Master of Computer Applications (MCA)

Web Technology

(OMCACO203T24)

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

Course Introduction	i
Unit 1 Web Technologies	1- 5
Unit 2 Web Application	6 - 10
Unit 3 Web Essentials	11 - 15
Unit 4 Web Browser	16 - 20
Unit 5 Markup Languages	21 - 25
Unit 6 Introduction to XHTML	26 - 29
Unit 7 Fundamental Elements of HTML	30 - 37
Unit 8 CSS in web design	38 - 41
Unit 9 Style Rule Cascading	42 - 46
Unit 10 CSS Box Model	47 - 53
Unit 11 Concept of Java Script	54 - 59
Unit 12 Control structures in Java Script	60 - 65
Unit 13 Scoping Rules in JavaScript	66 - 70
Unit 14 Objects in JavaScript	71 - 75

EXPERT COMMITTEE

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

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

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

COURSE COORDINATOR

Mr. Pawan Jakhar
(Department of Computer and Systems Sciences, JNU Jaipur)

UNIT PREPARATION

Unit Writer(s)	**Assisting & Proofreading**	**Unit Editor**
Mr. Pawan Jakhar (Department of Computer and Systems Sciences, JNU Jaipur) Unit 1-7	Mr. Satender Singh (Department of Computer and Systems Sciences, JNU Jaipur)	Mr. Hitendra Agarwal (Department of Computer and Systems Sciences, JNU Jaipur)
Ms. Heena Shrimali (Department of Computer and Systems Sciences, JNU Jaipur) Unit 8-14		

Secretarial Assistance

Mr. Mukesh Sharma

COURSE INTRODUCTION

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

This course is meticulously designed to provide a comprehensive overview of web development, starting from the foundational concepts to advanced programming techniques. It spans the breadth of essential technologies and methodologies crucial for effective web design and dynamic web applications.

This course has 3 credits and is divided into 14 Units. It delves into the functionalities of HTTP protocol focusing on its request-response pattern, which underpins web communication. Students will explore web browsers and servers, the pivotal components of web interactions. The latter part of this unit is dedicated to the principles of web design. It covers a range of topics from browser compatibility, bandwidth considerations, and caching to the aesthetics of website layout, user-centric design, and effective navigation systems. This unit aims to equip students with the skills to plan, design, and publish websites that are both functional and visually appealing.

It covers everything from simple formatting, links, lists, and tables to the more complex forms and frames, alongside an introduction to XHTML and HTML5. Students learn to manipulate everything from backgrounds, fonts, and text to mastering layout control with margins and positioning. The evolution from basic CSS to CSS2 and the latest CSS3 features are thoroughly explored to enhance the stylistic and responsive capabilities of web design.

The client-side scripting using JavaScript, an essential skill for adding interactive elements to web pages. Topics covered include variables, functions, loops, conditions, and event handling, along with JavaScript's interaction with the Document Object Model (DOM). Advanced JavaScript topics delve into object-oriented aspects and DOM manipulations. This unit also covers XML, its applications, and key components including DTDs, schemas, and the transformation of XML data using XSL and XSLT, integrating these with JavaScript to create dynamic HTML content.

A powerful server-side scripting language used for developing dynamic and interactive websites. Starting from the basic syntax, it covers control structures, data types, functions, and form processing. Advanced topics include cookies, sessions, and object-oriented programming in PHP.
This includes hands-on experience with PHPMyAdmin, an administration tool for MySQL databases, and handling common database bugs.

Course Outcomes:

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

1. Explain the core concepts to develop a dynamic webpage by the use of java script and HTML.
2. Identify and Incorporate aesthetics and formal concepts of layout and organization to design websites that effectively communicate using visual elements.
3. Apply and Select markup languages for processing, identifying, and presenting information in web pages.
4. Interpret the fundamental computer theory to basic programming techniques and use scripting languages and web services to transfer data and add interactive components to web pages..
5. Describe the Concept and plan an internet-based business that applies appropriate business models and web technologies and multiple web technologies to create advanced web components.
6. Create and design websites using appropriate security principles, focusing specifically on the vulnerabilities inherent in common web implementations and Incorporate best practices in navigation, usability and written content to design websites that give users easy access to the information they seek.

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

Web Technologies

Learning Objectives:

1. To understand the meaning of the Web and the concept of web engineering.
2. To explore the different categories into which web applications are divided.
3. To recognize the various characteristics offered by web applications.

Structure:

1.1 Introduction to Web Engineering

1.2 Categories of Web Applications

1.3 Characteristics of Web Applications

1.5 Summary

1.6 Keywords

1.7 Self-Assessment Questions

1.8 References

1.1 Introduction to Web Engineering

1.1.1 What is the Web?

The World Wide Web, sometimes referred to as the "Web," is the collective term for the part of the Internet consisting of web pages that can be viewed with a Web browser. Many people use the terms "Internet" and "Web" interchangeably, assuming they are similar. In reality, however, the word "Internet" refers to the global network of servers that enables information sharing via the World Wide Web. Consequently, although the Web makes up a significant portion of the Internet, the two are not synonymous.

1.1.2 What is web engineering?

The use of systematic, controlled, and quantitative techniques in the creation, management, and maintenance of web-based applications is known as web engineering.

The unorganized creation of websites and apps has highlighted the gap between web developers and traditional software developers. Taken as a whole, web engineering represents a growing body of theoretical and empirical study, as well as a deliberate, proactive strategy.

1.1.3 Web application

A web application, or web app, is an application program that is hosted on a remote server and distributed over the internet using a browser interface. By definition, web services are web applications, although not all websites have web apps, many do.

Web developers create apps for various users and purposes, ranging from businesses to individuals, for many different reasons. Examples of commonly used web apps include e-commerce stores, online calculators, and webmail services. While a few online apps are limited to a specific browser, most are accessible through any browser.

1.2 Categories of Web Applications

The web application can be categorized into the following various types:

1.2.1 Web application focused on documents

Document-oriented websites are static HTML pages stored on the server that are delivered directly to the client upon request. Using the appropriate tools, the web pages are updated manually. These applications respond quickly and are static, simple, and stable. However, due to their static nature and the lack of regular updates, these apps can be costly to maintain. They also tend to have consistency issues.

1.2.2 Interactive Online Application

HTML Forms and CGI both provide interactive online applications. They include forms, selection menus, radio buttons, and more. These applications are quick and easy to use. The web pages and links in this type of application are generated based on user input.

1.2.3 Web application for transactions

Users can make changes to these kinds of online apps. These programs allow organized database searches and are more interactive. The database system effectively and consistently handles data.

1.2.4 Workflow-based web application

These online applications enable the transfer of labor across public, private, and corporate entities. Web services are included for interoperability. It is essential, reliable, and adaptable to manage workflow with business autonomy. The best example of these uses is seen in B2B e-commerce systems.

1.2.5 Web application for collaboration

These kinds of apps are primarily used in group settings where communication within the group is crucial. These web applications include chat rooms, online forums, e-learning platforms, and websites that distribute content and allow updating, such as Wikipedia.

1.2.6 Portal-oriented web application

These web applications are ones in which several information and service sources areconnectedviaasinglepointofaccess.Thefinestexamplesofportal-orientedapplicationsinclude search engines and community portals.

1.2.7 Pervasive web application

These programs offer personalized features for every device from any location at anytime. It supports a small number of devices and has minimal interaction features. A prior understanding of the context in which the web application is utilized for dynamic modification is required. One type of web application is location-based services.

1.2.8 Web application based on knowledge

Applications of this type are used to provide machine and human expertise. Web technologies that are semantically based under pin knowledge management. Knowledge reuse, connecting, and web mining are some examples.

1.3 Characteristics of Web Engineering

Here are the key characteristics of web engineering:

- **Availability**: Ensuring that services remain uninterrupted, requiring constant availability.
- **Performance and Scalability**: Focusing on optimization and scalability to efficiently handle varying loads.
- **Security Measures**: Implementing a robust security framework to protect user data and information.
- **Functional Design**: Prioritizing functionality to ensure that online applications operate correctly.
- **User-Friendly Interface**: Committing to creating user-friendly online applications, emphasizing ease of navigation and interaction.
- **Content Management**: Referring to the effective arrangement and display of data to enhance the user experience.
- **Tiered Architecture**: Adopting a tiered architecture to facilitate scalability and modular development.
- **Systematic Development**: Using methodical, controlled, and quantitative techniques at every stage of creation, use, and maintenance.

1.5 Summary

- The World Wide Web, often referred to as the "Web," encompasses the segment of the Internet that consists of web pages accessible via a web browser.
- Web engineering involves the use of systematic, controlled, and quantitative techniques in the creation, management, and maintenance of web-based applications.
- The various types of web applications include document-oriented, interactive online applications, transaction-based applications, workflow-based applications, collaboration tools, portal-oriented applications, pervasive applications, and knowledge-based applications.
- Web applications offer a range of features to enhance their efficiency.

1.6 Keywords

- Web engineering: The use of systematic, controlled, and quantitative techniques for the creation, management, and maintenance of Web-based applications is known as web engineering.
- Collaboration-based applications: These types of applications are primarily used in group settings where communication within the group is crucial. These web applications include chat rooms, online forums, e-learning platforms, and websites that distribute content and allow updating, such as Wikipedia.

- Portal-oriented applications: These web applications are those in which multiple information and service sources are connected via a single access point. Notable examples of portal-oriented applications include search engines and community portals.

1.7 Self-Assessment Questions

1. What do you mean by the World Wide Web?
2. What is web engineering?
3. What are web applications?
4. List down the different types of web applications.
5. Explain some characteristics of web applications.

1.8 References

1. Web Engineering: A Practitioner's Approach (2008) by Roger S. Pressman
2. Web Engineering (2016) by Rajiv Chopra
3. Web Engineering(2004)by Woojong Suh

Unit 2

Web Application

Learning Objectives:

1. To understand the meaning of the Web application.
2. To thoroughly understand the workings of web applications
3. To understand the meaning of conventional software.
4. To analyze the different types of architecture required for software applications.
5. To understand the need for software applications

Structure:

2.1 Web Applications

2.2 Conventional Software

2.3 Need for an Engineering Approach.

2.4 Summary

2.5 Keywords

2.6 Self-Assessment Questions

2.7 References

2.1 Introduction to Web Application

2.1.1 What is a web application?

A web application is software that runs inside of a web browser. Businesses need to interact with one another and deliver services from a distance. They use web apps to simply and safely connect with clients. Web applications are designed to be the most often used features on websites, such as social network news feeds, shopping carts, product search and filtering, and instant chat. They enable you to enjoy advanced capabilities without the need to install or configure any software.

2.1.2 Architecture of web applications

Web applications have client-server architecture. Their code is divided into two sections: client-side scripts and server-side scripts.

Architecture on the client side

The functionality of buttons and drop-down menus is handled by the client-side script. When the end user clicks on the link to the online application, the web browser launches the client-side script and generates the text and graphic elements for user interaction. The user can, for example, read articles, watch videos, or fill out a contact form. For instance, a client request is sent to the server when the submit button is clicked.

Architecture on the server side

The data processing is done by the server-side script. The web application server responds once it has processed the requests from the clients. Requests are often made to supply new, updated, or preserved data.

If the user clicks the Read More button, material will be returned to them. If the user clicks the Submit button, the application server will store their information in the database. After completing the data request, the server will occasionally provide the whole HTML page to the client. This is known as server-side rendering.

