GCA (S5) 02
Exam. Code: WT

Web Technology

SEMESTER V

BACHELOR OF COMPUTER APPLICATION

Block 1



KRISHNA KANTA HANDIQUI STATE OPEN UNIVERSITY

Subject Experts

Prof. Anjana Kakati Mahanta, Gauhati University

Prof. (Retd.) Pranhari Talukdar, Gauhati University

Dr. Jyotiprokash Goswami, Associate Professor, Assam Engineering College

Course Co-ordinators

Dr. Tapashi Kashyap Das, Assistant Professor, KKHSOU

Ms. Sruti Sruba Bharali, Assistant Professor, KKHSOU

SLM Preparation Team

UNITS	CONTRIBUTORS
1, 2 & 3	Mr. Mridul Suklabaidya, Cotton University
4, 5 & 6	Mr. Sankhapani Bharali, Girls' Polytechnique, Guwahati
7	Mr. Anupam Gohain, Faculty, GATE Coach
8	Mr. Pritam Medhi, Research Scholar, Gauhati University

Editorial Team

Content : Dr. Tapashi Kashyap Das, KKHSOU

Language : Prof. (Retd.) Robin Goswami, Cotton College

Structure, Format & Graphics: Dr. Tapashi Kashyap Das, KKHSOU

June, 2019 ISBN: 978-93-89123-29-6

This Self Learning Material (SLM) of the Krishna Kanta Handiqui State University is made available under a Creative Commons Attribution-Non Commercial-ShareAlike4.0 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-781 017

City Office : Housefed Complex, Dispur, Guwahati-781 006; 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.

BACHELOR OF COMPUTER APPLICATION

Web Technology

Block-1

CONTENTS

		Pages
UNIT 1:	**Basics of Internet and Web** The Basics of Internet, World Wide Web, Web Page, Home Page, Web Site, Static, Dynamic and Active Web Page, Overview of Protocols– Simple Mail Transfer Protocol, Gopher, Telnet, Emails, FTP, Hyper Text Transfer Protocol, Client Server Computing Concepts, MIME, Web Surfing	**7-25**
UNIT 2:	**Web Client** Web Browser, Browsers e.g., Netscape navigator, Internet Explorer, Mozilla Firefox, Client-Side Scripting Languages- VB Script and JavaScript, Active X control and Plug-ins	**26-39**
UNIT 3:	**Web Sever** Web Server Architecture, Image Maps, CGI, API Web Database Connectivity-DBC, ODBC, Proxy Server	**40-56**
UNIT 4:	**Introduction to HTML** Introduction to HTML, Essential Tags, Tags and Attributes, Text Styles and Text Arrangements, Text, Effects, Exposure to Various Tags (DIV, MARQUEE, NOBR, DFN, HR, LISTING, Comment, IMG), Color and Background of Web Pages, Attributes of Image Tag, Hypertext, Hyperlink and Hypermedia, Links, Anchors and URLs, Links to External Documents	**57-73**
UNIT 5:	**HTML Lists and Tables** Lists, Unordered Lists, Ordered Lists, Definition Lists, Tables, TABLE, TR and TD Tags, Cell Spacing and Cell Padding, Colspan and Rowspan	**74-87**
UNIT 6:	**HTML Frames and Form** Frame, Frameset, FRAME Tag, NOFRAMES Tag, Forms, FORM and INPUT Tag, Text Box, Radio Button, Checkbox, SELECT Tag and Pull Down Lists, Hidden, Submit and Reset	**88-100**

UNIT 7: Cascading Style Sheet **101-112**

Cascading Style Sheets (CSS), Inline Style, Embedded Style, External Style Sheet, Imported Style Sheet, Creating a CSS file, Using CSS in a Web Page

UNIT 8: Dynamic HTML **113-141**

Dynamic HTML, Document Object Model, Features of DHTML, CSSP (Cascading Style Sheet Positioning) and JSSS (JavaScript assisted Style Sheet), Layers of Netscape, The ID Attribute, DHTML Events

COURSE INTRODUCTION

This course is on ***Web Technology***. Web Technology includes Web development and designing. Web design is about how to make the website unique and user friendly. The methods by which computers communicate with each other through the use of markup languages and multimedia packages is known as Web technology. The course emphasises the fundamentals of Web page development and prepares the learners to build real-world, Web-based applications and to use a wide variety of Web development tools effectively and efficiently.

The course is divided into two blocks:

Block 1 describes Internet and Web. Learners are introduced to the concepts of Internet Protocols commonly used in Web applications, Hypertext Markup Language and various features of CSS, Dynamic HTML. Markup languages are the languages in which the web is written. Learners will be able to design web pages including pictures, links, table, lists, frames and form after going through this block.

Block 2 starts with JavaScript. It describe the various concepts of web technologies, namely, ML, AJAX, XML, server-side scripting PHP.

BLOCK INTRODUCTION

This is the first block of the course ***Web Technology***. Various concepts associated with World Wide Web and Internet are discussed in this block. It has also described the concept of web server and web client. Different tags of Hypertext Markup Language, features of CSS and Dynamic HTML are other topics being discussed in this block. Learners will be able to design attactive web pages learning the features of HTML, CSS and Dynamics HTML .

This block comprises the following eight units:

Unit 1 is an introductory unit on Internet and Web. Concept of Web page, Home page, Website and various important protocols like SMTP, Telnet, FTP, HTTP etc.are introduced in this unit.

Unit 2 is on Web Client. This unit discusses some important web browsers, client-side scripting language and also Active X control and Plug-ins.

Unit 3 is on Web server. It describes the web server architecture in the beginning. It also introduces Image maps, CGI and ODBC at the end.

Unit 4 introduces HTML and the various tags of HTML with their attributes. This unit is intended to give the learner the ability to create HTML documents.

Unit 5 describes HTML lists and Tables. With this unit the learners will be able to create tables in a web page and add list of items with the use of different HTML lists and attributes.

Unit 6 is on HTML frames and forms. This unit describes how to devide the web page into frames so that one part can be kept fixed and the other part is kept movable as per user requirement. Learners will also be able to design HTML forms and take users input in such forms.

Unit 7 introduces Casecading Style Sheet. Inline Style Sheet, Embedded Style Sheet, External Style Sheet etc. are described in this unit.

Unit 8 is on Dynamic HTML. Different features of DHTML , DOM, CSS positioning etc. are discussed in this unit.

Each unit of this block includes some along-side boxes to help you know some of the difficult, unseen terms. Some “EXERCISES” have been included to help you apply your own thoughts. You may find some boxes marked with: “LET US KNOW”. These boxes will provide you with some additional interesting and relevant information. Again, you will get “CHECK YOUR PROGRESS” questions. These have been designed for you to self-check your progress of study. It will be helpful for you if you solve the problems put in these boxes immediately after you go through the sections of the units and then match your answers with “ANSWERS TO CHECK YOUR PROGRESS” given at the end of each unit.

UNIT 1: BASICS OF INTERNET AND WEB

UNIT STRUCTURE

1.1 Learning Objectives

1.2 Introduction

1.3 Basics of Internet

- 1.3.1 Usage of Internet
- 1.3.2 Advantages of Internet
- 1.3.3 Disadvantages and Limitations of Internet

1.4 World Wide Web

- 1.4.1 Web Page
- 1.4.2 Home Page
- 1.4.3 Website
- 1.4.4 Different Types of Web Documents
- 1.4.5 Advantages and Disadvantages

1.5 Protocols

1.6 Client Server Computing Techniques

1.7 MIME

1.8 Web Surfing

1.9 Let Us Sum Up

1.10 Further Reading

1.11 Answers to Check Your Progress

1.12 Model Questions

1.1 LEARNING OBJECTIVES

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

- learn about Internet and its working methods
- learn the advantage and disadvantage of using Internet
- describe World Wide Web
- define Web Page, Home page and different types of Web Documents
- gather some idea about Internet protocols
- describe Client-Server technique
- learn about Web Surfing.

1.2 INTRODUCTION

This is the first unit of the course. Here, we will discuss the basics of Internet and Web. Internet is a network of computers linking many different types of computers all over the world. It primarily consists of billions of interconnected Web Pages and thus creating World Wide Web. In 21st century, Internet facilities dominate all kinds of works such as household works, personal works and mainly official works. For internet only, people are getting connected across the world and contributing their innovative works to this world.

Different terminologies associated with Internet and Web as well as various protocols will be described in this unit.

1.3 BASICS OF INTERNET

Internet is a numerous number of connected computer networks all over the world to serve billions of people. It is a network of networks that consists of millions of private, public, academic, business, and government networks which span the globe and are linked by many networking technologies. With the Internet, it is possible to access almost any information, communicate with anyone, can do & get many more services. We can avail Internet connectivity from Internet Service Providers (ISP). Those service providers offer internet with monthly or daily rates.

Internet is also called a distributed system, because internet is decentralized in design that means no center point is there. Internet is made up of thousands of smaller networks. The Internet carries a vast range of information resources and services, such as the interlinked hypertext documents of the World Wide Web (WWW) and the infrastructure to support electronic mail. The Internet thrives and develops as its many users find new ways to create, display and retrieve the information that constitutes the Internet.

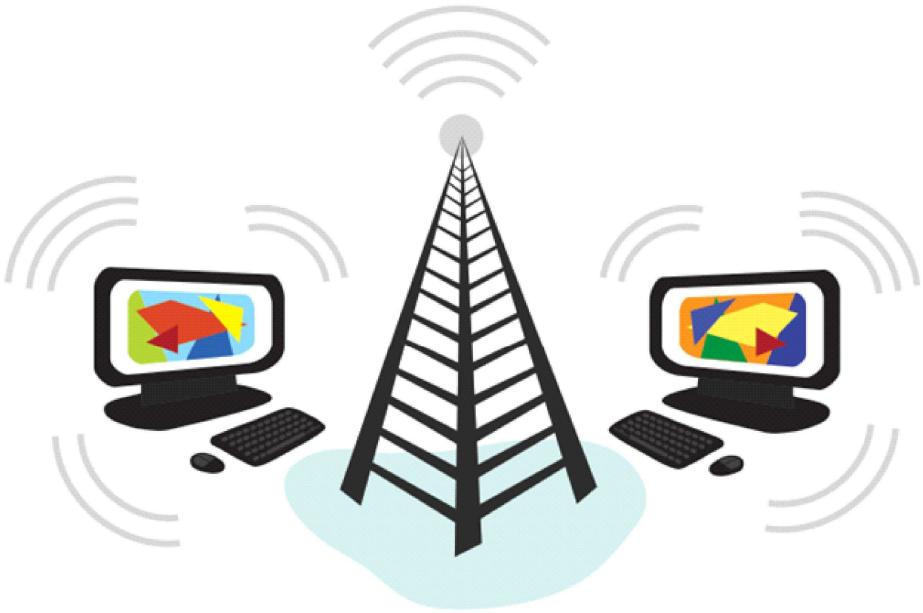


Figure 1.1: A simple picture of two connected computers enabling Internet features (Pic Courtesy: GCFGlobal)

1.3.1 Usage of Internet

Internet has been the most useful technology of the modern times which helps us not only in our daily lives, but also in our personal and professional lives. The internet helps us achieve this in several different ways. For the students and educational purposes the internet is widely used to gather information so as to do the research or to add to the knowledge of various subjects. Even the business professionals and the professionals like doctors, access the internet to filter the necessary information for their use. The internet is therefore the largest encyclopedia for everyone, in all age categories. The internet has served to be more useful in maintaining contacts with friends and relatives who live abroad permanently.

1.3.2 Advantages of Internet

In this section we will discuss some important Internet applications to show the advantages of Internet.

- **E-mail:** Electronic mail or e-mail is now an essential communication tool in business. With e-mail you can send and receive instant electronic messages, which works like writing letters. Your messages are delivered instantly to people anywhere in the world, unlike the traditional mail that takes a lot of time. Email is free, fast and very cheap when compared to telephone, fax and postal services.
- **24 hours a day – 7 days a week:** Internet is available, 24x7 days for usage.
- **Information:** Information is probably the biggest advantage internet is offering. There is a huge amount of information available on the internet for just about every subject, ranging from government law and services, trade fairs and conferences, market information, new ideas and technical support. You can almost find any type of data on almost any kind of subject that you are looking for by using search engines like Google, Yahoo, MSN, etc.
- **Online Chat:** You can access many 'chat rooms' on the web that can be used to meet new people, make new friends, as well as to stay in touch with old friends. You can chat in MSN and yahoo websites.
- **Services:** Many services are provided on the internet like net banking, job searching, purchasing tickets, hotel reservations, guidance services on array of topics engulfing every aspect of life.
- **Communities:** Communities of all types have sprung up on the internet. It's a great way to meet up with people of similar interest and discuss common issues.
- **E-commerce:** Along with getting information on the Internet, you can also shop online. There are many online stores and sites that can be used to look for products as well as buy them using your credit card. You do not need to leave your house and can do all your shopping from the convenience of your home. It has got a real amazing and wide range of products from household needs, electronics to entertainment.

- **Entertainment:** Internet provides facility to access wide range of Audio/Video songs, plays films. Many of which can be downloaded. One such popular website is YouTube.
- **Software Downloads:** You can freely download innumerable, software like utilities, games, music, videos, movies, etc from the Internet.

1.3.3 Disadvantages and Limitations of Internet

Though Internet has become an indispensible part of this digital age, it has some disadvantages. Some of them are:

- Theft of Personal information: Electronic messages sent over the Internet can be easily snooped and tracked, revealing who is talking to whom and what they are talking about. If you use the Internet, your personal information such as your name, address, credit card, bank details and other information can be accessed by unauthorized persons. If you use a credit card or internet banking for online shopping, then your details can also be 'stolen'.
- Negative effects on family communication: It is generally observed that due to more time spent on Internet, there is a decrease in communication and feeling of togetherness among the family members.
- Internet addiction: There is some controversy over whether it is possible to actually be addicted to the Internet or not. Some researchers claim that some people simply try to escape from their problems through the online world.
- Childrens use of the Internet has become a big concern. Most parents do not realize the dangers involved when their children log onto the Internet. When children talk to others online, they do not realize they could actually be talking to a harmful person. Moreover, pornography is also a very serious issue concerning the Internet, especially when it comes to young children. There are thousands of pornographic sites on the Internet that can be

easily found and can be absolutely harmful for children using the Internet.

- Virus threat: Today, not only are humans getting viruses, but computers are also. Computers are mainly getting these viruses from the Internet. Virus is a program which disrupts the normal functioning of your computer system. Computers attached to Internet are more prone to virus attacks and they can end up into crashing your whole hard disk.
- Spamming: It is often viewed as the act of sending unsolicited email. This multiple or vast emailing is often compared to mass junk mailings. It needlessly obstruct, the entire system. Most spam is commercial advertising, often for dubious products, get-rich-quick schemes, or quasi-legal services.

1.4 WORLD WIDE WEB

The *World Wide Web* (WWW) is the universe of network-accessible information in Internet sites that can be accessed by using a hypertext interface. Hypertext interface means online pages which are formatted in markup languages like HTML, XML etc. and accessed via HTTP. These online pages or web pages might take us to our related materials on another computer located anywhere in this world. The World Wide Web is a way of exchanging information between computers on the Internet, tying them together into a vast collection of interactive multimedia resources. The World Wide Web was originally designed by Tim Berners-Lee in 1991.

1.4.1 Web Page

A *web page* is a document commonly written in Markup Languages like HTML that is accessible using an Internet Browser with proper Internet connectivity. A webpage is accessed by entering page's address in the address bar of our browser and this page may contain text, graphics, and hyperlinks to other web pages and files. That Web Page's address we generally termed as URL (Uniform Resource Locator).



Figure 1.2: A cartoon image of a person viewing a webpage.

(Pic Courtesy: GCFGlobal)

1.4.2 Home Page

The **homepage** or **home page** is the name of the **main page** of any website where visitors can find hyperlinks to other pages on the site. By default, the homepage on all *web servers* is *index.html*, which, can also be index.htm, index.php, or whatever the developer decides.

1.4.3 Website

A Website is a related collection of World Wide Web (WWW) files that includes a beginning file called a homepage. A company or an individual tells you how to get to their Website by giving you the address of their home page. From the home page, you can get to all the other pages on their site. For example, the Website for Flipkart has the home page address of http://www.flipkart.com. Then this home page address leads to numerous number of pages of the Flipkart Website. Websites can have many functions and can be used in various fashions; a website can be a personal website, a corporate website for a company, a government website, an organization website, etc. Websites are typically dedicated to a

particular topic or purpose, ranging from entertainment and social networking to providing news and education. All publicly accessible websites collectively constitute the World Wide Web.

1.4.4 Different Types of Web Documents

Web documents can be of different types. Those are:

- **Static**: A static web document resides in a file that is associated with a web server. The author of a static document determines the contents at the time the document is written. Because the contents do not change, each request for a static document results in exactly the same response.
- **Dynamic**: A dynamic web document does not exist in a predefined form. When a request arrives the web server runs an application program that creates the document. The server returns the output of the program as a response to the browser that requested the document. Because a fresh document is created for each request, the contents of a dynamic document can vary from one request to another.
- **Active**: An active web document consists of a computer program that the server sends to the browser and that the browser must run locally. When it runs, the active document program can interact with the user and change the display continuously.

1.4.5 Advantages and Disadvantages

Advantages and disadvantages of different web documents are different.

- In case of **Static** web pages:

  Advantages: simplicity, reliability and performance. The browser can place a copy in a cache on a local disk.

  Disavantages: inflexibility, changes are time consuming because they require a human to edit the file.
- In case of **Dynamic** web pages:

  Advantages: ability to report current information (current stocks prices, current weather conditions, current availability of tickets

for a concert). Because both static and dynamic documents use HTML, a browser does not know whether the server extracted the page from a disk file or obtained the page dynamically from a computer program.

Disadvantages: increased cost and, like a static document, a dynamic document does not change after a browser retrieves a copy. Thus, information in a dynamic document begins to age as soon as it as been sent to the browser (stock prices). Server push. The server runs the programs periodically and sends the new document to the browser.

- In case of **Active** web pages:

  Advantages: Ability to update information continuously. For example, only an active document can change the display quickly enough to show an animated image. More important, an active document can access sources of information directly and update the display continuously. For example, an active document that dispays stock prices can continue to retrieve stock information and change the display without requiring any action from the user.

  Disadvantages:

  - An active document can run on an arbitrary computer instead on a server, the program must be written to avoid depending by particular features of a computer.
  - An active document is a potential risk because the document can export or import information.
  - Active documents require more sophisticated browser software (interpreters, virtual machines, etc.) and a powerful computer system to run the browser.

