XML tag : The XML tag has been shown in the first line of the example. Every XML document must be started with this line. As shown in the example the text in this line is enclosed with the tag pair <? and ?>.

Root element : It indicates the starting of actual content of the XML document. It is shown by the element <Books> in the line number 2.

Start and end element indicator : The start of an element is indicated by < and end is indicated by >. Every element is following this rule.

```
1.  <?xml version=”1.0" encoding=”iso-8859-1"?>
2.  <Books>
3.  <Book pyear=”2001">
4.  <Title>Learning ANSI C</Title>
5.  <Author>E.Balaguruswami</Author>
6.  </Book>
7.  <Book  pyear=”2002">
8.  <Title>Learning C++</Title>
9.  <Author>E.Balaguruswami</Author>
10. </Book>
11. <Book  pyear=”2005">
12. <Title>Database Systems</Title>
13. <Author>Korth,Sudarsana</Author>
14. </Book>
15. <Book  pyear=”2003">
16. <Title>Database Systems</Title>
17. <Author>Elmasri, Navathe</Author>
18. </Book>
19. </Books>
```

Start and end element declaration : It is similar to a HTML container tag. i.e. the element declaration is starting with a starting tag enclosed with <> and ends with same tag but with a slash “/” included in front of it. e.g., One example may be: the starting of an element declaration is displayed in line no. 3 and the end of declaration is in line no. 6.

Element name and value : All the items within a tag inside a root element are specified as element. e.g., **Title** or **Author** are the elements names. The element value is available between the starting and ending tags. If we consider line no 4, then “Title” is the element

name and “Learning ANSI C” is the element value.

Attribute name and value : Attributes are available along with the element name within the same starting element declaration. The attribute value is written along with the attribute using “=” operator. For example, in line no. 3, “pyear” is an attribute and “2001” is the attribute value.

13.5 Introduction to DTD

We can include a DOCTYPE declaration in a XML document which will refer a DTD file. DTD stands for ***document type declaration***. The DOCTYPE declaration is written in a <!DOCTYPE … > tag. The DTDs are of two types:

- Internal DTD
- External DTD

Internal DTD : In case of Internal DTD, the DTD is declared within the XML document itself in a separate block. It is shown in the following example.

```
<?xml version=”1.0" encoding=”iso-8859-1" standalone=”yes”?>
<?xml-stylesheet type=”TEXT/xsl” href=”abc.xsl”?>
<!Doctype Books[
    <!ELEMENT Books (Book+)>
    <!ELEMENT Book (Title?, Author?)>
    <!ATTLIST Book pub  CDATA #REQUIRED>
    <!ELEMENT Title (#PCDATA)>
    <!ELEMENT Author (#PCDATA)>
]>
<Books>
    <Book pub=”2001">
            <Title>Learning ANSI C</Title>
            <Author>E.Balaguruswami</Author>
            </Book>
            …
```

```
<Book pub="2003">
<Title>Database Systems</Title>
<Author>Elmasri, Navathe</Author>
</Book>
</Books>
```

***Points to be noted* :**

- We should include a clause standalone="yes" in the <?xml?> tag to indicate internal DTD.
- Elements are declared by using <!ELEMENT...> tag. Here we should use a data type #PCDATA within parenthesis for the DTD element. PCDATA stands for parsed character data.
- The characters **?, +** and ***** following an element name has some special meaning such as:
  - **element ?** indicates element can be repeated 0 or 1 times and it is also called single-valued.
  - **element +** indicates element can be repeated 1 or more times and it is also called multi-valued.
  - **element** * indicates element can be repeated 0 or more times and it is also called multi-valued.
- Attributes of an element are declared by using <!ATTLIST...> tag. It needs a data type CDATA stands for character data and a #REQUIRED keyword.

