KARPAGAM ACADEMY OF HIGHER EDUCATION
(Deemed to be University)
(Established Under Section 3 of UGC Act 1956)
Pollachi Main Road, Eachanari Post, Coimbatore - 641021
(For the candidates admitted from 2015 onwards)

DEPARTMENT OF COMMERCE (CA)

Semester VI

L	T	P	C
5	-	-	5

15CCU601 CORE - INTERNET AND WEB DESIGN

Course Objective:

This course design focuses on the structure of the website including the information architecture, the layout or the pages and the conceptual design with branding. HTML helps the students for developing dynamic web pages.

Learning Outcomes:

- To know the working of Internet, uses of search engines.
- Understand the fundamentals of HTML and use different formatting options
- To make the students expertise in creating Web Page.

UNIT I

Introduction to Internet - Resources of Internet – Hardware and Software Requirements of Internet. Internet Service Providers – Internet Services – Protocols – Concepts – Internet Client and Internet servers.

UNIT II

Introduction to HTML – Functions of HTML in Web Publishing – Basic Structural Elements and their Usage – Traditional Text and Formatting – Using Tables for Organisation and Layout – Forms – Frames and Frame sets - Style Sheets of Formatting - Advanced Layouts and Positioning with Style Sheets.

UNIT III

Using images with HTML – Merging Multimedia, Controls, and Plug-ins with HTML – Using the HTML Object Model and Creating Dynamic HTML Pages – Manipulating Objects and Responding to User Interaction.

UNIT IV

Scripting Basics – Client Side Image Maps – Introducing Java Script – Creating Simple Java Scripts – Using Java Scripts for Forms – Using Java Scripts with Style Sheets.

UNIT V

Introduction to ASP – Active Server Objects – Active Server Components – Cookies - Database Management with ASP. Emerging and Alternate Web Technologies – Active X Controls for the WWW - XML

Reference Books

1. Harley Hann, (2010), "*The Internet Complete Reference",* 2nd Edition, New Delhi, Tata McGraw-Hill Education Pvt. Ltd.
2. Shelly Powers, (2008). "*Dynamic Web Publishing Unleased*", 2nd Edition, Techmedia.
3. Scot Johnson et al, (2010), "*Using Active Server Pages",* Que.
4. Merser Dave, (2010). "*HTML"*, New Delhi: Tata McGraw Hill Publishing House.
5. Wyke R Allen, (2012), "*Java Script" Unleashed 2000*, New Delhi: Tech media.

Websites

1. www.webopedia.com
2. www.codecademy.com
3. www.w3schools.com/
4. www.javascriptkit.com
5. www.webteacher.com/javascript/
6. www.tizag.com/aspTutorial/
7. www.tutorialspoint.com/xml/

KARPAGAM ACADEMY OF HIGHER EDUCATION
(Deemed to be University)
(Established Under Section 3 of UGC Act 1956)
Pollachi Main Road, Eachanari Post, Coimbatore - 641021
(For the candidates admitted from 2015 onwards)
DEPARTMENT OF COMMERCE (CA)

SUBJECT : INTERNET AND WE DESIGN
SEMESTER : VI
SUBJECT CODE: 15CCU601 **CLASS : III B.COM CA**

LECTURE PLAN

S.NO	LECTURE DURATION (Hour)	TOPICS TO BE COVERED	SUPPORT MATERIALS
		UNIT I	
1	1	**Introduction**	R1: 2-3
2	1	What is Internet	R1: 3 - 4
3		Uses of Internet	R1:5
4	1	**Resources of Internet** Mail, The Web, Web Search Engine, Usenet, Gopher, Veronica and Jughead,	R1: 24 - 27
5	1	**Resources of Internet** Anonymous FTP Archie, Mailing Lists, Telnet, Talk Facilities, IRC, Mud	R1: 27 - 32
6	1	Hardware requirements	R1: 34 - 47
7	1	Software requirements	R1: 50 - 59
8	1	Internet Service Providers	W1
9	1	Internet Services	W1
10	1	Protocols & Concepts	W1, R1 : 20-24
11	1	Internet Client	R1 : 12 - 14
12	1	Internet servers	R1 : 15-17
13	1	**Recapitulation of Unit I**	
14	1	**Discussion of important Questions**	
		Total no. of Hours Planned for Unit - I	**14 Hrs**
		UNIT II	
1	1	Introduction to HTML	R2 :28-32
2	1	Functions of HTML in web publishing	R2 : 38-41

		➢ The Web's Defining Framework	
3	1	Basic structural elements and their usage	R2 : 44-55
4	1	Traditional text and formatting ➢ Text Layout,	R2 : 57 – 80
5	1	➢ Lists, Text Styles	
6	1	➢ The Font and Base font Elements	R2: 81 - 82
7	1	Using Tables for organisation and layout ➢ Table Elements	R2: 114- 120
8	1	➢ HTML Tables and CSSI Style Sheets	R2: 121 - 137
9	1	➢ HTML Table to create a Layout for Links	
10	1	Forms	R2 : 170- 189
11	1	Frames and Frame sets	R2 : 192-209
12	1	Style Sheets of formatting	R2: 86 - 110
13	1	Advanced layouts and positioning with style sheets ➢ CSS Positioning Attributes	R2 : 141 – 142
14	1	➢ Positioning Images and Other Elements	R2 : 143 -144
15	1	➢ Creating Page Columns	R2 : 145-152
16	1	**Recapitulation and discussion of important Questions**	
		Total no. of Hours Planned for Unit - II	**16Hrs**
UNIT III			
1	1	Using images with HTML ➢ Embedding Images within an HTML Document	W3 R2 : 214 - 224
2	1	➢ The Image Object and Accessing Images within Script	
3	1	Merging Multimedia, Controls, and Plug-Ins with HTML ➢ Introduction	R2 :234- 235
4	1	➢ Accessing Sound and Video in Web Pages	
5	1	➢ Using the <EMBED> Tag ➢ The <OBJECT> Tag	R2 : 235- 236
6	1	Using the HTML object Model and Creating Dynamic HTML Pages:	R2: 640- 651

		➢ Major Components	
7	1	➢ Scripting Object Models	
8	1	➢ New Microsoft Dynamic HTML Objects	R2 651- 647
9	1	➢ HTML Elements and the style Object	R2 : 648 - 660
10	1	➢ Netscape Scripting object Model	R2 : 661- 670
		➢ New Netscape dynamic Objects	
11	1	➢ Creating Cross-Browser Dynamic Pages	R2: 671 - 673
12	1	Manipulating Objects and Responding to User Interaction	
		➢ Introduction	R2 : 677 - 679
13	1	➢ Dynamic Positioning	
14	1	➢ Event Trapping and Assigning Event Handlers	R2 : 680 - 683
		➢ Event Bubbling	
15	1	**Recapitulation and discussion of important Questions**	
		Total no. of Hours Planned for Unit -III	**15 Hrs**
UNIT IV			
1	1	Scripting basics	R2 : 265 - 267
		➢ What is client side scripting	
2	1	➢ Script and Programs	R2 : 268 - 273
		➢ Client side scripting languages	
3	1	Client Side Image Maps	R2: 282 - 286
		➢ Server side Image Maps / Client side Image Maps	
4	1	➢ Making Client side Image Maps	R2: 286 - 298
5	1	➢ WYSIWYG image map editors,	
		➢ Enhancing client side image maps with scripting	
6	1	Introducing Java Script	R2 : 302- 304
		➢ What is Java Script, Statements	
7	1	➢ Blocks & Comments ,Data types, Variables, Expressions,	R2 : 305- 322
8	1	➢ Flow-control, Arrays, Functions	

9	1	Creating Simple Java Scripts	R2 : 325- 328
10	1	➢ Formatting Scripts	
		➢ Objects, Linking Scripts Windows Events	R2 : 332 - 337
11	1	Using Java scripts for forms	R2 : 348 - 358
		➢ Form Objects, Elements, Element Properties,	
12	1	Element Methods, Element Event Handlers, Form Validation	R2 : 358 - 371
13	1	Using Java Scripts with Style Sheets	R2 : 375- 376
		➢ Dynamic Style Sheets	
14	1	➢ Collections	R2 : 384
15	1	**Recapitulation and discussion of important Questions**	
		Total no. of Hours Planned for Unit - IV	**15 Hrs**
UNIT V			
1	1	Introduction to ASP	R3 : 9- 29,W6
2	1	Active server objects	R3 : 192 - 193
3	1	➢ Response & Request Objects	
4	1	Active server components	R3:342- 345,W6
5	1	➢ Browser capabilities	
6	1	➢ Other ASP components	W6
		➢ Sending & Receiving E-mail with ASP.	
7	1	Cookies	W6
8	1	Database Management with ASP	R3 : 429- 469
9	1	Emerging and alternate web technologies	R3 : 721 – 737,W6
10	1	Active X controls for the WWW	
11	1	Introduction to XML	R3 : 744 – 754, W7
12	1	➢ Features of XML	
13	1	**Recapitulation and discussion of important Questions**	
		Total no. of Hours Planned for Unit - V	**13 Hrs**
14	1	**Previous Year ESE Questions Discussion**	

		Previous Year ESE Questions Discussion	
15	1	**Previous Year ESE Questions Discussion**	
		Total no. of Hours Planned for Unit - V	**15 Hrs**
Total Planned Hours			**75 HRS**

SUPPORT MATERIAL

Reference Books

R1: Harley Hann, (2010), "*The Internet Complete Reference",* 2nd Edition, New Delhi, Tata McGraw-Hill Education Pvt. Ltd.

R2: Shelly Powers, (2008). "*Dynamic Web Publishing Unleased*", 2nd Edition, Techmedia.

R3: Scot Johnson et al, (2010), "*Using Active Server Pages",* Que.

R4: Merser Dave, (2010). "*HTML"* , New Delhi: Tata McGraw Hill Publishing House.

R5: Wyke R Allen, (2012), "*Java Script" Unleashed 2000*. New Delhi: Tech media.

Websites

W1: http://www.webopedia.com
W2:www.codecademy.com
W3:www.w3schools.com/
W4:www.javascriptkit.com
W5:http://www.webteacher.com/javascript/
W6: http://www.tizag.com/aspTutorial/
W7: http://www.tutorialspoint.com/xml/

UNIT I

SYLLABUS

Introduction to Internet - Resources of Internet – Hardware and Software Requirements of Internet. Internet Service Providers – Internet Services – Protocols – Concepts – Internet Client and Internet servers.

INTRODUCTION TO INTERNET

- Internet can be defined as an electronic medium, which is connecting the world with the help of the computers.
- Internet consists of large amount of data that can be accessed by the various users – and because of this it is also referred to as the 'Information Superhighway' of the world.
- With the help of the internet can easily be in touch with anyone in the whole world by sending electronic mail, by chatting etc., travel bookings can be made very easily, one can order books or buy anything online – in simple terms it can be said that internet provides a very strong connection or network between computers globally, bringing people and their working close to each other.
- The network of networks – 'The Internet' is not a personal property of any one i.e. it is not owned by anyone, which allows individuals and the various organizations to get connected to any other server or any other user.
- Internet has become such an important and defining tool in todays competitive and market oriented environment that it helps a lot in getting business and making money.

History & Development of the Internet:

- The Internet has its origins in the Cold War. In 1957, the Soviet Union launched the first satellite, Sputnik.
- The United States created the Advanced Research Projects Agency, or ARPA. The plan emerged to create a communication network that would not be vulnerable to a nuclear attack.
- The basic idea was to use a distributed network and break messages into blocks to route over this network.
- This specific military network was never built, but ARPA proceeded with working on a network for more general communication purposes, in particular to share research.
- In 1969, ARPANET was created, connecting computers at four universities across the country.
- The network used a packet-switching approach, with messages broken up into arbitrary packets that were routed across the network.
- By 1974, this protocol was refined as the Transmission Control Program, or TCP. The document describing TCP used the term 'internet' for the first time, as shorthand for ‘internetworking’.
- ARPANET grew in the next several years, with several other separate networks emerging at the same time, including the Computer Science Network (CSNET).
- The idea emerged to have multiple networks connected in a network of networks.
- In addition to the existing TCP protocol, the Internet Protocol, or IP, was developed to ensure packets of information routed over a network were delivered to the right destination.

- In 1982, the combined TCP/IP protocols were adopted as the communication standard for different networks. This is when the term 'Internet' became widely used to describe the concept of a worldwide network or connected networks.
- A high-speed backbone for the network was built by the National Science Foundation to connect supercomputing centers.
- Local networks were allowed to connect to this backbone, and this really made the Internet viable as a way to communicate between different networks.
- For a number of years, the Internet was mostly used by university researchers and defense contractors, but by the early 1990s, private companies also started using the Internet.
- Meanwhile, local networks were developing in other regions, in particular in Europe. These networks also started using TCP/IP, and gradually, the Internet expanded across the globe.

WHAT IS INTERNET?

The Internet or simply the Net is a worldwide network of computer networks. It is an interconnection of large and small networks around the globe.

APPLICATIONS OF INTERNET

By using Internet facility you can:

- Exchange messages using e-mail (Electronic mail)
- Transfer files as well as software (FTP)
- Browse through information on any topic on web (Internet Surfing)
- Communicate in real time with others connected to the
- Internet (Chatting over Internet)

- Search databases of government, individuals and organizations (Information Search).
- Set up a site with information about your company's products and services (Internet Hosting).
- Read news available from leading news groups.

Resources of Internet

- Mail
- The Web
- Web Search Engines
- Usenet
- Gopher
- Veronica and Jughead
- Anonymous ftp
- Archie
- Mailing Lists
- Telnet
- Talk Facilities
- Internet Relay Chat
- Muds and Other Imaginary Places

Mail

- *Mail referred to as E-mail*
- Electronic Mail or E-Mail is a method of exchanging messages between people using electronic devices

- Email operates across computer networks, which today is primarily the Internet.
- Some email systems required the author and the recipient to be online at the same time, in common with instant messaging.
- More plainly, e-mail is a message that may contain text, files, images, or other attachments sent through a network to a specified individual or group of individuals.
- The first e-mail was sent by Ray Tomlinson in 1971. By 1996, more electronic mail was being sent than postal mail.

The Web

- The Web is a large system of servers which offers all kinds of information to anyone on the Net.
- The information's can be a combination of text, pictures, audio clips, video clips, animations, and other stuff.
- For access this information, to use a client program called a browser.
- Web is so popular, because it is easy to use and it easy to create our own web information to share with people all over the Net.
- Within the web, information is stored in pages. Each page can hold not only information, but links to other pages.
- The idea of data containing links to other data is called hypertext.
- The purpose of the web is to fetch and display pages of hypertext.

Web Search Engine

- It is a software system that is designed to search for information on the World Wide Web.
- Many of these pages are constructed by individuals.

- It is tools that keep track of many web sites around the world.
- For Eg. Suppose want to search particular items.
- The result of a search is a custom list of links, pointing to whatever items the search engine found.
- To select or click any items on the web that may connect to the appropriate web server using client.
- Search engines are Google, Bing, Yahoo, Ask.com etc

Usenet

- It is a system of discussion groups in which individual articles are distributed throughout the world.
- It has thousands of different discussion groups.
- The name Usenet is can contraction o0f "User's Networks".
- Usenet is not a network, it system of discussion groups
- Usenet is a collection of newsgroups where the users can post messages and these posted messages are distributed via Usenet servers.

Gopher

- The Gopher system is similar to the web.
- To use a client to connect to servers all over the world, one at a time.
- All the information available through system is called gopherspace.
- The difference between gopherspace and the web is in the way in which information is organized.
- Within gopherspace, information is represented as series of simple menus.

- Each gopher server has a main menu, and series of sub-menus.
- To use a gopher, to select the menu item which to be want, and client will be fetch for it.
- Menu items can point to a number of different types of resources: files, pictures, other internet resources and so on.
- Advantage of gopher is that menu items can point to other menus.
- A single item on a particular menu can point to the main menu of a completely different gopher server.

Veronica and Jughead

- Veronica was a search engine system for the Gopher
- Veronica searches gopher menus on computers across the Internet analogous to the way archie searches for files available via FTP.
- Veronica is integrated into gopher - a user accesses veronica by selecting a gopher menu item, and the results of a search appear as a gopher menu.
- Jughead in a tool similar to veronica. It searches gopher sites as well, but only one at a time. Jughead servers can be located through veronica.

Anonymous FTP

- Anonymous FTP is a means by which archive sites allow general access to their archives of information.
- These sites create a special account called "anonymous".

- User "anonymous" has limited access rights to the archive host, as well as some operating restrictions.
- In fact, the only operations allowed are logging in using FTP, listing the contents of a limited set of directories, and retrieving files.
- Some sites limit the contents of a directory listing an anonymous user can see as well. Note that "anonymous" users are not usually allowed to transfer files to the archive site, but can only retrieve files from such a site.
- Traditionally, this special anonymous user account accepts any string as a password, although it is common to use either the password "guest" or one's electronic mail (e-mail) address.

Archie

- Archie is (was) an early search program that indexed files on anonymous File Transfer Protocol (FTP) servers and allowed users to search for specific files.
- An archive site is a host that acts as a repository of information,much like a conventional library.
- Information stored on these Internet hosts is made available for users to transfer to their local sites.
- Users run software to identify this information and transfer it to their own hosts. Such a transfer is done with a program that implements the File Transfer Protocol (FTP).

Mailing Lists

- A Mailing list is an organized system in which a group of people are sent messages pertaining to a certain topic.

- The messages can be articles, comments or whatever is appropriate to that topic.
- Subscribe or unsubscribe to a list by sending a message to the appropriate address.
- So the group of subscribers is referred to as "the mailing list", or simply "the list".

Telnet (Remote Connection)

- With some computer systems, need to enter a user name and a password in order to start a work session is called logging process.
- Whenever log in the system, connections can be established to the internet.
- The Internet was set up to allow people to work on remote computers.
- The service allowing logging in and using a remote computer is called telnet.
- To utilize this service, can use a telnet client to make the connection and then provide services of a terminal.
- Using a telnet client is like using a terminal to work with a remote host; host can be anywhere on the Net.

Talk Facilities

- A talk facility allows communicating with other people on the Net in real time, either by typing messages or by actual voice conversation.
- There are a large variety of talk facilities available on the Net.

- Talk to one person at a time in complete privacy, group discussion in private or also have public conversation also.

Internet Relay Chat

- Internet Relay Chat, usually called IRC is a public talk facility which can be used by anyone on the Net at any time.
- Within IRC, there are many, many conversations going on at any time, many of which are organized around a particular topic or idea.
- Each IRC Conversation is carried on a (Chat group) channel.
- To join an existing conversation channel and also to create a new channel also.
- Once the person has leaves a particular channel, it vanishes automatically.
- IRC is widely used by a great many people around the world.
- There are many people have made IRC friends they talk to regularly.

Muds and Other Imaginary Places

- The internet supports a wide variety of interactive virtual environments.
- Its place for to interact with people and situations which offers an imaginary setting of some type.
- The first such computerized environment was modeled after dungeons and dragons and became known as “Multiple User Dialogue”.
- There are now a large number of muds on the Net as well as variations called MUSHs, MOOs, MUSEs and MUCKs.

- Each of these environments has its own characteristics and its own abnk of enthusiasts. And there are all manner of variations.
- Some muds based on action: solving puzzles, fighting other characters, imaginary environment and so on.
- Main purpose of Muds is to provide a place to meet other people and talk.

Requirements of Internet

Need of requirements for connecting the one computer to another computer that is part of the Net.

i) Hardware Requirements

ii) Software Requirements

Hardware Requirements to the Internet:

For hardware requirements, internets needs 3 things

i) A computer

ii) Client programs to run on computer

iii) A way to connect computer to the Net.

Accessing the Internet via Local Network

- Consider the computing resources at a university.
- Throughout the campus, there are rooms of computers for people to use.
- Each of these computers is connected to a local area network; these networks are connected into one large campus network which is itself connected to the internet.

- Using of client, access the internet on the system.

Accessing the Internet via the Telephone System

- Use of the internet by connecting to another computer over a phone line.
- If it is direct internet access means able to easy use. Suppose to connect a computer over phone line which is difficult to use.
- For connect a computer over a phone line, to run a communications program on computer to dial the phone.
- Another computer answer to the phone, and program talk to a program at the other end and connection is established.
- Communication programs need MODEM- it is a hardware device acting as an interface between computer and the telephone system.

Modem

- Data inside a computer is stored in a form that is different from data transmitted over a phone line.
- Computer data is stored in a DIGITAL format; phone line can transmit those data to ANALOG format.
- The process of converting from a digital format to an analog format is called MODULATION.
- The process of converting analog data back to its digital form is called DEMODULATION.
- To act as an interface between computer and the phone system is called a "Modulator/Demodulator" or "MODEM".

To access the Internet over a phone line

- A computer.

- A modem.
- An available telephone line.
- Appropriate software, including a communications programs.
- A telephone number to use to connect to a remote internet computer.

Choosing a Computer

- Computers are different.
- To choose computer based on equipments.
- Computer requires a lot of more speed and memory, a large hard disk.
- The computer should also have CD drive, speakers and a microphone.
- And most important, Operating system is the master control program that runs the computer.

Choosing a Modem

- To choosing a modem is easy.
- A modem acts as the interface between computer and the phone line.
- There are two types of modems: internal and external.
- An internal resides on an adaptor card which fits into an expansion slot inside the computer.
- An external modem is a small box connected to the computer with a cable and it requires a cable and must be plugged into an electrical outlet.
- It is easy to move an external modem from one computer to another.
- Modem speeds are expressed in BITS PER SECOND or BPS.
- There are several standard modem speeds.
- Modems can also be classified by their symbol rate, measured in baud.

- The baud unit denotes symbols per second, or the number of times per second the modem sends a new signal. For example, the ITU V.21 standard used audio frequency-shift keying with two possible frequencies, corresponding to two distinct symbols (or one bit per symbol), to carry 300 bits per second using 300 baud.

ISDN

- It stands for "Integrated Services Digital Network."
- ISDN is a telecommunications technology that enables the transmission of digital data over standard phone lines.
- It can be used for voice calls as well as data transfers.
- There are two levels of service: the Basic Rate Interface (BRI), intended for the home and small enterprise, and the Primary Rate Interface (PRI), for larger users.
- Both rates divide their capacity across a number of channels:
  - B-channels carry payloads (e.g., data or voice streams)
  - D-channels carry control and signaling information.
- A BRI connection consists of two 64 Kbps B-channels and one 16 Kbps D-channel. Thus, a BRI delivers up to 128 Kbps of data.
- ISDN can deliver speeds up to 128 Kbps over home phone lines. Where it is still available for residential users, it - like modems -typically serves as a last-resort technique for low-end data connectivity, used only when DSL and cable modem services are not available.

Software Requirements of Internet

Host and Terminals

- A computer is accessed by a user working at a remote location.

- Typically, the term is used when there are two computer systems connected by modems and telephone lines.
- The system that contains the data is called the host, while the computer at which the user sits is called the remote terminal.
- A computer is connected to a TCP/IP network, including the Internet. Each host has a unique IP address.
- To provide the infrastructure for a computer service.
- For example, there are many companies that host files, programs, applications or even a Web server for companies and individuals.
- To connect to the Net, run a communications program to dial the number of an Internet Service provider.
- When the remote system answers the phone, modem or ISDN service talks to the other system and a connection is established.
- There are two possible types of connections.
    - A shell account and a PPP account
- A PPP account is better, but a shell has more advantage.

UNIX Hosts, VT-100 Terminals and Telnet

- Every computer has a master control program that manages the resources of the computer. This program is called the Operating System.
- The job of the operating system is to make efficient use of the computer hardware and to act as an interface for the person using the computer and for programs running on the computer.
- 1st first major operating system was DOS (Disk Operating System).

- On a Macintosh, the operating system is called MacOS.
- Another Operating system is UNIX. UNIX is unique, that on every type of computer from PCs to the largest supercomputers.
- UNIX is important to use because it is used to run many of the computers on the internet.
- Most of the Internet users use UNIX, because part of the internet culture was adopted directly from the UNIX.
- And one more reason is that UNIX is a multiuser timesharing system.
- There are two ways to access the Internet over telephone line.
  - Computer emulates a terminal and then connects to a timesharing host (usually a UNIX system).
  - For 1st case, computer is actually on the Internet (at least during the time phone connection is active).
  - For 2nd case, computer is not on the Internet, it is used access the remote computer, which is on the net.
- Terminal type or emulation specifies how the computer and the host computer to which are connected exchange information.
- Need to set your terminal type so that both computers communicate in the same way.
- Otherwise, terminal application will not have enough information to perform actions such as clearing the screen, moving the cursor around, and placing characters.

- UNIX was designed to work with many different types of terminals.
- There is one terminal which has become a de facto standard. It is called the VT-100.
- The most commonly emulated terminal type is the Digital Equipment Corporation (DEC) VT100 terminal. Most PC and Mac OS terminal emulation packages support this type.
- The VT100 terminal normally performs a two-part function. It is an input device to a computer – information entered through the keyboard is sent to the computer.
- It is simultaneously an output device for the computer – that is, data coming in from the computer is displayed on the video screen.
- If the computer to which connect does not automatically detect your terminal type, need to set it manually. The procedure for doing so varies from system to system.
- The telnet service to access a remote Internet host.
- Suppose to need an account for the remote computer should enter username and password.

Shell Accounts and PPP Accounts

- For connecting the computer to the telephone system either by using a phone line (with a modem) or an ISDN line (requires special equipments).
- To start work, establish a connection with remote internet host using of communication program with dial a phone.

- Once the connection is established, login to the system by typing of username and password which was provided by the Internet service provider.
- An Internet service provider offers two basic types of Internet accounts.
  - PPP Accounts
  - Shell Accounts

Shell Account

- In this type of Internet Connection, the user will get only textual matter of a Web Page.
- This connection does not support Graphics display.
- Shell Accounts were the only type of Internet access available for many years before the Internet entered in to the world of graphics and became more users friendly.

PPP Account

- PPP (Point-to-Point Protocol) is a communications protocol used to transmit network data over telephone lines.
- It allows you to connect your computer to the Internet itself, rather than logging on through an Internet Service Provider's host computer and using UNIX commands through a shell. This type of connection is communicated directly with other computers on the network using TCP/IP connections.
- It is part of the TCP/IP suite of programs necessary to connect to and use the Internet.
- To have a dial-up account with an Internet service provider, by using either PPP or SLIP to make connection to the Internet.
- PPP (Point-to-Point Protocol) is rapidly replacing SLIP (Serial Line Internet Protocol) as the more common standard.

- Whereas SLIP (Serial Line IP) is easy to install and to use, it does not provide error correction or certain negotiation features that are built into PPP.

Internet Service Providers

- An **Internet service provider** (**ISP**), also sometimes referred to as an **Internet access provider** (**IAP**), is a company that offers its customers access to the Internet.
- The ISP connects to its customers using a data transmission technology appropriate for delivering Internet Protocol packets or frames, such as dial-up, DSL, cable modem, wireless or dedicated high-speed interconnects.

INTERNET SERVICES

Dial-up

- This is the oldest method of providing access to the Internet.
- It uses a telephone line to perform a modem-to-modem connection.
- For that purpose, the computer is attached to a telephone line enabled modem device, which dials into the node of the ISP and starts transferring data between the servers that store websites the user wants to see and their Internet connected device.

