Bachelor of Computer Applications

(BCA)

HTML Programming

(OBCASE205T24)

Self-Learning Material

(SEM II)

Jaipur National University

Centre for Distance and Online Education

Established by Government of Rajasthan

Approved by UGC under Sec 2(f) of UGC ACT 1956

&

NAAC A+ Accredited

TABLE OF CONTENTS

EXPERT COMMITTEE

Prof. Sunil Gupta
(Computer and Systems Sciences, JNU Jaipur)

Dr. Deepak Shekhawat
(Computer and Systems Sciences, JNU Jaipur)

Dr. Shalini Rajawat
(Computer and Systems Sciences, JNU Jaipur)

COURSE COORDINATOR

Swarnima Gupta,
(Computer and Systems Sciences, JNU Jaipur)

UNIT PREPARATION

Unit Writer(s)	Assisting & Proofreading	Unit Editor
Swarnima Gupta, (Computer and Systems Sciences, JNU Jaipur) (Unit 1-7)	Mr. Kuldeep Sharma (Computer and Systems Sciences, JNU Jaipur)	Dr. Satish Pandey (Computer and Systems Sciences, JNU Jaipur)
Pawan Jakhar (Computer and Systems Sciences, JNU Jaipur) (Unit 8-10)		
Komal Sharma (Computer and Systems Sciences, JNU Jaipur) (Unit 10-14)		

Secretarial Assistance

Mr. Mukesh Sharma

COURSE INTRODUCTION

"Clean code always looks like it was written by someone who cares."
- Robert C. Martin

Welcome to the "HTML Fundamentals" course, a key stepping stone in the world of web development. This course is crafted to introduce you to HTML, the foundational language used to create and structure content on the web. As the backbone of any webpage, understanding HTML is essential for anyone looking to delve into web design and development.

HTML, which stands for Hypertext Markup Language, is the standard language for creating web pages and web applications. It provides the basic structure of a webpage by using a series of elements or tags to define and organize content. This course will guide you through the core components of HTML, offering a practical approach to learning how to construct and manage web content effectively.

This course has 3 credits and is divided into 14 Units. The course begins with an introduction to the fundamental concepts of HTML, including the structure and syntax of the language. You'll learn about HTML elements, which consist of opening and closing tags that delineate different types of content such as headings, paragraphs, and lists. Understanding these basic elements is crucial as they form the building blocks of any web page.

As you progress, you'll explore more advanced HTML elements and attributes. Topics include creating links, adding images, and incorporating multimedia such as audio and video. You'll also learn how to use forms to collect user input and how to format text and layout using HTML tags. These skills are vital for creating interactive and visually appealing websites.

A key component of this course is learning how to ensure your HTML code is both valid and accessible. You'll gain insights into best practices for writing clean, semantic HTML code that adheres to web standards. This includes understanding the importance of semantic HTML tags, which improve both the readability of your code and the accessibility of your web content for users with disabilities.

Throughout the course, you'll engage in practical exercises and projects that allow you to apply your knowledge in real-world scenarios. By the end of the course, you'll have a solid grasp of HTML and its role in web development. You'll be equipped with the skills to create well-structured, accessible web pages and understand how HTML fits into the broader context of web technologies.

Course Outcomes:

At the completion of the course, a student will be able to:

1. Remember about the concept of web application.
2. Illustrate the concepts of interactive web page(s) using HTML, CSS and JavaScript.
3. Build a responsive web site using HTML5 and CSS3.
4. Assess role of HTML and CSS in effective web development.
5. Develop an effective web application using HTML and CSS as per the plan.

Acknowledgements:

The content we have utilized is solely educational in nature. The copyright proprietors of the materials reproduced in this book have been tracked down as much as possible. The editors apologize for any violation that may have happened, and they will be happy to rectify any such material in later versions of this book.

Unit -1

Introduction to Web Development & Related Terms

Learning Outcomes

- Students will be able to understand concept of Web, Internet, Intranet and Extranet
- Students will have a clarity over the need and scope of different types of networks.
- Students will be able to understand the application areas of Internet.
- Students will be able to understand difference between Web and Internet
- Students will be able to understand how client server computing works

Structure

Overview of Web: -

The Internet originated in 1969 with the launch of the Advanced Research Project Agency Network (ARPANET), a national computer network sponsored by the United States. In 1992, the World Wide Web (WWW) was introduced at the Centre for European Nuclear Research (CERN) in Switzerland.

Development of the World Wide Web began in 1989 by Tim Berners-Lee and colleagues at CERN, an international scientific organization based in Geneva, Switzerland. They created the HyperText Transfer Protocol (HTTP) to standardize communication between servers and clients. Their text-based web browser became publicly available in January 1992.

Initially, the web was text-based without graphics, animation, audio, or video content. It provided a multimedia interface to resources available on the internet. Since then, the Internet and the Web have evolved into essential tools for accessing information globally.

The World Wide Web operates as a network service based on standardized protocols, allowing a designated server computer to distribute files across the Internet in a uniform manner. This web standard ensures that programs on different computer platforms can correctly format and display served information. These programs are commonly known as web browsers, facilitating access to information via the Internet.

History of the World Wide Web: The World Wide Web, often abbreviated as "the Web," functions as a global information system and represents the fastest-growing sector of the internet. Fundamentally, the Web comprises interconnected servers using hypertext. Hypertext highlights specific text that, when selected, displays additional information on the corresponding topic.

Similar to other internet services, the Web depends on data interaction and transfer between specialized web servers and personal computers using web browser software, such as Microsoft’s Internet Explorer

1.2 Concept of Intranet, Extranet and Internet: -

"At the point when at least two PCs are associated with one another to share or to move information or data starting with one spot then onto the next is called Computer Network."

There are various sorts of PC organization, for example, LAN, MAN and WAN and so on These sorts are separated by certain boundaries, for example, Distance covered, Media (transmission apparatuses) utilized for association.

Local Area Network (LAN): A Local Area Network (LAN) is a network of computers and peripheral devices connected within a confined area like a school, office building, or home. Its primary purpose is resource sharing such as files, printers, and applications. A basic LAN setup might involve connecting computers and a printer within a home or office environment. Typically, LANs utilize specific transmission mediums and typically comprise fewer than 5000 interconnected devices spread across multiple buildings.

Features of LAN:

1. Private Network: LANs are private networks managed internally without external regulatory oversight.
2. Higher Speed: LANs operate at higher speeds compared to other wide-area network (WAN) systems.
3. Media Access Control: LANs employ various media access control methods like Ethernet and Token Ring.

Metropolitan Area Network (MAN): A Metropolitan Area Network (MAN) spans across an entire city, school campus, or a small region. It is larger than a LAN, which is typically confined to a single building or site. Depending on its configuration, a MAN can cover areas ranging from several miles to tens of miles.

Features of MAN:

1. Geographic Coverage: MANs typically cover towns and cities within a maximum range of 50 km.
2. Medium: MANs commonly utilize optical fibers and cables for transmission.
3. Data Rates: MANs support data rates suitable for cloud computing applications.

Wide Area Network (WAN): A Wide Area Network (WAN) is an extensive network that spans a large geographical area, connecting multiple LANs across long distances using communication methods like telephone lines or radio waves. WANs are commonly used by

enterprises and organizations to facilitate communication and data sharing over significant distances.

Features of WAN:

1. Scalability: WANs have a high capacity, enabling them to connect a large number of computers over vast areas.
2. Resource Sharing: WANs facilitate the sharing of resources across different geographical locations.
3. Connectivity: WANs provide uplinks to connect LANs and MANs to the Internet.

Intranet and Extranet: Intranet, Extranet, and Internet are terms commonly used in computer networking. An Intranet is an internal network of an organization used for internal data exchange. It is completely isolated from the Internet and accessible only to the organization's employees. An Intranet is centrally managed by the organization and is protected by firewalls to secure it from external threats. It serves as a private network for the organization's exclusive use, providing controlled access and enhanced security compared to the Internet.

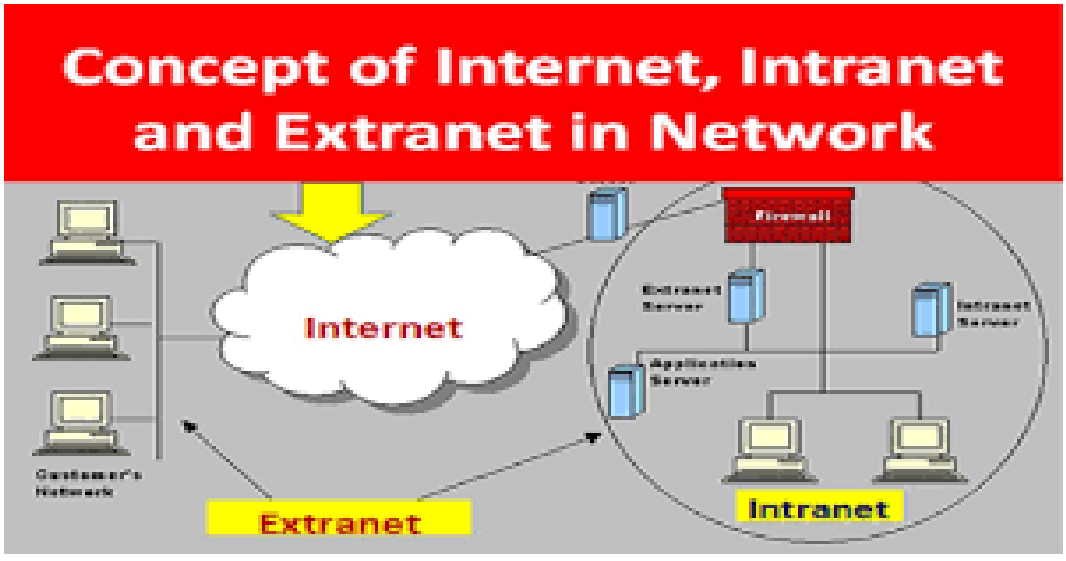


Fig.1 Concept of Internet

Extranet:

An Extranet is an organization which is intended for at least two associations. At some point this may conceivable that at least two associations are subject to one another for business

reason. In this way, it is gainful to frame an organization between the particular business parties. Extranet depends on a similar idea.

An Extranet is a controlled private organization permitting some chosen partners and different organizations to acquire data, normally about a Particular organization or instructive establishment and do as such without giving admittance to the association's whole organization.

An Extranet just contain data with respect to that particular accomplices or organizations. An extranet is likewise secured by firewall. Individuals from this organization can utilize this organization utilizing their ID and Password. This will offer non-interfered with assistance. This organization is administered by various associations (who are connected in this network). Following figure shows the Intranet, Extranet and web.

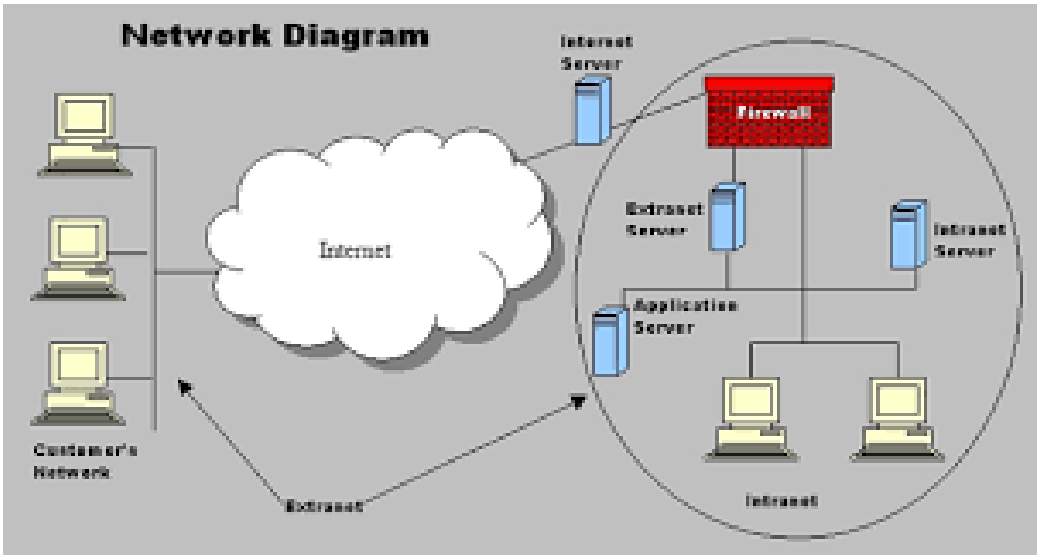


Fig. 2 Network Diagram

Concept of Internet: -

Internet is an organization which has no geological limits it implies it is accessible everywhere on the world. This Internet permits information to share from any spot to anyplace.

"Internet implies Network of Networks"

The Internet is a method of moving data through gadgets associated with one another utilizing wired or remote medium. Web isn't administered by any power. The power which is dealing with Internet is ISOC (Internet Society), this is intentional association whose goal is to advance worldwide data trade utilizing associated Networks. Anybody can get to Internet from anyplace. Web isn't claimed by anybody. utilizing Internet anybody can interface through Internet. In this Internet all the organization from the world are associated with one another and this structures an enormous huge Network of organizations.

Words You Should Know:

- Browser: A browser is software used to locate, retrieve, display, and send data over the Internet.
- Download: Downloading refers to copying data from a remote computer to a local computer.
- Upload: Uploading involves transferring data from a local computer to a remote computer.
- Email: Electronic mail, or email, is the transmission of messages stored on a computer via a communication network. Emails can be sent to individuals or groups and may include attachments such as sound files or graphics.
- Filter: Filtering software blocks access to certain websites. Examples include X Stop and AOL@School.
- Home Page: The starting page of a website that typically serves as an entry point to the site's content.
- HTML (Hyper Text Markup Language): HTML is used to create documents for the World Wide Web. It utilizes tags to structure and format content.
- HTTP (Hyper Text Transfer Protocol): HTTP is a protocol used for transmitting multimedia content over the Internet, including text, graphics, music, and video.
- Hypertext: Hypertext refers to text that contains links to other texts, enabling users to navigate between related information.
- Search Engine: A search engine retrieves information from other websites and indexes it in a database. It provides links to pages that match the user's search query.
- TCP/IP (Transmission Control Protocol/Internet Protocol): TCP/IP is the fundamental communication protocol of the Internet. It facilitates data transmission across networks, including private networks like intranets and extranets, as well as

the public Internet. Every device connected to the Internet uses TCP/IP for communication.

URL (Uniform Resource Locator): A Uniform Resource Locator, or URL, is the address used to access resources on the Internet. It specifies the location of a web page or other resources, indicating the protocol to be used (such as HTTP or HTTPS), the domain name, and optional path and parameters.

WWW (World Wide Web): The World Wide Web (WWW) refers to all the resources and users on the Internet that utilize the Hypertext Transfer Protocol (HTTP). It encompasses a vast array of interconnected documents and multimedia content accessible via web browsers.

What is an Internet Browser?
An Internet browser is essential software used for locating, retrieving, viewing, and sending data over the Internet. It enables users to:

- Send and receive emails instantly to recipients worldwide.
- Access newsgroups for discussions and information exchange on various topics.
- Navigate the World Wide Web to access text, graphics, and interactive content.

The three most popular browsers historically have been Netscape Navigator, Microsoft Internet Explorer, and Google Chrome. Different browsers may render websites slightly differently due to varying interpretations of web standards and technologies.

A URL: Explanation Every server on the Internet has an IP (Internet Protocol) address, which is a unique numerical identifier separated by dots. A URL provides a user-friendly way to access resources by translating the IP address of the server into a readable format, making it easier for users to navigate the Internet.
128.143.22.55 165.113.245.2

But people find it more difficult to recall numbers than word mixtures. This is how "word-based" addresses, or URLs, are assigned to addresses. Moreover, the IP address and the URL are identical.

The accepted practice for providing the address of any Internet resource that is necessary for the “World Wide Web” (WWW). This is how a URL looks:
/seminars.html on www.matisse.net
via telnet to well.sf.ca.us
Gopher at https://gopher.ed.gov

There are sections on the URL:
protocol for transfer/transport ://server (or region). general high level path/filename/area
The Transport protocol is defined in the first segment of a URL.

The HyperText Transport Protocol (http://) transports graphi A file can be moved between two computers using the File Transfer Protocol (FTP) by using http:// cal and hypertext files.

The Gopher client (gopher://) transfers text files. News group reader gopher://news: accesses a discussion thread gro news: up
The Telnet client, telnet:// telnet://, enables remote login to a different machine.
Domains: What Are They?
Indeed. Domains are a component of a website's address, or unified resource locator (URL), and they classify World Wide Web sites according to the type of owner.
Typical top-level domains include:

.com—	business websites;
.mil—	military portal
.org—	website for an organization (non-profits, etc.)
.int—	refers to establishments by international treaty;
.net—	refers to networks;
.biz—	refers to businesses and private websites;
.edu—	refers to educational websites (universities, schools, etc.)
.info—	individual websites;
.gov—	government agencies;
.name—	business and private websites;

- **Knowledge Check 1**
- **State true or false**

1) "WWW" (World Wide Web) was introduced in 1992 at the "Center for European Nuclear Research" (CERN) in Switzerland.
2) Internet is govern by private organization.
3) Anyone can use Intranet.

- **Outcome Based Activity**

Find out example of Intranet, Extranet and discuss its pros and cons.

1.3 Requirement of Internet:

To utilize Internet facility, you require following basic things:

1) A Personal Computer.
2) A Modem and phone line (on the off chance that you are utilizing dial up access) .
3) An Internet program (programming) or Browser and programming to associate you to the ISP.
4) A record with an "Internet Service Provider" (ISP)

1.4 Application areas of Internet.

1) **World Wide Web (WWW):** The World Wide Web (WWW) is a portion of the Internet that displays a single interface for text, images, animation, video, sound, and other mixed media. Since hypertext is a technique for data recovery, it is fundamental to Web activity.
2) **Electronic Mail** (E-Electronic Mail (E-Mail): It is the way toward trading messages electronically, through a correspondence's organization, utilizing the PC.
3) **FTP** stands for "File Transfer Protocol", and it is a configuration of rul es and a commercial package that enables a customer to log in to and transfer data between their PC and another PC.
4) **Telnet:** It links two computers together so that a person can communicate with another computer as if it were being used locally.
5) **Web Relay Chat (IRC):** With this support, individuals can engage in Internet Relay Chat (IRC): le to convey continuously also, carry on discussions by means of the PC with at least one individuals. It gives the client with the office to take part in concurrent (coordinated) on the web 'discussions' with different clients from anyplace on the planet.
6) **Visiting and Instant Messaging**: Chat programs permit Chatting and Instant Messaging: clients on the Internet to speak with one another by composing

continuously. Texting permits a client on the Web to contact another client as of now signed in and type a discussion.

7) **Web Telephony**: It alludes to the utilization of the Int Internet Telephony: ernet instead of the conventional phone organization framework, to trade spoken or other telephonic data.
8) **Video Conferencing**: It utilizes a similar innovation as Video Conferencing: IRC, yet additionally gives sound and video pictures. It empowers direct vis-à-vis correspondence across networks through web cameras, amplifiers, and other specialized apparatuses.
9) **Trade through Internet:** It alludes to purchasing and Commerce through Internet: selling merchandise and ventures on the web.
10) **Newsgroups (**Usenet): It is a worldwide talk about Newsgroups (Usenet): particle bunch that centers around a specific point and helps in get-together data about that theme.
11) **Mailing Lists (List worker**): It alludes to a huge c Mailing Lists (List worker): community of people who convey out dynamic conversations, coordinated around subject situated gatherings that are dispersed through email and this technique is known as mailing list.

1.5 Key difference between Web and Internet

Sr. No	**Parameter**	**WEB/World Wide Web/WWW/W3C/ World Wide Web Consortium**	**Internet**
1	Concept	Web is a content /Information you can see through web.	Internet is a network of connected devices that the web works on.
2	How it works	Web is collection of lots of information linked to each other.	Internet is a collection of different networks connected to each other to form a huge Network.
3	Basic requirement	To access web, you need browser software such as Google Chrome, Internet Explorer etc	To use Internet, you need 1) A Personal Computer. 2) A Modem and phone line (on the off chance that you are utilizing dial up access)

			3) A record with an Internet Service Provider (ISP)
4	Introduced by	Web was introduced by Tim Berners Lee.	Internet was introduced by a team while working on a project.

1.6 Client-Server computing:

Request and response are arranged in a way that facilitates client-server processing. The server receives a request from the client and responds with the optimal data. In order to easily associate with one another, the client and server should adhere to standard correspondence conventions. At the application layer, access to every correspondence convention is possible.

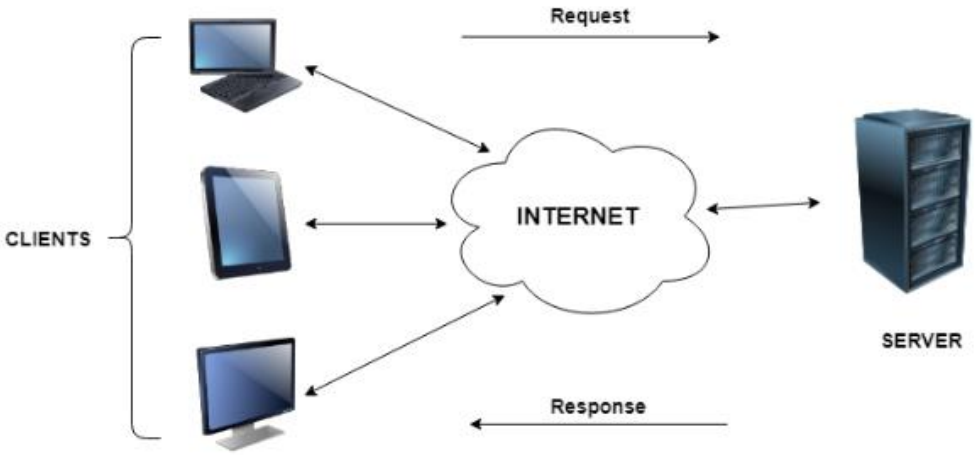


Fig.3 Internet

A server can only fulfill a certain amount of requests from clients at once. As a result, it makes use of a framework that requires it to respond to requests.

A web server is an example of a client-server figuring architecture. The site pages are returned to the clients who requested them.

Client-server processing, the client acts as a client at an inn and the server acts as a server there. This means that in this technique, the client makes requests of the server, and the server responds to the client.

Client-Server Computing's Benefits

Benefits of Computer Networking with Client-Server Architecture Client server computing offers several advantages, including:

1) All the required data is gathered in one location, such as the server. Thus, authorizing and validating users and protecting data is not hard.
2) The server does not have to be physically close to the clients. On the other hand, there are useful ways to obtain the information.
3) Since each hub in the client-server architecture is independent and only needs information from the server, replacing, redesigning, or migrating the hubs is not difficult.
4) Although the client and server hubs may not be as large as other hubs, they can nevertheless contribute to the information sharing process.

The following are some of the drawbacks of client-server computing:

1) The server may become overloaded if too many clients request information at once. This could cause an obstruction within the company.
2) All of the clients' requests cannot be fulfilled in the unlikely event that the server malfunctions for any reason. This causes the client-server organization to become disappointed.
3) A client-server approach is relatively expensive to set up and maintain.

Knowledge Check 2

❖ **State true or false**

1) Mozilla Firefox is a name of Browser.
2) client sends request to server; server gives response on client's requests.
3) Web is collection of computer Networks.

- **Outcome Based Activity**

Discuss Some Applications which are helpful for Education (Any Field)

1.7 Summary:

- The United States' sponsorship of an initiative to establish the Advanced Research Project Agency Network (ARPANET), a public PC network, launched the Internet in 1969. The World Wide Web, or Web, was first introduced in 1992 at the European Center for Nuclear Research (CERN) in Switzerland.

- Internet is an organization which has no geological limits it implies it is accessible everywhere on the world. This Internet permits information to share from any spot to anyplace.
- "Internet implies Network of Networks"
- With the help of Internet, we can do many things anytime and anywhere. It is used in all fields. It makes the task easier & simpler.

1.8 Self-Assessment Questions

1) What is World Wide Web? How it works?
2) Differentiate between Intranet, Extranet and Internet.
3) What is Internet? Explain the advantages & disadvantages of Internet.
4) Explain client -Server computing in detail.
5) Specify basic requirement for using Internet.
6) Explain how Internet works?

1.9 References/Reference Reading

1) Web Technology Theory and Practice -By M Srinivasan
2) Web Technology: A Computer Science Perspective by Jeffrey Jackson.
3) Web Technologies - Black Book

Unit - 2

Introduction to Website Development

Learning Outcomes

- Students will be able to understand concept of Web page, website
- Students will have a clarity over basic types of websites.
- Students will be able to understand different types of website development

Structure:

2.1 Meaning of web page, Website, Basic types of websites: -

A Web page is a page which contains hypertext. Web page is created using Hypertext Markup Language (HTML). A Web page may be static or Dynamic.

Static web page is just used to display data. Dynamic web page is a page which will respond on clients request.

Hypertext is Special text which will link other document to that text. Hypertext displays in blue or Specified color. Normal text and Hypertext identified very easily; Pointer of hypertext will indicate it very easily.

What is Website?

Unlike a Book has number of pages. Website is a huge collection of web pages.

Website may be static or dynamic. Depending on the client's requirement website is developed. It means website may be commercial or free source data providing site or any other.

A website which gives information it means this type of website not responding on any request of client, this type of website is called as Static website.

A website which responds on client's request is called as Dynamic website. This type of website uses client server technique for website development. client means user's machine from where client send request for any online operation to server machine; server is a central machine which stores all data related to website.

2.2 HTML Basics:-

2.2.1 Before you begin HTML:-

Before you begin, it's important that you know Windows or Unix. A working knowledge of Windows or Unix makes it much easier to learn HTML. You should be familiar with:

- Basic word processing using any text editor.
- How to create directories and files.
- How to navigate through different directories.
- Basic understanding on internet browsing using a browser like Internet Explorer or Firefox etc.

2.2.2 How to start with HTML?

Creating an HTML document is easy. To begin coding HTML you need only two things: a basic text editor and a web browser. Notepad is a fundamental example of a simple text editor, and you can use it for a fair amount of HTML coding. However, you might prefer using more advanced editor or IDE software for HTML programming, such as Notepad++

2.2.3 HTML is Forgiving?

An outstanding feature of all browsers is their ability to handle incorrect HTML tags or attributes gracefully. Instead of displaying an error, browsers ignore the faulty tag or attribute and only apply the valid ones. While HTML is a markup language essential for structuring text, its error tolerance doesn't imply that HTML itself is forgiving.

❖ **Knowledge Check 1**

Fill in the Blanks

1) A website which responds on client's request is called as ----------
2) ----------displays in blue or Specified color.
3) ----------page is just used to display data.

Outcome Based Activity

Find out some websites which are static and dynamic.