External DTD : Here the DTD declaration is made in a separate file with extension .dtd, which contains only the <!DOCTYPE name[… …]> parts of the above XML example. The reference to this .dtd file is to be included in the XML file is shown below:

```
<!DOCTYPE BOOKS SYSTEM "book1.dtd">
```

Provided that the DOCTYPE declaration for the same is available in the file "book1.dtd". The "book1.dtd" file must contain only the element declaration. It can be as follows for our example:

```
<!ELEMENT BOOKS (BOOK+)>
<!ELEMENT BOOK (TITLE, AUTHOR?)>
```

<!ELEMENT TITLE (#PCDATA)>

<!ELEMENT AUTHOR (#PCDATA)>



CHECK YOUR PROGRESS

Q.1. Select the correct option from the brackets:

a) XML (represents/ displays) data in a better way.

b) In order to include DTD in the same XML file, standalone clause should be (“Yes”/”No”) in the <?xml?> tag.

c) When “?” followed an element name, it indicates the special meaning (single-valued/ multi-valued).

d) External DTD file has extension (.dtd/ .doc)



13.6 LET US SUM UP

- The executable content tags can add more dynamism to a web page. Dynamic content can be developed by using Java applet and ActivX control. The executable content tags are: <APPLET> tag, <OBJECT> tag and <PARAM> tag.
- CSS is a powerful tool for adding style to a web page and ensure uniformity thorough out the page.
- XML stand for extensible markup language. XML can be used to exchange data between two applications
- HTML is an information presentation language whereas XML is a information description language.
- DTD stands for document type declaration.



13.7 ANSWER TO CHECK YOUR PROGRESS

Ans. to Q. No. 1 : a) Represents, b) Yes, c) single-valued, d) .dtd



12.8 FURTHER READING

- **Web enabled commercial Application Development using HTML, Javascript, DHTML and PHP** *–Ivan Bayross, BPB*
- **Web Technologies** *–A.S. Godbole and Atul Kahate, Tata McGraw-Hill.*
- Using HTML, XML and Java *–Eric Ladd and Jim O'Donnell, PHI*
- **Fundamental of Database Systems** *–Ramez Elmasri and Shamkant B. Navathe, Pearson.*



13.9 MODEL QUESTIONS

1. What is XML? What are the components of XML?
2. Write an XML document to store the details of Employees. Also design a web page to get the input of employees and to retrieve the full details from the XML document.
3. Explain the concept of DTD.

****

UNIT 14 : JAVA SCRIPT

UNIT STRUCTURE

14.1 Learning Objectives

14.2 Introduction

14.3 Client Side Programming

- 14.3.1 Introduction to JavaScript
- 14.3.2 Advantages of JavaScript

14.4 Basic Programming Concepts

- 14.4.1 Variables
- 14.4.2 Data-Types
- 14.4.3 Operators

14.5 Control Structures in JavaScript

- 14.5.1 Branching
- 14.5.2 Looping

14.6 Array

- 14.6.1 Dense Array
- 14.6.2 Array Within Array
- 14.6.3 Methods of Array

14.7 Functions

- 14.7.1 User-Defined Function
- 14.7.2 Built-in Function
- 14.7.3 Dialog Box

14.8 Let Us Sum Up

14.9 Answers to Check Your Progress

14.10 Further Reading

14.11 Model Questions

14.1 LEARNING OBJECTIVES

After going through this unit, you will be able to :

- describe client side programming
- discuss the basic programming concepts of JavaScript

- identify the different control structures used in JavaScript for branching and looping
- explain the Array
- describe function and its types

14.2 INTRODUCTION

In the previous unit, you have learnt the advanced HTML tags and basic concept of XML. This unit deals with the concept of JavaScript. In this unit you will learn about the client side programming. This unit also describes the basic programming idea of JavaScript. In addition, you will come across the different control structures used in JavaScript for branching and looping. This unit seeks to provide you a with the concept of array and functions used in JavaScript ansn you will also be able to learn about the working with form object. Besides you will be acquianted with the basic idea of built-in Objects in JavaScript. Finally, the unit will discuss the use of user defined objects.

14.3 CLIENT SIDE PROGRAMMING

A HTML document is purely static, i.e., it can only present the documents. Interaction is needed to makes it dynamic. To make it interactive, communication is to take place between the client's browser and the web server, where the actual content is stored. In order to fulfil this requirement client side scripting languages are developed. So a client side scripting language is capable of making interaction with the client; it can capture data inserted by the user , and validate data as per the server specification and dispatch corrected data to the web server. The client side programming languages are JavaScript, vbScript etc.

14.3.1 Introduction to JavaScript

JavaScript is a scripting language of the Web. It is the most popular scripting language on the internet and it works in all major browsers, such as Internet Explorer, Firefox, Chrome, Opera, and

Safari. JavaScript is used in millions of Web pages to add functionality to validate forms, detect browsers and much more.