1.5 PROTOCOLS

For communicating between two or more computers some rules and guidelines need to be followed. A protocol is simply a set of rules and guidelines for communicating data. Each device in the network has to follow these rules for successfully transmiting information.

Some of the important protocols which are used in transmission are:

- **Simple Mail Transfer Protocol:** As the name suggests, it is a protocol used in sending and receiving e-mails. SMTP is one of the most common and popular protocols for email communication over the Internet. SMTP, a TCP/IP protocol, usually is implemented to operate over Internet port no. 25. E-mail clients such as Outlook, Eudora etc. uses SMTP to send a message to the mail server, and the mail server uses SMTP to relay that message to the correct receiving mail server.
- **Gopher:** It's an application layer protocol used for searching, retrieving and viewing documents stored in remote servers over the internet. Gopher was conceived in 1991 as one of the Internet's first data/file access protocols to run on top of a TCP/IP network. A gopher server presents its content in hierarchical manners. It was developed at the University of Minnesota and named after the school's mascot. But when protocol like Hypertext Transfer Protocol (HTTP) came with modern features it dominate gopher type of protocols.
- **Telnet:** Telnet is a protocol for accessing remote computers and to provide a bidirectional interactive text-oriented communication facility using a virtual terminal connection over the internet or a TCP/IP Computer Network. Telnet is like command line system, where user can enter commands through the Telnet program and they will be executed as we were entering them directly on the server console. This enables you to control the server and communicate with other servers on the network. To start a Telnet session, you must log in to a server by entering a valid username and password. Telnet is a common way to remotely control Web servers.
- **E-mail:** Electronic mail or Email is the most frequently used service on the internet. Electronic mail or e-mail makes it possible to exchange messages through a local or worldwide communication networks. Most e-mail software and services allow for the attachment of files containing text, documents, sound, pictures, and even computer programs. Two types of E-mail type are there, first one enables subscribers of a network provider to send and receive

messages with e-mail software. Messages are processed on the sender's terminal. The second type, Web mail, is accessible through navigation software that enables a computer connected to Internet to send and receive mail, without the services of a network provider.

- **File Transfer Protocol (FTP):** FTP is a still-used and somewhat popular protocol for uploading and downloading files, a standard Internet protocol for transmitting files between computers on the Internet over TCP/IP connections. Using FTP, a client can upload, download, delete, and rename, move and copy files on a server. A user typically needs logon to the FTP server, although some servers make some or all of their content available without login, also known as anonymous FTP.
- **Hyper Text Transfer Protocol (HTTP):** Hypertext Transfer Protocol (HTTP) is an application-layer protocol used primarily on the World Wide Web. HTTP uses a client-server model where the web browser is the client and communicates with the webserver that hosts the website. The browser uses HTTP, which is carried over TCP/IP to communicate to the server and retrieve Web content for the user. HTTP is a widely used protocol and has been rapidly adopted over the Internet because of its simplicity. It is a stateless and connectionless protocol.



CHECK YOUR PROGRESS

Q.1: What is Internet? Write some uses of it.

.....

.....

.....

.....

.....

.....

.....

.....

.....

Q.2: Explain WWW. Write about Hypertext interface.

.....

.....

.....

.....

.....

.....

1.6 CLIENT/SERVER COMPUTING TECHNIQUES

Client/Server computing is a computing model in which client and server computers communicate with each other over a network. In client/ server computing, a server takes requests from client computers and shares its resources, applications and/or data with one or more client computers on the network, and a client is a computing device that initiates contact with a server in order to make use of a shareable resource. A server generally share their resources with clients. A client does not share any of its resources, but requests a server's content or service function. Clients therefore initiate communication sessions with servers which await incoming requests.

Characteristics of Client Server Computing: The salient points for client server computing are as follows:

- The client server computing works with a system of request and response. The client sends a request to the server and the server responds with the desired information.
- The client and server should follow a common communication protocol so that they can easily interact with each other. All the communication protocols are available at the application layer.
- A server can only accommodate a limited number of client requests at a time. So it uses a system based on priority to respond to the requests.
- The Denial of Service attacks the server's ability to respond to authentic client requests by inundating it with false requests.
- An example of a client server computing system is a web server. It returns the web pages to the clients that requested them.

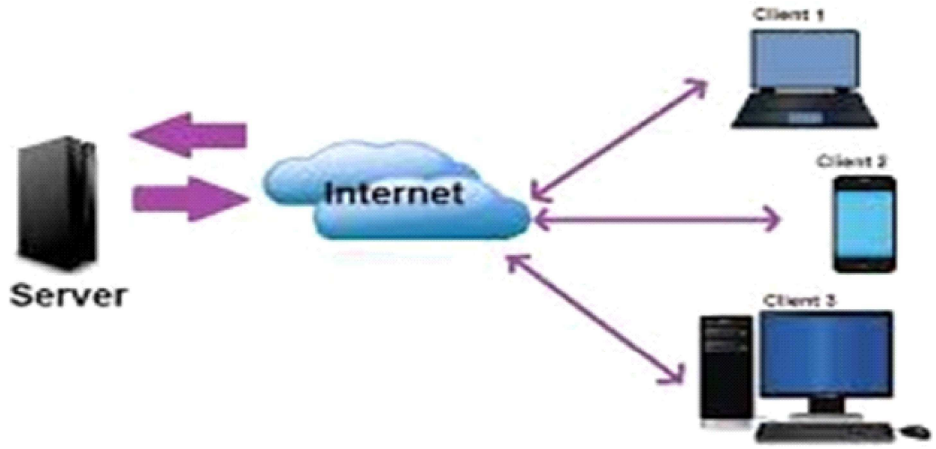


Figure 1.3: A typical Client Server model

1.7 MIME

MIME (Multi-Purpose Internet Mail Extensions) was proposed by Bell Communications in 1991, and the specification was originally defined in June 1992 for RFCs 1341 and 1342. MIME is an extension of the original Internet email protocol that helps extend the limited capabilities of email by allowing insertion of images, sounds and text in a message. Multimedia contents such as audio, video, images, application programs, and other kinds, as well as the ASCII text handled in the original protocol, the Simple Mail Transfer Protocol. MIME describes the message content type and the type of encoding used with the help of headers.

1.8 WEB SURFING

The term “surfing” was first coined by Mark McCahill. In the World Wide Web, surfing means to move from one Webpage to another, usually in an undirected manner. When surfing, the user typically visits pages based on what interests him/her at the moment.

Surfing is an activity that started with the advent of the World Wide Web. With hypertext links, users can go not only from one part of a document to another, but also from one document to another, including those located in remote sites. Surfing is a favorite pastime for millions of people around the world who have access to the Internet. Many users are hooked on it, spending countless hours doing casual searches or other online activities. Some of the advantages and disadvantages of surfing are as follows:

Advantages:

- Mostly information on various subjects are available.
- Search engines are powerful.
- Ability to do research from your home versus research libraries.
- Message boards where people can discuss ideas on any topic. Ability to get wide range of opinions. People can find others that have a similar interest in whatever they are interested in.
- The internet provides the ability of emails. Free mail service to anyone in the country.
- Platform for products like SKYPE, which allow for holding a video conference with anyone in the world who also has access.
- Many Connections between people from worldwide can be made via social networking.
- Things such as Yahoo Answers and other sites where kids can have readily available help for homework.
- News, of all kinds is available almost instantaneously. Commentary, on that news, from every conceivable viewpoint is also available.

Disadvantages:

- There is a lot of wrong information on the internet. Anyone can post anything, all sources are not reliable.
- There are predators that hang out on the internet waiting to get unsuspecting people in dangerous situations.
- Some people are getting addicted to the internet and thus causing problems with their family and career.
- Many Multimedia contents may be harmful for students.
- Easy to waste a lot of time on the internet. You can start surfing, and then realize far more time has passed than we realized.
- There are a lot of unscrupulous businesses that have sprung up on the internet to take advantage of people.
- Hackers can create viruses that can get into your personal computer and ruin valuable data, They can use the internet for identity theft.



CHECK YOUR PROGRESS

Q.3: What do you mean by Client Server Architecture?

.....

.....

.....

.....

.....

.....

.....

Q.4: What is Protocol? Give Some Examples.

.....

.....

.....

.....

.....

Q.5: Fill in the Blanks:

a) The internet is a numerous number of connected ..... all over the world to serve billions of people.

b) HTML is a ..... language.

Q.6: What is a Virus?

.....

.....

.....



1.9 LET US SUM UP

- The internet is a numerous number of connected computer networks all over the world to serve billions of people. It is a network of networks that consists of millions of private, public, academic, business, and government networks which spans the globe which are linked by many networking technologies.

- The internet helps us achieve this in several different ways. For the students and educational purposes the internet is widely used to gather information so as to do the research or add to the knowledge of various subjects. Even the business professionals and the professionals like doctors, access the internet to filter the necessary information for their use.
- The World Wide Web (WWW) is the universe of network-accessible information in Internet sites that can be accessed by using a hypertext interface. Hypertext interface means online pages which are formatted in markup languages like HTML, XML etc. and accessed via HTTP.
- A **web page** is a document commonly written in Markup Languages like HTML that is accessible using an Internet Browser with proper internet connectivity.
- The **homepage** or **home page** is the name of the **main page** of any website where visitors can find hyperlinks to other pages on the site.
- A Web site is a related collection of World Wide Web (WWW) files that includes a beginning file called a homepage.
- Protocols are some of rules and guidelines needed to be followed while communicating between systems.
- The client server computing works with a system of request and response. The client sends a request to the server and the server responds with the desired information.
- MIME is an extension of the original Internet email protocol that that helps extend the limited capabilities of email by allowing insertion of images, sounds and text in a message.
- Web surfing means to move from one Web page to another.



1.10 FURTHER READING

1) Godbole, Achyut S., Kahate, Atul. *Wed Technologies– TCP/IP to Internet Architectures*. Tata McGraw Hill.

2) Deitel & Deitel. *Web Technology*. Pearson Education.
3) Kamal, Raj. *Internet and Web Technologies*. Tata McGraw Hill.
4) https://www.webopedia.com,
5) https://searchenterprisedesktop.techtarget.com/,
6) https://www.tutorialspoint.com/,
7) https://www.computerhope.com, https://www.techopedia.com



1.11 ANSWERS TO CHECK YOUR PROGRESS

Ans. to Q. No. 1: The internet is a numerous number of connected computer networks all over the world to serve billions of people. It is a network of networks that consists of millions of private, public, academic, business, and government networks which spans the globe which are linked by many networking technologies. With the Internet, it's possible to access almost any information, communicate with anyone, can do & get many more services.

The internet helps us achieve this in several different ways. For the students and educational purposes the internet is widely used to gather information so as to do the research or add to the knowledge of various subjects. Even the business professionals and the professionals like doctors, access the internet to filter the necessary information for their use. The internet is therefore the largest encyclopedia for everyone, in all age categories. The internet has served to be more useful in maintaining contacts with friends and relatives who live abroad permanently.

Ans. to Q. No. 2: The World Wide Web (WWW) is the universe of network-accessible information in Internet sites that can be accessed by using a hypertext interface. Hypertext interface means online pages which are formatted in markup languages like HTML, XML etc. and accessed via HTTP. These online pages or web pages might take us to our related materials on another computer located anywhere in this world. The World Wide Web is a way of exchanging information

between computers on the Internet, tying them together into a vast collection of interactive multimedia resources.

Ans. to Q. No. 3: Client/Server computing is a computing model in which client and server computers communicate with each other over a network. In client/server computing, a server takes requests from client computers and shares its resources, applications and/or data with one or more client computers on the network, and a client is a computing device that initiates contact with a server in order to make use of a shareable resource. A server generally share their resources with clients. A client does not share any of its resources, but requests a server's content or service function. Clients therefore initiate communication sessions with servers which await incoming requests.

Ans. to Q. No. 4: For communicating between two or more computers some rules and guidelines needed to be followed. A protocol is simply a set of rules and guidelines for communicating data.

There are numerous number of protocols required for efficient and smooth communication. Some are: Hyper Text Transfer Protocol (HTTP), File Transfer Protocol (FTP), Simple mail Transfer Protocol (SMTP) etc.

Ans. to Q. No. 5: a) Computer Networks; b) Markup Language

Ans. to Q. No. 6: Virus is a program which disrupts the normal functioning of your computer systems. Computers attached to internet are more prone to virus attacks and they can end up into crashing your whole hard disk.



1.12 MODEL QUESTIONS

Q.1: What is Internet? Explain how it is helping people from Worldwide.

Q.2: Write down the advantages and disadvantages of Internet.

Q.3: What is World Wide Web?

Q.4: What do you mean by Website? How is it built?

Q.5: Explain the different types of Web documents.

Q.6: What do you mean by File Transfer Protocol?

Q.7: What is Telnet?

Q.8: Why E-mail is used for communication purpose, Explain.

Q.9: Write down some characteristics of Client Server Computing Techniques.

Q.10: What do you mean Web Surfing? Write down some advantages and Disadvantages of Web Surfing.

*** ***** ***

UNIT 2: WEB CLIENT

UNIT STRUCTURE

2.1 Learning Objectives

2.2 Introduction

2.3 Web Browsers

2.4 Client Side Scripting Language

2.4.1 VB Script

2.4.2 JavaScript

2.5 Active X

2.6 Plug-ins

2.7 Let Us Sum Up

2.8 Further Reading

2.9 Answers to Check Your Progress

2.10 Model Questions

2.1 LEARNING OBJECTIVES

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

- learn about the different web browsers and some of their features
- learn about the features of client side scripting language
- learn the basics of VBScript and JavaScript
- describe why ActiveX Control is needed in Web Browser
- define Plugins and its uses.

2.2 INTRODUCTION

In the previous unit we have learnt about the basics of Internet and Web, and the various terminologies associated with these two technologies. We have also learnt few important protocols and concepts of client-server computing.

In this unit we will discuss Web Client. A web client is an application that communicates with a web server, using Hypertext Transfer Protocol (HTTP). HTTP is the protocol behind the World Wide Web. With every web

transaction, HTTP is invoked. HTTP is behind every request for a web document or graphic, every click of a hypertext link, and every submission of a form. The Web is about distributing information over the Internet, and HTTP is the protocol used to do so. In 21st Century, millions of web applications are available in the World Wide Web. For solving particular task or providing services to users we generally build Web Applications. For building Web Applications we use languages like HTML, CSS, JavaScript, VB Script, AJAX and many more. We will describe the basics of HTML, CSS, JavaScript, XML etc. in later units of this course.

Web Clients or applications are of different types, some are built for the front end of the Web and some are for the back end of the system.

2.3 WEB BROWSER

The browser application retrieves (or fetches) code, usually written in HTML (Hypertext Markup Language) and other computer languages, from a web server. Then, it interprets this code and displays it as a web page for you to view. Browsers are used primarily for displaying and accessing websites on the internet, as well as other content created using languages such as Hypertext Markup Language (HTML) and Extensible Markup Language (XML). Browsers translate web pages and websites delivered using Hypertext Transfer Protocol (HTTP) into human-readable content. Today's browsers are fully-functional software suites that can interpret and display HTML Web pages, applications, JavaScript, AJAX and other content hosted on web servers. Many browsers offer plug-ins which extend the capabilities of the software so it can display multimedia information (including sound and video), or the browser can be used to perform tasks such as videoconferencing, to design web pages or add anti-phishing filters and other security features to the browser. Some Popular Web Browsers are–

- **Netscape Navigator:** Netscape Navigator was the oldest browser created by Netscape Communications, computer Services Company, best known for its Web browsers. Navigator was one of the two most popular Web browsers in the 1990s. This browser first introduced as Mosaic Netscape 0.9 on October 13, 1994, **Netscape** was popular

during the early 1990's and in a browser war with Microsoft Internet Explorer. The popularity and use of Netscape declined in the late 1990's, which led to the open source Mozilla project.

- **Mozilla Firefox:** The **Firefox** web browser was first released in beta on September 23, 2002, as the "Mozilla Browser," although it was internally code-named "Phoenix." Firefox 1.0 was officially released on November 9, 2004. Firefox is a Web browser that is smaller, faster, and in some ways more secure than the Mozilla browser from which much of its code was originally derived. Firefox included almost all the features offered by other browsers at that time, as well as a number of new features, including a bookmarks toolbar and tabbed browsing. Firefox's user interface was also easily customizable with the help of extensions, such as auto fill and stock tracker.
- **Internet Explorer:** A Web browser that made its debut in 1995 as Microsoft's response to Netscape, one of the first graphical-based Web browsers and, at the time, the dominant browser in use with control of over 90 percent of the market. It is included with the Windows Operating System. Often abbreviated as **IE** or **MSIE**, **Microsoft Internet Explorer** is an Internet browser that allows users to utilize Internet Explorer to listen to and watch streaming content, access online banking and make purchases over the Internet, and much more. As with most modern browsers, Internet Explorer supports Java, JavaScript, ActiveX, RSS, CSS and Ajax, while also offering features like tabbed browsing, private browsing, and built-in malware and phishing protection.
- **Google Chrome:** It is one of the most widely used web browser in the world today. Google Chrome is created by Google and based on the open source Chromium project. Google released Chrome in 2008 and issues several updates a year. It is available for operating systems like Windows, Linux and Mac OS etc. Each open website runs as its own process, which helps prevent malicious code on one page from affecting others (or the computer operating system at large). The browser also supports Web standards such as HTML5 and cascading style sheets (CSS), Java, JavaScript, ActiveX and Ajax etc.

2.4 CLIENT SIDE SCRIPTING LANGUAGE

The client-side environment used to run scripts is usually a browser. The processing takes place on the end users computer. The source code is transferred from the web server to the user's computer over the internet and is run directly in the browser.

The scripting language needs to be enabled on the client computer. Sometimes if a user is conscious of security risks they may switch the scripting facility off. When this is the case a message usually pops up to alert the user when the script is attempting to run. In client-side scripting, scripts are needed because:

- Scripts are embedded within and interact with the HTML of your site, selecting elements of it, then manipulating those elements to provide an interactive experience.
- Scripts interact with a Cascading Style Sheet (CSS) file that styles the way the page looks.
- It dictates what work the server-side code is going to have to accomplish (where utility should be built around these front-end functions), and returns data that's pulled from the site in a way that's readable by the browser. For example: If there's a form for updating a profile, the back end is built to pull specific data from the database to populate that form, while the front-end scripts populate the form with that information.
- Scripts put less stress on the server because they donot require processing on the server once they're downloaded, just when post-backs are made. "Post-backs" perform specific call-and-answers with the server-side code, and respond to the user immediately.