2.2 Conventional software

A conventional software model is a software development process built on a customary, structured methodology. This kind of software model divides the software development process into discrete, sequential phases, such as planning, analysis, design, implementation, testing, and maintenance. Since it is a linear development method, every step must be finished before moving on to the next.

2.2.1 Conventional Software Architecture

Data-centred architectures:

This design's central component will be a data store, which the other parts will periodically access in order to add, update, remove, or change the data that is currently there. The graphic depicts a typical data-centered strategy. The client software accesses a central repository.

Data flow architecture:

This kind of architecture is used when several computational manipulating components are used to transform input data into output data. Given that it has a collection of components called filters connected by lines and employs both pipes and filters, the diagram depicts pipe-and-filter architecture. Data is transferred from one component to the pipes.

Call and Return architectures:

Architectures that employ calls and returns are used to write simple programs to grow and adapt. This category has a wide variety of sub-styles.

Object oriented architecture:

A system's components contain data and the procedures needed to alter it. Message passing establishes coordination and communication between the components.

Layered architecture:

Several distinct levels are specified, and each layer carries out a particular set of actions. Every layer will perform specific tasks that gradually approach the machine instruction set.
Components will execute operating system interfacing (communication and coordination with the OS) at the inner levels and receive user interface actions at the outer layers.

2.3 Need for Software Applications

The following are points that highlight the importance of software applications:
Aids the user in accomplishing particular tasks: Application software should, in general, be created with the end user in mind. Their primary benefit will be assisting the user with specialized tasks across a range of industries, including business, entertainment, and education. Microsoft Word is a frequently used application that lets users create, edit, delete, and perform other actions on Word documents.

Manages and manipulates data: Applications are used by organizations to maintain and alter databases including data about clients, staff members, and other details. Customer relationship management (CRM) systems and enterprise resource management (ERM) systems are two well-known instances of application software.

Enables people to efficiently arrange information: Application software allows for the efficient development and management of data by individual users. For example, one of the most popular commercial tools for handling datasheets is Microsoft Excel.

2.4 Summary

1. Software that operates within a web browser is referred to as a web application.
2. Companies rely on online applications to securely and conveniently communicate with clients and provide remote services.
3. Common functionalities found on websites such as shopping carts, product search and filtering, instant messaging, and social network news feeds are all designed as web apps.
4. The architecture of web applications follows a client-server model, comprising client-side scripts and server-side scripts.
5. A traditional software model is structured around a conventional, methodical approach to software development.
6. This model divides the software development process into distinct, sequential phases including planning, analysis, design, implementation, testing, and maintenance.

2.5 Keywords

Client-side architecture: The client-side script handles the button and drop-down menu functionality. The web browser runs the client-side script and produces the text and visual components for user interaction when the end user clicks on the link to the online application.

Server-side architecture: The server-side script manages the data processing. After processing the client requests, the web application server responds. Requests typically involve updating, maintaining, or supplying additional data..

Layered architecture: Several distinct levels are specified, and each layer carries out a specific set of actions. Every layer will perform certain tasks that gradually approach the machine instruction set.

Object-oriented architecture: Data and the procedures needed to alter it are contained in a system's components. Message passing establishes coordination and communication between the components.

2.6 Self-Assessment Questions

1. In your words, what is a web application?
2. Explain the architecture of web applications.
3. What is conventional software?
4. Explain the different types of conventional software.
5. What is the need for software applications?

2.7 References

1. Web Engineering: A Practitioner's Approach (2008) by RogerS. Pressman
2. Web Engineering (2016) by Rajiv Chopra
3. Web Engineering (2004) by Woojong Suh

Unit 3

Web Essentials

Learning Objectives:

1. To understand the parts of web essentials
2. To understand the meaning of the Internet
3. To explore the workings of the Internet.
4. Understand the concept of internet protocols along with their different types.

Structure:

3.1 Web Essentials (The Internet)

3.2 Basic Internet Protocols

3.3 Summary

3.4 Keywords

3.5 Self-assessment questions

3.6 References

3.1 Web essentials

3.1.1 What is the Internet?

The internet is a global network of interconnected computer networks that enables communication and data exchange between billions of devices worldwide. It facilitates the transmission of information in various forms, including text, images, videos, and audio, across vast distances in real-time.

At its core, the internet relies on a set of standardized protocols, such as TCP/IP (Transmission Control Protocol/Internet Protocol), to ensure seamless communication between devices. These protocols define how data packets are transmitted, routed, and received across the network.

The internet encompasses a wide range of services and applications, including email, web browsing, social media, online gaming, cloud computing, and more. It serves as a platform for collaboration, information sharing, commerce, entertainment, and education, shaping virtually every aspect of modern society.

The internet has revolutionized the way people communicate, conduct business, access information, and interact with the world around them. It has democratized access to knowledge, empowered individuals and communities, and facilitated the rapid exchange of ideas on a global scale.

However, the internet also poses challenges and concerns, such as cyber security threats, privacy issues, digital divide, misinformation, and online harassment. Addressing these challenges requires ongoing efforts from various stakeholders, including governments, businesses, technology companies, and users.

Overall, the internet remains a transformative force that continues to evolve and reshape the way we live, work, and connect with one another in the digital age.

3.1.2 How Does the Internet Work?

Clients and servers are integral to the functioning of the Internet. In this setup, the client typically refers to a device like a laptop with a direct Internet connection, while servers are powerful computers indirectly connected to the Internet, hosting various websites. These servers are linked to the Internet through Internet Service Providers (ISPs) and are identifiable by their IP addresses.

Given that people find it challenging to remember long sequences of numbers, each website requires its own domain name. When users enter a domain name in their browser's search box, the server attempts to retrieve the corresponding IP address associated with that domain name. This process is facilitated by a DNS (Domain Name System) server in networking, which acts like an extensive phone directory. It searches for the IP address of the domain name requested, akin to finding someone's Aadhaar number in a large database once their name is known.

3.2 Basic Internet Protocols

A set of guidelines, known as internet protocols, governs data flow and communication across networks. Both the sender and recipient must adhere to these protocols to exchange data effectively. Here are some fundamental internet protocols:

1. **Internet Protocol/Transmission Control Protocol (TCP/IP)**:
TCP/IP comprises standard guidelines facilitating communication across diverse computers. Each machine connected to the Internet receives a unique serial number, known as an IP address, via the IP protocol. TCP defines the structure of IP packets and governs data transmission over the Internet. It ensures that packets contain information about the message's source, destination, and order of assembly. Additionally, TCP verifies the accurate transmission of messages to the intended location. TCP is often referred to as a connection-oriented protocol.

2. Hyper Text Transfer Protocol (HTTP):
HTTP, defined by the World Wide Web (www), is employed to transfer hypertexts over the Internet for information exchange. This protocol outlines the format and transmission requirements for data. It also specifies various responses that web browsers must provide when requesting to view a specific webpage. As HTTP facilitates the transfer of text, images, and multimedia items on the World Wide Web, users invariably engage with it when using a web browser.

3. Simple Mail Transfer Protocol (SMTP):
SMTP is essential for sending and forwarding emails. This protocol adds emails to the outgoing mail queue after retrieving the recipient's email address from the header. Upon sending the email to the recipient's address, SMTP removes it from the outgoing list. Messages or emails may include text, videos, pictures, and other media. SMTP also aids in configuring rules for communication servers.

4. Point-to-Point Protocol (PPP):
PPP establishes direct communication lines between two devices. It outlines the rules governing information exchange and authentication between two devices. For instance, PPP is utilized when connecting a PC to an Internet service provider's server or linking two routers for direct communication.

5. File Transfer Protocol (FTP):
FTP facilitates file transfer across systems using a client-server architecture. When receiving a file transfer request from another system, the File Transfer Officer (FTO) initiates a connection and verifies the identity of each computer using passwords and IDs. Subsequently, file transfer occurs between the

two computers.

6. Hyper Text Transfer Protocol Secure (HTTPS):

HTTPS is an extension of HTTP that utilizes the SSL/TLS protocol for authentication and encryption, ensuring secure communication across computer networks. Websites typically utilize HTTP, but an SSL certificate must be installed to enhance security, particularly if the website handles sensitive data. Verifying that a website uses HTTPS before entering sensitive information is crucial for security.

7. User Datagram Protocol (UDP):

UDP, a transport layer protocol, is unstable and connectionless. Unlike TCP, it does not establish a reliable connection between devices before data transmission and does not guarantee the receipt of data packets or their correct sequence. Instead, UDP transmits data packets to a destination without flow control or error checking.

3.3 Summary

- The internet is commonly described as an extensive, globally interconnected network of computers and electronic devices that facilitate communication and information exchange. It serves as a channel for internet communication, enabling users to share and retrieve information worldwide.
- Clients and servers are necessary for the Internet to function as it does. In this scenario, the client is a laptop with a direct Internet connection, while the servers are big computers that are indirectly connected to the Internet and house all of the websites.
- A collection of standards known as internet protocols regulates data flow and communication across networks. For effective data exchange, both the sender and recipient must comply with these protocols.
- Some of the basic protocols include TCP/IP, UDP, HTTP, PPP, SMTP, FTP, and many more.

3.4 Keywords

Hyper Text Transfer Protocol Secure (HTTPS):

HTTPS is an extension of HTTP that utilizes the SSL/TLS protocol for authentication and encryption, ensuring secure communication across computer networks. Websites typically utilize HTTP, but an SSL certificate must be installed to enhance security, particularly if the website handles sensitive data. Verifying that a website uses HTTPS before entering sensitive information is crucial for security.

User Datagram Protocol (UDP):

UDP, a transport layer protocol, is unstable and connectionless. Unlike TCP, it does not establish a

reliable connection between devices before data transmission and does not guarantee the receipt of data packets or their correct sequence. Instead, UDP transmits data packets to a destination without flow control or error checking.

The Internet Protocol/Transmission Control Protocol (TCP/IP) comprises standard guidelines facilitating communication across diverse computers. Each machine connected to the Internet is assigned a unique serial number, known as an IP address, via the IP protocol. TCP delineates the structure of IP packets and governs data transmission over the Internet.

3.5 Self-Assessment Questions

1. What do you mean by The Internet?
2. Illustrate the workings of the Internet.
3. What are internet protocols?
4. Explain the different types of internet protocols.
5. Differentiate between TCP/IP and UDP.

3.6 References

1. Internet Protocols: Advances, Technologies and Applications by Subrata Goswami, 2003
2. TCP/IP Protocol Suite E/4 Paperback–1 July2017 by Behrouz A.Forouzan
3. WEB TECHNOLOGIES Paperback– 22 November 2010 by Uttam K.Roy

Unit:4

Web Browser

Learning Objectives:

1. To understand the meaning of a Web browser and its functions.
2. To review the working of a URL
3. To understand the meaning of a web server and its features
4. To over view topics like virtual hosts and secure servers

Structure:

4.1 Web browser and its functions

4.2 URL

4.3 Web servers and their features

4.4 Virtual hosts and secure servers

4.5 Summary

4.6 Keywords

4.7 Self-Assessment Questions

4.8 References

4.1 Web browser

A web browser is software that enables users to access content on the World Wide Web. Once the user requests data from a web server, the web browser retrieves the requested information, and the webpage is displayed on the user's screen.