2.3 HTTP Protocol Request & Response

- The acronym for hypertext transfer protocol is HTTP.
- http is a stateless request/response protocol that runs on a dependable TCP/IP connection and is based on the client-server architecture paradigm. An HTTP server is a program, usually a web server like Apache, that accepts connections in order to serve http requests by sending http response messages. A web browser program that establishes a connection with a server in order to submit one or more http request messages is known as an HTTP client.
- Internet errors, sometimes known as http status codes, may be very annoying, particularly if you don't know the difference between a 404 and a 502 error.
- A frequent HTTP status code, such as "404 file not found," indicates that the web server is unable to locate the file you requested. It could be that the file has moved or been destroyed, or that you typed the erroneous URL or document name.
- The required data regarding the request and response or the object sent in the message body is provided via the HTTP header field. Four different kinds of HTTP message headers exist:

Header in general: It is often applicable for both request and response messages to use these header fields.

Request Header: Only request messages are applicable to these header fields. **Response Header**: Only response messages are applicable to these header fields. **Entity Header:** These header fields specify the meta data pertaining to the entity body.

As previously indicated, when you type a URL into the browser's address bar, the browser converts it into a request message based on the protocol you specify and sends the request message to the server.

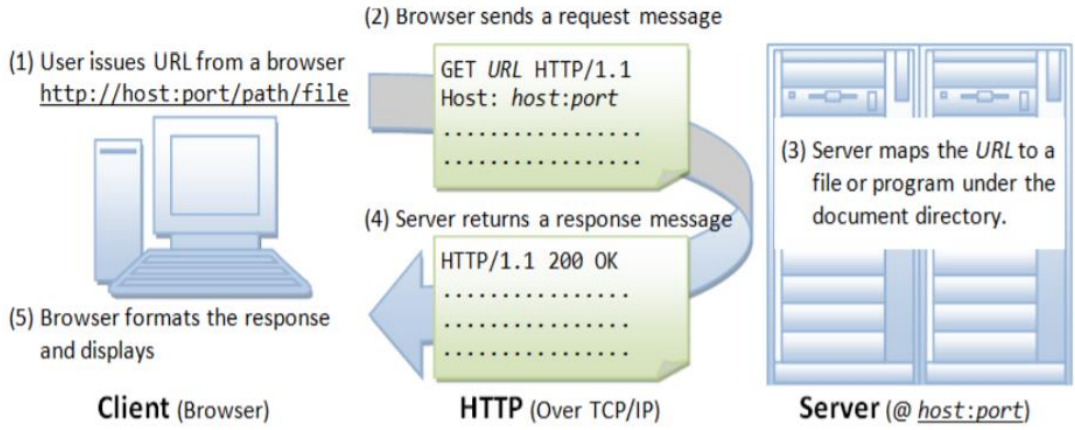


Fig. 2.1: Communication between HTTP client and HTTP server

2.4 Types of websites development: -

A Website always has two sides 1) Front End side and 2) Back End side. Front End side means client side and Back End side means Server side.

2.4.1 Client-Side web development: -

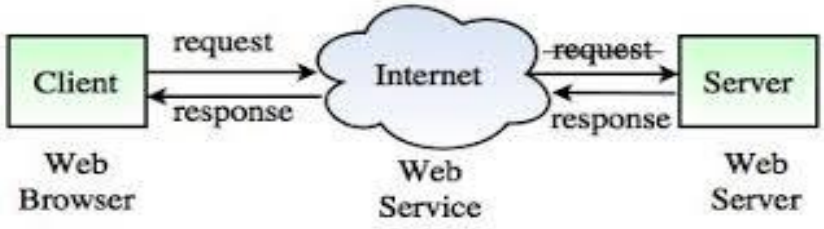


Fig.2.2 Client-Side Scripting

Client side means user side, this is used by client/user through Web browser. Website has two sides front end & back end. Front end side means client side. Web site is developed using specifically designed client-side scripting languages such as HTML, CSS, JAVASCRICPT

Client side always interact with the server. client send request for different operations such as registration, sign in, searching, update, sign out etc. These requests get response from server client server technique is used to complete all these operations. Different user interfaces are developed to complete these operations (Fig.2.2).

2.4.2 Server-Side web development: -

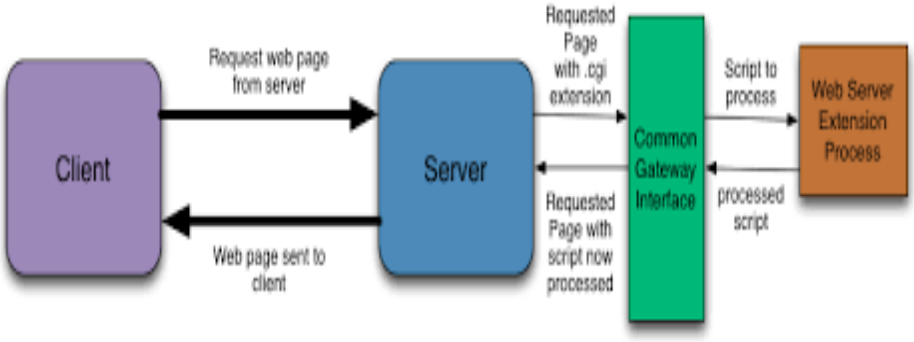


Fig.2.3 Server-Side Script

Server is a machine which gives response on client's request. Server side is also known as back end side of any system. Server is a central machine which stores all data related to website. When client send request to server, server will verify all login details and then give response for client's request (Fig.2.3).

The primary functions of the server side include database-related tasks like record fetching and updating, measuring client input, displaying the pages mentioned, web application design, interacting with servers, querying and interacting with databases, encoding data into HTML, and tasks involving information bases like update and erase.

To develop server-side programs some specific languages are used such as PHP, ASP.NET, JAVA, JSP, PYTHON, RUBY on Rails etc

Developers who develop Client side is known as client-side developer, developers who can develop server side they can know as server-side developers. Developer who knows both sides development is known as Full stack developer.

2.4.3 Full stack web development:

This type of web development consist of development of all aspects of web site. It means that full Stack web development consist of client side, server side & database operations. person who can develop client side, server side and database operations is known as Full Stack Developer.

Understanding client-side languages like HTML, CSS, and JavaScript, server-side languages like PHP, ASP.NET, Java, JSP, Python, RUBY on Rails, etc., as well as necessary frameworks and databases, is necessary for full stack web development.

❖ **Knowledge Check 2**

State TRUE or FLASE

1) Web page is created using Hypertext Markup Language (HTML).

2) PHP is a client side programming language.

3) Client always response on servers request .

Outcome Based Activity

a. **Study different types of Web development languages.**

2.5 Summary

- A Web page is a page which contains hypertext. Web page is created using Hypertext Markup Language (HTML). A Web page may be static or Dynamic.
- A Website always has two sides 1) Front End side and 2) Back End side. Front End side means client side and Back End side means Server side.
- Client side means user side, this is used by client/user through Web browser. Website has two sides front end & back end. Front end side means client side. Web site is developed using specifically designed client-side scripting languages such as HTML, CSS, JAVASCRICPT
- Server is a machine which gives response on client's request. Server side is also known as back end side of any system. Server is a central machine which stores all data related to website. When client send request to server, server will verify all login details and then give response for client's request.

2.6 Self-Assessment Questions

1) What do you mean by web page? what is a website means?

2) Specify different types of websites?

3) Differentiate between Client-side web development and server-side web development.

4) Explain the working of client server technique.

5) What is client-side scripting used for?

2.7 References

1) Web Technology Theory and Practice -By M Srinivasan

2) Web Technology: A Computer Science Perspective by Jeffrey Jackson.

3) Web Technologies - Black Book

Unit -3

Hypertext Markup Language –I

Learning Outcomes

- Students will be able to understand concept of Hypertext Markup Language.
- Students will be able to understand uses & limitations of HTML
- Students will be able to understand concept of tag, attributes, types of tags
- Students will be able to understand basic structure of HTML program
- Students will be able to understand different text formatting tags

Structure

3.1 Introduction to Hypertext Markup Language:

Now a days Web and Internet plays very important role in our day-to-day life. Life becomes easier because of this new era of Internet. Any person can get any information about anything from any place in the world using Internet and web. Website is a main source of information which is globally available to everyone.

Website is a collection of web pages. Each webpage is created using Hypertext Markup Language (HTML). HTML is markup language .HTML is simple, easy to use and modification.

HTML allows programmer to link different documents to a particular text or image. Linking is a key factor of a Website. HTML can run on any platform like Windows, Linux, etc.

Hypertext Markup Language is introduced by Tim Berners Lee in 1990 for making very basic pages in a website. From then on, the language has seen constant development and changes. Today, it has capability of making complex graphs and charts all by itself.

3.1.1 Uses of HTML:

1) HTML is used to create webpage which can store lots of information in textual, image or video format.
2) HTML is used to navigate Internet; it means you can click on different webpages for searching any information.
3) HTML can store all client-side data.
4) HTML is used to make text look attractive.
5) It is a versatile language and can be used on any platform.

3.1.2 Limitations of HTML: -

1) With only using HTML you can create only static web page.
2) You can not create responsive /interactive webpage using HTML.
3) HTML will never give you error message if your program is not correct, it just gives the output of correct program.
4) HTML is not a programming language in true sense.
5) Any type of calculation cannot be done in HTML and it cannot be used to display even system date.
6) Scripting languages like Script or JavaScript are required to handle calculations, validations, and events in HTML documents.
7) No separate/special debugger is provided.
8) Code complexity increases to make a more interactive webpage.
9) Complex HTML code is hard to read and understand.
10) Syntax errors are not identified or displayed by HTML.

❖ **Knowledge Check 1**

❖ **State True or False**

1) HTML is used to navigate Internet.
2) HTML is not a programming language.
3) HTML can store all client-side data.

❖ **Outcome Based Activity** 2

- Discuss different Markup Languages & identify the difference between them

3.2 Concept of Tag, Attributes, denotation of tags: -

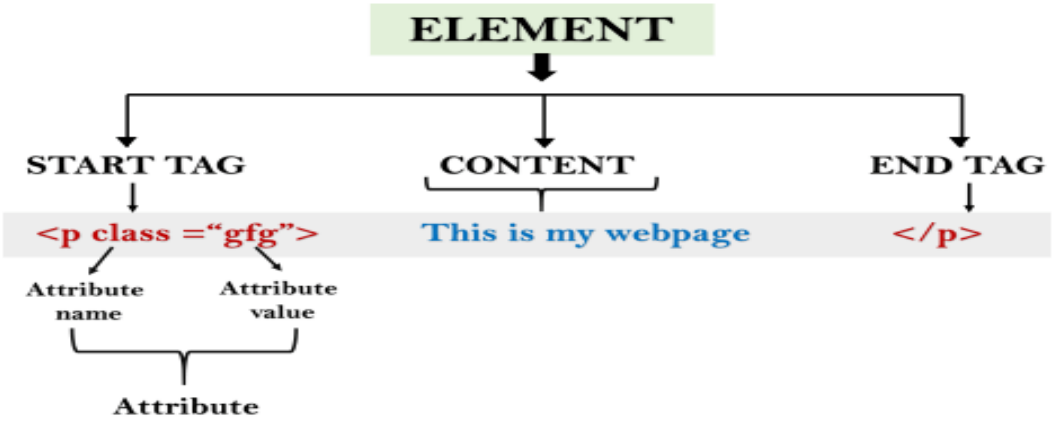


Fig.3.1 HTML Elements

Hyper Text: Hypertext refers to text within a document that contains links to other pages or documents. Clicking on these links allows users to navigate between different web pages or resources, creating a non-linear way of accessing information on the Internet.

Markup Language: A markup language is used to apply formatting and structure to a text document. It enhances the presentation of text by adding elements such as images, tables, links, and more, making the content interactive and dynamic.

Web Page: A web page, typically authored in HTML (HyperText Markup Language) and interpreted by a web browser, is accessed through a URL (Uniform Resource Locator). Web pages can be static, displaying fixed content, or dynamic, where content changes based on user interaction or data input.

HTML (HyperText Markup Language) serves as a markup language to create visually appealing web pages with styling that browsers can render effectively. HTML documents consist of tags that define and structure the content within the page.

Tags: HTML tags are keywords that define how content is structured and displayed by a web browser. Tags consist of an opening tag, content, and a closing tag enclosed within angle brackets <>. Some HTML tags are self-closing and do not require a closing tag.

Syntax:

- Paired Tags: These tags have both opening (<tag>) and closing (</tag>) tags to encapsulate content.
- Singular Tags: These tags are self-closing and do not require a closing tag.

When writing HTML:

- All tags must be enclosed within <> brackets.
- Each tag serves a specific purpose and defines different elements on a web page.
- Paired tags must be properly opened and closed (<tag> content </tag>), while singular tags do not need a closing tag (<tag>).

Understanding HTML tags and their usage is fundamental for creating structured and visually appealing web pages that can be rendered correctly by web browsers.

3.2.1) Basic structure of HTML Program :-

```
<!DOCTYPE html>

<html>
    <head>
        <title>

        </title>
    </head>

    <body>

    </body>
</html>
```

<!DOCTYPE html>: This tag is utilized to tells the HTML rendition. This at present tells that the form is HTML 5.

<html>: This is called HTML root component and used to wrap all the code.

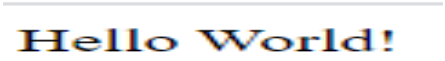
<head>: Head tag contains metadata, title, page CSS and so forth All the HTML components that can be utilized inside the <head> component are:

- <style>
- <title>
- <base>
- <noscript>
- <script>
- <meta>

<body>: Body tag is utilized to encased all the information which a website page has from writings to joins. The entirety of the substance that you see delivered in the program is contained inside this component. For example: -

```
<html>
   <head>
      <title>
         First Web Page
      </title>
   </head>

   <body>
      Hello World!
   </body>
</html>
```

Output: -



1) Heading Styles

Heading Tags: There are six levels of headings defined by HTML. These six heading elements are H1, H2, H3, H4, H5, and H6; with H1 being the highest level and H6 the least.

For example:-

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

```
        <title>
            First Web Page
        </title>
    </head>

    <body>
        <h1> Hypertext markup Language. </h1>
        <h2> Hypertext markup Language.. </h2>
        <h3> Hypertext markup Language.. </h3>
        <h4> Hypertext markup Language.. </h4>
        <h5> Hypertext markup Language.. </h5>
        <h6>Hypertext markup Language.. </h6>
    </body>
</html>
```

Output: -

Hypertext markup Language.

Hypertext markup Language..

Hypertext markup Language..

Hypertext markup Language..

Hypertext markup Language..

Hypertext markup Language..

<title>

The <title> tag is used inside the <head> tag to mention the document title.

3) Body Tag :

The <body> tag defines the document's body.

The <body> element contains all the contents of an HTML document, such as headings, paragraphs, images, hyperlinks, tables, lists, etc.

Following are the attributes of Body Tag ---

1) Bgcolor : This attributes is used to set background color .you can use name of the color or Hexa Decimal code of color or RGB code of color (RGB which stands for **red, green and blue**) is a colour model in which the colours **red, green and blue** are combined in various ways to reproduce a wide array of colours).

<body bgcolor=”red”>

<body bgcolor= #0000FF>

<body bgcolor= **rgb**(0, 255, 255)>

2) Background :- This attributes is used to set image as background. We can set image as background using path of the image file in double quotes.

<body background=”path of image file “>

<body background=”sky.jpg”>

3) Text :- This attribute is used to change default text color .default text color is black.

<body text=”blue”>

3.2.2. Text Formatting Tags:

** - Bold text :-**

In HTML, you can make text bold using the or tags. Here's how you can use them:

Using Tag:

This text is bold.

Using Tag:

I am bold.

Both tags will render the text in bold, but is semantically more meaningful, as it indicates that the text is of strong importance. In contrast, is used purely for visual presentation.

OUTPUT :-

I am bold

** - Important text: -**

The HTML element defines text with strong importance. The content inside is typically displayed in bold. This is paired tag.

**I am strong **

OUTPUT :-

I am strong

** - Italic text: -**

In HTML, you can make text italic using the or tags. Here's how to use them:

Using Tag:

This text is italic.

Using Tag:

This text is italic.

Both tags will render the text in italic, but is semantically more meaningful, as it indicates that the text should be emphasized. In contrast, is used purely for visual presentation.

OUTPUT :-

THis is Italic

<mark> - Marked text :-

In HTML, you can highlight or mark text using the <mark> tag. This tag is used to indicate text that should be highlighted or marked for reference.

Using <mark> Tag:

<mark>cake.</mark>

This will render the text with a yellow background by default, making it appear highlighted.

OUTPUT :-

Do not forget to bring cake today.

<small> - Smaller text :-

In HTML, you can make text smaller using the <small> tag. This tag is used to decrease the font size of the text relative to the surrounding text.

Using <small> Tag:

<small>This is example of smaller text</small>

This will render the text in a smaller font size compared to the normal text size.

OUTPUT :-

This is example of smaller text.

<del> - Deleted text :-

In HTML, you can indicate deleted text using the <del> tag. This tag is used to show that the text has been deleted or is no longer relevant.

Using <del> Tag:

This will render the text with a strikethrough, indicating that it has been deleted.

** yellow ~~blue~~ red. **

OUTPUT :-

yellow ~~blue~~ red.

<ins> - Inserted text :-

In HTML, you can indicate inserted text using the <ins> tag. This tag is used to show that the text has been added or inserted.

Using <ins> Tag:

This will render the text with an underline, indicating that it has been inserted.

** My favorite color is ~~blue~~<ins>red</ins>. **

OUTPUT :-

My favorite color is ~~blue~~ red.

<sub> - Subscript text :-

In HTML, you can create subscript text using the <sub> tag. This tag is used to display text slightly below the normal line of text, which is often used for chemical formulas or mathematical expressions.

Using <sub> Tag:

** This is _{subscripted} text. **

This will render the "2" as subscript text, displaying it below the baseline of the surrounding text.

OUTPUT :-

This is subscripted text.

H_2O

<sup> - Superscript text :-

In HTML, you can create superscript text using the <sup> tag. This tag is used to display text slightly above the normal line of text, which is often used for footnotes, exponents, or mathematical expressions.

Using <sup> Tag:

** This is ^{superscripted} text. **

This will render the "2" as superscript text, displaying it above the baseline of the surrounding text.

OUTPUT :-

This is superscripted text.

2^3

❖ **Knowledge Check 2**

Fill in the blanks

1) The HTML element defines ---------- text.

2) There are ---------- levels of headings defined by HTML.

3) All HTML tags must enclosed within ---------- these brackets.

❖ **Outcome Based Activity 2**

Execute more examples of subscript and superscript

3.3 Summary

- Hypertext Markup Language is introduced by Tim Berners Lee in 1990 for making very basic pages in a website. From then on, the language has seen constant

development and changes. Today, it has capability of making complex graphs and charts all by itself.

- Tags: The material is surrounded by HTML tags, which give it meaning. It is written in brackets (< and >).
- Attribute: In HTML, an attribute is added inside the start tag and gives additional information about the element. The name and value fields are found in an HTML attribute.
- Elements: A single HTML element is a part of an HTML file. Everything written inside tags in an HTML file is referred to as an HTML element.
- To develop a web page you have to follow the structure of HTML program . It consist of multiple tags such as <HTML>,<Head> ,<body>, text formatting tags

3.4 Self-Assessment Questions

1. Explain HTML? Explain structure of HTML program.
2. Explain limitations of HTML.
3. Explain features of HTML.
4. Explain different text formatting tags with example.
5. Explain different heading styles with example.
6. Explain concept of tag, attribute and value

3.5 References/Reference Reading

1) HTML, CSS -The complete Reference –
2) HTML, CSS -DESIGN and build websites by Jon duckett

Unit - 4

Hyper Text Markup Language -II

Learning Outcomes:

- Students will be able to understand concept of ordered list, unordered list and definition list
- Students will be able to create Nested list
- Students will have a clarity over adding image in webpage

Structure:

4.1 List Tag & its Attributes:

Normally the text type in body tag, browser will display this text in a continuous sequence means if user can write all the text & all the no of lines in body section according to the line size browser will complete first line then use second line for displaying.

Generally, to do any work properly we have to plan in a proper manner & for that we have to prepare to do list or sometimes we are going for shopping in this case also to do shopping of daily needs we have to prepare list. This will help to do task complete. On webpage we can display different items using list tag.

Using HTML, you can arrange items in lists in several ways. We can display list on webpage using list tag. The two most commonly used methods are: -

1) Bulleted or Unnumbered (Symbolic / Non-Sequential)
2) Numbered (Sequential)

Bulleted or Unnumbered (Symbolic / Non-Sequential)-This type of list uses symbols to display list such as asterisk (*) or tick (√) or circle etc.

Numbered (Sequential)-This is a type of list which uses a sequence such as alphabets (small (a-z) & capital letters(A-Z)) , Numbers , Roman numbers etc.

Sometimes user want to show his text in a specific sequence like serial no, alphabetical letters or a special symbol “Square “or “Circle”. To develop webpage which displays list, HTML supports 3 different types of list with its appropriate tag –

1) Unordered List (UL)
2) Ordered List (OL)
3) Definition List (DL)

4.1.1 Ordered List (OL) :-

If user want to assign a specific sequence using numbering or lettering then HTML supports type of list that is Ordered list. Ordered list is enclosed within <OL>&</OL>tag. Every item within order list is denoted by, <LI> tag .

To define which sequence user wants to assign in ordered list for that ordered list has following attributes

Sr.No.	Attribute	Description
1	Type	This specifies the numbering type for the list. For example: - Type =”1” – It will give counting numbers (1,2,3…) Type =” A” – It will give Uppercase letters (A,B,C…) Type =” a” – It will give Lowercase letters (a,b,c…) Type =” I” – It will give Uppercase Roman numbers (I,II,III…) Type =” i” – It will give Lowercase Roman numbers (i,ii,iii…)
2	Start	Alters the numbering sequence. This option can change starting value of list items.
3	Value	This changes the numbering sequence in the middle of an ordered list. It is to be specified with the tag .

For Example: -

```
<HTML>
<head> Example of ordered list
<title> List Tag </title >
</head>
<body>
<h1> Hardware Parts </h1>
<ol type =”1”>
        <li> Monitor
        <li> CPU
        <li> Monitor
</ol>
</body>
</HTML>
```

OUTPUT:

4.1.2 Unordered List:

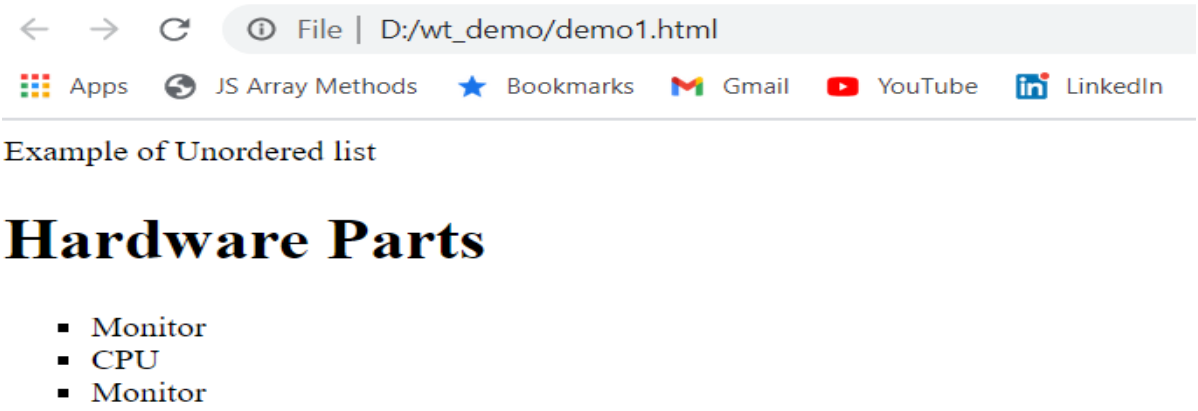
This type of list, which uses specific symbols such as circles or squares for each new line, is called an unordered list. An unordered list begins with the <UL> tag and ends with the </UL> tag. Each item in the list is called a list item and is denoted by the <li> tag. The attribute that can be specified with the <UL> tag is "type".

Sr.No.	Attribute	Description
1	Type	This specifies the numbering type for the list. For example: - Type =" filled circle" or "disc" – It will give solid circle or bullet to list items Type =" circle" – It will give empty circle to list items. Type =" Square" – It will give solid square to list items

For Example: -

```
<HTML>
<head> Example of Unordered list
<title> List Tag </title >
</head>
<body>
<h1> Hardware Parts </h1>
<ul type =square>
        <li> Monitor</li>
        <li> CPU</li>
        <li> Monitor </li>
```

```
</ul>
</body>
</HTML>
```



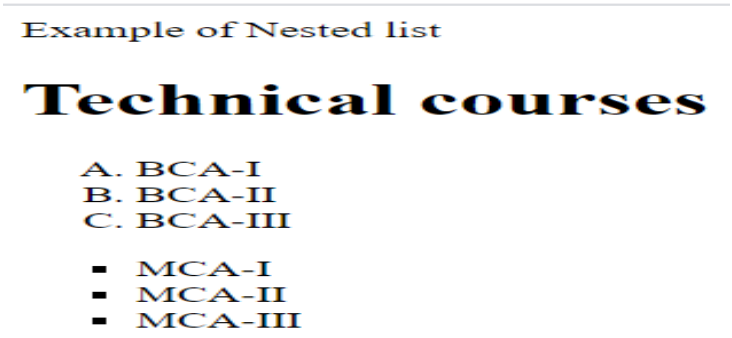
4.1.3 Nested List :-

User can nest several list to create hierarchical structure & can mix order & unorder list for each label of the list . The browser shows a different symbols to make viewing of the page easily .While using nested list be sure to match all the tags ends properly that is <ul> tag ends with . If you forgot one of them the list structure will be incorrect according to browser .user can make a nested list of ordered list & unordered list or combination of them. List are the main frame work to build menus inside the HTML documents.

For Example: -

```
<HTML>
<head> Example of Nested list
<title> List Tag </title >
</head>
<body>
<h1> Technical courses </h1>
<ol type ="A">
        <li> BCA-I</li>
        <li> BCA-II</li>
        <li> BCA-III </li>
</ol>
```

OUTPUT:-



4.1.4 Definition List

Definition lists print the data in key and description format.

Except the numbers or symbols sometime user want to assign only some place to the detail section of any topic. If user want to print first heading from first place on the line from next line, he want to print the description on that heading using a left margin then HTML supports third type of list that is Definition list .This list always appears within <DL>&</DL> Within definition list another two tags must be enclosed to define heading part & detail parts these tags are -

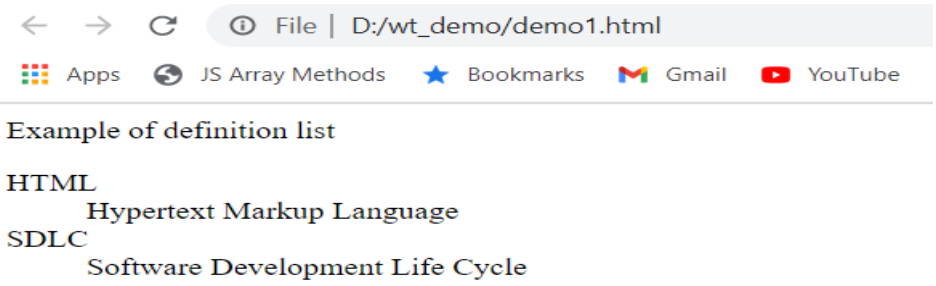
1) Definition Term <DT> :- It gives definition term means topic name or heading name. This will appear within <DT>&</DT>
2) Definition Description <DD> :-

It gives definition description or details of heading topic mention above in <DT> tag

For example:

```
<HTML>

<HEAD> Example of definition list </head>

<BODY>

<DL>

<DT> HTML</DT>

<DD> Hypertext Markup Language

<DT> SDLC

<DD> Software Development Life Cycle

</DL>

</BODY>

</HTML>
```

OUTPUT:



❖ **Knowledge Check 1**

Fill in the blanks.