DSL

- DSL, short for 'digital subscriber loop' or 'digital subscriber line', is an advanced version of the dial-up Internet access method.
- In contrast to dial-up, DSL uses high frequency to execute a connection over the local telephone network.

- This allows the Internet and the phone connections to be run on one and the same telephone line.
- The digital subscriber line technology ensures an Asymmetric Digital Subscriber Line (ADSL), where the upload speed is lower than the download speed, and a Symmetric Digital Subscriber Line (SDSL), offering equal upload and download speeds.

Cable Internet

- The cable Internet is among the most preferred methods for providing residential Internet access.
- It represents a broadband Internet access method, using the high-bandwidth cable television network to transmit data between the global network and the households.
- To use cable Internet, need a cable modem that will be connected with the CMTS (Cable Modem Termination System) of cable ISP.
- The cable Internet access can be offered together with a cable television subscription and separately, for customers' convenience. The second case incurs higher subscription fees due to the extra equipment installation costs.

Wireless Broadband (WiBB)

- This is a new-generation broadband Internet access technology, allowing the delivery of high-speed wireless Internet within a large area.
- Wireless broadband ISPs (WISPs) ensure connection speeds that come close to the wired broadband speeds provided by DSL and cable ISPs.
- To get wireless broadband, need to place a specific dish on house roof or apartment balcony and point it to the transmitter of WISP. This type of

Internet access is used as an alternative to the wired broadband connection in remote areas.

Wi-Fi Internet

- Wi-Fi (from Wireless Fidelity) has become one of the most widely distributed Internet access methods, with the growing usage of portable computers and Internet enabled mobile devices, such as smart phones, PDAs, game consoles, etc.
- In this sense, it is the most mobile Internet access method, able to use it everywhere within the scope of coverage, i.e. within the range of an Internet connected wireless network.
- Due to its ability to serve mobile devices, Wi-Fi is used in public places such as airports, hotels and restaurants to provide Internet access to customers.
- There are also specialized Wi-Fi hotspots where the service is either free or paid.
- Some of the largest cities in the world are in the process of building Wi-Fi networks that cover all the public places in the central areas.

ISDN

- Another online data transmission method worth considering is ISDN or the Integrated Services Digital Network.
- ISDN represents a telephone system network, integrating a high-quality digital transmission of voice and data over the ordinary phone line.
- Ensuring a much better data transmission over the phone line than an analog line could allow, the ISDN offers a fast upstream/downstream Internet connection speed of 128 kbit/s.

- This speed level can be considered as a broadband speed as opposed to the narrowband speed of standard analog 56k telephone lines.

Ethernet

- Another Internet access type worth mentioning is Ethernet - the most widespread wired LAN (local area network) technology, also used in wireless LANs.
- The Ethernet technology may ensure various speed levels and can thus be divided into several types: regular Ethernet, providing transmission speeds of upto 10 bits/s, fast Ethernet, offering up to 100 mbits/s, gigabit Ethernet, supporting 1 gbit/s and 10-Gbit Ethernet, coming at up to 10 gbits/s.

Internet Protocols

- A different type of networking operating systems uses different digital languages to control the communication process between the computers.
- These languages are called Network protocols.

Definition:

A protocol is a set of rules to govern the data transfer between the devices. The rules are used for the following purposes.

- For compressing the data.
- For sending device to indicate that it has finished sending a message.

Types of Protocols:

There are different types of Protocol such as:

1. Transmission control Protocol (TCP)
2. Internet Protocol (IP)
3. Internet Address Protocol (IP Address)
4. Post office Protocol (POP)
5. Simple mail transport Protocol (SMTP)
6. File Transfer Protocol (FTP)
7. Hyper Text Transfer Protocol (HTTP)
8. Ethernet
9. Telnet
10. Gopher

1. Transmission control Protocol (TCP)

This is a **communication protocol** that computers use to communicate over a network. TCP divides message into stream of packets which are sent and then reassembled at the destination.

2. Internet Protocol (IP)

Internet protocol is **addressing protocol**. It is always used together with TCP. IP addresses of packet, routes them through different nodes and networks until it reaches its final destination. TCP/IP is perhaps the most used standard protocol for connecting computer networks.

3. Internet Address Protocol (IP Address)

This is the address that identifies a computer on a network using TCP/IP. An IP address contains series of four numbers unique to the computer concerned Eg : 90.399.424.34. This address is usually supplied by a Internet Service Provider.

4. Post office Protocol (POP)

This is used to **receive incoming E-mail** .

5. Simple mail transport Protocol (SMTP)

This protocol is used for **sending and distributing outgoing E-Mail** .

6. File Transfer Protocol (FTP)

This is a system that allows users to **transfer files** from one computer to another computer. Files that can be transfered may include program files, text files and multimedia files ect. This method of file transfer is faster than that using HTTP.

7. Hyper Text Transfer Protocol (HTTP)

HTTP is used to transfer a hyper text between two or more computers. Hyper text is the text that is coded using the language called HTML. HTML codes are used to create links. This link may be in any format such as text or graphics.

HTTP is based on the **Client/server principles**. HTTP allows a client to establish connection with a server and make a request. The server accepts the connection initiated by the client and sends back the response. An HTTP request identifies the resources that the client is interested it and tells the server what action to take on the resource.

8. Ethernet

Ethernet is a most popular protocol used for LAN communication. It transfers the information in digital packets. Every computer that uses this protocol contains the **Ethernet Network Interface Card (NIC).**

This card is an unique address code embedded in microchip. This address is used to identify the system. When a packet is placed on a network, The packet is sent to every computer on the networks but only the computer which has the same address as Network Interface Card address is allowed to claim that packet and receive the digital information contained with it.

Only one packet is placed at a time in the network. If two systems are placed packets on the network on same time. When NIC tries to place a packet it first looks whether the packet is already exist or not. If it exits the NIC is forced to wait for random milliseconds before trying again. This continues until a 'gap' is found then the packet can be successfully transmitted on to a network.

9. Telnet

Telnet is a set of rules used to connect one computer to another computer. The process of this connection is called as **remote login**. The computer who request connection is called local computer, who accept the connection is called remote computer. If you type commands in local computer remote login these commands executed in the remote computer. You can see in your monitor what is the process going on in this remote computer.

Telnet also operates on the client/server principle. The establishment of connection and display data on the local computer uses a Telnet server program to accept the connection and send responses to requests for information back to the local computer.

10. Gopher

Gopher is a set of rules used to search, retrieve and display documents from remote sites. It is possible to initiate on-line connections with other systems through Gopher. It also operates on client/server principal.

Internet Server and Internet Client

Internet Client

- A client is a computer that connects to and uses the resources of a remote computer, or server.
- Many corporate networks comprise a client computer for each employee, each of which connects to the corporate server.
- The server provides resources like files, information, Internet and intranet access, and external processing power.
- In the case of processing, any work done on the server is referred to as "server-side" work. Any work done on the local client is similarly called "client-side."
- Client is also another name for a software program used to connect to a server.
- A client can also be another term used to describe a user.

Internet Server

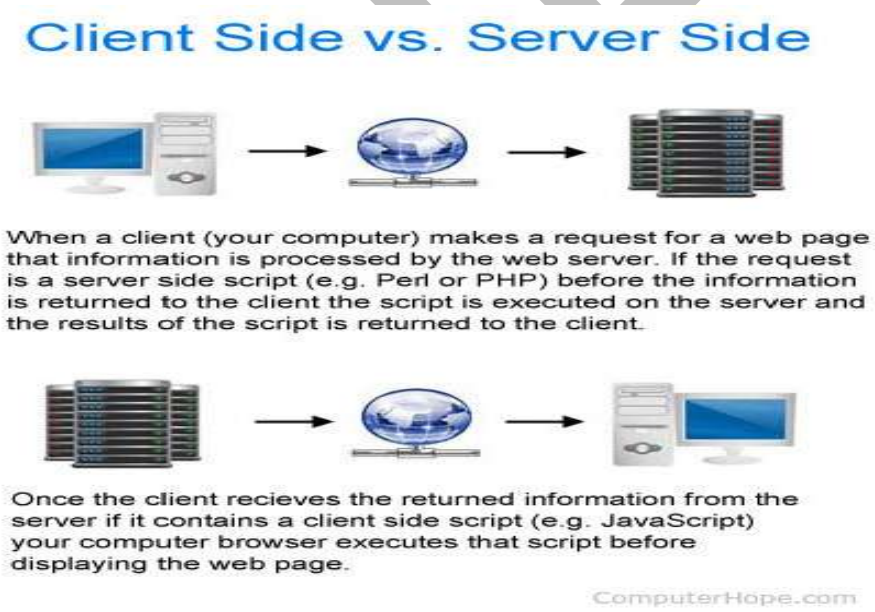
- A **server** is an instance of a computer program that accepts and responds to requests made by another program, known as a client.
- Servers are used to manage network resources.

- For example, a user may setup a server to control access to a network, send/receive e-mail, manage print jobs, or host a website.
- Some servers are committed to a specific task, often referred to as dedicated. As a result, there are a number of dedicated server categories, like print servers, file servers, network servers, and database servers.
- However, many servers today are shared servers which can take on the responsibility of e-mail, DNS, FTP, and even multiple websites in the case of a web server.
- Because they are commonly used to deliver services that are required constantly, most servers are never turned off.
- Consequently, when servers fail, they can cause the network users and company many problems.

Here are a few types of servers, among a great number of other possibilities:

- An application server is a program in a computer in a distributed network that provides the business logic for an application program.
- A proxy server is software that acts as an intermediary between an endpoint device, such as a computer, and another server from which a user or client is requesting a service.
- A mail server is an application that receives incoming e-mail from local users (people within the same domain) and remote senders and forwards outgoing e-mail for delivery.
- A virtual server is a program running on a shared server that is configured in such a way that it seems to each user that they have complete control of a server.

- A blade server is a server chassis housing multiple thin, modular electronic circuit boards, known as server blades. Each blade is a server in its own right, often dedicated to a single application.
- A file server is a computer responsible for the central storage and management of data files so that other computers on the same network can access them.
- A policy server is a security component of a policy-based network that provides authorization services and facilitates tracking and control of files.



POSSIBLE QUESTIONS

PART A (1 Mark)

(Online Examinations)

PART B (8 Marks)

1. Explain in detail the resources of internet.
2. What is the software requirements used in internet?
3. Write in detail the hardware requirement of internet.
4. Explain in detail the internet services.
5. Write in detail about the internet protocols.
6. Define client. Explain the internet client and server.
7. Explain the SHELL and PPP Account.
8. Discuss in detail the History of internet.
9. Describe about Internet Service Provider in Internet.
10. Write in detail about the Hardware and Software requirements of Internet.

KARPAGAM ACADEMY OF HIGHER EDUCATION

(Deemed to be University)
(Established Under Section 3 of UGC Act 1956)
Pollachi Main Road, Eachanari Post, Coimbatore - 641021
(For the candidates admitted from 2015 onwards)

CLASS: III B.COM CA **COURSE NAME: INTERNET AND WEB DESIGN**
COURSE CODE: 15CCU601 **UNIT: I (INTRODUCTION TO INTERNET)** **BATCH-2015-2018**

S.NO	QUESTIONS	OPTION 1	OPTION 2	OPTION 3	OPTION 4	ANSWER
1	_____ is a network in which the computers are connected directly.	LAN	WAN	NET	ARPANET	LAN
2	_____ is connected to satellites links.	WAN	LAN	WWW	None	WAN
3	Which of the following is the “Backbone” of the network?	LAN	WAN	NET	Network	LAN
4	LANs are connected by the special purpose of computers are called _____.	Routers	Resources	Networks	None	Routers
5	_____ is used to access resources.	Clients	Servers	Client - Servers	Internet	Clients
6	In web language, a client program is called a _____.	Browser	Address	Server	All the above	Browser
7	To send and receive electronic mail, one should familiar with _____.	mail client	client – server	internet	e-mail address	mail client
8	Within the internet, each separate computer is called _____.	Host	Node	Both 1 & 2	None	Host
9	Programmers write their programs using _____.	Protocols	Script	Both 1 & 2	None	Protocols
10	TCP stands for _____.	Transmission Control Protocol	Transiting Control Protocol	Transmission Control Process	Transmission Control Programs	Transmission Control Protocol
11	IP stands for _____.	Internet Protocol	I – Protocol	Internet – P	None	Internet Protocol
12	_____ will divide the message into a number of _____.	TCP, Packets	IP, Packets	TCP, Host	IP, Host	TCP, Packets
13	The data containing links to another data is called _____.	Hypertext	TCP	E-mail address	None	Hypertext
14	The name Usenet is a contraction of _____.	User network	User’s network	Network	None	User’s network
15	Which one of the following has main-menu and sub-menus?	Gopher	Usenet	Web	Telnet	Gopher
16	The FTP service allows to _____ from one computer to another.	Copy files	Send data	Connect	All the above	Copy files
17	FTP stands for _____.	File Transfer Protocol	File Transaction Protocol	Fixed Transfer Protocol	File Transiting Protocol	File Transfer Protocol

18	Anonymous “ftp” is one of the most important in _____.	Internet Services	Web Services	Gopher Services	Usenet Services	Internet Services
19	The role of _____ is to make the whole system manageable.	Archie	Gopher	Website	Usenet	Archie
20	The service allowing us to login and use a remote computer is called _____.	Telnet	Usenet	IP	None	Telnet
21	IRC stands for _____.	Internet Relay Chat	Internet Remote Chat	Intranet Relay Chat	Internet Relay Consumer	Internet Relay Chat
22	Each IRC conversation is carried on a _____.	Channel	Wires	Cables	Circuits	Channel
23	The Description of web search engines is _____.	Search the web	Receives messages	Gathering information	All the above	Search the web
24	Menu – Based information is a _____.	Gopher	Archie	Usenet	Mail	Gopher
25	Multi – linked information is a _____.	The web	Mailing lists	Talk facility	Using of e-mail	The web
26	_____ is a hardware device.	Modem	Cable	Wires	None	Modem
27	Computer data is stored in a _____ format.	Digital	Analog	Both 1 & 2	None	Digital
28	Phone lines transmit data in a _____ format.	Digital	Analog	Both 1 & 2	None	Analog
29	_____ is the process of converting from a digital to Analog format.	Modulation	Demodulation	Modem	Data	Modulation
30	_____ is the process of converting from Analog to digital format.	Modulation	Demodulation	Modem	Data	Demodulation
31	The device that acts as an interface between the computer & phone lines is	Modulator	Demodulator	Modem	All the above	All the above
32	Find the odd man out.	Modem	Computer	Telephone lines	Bits	Bits
33	_____ is the master control program that runs the computer.	Operating System	Hard disk	Motherboard	None	Operating System
34	Adaptor Card refers to which type of modem?	Internal modem	External modem	Both 1 & 2	None	Internal modem
35	An external modem is a _____ connected to computer with a cable.	Adaptor Card	Small Box	Big Box	None	Small Box
36	Modem is of _____ types.	1	2	3	4	2
37	Which of the following is easy to turn off and On to reset it?	External modem	Internal modem	Both 1 & 2	None	External modem
38	To restart the whole computer _____ is used.	Internal modem	External modem	Both 1 & 2	None	Internal modem
39	Modem speed is expressed in _____.	Bits Per Second	Kilobytes Per Second	Bits	None	Bits Per Second
40	Individual computer on the internet is referred by _____.	Host	Terminal	Multi-User	All the above	Host

41	The Basic Rate Interface Providers 3 Channels. Namely,	Two B channels, 1 D channel	Two D channels, 1 B channels	Three D channels only	Three B channels only	Two B channels, 1 D channel
42	Which of the channels carry the bulk of the data?	B channels	D channels	C channels	All the above	B channels
43	The B channel refers to _____.	Bearer channel	Board channel	Bulk of channel	Boarding channel	Bearer channel
44	The D channel refers to _____.	Data channel	Disconnect channel	Device channel	Digital channel	Data channel
45	D channel carries the _____ ISDN data.	Internal	External	Both 1 & 2	None	Internal
46	The 2 basic types of Internet Accounts is _____.	PPP & SHELL	SHELL & TCP	PPP & IP	None	PPP & SHELL
47	_____ can use both characters and draw shapes.	Graphical Interface	Text-based interface	Text only	None	Graphical Interface
48	Which of the following Account will “login” automatically?	PPP Account	SHELL Account	Both 1 & 2	None	PPP Account
49	The way to understand a domain name is to look at the sub-domains from _____.	Right to Left	Left to Right	Left only	None	Right to Left
50	E-mail addresses are _____.	Case Sensitive	Non Case Sensitive	Equivalent	None	Case Sensitive
51	Any data that is not simple text is referred to as _____.	Binary data	text	special data text	None	Binary data
52	A colon, a minus sign and a right parenthesis refers to _____.	Smiley	Special characters	Punctuation characters	Used in Usenet	Smiley
53	URL stands for _____.	Uniform Resource Locator	Unique Resource Locator	Uniform Resource Link	Unique Resource Link	Uniform Resource Locator
54	The top-level domain is the _____ sub-domain in the address.	Last	First	Period	None	Last
55	To transfer data from a remote computer to our computer is known as _____.	Download	Upload	Copy the file	None	Download
56	To transfer data from our computer to remote computer is known as _____.	Download	Upload	Copy the file	None	Upload
57	A formal description of the user name and address refers to _____.	Address	Address Book	Action	None	Address
58	A Program we use to access the Gopher System is _____.	Gopher Client	Gopher Server	Archie	Usenet	Gopher Client
59	A Server that provides access to Gopher Information is _____.	Gopher Space	Gopher Client	Gopher Server	None	Gopher Server
60	On the web, a designation within a URL that indicates a hypertext is _____.	Http	Html	Both 1 & 2	None	Http

UNIT II

SYLLABUS

Introduction to HTML – Functions of HTML in Web Publishing – Basic Structural Elements and their Usage – Traditional Text and Formatting – Using Tables for Organisation and Layout – Forms – Frames and Frame sets - Style Sheets of Formatting - Advanced Layouts and Positioning with Style Sheets.

Introduction to HTML:

- HTML stands for Hyper Text Markup Language.
- Hyper text is the text displayed on a computer display or other electronic devices with references to other text.
- The references can be otherwise called as hyperlinks which links the other document with current document.
- The document / file that contain a hypertext can be called as Hyper Document. It is of two types.
  - ✓ **Static:** Document is already prepared and stored in advance.
  - ✓ **Dynamic:** Changes according to the user's input.
- Using hypertext complex and dynamic systems of linking and cross referencing can be developed.
- HTML is a markup language which contains a set of markup tags is used to describe the web pages.
- HTML documents / web pages mainly contain HTML tags and plain text.

- The HTML tags are the keywords that are surrounded by angle brackets E.G:- "**<keyword>**".
- Each HTML tags will be in pairs called as start and end tags. With the end tag having a slash "**/**" in it.

History of HTML:

- 1960s-GML (Generalized Markup Language) was developed by Charles Goldfarb, Edward Musher and Raymond Lorie at IBM for organizing vast amount of documentation.
- 1978- ANSI took basics of GML and fashioned nationwide standard called GCA.
- Six years later ISO began to work on new global version called SGML (Standard Generalized Markup Language).
- 1989-Tim Berners Lee a physicist at CERN(Centre European pour la Recherche Nucleaire) created a method for the scientists to share papers. In the same year a memo proposing hypertext system was released.
- Late 1990- Tim Berners Lee specified HTML and wrote a browser and server software.
- Late 1991-public description of HTML was released containing 18 elements.
- 1995-HTML 2.0.
- Jan 1997-HTML 3.2 (W3C Recommendation).
- Dec 1997- HTML 4.0 (W3C).
- April 1998 – HTML 4.0 got minor edits.

- Dec 1999-HTML 4.0.1.
- Jan 2008-HTML 5.
- Mean while two languages related to hyper text where developed.
  - ✓ **DHTML→**combination of HTML 4, Cascading Style Sheets and JavaScript to create dynamic Web pages. These work on basis of DOM(Document Object Model) to change the content of the web page even if it's loaded into the browser.
  - ✓ **XML→**XML stands for Extensible Markup Language using which we can create our own markup languages. It is developed by W3C.

Basic Structure Elements and their Usage

The basic structure of a HTML document and the description of the tags present in it are as below.

1. **<!DOCTYPE>-** Defines the document type.
2. **<HTML>-** indicates the browser that it is a HTML document and </HTML> tells the browser that HTML document is completed.
3. **<HEAD>-** container for all the head elements.
4. **<BODY>-** contains all the contents of an HTML document DOCTYPE Element

<!DOCTYPE> Element

- ➢ The <!DOCTYPE> declaration must be the very first thing in your HTML document, before the <html> tag.
- ➢ The <!DOCTYPE> declaration is not an HTML tag; it is an instruction to the web browser about what version of HTML the page is written in.

- In HTML 4.01, the <!DOCTYPE> declaration refers to a DTD, because HTML 4.01 was based on SGML. The DTD specifies the rules for the markup language, so that the browsers render the content correctly.

```
<!DOCTYPE html>
<html>
<head>
<title>Title of the document</title>
</head>

<body>
The content of the document……
</body>

</html>
```

- **HTML 5**

  <!DOCTYPE html>

- **HTML 4.01 Strict**

  - This DTD contains all HTML elements and attributes, but does NOT INCLUDE presentational or deprecated elements (like font). Framesets are not allowed.
  - <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">

- **HTML 4.01 Transitional**

  - This DTD contains all HTML elements and attributes, INCLUDING presentational and deprecated elements (like font). Framesets are not allowed.
  - <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">

- **HTML 4.01 Frameset**

  - This DTD is equal to HTML 4.01 Transitional, but allows the use of frameset content.
  - <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN" "http://www.w3.org/TR/html4/frameset.dtd">

HTML Element

- The document begins with <HTML> and ends with </HTML>
- The HTML start tag contains no other content.
- Technically it have three attributes.
- The first is version, and it takes a URL value.
- The URL points to a location that has the Document Type Definition(DTD)for the version of HTML in use on that page.
- Another 2 attributes are lang and dir atributes
- Lang attributes-establish human language in which the web page is written.
- For example: an for english and fr from french
- Dir attributes-the direction of printing(right to left(rtl) or left to right(lrt))

<HTML version=http://www.w3.org/DTD/HTML4-strict.dtd lang=en dir=rt1>

HEAD Element

- The HEAD element contains *header information* about the document, such as title, keywords, description, and style sheet.
- It contains one required element is **TITLE.**
- The HEAD element has both start and end tags, beginning with <HEAD> and ending with </HEAD>.
- The text between the TITLE tags is displayed in the title bar of a Web browser.
- **<TITLE>- defines** title of the document, title in the browser toolbar, title for the page when it is added to favorites, title for the page in search-engine results. (E.g):- **<title>Sample Program</title>**

Metadata

A **HEAD** element can contain other elements known as metadata.

Metadata is information ,other than that containeds in the body of the HTML page.

It is data that is not shown to the person viewing the page in a web browser,its useful to user agents an search engines.

The Head element has two separtate methods for using metadata.

One is an attribute of HEAD element. i.e profile attribute.

Another one is used to separate elements that ate required to be containes within the HEAD element.

The profile is specified as aURL.

For <HEAD profile = "http://www.whatever.com/prfile/data.html">

There are several elements that can be contained with in the HEAD.

- BASE
- LINK
- META
- SCRIPT
- STYLE

BASE Element

<BASE>- specifies the base URL/target for all relative URLs in a document. Contains either a href attribute or target attribute or sometimes both. All the elements at the body of the html document will be referring the base tag's.

The LINK Element

The HTML link tag is used for defining a link to an external document. It is placed in in the <head> section of the document. The HTML link tag is commonly used for linking to an external style sheet. .eg. <link rel="stylesheet" type="text/css" href="/global.css" />

The BODY Element

- The body element contains all the contents of an HTML document, such as text, hyperlinks, images, tables, lists, etc.

  <BODY BACKGROUND= “Penguins.Jpg”,text= “000000”,Link= “0000FF”
  VLINK= “FFBC00” ALINK= “000000”>
  HAI
  </BODY>
- The code sets the background image to read in the file penguins.jpg and sets the text and link colors.
- The color can specifies by the hexadecimal numbers for RGB values.

The attributes of the body tag are

- **alink-** The alink attribute specifies the color of an active link in a document. The default color of the active link in a document will be red.
- **background-** specifies a background image for a document using the image’s URL. Proper image that is image with correct resolution should be chosen to display and if not then the display image will be inconvenient to the user/client.
- **bgcolor-** specifies the background color of a document. For specifying the color of the document the either the name of the particular color can be

used or the hexadecimal value for a color can be used or rgb value can be used.

- **link-** specifies the default color of unvisited links in a document. The default color of an unvisited in a HTML document will be blue.
- **text-** used to specify the color of text displayed in a document. Default color of the body text is black or it may differ based on the web browser of the client.
- **vlink-** used to specify the color of visited links displayed in a document. All the visited links in a HTML document will be pink in color.

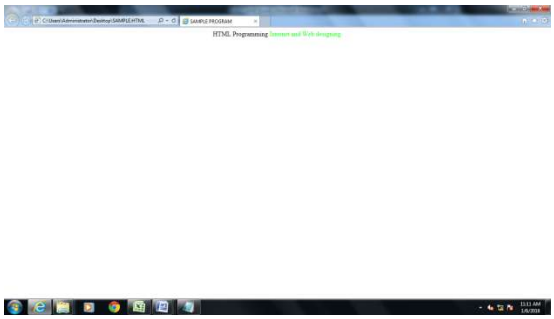
The DIV and SPAN Element

- The DIV and SPAN elements form a short of "BODY within the BODY".
- The tag defines a division or a section in an HTML document.
- The tag is often used to group block-elements to format them with styles.

  eg:
- The tag is used for grouping and applying styles to inline elements.
- There is a difference between the span tag and the div tag.
- The span tag is used with inline elements whilst the div tag is used with block-level content.

For example

```
<BODY >
<div align="center">
 HTML Programming
<span style="color:#00FF00">
Internet and Web designing
</BODY>
```



Traditional Text and Formatting

- Text Layout
- Lists
- Text styles
- The Font and BASEFONT Elements

Text Layout

- There are two basic types of text-affecting elements in HTML.
- The first performs text layout task, it set up paragraphs, line breaks or ordering text into lists.
- The second kind is text styles.

The P Element

- HTML paragraph tags are used to define the HTML paragraph element.
- The paragraph element begins with the HTML tag and ends with the HTML tag.
- The HTML paragraph element should not contain tables and other block elements.