Put simply, a web browser is a program that let us visit websites on the World Wide Web (www). It retrieves the data and presents it as a web page, allowing users to communicate with a web server. In our daily lives, we always browse websites using Chrome. One of the most widely used web browsers is, in fact, Chrome.

4.1.1 Functions of a web browser

Our reliance on the Internet has grown significantly. The features of web browsers and their applications are listed below:

- A web browser's primary purpose is to give users an interface through which to submit requests and get data and information from web servers.
- Using CSS and JavaScript, it presents the information to the user through websites, complete with multimedia components (pictures and videos).
- A web browser serves as a gateway to other websites. Users may access multiple websites by only inputting the website's URL on the web browser.
- Additionally, browsers are equipped with navigation capabilities that facilitate web browsing, such as a navigation bar, tabs, and other tools. It is also possible to reload, browse back and forth between websites, and open numerous web pages simultaneously.
- Ensuring safe and secure website access is another function of web browsers. Malicious programme detection, pop-up alerts, and other security precautions help achieve this.
- By including plugins or extensions, browsers can also have their functionality enhanced. The extensions that are most frequently installed include Calendar, AdBlock, and other valuable features.

4.2 URL (Uniform Resource Locator)

A unified resource identifier, or URL, is a sort of address that identifies a resource on the World Wide Web and the protocol used to access it. It serves as a way finding indicator for web resources so that users may reach websites.

Users can access a particular webpage, movie, or other online resource by using the URL. Google will offer numerous URLs for resources that are linked to your search query when you do an inquiry. The hyperlinks to the web pages can be accessed using the stated URLs.

4.2.1 Structure of a URL

The structure of the URL includes the following:

- The optional port number on the server
- The resource is accessed by a protocol that is contained in it.
- The server's location
- An identification for fragments
- The location of the resource is contained in the server's directory structure.

4.3 Web Servers

A web server is a program that responds to user network requests and provides them with files that are used to construct web pages. Hypertext Transfer Protocol is used for this interaction (HTTP).
In essence, web servers are computers that store the HTML files that make up a website. They then provide the required website to a client when the client requests for it. You can use your laptop to access Facebook and type the URL into Google's search box. Now, in order to access the Facebook webpage, the laptop will submit an HTTP request to the web server, another computer. All the files (typically in HTTP format) that comprise the website, such as text, photos, gif files, and so on, are stored on this machine, also known as the web server. You can access the website once the web server has processed your request and sent the necessary files for the website to your computer.
Popular web servers include “Apache HTTP Server”, “Microsoft Internet Information Services (IIS)”, “Nginx”, and “Lighttpd”.

4.3.1 Features of a web server

The following are some features that the web servers provide:

- Web servers are capable of supporting many websites due to their ability to accommodate larger data storage capacities.
- Where to store all log files is made possible by an easy-to-configure log filesetup. (Log files are sound for additional analysis of online traffic.)
- Because it aids in bandwidth regulation, network traffic may be regulated, and downtime can be avoided while heavy online traffic is flowing.
- FTP websites are simple to create since they facilitate the transfer of huge files from one location to another.
- Directory protection and website settings are simple to setup.

- Virtual directories are simple to create and may be easily mapped alongside actual directories.

4.4 Virtual hosts and secure server

4.4.1 Virtual hosts

A virtual host is a specialized hosting company that focuses on providing cloud-based infrastructure solutions. These solutions comprise virtual computers, servers, storage, and other hybrid platforms that enable the hosting of software, data, and services. Virtual hosting encompasses various technologies and service models that allow individuals and businesses to access computer infrastructure services and solutions over the Internet.

4.4.2 Secure server

Servers that utilize the Secure Sockets Layer (SSL) protocol to secure communication from unauthorized parties are commonly referred to as secure servers. SSL servers facilitate encrypted communication between web browsers and other web servers, employing cryptography to ensure that data transmitted between them remains confidential and secure.

In general, most Web servers have some degree of security; requiring a login and password for password-protected sites is one such example. While theoretically effective in preventing unauthorized access, password-protected security measures are nothing more than a list of authorized users maintained on the domain owner's server.

4.5 Summary

- A web browser is software that enables users to access content on the World Wide Web. Once the user requests data from a web server, the web browser retrieves the requested information, and the webpage is displayed on the user's screen.
- A unified resource identifier, or URL, is a sort of address that identifies are source on the World Wide Web together with the protocol that is used to access it. It serves as a way finding indicator for web resources so that users may reach websites.
- A web server is a program that responds to user network requests and provides them with files that are used to construct web pages.
- Popular web servers include Apache HTTP Server, Microsoft Internet Information Services (IIS), Nginx, and Lighttpd.

4.6 Keywords

Virtual Hosts - A virtual host is a specialized hosting company that focuses on providing cloud-based infrastructure solutions. These solutions comprise virtual computers, servers, storage, and other hybrid

platforms that enable the hosting of software, data, and services.

Secure server-Servers that utilize the Secure Sockets Layer (SSL) protocol to secure communication from unauthorized parties are commonly referred to as secure servers. SSL servers facilitate encrypted communication between web browsers and other web servers, employing cryptography to ensure that data transmitted between them remains confidential and secure.

4.7 Self-Assessment Questions

1. What do you mean by a web browser? Also, explain the functions of a web browser.
2. What is a web server? Write down some features of a web server.
3. What is a URL?
4. Explain the structure of a URL.
5. What are virtual hosts and secure servers? Explain both in detail with suitable examples.

4.8 References

1. The Internet For Dummies by John R. Levine, 2002

2. Web Engineering: A Practitioner's Approach (2008) by Roger S. Pressman

3. Web Server Technology by Nancy J. Yeager, 1996

Unit 5

Markup Languages

Learning Objectives:

1. To understand the concept of markup languages
2. To understand the basics of HTML
3. To overview the structure of HTML
4. To understand the different characteristics of HTML

Structure:

5.1 Markup Languages

5.2 Introduction to HTML

5.3 Characteristics

5.4 Summary

5.5 Keywords

5.6 Self-Assessment Questions

5.7 References

5.1 Markup Languages

A computer language includes basic terms, titles, or tags that help organize a page's overall layout and content. Examples of markup languages include BBC, HTML, SGML (Standard Generalized Markup Language), and XML (Extensible Markup Language).

An example of HTML (Hypertext Markup Language) code that produces bold text on a web page may be seen below:

bold

The output of the above code is **bold**.

Programming languages are not the same as markup languages. It consists of special symbols strewn across plain language that, if deleted or overlooked, keep the explicit content intact. Alternatively, such markers can be read in a predetermined way that improves the text's presentation to the reader (e.g., make this text bold, make this text an ordered list).

One excellent example of markup is the HTML that comprises this webpage. The HTML tags specify how your web browser will display the content's textual elements. However, the majority of the language would still be readable by humans if JavaScript, HTML, and CSS were eliminated.

But HTML, like other markup languages, has limitations when it comes to developing computational, dynamic, or interactive behaviours. JavaScript, Perl, or PHP (PHP: Hypertext Preprocessor) are examples of programming languages that may be used to generate a web page with more complex features, including search.

5.2 Introduction to HTML

HTML, or Hyper Text Markup Language, is a markup language used for creating web pages. It combines elements of markup and hypertext. Hypertext refers to the links between web pages. HTML structures the content of web pages within tags, making it understandable to computers for appropriate processing. This language allows for the annotation of text, making it machine-readable for proper manipulation. Most markup languages, including HTML, are designed to be human-readable. Tags in HTML specify how the content should be displayed or altered.

HTML was invented by Tim Berners-Lee in 1991. The original version of HTML was HTML 1.0, and the first standard version, HTML 2.0, was released in 1995.

HTML employs predefined components and tags to instruct the browser on how to display the material correctly. Don't forget to include closing tags. The beginning tag's effects are applied by the browser to the conclusion of the page if it is absent.

For example: write a paragraph Here, is an

HTML tag.

5.2.1 HTML Structure

Below is an outline of an HTML page's fundamental structure:

<!DOCTYPE html>- The document type declaration, or <!DOCTYPE html> is not actually a tag. It indicates that a given document is a HTML document. There is no case sensitivity in the doctype declaration.

<html>-It is referred to as the root element of HTML. It contains all the other components.

<head>-Awebpage'sunderneathcomponentsarecontainedintheheadtag.Onthefront end of a webpage, elements located within the head are not shown. The following HTML elements are used inside the <head> element:

<style>:With the aid of CSS, this HTML tag enables us to add styling to our web pages and make them visually appealing.

<title>-When you visit a website, the title appears at the top of your browser and contains the title of the page you are now reading.

<base>-This element designates the root URL for every relative URL with in a page.

<noscript>-It refers to a segment of HTML that appears when the user's browser disables scripting.

<script>:Java Script is used to provide functionality to this tag on the page.

<meta>: This tag contains the website's meta data, which has to load each time a page is accessed. For example, you may utilize the standard UTF-8 encoding on your website by using the
Meta data charset. Users may then access your web page in their preferred language as a result. The tag is self-closing.

<link>-HTML, CSS, and Java Script are all connected by the 'link' tag. It shuts on its own.

<body>: A web page's visible content is contained within the body tag. Put otherwise, the material that the browser displays on the front end is the body content.

Any text edit or may be used to build an HTML document. Use either.html or.htm to save the text file. The file may be viewed in the browser as a webpage when it has been saved as an HTML document.

5.3 Characteristics of HTML

Here are some features provided by HTML:

1. **User-Friendly and Simple**: HTML uses tags for annotations that organize content and enhance readability for both users and browsers. These tags also enable the inclusion of CSS (Cascading Style Sheets), which creates a unique visual presentation for digital content.

2. **Semantic Structure**: HTML5 introduces several tags that allow for precise annotation of elements for their specific roles. For example, the <title> tag is used for page titles, while <header>, <footer>, , <p>, and <a> tags are used for structuring and navigation within a page.
3. **Search Engine Optimization (SEO)**: HTML5's structured nature significantly aids SEO. Search engines like Google, Yahoo!, and Duck Duck Go use web crawlers to index content by mapping keywords, facilitated by HTML's structured format, making sites more accessible and searchable.
4. **Client-Side Storage**: HTML5 enhances client-side storage capabilities with technologies such as local Storage and session Storage, allowing developers to store data on the client side. This is an important advancement over the limitations of cookies in browser storage.
5. **Offline Functionality with Service Workers & Cache API**: HTML5 supports offline capabilities through Service Workers, Indexed DB, and the Cache API, enabling web applications to function without an active internet connection. This feature is integral to Progressive Web Applications (PWAs), as exemplified by apps like Flipkart.
6. **Game Development with Canvas**: HTML5 provides the Canvas element, which is crucial for developing interactive games, although 3D games may require additional technologies for more complex graphics.
7. **Platform Independence**: HTML is compatible with nearly all devices with a minimal operating system, accessible through a browser. This compatibility extends to older devices, such as those running Symbian that could display HTML pages.
8. **Media Support**: HTML supports embedding media files such as videos, audio, and images. HTML5 further simplifies media playback with <video> and <audio> elements, offering controls, button visuals, and programmability for managing playback.