1) Ordered list enclose within ----------tag

2) HTML stands for ----------

3) ----------attribute of <OL> Alters the numbering sequence.

o **Outcome Based Activity**

Develop a webpage which demonstrate the nested list.

4.2 Adding Graphics /Image in a webpage with its all attributes: -

Including of Images in webpages adds a new dimension to the distribution of information. The image makes the information much more attractive because "A picture is the meaning of thousand words. "Including an Image in webpage is quite easy. It requires only that you have available image for the display & that it be in one of the formats accepted by the browser.

Some graphical files which is standard for browser that are :-

1) .GIF :- GIF stands for Graphic Interchange Format .
2) .JPG :- JPG stands for Joint Photographic Expert Group
3) .BMP :- BMP stands for Bitmap

Images can be inserted in to webpage by using HTML tag <IMG> .The IMG tag takes following attributes

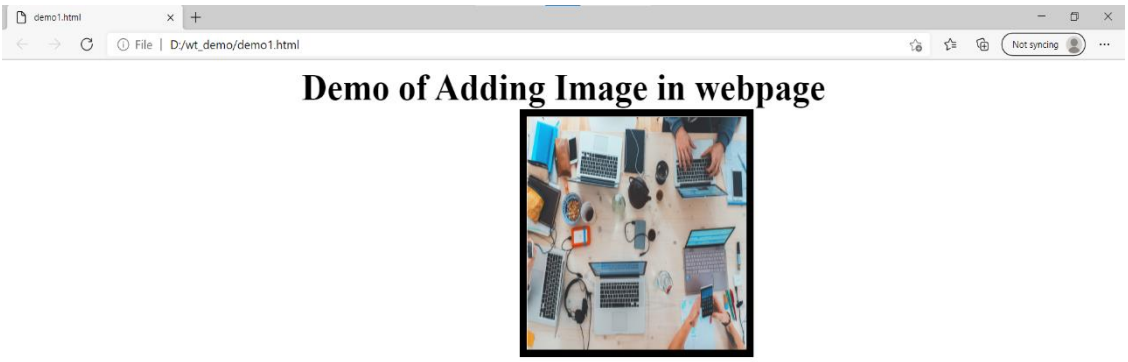
Sr.No.	Attribute	Description
1	Align	It controls the alignment of the text and Image also. It has six different values for Image alignment & Text alignment. Left, Right and Center values specifies the Image alignment across the page. By default, Image will always be displayed at Left side of webpage. Top, Middle and Bottom specifies alignment of the text after the Image tag according to Image position. By default, text will always displayed at Top of the Image.
2	Width	This attribute specifies the size of image in terms of width.

3	Height	This attribute specifies the size of image in terms of height.
4	Hspace (Horizontal Space)	It defines the amount of space to Left & Right of the image.
5	Vspace (Vertical space)	It defines the amount of space to Top & Bottom of the image.
6	Alt	It indicates the text to be displayed when user can moves cursor on Image in browser then the text mention in alt attribute will be display at the bott of the cursor same as tooltip.
7	SRC	It defines the path or address of Graphical file available on your machine which user want to display in webpage
8	Border	This defines whether border is assigned to the Image or not. If user want to assign a border then user can mention appropriate size of border

For example:

```
<HTML>

<HEAD><center><b><font size=30>

                                Demo of Adding Image in webpage

</font></b></center></head>

<BODY>

<imgSrc ="marvin-meyer-SYTO3xs06fU-unsplash.jpg”

                    Height = 300

                    Width = 300

                     Alt = “Use of Graphical Image “

                     Border = 10

                     Align = center

Vsapce = 50

                    Hspace =700>

</BODY>

</HTML>
```

OUTPUT:



❖ **Knowledge Check 2**

Fill in The blanks

1) ---------- defines the amount of space to Top & Bottom of the image.

2**)** By default, Image will always be displayed at ----------side of webpage.

3) JPG stands for ----------

❖ **Outcome Based Activity**

- Create a webpage that demonstrate adding of graphical image at right side of webpage using all its attribute

4.3 Summary:

- We can add content in list format using list tag. list is of three types 1) Ordered list 2) Unordered list and 3) Definition list.
- Ordered list will displays list in sequential manner. ordered list is enclosed within <OL>&</OL> .
- Unordered list will display list in symbolic manner. Unordered list is enclosed within <UL>&</UL>.
- Except the numbers or symbols sometime user want to assign only some spaces to the detail section of any topic. If user want to print first heading from first space on the line from next line, he wants to print the description on that heading using a left margin then HTML supports third type of list that is Definition list. This list always appears within <DL>&</DL>
- We can add graphics in webpage and adjust image using different attributes if <img>tag. Including of Images in webpages adds a new dimension to the distribution of information. The image makes the information much more attractive because “A picture is the meaning of thousand words.”

4.4 Self-Assessment Questions

1) Explain list in HTML. Explain advantages of adding list in webpage.
2) Explain ordered list with suitable example.
3) Explain Unordered list with suitable example.
4) Explain Definition list with example.
5) Explain Image tag with all its attributes.

6) Differentiate between ordered & Unordered list.
7) Explain Nested list .

4.5 References/Reference Reading

1) Web Technology Theory and Practice -By M Srinivasan
2) Web Technology: A Computer Science Perspective by Jeffrey Jackson.
3) Web Technologies - Black Book

Unit - 5

Hyper Text Markup Language -III

Learning Outcomes

- Students will be able to understand concept of how to create different clickable areas in a specific Image
- Students will be able to understand process of Image Maping
- Students will have a clarity over Table creation & all attributes of Table Tags

Structure:

5.1 Image Map,

- **Knowledge Check 1**
- **Outcome Based Activity**

5.2 Table Tag & its attributes

- **Knowledge Check 2**
- **Outcome Based Activity**

5.3 Use of ColSpan

5.4 Use of RowSpan

5.5 Summary

5.6 Self-Assessment Questions

5.7 References/Reference Reading

5.1 Image Map:-

A realistic image that allows a client to tap on different areas and coordinate with different objects is called an imagemap. Each heated zone is described in terms of its x and y coordinates (comparative with the upper left-hand corner) to create imagemaps. You designate a connection to which clients will be coordinated upon clicking within the zone for each set of coordinates.

Let's take an example where you have a world guide that you would want to use as a pictorial map. Each country may have its key spots highlighted on the guide to direct you to other pages.

A huge image with multiple hot spots that can be selected to take the user to distinct anchor destinations is called an image map.

An image map can be described using the HTML <map> tag. An image with interactive zones is called an image map. At least one <region> tag is used to describe each zone.

The idea behind an image map is that depending on where you click in the image, you should be able to do different things.

To construct an image map you need an image, plus some HTML code that portrays the interactive areas. Image map procedure in detail.

Step 1 :-

The <img> tag is used to insert the picture. The addition of a usemap property is the only way this image differs from others:

Sr.No.	Attribute	Description
1	Usemap	To establish a link between the image and the image map, the usemap esteem starts with a hash tag # and ends with the name of the image map.

For Example :-

<imgsrc="office.jpg" alt="Workplace" usemap="#myoffice">

Step 2:-

Create Image Map

Add a <map> component at that position. Using the required name attribute, the <map> component is linked to the image in order to create an image map.The most crucial thing to keep in mind in this situation is that the value of the name attribute and the usemap attribute of the <img> must match.

For Example :-

<map name="myoffice">

Step 3 :-

The Domains Add the clickable spots after that. An element is used to establish a clickable region.

Sr.No.	Attribute	Description
1	Shape	This property is used to specify which shape should be designated as a clickable area.Here are a few of the values: rect designates a rectangular area, Shape="rect" coordinates are provided in pairs, one for the x- and one for the y-axes. circle: denotes an area that is circular, Find the coordinates for the circle's center before adding a circle area. poly denotes a region that is polygonal. When many coordinate points are included in the shape="poly" expression, a polygon—a shape comprised entirely of straight lines—is produced.You can use this to make any kind of shape. default: specifies the whole area. To add the clickable portion to the image, you must additionally specify some coordinates.

For Example :-

```
<!DOCTYPE html>
<html>
<body>
<h2> Demo of Image Maps</h2>
<p>Click on the computer or the cup  to go to a new page and read more about the topic:</p>
<imgsrc="office.jpg" alt="office" usemap="#myoffice" width="800" height="379">
<map name="myoffice">
<area shape="rect" coords="38,44,270,350" alt="Computer" href="f2.html">
<area shape="circle" coords="337,300,44" alt="Cup" href="f3.html">
</map>
</body></html>
```

OUTPUT :-

Demo of Image Maps

Click on the computer or the cup to go to a new page and read more about the topic:

- **Knowledge Check 1**

Fill in the Blanks

1) ----------defines the entire region in imagemap.

2) ---------- attribute is use tell which shape you want to assign as clickable area in Imagemap.

3) ----------tag characterizes an image map.

- **Outcome Based Activity**

Create a web page that will demonstrate the use of Image Map

5.2 Table Tag & its attributes: -

Table isa two-dimensional matrix consisting of rows and columns. Sometimes web developer isrequired to design a webpage in tabular format. HTML supports for tabular documentation. Tables are inserted for displaying data in row and columns format in a webpage.

The Table tag (<table>) is used for creating a table.All tags related with table are included between <table>&</table> tags.

Each row of a table is describedbetween <TR>&</TR>tag. Table rows can be of two types: -

Header Row(TH):-

A Row that spans across the column of a table. A table header row is define using <TH>&</TH>.

The contents of a table header row is automatically central & bold in face.

Data Row (TD):-

Individual data set placed in horizontal column creates a data row. Each column is defined by <TD>&</TD>tag.

For a table it must be given a specific title for identifying information in a table. **The table heading is called Caption**.

Caption Tag: -

Caption can be provided to a table using <caption >&</caption >. This paired tag appeared within <table>&</table> tag. The caption tag has only one attribute that is align. It has two values Top & bottom by default it is displayed at Top of the table.

Attributes of table tag is as follows: -

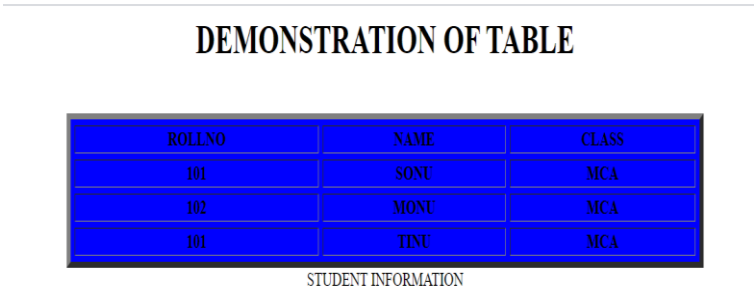
Sr.No.	Attribute	Description
1	Align	Horizontal alignment of a table is control by align attribute. By default, table will display at left hand side of the webpage. align attribute ha three values 1) Left 2) Center 3) Right
2	Width	It sets the width of a table as compare to width of the webpage. It defines with values in percentage format. Eg.-<table width =50%>
3	Border	It controls the border of the table to be place around the table. The border thickness is specified in pixels if user can not specify this

		attribute then browser will display all the table row & column without borders.
4	Bgcolor	If this specified, background color of the whole table.
5	Background	It specifies any graphical picture that user want to assign as a background of the table.
6	Bordercolor	If user can specify the border attribute then it also specify the border color using border color attribute.
7	Cellspacing	This attribute controls the distance between two adjacent cells.
8	Cellpadding	This attribute controls the distance between data in a cell & a boundary of a cell.
9	Title	The message in title attribute will display at bottom of the cursor when cursor moves & place on a table

For Example :-

```
<HTML>
   <H1 align=center> DEMONSTRATION OF TABLE </H1>
   <BODY>
   <TABLE BORDER=5
                BGCOLOR =BLUE
               CELLSPACING=5
               WIDTH=50%
               ALIGN=CENTER >
   <CAPTION ALIGN =BOTTOM >
                                  STUDENT INFORMATION
   </CAPTION >
   <TR>
   <TH> ROLLNO </TH>.
   <TH> NAME</TH>
   <TH> CLASS </TH>
   </TR>
   <TR>
   <TD> 101</TD>.
   <TD> SONU </TD>
   <TD> MCA </TD>
   </TR>
   <TR>
   <TD> 102</TD>.
   <TD> MONU </TD>
   <TD> MCA </TD>
   </TR>
   <TR>
   <TD> 101</TD>.
   <TD> TINU </TD>
   <TD> MCA </TD>
   </TR>
   </TABLE>
   </BODY>
   </HTML>
```

OUTPUT: -



5.3 Use of ColsSpan :-

Sometimes developer wants to create tabular structure in a way that column is again divided into number of columns for this table tag provides two special attributes mainly used in table header row means <TH> tag is sometimes used in table data row that is TD tag.

Colspan is used inside a <TH> tag to give the instruction to the browser to make a cell define by the tag. The colspan attribute can be set equal to number of columns that user want to set.

for example;-

< th colspan= 2>

In case, when user can use colspan attribute after that next row again contains Header row to specify heading to the column that can be divide about.

For Example:-

```
 <html>
<head>
<H1 align=center> DEMONSTRATION OF TABLE </H1>
<BODY>
</head>
<TABLE BORDER=5
        BGCOLOR =yellow
       CELLSPACING=5
       WIDTH=50%
       ALIGN=CENTER >
<CAPTION ALIGN =BOTTOM >
                          STUDENT INFORMATION
</CAPTION >
<TR>
<TH rowspan=2> ROLLNO </TH>.
<TH rowspan=2> NAME</TH>
<TH rowspan=2> CLASS </TH>
<TH colspan=2>Marks </TH>
```

```
</TR>
<TR>
<TH>Web Programming </TH>
<TH> C Programming </TH>
</TR>
<TR>
<TD> 101</TD>.<TD> SONU </TD> <TD> MCA </TD>
<TD> 75 </TD><TD> 82 </TD>
</TR>
<TR><TD> 102</TD>.<TD> MONU </TD><TD> MCA </TD>
<TD> 68 </TD><TD> 69</TD>
</TR>
<TR><TD> 101</TD><TD> TINU </TD><TD> MCA </TD>
<TD> 89 </TD><TD> 78 </TD>
</TR>
</TABLE>
</BODY>
</HTML>
```

OUTPUT :-

DEMONSTRATION OF TABLE

ROLLNO	NAME	CLASS	Marks	
			Web Programming	C Programming
101	SONU	MCA	75	82
102	MONU	MCA	68	69
101	TINU	MCA	89	78

STUDENT INFORMATION

5.4 Use of Rowspan :-

The rowspan attribute functions similarly to the colspan attribute, but instead of spanning multiple columns, it allows a cell to extend across multiple rows. You can set this attribute by assigning it a numeric value.

For Example :-

```
< TH Rowspan = 2>
```

For Example :-

```
 <html>
<head>
<H1 align=center> DEMONSTRATION OF TABLE </H1>
<BODY>
</head>
<TABLE BORDER=5
            BGCOLOR =yellow
           WIDTH=50%
           ALIGN=CENTER >
<CAPTION ALIGN =top >
 STUDENT INFORMATION
</CAPTION >
<TR>
<TH > ROLLNO </TH>.<TH > NAME</TH><TH > CLASS </TH>
</TR>
<TR><TD> 101</TD>.<TD> SONU </TD><TD rowspan = "2"> MCA </TD>
</TR><TR><TD> 102</TD><TD> MONU </TD></TR>
<TR><TD> 103</TD>.<TD> TINU </TD></TR>
</TABLE>
</BODY>
</HTML>
```

OUTPUT :-

DEMONSTRATION OF TABLE

STUDENT INFORMATION

ROLLNO	NAME	CLASS
101	SONU	MCA
102	MONU	
103	TINU	

Knowledge Check 2

Fill in the Blanks

1) ----------defines the entire region.
2) ---------attribute controls the distance between two adjacent cells.
3) Horizontal alignment of a table is control by ---------- attribute.
4) Individual data set placed in horizontal column creates a----------.

❖ **Outcome Based Activity**

Create a web page that will demonstrate the use of Table tag

5.5 Summary

- An imagemap is a realistic image that a customer may interact with by tapping on different areas and interacting with different objections. Each heated zone is described in terms of its x and y coordinates (comparative with the upper left-hand corner) to create imagemaps
- You designate a connection for clients to be coordinated to when they click inside the zone for each set of coordinates.
- A table is a matrix with two dimensions made up of rows and columns. A webpage may occasionally need to be designed in tabular style by a web developer.
- HTML supports for tabular documentation. Tables are inserted for displaying data in row and columns format in a webpage.

5.6 Self-Assessment Questions

1) Explain Image Map with example.
2) Explain Tags related with Image Map.
3) What is Table?
4) Explain tags related with table tag.
5) Explain Table tag with all its attributes.
6) Explain <area>tag from imagemap.
7) Explain use of <caption > tag in table tag.
8) Explain Colspan with suitable Example.
9) Explain Rowspan with suitable example.

5.7 References/Reference Reading

1) HTML, CSS -The complete Reference –
2) HTML, CSS -DESIGN and build websites by Jon duckett

Unit - 6

Hypertext Markup Language -IV

Learning Outcomes:

- Students will be able to understand concept of Linking & types of linking
- Students will be able to understand Anchor Tag & attributes of Anchor tag
- Students will be able to understand concept of Internal linking
- Students will be able to understand text as hyperlink & Image as hyperlink
- Students will be able to understand meaning of Form, Form tag & attributes of form tag
- Students will be able to understand various types form controls
- Students will be able to understand concept & difference between GET & POST
- Students will be able to understand the concept of Frameset & attributes of frameset
- Students will be able to understand concept of Targeted frames

Structure

6.1 Linking of document using Anchor Tag: -

HTML allows linking to other HTML documents as well as Images if user can work on a multiple webpages & he want to develop a web site using that pages then it is required that from one page operate all the available pages from any centralize point, for that from one page user can able to open another page from the same place. only necessary is a single click on text or Image.

Clicking on a section of a text or an Image in webpage, it will open an entire webpage. The text or Image that provides such type of linking called as **Hypertext** or **Hyperlink** or **Hotspot.**

Links are created in web page by using Anchor tag (<A>) .<A> is denoted by a paired tag <A>&</A> anything written between this tag become a hyperlink. Linking is of 2 types –

1) **Links to a specific place within same document /Internal Linking**
2) **Links to an external document / External Linking**

In every linking, link has been established using a anchor tag <A>. It has a main attribute Href (hyper reference) by using this the next web page or image can be specified at the time of specification. Href attribute denotes the path of HTML file or an image file just like SRC attribute in an Imgtag .

Syntax :-
<A Href =" Path of file ">
Hypertext or Hotspot or Hyperlink
</A>

6.2 Types of Linking :-

6.2.1 Links to a specific place within same document / Internal linking :-

This process can be perform using the following steps:-

A) For performing Internal linking user must assign a specific identity name to a specific section or location within a same document for marking the location for jumping from one place to another place .HTML gives another attribute of anchor tag i.e Name attribute .It is use for specifying or marking location in the document .

Syntax :-

<A Name=”s1”>

</A>

B) While linking specify the location name with the same file name using href attribute. For specify the location use ‘#’ symbol after file name without living single space.

Syntax :-

<A href =”path of file #s1 “>

6.2.2 Links to an external document :-

In this type of linking user can open another file from any file, it is called as External Linking. This is categorize in to two:-

A) Text as Hyperlink: -

For Example: -

Demonstration of external Linking

In this above example **“**Demonstration of external Linking” word becomes hyperlink or hotspot between link to another document demo1.html which is present in on d:\.

B) Image as Hyper Link :-

Just as a hypertext, user can use image for an hyper link or hotspot. This image can be made in hyper link by enclosing in <Img> tag between <A>&</A>tag .

For Example :-

<A href=”d:\demo1.html”>

<imgsrc=”d:\ bubbles.bmp”>

</A>

In this above example <img src=”d:\ bubbles.bmp”> this line indicates that this image becomes hyperlink r hotspot between link to an another document demo1.html

❖ **Knowledge Check 1**

Fill in the blanks

1) The text or Image that provides a type of linking called as----------
2) ---------- symbol is use to specify location in Internal Linking.
3) In Image as hyperlink ----------attribute denotes the image.

❖ **Outcome Based Activity**

- ❖ Create a webpage that will demonstrate Linking within a same document.
- ❖ Create a webpage that will demonstrate Image as hyperlink.
- ❖ Create a webpage that will demonstrate Text as hyperlink.

6.3 Form Tag & its Attribute: -

In the HTML documents whatever the tags we study before is for only one-way communication, in the same way the HTML documents open & show the messages. Forms are the tools to improve user interface in the web technology using forms user can design a web page on which user can able to communicate.

To create a form HTML supports number of controls to design a user interface all the controls & the tags include within <form>&</form>tag. A form is defined within this paired tag, it has following attributes: -

Sr.No.	Attribute	Description
1	Action	Forms are used to collect inputs from users, which are then submitted to a server. The `action` attribute in a form specifies the server location where the form data should be sent. This attribute provides the name of the executable program and the directory location, depending on the web server. Alternatively, a complete URL of the host can be entered.
2	Method	It has only two values GET and POST. Generally, the value used is POST. It is a recommended protocol with the POST method, the information from the user is put in to the data stream of http & the backend program can read the data as input.

3	Enctype	This attribute is used to inform the server how to handle the encryption process. The value of this attribute adheres to the communication standard.

6.4 Form Controls: -

In a Form there can be several elements or controls to get inputs from the users. They are as follows: -

- "Drop Down List"
- "Checkbox"
- "Radio Button"
- "Text field"
- "Text area"
- "Password"
- "Button" - i) Submit
  ii) Reset

1) Drop Down List: -

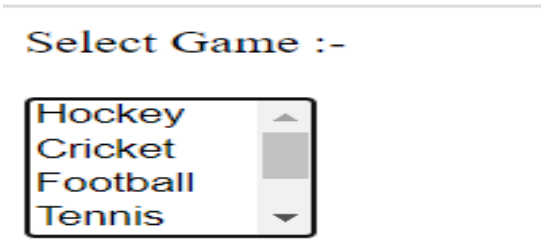
A Dropdown list present a list to the user. The Dropdown list definition begins with <select>& ends with </select >tag. It has three attributes: -

Sr.No.	Attribute	Description
1	Name	The Name attribute assigns a name for a variable. It is common for every element each option of the list is mention with <option> tag. sometimes user wants to assign a specific code for every option within a list & that code value is used for server purpose this is done with value attribute.
2	Size	When list is long, it is difficult to manage on the screen for the user .so user can define the no of items in the list to be displayed on screen for this size attribute.
3	Multiple	In the dropdown list we normally select only one item but in some cases in which user can be given choice to select more than one option .it is done with Multiple attributes.

For Example :-

```
<html>
<form>
Select Game :- <br><br>
<select name ="game" size = 4 multiple="multiple">
<option value ='1'> Hockey
<option value ='2'> Cricket
<option value ='3'> Football
<option value ='4'> Tennis
<option value ='5'> Swimming
<option value ='6'> Badminton
</select>
</form>
</html>
```

Output :-



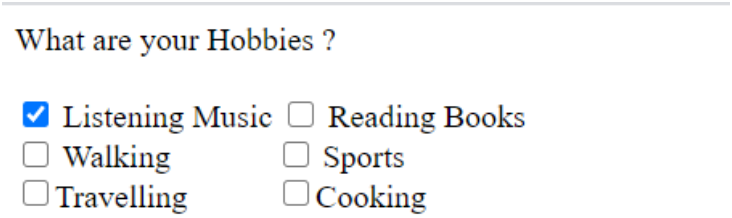
2) Checkbox :-

Checkboxes are another form of user interface to get input. We can get the reply for some 'Yes' or 'No' question using checkboxes. When this appears in a form, the user can select any box by clicking on the box in order to create a checkbox we use <Input> tag with the attribute type =" checkbox ". If you want to assign predefined selection then checkbox element supports one value.

For Example :-

```
<html>
<form>
What are your Hobbies? <br><br>
<input type=checkbox name="m"> Listening Music
<input type=checkbox name="n"> Reading Books <br>
<input type=checkbox name="o"> Walking
<input type=checkbox name="p"> Sports <br>
<input type=checkbox name="q">Travelling
<input type=checkbox name="r">Cooking <br>
</form>
</html>
```

OUTPUT :-



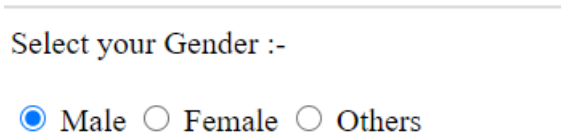
3) Radio Button :-

Radio buttons are used to select only one option from a list of option on the screen. It is exactly opposite in the function to the checkbox. Radio buttons are created using the <Input> tag with type = Radio attribute for predefined selection checked attribute can be used.

For Example :-

```
“<html>
<form>
Select your Gender  :-  <br><br>
<input type=Radio  name="M" checked> Male
<input type=Radio  name="n"> Female
<input type=Radio   name="o"> Others
</form>
</html>”
```

OUTPUT :-



4) Text Field :-

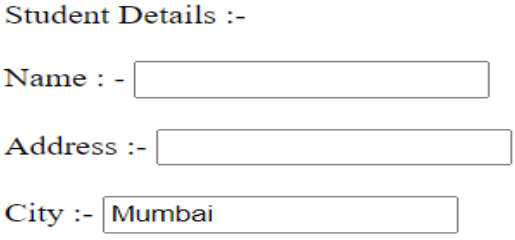
The Text Field provides a single line field in which the user can type text input. A Text field is created using the input tag using type =Text attribute. It has three main attributes: -

Sr.No	Attribute	Description
1	Size	Size attribute defines height of the textbox but in this case, user can enter only a single line.
2	Maxlength	This attribute is used to assign a limitation for textbox, means number of character type by the user up to specific limit.
3	Value	Value attribute is used to display a predefined value in a textbox.

For Example :-

```
“<html>
<form>
Student Details  :-  <br><br>
Name : -  <input type=Text  maxlength =30 ><br><br>
Address :-  <input type=text  maxlength =50 ><br><br>
City :-        <input type=Text   value="Mumbai">
</form>
</html>”
```

OUTPUT :-



5) Password: -

In forms, there is a special type of text field called a password field that collects and sends text input from the user. When the user types in this field, only asterisks (*) are displayed on the screen for security. A password field is created using the `<input>` tag with the `type="password"` attribute.

For Example :-

```
<html>
<form>
Student User ID  Details  :-  <br><br>
Enter your User Name : -  <input type=Text  value="xyz@gmail.com" ><br><br>
Enter your password  :-  <input type=password ><br><br>
</form>
</html>
```

OUTPUT :-

6) Text Area :-

The Text Area is a multi-line area in which user can type the input whenever long text is required then text area control can be used .The text area can be created using <text area > tag .It has three attributes :-

I) Name II) Rows III) Columns

Rows & columns restricts the size of the text area control.