  Some popular Client Server Scripts are:

2.4.1 VBScript

Microsoft VBScript (Visual Basic Script) is a general-purpose, lightweight and active scripting language developed by Microsoft that is modeled on Visual Basic. Nowadays, VBScript is the primary

scripting language for Quick Test Professional (QTP), which is a test automation tool.

Some features of VBScripts are:

- VBScript is a lightweight scripting language, which has a lightning fast interpreter.
- VBScript, for the most part, is case insensitive. It has a very simple syntax, easy to learn and to implement.
- Unlike C++ or Java, VBScript is an object-based scripting language and not an Object-Oriented Programming language.
- It uses Component Object Model **(COM)** in order to access the elements of the environment in which it is executing.
- Successful execution of VBScript can happen only if it is executed in Host Environment such as Internet Explorer **(IE)**, Internet Information Services **(IIS)** and Windows Scripting Host **(WSH).**

Example 2.1: A simple example of a VBScript to print "Hello World"

In text editor, and add the following lines of code. Then save it with the extension of html and then run it. It is very easy and simple scripting language.

```
<html>
<head>
<title>My First VBScript Code!!!</title>
</head>
<body>
<script type="text/vbscript">
document.write("Yes!!! I have started learning
VBScript.")
</script></body></html>
```



CHECK YOUR PROGRESS

Q.1: Write about some popular web browsers?

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Q.2: What is client side Scripting Language?

.....

.....

.....

.....

.....

.....

.....

.....

2.4.2 JavaScript

JavaScript is the most popular and one of the basic scripting language in the Web World. JavaScript is a scripting language used to create and control dynamic website content. It is an interpreted programming language with object-oriented capabilities.

JavaScript was first known as **LiveScript,** but Netscape changed its name to JavaScript, possibly because of the excitement being generated by Java. JavaScript made its first appearance in Netscape 2.0 in 1995 with the name **LiveScript**. The general-purpose core of the language has been embedded in Netscape, Internet Explorer, and other web browsers.

For instance, in-browser JavaScript is able to:

1) Add new HTML to the page, change the existing content, modify styles.
2) React to user actions, run on mouse clicks, pointer movements, key presses.
3) Send requests over the network to remote servers, download and upload files (so-called AJAX and COMET technologies).
4) Get and set cookies, ask questions to the visitor, show messages.
5) Remember the data on the client-side ("local storage").

JavaScript can be implemented using JavaScript statements that are placed within the **<script>... </script>** HTML tags in a web page.

You can place the **<script>** tags, containing your JavaScript, anywhere within your web page, but it is normally recommended that you should keep it within the **<head>** tags.

The **<script>** tag alerts the browser program to start interpreting all the text between these tags as a script. A simple syntax of your JavaScript will appear as follows:

```
<script ...>
    JavaScript code
</script>
```

The script tag takes two important attributes–

- **Language:** This attribute specifies what scripting language you are using. Typically, its value will be javascript. Although recent versions of HTML (and XHTML, its successor) have phased out the use of this attribute.

- **Type:** This attribute is what is now recommended to indicate the scripting language in use and its value should be set to "text/ javascript".

So our JavaScript segment will look like–

```
<script language = "javascript" type = "text/
javascript">
    JavaScript code
</script>
```

Let us take a simple example to print out "Hello World"–

```
<html>
    <body>
        <script language = "javascript" type =
        "text/javascript">
            document.write("Hello World")
        </script>
    </body>
</html>
```

2.5 ACTIVE X

ActiveX is a software framework created by Microsoft that adapts Component Object Model (COM) and Object Linking and Embedding (OLE) technologies for content downloaded from a network. ActiveX control is a control using Microsoft ActiveX technologies. An ActiveX control can be automatically downloaded and executed by a Web browser. ActiveX is not a programming language, but rather a set of rules for how applications should share information. Programmers can develop ActiveX controls in a variety of languages, including C, C++, Visual Basic, and Java. ActiveX controls can be downloaded as small programs or animations for Web pages, but they can also be used for any commonly-needed task by an application program in the latest Windows and Macintosh environments.

An ActiveX control is similar to a Java applet. Unlike Java applets, however, ActiveX controls have full access to the Windows operating system. This gives them much more power than Java applets, but with this power comes a certain risk that the applet may damage software or data on your machine. To control this risk, Microsoft developed a registration system so that browsers can identify and authenticate an ActiveX control before downloading it. Another difference between Java applets and ActiveX controls is that Java applets can be written to run on all platforms, whereas ActiveX controls are currently limited to Windows environments.

2.6 Plug-ins

A plug-in is a software component that adds a special additional features to a computer program. Plug-ins are commonly used in Internet browsers but also can be utilized in numerous other types of applications. A browser plug-in (such as Macromedia Flash or Apple QuickTime) allows you to play certain multimedia files within your Web browser. For example, if you wanted to watch a video on a website, you may need a plugin to play it because your browser doesn't have the tools it needs. You can think of it like getting a Blu-ray Player for your Blu-ray disc.

In popular technologies, like Internet browsers and audio/video applications, the ability to utilize plug-ins makes products more versatile and allows transparent and convenient customization according to the user's desired features. Plug-ins can also enable easier software upgrades or patches or additions by project collaborators.



CHECK YOUR PROGRESS

Q.3: Write a short note on JavaScript.

.....

.....

.....

.....

.....

Q.4: What are Plugins?

.....

.....

.....

.....

.....

Q.5: Multiple Choice Questions:

a) Among the following, which is a protocol?

i) HTTP ii) HTML

iii) XML iv) AJAX

b) JavaScript is a–

i) Scripting language used to create and control static website content.

ii) Not a Scripting Language.

iii) A Programming language used to develop applications.

iv) Scripting language used to create and control dynamic website content.

Q.6: What is VBScript?

.....

.....

.....



2.7 LET US SUM UP

- The browser application retrieves (or fetches) code, usually written in HTML (Hypertext Markup Language) and other computer languages, from a web server. Then, it interprets this code and displays it as a web page for you to view.
- The client-side environment used to run scripts is usually a browser. The processing takes place on the end users computer. The source code is transferred from the web server to the user's computer over the internet and run directly in the browser.

- The Scripting Languages we used to develop in client side are generally known as Client side scripting language.
- Microsoft VBScript (Visual Basic Script) is a general-purpose, lightweight and active scripting language developed by Microsoft that is modeled on Visual Basic.
- JavaScript is a scripting language used to create and control dynamic website content.
- ActiveX is a software framework created by Microsoft that adapts Component Object Model (COM) and Object Linking and Embedding (OLE) technologies for content downloaded from a network.
- A plug-in is a software component that adds a special additional features to a computer program.



2.8 FURTHER READING

1) Godbole, Achyut S., Kahate, Atul. *Wed Technologies– TCP/IP to Internet Architectures*. Tata McGraw Hill.
2) Deitel & Deitel. *Web Technology*. Pearson Education.
3) Kamal, Raj. *Internet and Web Technologies*. Tata McGraw Hill.
4) https://www.webopedia.com,
5) https://searchenterprisedesktop.techtarget.com/,
6) https://www.tutorialspoint.com/,
7) https://www.computerhope.com, https://www.techopedia.com



2.9 ANSWERS TO CHECK YOUR PROGRESS

Ans to Q. No. 1: There are numerous number of browsers present in the market. Some widely used browsers are:

i) **Mozilla Firefox:** The **Firefox** web browser was first released in beta on September 23, 2002, as the “Mozilla Browser,” although it was internally code-named “Phoenix.” Firefox 1.0 was officially released on November 9, 2004. Firefox is a Web browser that

is smaller, faster, and in some ways more secure than the Mozilla browser from which much of its code was originally derived. Firefox included almost all the features offered by other browsers at that time, as well as a number of new features, including a bookmarks toolbar and tabbed browsing. Firefox's user interface was also easily customizable with the help of extensions, such as auto fill and stock tracker.

ii) **Internet Explorer:** A Web browser that made its debut in 1995 as Microsoft's response to Netscape, one of the first graphical-based Web browsers and, at the time, the dominant browser in use with control of over 90 percent of the market. It is included with the Windows Operating System. Often abbreviated as **IE** or **MSIE**, **Microsoft Internet Explorer** is an Internet browser that allows users to utilize Internet Explorer to listen to and watch streaming content, access online banking and make purchases over the Internet, and much more. As with most modern browsers, Internet Explorer supports Java, JavaScript, ActiveX, RSS, CSS and Ajax, while also offering features like tabbed browsing, private browsing, and built-in malware and phishing protection.

iii) **Google Chrome:** It is one of the most widely used web browser in the world. Google Chrome is created by Google and based on the open source Chromium project. Google released Chrome in 2008 and issues several updates a year. It is available for operating systems like Windows, Linux and Mac OS etc.. Each open website runs as its own process, which helps prevent malicious code on one page from affecting others (or the computer operating system at large). The browser also supports Web standards such as HTML5 and cascading style sheets (CSS), Java, Javascript, ActiveX and Ajax etc.

Ans to Q. No. 2: Client Side Scripting is utilized when the client's (user's) browser has all the code and the page is modified on the basis of the client's (user's) information. The Web Browser executes the client

side scripting that locates within the user's computer. Client-side scripts are also known as embedded script (as they are often embedded within an HTML or XHTML document). The browser gets the page sent by the server & executes the client-side scripts. Client side scripting can't be utilized to join with the databases on the web server. Client side scripting cannot get the file system which lies at the web server.

Ans to Q. No. 3: JavaScript is the most popular and one of the basic scripting language in the Web World. JavaScript is a scripting language used to create and control dynamic website content but that might not make a lot of sense if you're new to tech. So let's replace "dynamic website content" with "things that move, refresh, or otherwise change on your screen without requiring you to manually reload a web page." It is an interpreted programming language with object-oriented capabilities.

Ans to Q. No. 4: A plug-in is a software component that adds a special additional features to a computer program. Plug-ins are commonly used in Internet browsers but also can be utilized in numerous other types of applications. A browser plug-in (such as Macromedia Flash or Apple QuickTime) allows you to play certain multimedia files within your Web browser.

Ans to Q. No. 5: a) I; b) IV

Ans to Q. No. 6: Microsoft VBScript (Visual Basic Script) is a general-purpose, lightweight and active scripting language developed by Microsoft that is modeled on Visual Basic.



2.10 MODEL QUESTIONS

Q.1: What is Web Browser?

Q.2: Write some history about the oldest Browser, Netscape Navigator.

Q.3: What is Mozilla Firefox?

Q.4: What are the languages used to develop a Web Browser?

Q.5: Write a brief note on Internet Explorer.

Q.6: What is client side Scripting Language?

Q.7: Name some client side scripting languages.

Q.8: Write a brief note on VBScript with an Example.

Q.9: Why is JavaScript most widely used in the world?

Q.10: What is ActiveX Control?

Q.11: Why is Browser Plugin used in a Web Browser?

*** ***** ***

UNIT 3: WEB SERVER

UNIT STRUCTURE

3.1 Learning Objectives

3.2 Introduction

3.3 Web Server and its Architecture

- 3.3.1 Apache Web Server
- 3.3.2 IIS Web Browser
- 3.3.3 Nginx Web Browser
- 3.3.4 Light Speed Web Browser

3.4 Image Map

- 3.4.1 Client side and Sever Side Image Map
- 3.4.2 Client Side Image Map
- 3.4.3 Server Side Image Map

3.5 Common Gateway Interface

3.6 Application Program Interface

3.7 Web Database Connectivity

- 3.7.1 Open Database Connectivity
- 3.7.2 Working

3.8 Proxy Server

3.9 Let Us Sum Up

3.10 Further Reading

3.11 Answers to Check Your Progress

3.12 Model Questions

3.1 LEARNING OBJECTIVES

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

- learn about Web Server and its architectures
- describe different types of modern day web servers
- familiar with Image Maps
- learn about Common Gateway Interface(CGI)
- explore yourself in the area of Application Program Interface (API)
- learn about open database connectivity with the web.

3.2 INTRODUCTION

We have acquainted you with different web browsers and some client-side scripting language in the previous unit. This unit is on web servers. Here we will discuss web server architecture, concept of Image map etc. and you will learn various database connectivities at the end of this unit.

3.3 WEB SERVER AND ITS ARCHITECTURE

Web Server is a computer where the web content is stored. Basically web server is used to host the web sites but there exist other web servers also such as gaming, storage, FTP, email etc. Web server is a program that uses HTTP to serve files that create web pages for the users in response to their requests, sent by their computers HTTP connection.

Any server that delivers an XML document to another device can be a web server. A better definition might be that a web server is an Internet server that responds to HTTP requests to deliver content and services.

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

- Sending the file to the client associated with the requested URL.
- Generating response by invoking a script and communicating with database.
- When the client sends request for a web page, the web server searches for the requested page if requested page is found then it will send it to client with an HTTP response.
- If the requested web page is not found, web server will send an **HTTP response: *Error 404 Not found*.**
- If client has requested for some other resources then the web server will contact to the application server and data store to construct the HTTP response.

Web Server Architecture follows the following two approaches:

- Concurrent Approach
- Single-Process-Event-Driven Approach

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

- Multi-process
- Multi-threaded
- Hybrid method.
- **Multi-Processing:** In this a single process (parent process) initiates several single-threaded child processes and distribute the incoming requests to these child processes. Each of the child processes is responsible for handling single request. It is the responsibility of parent process to monitor the load and decide if processes should be killed or forked.
- **Multi-threaded:** Unlike Multi-process, it creates multiple single-threaded process.
- **Hybrid:** It is a combination of above two approaches. In this approach multiple process are created and each process initiates multiple threads. Each of the threads handles one connection. Using multiple threads in a single process results in less load on system resources.

Single-Process-Event-Driven Approach: The single-process event-driven (SPED) approach is an approach which uses a single event-driven server process to perform concurrent processing of multiple HTTP requests.

Let us now discuss some commonly used web servers.

3.3.1 Apache Web Server

One of the most popular web servers in the world developed by the Apache Software Foundation. Apache is an open source software which supports almost all operating systems including Linux, Unix, Windows, FreeBSD, Mac OS X and more. About 60% of machines run on Apache Web Server. Customization of apache web server is easy as it contains a modular structure. It is also an open source which means that you can add your own modules to the server when you require and can make modifications that suit your requirements.

3.3.2 IIS Web Server

IIS is a Microsoft product. This server has all the features just like apache. But it is not an open source, and moreover adding personal modules is not easy and modification becomes a little difficult job. Microsoft developed this product and also maintains it. Thus it works with all the windows operating system platforms. Also, they provide good customer support if it had any issues.

3.3.3 Nginx Web Server

Another free open source web server is Nginx, it includes IMAP/POP3 proxy server. Nginx is known for its high performance, stability, simple configuration and low resource usage. This web server does not use threads to handle request; it rather uses a much more scalable event-driven architecture which uses small and predictable amounts of memory under load. It has become popular in the recent times and it is hosting about 7.5% of all domains worldwide. Most of the web hosting companies are using this in recent times.

3.3.6 Light Speed Web Server

LiteSpeed (LSWS) is a high-performance Apache drop-in replacement. LSWS is the 4th most popular web server on the internet and it is a commercial web server. Upgrading your web server to LiteSpeed will improve performance and lower operating costs. Lightspeed can run on Windows, Mac OS X, Linux and Solaris operating systems.

3.4 IMAGE MAP

An Image Map on the World Wide Web permits the web document writer to include in its document the ability for a viewer to “click” on a portion of a GIF or JPEG (or any other image) file. There are three components in a Web page image map:

1) **An image:** a normal web image, usually stored in the GIF or JPEG format.
2) **A set of map data:** a description of the mapped regions within the image.
3) **An HTML host entry:** the HTML code that positions the image within the web page and designates the image as having map functionality.

The regions in a map include shape descriptions (rectangle, circle, polygon, or background), numeric data defining the coverage of the regions within the image, and link URLs.

3.4.1 Server-side and Client-side Image Maps

The implementation of image maps requires the processing of the mouse click coordinates in conjunction with knowing the image areas and their associated hypertext links. The processing can take place in one of the two places:

- On the server where the Web pages are stored.
- In the Web browser.

Originally, the only option for processing image maps was to do so on the server. However, later versions of HTML have added features allowing client-side processing. Many people are now developing their new pages (and updating older pages) to process image maps in the Web browser itself, thus reducing the amount of processing required by the server

3.4.2 Server-side Image Maps

Server-side image mapping uses images, .map files and CGI scripts. Often server-side image maps use the Imagemap programme.

The programme processing image maps needs to be on the server, usually in the cgi-bin/ directory.

The HREF attribute of server side imagemaps is always a hyperlink to the cgi-bin directory:

```
<a href=”/cgi-bin/imagemap/user_dir/image.map”>
<img src=”/user_dir/image.gif”>
</a>
```

In fact, the HREF path can sometimes be interpreted as a combination of two paths:

- The location of the “imagemap” programme, such as /cgi-bin/ imagemap
- The location of the (coordinate) map for the image, user_dir/ image.map. This describes the various areas of the image.

A major disadvantage of server-side image maps is that their implementation can differ between different server systems. This means that the author of the website using image maps needs to communicate closely with the Web server’s administrator to ensure that the maps work correctly.

Since there are so many advantages to client-side image maps, and almost all current browser versions support them, we shall not explore server-side image maps further in this unit (although most image map programmes still provide features to output/edit server-side map files).

3.4.3 Client-side Image Maps

Clients-side image maps have a number of advantages over the server-side maps. These are:

- Reduced server load, since the mapping is done on the client.
- There is no need to communicate with server side scripts.
- The user can see the image maps associated hypertext URLs in the status bar of their browser before clicking.



CHECK YOUR PROGRESS

Q.1: What is a Web Server? Write about Apache Web Server.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Q.2: What is an Image map?

.....

.....

.....

.....

.....

.....

.....

.....

3.5 COMMON GATEWAY INTERFACE

CGI is the abbreviation of ***C**ommon **G**ateway **I**nterface.* It is a specification for transferring information between a World Wide Web server and a CGI *program*. A CGI program is any program designed to accept and return data that conforms to the CGI specification. The program could be written in any *programming language*, including *C*, *Perl*, *Java*, or *Visual Basic*.

An interesting aspect of a CGI enabled Web server is that computer programs can be created and deployed that can accept user input and create a webpage on the fly. Unlike static Web pages that display some preset information, these interactive web pages enable a client to send information to the Web server and get back a response that depends on the input.