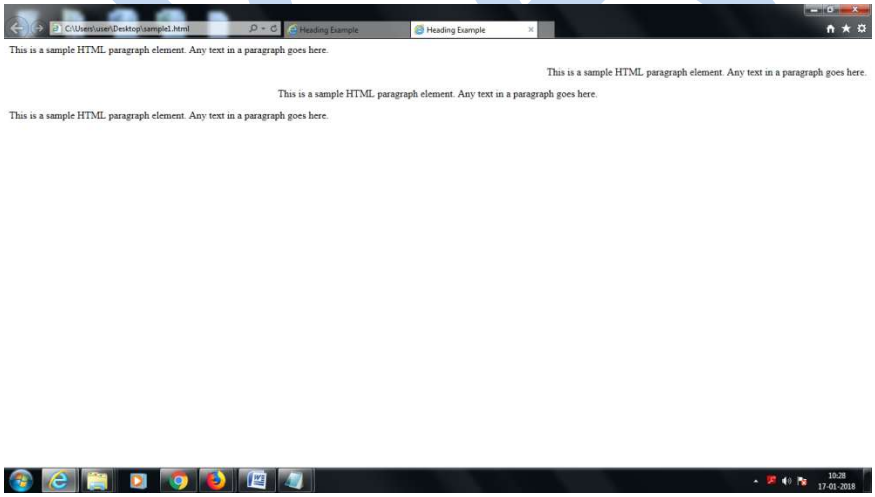
For example

This is a sample HTML paragraph element. Any text in a paragraph goes here.

- Paragraphs cab be aligned flush left, right justify or centered depending on the setting in the align attribute of the P elements

For example

```
<p align=left>
This is a sample HTML paragraph element. Any text in a paragraph goes here.
</p>
<p align=right>
This is a sample HTML paragraph element. Any text in a paragraph goes here.
</p>
<p align=center>
This is a sample HTML paragraph element. Any text in a paragraph goes here.
</p>
<p align=justify>
This is a sample HTML paragraph element. Any text in a paragraph goes here.
</p>
```

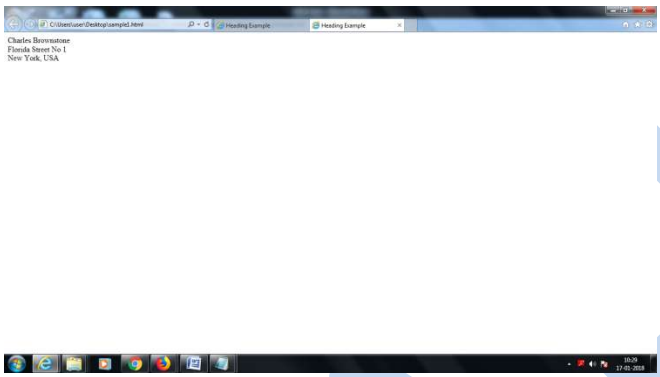


BR Element

The BR element forces a line break.

A line break is the same as a CR/LF (Carriage return/line feed) combination.

```
For example
<p>
        Charles Brownstone<br />
        Florida Street No 1<br />
        New York, USA<br />
</p>
```



CENTER Element

The CENTER element causes all text between its start and end tags to be centered between the margins.

<CENTER>This text will be center-aligned.</CENTER>

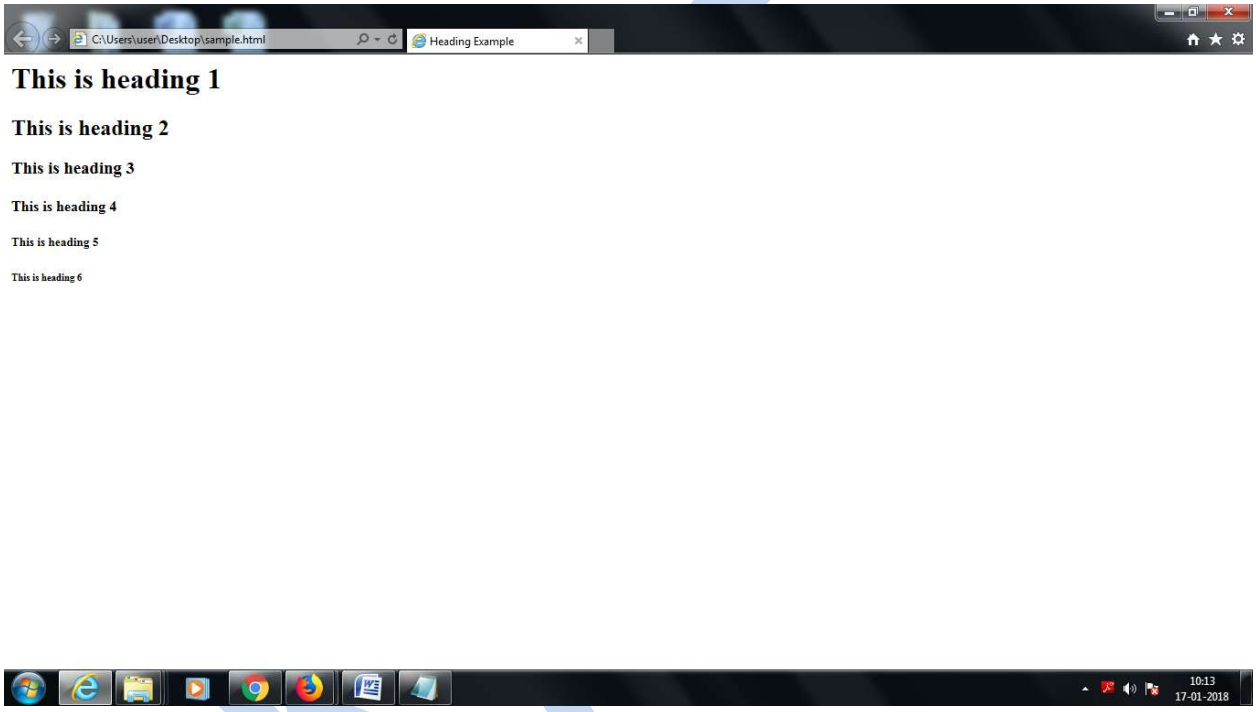
The HN Element

- The <h1> to <h6> tags are used to define HTML headings.
- <h1> defines the largest heading and <h6> defines the smallest heading.
- Heading elements are required to have end tags as well as start tags

```
For Example
<html>

   <head>
      <title>Heading Example</title>
   </head>
```

```
    <body>
       <h1>This is heading 1</h1>
       <h2>This is heading 2</h2>
       <h3>This is heading 3</h3>
       <h4>This is heading 4</h4>
       <h5>This is heading 5</h5>
       <h6>This is heading 6</h6>
    </body>

</html>
```

This will produce the following result –



The HR Element

- Horizontal rules are used to visually divide different segments of web page from one another.
- It does not has end tag
- By default, it is shaded line,two pixels thick, which stretches from one side of the page to the other.

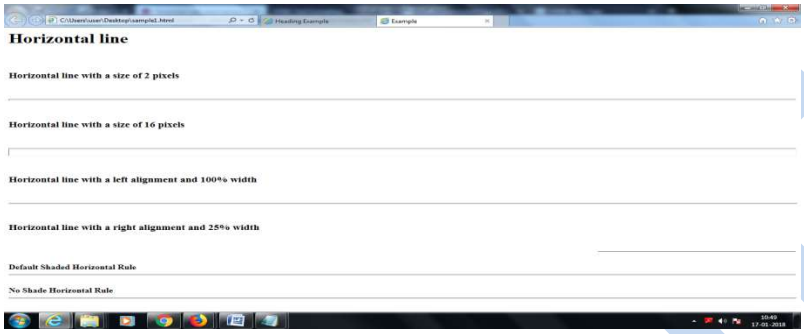
It has 4 attributes

- The size attributes to specify thickness of line in pixels
- The align attribute can be with Horizontal Rule.
- The width attributes to specify width of line in pixels or percentage of screen width.
- The noshade attribute to specify that a plain black line is to be drawn as opposed to (default) 3D shading. noshade has no value assigned to it.

For example

```
<html>

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

  <body>
<h1>Horizontal line</h1>
<br>
 <h3>Horizontal line with a size of 2 pixels</h3>
<br>
<hr size=2>
<br>
<h3>Horizontal line with a size of 16 pixels</h3>
<br>
<hr size=16>
<br>
<h3>Horizontal line with a left alignment and 100% width</h3><br>

        <hr width=100% align=left>
<br>
<h3>Horizontal line with a right alignment and 25% width</h3><br>

        <hr width=25% align=right>
<br>
<b>Default Shaded Horizontal Rule </b><br>
        <hr>
<br><b>No Shade Horizontal Rule </b>
        <hr  noshade>
```

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



LISTS

One of the methods for organizing information is by using lists.

It has three kinds of lists are ordered, unordered lists and definition lists:

- **<UL>** – An unordered list. This will list items using plain bullets.
- **<OL>** – An ordered list. This will use different schemes of numbers to list your items.
- **<DL>** – A definition list. This arranges your items in the same way as they are arranged in a dictionary.

Unordered Lists

- An unordered list is a collection of related items that have no special order or sequence.
- This list is created by using **<UL>** tag.
- Each list item starts with the <LI> tag.
- The list items in unordered lists are marked with bullets (small black circles), by default

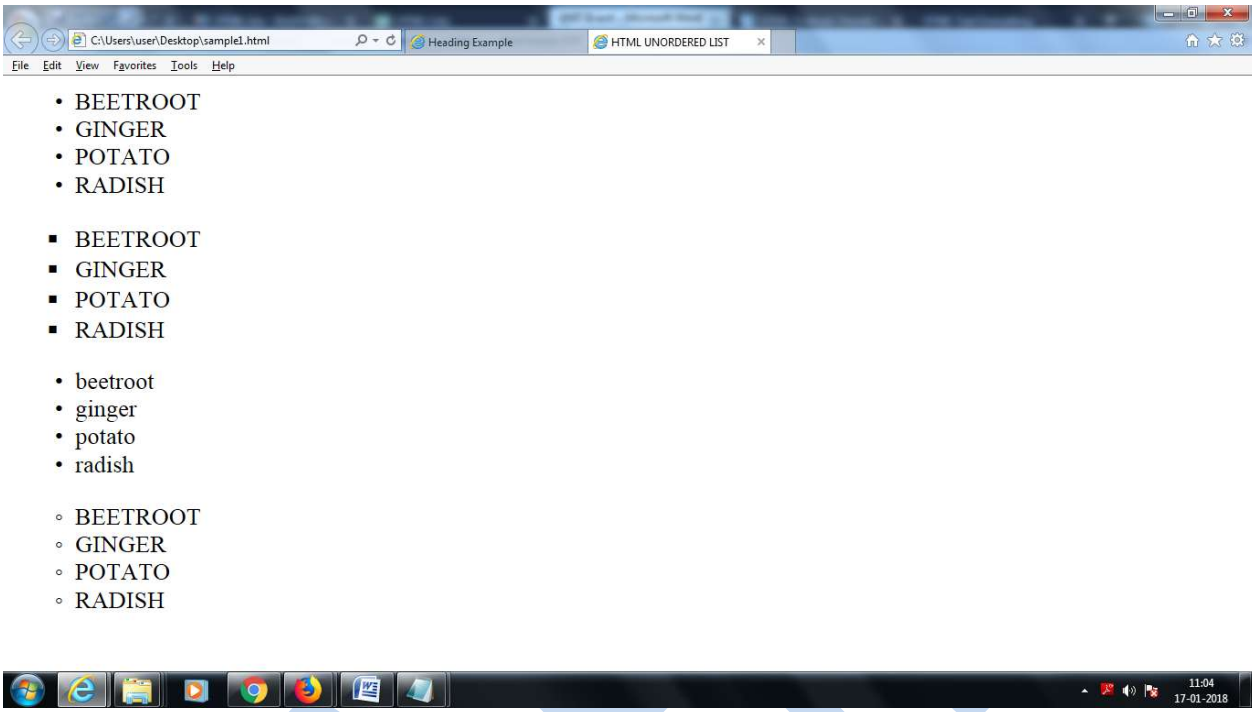
```
<UL>
    <LI>Chocolate Cake</LI>
```

```
    <LI>Black Forest Cake</LI>
    <LI>Pineapple Cake</LI>
</UL>
```

- The Type attribute, it defines the type of bullets used to denote the individual list items.
- The three options are disc, circle and square.

For example

```
<HTML>

   <HEAD>
      <TITLE>HTML UNORDERED LIST</TITLE>
   </HEAD>

   <BODY>
      <UL>
         <LI>BEETROOT</LI>
         <LI>GINGER</LI>
         <LI>POTATO</LI>
         <LI>RADISH</LI>
      </UL>

<UL type = "square">
         <LI>BEETROOT</LI>
         <LI>GINGER</LI>
         <LI>POTATO</LI>
         <LI>RADISH</LI>
      </UL>
<UL type = "disc">
         <li>beetroot</li>
         <li>ginger</li>
         <li>potato</li>
         <li>radish</li>
      </ul>
<ul type = "circle">
         <LI>BEETROOT</LI>
         <LI>GINGER</LI>
         <LI>POTATO</LI>
         <LI>RADISH</LI>
      </UL>

   </BODY>

</HTML>
```



Ordered Lists

- An ordered list starts with the <ol> tag.
- Each list item starts with the <li> tag.
- The list items are marked with numbers.

```
<ol>
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ol>
```

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

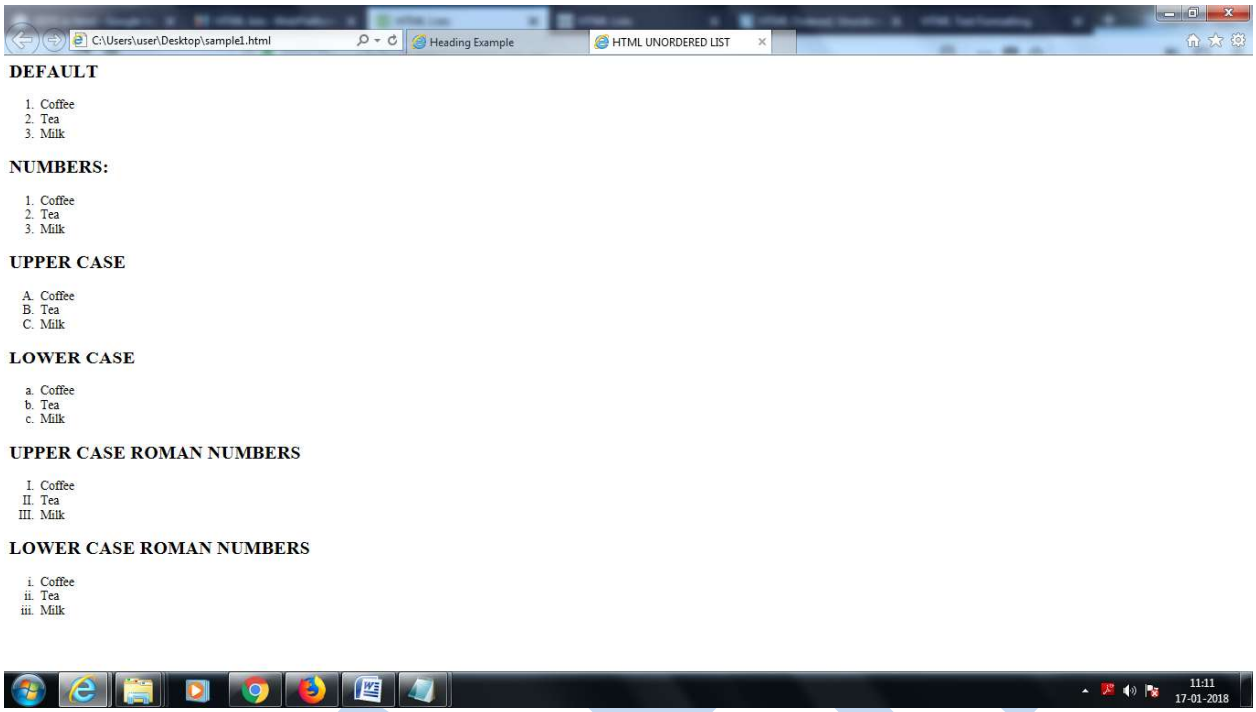
Type	Description
type="1"	The list items will be numbered with numbers (default)
type="A"	The list items will be numbered with uppercase letters

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

```
<HTML>
  <HEAD>
    <TITLE>HTML UNORDERED LIST</TITLE>
  </HEAD>

  <BODY>
<h2> DEFAULT</h2>
<ol>
 <li>Coffee</li>
 <li>Tea</li>
 <li>Milk</li>
</ol>
<h2> NUMBERS:</h2>
<ol type="1">
 <li>Coffee</li>
 <li>Tea</li>
 <li>Milk</li>
</ol>
<h2> UPPER CASE</h2>
<ol type="A">
 <li>Coffee</li>
 <li>Tea</li>
```

```
  <li>Milk</li>
</ol>
<h2> LOWER CASE</h2>
 <ol type="a">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ol>
<h2> UPPER CASE ROMAN NUMBERS</h2>
 <ol type="I">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ol>
<h2> LOWER CASE ROMAN NUMBERS</h2>
 <ol type="i">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ol>
</HTML>
```

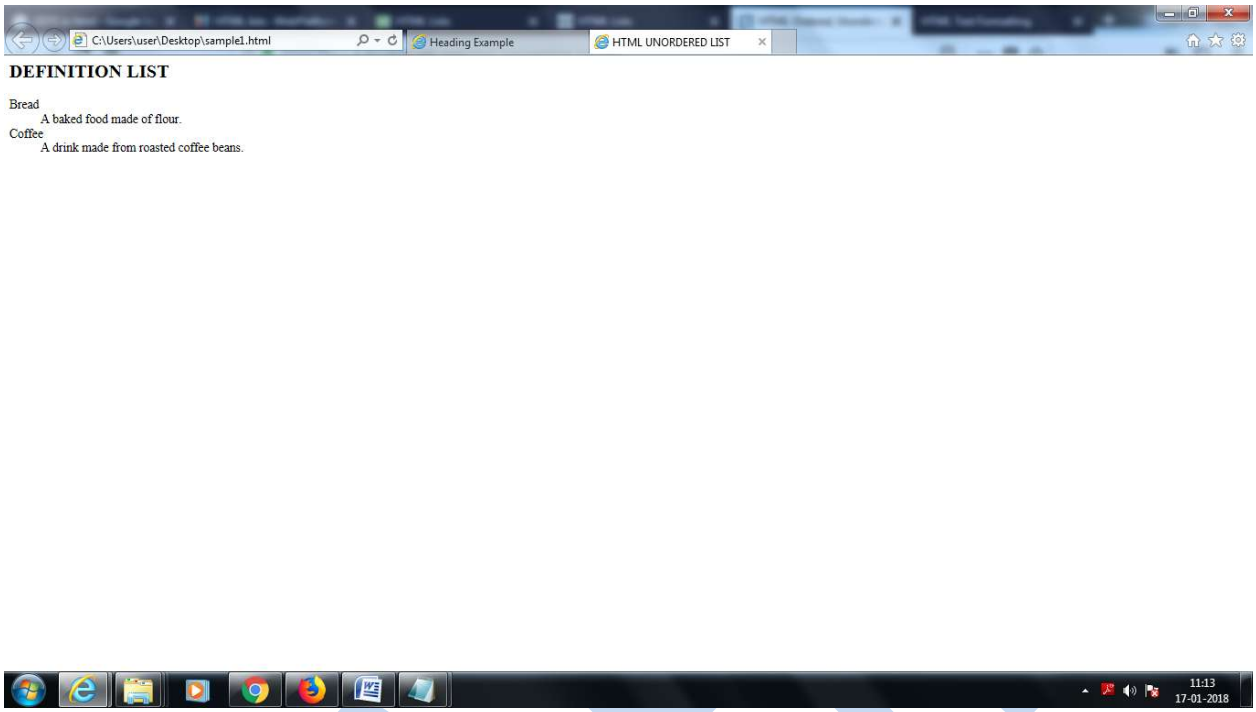


Definition Lists

- A definition list is a list of items, with a description of each item.
- The <dl> tag defines a definition list.
- The <dl> tag is used in conjunction with <dt> (defines the item in the list) and <dd> (describes the item in the list)

For example

```
<dl>
    <dt>Bread</dt>
    <dd>A baked food made of flour.</dd>
    <dt>Coffee</dt>
    <dd>A drink made from roasted coffee beans.</dd>
</dl>
```



Text Styles

- Text styles are inline elements that change the appearance of text.
- The most commonly used are the I element, which makes text italic, and the B element, which makes it bold.
- Others perform such tasks as creating a typewriter-style monospaced font, increasing or decreasing text size, or raising and lowering characters for superscripting and subscripting.

The B and STRONG Elements

The <B> and <STRONG> tags has the effect of text in bold print.

The tags can be inserted anywhere in text.

Both have required end tags (</B> and </STRONG).

For example

<HTML>

```
<HEAD>
  <TITLE>HTML UNORDERED LIST</TITLE>
</HEAD>

<BODY>
<strong>This bold text</strong> is in strong tags.
 <BR>
<b>This bold text</b> is in b tags.
</BODY>
</HTML>
```

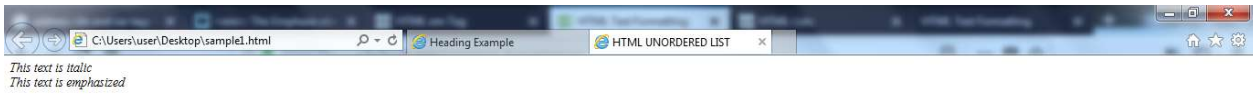
The I and EM Elements

- <I> and <EM> (emphasis) tags has the effect of rendering text in italicized print.

- The tags can be inserted anywhere in text.
- Both have required end tags (</I> and </EM>).
- Three other elements, ADDRESS, CITE and VAR, also place text in italics.
- ADDRESS element is used at the bottom oa a home page for the purpose os starting contact address and copy notices.
- The CITE element italicizes the title of a book or other citation.
- The VAR element indicates a program variable.

For example

```
<HTML>
  <HEAD>
    <TITLE>HTML UNORDERED LIST</TITLE>
  </HEAD>

  <BODY>
 <i>This text is italic</i>
   <BR>
 <em>This text is emphasized</em>
</BODY>
</HTML>
```



The TT Elements

- The **<tt> tag** was used to designate inline teletype text.
- It was intended to style text as it would appear on a fixed-width display, using the browser's default monotype font.
- The elements CODE, SAMP, KBD do the same thing as the TT elements.
- CODE is for displaying snippets of the program code.
- SAMP is for sample program output.
- KBD is for text to be entered by the user.

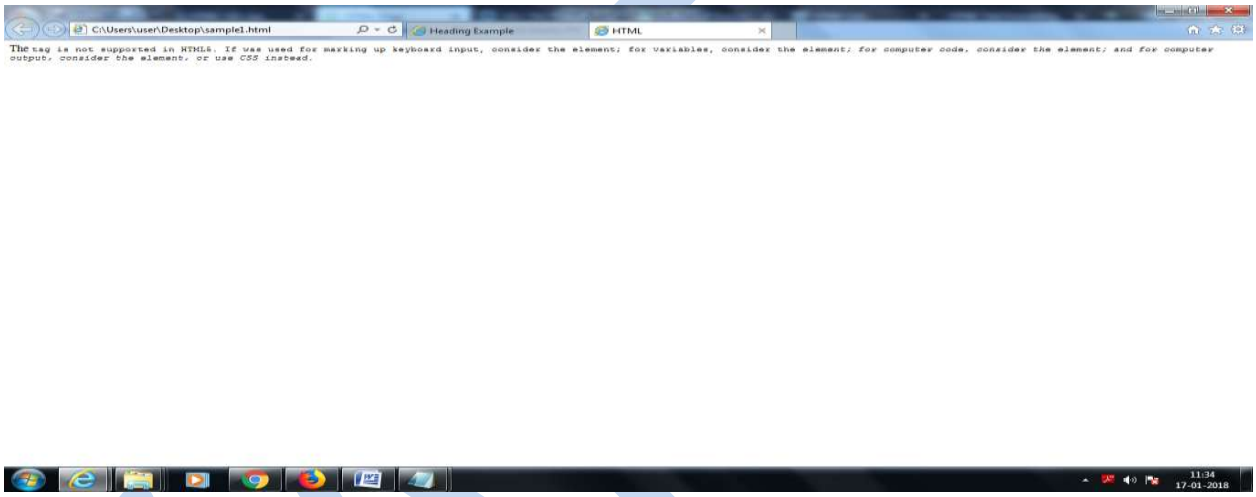
For example

```
<HTML>
  <HEAD>
    <TITLE>HTML UNORDERED LIST</TITLE>
  </HEAD>

  <BODY>
```

The <tt> tag is not supported in HTML5.

If <tt> was used for marking up keyboard input, consider the <kbd> element; for variables, consider the <var> element; for computer code, consider the <code> element; and for computer output, consider the <samp> element, or use CSS instead.

</HTML>



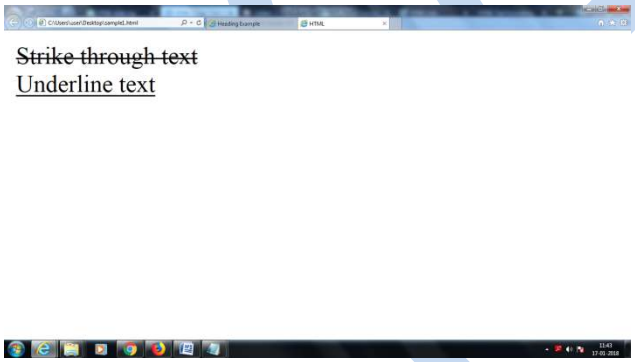
The STRIKE and U Elements

The `<strike>` (short for *strikethrough*) element renders the enclosed text in ~~strikethrough~~ style i.e. with a line drawn through it.

The `<u>` (short for underline) tag renders the enclosed text with an underline.

For example

```
<HTML>
  <HEAD>
    <TITLE>HTML </TITLE>
  </HEAD>
  <BODY>
<strike> Strike through text</Strike>
<br>
<U>Underline text</U>
</BODY>
</HTML>
```



The BIG, SMALL, SUP, and SUB elements

- These four elements actually change the size or position of the affected text.
- The BIG element increases the size of the affected text, and the SMALL element decreases it.

- The SUP and SUB elements are used, respectively, to create superscripts and subscripts.

Example program

```
<HTML>
  <HEAD>
    <TITLE>HTML </TITLE>
  </HEAD>
  <BODY>
<big> biggest text</big>
<br>
<small>smallest text</small>
<br>
H<sub>2</sub>o
<br>
E=mc<sup>2</sup>
</BODY>
</HTML>
```

biggest text
smallest text
H_2o
$E=mc^2$

The Q and BLOCKQUOTE Elements

- The Q and BLOCKQUOTE elements very similar functions; they are both used to denote quoted material.
- Both Q and BLOCKQUOTE use the cite attribute to provide a URL that points to the online source (if any) for the quotation.

