9

Website Design

Government of Nepal
Ministry of Education, Science and Technology
Curriculum Development Centre
Sanothimi, Bhaktapur
Phone : 5639122/6634373/6635046/6630088
Website- https://www.moecdc.gov.np
Email- info@moecdc.gov.np

Technical and Vocational Stream
Learning Resource Materials

Website Design
(Grade 9)
Computer Engineering

Government of Nepal
Ministry of Education, Science and Technology
Curriculum Development Centre
Sanothimi, Bhaktapur

Publisher: Government of Nepal
Ministry of Education, Science and Technology
Curriculum Development Centre
Sanothimi, Bhaktapur

Layout by Khados Sunuwar

Preface

The curriculum and curricular materials have been developed and revised on a regular basis with the aim of making education objective-oriented, practical, relevant and job oriented. It is necessary to instill the feelings of nationalism, national integrity and democratic spirit in students and equip them with morality, discipline, self-reliance, creativity and thoughtfulness. It is essential to develop linguistic and mathematical skills, knowledge of science, information and communication technology, environment, health and population and life skills in students. It is also necessary to bring the feeling of preserving and promoting arts and aesthetics, humanistic norms, values and ideals. It has become the need of the present time to make them aware of respect for ethnicity, gender, disabilities, languages, religions, cultures, regional diversity, human rights and social values to make them capable of playing the role of responsible citizens with applied technical and vocational knowledge and skills. This learning resource material for computer engineering has been developed in line with the Secondary Level computer engineering Curriculum with an aim to facilitate the students in their study and learning on the subject by incorporating the recommendations and feedback obtained from various schools, workshops, seminars and interaction programs attended by teachers, students, parents and concerned stakeholders.

In bringing out the learning resource material in this form, the contribution of the Director General of CDC Mr. Yubaraj Paudel and members of the subject committee Dr. Baburam Dawadi, Dr. Sarbim Sayami, Mrs. Bibha Sthapit, Mrs. Trimandir Prajapati is highly acknowledged. This learning resource material is compiled and prepared by Mr. Bimal Thapa, Mr, Rajendra Rokaya, Mr. Suresh Shakya. The subject matter of this material is edited by Mr. Badrinath Timsina and Mr. Khilanath Dhamala. Similarly, the language is edited by Mr. Narendra Bahadur Bogati. CDC extends sincere thanks to all those who have contributed to developing this material in this form.

This learning resource material contains a wide coverage of subject matters and sample exercises which will help the learners to achieve the competencies and learning outcomes set in the curriculum. Each chapter in the material clearly and concisely deals with the subject matters required for the accomplishment of the learning outcomes. The Curriculum Development Centre always welcomes creative and constructive feedback for the further improvement of the material.

2082 BS

Curriculum Development Centre
Sanothimi, Bhaktapur

Table of Content

Guidelines to Teachers

A. Facilitation Methods

The goal of this course is to combine the theoretical and practical aspects of the contents needed for the subject. The nature of contents included in this course demands the use of practical or learner focused facilitation processes. Therefore, the practical side of the facilitation process has been focused much. The instructor is expected to design and conduct a variety of practical methods, strategies or techniques which encourage students engage in the process of reflection, sharing, collaboration, exploration and innovation new ideas or learning. For this, the following teaching methods, strategies or techniques are suggested to adopt as per the course content nature and context.

Brainstorming

Brainstorming is a technique of teaching which is creative thinking process. In this technique, students freely speak or share their ideas on a given topic. The instructor does not judge students' ideas as being right or wrong, but rather encourages them to think and speak creatively and innovatively. In brainstorming time, the instructor expects students to generate their tentative and rough ideas on a given topic which are not judgmental. It is, therefore, brainstorming is free-wheeling, non-judgmental and unstructured in nature. Students or participants are encouraged to freely express their ideas throughout the brainstorming time. Whiteboard and other visual aids can be used to help organize the ideas as they are developed. Following the brainstorming session, concepts are examined and ranked in order of importance, opening the door for more development and execution. Brainstorming is an effective technique for problem-solving, invention, and decision-making because it taps into the group's combined knowledge and creative ideas.

Demonstration

Demonstration is a practical method of teaching in which the instructor shows or demonstrates the actions, materials, or processes. While demonstrating something the students in the class see, observe, discuss and share ideas on a given topic. Most importantly, abstract and complicated concepts can be presented into visible form through demonstration. Visualization bridges the gap between abstract ideas and concrete manifestations by utilizing the innate human ability to think visually. This enables students to make better decisions, develop their creative potential, and obtain deeper insights across a variety of subject areas.

Peer Discussion

Peer conversation is a cooperative process where students converse with their peers to exchange viewpoints, share ideas, and jointly investigate subjects that are relevant or of mutual interest. Peer discussion is an effective teaching strategy used in the classroom to encourage critical thinking, active learning, and knowledge development. Peer discussions encourage students to express their ideas clearly, listen to opposing points of view, and participate in debate or dialogue, all of which contribute to a deeper comprehension and memory of the course material. Peer discussions also help participants develop critical communication and teamwork skills by teaching them how to effectively articulate their views, persuasively defend their positions, and constructively respond to criticism.

Peer conversation is essential for professional growth and community building outside of the classroom because it allows practitioners to share best practices, work together, and solve problems as a group. In addition to expanding their knowledge horizon and deepening their understanding, peer discussions help students build lasting relationships and a feeling of community within their peer networks.

Group Work

Group work is a technique of teaching where more than two students or participants work together to complete a task, solve a problem or discuss on a given topic collaboratively. Group work is also a cooperative working process where students join and share their perspectives, abilities, and knowledge to take on challenging job or project. Group work in academic contexts promotes active learning, peer teaching, and the development of collaboration and communication skills. Group work helps individuals to do more together than they might individually do or achieve.

Gallery Walk

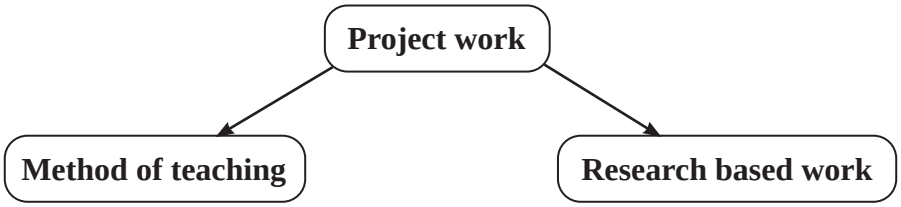
Gallery walk is a critical thinking strategy. It creates interactive learning environment in the classroom. It offers participants or students a structured way to observe exhibition or presentation and also provides opportunity to share ideas. It promotes peer-to-peer or group-to-group engagement by encouraging participants to observe, evaluate and comment on each other's work or ideas. Students who engage in this process improve their communication and critical thinking abilities in addition to their comprehension of the subject matter, which leads to a deeper and more sophisticated investigation of the subjects at hand.

Interaction

The dynamic sharing of ideas, knowledge, and experiences between people or things is referred to as interaction, and it frequently takes place in social, academic, or professional settings. It includes a broad range of activities such as dialogue, collaboration or team work, negotiation, problem solving, etc. Mutual understanding, knowledge sharing, and interpersonal relationships are all facilitated by effective interaction. Interaction is essential for building relationships, encouraging learning, and stimulating creativity in both in-person and virtual contexts. Students can broaden their viewpoints, hone their abilities, and jointly achieve solutions to difficult problems by actively interacting with others.

Project Work

Project work is a special kind of work that consists of a problematic situation which requires systematic investigation to explore innovative ideas and solutions. Project work can be used in two senses. First, it is a method of teaching in regular class. The next is: it is a research work that requires planned investigation to explore something new. This concept can be presented in the following figure.



Project work entails individuals or teams working together to achieve particular educational objectives. It consists of a number of organized tasks, activities, and deliverables. The end product is important for project work. Generally, project work will be carried out in three stages. They are:

- Planning
- Investigation
- Reporting

B. Instructional Materials

Instructional materials are the tools and resources that teachers use to help students. These resources/materials engage students, strengthen learning, and improve conceptual comprehension while supporting the educational goals of a course or program. Different learning styles and preferences can be accommodated by the variety of instructional

resources available. Here are a few examples of typical educational resource types:

- Daily used materials
- Related Pictures
- Reference books
- **Slides and Presentation:** PowerPoint slides, keynote presentations, or other visual aids that help convey information in a visually appealing and organized manner.
- **Audiovisual Materials:** Videos, animations, podcasts, and other multimedia resources that bring concepts to life and cater to auditory and visual learners.
- **Online Resources:** Websites, online articles, e-books, and other web-based materials that can be accessed for further reading and research.

Maps, Charts, and Graphs: Visual representations that help learners understand relationships, patterns, and trends in different subjects.

Real-life Examples and Case Studies: Stories, examples, or case studies that illustrate the practical application of theoretical concepts and principles.

C. Assessment

Formative Test

Classroom discussions: Engage students in discussions to assess their understanding of concepts.

Quizzes and polls: Use short quizzes or polls to check comprehension during or after a lesson.

Homework exercises: Assign tasks that provide ongoing feedback on individual progress.

Peer review: Have students review and provide feedback on each other's work.

Summative Test

Exams: Conduct comprehensive exams at the end of a unit or semester.

Final projects: Assign projects that demonstrate overall understanding of the subject.

Peer Assessment

Group projects: Evaluate individual contributions within a group project.

Peer feedback forms: Provide structured forms for students to assess their peers.

Classroom presentations: Have students assess each other's presentations.

Objective Test

Multiple-choice tests: Use multiple-choice questions to assess knowledge.

True/False questions: Assess factual understanding with true/false questions.

Matching exercises: Evaluate associations between concepts or terms.

Portfolio Assessment

Compilation of work: Collect and assess a variety of student work samples.

Reflection statements: Ask students to write reflective statements about their work.

Showcase events: Organize events where students present their portfolios to peers or instructors.

Observational Assessment

Classroom observations: Observe students' behavior and engagement during class.

Performance observations: Assess practical skills through direct observation.

Field trips: Evaluate students' ability to apply knowledge in real-world settings.

Abbreviations

AJAX: Asynchronous JavaScript and XML

API: Application Programming Interface

BEM: Block Element Modifier

CSS: Cascading Style Sheets

CSSOM: CSS Object Model

DOM: Document Object Model

ES: ECMAScript

HSL: Hue, Saturation, Lightness (color model)

HTML: HyperText Markup Language

HTTP: HyperText Transfer Protocol

ITCSS: Inverted Triangle CSS

JSON: JavaScript Object Notation

JSP: JavaServer Pages (not directly JavaScript, but related to web development)

JSX: JavaScript XML (syntax extension for React)

LESS: Leaner Style Sheets

Node.js: Node JavaScript (runtime environment)

NPM: Node Package Manager

OOCSS: Object-Oriented CSS

RGB: Red, Green, Blue (color model)

RGBA: Red, Green, Blue, Alpha (color model with opacity)

Sass: Syntactically Awesome Style Sheets

SCSS: Sassy Cascading Style Sheets (a syntax of Sass)

SMACSS: Scalable and Modular Architecture for CSS

SVG: Scalable Vector Graphics

URL: Uniform Resource Locator

V8: V8 JavaScript Engine (used in Google Chrome and Node.js)

W3C: World Wide Web Consortium

XHTML: Extensible HyperText Markup Language

Basics in Website Design

Unit 1

1.1 Brief History of the Internet

Many of you may have heard the word 'Internet'. Nowadays, it is widely used to search for information on any topic, interact with others, shop, stay up to date on current events, book tickets, play games, send greeting cards, and for a variety of other uses.

Internet technology refers to the devices, software, hardware, and transmission protocols used to connect computers so that data can be received or sent from one to another within a small network or as part of a small network within a larger network, like the Internet. The Internet is a network of networks, a worldwide communication system that connects hundreds of different networks. As a result, almost every computer on any network can communicate with any other computer on another network. These connections let users to exchange messages, converse in real-time (viewing messages and responses right away), share data and programs, and access endless amounts of information.

Today, the Internet connects thousands of networks and hundreds of millions of people worldwide. It is a vast, cooperative community with no centralized ownership. This

absence of ownership is a crucial aspect of the Internet since it implies that no single person or group controls the network. Anyone with access to the Internet can use it. If you can use a computer and it is connected to the Internet, you are free to not only use the resources provided by others, but also to generate your own resources, such as publishing documents on the World Wide Web, exchanging e-mail messages, and performing a variety of other jobs. The roots of the Internet were sown in 1969 when the Advanced Research Projects Agency (ARPA) of the United States Department of Defense began connecting computers at various universities and defense companies. The resulting network was named ARPANET.

1.2 World Wide Web (WWW)

The World Wide Web was founded in 1989 at the European Particle Physics Laboratory in Geneva, Switzerland. The term "World Wide Web" refers to the interlinked collection of hypertext documents and multimedia information available on the Internet. You use a Web browser to search, find, view, and download information from the Internet. In recent years, the Web has emerged as one of the most widely used Internet services. The Web is a collection of Internet host systems that provide these services via the Internet via HTTP (Hypertext Transfer Protocol). Web-based information is typically given in the form of hypertext and hypermedia via HTML.

Many people think that the Web and the Internet are the same thing, but this is incorrect. They are two distinct things. The Web is a service (a means for accessing documents) powered by the Internet (a massive network).

The World Wide Web, commonly known as the Web, is a collection of websites or web pages stored on web servers and local computers and accessible over the Internet. Web components include URLs, HTTPS, and HTML.

1.3 Web Standards

Web standards are a formal set of specifications and technology used to create websites and web applications. It is a technical document that defines several components of the World Wide Web (WWW). It is established by standards bodies, such as institutions, who ask representatives from various technology businesses to collaborate. The purpose of the World Wide Web Consortium (W3C) is "to lead the World Wide Web to its full potential by developing protocols and

guidelines that ensure long-term growth for the Web". The W3C fulfills its objective by developing Web standards and recommendations. In 1994, the W3C issued over a hundred of these standards, known as W3C Recommendations.

1.4 Web Protocols

A set of rules followed for interconnection and communication between computers in a network is called web protocol. So it is rules followed while sending and receiving information using network software. NCP (Network Control Protocol) was the first protocol. Some common protocols are as follows:

a. The Internet uses TCP/IP (Transmission Control Protocol/Internet Protocol).

b. HTTP (Hyper Text Transfer Protocol) is used to transfer HTML documents on the WWW.

c. SMTP (Simple Mail Transfer Protocol) is used to send emails.

d. POP (Post Office Protocol) retrieves e-mail from a mail server.

e. FTP (File Transfer Protocol) transfers files between computers.

f. UDP refers to User Datagram Protocol. It is one of the main protocols in the Internet Protocol stack, alongside TCP (Transmission Control Protocol). It is used to transmit data over a short distance.

g. IMAP stands for Internet Message Access Protocol. It is a protocol used by email clients to get emails from a mail server via a TCP/IP connection.

1.5 Web Browser

A Web browser is a software application that finds hypertext documents on the Web and then opens them on the user's computer. Examples include Mozilla Firefox, Google Chrome, Microsoft Edge, Brave, Safari, and Opera. It is a client application that allows the client computer to connect to a Web server or other Internet servers like FTP and Gopher. A browser can also comprehend and display documents. Mosaic was the original online browser, and Mozilla Firefox and Google Chrome are the most used browsers today.

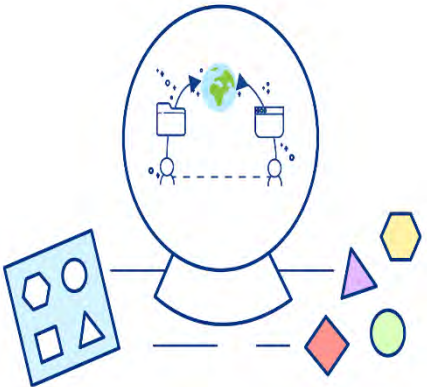
www

Web Site

A website is a collection of related web pages, including text, graphics, audio, video, and links, typically identified by a common domain name and published on a web server. Examples are moecdc.gov.np, gorkhapatraonline.com, neb.gov.np etc.

Web Pages

Web pages, created using Hypertext Markup Language (HTML), are hypertext documents on the internet that can be interpreted and displayed by web browsers, and are used to distribute news, educational services, product information, catalogues, and more.



Web Standards

Home Page

The home page is the first page loaded in a web browser, and can be changed using Tools options when typing a domain name in the address bar.

1.6 Search Engine

Assume you wish to look through Chapter 1 of your book. What would you do? You would look at the contents page, search for the chapter, find the page number, and then open it. Similarly, it is not always easy to find what you're looking for on the internet.

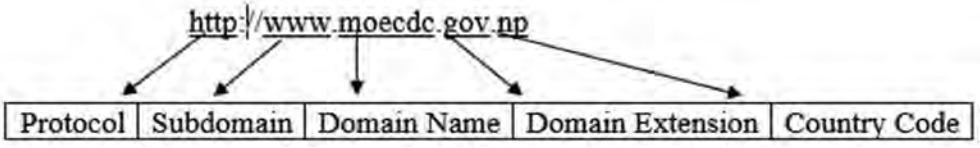
Search Engine

That's because there are tens of millions of distinct web sites, each with billions of unique pages! Search engines are, well, "engines" or "robots" that cruise the internet seeking for new web pages by inputting one or more words. These robots

read web pages and save the text (or portions of the content) to a big database or index that you can access. The engine then displays a list of web pages containing information relating to your words, known as a keyword. None of them cover the entire Internet, but several are rather huge. Google, Bing, Yahoo Search, Ask.com, AOL.com, and Baidu are among the most prominent search engines. "KidRex" is a popular search engine among schoolchildren. It generates a list of websites acceptable for school children based on the keywords entered.

URLs

The hypertext transfer protocol makes use of a specific format for Internet addresses known as the universal resource locator, or URL. In a URL, type identifies the type of server on which the file is located, address is the server address (HTTP, FTP), and path is the place within the server's file structure. The path contains a list of directories where the desired file is located. The URL is the key to accessing the web. When you enter a URL into the browser, it locates the URL's page (index.html, index.htm, index.php) and transfers it to your computer.



From the above Example

http:// or https://	Hypertext Transfer Protocol / Hypertext Transfer Protocol Secure
www	Subdomain (World Wide Web)
moecdc	Domain Name (eg. google, youtube, baidu, gorkhapatraonline)
.gov	Domain Extension (eg. .gov=Government, .com=Commercial, .edu=Education, .mil=Military, .net=Networ, .org=Organization)

.np	Country Code (eg. .np=Nepal, .us=USA, .ca=Canada, .in=India, .uk= United Kingdom)

1.7 Web Domain

A web domain is a presence on the Internet, such as a website. A web domain name is a substitute for an Internet Protocol (IP) address. For example, we can change the IP address 202.45.144.44 with a domain name like https://www.moecdc.gov.np. The Domain Name System (DNS) converts IP addresses into domain names. DNS, which stands for Domain Name System (or service or server), is an Internet service that converts domain names into IP addresses over the Internet or local networks using TCP/IP. DNS instantly translates the website name entered into the web browser address bar into the IP address of the web server that hosts the site.

Web Server

Web servers are internet-based computers that host websites and serve pages upon request. They have an IP address and possibly a domain name. When a user enters a URL, the server retrieves the specific page "index.html" and sends the HTML content to the client browser. Some of the popular web server applications are Apache web servers, Microsoft IIS (Internet Information Server), NGNIX, Google Web Server (GWS), etc.\

Web Domain

1.8 Web Hosting

Web hosting is a service that stores your website's data as well as the software, physical hardware, and network infrastructure that allows others to access your website via the internet. A web hosting service provider offers the technologies and

services required for a website to be seen on the Internet. Websites are hosted on special or main computers known as servers. When Internet users wish to see a website, they only need to type the website address or domain into their browser. Their machine will then connect to the server, and web pages will be transmitted to them through the browser. Web hosting service providers give a wide range of hosting alternatives, from pricey to economical. The cost is essentially determined by the following:

Web hosting

- The amount of storage space and computing capacity
- The degree of shares computing resources with other sites
- The additional capabilities and services offered
- The degree of control and flexibility
- The extent to manage web site

Steps to Web Hosting

Step1	Find a reliable and experienced web host company
Step2	Search via the deals offered by web hosting company
Step3	Setting up the whole domain
Step4	Connectivity to the host
Step5	Finally testing the website after upload

Exercises

Choose the correct answer from the given alternatives.

1. Which is of the following called a network of networks?

   a. Outernet b. LAN c. Internet d. MAN

2. When was started Internet?

   a. 1939 b. 1949 c. 1959 d. 1969

3. What does WWW stands for

   a. Word Wide Web b. World Well Web

   c. World Wide Web d. World Wide Wisdom

4. Which is the formal set of standards and technologies?

   a. World standards b. Web standards

   c. Website standards d. Server standards

5. Which protocol is used to transfer HTML documents in WWW.?

   a. HTTP b. TCP/IP c. FTP d. POP

6. Which is Web browser software?

   a. Altavista b. Yahoo c. Google d. MS Edge

7. Which are search engine website?

   a. Mozila b. Opera c. Google d. MS Edge

Write short answers to the following questions.

1. Define the term “Internet Technology”.
2. What is the Web Browser? What the name any four web browsers.
3. What is search engine? Give any two examples.
4. What is www.? Brifly explain.
5. How does a search engine let you search for information on the Web?

Write long answers to the following questions.

1. What are the key milestones in the history of the internet, from its inception to the present day, and how has it evolved into the global network we use today?

2. What is the significance of search engines, such as Google, Bing, and Yahoo, in helping users find relevant information on the web?
3. What are the web domains, and how do they serve as unique identifiers for websites, email addresses, and online services on the internet?
4. What is web hosting? What are the different types of hosting services?