Java and JavaScript are two completely different languages in both concept and design. Java (developed by Sun Microsystems) is a powerful and much more complex programming language - in the same category as C and C++.

A scripting language is a lightweight programming language.

- JavaScript was designed to add interactivity to HTML pages
- JavaScript is usually embedded directly into HTML pages
- JavaScript as an interpreted language means that scripts execute without preliminary compilation
- Everyone can use JavaScript without purchasing a license

The HTML <SCRIPT> tag is used to insert a JavaScript into an HTML page. A JavaScript program code is included between the tag pair <SCRIPT> and </SCRIPT>, which is contained either in the header section or in the body section of the HTML page.

The **<SCRIPT> Tag :** It is a container tag and it is used to include JavaScript code into a HTML document.

Syntax :

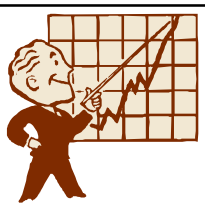
```
<SCRIPT Language="JavaScript">
        // JavaScript code
</SCRIPT>
```

14.3.2 Advantages of JavaScript

- JavaScript is an interpreted language, which is interpreted by a JavaScript enabled browser just like a HTML code.
- JavaScript does not require any other special editor for programming. Just simple editor is enough, where any HTML code can be created.
- It has comparatively fewer commands;so the programmer needs to know limited syntax.
- It requires minimum storage spaces which lead to better

performance.

- JavaScript can properly execute the event handling like an event driven programming language. Example of events may be: press a button, click mouse button, move mouse over etc.
- Error detection-correction process as well as debugging is easy.



CHECK YOUR PROGRESS

Q.1. Fill in the blanks :

i. A HTML document is a purely _____.

ii. JavaScript is an _____.

iii. _____ Tag is a container tag and it is used to include JavaScript code into a HTML document.

14.4 BASIC JAVASCRIPT CONCEPTS

A few points on JavaScript :

- JavaScript is case sensitive, i.e. it can understand the difference between the capital letter and small letters.
- Every JavaScript statement is ended with a semicolon.
- JavaScript uses a **prompt ()** method to receive data from user in the client browser. Here contents are delimited by using comma.
- JavaScript uses **document.write()** methods to display message or content of variable into client browser. Here also contents are delimited by using comma. Here document is a JavaScript object and write is a method of it.
- In document.write() we can even put any HTML tag, delimited by comma.
- JavaScript uses an **alert()** method, which is used to display message or content of variable. Here contents are delimited by

'+' operator.

- JavaScript variables are loosely typed. i.e. type of a variable can be decided by looking at its value. A variable can change its type according to the value stored in it. i.e. the declaration of a variable in JavaScript is not mandatory prior to its use.

14.4.1 Variables

Definition : A variable is a place in memory which can hold a data element. When a variable is defined a memory location is assigned internally to it. Value of a variable can vary during the execution of a program. A variable can be declared by using **var** keyword.

Syntax :

var <variable name>=value;

e.g.

var number=100;

var name="Gagan";

Rules for giving variable name :

- A variable may start with a uppercase, lowercase, a underscore character (_) or a $ sign.
- Rest of the characters may be letters, underscore (_), doller ($) or digits (0…9).
- If multiple words are used as a variable name then the first character should be lowercase and first character of next words should be capital. e.g. interestRate, discountAmount etc.

14.4.2 Data-Types

Data type means types of data that can hold by a variable. In JavaScript there are four different primitive data types such as: **Number, Boolean, String** and **Null.** JavaScript also supports complex types like: **array** and **object**.

Number : It can take integer, floating point number or NaN

(stands for Not a Number) value. Integer can be decimal, hexadecimal or octal. But at the time of display others are converted into decimal.

Boolean : It is represented by **true** and **false**. These are converted to 1 and 0 automatically when used in numerical expression.

String : Strings are represented by putting alphabets or digits within single or double quotation mark.

Null : This type contains a single value i.e. null, which means empty or nonexistent reference.

14.4.3 Operators

Arithmetic Operators : The normal arithmetic operators can be used here also. Such as:

Operator	Name	Meaning
+	plus	for addition
–	minus	for subtraction
/	slash	for division
*	asterisk	for multiplication
%	modulus	for getting remainder of two integer number division
++	increment	increment by 1
—	decrement	decrement by 1

Relational Operators : these are used to compare two quantities. They are used to construct a condition. They produce Boolean values.