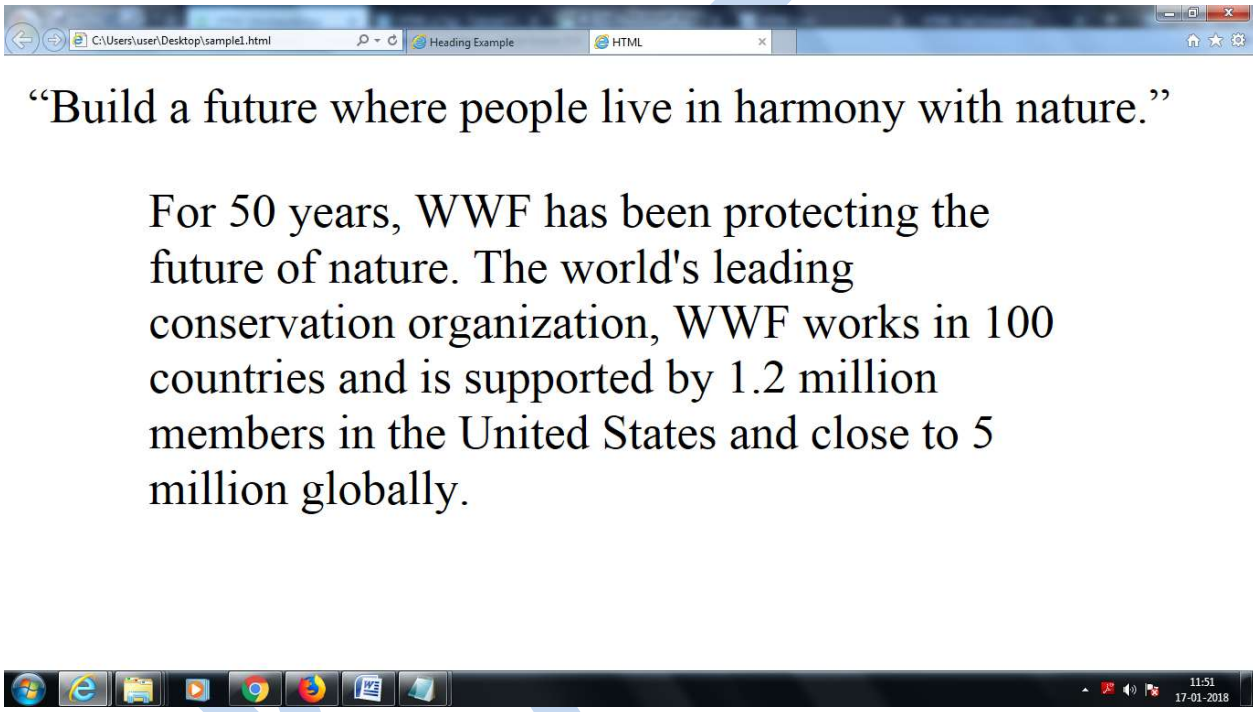
```
<HTML>
  <HEAD>
    <TITLE>HTML </TITLE>
  </HEAD>
  <BODY>
<q>Build a future where people live in harmony with nature.</q>
<br>
 <blockquote cite="http://www.worldwildlife.org/who/index.html">
```

For 50 years, WWF has been protecting the future of nature. The world's leading conservation organization, WWF works in 100 countries and is supported by 1.2 million members in the United States and close to 5 million globally.

</blockquote>

</BODY>

</HTML>



The INS and DEL Elements

- The <INS> tag defines a text that has been inserted into a document.
- The <DEL> tag defines text that has been deleted from a document.
- Use <DEL> and <INS> to markup updates and modifications in a document.
- Browsers will normally strike a line through deleted text and underline inserted text.

The FONT and BASEFONT Elements

The tag is used for modifying the type of a text, the size and the color.

Use "*size*", "*color*" and "*face*" attributes to personalize text and the <basefont> tag to set the color, size and the style of a whole text.

The font size

Set the size of your font. There are accepted values between 1 (the smallest) and 7(the biggest). The standard value of a text is 3.

This is font size 5

The font color

Set the color of the text

This text is hexcolor #990000

This text is red

Font face

Pick a letter style using the font's face attribute.

This paragraph has had its font formatted with the font's face atribute

Basefont

With the help of the "basefont" tag you can set the default value of the font in your web page. We recommend to specify a "basefont" in case you will use the font in HTML.

<basefont size="2" color="green">

For example

```
<HTML>
  <HEAD>
    <TITLE>HTML </TITLE>
  </HEAD>
  <BODY>
<p><font size="7" face="Georgia, Arial" color="maroon">C</font>ustomize your font to achieve a desired look.</p>
<basefont size="2" color="blue">
<p>This paragraph has had its font formatted with the basefont tag</p>
</BODY>
</HTML>
```

Customize your font to achieve a desired look.

This paragraph has had its font formatted with the basefont tag

KARPAGAM ACADEMY OF HIGHER EDUCATION

(Deemed to be University)
(Established Under Section 3 of UGC Act 1956)
Pollachi Main Road, Eachanari Post, Coimbatore - 641021
(For the candidates admitted from 2015 onwards)

CLASS: III B.COM CA **COURSE NAME: INTERNET AND WEB DESIGN**
COURSE CODE: 15CCU601 **UNIT:II (INTRODUCTION TO HTML)** **BATCH-2015-2018**

S.NO	QUESTIONS	OPTION 1	OPTION 2	OPTION 3	OPTION 4	ANSWER
1	An HTML is simply a _____.	Text file	Text	File	Language	Text file
2	An element is made up of _____ tags.	Start tag	End tag	Both 1 & 2	None	Both 1 & 2
3	The <html>, <head>, <body> tag pairs are used to specify the _____.	General Structure of the document	Structure of the text file	A text	None	General Structure of the document
4	In html <a> stands for _____.	Anchor tag	Angle tag	Alphabet tag	None	Anchor tag
5	Html tags are keywords surrounded by the _____.	Angle brackets	Prompt symbol	Mathematical symbols	None	Angle brackets
6	HTML is a language for describing _____.	Files	Web pages	Tags	All the above	Web pages
7	A Markup language is a set of _____.	Markup tags	End tag	Start tag	Scripting language	Markup tags
8	The Start and End tags are also called the _____ and _____ tags.	Opening, Closing	Open, Prompt	Angle, Brackets	None	Opening, Closing
9	The Html file name must end with _____ extension.	.htm	.html	Both 1 & 2	None	Both 1 & 2
10	The html file is often called a _____.	Html document	Document	Internet explorer	None	Html document
11	HTML mark up tags is usually called _____.	Html tags	Tags	Markup language	None	Html tags
12	The text between the <html> and </html> tags describes the _____.	Web page	Web browser	Internet explorer	None	Web page
13	The text between the and tags is displayed in _____.	Bold font	Break line	New line	All the above	Bold font
14	The text between the <body> and </body> tags is displayed in the _____.	Web browser	Web page	Paragraphs	None	Web browser
15	Headings are defined with the <h1> to _____ tags.	<h6>	<h7>	<h5>	<h8>	<h6>
16	The heading, <h7> defines the _____ of the heading.	Largest font	Smallest font	Middle font	None of these	None of these
17	The tag defines a _____.	New line	Line break	Both 1 & 2	Bold	Both 1 & 2
18	The tag is a _____.	Empty tag	End tag	Empty line	None	Empty tag

19	Comments are _____ by the browser.	Ignored	Displayed	Defined	None	Ignored
20	Comments _____ inserted in html code.	Can be	Cannot be	Either 1 or 2	None	Can be
21	Which of the following element has no end tags?	<p>	<body>	 	<h1>	
22	The html element defines the _____ of the document.	Whole	Part	Body	None	Whole
23	Html tags are _____.	Case sensitive	Not case sensitive	Lower case tags	None	Not case sensitive
24	Html Attributes are always specified in the _____ tag.	Start	End	Paragraph	Comment line	Start
25	Attribute values are _____.	Case sensitive	Not case sensitive	Additional information	None	Case sensitive
26	The _____ tag defines how to divide the window into frames.	Frameset	Table	Frame	None	Frameset
27	Which of the following defines a sub window?	Frames	Frames set	Table	. None	Frames
28	Find the odd man out.	Frames	Table	Tags	.hm	.hm
29	The letters “td” stands for _____.	Table data	Tags	Tables	Table definition	Table data
30	Headings in the table are defined with the _____ tag.	<th>	td	Start	<h1>	<th>
31	To avoid empty cells we specify _____.	( )	Hyphen	Null character	None	( )
32	<th> specifies a _____.	Table header	Table heading	t head	Table	Table header
33	tag stands for _____.	Underline	Unordered lists	Bullets	None	Unordered lists
34	Find the odd man out.	Check boxes	Text area fields	Radio buttons	Choices	Choices
35	A form is defined with the _____.	<html>	<form>	Both 1 & 2	None	<form>
36	_____ is used select one or more options.	Check boxes	Radio buttons	Option buttons	None	Check boxes
37	Some html elements have no _____.	Content	Source	Attributes	All the above	Content
38	The image tag ends with the _____	<img>	</img>	</image>	None of these	None of these
39	SRC stands for _____.	Source	Attribute	Server	None	Source
40	The <map> tag defines a _____.	image map	map	map image	images	map
41	Which one of the following is difficult to read?	Background color, text color	text color	improper statements	all the above.	Background color, text color

42	World Wide Web consortium stands for _____.	W3C	3WC	CW3	C3W	W3C
43	Choose the correct HTML tag for the largest heading?	<h6>	<Head>	<h1>	<Heading>	<h1>
44	What is the correct html for adding a background color?	<body color=”yellow”>	<Background>yellow</background>	<body bgcolor=”yellow”>	None	<body bgcolor=”yellow”>
45	Choose the correct HTML tag to make a text bold?	<bld>	<Bold>	<Strong>	Both 2 & 3	Both 2 & 3
46	Choose the correct HTML tag to make a text bold?	<b>	<Bold>	<Strong>	All the above	All the above
47	How can you make a list that lists the items with numbers?	<List>			<Dl>	
48	_____ are used to select different kinds of user input.	HTML forms	HTML table	HTML lists	None	HTML forms
49	In HTML, Comments lines are denoted by_____.	<!-- & -->	<!- & - >	!-- & ->	! & !	<!-- & -->
50	Joint Photographic elements refer to _____.	.jpg	.jpeg	both 1 & 2	None	both 1 & 2
51	The Basic Structural Elements in HTML are _____.	DOCTYPE	BODY	HEAD	All the above	All the above
52	What is the correct tag for inserting a line break?	 	<lb>	<break>	<none>	
53	_____ element will follow the head tag.	Body	Title	Both 1 & 2	None	Body
54	Table values are displayed in a _____ format.	Tabular	Frames	Both 1 & 2	None	Tabular
55	For a <p> tag the end tag is required or not?	Required	Not Required	Optional	None	Optional
56	Choose the correct HTML tag to make a text italic?	<ii>	<i>	<Italic>	<it>	<i>
57	Choose the correct HTML tag to make a text underline?		<u>	<und>	<underline>	<u>
58	The base element specifies an absolute _____.	URL address	IP address	TCP address	None	URL address
59	How can you make a list that lists the items with bullets?	<List>			<Dl>	
60	How can you make an e-mail link?	<a href=mailto:XXX@YYY>	<a href = “XXX@YYY”>	<mail>XXX@YYY</mail>	<mail href =”XXX@YYY”>	<a href=mailto:XXX@YYY>

UNIT III

SYLLABUS

Using images with HTML – Merging Multimedia, Controls, and Plug-ins with HTML – Using the HTML Object Model and Creating Dynamic HTML Pages – Manipulating Objects and Responding to User Interaction.

Embedded Images within an HTML Document

- Images are embedded within a Web document page use of the IMG tag.
- This tag contains the source of the image file and other information and does have an end tag.
- The images are commonly GIF or JPEG files, though new image formatting specification the PNG format is now supported.

Image Format

GIF: Graphic Interchange Format uses a maximum of 256 colors, and uses combinations of these to create colors beyond that number. The GIF format is best for displaying images that have been designed using a graphics program, like logos, icons, and buttons.

GIF images come in two different versions and have some extra functionality that JPEGs do not. You can save GIF images in GIF 87 or GIF 89a format. GIF 89a is newer and has the fol owing features that GIF 87 and JPEG files do not:

JPG, JPEG: Joint Photographic Expert Group, is a good format for photographs because JPEG files can contain millions of colors.JPEG images doesn't give you the option of including transparency or of interlacing images, but they do allow you to specify the degree of file compression so that you can create a balance between

image quality and file size. A new addition to the JPEG format is Progressive JPEG. Pro-JPEGs boast superior compression to regular JPEGs. They also give you a wider range of quality settings. Pro-JPEGs also support.

Limited Support or Non-Supported Image Formats

PNG: Portable Network Graphics, is good for combinations of text and graphics within one image. PNG permits truecolor images, variable transparency, platform-independent display, and a fast 2D interlacing scheme. Currently only supported by Internet Explorer.

BMP: MS Windows BitMaP**TIFF:** Tagged Image File Format

PCX: Originally developed by ZSOFT for its PC Paintbrush program, PCX is a graphics file format for graphics programs running on PCs.

Images are a staple of any web designer, so it is very important that you understand how to use them properly. In order to place an image onto a website, one needs to know where the image file is located within the file tree of the web server -- the URL (Unified Resource Locator).

Use the <img /> tag to place an image on your webpage. Like the <br /> tag, <img /> tag does not require a formal ending tag. Instead, all we need to do to close this tag out with a slash (/) placed just inside the ending bracket (/>).

Img Attributes

Images are embedded within a web page using the IMG tag.

<IMG src="filename.gif" align=left/right width=XXX height=XXX alt="text">

For example

<IMG src= “Penguins.jpg” width=100 height=100 alt=”penguins” hspace=20>

An image source value is essentially the URL of a picture file and tells the web browser where the image is located so that it can then display the image correctly.

Attribute	Value	Description
align	top bottom	Specifies the alignment of an image according to surrounding elements
alt	text	Specifies an alternate text for an image
border	pixels	Specifies the width of the border around an image
crossorigin	anonymous use-credentials	Allow images from third-party sites that allow cross-origin access to be used with canvas
height	pixels	Specifies the height of an image
hspace	pixels	Specifies the whitespace on left and right side of an image
ismap	ismap	Specifies an image as a server-side image-map
longdesc	URL	Specifies a URL to a detailed description of an image

sizes		Specifies image sizes for different page layouts
src	URL	Specifies the URL of an image
srcset	URL	Specifies the URL of the image to use in different situations
usemap	#mapname	Specifies an image as a client-side image-map
vspace	pixels	Specifies the whitespace on top and bottom of an image
width	pixels	Specifies the width of an image

Set Image Location

Usually we keep all the images in a separate directory. So let's keep HTML file test.htm in our home directory and create a subdirectory images inside the home directory where we will keep our image test.png.

Example

Assuming our image location is "image/test.png", try the following example:

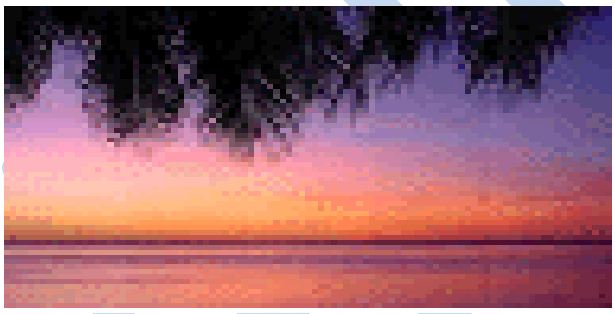
```
<html>
<head>
<title>Using Image in Webpage</title>
</head>
<body>
<p>Simple Image Insert</p>
```

```
<img src="images/test.png" alt="Test Image" />
</body>
</html>
```

IMG Width and Height Attributes

The IMG width and Height of the image define the area of the window the image occupies, not necessarily the size of the image.

```
<img src="sunset.gif" height="50" width="100" />
```



Alt Attribute

The *alt* attribute specifies alternate text to be displayed if for some reason the browser cannot find the image, or if a user has image files disabled in their web browser settings. Text-only browsers greatly depend on the *alt* attribute since they are not capable of displaying pictures.

```
<img src="http://example.com/brokenlink/sunset.gif" alt="Beautiful Sunset" />
```

The hspace, vspace and align attributes

- The white space attributes, hspace and vspace.

- To 10 pixels of horizontal space to surround the image use the img tag attribute hspace="10", and to 15 pixels of vertical space use vspace="15" as here:

```
<img src="petrock.jpg" align="left" hspace="10" vspace="15">
```

The text will appear to the right of the image. It should look less cramped than the text on the previous page because the IMG tag used the hspace="10" and vspace="15" attributes.

Now there are 10 pixels of space between the right edge of the image and the start of the text and the right. And there are 15 pixels of space between the bottom edge of the image and the paragraph that follows it.

The align attribute , which determines how the image is positioned with respect to surrounding HTML elements.

Attribute values are:

Left-align image to left of other elements

Right align image to right of other elements

Top- align image top elements

Center- align image to center of other elements

Bottom- align image to bottom of other elements

Applying Styles and Borders

- A border is created around an image when the image is surrounded by a hypertext link.
- This can be controlled by setting attribute to a value of zero (o).
- If an image is created outside a link , to create a border by setting the border value to an integer representing the size.

<img src= “subshine.jpg” width=100 height=100 border=6>

Border with style

Use the border-radius property to create rounded images:

```
<html>
<head>
<style>
img {
   border: 1px solid #ddd;
   border-radius: 4px;
   padding: 5px;
   width: 150px; }
</style>
</head>
<body>
```

```
<h2>Thumbnail Images</h2>
<p>Use the border property to create thumbnail images:</p>
<img src="paris.jpg" alt="Paris" style="width:150px">
</body>
</html>
```

Another Example

```
<html>
<head>
<style>
img {
    border-radius: 50%;
}
</style>
</head>
<body>

<h2>Rounded Images</h2>
<p>Use the border-radius property to create circled images:</p>

<img src="paris.jpg" alt="Paris" width="300" height="300">

</body>
</html>
```

Creating Image Rollover Effects

The use of changing images is to create a rollover effect for images that represent active links .

When the reader moves his or her mouse over the images, the image takes on a different appearance, adding a highlight that provides feedback to the reader.

Create A Rollover Image Effect (Hover Effects)

The code shows how to show a hover effect on Box Shadow.

Event is Mouse over on Image

For Example

<!DOCTYPE html>

```
<html >
<head>
<style type="text/css">
        .box
        {
                width: 150px;
                border: 1px solid #9325BC;
                padding: 10px;
        }
        .box:hover
        {
                -moz-box-shadow: 0 0 10px #ccc;
                -webkit-box-shadow: 0 0 10px #ccc;
                box-shadow: 0 0 10px #ccc;
        }
</style>
</head>
<body>
        <div class="box">
                Move your mouse here
        </div>
</body>
</html>
```

Image Hover Effects (Image Rollovers)

The CSS code shows how to present an Image hover effects.

To Move the mouse over the Image.

```
<!DOCTYPE html>
<html >
<head>
<style type="text/css">
        .imgBox
        {
                width: 441px;
                height: 248px;
                background:
url(http://www.corelangs.com/css/box/img/water.jpg) no-repeat;
        }
        .imgBox:hover
        {
                -moz-box-shadow: 0 0 10px #ccc;
                -webkit-box-shadow: 0 0 10px #ccc;
                box-shadow: 0 0 10px #ccc;
        }
</style>
</head>
<body>
        <div class="imgBox">
```

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

CSS Image Opacity / Transparency

Opacity is the opposite of transparency, allowing no light to pass through. You can create transparent images in CSS with the opacity property .

CSS Image fade effect

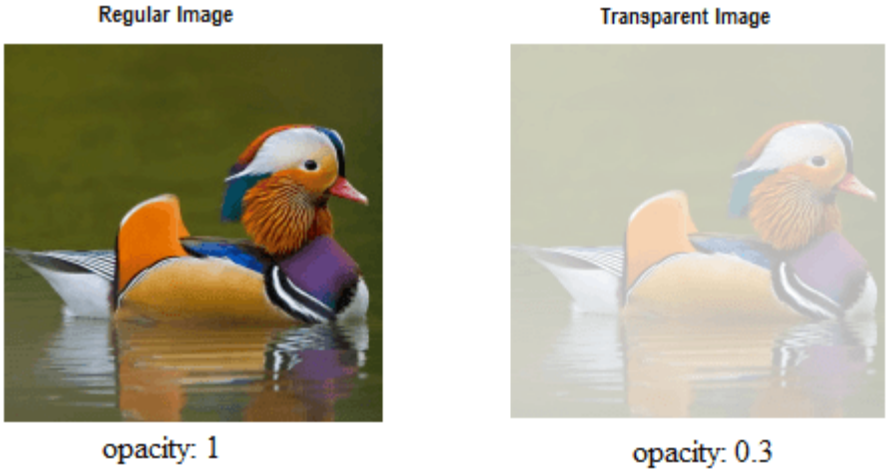


Image Blur effect

```
img
{
    opacity: 0.3;
}
```

Creating transparent/Opacity images - mouse over effect

The CSS program shows how to implement Opacity on an Image while mouse hover. When you mouse over the image, you can see the image is fading.

Image Opacity on Hover

Move your mouse over the Image.

Source Code

```
<!DOCTYPE html>
<html >
<head>
<style type="text/css">
.pic{
        width:190px;
        height:190px;
        opacity: 1;
        filter: alpha(opacity=100);
        background:  url(http://www.corelangs.com/css/box/img/duck.png)  no-
repeat;
}
.pic:hover
{
        opacity: 0.3;
        filter: alpha(opacity=30);
}
```

```
</style>
</head>
<body>
        <div class="pic">
        </div>
</body>
</html>
```

Text Overlay on Image Hover

Layering images over one another is a great way to give an Image a new Look. Here, the CSS code that display text on image while mouse hover

Move your mouse over the Image.

The tiger is the largest cat species, reaching a total body length of up to 3.3 m and weighing up to 306 kg. Its most recognizable feature is a pattern of dark vertical stripes on reddish-orange fur with a lighter underside.

Source Code

```
<!DOCTYPE html>
<html >
<head>
                <style type="text/css">
.pic{
```

```
        width:400px;
        height:217px;
        background: url(http://www.corelangs.com/css/box/img/tiger.png) no-
repeat;
}
.text{
        width:340px;
        height:217px;
        background:#FFF;
        opacity:0;
}
.pic:hover .text
{
        opacity:0.6;
        text-align:justify;
        color:#000000;
        font-size:20px;
        font-weight:700;
        font-family:"Times New Roman", Times, serif;
        padding:30px;
}
                        </style>
</head>
<body>
<div class="pic">
  <div class="text">
        The tiger is the largest cat species, reaching a
```

total body length of up to 3.3 m and weighing up
to 306 kg. Its most recognizable feature is a pattern
of dark vertical stripes on reddish-orange fur
with a lighter underside.

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

CSS hover image swap - onmouseover Event

The following CSS code shows how to change an Image on mouse over. On mouse hover one div containging image is placed over the other image.

CSS Ful Screen overlay

```
<!DOCTYPE html>
<html >
<head>
                <style type="text/css">
.imgBox
{
        width: 191px;
        height: 191px;
        background:  url(http://www.corelangs.com/css/box/img/duck.png)  no-repeat;
}
.imgBox:hover {
        width: 191px;
```

```
        height: 191px;
        background: url(http://www.corelangs.com/css/box/img/peng.png) no-repeat;
        }
                </style>
</head>
<body>
<div class="imgBox">
</div>
</body>
</html>
```

Div Hover

We can change the background color of a Div on Hover. The following CSS code shows how to change div background color on hover.

Move your mouse over the Div.

CSS Div hover effects

Source code

```
<!DOCTYPE html>
<html >
<head>
<style type="text/css">
        .picColor{
                width:320px;
                height:240px;
                border:5px solid #000000;
        }
        .picColor:hover
        {
                background:#FF2400;
                width:320px;
                height:240px;
        }
</style>
</head>
<body>
        <div class="picColor">
        </div>
</body>
</html>
```

CLIENT-SIDE IMAGE MAPS

Image Maps

- An image map is positional information XHTML and HTML which has details of coordinates related to a unique image.
- Unlike a normal image link where the entire area of image is linked to a single destination, an image map is created to hyperlink sections in image to different destinations.
- Image maps provide a convenient way of linking different sections of an image without the need to create image files for the image.
- An image map is also known as a clickable image map.
- Image maps are used on both the server side and client side.
- Server-side image maps aid the web browser in sending the positional details to the server when the user clicks a section of an image.
- It also helps the server to determine pixel-by-pixel content and what it needs to return in response to user actions.
- Client-side image maps do not need any special logic to be executed on the server or JavaScript.

Client-Side Image Maps Vs Server-Side Image Maps.

The two principal types of image maps are client-side image maps and server-side image maps.

With a server-side image map, when a user clicks on a part of the map, the vertical and horizontal coordinates of the cursor location are transmitted back to the web page's server, which then calculates what region of the map those coordinates belong to and responds appropriately.

With the newer client-side type of image map, the user's browser determines which region the user has selected.

Client-side image maps are generally more efficient because the web server is not involved in determining what a click on the map means.

More important for accessibility, with client-side image maps the browser can "know" what the various regions of the map mean, i.e. what link or script they activate, and can therefore pass this information to the user *before* the link or script is activated.

To make client-side image maps accessible, each region within the map can be assigned an alt description that can be read by a screen reader or other assistive technology device.

The same cannot be said of server-side image maps. Server-side image maps must be accompanied by redundant text links.

For instance, in the example above, the agency has made the image map accessible by providing a duplicate list of links - here, the names of states - for each of the areas identified in each region of the map.

To determine whether a web page includes client-side image maps, you can review the page's HTML source coding for any tags that begin with the word map. For instance, <MAP NAME="usemap">.

Server-side image maps are slightly more difficult to identify within the page's HTML source coding, because there are several tags that will send the horizontal and vertical coordinates of the cursor location back to the server.

Some indications that a web page includes a server-side image map are the presence of the ISMAP attribute in an image tag or a form input tag specifying a parameter of IMAGE.

If the HTML source code for the page includes tags of one of the following formats, then the page may include a server-side image map:

<IMG SRC="picture.jpg" ISMAP> or <INPUT TYPE="image">

HTML Elements Used to Create Image Maps

There are three HTML elements used to create image maps:

- img: specifies the location of the image to be included in the map.
- map: is used to create the map of clickable areas.
- area: is used within the map element to define the clickable areas.

A server-side image maps: is enabled by the *ismap* attribute for the <img> tag and requires access to a server and related image-map processing applications.

A client-side image maps: is created with the *usemap* attribute for the <img> tag, along with corresponding <map> and <area> tags.

MERGING MULTIMEDIA, CONTROLS AND PLUG-INS WITH HTML

Accessing sound and Video in Web Pages

- Multimedia was first used in WebPages to include sound.
- Both Netscape Navigator and Internet Explorer support the use of sound in a Web page.
- Beginning with Netscape's Navigator version 3.x, the browser used the Live-audio plug-in for sound, and the LiveVideo plug-ins for video.
- To implement sound files in different ways.

Using the <EMBED> Tag

•

The EMBED element is a simple-to-use technique that includes multimedia in a web page.

•

Both IE and Navigator show a visible control in a web page if the following embedded tag syntax is used:

<EMBED src="example.mid" controls=console width=144 height=60>

•

Each browser determines what plug-ins are currently installed to support files with the extension type of .mid and the browser then uses the associated plug-in.

•

Netscape will most likely use the Live-Audio plug-in, and microsoft use the DirectShow control.

•

The EMBED element can also be used to support video files.