Project works

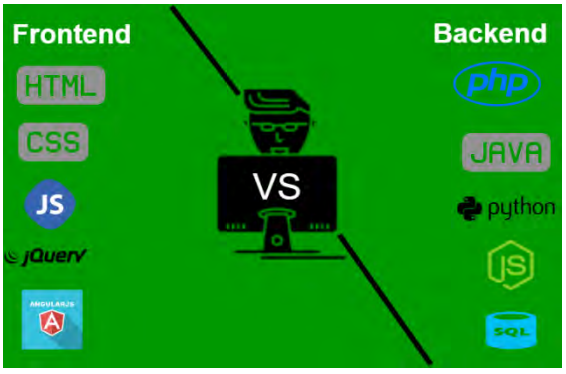
1. Prepare a list of search engines using the Internet.
2. Practice using your browser. Launch your browser and practice navigating the Web. Try using URLs based on the names of people of companies you want outlearn more about. As you visit different sites, look for hyperlinked text and graphics; click them, and see where they lead.
3. Search, search, search. Pick a topic and search the Web for information about it. Pick a keyword to use in your search, then visit three search engines and use each of them to conduct a search using your chosen keyword. Use Bing, Altavista, and Google for your searches.
4. Prepare an article on “Internet Safety”. Using the Internet, find some information about the topic and include in your article and also mention the name of the website you visited.

Unit 2 Website Design Principles

Website development is a specialized area that focuses on creating online pages. When a user navigates to a website address, our web browser (Google Chrome, Torch, IE, etc.) downloads and runs the web page or web app. It is typically downloaded once and installed as a standalone program on a computer or smartphone. Web development is an intriguing professional path. HTML, CSS, and JavaScript are the three major web building blocks. There are two main categories: front end and back end.

Front End

The front end of a web page is the presentation layer, made up of HTML, CSS, and JavaScript. HTML allows content to be placed, CSS style the page, and JavaScript makes it dynamic, allowing content to change on a mouse click and submit data. This layer is crucial in software development.



Front and back End softwares

Back End

Back end refers to the hidden realm of servers and databases, providing data storage and servers for the front end. It's often referred to as the data access layer or server in software development. It computes data and content for the front end, often using a database language like SQL.

2.1 Basic Principles of Website Development

The basic principles of website development are as follows:

a. **Simple is the Best:** Over-designed websites may divert visitors from the main purpose, so simplicity is the key in effective web page design. Keep it simple for easy navigation and navigation.

b. **Consistency:** Consistency in website design is crucial, ensuring consistent fonts,

sizes, headings, sub-headings, and button styles across all pages. Finalize fonts and colors throughout development.

c. **Typography & Readability:** Modern sans-serif fonts, Times New Roman, Arial, and Helvetica are suitable for body texts and design elements like headlines and buttons. Proper combination of typefaces enhances website design.

d. **Mobile Compatibility:** The increasing use of smartphones and tablets necessitates responsive web design for various screen sizes, with various design services transforming desktop design into adaptive and adaptive.

e. **Color Palette and Imagery:** Selecting the right color palette for your website is crucial for creating a pleasing atmosphere and attracting users, as poor combinations can lead to distraction.

f. **Easy Loading:** Optimize website loading time by reducing HTTP requests and combining code into a central CSS or JavaScript file.

g. **Easy Navigation:** Websites with easy navigation, logical page hierarchy, bread scrums, and clickable buttons can increase visitor retention.

h. **Communication:** Efficient website communication leads to increased visitor engagement. Effective strategies include using headlines, sub-headlines, cutting waffles, and bullet points to organize information effectively.

2.2 Phases of Website Development

There are several stages to website development. The website emphasizes the need of focusing on business growth. The various stages of website development are as follows:

Phase 1: Information Gathering: The initial stage of website development involves gathering information, analyzing client needs, and presenting the client's vision, ensuring understanding of the goal and purpose of the website.

Phase 2: Planning: The planning phase involves arranging tasks for website completion, developing a sitemap, deciding menus, contents, and navigational system after gathering information.

Phase 3: Designing: The creative phase of website design involves the designer putting their heart and soul into understanding client expectations, sketching them, designing logos, and selecting templates.

Phase 4: Development: The development phase, also known as the 'implementing

phase', is crucial for website design, involving the creation of databases, logic, and programming.

Phase 5: Testing & Delivery: Website testing involves preparing test cases, including content, functional, and design testing. After website testing, files are uploaded to the server, WordPress is installed and configured, essential plugins are added, and the website is officially launched, ensuring proper functionality and public viewing.

Phase 6: Maintenance: Website maintenance is limited to a specific time frame, involving content and design updates, with additional charges for extension services.

Content Management System (CMS)

A content management system (CMS) is a set of software programs for creating and managing digital material. It is easier to design and administer a website using a human-friendly interface rather than working directly with code.

Features of CMS

The fundamental functions of CMS include indexing, search and retrieval, format management, revision control, and publishing.

a. Users may easily access all material using intuitive indexing, search, and retrieval options, including qualities like publication dates, keywords, and authors.

b. Format management converts scanned paper documents and legacy electronic documents to HTML or PDF.

c. Revision tools enable content to be changed and edited after original publication. d. Publishing functionality enables individuals to use approved organizational templates.

WordPress, Joomla, and Drupal are popular content management systems (CMS). WordPress, the most popular CMS, powers around 40% of all websites because to its user-friendly interface and huge plugin ecosystem. Joomla is a more sophisticated but flexible platform designed for advanced users and developers. Drupal is known for its resilience and scalability, making it an excellent solution for large and complicated websites. Other popular CMS platforms include Shopify for e-commerce, Magento for large-scale online stores, and Squarespace and Wix for customers looking for simple, all-in-one solutions with built-in hosting and design features.

WordPress

WordPress is the most widely used content management system. It is free to download and use. It is also simple to learn, adaptable, and search engine optimized. Additionally, thousands of themes and plug-ins make it one of the most configurable platforms. If you'd like to get started with Word Press, check out our posts on:

- How to start a blog with Word Press
- How to make a website with Word Press

2.3 Importance of Websites in Contemporary World

In this highly modern, contemporary, and digital age, it appears that practically everyone requires a website to establish an online presence. More than four billion people worldwide use the internet, and more than five billion queries are conducted each day. The internet and WWW are currently tremendously popular. The user is linked to the online world, enterprises, corporations, organizations, and individuals from numerous trades and industries. They should have a website to have an online presence and be found on the Internet.

There is a wide range of trade and business, and having a website will bring various benefits based on the goals of the community. Some will benefit more from extra digital marketing chances such as online adverts, as well as the ability to gather vital facts and details about their trade or business online. According to the corporate organization, we now have control of the world and access to information. The world of the internet is great. It is still transforming how people communicate with one another, share knowledge, and live a fulfilling life.

A lot of people spend the most of their time on the internet. This could be to purchase a product, use a service, read a blog, enjoy themselves, or for any other reason. Given the amount of time individuals spend on the internet, businesses have also gone online. Business owners of all sizes must have a website and a social media presence. If you own a business and don't have a website, you could be losing a lot of prospective clients online. Knowing the value of a website is critical for exponentially growing your business.

The importance of having website are as follows:

a. Businesses should have a website.

b. It is cost-effective. Websites are cost-effective and yield positive returns.

c. People seek online for relevant local companies. Vital for digital marketing campaigns.

d. Targeted marketing can increase online sales.

e. Establishes client trust

f. Offers 24/7 accessibility for convenience

g. Boosts company visibility and audience reach

h. Encourages consumer engagement and feedback.

Exercises

Choose the correct answer from the given alternatives.

1. What are the basic principles of website development?

   a Consistency b. Easy Loading

   c. Mobile Compatibility d. All of them

2. Which is the primary stage of website development?

   a. Planning b. Website design

   c. Gathering information d. Testing

3. Creating Database, logic & actual programming is done in ……. phase.

   a. Planning b. Website design

   c. Gathering information d. Development

4. Which is arranging tasks for website completion?

   a. Planning b. Website design

   c. Gathering information d. Development

5. What does means updating the contents & design of the website mean?

   a. Planning b. Website design

   c. Gathering information d. Maintenance

6. Which are the types of website testing?

   a. Content b. Functional c. Design d. All of them

Write short answer to the following questions.

1. Define the term “Website development”.
2. What is Design phase?
3. How website development is important?
4. What is WordPress?
5. Explain development phase.
6. What are the types of website testing?

Write long answer to the following questions.

1. What are the basic principles of website development?
2. What are the different phases of website development?
3. What are the importance of website?

Project works

1. Collect information of different application program for development of website and demonstrate in your class room.
2. Make a presentation or conduct a speech competition on the topic ' Importance of Website.

HTML Basics

Unit 3

3.1 Introduction

The language used to create web pages is called Hypertext Markup Language (HTML), and hypertext is text that contains links to other writings. By clicking on a link in a hypertext document, a user can swiftly go to various material. For example, software such as dictionaries and encyclopaedias has long incorporated hypertext in definitions so that users can rapidly learn more about specific words or topics. Although hypertext is typically connected with web pages.

A markup language is a computer language that employs tags to define items within a page. It is human-readable, and markup files use standard terms rather than traditional programming terminology. There are various markup languages, but the two most used are HTML and XML.

3.2 HTML Documents

Tim Berners-Lee devised HTML at CERN, the European Particle Physics Laboratory. The World Wide Web Consortium (W3C) is the standards group in charge of the evolution of HTML. The W3C publishes a list of guidelines for the usage of HTML and other related technologies, which you should aim to follow when creating your own web pages.

3.3 Basic Structure of HTML Document

Every HTML document has a rigid structure. The HTML document is mainly divided in two sections. The entire document is enclosed within <HTML> </HTML> tags. Within these tags two distinct sections are Document **Head section** contained in <HEAD> </HEAD> and **Body section** contained in <BODY> </BODY>.

<HTML>

<HEAD>

<TITLE>…title section</TITLE>

..head section

</HEAD>

<BODY>

..body section

</BODY>

</HTML>

<HTML> This tag recognizes the document as an HTML document.

<HEAD> …</HEAD> This tag identifies the beginning and end of the Header section.

<TITLE> This tag is used to place the text in the title bar of the browser window.

<BODY> …</BODY> This tag recognizes the beginning and end of the Body section.

There are some attributes of the <BODY> tag which are used to specify background color, text color, font size, font weight and so on.

BGCOLOR	Changes the default background color to whatever color is specified with this tag.
BACKGROUND	Specifies the name of the image (.gif) file that will be used as the background of the document.
TEXT	Changes the body text color from its default value to the color specified with this attribute.

3.4 HTML Tags

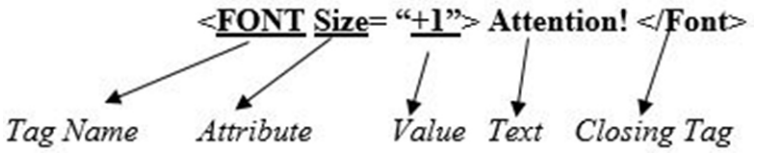
HTML Tags are instructions that are embedded directly into the text of the document. An HTML tag is a signal to a browser that it should do something other than just throw text up on the screen. By convention all HTML tags being with an open left angle bracket (<) and end with a close right angle bracket (>). HTML tags can be two types:

a. **Paired or Container Tag:** Most tags come in pairs and surround the material they affect. They work like a light switch: the first tag turns the action on, and the second turns it off. For example the <B> tag is a container tag. It is also called paired tag. In paired tags, the first tag (<B>) is often called the opening tag and the second tag (</B>) is called the closing tag. The opening tag activates the effect and the closing tag turns the effect off.

b. Singular or Empty Tag: A empty tag does not have a companion tag. Tags that are used by themselves are called empty tags. It is also called Singular or stand-alone tag. For instance, the
 tag creates a blank line and doesn't have an "off switch". <HR>, <Meta>, <Input>, <!>,<Img>, etc. are the some examples of empty tags.

3.5 HTML Attributes

Some elements may include an attribute, which is additional information contained within the start tag. For example, you can specify the FONT (size, color, or face) by inserting the corresponding attribute in the HTML code.



One tag can have more than one attribute and value.

<FONT FACE="MS Sans Serif" Color="red" Size= "+1"> Attention! </Font>

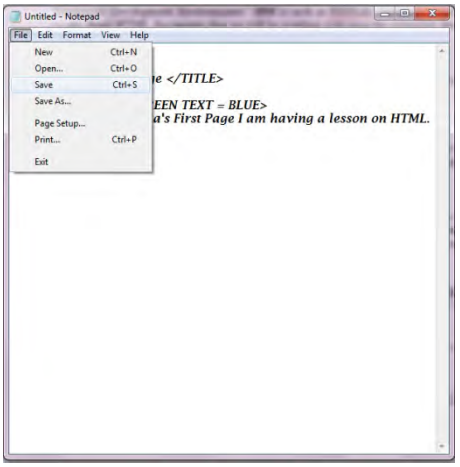
HTML is not case sensitive. <font> is equivalent to <FONT> or <FONt>.

Creating HTML Documents in Notepad:

Now let's put some of this training to use. All you need to do is write an HTML file in a text editor. Every major operating system includes at least one text editor: Mac OS has SimpleText, UNIX/Linux has vim and emacs, and Windows has Notepad. In addition, there are various HTML "Integrated Development Environments" (IDEs) such as PHPEdit, Textpad, Editplus, Notepad ++, and so on. We will be working with basic HTML texts with the extensions.html or.htm.

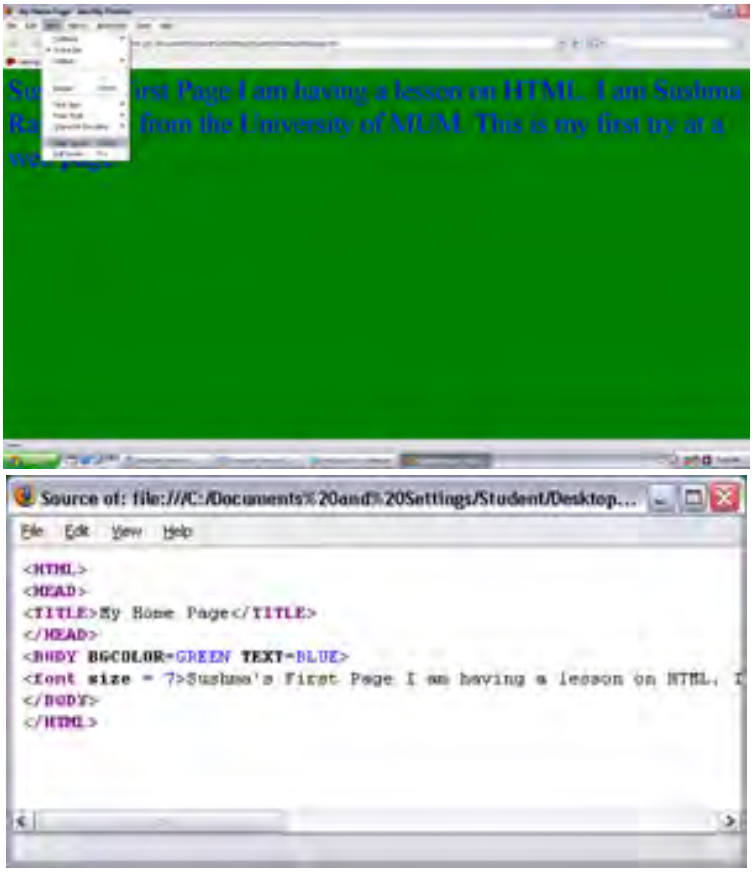
Here are the Steps to Create and Save the HTML Document in Notepad.

- In MS-Windows, Click the Start Button
- Type Notepad in the search text box and hit Enter. (Notepad window will appear)
- Type the HTML code in the Notepad window as shown in the figure below.
- Save the file from File Menu>Save.



- Notepad will display a Save as dialog box asking for a file name as shown in the figure below.
- In the File name text box, type “Firstpage.htm” or “Firstpage.html”. Type the filename and .html or .htm extension enclosed in the double quotes else the file might be saved with TXT extension.
- Click on the save button.

To view your Web page, double-click "Firstpage.htm" in your default browser (such as Internet Explorer or Mozilla Firefox). To access your page's source code, simply click on the access menu and select Source. You can now see the HTML source code as shown in the figure below.



3.6 HTML Comments

The comment tag used to insert a comment in the HTML source code which will not be displayed on the web page itself. We can use comment to explain tag/code, which

can help us when we edit the source code later. Comments in HTML always begin with <!- - and end with - ->. The browser ignores any text or tags placed inside a comment.

<!-- comment here-->

Comment tag requires exclamation (!) symbol after opening bracket, but not before the closing bracket.

Example 1

```
<HTML>
        <HEAD>
                <TITLE> An example of comment </TITLE>
        </HEAD>
<BODY>
<!—body tag includes all the tags and other text-->
The comment tag used to insert comment in the HTML source code
</BODY>
</HTML>
```

Exercise

Choose the correct answer from the given alternatives.

1. Which language is used to develop web pages?

   a. HTML b. CSS c. HTML 5 d. All of them

2. XML stands for…….

   a. Extra Markup Language b. Extensible Makeup Language

   c. Extensible Markup Language d. Extensible Markup Landmark

3. ……is the standards organization that controls the evolution of HTML.

   a. W2C b. W3C c. W4C d. W5C

4. HTML document is mainly divided in ……sections.

   a. Two b. Three c. Four d. Five

5. Which tag identifies the beginning and end of the Header section?

   a. <HTML> b. <HEAD> c. <TITLE> d. <BODY>

6. Which tag recognizes the beginning and end of the Body section?

   a. <HTML> b. <HEAD> c. <TITLE> d. <BODY>

7. Which tag used to insert a comment in the HTML?

   a. <HTML> b. <B> c. <!> d. <HR>

Write short answer to the following questions.

1. Describe about HTML document.
2. What are the basic structure of HTML?
3. Define html comments with examples.
4. What is a tag? Write its types.
5. Write any six HTML tag?
6. Explain HTML attributes.
7. What are the comment tag? Give examples.

Write long answer to the following questions.

1. Write the basic structure of HTML? Explain each elements.
2. What are the attribute of body tag?
3. What are the HTMLstructures?
4. What is tag in HTML? Describe the different types of tags in HTML.

Project works

1. Make a list of different HTML tags used to develop website and demonstrate in your class room.
2. Make a presentation on the topic ' Basic Structure of HTML document'.

Unit 4 HTML Elements

4.1 Introduction

HTML elements are the foundational components of web pages. They are like the components that combine to make a tasty cake. Each element has a specific role and represents a portion of the content on a webpage. Imagine a web page as a paper with various elements such as headings, paragraphs, graphics, and links. Each of these sections is represented by an HTML element.

4.2 Headings

HTML supports six different levels of headings. The highest-level header format is <H1> and the lowest level is <H6>. All the styles appear in BOLDFACE and the size of the heading depends on the level chosen, i.e. <H1> to <H6>

Input:	**Output:**
This is a heading one.	This is a heading one.
This is a heading two.	This is a heading two.
This is a heading three.	This is a heading three.
This is a heading four.	This is a heading four.
This is a heading five.	This is a heading five.
This is a heading six.	This is a heading six.

[Note: Here as the number increases next to <H> like (1,2,3…..) the font size actually decreases.]

4.3 Paragraphs

A blank line always separates paragraphs in textual material. The tag that provides this functionality is **<P>**. At this tag, the browser moves onto a new line skipping one line between the previous line and the new line.

ALIGN=LEFT | CENTER | RIGHT attribute can be used to set the alignment of the paragraph. Left alignment is set by default.

Input:

<P> Please forward any inquires to info@moecdc.gov.np</P>

<P ALIGN=RIGHT> DATE: </P>

4.4 Line Breaking

When text needs to start from a new line and not continue on the same line (without skipping a blank line), the **
** tag should be used. This tag simply jumps to the start of the next line.

Input:

Government of Nepal,
 Ministry of Education & Sports,

Curriculum Development Centre,
 Tel.:(977-1) 6630588, 6634119.

Output:

Government of Nepal,

Ministry of Education & Sports,

Curriculum Development Centre,

Tel.:(977-1) 6630588, 6634119

4.5 Horizontal Line

The tag <HR> draws line and a horizontal ruler. This tag draws a horizontal line across the whole page, wherever specified. The attributes of the <HR> tag are:

Attributes	Description
ALIGN	Aligns the line on the Browser screen, which is by default, aligned to the center of the screen. ALIGN=LEFT will align the line to the left of the screen ALIGN=RIGHT will align the line to the right of the screen ALIGN=CENTER will align the line to the center of the screen
SIZE	Change the size of the rule.
WIDTH	Sets the width of the rule. It can be set to a fixed number of pixels, or to a percentage of the available screen width.

Input:

Welcome to Nepal!

<HR ALIGN= “LEFT” WIDTH= “20” SIZE= “4”>

4.6 Text Formatting

Formatting tags are used to change the appearance of the text in web page. Formatting tags can be used set the font face, font color, font size, bold, italics styles of the text.

Text Styles

To format your text, you can use the following tags. <B>…</B> tag to display text in BOLDFACE style, <I>…<I> tag for ITALICS, <U>…</U> tag is uses to UNDERLINE. The superscript […] and subscript _… tags are used for placing the text above and below the specified text.

Input

this text is bold

 this text is italicize

<u>this text is underlined</u>

A²B²

H₂O

Output

this text is bold

this text is italicize

<u>this text is underlined</u>

A^2B^2

H_2O

Text Effects

Centering (Text, Image etc.)

<CENTER>…..</CENTER> tags is used to center everything found between them – text, lists, images, rules, tables or any other page element.

Input

<CENTER> Welcome to Pokhara! <CENTER>

Controlling Font Size and Color:

All text specified within the tags<FONT> and </FONT> will appear in the font, size and color as specified as attributes of the tag <FONT>. The attributes are:

FACE	Sets the font to the specified font name.
SIZE	Sets the size of the text. Size can take values between 1 and 7. The default sized used is 3.
COLOR	Sets the color of the text. COLOR can be set to an English language color name or to a hexadecimal triplet.