For Example :-

```
<html>
<form>
Student Feedback :-  <br><br>
Enter your experience about this seminar  : -   <br><br>
<textarea  name="t1" rows=10 cols=45 >
</textarea>
</form>
</html>
```

OUTPUT:-

Student Feedback :-

Enter your experience about this seminar : -

7) Submit & Reset Button :-

In every form, two important buttons are typically present: Submit and Reset. The Submit button is used to finalize the user input. When the user clicks on the Submit button, the data provided by the user is sent to the server. The Submit button is created using the `<input>` tag with `type="submit"` and `value="Submit Information"` attributes. The value attribute specifies the caption displayed on the Submit button, which can be customized by the user.

The Reset button, on the other hand, is used to clear the values of all fields in the form that were entered by the user. It allows the user to start over with fresh input. The Reset button is created using the `<input>` tag with `type="reset"`, and similar to the Submit button, the value attribute is used to specify the caption displayed on the Reset button.

For Example :-

```
<html>
<form>
<br>
<br>
<input type=reset value="Reset ">
<input type=Submit value="Submit ">
</form>
</html>
```

OUTPUT :-

6.5 GET & POST Method :-

GET Method:-

GET technique is utilized to attaches structure information to the URL in name or worth pair. On the off chance that you use GET, the length of URL will stay restricted. It encourages clients to present the bookmark the outcome. GET is better for the information which doesn't need any security or having pictures or word archives.

Features of GET:

1) It is not difficult to bookmark information utilizing GET strategy.
2) The length limitation of GET strategy is restricted.
3) You can utilize this strategy just to recover information from the location bar in the program.
4) This technique empowers you to handily store the information.

POST Method: -

POST is a strategy that is upheld by HTTP and portrays that a web server acknowledges the information remembered for the body of the message. POST is regularly utilized by World Wide Web to send client created information to the web server or when you transfer document.

Features of POST: -

1) POST method request gets input from the request body and query string.
2) Information passed utilizing the POST technique won't obvious in question boundaries in program URL.
3) Boundaries of POST strategies are not saved in program history.
4) There is no limitation in sending the length of information.
5) It encourages you to safely pass touchy and private data like login subtleties to server.

6.6 Dividing a screen into multiple Sections / Frames using Frameset Tag: -

In web page using linking multiple files can be handle using different anchor tags but another file takes or occupy the whole area of the page but sometimes if user thinks that in only one page the multiple files can be shown at a time it is not possible using a linking tag. This feature allows the programmer to create several HTML pages that can be simultaneously view in the browser window. HTML supports frame feature for that using it web developer can able to divide browser window in to multiple rows &columns. Each Row or Column is called Frame. Working with frames requires two steps:-

1) User can create the layout of the window first
2) Divides browser screen according to that layout & define contents for each frame

Frameset Tag: -

The frameset tag is of a container type that is it must be open &close. The splitting of a browser screen in to frames is done with <frameset >&</ frameset > tag. The frameset tag contains following attributes: -

Sr.No.	Attribute	Description
1	Rows	It is used to divide the browser screen in to multiple rows depending on the requirement of sight of user. The value of each row can be given in three different forms.

		1) In to no of pixels 2) Express as a percentage of screen width 3) The symbol '*' indicate remaining space.
2	Cols	This is use to create or divide the screen in to number of columns. The value of column can be given same as above three ways. 1) In to no of pixels 2) Express as a percentage of screen width 3) The symbol '*' indicate remaining space.
3	Border	To show the divider or boundary of each frame this attribute must be specified in frameset tag.
4	Border color	To specify the color of the border this attribute can be used.
5	Frame spacing	To show the distance between two consecutive frames some value can be given in frame spacing attribute in to a no of pixels.

For Example: -

```
<frameset rows =”30%, 35%,*” border=5 bordercolor=red framespacing=200>
```

Frame Tag :-

After dividing a browser screen in to a number of rows or columns the next job of user is to assign a specific file to that row or column. This function is performed by < Frame >tag. The frame tag is always include in <frameset >&</frameset >tag. The attributes of frame tags are:

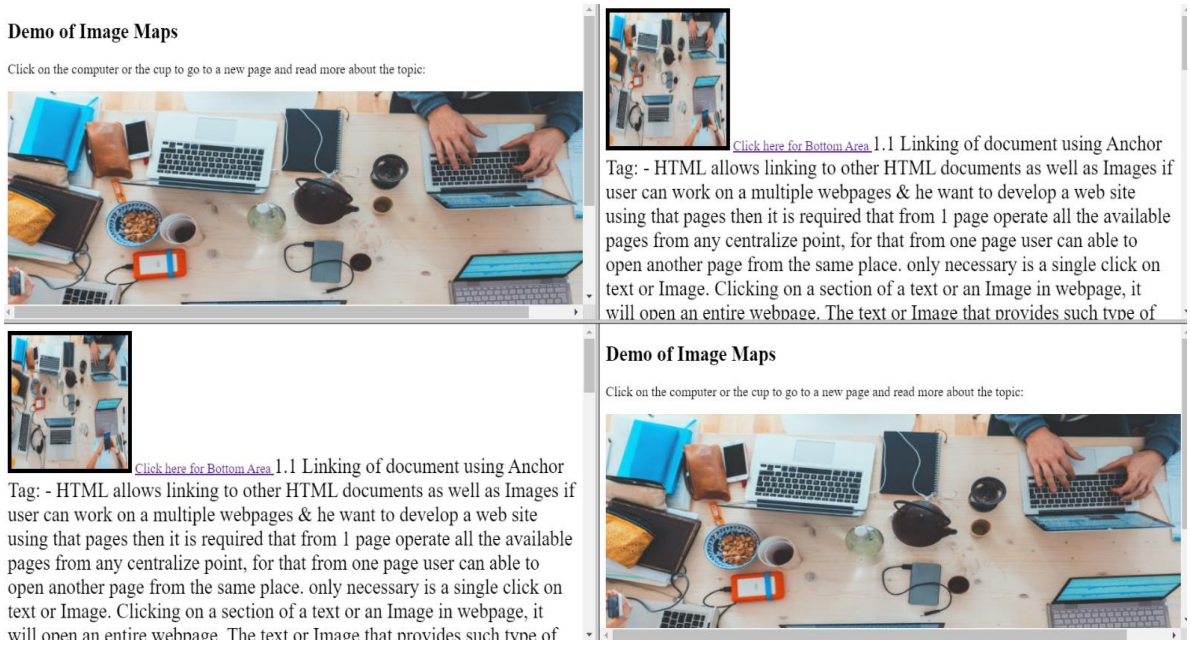
Sr.No.	**Attribute**	**Description**
1	SRC	It specifies URL (Uniform Resource Locator) of the HTML document to be displayed in that frame the name of the file is given in ““(double quotes)
2	Name	A frame is given a name to identify when specifying a target of HTML document.
3	Margin	The Margin-width and Margin -Height specify the left, right, Top & Bottom margin to be maintain the frame.
4	Scrolling	This controls the appearance of horizontal & vertical scroll bars in a frame. By default, scroll bars will appear in a frame according to size of the HTML page. This attribute has 3 values.

		1) Yes 2) No 3) Auto Auto:- Auto value means in this browser decide whether the scroll bar is needed or not. If contents of file is more than the frame. The scroll bar will appear automatically.
5	Noresize	It disables (No Permission) the frame resizing capability. If it is given in frame tag then user cannot allow to decrease or increase size of the frame in browser screen.

For Example :-

```
<frameset rows="50% ,50%">
<frameset cols="50%,*">
<frame src="demo1.html">
<frame src="demo2.html">
</frameset>
<frameset cols="50%,*">
<frame src="demo2.html">
<frame src="demo1.html">
</frameset>
</frameset>
```

OUTPUT :-



6.7 Targeted frame: -

Whenever in a frame a document can be used to which contains hypertext or hyperlink functions which loads a document in a same frame and file reference documents will be replace. The current document which is in the frame sometimes web developer wants to open hyperlink documents in another frame on the same browser screen. The frame in which the hyperlink file is open it needs to be have a specific name means using a hyperlink in one frame document will open in another frame is called Targeted Frame .

Specific name for each frame can be given by using name attribute of the frame tag. In second step second part must be perform in hyperlinking document in that hyperlink tag will have following information:-

1) The file name with extension .HTML that has to be open mention using href attribute.

Example: - (example of Frame division)

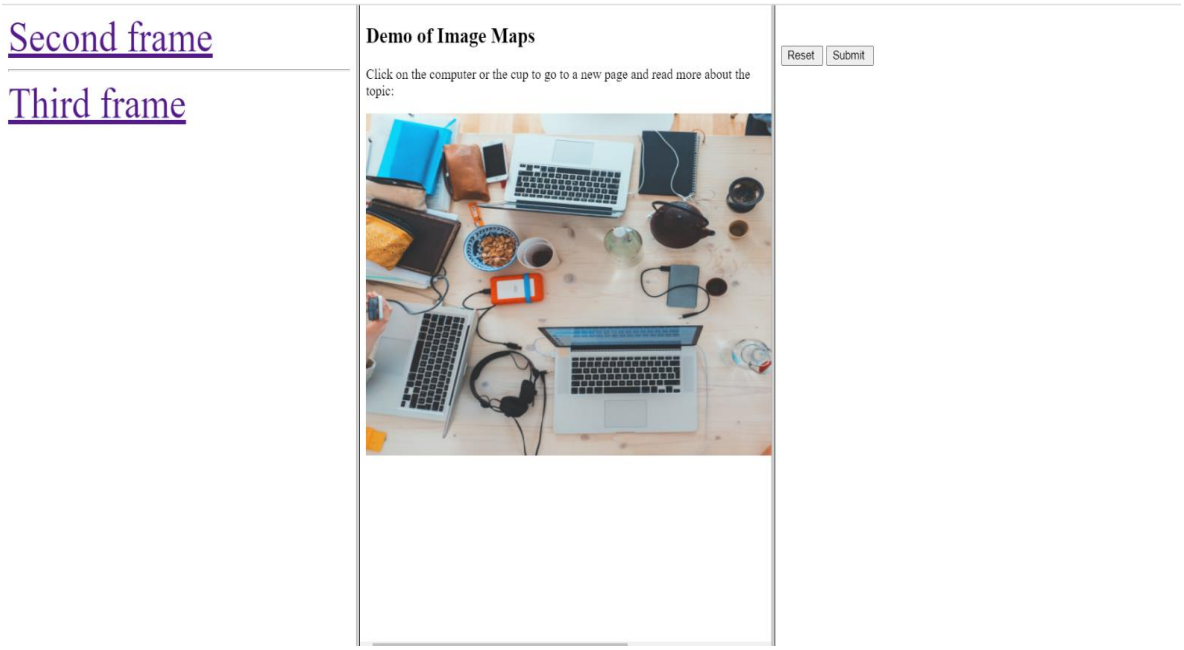
```
<frameset cols="30%,*">
<frame src="tar_frame2.html" name="f1">
<frameset cols="50%,50%">
<frame src="" name="f2">
<frame src="" name="f3">
```

2) The name of the frame where the file has to be open for that the Target attribute is used in anchor tag to show the destination for opening a file.

Example:- (Example of Target Frame)

```
<a href="D:\wt_demo\demo1.html" target="f2">
<font size=50>Second frame <br></font>
</a>
<hr>
<a href="D:\wt_demo\demo3.html" target="f3">
<font size=50> Third frame <br></font>
</a>
```

OUTPUT :-



❖ **Knowledge Check 2**

Fill in the Blanks

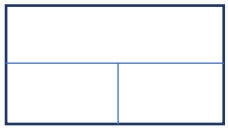
1) ---------- are used to get inputs from user.

2) Method attribute of form tag has only two values ---------- and ----------.

3) The ---------- is a multi-line area in which user can type the input whenever long text is required then text area control can be used.

4) ---------- value of scrolling attribute means that browser decide whether the scroll bar is needed or not.

❖ **Outcome Based Activity**



6.8 Summary :-

- HTML allows linking to other HTML documents as well as Images if user can work on a multiple webpages & he want to develop a web site using that pages then it is required that from one page operate all the available pages from any centralize point, for that from one page user can able to open another page from the same place. only necessary is a single click on text or Image.

- Links are created in web page by using Anchor tag (<A>) .<A> is denoted by a paired tag <A>&</A> anything written between this tag become a hyperlink. Linking is of 2 types –
    - **Links to a specific place within same document /Internal Linking**
    - **Links to an external document / External Linking**
- In the HTML documents whatever the tags we study before is for only one-way communication, in the same way the HTML documents open & show the messages. Forms are the tools to improve user interface in the web technology using forms user can design a web page on which user can able to communicate.
- - The GET method is used to attach form data to the URL in name or value pairs. When using GET, the length of the URL remains limited. This method allows users to bookmark the result. GET is suitable for data that does not require security or involves images or word documents.
- POST is an HTTP-supported method that indicates a web server accepts data included in the body of the message. POST is commonly used by the World Wide Web to send user-generated data to the web server or when uploading a file.
- In web page using linking multiple files can be handle using different anchor tags but another file takes or occupy the whole area of the page but sometimes if user thinks that in only one page the multiple files can be shown at a time it is not possible using a linking tag. This feature allows the programmer to create several HTML pages that can be simultaneously view in the browser window. HTML supports frame feature for that using it web developer can able to divide browser window in to multiple rows & columns. Each Row or Column is called Frame. Working with frames requires two steps: -

  1) User can create the layout of the window first

  2) Divides browser screen according to that layout & define contents for each frame
- The frame in which the hyperlink file is open it needs to be have a specific name means using a hyperlink in one frame document will open in another frame is called Targeted Frame.

6.9 Self-Assessment Questions

1) Explain importance of linking in world wide web.
2) Explain different types of linking.
3) Explain Internal linking with suitable example.
4) Explain External linking with suitable example.
5) Explain <Anchor> tag with suitable example.
6) Explain <form> tag with all its attributes.
7) Explain need of user interface in web site.
8) Explain <frameset> tag with all its attribute.
9) Explain the concept of Targeted Frame and Steps to create targeted frame with suitable example.
10) Differentiate between GET & POST.
11) Explain <Select> tag with example.
12) Explain <Input> tag with example.

6.10 References/Reference Reading

1) HTML, CSS -The complete Reference –
2) HTML, CSS -DESIGN and build websites by Jon duckett
3) Mastering HTML, CSS & JAVASCRIPT web Publishing by Laura Lemay , Rafe Colburn ,Jennifer Keymen
4) Learning Web Design by- Jennifer Niederst Robins

Unit -7

HTML -5

Learning Outcomes

- Students will be able to understand features of HTML 5
- Students will have a clarity over the different tags of HTML 5
- Students will be able to understand the different events in HTML 5 with examples
- Students will be able to understand Input tags in HTML 5 with example
- Students will be able to understand Graphics & Media tags in HTML 5.

Structure

7.1 Key features of HTML-5 :-

HTML5 is the following significant amendment of the HTML standard supplanting HTML 4.01, XHTML 1.0, and XHTML 1.1. HTML5 is a norm for organizing and introducing content on the World Wide Web.HTML5 is a participation between the World Wide Web Consortium (W3C) and the Web Hypertext Application Technology Working Group (WHATWG).

The new standard joins highlights like video playback and simplified that host been beforehand subject to third-gathering program modules like Adobe Flash, Microsoft Silverlight, and Google Gears.

The most recent renditions of Apple Safari, Google Chrome, Mozilla Firefox, and Opera all help numerous HTML5 highlights and Internet Explorer 9.0 will likewise have uphold for some HTML5 usefulness. The portable internet browsers that come pre-introduced on iPhones, iPads, and Android telephones all have brilliant help for HTML5**.**

Features of HTML 5: -

1) It permits you to play a video and sound record.
2) It permits you to draw on a Canvas.
3) It encourage you to configuration better structures and construct web applications that work disconnected.
4) It gives you advance highlights to that you would ordinarily need to compose JavaScript to do.

The DOCTYPE: -

DOCTYPEs in more seasoned renditions of HTML were longer in light of the fact that the HTML language was SGML based and consequently required a reference to a DTD.

HTML 5 creators would utilize basic grammar to determine DOCTYPE as follows −

<!DOCTYPE html>

The above statement is case-sensitive.

7.2 HTML5 Tags: -

<section>: This element defines a thematic grouping of content, which is typically accompanied by headers (h1-h6) to establish the structure of the document.

<article>: This element designates a standalone segment of content within a document, such as a blog entry or a news article.

<aside>: This element represents content that is related but not central to the main content of the page.

<header>: This element signifies the introductory content or navigational links for its nearest ancestor sectioning content or sectioning root element.

<footer>: This element represents the concluding content of its nearest ancestor sectioning content or sectioning root element, often containing information about the author, copyright details, and other metadata.

<nav>: This element represents a section of the document intended specifically for navigation links.

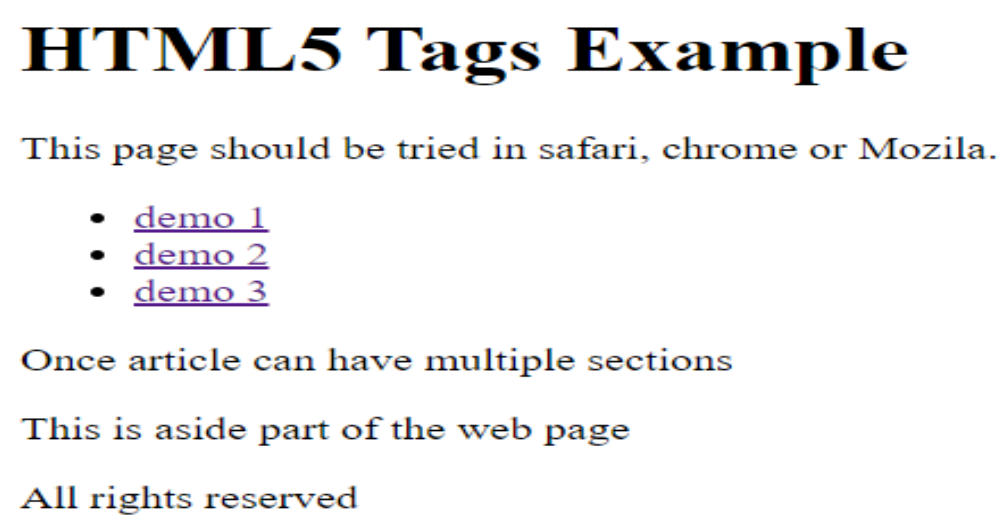
<dialog>: This element is used for marking up interactive components like dialogues and modals.

<figure>: This element is used to encapsulate a piece of media content, like an image or a video, along with a caption.

For example :-

```
<!DOCTYPE html>
<html>
<head>
<meta charset = "utf-8">
<title>...</title>
</head>
<body>
<header role = "banner">
<h1>HTML5 Tags Example</h1>
<p>This page should be tried in safari, chrome or Mozila.</p>
</header>
<nav>
<ul>
<li><a href = "demo1.html">demo 1 </a></li>
<li><a href = "demo2.html">demo 2</a></li>
<li><a href = "demo3.html">demo 3
</a></li>
</ul>
</nav>
<article>
<section>
<p>Once article can have multiple sections</p>
</section>
</article>
<aside>
<p>This is  aside part of the web page</p>
</aside>
<footer>
<p>All rights reserved</p>
</footer>
</body>
</html>
```

Output :-



❖ **Knowledge Check 1**

State whether the statement is True or False

1) Dialog tag can't be utilized to increase a discussion.

2) HTML 5 encourages you to configuration better structures and construct web applications that work disconnected.

3) Nav tag addresses a footer for a segment and can contain data about the creator, copyright data, and so on.

❖ **Outcome Based Activity**

Discuss the basic difference between HTML 4 & HTML 5

7.3 Events in HTML5: -

"In programming, an event is an activity that happens because of the client or another source, for example, a mouse click."

We can compose event handlers in JavaScript or VBScript, and these event handlers are represented by event attribute properties in HTML5. The HTML5 specification defines various event attributes as listed below. We can use the following set of attributes to execute any JavaScript or VBScript code provided as a value when any event occurs for an HTML5 element.

Mainly there are five different types of events. They are as follows: -

1) Window Events
2) Form Events
3) Keyboard Events
4) Mouse Events
5) Drag Events

1) Window Events: -

Events set off for the window object (applies to the <body> tag):

Attribute	Value	Details
onafterprint	*script*	Executes after the associated document is printed.
onbeforeprint	*script*	Executes before the associated document is printed.
onbeforeunload	*script*	Executes before a document being unloaded.
onerror	*script*	Executes when document errors occur.
onhashchange	*script*	Executes when the fragment identifier part of the document's URL i.e. the portion of a URL that follows the sign (#) changes.
onload	*script*	Executes when the document has finished loading.
onmessage	*script*	Executes when the message event occurs i.e. when user sends a cross-document message or a message is sent from a worker with postMessage() method.
onoffline	*script*	Executes when the network connection fails and the browser starts working offline.
ononline	*script*	Executes when the network connections returns and the browser starts working online.
onpagehide	*script*	Executes when the page is hidden, such as when a user is moving to another webpage.
onpageshow	*script*	Executes when the page is shown, such as when a user navigates to a webpage.
onpopstate	*script*	Executes when changes are made to the active history.
onresize	*script*	Executes when the browser window is resized.
onstorage	*script*	Executes when a Web Storage area is updated.
onunload	*script*	Executes immediately before the document is unloaded or the browser window is closed.

2) Form Events :-

Attribute	Value	Details
"Onblur"	*"script"*	Executes when an element loses focus.
"onchange"		Executes when the value or state of the element is changed.
"onfocus"		Executes when the element receives focus.
"oninput"		Executes when the value of an element is changed by the user.
"oninvalid"		Executes when a submittable element does not satisfy their constraints during form validation.
"onreset"		Executes when the user resets a form.
"onselect"		Executes when some text is being selected or the current selection is changed by the user.
"onsearch"		Executes when the user writes something in a search input field.
"onsubmit"		Executes when a form is submitted.

3) Keyboard Events: -

Attribute	Value	Details
"Onkeydown"	*"Script"*	Executes when the user presses a key.
"Onkeypress"		Executes when the user presses an alphanumeric key.
"Onkeyup"		Executes when the user releases a key.

4) Mouse Events :-

Attribute	Value	Details
"onclick"	*"Script"*	Executes when the user clicks the left mouse button on the element.
"ondblclick"		Executes when the user double-clicks on the element.
"oncontextmenu"		Executes when a context menu is triggered by the user through right-click on the element.
"ondrag"		Executes when the user drags an element. The ondrag event Executes throughout the drag operation.
"ondragend"		Executes when the user releases the mouse button at the end of a drag operation.

“ondragenter”		Executes when the user drags an element to a valid drop target.
“ondragleave”		Executes when an element leaves a valid drop target during a drag operation.
“ondragover”		Executes when an element is being dragged over a valid drop target.
“ondragstart”		Executes when the user starts to drag a text selection or selected element.
“ondrop”		Executes when the mouse button is released during a drag-and-drop operation i.e. when dragged element is being dropped.
“onmousedown”		Executes when the mouse button is pressed over an element.
“onmousemove”		Executes when the user moves the mouse pointer over an element.
“onmouseout”		Executes when the user moves the mouse pointer outside the boundaries of an element.
“onmouseover”		Executes when the user moves the mouse pointer onto an element.
“onmouseup”		Executes when the user releases the mouse button while the mouse is over an element.
“onmousewheel”		Deprecated Use the onwheel attribute instead.
“onscroll”		Executes when the user scrolls the contents of an element by scrolling the element's scrollbar.
“onshow”		Executes when a contextmenu event was fired onto an element that has a contextmenu attribute.
“ontoggle”		Executes when the user opens or closes the <details> element.
“onwheel”		Executes when the user scrolls the contents of an element by rolling the mouse wheel up or down over an element.

5)Drag Events :-

Attribute	Value	Description
“Ondrag”	*“Script”*	This will run when an element is dragged
“Ondragend”		This will run at the end of a drag operation

“Ondragenter”		This will run when an element has been dragged to a valid drop target
“Ondragleave”		This will run when an element leaves a valid drop target
“Ondragover”		This will run when an element is being dragged over a valid drop target
“Ondragstart”		This will run at the start of a drag operation
“Ondrop”		This will run when dragged element is being dropped
“Onscroll”		This will run when an element's scrollbar is being scrolled

Input tag in HTML5: -

(Type, Auto focus, placeholder, required etc. attributes.) :-

Input tag is very important tag for insertion of records through forms. Input tag is used to create different controls through which any user can insert data in system. Following are some new attributes of Input tag.

Attribute	Details	Example
Type	This is use to Specify the type of control to insert. Button Checkbox color date “datetime” “datetime-local” “email” “file” “hidden” “image” “month” “number” “password” “radio” “range” “reset”	1)<input type="text"> 2)<input type=" radio">

	“search” “submit” “tel” “text” “time” “url” “week”	
Autofocus	Specifies that an <input> element should automatically get focus when the page loads.	<input type="text" id="f1" name="f1" autofocus>
placeholder	Specifies a short hint that describes the expected value of an <input> element	
required	Specifies that an input field must be filled out before submitting the form	<input type="text" id="username" name="u1" required>
checked	This Boolean attribute specifies that the control is selected by default (for type="radio" or type="checkbox").	

7.5 Graphics in HTML5 :-

The HTML <canvas> tag serves the purpose of creating graphics on a webpage. The graphic on the left-hand side is generated using <canvas>, showcasing four elements: a red rectangle, a gradient rectangle, a multicolor rectangle, and a multicolor text.

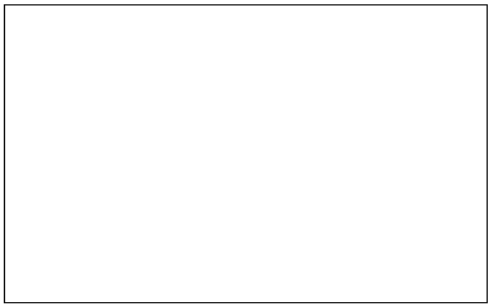
HTML Canvas is a feature used for drawing graphics dynamically on a webpage via JavaScript. The <canvas> element acts as a container for these graphics, but actual drawing operations are performed using JavaScript.

Canvas provides various methods for drawing paths, shapes like boxes and circles, text, and incorporating images. Essentially, a canvas represents a rectangular region on an HTML page, initially devoid of any border or content.

```
<!DOCTYPE html>
<html>
<body>
<canvas id="myCan" width="300" height="200" style="border:1px solid #000000;">
This is  the HTML canvas tag.
</canvas>
</body>
</html>
```

Example :-

OUTPUT: -



7.6 Media tags in HTML5 :-

Multimedia on the web encompasses audio, music, videos, movies, and animations.

What is Multimedia?

Multimedia comes in a wide range of arrangements. It very well may be nearly anything you can hear or see, similar to pictures, music, sound, videos, records, movies, animations, and that's just the beginning. Site pages frequently contain multimedia components of various kinds and arrangements.