5.4 Summary

- A computer language consisting of simple terms, titles, or tags that help organize a page's layout and content. Examples of markup languages include BBC, HTML, SGML (Standard Generalized Markup Language), and XML (Extensible Markup Language).
- HTML stands for Hyper Text Markup Language, used in conjunction with a markup language to create web pages. HTML combines markup language and hypertext, referring to the links between web pages.
- HTML uses predefined elements and tags to guide the browser in displaying content correctly. It is important to include both opening and closing tags; without a closing tag, the browser may not apply the intended formatting correctly.
- The structure of HTML includes elements like <!DOCTYPE html>, <html>, <head>, and <body>.

5.5 Keywords

Structure of Semantics- One of the most anticipated aspects of HTML is this. Several tags are included in HTML5 to allow you to annotate certain components for their specific purposes. To give the content of a page a title, for instance, utilize the<title>tag.

<!DOCTYPE html>- The document type declaration, or <!DOCTYPE html> is not actually a tag. It indicates that a given document is an HTML document. There is no case sensitivity in the doctype declaration.

<head>-A webpage's underneath components are contained in the head tag. On the front end of a webpage, elements located within the head are not shown.

5.6 Self-Assessment Questions

1. What are markup languages?
2. What is HTML?
3. Explain the structure of HTML.
4. List down some important features of HTML.
5. Write an HTML sample code.

5.7 References

1. Web Technologies: HTML, Java script, PHP, Java, JSP, Asp.Net, Xml And Ajax, Black Book (2009)
2. ResponsiveWebDesignwithHTML5andCSS:DevelopFuture-proofResponsiveWebsitesUsingthe Latest HTML5 and CSS Techniques by BenFrain,2020
3. Learning Web Design: A Beginner's Guide to HTML, CSS, Java script, and Web Graphic by Jennifer Niederst Robbins

Unit 6

Introduction to XHTML

Learning Objectives:

1. To understand the meaning of XHTML.
2. To explore the relationship between XHTML and HTML.
3. To understand the syntax of an XHTML code.
4. Understand the different semantics elements of XHTML.
5. To understand the topic of Abstract syntax of XHTML

Structure:

6.1 Introduction to XHTML

6.2 Syntax and Semantics,

6.3 XHTML Abstract Syntax

6.4 Summary

6.5 Keywords

6.6 Self-Assessment Questions

6.7 References

6.1 Introduction to XHTML

Extensible HyperText Markup Language, or XHTML for short, is an application of XML (Extensible Markup Language) that reformulates HTML. HTML establishes the components that can be included in a document and how they are to be presented, whereas XML is a meta-language that lets users create their own unique markup languages. By fusing the HTML structure with XML's flexibility, XHTML makes documents more widely readable by XML parsers and compatible with web browsers.

6.1.1 Relationship with HTML

XHTML is fundamentally developed from HTML and mostly keeps the functionality and grammar of its parent language. Meanwhile, the main distinction is in its more rigid syntax and regulations. For instance, XHTML requires that all elements be correctly nested, closed, and in lowercase, whereas HTML is more tolerant of syntax problems. This implies that an XHTML page must rigorously abide by its standards in order to be presented correctly. Still, an HTML document may render successfully even with a few little errors.

6.2 XHTML Syntax Semantics

6.2.1 XHTML Syntax

Nearly all of the components that are valid in HTML are also valid in XHTML, and the grammar of the two languages is highly similar. However, you must take extra care while writing an XHTML page to ensure that your HTML document complies with XHTML. The following are crucial things to keep in mind while creating a new XHTML document or transforming an HTML document into an XHTML document:

- To begin an XHTML document, write a DOCTYPE declaration.
- All XHTML tags and attributes should only be written in lowercase.
- Properly close all XHTML tags.
- Place each tag in its proper place.
- Cite every property value.
- Prohibit the reduction of attributes.
- Substitute the id attribute for the name attribute.
- Deprecate the script tag's language property.

6.2.2 XHTML with Semantic Content

"Semantic" refers, in the context of web design and development, to using markup to communicate meaning regarding the content rather than merely its display. The need to provide material that is both machine- and human-readable has grown as web technologies have advanced. As a transitional

language between the more flexible HTML and the more rigid XML structure, XHTML is crucial to the movement of the web towards more semantics.

Following are the elements of Semantic XHTML:

- Heading elements (<h1>,<h2>,etc.): These reveal a document's hierarchy and structure.
- ,<ol>,and are the elements of a list: They indicate things on lists.
- For citations and references, use Link (<a>) and Block quote (<Blockquote>).
- Emphasis() and Strong (): These terms refer to significance and Heightened emphasis, respectively.

6.3 XHTML Abstract Syntax

The abstract syntax of XHTML, which is an XML adaption of HTML, has specific guidelines for content organization. The following are important points:

DOCTYPE Declaration: A DOCTYPE declaration has to be at the beginning of every XHTML document. DOCTYPE declarations come in three different forms:

- Strict
- Transitional
- Frameset

All tags need to be lowercase: Markup in XHTML is case-sensitive. It is therefore necessary to write all XHTML tags and attributes in lowercase.

Closing tags must be used: A closing tag that is equal to an XHTML must exist. Closing tags ought to be used on even empty components.

Attribute Quotations: The values of every XHTML attribute need to be quoted. In the absence of such information, your XHTML document is regarded as invalid.

Attribute Reduction: Attributes cannot be minimized in XHTML. It is necessary to provide the characteristic and its value directly.

The id attribute: The name attribute is swapped out with the id attribute. XHTML advises using id = "id" rather than name = "name".

The attribute of language: The language attribute of the script element in XHTML is obsolete; instead, use the type attribute.

Nested Tags: XHTML tags need to be correctly nested. If not, it will be presumed that your document is an improper XHTML document.

6.4 Summary

- Extensible Hyper Text Markup Language, or XHTML for short, is an application of XML (Extensible Markup Language) that reformulates HTML.
- XHTML is fundamentally developed from HTML and mostly keeps the functionality and grammar of its parent language. Meanwhile, the main distinction is in its more rigid syntax and regulations.
- Nearly all of the components that are valid in HTML are also valid in XHTML, and the grammar of the two languages is highly similar.
- "Semantic" refers, in the context of web design and development, to the use of markup to communicate meaning regarding the content rather than merely its display.
- The abstract syntax of XHTML, which is an XML adaption of HTML, has specific guidelines for content organization.

6.5 Keywords

1. **XHTML:** Extensible Hyper Text Markup Language, or XHTML for short, is an application of XML (Extensible Markup Language) that reformulates HTML. HTML establishes the components that can be included in a document and how they are to be presented. In contrast, XML is a meta-language that lets users create their own unique markup languages.
2. **Semantics:** "Semantic" refers, in the context of web design and development, to the use of markup to communicate meaning regarding the content rather than merely its display. The need to provide material that is both machine-and human-readable has grown as web technologies have advanced.

6.6 Self-Assessment Questions

1. What is XHTML?
2. Explain the relationship between XHTML and HTML.
3. Explain the syntax to write the XHTML code.
4. What are semantics in XHTML?
5. What is XHTML Abstract Syntax?

6.7 References

1. Beginning HTML, XHTML, CSS, and Java Script Paperback–Import, 29 December 2009 by Jon Duckett
2. HTML5 Black Book, CoversCSS3, Java Script, XML, XHTML, AJAX, PHP, and j Query, 2ed|BS| ePaperback–1 January 2016 by DT Editorial Services
3. HTML & XHTML The Definitive Guide, 6/Ed (English, Paperback, Musciano Chuck)

Unit: 7

Fundamental Elements of HTML

Learning Objectives:

1. To understand the different fundamental elements of HTML
2. To understand the types of lists and their syntaxes
3. To overview the topic of frames in HTML
4. To understand the form element in HTML and its different tags.
5. To learn the formation steps of an HTML page

Structure:

7.1 Fundamental HTML elements

7.2 Creating HTML pages

7.3 Summary

7.4 Keywords

7.5 Self-Assessment Questions

7.6 References

7.1 Fundamental HTML elements

A “start tag, some content, and an end tag define an HTML element”. For example,

sample paragraph .

Following are some types of HTML elements:

Nested HTML elements:

HTML elements have the ability to nest or contain other components. HTML documents are made up entirely of nested HTML components.

Empty HTML Elements:

Elements in HTML that are empty are those that have no content.

Without a closing tag, the br>tag is an open element that defines a line break.

7.1.1 Lists in HTML

Information lists are specified using HTML Lists. There might be one or more list components in every List. Three categories of HTML lists exist:

- Lists that are numbered or ordered (ol)
- List without order or List with bullets (ul)
- List of Definitions or Descriptions (dl)

Ordered or number list (ol)

All of the list items in ordered HTML lists are, by default, indicated with numbers. It is sometimes referred to as a “numbered list”. The “<ol>” tag initiates ordered List, whereas “ tag starts list elements”.

Example:

1. Apple
2. Banana

3. Grapes

Unordered lists(ul)

All of the list elements in an HTML Unordered list have bullets next to them. Another name for it is a list with bullets. List elements begin with the <li> tag, while the unordered List is introduced with the <ul> tag.

Example:

- Apple
- Banana
- Grapes

Description list or definition lists (dl)

Another list format that HTML and XHTML support is the HTML Description list. The term "definition list" refers to a list of entries that resembles an encyclopaedia or dictionary.

The three tags are present in the HTML definition list:

- “The List's beginning is defined by the<dl>tag”.
- “A term is illustrated with a<dt>tag”.
- “The word definition (description) is determined by the <dd>tag”.

Example:

p>Different types of fruits</p>

<dl>

<dt>Apple</dt>

<dd>Apple contains vitamin A</dd>

<dt>banana </dt>

```
  <dd>Banana contain vitamin B</dd>
  <dt>guava</dt>
  <dd>guava contains vitamin C</dd>
</dl>
```

HTML Nested lists

A nested list is a list within another list. A list of lists that has bullets inside of numbered lists is referred to as a nested list.

7.1.2 Tables in HTML

The HTML table tag may show data in tabular form (row*column). Multiple columns may be arranged in a row.

With the aid of “<tr>”, “<td>”, and “<th>” components, we can use “<table>”element to construct a table that displays data in tabular form. The “<tr> tag defines the table row”, “the <th> tag defines the table header”, and “the <td> tag defines the table data” in each table.

The page's layout, including the header, navigation bar, body text, and footer sections, is controlled by HTML tables. However, it is advised to manage the page layout using the div element rather than the table.

Example code to create a table in HTML:

```
  <table>
<tr><th>name</th><th>class</th><th>marks</th></tr>
<tr><td>aman</td><td>9th</td><td>68</td></tr>
<tr><td>rohit</td><td>10th</td><td>89</td></tr>
<tr><td>akash</td><td>11th</td><td>78</td></tr>
<tr><td>shweta</td><td>12th</td><td>85</td></tr>
```

</table>

Output:

Name	Class	Marks
Aman	9th	68
Rohit	10th	89
Akash	11th	78
Shweta	12th	85

7.1.3 Frames in HTML

The HTML<frame> element designates the specific space inside an HTML file that can be used to show another HTML webpage.

When used in conjunction with <frameset>, the <frame> element splits a web page into several portions or frames, each of which may hold a distinct website.

Syntax:

<frame-src="URL">

7.1.4 Forms in HTML

A portion of a page, including controls like “text fields, password fields, checkboxes, radio buttons, submit buttons, menus, and so on”, is called an HTML form.