Input

<FONT face= “Comic Sans MS” Size=6 Color=Red> Welcome to Nepal VisitNepal 2020! </Font>

If you would like to specify a certain text or background color, you can do so by using color codes.

RGB color codes are represented as hexadecimal values. The RGB color codes contain three sets of numbers representing the amount of Red, Green and Blue contained in a color. These codes must be used within your HTML to specify your selected colors.

Here are a few of the basic color codes:

Black	#000000	Silver	#C0C0C0
White	#FFFFFF	Lime	#00FF00
Red	#FF0000	Gray	#808080
Green	#00CC00	Maroon	#800000
Blue	#0000FF	Purple	#800080
Yellow	#FFFF00	Aqua	#00FFFF

Input

<font color=”#FF0000”>RED </FONT>

<font color=”#00CC00”>Green </FONT>

<font color=”#0000FF”>Blue </FONT>

4.7 Lists

In HTML, lists are used to group similar elements.

TYPES OF LISTS

Unordered List (Bullets)

An unordered list starts with the tag <UL> and ends with </UL>. Each list of items starts with the tag <LI>. The attributes that can be specified with <LI> are

TYPE	**:**	Specifies the type of bullet. **TYPE=FILLROUND** will give a solid round black bullet **TYPE=SQUARE** will give a solid square black bullet

Example 2

Some of these products include:

```
<UL TYPE =FILLROUND>
        <LI> Pen drive </LI>
        <LI> Hard Disk </LI>
        <LI> Monitors </LI>
</UL>
```

Output

Some of these products include:

- Pen drive
- Hard Disks
- Monitors

Ordered Lists (Numbering)

An ordered lists start with the tag <OL> and end with </OL>. Each list items start with the tag <LI>. The attributes that can be specified with <LI> are:

TYPE	**:**	Controls the numbering scheme to be used. TYPE = “1” will give counting numbers (1,2, ……) TYPE = “A” will give Uppercase letters (A,B,……) TYPE = “a” will give Lowercase letters (a,b,……..) TYPE = “I” will give Uppercase Roman Numerals (I, II, ……) TYPE = “i” will give Lowercase Roman Numerals (i, ii, iii, …….)

START	**:**	Alters the numbering sequence. Can be set to any numeric value.
VALUES	**:**	Changes the numbering sequence in the middle of an ordered list. It is to be specified with the <LI> tag.

Example

Some of these products include:

```
<OL TYPE = "1" START=3>
        <LI> Pen drive
        <LI> Hard Disks
        <LI> Monitors
</OL>
```

Output

Some of these products include:

3. Pen drive
4. Hard Disks
5. Monitors

Definition Lists

Definition list values appear within tags <DL> and </DL>. Definition lists consist of two parts:

Definition term	**:**	appears after the tag <DT>
Definition description	**:**	appears after the tag <DD>

Example

```
<DL>
        <DT> Keyboard
                <DD> An input device
        <DT> Printer
                <DD> An output device
```

</DL>

Output:

Keyboard

An input device

Printer

An output device

4.8 Tables and Frames

Tables

A table is a two-dimensional matrix, consisting of rows and columns. Tables are intended for displaying data in rows and columns on a web page. All table-related tags are included between the <TABLE> </TABLE> tag. Each row of a table is described between the <TR> </TR> tag. Each column of a table is described between the <TD> </TD> tag.

Table rows can be of two types:

Sunday	Monday	Tuesday	Wednesday
5	6	7	8
12	13	14	15

Header rows

A table header row is defined using **<TH> </TH>** tag. The content of a table header tow is automatically centred and appears in boldface.

Data rows (Individual data cells placed in the horizontal plane creates a data row)

There could be a single data cell (i.e. a single column table) or multiple data cells (i.e. a multi-column table)

Data cells hold data that must be displayed in the table. A data row is defined using <TR> </TR> tags. Text matter displayed in a data row is left-justified by default. Any special formatting like boldface or italics is done by including appropriate formatting tags inside the <TR> </TR> tags. An image can also be displayed in a data cell.

The attributes that can be included in the <TABLE> tag are:

ALIGN	**:**	Horizontal alignment is controlled by the ALIGN attribute. It can be set to LEFT, CENTER, or RIGHT.
VALIGN	**:**	Controls the vertical alignment of cell contents. It accepts the values TOP, MIDDLE or BOTTOM.
WIDTH	**:**	Sets the WIDTH to a specific number of pixels or to a percentage of the available screen width. If width is not specified, the data cell is adjusted based on the cell data value.
BORDER	**:**	Controls the border to be placed around the table. The border thickness is specified in pixels.
CELLPADDING	**:**	This attribute controls the distance between the data in a cell and the boundaries of the cell.
CELLSPACING	**:**	Controls the spacing between adjacent cells.
COLSPAN	**:**	The COLSPAN attribute inside a <TH> or <TD> tag instructs the browser to make the cell defined by the tag to take up more than one column. The COLSPAN attribute can be set equal to the number of columns the cell is to occupy. This attribute is useful when one row of the table needs to be a certain number of columns wide.
ROWSPAN	**:**	The ROWSPAN attribute works in the same way as the COLSPAN attribute except that it allows a cell to take up more than one row. The attribute can be set by giving a numeric value. For example ROWSPAN = 3

Caption Tag

Often tables need to be given a heading, which gives the reader a context for the information in the tables. Table Headings are called Captions. Captions can be provided to a table by using the <CAPTION> </CAPTION> tags. The paired tag appears within the <TABLE> </TABLE> tags. The table caption can be made to appear above or below the table structure with the help of the attribute ALIGN, as explained below in Table.

ALING : It controls placing of the caption with respect to the table.

ALIGN = BOTTOM will place the caption immediately below the table.

ALIGN=TOP will place the caption immediately above the table.

Bypassing a rows <TR> tag the VALIGN and ALIGN attributes, vertical or the horizontal alignment can be made identical for every cell in a given row.

Bypassing the <TH> and/or <TD> tags, VALIGN or ALIGN attributes, vertical or horizontal alignments in both header and data cells can be done. Any alignment specified at the cell level overrides any default alignments and any alignments specified in a <TR> tag.

Note:

- Alignments specified in <TD> or <TH> apply only to the cell being defined.
- Alignments specified in a <TR> tag apply to all cells in a row, unless overridden by an alignment specification in a <TD> or <TH> tag.

Using the WIDTH and BORDER attribute

Example

```
<HTML>
        <HEAD>
        <TITLE> Table Attributes </TITLE>
</HEAD>
<BODY BGCOLOR=LIGHTGREY>
        <B> Specifying the BORDER and WIDTH of the Table! </B>
        <BR> <BR> <BR> <BR>
<CENTER>
<TABLE BORDER=5 WIDTH=50%>
<CAPTION ALING=bottom>
<B> Personal Information </B>
</CAPTION>
<TR>
        <TH> NAME </TH>
        <TH> AGE </TH>
```

```
</TR>
        <TR ALIGN = CENTER>
<TD> Suman Raut </TD>
        <TD> 16 </TD>
</TR>
        <TR ALIGN=CENTER>
        <TD> Pooja Niraula </TD>
        <TD>32 </TD>
        </TR>
</TABLE>
</CENTER>
</BODY>
</HTML>
```

Note:

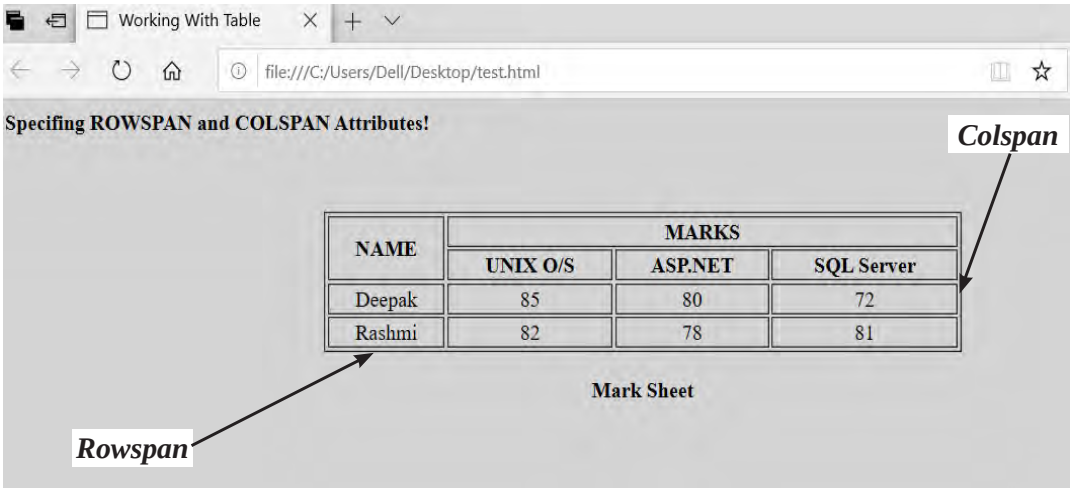
- If the WIDTH attribute is associated with the <TH> tag then the width of an individual column can be adjusted.

Using the COLSPAN AND ROWSPAN attributes.

Example

```
<HTML>
  <HEAD>
          <TITLE> Working With Table </TITLE>
        </HEAD>
        <BODY BGCOLOR=LIGHTGREY>
            <B> Specifying ROWSPAN and COLSPAN Attributes! </B>
            <BR> <BR> <BR> <BR>
<CENTER>
   <TABLE BORDER=1 WIDTH=50% ALIGN=CENTER>
   <TR>
      <TH ROWSPAN = 2> NAME
      <TH COLSPAN = 3> MARKS
   </TR>
```

```
<TR>
    <TH> UNIX O/S
    <TH> ASP.NET
    <TH> SQL Server
</TR>
<TR ALIGN=CENTER>
    <TD>Deepak
    <TD>85
    <TD>80
    <TD>72
</TR>
<TR ALIGN=CENTER>
    <TD> Rashmi
    <TD> 82
    <TD> 78
    <TD> 81
</TR>
<CAPTION ALIGN = bottom> <B> <BR> Mark Sheet </B> </CAPTION>
            </TABLE>
        </CENTER>
    </BODY>
</HTML>
```



Frames

HTML frames divide your browser window into multiple portions, each of which can load a different HTML content. A frameset is a collection of frames within a browser window. The window is divided into frames in the same way that tables are organized: by rows and columns.

Creating Frames

To use frames on a page we use <frameset> tag instead of <body> tag. The <frameset> tag defines how to divide the window into frames. The rows attribute of <frameset> tag defines horizontal frames and cols attribute defines vertical frames. Each frame is indicated by <frame> tag and it defines which HTML document shall open into the frame.

```
<html>
<head>
<title>HTML Frames</title>
</head>
<frameset rows="10%,80%,10%">
  <frame name="top" src="/html/top_frame.htm" />
  <frame name="main" src="/html/main_frame.htm" />
  <frame name="bottom" src="/html/bottom_frame.htm" />
  <noframes>
<body>
  Your browser does not support frames.
</body>
  </noframes>
</frameset>
</html>
```

4.9 Hyperlinks

HTML supports linking to other HTML documents as well as images. When you click on a part of text or an image on one online page, you will be taken to another web page or to a specific point on the same page. The text or image that creates such linkage is known as hypertext, hyperlink, or hotspot. The browser distinguishes between hyperlinks and regular text. Every Hyperlink,

- Appears blue in color
- The hyperlink text/images are underlined
- When the mouse cursor is placed over it, the standard arrow-shaped mouse cursor changes to the shape of a pointing hand

Links are created in a web page by using the <A> </A> tag. Anything written between the <A> </A> tag becomes a hyperlink/hotspot. When you click on the links, you will be taken to a different website or image. The document to be navigated to must be supplied. By using the HREF attribute of the <A> tag the next navigable web page or images can be specified.

<A HREF = "filename.htm">

Hyperlinks can be of two types

- Links to an external document
- Links (jumps) to a specific place within the same document

External Document References

Example <A HREF = "details.htm"> Visit my Home page </A>

INPUT

```
<HTML>
<HEAD>
    <TITTLE> USING LINKS TO CONNECT TO EXTERNAL SITES </TITLE>
   <BODY BGCOLOR= "GREEN">
<H2 ALIGN= "CENTER">VISIT THE SITES TO KNOW ABOUT NEPAL</H2>
<P>
<OL>
        <LI> <A HREF= "http:// WWW.welcomenepal.com"> WWW.welcomenepal.com </A>
        <LI> <A HREF= "http://WWW.explorenepal.com"> WWW.explorenepal.com </A>
        <LI> <A HREF= "http:// WWW.lonelyplanet.com"> WWW.lonelyplanet.com </A>
        <LI> <A HREF= "http:// mofa.gov.np"> mofa.gov.np </A>
```

</OL>

</BODY>

</HTML>

Intra-Page Links or Bookmarks

Link created in a page can be opened at a different location on the same page. Such links are called intra page links or bookmarks. It is also possible to jump to a particular location on another web page using bookmarks.

Using Named Anchors

To use the intra-page links named Anchors are used.

For instance, <A NAME= “top”> </A>

This is a different use of the <A> anchor tag; it is necessary to give a name at the specific point on the page where the tag occurs. The </A> tag must be included, but no text is required between <A> and </A>.

4.10 Multimedia (Image, Audio, Video)

The <MARQUEE> tag is useful for creating scrolling effect.

Attribute	Description	Values
Behavior	Scrolling behavior	alternate, slide, scroll
Direction	Scrolling direction	left, right, up, down
Bgcolor	Background color	color name or value
scrolldelay	Delay in scrolling text	A number in milliseconds
Height	Height of scroll area	A number is pixels
width	Width of scroll area	A number in pixels

Input

<MARQUEE> Computer Engineering !!!</MARQUEE>

<marquee behavior=”scroll” direction=”down” bgcolor=”#CCFF00” scrolldelay=”1000”> Running Text! </marquee>

ADDING IMAGE TO HTML DOCUMENTS

HTML, in addition to text, allows you to include static and/or animated images in your

HTML pages. HMTL mostly uses two picture file formats:.gif and.jpg. Images can be developed to meet the needs of a web page and stored in these file types. Using programs like Gif Constructor and Adobe Photoshop.

Once an image is ready and stored in above-mentioned format, it can be inserted into a web page using <IMG> tag, which takes the name of the image file (filename.gif, filename.jpg or filename.jpeg) as the value of SRC attribute.

Inserting Images <IMG> Tag

The <IMG> tag takes the following attributes:

Attribute	Value
SRC	Location and name of the image file
ALIGN	Controls alignment of the text following the image **ALIGN = TOP** indicates the text after the image to be written at the tip, next to the image. **ALIGN = MIDDLE** indicates the text after the image to be written at the middle, next to the image. **ALIGN = BOTTOM** indicates the text after the image to be written at the bottom, next to the image. Controls the image's alignment with respect to the VDU screen. **ALIGN = LEFT** indicates the image is aligned to the left with respect to the screen. **ALIGN = CENTER** indicates the image is aligned to the center with respect to the screen. **ALIGN = RIGHT** indicates the image is aligned to the right with respect to the screen.
BORDER	Specifies the size of the border to place around the image.
WIDTH	Specifies the width of the image in pixels.
HEIGHT	Specifies the height of the image in pixels.
HSPACE	Indicates the amount of space to the left and right of the image.
VSPACE	Indicates the amount of space to the top and bottom of the image.
ALT	Indicates the text to be displayed in case the Browser is unable to display the image specified in the SRC attribute

Example

<IMG WIDTH = 447 HEIGHT = 57 BORDER =0 HSPACE = 0 SRC = “IMAGE1.GIF” ALIGN=CENTER>

4.11 Forms

HTML forms are basic for web Client-Server capabilities. Forms are the client-side GUI applications. To create a form, use the <FORM> tag. Inside the opening and closing FORM tags individual form elements plus any other HTML content are used to create a layout for the form (paragraphs, heading, tables, and on). As many forms on a page can be used but you can’t include a <FORM> tag inside another <FORM> tag.

The opening tag of the FORM tag includes attributes namely Name, Method and Action. Name is used for defining the name of the form. The Method attribute can be either GET or POST, which determines how the form data is being sent to the server. The Action attribute is a pointer to the script that process the form on the server-side.

Different types of form elements included in the <FORM> are text boxes, radio buttons, checkboxes, drop-down boxes, multiline, scrollable text areas and password boxes.

Each of the form element will be placed inside a <FORM>… </FORM> tag.

Elements of HTML Form

a. Text fields

Text fields enable the client to type text into a single-line field. To create a text field, you can either use TYPE=“TEXT” in the <INPUT> tag or leave off the TYPE specification altogether. The default TYPE for the <INPUT> tag is “text”.

NAME indicates the name of this field as passed to the script processing the form.

SIZE indicates the length of the text-entry field, in characters; the field is 20 characters by default.

Example

```
<form >
    User Name : <Input type= “text” Name= “userName” >
<br><br>
```

Password :   <Input type= “password” Name= “passwd” >

</form>

Text box

Note: <input type = “password”> password text fields are identical to ordinary text fields, except that all the characters types are echoed back in the browser as asterisks or bullets. Here   is used for one space

b. Text Area

Text areas are input fields that contain many lines of text, making them extremely useful for forms that require extensive input. For example, if you wanted to create a form that enables readers to compose electronic mail, you might use a text area for the body of the message.

To include a text are element in a form, use the <Textarea> … </Textarea>.

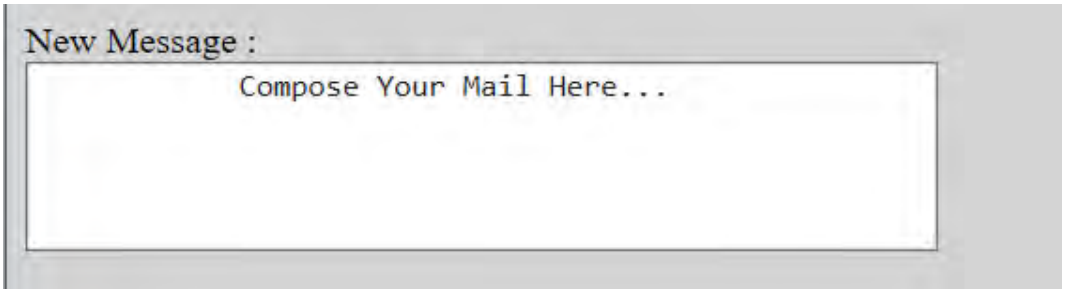
<**TextArea**> includes three attributes:

NAME	The name to be sent when the form is submitted
ROWS	The height of the text area element, in rows of text
COLS	The width of the text area element in columns (characters)

<Textarea Name= “theBody” Rows= “7” Cols = “30”> Enter your message here </textarea>

Example

```
<form>
        New Message : <br>
<textarea rows= “5” cols= “50” name= “details”>
          Compose Your Mail Here...
        </textarea>
</form>
```



c. Radio Button

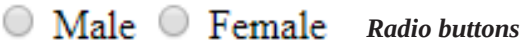
Radio buttons indicate a list of items, of which only one can be chosen. If one radio button in a list is selected, all the other radio buttons in the same list are deselected.

Example

<INPUT TYPE= “RADIO” NAME = “r1” VALUE = “MALE”>Male

<INPUT TYPE= “RADIO” NAME = “r1” VLAUE = “FEMALE”>Female

textarea



Radio buttons

d. Checkbox

Checks boxes make it possible to choose multiple items in a list. Each checkbox can be either checked or unchecked.

Example

<Input Type= “checkbox” Name= “Red”> Red

<Input Type= “checkbox” Name = “Green”> Green

<Input Type = “checkbox” Name = “Blue”> Blue

e. Select box

Select has the ability to create pull-down menus and scrolling lists. Selection enables to select one or more items for a menu or a scrolling list. They’re similar in functionality to radio buttons or checkboxes, but they’re display in a different way on-screen. The <select> tag and individual options within the selection indicate by the **<option>** tag are used for creating such elements.

Example

```
<HTML>
  <HEAD>
      <TITLE> Working With Select box </TITLE>
  </HEAD>
<BODY BGCOLOR=”#aabbcc”>
<form>
<P> <b> Select a Country:</b>
        <Select Name= “Scount”>
                <option value= “Nepal” selected> Nepal </option>
                <option value=”Afghanistan”> Afghanistan </option>
                <option value= “Bangladesh”> Bangladesh </option>
                <option value= “Bhutan”> Bhutan </option>
                <option value= “India”> India </option>
                <option value= “Maldives”> Maldives </option>
                <option value= “Pakistan”> Pakistan </option>
                <option value= “Srilanka”> Srilanka </option>
        </select> </P>
<b> Select a Capital:</b>
        <Select Name= “Capital”>
                <option value= “Kathmandu” selected> Kathmandu </option>
                <Option Value= “Kabul”> Kabul
                <Option Value= “Dhaka”> Dhaka
                <Option Value = “Thimphu”> Thimphu
                <Option Value = “NewDelhi”> New Delhi
                <Option Value = “Male”> Male
                <option Value = “Islamabad”> Islamabad
                <option Valu = “Colombo”> Colombo
        </Select> </P>
</form>
```

</HTML>

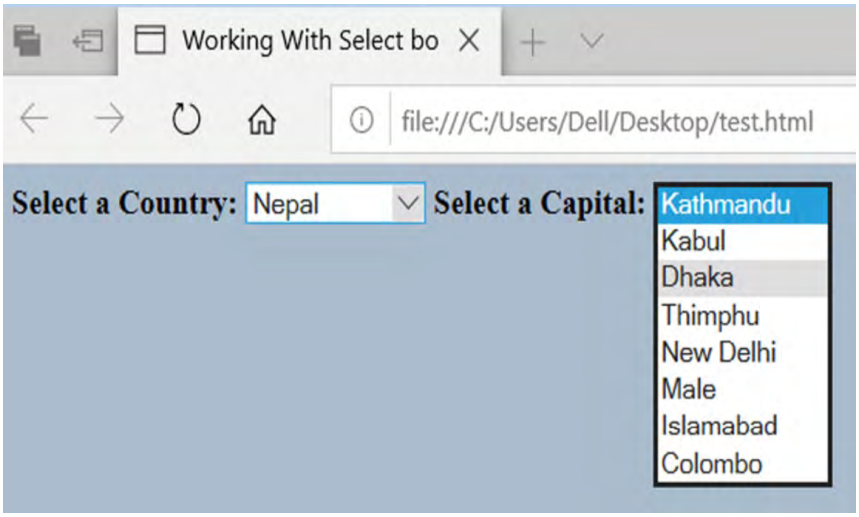


Fig. Select box

f. Submit Button

Submit buttons tell the browsers to send the form data to the server. You should include at least one submit button on every form.