A Web search engine is a good example of an interactive web page. The client enters one or more keywords, and the Web index returns a list of Web pages that satisfy the search criteria entered. The Web page returned by the Web index is also dynamic, because the content of that page depends on what the client types in as search words– it's not a predefined static document.

To create an interactive Web page, HTML elements are used to display a form that accepts a client's input and passes this to special computer programs on the Web server, these computer programs process a client's input and return requested information, usually in the form of a web page constructed on the fly by the computer program.

These programs are known as gateways because they typically act as a conduit between the Web server and an external source of information, such as a database. Gateway programs exchange information with the Web server using a standard known as The Common Gateway Interface. This is the reason CGI programming is used to describe the task of writing computer programs that handle client requests for information.

The term 'gateway' describes the relationship between the WWW server and external applications that handle data access and manipulation chores on its behalf. A gateway interface handles information requests in an orderly fashion, and then returns an appropriate response.

In other words, CGI allows a WWW server to provide information to WWW clients that would otherwise not be available to those clients. This could, for example, allow a WWW client to issue a query to an Oracle database and receive an appropriate response in the form of a custom built Web document. Some common uses of CGI include: Gathering user feedback about a product line through an HTML form. Querying an Oracle database and rendering the result as an HTML document.

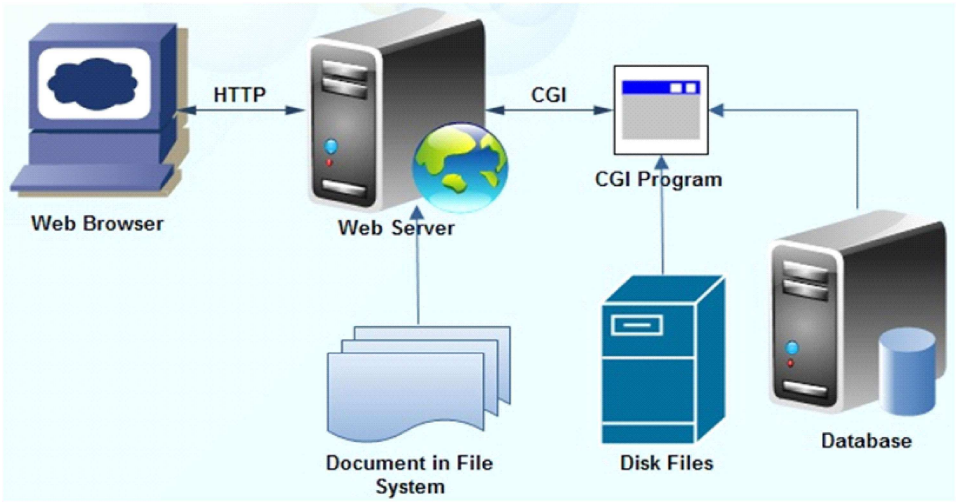


Figure 3.1: Working of CGI

The sequence of events for creating a dynamic HTML document on the fly through CGI scripting is as follows:

1) A client makes an HTTP request by means of a URL. This URL could be typed into the 'Location' window of a browser, be a hyperlink or be specified in the 'Action' attribute of an HTML <form> tag.
2) From the URL, the Web server determines that it should activate the gateway program listed in the URL and send any parameters passed via the URL to that program.
3) The gateway program processes the information and returns HTML text to the Web server. The server, in turn, adds a MIME header and returns the HTML text to the Web browser.
4) The Web browser displays the document received from the Web server.

3.6 APPLICATION PROGRAMMING INTERFACES

Application Programming Interfaces (APIs) are a set of tools that allow software developers to create software applications. Basically, an API specifies how software components should interact. Additionally, APIs are used when programming graphical user interface *(GUI)* components. A good API makes it easier to develop a *program* by providing all the building blocks. A *programmer* then puts the blocks together. A Web API is a kind of a framework which makes building HTTP services easier than ever. It can be

used almost everywhere including wide range of clients, mobile devices, browsers etc.

The advantage of using APIs are:

Faster development: Since the functionality is already implemented, we do not have to implement that functionality by ourselves; instead we just need to call the API.

Error handling: The implementation of the API is well tested and has minimal bugs. This could eliminate any bugs from creeping in had the functionality been coded specifically instead of using the API.

Some examples of API are:

- **Google Maps** is a web API. By creating a simple HTML page and adding a few lines of Javascript, a software developer can embed a fully featured map display in their own application. With a few more lines of code, a developer can add custom markers, lines and popups. All this is done without having to download or maintain large sets of map data.
- The **Amazon** Product Advertising API allows developers to advertise Amazon products on their website for profit.
- **Microsoft Windows** provides many APIs that allow developers to create rich graphical applications. These APIs allow developers to instruct Windows to display a new form, perform an action when the user clicks a button, and to read and write data to the disk.

3.7 WEB DATABASE CONNECTIVITY

A web database is a database application designed to be managed and accessed through the Internet. The content of any website for a business or organization comprises data, including text, images, media and numerical values. Using a database to store this data is an efficient approach for many sites. This process involves connecting to the database, querying it for data and presenting the data in HTML, often by using a server side scripting language such as PHP. Retrieving data from **databases** and developing dynamic **web** pages are essential and important problems to construct responsive **web** site.

3.7.1 Open Database Connectivity

Open Database Connectivity or ODBC is an application programming interface (API) that lets software connect with database management systems while remaining independent of them. This is important, because it allows applications to interact with multiple databases simultaneously using *SQL* (Structured Query Language). It was developed back in 1992 with the sole purpose of facilitating the work with *database management systems* (DBMS). It was created by the Microsoft corporation and is since then included in every copy of Microsoft Windows.

Some of the advantages of ODBC are:

- ODBC provides a consistent interface regardless of the kind of database server used.
- You can have more than one concurrent connection.
- Applications do not have to be bound to each database on which they will run.

3.7.2 Working

ODBC consists of four components, which work together to enable functions. ODBC allows programs to use SQL requests that access databases without knowing the proprietary interfaces to the databases. ODBC handles the SQL request and converts it into a request that each database system understands.

The four different components of ODBC are:

- **Application**: Processes and calls the ODBC functions and submits the SQL statements;
- **Driver manager:** Loads drivers for each application;
- **Driver**: Handles ODBC function calls, and then submits each SQL request to a data source; and
- **Data source**: The data being accessed and its database management system (DBMS) OS.

3.8 PROXY SERVER

Proxy server is a computer that sits between a client computer and the Internet, and provides indirect network services to a client. It may reside on the user's local computer, or at various points between the user's computer and destination servers on the Internet. A proxy server intercepts all client requests, and provides responses from its cache or forwards the request to the real server. A client computer is connected to the proxy server, which acknowledges client requests by providing the requested resource/ data from either a specified server or the local cache memory. Client requests include files or any other resources available on various servers. The proxy server's most important role is providing security.

Following are some important reasons to use proxy servers:

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



CHECK YOUR PROGRESS

Q.3: What are APIs? Write some advantages.

.....

.....

.....

.....

.....

.....

.....

.....

.....

Q.4: What is Proxy Server and why it is used?

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Q.5: Which of the following is true?

i) Web Server and Web Browser is similar.

ii) Web Server is a computer where the web content is stored and Web Browser request those content.

iii) Web Browser is a computer where the web content is stored and Web Server request those content.

iv) Web Server & Web Browser both follows same protocol while storing data.

Q.6: Name two approaches of Web Server Architecture.

.....

.....

.....

Q.7: What do you mean by Google Maps?

.....

.....

.....

.....



3.9 LET US SUM UP

- Web server is used to host the web sites but there exist other web servers also such as gaming, storage, FTP, email etc. Web server is a program

that uses HTTP to serve files that create web pages to users in response to their requests, which is sent by their computers HTTP connection.

- Web Server Architecture follows the following two approaches: Concurrent Approach, Single-Process-Event-Driven Approach.
- An Image Map on the WWW permits the web document writer to include in its document the ability for a viewer to "click" on a portion of a GIF or JPEG (or any other image) file.
- CGI is the abbreviation of ***C**ommon **G**ateway **I**nterface.* It is a specification for transferring information between a World Wide Web server and a CGI *program*. A CGI program is any program designed to accept and return data that conforms to the CGI specification. The program could be written in any *programming language*, including *C*, *Perl*, *Java*, or *Visual Basic*.
- **APIs** are a set of tools that allow software developers to create software applications. Basically, an API specifies how software components should interact.
- A Web database is a database application designed to be managed and accessed through the Internet. The content of any website for a business or organization comprises data, including text, images, media and numerical values. Using a database to store this data is an efficient approach for many sites.
- Open Database Connectivity or ODBC is an application programming interface (API) that lets software connect with database management systems while remaining independent of them.
- Proxy server is a computer that sits between a client computer and the Internet, and provides indirect network services to a client. It may reside on the user's local computer, or at various points between the user's computer and destination servers on the Internet.



3.10 FURTHER READING

1) Godbole, Achyut S., Kahate, Atul. *Wed Technologies– TCP/IP to Internet Architectures*. Tata McGraw Hill.
2) Deitel & Deitel. *Web Technology*. Pearson Education.

3) Kamal, Raj. *Internet and Web Technologies*. Tata McGraw Hill.
4) https://www.webopedia.com,
5) https://searchenterprisedesktop.techtarget.com/,
6) https://www.tutorialspoint.com/,
7) https://www.computerhope.com, https://www.techopedia.com



3.11 ANSWERS TO CHECK YOUR PROGRESS

Ans. to Q. No. 1: Web Server is a computer where the web content is stored. Basically web server is used to host the web sites but there exists other web servers also such as gaming, storage, FTP, email etc. Web server is a program that uses HTTP to serve files that create web pages to users in response to their requests, which is sent by their computers HTTP connection.

One of the most popular web server in the world developed by the Apache Software Foundation. Apache is an open source software which supports almost all operating systems including Linux, Unix, Windows, FreeBSD, Mac OS X and more. About 60% of machines run on Apache Web Server. Customization of apache web server is easy as it contains a modular structure. It is also an open source which means that you can add your own modules to the server when to require and make modifications that suit your requirements.

Ans. to Q. No. 2: An Image Map on the WWW permits the web document writer to include in its document the ability for a viewer to “click” on a portion of a GIF or JPEG (or any other image) file. There are three components in a Web page image map:

1) **An image:** a normal web image, usually stored in the GIF or JPEG format.
2) **A set of map data:** a description of the mapped regions within the image.
3) **An HTML host entry:** the HTML code that positions the image within the web page and designates the image as having map functionality.

The regions in a map include shape descriptions (rectangle, circle, polygon, or background), numeric data defining the coverage of the regions within the image, and link URLs.

Ans. to Q. No. 3: APIs are a set of tools that allow software developers to create software applications. Basically, an API specifies how software components should interact. Additionally, APIs are used when programming graphical user interface (GUI) components. A good API makes it easier to develop a program by providing all the building blocks. A programmer then puts the blocks together. A Web API is a kind of a framework which makes building HTTP services easier than ever. It can be used almost everywhere including wide range of clients, mobile devices, browsers etc.

The advantage of using APIs are:

1) Faster development: Since the functionality is already implemented, you don't have to implement that functionality yourself; instead you just need to call the API.
2) Error handling: The implementation of the API is well tested and has minimal bugs. This could eliminate any bugs from creeping in had the functionality been coded specifically instead of using the API.

Ans. to Q. No. 4: Proxy server is a computer that sits between a client computer and the Internet, and provide indirect network services to a client. It may reside on the user's local computer, or at various points between the user's computer and destination servers on the Internet. A proxy server intercepts all client requests, and provide responses from its cache or forwards the request to the real server. A client computer is connected to the proxy server, which acknowledges client requests by providing the requested resource/ data from either a specified server or the local cache memory. Client requests include files or any other resources available on various servers. The proxy server's most important role is providing security.

Following are some important reasons to use proxy servers:

- Monitoring and Filtering

- Improving performance
- Translation
- Accessing services anonymously
- Security

Ans. to Q. No. 5: ii) Web Server is a computer where the web content is stored and Web Browser request those content.

Ans. to Q. No. 6: Web Server Architecture follows the following two approaches:

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

Ans. to Q. No. 7: Google Maps is a Web-based service that provides detailed information about geographical regions and sites around the world. In addition to conventional road maps, Google Maps offers aerial and satellite views of many places. In some cities, Google Maps offers street views comprising photographs taken from vehicles.



3.12 MODEL QUESTIONS

Q.1: What is Web Server?

Q.2: Explain the Web Server architecture.

Q.3: Write about Apache & IIS Web Server.

Q.4: What do you mean by Image Map?

Q.5: Distinguish between Server-side and Client-side image maps.

Q.6: Write a brief note on CGI.

Q.7: Explain the sequence of events for creating a dynamic HTML document on the fly through CGI.

Q.8: Why are APIs considered very important for systems?

Q.9: Explain Web database connectivity.

Q.10: What is ODBC? Write down its components.

*** ***** ***

UNIT 4: INTRODUCTION TO HTML

UNIT STRUCTURE

4.1 Learning Objectives

4.2 Introduction

4.3 Basics of HTML

4.3.1 HTML Tags

4.3.2 HTML Editor

4.4 Document Structure Tags

4.5 Formatting Tags

4.6 Hypertext, Hyperlink and Hypermedia

4.7 Anchor and URL

4.8 Inserting Images

4.9 Let Us Sum Up

4.10 Further Reading

4.11 Answers to Check Your Progress

4.12 Model Questions

4.1 LEARNING OBJECTIVES

After going through this unit, you will able to:

- learn about HTML, HTML editor
- create simple web pages using basic HTML tags
- explain the use of document structure tags
- learn formatting tags
- define hypertext, hyperlink and hypermedia
- learn the use of anchors and URLs
- link differnent web pages
- insert image in HTML document.

4.2 INTRODUCTION

In the previous unit we have acquainted you with various concepts associated with World Wide Web. Web pages, web servers, web browsers search engines etc. are discussed in this unit.

In this unit we will learn how to create simple web pages using HTML tags. Here, we will discuss mainly formatting tags, image tags and tags related with linking different web pages.

4.3 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 we can design web pages. When we see a web page on the Internet and if we click at 'view source' by right clicking on the page, we will be able to visualize the HTML code behind the page. The HTML language is based on tags. A tag tells the browser what to do. 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 us error if we make mistake in writing codes. The only possibility is that we will not get our desired output. It displays outputs only for the correct parts of what we have written.

4.3.1 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. For example, the first tag is <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 does not have any closing counterpart. So it does not contain anything. It affects at the position where it is applied. For example, <BR> tag. It breaks a line in the output at the position where it appears.

4.3.2 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.

4.4 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.

Let us now discuss the most common HTML tags.

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

Syntax:

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

ii) **<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>
```

iii) **<TITLE> tags:** This tag is used for creating the title of the web page document. It is a container tag.

Syntax:

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

iv) **<BODY> tags:** It is used for creating 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.

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>
```



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 hexadecimal 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".

This will change the background of the web page to yellow. The text "Krishna Kanta Handique State Open University" will be in blue colour.

4.5 FORMATING TAGS

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

i) **<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.

ii) **<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.

iii) **<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>
```

iv) **<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 >
```

v) **<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>
```

vi) **<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>
```

vii) **<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>
```

viii) **<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>
```

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

Syntax :

```
<Sub> … subscript text goes here... </Sub>
```

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

Syntax:

```
<Sup> … superscript text goes here... </Sup>
```

xi) **<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>
```

xii) **<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>
```

xiii) **<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>
```

xiv) **<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>
```

xv) **
 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:

```
<BR>
```

xvi) **<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>
```

xvii) **<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>
```

xviii) **<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.

xix) **<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.

xx) **<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>
```

xxi) **<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>
```

xxii) **<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.1: Fill in the blanks:

i) To change the style of the font, ... attribute is used.

ii) To change the background colour of a we page, ... attribute is used.

iii) <B>...</B> tag is used to change the enclosed tag to be .....

iv)
 tag is a ..... tag.

v) <H1> creates the ..... possible heading whereas <H6> creates the ..... possible heading.

Q.2: Do all HTML tag come in pair?

.....

.....

4.6 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 a text which is not constrained to be linear. It is a 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 the 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:

```
<a href=''http://www.kkhsou.org/"
target=''_blank">Go</a>
```

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.

4.7 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.

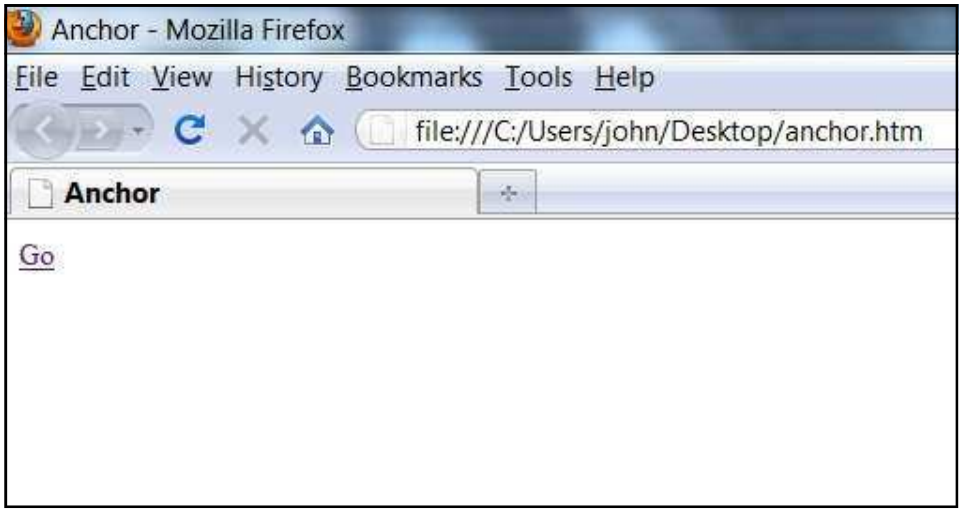
The **<a>** tag supports the following standard attributes:

access key	Specified 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:



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:

```
<img src=''http://www.webinfo.com/images/email.gif"
width=''..." height=''..."/>
```

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

```
<img src=''../images/email.gif" width=''..."
height=''..." />
```



CHECK YOUR PROGRESS

Q.3: 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.4: 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.

4.8 INSERTING IMAGE

To insert image in a webpage, <img> tag is used.

<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.



4.7 LET US SUM UP

- HTML, which stands for Hyper Text Markup Language, is the Universal language on which the web documents(pages) are written (e.g., 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.
- 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.
- <img> tag is used to insert images in a web page.