Following are some of file formats used for adding Media in web page :-

Sr.No	Media	Supporting File Formats	Format Extension
1	**Audio**	MIDI, WAV, Ogg, mp3,mp4 ,WMA ,Real audio ,AVI	MIDI - .mid or .midi WAV - .wav Ogg - .ogg mp3 - .mp3 mp4 - .mp4 WMA - .wma

			Real audio -.rm or .ram
2	**Video**	MP4, WebM, MPEG,Ogg, AVI	MP4 --.mp4 WebM-- .webm MPEG-- .mpeg Ogg -- .ogg AVI -- .avi Flash - .swf .flv

Adding Video in Web page :-

To add video file in web page we have to use <video >tag .It is a paired tag .it has different attributes they are as follows :-

Sr.No	**Attribute**	**Details**
1	Controls	This attribute adds video controls, like play, pause, and volume.
2	Height	This attribute adjusts the height of the video.
3	Weight	This attribute adjusts the width of the video.
4	Source	You can define other video files that the browser can select by using this feature. The first format that the browser recognizes will be used.
5	Autoplay	Autoplay attribute will play your video start playing automatically
6	Muted	This will keep your video on mute mode.

For Example :-

```
<!DOCTYPE html>
<html><body>
<video width="420" height="440" autoplay muted controls>
<source src="D:\wt_demo\cute_baby.mp4"  type="video/mp4">
<source src="D:\wt_demo\Cute_baby.ogg"  type="video/ogg">
</video></body>
</html>
```

OUTPUT :-

Adding Audio in Web page :-

To add Audio file in web page we have to use <Audio >tag .It is a paired tag .it has different attributes they are as follows :-

Sr.No	Attribute	Details
1	Controls	This attribute adds audio controls, like play, pause, and volume.
2	Source	You can define other audio files that the browser can select by using this feature. The first format that the browser recognizes will be used.
3	Autoplay	Using the autoplay attribute will cause an audio file to play automatically.

For Example: -

```
<!DOCTYPE html>
<html>
<body>
<audio  autoplay muted controls>
<source src="D:\wt_demo\AUD-20190806-WA0009.mp3"  type="audio/mp3">
<source src="D:\wt_demo\AUD-20190806-WA0009.mp3"  type="audio/ogg">
 Demonstration of Audio Tag
</audio>
</body>
</html>
```

OUTPUT:-

- **Knowledge Check 2**

  Fill in the Blanks

1) ----------tag is utilized to draw graphics on a page.

2) ----------adds audio controls, like play, pause, and volume.

3) In programming, an ---------- is an activity that happens because of the client or another source.

- **Outcome Based Activity**

  Demonstrate the Audio & Video tag.

  Demonstrate the events in HTML 5 .

7.7 Summary :-

- HTML5 is the following significant amendment of the HTML standard supplanting HTML 4.01, XHTML 1.0, and XHTML 1.1. HTML5 is a norm for organizing and introducing content on the World Wide Web.HTML5 is a participation between the World Wide Web Consortium (W3C) and the Web Hypertext Application Technology Working Group (WHATWG).
- The new standard joins highlights like video playback and simplified that host been beforehand subject to third-gathering program modules like Adobe Flash, Microsoft Silverlight, and Google Gears.
- These are importantTags of HTML 5 -Section, Article, aside, header, foot figure etc.
- In programming, an event is an activity that happens because of the client or another source, for example, a mouse click."
- Our event handlers can be written in Script or JavaScript, and you can use them to estimate the event label property. Different event credits are characterized by the HTML5 specific, as seen below.
- Mainly there are five different types of events. They are as follows: -
    - Window Events
    - Form Events
    - Keyboard Events
    - Mouse Events
    - Drag Events
- Input tag is very important tag for insertion of records through forms. Input tag is used to create different controls through which any user can insert data in system. Type, Auto focus, placeholder, required etc. attributes
- The HTML <canvas> element is employed for rendering graphics on a webpage. The image on the left has been generated using <canvas>, featuring four elements: a red square, a gradient square, a multicolor square, and a multicolored book.
- The HTML <canvas> element is utilized for dynamically drawing graphics using JavaScript. It acts as a container for graphics, and JavaScript is employed to perform the actual drawing operations.
- Multimedia comes in a wide range of arrangements. It very well may be nearly anything you can hear or see, similar to pictures, music, sound, videos, records, movies, animations, and that's just the beginning. Site pages frequently contain

multimedia components of various kinds and arrangements. Controls, Height, Weight, Source, Autoplay, Muted these are some important attributes used with <audio >& <video> tags.

7.8 Self-Assessment Questions

1) Explain HTML5. What are the unique features of HTML 5?
2) Differentiate between HTML 4 & HTML 5.
3) Explain in details tags available in HTML 5.
4) Explain Input tag & its attributes with suitable example.
5) Explain adding Graphics in HTML 5.
6) State the importance of Multimedia in digital world.
7) Explain <Audio> tag with suitable example.
8) Explain <Video> tag with suitable example.
9) What do you mean Events? Explain events in HTML 5.
10) Explain different file formats available for adding audio or video in web page.
11) Explain Different mouse events in HTML 5 with suitable example.
12) Explain Different Window events in HTML 5with suitable example.
13) Explain Different keyboard events in HTML 5with suitable example.

7.9 References/Reference Reading

1) HTML, CSS -The complete Reference –
2) HTML, CSS -DESIGN and build websites by Jon duckett
3) HTML, Javascript & CSS -Ivan Bayross

Unit - 8

Cascading Style Sheet (CSS)

Learning Outcomes

- Students will be able to understand meaning of Cascading Style Sheet (CSS)
- Students will have a clarity over the use of CSS
- Students will be able to understand types of CSS
- Students will be able to understand the different types of selectors in CSS
- Students will be able to understand the Properties, Values

Structure

8.1 Introduction to CSS

8.2 Use of CSS

8.3 Types of CSS

- **Knowledge Check 1**
- **Outcome Based Activity**

8.4 CSS Syntax

8.5 Selectors,Properties, Values

8.6 CSS Properties ---

Background, Text, Fonts,

- Link, List, Table,

Border, Margin,

- **Knowledge Check 2**
- **Outcome Based Activity**

8.7 Summary

8.8 Self-Assessment Questions

8.9 References/Reference Reading

8.1 Introduction to Cascading Style Sheet (CSS)-

In essence, CSS is utilized to define styles for web pages, encompassing layout, design, and responsive adjustments for diverse devices and screen sizes. CSS, short for Cascading Style Sheets, is a specialized language designed to enhance the visual presentation of web pages. It allows developers to apply styles independently of the HTML structure, providing substantial control over the appearance of web documents. Although CSS is relatively simple to learn, it offers extensive capabilities for managing the look and feel of HTML elements across various media, such as screens and printed pages.

By using CSS, developers can efficiently style multiple web pages simultaneously through external stylesheets, which are stored in separate CSS files. This approach streamlines the process of maintaining consistent design and layout across a website. CSS is fundamental for defining the visual aspects of web pages, including layout, design, and responsive adjustments for different devices and screen sizes.

8.2 Uses of Cascading Style Sheet (CSS)

1) Compatibility: -

When employing CSS, users can rest assured with its backward compatibility, ensuring smooth integration with older language versions. This means that even if CSS applications are developed in older programming language versions, they can seamlessly incorporate with new advancements without necessitating the removal of CSS predecessors. Consequently, developers can easily update existing code without the need to eliminate changes associated with the previous version of CSS.

2) E-Commerce Domain: -

CSS has significantly impacted the e-commerce industry by enhancing the styling and appearance of application frameworks, benefiting businesses of all sizes. CSS is integral to the design of e-commerce websites, allowing for the customization and updating of various add-ins and library files. By using CSS within HTML frameworks, developers can build e-commerce platforms from scratch, ensuring an appealing and functional user experience. This flexibility and control over the visual elements make CSS a crucial tool in the development and evolution of e-commerce web applications.

3) Website Maintenance: -

Another primary reason why the use of CSS is important is its significant role in website maintenance. CSS simplifies the process of maintaining a website by enabling easier updates

and changes. With a CSS file, the website's appearance becomes more adaptable, allowing for convenient modifications. Furthermore, it streamlines HTML formatting and the alteration of related data elements. This ease of updating and managing the visual and structural aspects of a website makes maintenance more efficient from a development standpoint.

4) Easy Accessibility: -

From an accessibility perspective, utilizing CSS offers superior arrangements that allow users to modify the user interface to meet business requirements. It also enables web pages to be easily rendered on various devices, such as screen readers and PDAs. This has a significant impact when considering changes to the look and feel of a web page from both the end-user and business perspectives.

5**) End-User and Server-side Representation**

CSS files primarily handle the styling of web pages and can indirectly influence server-side responses by improving user interface (UI) design. While CSS itself does not interact directly with server-side processes, effective use of CSS can enhance the overall user experience, making the web representation more appealing and functional from the end user's perspective. This, in turn, can contribute to a more efficient and user-friendly interface on the server side.

8.3 Types of CSS

CSS (Cascading Style Sheets) defines how HTML elements are displayed across different media, such as screens, paper, or other formats. It significantly reduces the time required to style a website by allowing the design of multiple web pages to be controlled simultaneously. CSS sets attributes such as font size, font family, color, and background color for the web page.

CSS also enables the addition of dynamic effects and animations to a website. It can be used to create interactive elements like buttons, effects, loaders, spinners, and animated backgrounds, enhancing the overall user experience.

Without CSS, a website would lack visual appeal. There are three types of CSS:

- Inline CSS
- Internal/Embedded CSS
- External CSS

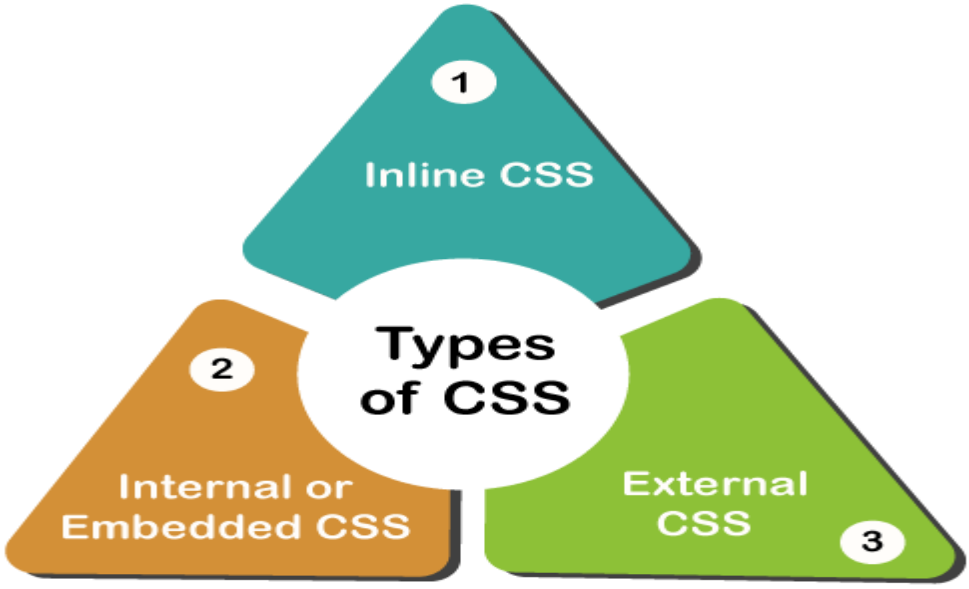


Fig.1 type of CSS

1) Inline CSS

Without the use of selectors, inline CSS styles particular HTML elements by appending a style property straight to each HTML tag. While using only inline CSS can make managing a website cumbersome, it can be useful in certain situations where access to external CSS files is unavailable or when styles need to be applied quickly to individual elements. In the example below, inline CSS is applied to <p> and <h1> tags.

Advantages of Inline CSS:

- CSS rules can be created directly within the HTML page.
- There is no need to create or transfer a separate CSS file.

Disadvantages of Inline CSS:

- Adding CSS rules to HTML elements can be time-consuming and can clutter the HTML structure.
- Styling multiple elements this way can increase page size and download time.

For Example:

```
“<!DOCTYPE html>
        <html>
        <body style="background-color:white;">
        <h1 style="color:blue;padding:20px;">Inline CSS Tutorials</h1>
        <p style="color:pink;">It will be used for if you want to apply some style to
particular line .</p>
        </body>
        </html>”
```

OUTPUT:

2) Internal CSS:

Internal CSS utilizes the <style> tag within the <head> section of an HTML document. While this approach is suitable for styling single pages, it becomes inefficient for multiple pages since the styles must be repeated on each page individually. To use internal CSS, follow these steps:

1. Open the HTML page and locate the <head> section.
2. Add the following code within the <head>:
   <style type="text/css">
3. Insert the CSS rules on new lines within the <style> tags.

Advantages of Internal CSS:

- Internal CSS eliminates the need to transfer multiple files since the styles are embedded within the HTML document.

Disadvantages of Internal CSS:

- Adding CSS directly in the HTML document can increase the page size and loading time of the webpage.

For Example: -

```
“<!DOCTYPE html>
        <html>
<head>
        <style>
        body {
           background-color: black;
        }
        h1 {
           color: red;
           padding: 50px;
        }
        </style>
        </head>
        <body>
        <h1>CSS types</h1>
        <h1>Internal Cascading Style sheet </h1>
        </body>
        </html>”
```

OUTPUT :-

3) External Style Sheet :-

In external CSS, web pages are linked to an external .css file, which is created using a text editor. This method is more efficient for styling a website because it allows for global changes by editing a single .css file.

Advantages of External CSS:

- It results in a cleaner and more organized HTML structure with smaller file sizes.
- The same .css file can be used across multiple web pages, ensuring consistency and reducing redundancy.

Disadvantages of External CSS:

- Web pages may not be displayed correctly until the external CSS file is fully loaded.
- Using multiple external CSS files can increase the download time of a website.

To use the external CSS, follow the steps, given below:

1. Use a text editor to create a new.css file and include the Cascading Style Sheet rules.

```
body {
  background-color: yellow;
}

h1 {
  color: navy;
  margin-left: 20px;
}
```

mystyle.css

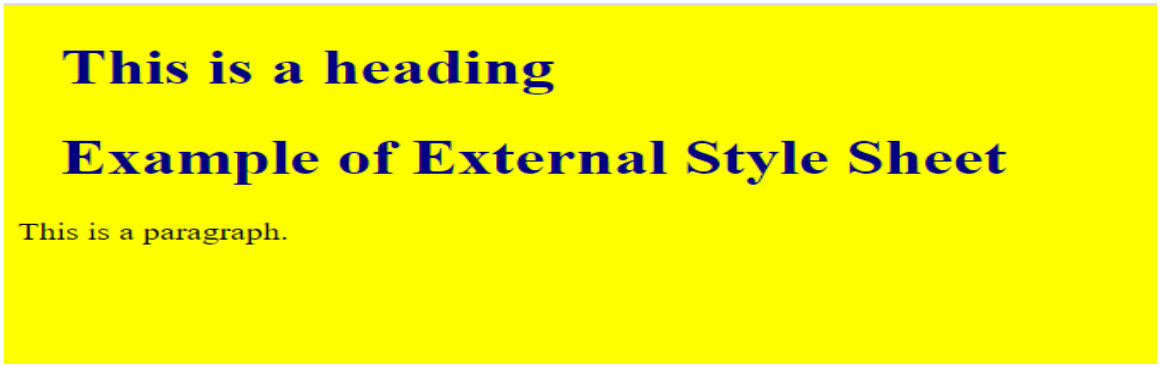
2. Add a reference to the external CSS file right after <title> tag in the <head> section of HTML sheet:

   <link rel="stylesheet" type="text/css" href="style.css" />

```
<!DOCTYPE html>
<html>
<head>
<link rel="stylesheet" href="mystyle.css">
</head>
<body>
<h1>This is a heading</h1>
<h1>Example of External Style Sheet </h1>
<p>This is a paragraph. </p>
</body>
</html>
```

Demo-h5.html

OUTPUT:



❖ **Knowledge Check 1**

Fill in the Blanks

1) **CSS stands for** ----------
2) ---------- CSS is used to style a specific HTML element.
3) ---------- is used to define styles for your web pages.

❖ **Outcome Based Activity**

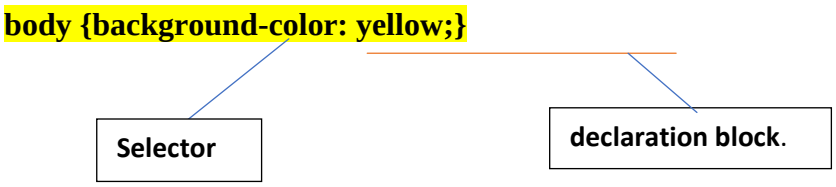
Demonstrate the use of 1) Inline CSS

2) Internal CSS

3) External CSS (name it as “mystyle.css”)

8.4 CSS Syntax

The CSS command is divided into two parts. A **declaration block** and a **selector** make up a CSS rule.

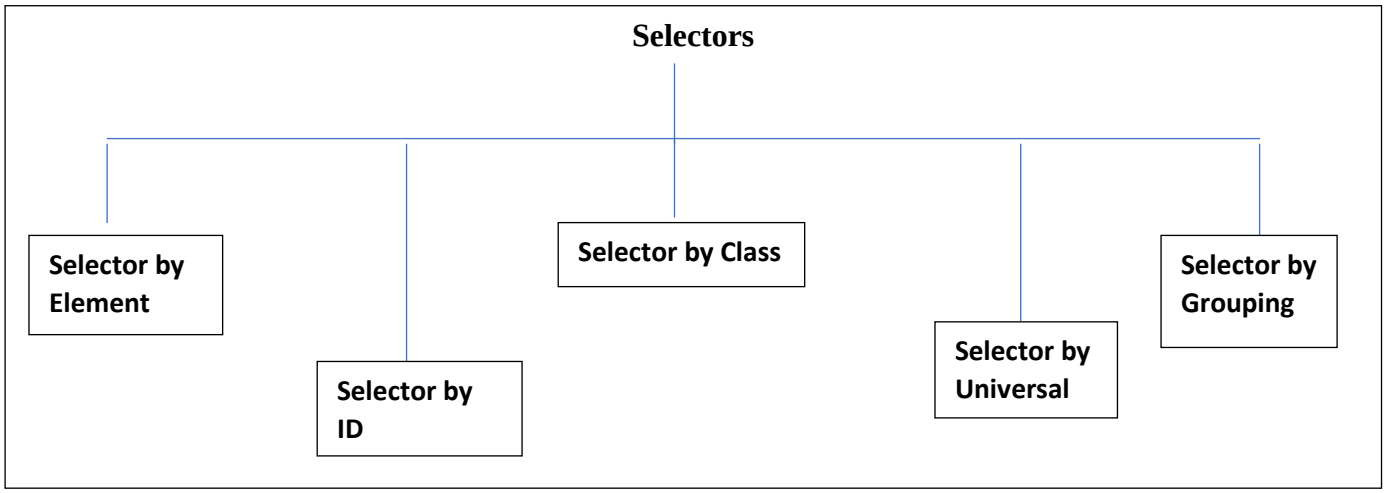


Selector: A selector targets the HTML element you want to style.

Declaration Block: The declaration block contains one or more declarations, each separated by semicolons. Each declaration consists of a CSS property name and a value, separated by a colon. Multiple declarations are enclosed within curly braces.

8.5 Selectors:

The selector focuses to the HTML element you need to style. CSS selectors are used to "find" (or select) the HTML elements you want to style.

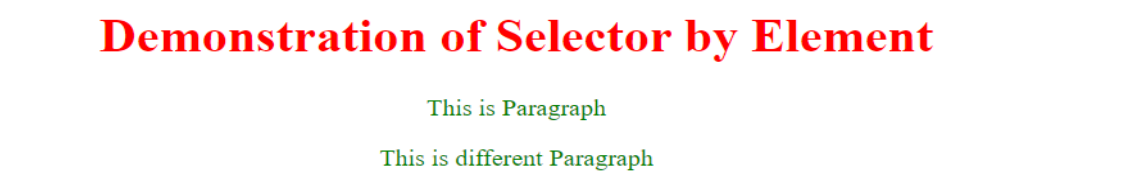


1) Selector by Element:

The element selector selects HTML elements based on the element name.

```
“<!DOCTYPE html>
<html><head>
<style>
h1 {
  text-align: center;
  color: red;}
p {
  text-align: center;
  color: green;
}
</style>
</head><body>
<h1>Demonstration of Selector by Element </h1>
<p id="para1">This is Paragraph</p>
<p>This is different Paragraph </p>
</body></html>”
```

1) OUTPUT : Selector by Element:



2) Selector by ID

The HTML element's id attribute is used by the id selector to target a particular element. The id selector is used to choose a single unique element because each element's id is unique inside a page. Use the hash (#) character before the element's id to choose an element with a specified id.

For Example:

```
“<!DOCTYPE html>
<html>
<head>
<style>
#h1 {
  text-align: center;
  color: blue;}
#p1 {
  text-align: center;
  color: purple;
}
</style>
</head>
<body>
<h1 id="h1">Demonstration of Selector by ID </h1>
<p id="p1">This is Paragraph</p>
</body>
</html>”
```

OUTPUT:

3) **Selector by Class:**

The class selector targets HTML elements that possess a specific class attribute. To select elements with a particular class, precede the class name with a period (.) character.

```
“<!DOCTYPE html>
<html>
<head>
<style>
.s {
  text-align: center;
  color: green;
}
</style>
</head>
<body>
<h1 class="s">Demonstration of Selector by Class </h1>
</body>
</html>”
```

OUTPUT:

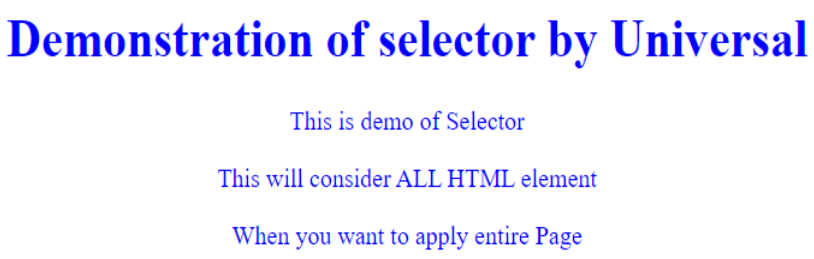
4) Selector by Universal:

The universal selector (*) selects all HTML elements on the page.

For Example:

```
“<!DOCTYPE html>
<html>
<head>
<style>
* {
  text-align: center;
  color: blue;
}
</style>
</head>
<body>
<h1>Demonstration of selector by Universal </h1>
<p>This is demo of Selector </p>
<p id="para1">This will consider ALL HTML element </p>
<p>When you want to apply entire page </p>
</body>
</html>”
```

OUTPUT:



5) Selector by Grouping

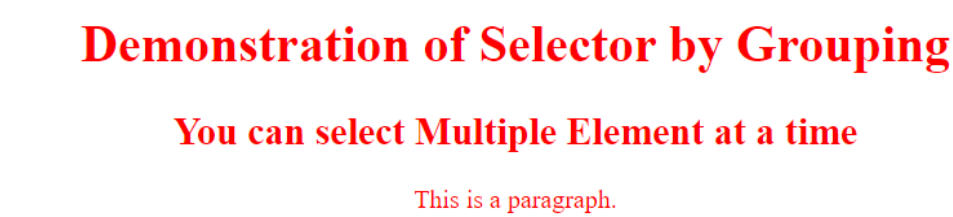
The grouping selector selects all the HTML elements with the same style definitions.

- It will be better to bunch the selectors, to minimize the code.
- To bunch selectors, separate each selector with a comma.

For Example :

```
“<!DOCTYPE html>
<html>
<head>
<style>
h1, h2, p {
  text-align: center;
  color: red;
}
</style>
</head>
<body>
<h1>Demonstration of Selector by Grouping </h1>
<h2>You can select Multiple Element at a time </h2>
<p>This is a paragraph.</p>
</body>
</html>”
```

OUTPUT:



CSS Properties ---

[Background, Text, Fonts, Link, List, Table, Border, Margin]

1) Background Properties:

The background shorthand CSS property sets all background style properties at once, such as color, image, origin and size, or repeat method.

Property	Description	Values
1. background-color	This is used to specify the background color of the element.	#b0d4de;
2. background-image	The background-image property is used to set an image as a background of an element. By default the image covers the entire element. You can set the background image for a page like this.	url("paper1.gif");
3. background-repeat	By default, the background-image property repeats the background image horizontally and vertically. Some images are repeated only horizontally or vertically.	repeat-x;
4. background-attachment	The background-attachment property is used to specify if the background image is fixed or scroll with the rest of the page in browser window. If you set fixed the background image then the image will not move during scrolling in the browser.	fixed;
5. background-position	The background-position property is used to define the initial position of the background image. By default, the background image is placed on the top-left of the webpage.	Center top bottom Left right

2) Text Properties:

Property	Description	Values
"color"	Sets the color of a text	RGB, hex, keyword
"line-height"	Sets the distance between lines	normal, *number*, *length*, *%*
"letter-spacing"	Increase or decrease the space between characters	normal, *length*
"text-align"	Aligns the text in an element	left, right, center, justify

“text-decoration”	Adds decoration to text	none, underline, overline, line-through
“text-indent”	Indents the first line of text in an element	*length, %*
“text-transform”	Controls the letters in an element	none, capitalize, uppercase, lowercase

3) Font Properties:

Property	**Description**	**Values**
“font”	Sets all the font properties in one declaration	*font-style, font-variant, font-weight, font-size/line-height, font-family,* caption, icon, menu, message-box, small-caption, status-bar, inherit
“font-family”	Specifies the font family for text	*family-name, generic-family*, inherit
“font-size”	Specifies the font size of text	xx-small, x-small, small, medium, large, x-large, xx-large, smaller, larger, *length, %*, inherit
“font-style”	Specifies the font style for text	normal, italic, oblique, inherit
“font-variant”	Specifies whether or not a text should be displayed in a small-caps font	normal, small-caps, inherit
“font-weight”	Specifies the weight of a font	Normal, bold, bolder, lighter, 100, 200, 300, 400, 500, 600, 700, 800.

4) Link Properties: Links can be styled with any CSS property.