An HTML form makes it easier for the user to input information suchas “name, email address, password, phone number, etc.”, that has to be transmitted to the server for processing. HTML forms are necessary to gather data from site visitors.

Syntax:

<formaction="serverURL" method="get|post">

//inputcontrols,e.g.textfield,textarea,radiobutton,button

</form>

Following are some tags used in forms:

<form>-It describes an HTML form that may be used to enter data.

<input>-An input control is defined by it.

<textarea>-A multi-line input control is defined.

<label>-It establishes an input element's label.

<fieldset>-It forms a grouping of the related elements.

<legend>-It establishes a<fieldset>element's caption.

<select>-A drop-down list is defined by it .

7.2 Creating HTML pages

7.2.1 How to create HTML pages?

Following are some steps through which you can easily create HTML pages:

First, launch the text editor

To write HTML code, we must first open a text editor such as Notepad or Notepad++. The screen shot of Notepad++, the text editor used to write HTML code, is seen in the image below.

Enter the HTML code in step two.

At this stage, we need to type the HTML code in a text editor. HTML code consists of several tags, each beginning with an opening tag and ending with a closing tag.

Save the HTML code in step three.

Once the entire HTML code has been typed, the file has to be saved in a folder ending in.html. By selecting the File menu and then the Save As option, we can quickly save the HTML file. Type the file name with the.html extension after that.

Run the HTML file in step four.

The final step involves opening an HTML file from its stored location and running it. The default browser will open and run the file.

7.3 Summary

- A "start tag, some content, and an end tag define an HTML element".
- HTML elements can nest or contain other components. HTML documents are made up entirely of nested HTML components.
- Elements in HTML that are empty are those that have no content.
- Information lists are specified using HTML Lists. There might be one or more list components in every List.
- Data may be shown in "tabular form (row*column) using the HTML table tag". Multiple columns may be arranged in a row.
- The HTML <frame> element designates the specific space inside an HTML file that can be used to show another HTML webpage.

7.4 Keywords

Forms: A portion of a page, including controls like "text fields, password fields, checkboxes, radio buttons, submit buttons, menus, and so on", is called an HTML form.

An HTML form makes it easier for the user to input information such as "name, email address, password, phone number, etc.", that has to be transmitted to the server for processing. HTML forms are necessary to gather data from site visitors.

Frames: The HTML<frame>element designates the specific space inside an HTML file that can be used to show another HTML webpage.

When used in conjunction with <frameset>, the <frame> element splits a web page into several portions or frames, each of which may hold a distinct website.

7.5 Self-Assessment Questions

1. What are HTML elements?
2. What is a list in HTML? Explain with an example.
3. What is a form element in HTML? Explain its various tags.
4. How to create frames in HTML
5. How to create an HTML page? Explain with all the necessary steps.

7.6 References

1. Web Technologies: HTML, Java script, PHP, Java, JSP, Asp.Net, Xml And Ajax, Black Book (2009)
2. Responsive Web Design with HTML 5 and CSS: DevelopFuture-proof Responsive Websites Using the Latest HTML 5 and CSS Techniques by Ben Frain, 2020
3. Learning Web Design: A Beginner's Guide to HTML, CSS, JavaScript, and Web Graphic by Jennifer Niederst Robbins

Unit 8

CSS in Web Design

Learning Objectives:

1. To understand the concept of CSS in web design
2. To explore the need for using CSS
3. To learn the features offered by CSS
4. Understand the syntax of writing CSS
5. Explore the different types of CSS

Structure:

8.1 Introduction to CSS

8.2 Features of CSS

8.3 Syntax for CSS

8.4 Types of CSS

8.5 Summary

8.6 Keywords

8.7 Self-Assessment Questions

8.8 References

8.1 Introduction to CSS

Cascading Style Sheets, commonly known as CSS, is a simple programming language designed to make it easier to create visually appealing web pages. CSS allows you to apply styles to web pages independently of the HTML that structures each page. It defines the ideal appearance of a webpage, including font choices, colors, spacing, and more. In essence, CSS enables you to customize the visual presentation of your website. Developers and designers use CSS to control how elements are displayed, including their position in the browser.

CSS uses rule sets to define styles, whereas HTML utilizes tags for structuring content. While CSS is straightforward to learn and understand, it provides significant control over the display of an HTML document.

8.1.1 Why to use CSS?

CSS is widely utilized in both web-based and off line applications:

- All current websites use CSS to enhance their visual design.
- Embedded device displays frequently use CSS to layout their user interfaces.
- You may also add CSS to feeds and feed items using RSS clients.
- Chat windows in instant messaging programs are likewise formatted with CSS.

8.2 Features of CSS

The following are the critical features of CSS:

Possibility for Web Design Career: Proficiency in CSS and HTML is a prerequisite for anybody wishing to start a web design career.

Website Design: With CSS, we can manage various styles such as text color, font style, paragraph spacing, column dimensions, and layout. It also allows us to control background colors and images, adjust layout designs, and provide different display settings for various screen sizes and devices, among many other effects.

Web Control: CSS is simple to learn and provides the ability to control HTML content. It is connected with the markup languages XHTML and HTML.

Other Languages: After learning the fundamentals of HTML and CSS, we may go on to other related technologies like Angular, PHP, and JavaScript.

8.3 Syntax for CSS

A CSS rule consists of a selector and a declaration block.

The selector specifies the HTML element you want to style.

The declaration block contains one or more declarations, each separated by a semicolon. In each declaration, a colon (:) separates the CSS property from its value. Semicolons separate individual CSS declarations, and curly braces enclose the declaration block.

Example:

```
p {
  colour:green;
  text-align:bottom-left;
}
```

8.4 Types of CSS

The three kinds of CSS are listed below:

- **Inline CSS**: Inline CSS styles an HTML element directly by using the style attribute within an HTML tag. While this method can be cumbersome for managing styles across a website, there are situations where inline CSS is useful, particularly when you cannot use style sheets or access CSS files.
- **Internal/Embedded CSS**: Internal CSS is included within the <style> tag in the <head> section of an HTML document. This method is efficient for styling individual pages, but applying styles across multiple pages can be cumbersome as the styles need to be repeated on each page.
- **External CSS**: External CSS involves linking an HTML document to a separate CSS file (commonly named styles.css). This approach provides a more efficient way to style web pages, as changes to the CSS file automatically apply to all linked pages, making it easier to maintain consistent styling across an entire website.

8.5 Summary

- ❖ Cascading Style Sheets, or CSS, is a simple programming language designed to make the creation of visually appealing web pages easier. CSS allows you to apply styles to web pages.
- ❖ It defines the ideal appearance of a webpage, including recommended fonts, colors, spacing, and more, enabling you to customize your site's look.
- ❖ CSS is widely used in both web-based and offline applications.

- ❖ CSS provides control over many design aspects, including text color, font style, paragraph spacing, column sizes, layout, background color and images, and the adaptation of layouts for different screen sizes and devices, among other effects.
- ❖ A CSS rule consists of a declaration block and a selector. In each declaration, a colon (:) separates the CSS property name from its value. Declarations are separated by semicolons, and declaration blocks are enclosed in curly braces.

8.6 Keywords

- **Inline CSS**: Inline CSS allows you to style an HTML element by directly applying a style property within the element's tag. While convenient for quick changes, managing a website solely with inline CSS can be cumbersome.
- **Internal/Embedded CSS**: Internal CSS is defined within the <style> tag in the <head> section of an HTML document. This method is useful for styling individual pages efficiently but requires repeating styles across multiple pages if the same styles are needed.
- **External CSS**: External CSS involves linking an HTML document to a separate CSS file, typically named external.css. This approach is more efficient for styling multiple pages, as changes in the CSS file automatically update all linked pages.

8.7 Self-Assessment Questions

1. What do you mean by CSS?
2. Why is CSS used?
3. List down some features of CSS.
4. Explain the syntax to write codes in CSS.
5. Explain the different types of CSS.

8.8 References

1. Beginning HTML, XHTML, CSS, and Java Script Paperback–Import, 29 December 2009 by Jon Duckett
2. HTML 5 Black Book, Covers CSS3, Java Script, XML, XHTML , AJAX, PHP, and jQuery, 2ed|BS|e Paperback–1January 2016 by DT Editorial Services
3. Mastering HTML, CSS & Java script Web Publishing (English, Paperback, Lemay Laura)

Unit 9

Style Rule Cascading

Learning Objectives:

1. To understand the concept of style rule cascading
2. To understand the topic of selectors in CSS and its types
3. To know how property and value are defined after selecting a selector
4. To understand the subject of inheritance in CSS
5. To review various text properties in CSS

Structure:

9.1 Style rule cascading

9.2 Inheritance

9.3 Text properties

9.4 Summary

9.5 Keywords

9.6 Self-Assessment Questions

9.7 References

9.1 Style rule cascading

A Cascading Style Sheet rule instructs the browser on displaying and using HTML. A rule can specify one HTML tag's appearance, or you can create your own rule that can be applied wherever and anywhere you like. The rule consists of three components: the selector, the property, and the value.

9.1.1 The selector

To choose the content you wish to style, apply CSS selectors. The CSS rule set comprises selectors. CSS selectors select HTML elements based on criteria such as id, class, type, attribute, etc. Following are the types of selectors in CSS:

CSS Elements Selector- The HTML element is chosen by name using the element selector.

CSS ID Selector - To choose a particular HTML element, the id selector uses the element's id property. Since an ID is always unique on a page, it is used topic k just a distinct component. The hash symbol (#) is typed first, and then the element's id.

CSS Class Selector - the class select or chooses HTML elements having a specific class attribute. It is applied with a historical flare. (complete stop) and then the name of the class.

CSS Class Selector for a particular element - Use the element name with class selection if you wish to indicate that only a single HTML element should be impacted. **CSS Universal Selector-** A wildcard character is the global selection. Every element on the site is chosen by it.

CSS Groups Selector - Use the grouping selector to choose every element with the exact style specifications. To reduce the amount of code, utilize the grouping selector.

9.1.2 Property and Value

After your selection is set up, you must define its properties and values. What you're trying to modify is the selector's property. Font, Colour, Background, Margin, and Text are a few examples.

Curly brackets { } enclose the value and the property. That would then make the entire thing's syntax:

Selector{Property:Value} For example:

H1{color: green}

9.2 Inheritance in CSS

The practice of passing down specific style attributes of an HTML element to its offspring, or child elements, is known as inheritance. This implies that unless the child elements have a specified style property of their own, they will automatically inherit any style property applied to the parent element. One of CSS's most valuable concepts is inheritance, which helps you write less code overall and makes it easier to maintain.

9.2.1 How does inheritance work in CSS?

Inheritance in CSS functions by transferring specific style property values from an element's parent to its descendant elements. Not every CSS property is passed down automatically. Certain attributes are automatically inherited, such as colour and font family; other properties must be explicitly inherited using the inherit keyword.

9.3 Text Properties

Word spacing, alignment, justification, and text transformation are just a few of the various manipulations that may be done with text using the CSS text formatting attributes.

Syntax:

```
Selector{
   property-name:/*value*/
}
```

Following are some text properties in CSS:

Text Colour: This property allows you to choose the colour of the text. You may use the RGB value "RGB (255,0,0)" or a colour name like "red" to determine the colour.

Text-align: This CSS property allows you to define the text's horizontal alignment inside a block element or table-cell box.

Text-decoration: Text content may be "decorated" with the text-decoration property.