4.8 FURTHER READING

1) Batross, I. (2009). *Web Enabled Commercial Application Development Using HTL, DHTML, Javascript, Peril CGI*. Bpb Publications.
2) Duckett, J. (2011). *Beginning html, xhtml, css, and javascript*. John Wiley & Sons.
3) Sikos, L. (2014). *Web Standards: Mastering HTML5, CSS3, and XML*. Apress.



4.9 ANSWERS TO CHECK YOUR PROGRESS

Ans. to Q. No. 1: i) font face; ii) bgcolor; iii) bold; iv) standalone; v) largest, smallest

Ans. to Q. No. 2: No,there are single HTML tags that do not need a closing tag. Examples are <img> tag and
tag.

Ans. to Q. No. 3: i) hyperlinks; ii) anchor; iii) two; iv) resource; v) relative

Ans. to Q. No. 4: i) false; ii) true; iii) false; iv) true; v) true; vi) false



4.9 MODEL QUESTIONS

Q.1: What is HTML? Mention the uses of HTML.

Q.2: What are HTML tags and attributes? Explain with examples.

Q.3: What is DIV tag and how can it be used? Explain with an example.

Q.4: Define the terms "Hypertext", "Hyperlink" and "Hypermedia" with appropriate example.

Q.5: What are Anchors? Mention some attributes along with an example?

Q.6: What is an URL? Mention the types of URL?

Q.7: What is header and body tag? Write a simple HTML code to show the use of these two tags.

*** ***** ***

UNIT 5: HTML LISTS AND TABLES

UNIT STRUCTURE

5.1 Learning Objectives

5.2 Introduction

5.3 HTML List

5.4 HTML Table

5.5 Let Us Sum Up

5.6 Further Reading

5.7 Answers to Check Your Progress

5.8 Model Questions

5.1 LEARNING OBJECTIVES

After going through this unit, you will able to :

- create HTML list
- learn about the different types of HTML lists
- create HTML tables
- use the different attributes in designing HTML tables.

5.2 INTRODUCTION

In the previous unit we have learnt abount the basic HTML tags and formatting tags. We have also learnt to insert images in a HTML document.

In this unit we will mainly discuss how to create HTML list and tables in a web page. We will learn to add various attributes associated with HTML list ans tables.

5.3 HTML LIST

HTML lists are used to present list of information in an organized and sequential manner. There are three different types of list in HTML and each one has a specific purpose and meaning:

i) Unordered list

ii) Ordered list

iii) Definition list

i) Unordered Lists: An unordered list is a collection of related items that have no special order or sequence. An unordered list is created using the <ul> tag, and each list item starts with the <li> tag. The <ul> tag is a container tag. The list items in unordered lists are marked with bullets (small black circles), by default. It is normally known as bulleted list.

For example, let us consider the following HTML code:

```
<ul>
    <li>MCA</li>
    <li>M.Sc IT</li>
    <li>PGDCA</li>
</ul>
```

The output of the above example will look something like this:

- MCA
- M.Sc IT
- PGDCA

We can use ***type*** attribute for <ul> tag to specify the type of bullet we like. By default, it is a disc. Following are the possible options–

```
<ul type = "square">
<ul type = "disc">
<ul type = "circle">
```

ii) Ordered Lists: If we are required to put our items in a numbered list instead of bulleted, then HTML ordered list will be used. This list is created by using <ol> tag. The numbering starts at one and is incremented by one. An ordered list is created using the <ol> tag, and each list item starts with the <li> tag. For example,

```
<ol>
    <li> Flow chart</li>
    <li>Algorithm</li>
    <li>Programming</li>
</ol>
```

The output of the above code will look something like this:

1) Flow chart
2) Algorithm
3) Programming

The ***type*** attribute of the *<ol>* tag, defines the type of the list item marker:

Type	Description
type="1"	The list items will be numbered with numbers (default)
type="A"	The list items will be numbered with uppercase letters
type="a"	The list items will be numbered with lowercase letters
type="I"	The list items will be numbered with uppercase roman numbers
type="i"	The list items will be numbered with lowercase roman numbers

For example,

```
<ol type="a">
    <li>Water</li>
    <li>Milk</li>
    <li>Tea</li>
</ol>
```

The above code will give the output as:

a) Water
b) Milk
c) Tea

Similarly, if we use <ol type="i"> in the above code, then the output will look like:

i) Water
ii) Milk
iii) Tea

The start Attribute: We can use ***start*** attribute for <ol> tag to specify the starting point of numbering we need. Following are the possible options:

```
<ol type = "1" start = "3">- Numerals starts with 3.
<ol type = "i" start = "4">- Numerals starts with iv.
<ol type = "a" start = "3">- Letters starts with c.
<ol type = "A" start = "3">- Letters starts with C.
```

This can be used as follows:

```
<ol type = "i" start = "3" >
```

iii) Definition Lists: HTML supports definition lists where entries are listed like in a dictionary or encyclopedia. The definition list is the ideal way to present a glossary, list of terms, or other name/value list. Definition List makes use of following three tags:

<dl> – defines the start of the list

<dt> – defines the term(name)

<dd> – describes each term

</dl> – defines the end of the list

For example,

```
<dl>
    <dt>Coffee</dt>
    <dd>- black hot drink</dd>
<dt>Milk</dt>
    <dd>- white cold drink</dd>
</dl>
```

The output will look like this:

Coffee

– black hot drink

Milk

– white cold drink

Inside a list item we can put text, line breaks, images, links, etc. We can also place an entire list inside a list item to create nested list.

- **Nested HTML Lists:** When a list is made inside another list, then it is called nested list. For example,

```
<ul>
    <li>Coffee</li>
    <li>Tea
    <ul>
        <li>Black tea</li>
        <li>Green tea</li>
    </ul>
```

```
    </li>
    <li>Milk</li>
</ul>
```

The output will be like this:

- Coffee
- Tea
  - Black tea
  - Green tea
- Milk



CHECK YOUR PROGRESS

Q1: Fill in the gaps:

a) An ..... list is created using the <ul> tag.

b) The unordered list is also termed as ..... list.

c) An ..... list is a collection of related items that have no special order or sequence.

d) Each list item starts with the ..... tag.

Q.2: Which attribute and tag is used to change the starting number of an ordered list?

.....

.....

Q.3: Which attribute and tag is used to change the design of bullet in an unordered list?

.....

.....

5.4 HTML TABLE

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.

The **<th>** tage is used to define a table header. By default, table headings are bold and centered.

<TABLE> tag: It is a container tag. The starting tag is <table >and the ending table tag is </table>. All the related tags for creating a table are included within this tag. For example, to create a table as shown in Figure 11.1, we have to write the following HTML code.

Example 5.1:

```
<table border='1' style="width:40%">
<tr>
    <th>Firstname</th>
    <th>Lastname</th>
    <th>Age</th>
</tr>
<tr>
    <td>Baroon</td>
    <td>Sarma</td>
    <td>24</td>
</tr>
<tr>
    <td>Megha</td>
    <td>Baruah</td>
    <td>21</td>
</tr>
</table>
```

First Name	Last Name	Age
Baroon	Sarma	24
Megha	Baruah	21

Figure 5.1: A HTML table of 3 rows and 3 columns

We can change the design of the table according to our requirements using table attributes.

- 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. 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 5.2:

```
<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 gives the following output in Mozilla Firefox: Fire

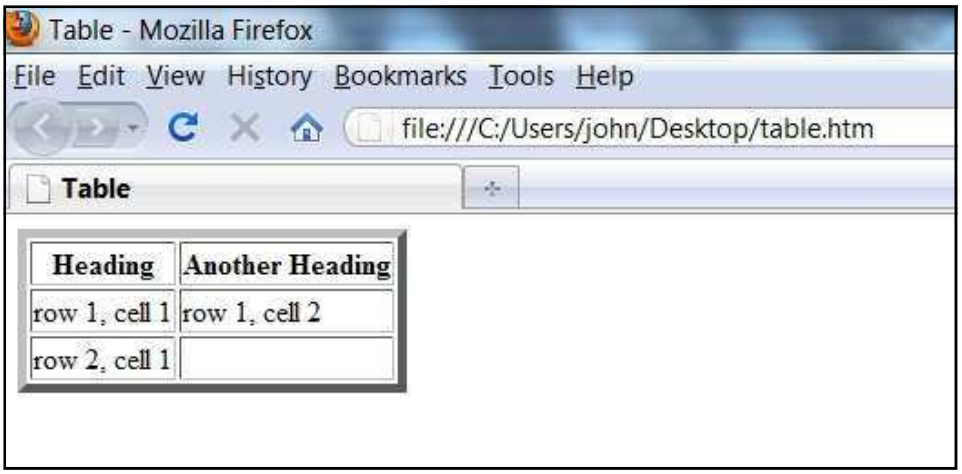


Figure 5.2: HTML table

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

Attribute	Description
align	Spefifies 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 will 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.

Example 5.3:

```
<table border="1" celpadding="5">
<tr><th colspan="2">Production</th>
    </tr>
<tr><td>Kaberi Sarma</td>
        <td>38000</td>
    </tr>
<tr><td>Gautam Baruah</td>
        <td>43000</td>
    </tr>
<tr><td>Dipankar Das </td>
    <td>45000</td>
</tr>
<tr><th colspan="2">Sales</th>
    </tr>
<tr><td>Rajib Saikia</td>
        <td>38000</td></tr>
        <tr><td>Meghali Dutta</td>
        <td>18000</td>
    </tr>
<tr><td>Jumi Saikia</td>
        <td>26000</td>
    </tr>
</table>
```

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

Production		Sales	
Kaberi Sarma	38000	Rajib Saikia	38000
Gautam Baruah	43000	Meghali Dutta	18000
Dipankar Das	45000	Jumi Saikia	26000

Figure 11.3: Use of *colspan* in the cells containing *Production* and *Sales*

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.

Example 5.4:

```
<table border="2" celpadding="4">
   <tr>
      <th rowspan="3" bgcolor="lime">Production</th>
      <td>Plabita Das</td> <td>550</td>
   </tr>
      <tr><td>Rita Bhuyan</td> <td>380</td></tr>
      <tr><td>Pankaj Sarma</td> <td>300</td></tr>
   <tr>
      <th rowspan="3" bgcolor="lime">Sales</th>
      <td>Hrishi Sarma</td> <td>500</td>
   </tr>
      <tr><td>Rahul Baruah</td> <td>270</td></tr>
      <tr><td>Mousumi Dutta</td> <td>550</td>
            </tr>
</table>
```

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

Production	Plabita Das	550
	Rita Bhuyan	380
	Pankaj Sarma	300
Sales	Hrishi Sarma	500
	Rahul Baruah	270
	Mousumi Dutta	550

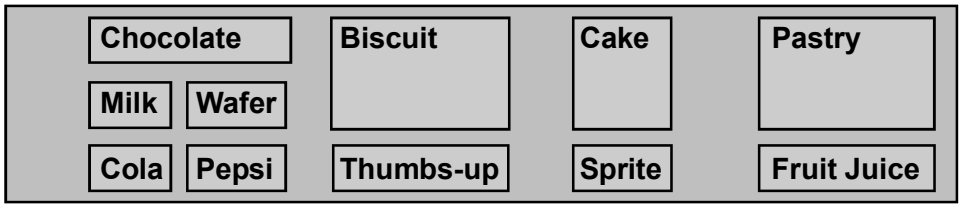
Figure 5.4: Use of *rowspan* in the cells containing *Production* and *Sales*

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

Example 5.5:

```
<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">Biscuit</td>
<td rowspan="2">Cake</td>
<td rowspan="2">Pastry</td>
</tr>
<tr>
<td>Milk</td>
<td>Wafer</td>
</tr>
<tr>
<td>Cola</td>
<td>Pepsi</td>
<td>Thumbs-up</td>
<td>Sprite</td>
<td>Fruit Juice</td>
</tr>
</table>
</body>
</html>
```

The above code gives the following output in Mozilla Firefox:





CHECK YOUR PROGRESS

Q.4: Write the HTML code to create the following table:

PG	MCA
	M.Sc.IT



5.5 LET US SUM UP

- HTML lists are used to present list of information in an organized and sequential manner.
- An unordered list is created using the <ul> tag, and each list item starts with the <li> tag
- Unordered lists are also called bulleted list.
- An ordered list is created using the <ol> tag.
- The *type* attribute of the *<ol>* tag defines the type of the list item marker.
- The type can be in uppercase letters, lowercase letters uppercase roman, lowercase roman numbers etc.
- The starting point or number of an ordered list can be changed by specifying the number in the start attribute.
- A definition list is a list of terms and corresponding definitions.
- The opening list tag must be <DL>. It is followed by an optional list header (<LH>caption</LH>) and then by term names (<DT>) and definitions (<DD>).
- The <table> tag defines an HTML table.
- An HTML table consists of the <table> element and one or more <tr>, <th>, and <td> elements.

- The <tr> element defines a table row, the <th> element defines a table header, and the <td> element defines a table cell.
- The "align", "bgcolor", "border", "cellpadding", "cellspacing", "width" etc. are some of the attributes of table.
- Cellpadding specifies the space between the cell wall and the cell content.
- Cellspacing specifies the space between cells.
- The rowspan attribute specifies the number of rows a cell should span.
- The colspan attribute defines the number of columns a cell should span.



5.6 FURTHER READING

1) Batross, I. (2009). *Web Enabled Commercial Application Development Using HTL, DHTML, Javascript, Peril CGI*. Bpb Publications.
2) Duckett, J. (2011). *Beginning html, xhtml, css, and javascript*. John Wiley & Sons.
3) Sikos, L. (2014). *Web Standards: Mastering HTML5, CSS3, and XML*. Apress.



5.7 ANSWERS TO CHECK YOUR PROGRESS

Ans. to Q. No. 1: a) unordered; b) bulleted; c) unordered; d)

Ans. to Q. No. 2: The <ol> tag and *start* attribute. For example,

```
<ol type = "1" start = "3"> - Number starts from 3.
```

Ans. to Q. No. 3: We can use *type* attribute for <ul> tag to specify the type of bullet we like.

Ans. to Q. No. 4: The HTML code will be as follows:

```
<html>
<body>
<table style= "width: 30%" align = "center">
<tr>
   <td bgcolor = "yellow" rowspan = 2 width = 75> PG</td>
   <td bgcolor = "pink" height =100> MCA</td>
</tr>
```

```
<tr>
   <td bgcolor = "lime" height =100> M.ScIT</td>
</tr>
</table>
</body>
</html>
```



5.8 MODEL QUESTIONS

Q.1: How are the HTML lists created? What are the different categories of HTML list?

Q.2: What is a definition list? What are the tags associated with HTML definition list?

Q.3: How is an ordered and unordered list created? Write the HTML code to create the following list:

```
<ul>
    <li>MCA</li>
    <li>M.Sc IT</li>
    <li>PGDCA</li>
</ul>
```

Q.4: Write the HTML codes to create a HTML table with 6 rows and 5 columns. The table should contain marks of 5 students in three different subjects, roll number and total marks.

Q.5: What are rowspan and colspan? Explain with an example each.

Q.6: Write the use of cellpadding and cellspacing.

Q.7: Why do we use HTML tables?

Q.8: Write the HTML code to add colour to a particular table cell.

Q.9: Can we create an ordered list inside an unordered list? Explain.

Q.10: What are definition lists? How are they used? Give an example.

*** ***** ***

UNIT 6: HTML FRAMES AND FORMS

UNIT STRUCTURE

6.1 Learning Objectives

6.2 Introduction

6.3 HTML Frames

6.4 HTML Forms

6.5 Let Us Sum Up

6.6 Further Reading

6.7 Answers to Check Your Progress

6.8 Model Questions

6.1 LEARNING OBJECTIVES

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

- learn about HTML frames and forms
- learn to use the different tags associated with frames and forms
- use check box, combo box, radio button etc. in designing HTML forms.

6.2 INTRODUCTION

In the previous units, we learned about HTML, which is the basic markup language used for building the primary structure of any web page. We have learnt about the various fundamental HTML tags like head, body, img, href etc. and have a got familiarized with the various tags associated with HTML tables and lists.

In this unit we will discuss how to create HTML frames and forms. We will also discuss form handling which is an integral part of any site dealing with using and processing user related data.

6.3 HTML 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. Framed layout gives a better look to a

web page. A few frame tags are used to create a framed layout. Let us discuss the most important <frameset> and <frame> tag with their attributes.

<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: it divides the window either *horizontally* or *vertically*.

Syntax:

```
<FRAMESET Rows=''row size" Cols=''column size">
    … Frame tags
</FRAMESET>
```

Attribute of FRAMESET tag: Though it has two attributes rows and cdumns (cols), 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

For example, let us consider the following example 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. Suppose, we have created two HTML document "frame_a.htm'' and "frame_b.htm". 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 6.1:

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

The HTML code for **frame_a.htm** and **frame_b.htm** are as follows:

```
<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 code in Example 6.1 gives the following output in Mozilla Firefox:

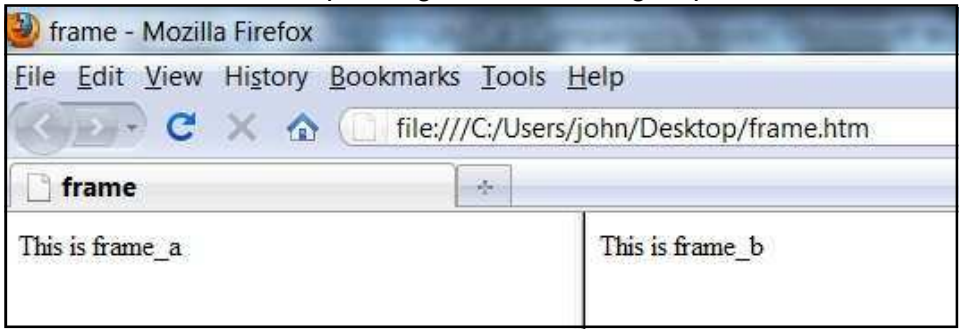


Figure 6.1: A frameset with two columns

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 by 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.