5) List Properties:

Property	**Description**	**Values**
“list-style”	Sets all the properties for a list in one declaration	*list-style-type, list-style-position, list-style-image*, inherit
“list-style-image”	Specifies an image as the list-item marker	URL, none, inherit
“list-style-position”	Specifies where to place the list-item marker	inside, outside, inherit

“list-style-type”	Specifies the type of list-item marker	none, disc, circle, square, decimal, decimal-leading-zero, armenian, georgian, lower-alpha, upper-alpha, lower-greek, lower-latin, upper-latin, lower-roman, upper-roman, inherit ”

6) Table Properties:

Property	Description	Values
border	To specify table borders in CSS, use the border property.	1px solid black;
Width	The table above might seem small in some cases. If you need a table that should span the entire screen (full-width), add width	100%;
border-collapse	border-collapse property sets whether the table borders should be collapsed into a single border	collapse;
Width and Height	You can define Height & width of table	Width :100%; height: 70px;
text-align	property sets the horizontal alignment (like left, right, or center) of the content in <th> or <td>.	Left ,Right, Center
padding	To control the space between the border and the content in a table, use the padding property on <td> and <th> elements	15px;

7) Border properties :

Property	Description	Values
“border”	Sets all the border properties in one declaration	*border-width, border-style, border-color*
“border-bottom”	Sets all the bottom border properties in one declaration	*border-bottom-width, border-bottom-style, border-bottom-color*
“border-bottom-color”	Sets the color of the bottom border	*border-color*
“border-bottom-style”	Sets the style of the bottom border	*border-style*

“border-bottom-width”	Sets the width of the bottom border	*border-width*
“border-color”	Sets the color of the four borders	*color_name, hex_number, rgb_number,* transparent, inherit
“border-left”	Sets all the left border properties in one declaration	*border-left-width, border-left-style, border-left-color*
“border-left-color”	Sets the color of the left border	*border-color*
“border-left-style”	Sets the style of the left border	*border-style*
“border-left-width”	Sets the width of the left border	*border-width*
“border-right”	Sets all the right border properties in one declaration	*border-right-width, border-right-style, border-right-color*
“border-right-color”	Sets the color of the right border	*border-color*
“border-right-style”	Sets the style of the right border	*border-style*
“border-right-width”	Sets the width of the right border	*border-width*
“border-style”	Sets the style of the four borders	none, hidden, dotted, dashed, solid, double, groove, ridge, inset, outset, inherit
“border-top”	Sets all the top border properties in one declaration	*border-top-width, border-top-style, border-top-color*
“border-top-color”	Sets the color of the top border	*border-color*
“border-top-style”	Sets the style of the top border	*border-style*
“border-top-width”	Sets the width of the top border	*border-width*
“border-width”	Sets the width of the four borders	thin, medium,thick, *length,* inherit”

8)Margin Properties:

Property	**Description**	**Values**
• margin-top • margin-right • margin-bottom	The top, right, bottom, and left margin can be changed independently using	**auto** - the browser calculates the margin **length** - specifies a margin in px, pt, cm, etc. **%** - specifies a margin in % of

• margin-left	separate properties. A short hand margin property can also be used, to change all margin sat once.	the width of the containing element **inherit** - specifies that the margin should be inherited from the parent element

Knowledge Check 2

Fill in the blanks

1) The ----------selector selects HTML elements with a specific class attribute.

2) The universal selector (*) selects ---------- HTML elements on the page.

3) The selector focuses to the HTML element you need to style.

Outcome Based Activity

Demonstrate the use of Border properties

8.6 Summary

- CSS, short for Cascading Style Sheets, is a carefully crafted language designed to streamline the process of making web pages presentable. It allows for the application of styles to web pages, independently of the HTML that constitutes each page.
- There are three primary types of CSS:
  Inline CSS
  Internal/Embedded CSS
  External CSS
- A CSS command consists of two sections: a Selector and a declaration block.
- The selector identifies the HTML element you wish to style. CSS selectors are instrumental in pinpointing the HTML elements you want to style.
- Selectors come in five different types:
  Selector by Element
  Selector by ID
  Selector by Class
  Selector by Universal

Selector by Grouping

- CSS offers a variety of properties (such as Background, Text, Fonts, Link, List, Table, Border, Margin) to enable the attractive design of web pages.

8.7 Self-Assessment Questions

1) What is Cascading Style Sheet? Explain different uses of CSS.
2) Explain different types of CSS with suitable example.
3) Explain Inline CSS with suitable example.
4) Explain Internal CSS with suitable example.
5) Explain External CSS with suitable example.
6) Explain syntax of CSS command with example.
7) Explain the term Selector. Explain different types of Selectors with example.
8) Explain Selector by Class with proper example.
9) Explain Selector by Universal with proper example.
10) Explain Selector by ID with proper example.
11) Explain Selector by Element with proper example.
12) Explain various properties of CSS.
13) Explain the Background properties of CSS.
14) Explain the Table Properties of CSS with suitable example.
15) Differentiate between Inline CSS & Internal CSS.

8.9 References/Reference Reading

1) HTML, CSS -The complete Reference –
2) HTML, CSS -DESIGN and build websites by Jon duckett
3) HTML, JavaScript & CSS -Ivan Bayross

Unit - 9

Java Script -I

Learning Outcomes

- Students will be able to understand basic concept of JavaScript
- Students will have a clarity over the key features of JavaScript
- Students will be able to understand the JavaScript Variables
- Students will be able to understand the Data Types in JavaScript
- Students will be able to understand Operators in JavaScript

Structure

9.1 Introduction to JavaScript

JavaScript is a lightweight, cross-platform, and interpreted scripting language widely known for enhancing website functionality. It is also utilized in various non-programming environments. JavaScript serves both client-side and server-side development purposes. It boasts a standard library of objects like Array, Date, and Math, along with essential language components such as operators, control structures, and expressions. Originally designed to "make web pages alive," JavaScript programs, known as scripts, can be embedded directly within HTML web pages and execute automatically as the page loads.

Client-side: JavaScript offers objects to control programs and their Document Object Model (DOM) on the client-side. Client-side extensions enable applications to manipulate elements on HTML forms and respond to user events like mouse clicks, form data input, and page navigation. Popular client-side libraries include AngularJS and ReactJS.

Server-side: JavaScript provides objects for running JavaScript on a server. Server-side extensions facilitate communication with databases, data flow management within applications, and server-side file operations. The widely used framework for server-side JavaScript is node.js.

9.2 Key features of JavaScript:

Key features of JavaScript:

1. Validating User Input: JavaScript is adept at validating user input in forms, checking for errors and ensuring data accuracy before submission to the server, thus saving time.
2. Basic Client-side Calculations: Being a client-side technology, JavaScript can perform basic computations in the browser, reducing the need for server interactions. This is particularly beneficial for frequent calculations, as server interactions would be slower.
3. More Control: JavaScript empowers browsers with greater control, reducing reliance on web servers. It provides additional functionalities to browsers, helping to alleviate server load and network traffic.
4. Platform Independence/Architecture Neutrality: JavaScript's interpretive nature eliminates the need for compilation and ensures compatibility across different platforms, including Windows, Macintosh, and other Netscape-supported systems. It can be seamlessly embedded into HTML and other content.

5. Control Structures: JavaScript offers various control structures such as If, If-else, switch case, for, do-while, etc., to enhance web page interactivity.

- **Knowledge Check 1**

State whether statement is True or False:

1) JavaScript can easily run on any platform.
2) JavaScript is not use to validate user's input.
3) The programs in this language are called scripts

9.3 JavaScript Variables: -

In JavaScript, a variable serves as a label for storage locations. There are two types of variables: Local variables and Global variables.

Certain guidelines govern the declaration of JavaScript variables, also known as identifiers:

1. Names must commence with a letter (from a to z), underscore (_), or dollar sign ($).
2. Following the first letter, digits (from 0 to 9) can be used, such as value1.
3. JavaScript variables are case-sensitive; hence, variables like x and X are distinct.

For example :-

```
var a = 10;
var _A="sonu";
```

Program to print addition of two Numbers.

```
<script>
var a= 100;
var b = 200;
var c=a+b;
document.write("addition of two nos is :-" + c);
</script>
```

OUTPUT :-

· The variable s1 holds the string "This is using Double quote" enclosed within double quotes.

· The variable s2 holds the string 'This is using Single quote' enclosed within single quotes.

· These strings are then displayed in separate paragraphs on the HTML page.

JavaScript Variable Scope :-

The scope of a variable refers to the portion of the program where it is defined. In JavaScript, variables have two scopes:

- Global Variables: A global variable has global scope, meaning it can be defined anywhere in your JavaScript code and is accessible throughout the program.
- Local Variables: A local variable is visible only within the 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. Consider the following example:

```
<html>
        <body>
        <script>
        var data=200;//global variable
        function a(){
        var data=400;   // Local Variable
         document.writeln("This is Local Variable  :- " + data + “<br>”);

        }
        function b(){
        document.writeln("This is Global Variable  :- " +data);
        }
        a();//calling JavaScript function
        b();
         </script>  </body></html>
```

OUTPUT :-

9.4 Data Types in JavaScript: -

One of the fundamental characteristics of a programming language is its support for data types, which determine the types of values that can be represented and manipulated. In JavaScript, you can work with three primitive data types:

- Numbers, such as 785, 160.50, and so on.
- Text strings, represented within double or single quotes, for example, "string".
- Boolean, representing true or false values.

Additionally, JavaScript defines two special data types: null and undefined, each representing only a single value. Alongside these primitive data types, JavaScript also supports a composite data type known as an object.

It's worth noting that JavaScript does not distinguish between number values and floating-point values. All numbers in JavaScript are represented as floating-point values.

Data types	**Details**
String	represents sequence of characters e.g. "Good Morning"
Number	represents numeric values e.g.520
Boolean	represents Boolean value either false or true
Undefined	represents undefined value
Null	represents null i.e. no value at all

Example of String data type :-

```
<!DOCTYPE html>
<html>
<body>
<p>Demonstration of String data type :</p>
<p id="demo"></p>
<script>
var s1= "This is using Double  quote";
var s2 = 'This is using Single   quote';
```

OUTPUT

- The variable s1 contains a string "This is using Double quote" enclosed in double quotes.
- The variable s2 contains a string 'This is using Single quote' enclosed in single quotes.
- The content of both variables is displayed in separate paragraphs with the help of the document. Get Element By Id method in JavaScript.

Example of Number data type :

```
<!DOCTYPE html>
<html>
<body>
<p> Demonstration of Number Data type </p>
<p id="demo"></p>
<script>
```

9.5 Operators in JavaScript :-

JavaScript consist of following Operators :-

1) "Arithmetic Operators"
2) "Comparison (Relational) Operators"
3) "Bitwise Operators"
4) "Logical Operators"
5) "Assignment Operators"
6) "Special Operators"

1) Arithmetic Operators: - (consider the values for variable a=10 b=20)

Sr. No	Operator	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

For Example :-

```
“<html> <body>
  <script type = "text/javascript">
      <!--
        var p= 33;
        var q = 10;
        var r= "Test";
        var linebreak = "<br />";
        document.write("p + q = ");
        result = p + q;
        document.write(result);
        document.write(linebreak);
        document.write("p - q = ");
        result = p - q;
        document.write(result);
        document.write(linebreak);
        document.write("p / q = ");
        result = p / q;
        document.write(result);
        document.write(linebreak);
        document.write("p % q = ");
        result = p % q;
        document.write(result);
        document.write(linebreak);”
```

```
“document.write("p + q + r  = ");
        result = p+ q + r;
        document.write(result);
        document.write(linebreak);

        p = ++p;
        document.write("++p = ");
        result = ++p;
        document.write(result);
        document.write(linebreak);

        q = --q;
        document.write("--q = ");
        result = --q;
        document.write(result);
        document.write(linebreak);0
      //-->
    </script>

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

OUTPUT :-

· The variables p and q contain the values 33 and 10, respectively.

· (p + q) adds the values of p (33) and q (10), resulting in 43.

· he result is displayed as "p + q = 43".

2) Comparison (Relational) Operators: -

JavaScript supports the following comparison operators −

Assume variable X holds 10 and variable Y holds 20, then −

Sr. No	Operator	Description
1	= = (Equal)	Checks if the value of two operands are equal or not, if yes, then the condition becomes true. **Ex:** (X == Y) 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:** (X != Y) 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:** (X > Y) 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:** (X < Y) 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:** (X >= Y) 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:** (X <= Y) is true.

For Example :-

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

        var x = 10;
        var y = 20;
        var linebreak = "<br />";

        document. Write ("(x == y) -------");
        result = (x == y);
        document. Write(result);
        document. Write(linebreak);

        document. Write("(x < y)------- ");
        result = (x < y);
        document. Write(result);
        document. Write(linebreak);

        document. Write("(x > y)------ ");
        result = (x > y);
        document.write(result);
        document.write(linebreak);
```

```
document.write("(x != y) ---- ");
        result = (x != y);
        document.write(result);
        document.write(linebreak);

        document.write("(x >= y) ---- ");
        result = (x >= y);
        document.write(result);
        document.write(linebreak);

        document.write("(x <= y) ---- ");
        result = (x <= y);
        document.write(result);
        document.write(linebreak);
    </script>
    Demonstration of Comparison
Operators
  </body> </html>
```

OUTPUT :- (x == y) compares the value of x (10) with the value of y (20) using the equality operator (==). Since x is not equal to y, the result is false.

3) Logical Operators: -

JavaScript supports the following logical operators −

Assume variable X holds 10 and variable Y holds 20, then –

Sr.No	Operator	Description
1	&& (Logical AND)	If both the operands are non-zero, then the condition becomes true. **Ex:** (X && Y) is true.
2	|| (Logical OR)	If any of the two operands are non-zero, then the condition becomes true. **Ex:** (X || Y) 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:** ! (X && Y) is false.

For Example :-

```
<html>
  <body>
    <script type = "text/javascript">
        var X = true;
        var Y = false;
        var linebreak = "<br />";
        document. write ("(X && Y) -- ");
        result = (X && Y);
        document. write(result);
        document. write(linebreak);
        document. write("(X || Y) -- ");
        result = (X || Y);
        document. write(result);
        document. write(linebreak);
       document. write ("!(X && Y) -- ");
        result = (!(X && Y));
        document. write(result);
        document. write(linebreak);
    </script>
    <p>Demonstration of Logical Operators. </p>
  </body> </html>
```

OUTPUT :-

· (X && Y) performs a logical AND operation between X (true) and Y (false), resulting in false.

· (X || Y) performs a logical OR operation between X (true) and Y (false), resulting in true.

·!(X && Y) performs a logical NOT operation on the result of (X && Y), which is false, resulting in true.

4) Bitwise Operator :-

JavaScript supports the following bitwise operators −

Assume variable A holds 2 and variable B holds 3, then

Sr. No	Operator	Description
1	& (Bitwise AND)	It performs a Boolean AND operation on each bit of its integer arguments. **Ex:** (A & B) is 2.
2	| (Bit Wise 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. **Ex:** (~B) is -4.
5	<< (Left Shift)	It moves all the bits in its first operand to the left by the number of places specified in the second operand. New bits are filled with zeros. Shifting a value left by one position is equivalent to multiplying it by 2, shifting two positions is equivalent to multiplying by 4, and so on. **Ex:** (A << 1) is 4.
6	>> (Right Shift)	Binary Right Shift Operator. The left operand's

		value is moved right by the number of bits specified by the right operand. **Ex:** (A >> 1) is 1.
7	>>> (Right shift with Zero)	This operator is just like the >> operator, except that the bits shifted in on the left are always zero. **Ex:** (A >>> 1) is 1.

For Example :-

```
<html>
 <body>
   <script type = "text/javascript">
        var a = 2;
        var b = 3;
        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>Demonstration of Bitwise Operators
</p>
  </body>
</html>
```

OUTPUT :-

· (a & b) performs a bitwise AND operation between a (2) and b (3), resulting in 2.

· (a | b) performs a bitwise OR operation between a (2) and b (3), resulting in 3.

· (a ^ b) performs a bitwise XOR (exclusive OR) operation between a (2) and b (3), resulting in 1.

5) Assignment Operator :-

JavaScript supports the following assignment operators −

Sr.No	Operator	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 & 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 & 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 & 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 & Assignment)	It takes modulus using two operands and assigns the result to the left operand. **Ex:** C %= A is equivalent to C = C % A

For Example: -

```
<html>
  <body>
    <script type = "text/javascript">
        var a = 50;
        var b = 10;
        var linebreak = "<br />";
document.write("Output of  :(a = b) --");
        result = (a = b);
        document.write(result);
        document.write(linebreak);
document.write("Output of  : (a += b) -- ");
        result = (a += b);
        document.write(result);
        document.write(linebreak);


        document.write("Output of  : (a -=
b) -- ");
        result = (a -= b);
        document.write(result);
        document.write(linebreak);
```

```
  document.write("Output of  : (a *= b) -- ");
        result = (a *= b);
        document.write(result);
        document.write(linebreak);
       document.write("Output of :  (a /= b) -- ");
        result = (a /= b);
        document.write(result);
        document.write(linebreak);
       document.write("Output of : (a %= b) -- ");
        result = (a %= b);
        document.write(result);
        document.write(linebreak);
    </script>
    <p>Demonstration of Assignment Operator
</p>
  </body> </html>
```

OUTPUT :-

- **Knowledge Check 2**

Fill in the Blanks:

1) At the time of variable creation Name should begin with --------, ---------- & ---------

2) Boolean represents Boolean value either ------- or ----------.

3) In ----------type of operators **,** If both the operands are non-zero, then the condition becomes true.

· (a = b) assigns the value of b (10) to a, resulting in a becoming 10. The value assigned to a (10) is then returned as the result.

· (a += b) adds the value of b (10) to the current value of a (10)

becoming 20. The updated value of a (20) is then returned as the result.

9.6 Summary:

JavaScript is a lightweight, cross-platform, and interpreted scripting language renowned for its role in advancing website pages. It finds utility in numerous non-programming contexts as well. JavaScript facilitates both client-side and server-side development. It encompasses a standard library of objects like Array, Date, and Math, along with core language components such as operators and control structures.

Initially developed to "make web pages alive," JavaScript boasts key features including:

- User Input Validation
- Basic Client-Side Calculations
- Enhanced Control
- Platform Independence
- Control Structures

JavaScript variables serve as placeholders for storage locations, categorized into two types: local variables and global variables.

JavaScript supports various data types, including:

- Numbers (e.g., 785, 160.50)
- Text strings (e.g., "string")
- Boolean values (e.g., true or false)

Additionally, JavaScript incorporates several types of operators:

- Arithmetic Operators

- Comparison (Relational) Operators
- Bitwise Operators
- Logical Operators
- Assignment Operators
- Special Operators

9.7 Self-Assessment Questions

1) Explain Features of JavaScript.
2) Explain process to create variable in JavaScript.
3) Explain difference between Local Variable & Global Variable with suitable example.
4) Explain Data types of JavaScript with example.
5) Explain Different types of Operators in JavaScript.
6) Explain Local Variable & Global Variable with example.

9.8 References

1) HTML, CSS -The complete Reference –
2) HTML, CSS -DESIGN and build websites by Jon duckett
3) HTML, JavaScript & CSS -Ivan Bayross

Unit - 10

Java Script -II

Learning Outcomes

- Students will be able to understand importance of Control structures in JavaScript
- Students will have a clarity over the If statements, If---else statement
- Students will be able to understand the structure & execution method of Do---while
- Students will be able to understand the structure & execution method ofFor loops
- Students will be able to understandthe structure & execution method ofSwitch case

Structure

10.1 Control structure in JavaScript: -

Control structures are essential components in programming that determine the flow of control within programs. They enhance the clarity and understanding of algorithms or programs by organizing them into distinct modules known as logic or control structures. These structures analyze and dictate the direction of program flow based on specific conditions or criteria. There are three fundamental types of control structures:

1. Sequence Logic, or Sequential Flow: This involves the sequential execution of code statements, where each statement is executed one after another, akin to following a recipe.
2. Selection Logic, or Conditional Flow: This type of logic is used for making decisions or branching, selecting between two or more alternative paths. It includes constructs like if statements, if/else statements, and switch statements.
3. Iteration Logic, or Repetitive Flow: Iteration logic facilitates repetition, allowing a piece of code to be executed multiple times in succession. Common constructs for iteration include while loops, do/while loops, and for loops.

10.1.1 If statement: -

The "if" statement is a conditional branching construct in programming. It allows for the execution of a group of statements if a specified condition evaluates to true. If the condition is false, the subsequent statement is skipped. Here's the syntax of the "if" statement:

```
if (condition) {
  // Code block to be executed if the condition is true
}
```

For example :-

```
<html>
<body>
<script type="text/javascript">
        var t1 = prompt ("Enter Number");
        if (t1> 0) {
alert ("Given number is Positive!!!");
        }
</script>
</body>
</html>
```

OUTPUT :-

10.1.2 If – Else Statement :-

If – Else is a two-way choice statement. It is utilized to settle on choices and execute statements conditionally.

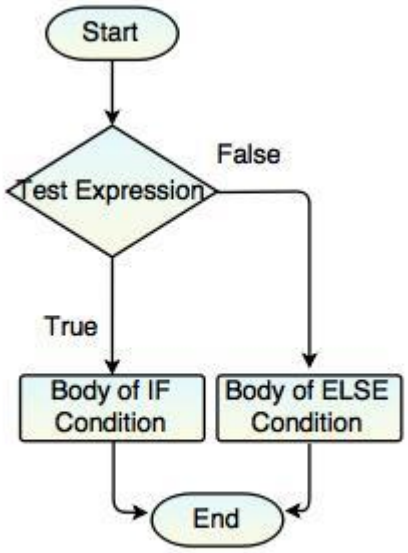


Fig. Flow Diagram of IF - ELSE Statement

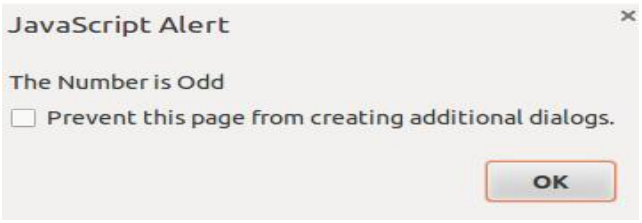
Syntax of If—Else statement: -

```
 if (condition) {
   // Code block to be executed if the condition is true
} else {
```

For Example: -

```
<html>
    <head>
    <script type="text/javascript">
        var no = prompt("Enter a Number to findOdd or Even");
        s = parseInt(no);
        if (isNaN(s))
        {
            alert("Please Enter a Number");
        }
        else if (s  == 0)
        {
            alert("The Number is Zero");
        }
        else if (s % 2)
        {
            alert("The Number is Odd");
        }
        else
        {
            alert("The Number is Even");
        }
    </script>
    </head>
</html>
```

OUTPUT:-



❖ **Knowledge Check 1**

State whether the statement is True or False

1) Iterative statements are not utilized for repetition.

2) If statement is also called as decision making statements.

3) Control Structures are only an approach to determine flow of control in programs.

❖ **Outcome Based Activity**

2 Demonstrate If statement -Execute the program for Largest No between two Numbers

3 Demonstrate If—else statement -Execute the program for Largest No between three Numbers

10.1.3 **D**

o---While: -

Do-While loop is an exit-controlled circle proclamation. Similar to the While Loop , the solitary distinction is condition will be checked toward the finish of the loop. The loop is executed in any event once, regardless of whether the condition is False.

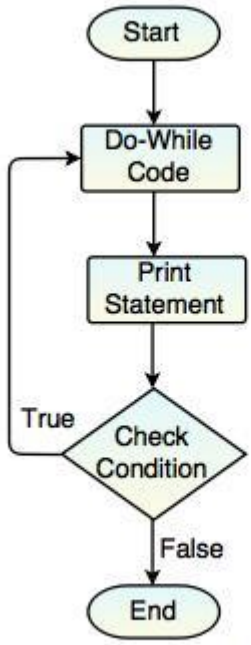


Fig. Flow Diagram of Do-While Loop

Syntax of the Do—while ;-

```
do{
   //Statements;
}
while(condition);
```

For Example:-

```
<html>
<body>
    <script type ="text/javascript">
    var i = 0;
    do
    {
        document.write(i+"<br>")
        i++;
    }
    while (i<= 5)
    </script>
</body></html>
```

OUTPUT :-

0

1

2

3

4

5

10.1.4 While loop :-

While loop is an entry-controlled loop statement. It is the most essential loop in JavaScript**.**

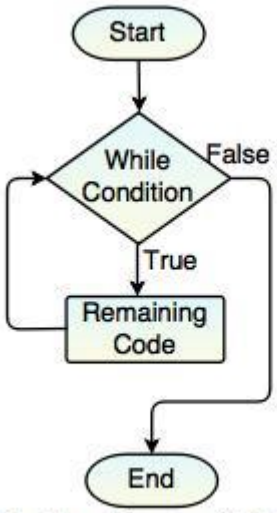


Fig. Flow Diagram of While Loop

Syntax of While loop :-

```
While (condition) {
   //Statements;
}
```

For Example :-Program to print Fibonacci series

```
"<html>
<body>
   <script type="text/javascript">
      var no1=0,no2=1,no3=0;
      document.write("Fibonacci Series:"+"<br>");
      while (no2<=10){
         no3 = no1+no2;
         no1 = no2;
         no2 = no3;
         document.write(no3+"<br>");
      }
   </script></body></html>"
```

OUTPUT:-

```
Fibonacci Series:
1
2
3
5
8
13
```

❖ **Knowledge Check 2**

Fill in the blanks

1) he ------- is executed in any event once, regardless of whether the condition is----- .
2) Do –while is ----------type of control structure.

❖ **Outcome Based Activity**

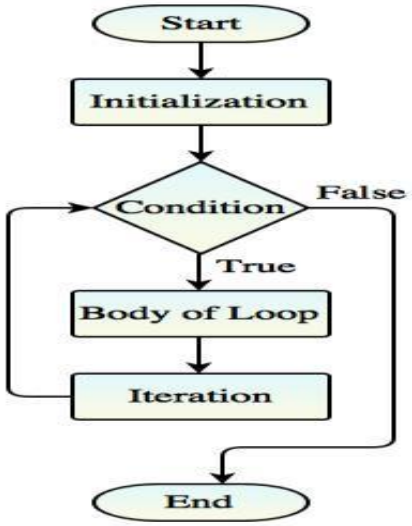
Demonstrate Do-while loop -Execute the program for first 500 Odd numbers using Do-while loop.

10.1.5 For loops: -

For loop is a minimized type of looping. It incorporates three significant parts:

1. “Loop Initialization”
2. “Test Condition”
3. “Iteration”

All these three sections arrive in a solitary line isolated by semicolons (;).



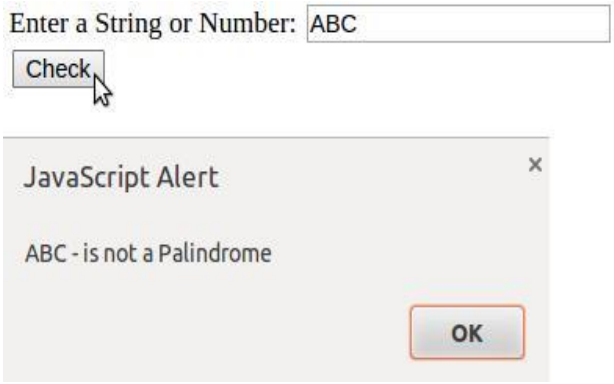
Syntax of For Loop :-

for(initialization; test-condition; increment/decrement)

{

//Statements;

}



10.1.6 Switch case :-

The switch statement is used to execute different actions based on different conditions. It allows for the comparison of a single expression with multiple different values.