<input type = “Submit” value= “Submit”>

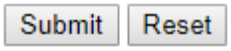
g. Reset Button

Reset button is used for renewing the elements of the form. It allows the client to fill new entries by clicking on the reset button.

<input type = “reset” value = “reset”>

Example

```
<form>
        <input type= “submit” name= “submit” value= “Submit”>
        <input type= “reset” name= “reset” value= “Reset”>
</form>
```



Exercise

Choose the correct answer from the given alternatives.

1. What does HTML stand for?

   a. Hyperlinks and Text Markup Language

   b. Hypertext Markup Language

   c. Home Tool Markup Language

   d. All of above

2. Choose the correct HTML tag for the largest heading.

   a. <h1> b. <head> c. <heading> iv. <h6>

3. What is the correct HTML tag for inserting a line break?

   a.
 b. <lb> c. <break>

4. What is the correct HTML for adding a background color?

   a. <body bgcolor="yellow">

   b. <background> yellow </background>

   c. <body color="yellow">

5. Choose the correct HTML tag to make a text bold.

   a. <bb> b. <bold> c. <b> d. <bld>

6. Choose the correct HTML tag to make a text italic.

   a. <i> b. <ii> c. <italics>

7. What is the correct HTML for making a hyperlink?

   a. W3Schools.com

   b. W3Schools

   c. W3Schools.com

   d. <a>http://WWW.w3schools.com

8. How can you make an e-mail link?

   a. <mail> xxx@yyy </mail> b. <a href="mailto:xxx@yyy">

   c. <mail href="xxx@yyy"> d. <a href="xxx@yyy">

9. Write the purposes and syntaxes of the following HTML tags.

a. <P>	b. <SUB>	c. <HR>	d. <A>
e. <TABLE>	f. <IMG>	g. <CENTER>	h. <STRONG>
i. <IMG>	j. <U>	k. <UL>	l. <B>

Write short answers to the following questions.

1. Name the commonly used web browsers for viewing the HTML pages.
2. What do you mean by attributes?
3. Differentiate between paired tag and singular tag with examples.
4. How is the FONT tag used? What are its attributes?
5. Name the tags for breaking paragraph and line.
6. What are the different types of lists used in HTML?
7. What is hyperlink?
8. What is the use of intra page links? Illustrate with examples.
9. Is it possible to give a background to the web page? If yes, then how?
10. What are the main tags used in <TABLE> tag?
11. What are the different types of input types?

Write long answers to the following questions.

1. Which HTML tag is used to display the data in the tabular form?
2. What are some common lists that are used when designing a page?
3. What is the difference between HTML elements and tags?
4. Describe table tag with its attributes. Write an HTML code to form a table to show the below values in a tabular form with heading as Roll No., Student name and Subject Name.

Practical Works

1. Design a web page having the following specifications.

   The italic tag, The center tag, The Paragraph tag, The Break tag, The font tag and its attributes

2. Design a web page using the image files according to the following specifications.
   a. Use a Border for Image
   b. Resize the Width and Height of the image to 100 pixels each.

HTML5 Basics

Unit 5

5.1 Introduction

HTML5 is the most recent version of HTML, allowing elements to be loaded more easily through a single interface. Previous versions of web pages may not work properly in other browsers because those HTML versions require extra plugins. HTML5 resolves this compatibility issue. For example, to play video, the browser does not require a Flash plugin; instead, the HTML5 element runs on its own.

Advantages of HTML5

a. Writing, maintaining and redesigning web documents in HTML5 is easier.

b. HTML5 provide a better way for Search Engine Optimization (SEO).

c. The web pages developed in HTML5 are easily accessible to visually impaired users.

d. It provides better browsing of web pages in mobile devices also.

e. Web pages in HTML5 are better for the users with slower internet connectivity.

f. The web pages written in HTML5 have few chances to have design failure.

g. Adding media and making web pages more interactive in HTML5 is easier.

5.2 New Features in HTML5

Some of the features of HTML5 are given below:

a. Introduction to new semantic (meaningful) markup.

b. Separation of design part and content part in document.

c. Enhancement of accessibility of the web pages to all types of users. (Including blind and visually impaired)

d. Encouragement of responsive design practices.

e. Reduction of overlap between HTML, CSS and JavaScript.

f. Introduction of new media element to replace plugins such as Flash or Java

5.2.1 HTML5 Semantic Elements

In previous version of HTML, the tag was used to contain all the structural

elements like headers, navigations, menus, main content, footer and others. But in HTML5, new semantic elements are introduced to define the basic structure of the document. These semantic elements clearly describe its purpose and meaning to both the browser and the developer. List of new semantic elements added in HTML5 are <article>, <aside>, <details> and <summary>, <figure> and <figcaption>, <nav>, <header>, <footer>, <main>, <section>, <mark>, <time>, etc.

<article>

An <article> element represents the self-contained composition in a document, application, page, or site which can be reused by distributing independently. The element can be used in pages like blogs, forums, magazines newspaper articles, or any independent item of the content.

Example

```
<!DOCTYPE html>
<html>
<head>
        <title>Article Tag</title>
</head>
<body>
        <article>
                <h1>About Article Tag</h1>
                <p>An article tag is the semantic tag. </p>
        </article>
</body>
</html>
```

<aside>

An <aside> element is used to define the additional information of the content surrounding or aside of it. The content inside the <aside> element should be relatable to the sorrounding

content. It place the content in the sidebar, i.e. aside to the existing relatable content.

Example

```
<!DOCTYPE html>
<html>
<head>
        <title>Aside Tag</title>
</head>
<body>
<p>This is about the Semantic Element<p>
        <aside>
                <h4>Aside Tags</h4>
        <p>This is about the aside tag.
                </p>
        </aside>
</body>
</html>
```

<nav>

The <nav> element is used to add navigation links in the web pages. Only major navigation links, like the main menus of the pages, are included in these tags instead of all kind of links. For Example the <nav> tag cannot be placed in the <footer> tag for defining links in the footer of the website.

Example

```
<!DOCTYPE html>
<html>
<body>
<nav>
        <a href="/learn-html.html">HTML</a> | <a href="/learn-css.
html">CSS</a> | <a href="/learn-javascript.html">JavaScript</a> | <a href="/
learn-php.html">PHP</a> |
        </nav>
```

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

<header>

The <header> element is used to add the content like logos, search bars, menus or navigation links at the top of the page.

Example

```
<!DOCTYPE html>
<html>
        <body>
         <header>
        <nav>
    <ul style=”padding:0;”>
     <li>Home</li>
     <li>About us</li>
    </ul>
   </nav>
   <h1>Welcome to our page</h1>
   <hr>
        </header>
        </body>
</html>
```

<footer>

The <footer> element is used to add the content in the footer of the pages. The footer contents may be the useful links, copyright information, contact details etc. of the site. It is also written within the <body> tag like other.

Example

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

```
        <footer>
            <a href="https://facebook.com">Facebook</a>|
            <a href="https://google.com">Google</a>|
            <p>@CDC, All Right Reserved, 2081</p>
        </footer>
    </body>
    </html>
```

<main>

The <main> element is used to add the main central content of the web pages. There shouldn't be more than one <main> element in each page. The content inside <main> element should be unique, i.e., shouldn't be repeated in sidebars, footer etc.

Example

```
    <!DOCTYPE html>
    <html>
        <head><title>main tags</title></head>
        <body>
            <h3> About Main Tag</h3>
        <main> This is the main dominant content of the page.</main>
        </body>
    </html>
```

<section>

The <section> tag is used to add the groups of logically connected content in one section. It is similar to <article> tag but it is thematic grouping of content whereas <article> is intended to be independently distributable.

Example

```
    <!DOCTYPE html>
    <html>
        <head><title>Section Tags</title></head>
        <body>
```

```
                    <h3> About Section Tag</h3>
            <section>
  <p>Top Stories</p>
  <section>
            <p>News</p>
            <article>Story 1</article>
            <article>Story 2</article>
            <article>Story 3</article>
            </section>
<section>
   <p>Sport</p>
   <article>Story 1</article>
   <article>Story 2</article>
   <article>Story 3</article>
  </section>
</section>
            </body>
</html>
```

<mark>

The <mark> element is used to highlight the desired text.

Example

```
<!DOCTYPE html>
<html>
            <head><title>Mark Tags</title></head>
            <body>
                    <h3> About mark Tag</h3>
            <p>This resource is developed by <mark>CDC</mark> for educational purpose</p>
            </body>
</html>
```

<time>

The <time> element defines a human-readable time on a 24-hour clock or a precise date in the Gregorian calendar.

Example

```
<!DOCTYPE html>
<html>
        <head><title>Time Tags</title></head>
        <body>
                <h3> About Time Tag</h3>
<p>The game will be held on<time datetime="2018-09-28 19:00">September 28</time>.</p>
        <p>It will start at <time>19:00</time></p>
        </body>
</html>
```

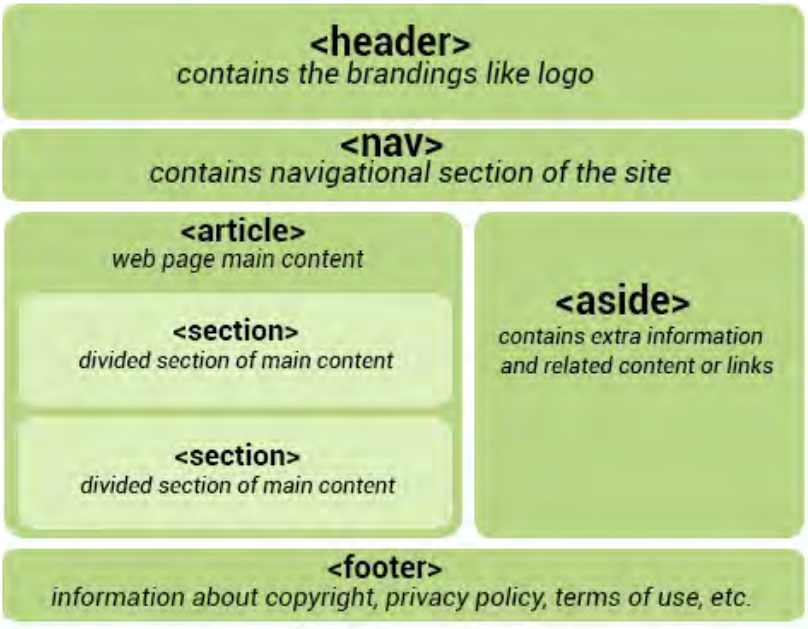


Fig: Semantic Structure of HTML5 documents.

5.2.2 HTML5 Audio and Video

Earlier, native web technologies such as HTML didn't allow embedding video and audio on the Web. Plugin-based technologies became popular for handling such

content, but they had many problems, including not working well with HTML/CSS features, security and accessibility issues. Later, HTML5 specification introduced such features with the <video> and <audio> elements.

HTML5 Audio

The <audio> tag is used to add audio files in the webpages. The formats may not be supported by all the browser, however a WAV file is common sound that is supported by most browser versions.

Syntax

<audio src="URL" controls> </audio>

In the code, the src attribute refers to the URL of the audio file, and the controls attribute adds a control panel (play button, scroll bar, volume button etc).

Syntax for More than one Audio Format

When defining different file formats, we recommend to define MIME-type for each file, in order to let the browser localize supported file. The MIME-type is defined by the help of the type attribute.

```
<audio controls="controls" autoplay   loop>
    <source src="URL1" type="audio/mp3" />
    <source src="URL2" type="audio/wma" />
    <source src="URL3" type="audio/x-wav" />
</audio>
```

The most popular audio formats are the following ones:

a. MP3 – the most popular audio format, which uses lossy compression and permits to reduce the file size. Despite the popularity among the users, TV companies and radio stations use more modern codecs ISO-MPEG, like AAC or MPEG-H.

b. AAC (Advanced Audio Codec) — closed codec, MP3 analog, but comparing to the latter, it provides higher quality with the same or stronger compression degree.

c. Ogg Vorbis— free format with an open code, supported in Firefox, Opera and Chrome. Provides good quality sound, but not sufficiently supported by device players.

Because not all browsers support all audio formats, the audio file is encoded/ decoded using an audio codec (a digital electronic device or computer-based

software application that aids in the compression and decompression of a digital audio data).

Example

```
<!DOCTYPE html>
    <html>
     <head>
      <title>HTML5 audio</title>
     </head>
     <body>
      <audio controls>
       <source src=”audio.ogg” type=”audio/ogg”>
       <source src=”audio.mp3” type=”audio/mpeg”>
      </audio>
      <p>Click the play button</p>
     </body>
    </html>
```

HTML5 Video

The <video> tag is used to add videos on the webpages.

Syntax:

```
<video src=”example.webm” controls></video>
```

The src attribute indicates the URL of the file, and the controls attribute is used to display control elements. Each browser supports particular codec, that’s why, in order to provide video playback in all browsers, the video file must be placed in a few formats. Like in the case of audio files, all formats of video files are included in the <source> element, starting with the most preferred one. Every video file must have its MIME-type, which is defined by the type attribute. The popular video formats are MP4/MPEG-4, OGG and WebM +.

Example

```
<!DOCTYPE html>
<html>
  <head>
    <title>About video element</title>
```

```
</head>
<body>
  <video controls>
    <source src=”video.ogv” type=”video/ogg”>
    <source src=”video.mp4” type=”video/mp4”>
  </video>
  <p>Some information about video</p>
</body>
</html>
```

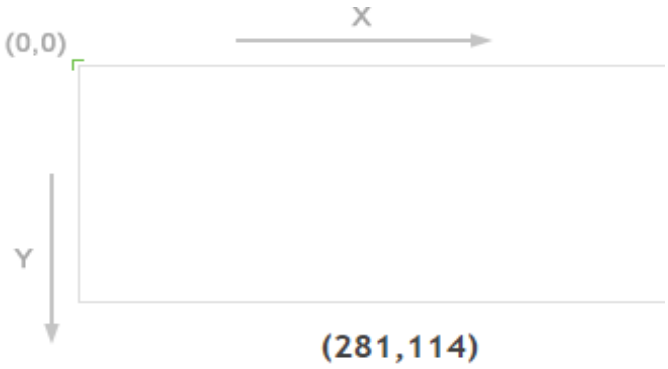
Audio and Video Attributes

Attribute	Description
autoplay	Allows the audio/video to automatically start playing while the rest of the page is loading.
autobuffer	Allows the audio/video to automatically begin buffering.
controls	Allows controlling the audio/video playback, including volume, pause/ resume playback.
loop	Allows the audio/video to play again whenever it finishes.
muted	Makes the media play with the turned off sound by default.
preload	Specifies buffering large files. It can have one of these values:
	“none”, which does not buffer the file.
	“auto”, which buffers the media file.
	“metadata”, which buffers only the metadata for the file.
src	The URL of the audio/video to embed. It is optional.
poster	This is a URL of an image displayed before the video is played.
width	Specifies the display area width of the video, in CSS pixels.
height	Specifies the display area height of the video, in CSS pixels.

5.2.3 HTML5 Canvas

The canvas is a two-dimensional rectangular area. The coordinates of the top-left corner of the canvas are (0, 0) which is known as origin, and the coordinates of the

bottom-right corner are (canvas width, canvas height). Here's a simple demonstration of canvas default coordinate system.



The <canvas> element is used to draw graphics on the webpage using JavaScript. It can be used to draw graphs, make photo compositions or do animations. By default the <canvas> element has 300px of width and 150px of height without any border and content.

Example

```
<!DOCTYPE html>
<head>
<title>Drawing on Canvas</title>
<script>
   window.onload = function() {
      var canvas = document.getElementById("myCanvas");
      var context = canvas.getContext("2d");
     context.moveTo(50, 150);
      context.lineTo(250, 50);
      context.stroke();
     };
</script>
</head>
<body>
   <canvas id="myCanvas" width="300" height="200" border="1px"></canvas>
</body>
</html>
```

5.2.4 HTML5 SVG

The <svg> element allows to embed SVG graphics in webpage. SVG stands for Scalable Vector Graphics which defines graphics for web. SVG has several methods for drawing paths, boxes, circles, text, and graphic images.

Syntax:

```
<svg xmlns = “http://www.w3.org/2000/svg”>
  ...
</svg>
```

Example

```
<!DOCTYPE html>
<html>
  <head>
    <title>SVG</title>
    <meta charset = “utf-8” />
  </head>

  <body>
    <h2 align = “center”>HTML5 SVG Circle</h2>

    <svg id = “svgelem” height = “200” xmlns = “http://www.w3.org/2000/svg”>
      <circle id = “redcircle” cx = “50” cy = “50” r = “50” fill = “red” />
    </svg>
  </body>
</html>
```

5.2.5 HTML5 Drag and Drop

Drag and Drop is the concept to make user more comfortable with moving, deleting, reordering items with the help of mouse clicks. In previous version of HTML user has to take help of JavaScript programming to achieve the required drag and drop option. But in HTML5 there is Drag and Drop API which makes easy to code and is supported by all major browsers. There are number of events which are fired during

various stages of the drag and drop operation. In this level we will just get to know about the Drag and Drop events with description.

S.N.	Events & Description
1	**dragstart:** Fires when the user starts dragging of the object.
2	**dragenter:** Fired when the mouse is first moved over the target element while a drag is occurring. A listener for this event should indicate whether a drop is allowed over this location. If there are no listeners, or the listeners perform no operations, then a drop is not allowed by default.
3	**dragover:** This event is fired as the mouse is moved over an element when a drag is occurring. Much of the time, the operation that occurs during a listener will be the same as the dragenter event.
4	**dragleave:** This event is fired when the mouse leaves an element while a drag is occurring. Listeners should remove any highlighting or insertion markers used for drop feedback.
5	**drag:** Fires every time the mouse is moved while the object is being dragged.
6	**drop:** The drop event is fired on the element where the drop was occurred at the end of the drag operation. A listener would be responsible for retrieving the data being dragged and inserting it at the drop location.
7	**dragend:** Fires when the user releases the mouse button while dragging an object.

HTML5 Forms (new attributes for <input> tag)

New Input Types

The input element's type attribute now has the following new values.

Input type: color

Select a color from a color picker:

Example

```
<!DOCTYPE html>
<html>
<body>
        <form action="#">
         Select your favorite color: <input type="color" name="favcolor"><br>
```

```
<input type="submit">
</form>
</body>
</html>
```

Input type: date

Define a date control:

Example

```
<!DOCTYPE html>
<html>
<body>
<form action="#">
Birthday: <input type="date" name="bday">
<input type="submit">
</form>
</body>
</html>
```

Input type: datetime

Define a date and time control (with time zone):

Example

```
<!DOCTYPE html>
<html>
<body>
<form action="#">
Birthday (date and time): <input type="datetime" name="bdaytime">
<input type="submit">
</form>
</body>
</html>
```

The input type datetime has been removed from the HTML standard. Use datetimelocal instead.

Input type: datetime-local

Define a date and time control (no time zone):

Example

```
<!DOCTYPE html>
<html>
<body>
<form action="#">
 Birthday (date and time): <input type="datetime-local" name="bdaytime">
 <input type="submit">
</form>
</body>
</html>
```

Input type: email

Define a field for an e-mail address (will be automatically validated when submitted):

Example

```
<!DOCTYPE html>
<html>
<body>
<form action="#">
 E-mail: <input type="email" name="usremail">
 <input type="submit">
</form>
</body>
</html>
```

Input type: file

Define a file-select field and a "Browse..." button (for file uploads):

Example

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

```
<body>
<form action=”#”>
 Select a file: <input type=”file” name=”img”>
 <input type=”submit”>
</form>
</body>
</html>
```

Input type: hidden

Define a hidden field (not visible to a user). A hidden field often stores a default value, or can have its value changed by a JavaScript.

Example

```
<!DOCTYPE html>
<html>
<body>
<form action=”#”>
 First name: <input type=”text” name=”fname”><br>
 <input type=”hidden” name=”country” value=”Norway”>
 <input type=”submit” value=”Submit”>
</form>
</body>
</html>
```

Input type: image

Define an image as a submit button:

Example

```
<!DOCTYPE html>
<html>
<body>
<form action=”#”>
 First name: <input type=”text” name=”fname”><br>
 Last name: <input type=”text” name=”lname”><br>
```

```
<input type=”image” src=”img_submit.gif” alt=”Submit” width=”48” height=”48”>
</form>
</body>
</html>
```

Input type: month

Define a month and year control (no time zone):

Example

```
<!DOCTYPE html>
<html>
<body>
<form action=”#”>
Birthday (month and year): <input type=”month” name=”bdaymonth”>
<input type=”submit”>
</form>
</body>
</html>
```

Input type: number

Define a field for entering a number (You can also set restrictions on what numbers are accepted):

Example

```
<!DOCTYPE html>
<html>
<body>
<form action=”#”>
Quantity (between 1 and 5): <input type=”number” name=”quantity” min=”1”
max=”5”>
<input type=”submit”>
</form>
</body>
```

```
</html>
```

Use the following attributes to specify restrictions:

- max - specifies the maximum value allowed
- min - specifies the minimum value allowed
- step - specifies the legal number intervals
- value - Specifies the default value

Input type: range

Define a control for entering a number whose exact value is not important (like a slider control). You can also set restrictions on what numbers are accepted.