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=''00">
    <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 id for an element
style	Specifies an inline style for an element
title	Specifies extra information about an element

Example 6.2:

```
<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:

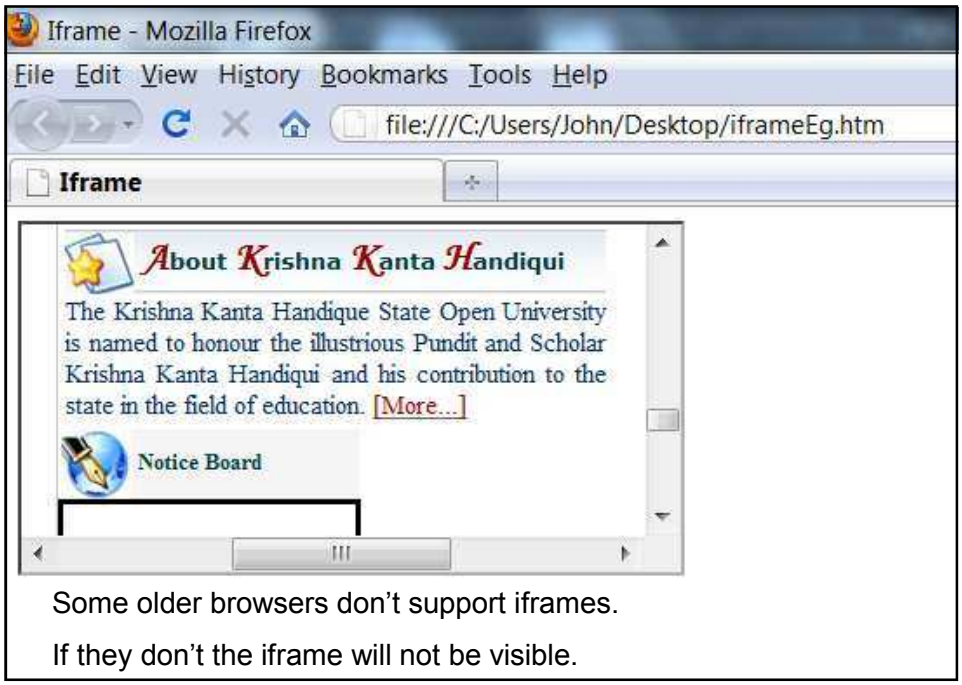


Figure 6.2: The iframe tag

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 the users that frames are required. The noframes element goes inside the frameset element. This tag is supported by all major browsers.

Example 6.3:

```
<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.1: Fill in the blanks:

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

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

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

iv) The HTML tag used for dividing the window vertically into three pars as 20%, 30% and 50% is .....

6.4 HTML FORMS

HTML forms are mainly used to collect information from different kinds of user and to pass data to a server.

A form can contain input elements like text fields, check boxes, radio-buttons, submit buttons and more. A form can also contain select lists, text area, field set, legend, and label elements. 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 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. The type of <input>

tag is specified with the type attribute. An input element can be of type text field, checkbox, password, radio button, submit button, and more. The most common input types are described below:

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

<input type="text" /> defines a one-line input field that a user can enter text into. The syntax is as follows:

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

The output will be as follows:

First name:

Last name:

The default width of a text field is 20 characters.

Radio Buttons: Radio Buttons are used when we want the user to select only one of a limited number of choices. <input type="radio" /> defines a radio button. Radio buttons allows an user to select only 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>
```

The output in a browser window will be like this:

- Male
- Female

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

The output in a browser window will be like this:

- I have a bike
- I have a car

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.

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

The output in a browser window will be like this:

Username: [ ] Submit

Password Field:

```
<input type=''password" /> defines a password field:
<form>
Password: <input type="password" name="pwd" />
</form>
```

The output in a browser window will be like this:

Password:

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

Text Area: When the single line approach of a regular text input field is not 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 6.4:

```
<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 which the content is to be sent. 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 6.5: 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 <input type=''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:

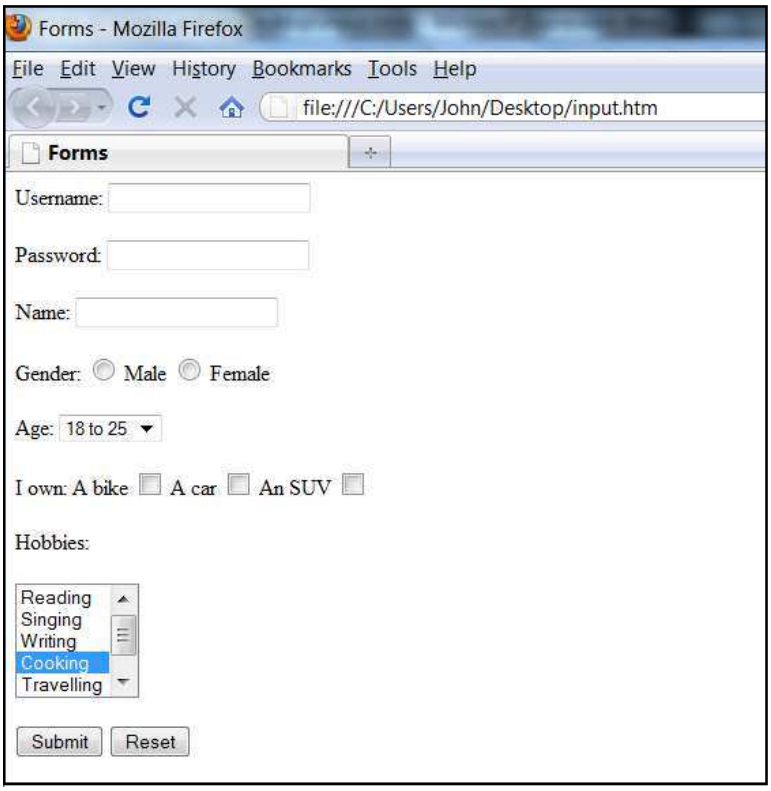


Figure 6.6: A Form example

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.



CHECK YOUR PROGRESS

Q.2: 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.



6.5 LET US SUM UP

- 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.



6.6 FURTHER READING

1) Brooks, D. R. (2017). *Functions. In Programming in HTML and PHP* (pp. 129-147). Springer, Cham.

2) Jackson, J. C. (2007). *Web Technologies: A Computer Science Perspective*. Pearson/Prentice Hall.

3) Willard, W. (2002). *HTML: A Beginner's Guide*. Osborne.

4) WWW.W3schools.com.

6.7 ANSWERS TO CHECK YOUR PROGRESS

Ans. to Q. No. 1: i) Frame; ii) <noframes>; iii) <iframe>; iv) <frameset cols="20%,30%,50%">

Ans. to Q. No. 2: i) form; ii) select; iii) input; iv) reset; v) Checkboxes; vi) method

6.8 MODEL QUESTIONS

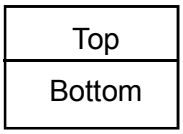
Q.1: What is the use of frames in HTML?

Q.2: Describe all the various tags associated with HTML frames.

Q.3: Giving an example to describe the working of HTML form.

Q.4: Write HTML frame code to divide the webpage in the following format:





Q.5: What is an IFrame in HTML

Q.6: What is Checkbox control in HTML form? Give the HTML code to add Checkbox.

Q.7: What is Radio Button control in HTML form? For what kind of input Radio Button are used?

Q.8: Write HTML code to insert multiline text input in a HTML form.

*** ***** ***

UNIT 7: CASCADING STYLE SHEET

UNIT STRUCTURE

7.1 Learning Objectives

7.2 Introduction

7.3 Cascading Style Sheet

- 7.3.1 Understanding Style Rules
- 7.3.2 The id and Class Selector
- 7.3.3 Different Types of CSS

7.4 Inline Style Sheet

7.5 Embedded Style Sheet

7.6 External Style Sheet

7.7 Imported Style Sheet

7.8 Creating a CSS file

7.9 Using CSS in a web page

7.10 Let Us Sum Up

7.11 Further Reading

7.12 Answers to Check Your Progress

7.13 Model Questions

7.1 LEARNING OBJECTIVES

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

- define Cascading Style Sheets
- describe Inline Style Sheets
- describe Embedded Style Sheets
- describe External Style Sheets
- describe Imported Style Sheets
- create CSS files
- use CSS in web pages.

7.2 INTRODUCTION

In the previous unit you have been introduced to HTML frames and forms.

In this unit, you will learn about Cascading Style Sheets and you be introduced to Inline Style Sheets, Embedded Style Sheets, External Style Sheets, and Imported Style Sheets. You will also learn to create CSS files and to use CSS in web pages.

7.3 CASCADING STYLE SHEETS (CSS)

CSS stands for Cascading Style Sheets. CSS define how HTML elements are to be displayed. CSS enables us to change the appearance and layout of all the pages in a web site, just by editing in one single location. CSS helps us save a lot of work since we have to specify the details only once for an element. Due to less coding, pages are loaded faster. Since we need to edit in only one place, the maintenance is easy. Also CSS has a much wider array of attributes than HTML, hence having superior styles to HTML.

7.3.1 Understanding Style Rules

A style rule in CSS is composed of two parts: a *selector* and *declaration*. The selector indicates the element to which the rule is applied. The declaration determines the property values of a selector.

The declaration is composed of a property and a value. The property specifies a characteristic, such as color, font-family, position, etc. and is followed by a colon (:). The value expresses specification of a property, such as red for color, arial for font family, 12 pt for font-size, etc. and followed by a semicolon (;).

For example, *th{color:red;}* and *p{font-family:arial}.* Here *th* and *p* are selectors and color:red and font-family:arial are declarations where color and font-family are properties and red and arial are values.

A CSS declaration always ends with a semicolon and declaration groups are surrounded by curly brackets. To make the CSS more readable we can put one declaration on each line. For example,

```
p{
    text-align:center;
    color:black;
    font-family:arial;
}
```

7.3.2 The id and Class Selector

In addition to setting a style for a HTML element, CSS allows us to specify our own selectors called “id” and “class”.

The id selector is used to specify a style of a single, unique element. The id selector uses the id attribute of the HTML element and is defined with a “#”. The style rule below will be applied to the element with id=”para1". Example,

```
#para1
{
    text-align:center;
    color:red;
}
```

The class selector is used to specify a style or a group of elements. Unlike the id selector, the class selector is most often used on several elements. This allows us to set a particular style for any HTML elements with the same class. The class selector uses the HTML class attribute and is defined with a “.”.

In the example below, all HTML elements with class=“center” will be center aligned. Example,

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

In the example below, all p elements with class=‘‘center” will be center aligned. Example,

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

7.3.3 Different Types of CSS

There are three types of CSS.

1) Inline Style Sheet
2) Embedded Style Sheet
3) External Style Sheet



CHECK YOUR PROGRESS

Q.1: What does CSS stands for?

.....

Q.2: Name the two parts of a style rule in CSS.

.....

.....

Q.3: Name the different types of CSS.

.....

7.4 INLINE STYLE SHEET

An inline style sheet is used when a style is to be applied to the current element of a HTML document. Here, we apply the style to the instance to which we want to apply the style. Example,

```
<html>
   <head>
   </head>
   <body>
      <p style=''color:red;margin-left:20px”>
      This is a
      paragraph.</p>
      <p>This is an another paragraph</p>
</body>
</html>
```

As we can see in the above example, the style is applied to the first *<p>* element with the property *color* having value *red* and *margin-left* having

value *20px*. This style is applicable only to the current instance of the element <*p*>. This style is not applicable to the second <*p*> element.

7.5 EMBEDDED STYLE SHEET

An embedded style sheet is used when a single document has a unique style. It embeds style sheet information into a HTML document using the <*style*> element inside the head section. It is also known as Internal Style Sheet. Example,

```
<html>
   <head>
      <style type=''text/css”>
         hr {color:red;}
         p {margin-left:20px;}
         body {background-image:url(“images/
         back40.gif”);}
      </style>
   </head>
   <body>
      <p>This is a paragraph.</p>
      <p>This is an another paragraph.</p>
   </body>
</html>
```

As we can see in the above example, the style sheet information is embedded into the HTML file using the <*style*> element. Here the style is applied to the whole document and so for both the instances of <*p*> element, the style is visible.

7.6 EXTERNAL STYLE SHEET

An external style sheet is used when the style is applied to many pages. With an external style sheet, we can change the look of an entire website by changing one file. Each page must link to the style sheet using the <*link*> tag. The <*link*> tag goes inside the head section.

```
<html>
   <head>
      <link rel=‘‘stylesheet” type=‘‘text/css”
      href=‘‘mystyle.css”/>
   </head>
   <body>
      <p>This is a paragraph.</p>
      <p>This is an another paragraph.</p>
   </body>
</html>
```

As we can see in the above example, an external style sheet “*mystyle.css”* is applied by using the <link> tag. When we do any type of change in the file “mystyle.css” or change the file name inside the <link> tag, we can see the change in style in the whole web page.

7.7 IMPORTED STYLE SHEET

An imported style sheet is used when we want to import a style sheet somewhere into a HTML page or into another style sheet.

We can import style sheet information into a HTML document using the *<style>* element with *@import* rule inside the head section.

```
<html>
   <head>
   <style>
      @import url(http://www.yoursite/stylesheets/
      mystyle.css);
   </style>
   </head>
   <body>
      <p>This is a paragraph.</p>
      <p>This is an another paragraph.</p>
   </body>
</html>
```

As we can see in the above example, the *@import* rule is used inside the *<style>* element which imports the style sheet from the given url.

We can also import style into the current style sheet by using the *@import* style at the top of the CSS document.

```
@import "navigation.css";
```

or

```
@import url("navigation.css");
```

7.8 CREATING A CSS FILE

A cascading style sheet (CSS) file can be created using any text editor. Firstly the file should not contain any html tags. Then start writing the style sheet by using any selector and its declaration (comprising a property and its value). The style sheet should be saved with a .css extension. An example of the contents of a style sheet file is shown below.

```
hr {color:red;}
p {margin-left:20px;}
body {background-image:url("images/back40.gif");}
#para1
{
    text-align:right;
    color:blue;
}
.center
{
    text-align:center;
}
```

7.9 USING CSS IN A WEB PAGE

A cascading style sheet (CSS) can be used in a web page by inserting the CSS in the web page by using the different style sheets i.e. Inline Style Sheet, Embedded Style Sheet, External Style Sheet and Embedded Style Sheet.

Following are some properties of a webpage which can is used by CSS.

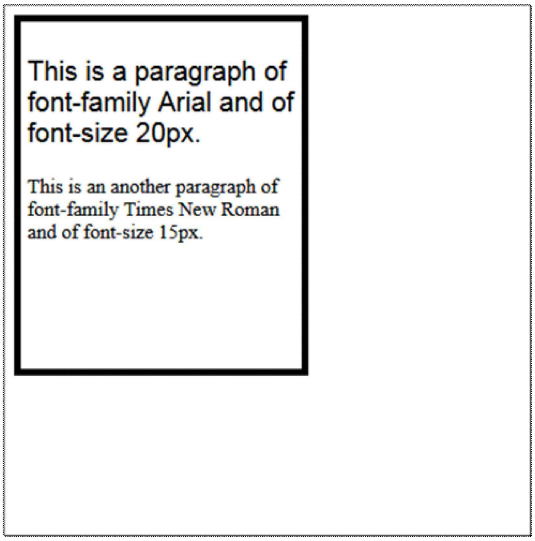
	Properties
Background	background-attachment, background-color, background-image, background-position, background-repeat
Border	border-bottom-color, border-bottom-style, border-bottom-width, border-color, border-left-color, border-left-style, border-left-width, border-right-color, border-right-style, border-right-width, border-style, border-top-color, border-top-style, border-top-width
Classification	clear, cursor, display, float, position, visibility
Dimension	height, line-height, max-height, min-height, min-width, width
Font	font-family, font-size, font-size-adjust, font-stretch, font-style, font-variant, font-weight
List and Marker	list-style-image, list-style-position, list-style-type, marker-offset
Margin	margin-bottom, margin-left, margin-right, margin-top
Outlines	outline-color, outline-style, outline-width
Padding	padding-bottom, padding-left, padding-right, padding-top
Positioning	bottom, clip, left, overflow, position, right, top, vertical-align, z-index
Table	border-collapse, border-spacing, caption-side, empty-cells, table-layout
Text	color, direction, line-height, letter-spacing, text-align, text-decoration, text-indent, text-shadow, text-transform, unicode-bidi, white-space, word-spacing

Let us use some of the properties to understand how CSS can be used in a web page. Let us create an html file using any available editor

tool. Let us say we use Notepad in windows to create the html file. We use inline cascading style sheet as the cascading style and hence the style is applied to the each current element of the HTML document. Here we have created an element which contains two <p> elements.

```
<html>
   <head>
   </head>
   <body>
      <div style=''border-style:solid;border-
      color:black;border-
      width:thick;width:200px;height:250px;
      padding:5px”>
         <p style=''font-family:arial;font-
         size:20px”>This is a paragraph of font-
         family Arial and of font-size 20px.</p>
         <p style=''font-family:times;font-
         size:15px”>This is an another paragraph
         of font-family Times New Roman and of font-
         size 15px.</p>
      </div>
   </body>
</html>
```

The output for the above written code for the html file is given below.



We can also use Embedded, Internal or External Style Sheet as explained earlier by using the different properties of a web page.



CHECK YOUR PROGRESS

Q.4: In case of Embedded Cascading Style Sheet which element is used to embed the style?

.....

Q.5: Which tag is used for an External Style sheet?

.....



7.10 LET US SUM UP

- CSS stands for Cascading Style Sheets.
- CSS enables us to change the appearance and layout of all the pages in a web site.
- A style rule in CSS is composed of two parts: a selector and declaration.
- The declaration is composed of a property and a value.
- A CSS declaration always ends with a semicolon and declaration groups are surrounded by curly brackets.
- CSS allows us to specify our own selectors called “id” and “class”.
- The id selector is used to specify a style of a single, unique element.
- The class selector is used to specify a style or a group of elements.
- There are four ways to insert CSS are Inline Style Sheet, Embedded Style Sheet, External Style Sheet and Imported Style Sheet.
- An inline style sheet is used when a style is to be applied to the current element.
- An embedded style sheet is used when a single document has a unique style.
- An external style sheet is used when the style is applied to many pages.

- An imported style sheet is used when we want to import a style sheet somewhere into a HTML page or into another style sheet.
- A style sheet should be saved with a .css extension.



7.11 FURTHER READING

1) Grannell, Craig. (2008). *The Essential Guide to CSS and HTML Web Design*. Apress.
2) Meyer, Eric A. (2010). *Smashing CSS: Professional Techniques for Modern Layout*. Wiley.
3) Meyer, Eric A. (2011). *CSS Pocket Reference: Visual Presentation for the Web*. O'Reilly Media.



7.12 ANSWERS TO CHECK YOUR PROGRESS

Ans. to Q. No. 1: Cascading Style Sheet.

Ans. to Q. No. 2: Selector and Declaration.

Ans. to Q. No. 3: Inline Style Sheet, Embedded Style Sheet and External Style Sheet.

Ans. to Q. No. 4: <style>

Ans. to Q. No. 5: <link>



7.13 MODEL QUESTIONS

Q.1: What is CSS?