Operator	**Meaning**
==	Equal
>	Greater than
<	Less than
>=	Greater than and Equal
<=	Less than and Equal
!=	Not Equal
===	Strictly Equal

!==	Not Strictly Equal

Logical Operator : These are used to combine two or more condition. They produce Boolean values.

Operator	**Meaning**
&&	AND
||	OR
!	NOT

String Operator : It is a "+" symbol used to perform concatenation.

Assignment Operator : It is used to assign a value or expression of right side of an = sign to a single variable to the left hand side. It has several variations as shown below:

Operator	Example	Result
=	a=5;	a is 5
+=	b=7;b+=a;	b is 12
-=	c=4; b-=c;	b is 8
*=	b*=c;	b is 32
/=	b/=8;	b is 4
%=	a%=3;	a is 2

Delete operator : It is used to delete property of an object or an array element.

e.g. delete array1[2];

New operator : It is used to create an instance of an object.

e.g. array1= new array1();

Example Program 1.

```
<HTML>
    <HEAD>
    <SCRIPT Language="JavaScript">
        var num=10;
        var name="TARAJAN";
    </SCRIPT>
    </HEAD>
<BODY >
```

```
<SCRIPT Language="JavaScript">
    // use of document.write().
    document.write("Number is = ",num,"<br/>");
    document.write("Name is = ",name,"<br/>");
    // use of prompt().
    num=prompt("Enter a distance ",num);
    name=prompt("Enter a Destination ",name);
    document.write("Destination is = ",name,"<br/>");
    // use of alert().
    alert("Distance is = "+num+" k.m. ");
    document.write("It is",num, " k.m. apart<br/>");
    // use of conditional expression.
    str1=(num>100)?"More than 100":"Below 100";
    document.write("Num is  ",str1,"<br/>");
</SCRIPT>
</BODY>
</HTML>
```



CHECK YOUR PROGRESS

Q.2. State True or False :

i. JavaScript is case sensitive, i.e. it can understand the difference between the capital letter and small letters.

ii. In document. write () we can put only limited HTML tag.

iii. Data type means types of data that can be hold by a variable

iv. JavaScript does not support complex types like array and object.

v. '–' symbol is used to perform concatenation.

14.5 CONTROL STRUCTURES IN JAVASCRIPT

Control structures like conditional statements are used in JavaScript

to perform different actions based on different conditions.

14.5.1 Branching

If statement : it is similar to C language. Here a condition is tested if it is true then it will execute the true part i.e. statement set 1, if it is false then it will execute the false part i.e. statement set 2,

Syntax :

```
if (<condition>)
        {
            Statement set 1;
        }
[else
        {
            Statement set 2;
        } ]
```

Switch statement : It is more suitable for the situation where we need to select one option from multiple options.

Syntax :

```
switch (choice)
        {
            Case 1 :
                Statement set 1;
                break;
            Case 2 :
                Statement set 1;
                break;
            Case 3 :
                Statement set 1;
                break;
            default :
                Statement set 1;
        }
```

Immediate if (conditional expression) : JavaScript has a conditional expression operator ? and : , using which immediate if is constructed as stated below.

Syntax :

condition ? value if true : value if false

e.g. b=100;

str1=a>b ? “More than 100” : “Below 100”;

14.5.2 Looping

Looping means to execute the same set of statement repeatedly for several times (zero or more times).

for loop : It is same as C language. It is more suitable when the number of iteration is known earlier.

Syntax :

```
for(initialization; termination; incrementation/decrementation)
        {
            Body;
        }
```

e.g.

```
for(i=0;i<100;i++)
        {
            document.write(I,”<br/>”);
        }
```

while loop : It is also same as C language. It is executed until a condition is satisfied. Once the condition becomes false, the loop is terminated.

Syntax :

```
while(condition)
        {
            Body;
        }
```

e.g.

```
i=0;
while(i<100)
    {
        document.write(I,"<br/>");
        i++;
    }
```

do … while loop : It is also same as while loop. The only difference is that here the condition is tested at the end of the loop, whereas in case of while the condition is tested at the beginning of the loop.