Text-indent: This style, whose size can be in points, centimetres, or pixels, is used to indent a paragraph's initial line.

Text-justify: Text-align may be made to justify using this attribute. The words are stretched out into whole lines.

Text-shadow: This is used to give the text a shadow.
Letter-spacing: The space between the text's letters is specified using this attribute.
Line height: This property controls the distance between lines.
Direction: The text's direction may be adjusted with this parameter.
Word-spacing: This specifies the distance in units of words within a line.

9.4 Summary

- A Cascading Style Sheet rule instructs the browser on displaying and using HTML. One HTML tag's appearance can be specified by a rule, or you can create your own rule that can be applied wherever and anywhere you like.
- Comprises selectors. CSS selectors choose HTML elements based on criteria such as id, class, type, attribute, etc.
- The practice of passing down certain style attributes of an HTML element to its offspring, or child elements, is known as inheritance. This implies that unless the child elements have a specified style property of their own, they will automatically inherit any style property applied to the parent element.
- Word spacing, alignment, justification, and text transformation are just a few of the various manipulations that may be done with text using the CSS text formatting attributes.

9.5 Keywords

Cascading: A Cascading Style Sheet rule instructs the browser on how to display and use HTML. A rule can specify one HTML tag's appearance, or you can create your own rule that can be applied wherever and anywhere you like. The rule consists of three components: the selector, the property, and the value.

Selector: To choose the content you wish to style, apply CSS selectors. The CSS rule set comprises selectors. CSS selectors select HTML elements based on criteria such as id, class, type, attribute, etc.

Inheritance: The practice of passing down specific style attributes of an HTML element to its offspring,

or child elements, is known as inheritance. This implies that unless the child elements have a specified style property of their own, they will automatically inherit any style property applied to the parent element.

9.6 Self-Assessment Questions

1. What is Cascading in CSS?
2. Explain the term selector. Also, explain the different types of selectors.
3. What is property and value in CSS?
4. What is inheritance in CSS?
5. Explain the different types of text properties in CSS.

9.7 References

1. Beginning HTML, XHTML, CSS, and Java Script Paperback–Import, 29 December 2009 by Jon Duckett
2. HTML5 Black Book, Covers CSS 3, JavaScript, XML, XHTML, AJAX, PHP, and jQuery, 2ed | BS | e Paperback – 1 January 2016 by DT Editorial Services
3. HTML & XHTML The Definitive Guide, 6/Ed(English, Paperback, Musciano Chuck)

Unit 10

CSS Box Model

Learning Objectives:

1. To understand the concept of the CSS Box Model
2. To explore the different properties of the CSS Box Model
3. To understand the concept of Normal Flow Box Layout
4. To overview the topic of positioning in CSS and understand different types of positioning

Structure:

10.1 CSS Box Model

10.2 Normal flow box layout

10.3 Positioning

10.4 Summary

10.5 Keywords

10.6 Self-Assessment Questions

10.7 References

10.1 CSS Box Model

The CSS box model is a container that holds the content as well as various attributes like padding, margins, and borders. It is employed in the layout and design of websites. It functions as a tool set for adjusting how multiple pieces are arranged. The web browser displays each element as a rectangular box using the CSS box paradigm.

10.1.1 Properties of CSS Box Model

The following are the properties of a Box Model in CSS:

- **Content:** The information displayed in the box, including any text and photos
- **Padding:** Makes space for the content. The cushion is transparent.
- **Border:** A border around the text and padding
- **Margin:** Removes stuff beyond the boundary. The margin is observable.
- **Content Area:** This section contains text, pictures, and other types of media content. The content box determines its width and height, and the content edge sets its boundaries.
- **Padding Area:** This refers to the padding of the element. Actually, this region is the area inside the border box and around the content area.
- **Border Area:** The space between the box's padding and margin is known as the border area. The border's height and breadth indicate its size.
- **Margin Area:** The area that is between the border and the margin is known as the margin area. The margin-box width and height make up the measurements of the margin area. It's helpful to divide the elements

10.1.2 Height and width

Box height and width are set using CSS properties called Height and Width. You may set its value using auto, length, or percentage.

Syntax:

```
p {
   width: 80px; height:70px;
}
```

10.1.3 Margins and Padding

CSS Margins: To add space around an element, use CSS margins. We are able to adjust the margin widths for the top, right, bottom, and left sides separately.

The values of margin attributes can be as follows:

- Length in points, pixels,cm,etc.
- % of the element's width.
- Browser-calculated margin:auto.

Syntax:

```
body {
   margin:value;
}
```

CSS Padding: Within any specified border, space is created around an element using CSS padding. Different cushioning may be configured for each of the four sides (top, right, bottom, and left). In order to use padding properties, border properties must be included. The following values are possible for padding properties:

- Length in points, pixels, cm,etc.
- % of the element's width.

Syntax:

```
body {
   padding: value;
}
```

10.1.4 CSS border

The border's style, colour, and width may all be change dusing CSS border attributes. Different characteristics can be configured for each of the four borders—the top, right, bottom, and left borders. Following are some border properties:

1. Border Style: The border-style attribute specifies the type of border. Setting the border style is a prerequisite for using any other border property.

2. Border Width: This determines the border's width. Border width units might be pixels, points, centimetres, or thin, medium, and thick.

3. Border Colour: The border's colour may be adjusted using this option. The RGB value, hex value, or colour name can all be used to set colour. The border takes on the colour of the element if the colour is not specified.

4. Border individual sides: Using the border property, we may give each border a different width, style, and colour. However, each border's side must have a value assigned to it.

5. The border-radius: This attribute is used to create a more aesthetically pleasing border by rounding the corner.

10.2 Normal Flow Box Layout

Before their layout is altered, Block and Inline items are shown on a page using Normal Flow, also known as Flow Layout. In your layout, the flow is just a collection of elements connected and functioning as a unit. When anything is removed from the flow, it functions on its own.

When a document is written in the Writing Mode, words are presented in sentences in the inline direction, which is how inline components are displayed in a regular flow. Block components appear one after the other in the document's Writing Mode, like paragraphs. Thus, in English, block components begin at the top and show sequentially, while inline elements appear one after the other, starting on the left.

10.3 Positioning in CSS

An element's position may be set using the CSS position attribute. It is also helpful for scripting animation effects and for positioning a component behind another.The characteristics on the top, bottom, left, and right can be used to place an element. Only when the position property has been set may these properties be utilized. The calculated position attribute of a position element can be fixed, relative, absolute, or sticky.

Following are the types of positioning in CSS:

1) Static Positioning in CSS

This is where HTML components are placed by default. An element is always positioned in accordance with the page's natural flow.The attributes on top, bottom, left, and right have no effect on it.

Syntax:

```
div. static { position:static;}
```

2) Fixed Positioning in CSS

The text may be fixed in the browser with the aid of the selected positioning feature. Even when you scroll the window, this fixed test remains in place in relation to the browser window.

Syntax:

```
div. fixed { position:fixed;
 width: 500px;
 }
```

3) Relative Positioning in CSS

To set an element in relation to its standard location, using the relative positioning attribute.

Syntax:

```
div. relative { position:relative;
 left: 50px;
 }
```

4) Absolute Positioning in CSS

An element can be positioned absolutely with respect to its initial parent element, whose position is not static. An element may be placed anywhere on a page using absolute positioning. If no such component is discovered, then the containing block is HTML.

Syntax:

```
    div. relative { position:relative;
     width: 700px; height: 500px;
     }
div. absolute { position:absolute;
 top: 10px;
 width: 500px; height:300px;
}
```

10.4 Summary

- The CSS box model is a container that holds the content as well as various attributes like padding, margins, and borders. It is employed in the layout and design of websites.
- The web browser displays each element as a rectangular box using the CSS box paradigm.
- The different properties of a box model include Content, Padding, Margin, and Border.
- Before their layout is altered, Block and Inline items are shown on a page using Normal Flow, also known as Flow Layout. In your layout, the flow is just a collection of elements connected and functioning as a unit.
- An element's position may be set using the CSS position attribute. It is also helpful for scripting animation effects and for positioning a component behind another. The characteristics on the top, bottom, left, and right can be used to set an element.

10.5 Keywords

CSS Padding: Within any specified border, space is created around an element using CSS padding. Different cushioning may be configured for each of the four sides (top, right, bottom, and left).

CSS Margins: To add space around an element, use CSS margins. We are able to adjust the margin widths for the top, right, bottom, and left sides separately.

Absolute Positioning in CSS: An element can be positioned absolute concerning its initial parent element whose position is not static. A component may be placed anywhere on a page using absolute positioning.

10.6 Self-Assessment Questions

1. What do you mean by the CSS box model?
2. Explain some properties of the CSS box model.
3. Explain the different types of CSS borders.
4. What is a Normal Flow Box Layout?
5. What is positioning? Explain the different types of positioning.

10.7 References

1. Beginning HTML, XHTML, CSS, and Java Script Paperback–Import,29 December 2009 by Jon Duckett
2. HTML5 Black Book, Covers CSS 3, JavaScript, XML, XHTML, AJAX, PHP, and jQuery, 2ed | BS | e Paperback – 1 January 2016 by DT Editorial Services
3. Mastering HTML, CSS & Java script Web Publishing (English, Paperback, Lemay Laura)

Unit 11

Concept of Java Script

Learning Objectives:

1. To understand the basic concept of Java Script
2. To highlight the different features of Java Script
3. To understand the Method of memory management in Java Script
4. To learn the methods to take inputs from users in Java Script
5. To explore the other operators used in Java Script

Structure:

11.1 Introduction to Java Script

11.2 Obtaining user inputs

11.3 Memory concepts

11.4 Operators

11.5 Summary

11.6 Keywords

11.7 Self-Assessment Questions

11.8 References

11.1 Introduction to Java Script

Many websites utilize JavaScript (js), a lightweight object-oriented programming language, to script their web pages. When applied to an HTML page, this fully functional programming language that is interpreted allows for dynamic website interaction. It was first made available in 1995 so that users of the Netscape Navigator browser could add programs to their webpages. Every other graphical web browser has since embraced it. JavaScript allows users to create interactive, contemporary web apps that don't require page refreshing. The typical website uses Js to offer simplicity and several sorts of interactivity.

11.1.1 Features of Java Script

Following are some of the critical features of Java Script:

1) JavaScript adheres to the C programming language's structure and grammar. As a result, it is an organized programming language.
2) Some types in JavaScript are implicitly cast since the language is loosely typed (depending on the action).
3) Java Script is an object-oriented programming language that does not utilize classes for inheritance but employs prototypes.
4) It is an interpretive, light weight language.
5) The language is case-sensitive.
6) Numerous operating systems, such as Windows and macOS, support Java Script.

11.2 Obtaining user inputs in Java Script

In Java Script, there are several methods for obtaining user input. We'll look at three of the most popular techniques in this post:

I. Making use of the prompt () function
II. Event detection on HTML form elements
III. Applying input events to the add Event Listener () function

11.2.1 Employing the function prompt ()

Using the prompt () function is the most straightforward Method for obtaining user input in Java Script. This function shows a pop-up dialogue box with two buttons, "OK" and "Cancel," an input area where the user may write text and a message.

After the user hits "OK," the function returns a string representing the input value. The Method returns null if they click "Cancel" or shut the dialogue window.