Q.2: How do you use a CSS selector in a style rule?

Q.3: How will you write a CSS declaration in a style rule?

Q.4: What is an id and what is a class selector?

Q.5: What are the different types of CSS?

Q.6: What is an inline style sheet?

Q.7: What is an embedded style sheet?

Q.8: What are the uses of embedded style sheet?

Q.9: Why is the external style sheet useful?

Q.10: What is an imported style sheet?

*** ***** ***

UNIT 8 : DYNAMIC HTML

UNIT STRUCTURE

8.1 Learning Objectives

8.2 Introduction

8.3 Dynamic HTML

8.4 Document Object Model

8.5 Features of DHTML

8.6 Cascading Style Sheet Positioning

8.7 JavaScript Assisted Style Sheet

8.8 The ID Attribute

8.9 DHTML Events

8.10 Let Us Sum Up

8.11 Further Reading

8.12 Answers to Check Your Progress

8.13 Model Questions

8.1 LEARNING OBJECTIVES

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

- learn about Dynamic HTML
- use DHTML in web-pages
- learn the concept of Document Object Model
- learn the various features of DHTML
- learn how to position Cascading Style Sheets
- describe what are JavaScript assisted Style Sheets
- use the ID attribute
- learn what are DHTML events.

4.2 INTRODUCTION

In the previous units, you were introduced to various HTML tags and also their different attributes having their own individual functions which can be used in designing web-pages Style Sheets that offer a very flexible and fast way to designing web-pages also featured in the previous unit. Style

Sheets also allow us to control the style and layout of multiple web pages all at once without much effort.

In this unit, we will discuss Dynamic HTML which enables us to dynamically change the rendering and content of a web page as the users interact with it. We will learn to create dynamic web pages to improve the user interactivity.

8.3 DYNAMIC HTML

Dynamic HTML (DHTML) is an umbrella term for a collection of technologies used together to create interactive and animated web sites by using a combination of a static markup language such as HTML, a client-side scripting language such as JavaScript, a presentation definition language such as CSS, and the Document Object Mode.

DHTML allows scripting languages to change variables in a web page's definition language, which in turn affects the look and function of static HTML page content, after the page has been fully loaded and during the viewing process. Thus, the dynamic characteristic of DHTML is the way it functions while a page is viewed. DHTML is a browser feature or enhancement that gives the browser the ability to be dynamic. Dynamic HTML is not really a new specification of HTML, but rather a new way of looking at and controlling the standard HTML codes and commands. When thinking of dynamic HTML, we need to remember the qualities of standard HTML, especially that once a page is loaded from the server, it will not change until another request comes to the server. Dynamic HTML gives more control over the HTML elements and allows them to change at any time, without returning to the Web server.

With DHTML, we can easily add effects to our web pages. For example, we can:

- Hide web-content until a given time elapses or the user interacts with the page.
- Animate text and images, independently moving each element from any starting point to any ending point, following a predetermined path or one chosen by the user.

- Embed a ticker that automatically refreshes its content with the latest news, stock quotes, or other data.
- Use a form to capture user input, and then instantly process and respond to that data.

DHTML achieves these effects by modifying the in-memory representation of the current document and automatically reformatting it to show changes. It does not reload the document, load a new document, or require a distant server to generate new content. Instead, it uses the user's computer to calculate and carry out changes. This means a user does not have to wait for text and data to complete the time-consuming roundtrips to and from a server before seeing the results. Furthermore, DHTML does not require additional support from applications or embedded controls to make changes. Typically, DHTML documents are self-contained, using styles and a script to process user input and directly manipulate the HTML elements, attributes, styles, and text of the document.

In short, DHTML eliminates the shortcomings of static pages. We can create innovative Web sites, on the Internet or on an Intranet, without having to sacrifice performance for interactivity. Not only does DHTML enhance the user's perception of web-documents, it also improves server performance by reducing requests to the server.

8.4 DOCUMENT OBJECT MODEL

The *Document Object Model (DOM)* is a cross-platform and language-independent convention for representing and interacting with objects in HTML, XHTML and XML documents. Aspects of the DOM may be addressed and manipulated within the syntax of the programming language in use. The public interface of a DOM is specified in its *application programming interface* (API). It defines the logical structure of documents and the way a document is accessed and manipulated. With the Document Object Model, programmers can build documents, navigate their structure, and add, modify, or delete elements and content. Anything found in an HTML document can be accessed, changed, deleted, or added using the Document Object Model. One important objective for the Document Object

Model is to provide a standard programming interface that can be used in a wide variety of environments and applications. The DOM allows access to any part of a Web page to change it with DHTML. Every part of a Web page is specified by the DOM and using its consistent naming conventions we can access them and change their properties. To become proficient in DHTML, we must familiarize ourselves with the DOM which is simply a framework containing objects displayed by a browser, such as forms, images, layers, etc. Developers can employ this hierarchical structure to access and manipulate what appears onscreen.

DOM is the foundation of DHTML, providing a structured interface that allows us to access and manipulate virtually anything within the document. The HTML elements in the document are available as individual objects, meaning that we can examine and modify an element and its attributes by reading and setting properties and by calling methods. The text between elements is also available through DOM properties and methods.

DOM Nodes: According to the DOM, everything in an HTML document is a node. The DOM says:

- The entire document is a document node
- Every HTML element is an element node
- The text in the HTML elements are text nodes
- Every HTML attribute is an attribute node
- Comments are comment nodes

Let us consider the following HTML document:

Example 8.1:

```
<html>
<head>
<title>DOM Tutorial</title>
</head>
<body>
<h1>DOM Lesson one</h1>
<p>Hello world!</p>
</body>
</html>
```

In the above code, the root node is <*html*>. All other nodes in the document are contained within <*html*>.

The <*html*> node has two child nodes: <*head*> and <*body*>.

The <*head*> node holds a <*title*> node. The <*body*> node holds a <*h1*> and <*p*> node.

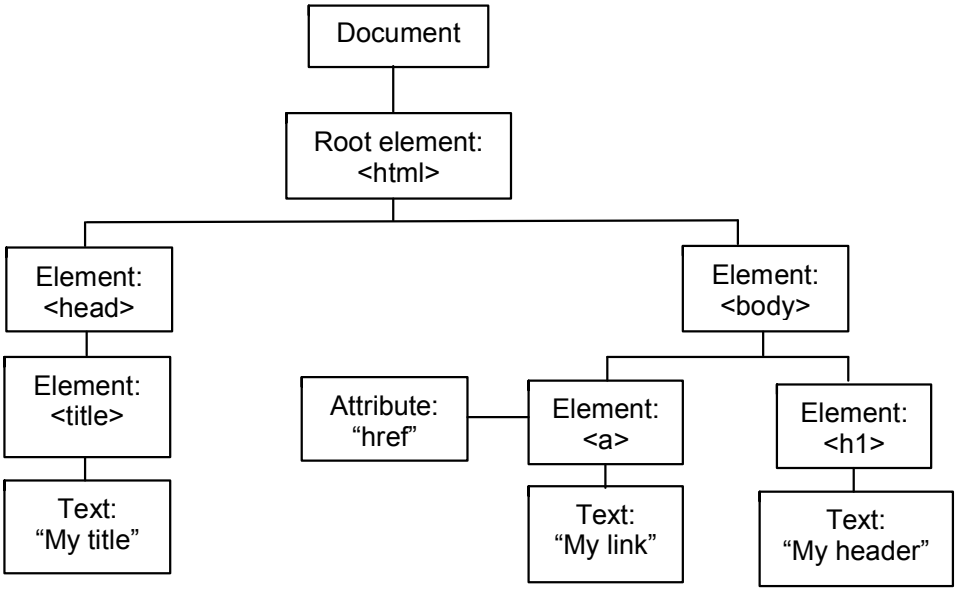
A common error in DOM processing is to expect an element node to contain text. However, the text of an element node is stored in a text node.

In this example: <*title*>*DOM Tutorial*</*title*>, the element node <*title*>, holds a text node with the value "DOM Tutorial".

"DOM Tutorial" is not the value of the <*title*> element!

The HTML DOM Node Tree: The HTML DOM views a HTML document as a *tree-structure*. The tree structure is called a *node-tree*. All nodes can be accessed through the tree. Their contents can be modified or deleted, and new elements can be created.

The node tree below shows the set of nodes, and the connections between them. The tree starts at the root node and branches out to the text nodes at the lowest level of the tree:



Node Parents, Children, and Siblings: The nodes in the node tree have a hierarchical relationship to each other. The terms parent, child, and sibling are used to describe the relationships. Parent nodes have children. Children on the same level are called *siblings* (brothers or sisters).

- In a node tree, the top node is called the **root**
- Every node, except the root, has exactly one **parent node**
- A node can have any number of **children**
- A **leaf** is a node with no children
- **Siblings** are nodes with the same parent

In the DOM, HTML documents consist of a set of node objects that can be accessed with JavaScript or other programming languages. The programming interface of the DOM is defined by standard properties and methods.

Properties are often referred to as something that is (i.e. the name of a node). Some DOM properties are:

- x.innerHTML – the text value of x
- x.nodeName – the name of x
- x.nodeValue – the value of x
- x.parentNode – the parent node of x
- x.childNodes – the child nodes of x
- x.attributes – the attributes nodes of x

where x is a node object (HTML element).

Methods are often referred to as something that is done (i.e. remove a node). Some DOM methods are:

- x.getElementById(id) – get the element with a specified id
- x.getElementsByTagName(name) – get all elements with a specified tag name
- x.appendChild(node) – insert a child node to x
- x.removeChild(node) – remove a child node from x

where x is a node object (HTML element).

The innerHTML Property: The easiest way to get or modify the content of an element is by using the *innerHTML* property which is supported by all major browsers. The *innerHTML* property is useful for returning or replacing the content of HTML elements (including <*html*> and <*body*>), it can also be used to view the source of a page that has been dynamically modified.

Example 8.2: The following code gets the *innerHTML* (text) from the <*p*> element with id="intro":

```
<html>
<body>
<p id=''intro”>Hello World!</p>
<script type=''text/javascript”>
txt=document.getElementById(“intro”).innerHTML;
document.write(“<p>The text from the intro
paragraph: “+ txt’’ + “</p>”);
</script>
</body>
</html>
```

In the Example 8.2, *getElementById* is a method, while *innerHTML* is a property.

Accessing Nodes: With the DOM, we can access every node in an HTML document in three ways.

1) By using the **getElementById()** method which returns the element with the specified ID.
2) By using the **getElementsByTagName()** method which returns all elements with a specified tag name.
3) By navigating the node tree, using the node relationships.

getElementById() : This method returns the element with the specified ID.

For example: `document.getElementById(“intro”);`

getElementsByTagName() : This method returns all elements with a specified tag name.

For example: `document.getElementsByTagName(“p”);`

Navigating Node Relationships: The three properties; parentNode, firstChild, and lastChild, follow the document structure and allow short-distance travel in a document.

```
<html>
<body>
<p>Hello World!</p>
<div>
<p>The DOM is very useful!</p>
```

```
<p>This example demonstrates node relationships.</p>
</div>
</body>
</html>
```

In the above code, the first *p* element is the first child node (firstChild) of the body element, and the *div* element is the last child node (lastChild) of the body element. The parent node (parentNode) of the first *p* element and the *div* element is the *body* element, and the parent node of the *p* elements inside the *div* element, is the *div* element.

The firstChild property can also be used to access the text of an element. This is shown in Example 8.3.

Example 8.3:

```
<html>
<body>
<p id=''intro”>Hello World!</p>
<script type=''text/javascript”>
x=document.getElementById(“intro”);
document.write(x.firstChild.nodeValue);
</script>
</body>
</html>
```

8.5 FEATURES OF DHTML

There are four primary features of DHTM. They are:

1) Changing the tags and properties
2) Real-time positioning
3) Dynamic fonts (Netscape Communicator)
4) Data binding (Internet Explorer)

1) Changing the tags and properties: This is one of the most common uses of DHTML. It allows us to change the qualities of an HTML tag depending on an event outside of the browser (such as a mouse click, time, or date, and so on). We can use this to preload information onto a page, and not display it unless the reader clicks on a specific

link. Such events that trigger other changes to occur are discussed at length later in this unit.

2) **Real time positioning:** When most people think of DHTML this is what they expect– objects, images, and text moving around the Web page. This can allow us to play interactive games with our readers or animate portions of our screen. While straight HTML will allow us some control of the position of elements through tables and using the non-breaking space ( ), we can't position things very precisely. With Cascading Style Sheets (CSS) we can place an item on a page at exact x and y coordinates. We will look into this feature in the next section in more detail.

3) **Dynamic fonts** (Netscape Communicator)**:** This is a Netscape only feature. Netscape developed this to get around the problem designers had with not knowing what fonts would be on a reader's system. With dynamic fonts, the fonts are encoded and downloaded with the page, so that the page always looks as the designer intended it to be. One of the most common uses of graphics on Web pages is to create titles with the specific fonts we want displayed on our Web site. The problem is, when we use a font tag or the font-family CSS property, we have to hope that the readers have the fonts that we specify on their computer.

   For example, a company might have a font developed for them for use as their official typeface, say "Company Font A". In order to get the most return on the investment of creating a new font, the company wants the Web site to be displayed using that font. The Web developers could use the CSS property:

   font-family: Company Font A;

   But this would only work on those computers that had that font installed. In other words, only corporate computers would see the font. Using dynamic fonts or embedded fonts, a Web developer can use any font they wish to (Note: not all fonts can be embedded or used as dynamic fonts) on their Web pages and have them embedded so that their readers can view them as well. What this

basically means is that the font definition is downloaded when the page downloads.

4) **Data Binding:** Data binding is a DHTML feature that lets us easily bind individual elements in our document to data from another source, such as a database or comma-delimited text file. When the document is loaded, the data is automatically retrieved from the source and formatted and displayed within the element. One practical way to use data binding is to automatically and dynamically generate tables in our document. This can be done by binding a table element to a data source. When the document is viewed, a new row is created in the table for each record retrieved from the source, and the cells of each row are filled with text and data from the fields of the record. Because this generation is dynamic, the user can view the page while new rows are created in the table. Additionally, once all the table data is present, we can manipulate (sort or filter) the data without requiring the server to send additional data. The table is simply regenerated, using the previously retrieved data to fill the new rows and cells of the table.

   Another practical use of data binding is to bind one or more elements in the document to specific fields of a given record. When the page is viewed, the elements are filled with text and data from the fields in that record, sometimes called the “current” record. A simple example is a form letter in which the name, e-mail address, and other details about an individual are filled from a database. To adapt the letter for a given individual, we simply specify which record should be the current record. No other changes to the letter are needed.

   Yet another practical use is to bind the fields in a form to fields in a record. Not only can the user view the content of the record, but the user can also change that content by changing the settings and values of the form. The user can then submit these changes so that the new data is uploaded to the source– for example, to the HTTP server or database.



CHECK YOUR PROGRESS

Q.1: Fill in the blanks:

i) ..... enables us to dynamically change the content of a Web page.

ii) DHTML eliminates the shortcomings of ..... pages.

iii) ..... is simply a framework containing ..... displayed by a browser.

iv) The ..... of an element node is stored in a text .....

v) The HTML DOM views a HTML document as a .....

vi) The easiest way to get or modify the content of an element is by using the ..... property.

vii) The ..... method returns the element with the specified ID.

8.6 CASCADING STYLE SHEET POSITIONING

We have already introduced Casecading Style Sheet(CSS) in the previous unit. Using the features of Cascading Style Sheet (CSS), we can often make web site much more attractive, and at the same time, enforce style rules that help to unify the entire appearance of the web site across all its pages.

CSS offers various kinds of benefits. For example, a single style sheet can cascade its effects through all the pages in a web site. Suppose, we want to change our site's default body font from "Arial" to "Times New Roman", we need to make that change only once within the style sheet file, rather than hunting down all the attributes throughout the entire set of HTML code files that make up the site. Another benefit of using CSS is that the style sheet only needs to be downloaded once to the user's computer. Thereafter, it is called up from a local cache, resulting in smaller HTML

pages. The web pages containing CSS load faster into the user's browser hence allowing faster web access and navigation.

Positioning is the ability to place an HTML element at a specific point on a page, relative to another element or the browser window itself. While using CSS positioning, the first thing to be done is to use the CSS property position to tell the browser if we are going to use absolute or relative positioning. Then by assigning these coordinates, we can place elements such as images, controls, and text exactly where needed. We can also assign a z-index to define the order in which overlapping elements are stacked on top of one another. There are four basic types of CSS positioning. They are:

1) Static positioning
2) Absolute positioning
3) Relative positioning
4) Fixed positioning

The CSS positioning properties allow us to specify the *left*, *right*, *top*, and *bottom* position of an element. It also allows us to set the shape of an element, place an element behind another, and to specify what should happen when an element's content is too big to fit in a specified area.

1) Static positioning: Static is the default position for any element on a Web page. If we do not define the position, it will be static and will display on the screen based on where it is in the HTML document and how it would display inside the normal flow of the document.

If we apply positioning rules like top or left to an element that has a static position, those rules will be ignored.

2) Absolute positioning: Absolute positioning is the easiest to understand.

The CSS position property:

```
position: absolute;
```

tells the browser that whatever is going to be positioned should be removed from the normal flow of the document and will be placed in an exact location on the page. It is also taken out of the normal flow of the document. It will not affect how the other elements before it or after it in the HTML are positioned on the Web page.

If we want to position an element 10ems from the top of the document window, we would use the “top” offset to position it there with absolute positioning:

```
position: absolute;
top: 10em;
```

This element will then always display 10em from the top of the page, no matter what else is displayed there in normal flow. Absolute positioning takes its starting point from the upper left corner of the browser pane, this is coordinate 0,0. One thing to remember is that once we use absolute positioning, this object does not affect any other object in the flow. Absolute positioning allows us to position an object precisely where we want it to be. This gives a huge amount of control over our Web site. Even if the browser window is smaller than the page requires.

Example 8.4: Absolute positioning example:

```
<html>
<head>
<title>Absolute Positioning</title>
<style type=”text/css”>
h2
{
   position:absolute;
   left:100px;
   top:150px;
}
</style>
</head>
<body>
<h2>This is a heading with an absolute position</h2>
<p>With absolute positioning, an element can be placed
anywhere on a page. The heading below is placed 100px
from the left of the page and 150px from the top of
the page.</p>
```

```
</body>
</html>
```

The above code gives the following output in Mozilla Firefox.