Example

```
<!DOCTYPE html>
<html>
<body>
<form action="#">
Points: <input type="range" name="points" min="0" max="10">
<input type="submit">
</form>
</body>
</html>
```

Use the following attributes to specify restrictions:

- max - specifies the maximum value allowed
- min - specifies the minimum value allowed
- step - specifies the legal number intervals
- value - Specifies the default value

Input type: reset

Define a reset button (resets all form values to default values):

Example

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

```
<form action="#">
 Email: <input type="email" name="email"><br>
 Pin: <input type="text" name="pin" maxlength="4"><br>
 <input type="reset" value="Reset">
 <input type="submit" value="Submit">
</form>
<p>Click on the reset button to reset the form.</p>
</body>
</html>
```

Tip: Use the reset button carefully! It can be annoying for users who accidentally activate the reset button.

Input type: search

Define a search field (like a site search, or Google search):

Example

```
<!DOCTYPE html>
<html>
<body>
<form action="#">
 Search Google: <input type="search" name="googlesearch"><br>
 <input type="submit">
</form>
</body>
</html>
```

Input type: tel

Define a field for entering a telephone number:

Example

```
<!DOCTYPE html>
<html>
<body>
<form action="#">
```

```
Telephone: <input type="tel" name="usrtel"><br>
<input type="submit">
</form>
</body>
</html>
```

Input type: time

Define a control for entering a time (no time zone):

Example

```
<!DOCTYPE html>
<html>
<body>
<form action="#">
Select a time: <input type="time" name="usr_time">
<input type="submit">
</form>
</body>
</html>
```

Input type: url

Define a field for entering a URL:

Example

```
<!DOCTYPE html>
<html>
<body>
<form action="#">
Add your homepage: <input type="url" name="homepage"><br>
<input type="submit">
</form>
</body>
</html>
```

Tip: Safari on iPhone recognizes the url input type, and changes the on-screen keyboard to match it (adds .com option).

Input type: week

Define a week and year control (no time zone):

Example

```
<!DOCTYPE html>
<html>
<body>
<form action="demo_form.asp">
 Select a week: <input type="week" name="year_week">
 <input type="submit">
</form>
</body>
</html>
```

Exercise

Choose the correct answer from the given alternatives.

1. Which of the following tag is used to draw graphics via JavaScript:

   a. <fig> b. <svg> c. <section> d. <header>

2. The tag used to define navigation link is

   a. b. <footer> c. <nav> d. <time>

3. A page can contain more than of main tag.

   a. True b. False

4. _____ specify illustrations, diagram, photos in pages.

   a. <mark> b. <fig> c. <aside> d. <details>

5. Which of the following attribute triggers event at the end of a drag operation?

   a. Ondrageleave b. Ondrag c. Ondragend d. Ondragenter

6. Which of the following is not HTML5 tag?

   a. b. <svg> c. <canvas> d. <audio>

7. HTML5 documents may contains a _____ element, which is used to set the header section of a document.

   a. <footer> b. <svg> c. <section> d. <header>

8. Following audio format is supported across all browsers.

   a. Mp3 b. Ogg c. None of above d. Both of above

Write short answers to the following questions.

1. What is HTML5?
2. Is HTML5 backward compatible with old browsers?
3. List out the features of HTML5.
4. What is the use of <figure> tag in HTML5?
5. How many tags are removed completely in HTML5?

Write long answers to the following questions.

1. Explain new form input types in HTML5.
2. What are the differences between SVG and Canvas element?
3. Explain the usage of following HTML5 semantic elements.
    a. <article>
    b. <footer>
    c. <header>
    d. <section>
    d. Explain the advantages and disadvantages of HTML5.
    e. Write down any four differences between HTML and HTML5.

Practical works

a. Complete the following exercises as directed by your instructor.
    a. Make a chart of all the semantic tags with its main function in chart paper.
    b. Create a simple registration form including all new input types in HTML5.

b. Make PowerPoint presentation on the following topic.
    a. HTML5 graphics.
    b. How semantic elements helps on effective web page development?

Unit 6 Cascading Style Sheets (CSS)

6.1 Introduction

CSS (Cascading Style Sheets) styling language used to style the webpages. It defines the layout of the webpage and make web pages look good and presentable. Moreover, CSS describes how the HTML elements in the document are to be rendered on screen. CSS is used to define styles of web pages, including the design, layout and variations in display for different devices and screen sizes. CSS saves time and work while developing stylish webpages by controlling the layout of multiple pages at a time using external stylesheet (further detail on types of CSS).

CSS Syntax

CSS can be defined as follows. It consists of a selector and a declaration block.

```
Selector
        {
        property1: value1;
        property2: value2;
        ……….
        propertyN: valueN;
        }
```

Syntax Description

- The selector points to the HTML element you want to style. (further details in 6.3)
- The declaration block contains one or more declarations separated by semicolons.
- Each declaration includes a CSS property name and a value, separated by a colon.
- Multiple CSS declarations are separated with semicolons, and declaration blocks are surrounded by curly braces.

Example

h1 {
color: blue;
font-size:12px;
}
	• h1 is a selector in CSS (it points to the HTML element you want to style: <h1>).
	• color is a property, and blue is the property value
	• font-size is a property, and 12px is the property value

p {
color: red;
text-align: center;
}
	• p is a selector in CSS (it points to the HTML element you want to style: <p>).
	• color is a property, and red is the property value
	• text-align is a property, and center is the property value

6.2 Types of CSS

There are three types of CSS:

- Inline CSS
- Internal or Embedded CSS
- External CSS

Inline CSS

Inline CSS is generally used to style a specific HTML element only. We can write inline CSS by adding the style attribute to each HTML element.

Syntax:

<htmltag style="cssproperty1:value; cssproperty2:value;"> </htmltag>

Example

- <h1 style="color: blue"> Hello world! </h1>
- <h2 style="color:red;margin-left:40px;">This is inline css</h2>
- Following example shows complete html document with inline CSS.

```
<!DOCTYPE html>
<html>
        <body style="background-color:black;">
        <h1 style="color:white;padding:30px;">My Title</h1>
        <p style="color:white;">Something about me.p>
```

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

Advantages of Inline CSS:

- We can easily and quickly insert CSS rules to an HTML page.
- This method is useful for testing or previewing the changes, and performing quick-fixes to your website.
- You don't need to create and upload a separate document as in the external style.

Disadvantages of Inline CSS:

- It is time-consuming
- It makes your HTML structure messy.
- Styling multiple elements can affect your page's size and download time.

Internal CSS

The internal CSS is used to add a specific style for a single document or page in the website. It is defined within <head> section of the HTML page inside the <style> tag. Internal CSS is also known as embedded CSS since it is embedded within HTML file. This can be used when a single HTML document must be styled uniquely.

Syntax

```
<!DOCTYPE html>
<html>
<head>
<style>
Selector {
        property1: value1;
        property2: value2;
        …….
propertyN:valueN
}
</style>
</head>
```

```
<body>…….</body>
</html>
Example
<!DOCTYPE html>
<html>
<head>
<style>
body {
        background-color: blue;
        }
h1 {
        color: red;
        padding: 60px;
        }
</style>
</head>
<body>
        <h1>This is internal css</h1>
        <p> Internal CSS is used to style single page </p>
</body>
</html>
```

Advantages of Internal CSS

- Since you'll only add the code within the same HTML file, you don't need to upload multiple files.
- You can use class and ID selectors in this style sheet.

Disadvantages of Internal CSS

- Adding the code to the HTML document can increase the page's size and loading time.
- Since you'll only add the code within the same HTML file, styling multiple pages will become time-consuming.

External CSS

External CSS is used when multiple pages are needed to be styled at once. With an external CSS, you can change the look of an entire website by changing just one file. We can write external CSS in a separate .CSS file. Each HTML page must include a reference to the external CSS file inside the <link> tag, inside the <head> tag.

First of all create an external CSS file using an editor, simply Notepad with any name like example.css. The file will contain all the styles required to style your HTML document (let's say example.html. Then include the example.css in the example.html using link tag as follows.

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

Example

The example.css will look like as follows

```
body {
   background-color: blue;
            }
h1 {
  color: red;
  padding: 60px;
  }
```

The example.html with external CSS will look like as follows

```
<!DOCTYPE html>
<html>
   <head>
      <link   rel="stylesheet"   type="text/css"   href="example.css"/>
   </head>
<body>
<h1>My Heading</h1>
<p>This is my paragraph.</p>
</body>
</html>
```

Advantages of External CSS

- Since the CSS code is in a separate file, your HTML files will have a cleaner structure and are smaller in size.
- You can use the same .css file for multiple pages when you want the same look for each page.

Disadvantage of External CSS

- Your pages may not be rendered correctly until the external CSS is loaded.
- Uploading multiple external CSS files can increase the download time of a website.

Difference between Inline, Internal, and External CSS

Inline CSS	Internal CSS	External CSS
Inline CSS is used to style a specific HTML element.	Internal CSS is used to style a specific HTML page.	External CSS is used to change the look of an entire website by changing just one file.
You can write inline CSS using the style attribute.	You can write Internal CSS using the <style> tag.	You can write External CSS in a .css file.
It doesn't allow you to use any selectors.	It allows you to use selectors. eg:- id, class, tag name, etc.	It also allows you to use selectors.
It takes time to use as each element need to add.	It is also time-consuming but in comparison to Inline CSS is less.	It saves time as you can use the same file on multiple pages for the same look.

6.3 CSS Selectors

Selectors are patterns used to select the element(s) you want to style. CSS selectors are used to "find" (or select) the HTML elements you want to style. It is a pattern of elements and other terms that tell the browser which HTML elements should be selected to have the CSS property values inside the rule applied to them. The element or elements which are selected by the selector are referred to as the subject of the selector. A CSS selector is the first part of a CSS Rule.

Types of CSS Selector

The most basic types of CSS selectors are as follows:

- Element Selector

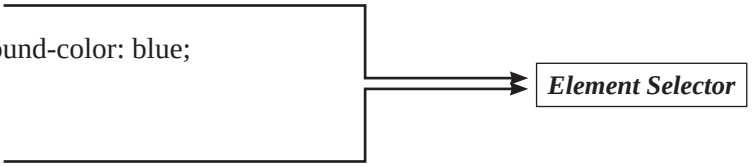
- ID Selector
- Class Selector
- CSS Universal Selector
- CSS Group Selector

Element Selector

The Element Selector use HTML tag name as a selector to write the CSS rule. Thus, it is also called as Tag Name Selector.

Example

```
<!DOCTYPE html>
<html>
<head>
<style>
body {
   background-color: blue;
}
h1 {
   color: red;
   padding: 60px;
}
</style>
</head>
<body>
        <h1>Element Selector</h1>
        <p> This is element selector </p>
</body>
</html>
```



In above code snippet, tag name (body, h1) is used as selector.

ID Selector

In ID Selector, the value id attribute of an HTML element is used to style the specific element. An id is always unique within the page so it is chosen to select a single, unique element. It is written with the hash character (#), followed by the id of the element.

```
<!DOCTYPE html>
<html>
<head>
<style>
#myP{
   background-color: blue;
}
#myh1 {
   color: red;
   padding: 60px;
}
</style>
</head>
<body>
        <h1 id =”myh1”>Id Selector</h1>
        <p id=”myP”> This is id selector </p>
</body>
</html>
```

ID Selector to style element h1 whose value of id attribute is ‘myh1’ and body whose value of id attribute is ‘myP’

In above code snippet, id attribute myh1 and myP is used to style the element h1 and p respectively.

Class Selector

In the class selector the value of class attribute of HTML elements is used to style the elements. It is used with dot (.) symbol followed by the class name. We can define class

attribute to a HTML element to identify more than one HTML element and that class name can be used as a selector. A class name should not be started with a number.

```
<!DOCTYPE html>
<html>
<head>
<style>
.myh{
   background-color: blue;
}
.myP {
   color: red;
   padding: 60px;
}
</style>
</head>
<body>
        <h1 class =”myh”>Class Selector</h1>
        <h2 class=”myh”>Hello</h2>
        <p class=”myP”> This is class selector </p>
        <p> This is paragraph without style</p>
</body>
</html>
```

Class Selector to style element h1 whose value of class attribute is ‘myh1’ and p whose value of class attribute is ‘myP’

In above code snippet, class attribute myh1 and mybody is used to style the element h1 and p respectively.

CSS Universal Selector

The universal selector is used to give same style to all the pages. It selects all the elements on the pages. It uses asterisk (*) symbol to match element of any type. It acts like a wildcard and matches all element types in an HTML page.

```
<!DOCTYPE html>
<html>
<head>
<style>
* {
   color: blue;
}
</style>
</head>
<body>
        <h1>Universal Selector</h1>
        <p> It selects all the elements on the pages. </p>
</body>
</html>
```

Group Selector

The group selector is used to select all the elements with the same style definitions. Group selector is used to minimize the code. Commas are used to separate each selector in grouping.

```
<!DOCTYPE html>
<html>
<head>
<style>
h1, p {
   color: blue;
}
</style>
</head>
<body>
        <h1>Universal Selector</h1>
        <p> It selects all the elements on the pages. </p>
</body>
</html>
```

6.4 CSS Basic Properties

The CSS basic properties are described below.

6.4.1 CSS Font

CSS Font property is used to style the texts. By the use of CSS font property, the text size, color, style etc. can be change so that the look and feel of the web pages are presentable. CSS provide several properties for styling the font of the text, including changing their face, controlling their size and boldness, managing variant, and so on.

The font properties are:

- font-family,
- font-style
- font-weight,
- font-size
- font-variant

Font Family

The font-family property is used to specify the font to be used to render the text. This property can hold several comma-separated font names as a fallback system, so that if the first font is not available on the user's system, browser tries to use the second one, and so on.

Example

```
body {
   font-family: Arial, Helvetica, sans-serif;
}
```

Note: If the name of a font family contains more than one word, it must be placed inside quotation marks, like "Times New Roman", "Courier New", "Segoe UI", etc.

Font Style

The font-style property is used to set the font face style for the text content of an element. The font style can be normal, italic or oblique. The default value is normal.

```
p {
  font-style: normal;
```

```
}
```

Note: At first glance both oblique and italic font styles appear the same thing, but there is a difference. The italic style uses an italic version of the font while oblique style on the other hand is simply a slanted or sloped version of the normal font.

Font Size

The font-size property is used to set the size of font for the text content of an element. There are several ways to specify the font size values. They are: keywords, percentage, pixels, ems.

Setting Font Size with Pixels

Setting the font size in pixel values (e.g. 14px, 16px, etc.) is a good choice when you need the pixel accuracy. Pixel is an absolute unit of measurement which specifies a fixed length.

Example

```
h1 {
   font-size: 24px;
}
```

Defining the font sizes in pixel is not considered very accessible, because the user cannot change the font size from the browser settings.

Setting Font Size with EM

The em unit refers to the font size of the parent element. When defining the font-size property, 1em is equal to the size of the font that applies to the parent of the element.

So, if you set a font-size of 20px on the body element, then 1em = 20px and 2em = 40px.

However, if you haven't set the font size anywhere on the page, then it is the browser default, which is normally 16px. Therefore, by default 1em = 16px, and 2em = 32px.

Example

```
h1{
   font-size: 2em;    /* 32px/16px=2em */
}
```

Setting Font Size with Keywords

CSS provide several keywords that you can use to define font sizes.

An absolute font size can be specified using one of the following keywords:

- xx-small,
- x-small,
- small,
- medium,
- large,
- x-large,
- xx-large.

Whereas, a relative font size can be specified using the keywords: smaller or larger.

```
body {
   font-size: large;
}
h1 {
   font-size: larger;
}
p {
   font-size: smaller;
}
```

Note: The keyword medium is equivalent to the browsers default font-size, which is normally 16px. Likewise, xx-small is the equivalent of 9 pixels, x-small is 10 pixels, small is 13 pixels, large is 18 pixels, x-large is 24 pixels, and xx-large is 32 pixels.

Tip: By setting a font size on the body element, you can set the relative font sizing everywhere else on the page, giving you the ability to easily scale the font size up or down accordingly.

Font Weight

The font-weight property specifies the weight or boldness of the font. This property can take one of the following values: normal, bold, bolder, lighter, 100, 200, 300, 400, 500, 600, 700, 800, 900 and inherit. The numeric values 100-900 specify the font weights, where each number represents a weight greater than its predecessor. 400 is same as normal & 700 is same as bold. The bolder and lighter values are relative to the inherited font weight, while the other values such as normal and bold are absolute font weights.

Example

```
p {
   font-weight: bold;
}
```

Note: Most of the fonts are only available in a limited number of weights; often they are available only in normal and bold. In case, if a font is not available in the specified weight, an alternate will be chosen that is the closest available weight.

Font Variant

The font-variant property allows the text to be displayed in a special small-caps variation. Small-caps or small capital letters are slightly different to normal capital letters, in which lowercase letters appear as smaller versions of the corresponding uppercase letters. The value normal removes small caps from the text which is already formatted that way.

Example

```
p {
   font-variant: small-caps;
}
```

6.4.2 CSS Colors and Background

CSS Colors

The CSS color property defines fore color of the text of an element.

Example

```
body {
   color: #ff5722;
}
```

Defining Color Values

Colors in CSS most often specified in the following formats:

- a color keyword - like "red", "green", "blue", "transparent", etc.
- a HEX value - like "#ff0000", "#00ff00", etc.
- an RGB value - like "rgb(255, 0, 0)"

CSS Background

The CSS properties are used to style the background of the web pages. CSS provide several properties for styling the background of an element, including coloring the background, placing images in the background and managing their positioning, etc.

The CSS Background properties are:

- background-color
- background-image
- background-repeat
- background-attachment
- background-position

background-color

The background-color property is used to set the background color of an element.

Example

```
body {
    background-color: #f0e68c;
}
```

Color values in CSS are most often specified in the following formats:

- a color name - like "red"
- a HEX value - like "#ff0000"
- an RGB value - like "rgb(255, 0, 0)"

Background-image

The background-image property set an image as a background of an HTML element.

Example

```
body {
    background-image: url("images/tile.png");
}
```

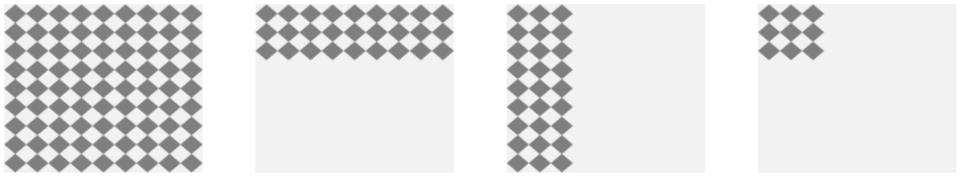
By default, browser repeats or tiles the background image both horizontally and vertically to fill the entire area of an element. You can control this with background-repeat property.

Background-repeat

The background-repeat property allows you to control how a background image is repeated or tiled in the background of an element. You can set a background image to repeat vertically (y-axis), horizontally (x-axis), in both directions, or in neither direction.

Example

```
body {
   background-image: url("texture.png");
   background-repeat: no-repeat;
}
```



Background-position

The background-position property is used to control the position of the background image. If no background position has been specified, the background image is placed at the default top-left position of the element i.e. at (0,0).

Example

```
body {
   background-image: url("images/robot.png");
   background-repeat: no-repeat;
   background-position: right top;
}
```

background-attachment

The background-attachment property determines whether the background image is fixed with regard to the viewport or scrolls along with the containing block.

Example

```
body {
   background-image: url("images/bell.png");
   background-repeat: no-repeat;
   background-attachment: fixed;
}
```

The Background Shorthand Property

It is also possible to specify all these properties in one single property to shorten the code or avoid extra typing. This is called a shorthand property.

Example

```
body {
   background-color: #f0e68c;
   background-image: url("images/smiley.png");
   background-repeat: no-repeat;
   background-attachment: fixed;
   background-position: 250px 25px;
}
```

Above code can be written in shorthand as follows:

```
body {
   background: #f0e68c url("images/smiley.png") no-repeat fixed 250px 25px;
}
```

When using the background shorthand property, the order of the property values should be. background: color image repeat attachment position;

6.4.3 CSS Borders

The CSS border property is used to style the border of an element. The CSS border properties are:

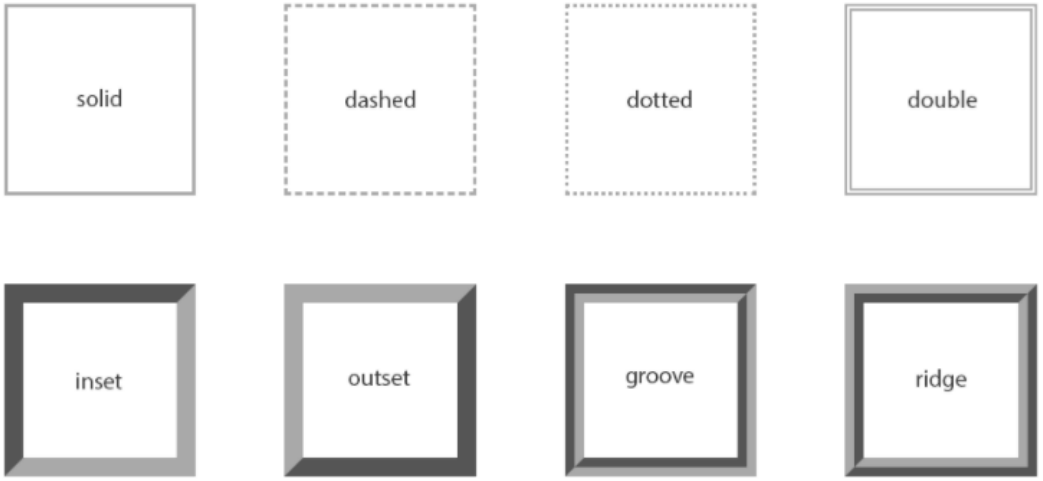
- border-style
- border-size
- border-width

Border-style

The border-style properties set the line style of the border. The border-style property can have the values like: none, hidden, solid, dashed, dotted, double, inset, outset, groove, and ridge. A border style should be specified in order to make the border appear around an element, because the default border style is none.