Example:

Let user Age=prompt ("Please enter your age:"); console.log ("I am , " + user Age + "20");

11.2.2 HTML Form Elements: Listening for Events

Using HTML form elements like <input>, <textarea>, and <select> is a more popular and effective technique to obtain user input with JavaScript. These components let users input or choose data that may be accessed and changed using JavaScript.

You'll need to listen for particular events (like "click" or "submit") that happen when a user interacts with the form components in order to obtain the input data. To do this, add event attributes to the HTML elements (such as click or submit) and provide a JavaScript function that will be called when the event takes place.

11.2.3 Utilising Input Events with the add Event Listener () Method

The add Event Listener () function in JavaScript provides an additional means of obtaining user input. With this technique, you may affix an event listener to a specific HTML element, which then waits for a given event to transpire before carrying out a function in response.

Using event attributes directly in the HTML markup is less flexible and controllable than using the add Event Listener () function. It lets you simply handle events in external JavaScript files, add numerous event listeners to a single element, and keep your JavaScript code isolated from the HTML.

11.3 Memory Concepts in Java Script

In JavaScript, the runtime environment automatically takes care of memory management, which is

usually Node.js or the JavaScript engine found in web browsers.To manage memory and ensure developers don't have to manually allocate or deallocate memory, JavaScript uses a garbage collector. The steps of the memory life cycle are as follows, regardless of the programming language:

1. Allocates the necessary memory: Memory is allocated to the generated object by Java Script.
2. Utilize the memory that has been allotted.
3. When not in use, release the RAM assigned to it so that it may be used for other things. An engine that uses JavaScript handles it.

11.4 Operators in Java Script

Operators in Java Script are symbols that are used to manipulate operands. The JavaScript operator types are as follows.

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

11.4.1 Arithmetic operators

An arithmetic operator returns a single numerical value after accepting numerical values as operands, which can be literals or variables. The addition (+), subtraction (-), multiplication (*), and division (/) operators are the conventional arithmetic operators. When used with floating point values, these operators function as they do in the majority of other programming languages (notice, in particular, that division by zero generates Infinity).

11.4.2 Comparison (relational) Operators

A comparison operator determines if the comparison is valid by comparing its operands and returning a logical value. Values in the operands can be object, logical, string, or numeric. Using Unicode values, strings are compared according to conventional lexicographical ordering. JavaScript often tries to transform the two operands to a type that is suitable for comparison if they are not of the same kind.

Usually, this behavior leads to a numerical comparison of the operands. The === and!== operators, which carry out rigorous equality and inequality comparisons, are the only instances of type conversion inside comparisons.

11.4.3 Bitwise Operators

Rather than treating their operands as decimal, hexadecimal, or octal integers, a bitwise operator considers them as a collection of 32 bits (zeros and ones). For instance, the binary equivalent of nine, in decimal notation, is 1001. Although bitwise operators operate on such binary representations, the values they produce in Java Script are conventional numerical values.

11.4.4 Logical Operators

When employed with Boolean (logical) values, logical operators usually yield a Boolean value. When used with non-Boolean values, the && and || operators, however, may cause a non-Boolean value since they really return the value of one of the given operands.

11.4.4 Assignment Operators

Based on the value of its right operand, an assignment operator sets the value of its left operand. Equal (=), a primary assignment operator, transfers the value of its right operand to its left operand. In other words, the assignment statement x=f()assigns x the value of f().

11.5 Summary

- JavaScript allows users to create interactive, contemporary web apps that don't require page refreshing. Js is used by the typical website to offer simplicity and several sorts of interactivity.

- In JavaScript, there are several methods for obtaining user input. Like Making use of the prompt() function, Event detection on HTML form elements, and Applying input events to the add Event Listener () function.

- In JavaScript, the runtime environment automatically takes care of memory management, which is usually Node.js or the JavaScript engine found in web browsers. To manage memory and ensure developers don't have to manually allocate or deallocate memory, JavaScript uses a garbage

collector.

- ❖ Operators in Java Script are symbols that are used to manipulate operands.

11.6 Keywords

Arithmetic operators: An arithmetic operator returns a single numerical value after accepting numerical values as operands, which can be literals or variables. The addition (+), subtraction (-), multiplication (*), and division (/) operators are the conventional arithmetic operators.

Bitwise Operators: Rather than treating their operands as decimal, hexadecimal, or octal integers, a bitwise operator considers them as a collection of 32bits (zeros and ones).

Logical Operators: When employed with Boolean (logical) values, logical operators usually yield a Boolean value. When used with non-Boolean values, the && and || operators, however, may cause a non-Boolean value.

11.7 Self-Assessment Questions

1. What do you mean by Java Script?
2. List down some features of JavaScript.
3. Explain the memory management in JavaScript.
4. How to take user inputs in JavaScript?
5. What are the operators in Java Script? List down the types of operators.

11.8 References

1. Java Script and JQuery: Interactive Front-End Web Development by Jon Duckett, 2013
2. Java Script for Gurus Paperback–1 January2020 by Ockert J.du Preez
3. Java Script from Beginner to Professional Paperback – Import, 15 December 2021 by Laurence Lars Svekis

Unit 12

Control structures in Java Script

Learning Objectives:

1. To understand the basic concept of Control structures in JavaScript
2. To understand the different conditional statements
3. To learn about the constructs of Looping in JavaScript
4. To understand the Break and Continue statements
5. To explore the other functions in JavaScript

Structure:

12.1 Control Structures in Java Script

12.2 Looping constructs

12.3 Break and continue statements

12.4 Programmer-defined functions

12.5 Summary

12.6 Keywords

12.7 Self-Assessment Questions

12.8 References

12.1 Control Structures in Java Script

The control structures of JavaScript, C, and Java are comparable. Switch and if...else provide functionality for conditional statements. The while, do...while, and for constructs support loops.

12.1.1 Conditional statements in Java Script

In JavaScript, conditional statements decide whether or not a piece of code may execute and govern behaviour. JavaScript contains several different kinds of conditionals, such as:

"If" statements: these indicate that a block of code will be executed if a condition is met.

Syntax:

```
if(expression){
//material that has to be assessed
}
```

"Else"statements: they indicate how a block of code should be executed if the same condition is false.

Syntax:

```
if(expression){
//material that has to be assessed if the condition is true
}
else{
//material that has to be assessed if the condition is false
}
```

If...else if statement: Only when an expression is true across several expressions does it assess the content. Below is the JavaScript if else if statement's signature.

Syntax:

```
if(expression1){
//material to be assessed if expression 1 is true
}
else if (expression 2){
```

```
//material to be assessed if expression 2 is true
}
Elseif (expression3){
//material to be assessed if expression 3 is true
}
else{
//material to be assessed if no expression is true
}
```

12.2 Looping Constructs in JavaScript

The for, while, do while, or for-in loops in JavaScript are used to iterate the code. It reduces the code's bulk. Mostly, arrays utilize it.
In JavaScript, loops come in four different varieties:

- For loop

The items are iterated several times in the JavaScript for loop. If the number of iterations is known, it ought to be utilized. Below is the for loop's syntax:

```
For (initialization; condition; increment)
{
  code to be executed
}
```

- while loop

The items are iterated through an endless number of times by JavaScript while looping. If the number of iterations is unknown, this should be used. The while loop's syntax is seen below:

```
while (condition)
{
  Code to be executed
}
```

- do-while loop

The do while loop in JavaScript iterates through the components an unlimited number of times.

Nevertheless, the code is run at least once, whether the condition is true or false. Below is the syntax for the do while loop:

```
do{
   code to be executed
}while (condition);
```

- for-in loop

An object's properties can be iterated using JavaScript for-in loop.

12.3 Break and Continue Statements

12.3.1 Break statements

To break out of a loop, use the break statement. It has the ability to "jump out" of a switch () expression. It ends the loop and resumes running the code that follows.

12.3.2 Continue Statements

One loop iteration is "jumped over" by the continue statement. It ends one iteration in the loop and starts the subsequent iteration.

12.4 Programmer-defined functions

JavaScript's fundamental building element is a function. We may encapsulate a programming block and reuse it repeatedly by using functions. JavaScript code is easier to understand, maintain, reuse, and organize when it has functions.

The function keyword in JavaScript, the function name, and brackets are the order in which a function can be defined. The brackets can optionally include a list of input parameters. Curly brackets surround the code block that must be run when the function is invoked.

The following defines a function named welcome:

```
function welcome()
{alert("Hello World!");
}
```

12.4.1 Parameters of Function

Parameters are ways to pass values to a function. The calling code will send the values of any arguments that a function may have.

12.4.2 Return a Value

A function can use the return keyword followed by a variable or value to return a value to the caller code.

12.4.3 Function Expression

In JavaScript, a function expression is a function that is saved as a value, and that may be provided as an argument to another function or assigned to a variable.

12.4.4 Nested Functions

A function in JavaScript may contain one or more inner functions. The outer function's domain includes these nested functions. Variables and parameters of the external function are accessible to the internal function. Nevertheless, variables specified inside internal functions are not accessible by external functions.

12.5 Summary

- The control structures of JavaScript, C, and Java are comparable. Switch and if...else provide functionality for conditional statements. The while, do...while, and for constructs support loops.
- In JavaScript, conditional statements decide whether or not a piece of code may execute and govern behavior.
- The for, while, do while, or for-in loops in JavaScript are used to iterate the code. It reduces the code's bulk. Mainly, arrays utilize it.
- JavaScript's fundamental building element is a function. We may encapsulate a programming block and reuse it repeatedly by using functions.

12.6 Keywords

Break Statements: Use the break statement to break out of a loop. It has the ability to "jump out" of a switch () expression. It ends the loop and resumes running the code that follows.

Continue Statements: One loop iteration is "jumped over" by the continue statement. It ends one iteration in the loop and starts the subsequent iteration.

Nested functions: A function in JavaScript may contain one or more inner functions. The outer

function's domain includes these nested functions. Variables and parameters of the external function are accessible to the internal function.

12.7 Self-Assessment Questions

1. What are control structures in JavaScript? Explain conditional statements in detail.
2. What are looping constructs in JavaScript?
3. What are break and continue statements?
4. What are functions in JavaScript? Explain with an example.
5. What are nested functions?

12.8 References

1. Java Script and JQuery: Interactive Front-End Web Development by JonDuckett, 2013
2. Java Script for Gurus Paperback–1 January2020 by Ockert J.duPreez
3. Java Script from Beginner to Professional Paperback–Import, 15 December2021 by Laurence Lars Svekis

Unit 13

Scoping Rules in JavaScript

Learning Objectives:

1. To understand the basic concept of Scoping rules in JavaScript
2. To understand the concepts of Recursion and Iteration
3. To know how to declare and allocate arrays in JavaScript
4. To understand how arrays are passed in a function

Structure:

13.1 Scoping Rules

13.2 Recursion and Iteration

13.3 Arrays in JavaScript

13.4 Passing arrays to function

13.5 Summary

13.6 Keywords

13.7 Self-Assessment Questions

13.8 References

13.1 Scoping Rules

The area of execution, or the zone where values and expressions can be referenced, is known as the scope.

In JavaScript, there are two scopes: local and global.

Global Scope: Global scope refers to the ability to access variables declared globally—that is, outside of any function—from anywhere in a program. When defined outside of a block, let, and const are quite comparable to function scope variables declared with var.