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

```
<head>
   <title>HTML embed Tag</title>
</head>

<body>
   <embed src = "/html/yourfile.mid" width = "100%" height = "60" >
      <noembed><img src = "yourimage.gif" alt = "Alternative Media"
></noembed>
   </embed>
</body>

</html>
```

The <embed> Tag Attributes

Following is the list of important attributes which can be used with <embed> tag.

r.No	Attribute & Description
1	align Determines how to align the object. It can be set to either center, *left or right*.
2	autostart This boolean attribute indicates if the media should start automatically. You can set it either true or false.
3	loop Specifies if the sound should be played continuously (set loop to true), a certain number of times (a positive value) or not at all (false)

4	playcount Specifies the number of times to play the sound. This is alternate option for *loop* if you are usiong IE.
5	hidden Specifies if the multimedia object should be shown on the page. A false value means no and true values means yes.
6	width Width of the object in pixels
7	height Height of the object in pixels
8	name A name used to reference the object.
9	src URL of the object to be embedded.
10	volume Controls volume of the sound. Can be from 0 (off) to 100 (full volume).

Video Types

You can use various media types like Flash movies (.swf), AVI's (.avi), and MOV's (.mov) file types inside embed tag.

- **.swf files** − are the file types created by Macromedia's Flash program.
- **.wmv files** − are Microsoft's Window's Media Video file types.
- **.mov files** − are Apple's Quick Time Movie format.
- **.mpeg files** − are movie files created by the Moving Pictures Expert Group.

Live Demo

```
<!DOCTYPE html>
<html>

   <head>
      <title>HTML embed Tag</title>
   </head>

   <body>
      <embed src = "/html/yourfile.swf" width = "200" height = "200" >
         <noembed><img src = "yourimage.gif" alt = "Alternative Media"
></noembed>
      </embed>
   </body>

</html>
```

Netscape's LiveAudio Plug-In

The LiveAudio plug-in play files with extensions of .wav, .aiff, .au and .mid.

The LiveAudio plug-in attributes are:

SRC - Audio Source File URL

AUTOSTART - Whether file starts playing as soon as it is loaded.

LOOP - Whether to loop sound file

Starttime - specifying a time in the source code to begin playing, specified in Minutes:Seconds

EndTime - Specifying a time in the source code to end playing

Volume - A value between 0 and 100, representing percentage of sound volume.

Width - Width of display

Height - height of display

Netscape's LiveAudio Plug-In

The methods that can be called from script for the LiveAudio plugin are the following:

Play

Stop

Pause

Start-time

End-time

Setvol

Isready

Isplaying

Ispaused

Getvolume

The <OBJECT> tag

The object element is used for embedding generic content into a web page. The object element has gained favor because of its use of the PARAM element, embedded within the beginning and ending <object> tags, that provides name-value pairs as a method of passing attributes to the embedded element. The attributes specific to the OBJECT element are:

CLASSID - Location of an object's implementation

CODEBASE - Path used to resolve URL references

CODETYPE - Internet media type

DATA - Location of data rendered by control

STANDBY - message to display while control is downloading

The <OBJECT> tag and the Future

The attributes of PARAM element are:

Name

Value

Valuetype Example

<object id="thename" classid="CLSID:05589FA1
-C356-11CE-BF01-
00AA0055595A">
<param name="filename" value="f.avi">
</object>

Embedding Sound and Video into an Html Page

With DirectShow, a multimedia file can be inserted into a page using a hypertext link, embedded into the page using the <embed> tag, or embedded as an ActiveX control using the <object> tag.

Three different techniques of using DirectShow to play a .mid file <html><body>
 Song

<object id="themusic" classid="CLSID:05589FA1
-C356-11CE-BF01-
00AA0055595A">

<param name="filename" value="f.avi">

</object>

<EMBED SRC="EXAMPLE.MID">

The DirectShow object Properties, Events & Methods

Properties are:

Filename - url of multimedia source file

AllowHideControls - whether the web page reader can hide or display controls

Appearance

whether the control has an inset border or flat border

Autostart - starts the multimedia stream play as soon as the control loads

Bordestyle - controls appearance of border

Currentstate - whether control is stopped, playing or paused.

Enabled - whether the control is enabled

Fullscreenmode - to display video fullcreen

Playcount - number of times to loop playback

Readystate - controls readiness state

Volume – controls sound volume

The DirectShow object Properties, Events & Methods

Methods:

AboutBox - version and copyright information about the control

IsSoundCardEnabled - whether sound card is installed and enabled

Pause - pause playback

Run - run multimedia stream

Stop - stop stream playback

The DirectShow object Properties, Events & Methods

Events:

Displaymodechange - when display mode property is changed

Error - when an error occurs

openComplete - when source code has finished loading

PositionChange - When media position is changed

StateChange - player state changes

Timer - for timing events

Microsoft's Direct Animation Technology

DirectAnimation is a set of objects, a set of built-in controls, and an API that are accessible from script, java, C++ or other code.

With these sets you can integrate the use of scripting, java and controls. The DirectAnimation controls are

1. Structured Graphics - creates a 2D images
2. Path – controls the movement of any object
3. Sequencer - controls and synchronizes the actions of several different elements
4. Sprite - controls the animation frames

Internet Explorer Built-In Filters

Microsoft created the css1 style visual filters to control the appearance of ordinary HTML elements.

The visual filter effects can do such things as make an element semi-transparent, rotate an image, remove colors, or add specific lighting effects to a page.

Visual filter effects:

1. Glow - adds a glow outlining the element

2. Gray – drops the element's color palette
3. Light - creates a light source on a page
4. Shadow - creates solid shadow

When and when not to use Multimedia

There are some limitations to using multimedia. Restrict download sizes If you want to include the use of video or sound, such as company theme song, you might want to consider placing these on a separate page so these bandwidth - hogging files do not impede your reader's access to other information.

Copyright Information

If you don't want your image, video or song saved and used by another site, you must make sure that all copyright information on the multimedia file is displayed prominently on the same page the file is accessed from.

Speaking of copyright, do not ever copy and reuse images, videos, or sounds that are not public domain and that you do not have permission to use.

Cross- Browser Compatibility

Suppose we want to use one browser's specific technique, but our page is accessed by both of these popular browsers, use the browser-specific technique to enhance an effect, not supply the entire effect.

Cross- Platform Compatibility Issues

- Not all multimedia files play equally well on all platforms.
- The rule of thumb is to deliver content to the lowest common denominator
  - what works for all platforms

– or to provide different links to different multimedia types, and let the reader choose which format to use.

USING THE HTML OBJECT MODEL AND CREATING DYNAMIC HTML PAGES

Dynamic HTML

Dynamic HTML is a collective term for a combination of Hypertext Markup Language (HTML) tags and options that can make Web pages more animated and interactive than previous versions of HTML.

Much of dynamic HTML is specified in HTML 4.0. Simple examples of dynamic HTML capabilities include having the color of a text heading change when a user passes a mouse over it and allowing a user to "drag and drop" an image to another place on a Web page.

Dynamic HTML can allow Web documents to look and act like desktop applications or multimedia productions.

- The Concepts and Features in Dynamic HTML
- An object-oriented view of a Web page and its elements
- Cascading style sheets and the layering of content
- Programming that can address all or most page elements
- Dynamic fonts

An Object-Oriented View of Page Elements

Each page element (division or section, heading, paragraph, image, list, and so forth) is viewed as an "object." (Microsoft calls this the "Dynamic HTML Object Model."

Netscape calls it the "HTML Object Model." W3C calls it the "Document Object Model.")

For example, each heading on a page can be named, given attributes of text style and color, and addressed by name in a small program or "script" included on the page.

This heading or any other element on the page can be changed as the result of a specified event such a mouse passing over or being clicked or a time elapsing Or an image can be moved from one place to another by "dragging and dropping" the image object with the mouse. (These event possibilities can be viewed as the reaction capabilities of the element or object.)

Any change takes place immediately (since all variations of all elements or objects have been sent as part of the same page from the Web server that sent the page). Thus, variations can be thought of as different properties of the object.

Style Sheets and Layering

A describes the default style characteristics (including the page layout and font type style and size for text elements such as headings and body text) of a document or a portion of a document.

For Web pages, a style sheet also describes the default background color or image, hypertext link colors, and possibly the content of page.

Style sheets help ensure consistency across all or a group of pages in a document or a Web site.

Dynamic HTML includes the capability to specify style sheets in a "cascading style sheet" fashion (that is, linking to or specifying different style sheets or style

statements with predefined levels of precedence within the same or a set of related pages).

As the result of user interaction, a new style sheet can be made applicable and result in a change of appearance of the Web page.

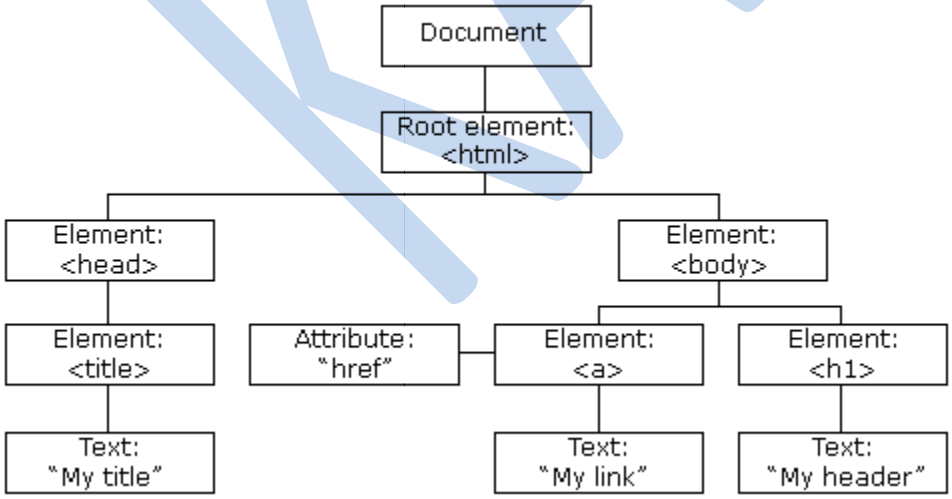
We can have multiple layers of style sheet within a page, a style sheet within a style sheet within a style sheet.

A new style sheet may only vary one element from the style sheet above it.

HTML Document Object Model

The HTML Document Object Model (HTML DOM) defines a standard way for accessing and manipulating HTML documents.

The DOM presents an HTML document as a tree-structure (a node tree), with elements, attributes, and text.



What is the DOM?

The DOM is a W3C (World Wide Web Consortium) standard.

The DOM defines a standard for accessing documents like HTML and XML:

"The W3C Document Object Model (DOM) is a platform and language-neutral interface that allows programs and scripts to dynamically access and update the content, structure, and style of a document."

The DOM is separated into 3 different parts / levels:

- Core DOM - standard model for any structured document
- XML DOM - standard model for XML documents
- HTML DOM - standard model for HTML documents

The DOM defines the objects and properties of all document elements, and the methods (interface) to access them.

What is the HTML DOM?

The HTML DOM is:

- A standard object model for HTML
- A standard programming interface for HTML
- Platform- and language-independent
- A W3C standard

The HTML DOM defines the objects and properties of all HTML elements, and the methods (interface) to access them.

HTML DOM Event Object

Events

By using JavaScript, we have the ability to create dynamic web pages. Events are actions that can be detected by JavaScript.

Every element on a web page has certain events which can trigger JavaScript functions. 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 box in an HTML form
- Submitting an HTML form
- A keystroke

Note: Events are normally used in combination with functions, and the function will not be executed before the event occurs!

onload and onUnload

The onload and onUnload events are triggered when the user enters or leaves the page.

The onload event is often used to check the visitor's browser type and browser version, and load the proper version of the web page based on the information.

Both the onload and onUnload events are also often used to deal with cookies that should be set when a user enters or leaves a page.

For example, you could have a popup asking for the user's name upon his first arrival to your page.

The name is then stored in a cookie. Next time the visitor arrives at your page, you could have another popup saying something like: "Welcome John Doe!".

onFocus, onBlur and onChange

The onFocus, onBlur and onChange events are often used in combination with validation of form fields.

Below is an example of how to use the onChange event. The checkEmail() function will be called whenever the user changes the content of the field:

```
<input type="text" size="30"
id="email" onchange="checkEmail()">
```

onMouseOver and onMouseOut

onMouseOver and onMouseOut are often used to create "animated" buttons.

Below is an example of an onMouseOver event. An alert box appears when an onMouseOver event is detected:

```
<a href="http://www.w3schools.com"
onmouseover="alert('An onMouseOver event');return false">
<img src="w3schools.gif" width="100" height="30">
</a>
```

Manipulating objects and responding to user interaction

The Life Cycle of an Event

An event has a life cycle that begins with the action or condition that initiates the event and ends with the final response by the event handler or Internet Explorer. The life cycle of a typical event consists of the following steps.

1. The user action or condition associated with the event occurs.
2. The **event** object is instantly updated to reflect the conditions of the event.
3. The event fires. This is the actual notification in response to the event.
4. The event handler associated with the source element is called, carries out its actions, and returns.
5. The event bubbles up to the next element in the hierarchy, and the event handler for that element is called. This step repeats until the event bubbles up to the window object or a handler cancels bubbling.
6. The final default action, if any, is taken, but only if this action has not been canceled by a handler.

POSSIBLE QUESTIONS

PART A (1 Mark)

(Online Examinations)

PART B (8 Marks)

1. Explain in detail about the multimedia plug-ins.
2. Write about the creation of static HTML object.
3. Explain in detail about the image formats and image attributes.
4. Explain in detail the Netscape live Audio plug –in.
5. How will you create an image rollover effects in HTML? Describe.
6. Describe the image element attribute with examples.
7. Elucidate the four elements of Table with example.
8. Write the basic Text Layout with HTML Elements.
9. What are the HTML object models available? Explain with example.
10. Write about new Netscape dynamic HTML object.

KARPAGAM ACADEMY OF HIGHER EDUCATION

(Deemed to be University)
(Established Under Section 3 of UGC Act 1956)
Pollachi Main Road, Eachanari Post, Coimbatore - 641021
(For the candidates admitted from 2015 onwards)

CLASS: III B.COM CA **COURSE NAME: INTERNET AND WEB DESIGN**
COURSE CODE: 15CCU601 **UNIT: III (Using Images with HTML)** **BATCH-2015-2018**

S.NO	QUESTIONS	OPTION 1	OPTION 2	OPTION 3	OPTION 4	ANSWER
1	Images are embedded within a web document page with the use of _____.	Img tag	text file	http	none	Img tag
2	Does the image tag has an end tag?	yes	no	optional	none	no
3	When the mouse is over the image, the function refers to _____.	change-high	change-low	change	none	change-high
4	When the mouse leaves the image area, the function refers to _____.	change-high	change-low	change	none	change-low
5	_____ was the first used in web pages to include sound.	multimedia	Netscape navigator	internet explorer	all the above	multimedia
6	Passing a Boolean value _____ the sound playing.	begins	ends	pause	none	begins
7	Which of these attributes will specify “location of an objects implementation?	class id	data	code type	code base	class id
8	Which of the following options are object tag attributes?	class id	code base	code type	All the above	All the above
9	The direct animation controls are:	Structured graphics	Path	Sprite	All the above	All the above
10	World Wide Web consortium stands for _____.	W3C	3WC	CW3	C3W	W3C
11	The html element defines the _____ of the document.	Whole	Part	Body	None	Whole
12	_____ are used to select different kinds of user input.	HTML forms	HTML table	HTML lists	none	HTML forms
13	What is the correct HTML for making a text area?	<input type = “text area”>	<input type = “text box”>	<text area>	none	<text area>
14	How many different layers are in HTML?	2	3	4	6	2
15	The event that captures one or more event and routes to object is _____.	Capture event	Route event	Release event	none	Capture event
16	For every web page we create, it must have_____.	<html>	<head>	<title> and <body> tags	All the above	All the above
17	Which of these functions will encloses the entire HTML document?	<html>…</html>	<body>…</body>	<hr>	both 1 & 2	<html>…</html>
18	Which of these is the fourth level heading?	H1	H2	H3	H4	H4

19	The tag which creates a link is called?	<a> tag	Strong tag	Bold tag	none	<a> tag
20	Which of these specifies the address of the document?	href	<a>	both 1 & 2	none	href
21	Which of these specifies belongs to HTML lists?	Unordered lists	Ordered lists	Definition lists	All the above	All the above
22	In html, another name for <bold> tag is _____.	Strong tag	Big tag	H1	none	Strong tag
23	In which of these tag BGCOLOR we can specify_____.	<body> tag	<head> tag	<title> tag	<p> tag	<body> tag
24	In image tag, “the address of the image” specifies _____.	src	http	Location of the directory	none	src
25	SRC stands for _____.	Source	Screen	Scream	none	Source
26	_____ is the arrangement of text /graphics into vertical columns and horizontal rows.	Table	Frames	Both 1 & 2	none	Table
27	Does the table tag have the end tag?	yes	no	optional	none	yes
28	Find the odd man out?	<tr>	<th>	<td>	<hr>	<hr>
29	The amount of space between the cells in the table refers to _____.	cell spacing	cell padding	table	cells	cell spacing
30	To display long quotations _____ is used.	Block quote	Quote	Single quote	none	Block quote
31	To display short quotations _____ is used.	block quote	quote	Single quote	none	quote
32	Which of these creates a definition lists?	<dl>	<dt>	<tr>	<hr>	<dl>
33	Which of these creates a definition term?	<dl>	<dt>	<tr>	<hr>	<dt>
34	“Used to define links and anchors” refers to _____.	Anchor tag	Link tag	Bold tag	frames	Anchor tag
35	The abbreviation for URL is _____.	Uniform Resource Locator	Uniform Relay Locator	Unique Resource Locator	Universal Resource Locator	Uniform Resource Locator
36	The <th> tag specifies the _____.	table heading	table footer	table row	table	table heading
37	The method for copying a file, from one computer to another through the internet is ?	FTP	TCP	IP	none	FTP
38	The abbreviation for http is _____.	hypertext transfer protocol.	hyper tool transfer protocol.	hypertext transfer process.	hypertext tag protocol.	hypertext transfer protocol.
39	A computer network limited to a small area known as _____.	LAN	WAN	Both 1 & 2	None	LAN
40	All HTML tags are enclosed with an _____.	angle brackets < >	square brackets [ ]	Set braces { }	Simple Bracket ( )	angle brackets < >
41	To create a bulleted list we need to add an _____.	tag	tag	tag	none	tag

42	The bullet option takes the form _____.	Disc	Circle	Square	All the above	All the above
43	_____ are the most fundamental part of the WWW.	Links	Web search	Text area	Images	Links
44	Find the odd man out?	List	Link	Tables	Script	Script
45	DOCTYPE information appears on the _____ of an HTML document.	first line	bottom of the line	middle of the line	title bar	first line
46	SGML stands for _____.	Standard Generalized Markup Language	Standard General Markup Language	Source General Markup Language	none	Standard Generalized Markup Language
47	_____ is the element to denote a citation		<cite>		none	<cite>
48	The element showing subscripts refers to _____.	<sup>	<sub>	<s>	none	<sub>
49	The element showing superscripts refers to _____.	<sup>	<sub>	<s>	none	<sup>
50	_____ is the tag used to emphasize the text.	<emphasize>	<emp>		none	
51	Which is the correct CSS syntax?	body {color:black}	{body;color:black}	{body:color=black)body}	body:color=blak	body {color:black}
52	_____ is used to insert a comment in the HTML source code.	head tag	title tag	The comment tag	none	The comment tag
53	_____ is the HTML attribute which defines the body of an HTML document.	<head>	<body>	<title>	none	<body>
54	Choose the correct HTML tag to make a text bold?	<bld>	<bb>	<b>	none	<b>
55	Is HTML tags are surrounded by two characters?	yes	no	optional	none	yes
56	_____ is the tag defines how to divide the window into frames.	<frame>	<frameset>	<framest>	none	<frameset>
57	The _____ attribute specifies a background-image for an HTML page.	background	Bground	Backgrd	none	background
58	Which of these will define the document's body?	<body>	<bg>	<body Doc>	none	<body>
59	_____ element is used to include style information to the browser.	<style>	<body>	<html>	none	<style>
60	_____ is the tag used to define an image	<img>	<image>	<im>	none	<img>

UNIT IV

SYLLABUS

Scripting Basics – Client Side Image Maps – Introducing Java Script – Creating Simple Java Scripts – Using Java Scripts for Forms – Using Java Scripts with Style Sheets.

Client –Side Image Maps

Client side image maps are enabled by the *usemap* attribute for the <img /> tag and defined by special <map> and <area> extension tags.

The image that is going to form the map is inserted into the page using the <img /> element as normal, except it carries an extra attribute called usemap. The value of the usemap attribute is the value of the name attribute on the <map> element, which you are about to meet, preceded by a pound or hash sign.

The <map> element actually creates the map for the image and usually follows directly after the <img /> element. It acts as a container for the <area /> elements that actually define the clickable hotspots. The <map> element carries only one attribute, the name attribute, which is the name that identifies the map. This is how the <img /> element knows which <map> element to use.

The <area> element specifies the shape and the coordinates that define the boundaries of each clickable hotspot. Here's an example from the image map:

<!– Create Mappings –>

```
<map name="html">
<area shape="circle"
coords="154,150,59" href="link1.htm" alt="link 1"
target="_self" />
<area shape="poly"
coords="272,79,351,79,351,15,486,15,486,218,272,218,
292,166,292,136,270,76" alt="link 2"
href="link2.htm" target="_self" />
<area shape="rect"
coords="325,224,488,286" alt="link 3"
href="link3.htm" target="_self" />
</map>
```

The actual value of coords is totally dependent on the shape in question.

Attribute name	Values	Notes
alt		Specifies alternative text for a clickable area in an image map.
coords	values	Defines the shape and size of a clickable area in an image

		map.
href		Defines the URL of the linked document or resource.
nohref		Specified that an area of an image map did not link to another resource.
shape		In conjunction with the coords attribute, specifies the shape, size, and placement of a clickable area in an image map.
target		Specifies the context in which to open the linked resource.
title		Defines the title text of the clickable area. The title text will appear as a tooltip in most browsers.

Server-Side Image Maps:

- To add an image to anchor simply by placing an <img> tag within the body of the <a> tag.
- Make that embedded image into a mouse-sensitive one by adding the *ismap* attribute to the <img> tag.
- This special <img> attribute tells the browser that the image is a special map containing more than one link.
- When the user clicks some place within the image, the browser passes the coordinates of the mouse pointer along with the URL specified in the <a> tag to the document server.
- The server uses the mouse-pointer coordinates to determine which document to deliver back to the browser.

- When *ismap* is used, the href attribute of the containing <a> tag must contain the URL of a server application like amap file or cgi script etc. to process the incoming request based on the passed coordinates.
- The coordinates of the mouse position are screen pixels counted from the upper-left corner of the image, beginning with (0,0).
- The coordinates, preceded by a question mark, are added to the end of the URL.

For example, if a user clicks 50 pixels over and 30 pixels down from the upper-left corner of the image displayed from the following link:

```
<a href="/cgi-bin/logo.map" target="_self" >

<img ismap src="/images/html.gif" alt="HTML" border="0"/>  </a>
```

ISMAP

The ismap attribute is a boolean attribute.

When present, it specifies that the image is part of a server-side image-map (an image-map is an image with clickable areas).

When clicking on a server-side image-map, the click coordinates are sent to the server as a URL query string.

USEMAP

The usemap attribute specifies an image as a client-side image-map (an image-map is an image with clickable areas).

The usemap attribute is associated with a <map> element's name attribute, and creates a relationship between the <img> and the <map>.

Java Scripts - Introduction

JavaScript was designed to 'plug a gap' in the techniques available for creating web-pages.

HTML is relatively easy to learn, but it is static.

It allows the use of links to load new pages, images, sounds, etc., but it provides very little support for any other type of interactivity.

To create dynamic material it was necessary to use either:

- CGI (Common Gateway Interface) programs Can be used to provide a wide range of interactive features, but...Run on the server, i.e.:
  - A user-action causes a request to be sent over the internet from the client machine to the server.
  - The server runs a CGI program that generates a new page, based on the information supplied by the client.
  - The new page is sent back to the client machine and is loaded in place of the previous page.
  - Thus every change requires communication back and forth across the internet.
  - Written in languages such as Perl, which are relatively difficult to learn.

Javascript only runs on Netscape browsers (e.g., Netscape Navigator). However, Microsoft soon developed a version of JavaScript for their Internet Explorer browser. It is called ***JScript***.

The two languages are almost identical, although there are some minor differences.

Internet browsers such as Internet Explorer and Netscape Navigator provide a range of features that can be controlled using a suitable program. For example, windows can be opened and closed, items can be moved around the page, colours can be changed, information can be read or modified, etc..

Simple Java Script Program

```
<html>

<body>

<script type="text/javascript">

document.write("Hello World!")

</script>

</body>

</html>
```

JavaScript Data types

One of the most fundamental characteristics of a programming language is the set of data types it supports. These are the type of values that can be represented and manipulated in a programming language.

JavaScript allows you to work with three primitive data types:

- Numbers, e.g., 123, 120.50 etc.
- Strings of text, e.g. "This text string" etc.
- Boolean, e.g. true or false.

JavaScript also defines two trivial data types, null and undefined, each of which defines only a single value. In addition to these primitive data types, JavaScript supports a composite data type known as object. We will cover objects in detail in a separate chapter.

JavaScript Variables

Like many other programming languages, JavaScript has variables. Variables can be thought of as named containers. You can place data into these containers and then refer to the data simply by naming the container.

Before you use a variable in a JavaScript program, you must declare it. Variables are declared with the **var** keyword as follows.

```
<script type="text/javascript">
<!--
var money; var
name;
//-->
</script>
```

To declare multiple variables with the same **var** keyword as follows:

```
<script type="text/javascript">
<!--
var money, name;
//-->
</script>
```

Storing a value in a variable is called **variable initialization**. You can do variable initialization at the time of variable creation or at a later point in time when you need that variable.

For instance, you might create a variable named **money** and assign the value 2000.50 to it later. For another variable, you can assign a value at the time of initialization as follows.

```
<script type="text/javascript">
<!--
var name = "Ali"; var
money;
money = 2000.50;
//-->
</script>
```

JavaScript is **untyped** language. This means that a JavaScript variable can hold a value of any data type. Unlike many other languages, you don't have to tell JavaScript during variable declaration what type of value the variable will hold. The value type of a variable can change during the execution of a program and JavaScript takes care of it automatically.

JavaScript VariableScope

The scope of a variable is the region of your program in which it is defined. JavaScript variables have only two scopes.

- **Global Variables:** A global variable has global scope which means it can be defined anywhere in your JavaScript code.
- **Local Variables:** A local variable will be visible only within a function where it is defined. Function parameters are always local to that function.

Within the body of a function, a local variable takes precedence over a global variable with the same name. If you declare a local variable or function parameter with the same name as a global variable, you effectively hide the global variable. Take a look into the following example.

```
<script type="text/javascript">
<!--
var myVar = "global"; // Declare a global variable
function checkscope( ) {
    var myVar = "local";     // Declare a local variable
    document.write(myVar);
}
//-->
</script>
```

It will produce the following result:

```
Local
```

JavaScript VariableNames

While naming your variables in JavaScript, keep the following rules in mind.

- You should not use any of the JavaScript reserved keywords as a variable name. These keywords are mentioned in the next section. For example, **break** or **boolean** variable names are not valid.
- JavaScript variable names should not start with a numeral (0-9). They must begin with a letter or an underscore character. For example, **123test** is an invalid variable name but **_123test** is a valid one.
- JavaScript variable names are case-sensitive. For example, **Name** and **name** are two different variables.