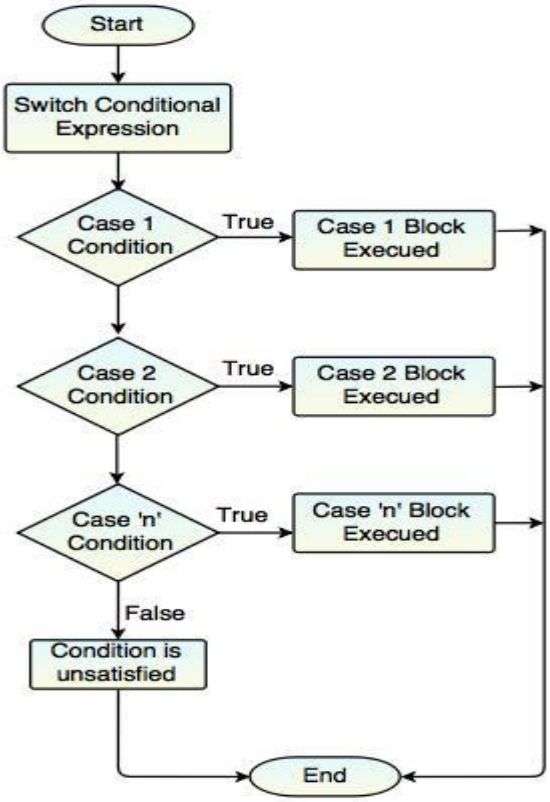


Fig. Flow Diagram of Switch Statement

Syntax of Switch Case :-

```
Switch (expression)
{
   case condition 1:
      //Statements;
      break;
   case condition 2:
      //Statements;
      break;
   case condition 3:
      //Statements;
      break;
case condition n:
      //Statements;
      break;
```

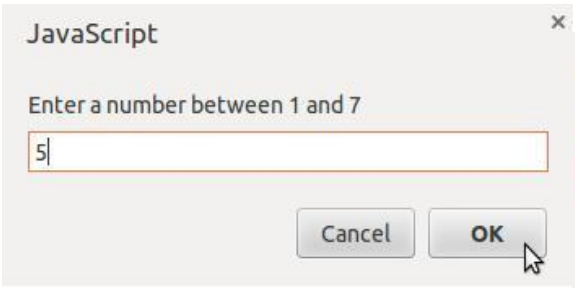
```
    default:
        //Statement;
}
```

For Example :-

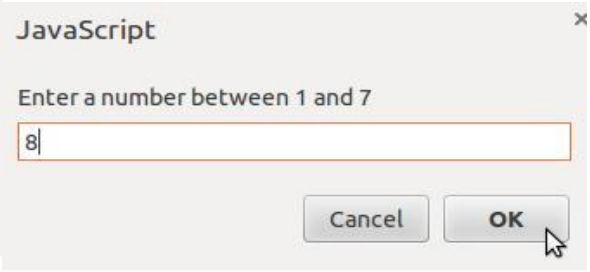
```
"<html>
<head>
    <script type="text/javascript">
    var day = prompt("Enter a number between 1 and 7");
    switch (day)
    {
        case (day="1"):
            document.write("Sunday");
            break;
        case (day="2"):
            document.write("Monday");
            break;
        case (day="3"):
            document.write("Tuesday");
            break;
        case (day="4"):
            document.write("Wednesday");
            break;
        case (day="5"):
            document.write("Thursday");
            break;
        case (day="6"):
            document.write("Friday");
            break;
        case (day="7"):
            document.write("Saturday");
            break;
        default:
            document.write("Invalid Weekday");
```

```
        break;
    }
    </script></head></html>”
```

OUTPUT:-



Thursday



Invalid Weekday

10.2 Summary :-

- Control structures are essential mechanisms for determining the flow of control in programs. They enhance clarity and understanding by organizing algorithms or programs into independent modules known as logic or control structures. These structures analyze specific conditions or boundaries to direct the program flow.
- The "if" statement serves as a conditional branching construct. When the condition evaluates to true, a group of statements is executed, and if false, the subsequent statement is skipped.
- The "if-else" statement provides a two-way choice, allowing decisions to be made and statements to be executed conditionally.
- The "do-while" loop is an exit-controlled loop where the condition is checked at the end of the loop. It executes the loop at least once, regardless of the initial condition.
- The "while" loop is an entry-controlled loop, representing the most basic loop structure in JavaScript.
- The "for" loop is a concise form of looping, comprising three essential parts: loop initialization, test condition, and iteration.
- The "switch" statement allows for performing various actions under different conditions, comparing expressions against multiple distinct values.

10.3 Self-Assessment Questions

1) What do you mean by control structures? Explain the need of control structures in programming.
2) Explain decision making statements with suitable examples.
3) Explain Iterative statements with syntax & examples.
4) Explain while loop with example.
5) Explain switch case statement with example.
6) Differentiate between while & do-while loop.
7) Explain for loop with syntax & example.

10.4 References/Reference Reading

1) HTML, CSS -The complete Reference –
2) HTML, CSS -DESIGN and build websites by Jon duckett.
3) HTML, JavaScript& CSS -Ivan Bayross

Unit -11

Java Script -III

Learning Outcomes

- Students will be able to understand meaning of DOM in JavaScript
- Students will have a clarity over the Key features of DOM
- Students will be able to understand some real examples of DOM
- Students will be able to understand the structure & use of HTML DOM document object
- Students will be able to understand the structure & use of six global DOM objects

Structure

11.1 What is DOM?

The W3C's Document Object Model (DOM) is an interface that is independent of any particular platform or programming language. It allows programs and scripts to dynamically access, update, and manipulate the content, structure, and style of web documents.

The Document object in the DOM represents the HTML document displayed in a browser window. This object includes various properties that reference different elements, enabling developers to access and modify the document's content. The method of accessing and modifying this content is known as the Document Object Model, or DOM.

Every webpage is displayed within a browser window, which can be seen as an object in itself. The Document object stands for the HTML content shown in that window. This object has multiple properties referencing other objects, making it possible to access and change the document's content.

The process of accessing and modifying the content of a document is known as the Document Object Model, or DOM. In the DOM, objects are arranged hierarchically, similar to a chain of command. This hierarchy is used to organize objects within a web document.

Here are some key components of this hierarchy (Fig.1):

- **Window Object**: This is at the top of the hierarchy and represents the browser window.
- **Document Object**: Each HTML document loaded into a window becomes a document object, encompassing the page's content.
- **Form Object**: This object includes everything enclosed within <form>...</form> tags.
- **Form Control Elements**: Within the form object, this includes all defined elements such as text fields, buttons, radio buttons, and checkboxes.

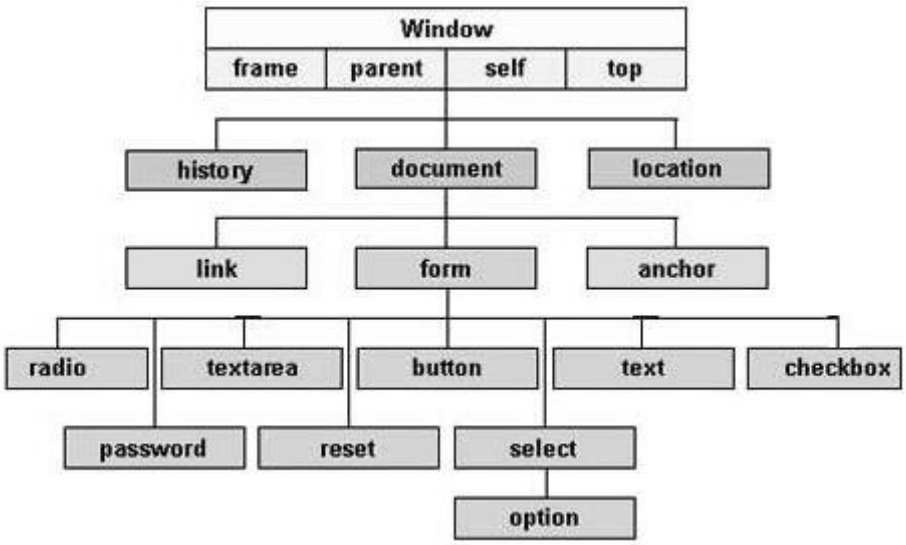


Fig.1 Document Object Model

11.2 Key Features of DOM :-

- The DOM establishes a standard for accessing documents.
- The DOM is a W3C (World Wide Web Consortium) standard.
- The W3C DOM standard is divided into three distinct parts:
- Core DOM: a standard model applicable to all document types.
- XML DOM: a standard model specifically designed for XML documents.
- HTML DOM: a standard model tailored for HTML documents.

11.3 Some Examples of DOM :-

The Legacy DOM − This is the model which was presented in early forms of JavaScript language. It is all around supported by all programs, yet permits access just to certain vital portions of documents, like forms, form components, and pictures.

- **The W3C DOM** − This document object model permits access and adjustment of all document content and is normalized by the World Wide Web Consortium (W3C). This model is supported by practically every one of the advanced programs.
- **The IE4 DOM** - was introduced with Version 4 of Microsoft's Internet Explorer. Subsequent versions, starting from IE 5, included support for most essential features of the W3C DOM.

❖ **Knowledge Check 1**

Fill in the Blanks

1)----------standard model for all document types

2) The document contains the ----------of the page.

3) Each site page dwells inside a program window which can be considered as an ----------.

❖ **Outcome Based Activity**

Create your own program & Identify the Parent & child element in program .

11.4 The HTML DOM Document Object

The document object represents your entire webpage. To interact with any element on an HTML page, you generally start by accessing the document object. Here are some examples of how you can use the document object to access and modify HTML elements.

Finding HTML Element: -

Method	Description
document. get Element By Id (*id*)	Find an element by element id
document. get Elements By Tag Name(*name*)	Find elements by tag name
document. Get Elements By Class Name (*name*)	Find elements by class name

The DOM provides various methods to locate HTML elements:

- Locating HTML elements by their id
- Locating HTML elements by their tag name
- Locating HTML elements by their class name
- Locating HTML elements using CSS selectors
- Locating HTML elements through HTML object collections

For Example :- 1) Elements by id

```
<!DOCTYPE html>
<html>
<body>
<h2>Finding HTML Elements by Id</h2>
<p id="intro">Hello World!</p>
<p>This example demonstrates the <b>getElementsById</b>
method.</p>
<p id="demo"></p>
<script>
var e = document.getElementById("intro");
document.getElementById("demo").innerHTML =
"The text from the intro paragraph is " + e.innerHTML;
</script>
</body>
</html>
```

OUTPUT:-

Finding HTML Elements by Id

Hello World!

This example demonstrates the **getElementsById** method.

The text from the intro paragraph is Hello World!

Changing HTML Elements :-

Property	**Description**
element.innerHTML = *new html content*	Change the inner HTML of an element

element.attribute = new value	Change the attribute value of an HTML element
element.style.*property = new style*	Change the style of an HTML element
Method	**Description**
element.setAttribute(attribute, value)	Change the attribute value of an HTML element

Adding or Deleting Element:

Method	**Description**
document. Create Element (*element*)	Create an HTML element
document. remove Child (*element*)	Remove an HTML element
document. app end Child (*element*)	Add an HTML element
document. Replace Child (*new, old*)	Replace an HTML element
document. write (*text*)	Write into the HTML output stream

Adding HTML Elements :

To insert a new element into the HTML DOM, you first need to create the element (element node) and then append it to an existing element.

```
document.createElement(element);

document.createTextNode(some text);

parentElement.appendChild(childElement);
```

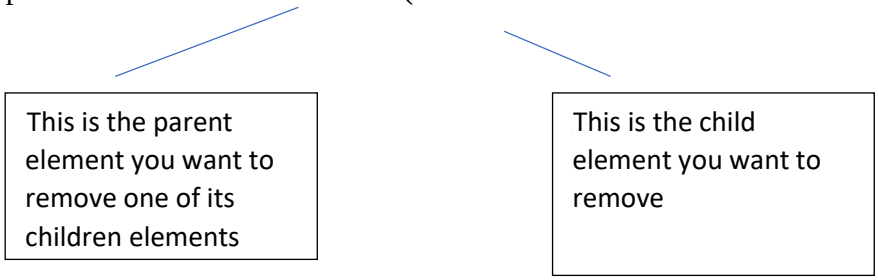
This is the element you want to append the child element to

This is the child element you want to nest inside the parent element

Removing HTML Element :-

To delete an HTML element, you first need to identify its parent element. Then, you can use the appropriate syntax to remove the targeted element.

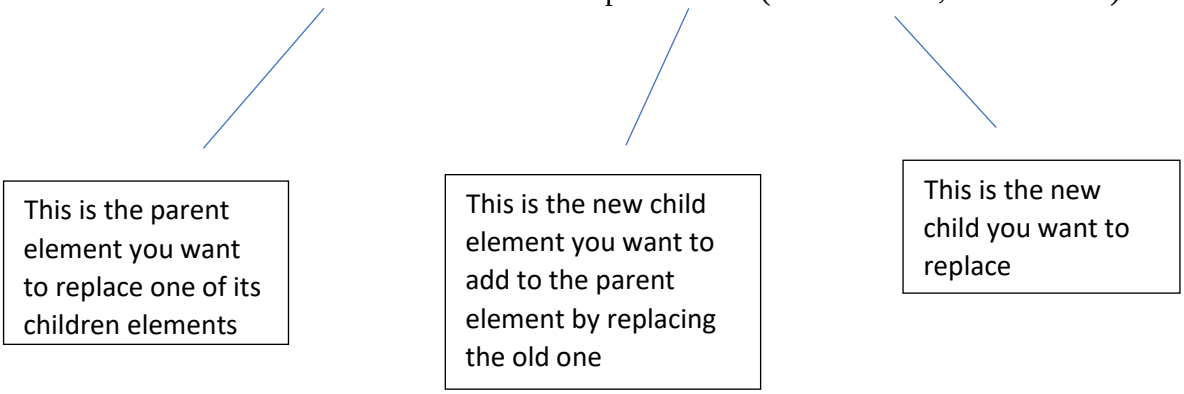
parent Element. Remove Child(child Elem



Replacing HTML Elements : -

To replace an element, use the replace Child () method:

Parent Element. Replace Child (new Element, old Element)



Adding Event Handler: -

Method	**Description**
document. Get Element By Id (id).on click = function(){code}	Adding event handler code to an onclick event

11.5 The six global DOM objects : -

Every JavaScript program can refer to the following global objects:

Name	Description
document	current HTML page and its content
history	list of pages the user has visited
location	URL of the current HTML page
navigator	info about the web browser you are using

screen	info about the screen area occupied by the browser
window	the browser windows

The window object represents the entire browser window and is the top-level object in the DOM hierarchy. All global code and variables are technically part of the window object.

Properties:

- document
- history
- location
- name

Methods:

- Popup boxes: alert, confirm, prompt
- Timers: set Interval, set Timeout, clear Interval, clear Timeout
- Window control: open, close, blur, focus, move By, move To, print, resize By, resize To, scroll By, scroll To

The Document Object

The document object represents the current web page and its elements.

Properties:

- anchors
- body
- cookie
- domain
- forms
- images
- links
- referrer
- title
- URL

Methods:

- Element selection: get Element By Id, get Elements By Name, get Elements ByTag Name
- Document control: close, open, write, writeln

The Location Object

The location object represents the URL of the current web page.

Properties:

- host
- hostname
- href
- pathname
- port
- protocol
- search

Methods:

- URL manipulation: assign, reload, replace

The Navigator Object

The navigator object provides information about the web browser.

Properties:

- appName
- app Version
- browser Language
- cookie Enabled
- platform
- user Agent

The Screen Object

The screen object contains information about the client's display screen.

Properties:

- avail Height
- avail Width
- color Depth
- height
- pixel Depth
- width

The History Object

The history object provides access to the list of sites the browser has visited in the current window.

Properties:

- length

Methods:

- Navigation: back, forward, go

Note: For security reasons, sometimes browsers restrict scripts from accessing history properties.

❖ **Knowledge Check 2**

Fill in the blanks:

1)---------- is a current HTML page and its content.

2) ----------object means the URL of the current web page

3) ----------method is to replace an element.

❖ **Outcome Based Activity**

Create a web page which will demonstrate using HTML DOM.

11.6 Summary:

- The W3C Document Object Model (DOM) is a platform and language-neutral interface that enables programs and scripts to dynamically access and modify the content, structure, and style of documents.
- The Document object represents the HTML document displayed in a browser window. It includes various properties referencing different objects, allowing access to and modification of the document's content. This process is referred to as the Document Object Model, or DOM.

Features of the DOM:

- The DOM defines a standard method for accessing documents.
- The DOM is a standard maintained by the W3C (World Wide Web Consortium).
- The W3C DOM standard is divided into three main parts: Core DOM, XML DOM, and HTML DOM.

The Document object represents your webpage. If you need to access any element within an HTML page, you usually start by interacting with the document object. Here are some examples of how you can use the document object to access and manipulate HTML elements:

- Adding an HTML element
- Removing an HTML element

- Replacing an HTML element
- Changing an HTML element

JavaScript Global Objects:

JavaScript includes six global objects:

- document
- history
- location
- navigator
- screen
- window

11.7 Self-Assessment Questions

1) What is DOM? Advantages of using DOM.
2) Explain the concept on DOM. Explain features of DOM.
3) Explain HTML DOM document object.
4) Explain HTML DOM object for adding new.
5) Explain six global DOM object .
6) Explain method for adding element with suitable example.

11.8 References/Reference Reading

- HTML, CSS -The complete Reference –
- HTML, CSS -DESIGN and build websites by Jon duckett
- HTML, JavaScript & CSS -Ivan Bayross
- w3schools.com

Unit – 12

Java Script -IV

Learning Outcomes

- Students will be able to understand meaning of Function
- Students will have a clarity over the Benefits of using function
- Students will be able to understand Key Steps in using Function
- Students will be able to understand Types of functions- Built in Functions, User defined function,
- Students will be able to understand Validation in JavaScript

Structure

12.1 Meaning of Function: -

JavaScript functions are essential for performing tasks efficiently by allowing code reuse. They provide a way to define a block of code, assign it a name, and execute it whenever needed. In JavaScript, you can define a function using the function keyword.

Functions in JavaScript enable you to encapsulate logic and execute it multiple times throughout your codebase. By naming functions, you can easily reference and call them whenever necessary. This reusability enhances the maintainability and readability of your code.

12.2 Benefits of using function: -

JavaScript functions offer several advantages:

1. **Code Reusability**: Functions enable us to reuse code, saving us from rewriting the same logic multiple times.
2. **Reduced Code Length**: Functions help minimize the length of our programs by allowing us to encapsulate common tasks into reusable blocks of code.
3. **Reusable Code Blocks**: Functions act as reusable code blocks that execute whenever called, enhancing the modularity and maintainability of our codebase.
4. **Efficient Tool**: Functions are powerful tools for organizing and optimizing code, leading to more efficient development processes.
5. **Task Automation**: They are particularly useful for automating repetitive tasks, allowing us to call the same function multiple times to achieve the desired outcome.
6. **Built-in Functions**: JavaScript provides a plethora of built-in functions, further enhancing the language's versatility and capability.

12.3 Key Steps in using Function: -

To create a function in JavaScript, you need to follow two steps:

1. **Define a function:** A function in JavaScript is defined using the 'function' keyword, followed by a unique function name, a list of parameters (which can be empty), and a statement block enclosed in curly braces.
2. **Calling a function:** After defining a function, the next step is to call it. To invoke a function later in your code, you simply use its name followed by parentheses, optionally passing any required arguments.

- **Knowledge Check 1**

  Fill in the Blanks

1) A JavaScript function can be defined utilizing ----------reserve word.
2) ----------is a main benefit of function which saves lots of time.
3) Before ----------we have to ----------the function.

- **Outcome Based Activity**

Discuss the advantages of using functions in programming.

12.4 Types of functions: -

Mainly functions are categorized in two types: -

12.4.1 Built -in functions: -

These are the functions which are provided by language. Built in functions are categorize in to following: -

1) Number functions
2) String functions
3) Date functions
4) Math functions
5) Array functions

Common Built-in Functions:-

Functions	**Description**
Is Nan()	**Returns true,** only if the object is Not a Number. **Returns false,** only if the object is a number.
Parse Float (string)	When a string starts with a number, the function reads the string until it reaches the end of the number, at which point it breaks off the remaining text and returns the outcome. The function returns NaN in the event that the string doesn't start with a number.
Parse Int (string)	The function reads the string till it reaches the end of the integer if the string starts with one, then it cuts off the remaining portion of the string and returns the result.

	The function returns NaN (Not a Number) if the string does not start with an integer.
String (object)	Converts the object into a string.
eval()	returns the outcome of an arithmetic expression's evaluation.

Example of Built-in functions: -

```
<html>
<head>
<title>JavaScript Math sqrt() Method</title>
</head>
<body>
<script type = "text/javascript">
document.write("Demonstration of Built in functions <br>" );
      var s = Math.sqrt( 25 );
document.write("Square root of 25 is :" + s +" <br>" );

      var s = Math.sqrt( 256 );
document.write("<br />Square root of 256 is :" + s );
</script>
</body>
</html>
```

OUTPUT:

```
Demonstration of Built in functions
Square root of 25 is : 5

Square root of 256 is : 16
```

-

12.4.2 User Defined Functions

To create a function, we have to perform two steps:

1) **Define a function: -**
2) In JavaScript, a function must be defined before it can be used. The most common way to define a function is by using the function keyword, followed by a unique function name, a list of parameters (which can be empty), and a statement block enclosed in curly braces.
3) **Syntax for Function Creation:** The function keyword in JavaScript is used to declare functions. It is followed by a function name and parentheses (). Function names can consist of letters, digits, underscores, and dollar signs, following the same rules as variable names. Parameters, if any, are listed inside the parentheses. The code that the function will execute is enclosed within curly braces {}.

Syntax :-

```
function name (parameter1, parameter2, parameter3)
{
/code to be executed
}
```

Function parameters are recorded inside the parentheses () in the function definition.

Function contentions are the qualities gotten by the function when it is conjured.

Inside the function, the contentions (the parameters) carry on as local variables.

For Example: -

```
<script type = "text/javascript">
Function myfunc()
{
alert ("Hello friends!");
}
```

It defines a function called myfunc () that takes no parameters

1) **Calling a Function:** Calling a function is the subsequent step after its creation. To invoke a function later in the script, you simply use its name, as demonstrated in the accompanying code.

```
 <html>
<head>
<script type = "text/javascript">
      function myfunc() {
        document. Write ("Hello friends!");
      }
</script>
</head>
<body>
<p>Click the following button to call the function</p>
<form>
<input type = "button" onclick = "myfunc()" value = "Click Here">
</form>
<p>Demonstrations of Function ..</p>
</body> </html>
```

OUTPUT :-

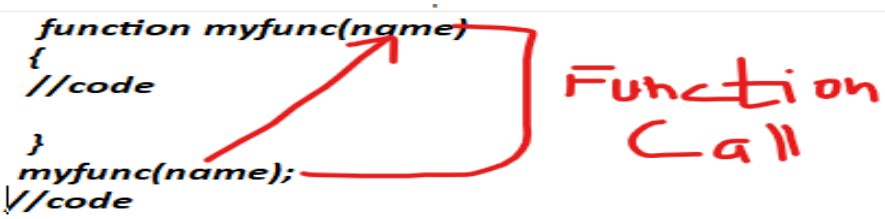
Click the following button to call the function

Click Here

Demonstrations of Function ..

Function Parameters: -

Similarly, functions in JavaScript can be declared with parameters. A value provided when defining a function is referred to as a parameter.

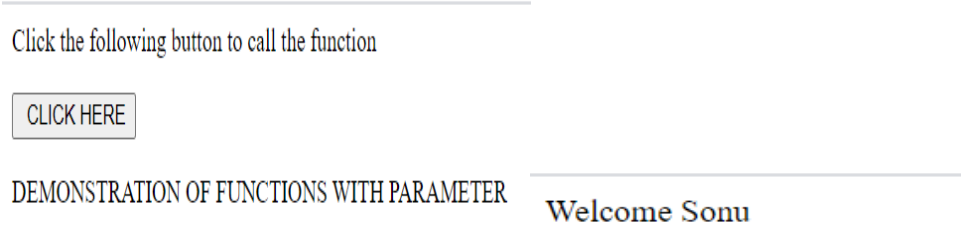


Working of JavaScript Functions with Parameters

For Example :-

```
<html>
<head>
<script type = "text/javascript">
        function myfunc(name)
        {
document.write ("Welcome  " + name);
        }
</script>
</head>
<body>
<p>Click the following button to call the function</p>
<form>
<input type = "button" onclick = "myfunc('Sonu')" value = " CLICK
HERE">
</form>
<p> DEMONSTRATION OF FUNCTIONS WITH PARAMETER </p>
</body>
</html>
```

OUTPUT :-



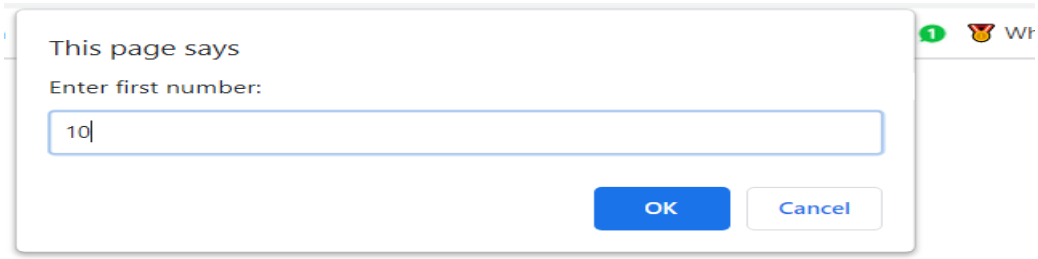
Function with Return Statement :-

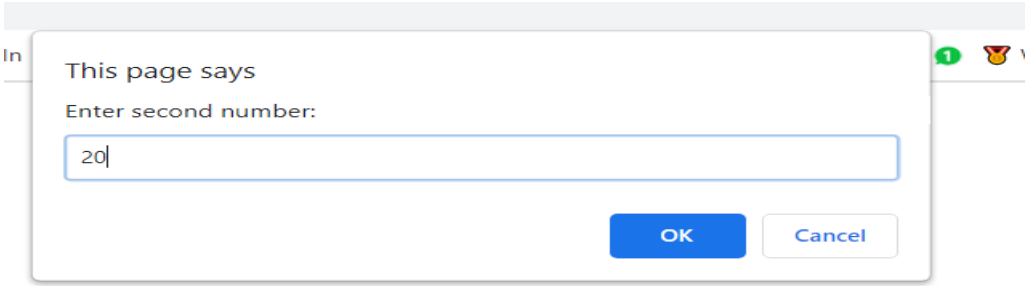
In programming, the return statement is used to send a value back to the function caller. It marks the end of the function's execution. Any code written after the return statement will not be executed. If a function does not explicitly return a value, it will return undefined.

For Example :-

```
<html>
<head>
<script type = "text/javascript">
    function add(a, b) {
   return a + b;
  }
  // take input from the user
  let n1 = parseFloat(prompt("Enter first number: "));
  let n2 = parseFloat(prompt("Enter second number: "));
 // calling function
 let result = add(n1,n2);
 // display the result
document.write ("The sum is " + result);
</script></head></html>
```

OUTPUT :-





The sum is 30

Local & Global Variable: -

Variables in programming store data that can be modified during the execution of a program. In JavaScript, the var keyword is used to declare variables. JavaScript recognizes two types of variables, and it also defines where in your program you can use the variables and functions you've created.