Local Scope: Variables defined inside a function are considered local to that function and have a local scope. After a function begins, local variables are generated, and after the function is run, they are removed. Due to their Function Scope, local variables are only accessible from within the function.

13.2 Recursion and Iteration

13.2.1 Recursion in JavaScript

In the JS framework, recursion is the process by which a function calls itself within its definition in order to resolve an issue. This effective programming method makes it easier to create online apps that manage the intricate, layered data structures that are typical of JavaScript. In JavaScript, recursion is very helpful for navigating and working with hierarchical data, such as DOM elements and UI site components. It divides a complicated issue into smaller, more manageable parts that are all resolved using the same recursive reasoning.

13.2.2 Iteration in Javascript

In programming or JS frameworks, iteration is the process of looping over a series of instructions until a specific condition is satisfied. Iteration is frequently used in JS frameworks to run code blocks repeatedly. Processing arrays or groups of UI elements is a widespread usage for this. In contrast to recursion, this approach makes use of a simple loop structure rather than self-calling function calls.

13.3 Arrays in JavaScript

A JavaScript array is a type of data structure that lets you arrange and store several items in one variable. It's a dynamic and adaptable item. Numerous data kinds, such as integers, texts, objects, and even other arrays, can be stored in it. JavaScript arrays are zero-indexed, meaning that the index for the first element is 0, the index for the second element is 1, and so on.

JavaScript arrays can be created with the Array constructor or the square bracket-based shortcut array literal syntax. The size of an array can change dynamically as elements are added or deleted.

13.3.1 Declaration of Arrays

An array may be declared in essentially two ways: using an array literal or an array constructor.

1. Using Array Literal to Create an Array

Square brackets [] are used to define and initialize an array when creating an array literal. This approach is short and popular since it's so easy to use.

Syntax:

let array Name=[value1,value2,...];

2. Using the Array Constructor to Create an Array (a new JavaScript keyword)

Using the Array constructor function to create arrays is referred to as the "Array Constructor" method. This method may be used to build arrays with a specific length or number of entries, allowing for dynamic initialization.

Syntax:

let array Name=new Array();

13.3.2 Allocation of Arrays

Javascript reads the first statement, then allocates a contiguous memory for "arr." After reading the second statement, JavaScript allocates memory for that particular element. The problem is that memory allocation for index 3 through index 49 will not occur. Instead, it will just set aside memory for index 50.

13.4 Passing Arrays to a Function

The following are two ways to pass an array as a function parameter:

Method1: Applying the function apply ()

Using this method, you may call a function an array or object with an array-like structure as the parameters. There are two parameters in it. The arguments array holds the array of arguments to be supplied, and this value calls the function. The function that needs to be given as an array of parameters is applied using the apply() method. The arguments array is used to specify the second parameter, whereas the value of the first parameter is stated as "null." The function will be called as a result with

the given parameters array.

Syntax:

Array To Pass=[10,15,20];

Un modifiable Function. Apply (null, array To Pass);

Method2: Applying the spread syntax

When zero or more arguments are anticipated, the spread syntax is used. It may be applied to expanders (such as function parameters) that may not have a set number of expected arguments. To fill in the function's parameters from the array, the necessary function is called as provided by the arguments array using the spread syntax.

Syntax:

Array To Pass = [10, 15, 20]; un modifiable Function(...array To Pass);

13.5 Summary

- The area of execution, or the zone where values and expressions can be referenced, is known as the scope. In JavaScript, there are two scopes: local and global.
- In the JS framework, recursion is the process by which a function calls itself within its definition in order to resolve an issue. This effective programming method makes it easier to create online apps that manage the intricate, layered data structures that are typical of JavaScript.
- In programming or JS frameworks, iteration is the process of looping over a series of instructions until a specific condition is satisfied. Iteration is frequently used in JS frameworks to run code blocks repeatedly.
- A java Script array is a type of data structure that lets you arrange and store several items in one variable. It's a dynamic and adaptable item. Numerous data kinds, such as integers, texts, objects, and even other arrays, can be stored in it.

13.6 Keywords

Global Scope: Global scope refers to the ability to access variables declared globally—that is, outside of any function—from anywhere in a programme. When defined outside of a block, let, and const are quite comparable to function scope variables declared with var.

Local Scope: Variables defined inside a function are considered local to that function and have a local scope. After a function begins, local variables are generated, and after the function is run, they are

removed. Due to their Function Scope, local variables are only accessible from within the function.

13.7 Self-Assessment Questions

- What is scoping in JavaScript? Explain its different types.
- What is recursion in JavaScript?
- What is iteration in JavaScript?
- Explain the concept of arrays in JavaScript.
- How to pass an array in a function parameter?

13.8 References

1. JavaScript and JQuery: Interactive Front-End Web Development by Jon Duckett,2013
2. JavaScript for Gurus Paperback– 1 January 2020 by OckertJ.du Preez
3. JavaScript from Beginner to Professional Paperback – Import, 15 December 2021 by Laurence Lars Svekis

Unit 14

Objects in JavaScript

Learning Objectives:

- To understand the concepts of objects in JavaScript
- To overview different types of objects
- To understand the meaning and working of a Cookie
- To understand event handling in JavaScript and view different event handlers
- To understand the working of data validation using a regular expression

Structure:

14.1 Objects in JavaScript

14.2 Cookies in JavaScript

14.3 Data Handling

14.4 Data validation using regular expression

14.5 Summary

14.6 Keywords

14.7 Self-Assessment Questions

14.8 References

14.1 Objects in JavaScript

One of JavaScript's data types is represented by the Object type. It is employed to hold more complicated things and different keyed collections. The constructor Object () can be used to generate objects.

In JavaScript, almost every object is an instance of an Object; an object often inherits attributes from Object. prototype, including methods; however, these properties can also be shadowed (also known as overridden). Only objects with a null prototype or descended from other objects with a null prototype are exempt from inheriting from the Object.prototype.

14.1.1 Strings

A string in JavaScript is an object that symbolizes a series of characters. In JavaScript, there are two methods for creating strings.

- Using a string literal
- Through a string object (new keyword)

14.1.2 Date

The year, month, and day may be obtained using the JavaScript Date object. JavaScript's Date object may be used to display a timer on the page. To generate a Date object, you can utilize several Date constructors. It offers ways to obtain and set the hour, minute, seconds, day, month, and year. To build a Date object, you can use one of four Date constructor variants:

- Date ()
- Date (milliseconds)
- Date (date String)
- Date (year, month, day, hours, minutes, seconds, milliseconds)

14.1.3 Boolean

An object termed as JavaScript Boolean represents a value in one of two states: true or false. The JavaScript Boolean object has false as its default value.

14.1.4 Window

The window object represents a window in a browser. The browser automatically creates a window object. The window is a browser object—it is not a JavaScript object. Among the javascript objects are dates, arrays, and strings.

14.1.5 Document

The document object represents the whole HTML document. An HTML document becomes a document object when it loads in the browser. This root element means the HTML document. It contains attributes and functions. We can incorporate dynamic material into our webpage with the aid of document objects. It is an object of the window. Therefore, syntax is a window.document.

14.2 Cookies in JavaScript

A cookie is a little piece of data that lives on both the client and server sides. This data is stored by a web browser while you are surfing. Information is often included in a cookie as a string that is divided into name-value pairs and separated by semicolons. It preserves a user's state and keeps track of their information across all web pages.

14.2.1 Working of a cookie

- Any request that a user submits to the server is regarded as coming from a separate user each time.
- Therefore, we must include the cookie with the server response in order to identify the previous user.
- On the client-side browser,
- The cookie is now automatically included in requests that a user sends to the server. The server is able to identify users because of the cookie.

14.3 Handling events using JavaScript

An event is the modification of an object's state. HTML has a number of events that indicate if an action is taken by the user or the browser. JS responds to these events when HTML contains JavaScript code and permits execution. Event handling is the process of responding to events. As a result, Event Handlers in JavaScript manage HTML events.

For instance, provide JS code that will run the action to be taken on the event when a user clicks over the browser.

There are two methods that are advised for handler activation. One way to have event handler code execute when an event occurs is to set it to the relevant event property of the target element or use the add Event Listener () function to register the handler as an element listener. The handler will receive an object that complies with the Event interface (or a derived interface) in either scenario. The primary distinction is that the event listener techniques allow for the addition (or removal) of numerous event handlers.

14.3.1 Event Target .add Event Listener

Using the Event Target .add Event Listener function is the most adaptable approach to put an event handler on an element. With this method, listeners may be added to a component in multiples and deleted when necessary (using Event Target .remove EventListener).

14.3.2 Using On event Properties

JavaScript objects that trigger events are often associated with "onevent" properties, which are named by appending "on" to the event name. When the event is fired, these properties are called to launch related handler codes; your own code may also contact them directly.

You may just provide an event handler code to the relevant event property to set it. Each event in an element can only have one associated event handler. A different function can be assigned to the exact property in order to replace the handler if necessary.

14.3.3 Using an Abort Signal

The ability to simultaneously clear up numerous event handlers with an abort signal is a noteworthy feature of event listeners.

To do this, send the identical Abort Signal to each add Event Listener() method for each event handler you wish to be able to delete collectively. The event handlers that were created with the Abort Signal will all be removed when you use abort() on the controller that owns the signal.

14.4 Data Validation Using Regular Expression

Making sure the data you're dealing with satisfies specific requirements or limitations requires data validation, which is a crucial component of programming. Regular expressions and regular functions can be used in JavaScript for data validation. Character combinations in strings can be matched using regular expressions or regexes.

To suit your own needs, modify the validation criteria and regular expressions. When used for data validation, regular expressions may be quite effective; however, to find the best patterns for your use case, you may need to explore and test a little.

14.5 Summary

- One of JavaScript's data types is represented by the Object type. It is employed to hold more complicated things and different keyed collections. The constructor Object() can be used to generate objects.
- A cookie is a little piece of data that lives on both the client and server sides. This data is stored by a web browser while you are surfing.

- An event is the modification of an object's state. HTML has several events that indicate if an action is taken by the user or the browser. When HTML contains JavaScript code, JS responds to these events and permits execution.
- Making sure the data you're dealing with satisfies specific requirements or limitations requires data validation, which is a crucial component of programming. Regular expressions and regular functions can be used in JavaScript for data validation.

14.6 Keywords

Cookie: A cookie is a little piece of data that lives on both the client and server sides. This data is stored by a web browser while you are surfing. Information is often included in a cookie as a string that is divided into name-value pairs and separated by semicolons.

Document Object: The whole HTML document is represented by the document object. An HTML document becomes a document object when it loads in the browser. The HTML document is meant by this root element.

Window Object: A window in a browser is represented by the window object. The browser automatically creates a window object. The window is a browser object — it is not a JavaScript object.

14.7 Self-Assessment Questions

- What is an Object in JavaScript?
- Explain the different types of objects in JavaScript.
- What is a Cookie? Also, explain the workings of a cookie.
- What is event handling in JavaScript? Explain the types of event handling.
- What is data validation using a regular expression?

14.8 References

1. JavaScript and JQuery: Interactive Front-End Web Development by Jon Duckett, 2013
2. JavaScript for Gurus Paperback– 1 January 2020 by Ockert J.duPreez
3. JavaScript from Beginner to Professional Paperback– Import, 15 December 2021 by Laurence LarsSvekis