Example

```
h1 {
   border-style: dotted;
}
```



The values none and hidden displays no border, however, there is a slight difference between these two values.

In order to specify individual border style following properties are used.

- border-top-style for top border style
- border-right-style for right border style
- border-bottom-style for bottom border style
- border-left-style for left border style

Example

```
p {
  border-top-style: dotted;
  border-right-style: solid;
  border-bottom-style: dotted;
  border-left-style: solid;
}
```

border-size

The border-width property specifies the width of the border area. A border style should be specified in order to apply border width.

Example

```
p {
   border-style: solid;
   border-width: 10px;
}
```

In order to specify individual border size following properties are used.

- border-top-size for top border size
- border-right-size for right border size
- border-bottom-size for bottom border size
- border-left-size for left border size

Example

```
p {
  border-top-size: 1px;
  border-right-size: 2px;
  border-bottom-size: 1px;
  border-left-size: 2px;
}
```

Border-color

The border-color properties specify the color of the border. A border style should be specified in order apply the border color.

Example

```
p {
   border-style: solid;
   border-color: #ff0000;
}
```

In order to specify individual border color following properties are used.

- border-top-color for top border color
- border-right-color for right border color
- border-bottom-color for bottom border color
- border-left-color for left border color

Example

```
p {
  border-top-color: 1px;
  border-right-color: 2px;
  border-bottom-color: 1px;
  border-left-color: 2px;
}
```

The Border Shorthand Property

The border CSS property is a shorthand property for setting one or more of the individual border properties border-width, border-style and border-color in a single rule.

```
p {
   border: 5px solid #00ff00;
}
```

6.4.4 CSS Margins and Paddings

CSS Margins

The CSS margin properties are used to create spacing around the border of an element's box.

Example

```
h1 {
   margin:50px:
}
```

The margin properties can be specified using the following values:

- length - specifies a margin in px, em, rem, pt, cm, etc.
- % - specifies a margin in percentage (%) of the width of the containing element.
- auto - the browser calculates a suitable margin to use.
- inherit - specifies that the margin should be inherited from the parent element.

In order to specify individual margin following properties are used.

- margin-top for specifying margin at top of element.
- margin-right for specifying margin at right of element.
- margin-bottom for specifying margin bottom of element.
- margin-left for specifying margin at top left of element.

Example

```
p {
  margin-top: 100px;
  margin-bottom: 100px;
  margin-right: 150px;
  margin-left: 80px;
}
```

CSS Padding

The CSS padding properties is used to create spacing around the element's content, inside of any defined border.

Example

```
h1 {
   padding:50px:
}
```

The margin properties can be specified using the following values:

- length - specifies a padding in px, pt, cm, etc.
- % - specifies a padding in % of the width of the containing element
- inherit - specifies that the padding should be inherited from the parent element

In order to specify individual margin following properties are used.

- padding-top for specifying padding at top of element.
- padding-right for specifying padding at right of element.
- padding-bottom for specifying padding bottom of element.
- padding-left for specifying padding at top left of element.

Example

```
p {
  padding-top: 100px;
  padding-bottom: 100px;
  padding-right: 150px;
  padding-left: 80px;
}
```

6.4.5 CSS Text

The CSS text properties is used to define various text styles such as color, alignment, spacing, decoration, transformation, etc. very easily and effectively. The properties provide precise control over the visual appearance of the characters, words, spaces, and so on.

The commonly used text properties are:

- text-color
- text-align
- text-decoration

- text-transform
- text-indent
- line-height
- letter-spacing
- word-spacing

Text-color

The color of the text is defined by the CSS color property.

Example

```
body {
   color: #434343;
}
```

Although, the color property seems like it would be a part of the CSS text, but it is actually a standalone property in CSS.

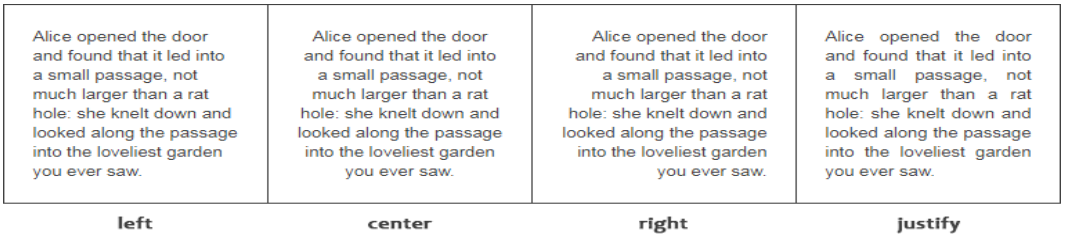
Text-align

The text-align property is used to set the horizontal alignment of the text. Text can be aligned in four ways: to the Left, Right, Centre, Justified (straight left and right margins).

Example

```
h1 {
   text-align: center;
}
```

When text-align is set to justify, each line is stretched so that every line has equal width (except the last line), and the left and right margins are straight. This alignment is generally used in publications such as magazines and newspapers.



Text-decoration

The text-decoration property is used to set or remove decorations from text. This property typically accepts one of the following values: underline, overline, line-through. It is recommended to avoid underline text that is not a link, as it might confuse the visitor.

Example

```
h1 {
   text-decoration: overline;
}
h2 {
   text-decoration: line-through;
}
h3 {
   text-decoration: underline;
}
```

Text-transform

The text-transform property is used to set the cases for a text. Texts are often written in mixed case. However, in certain situations you may want to display your text in entirely different case. Using this property you can change an element's text content into uppercase lowercase letters, capitalize the first letter of each word without modifying the original text.

Example

```
h1 {
   text-transform: uppercase;
}
h2 {
   text-transform: capitalize;
}
h3 {
   text-transform: lowercase;
}
```

text-indent

The text-indent property is used to set the indentation of the first line of text within a block of text. It is typically done by inserting the empty space before the first line of text.

The size of the indentation can be specified using

- percentage (%)
- length values in pixels
- ems

Example

```
p {
   text-indent: 100px;
}
```

Line-height

The line-height property is used to set the height of the text line. It is also called leading and commonly used to set the distance between lines of text. The value of this property can be a number, a percentage (%), or a length in pixels, ems, etc.

Example

```
p {
   line-height: 1.2;
}
```

Letter-spacing

The letter-spacing property is used to set extra spacing between the characters of text. This property can take a length value in pixels, ems, etc. It may also accept negative values. When setting letter spacing, a length value indicates spacing in addition to the default inter-character space.

Example

```
h1 {
   letter-spacing: -3px;
}
p {
```

```
    letter-spacing: 10px;
}
```

Word-spacing

The word-spacing property is used to specify additional spacing between the words. This property can accept a length value in pixels, ems, etc. Negative values are also allowed. Word spacing can be affected by text justification. Also, even though whitespace is preserved, spaces between words are affected by the word-spacing property.

```
p.normal {
    word-spacing: 20px;
}
p.justified {
    word-spacing: 20px;
    text-align: justify;
}
p.preformatted {
    word-spacing: 20px;
    white-space: pre;
}
```

6.4.6 CSS Height/Width

The height and width properties are used to set the height and width of an element. The height and width properties may have the following values:

- auto - This is default. The browser calculates the height and width
- length - Defines the height/width in px, cm etc.
- % - Defines the height/width in percent of the containing block
- initial - Sets the height/width to its default value
- inherit - The height/width will be inherited from its parent value

Example

```
p {
  height: 200px;
```

```
    width: 50%;
    background-color: red;
}
```

6.4.7 CSS Position and Float

CSS Position

The CSS position is used for positioning the elements in required and appropriate position in webpages. The appropriate positioning of elements are necessary for good layout and design of the webpages. There are four ways to position the elements in pages. They are:

- Static Position
- Relative Position
- Absolute Position
- Fixed Position

Static Position

Static position is set by default according to the flow of elements. Static positioned elements are not affected by the top, bottom, left, right, and z-index properties.

Relative Position

In relative position element is positioned relative to its normal position. The element's box position is calculated according to the normal flow. Then the box is shifted from this normal position according to the properties — top or bottom and/or left or right. A relatively positioned element can be moved and overlap other elements, but it keeps the space originally reserved for it in the normal flow.

Example

```
.myelement {
    position: relative;
    left: 100px;
}
```

Absolute Position

In an absolutely position element is positioned relative to the first parent element that has a position other than static. If no such element is found, it will be positioned on a page

relative to the 'top-left' corner of the browser window.

Example

```
.myelement {
   position: absolute;
   top: 200px;
   left: 100px;
}
```

Fixed Position

The fix position is sub type of absolute position in which element is fixed with respect to the browser's viewport and does not move when scrolled.

Example

```
.myelement {
   position: fixed;
   top: 200px;
   left: 100px;
}
```

CSS Float

The CSS float is use to float elements to the left or right, but only applies to the elements that generate boxes that are not absolutely positioned. Any element that follows the floated element will flow around the floated element on the other side.

The float property may have one of the three values:

- Left: The element floats on the left side of its containing block.
- Right: The element floats on the right side of its containing block.
- None: Removes the float property from an element.

Example

```
img {
   float: left;
}
```

6.4.8 CSS Overflow

The CSS overflow is used to specify whether to clip content, render scroll bars or display overflow content of a block-level element.

The float property may have one of the three values:

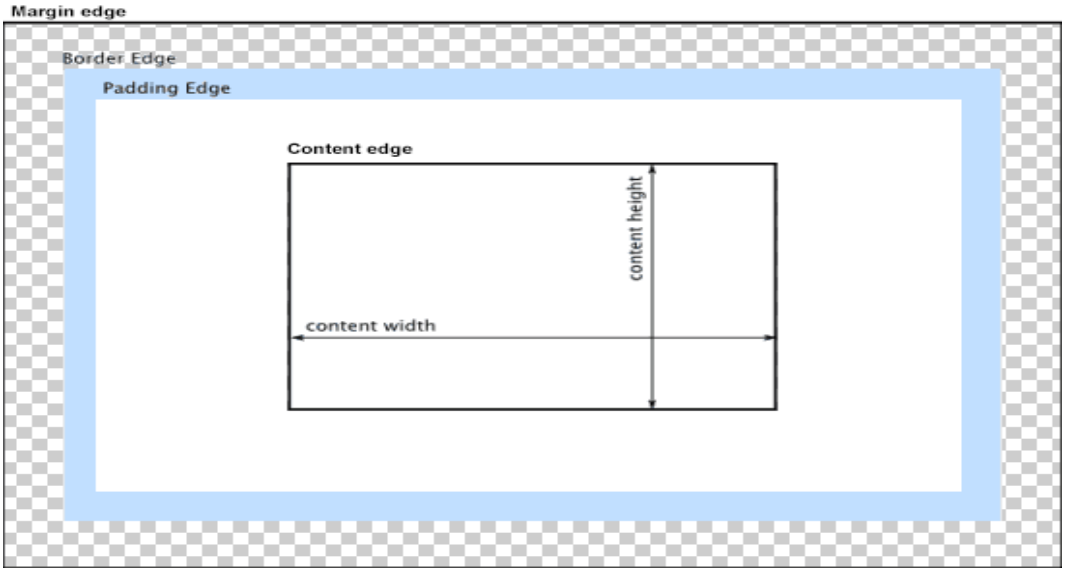
- Visible: The default value. Content is not clipped; it will be rendered outside the element's box, and may thus overlap other content.
- Hidden: Content that overflows the element's box is clipped and the rest of the content will be invisible.
- Scroll: The overflowing content is clipped, just like hidden, but provides a scrolling mechanism to access the overflowed content.
- Auto: If content overflows the element's box it will automatically provide the scrollbars to see the rest of the content, otherwise scrollbar will not appear.

Example

```
.myelement {
    width: 250px;
    height: 150px;
    overflow: scroll;
}
```

6.4.9 CSS Box Model

CSS Box Model refers to the standard model according to which each element in the document is laid out so that browser rendering engine represents each element as a rectangular box. Every element that can be displayed on a web page is comprised of one or more rectangular boxes. CSS determines the size, position, and properties (color, background, border size, etc.) of these boxes. These boxes can have different properties and can interact with each other in different ways, but every box has a content area and optional surrounding padding, border, and margin areas.



Content Area

The content area, bounded by the content edge, contains the "real" content of the element, such as text, an image, or a video player. Its dimensions are the content width (or content-box width) and the content height (or content-box height). It often has a background color or background image.

Padding Area

The padding area, bounded by the padding edge, extends the content area to include the element's padding. Its dimensions are the padding-box width and the padding-box height. The thickness of the padding is determined by the padding-top, padding-right, padding-bottom, padding-left, and shorthand padding properties.

Border Area

The border area, bounded by the border edge, extends the padding area to include the element's borders. Its dimensions are the border-box width and the border-box height. The thickness of the borders is determined by the border-width and shorthand border properties.

Margin Area

The margin area, bounded by the margin edge, extends the border area to include an empty area used to separate the element from its neighbors. Its dimensions are the margin-box width and the margin-box height. The size of the margin area is determined by the margin-top, margin-right, margin-bottom, margin-left, and shorthand margin properties. When

margin collapsing occurs, the margin area is not clearly defined since margins are shared between boxes.

6.4.10 CSS Navigation Bar

The navigation bars are used to navigate from one page to another in website. The CSS navigation is used to style those navigation bars. It is important for a website to have easy-to-use navigation. Generally, there are two types of navigation bars: Horizontal and Vertical

Horizontal Navigation Bar

In horizontal navigation bar, the navigation links are in horizontal directions i.e., from left to right. There are two ways to create a horizontal navigation bar: inline and floating.

Inline List Items

In this method, the display properties of list item <li> is set as inline as in example below.

```
<!DOCTYPE html>
<html>
<head>
<style>
ul {
  list-style-type: none;
  margin: 0;
  padding: 0;
}
li {
  display: inline;
}
</style>
</head>
<body>
<ul>
  <li><a href="#home">Home</a></li>
  <li><a href="#news">News</a></li>
```

```
  <li><a href="#contact">Contact</a></li>
  <li><a href="#about">About</a></li>
</ul>
</body>
</html>
```

Output:

Floating List Items

Another way of creating a horizontal navigation bar is to float the <li> elements, and specify a layout for the navigation links as in example below.

```
<!DOCTYPE html>
<html>
<head>
<style>
ul {
  list-style-type: none;
  margin: 0;
  padding: 0;
  overflow: hidden;
}
li {
  float: left;
}
li a {
  display: block;
  padding: 8px;
  background-color: #dddddd;
}
</style>
```

```
</head>
<body>
<ul>
  <li><a href="#home">Home</a></li>
  <li><a href="#news">News</a></li>
  <li><a href="#contact">Contact</a></li>
  <li><a href="#about">About</a></li>
</ul>
</body>
</html>
```

Output

Vertical Navigation Bar

In vertical navigation bar, the navigation links are in vertical directions i.e., from top to bottom. To build a vertical navigation bar, you can style the <a> elements inside the list.

Example

```
<!DOCTYPE html>
<html>
<head>
<style>
ul {
  list-style-type: none;
  margin: 0;
  padding: 0;
  width: 60px;
}
li a {
  display: block;
  background-color: #dddddd;
}
```

```
</style>
</head>
<body>
<ul>
  <li><a href="#home">Home</a></li>
  <li><a href="#news">News</a></li>
  <li><a href="#contact">Contact</a></li>
  <li><a href="#about">About</a></li>
</ul>
</body>
</html>
```

Output:

6.5 CSS Advance Properties

6.5.1 CSS Rounded Corners

The border-radius property is used to make the corner rounded of any element.

Example

```
<!DOCTYPE html>
<html>
<head>
<style>
p{
  border-radius: 25px;
  background: #73AD21;
  padding: 20px;
  width: 200px;
  height: 150px;
```

```
}
</style>
</head>
<body>
<h1>The border-radius Property</h1>
<p>Rounded corners for an element with a specified background color:</p>
</body>
</html>
```

In order to specify individual border corner following properties are used.

- border-top-left-radius: Defines the shape of the border of the top-left corner
- border-top-right-radius: Defines the shape of the border of the top-right corner
- border-bottom-right-radius: Defines the shape of the border of the bottom-right corner
- border-bottom-left-radius: Defines the shape of the border of the bottom-left corner

6.5.2 CSS Border Images

The CSS border-image property allows you to specify an image to be used instead of the normal border around an element.

The property has three parts:

- The image to use as the border
- Where to slice the image
- Define whether the middle sections should be repeated or stretched

Example

```
p {
  border: 10px solid transparent;
  padding: 15px;
  border-image: url(border.png) 30 round;
}
```

The border-image property is actually a shorthand property for the border-image-source, border-image-slice, border-image-width, border-image-outset and border-image-repeat properties.

6.5.3 CSS Text Effects

The CSS text effect consists of following properties:

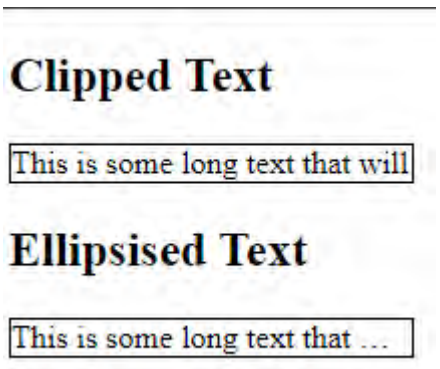
- text-overflow
- word-wrap
- word-break
- writing-mode

text-overflow

The CSS text-overflow property specifies how to indicate overflowed content of the element, that is not displayed in the pages. The values used for this property are: clip and ellipsis.

Example

```
<!DOCTYPE html>
<html>
<head>
<style>
.p1 {
  white-space: nowrap;
  width: 200px;
  border: 1px solid #000000;
  overflow: hidden;
  text-overflow: clip;
}
.p2 {
 white-space: nowrap;
  width: 200px;
  border: 1px solid #000000;
  overflow: hidden;
  text-overflow: ellipsis;
}
```



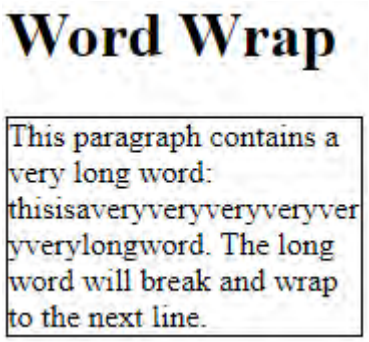
```
</style>
</head>
<body>
<h2>Clipped Text</h2>
<p class="p1">This is some long text that will not fit in the box</p>
<h2>Ellipsised Text</h2>
<p class="p2">This is some long text that will not fit in the box</p>
</body>
</html>
```

Word-wrap

The CSS word-wrap is used to break long words and wrap onto the next line. It allows to fit long word within area instead of expanding it outside.

Example

```
<!DOCTYPE html>
<html>
<head>
<style>
p.test {
  width: 11em;
  border: 1px solid #000000;
  word-wrap: break-word;
}
</style>
</head>
<body>
<h1>Word Wrap</h1>
    <p class="test">This paragraph contains a very long word: thisisaveryveryveryveryveryverylongword. The long word will break and wrap to the next line.</p>
</body>
</html>
```



Word-break

The CSS word-break property specifies line breaking rules, i.e., how a word should be broken or split when reaching the end of a line. It has two values keep-all and break-all.

Example

```
<!DOCTYPE html>
<html>
<head>
<style>
p.test1 {
  width: 140px;
  border: 1px solid #000000;
  word-break: keep-all;
}
p.test2 {
  width: 140px;
  border: 1px solid #000000;
  word-break: break-all;
}
</style>
</head>
<body>
<h1>The word-break Property</h1>
    <p class="test1">Keep All: This paragraph contains some text. This line will-break-at-hyphens. </p>
    <p class="test2">Break All- This paragraph contains some text. The lines will break at any character. </p>
</body>
</html>
```

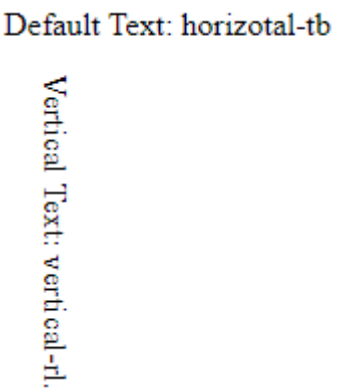
Keep All: This paragraph contains some text. This line will-break-at-hyphens.

Break All- This parag raph contains some te xt. The lines will brea k at any character.

Writing-mode

The CSS writing-mode property specifies whether lines of text are laid out horizontally or vertically. The possible values of this property are: horizontal-tb, vertical-rl,

```
<head>
<style>
.test1 {
  writing-mode: horizontal-tb;
}
 .test2 {
   writing-mode: vertical-rl;
}
</style>
</head>
<body>
<h1>The writing-mode Property</h1>
<p class="test1">Default Text: horizotal-tb</p>
<p class="test2">Vertical Text: vertical-rl.</p>
</body>
</html>
```



6.5.4 CSS Gradients

The CSS Gradients is used to display the smooth transitions between two or more specified colors. There are three types of CSS Gradients, they are:

- Linear Gradients (goes down/up/left/right/diagonally)
- Radial Gradients (defined by their center)
- Conic Gradients (rotated around a center point)

Linear Gradients

The linear gradient is defined by its diagnol.

Syntax:

background-image: linear-gradient(direction, color-stop1, color-stop2, ...);

Example CSS:

```
#grad {
  background-image: linear-gradient(to right, red , yellow);
}
```

Radial Gradients

A radial gradient is defined by its center.