Local Variable: Local variables in JavaScript are those defined within functions. They have local scope, meaning they are accessible only within the functions where they are declared

```
<html>
        <body>
        <script>
        var data=200;//global variable
        function a(){
        var data=400;   // Local Variable
document.writeln("This is Local Variable  :- " + data + “<br>”);
        }    function b(){
```

OUTPUT :-

This is Local Varible :- 400
This is Global Varible :- 200

Global Variable: Local variables are those defined inside functions while using JavaScript. They can only be accessible within the functions in which they are specified since they have a local scope.

```
<!DOCTYPE html>
<html>
<body>
<h2>JavaScript Global variable Demo</h2>
<p id="demo"></p>
<script>
  var name = "I am Global variable";
myFunc();
   function myFunc() {
document.getElementById("demo").innerHTML = "I can display outside of function  &"  +  name;
   }
</script></body></html>
```

OUTPUT :-

JavaScript Global variable Demo

I can display outside of function &I am Global variable

12.5 Validation in JavaScript :-

It is critical to approve the structure presented by the client since it can have wrong qualities. Thus, approval is must to validate client.

JavaScript enables the office to approve the customer-side structure, which speeds up information handling compared to worker-side approval. Most site designers err on the side of JavaScript structure approval.

Through JavaScript, we can approve name, secret word, email, date, portable numbers and more fields.

- **Knowledge Check 2**

  State whether the statement is True or False

  1) global variables will be variables that are defined inside of functions.

  2) A large portion of the web designers lean toward JavaScript structure approval.

  3) Function parameters are recorded inside the parentheses ()

- **Outcome Based Activity**

  Demonstrate different types of functions .

12.9 Summary

- JavaScript functions are commonly used to reuse code.
- To create a function, two steps are involved:
  1. Defining a function
  2. Calling a function as defined
- Functions are primarily categorized into two types:
  - Built-in functions
  - User-defined functions
- **Variables:** Variables in JavaScript store data that can be changed. JavaScript uses the var keyword to declare variables. It also defines where in your program you can use the variables and functions you've created.
- **Parameters** can be defined with a function. A value provided during the function declaration is referred to as a parameter.
- To return a value from a function call, the return statement is used. The function execution ends upon encountering a return statement. Any code after the return statement is not executed. If no explicit value is returned, the function returns undefined.

12.10 Self-Assessment Questions

1) Explain meaning of Function. Explain advantages og using function in programming.
2) Explain types of functions with suitable example
3) Explain how to use built in functions.
4) Explain steps to create user defined function.
5) Explain function with return type.
6) Write a program by using passing parameters.
7) What is variable? explain types of variables.

12.11 References

- HTML, CSS -The complete Reference –
- HTML, CSS -DESIGN and build websites by Jon duckett
- HTML,Javascript& CSS -Ivan Bayross
- https://www.tutorialride.com/javascript/javascript-built-in-functions.htm

Unit - 13

Java Script -V

Learning Outcomes

- Students will be able to understand meaning of events in JavaScript
- Students will have a clarity over the different types of events in JavaScript
- Students will have a clarity over the event handling in JavaScript

Structure

13.1 Meaning of Events

In programming, an event refers to an action or occurrence triggered by the user or another source, such as a mouse click. An event handler is a mechanism that processes the event, enabling a programmer to execute code when the event occurs.

13.2 Working of Events in JavaScript

Events in JavaScript can be initiated by user interactions or by software and hardware processes. An event represents a signal emitted by hardware. Examples of events include:

1. A web browser fully loading a webpage.
2. Creating or modifying a file in a filesystem.
3. Hardware sensors, like webcams or microphones, capturing data.
4. Incoming network traffic.
5. Errors occurring at the browser or system level.

13.3 Why Events in Programming

Event-driven programming is a paradigm used in graphical user interfaces and various applications, such as JavaScript-based web applications. It focuses on executing specific actions in response to user input or system events.

- **Knowledge Check 1**

State weather statement is true or false

1) An event is a hardware condition that is signaled

2) An event is an activity that happens because of the user or another source,

3) An event handler is a mechanism that manages events, allowing a programmer to write code that is executed when the event occurs.

- **Outcome Based Activity**

Discuss how event-based coding helps the programmer to improve programming.

13.4 Different Events in JavaScript

Event Listeners (Handlers)

- **Recognizing and Processing Events**: Event listeners are responsible for identifying and handling events. They are objects that respond to specific events.

- **Notification:** A listener is an object that, if registered, is notified by the event source when an event occurs.

- **Registration:** Event listeners register their interest in certain events with event sources.

- **Passive Waiting:** Listeners typically remain inactive until an event happens.

Common events are from input devices:

- ✓ **mouse events -** mouse up, mouse down, mouse move
- ✓ **keyboard events** - key up, key down
- ✓ **Form Event -**
- ✓ **window events** - window resized, window dragged, focus lost

1) **Mouse Events: -**

Event Performed	Event Handler	Description
click	onclick	When mouse click on an element
mouseover	onmouseover	When the cursor of the mouse comes over the element
mouseout	onmouseout	When the cursor of the mouse leaves an element
mousedown	onmousedown	When the mouse button is pressed over the element
mouseup	onmouseup	When the mouse button is released over the element
mousemove	onmousemove	When the mouse movement takes place.

For Example :- on click Event

```
<html> <body>
    <script language="Javascript" type="text/Javascript">
        function clik() {
document.write("This is Onclick Event ");
        }
    </script>
```

```
<form>
<input type="button" onclick="clik()" value="Click Here"/>
</form>
  </body>
</html> “
```

OUTPUT :-Button before clicking

OUTPUT :- Button after clicking

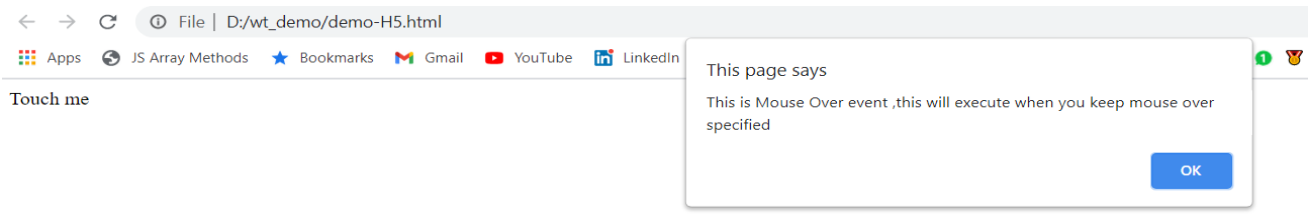
This is Onclick Event

```
  “<html>
  <head>
</head>
<body>
<script language="Javascript" type="text/Javascript">
  function mover()
  {
    alert("This is Mouse Over event ,this will execute when you keep mouse over specified");
  }
</script>
<p onmouseover="mover()"> Touch me</p>
</body> </html>”  “”
```

Click Here

For Example :-Mouse Over

OUTPUT – Mouse Over :-



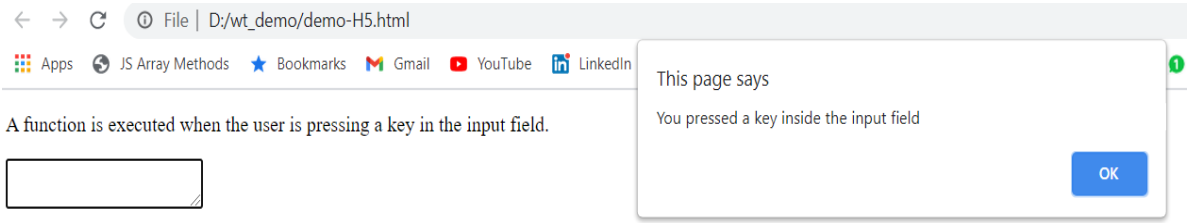
2) **Keyboard Events: -**

Event Performed	Event Handler	Description
Keydown	onkeydown	The event occurs when the user is pressing a key
Keyup	onkeyup	The event occurs when the user releases a key
Keypress	Onkeypress	The event occurs when the user presses a key

For Example: - 1) key down

```
“<!DOCTYPE html>
 <html>
<body>
<p>A function is executed when the user is pressing a key in the input field.</p>
<textarea  onkeydown="keyp()">
</textarea>
<script>
function keyp() {
  alert("You pressed a key inside the input field");
}
</script>
</body>
</html>”
```

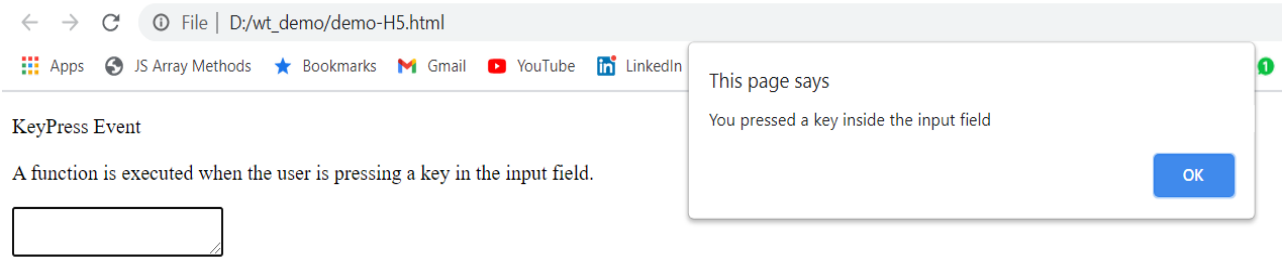
OUTPUT :-



For Example – 2) Key press Event: -

```
<!DOCTYPE html>
<html>
<body>
    <p> KeyPress Event </p>
<p>A function is executed when the user is pressing a key in the input field.</p>

<textarea  onkeypress="keyp()">
</textarea>
<script>
function keyp() {
 alert("You pressed a key inside the input field");
}
</script>
</body>
</html>
```

OUTPUT :- 2) Key press Event: -



3) **Form Events: -**

Event Performed	Event Handler	Description
focus	onfocus	When the user focuses on an element
submit	onsubmit	When the user submits the form
blur	onblur	When the focus is away from a form element
change	onchange	When the user modifies or changes the value of a form element

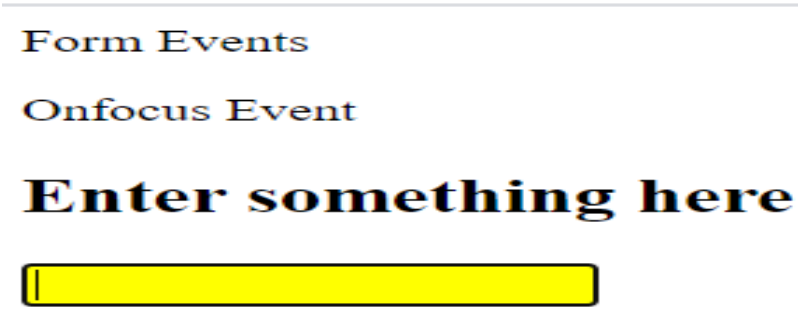
For Example: - 1) Focus Event

```
<html>
        <head> Form  Events</head>
        <body>
         <p> Onfocus Event </p>
        <h2> Enter something here</h2>
        <input type="text" id="d1" onfocus="focuse()"/>
        <script>

          function focuse()
          {
             document.getElementById("d1").style.background= " yellow ";
          }
        </script>
        </body>
        </html>
```

OUTPUT - 1) Focus Event

Before entering in Input field

After entering in Input field

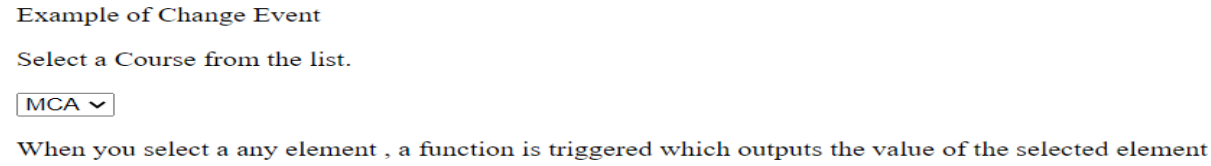


For Example: -2) Change Event

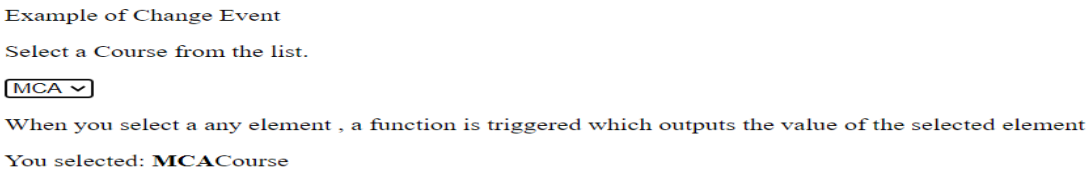
```
<!DOCTYPE html>
<html>
<body>
<p> Example of Change Event </p>
<p>Select a Course from the list. </p>
<select id="s1" onchange="ch()">
  <option value="MCA">MCA</option>
  <option value="MBA">MBA</option>
  <option value="BCA">BCA</option>
  <option value="BBA">BBA</option>
</select>
<p>When you select a any element , a function is triggered which outputs the value of the selected element </p>
<p id="demo"></p>
<script>
function ch() {
  var g= document.getElementById("s1").value;
  document.getElementById("demo").innerHTML = "You selected: " + "<b>" + g +"</b>" + "Course ";
}
</script></body></html>
```

OUTPUT :- 2) Change Event

Before Selection of list element



After Selection of list element



4)Window Events: -

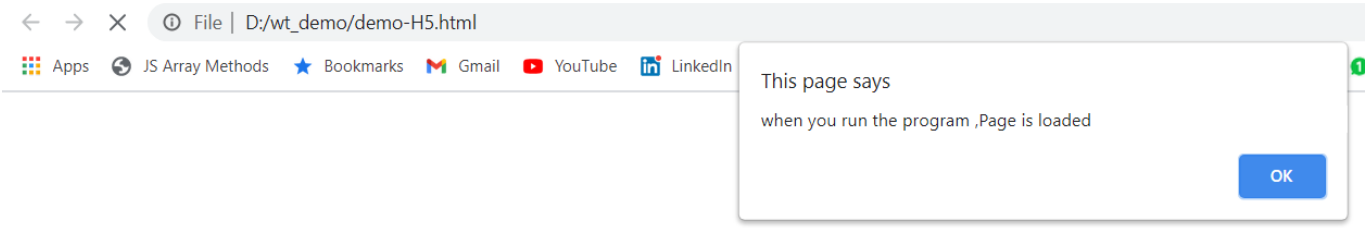
Event Performed	Event Handler	Description
load	onload	When the browser finishes the loading of the page
unload	onunload	When the visitor leaves the current webpage, the browser unloads it
resize	onresize	When the visitor resizes the window of the browser

For Example: - Load Event

```
<!DOCTYPE html>
<html>
<body onload="pload()">
<h1>Hello World!</h1>
<script>
function pload() {
  alert("when you run the program ,Page is loaded");
}
</script>
</body>
</html>
```

OUTPUT: - Load Event

Output after running program (page loaded)



Output after clicking on ok button

Hello World!

13.5 addEventListener()

The addEventListener() method is utilized to bind an event handler to a specific HTML element. Importantly, it does not replace existing event handlers. Events play a crucial role in JavaScript, enabling web pages to respond to user actions and API interactions. An event listener in JavaScript is a function that waits for a specific event to occur. The addEventListener() function is an integral part of JavaScript, facilitating the attachment of multiple event handlers to a single element without overwriting existing ones.

```
“element.addEventListener(event, function, useCapture);”
```

The addEventListener() method has three parameters, with "event" and "function" typically being used. The third parameter is optional and can be defined as follows:

Parameter Values:

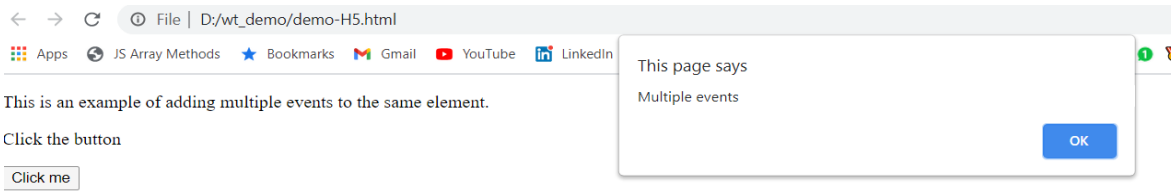
- **Event:** This is a mandatory parameter. It is specified as a string that denotes the name of the event.
- **Function:** Also mandatory, this parameter is a JavaScript function that executes when the event occurs.
- **UseCapture:** This parameter is optional. It is a Boolean value that determines whether the event handler executes in the capturing phase (true) or the bubbling phase (false). The default value is false.

For Example :- Use of Multiple Event

```
“<!DOCTYPE html>
 <html>
<body>
<p> This is an example of adding multiple events to the same element. </p>
<p> Click the button  </p>
<button id = "btn"> Click me </button>
<p id = "p1"></p>
<p id = "p2"></p>
<script>
function fun() {
   alert("Multiple events ");
}
function fun1() {
  document.getElementById("p1").innerHTML =  "This is second function";
}
function fun2() {
  document.getElementById("p2").innerHTML =  "This is third function";
}
var mybtn = document.getElementById("btn");
```

```
mybtn.addEventListener("click", fun);
mybtn.addEventListener("click", fun1);
mybtn.addEventListener("click", fun2);
</script>
</body>
</html>”
```

OUTPUT :-



- **Knowledge Check 2**

Fill in the Blanks

1) In the following syntax find out the missing part –

element. ---------- (event, function, useCapture);

2) ---------- events executes, When the visitor leaves the current webpage, the browser unloads it.

3) ---------events executes, when mouse click on an element

- **Outcome Based Activity**

Write & execute a program for resize event.

13.6 Summary

- In programming, an event refers to an action or occurrence triggered by a user or another source, such as a mouse click.
- An event handler is a mechanism that processes these events, enabling programmers to execute code when events occur.
- Events can be triggered by human interaction or by software and hardware processes that generate signals.
- Examples of Events:

1. A web browser fully loading a webpage.
2. Creating or modifying a file in a file system.

- There are various types of events categorized as follows:
    - Mouse Events: Examples include mouseup, mousedown, and mousemove.
    - Keyboard Events: Examples include keyup and keydown.
    - Form Events.
    - Window Events: Examples include resize, drag, and focus.

13.7 Self-Assessment Questions

1) Explain meaning of event. explain uses of events in programming .
2) Explain Mouse events with suitable example
3) Explain keyboard events with suitable example
4) Explain Form events with suitable example
5) Explain window events with suitable example
6) Explain addEventListener () with example.
7) Write a program which demonstrate use of form events.

13.8 References

1) HTML, CSS -The complete Reference –
2) HTML, CSS -DESIGN and build websites by Jon duckett
3) HTML, JavaScript & CSS -Ivan Bayross

https://www.w3schools.com/jsref/obj_keyboardevent.asp

https://www.javatpoint.com/javascript-addeventlistener

Unit – 14

Steps To Create A Website

Learning Outcomes

- Students will be able to understand Requirement for any website creation
- Students will gain an understanding of the steps involved in creating a website.
- Students will learn about website security, including firewalls, and how to maintain a website.
- Students will be able to understand the promotion and legal obligations for online business

Structure

14.1 What are the requirements to create a website?

In the previous chapter, we explored various techniques for creating user-friendly web pages. In this chapter, we delve into the basics of website creation, detailing the necessary steps and requirements. A website is essential for businesses to establish an online presence, whether for selling products or providing information and contact details.

Before starting to build your website, it's crucial to define your goals. Researching competitor websites can offer insights into effective strategies for your own site.

List of Operations in Website Creation:

14.1.1 Domain Name Registering a domain name forms the URL of your website (e.g., 'xyz.com'). It plays a critical role in how users find and perceive your site. Owning multiple versions of the domain name can help protect your brand. You can register your domain name through a domain registrar, typically on an annual basis.

14.1.2 Web Hosting Web hosting involves renting server space from a hosting company to store your website files. There are three main types:

1. **Shared Hosting:** Affordable option sharing server resources with other websites.
2. **VPS Hosting:** Offers more resources and capabilities with a virtual private server.
3. **Dedicated Hosting:** Provides exclusive server access, suitable for large businesses. Hosting costs are recurring, usually billed annually.

14.1.3 Business Email Address Having an email address linked to your domain name (e.g., 'me@mywebsiteURL.com') adds professionalism. This can be set up through your domain registrar or services like Google G-Suite.

14.1.4 Logo Design A logo visually represents your business and enhances brand recognition across various platforms, including websites, business cards, and signage. Designing a logo involves careful consideration and can be created by a graphic designer.

14.1.5 Favicon A favicon is a small icon that appears in browser tabs, enhancing website recognition. It can be created using your logo graphic file.

14.1.6 Images Images are crucial for visual appeal and conveying messages about your business, products, or services. Original, high-quality images are ideal for websites to enhance engagement and credibility. Properly labeled images can also improve visibility in search engines. Professional photographs are recommended; alternatively, high-quality images can be sourced from platforms like unsplash.com, pixabay.com, pxhere.com, or librestock.com.

- **Knowledge Check 1**

Fill in the Blanks

1) ----------are significant, for the website to look great as well as to show individuals what your items resemble or how you play out your administrations.

2) ----------is a example of favicon .

3) A logo is made by a ----------.

- **Outcome Based Activity**

Study the working of **unsplash.com** for getting Images.

14.1.7 Content of the Website: -

Creating content for your website can often pose a significant challenge. However, if you find it daunting, hiring a copywriter can be a worthwhile investment. Primarily, focus on crafting compelling content for your Home page that clearly outlines what you offer, who you are, and why visitors should choose you. Other essential pages where well-crafted content is crucial include:

Services: Detail the services you provide, highlighting their features and benefits comprehensively.

Products: Provide detailed information and reviews for each product, showcasing their unique features and advantages.

About Me/Us: This page offers insight into the individuals behind your offerings. Include profile photos and a compelling write-up that covers your background, journey, motivations, qualifications, and the inspiration behind your services/products.

FAQ (Frequently Asked Questions): Compile a list of common questions with clear, helpful responses to address visitor inquiries.

Testimonials/Reviews: Gather customer testimonials or reviews, ideally with accompanying photos for credibility.

Blog: Start with at least one blog post to engage your audience and provide valuable content regularly.

14.1.8 Web Designer: If you opt for a web designer, they will collaborate with you to design your website based on factors like services/products, target audience, preferences, and SEO requirements. Design fees are typically one-time costs, with additional work charged hourly. Choose a designer who communicates well and understands your long-term needs.

14.1.9 Secure Socket Layer (SSL): SSL encryption is crucial for website security, encrypting data transfers and changing your URL from HTTP to HTTPS. Websites without SSL may be flagged as insecure. Your web designer can install a free SSL certificate or help you purchase and install one.

14.1.10 Google Analytics: Integrating Google Analytics provides valuable insights into website performance, essential for tracking campaigns and optimizing your site's effectiveness. This service is typically included with your web design.

14.1.11 XML Sitemap: Your designer will enable XML sitemap functionality, aiding search engines in efficiently indexing all your website pages for better search engine visibility.

14.1.12 Website Security & Firewall: Implement additional security measures, such as security plugins and best practices like secure usernames and passwords, to protect against hacking attempts and malware.

14.1.13 Website Maintenance: Regular updates to your website's core framework, themes, and plugins are crucial for optimal performance and security. Consider learning to perform maintenance tasks yourself or opt for a maintenance service.

- **Knowledge Check 2**

Fill in the blanks :-

1) FAQ stands for ----------
2) ---------- & ---------- will give you the insights showing the adequacy or on the off chance that anything needs tweaking.
3) Your website URL convention will change from ---------- to ----------after using SSL.

- **Outcome Based Activity**

Study the working of Google Analytics. & How it is beneficial for any website?

14.2 Promoting Your Business Online: -

When setting up your marketing strategy, consider how to promote your business online. Online promotion can be more cost-effective than traditional methods like newspaper ads or TV commercials. It also allows for easy measurement of success, as you can track responses to your promotions.

Online tools enable you to reach a targeted audience with personalized messages, fostering ongoing customer relationships. Simple methods for online business promotion include:

- **Email Marketing:** Send emails to customers or contacts to promote events, sales, new products/services, or newsletters related to your business.

- **Banner Ads:** Display ads on various websites linking back to your own.
- **Pay Per Click (PPC) Advertising:** Also known as sponsored links or ads, PPC generates website traffic by displaying your site when users search for specific keywords.
- **Social Media Engagement:** Utilize platforms like Facebook, Twitter, YouTube, or photo-sharing sites to interact with customers.
- **Blogging:** Maintain an online journal (blog) to keep customers updated with business news.
- **Coupon Platforms and Location-Based Marketing Tools:** Utilize these tools to attract local customers and offer promotions.
- **Mobile-Friendly Website:** Ensure your website is optimized for mobile devices.
- **Mobile Apps:** Develop or use apps to enhance customer engagement.
- **Handling Online Reviews:** Manage customer reviews and feedback effectively to build trust.

14.3 Legal Obligations for Online Business:

Online Security and Fraud: Ensuring online security and preventing fraud are essential legal obligations to protect sensitive customer information. Robust security measures not only mitigate risks but also enhance customer trust.

Data Protection: Secure personal computers and mobile devices with strong passwords to prevent unauthorized access and data breaches. Regularly backing up data and establishing protocols for handling sensitive information within your business are crucial safeguards.

Online Scams: Beware of online scams, including phishing and pharming, designed to deceive individuals into divulging personal information. Educate yourself and your staff to recognize these scams and take preventive measures.

14.3 Summary:

In the previous chapter, we explored techniques for creating user-friendly web pages. Now, we'll delve into the basics of website creation, covering requirements and steps involved.

Having a website is essential in today's digital landscape, as customers expect businesses to have an online presence. Whether selling products or providing information, a website is indispensable. Before starting, define your website's objectives and study competitors' sites for insights.

Operations involved in creating a website include:

- Domain Name Registration
- Web Hosting
- Business Email Setup
- Logo and Branding
- Favicon Creation
- Image Selection
- Content Creation
- Web Design
- SSL Installation
- Google Analytics Integration
- XML Sitemap Creation
- Website Security
- Regular Maintenance

A web hosting provider offers server space for hosting your website files, ensuring online accessibility. Establishing a recognizable logo and using compelling images are crucial for visual branding and engaging customers.

Content creation for your website, though challenging, can be aided by hiring a copywriter. Some prefer DIY website creation with drag-and-drop tools like WordPress.

Online tools and analytics help target audiences effectively and maintain customer relationships. It's vital to stay updated on legal obligations, particularly concerning online security and fraud prevention.

14.4 Self-Assessment Questions

1) Explain Basic requirement for creation of website.
2) Explain the term Web Hosting in detail.
3) Explain the importance of Content in any website.
4) Who is web Designer?
5) What is Site Map? Explain XML Sitemap.
6) What is SSL? Explain working of SSL?
7) Which precautions we have to take while doing Online transaction?
8) What is Google Analytics? Explain Importance of Google Analytics in any website.
9) What do you mean by Favicon?
10) What do you mean by Website Maintenance?

14.6 References/Reference Reading

1) HTML, CSS -The complete Reference –

2) HTML, CSS -DESIGN and build websites by Jon duckett

3) HTML, JavaScript & CSS -Ivan Bayross