JavaScript ReservedWords

A list of all the reserved words in JavaScript is given in the following table. They cannot be used as JavaScript variables, functions, methods, loop labels, or any object names.

abstract	else	Instanceof	switch
boolean	enum	int	synchronized
break	export	interface	this
byte	extends	long	throw
case	false	native	throws
catch	final	new	transient
char	finally	null	true
class	float	package	try
const	for	private	typeof

continue	function	protected	var
debugger	goto	public	void
default	if	return	volatile
delete	implements	short	while
do	import	static	with
double	in	super	

Operator

JavaScript supports the following types of operators.

- Arithmetic Operators
- Comparison Operators
- Logical (or Relational) Operators
- Assignment Operators
- Conditional (or ternary) Operators

Arithmetic Operators

JavaScript supports the following arithmetic operators: Assume variable A holds 10 and variable B holds 20, then:

S. No.	Operator and Description
1	**+ (Addition)** Adds two operands **Ex:** A + B will give 30
2	**- (Subtraction)** Subtracts the second operand from the first **Ex:** A - B will give -10
3	*** (Multiplication)** Multiply both operands **Ex:** A * B will give 200
4	**/ (Division)**
	Divide the numerator by the denominator **Ex:** B / A will give 2
5	**% (Modulus)** Outputs the remainder of an integer division **Ex:** B % A will give 0

6	**++ (Increment)** Increases an integer value by one **Ex:** A++ will give 11
7	**-- (Decrement)** Decreases an integer value by one **Ex:** A-- will give 9

Note: Addition operator (**+**) works for Numeric as well as Strings. e.g. "a" + 10 will give "a10".

Comparison Operators

JavaScript supports the following comparison operators: Assume variable A holds 10 and variable B holds 20, then:

S.No	Operator and Description
1	**== (Equal)** Checks if the value of two operands are equal or not, if yes, then the condition becomes true. **Ex:** (A == B) is not true.
2	**!= (Not Equal)** Checks if the value of two operands are equal or not, if the values are not equal, then the condition becomes true. **Ex:** (A != B) is true.
3	**> (Greater than)** Checks if the value of the left operand is greater than the

	value of the right operand, if yes, then the condition becomes true. **Ex:** (A > B) is not true.
4	**< (Less than)** Checks if the value of the left operand is less than the value of the right operand, if yes, then the condition becomes true. **Ex:** (A < B) is true.
5	**>= (Greater than or Equal to)** Checks if the value of the left operand is greater than or equal to the value of the right operand, if yes, then the condition becomes true. **Ex:** (A >= B) is not true.
6	**<= (Less than or Equal to)** Checks if the value of the left operand is less than or equal to the value of the right operand, if yes, then the condition becomes true. **Ex:** (A <= B) is true.

Example

The following code shows how to use comparison operators in JavaScript.

```
<html>
<body>

<script type="text/javascript">
<!--var a = 10; var b = 20;
var linebreak = "<br />";

document.write("(a == b) => "); result =
(a == b); document.write(result);
```

```
document.write(linebreak);
document.write("(a < b) => "); result = (a < b); document.write(result);
document.write(linebreak);
document.write("(a > b) => "); result = (a > b); document.write(result);
document.write(linebreak);
document.write("(a != b) => "); result = (a != b); document.write(result);
document.write(linebreak);
document.write("(a >= b) => "); result = (a >= b); document.write(result);
document.write(linebreak);
document.write("(a <= b) => "); result = (a <= b); document.write(result);
document.write(linebreak);
//-->
</script>
<p>Set the variables to different values and different operators and then try...</p>
</body>
</html>
```

Output

```
(a == b) => false
(a < b) => true
(a > b) => false
(a != b) => true
(a >= b) => false
(a <= b) => true

Set the variables to different values and different operators and then
try...
```

Logical Operators

JavaScript supports the following logical operators: Assume variable A holds 10 and variable B holds 20, then:

S.No	Operator and Description
1	**&& (Logical AND)** If both the operands are non-zero, then the condition becomes true. **Ex:** (A && B) is true.
2	**|| (Logical OR)** If any of the two operands are non-zero, then the condition becomes true. **Ex:** (A || B) is true.

3	**! (Logical NOT)** Reverses the logical state of its operand. If a condition is true, then the Logical NOT operator will make it false. **Ex:** ! (A && B) is false.

Example

```
<html>
<body>
<script type="text/javascript">
<!--
var a = true; var b = false;
var linebreak = "<br />";
document.write("(a &&                    b) => "); result = (a && b);
document.write(result); document.write(linebreak);

document.write("(a || b) => "); result = (a || b); document.write(result);
document.write(linebreak);
document.write("!(a && b) => "); result = (!(a && b)); document.write(result);
document.write(linebreak);
//-->
</script>
```

Bitwise Operators

JavaScript supports the following bitwise operators: Assume variable A holds 2 and variable B holds 3, then:

S.No	Operator and Description
1	& (Bitwise AND) It performs a Boolean AND operation on each bit of its integer arguments. Ex: (A & B) is 2.
2	| (BitWise OR) It performs a Boolean OR operation on each bit of its integer arguments. Ex: (A | B) is 3.
3	^ (Bitwise XOR) It performs a Boolean exclusive OR operation on each bit of its integer arguments. Exclusive OR means that either operand one is true or operand two is true, but not both. Ex: (A ^ B) is 1.
4	~ (Bitwise Not) It is a unary operator and operates by reversing all the bits in the operand.

Example

Try the following code to implement Bitwise operator in JavaScript.

```
<html>
<body>

<script type="text/javascript">
<!--
var a = 2; // Bit presentation 10 var b =
3; // Bit presentation 11 var linebreak =
"<br />";
document.write("(a &  b) => "); result
= (a & b);
document.write(result);
document.write(linebreak);
document.write("(a | b) => "); result =
(a | b); document.write(result);
document.write(linebreak);
document.write("(a ^ b) => "); result =
(a ^ b); document.write(result);
document.write(linebreak);
document.write("(~b) => "); result =
(~b); document.write(result);
document.write(linebreak);
document.write("(a << b) => "); result
= (a << b); document.write(result);
document.write(linebreak);
```

```
document.write("(a >> b) => "); result
= (a >> b); document.write(result);
document.write(linebreak);
//-->
</script>
<p>Set the variables to different values and different operators and then
try...</p>
</body>
</html>
```

Output

```
(a & b) => 2 (a |
b) => 3 (a ^ b) =>
1 (~b) => -4
(a << b) => 16 (a >>
b) => 0
```

Assignment Operators

JavaScript supports the following assignment operators:

S.No	Operator and Description
1	**= (Simple Assignment)** Assigns values from the right side operand to the left side operand

	Ex: C = A + B will assign the value of A + B into C
2	**+= (Add and Assignment)** It adds the right operand to the left operand and assigns the result to the left operand. **Ex:** C += A is equivalent to C = C + A
3	**-= (Subtract and Assignment)** It subtracts the right operand from the left operand and assigns the result to the left operand. **Ex:** C -= A is equivalent to C = C - A
4	***= (Multiply and Assignment)** It multiplies the right operand with the left operand and assigns the result to the left operand. **Ex:** C *= A is equivalent to C = C * A
5	**/= (Divide and Assignment)** It divides the left operand with the right operand and assigns the result to the left operand.
	Ex: C /= A is equivalent to C = C / A
6	**%= (Modules and Assignment)** It takes modulus using two operands and assigns the result to the left operand.

	Ex: C %= A is equivalent to C = C % A

Note: Same logic applies to Bitwise operators, so they will become <<=, >>=, >>=, &=, |= and ^=.

Example

Try the following code to implement assignment operator in JavaScript.

```
<html>
<body>
<script type="text/javascript">
<!--
var a = 33; var b = 10;
var linebreak = "<br />";
document.write("Value of a => (a = b) => "); result = (a = b);
document.write(result); document.write(linebreak);
document.write("Value of a => (a += b) => "); result = (a += b);
document.write(result); document.write(linebreak);
document.write("Value of a => (a -= b) => "); result = (a -= b);
document.write(result);
document.write(linebreak);
document.write("Value of a => (a *=  b) => "); result = (a *= b);
document.write(result); document.write(linebreak);
document.write("Value of a => (a /= b) => "); result = (a /= b);
document.write(result); document.write(linebreak);
```

```
document.write("Value of a => (a %= b) => "); result = (a %= b);
document.write(result); document.write(linebreak);
//-->
</script>
<p>Set the variables to different values and different operators and then try...</p>
</body>
</html>
```

Output

```
Value of  a  =>  (a  = b) => 10
Value of  a  =>  (a  += b) => 20
Value of  a  =>  (a  -= b) => 10
Value of  a  =>  (a  *= b) => 100
Value of  a  =>  (a  /= b) => 10
Value of  a  =>  (a  %= b) => 0
```

Miscellaneous Operators

The conditional operator (? :) and the typeof operator.

Conditional Operator (? :)

The conditional operator first evaluates an expression for a true or false value and then executes one of the two given statements depending upon the result of the evaluation.

S.No	Operator and Description
1	**? : (Conditional)** If Condition is true? Then value X : Otherwise value Y

Example

Try the following code to understand how the Conditional Operator works in JavaScript.

```
<html>
<body>
<script type="text/javascript">
<!--
var a = 10; var b
= 20;
var linebreak = "<br />";
document.write ("((a > b) ? 100 : 200) => "); result =
(a > b) ? 100 : 200; document.write(result);
```

```
document.write(linebreak);

document.write ("((a < b) ? 100 : 200) => ");
result = (a < b) ? 100 : 200;
document.write(result);
document.write(linebreak);

//-->
</script>

<p>Set the variables to different values and different operators and then try...</p>
</body>
</html>
```

OUTPUT

```
((a > b) ? 100 : 200) => 200
((a < b) ? 100 : 200) => 100
```

Conditional Statements

JavaScript supports the following forms of **if..else** statement:

- if statement
- if...else statement
- if...else if... statement

if Statement

The '**if**' statement is the fundamental control statement that allows JavaScript to

make decisions and execute statements conditionally.

Syntax

The syntax for a basic if statement is as follows:

```
if (expression){
    Statement(s) to be executed if expression is true
}
```

Here a JavaScript expression is evaluated. If the resulting value is true, the given statement(s) are executed. If the expression is false, then no statement would be not executed. Most of the times, you will use comparison operators while making decisions.

Example

Try the following example to understand how the **if** statement works.

```
<html>
<body>
<script type="text/javascript">
<!--
var age = 20; if( age >
18 ){
    document.write("<b>Qualifies for driving</b>");
}
//-->
</script>
```

```
<p>Set the variable to different value and then try...</p>
</body>
</html>
```

Output

Qualifies for driving

if...else Statement

The '**if...else**' statement is the next form of control statement that allows JavaScript to execute statements in a more controlled way.

Syntax

The syntax of an **if-else** statement is as follows:

```
if (expression){
Statement(s) to be executed if expression is true
}else{
Statement(s) to be executed if expression is false
}
```

Here JavaScript expression is evaluated. If the resulting value is true, the given statement(s) in the 'if' block, are executed. If the expression is false, then the given statement(s) in the else block are executed.

Example

```
<html>
```

```
<body>

<script type="text/javascript">
<!--
var age = 15;

if( age > 18 ){
    document.write("<b>Qualifies for driving</b>");
}else{
    document.write("<b>Does not qualify for driving</b>");
}
//-->
</script>

<p>Set the variable to different value and then try...</p>
</body>
</html>
```

if...else if...Statement

The '**if...else if...**' statement is an advanced form of **if...else** that allows JavaScript to make a correct decision out of several conditions.

Syntax

The syntax of an if-else-if statement is as follows:

```
if (expression 1){
    Statement(s) to be executed if expression 1 is true
```

```
}else if (expression 2){
    Statement(s) to be executed if expression 2 is true
}else if (expression 3){
    Statement(s) to be executed if expression 3 is true
}else{
    Statement(s) to be executed if no expression is true
}
```

There is nothing special about this code. It is just a series of **if** statements, where each **if** is a part of the **else** clause of the previous statement. Statement(s) are executed based on the true condition, if none of the conditions is true, then the **else** block is executed.

Example

```
<html>
<body>

<script type="text/javascript">
<!--
var book = "maths";
if( book == "history" ){
    document.write("<b>History Book</b>");
}else if( book == "maths" ){
    document.write("<b>Maths Book</b>");
}else if( book == "economics" ){
    document.write("<b>Economics Book</b>");
}else{
```

```
    document.write("<b>Unknown Book</b>");
  }
  //-->
  </script>
  <p>Set the variable to different value and then try...</p>
  </body>
  </html>
```

Outpu

Maths Book

Switch

The objective of a **switch** statement is to give an expression to evaluate and several different statements to execute based on the value of the expression. The interpreter checks each **case** against the value of the expression until a match is found. If nothing matches, a **default** condition will be used.

Syntax

```
switch (expression)
{
   case condition 1: statement(s)
                     break;
   case condition 2: statement(s)
                     break;
     ...
   case condition n: statement(s)
```

```
                    break;
   default: statement(s)
}
```

The **break** statements indicate the end of a particular case. If they were omitted, the interpreter would continue executing each statement in each of the following cases.

Example

```
html>
<body>

<script type="text/javascript">
<!--
var grade='A';
document.write("Entering switch block<br />");
switch (grade)
{
   case 'A': document.write("Good job<br />");
             break;
   case 'B': document.write("Pretty good<br />"); break;
   case 'C': document.write("Passed<br />"); break;
   case 'D': document.write("Not so good<br />"); break;
   case 'F': document.write("Failed<br />"); break;
   default:  document.write("Unknown grade<br />")
}
document.write("Exiting switch block");
```

```
//-->
</script>

<p>Set the variable to different value and then try...</p>
</body>
</html>
```

Output

```
Good job
Exiting switch block
```

While Loop

While writing a program, you may encounter a situation where you need to perform an action over and over again. In such situations, you would need to write loop statements to reduce the number of lines.

JavaScript supports all the necessary loops to ease down the pressure of programming.

The while Loop

The most basic loop in JavaScript is the **while** loop which would be discussed in this chapter. The purpose of a **while** loop is to execute a statement or code block repeatedly as long as an **expression** is true. Once the expression becomes **false**, the loop terminates.

Syntax

The syntax of **while loop** in JavaScript is as follows:

```
while (expression){
    Statement(s) to be executed if expression is true
}
```

Example

Try the following example to implement while loop.

```
<html>
<body>
<script type="text/javascript">
<!--
var count = 0; document.write("Starting
Loop "); while (count < 10){
   document.write("Current Count : " + count + "<br />");
   count++;
}
document.write("Loop stopped!");
//-->
</script>

<p>Set the variable to different value and then try...</p>
</body>
</html>
```

Output

Current Count : 1
Current Count : 2
Current Count : 3
Current Count : 4
Current Count : 5
Current Count : 6
Current Count : 7
Current Count : 9

The do...whileLoop

The **do...while** loop is similar to the **while** loop except that the condition check happens at the end of the loop. This means that the loop will always be executed at least once, even if the condition is **false**.

Syntax

The syntax for **do-while** loop in JavaScript is as follows:

```
do{
    Statement(s) to be executed;
} while (expression);
```

```
<html>

<body>
```

```
<script type="text/javascript">
<!--
var count = 0;
document.write("Starting Loop" + "<br />"); do{
document.write("Current Count : " + count + "<br />"); count++;
}while (count < 5); document.write ("Loop stopped!");
//-->
</script>
<p>Set the variable to different value and then try...</p>
</body>
</html>
```

The forLoop

The '**for**' loop is the most compact form of looping. It includes the following three important parts:

- The **loop initialization** where we initialize our counter to a starting value. The initialization statement is executed before the loop begins.
- The **test statement** which will test if a given condition is true or not. If the condition is true, then the code given inside the loop will be executed, otherwise the control will come out of the loop.
- The **iteration statement** where you can increase or decrease your counter.

- To put all the three parts in a single line separated by semicolons.

Syntax

```
for (initialization; test condition; iteration statement){
        Statement(s) to be executed if test condition is true
}
```

```
<html>
<body>
<script type="text/javascript">
<!--
var count;
document.write("Starting Loop" + "<br />"); for(count = 0; count < 10; count++){
document.write("Current Count : " + count ); document.write("<br />");
}
document.write("Loop stopped!");
//-->
</script>
<p>Set the variable to different value and then try...</p>
</body>
</html>
```

Function

A function is a group of reusable code which can be called anywhere in your program. This eliminates the need of writing the same code again and again. It helps programmers in writing modular codes. Functions allow a programmer to divide a big program into a number of small and manageable functions. Like any other advanced programming language, JavaScript also supports all the features necessary to write modular code using functions. You must have seen functions like alert() and write() in the earlier chapters. We were using these functions again and again, but they had been written in core JavaScript only once. JavaScript allows us to write our own functions as well.

Function Definition

Before we use a function, we need to define it. The most common way to define a function in JavaScript is by using the function keyword, followed by a unique function name, a list of parameters (that might be empty), and a statement block surrounded by curly braces.

Syntax

```
<script type="text/javascript">
<!--
function functionname(parameter-list)
{
   statements
}
//-->
</script>
```

For Example

```
<script type="text/javascript">
<!--
function sayHello()
{
    alert("Hello there");
}
//-->
</script>
```

Calling a Function

To invoke a function somewhere later in the script, you would simply need to write the name of that function

```
<html>
<head>
<script type="text/javascript">
function sayHello()
{
 document.write ("Hello there!");
}
</script>
```

```
</head>

<body>

<p>Click the following button to call the function</p>

<form>

<input type="button" onclick="sayHello()" value="Say Hello">

</form>

<p>Use different text in write method and then try...</p>

</body>

</html>
```

Function Parameters

Till now, we have seen functions without parameters. But there is a facility to pass different parameters while calling a function. These passed parameters can be captured inside the function and any manipulation can be done over those parameters. A function can take multiple parameters separated by comma.

Example

```
<html>

<head>

<script type="text/javascript">
```

```
function sayHello(name, age)
{
 document.write (name + " is " + age + " years old.");
}
</script>
</head>
<body>
<p>Click the following button to call the function</p>
<form>
<input type="button" onclick="sayHello('Zara', 7)" value="Say Hello">
</form>
<p>Use different parameters inside the function and then try...</p>
</body>
</html>
```

The return Statement

A JavaScript function can have an optional return statement. This is required if you want to return a value from a function. This statement should be the last statement in a function.

For example, you can pass two numbers in a function and then you can expect the function to return their multiplication in your calling program.

Example

```
<html>

<head>

<script type="text/javascript">

function concatenate(first, last)

{

 var full;

 full = first + last;

 return full;

}

function secondFunction()

{

 var result;

 result = concatenate('Zara', 'Ali');

 document.write (result );

}

</script>
```

```
</head>

<body>

<p>Click the following button to call the function</p>

<form>

<input type="button" onclick="secondFunction()" value="Call Function">

</form>

<p>Use different parameters inside the function and then try...</p>

</body>

</html>
```

Using JavaScript Using Forms

There are few differences between a straight HTML form and a JavaScript-enhanced form. The main one being that a JavaScript form relies on one or more event handlers, such as onClick or onSubmit. These invoke a JavaScript action when the user does something in the form, like clicking a button. The event handlers, which are placed with the rest of the attributes in the HTML form tags, are invisible to a browser that don't support JavaScript. Because of this trait, you can often use one form for both JavaScript and non-JavaScript browsers.

Typical form control objects -- also called "widgets" -- include the following:

- Text box for entering a line of text
- Push button for selecting an action
- Radio buttons for making one selection among a group of options

- Check boxes for selecting or deselecting a single, independent option

```
<FORM NAME="myform" ACTION="" METHOD="GET">
Enter something in the box: <BR>
<INPUT TYPE="text" NAME="inputbox" VALUE=""><P>
<INPUT TYPE="button" NAME="button" Value="Click"
onClick="testResults(this.form)">
</FORM>
```

- FORM NAME="myform" defines and names the form. Elsewhere in the JavaScript you can reference this form by the name myform. The name you give your form is up to you, but it should comply with JavaScript's standard variable/function naming rules (no spaces, no weird characters except the underscore, etc.).
- ACTION="" defines how you want the browser to handle the form when it is submitted to a CGI program running on the server. As this example is not designed to submit anything, the URL for the CGI program is omitted.
- METHOD="GET" defines the method data is passed to the server when the form is submitted. In this case the atttibute is puffer as the example form does not submit anything.
- INPUT TYPE="text" defines the text box object. This is standard HTML markup.
- INPUT TYPE="button" defines the button object. This is standard HTML markup except for the onClick handler.
- onClick="testResults(this.form)" is an event handler -- it handles an event,

in this case clicking the button. When the button is clicked, JavaScript executes the expression within the quotes. The expression says to call the testResults function elsewhere on the page, and pass to it the current form object.

Getting a value from a form object

```
<HTML>
<HEAD>
<TITLE>Test Input</TITLE>
<SCRIPT LANGUAGE="JavaScript">
function testResults (form) {
    var TestVar = form.inputbox.value;
    alert ("You typed: " + TestVar);
}
</SCRIPT>
</HEAD>
<BODY>
<FORM NAME="myform" ACTION="" METHOD="GET">Enter something in the box:
<BR>
<INPUT TYPE="text" NAME="inputbox" VALUE=""><P>
<INPUT TYPE="button" NAME="button" Value="Click"
onClick="testResults(this.form)">
</FORM>
```

```
</BODY>
</HTML>
```

Setting a value in a form object

The value property of the inputbox, shown in the above example, is both readable and writable.

```
<HTML>
<HEAD>
<TITLE>Test Input </TITLE>
<SCRIPT LANGUAGE="JavaScript">
function readText (form) {
    TestVar =form.inputbox.value;
    alert ("You typed: " + TestVar);
}
function writeText (form) {
    form.inputbox.value = "Have a nice day!"
}
</SCRIPT>
</HEAD>
```

```
<BODY>

<FORM NAME="myform" ACTION="" METHOD="GET">

Enter something in the box: <BR>

<INPUT TYPE="text" NAME="inputbox" VALUE=""><P>

<INPUT TYPE="button" NAME="button1" Value="Read"
onClick="readText(this.form)">

<INPUT TYPE="button" NAME="button2" Value="Write"
onClick="writeText(this.form)">

</FORM>

</BODY>

</HTML>
```

- When you click the "Read" button, JavaScript calls the readText function, which reads and displays the value you entered into the text box.
- When you click the "Write" button, JavaScript calls the writeText function, which writes "Have a nice day!" in the text box.

Reading other form object values

The text box is perhaps the most common form object you'll read (or write) using JavaScript

In addition to text boxes, JavaScript can be used with:

Hidden text box (TYPE="hidden").

Radio button (TYPE="radio")

Check box (TYPE="checkbox")

Text area (<TEXT AREA>)

Lists (<SELECT>)

Using Hidden Text Boxes

From a JavaScript standpoint, hidden text boxes behave just like regular text boxes, sharing the same properties and methods. From a user standpoint, hidden text boxes "don't exist" because they do not appear in the form.

Using Radio Buttons

Radio buttons are used to allow the user to select one, and only one, item from a group of options.

<INPUT TYPE="radio" NAME="rad" VALUE="radio_button1" onClick=0>

<INPUT TYPE="radio" NAME="rad" VALUE="radio_button2" onClick=0>

<INPUT TYPE="radio" NAME="rad" VALUE="radio_button3" onClick=0>

<INPUT TYPE="radio" NAME="rad" VALUE="radio_button4" onClick=0>

Using Check Boxes

Check boxes are stand-alone elements; that is, they don't interact with neighboring elements like radio buttons.

<HTML>

<HEAD>

```
<TITLE>Checkbox Test</TITLE>
<SCRIPT LANGUAGE="JavaScript">
function testButton (form){
    alert (form.check1.checked);
}
</SCRIPT>
</BODY>
<FORM NAME="testform">
<INPUT TYPE="button" NAME="button" Value="Click"
    onClick="testButton(this.form)"><BR>
<INPUT TYPE="checkbox" NAME="check1" Value="Check1">Checkbox 1<BR>
<INPUT TYPE="checkbox" NAME="check2" Value="Check2">Checkbox 2<BR>
<INPUT TYPE="checkbox" NAME="check3" Value="Check3">Checkbox 3<BR>
</FORM>
</BODY>
</HTML>
```

Using Text Areas

Text areas are used for multiple-line text entry. The default size of the text box is 1 row by 20 characters.

```
<TEXTAREA NAME="myarea" COLS="40" ROWS="7">
</TEXTAREA>
```

```
<HTML>
<HEAD>
<TITLE>Text Area Test</TITLE>
<SCRIPT LANGUAGE="JavaScript">
function seeTextArea (form) {
    alert (form.myarea.value);
}
</SCRIPT>
</HEAD>
<BODY>
<FORM NAME="myform">
<INPUT TYPE="button" NAME="button3" Value="Test"
    onClick="seeTextArea(this.form)">
<TEXTAREA NAME="myarea" COLS="40" ROWS="5">
```

```
</TEXTAREA>

</FORM>

</BODY>

</HTML>
```

Using Selection Lists

List boxes let you pick the item you want out of a multiple-choice box. The listbox itself is created with the <SELECT> tag, and the items inside it are created by one or more <OPTION> tags. We can have any number of <OPTION> tags in a list. The list is terminated with a </SELECT> tag.

```
<HTML>

<HEAD>

<TITLE>List Box Test</TITLE>

<SCRIPT LANGUAGE="JavaScript">

function testSelect(form) {

    alert (form.list.selectedIndex);

}

</SCRIPT>

</HEAD>

<BODY>
```

```
<FORM NAME="myform" ACTION="" METHOD="GET">

<INPUT TYPE="button" NAME="button" Value="Test"
onClick="testSelect(this.form)">

<SELECT NAME="list" SIZE="3">

<OPTION>This is item 1

<OPTION>This is item 2

<OPTION>This is item 3

</SELECT>

</FORM>

</BODY>

</HTML>
```

testSelect function:

```
function testSelect (form) {

    Item = form.list.selectedIndex;

    Result = form.list.options[Item].text;

    alert (Result);

}
```

Other events you can trigger within a form

The event handlers used with form object are:

onFocus -- an event is triggered with a form object gets input focus (the insertion point is clicked there).

onBlur -- an event is triggered with a form object loses input focus (the insertion point is clicked out of there).

onChange -- an event is triggered when a new item is selected in a list box. This event is also trigged with a text or text area box loses focus and the contents of the box has changed.

onSelect -- an event is triggered when text in a text or text area box is selected.

onSubmit -- an event is triggered when the form is submitted to the server (more about this important hander later in the column).

Submitting the form to the server

Many forms are designed to send data back to a CGI program running on the server. This is referred to as submitting the form, and is accomplished using either of two JavaScript instructions: the onSubmit event handler or the submit method. In most instances, you use one or the other, not both!

Place the onSubmit event hander in the <FORM> tag. This tells JavaScript what it should do when the user clicks the Submit button (this is a button defined as TYPE="submit").

Place the submit instruction anywhere in your JavaScript. It can be activated by any action, such as clicking a form button that has been defined with the onClick event handler.