Syntax:

background-image: radial-gradient(shape size at position, start-color, ..., last-color);

Example

```
#grad {
  background-image: radial-gradient(red, yellow, green);
}
```

Conic Gradients

A conic gradient is a gradient with color transitions rotated around a center point.

Syntax:

background-image: conic-gradient([from angle] [at position,] color [degree], color [degree], ...);

Example

```
#grad {
  background-image: conic-gradient(red, yellow, green);
}
```

6.5.5 CSS Shadows

The CSS shadow is used to add the shadow to the text and elements. The CSS shadows consists of two properties: text-shadow and box-shadow.

Text-shadow

The text-shadow property is used to add shadow to the text.

Example

```
<!DOCTYPE html>
<html>
<head>
<style>
h1 {
  text-shadow: 2px 5px 4px red;
```

```
}
</style>
</head>
<body>
<h1>Text-shadow effect!</h1>
</body>
</html>
```

box-shadow

The CSS box-shadow property is used to apply one or more shadows to an element.

Example

```
<!DOCTYPE html>
<html>
<head>
<style>
p {
  width: 300px;
  height: 100px;
  padding: 15px;
  background-color: blue;
  box-shadow: 10px 10px;
}
</style>
</head>
<body>
<h1>The box-shadow Property</h1>

<p>This is a div element with a box-shadow</p>
</body>
</html>
```

6.5.6 CSS Buttons

In CSS the default buttons can be styled using different basic CSS properties. The list of properties are given in the table below.

Properties	Description	Example
background-color	To change the button color	.button{background-color:red;}
color	To change the text color of button	.button{color:green;}
border	To change the border style of button	.button{border:none;}
box-shadow	To add the shadow in the button	.button{box-shadow:10px 10px}
Padding	To create more space inside the button's borders	.button{ padding:15px;}

: hover
: focus
:active
	To style the state of the button	.button:focus {
		outline-color: transparent; outline-style:solid;
		box-shadow: 0 0 0 4px #5a01a7;
		transition: 0.7s;
		}

Note: Other CSS properties can also be used to make more stylish buttons according to the requirement.

6.5.7 CSS Tooltips

The CSS Tooltips is used to create a tooltip that appears when the user moves the mouse over an element.

Example

```
<!DOCTYPE html>
<html>
<style>
.tooltip {
  position: relative;
```

```
  display: inline-block;
  border-bottom: 1px dotted black;
}
.tooltip .tooltiptext {
  visibility: hidden;
  width: 120px;
  background-color: black;
  color: #fff;
  text-align: center;
  border-radius: 6px;
  padding: 5px 0;
  /* Position the tooltip */
  position: absolute;
  z-index: 1;
}
.tooltip:hover .tooltiptext {
  visibility: visible;
}
</style>
<body style=”text-align:center;”>
<h2>Basic Tooltip</h2>
<p>Move the mouse over the text below:</p>
<div class=”tooltip”>Hover the Mouse Here
  <span class=”tooltiptext”>This is tool tip</span>
</div>
</body>
</html>
```

6.5.8 CSS Style Images

We can style the images in html by various ways as described below:

Rounded Images

The border-radius property is used to create rounded images

Example

```
<!DOCTYPE html>
<html>
<head>
<style>
img {
  border-radius: 8px; // or border-radius: 50% (to create circled image)
}
</style>
</head>
<body>
<h2>Rounded Image</h2>
<img src=”example.jpg” alt=”example” width=”300” height=”300”>
</body>
</html>
```

Thumbnail Images

The border property is used to create thumbnail images.

```
<!DOCTYPE html>
<html>
<head>
<style>
img {
  border: 1px solid #ddd;
  border-radius: 4px;
  padding: 5px;
```

```
  width: 150px;
}
</style>
</head>
<body>
<h2>Thumbnail Image</h2>
<img src=”example.jpg” alt=”example” style=”width:150px”>
</body>
</html>
```

Responsive Images

Responsive images will automatically adjust to fit the size of the screen. The max-height and height property is used to achieve this.

Example

```
<!DOCTYPE html>
<html>
<head>
<style>
img {
  max-width: 100%;
  height: auto;
}
</style>
</head>
<body>
<h2>Responsive Image</h2>
<p>Responsive images will automatically adjust to fit the size of the screen.</p>
<p>Resize the browser window to see the effect:</p>
<img src=”img_5terre_wide.jpg” alt=”Cinque Terre” width=”1000” height=”300”>
</body>
</html>
```

Centered Image

To center an image, the left and right margin property as well as display property is used. Set the left and right margin to auto and make it into a block element

Example

```
<!DOCTYPE html>
<html>
<head>
<style>
img {
  display: block;
  margin-left: auto;
  margin-right: auto;
}
</style>
</head>
<body>
<h2>Center an Image</h2>
<img src=”example.jpg” alt=”example” style=”width:50%”>
</body>
</html>
```

6.5.9 CSS Animation

The animation property in CSS can be used to animate many other CSS properties such as color, background-color, height, or width. Each animation needs to be defined with the @ keyframes at-rule which is then called with the animation property. Each @keyframes at-rule defines what should happen at specific moments during the animation. For example, 0% is the beginning of the animation and 100% is the end. These keyframes can then be controlled either by the shorthand animation property, or its eight sub-properties, to give more control over how those keyframes should be manipulated.

Sub-properties

- animation-name: declares the name of the @keyframes at-rule to manipulate.

- animation-duration: the length of time it takes for an animation to complete one cycle.
- animation-timing-function: establishes preset acceleration curves such as ease or linear.
- animation-delay: the time between the element being loaded and the start of the animation sequence (cool examples).
- animation-direction: sets the direction of the animation after the cycle. Its default resets on each cycle.
- animation-iteration-count: the number of times the animation should be performed.
- animation-fill-mode: sets which values are applied before/after the animation. For example, you can set the last state of the animation to remain on screen, or you can set it to switch back to before when the animation began.
- animation-play-state: pause/play the animation.

Example

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
  width: 100px;
  height: 100px;
  background-color: red;
  animation-name: example;
  animation-duration: 4s;
}
@keyframes example {
  from {background-color: red;}
  to {background-color: yellow;}
}
</style>
</head>
<body>
```

```
<h1>CSS Animation</h1>
<div></div>
</body>
```

6.5.10 CSS Transition

CSS transition properties allow elements to change values over a specified duration, animating the property changes, rather than having them occur immediately. So, if we have, say, a box with a red background that we want to change to a green background when it is hovered, we can reach right for the transition property to move between background colors.

To create a transition effect, you must specify two things:

- the CSS property you want to add an effect to
- the duration of the effect

If the duration part is not specified, the transition will have no effect, because the default value is 0.

Properties of CSS Transition

- transition: A shorthand property for setting the four transition properties into a single property
- transition-delay: Specifies a delay (in seconds) for the transition effect
- transition-duration: Specifies how many seconds or milliseconds a transition effect takes to complete
- transition-property: Specifies the name of the CSS property the transition effect is for
- transition-timing-function: Specifies the speed curve of the transition effect

Example

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
  width: 100px;
```

```
  height: 100px;
  background: red;
  transition-property: width;
  transition-duration: 2s;
  transition-timing-function: linear;
  transition-delay: 1s;
}
div:hover {
  width: 300px;
}
</style>
</head>
<body>
<h1>The transition Properties Specified One by One</h1>
<p>Hover over the div element below, to see the transition effect:</p>
<div></div>
</body>
</html>
```

6.6 CSS Measurement Units

CSS measurement units are different ways in CSS to express the length. A whitespace cannot appear between the number and the unit. However, if the value is 0, the unit can be omitted. There are two types of CSS measurement units: They are: Absolute and Relative.

Absolute Measurement Units

Absolute units are used to measure fixed length. They are not relative to anything else and are considered to be of the same size. These measurement units are not recommended because of the variation of screen size. These units are recommended only when the output medium is known, such as for printing.

Some of the absolute units are as follows:

- px: Pixel is a small unit but still visible and is relative to the viewing device.
- cm: Centimeters, 1cm=37.8px

- mm: millimeters, is a smaller unit as compared to cm, 1 mm=1/10th of 1cm.
- in: inches, is a large unit compared to pixels, 1in=96px
- pc: pica is a typographic unit, 1pc=1/6th of 1in
- pt: Points is the smallest unit of measure, 1pt=1/72 of 1in

Example

```
<!DOCTYPE html>
<html>
<head>
<style>
body{
text-align: center;
}
</style>
</head>
<body>
<h1> Absolute units </h1>
<p style = "font-size: 20px;" > It has a font-size: 20px; </p>
<p style = "font-size: 1.2cm;" > It has a font-size: 1.2cm; </p>
<p style = "font-size: .7in;" > It has a font-size: .7in; </p>
<p style = "font-size: 18pt;" > It has a font-size: 18pt; </p>
<p style = "font-size: 2pc;" > It has a font-size: 2pc; </p>
<p style = "font-size: 10mm;" > It has a font-size: 10mm; </p>
</body>
</html>
```

Relative Measurement Units

Relative measurement units are used to measure the length which is related to another length. Relative units are good for styling the responsive site because they scale relative to the window size or the parent. They specify the length, which is related to another length property.

Depending on the device, if the size of the screen varies too much, then the relative length units are the best because they scale better between the different rendering mediums. We can use the relative units as the default for the responsive units. It helps us to avoid updating styles for different screen sizes.

Some of the relative units are as follows:

- **rem:** This unit is relative to the font size of the root element.
- **em:** This unit is relative to the font size of the element itself. For example, 3em signifies that the new size will be 3 times the size of the current font.
- **%:** This denotes a percentage value, and defines the size of an element relative to its parent's element.
- **ex:** This unit is relative to the x-height of the font in use.
- **vh:** The viewport height, denotes the element's height is 1% of the viewport's height.
- **vw:** The viewport width, denotes the element's width is 1% of the viewport's width.
- **vmin:** This denotes the size relative to 1% of the viewport's smaller dimension.
- **vmax:** This denotes the size relative to 1% of the viewport's larger dimension.

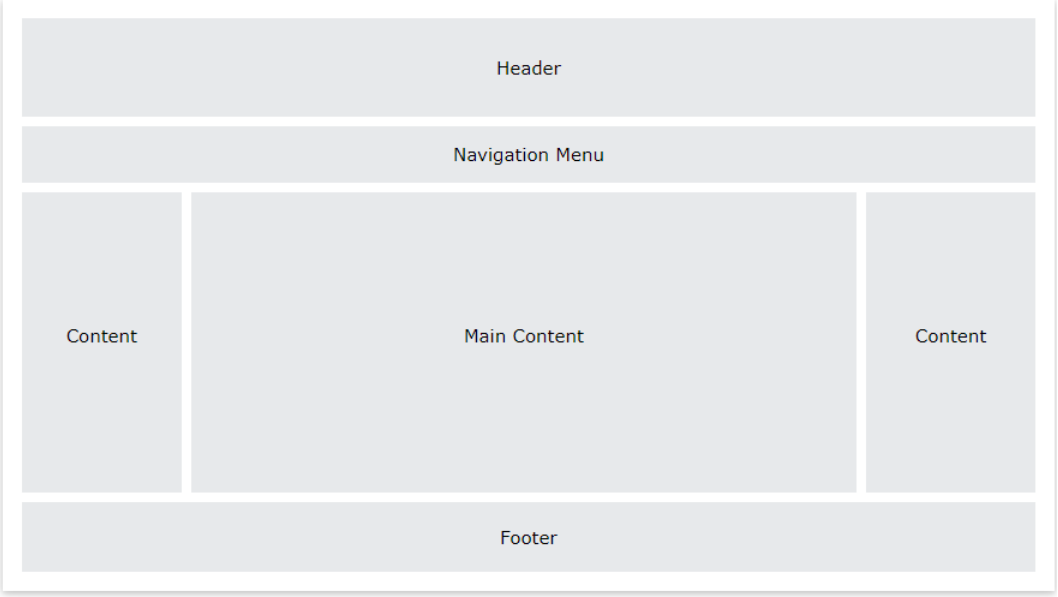
Example

```
<!DOCTYPE html>
<html>
<head>
<style>
body{
text-align: center;
}
p{
line-height: 0.1cm;
color: blue;
}
</style>
</head>
<body>
```

```
<h1> Relative units </h1>
<p style = "font-size: 2em;" > It has a font-size: 2em; </p>
<p style = "font-size: 8ex;" > It has a font-size: 8ex; </p>
<p style = "font-size: 6ch;" > It has a font-size: 6ch; </p>
<p style = "font-size: 4rem;" > It has a font-size: 4rem; </p>
<p style = "font-size: 4vw;" > It has a font-size: 4vw; </p>
<p style = "font-size: 10vh;" > It has a font-size: 10vh; </p>
<p style = "font-size: 10vmin;" > It has a font-size: 10vmin; </p>
<p style = "font-size: 8vmax;" > It has a font-size: 8vmax; </p>
<p style = "font-size: 400%;" > It has a font-size: 400%; </p>
</body>
</html>
```

6.7 CSS Website Layout

CSS Website Layout describes how to correctly arrange your boxes in relation to the viewport as well as to one another. A website is often divided into headers, menus, content and a footer. In general, a website would look like as in figure below:



Header

A header is usually located at the top of the website. It often contains a logo or the website

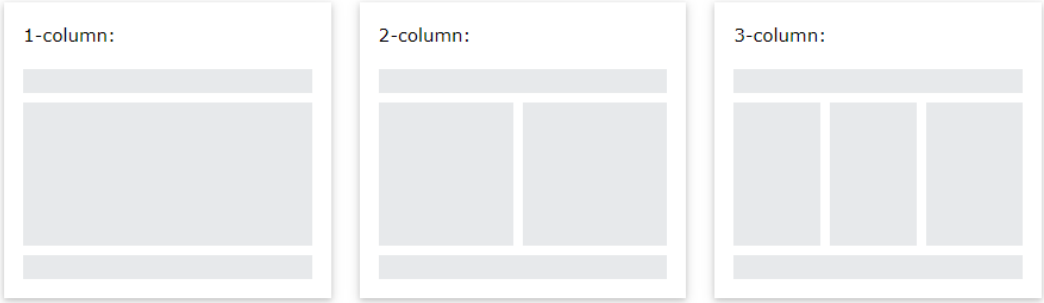
name.

Navigation Bar

A navigation bar contains a list of links to help visitors navigating through your website

Content

The layout in this section, often depends on the target users. The most common layout is of the following:



Footer

The footer is placed at the bottom of web page. It often contains information like copyright and contact info

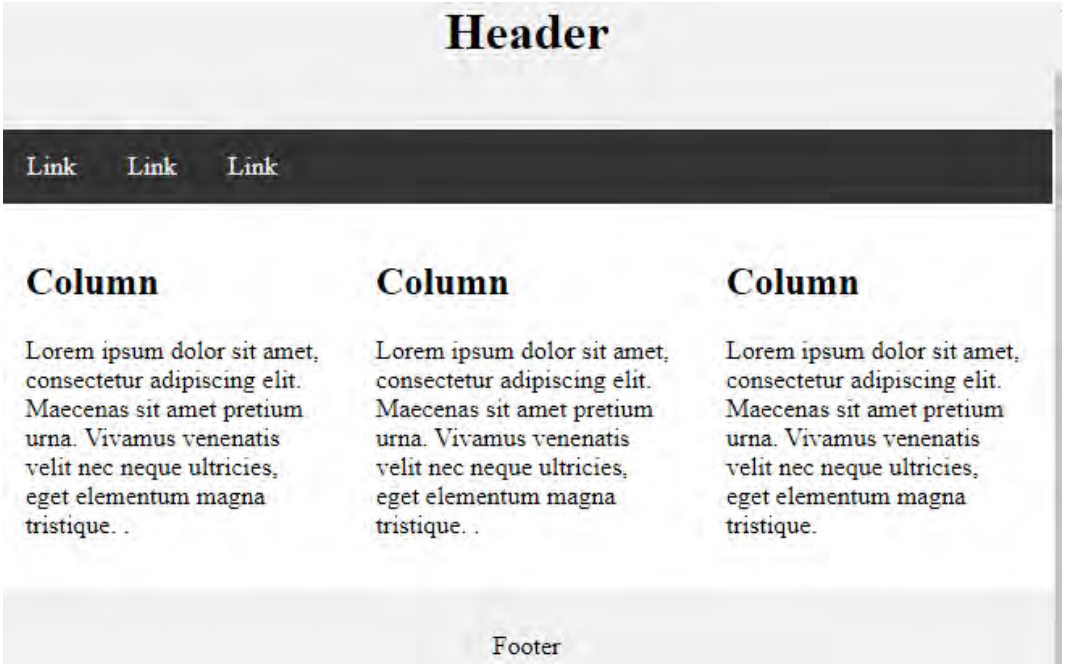
Example

```
<!DOCTYPE html>
<html lang="en">
<head>
<title>CSS Website Layout</title>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<style>
* {
  box-sizing: border-box;
}
body {
  margin: 0;
}
```

```
/* Style the header */
.header {
  background-color: #f1f1f1;
  padding: 20px;
  text-align: center;
}

/* Style the top navigation bar */
.topnav {
  overflow: hidden;
  background-color: #333;
}

/* Style the topnav links */
.topnav a {
  float: left;
  display: block;
  color: #f2f2f2;
  text-align: center;
  padding: 14px 16px;
  text-decoration: none;
}

/* Change color on hover */
.topnav a:hover {
  background-color: #ddd;
  color: black;
}
```

```
/* Create three equal columns that floats next to each other */
.column {
  float: left;
  width: 33.33%;
  padding: 15px;
}

/* Clear floats after the columns */
.row:after {
  content: "";
  display: table;
  clear: both;
}

/* Style the footer */
.footer {
  background-color: #f1f1f1;
  padding: 10px;
  text-align: center;
}
</style>
</head>
<body>

<div class="header">
  <h1>Header</h1>
 </div>

<div class="topnav">
  <a href="#">Link</a>
```

```
  <a href="#">Link</a>
  <a href="#">Link</a>
</div>

<div class="row">
  <div class="column">
    <h2>Column</h2>
     <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Maecenas sit amet pretium urna. Vivamus venenatis velit nec neque ultricies, eget elementum magna tristique. .</p>
  </div>

  <div class="column">
    <h2>Column</h2>
     <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Maecenas sit amet pretium urna. Vivamus venenatis velit nec neque ultricies, eget elementum magna tristique. .</p>
  </div>

    <div class="column">
    <h2>Column</h2>
     <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Maecenas sit amet pretium urna. Vivamus venenatis velit nec neque ultricies, eget elementum magna tristique. </p>
  </div>
</div>
<div class="footer">
  <p>Footer</p>
</div>
</body>
</html>
```


Header
Link
Link
Link
Column
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Maecenas sit amet pretium urna. Vivamus venenatis velit nec neque ultricies, eget elementum magna tristique. .
Column
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Maecenas sit amet pretium urna. Vivamus venenatis velit nec neque ultricies, eget elementum magna tristique. .
Column
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Maecenas sit amet pretium urna. Vivamus venenatis velit nec neque ultricies, eget elementum magna tristique.
Footer

Exercise

Choose the correct answer from the given alternatives.

1. CSS is the acronym for

   a. Cascade style sheets
   b. Color and style sheets
   c. Cascading style sheets
   d. Cascade and Style Sheet

2. Which of the following is the correct syntax for referring the external style sheet?

   a. <style src = example.css>
   b. <style src = “example.css” >
   c. <link rel=”stylesheet” type=”text/css” href=”example.css”>
   d. <stylesheet> example.css </stylesheet>

3. The CSS property used to control the element’s font-size is -

   a. Text-style b. Text-size c. Font-size d. Font-style

4. Which of the following property is used as the shorthand property for the border properties?

   a. Border-all b. Border-top c. Border d. Border-lrtb

5. Which of the following syntax is correct in CSS to make each word of a sentence start with a capital letter?

   a. Text-style: capital
   b. Transform: capitalize
   c. Text-transform: capital
   d. Text-transform: capitalize;

6. The CSS property used to draw a border around element is?

   a Line b Border c Border-line d Outline

7. The correct syntax to give a line over text is -

   a. Text-decoration: line-through
   b. Text-decoration: none
   c. Text-decoration: overline
   d. Text-decoration: underline

Write short answers to the following questions.

1. Describe CSS with its syntax.
2. What is Selector in CSS?

3. Explain CSS Margin and Padding.
4. How do you style the background of element at its individual side differently? Explain with example.
5. Explain CSS height and width properties.

Write long answer to the following questions.

1. Explain the types of CSS with example code.
2. Describe CSS navigation bar.
3. Explain the Box Model of CSS.
4. How do you style your text with CSS Text properties? Explain
5. Explain any four CSS advance properties.
6. Explain the general CSS Website Layout.
7. What are the CSS measurement units? Explain.
8. Explain CSS Box Model.
9. What is CSS selector? Describe different types of CSS selectors with examples.

Unit 7 JavaScript Fundamentals

7.1 Introduction

JavaScript is a scripting programming language that makes webpages lively and interactive. It was invented by Brendan Eich in 1995. (co-founder of the Mozilla project, the Mozilla Foundation, and the Mozilla Corporation). It appeared in Netscape Navigator browser for the first time. The language was initially called LiveScript and was later renamed JavaScript. JavaScript and Java are not same. Java is very complex programming language whereas JavaScript is only a scripting language. The syntax of JavaScript is mostly influenced by the programming language C. Typically, JavaScript is used with HTML and CSS to enhance a web page's functionality, such as validating forms, creating interactive maps, and displaying animated charts. JavaScript is widely also being used in game development and Mobile application development.

Features of JavaScript

- JavaScript is supported by almost all popular web browsers.
- JavaScript can also be referred as structured programming language as it follows the syntax and structure of C programming language.
- JavaScript is an object-oriented programming.
- It is a light-weighted and interpreted language.
- It is a case-sensitive language.
- JavaScript is supported by several operating systems including, Windows, macOS, etc.
- It provides good control to the users over the web browsers.