Syntax :

```
do
    {
        Body;
    }
        while(condition);
```

e.g.

```
i=0;
do
    {
        document.write(I,"<br/>");
        i++;
    }
while(i<100);
```

break and continue statement : These two statements are used in loops.

- **break** statement is used to come out of a loop canceling the rest of the iterations.
- **continue** statement is used to jump to the end of the loop ignoring the statements from its occurrence up to the end of the loop for that iteration only. The next iterations will be executed.

Example program 2 :

```
<html>
```

```
<Head>
<script language="JavaScript">
    var num1=0,num2=0,i=0;
</script>
</head>
<body >
<script language="JavaScript">
    num1=prompt("Enter a small number ",num1);
    num2=prompt("Enter a large number ",num2);
    alert("yoU have inserted ! "+num1+" and "+num2);
    for(i=num1; i<=num2;i++)
        document.write(i,"<br/>");
    alert("Over! ");
    document.write("Over! ");
</script>
</body>
</html>
```

14.6 ARRAY

An **array** is a common name given to set of variables. It has an index number to be written within square brackets and started with the value 0. Elements of an array are not necessarily of same type, they can be of any type. The array must be declared before using it. Array is a JavaScript object. Array has several methods for the manipulation of its elements. Array is declared as follows:

Syntax :

```
arrayname = new Array(size);
```

Or `arrayname = new Array();`

Example of array declaration :

```
name = new Array(12);
```

Use of array :

```
name[0]="Kamal";
name[1]="Amal";
```

…

name[11]=”Ratul”;

14.6.1 Dense Array

It is a JavaScript array where initialization is done at the time of declaration. i.e. Both declaration and initialization are taking place at the same place.

Syntax :

arrayname = new Array(value0, value1, value2, … valuen-1);

e.g.

address= new Array(“Sankar”,”Ulubari”,”Guwahati”, 781007);

14.6.2 Array Within Array

An array can be an element of another array, thereby forming array within array. It can also be considered as multidimensional array. The inner array can be accessed by using double dimension.

e.g.

address= new Array(“Sankar”,”Guwahati”, new array(9861288771,9435165455),”Mridul”,”Jorhat”, new array(9976512123, 9707013422,9854329975));

i.e. the array elements for the above array are as follows:

address[0]=”Snkar”

address[1]=”Guwahati”

address[2][0]=”9861288771"

address[2][1]=” 9435165455"

address[3]=”Mridul”

address[4]=” Jorhat”

address[5][0]=” 9976512123"

address[5][1]=” 9707013422"

address[5][2]=” 9854329975"

14.6.3 Methods of Array

There are a few methods which are working on array. Such as: join(), reverse().

Join() : This method is used to join all the elements of the array like a single string and return the joint string. We can use the delimiter as argument of join(), that we want between the elements in the joint string. If we do not put any delimiter the default is comma and followed by a space.

e.g.

```
jname = address.join();
document.write(jname,"<br/>");
```

Reverse() : Using this method the elements of an array can be reversed.

e.g.

```
1. address.reverse();                    // it will reverse the array
2. rname1=address.reverse();             // it will reverse the array
                                         //address and stored them
                                         //into another array rname1
   document.write(rname,"<br/>");        // display array
3. rname2=address.reverse().join();      // it will reverse  as well as
                                         // join the array and store
                                         //them to a string rname2
   document.write(rname,"<br/>");
```

In the above example 1 the original array 'address' is reversed. In example 2, array is again reversed and store the reverse array into a new array rarray1. In example 3, only a reverse string is produced i.e. rname2.

Length property : In an array we can use the length property to find the size of the array.

e.g. n=address.length;



CHECK YOUR PROGRESS

Q3. i. What is an array?

.....

ii. Write down the syntax for dense array.

.....

14.7 FUNCTIONS

Functions are considered to be a subprogram or blocks of code, which performs a subtask. A function can return a value. Function is used to eliminate repetition of coding. It turns the program into a structured one which increases readability.

Functions are of two types :

- User-defined function.
- Built-in function

14.7.1 User-Defined Function

User defined functions are those function which are developed by the programmer itself in the coding. The user defined functions are to be declared first before using it. Once we declare the function we can call it by giving its name. A function is declared by using the keyword function.