Figure 8.1: Absolute Positioning

3) **Relative positioning:** Relative positioning uses the same four positioning properties as absolute positioning. But instead of basing the position of the element upon the browser view port, it starts from where the element would be if it were still in the normal flow.

For example, if we have three paragraphs and the third has a position: relative style placed on it, its position will be offset based on its current location – not from the original sides of the view port.

```
<p>Paragraph 1.</p>
<p>Paragraph 2.</p>
<p style=''position: relative;left: 2em;”>Paragraph
3.</p>
```

In the above code, the third paragraph will be positioned 2em from the left side of the container element, but will still be below the first two paragraphs. It would remain in the normal flow of the document, and just be offset slightly. If we change it to position: absolute; anything following it would display on top of it, because it would no longer be in the normal flow of the document.

Relative positioning allows us to place an object in relation to where it would normally have been positioned if only HTML were

the only position control. What this means is that we position an HTML element within the current flow of the document. It gives us less control over the actual layout of the page, but allows to position elements within the page itself. For example: If we place text on a page, and then place an image after it, the image will be positioned directly after the text (barring any alignment tags). If we add a relative position style tag to the image, we can place it further down the page, or even directly on top of the text.

Example 8.5: Relative positioning example:

```
<html>
<head>
<style type=‘‘text/css”>
h2.pos_left
{
   position:relative;
   left:-20px;
}
   h2.pos_right
{
   position:relative;
   left:20px;
}
</style>
</head>
<body>
<h2>This is a heading with no position</h2>
<h2 class=‘‘pos_left”>This heading is moved left
according to its normal position</h2>
<h2 class=‘‘pos_right”>This heading is moved right
according to its normal position</h2>
<p>Relative positioning moves an element RELATIVE
to its original position.</p>
```

```
<p>The style "left:-20px" subtracts 20 pixels from
the element's original left position.</p>
<p>The style "left:20px" adds 20 pixels to the
element's original left position.</p>
</body>
</html>
```

The above code gives the following output in Mozilla Firefox.



Figure 8.2: Relative Positioning

4) **Fixed positioning:** Fixed positioning is a lot like absolute positioning. The position of the element is calculated in the same way as the absolute model - from the sides of the view port. But fixed elements are then fixed in that location, like a watermark. Everything else on the page will then scroll past that element.

```
position: fixed;
```

When we fix an element in place on our site, it will print in that location when the Web page is printed out. For example, if the element is fixed at the top of the page, it will appear at the top of every printed page– because it is fixed to the top of the page. We can use media types to change how the printed pages display fixed elements:

```
@media screen {
h1#first { position: fixed; }
 }
@media print {
```

```
h1#first { position: static; }
}
```

Example 8.6: Fixed positioning example:

```
<html>
<head>
<title>Fixed Positioning</title>
<style type=‘‘text/css”>
p.pos_fixed
{
position:fixed;
top:30px;
right:5px;
}
</style>
</head>
<body>
<p class=‘‘pos_fixed”>Some more text</p>
<p><b>Note:</b> Internet Explorer 7 (and higher)
supports the fixed value if a !DOCTYPE is
specified.</p>
<p>Some text</p><p>Some text</p><p>Some text</
p><p>Some
text</p><p>Some   text</p><p>Some   text</
p><p>Some
text</p><p>Some   text</p><p>Some   text</
p><p>Some
text</p><p>Some   text</p><p>Some   text</
p><p>Some
text</p><p>Some   text</p><p>Some   text</
p><p>Some text</p>
</body>
</html>
```

The above code gives the following output in Mozilla Firefox.

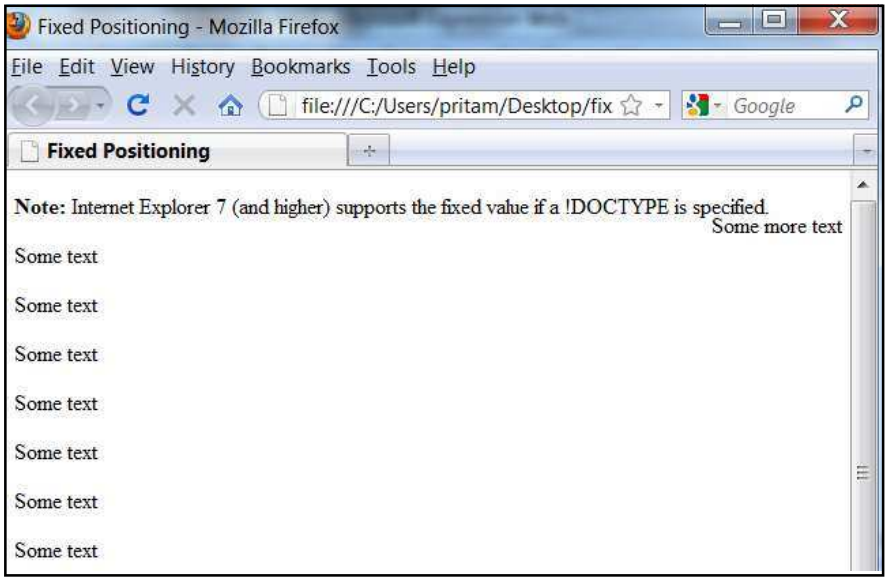


Figure 8.3: Fixed Positioning



CHECK YOUR PROGRESS

Q.2: Fill in the blanks:

i) There are ..... basic types of CSS positioning.

ii) There are three ways of inserting a .....

iii) The CSS ..... property sets an image as the background.

iv) The CSS syntax is made up of ..... parts.

v) CSS comments are ignored by a .....

vi) CSS margin properties define the ..... around elements.

8.7 JAVASCRIPT ASSISTED STYLE SHEET

In the previous section we have seen how we can add style to a web page using the various CSS positioning rules. In case we want to add style to our page elements we can use all the CSS styling rules in a number of ways. But if we want to change the format of page elements (e.g., change their color, show/hide them, animate them, etc.), we can use JavaScript to achieve these effects.

Dynamic Styles: Dynamic styles are a key feature of DHTML. By using CSS, we can quickly change the appearance and formatting of elements in a document without adding or removing elements. This helps keep the documents small and the scripts that manipulate the document fast. The object model provides programmatic access to styles. This means that we can change inline styles on individual elements and change style rules using simple script-based programming. When the color of an element is changed dynamically, it still fits into the same physical space in the document as it did before the color change. This is a relatively simple task for the browser to carry out. However, there are other styles, such as fontSize or fontFamily, that, when changed, actually change the size of the element and the amount of space required to display that element in the document.

Suppose we have the following HTML :

```
<p id=''introtext”>This is some introductory text</p>
```

We can very easily change the color of this text from within JavaScript. The first step is to access the element using the *getElementById()* method of the document object

```
var myElement = document.getElementById(“introtext”);
```

Having gained access to the element object we can access the *color* property of the *style* object for this element and change the color. The following example contains some text contained in a *<p>* tag assigned the *id* introtext. A button is created which calls the *buttonPressed()* function when clicked. The *buttonPressed()* function in turn accesses the element via the *id* and changes the color style to ‘green’.

```
myElement.style.color=''green”;
```

Let us put this together in an example in which we have a button configured to change the color of some text when it is pressed:

```
<html>
<head>
<title>JavaScript Radio Example</title>
<script type=''text/javascript”>
function buttonPressed()
```

```
{
var myElement=document.getElementById("introtext");
myElement.style.color=''green";
}
</script>
</head>
<body>
<p id=''introtext">''This is some introductory text.
The color of the text sould change when the button
is pressed"</p>
<form action=''name''=''orderForm">
<input type=''BUTTON" value=''Click to change text
color" onClick=''buttonPressed()">
</form>
</body>
</html>
```

Example 8.7: The following example demonstrates how the font size increases and changes its color when the user moves the mouse pointer over it.

```
<html>
<head>
<title>Dynamic Styles</title>
<script type=''text/javascript">
function doChanges(e)
 {
e.style.color = "green";
e.style.fontSize = "20px";
}
</script>
</head>
<body>
<h3 onmouseover=''doChanges(this)" style=''color:
black;font-size:18px">Welcome to Dynamic HTML!</h3>
```

```
<p>You can do the most amazing things with the
least bit of effort.</p>
</body>
</html>
```

The above example demonstrates the following:

- **HTML element:** In this case, the *H3* tag is the target element.
- **Inline style:** The element is initially displayed in black with a font size of 18px.
- **Event attribute:** The *onmouseover* attribute defines the action that occurs when the mouse pointer is moved over the element.
- **Event handler:** The function that responds to the event is declared in the head of the document. A DHTML object that represents the target element is passed as a function parameter using the this pointer.
- **The style object:** The style object contains the information that was set in the inline style when the element was defined. To change the color and font size, the function modifies the *color* and *font size* properties of the element. The browser immediately updates the onscreen text to display these new attribute values.

8.8 THE ID ATTRIBUTE

The ID attribute specifies a unique id for an HTML element and it must be unique within the HTML document. When we know a page will contain objects that we want to manipulate from script, we can uniquely identify an object by using its ID property. We can set the ID property by specifying the ID attribute within the tag itself. The attribute for an element usually has a corresponding property in the DHTML object model. For example, all the following <p> elements within the body of the page are specifically identified by using the ID attribute:

```
<html>
<body>
<center>
<h1>
```

This is the heading on a simple page.

```
</h1>
</center>
<p id=''para1''>
```
This is the first paragraph on a simple page.
```
</p>
<p id=''para2''>
```
This is the second paragraph on a simple page.
```
</p>
</body>
</html>
```

Syntax : `<element id=''value">`

The ID attribute performs several actions for Web pages:

- *A style sheet selector:* This is the function most people use the id attribute for. Because they are unique, we can be sure that we will be styling just the one item on our Web page when we style using an id property.
- *Named anchors for linking to:* Now, most modern Web browsers allow us to target precise locations in a Web document by pointing to the id at the end of the URL. We simply add the id to the end of the page URL, preceded by a pound-sign (#).
- *A reference for scripts:* If we write any DHTML, we will want to use the id attribute so as to make changes to the precise element on the page with our scripts.
- *A name for object elements:* This acts in the same way as for scripts, only we identify objects like Flash or Java.
- *Other processing:* The id allows us to process Web documents in whatever way needed. For example, we might extract the HTML into a database, and the id attribute identifies fields.

Rules for Using the ID Attribute: There are a few rules to be followed to have a valid document that uses the id attribute anywhere in the document:

- The id must start with a letter (a-z or A-Z)
- All subsequent characters can be letters, numbers (0-9), hyphens (–), underscores (_), colons (:), and periods (.).
- Each id must be unique within the document.

Using the ID Attribute: Once we have identified a unique element of our Web site, we can use style sheets to style just that one element:

```
<p id=''paragraph1''>This is my first paragraph.</p>
p#paragraph1 { background: #0cf; }
```

We can also link to that specific element without adding any <a> tags:

```
<a href=''#paragraph1''>link to the first paragraph</a>
```

Reference that paragraph in the scripts with the "*getElementById*" JavaScript method:

```
document.getElementById("paragraph1")
```

The id attribute can be used by a JavaScript (via the HTML DOM) or by CSS to make changes or style the element with the specified id.

Example 8.8: Use the id attribute to change a text with a JavaScript:

```
<html>
<head>
<script type=''text/javascript">
function change_header()
{
document.getElementById("myHeader").innerHTML=
''Nice day!";
}
</script>
</head>
<body>
<h1 id=''myHeader">Hello World!</h1>
<button onclick=''change_header()">Change text</
button>
</body>
</html>
```

8.9 DHTML EVENTS

DHTML events are nothing but handlers that can trigger actions in a browser. The Event object represents the state of an event, such as the

element in which the event occurred, the state of the keyboard keys, the location of the mouse, and the state of the mouse buttons.

Events are normally used in combination with functions, and the function will not be executed before the event occurs. Every element on an HTML page has events which can trigger a JavaScript. For example, we can use the onClick event of a button element to indicate that a function will run when a user clicks on the button. We define the events in the HTML tags.

Examples of events:

- A mouse click
- A web page or an image loading
- Mousing over a hot spot on the web page
- Selecting an input field in an HTML form
- Submitting an HTML form
- A keystroke

Some of the attributes that can be inserted into HTML tags to define event actions are:

onclick	Mouse clicks on object
ondblcick	Mouse double-clicks an object
onblur	Anelement loses focus
onabort	Loading of an element is interrupted
oncharge	The content of a field changes
onreset	The reset button is clicked
onselect	Text is selected
onload	A page or an image has finished loading
onunload	The user exits the page
onsubmit	The submit button is clicked
onresize	A window or frame is resized
onerror	An error occurs when loading a document or an image
onmousedown	A mouse button is pressed
onmousemove	The mouse is moved
onfocus	An element gets focus

Example 8.9: In the following example, the content of the *h1* element will change when a user clicks on it:

```
<html>
<body>
<h1 onclick=''this.innerHTML=''Ooops!''>Click on
this text</h1>
</body>
</html>
```

We can also add the script in the head section, and then call a function from the event handler as shown below:

```
<html>
<head>
<script type=''text/javascript”>
function changetext(id)
{
id.innerHTML=''Ooops!”;
}
</script>
</head>
<body>
<h1 onclick=''changetext(this)”>Click on this text</h1>
</body>
</html>
```

Example 8.10: This example asks for user input, and on clicking OK button redirects the user to www.kkhsou.org.

```
<html>
<head>
<script type=''text/javascript”>
function confirmInput()
{
fname=document.forms[0].fname.value;
alert(“Hello'' + fname + “! You will now be redirected to
www.kkhsou.org”);
}
</script>
```

```
</head>
<body>
<form onsubmit=''confirmInput()" action=''http://
www.kkhsou.org/">
Enter your name: <input id=''fname" type=''text"
size=''20">
<input type=''submit">
</form>
</body>
</html>
```

Example 8.11: In this example an alert box is triggered whenever the text box is clicked (gets focus).

```
<html>
<head>
<script type=''text/javascript">
function message()
{
alert("This alert box was triggered by the onfocus
event handler");
}
</script>
</head>
<body>
<form>
Enter your name: <input type=''text"
onfocus=''message()" size=''20''>
</form>
</body>
</html>
```



CHECK YOUR PROGRESS

Q.3: Fill in the blanks:

i) There are ..... rules for using the ID attribute.

ii) DHTML events are normally used in combinationwith .....

iii) Dynamic styles are a key feature of .....

iv) The ..... model provides programmatic access to styles.

v) ID allows us to ..... Web documents.



8.10 LET US SUM UP

- DHTML is the art of making ***dynamic*** and ***interactive*** web pages.
- DHTML is a ***browser feature*** or ***enhancement*** that gives the browser the ability to be dynamic.
- The DOM defines the logical structure of documents and the way a document is ***accessed*** and ***manipulated***.
- DOM provides a standard ***programming interface*** that can be used in a wide variety of environments and applications.
- In the DOM, HTML documents consist of a set of node objects that can be accessed with ***JavaScript*** or other programming languages.
- With ***Dynamic Fonts*** the font definition is downloaded when the page downloads.
- A single style sheet can ***cascade*** its effects through all the pages in a Web site.
- An ***external style sheet*** is ideal when the style is applied to many pages.
- ***CSS Positioning*** is the ability to place an HTML element at a specific point on a page, relative to another element or the browser window itself.

- HTML styles of a document can be dynamically changed using ***JavaScript***.
- Each ***ID*** attribute must be unique within the document.
- Every element on an HTML page has events which can ***trigger*** a JavaScript.



8.11 FURTHER READING

1) Godman, D. *Dynamic HTML– the Definitive References*. Shroff Publishers.
2) Jaworski, J. *Mastering JavaScript and Jscript*. BPB Publication.
3) WWW.W3schools.com.



8.12 ANSWERS TO CHECK YOUR PROGRESS

Ans. to Q. No. 1: i) DHTML; ii) static; iii) DOM, objects; iv) text, node; v) tree-structure; vi) innerHTML.; vii) getElementById()

Ans. to Q. No. 2: i) four; ii) stylesheet; iii) background-image; iv) three; v) browser; vi) space

Ans. to Q. No. 3: i) three; ii) functions; iii) DHTML; iv) object; v) process



8.13 MODEL QUESTIONS

Q.1: What is the Document Object Model? Describe its structure.

Q.2: What is a DOM node tree? Define Parents, Children and Siblings in this context.

Q.3: Describe briefly the DHTML features.

Q.4: Write a short note on Data Binding.

Q.5: What are style sheets? What are the advantages of using them?

Q.6: What are class selectors and ID selectors in CSS?

Q.7: Describe the different methods to insert a style sheet in an HTML document.

Q.8: How many types of CSS positioning are there? Explain with examples.

Q.9: What are JavaScript assisted style sheets and how can we use them? Give examples.

Q.10: Why should each ID attribute in an HTML document be unique?

Q.11: What are DHTML events? Give examples.

*** ***** ***

Name of the Paper: .....
Course Code: .....

Centre for Internal Quality Assurance (CIQA)
Krishna Kanta Handiqui State Open University
City Office: Housefed Complex, Guwahati-781006

Learner's Feedback on Course

Dear Learner,

Regarding the course as mentioned above, we would like to know your opinions and comments so as to improve the quality of self learning materials in future. Please respond to the following statements by ticking the number you feel most reflect your opinion. After completion of the additional comments, please detach the page and send/mail the same to us at the address given below.

The Director, Centre for Internal Quality Assurance, KKHandiqui State Open University
Housefed Complex, Dispur, Guwahati-781006
(E-mail id: ciqa@kkhsou.in)

1) Approximately how many hours did you spend for studying the units in the course? ☐☐☐

2) Please give your opinions (by ✓ mark) to the following items based on your reading of the block:

Sl. No.	Statements	Strongly Agree	Agree	Neither Agree nor Disagree	Disagree	Strongly Disagree
I)	The SLMs of the course fulfil the learning objectives set out in the beginning of each unit					
II)	The units of the SLM could satisfy my academic needs and expectations					
III)	The Unit writers have excellent knowledge about the course contents					
IV)	Language and contents of the units were presented at a level which I could readily understand					
V)	Ample opportunity for participation in the activities provided in the units					
VI)	Used enough Illustrations (Diagrams, tables etc.) for conceptual clarity					
VII)	Quality of content is engaging, relevant, and up-to-date					
VIII)	The self check questions are very helpful					
IX)	The Possible/model questions and the answers to check my progress have benefited me a lot					

Additional Comments: *(Please feel free to provide your open comments)*

1) Which aspects of the SLM, according to you, worked well?

.....

.....

2) What sort of changes/improvements do you feel KKHSOU could implement to improve the overall quality of the SLM?

.....

.....

Thank you for taking the time to complete this form.