7.2 JavaScript in Different Browser

Now a days, almost all web pages contain JavaScript, a scripting programming language that runs on visitor's web browser. It makes web pages functional for specific purposes and if disabled for some reason, the content or the functionality of the web page can be limited or unavailable.

The instructions to enable (activate) JavaScript in five most commonly used browsers are described below.

Google Chrome

- On the web browser menu click on the "Customize and control Google Chrome" and select "Settings".
- In the "Settings" section click on the "Show advanced settings..."
- Under the "Privacy" click on the "Content settings...".
- When the dialog window opens, look for the "JavaScript" section and select "Allow all sites to run JavaScript (recommended)".
- Click on the "OK" button to close it.
- Close the "Settings" tab.
- Click on the "Reload this page" button of the web browser to refresh the page.

Internet Explorer

- On web browser menu click "Tools" icon and select "Internet Options".
- In the "Internet Options" window select the "Security" tab.
- On the "Security" tab click on the "Custom level..." button.
- When the "Security Settings - Internet Zone" dialog window opens, look for the "Scripting" section.
- In the "Active Scripting" item select "Enable".
- When the "Warning!" window pops out asking "Are you sure you want to change the settings for this zone?" select "Yes".
- In the "Internet Options" window click on the "OK" button to close it.
- Click on the "Refresh" button of the web browser to refresh the page.

Internet Explorer < 9

- On web browser menu click "Tools" and select "Internet Options"
- In the "Internet Options" window select the "Security" tab.
- On the "Security" tab click on the "Custom level..." button.
- When the "Security Settings - Internet Zone" dialog window opens, look for the "Scripting" section.

- In the "Active Scripting" item select "Enable".
- When the "Warning!" window pops out asking "Are you sure you want to change the settings for this zone?" select "Yes".
- In the "Internet Options" window click on the "OK" button to close it.
- Click on the "Refresh" button of the web browser to refresh the page.

Mozilla Firefox

- In the address bar, type about:config and press Enter.
- Click "I'll be careful, I promise" if a warning message appears.
- In the search box, search for javascript.enabled
- Toggle the "javascript.enabled" preference (right-click and select "Toggle" or double-click the preference) to change the value from "false" to "true".
- Click on the "Reload current page" button of the web browser to refresh the page.

Opera

- Click on Opera icon "Menu" and than "Settings".
- 2. Click on "Websites" and then choose "Allow all sites to run JavaScript (recommended)"
- 3. Click on the "Reload" button of the web browser to refresh the page.

Apple Safari

- On the web browser menu click on the "Edit" and select "Preferences".
- In the "Preferences" window select the "Security" tab.
- In the "Security" tab section "Web content" mark the "Enable JavaScript" checkbox.
- Click on the "Reload the current page" button of the web browser to refresh the page.

7.3 JavaScript in Html Documents

The <script> tag is used to add JavaScript code in HTML documents.

Syntax:

<script type=”text/javascript” >

--- JavasScript code here.

</script>

Example

- The script tag specifies that we are using JavaScript.
- The text/javascript is the content type that provides information to the browser about the data.

```
<html>
<head>
        <title>My First JavaScript code!!!</title>
        <script type=”text/javascript”>
                alert(“Hello World!”);
        </script>
</head>
<body>
</body>
</html>
```

Note: type=”text/javascript” is not necessary in HTML5. Simply:

```
<script > alert(“Hello World!”);
</script>
```

There are three places in HTML where we can put JavaScript Code.

1. Between the body(<body>..</body>) tag of html
2. Between the hea (<head>..</head>) tag of html
3. In external file (with .js extension)

Between the body(<body>..</body>) tag of html

The JavaScript is placed with body tag if the script is to be run on the page loads so that the script generates content in the page.

Example

```
<html>
<head>
<title>Javacript in body section</title>
</head>
<body>
<script type=”text/javascript”>
  document.write(“Hello World”);
</script>
</body>
</html>
```

Between the Head (<head>..</head>) tag of HTML

The JavaScript is place within head tag if the script is to be run on some event, such as

when a user clicks somewhere, or key press etc.

Example

```
<html>
<head><title>JavaScript in head section</title>
<script type=”text/javascript”>
function ShowHelloWorld(){
 alert(“Hello World!!!”);
}
</script>
</head>
<body>
<p>Welcome to World</p>
<form>
<input type=”button” value=”click” onclick=”ShowHelloWorld()”/>
</form>
</body>
</html>
```

In External File (with .js extension)

The JavaScript code can be written in separate file with .js extension and then embed it in HTML file by using script tag and src attribute. The path of external JavaScript file is used as value of src attribute of script tag as in following example.

Example.js

```
function ShowHelloWorld(){
 alert(“Hello World!!!”);
}
```

Example.html

```
<html>
<head>
<script type=”text/javascript”  src=”example.js”></script>
```

```
</head>
<body>
<p>Welcome to JavaScript</p>
<form>
<input type=”button” value=”click” onclick=” ShowHelloWorld()”/>
</form>
</body>
</html>
```

7.4. Variables and Datatypes

Variables

The variables in JavaScript are the name of the memory location that stores value. The value stored in variable is changeable.

Syntax to define variable in JavaScript

```
var  variable_name;
```

OR

```
var variable_name = value;
```

Syntax Description

- var is the keyword used to define variable.
- variable_name is the name give to variable.
- value is the initial value assigned to variable.

Example

```
var number = 1;
```

Rules to define variables

- Name must start with a letter (a to z or A to Z), underscore (_), or dollar($) sign.
- After first letter we can use digits (0 to 9), for example value1.
- JavaScript variables are case sensitive, for example x and X are different variables.

Correct JavaScript variables

```
var x = 10;
var _value="sonoo";
```

Incorrect JavaScript variables

```
var  123=30;
var *aa=320;
```

Types of Variable

There are two types of variable in JavaScript: Local and Global.

1. Local Variable

A variable declared inside the function is known as global variable. It can't be used outside the block or function.

Example

```
<script>
function  display(){
var n=10;//local variable
}
</script>
```

2. Global Variable

A variable declared outside the function is known as global variable. It is is accessible from any function.

Example

```
<script>
var n=200;//global variable
function display(){
document.writeln(n);
}
function buffer(){
document.writeln(n);
}
display ();//calling JavaScript function
buffer();
</script>
```

Example showing both local and global JavaScript variables

```
<!DOCTYPE html>
<html>
<body>
        <h1>Demo: JavaScript Global and Local Variables </h1>
        <script>
                var greet = “Hello “ // global variable
                function myfunction(){
                        var msg = “JavaScript!”; //local variable
                        alert(greet + msg); // access global and local variable
                }
                myfunction();
                alert(greet);//can access global variable
                alert(msg); //error: can’t access local variable
    </script>
</body>
</html>
```

Datatypes

Datatypes are used to define the types of the data and value that can a variable can hold. In JavaScript, there are 2 types of datatypes.

1. Primitive datatype
2. Non-primitive datatype

Primitive Datatype

There are five types of primitive data types in JavaScript. They are as follows:

Data Type	Description
String	represents sequence of characters e.g. “hello”
Number	represents numeric values e.g. 100
Boolean	represents boolean value either false or true

Undefined	represents undefined value
Null	represents null i.e. no value at all

Non Primitive Datatype

There are three types of non-primitive datatype. They are given below:

Data Type	Description
Object	represents instance through which we can access members
Array	represents group of similar values
RegExp	represents regular expression

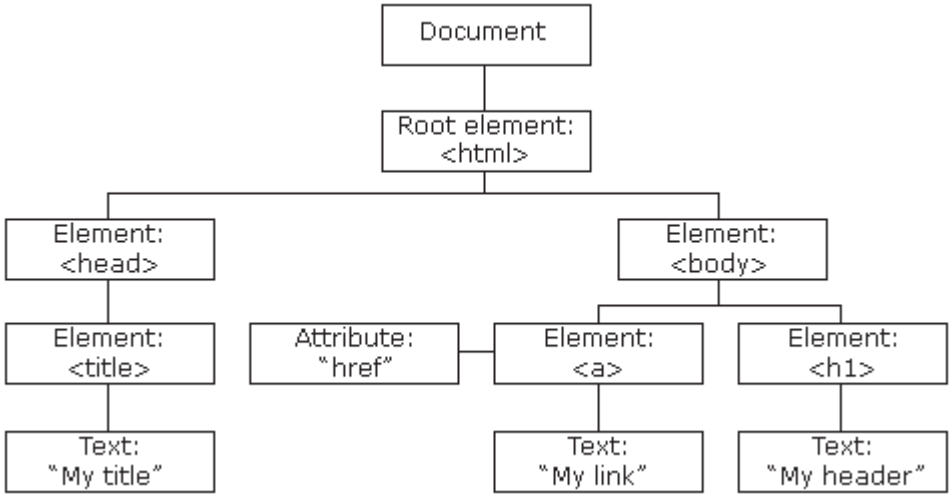
7.5 JavaScript HTML DOM

The HTML DOM (Document Object Model) is a programming API for documents. The document object represents the whole html document. When a web page is loaded, the browser creates a Document Object Model of the page. By the help of document object, we can add dynamic content to our web page.

The HTML DOM is an API (Programming Interface) for JavaScript:

- JavaScript can add/change/remove HTML elements
- JavaScript can add/change/remove HTML attributes
- JavaScript can add/change/remove CSS styles
- JavaScript can react to HTML events
- JavaScript can add/change/remove HTML events

The HTML DOM model is constructed as a tree of Objects:



JS HTML DOM Methods

HTML DOM methods are actions you can perform on HTML Elements. Using these methods, the values of the properties of the HTML elements can be changed or set.

Example

```
<!DOCTYPE html>
<html>
<body>
<h2>My First Page</h2>
<p id="para"></p>
<script>
document.getElementById("para ").innerHTML = "Hello JavaScript!";
</script>
</body>
</html>
```

In the example above, get element by Id is a method, while inner HTML is a property.

JavaScript HTML DOM Document and Elements

The HTML DOM document object is the owner of all other objects in the web page. In order to access any other element in HTML, the document object is accessed at first.

These are some ways to use document object to access the HTML elements

Finding HTML Elements

Method	Description
document.getElementById(id)	Find an element by element id
document.getElementsByTagName(name)	Find elements by tag name
document.getElementsByClassName(name)	Find elements by class name

Changing HTML Elements

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

Adding and Deleting Elements

Method	Description
document.createElement(element)	Create an HTML element
document.removeChild(element)	Remove an HTML element
document.appendChild(element)	Add an HTML element
document.replaceChild(new, old)	Replace an HTML element
document.write(text)	Write into the HTML output stream

Adding Events Handlers

Method	Description
document.getElementById(id).onclick = function(){code}	Adding event handler code to an on click event

JavaScript HTML DOM Node Lists

A Node List object is a collection of nodes (DOM elements) taken from the HTML

document. A node list can be live or static, that means the changes to the DOM are either applied automatically to the collection or don't affect the elements of the collection at all. The querySelectorAll() method returns a static NodeList, while the Node.child Nodes property returns a live NodeList. The nodes can be accessed by an index number.

Example

```
<!DOCTYPE html>
<html>
<body>
<p>Hello World!</p>
<p>The DOM is very useful!</p>
<p id="demo"></p>
<script>
var myNodelist = document.getElementsByTagName("p");
document.getElementById("demo").innerHTML ="The innerHTML of the second paragraph is: " +myNodelist[1].innerHTML;

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

Node List Length

The length property defines the number of nodes in a node list.

Example

```
<!DOCTYPE html>
<html>
<body>
<p>Hello World!</p>
<p>How many paragraphs are in this document?</p>
<p>This example demonstrates the length property of a novelist.</p>
<p id="demo"></p>
<script>
```

```
var myNodelist = document.getElementsByTagName("p");
document.getElementById("demo").innerHTML = myNodelist.length;
</script>
</body>
</html>
```

7.6 Control Flow Statement

Control flow statements are used to control the flow of execution in a script. Statements are separated by semicolon (;). JavaScript statement constitutes the JavaScript code which is translated by the browser line by line.

Conditional Statement

Statements that are used to execute a set of statements based on a condition are known as conditional statements. It is also known as a selection statement. In JavaScript selection statements are: if, if-else, and switch.

if statement

The if statement is used to specify a block of JavaScript code to be executed if a condition is true.

Syntax:

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

Example

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

if-else statement

The if-else statement is used to specify a block of JavaScript code to be executed if a condition is true and execute the next block of code if the condition is false.

Syntax:

```
if(expression)
{
   statement(s);
}
else
{
   statement(s);
}
```

Example

```
<script type="text/javascript">
  <!--
     var age = 20;
     if( age > 18 ){
        document.write("<b>valid age for driving</b>");
     }
else{
        document.write("<b>not valid age for driving</b>");

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

Switch Case Statement

A block of statements, in which the execution of code depends upon different cases. The interpreter checks each case against the value of the expression until a match is found. If nothing matches, a default condition will be used.

Syntax

```
switch(expression)
{
  case label1:
    statement(s);
    break;
  case label2:
    statement(s);
    break;
  ...
  default:
    statement(s);
}
```

Example

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

```
        default:  document.write("Unknown grade<br/>")
      }
      document.write("Exiting switch block");
    //-->
</script>
```

Loop Statement

Statements that are used to execute a set of statements repeatedly based on a condition are known as loop statements or iteration statements. Some of the loop statements supported by JavaScript are for, while, do-while.

For Loop

A for loop will execute a code block several times. It contains initialization, condition, and increment or decrement in a single line.

Syntax

```
for (initialize; condition; increment/decrement)
{
//code block to be executed
}
```

With initialize, it starts the loop, here a declared variable is used. Then the exit condition for the loop is checked in the condition part. When this condition returns true, the code block inside is executed. When, in case, the condition returns false or fails, it goes to the increment/decrement part, and the variable is assigned an updated value. Values are updated until the condition is satisfied.

Example

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

```
    document.write("Loop stopped!");
  </script>
```

While Loop

In the while loop, a statement or code block is executed repeatedly as long as the expression is true. Once the expression becomes false, the loop will be exited.

Syntax

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

Example

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

Do while Loop

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

Syntax

```
do{
  Statement(s) to be executed;
}
<script type="text/javascript">
    var count = 0;
```

```
document.write("Starting Loop" + "<br/>");
do{
  document.write("Current Count : " + count + "<br/>");
  count++;
}while (count < 0);
document.write("Loop stopped!");
</script>
```

7.7 Functions

A function is a block of code that can be executed again and again. Functions make the code modular which improves maintenance. In JavaScript, a function keyword is used to declare or define a function.

Syntax

```
function functionname (p1, p2) {
statements;
return; //This is optional
}
```

- To define a function use the function keyword followed by the function name, followed by parentheses ().
- In parenthesis, we define parameters or attributes.
- The group of reusable statements (code) is enclosed in curly braces {}. This code is executed whenever a function is called.

Example

```
<html>
  <body>
    <button onclick=" this.innerHTML=Date()">The time is?</button>
    <p>Click to display the date.</p>
    <button onclick="displayDate()"
    <script>
      function displayDate() {
        document.getElementById("demo").innerHTML = Date();
```

Hello World!

The DOM is very useful!

The innerHTML of the second paragraph is: The DOM is very useful!

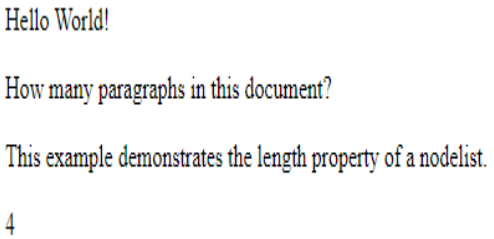
```
    }</script>
    <p id="demo"></p>
  </script>
 </body>
</html>
```

7.8 JavaScript Popup Boxes (Prompt, Confirm, Alert)

JavaScript provides built-in global functions to display messages to users for different purposes, e.g., displaying a simple message displaying message and taking the user's confirmation or displaying a popup to take the user's input value.

Prompt

Prompt Box is used to take the user's input to dc
up, the user will have to click either "OK" or "
value. If the user clicks "OK" the box returns t
the box returns null. The built-in prompt() functi



Syntax

prompt([string message], [string defaultValue]);

The prompt() function takes two string parameters. The first parameter is the message to be displayed, and the second parameter is the default value which will be in input text when the message is displayed.

Example

```
<!DOCTYPE html>
<html>
<body>
        <h1>Example of prompt</h1>
        <p id="msg"></p>

        <script>
                var age = prompt("Please enter preferred tenure in years", "15");

                document.getElementById("msg").innerHTML = "You have
```

```
entered " + age + " years";
   </script>
</body>
</html>
```

Confirm

The confirm box is used to take the user's confirmation to proceed on the webpage. For example, the user's confirmation needs to be taken before saving updated data or deleting existing data. The built-in function confirm() is used for taking those data and proceeding. When a confirm box pops up, the user will havepop-upick either "OK" or "Cancel" to proceed. If the user clicks "OK", the box returns true. If the user clicks "Cancel", the box returns false.

Syntax

window.confirm("some text") or confirm(“sometext”);

Example

```
<!DOCTYPE html>
<html>
<body>
        <h1>Example of confirm</h1>
        <p id="msg"></p>
        <script>
                var userChoice;

                if (window.confirm("Do you want to save changes?") == true) {
                        userChoice = "Data saved successfully!";
                } else {
                        userChoice = "Save Canceled!";
                }
                document.getElementById("msg").innerHTML = userChoice;
   </script>
</body>
```

```
</html>
```

Alert

The alert box is used to display the alert messages to the webpage visitors. When an alert box pops up, the user will have to click "OK" to proceed.

Syntax

window.alert("sometext"); or alert(“sometext”);

Example

```
<!DOCTYPE html>
<html>
<body>
<h2>JavaScript Alert</h2>
<button onclick="myFunction()">Try it</button>
<script>
function myFunction() {
  alert("Hello JavaScript!");
}
</script>
</body>
</html>
```

7.9 JavaScript Objects

JavaScript object is a non-primitive data-type that allows you to store multiple collections of data. JavaScript object is an unordered collection of key-value pairs. Each key-value pair is called a property. The key of a property can be a string. The value of a property can be any value, e.g., a string, a number, an array, and even a function.

The syntax to declare an object is:

```
const object_name = {
  key1: value1,
  key2: value2
```

```
}
Example
const student = {
   name: ‘John’,
   class: 9;
};
```

Here, an object name student is defined. name and class are keys, and John and 9 are values respectively.

Accessing Object Properties

You can access the value of a property by using its key. There are two ways to perform this operation.

1. Using dot Notation

```
Syntax:
objectName.key
Example
<html>
<body>
<script>
const student = {
   name: 'John',
   class: 9
};
document.write(student.name+" "+ student.class);
</script>
</body>
</html>
```

2. Using bracket Notation

```
Syntax:
objectName[“propertyName”]
```

Example

```
<html>
<body>
<script>
const student = {
   name: ‘John’,
   class: 9
};
document.write(student[“name”] +” “+ student[“class”]);
</script>
</body>
</html>
```

Exercise

Choose the correct answer from the given alternatives.

1. JavaScript is an ….language.

   a. Object-Oriented b. Procedural

   c. Structured d. Object-Based.

2. Which of the following is the correct syntax for referring the external JavaScript?

   a. <script type=”text/javascript” src=”example.js” ></script>

   b. <style src = “example.css” >

   c. <script>example.js</script>

   d. None of above.

3. Which of the following keywords is used to define a variable in JavaScript?

   a. Int b. Var c. Let d. Const

4. How do you write “Hello World” in an alert box?

   a. AlertBox(“Hello World”); b. Alert(“Hello World”);

   c. MsgBox(“Hello World”); d. Msg(“Hello World”);

5. How do you create a function in JavaScript?

   a. Function myFunction() b. Function:myFunction()

   c. Function = myFunction() d. None of above.

6. Where is the correct place to insert a JavaScript?

   a. Both the <head> section and the <body> section are correct

   b. The <head> section

   c. The <body> section

   d. None of above.

Write short answers to the following questions.

1. What is JavaScript?
2. List out the datatypes in JavaScript. Explain any two of them.
3. List out the features of JavaScript.

4. What is JavaScript HTML DOM.? Describe in short.
5. How do you declare a function in JavaScript? Explain with an example.

Write long answers to the following questions.

1. Explain different loop statements in JavaScript.
2. What do you define JavaScript object? How can you access it in the program? Explain with an example.
3. Explain the pop-up boxes available in JavaScript with examples.
4. Describe the JavaScript variable with its type.
5. How do you enable JavaScript in the Mozilla Firefox browser? Write down the steps.

Project works

1. Prepare a chart for different types of variables and datatypes in JavaScript.
2. Make a presentation on the topic: Importance of JavaScript in Website.
3. Implement pop up boxes of JavaScript in the simple webpage.

Bibliography

"Web Development and Design Foundations with HTML5", 10th Edition by Felke-Morris

"Learning SQL" By Alan Beaulieu

"SQL: The Ultimate Beginners Guide: Learn SQL Today" By Steve Tale

"SQL: QuickStart Guide — The Simplified Beginner's Guide To SQL' By Clydebank Technology

Mark Mayers, A Smarter Way to Learn JavaScript

Elizabeth Robinson, Head First JavaScript Programming: A Brain-Friendly Guide

References

Book References

Robbins, Jennifer Niederst – Learning Web Design, O'Reilly Media.

Duckett, Jon – HTML and CSS: Design and Build Websites, Wiley.

Powell, Thomas A. – Web Design: The Complete Reference, McGraw-Hill.

"Web Development and Design Foundations with HTML5", 10th Edition by Felke-Morris

Web References

https://www.canva.com

https://www.w3schools.com

https://en.wikipedia.org

https://www.geeksforgeeks.orgc

https://www.tutorialspoint.com

https://www.squarespace.com

https://www.figma.com