Syntax :

```
function fun_name(parameter list delimited by comma)
        {
            Body
        }
```

Though a function can be declared anywhere within the HTML document yet it is better to place it in the header section. It is because, the header section is executed before the body section.

e.g.

```
<html><head><title>Example of Function:: Display Addition chart
                    </title>
<script language="JavaScript">
var a=0;
function addchart(num)
        {
        var n=0,i=0;
        for(i=1;i<=10;i++)
            document.write(num," + ",i," = ",num+i," <br/>");
        }
        </script>
        </head>
        <body>
            <script language="JavaScript">
            a=prompt("Enter a number :",a);
            a=parseIns(a);
            addchart(a);
        </script>
        </body>
</html>
```

14.7.2 Built-in Function

Built-in functions are those functions which are already developed by the language developer. They are the standard functions, which can be accessed at any time within the <script> tag. In JavaScript there are several such built-in functions. Such as:

eval() : It is used to convert a string value to a numeric value.

parseInt() : It is also used to convert a string value to a numeric value.

parseFloat() : It is also used to convert a string value to a float value.

14.7.3 Dialog Box

In JavaScript there are a few dialog boxes for displaying messages and content of variables. These dialog boxes are very effective as well as attractive for interacting with the user at run time. They appear as a separate window. e.g. the dialog boxes are: prompt, alert and confirmed.

prompt() : It is used to receive data from user at run time. It also displays a message.

Syntax :

prompt("message", "Default value");

alert() : It is used to display a small message in a separate window.

Syntax :

alart("message");

Confirm Dialog Box : It is used for the confirmation of the user. A confirmed dialog box may contain **ok** and **cancel** button.

Syntax :

confirm("message");



CHECK YOUR PROGRESS

Q.4. Fill in the blanks :

i. A function can return a _____.

ii. The user defined functions are to be _____ before using it.

iii. A function is declared by using the _____ function.



14.8 LET US SUM UP

- A HTML document is purely static. i.e. It can only present the documents.

- JavaScript is a high level object oriented language.
- JavaScript is case sensitive, i.e. it can understand the difference between the capital letter and small letters.
- When a variable is defined a memory location is assigned internally to it.
- Data type means types of data that can be held by a variable. In JavaScript there are four different primitive data types such as: Number, Boolean, String and Null.
- Looping means to execute the same set of statement repeatedly for several times (zero or more times).
- An array is a common name given to set of variables. It has an index number to be written within square brackets and started with the value 0.
- Functions are considered to be a subprogram or blocks of code, which performs a subtask. A function can return a value.
- Built-in functions are those which are already developed by the language developer. They are the standard functions, which can be accessed at any time within the <script> tag.



14.9 ANSWERS TO CHECK YOUR PROGRESS

Ans. to Q. No. 1 : i. static, ii. interpreted language**,** iii. <SCRIPT>

Ans. to Q. No. 2 : i. True, ii. False, iii. True, iv. False, v. False

Ans. to Q. No. 3 : i. An array is a common name given to set of variables

ii. arrayname = new Array(value0, value1, value2, … valuen-1)

Ans. to Q. No. 4 : i. Value**,** ii. declared first**,** iii. keyword



14.10 FURTHER READING

- **Web enabled commercial Application Development using HTML, Javascript, DHTML and PHP** - *Ivan Bayross*



14.11 MODEL QUESTIONS

Q.1. What is JavaScript? Describe in brief.

Q.2. Write short note on :

a. Variable

b. Data type

c. Operators

Q.3. Write in short about the control structure in Java Script.

Q.4. What is an array? What is its use?

Q.5. Write in short about Function.

Q 6. Write a function to find sum of two numbers using javascript

Q7. Write a function to reverse the digits of a number using javascript (example 357 in reverese 753)

****

UNIT 15 : FORMS AND OBJECT

UNIT STRUCTURE

15.1 Learning Objectives

15.2 Introduction

15.3 JavaScript Form Object

15.4 Elements of Form Object

15.5 Properties of Form Object

15.6 Events of Form Elements

15.7 Methods of Form and Form Elements

15.8 Built-in Objects in JavaScript

- 15.8.1 String Objects
- 15.8.2 Math Object
- 15.8.3 Date Objects

15.9 User Defined Objects in JavaScript

- 15.9.1 Creating User Defined Objects
- 15.9.2 Creating Objects Within Objects

15.10 Let Us Sum Up

15.11 Answers to Check Your Progress

15.12 Further Reading

15.13 Model Questions

15.1 LEARNING OBJECTIVES

After going through this unit, you will be able to :

- illustrate the form object and its elements, properties, methods, events and event-handlers
- identify built-in Objects in JavaScript like string object, math object and date objects
- discuss the use of user defined objects.