Using onSubmit

Below example of using the onSubmit event handler to send mail. The onSubmit event handler tells JavaScript what to do when the user clicks the Submit button: call the mailMe() function, where additional mail fields are appended to a mailto: URL. Navigator automatically opens a new mail window with the fields filled in. Write the body of the message, and send the mail off to the recipient.

```
<HTML>

<HEAD>

<TITLE>onSubmit Test</TITLE>

<SCRIPT LANGUAGE="JavaScript">

function mailMe(form){

    Subject=document.testform.inputbox1.value;

    CC= document.testform.inputbox2.value;

    BCC= document.testform.inputbox3.value;

    location = "mailto:jwedit@javaworld.com?subject="+Subject+"&Bcc="+

        BCC+"&cc="+CC;

    return true;

}

</SCRIPT>

</HEAD>
```

```
<BODY>

<FORM NAME="testform" onSubmit="return mailMe(this.form)" >Subject of message: <BR>

<INPUT TYPE="text" NAME="inputbox1" VALUE="This is such a great form!" SIZE=50>

<P>Send cc to: <BR>

<INPUT TYPE="text" NAME="inputbox2" VALUE="" SIZE=50><P>

Send blind cc to: <BR>

<INPUT TYPE="text" NAME="inputbox3" VALUE="" SIZE=50><P>

<INPUT TYPE="submit"><BR>

</FORM>

</BODY>

</HTML>
```

JavaScript Style Sheets

JavaScript Style Sheet Syntax

Netscape 4 implements JSS by extending several existing HTML tags and defining a few new objects that store your document's styles.

External, document-level, and inline JSS

As with CSS, you may reference and load external JSS files with the <link> tag. For example:

<link href="styles.js" rel=stylesheet type=text/JavaScript>

The only real difference between this one and the one for a CSS external style sheet is that the type attribute of the <link> tag is set to text/JavaScript instead of text/CSS. The referenced file, styles.js, contains JavaScript statements that define styles and classes that Netscape will then use to control display of the current document.

The contents of the <style> tag for JSS are quite different from those for CSS, however. For example:

```
<style type=text/JavaScript>
<!--
   tags.BODY.marginLeft = "20px";
   tags.P.fontWeight = "bold";
  // -->
</style>
```

Defining styles for tags

JavaScript defines a new document property, tags, that contains the style properties for all HTML tags. To define a style for a tag, to simply set the appropriate property of the desired style property within the tag property of the document object. For example:

```
document.tags.P.fontSize = '12pt';
document.tags.H2.color = 'blue';
```

These two JSS definitions set the font size for the <p> tag to 12 points and render all <h2> tags in blue.

The equivalent CSS definitions are:

P {font-size : 12pt}

H2 {color : blue}

We could have written the previous two styles as:

The correct versions are:

tags.P.fontSize = '12pt';

tags.BODY.color = 'blue';

tags.P.color = 'red';

It can be very tedious to specify a number of properties for a single tag, so you can take advantage of the JavaScript with statement to reduce your typing burden.

These styles:

tags.P.fontSize = '14pt';

tags.P.color = 'blue';

tags.P.fontWeight = 'bold';

tags.P.leftMargin = '20%';

can be more easily written as:

with (tags.P) {

```
fontSize = '14pt';
color = 'blue';
fontWeight = 'bold';
leftMargin = '20%';
}
```

To apply similar styles to diverse tags just as easily:

```
with (tags.P, tags.LI, tags.H1) {
fontSize = '14pt';
color = 'blue';
fontWeight = 'bold';
leftMargin = '20%';
}
```

Defining style classes

Like CSS, JSS lets you target styles for specific ways in which a tag may be used in your document. JSS uses the classes property to define separate styles for the same tag. There are no predefined properties within the classes property; instead, any property you reference is defined as a class to be used by the current document. For example:

```
classes.bold.P.fontWeight = 'bold';
with (classes.abstract.P) {
```

```
leftMargin = '20pt';

rightMargin = '20pt';

fontStyle = 'italic';

textAlign = 'justify';

}
```

The first style defines a class of the <p> tag named bold whose font weight is set to bold. The next style uses the with statement to create a class of the <p> tag named abstract with the specified properties. The equivalent CSS rules would be:

```
P.bold {font-weight : bold}

P.abstract {left-margin : 20pt;

        right-margin : 20pt;

        font-style : italic;

        text-align : justify

       }
```

Once defined, use a JSS class just like any CSS class: with the class attribute and the class name.

Like CSS, JSS also lets you define a class without defining the tag that will use the class. To define generic classes that you can later apply to any tag. To create a generic style class in JSS, use the special tag property all:

```
classes.green.all.color = "green";
```

To can then add class="green" to any tag to have Netscape render its contents in green. The equivalent CSS is:

.green {color : green}

POSSIBLE QUESTION

Part A (1 Mark)

(Online Examinations)

Part B (8 Marks)

1. Discuss about basics of java scripts.
2. Explain the form elements in java script with examples.
3. Discuss about an altering of status bar in java script.
4. Explain the control flow statement in java script.
5. What are the form elements properties used in java script? Explain.
6. Discuss about Client side Image Maps.
7. Enlighten the concept of JavaScript Language.
8. Explain the java script form element methods and event handlers with examples.
9. Describe on java script with stylesheet.
10. What are the data types used in java script? Explain.

KARPAGAM ACADEMY OF HIGHER EDUCATION

(Deemed to be University)
(Established Under Section 3 of UGC Act 1956)
Pollachi Main Road, Eachanari Post, Coimbatore - 641021
(For the candidates admitted from 2015 onwards)

CLASS: III B.COM CA **COURSE NAME: INTERNET AND WEB DESIGN**

COURSE CODE: 15CCU601 **UNIT: IV (INTRODUCTION TO JAVA SCRIPT)** **BATCH-2015-2018**

S.NO	QUESTIONS	OPTION 1	OPTION 2	OPTION 3	OPTION 4	ANSWER
1	A client – side script is typically a small program embedded within an _____.	Html document	Xml document	Dhml document	All the above	Html document
2	Client-side scripts are referred to as scripts rather than _____.	Programs	Programmer	Application	Both 1 and 3	Programs
3	In a programming language or a scripting language the results are saved in _____.	Text file	Source file	Document	None	Text file
4	Which of these following was the first client-side scripting language?	Java script	Vb script	Both 1 and 2	None	Java script
5	The purpose of java script is to provide a _____.	Web pages	Programming languages	Both 1 and 2	None	Web pages
6	The Microsoft implementation of java script is _____.	Jscript	J - script	Script of J	Script – J	Jscript
7	In Jscript, the closing comment characters are preceded by _____.	2	3	4	1	2
8	Client – side scripts, are programming languages with _____.	Data Structures	Control Structures	Conditional Statements	All the above	All the above
9	_____ are one of the tasks that no need to perform on server- side.	Image maps	Bitmaps	Client	None	Image maps
10	The purpose of an image maps is to define _____.	Multiple hotspots	Single hotspots	Pixels	None	Multiple hotspots
11	Server – slide image maps were the _____ image map form.	First	Second	Third	Fourth	First
12	_____ image maps function in a straight forward manner.	Server – slide	Client – slide	Both 1 and 2	None	Server – slide
13	The Attributes of <Area> tag is _____.	Shape	Target	Name	All the above	All the above
14	The Attribute of shape is _____.	rect	coords	map	image	rect
15	The target Attribute will specify the _____ document is loaded.	Window's name	Frame name	Both 1 and 2	None	Both 1 and 2
16	HREF attribute used in the Area tag for defining _____.	Links	List	Tags	None	Links
17	Usemap attribute is similar to _____ attribute.	Ismap	Shape	Href	Target	Ismap
18	The more popular of WYSIWYG image map editors are _____.	Map edit	Hot spots	Live image	All the above	All the above

19	Which of these following will support the map edit in editors?	Client-side image maps	Server-side image maps	Both 1 and 2	None	Both 1 and 2
20	Java script is a _____ scripting language.	Client-side	Server-side	Both 1 and 2	None	Client-side
21	Java script uses the _____ end of a statement.	Semi-colon	Colon	Full stop	Slash.	Semi-colon
22	What is the symbol used in block statement?	{ }	( )	< >	[ ]	{ }
23	In JavaScript _____ types of comments are specified.	2	3	4	1	2
24	A single – line comment is denoted with _____.	2 slashes	2 asterisks	2 Ampersand	1 slash and 1 asterisk	2 slashes
25	Find the odd man out.	String	Number	Object	And	And
26	Creating variables in java script must be a _____.	case- sensitive	incase – sensitive	simple source code	None	case- sensitive
27	Numerical expressions deal with _____ .	Number data type	Data types	Functions	Strings	Number data type
28	Methods that determine how our script continues are known as _____.	Conditional expression	Biconditional expression	Logical expression	None	Conditional expression
29	_____ statement can be used to determine a course of action.	If	If – else	If then	While	If
30	_____ will accept both parameters and arguments.	Functions	Arrays	Loops	Variables	Functions
31	JavaScript is an _____ scripting language.	Object - based	Object	Object – model	Element based	Object - based
32	Which of these are the form properties in Jscript?	Action	Encoding	Method	All the above	All the above
33	_____ is used to communicate our form data to the server.	Method	Target	Encoding	Action	Method
34	The data that passed to the server from our script is _____.	Encoding	Action	Target	Method	Encoding
35	The property that handles our form input on the server is _____.	Action	Encoding	Target	Method	Action
36	The methods that are associated with the form object are _____ and _____.	Submit ( ), reset ( )	Submit ( ), Set ( )	Reset ( ), Class ( )	Object ( ), Class ( )	Submit ( ), reset ( )
37	The button element represents a button on the _____.	Current form	Loaded form	Icons	None	Current form
38	The Radio button functions like a checkbox, but offers _____.	different visual style	similar visual style	same visual style	None	different visual style
39	In select element, a single selection box shows the _____ in the visible field.	current selection	default selection	Both 1 and 2	None	current selection
40	The elements that allow to enter multiple lines of text is _____.	text area element	text element	password element	hidden element	text area element
41	How many form elements methods are there?	4	2	3	5	4

42	The _____ method can be used to remove focus from a current element.	blur ( )	click ( )	focus ( )	select ( )	blur ( )
43	_____ method is similar to blur ( ) method.	click ( )	focus ( )	select ( )	None	click ( )
44	_____ method is opposite to blur ( ) method.	click ( )	focus ( )	select ( )	None	focus ( )
45	The event that occurs when someone click on the element is _____.	Onclick	Ondblclick	Onmouseover	None	Onclick
46	_____ property is used to find the number of style sheet in document.	Length	Width	Count	None	Length
47	To replace an existing style sheet with another, _____ tag is used.	Link	Script	Image	None	Link
48	The _____ element is used to add structure to a block of text.	Division	Link	Head	None	Division
49	For horizontal rule tag the end tag is _____.	Required	Optional	Illegal	None	Illegal
50	_____ tag provides information about the document	Meta	Html	Head	Title	Meta
51	To download files from servers, the software's used is _____.	FTP	TCP	IP	TCP/ IP	FTP
52	Which of these following browsers will JavaScript works?	Opera	Fire fox	Internet Explorer	All the above	All the above
53	_____ and JavaScript are two completely different language?	Java	VbScript	C	C++	Java
54	The <script> tag is used to insert a _____ into an HTML page.	JavaScript	Title	Paragraph	None	JavaScript
55	Does the external script cannot contain the <script> tag?	yes	no	optional	none	yes
56	JavaScript is primarily _____ scripting language for the World Wide Web.	client-side	server-side	applications	none	client-side
57	In JavaScript, a Single-line comment is denoted with _____.	1 slashes	2 slashes	2 asterisk	none	2 slashes
58	Find the odd man out?	strings	numbers	Booleans	dimension	dimension
59	In Password field, the entry is viewed with the _____ character.	*	@	&	$	*
60	A client-side script is typically a _____ embedded with an HTML document.	small program	large program	both 1& 2	none	small program

UNIT V

SYLLABUS

Introduction to ASP – Active Server Objects – Active Server Components – Cookies - Database Management with ASP. Emerging and Alternate Web Technologies – Active X Controls for the WWW - XML

Introduction to Active Server Pages

Active Server Pages were introduced by Microsoft in 1996 as a downloadable feature of Internet Information Server 3.0. The concept is pretty simple: an Active Server Page allows code written in the JavaScript or VBScript languages to be embedded within the HTML tags of a Web page and executed on the Web server. There are great advantages to this, not the least of which is security. Since your code is executed on the Web server, only HTML tags are sent to the browser. The result is that the ASP code is "invisible" to the end user.

Another upside to the "server-side script" concept is that it allows things like database connections to be made from the Web server rather than from the client.

Creating an Active Server Page, the server-side script can be identified in one of the following two ways:

Use the server-side script block identifiers "<%" and "%>".

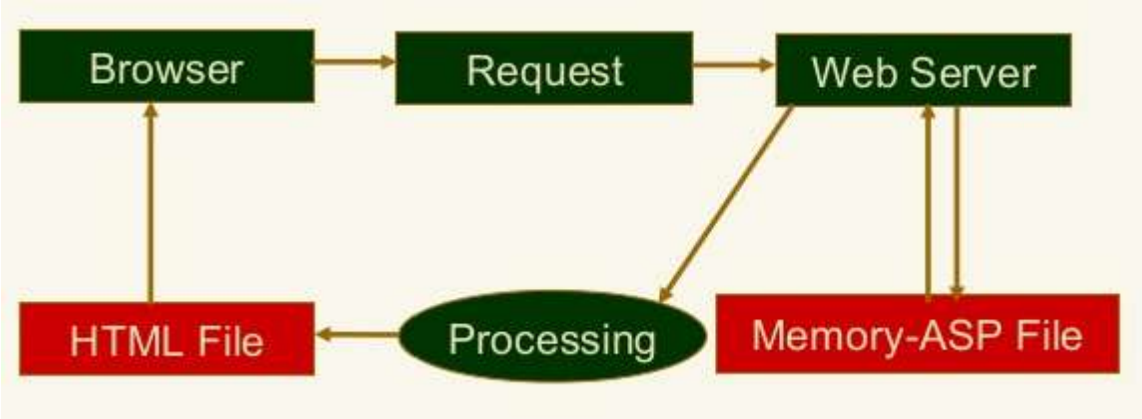
Use the <SCRIPT> tag with the RUNAT="SERVER" attribute.

Example uses a server-side VBScript to display a message in the browser window.

<HTML>

```
<HEAD>
<TITLE>Sample ASP Clock</TITLE>
</HEAD>
<BODY BGCOLOR=”BLACK”>
<FONT COLOR=”GREEN”>
<SCRIPT LANGUAGE=”VBScript” RUNAT=”SERVER”>
RESPONSE.WRITE NOW()
</SCRIPT>
</BODY>
</HTML>
```

Processing of HTML file



Active Server Objects

Active Server Pages are made up of the objects that will be "processed" by the server. The seven basic objects are:

Request Object

The Request object lets you tap into the information passed through an HTTP request. You can use the Request object to parse encoded URLs, access information from a form, and read cookies, client certificates, and the HTTP headers.

It allows data to be read that was sent by the client browser: Form, Querystring, and HTTP Cookie. It also provides information on the server, the client browser, and retrieve HTTP Cookie stored on the visitor's machine. Can retrieve data from a form using both methods HTTP:

Example:

Request.Form reads data sent by POST.

Request.QueryString reads data sent by GET.

```
<%
Response.Write "Welcome" & server.HTMLEncode(Request.QueryString("name")) & "!"
%>
```

Response Object

The Response object is your key to sending information to the user. We can write to the screen, redirect to another page, and create cookies using the Response object.

It can send information to the client, such as the writing of the text on a page or HTTP Cookie.

```
<%
If Len(Request.QueryString("name")) > 0 Then
    Response.Cookies("name") = Request.QueryString("name")
End If
Response.Write "Welcome " & Server.HTMLEncode(Response.Cookies("name")) & "!"
%>
```

Application Object

The running Web server is an application. Using the Application object, we can control features related to starting and stopping the application, as well as store information that should be accessed by the application as a whole.

It stores global variables.

```
<%
Application("Ali") = "My ASP Application"
Response.Write "Welcome to " & Server.HTMLEncode(Application("Ali")) & "!"
%>
```

Server Object

The Server object lets you perform routine functions, such as mapping a virtual path to a physical one and creating an instance of a component.

It allows connections to databases (ADO), filesystem, and use of components installed on the server.

```
<%

Dim oAdoCon, oAdoRec, oAdoStm, oCdoCon, oCdoMsg, oSciDic, oSciFsm, oMswAdr

Set oAdoCon = Server.CreateObject("ADODB.Connection")

Set oAdoRec = Server.CreateObject("ADODB.Recordset")

Set oAdoStm = Server.CreateObject("ADODB.Stream")

Set oCdoCon = Server.CreateObject("CDO.Configuration")

Set oCdoMsg = Server.CreateObject("CDO.Message")

Set oSciDic = Server.CreateObject("Scripting.Dictionary")

Set oSciFsm = Server.CreateObject("Scripting.FileSystemObject")

Set oMswAdr = Server.CreateObject("MSWC.AdRotator")

%>
```

Session Object

Using the Session object, you can store information related to each user who is visiting your site.

It stores variables accessible only to a single visitor.

```
<%
If Len(Request.QueryString("name")) > 0 Then
    Session("name") = Request.QueryString("name")
End If
Response.Write "Welcome " & Server.HTMLEncode(Session("name")) & "!"
%>
```

The Err object

It allows the management of errors.

```
<%
On Error Resume Next
Response.Write 1 / 0 ' Division by zero
If Err.Number <> 0 Then
    Response.Write "Error Code: " & Server.HTMLEncode(Err.Number) & "<br />"
```

```
    Response.Write "Error Source: " & Server.HTMLEncode(Err.Source) & "<br />"
    Response.Write "Error Description: " & Server.HTMLEncode(Err.Description) & "<br />"
    Err.Clear
End If
%>
```

Active Server Components

Ad Rotator

The Ad Rotator streamlines the process of setting up a delivery system for your banner ads. In a separate file, you store information regarding the banner. The component then delivers a randomly selected banner every time the page is loaded.

Browser Capabilities

The Browser component lets you determine what browser a user is using and what features are supported by that browser.

Collaboration Data Objects (CDO)

Tied in with the IIS SMTP server, CDO lets you send and receive email. With CDO, for example, you can process a form without relying upon a Perl script and CGI.

Content Linking

This is a handy object for creating a linear or sequential pathway through your site or a subsection of the site. You maintain a simple text file that lists the files in the

proper sequence. Simple next and previous links then can be added to the page, and a table of contents can be easily generated.

Content Rotator

If you have a need for rotating content, this will be a favorite component. It is easy to use and allows you to add dynamic content to any page without using a database. In a separate text file, you store chunks of HTML code that you want alternately dropped into a space on the page. The Content Rotator will display one of the chunks each time the page is reloaded.

Database Access

Using this component, you can hook into a database to write contents to the browser screen and to create or update existing database files.

ASP Cookies

The Cookies collection is used to set or get cookie values. If the cookie does not exist, it will be created, and take the value that is specified.

Note: The Response.Cookies command must appear before the <html> tag.

Syntax

Response.Cookies(name)[(key)|.attribute]=value

variablename=Request.Cookies(name)[(key)|.attribute]

Parameter	Description
Name	Required. The name of the cookie valueRequired for the Response.Cookies command. The value of

	the cookie
Attribute	Optional. Specifies information about the cookie. Can be one of the following parameters:
1. Domain	Write only. The cookie is sent only to requests to this domain
2. Expires	Write only. The date when the cookie expires. If no date is specified, the cookie will expire when the session ends
3. HasKeys	Read only. Specifies whether the cookie has keys (This is the only attribute that can be used with the Request.Cookies command)
4. Path	Write only. If set, the cookie is sent only to requests to this path. If not set, the application path is used
Secure	Write only. Indicates if the cookie is secure
key	Optional. Specifies the key to where the value is assigned

Examples

The "Response.Cookies" command is used to create a cookie or to set a cookie value:

<%

Response.Cookies("firstname")="Alex"

%>

In the code above, we have created a cookie named "firstname" and assigned the value "Alex" to it.

It is also possible to assign some attributes to a cookie, like setting a date when a cookie should expire:

<%

Response.Cookies("firstname")="Alex"

Response.Cookies("firstname").Expires=#May 10,2002#

%>

Now the cookie named "firstname" has the value of "Alex", and it will expire from the user's computer at May 10, 2002.

The "Request.Cookies" command is used to get a cookie value.

In the example below, we retrieve the value of the cookie "firstname" and display it on a page:

<%

fname=Request.Cookies("firstname")

response.write("Firstname=" & fname)

%>

Output:

Firstname=Alex

A cookie can also contain a collection of multiple values. We say that the cookie has Keys.

In the example below, we will create a cookie-collection named "user". The "user" cookie has Keys that contains information about a user:

```
<%
Response.Cookies("user")("firstname")="John"
Response.Cookies("user")("lastname")="Smith"
Response.Cookies("user")("country")="Norway"
Response.Cookies("user")("age")="25"
%>
```

The code below reads all the cookies your server has sent to a user. Note that the code checks if a cookie has Keys with the HasKeys property:

```
<html>
<body>
<%
dim x,y
for each x in Request.Cookies
  response.write("<p>")
```

```
if Request.Cookies(x).HasKeys then
  for each y in Request.Cookies(x)
    response.write(x & ":" & y & "=" & Request.Cookies(x)(y))
    response.write("<br>")
  next
else
  Response.Write(x & "=" & Request.Cookies(x) & "<br>")
end if
response.write "</p>"
next
%>

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

Output:

firstname=Alex

user:firstname=John

user:lastname=Smith

user:

country=Norway

user:

age=25

Cookies

Cookies were originally invented by Netscape to give 'memory' to web servers and browsers. The HTTP protocol, which arranges for the transfer of web pages to your browser and browser requests for pages to servers, is state-less, which means that once the server has sent a page to a browser requesting it, it doesn't remember a thing about it. So if you come to the same web page a second, third, hundredth or millionth time, the server once again considers it the very first time you ever came there.

This can be annoying in a number of ways. The server cannot remember if you identified yourself when you want to access protected pages, it cannot remember your user preferences, it cannot remember anything. As soon as personalization was invented, this became a major problem.

Cookies were invented to solve this problem. There are other ways to solve it, but cookies are easy to maintain and very versatile.

How cookies work

A cookie is nothing but a small text file that's stored in your browser. It contains some data:

- A name-value pair containing the actual data
- An expiry date after which it is no longer valid
- The domain and path of the server it should be sent to

Cookies can be read by JavaScript too. They're mostly used for storing user preferences.

Setting Cookies

cookies.txt :

During the browsing session browser stores the cookies in memory, at the time of quitting they goto the file called cookies.txt. Different browser store cookies files in a different location in the disk.

For example on windows 2000 firefox stores the cookies into C:\Documents and Settings\your_login_name_here\LocalSettings\ApplicationData\Mozilla\Firefox\Profiles\default .7g1\Cache. Note that the "default.7g1" folder name may vary depending on the version of Firefox you have Everytime When you open your browser, your cookies are read from the stored location and at the closing, your browser, cookies are reserved to disk. As a cookie expires it is no longer saved to the hard drive.

There are six parts of a cookie : name, value, expires, path, domain, and security. The first two parts i.e. name and value are required and rest parts are optional.

Syntax

document.cookie="NAME=VALUE; expires=DATE; path=PATH; domain=DOMAIN; secure";

The different components of the above syntax are discussed with a heading specifying the component.

name and value

The first part of the cookie string must have the name and value. The entire name/value must be a single string with no commas, semicolons or whitespace characters. Here is an example which stores the string "George" to a cookie named 'myName". The JavaScript statement is :

document.cookie = "myName=George";

Using encodeURIComponent() and decodeURIComponent() function.

Using encodeURIComponent() function it is ensured that the cookie value string does not contain any commas, semicolons, or whitespace characters. which are not allowed in cookie values.

encodeURIComponent("Good Morning");

Returns the string as :

Good%20Morning

while decodeURIComponent("Good%20Morning") decode the string as :

Good Morning.

expires

The cookie has a very limited lifespan. It will disappear when a user exits the browser. To give more life to the cookies you must set an expiration date in the following format.

DD-Mon-YY HH:MM:SS GMT

Here is an example where the cookie will live upto Mon, 12 Jun 2011:00:00:00 GMT

document.cookie = "VisiterName=George; expires=Mon, 12 Jun 2011:00:00:00 GMT; ";

In the above example we have written the date in the pages, but in real life, you can use the Date object to get the current date, and then set a cookie to expire six months or ten months.

Here is the example

cookieExpire = new Date();

cookieExpire.setMonth(cookieExpire.getMonth() + 10);

document.cookie = "VisitorName=George; expires="+ expireDate.toGMTString() + ";";

Copy

The above JavaScript code will create a new cookie called VisitorName with the value of 'George' and it will expire after 10 months from the current date.

path

If a page www.w3resource.com/javascript/ sets a cookie then all the pages in that directory (i.e. ../javascript) and its subdirectories will be able to read the cookie. But a page in www.w3resource/php/ directory can not read the cookie. Usually, the path is set to root level directory ('/') , which means the cookie is available for all the pages of your site. If you want the cookie to be readable in a specific directory called php, add path=/php;.

Here is the example where we have set a specific path called 'php'

```
document.cookie= "VisiterName=George;expires=Mon,12Jun2011:00:00:00"
  + ";path=/php;";
```

In the following code here we have specified that the cookie is available to all subdirectories of the domain.

```
document.cookie= "VisiterName=George;expires=Mon,12Jun2011:00:00:00"
  + ";path=/;";
```

domain

Some websites have lots of domains. The purpose of the 'domain' is to allow cookies to other subdomains. For example a web portal call 'mysite'. It's main site is http://www.mysite.com with a matrimonial site (http://matimonial.mysite.com), a financial site (http://financial.mysite.com) and a travel site (http://travel.mysite.com). By default, if a web page on the travel site sets a cookie, pages on the financial site cannot read that cookie. But if you add domain = mysite

to a cookie, all domain ending with 'mysite' can read the cookie. Note that the domain name must contain at least two dots (.), e.g., ".mysite.com"

Here is the example.