15.2 INTRODUCTION

We have already learnt about HTML *Form* and all its elements. HTML form provides the facility for data gathering in a web page. It has several GUI controls for that purpose. Also HTML is capable of sending gathered data to the web server and the server side scripting languages in the web server processes those data. Here, **role of JavaScript is to validate the gathered data** in the client's machine itself. This will reduce some unnecessary traffic. Otherwise, lots of traffic is to be involved between the server and the client machine in order to send correct data, as the server side program will not allow invalid data.

15.3 JavaScript Form Object

When the <FORM> tag is encountered the JavaScript enabled browsers creates the form object and stores them as a **form array** in memory. Each form object is described between <FORM> and </FORM> tags. i.e. there can be multiple occurrence of <FORM>.. </FORM> tag pair.

15.4 Elements of Form Object

The elements of a form object are: Text, Password, TextArea, Button, Radio, CheckBox, Select, Submit, Reset, Hidden and Fileupload. This property is used to refer to the respective form object element. The basic concepts of these elements are already given in the earlier unit.

15.5 Properties of Form Object

The form object has several properties such as: Name, Value, DefaultValue, Checked, DefaultChecked, Length, Index, Text, SelectIndex, DefaultSelect and Selected.

Name and **value** properties are used by all the form elements.

DefaultValue property is used by Text, TextArea and Password elements of a form.

Checked property is used by Radio button and CheckBox elements of a form.

DefaultChecked property is used by Radio Button and CheckBox elements of a form.

Length property is used by Radio Button element of a form.

Index property is used by Radio Button and Select elements of a form.

Text property is used by Radio Button and Select elements of a form.

SelectIndex property is used by Radio Button and Select elements of a form.

DefaultSelect property is used by Radio Button and Select elements of a form.

Selected property is used by Radio Button and Select elements of a form.

15.6 Events of Form Elements

JavaScript uses several events for many of its form elements. Events are the activities of an user in the web page as a result of which a piece of program or a function will be executed. The events are such as: Focus(), Blur(), Select(), Change(), Click(), Clicked() etc.

Focus() event is used by the form elements Text, Password, TextArea and Select. The event handler for Focus() is **onFocus**().

Blur() event is used by the form element Text, Password, TextArea and Select. The event handler for Blur() is **onBlur**().

Select() event is used by the form element Text, Password, TextArea and Select. The event handler for Select() is **onSelect**().

Change() event is used by the form element Text, Password and Select. The event handler for Change() is **onChange**()

Click() event is used by the form element Button, Submit, Reset, CheckBox, and Select. Here the event handler is used is **onClick**().

Clicked() event is used by the form element Radio Button. The event handler used for Clicked() is **onClicked().**

15.7 Methods of Form and Form Elements

Methods of Form : The methods of form are used to send data captured by the form elements. These are either GET or POST.

The **GET** method appends the captured data with the URL and send to the web server. It is suitable for sending small amount of data. The maximum limit of data here is 1024 bytes. It is the default method.

The **POST** method is used to send data as a separate bit-stream to the web server. It is used when a large amount data is to be sent.

Methods of Form Elements : These methods are used to handle the events that occurred in different form elements. Such as: onFocus(), onBlur(), onSelect(), onChange(), onClick(), onClicked() etc.

onFocus() method is used by the form elements Text, Password, TextArea and Select.

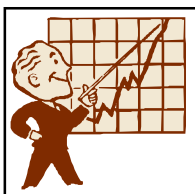
onBlur() method is used by the form element Text, Password, TextArea and Select.

onSelect() method is used by the form element Text, Password, TextArea and Select.

onChange() method is used by the form element Text, Password and Select.

onClick() method is used by the form element Button, Submit, Reset, CheckBox, and Select.

onClicked() method is used by the form element Radio Button.



CHECK YOUR PROGRESS

Q.1. State true or false of the following

i. 'Password' is an element of a form object.

ii. Checked properly is used by submit button.

iii. Click () event is used by the form element Text.

iv. The GET method is used to append the captured data with the URL and send to the Web server.

v. The POST mothod is used to send data together in bit-stream.

15.8 BUILT-IN OBJECTS IN JAVASCRIPT

In JavaScript some of the built-in Objects are used for data processing, related to string, math and date objects.

15.8.1 String Objects

A string in JavaScript is an object and it has properties and methods for the manipulations.

String **property** is the **length,** using which we can find out the number of characters in a string.

String **methods** are: big(), blink(), bold(), charat(), italics(), tolowercase(), touppercase(), substring(). The functions performed by the methods are easily understood by its name. Only a few methods are discussed below.

charat() method: It takes an index as argument and it return the respective character at that index in the string. e.g. str1="WELCOME", then str1.chatat(0) will return "W" and str1.chatat(5) will return "M".

substring() method: Using it we can extract a substring from a string. It takes two arguments: first argument indicates starting index of the substring to be extracted. The second argument indicates the last index of the substring to be extracted +1. i.e. next to the last index of the substring to be extracted. e.g. if str1="WELCOME" then str1.substring(0,3) will return "WEL" and str1.substring(3,6) will return "COM".

15.8.2 Math Object

The math object provides several useful properties and method which helps JavaScript a lot to carry out any mathematical operation smoothly.

Math object has the **properties** like **E, LN10, LN2** and **PI** for Eular's constant, natural logarithm of 10, natural logarithm of 2 and Pi value respectively.

Math objects also has **methods** like: **abs(), ceil(), cos(), floor(), pow(), random(), sin(), sqrt()** and **ten().**

15.8.3 Date Objects

The date object of JavaScript has the methods like: getDate(), setDate(), getHours(), setHours(), getTime() and setTime() for the manipulation of date and time.

15.9 USER DEFINED OBJECTS IN JAVASCRIPT

JavaScript provides facilities for creating user defined object also which can satisfy a particular need.

15.9.1 Creating User Defined Objects

The user defined object can be created with the help of a method. e.g. **address** is a user defined object. Then its **properties** may be described by the name, po, city and contact and they can be referred as:

address.name

address.po

address.city

address.contact

Say one methods may be display(), then it can be referred as:

address.display()

e.g. function address(name, po, city, contact)

```
{
    this.name=name;
    this.po=po;
    this.city=city;
    this.contact=contact;
}
```

15.9.2 Creating Objects within Objects

An object can be used as property of another object, such as: in the previous example, suppose we want to create contact as an object with the properties like: landPhone, mobilePhone, emailAddress. In this case the properties will be specified as:

```
address.name
address.po
address.city
address.contact.landPhone
address.contact.mobilePhone
address.contact.emailAddress
e.g.
function contact(landPhone, mobilePhone, emailAddress)
    {
        this.contact.landPhone= landPhone;
        this.contact.mobilePhone= mobilePhone;
        this.contact.emailAddress= emailAddress;
    }
function address(name, po, city, contact)
    {
        this.name=name;
        this.po=po;
        this.city=city;
        this.contact=contact;
    }
```

Use of user defined onjects :

Kamal = new contact(“2556778”, “9856645453”, “kamalnath@yahoo.co.in”);

Champak = new contact(“2556557”, “9707045683”, “champak@yahoo.com”);

add1=new address(“Kamal”, “Chandmari”, “Guwahati”, kamal);

add2=new address(“Champak”, “Noonmati”, “Guwahati”, champak);



15.10 LET US SUM UP

- <FORM> tag is used to create form object and stores them as a **form array** in memory.
- Every form object has some elements such as Text, Password, TextArea, Button, Radio, CheckBox, Select, Submit, Reset, Hidden and Fileupload.
- Every form object has several properties such as: Name, Value, DefaultValue, Checked, DefaultChecked, Length, Index, Text, SelectIndex, DefaultSelect and Selected.



15.11 ANSWERS TO CHECK YOUR PROGRESS

Ans. to Q. No. 1 : i. True, ii. False, iii. False, iv. True, v. False



15.12 FURTHER READING

- **Web enabled commercial Application Development using HTML, Javascript, DHTML and PHP** - *Ivan Bayross*



15.13 MODEL QUESTIONS

Q.1. What is form object? How to create it?

Q.2. What are the elements of form objects? Where they are used?

Q.3. Describe the properties of form object..

Q.4. What is the basic difference between Built in Object and User Defined Object in Java Script? Write in brief about them.

****