```
document.cookie= "VisiterName=George;expires=Mon,12Jun2011:00:00:00"
  + ";path=/" + domain = mysite.com;";
```

Secure

The last part of the cookie string is the secure part which is a boolean value. The default value is false. If the cookie is marked secure (i.e. secure value is true) then the cookie will be sent to web server and try to retrieve it using a secure communication channel. This is applicable for those servers which have SSL (Secure Sockets Layer) facility.

JavaScript: Creating Cookies

We have already learned the various parts of the cookie like name, value, expires, path, domain, and security. Let's create a simple cookie.

In the following example, we have written a function name 'CookieSet' and set some of its attributes.

Example:

In the following web document four parameters name, value, expires and path parts sent to the function 'CookieSet'. The secure and domain parts are not essential here. Empty values set to expires and path parts and there is a checking for expires

and path parts. If expires receive empty value it will set the expires date 9 months from the current date and if path receives empty value then current directory and subdirectories will be the path. The toUTCString converts the date to a string, using the universal time convention.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head>
<meta http-equiv="content-type" content="text/html; charset=iso-8859-1" />
<title>JavaScript creating cookies - example1</title>
</head>
<body>
<h1 style="color: red">JavaScript creating cookies - example1</h1>
<hr />
<script type="text/javascript">
//This is done to make the following JavaScript code compatible to XHTML.
<![CDATA[
function CookieSet (cName, cValue, cPath, cExpires)
{
```

```
cvalue = encodeURIComponent(cValue);
if (cExpires == "")
{
var cdate = new Date();
cdate.setMonth(cdate.getMonth() + 9);
cExpires = cdate.toUTCString();
}
if (cPath != "")
{
cPath = ";Path=" + cPath;
}
document.cookie = cName + "=" + cValue +"expires=" + cExpires + cPath;
}
CookieSet("Name","George ","","");
alert(document.cookie)
//]]>
</script>
</body>
```

```
</html>
```

Example: **Receive real data from the user and store it in a cookie.**

The following web document receives real data from the user and stores it in a cookie for next one year from the current date.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"

"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">

<head>

<meta http-equiv="content-type" content="text/html; charset=iso-8859-1" />

<title>JavaScript creating cookies - receive real data. example1</title>

</head>

<body>

<h1 style="color: red">JavaScript creating cookies, receive real data. -
example1</h1>

<hr />

<script type="text/javascript">

//This is done to make the following JavaScript code compatible to XHTML.
<![CDATA[

var visitor_name = prompt("What's your name?","");
```

```
var expr_date = new Date("December 30, 2012");

var cookie_date = expr_date.toUTCString();

final_cookie = "Name =" + encodeURIComponent(visitor_name) + ";expires_on = " +
cookie_date;

document.cookie = final_cookie;

alert(final_cookie);

//]]>

</script>

</body>

</html>
```

Delete a cookie

Deleting a cookie is extremely easy. To delete a cookie first accept the name of the cookie and create a cookie with the same name and set expiry date one day ago from the current date. As the expiry date has already passed the browser removes the cookie immediately. It is not confirmed that the cookie has deleted during the current session, some browser maintains the cookie until restart the browser. To see the example read A real example on Cookies page.

Reading Cookies

When a browser opens a web page, the browser reads the cookies (provided it has already stored it) that have stored on your machine. We used document.cookie to retrieve the information about the cookie.

Example:

In the following web document, we receive a name from the visitor and stored it in the cookie called "Name".

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head>
<meta http-equiv="content-type" content="text/html; charset=iso-8859-1" />
<title>JavaScript creating cookies - receive real data. example1</title>
</head>
<body>
<h1 style="color: red">JavaScript creating cookies, receive real data. - example1</h1>
<hr />
<script type="text/javascript">
```

```
//This is done to make the following JavaScript code compatible to XHTML.
<![CDATA[

var visitor_name = prompt("What's your name?","");

var curdate = new Date();

curdate.setMonth(curdate.getMonth() + 6);

var cookie_date  = curdate.toUTCString();

final_cookie = "my_cookie=" + encodeURIComponent(visitor_name) + ";expires_on = "
+ cookie_date;

document.cookie = final_cookie;

alert(final_cookie);

//]]>

</script>

</body>

</html>
```

Example: Retrieves values from cookie

In the following web document, we will read the stored cookie and retrieves its value.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head>
<meta http-equiv="content-type" content="text/html; charset=iso-8859-1" />
<title>JavaScript : Retrieve values from a cookie - example1</title>
</head>
<body>
<h1 style="color: red">JavaScript : Retrieve values from a cookie - example1</h1>
<hr />
<script type="text/javascript">
//This is done to make the following JavaScript code compatible to XHTML.
<![CDATA[
var search_cookie = "my_cookie" + "="
if (document.cookie.length > 0)
```

```
{
// Search for a cookie.
offset = document.cookie.indexOf(search_cookie)
if (offset != -1)
{
offset += search_cookie.length
// set index of beginning of value
end = document.cookie.indexOf(";",offset)
if (end == -1)
{
end = document.cookie.length
}
alert(decodeURIComponent(document.cookie.substring(offset, end)))
}
}
//]]>
</script>
</head>
```

```
</body>

</html>
```

JavaScript: Cookies A Real Example

In the following web document, if a visitor registered his name, his name will be displayed if he returns back to this page for the next nine months. When the visitor registered his name a cookie is stored in the visitor hard drive, to delete this cookie see the next example.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"

"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">

<head>

<meta http-equiv="content-type" content="text/html; charset=iso-8859-1" />

<title>JavaScript Cookie</title>

<script type="text/javascript">

//This is done to make the following JavaScript code compatible to XHTML.
<![CDATA[

function register(name)
```

```
{
var curdate = new Date();
curdate.setMonth(curdate.getMonth() + 9);
cookieExpires = curdate.toUTCString();
final_cookie = "mycookie=" + encodeURIComponent(name) + ";expires_on = " + cookieExpires;
document.cookie = final_cookie;
}
function getCookie(cookie_name)
{
var search_cookie = cookie_name + "="
if (document.cookie.length > 0)
{
start_position = document.cookie.indexOf(search_cookie)
if (start_position!= -1)
{
start_position += search_cookie.length
end_position = document.cookie.indexOf(";", start_position)
```

```
if (end_position == -1)
end_position = document.cookie.length
return (decodeURIComponent(document.cookie.substring(start_position, end_position)))
}
}
}
//]]>
</script>
</head>
<body>
<h1 style="color: red">JavaScript Cookie</h1>
<hr />
<script type="text/javascript">
//This is done to make the following JavaScript code compatible to XHTML.
<![CDATA[
var username = getCookie("mycookie")
if (username)
{
```

```
document.write("Welcome Back, ", username)
}
if (username == null)
{
document.write("You haven't been here in the last nine months...")
document.write("When you return to this page in next nine months, ");
document.write("your name will be displayed...with Welcome.");
document.write('<form onsubmit = "return false">');
document.write('<p>Enter your name: </p>');
document.write('<input type="text" name="username" size="40">');
document.write('<input type = "button" value= "Register"');
document.write('onClick="register(this.form.username.value); history.go(0)">');
document.write('</form>');
}
//]]>
</script>
</body>
</html>
```

Database Management with ASP

Database connectivity with ADO (ActiveX Data Objects) and how we can use this to communicate and manage a database server.

To open a connection to a database, run INSERT, UPDATE, DELETE, and SELECT statements as well as execute a stored procedure.

Active Server Pages, being a scripted language, is a stateless language. By this, we mean that it doesn't preserve the state of the application between page loads. Nearly all ASP hosting solutions will include some sort of database hosting. Most common is MySQL, followed by Microsoft SQL Server. No matter what database you are using, ADO will allow you to manage the database through Active Server Pages.

While it's true that you can store persistant data in the Application object. This is basically just stored in a memory cache that will be flushed whenever IIS or the server is restarted. It also causes problems when trying to create an application which will be used in a server cluster (where each cluster has its own version of the Application cache.)

Connecting to a Database

Here is a sample showing how to open a connection to a database. To do this, we call the Server.CreateObject to create an instance of the built-in ADODB Connection object. Not only is this used to create the initial connection to the database, it will also be used to create the Recordset and Command Objects later.

to replace the text mysqluser, mysqlpass, and mysqldbname with the correct values for your database. Note that this is accessing a MySQL server on the local server (localhost) using ODBC.

```
' create a connection object

Set oConnection = Server.CreateObject("ADODB.Connection")

' open a connection to a data source

oConnection.Open "Driver={MySQL ODBC 3.51 Driver};server=localhost;port=3306;uid=mysqluser;pwd=mysqlpass;database=mysqldbname;Option=16384"
```

INSERT, UPDATE, and DELETE

The following will use the ADODB.Connection object to run an INSERT query on the database. The code is the same whether you are doing INSERT, UPDATE, or DELETE. You only need to change the sQuery variable with your modified query.

SELECT

The following code can be used to run a SELECT statement against a database. You must first open a connection to a database using the code described in the section Connecting to a Database.

```
sQuery = "SELECT companyid, companyname FROM tblCompany"


' create the recordset object to hold the result sets

Set rs = Server.CreateObject("ADODB.Recordset")

' set the number of records to fetch into memory at once (performance)

rs.CacheSize = 256
```

```
' execute the query and populate the recordset

rs.Open sQuery, oConnection, adOpenKeySet, adLockReadOnly, adCmdText

'More examples of SELECT (based on criteria) are:

'*example 1.

sQuery = "SELECT companyid, companyname FROM tblCompany WHERE companyid = 1" '1. See details below
```

1. When sQuery in above code snippet is set as example 1. it will fetch records with companyid having value 1, in this SQL statement WHERE clause is used to define that criteria.

The constants adOpenKeySet, adLockReadOnly, and adCmdText are described in the section ADO Constants. Now that you have opened a recordset, what can you do with it?

Retrieving the Recordset

The following code will show you how to use the recordset object (rs) described in the SELECT section to retrieve values from your database query.

```
' EOF checks for the end of a recordset

If Not rs.EOF Then

    ' The Fields collection accesses fields from a single record

    sFirstName = rs.Fields("firstname").Value

        ' You can enumerate all fields in the record
```

```
        For Each oFld In rs.fields
            Response.Write "Field = " & oFld.Name & "<br>"
            Response.Write "Value = " & oFld.Value & "<br>"
        Next
End If
            ' You can also loop through all records
Do Until rs.EOF
        Response.Write "Name = " & rs.Fields("firstname").Value & "<br>"
        ' make sure you don't forget this next step
        ' MoveNext will move to the next record in the resultset
        rs.MoveNext
Loop
' following is always a good idea to clean up resources ASAP
rs.Close : rs = 0
```

Closing a Database Connection

You should always close your database connection when you are done accessing the database. It is not required, but it is always a good idea.

```
' it's a good idea to close your recordset first (if necessary)
```

rs.Close : rs = 0

' this will close the database connection

oConnection.Close

ADO Constants

Variable	Purpose
'adOpenKeySet	Open a keyset Const adOpenKeySet = 1
adLockReadOnly	Create a database lock for read only access to a table. This is a forward-only cursor which provides the most efficient method for retrieving results. Const adLockReadOnly = 1
adLockPessimistic	This locking method will lock a database record as soon as edits have been made to the database. (Not really used in Active Server Pages) Const adLockPessimistic = 2
adLockOptimistic	Use optimistic record locking on the database records. Meaning a record will only be locked when records have been modified and edits are committed back to the database. (Not

	really used in Active Server Pages) Const adLockOptimistic = 3
adCmdText	Indicates the SQL command being passed is text (an SQL statement) Const adCmdText = 1
adCmdTable	Indicates the SQL command being passed is the name of a database table to open (all rows and fields) Const adCmdTable = 2
adCmdStoredProc	Indicates the SQL command being passed is the name of a stored procedure to execute. When using this call, the parameters must be defined separately. Const adCmdStoredProc = 4
adCmdUnknown	Indicates the SQL command being passed is unknown - the ADO library will do it's best to interpret what type of command was intended. Const adCmdUnknown = 8
adStateOpen	Indicates the current state of the result set. This indicates the recordset is open and data retrieval operations can be performed.

	Const adStateOpen = 1
adStateClosed	Indicates the current state of the result set. This indicates the recordset is closed and data retrieval are forbidden. Const adStateClosed = 0
adExecuteNoRecords	Indicate that no recordset is returned. SQL command is stored procedure call, INSERT, UPDATE, or DELETE statement. Const adExecuteNoRecords = 128
adParamInput	Stored procedure parameter is an input (this is the default) meaning we are just passing data into the procedure. Const adParamInput = 1
adParamOutput	Stored procedure parameter is an output meaning that data will be returned by the procedure. Const adParamOutput = 2
adParamInputOutput	Stored procedure parameter is both an input and an output. Data is passed into the procedure and then returned after execution is complete. Const adParamInputOutput = 3
adParamReturnValue	Indicates that the variable holds the return value from a

	procedure Const adParamReturnValue = 4
adVarChar	Variable character (string) SQL data type Const adVarChar = 200
adChar	Fixed length character SQL data type Const adChar = 129
adInteger	Integer SQL data type Const adInteger = 3
adCurrency	Currency SQL data type Const adCurrency = 6

```
' build the query to run

sQuery ="INSERT INTO Company VALUES ("Microsoft")"

' execute the insert statement

oConnection.Execute sQuery, nRecordsAffected, adCmdText + adExecuteNoRecords

Response.Write "Records Affected: " & nRecordsAffected
```

XML

XML stands for Extensible Markup Language. XML was developed by the W3C (World Wide Web Consortium) between 1996 and 1998 to provide a universal format for describing structured documents and data. The XML is a subset of SGML (Standard Generalized Markup Language) that was designed to comply with the basic need of making data storage independent of any one software package. SGML is a meta language, i.e., a language used for describing other mark up languages.

The SGML existed a long before XML, but never given much acceptance. Basically, SGML is a very complicated language, when used extensively for web. It requires too much of an investment in time, tools and training.

HTML, too, cited complaints like :

- HTML lacks syntactic checking. There is no way to validate HTML code.
- HTML lacks internal structure.
- HTML has not been standardized as with SGML.
- HTML cannot handle data interchange, i.e., HTML tags does not identify the information contained in the document.
- There is no concept of reusing code. HTML cannot reuse its code.

HTML is not extensible.

XML uses the features of SGML. XML is designed in such a way that it is inter-practicable with both SGML and HTML. Also it provides ease of implementation with both. So, XML takes the best of SGML and combines it with some of the best features of HTML. Various features of XML are :

- XML is an abbreviation for Extensible Mark up Language.

- XML is a language developed for the web which is different to any other type of scripting or programming language available before it.
- XML is a framework for defining mark up languages.
- XML is known as a simplified subset of the SGML, the goal of which is to enable generic SGML to be served, received and processed on the web.
- XML is a W3C recommendation. It is recommended by a group of vendors, including Microsoft and Sun, called the World Wide Web Consortium (W3C).
- XML has no fixed collection of mark up tags. You can define your own tags.
- XML allows data to be self-describing, i.e., it is designed to describe data and to focus on what data is rather than specifying how to display it.
- XML is hierarchical in nature which means a nested tag pair cannot overlap with another tag.
- In XML, every opening tag must have a corresponding closing tag.
- XML can be categorized as both a meta-language and a markup language

Benefits of XML

- XML is a simple language as information contained in an XML program can be read and can be easily processed by any computer.
- As XML is W3C standard, various software companies have openly accepted and implemented it in their operations.
- XML can handle high volumes of information especially over the internet and World Wide Web.
- XML can be used with existing web protocols (such as HTTP and MIME) and mechanisms (such as URL's), and it does not impose any additional requirements.

- XML operates on data orientation than readability by humans. Although being humanly readable is one of XML's design goals.
- There are no fixed tags for XML. Any user can add his own set of tags. XML tags do not describe presentation but meaning.
- XML can support data written in some other language too.
- XML data can be sent to some cell phones, which is then formatted by the specification of the cell phone software designer to display text, images and even play sounds.

- **How XML is different from HTML.**
- XML and HTML are identical in many respects. Both are derived from SGML, both have start tag and end tag and both have elements and attributes associated with the tags. So they appear very similar. However, XML is not like HTML. Various differences encountered are :
- The key concept to remember is that XML was designed to describe data and to focus on what the data actually is. HTML, on the other hand, was designed to display content and to focus on how the content looks. In other words, HTML is about displaying information, XML is about describing information.
- There are predefined set of HTML tags designed for we pages and the tag semantics are fixed in the HTML standard whereas XML can have user-defined tags. The tags in XML are not predefined and neither do their semantics.
- In XML, every opening tag must have a corresponding closing tag. However, in HTML pairing is optional for certain tags, and only the opening tag is required.

For example, browsers will accept the following HTML code :

<P>The paragraph contains opening tag only.

<P>This paragraph, too, has opening tag only.

XML, however, has been designed to eliminate this possibility with the rigid rule that every opening tag must have a matching closing tag.

In XML, a nested tag pair cannot overlap with another tag whereas HTML allow tags to appear without satisfying the rules of proper nesting.

For example, Following lines of HTML code contain tags that aren't properly nested :

<STRIKE><BOLD>In this example, code is bold.</BOLD></STRIKE>

^{<SMALL>This is second example.}</SMALL>

These lines of code, however, would be illegal in XML. In XML, the closing tag must match the last-used opening tag.

HTML is not extensible whereas XML adds extensibility. XML's powerful linking mechanisms allow you to link to material without requiring the target to be physically present in the object.

HTML is not international, where as, XML is based on Unicode. Unicode enables XML to handle not just Western-accented characters, but also Asian languages.

XML interact with data

XML is steered to make the data controlling easy and straightforward when integrated for use on the Web. With the combination of XML into existing and new IT developments, it becomes easy to define various documents and their types. But XML is always intended to keep data separated from your HTML. It is important to learn that XML is used to share, exchange, store and carry data.

XML keeps data separated from HTML

The XML was designed to describe data and to focus on what the data actually is. HTML, on the other hand, was designed to display content and to focus on how the content looks. XML is about describing information. HTML is still used to format and display data.

XML can be used to store data inside HTML documents

XML data can be stored inside HTML pages as "Data Islands". As HTML provides a way to format and display your data, XML stores data inside the HTML documents. The data contained in an XML file is of little value unless it can be displayed and HTML files are used for that purpose.

XML can be used as a format to exchange information

In the real world, computer systems and databases contain data in incompatible formats. It is very time-consuming for the developers to exchange data between such systems over the internet. But XML's powerful linking mechanisms allow you to link to material without requiring the link target to be physically present in the object.

XML can be used to share data

XML can also be used to share data with plain text files. The main benefit of XML is that you can take data from a program like MSSQL (Microsoft SQL), convert it into XML, then share that XML with other programs, platforms, etc. Each of these receiving platforms can then convert the XML into a structure the platform uses normally and thus you can communicate between two potentially very different platforms.

A sample XML document for a web page is shown below. It contains all the important parts that are usually contained in XML documents.

Example

```
<?xml version ="1.0"?>

<My.page>

<head>

<title> My home page </title>

<header source = "imagehead.gif "/>

</head>

<body>

<main.title> Welcome to my home page </main.title>

<rule/>

<text>
```

<para>

SORRY! This page is still under construction. Visit this site soon.

</para>

</text>

</body>

<footer source= imagefoot.gif "/>

</My.page>

XML Declaration

[i.e., <? xml version = "1.0" ?>]

The XML declaration identifies what follows it as XML code.

It states which version of the XML standard the code compiles with, and it specifies whether the document can be treated as a stand-alone document or whether DTD (Document Type Definition, [defined later in the tutorial]) must also be retrieved in order to make full sense of the contents.

The XML declaration is a processing instruction (identified by the ? at its start and end).

The Root Element

[i.e., from < My.page> to </ My.page>]

Each XML document must have only one root element, and all the other elements must be completely enclosed in that element.

In XML, an element normally consists of 3 things :

- a start tag.
- content (either text or other elements).
- an end tag.

The name you use in the element start tag must exactly match the name you use in the end tag.

XML is case sensitive, recognizing the difference between uppercase letters and lowercase letters.

An Empty Elements

[i.e., < rule/ >]

Empty elements are a special case in XML.

XML requires you to be much more explicit when dealing with empty elements.

To do so, there is a special empty tag close delimiter, /> , as in the following :

<empty-element />

To use an end tag instead of the special empty tag close delimiter. The element declaration :

< graphic source= " image.gif "/> is therefore interchangeable with

< graphic source= " image.gif ">.....</ graphic>

Attributes

[i.e., <header source="imagehead.gif"/> , <footer source = 'imagefoot.gif"/>]

Element tags can include one or more optional or mandatory attributes.

Attributes can only be specified in the element start tag.

The syntax for specifying an attribute is :

Syntax

<attribute.type.name attribute.name = "attribute.value" >

Example

< fruit taste = "sharp" >

XML deals with multiple declaration of attributes in a unique manner. If an element appears once with one set of attributes and then appears again with a different set of attributes, the two sets of attributes are merged.

The first declaration of an attribute for a particular element is the only one that counts, and any other declarations are ignored.

POSSIBLE QUESTION

Part A (1 Mark)

(Online Examinations)

Part B (8 Marks)

1. Describe in detail about the ASP objects.
2. Write a short note on XML.
3. Explain the benefits of ActiveX controls.
4. Explain the XML document type declaration.
5. Describe in detail about the cookies in ASP.
6. List the Active Server Components and explain of each with example.
7. Give explanation for ASP forms and User Input.
8. How will you access the database in ASP? Explain.
9. Why we need XML? Write the difference between SGML and XML with Example.
10. Write about ASP with example.

KARPAGAM ACADEMY OF HIGHER EDUCATION

(Deemed to be University)
(Established Under Section 3 of UGC Act 1956)
Pollachi Main Road, Eachanari Post, Coimbatore - 641021
(For the candidates admitted from 2015 onwards)

CLASS: III B.COM CA **COURSE NAME: INTERNET AND WEB DESIGN**

COURSE CODE: 15CCU601 **UNIT: V (ACTIVE SERVER PAGES)** **BATCH-2015-2018**

S.NO	QUESTIONS	OPTION 1	OPTION 2	OPTION 3	OPTION 4	ANSWER
1	The Abbreviation for ASP denotes _____.	Active Server Pages	Activex Server Pages	Active Source Pages	Active Script Pager	Active Server Pages
2	Traditional corporate websites are often examples of _____.	Static content	Dynamic content	Both 1 & 2	None	Static content
3	The ASCII text is transferred via _____.	TCP/IP	TCP Only	IP Only	Anonymous FTP	TCP/IP
4	TCP protocol interacts the communication between the _____ and _____.	Client and server	Client and protocol	Server and protocol	None	Client and server
5	_____ was one of the first methods used to create dynamic HTML.	Common Gateway Interface	Creating Gateway Interface	Common Gateway Interaction	Common Gateway Internet	Common Gateway Interface
6	_____ usually represents a hierarchical organization chart.	Object model	Website	Dynamic object model	None	Object model
7	_____ usually represents a hierarchical organization chart.	Object model	Website	Dynamic object model	None	Object model
8	Common Object model is a standard communication mechanism between _____.	Components	Protocols	Scripts	None	Components
9	ASP is a text – based files of a combination of _____ and _____.	Html tags and ASP Scripts	Html tags and Dhtml	XML and Dhtml	None	Html tags and ASP Scripts
10	To control the application logic processed by the server the components used is ___.	Variables	Operators	Statements	All the above	All the above
11	The base directory for our ASP application is referred to as a _____.	Virtual root	Physical root	Root node	None	Virtual root
12	ASP programming is the processing of _____ scripts.	Sever-side	Client-side	Server –client-side	None	Sever-side
13	To differentiate between HTML and Active Script function the delimiters used is __.	<% %>	<! !>	<@ @>	<$ $>	<% %>
14	_____ is used to track and manage individual user session in an application.	Session object	Application object	Request object	None	Session object
15	The Session object has _____ events.	2	3	4	No	2
16	How many fundamental Active Server Page objects are there?	6	5	4	2	6
17	_____ is used to manage all information in the ASP application	Application object	Session object	Server Object	Response object	Application object
18	To control transaction processing using the MTS which object taken place?	The object context object	Server object	Request object	Response object	The object context object

19	To control transaction processing using the MTS which object taken place?	The object context object	Server object	Request object	Response object	The object context object
20	_____ is responsible for sending output from the server to the requesting client.	Request object	Response object	Server object	All the above	Response object
21	The most common Response object methods are _____.	Write	Redirect	Response	All the above	All the above
22	Which of this object will enrich the values on the client's browser?	Cookies	Variables	Functions	None	Cookies
23	The method which uses to send HTTP output to the browser is _____.	Write method	Redirect method	Response method	All the above	Write method
24	The Session object has _____ events.	2	3	4	No	2
25	For retrieving information from the client browser the object used is _____.	Request object	Response object	Object context object	Session object	Request object
26	To display and rotate different advertisements on web pages is _____.	Ad Rotator Component	Advertisement method	Both 1 & 2	None	Ad Rotator Component
27	When the page is refreshed, the Ad Rotator Components displays a _____.	New banner image	Old banner image	Constant	None	New banner image
28	In Ad Rotator Component required files are _____	Rotation Schedule File	Redirection file	Both 1 & 2	None	Both 1 & 2
29	Text file that manages the frequency or schedule of the displayed banner is _____.	Redirection file	Rotation schedule file	Properties	None	Rotation schedule file
30	The requesting browser to the destination site for given banner refers to _____.	Redirection file	Rotation file	Both 1 & 2	None	Redirection file
31	ASP can access databases through the use of the	Active data object	Data Active object	Database	None	Active data object
32	To create a record set quickly which component is used _____.	ADO	DAO	RDO	None	ADO
33	Which method adds a new parameter to the collection?	Append	Delete	Refresh	None	Append
34	Which method saves the changes made to the current record?	Update	Save	Save as	All the above	Update
35	____ is the collection of objects that enable ASP developers to connect to databases.	ADO	Record set	DAO	Data	ADO
36	Which of the following offers a better approach to accessing our data?	OLEDB	ODBC	Both 1 & 2	None	OLEDB
37	The major objects we use with ADO are _____.	Connection	Command	Record set	All the above	All the above
38	_____ property is used to set the permissions for a connection.	Mode	Connection string	Connection Time out	None	Mode
39	Error collection belongs to _____ object?	Connection	String	Source	Help file	Connection
40	Which of the following are the cursor types for record sets?	Forward only	Key set	Dynamic	All the above	All the above
41	The Abbreviation for "BOF" stands for _____.	Beginning of file	Bottom of file	Back of file	Beginning of format	Beginning of file

42	The Abbreviation of “EOF” stands for _____.	End of file	End of format	Entering of file	Exit of file	End of file
43	The actions that we want to perform on a record set is _____.	Update an existing record	Add a new record	Deleting an existing record	All the above	All the above
44	_____ record removes the current record from the record set.	Delete	New	Existing	None	Delete
45	The abbreviation for SQL stands for _____.	Structured Query Language	Statement Query Language	Select Query Language	Simple Query Language	Structured Query Language
46	The Abbreviation for “XML” stands for _____.	Extensible Markup Language	X-Markup Language	Extra markup language	None	Extensible Markup Language
47	In XML, a link that contains only one locator is known as _____.	Simple link	Extended link	Both 1 & 2	None	Simple link
48	For the script tag, the end tag is _____.	Required	Not required	Optional	None	Required
49	Which of these tags specify choices in a SELECT element?	<option>	<select>	Both 1 & 2	None	<option>
50	In HTML, the file name is saved with an extension named _____.	Filename.html	Filename.htm	Both 1 & 2	Filename.h	Both 1 & 2
51	In html, SRC stands for _____.	Source	Screen	Standardized	Attributes	Source
52	In html, for the <img> tag the end tag is _____.	Required	Not Required	Illegal	Optional	Illegal
53	“Defines the layout of frames within a window” refers to _____.	Frameset	Frames	Table	All the above	Frameset
54	“Defines a list item within a list” denotes to _____.			<menu>	<dl>	
55	The paragraph ie., <p> tag the end tag is _____.	Required	Not required	Optional	Illegal	Optional
56	Find the odd man out.	Tables	Frames	Style	Language	Language
57	When element contains the entire document?	<Html> </html>	<Body> </body>		<Center>	<Html> </html>
58	Definition lists are specified by the _____.	<dl> tag	<d> tag	<dd> tag	<dt> tag	<dl> tag
59	_____ is the main resources for “HTML”	Tags	Document	Structure	HTTP	Tags
60	HTML is a subset of _____.	SGML	DHTML	XML	Script	SGML