Introduction to HTML

* HTML Stands for Hyper Text Markup Language
* HTML used to develop static web pages
* Current version of HTML is HTML5.X
* HTML Released by "Tim Berners Lee" in 1991
* we will execute HTML with the help of "Browsers" Ex.

Google Chrome Mozilla

Opera IE

Safari

Netscape Navigator

* IDE is used to develop the software applications Ex.

Notepad Edit Plus Notepad++

Visual Studio Code Eclipse

* Visual Studio Code is Recommended IDE to develop web applications
* Visual Studio Code provided by "Microsoft" and "open-source IDE"
* CSS Stands for Cascading Style Sheet
* CSS used to apply styles to web pages Ex.

color

background-color margin

padding border border-radius

* Current version of CSS is CSS3.X
* Extension of CSS files is ".CSS"
* JavaScript is the Scripting Language
* JavaScript used to develop Dynamic Web Pages
* JavaScript also used to implement the Forms Validations
* Current version of JavaScript is ES13
* ES Stands for ECMA Script
* Extension for JavaScript files is ".js"
* HTML is TAG Based Markup Language Ex.

<table></table>

<h1></h1>

<a></a>

<br>

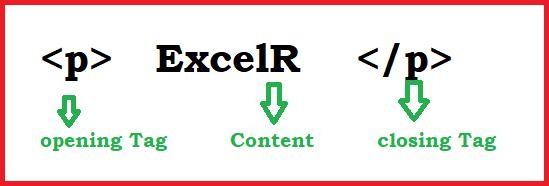
<img>

<p></p>

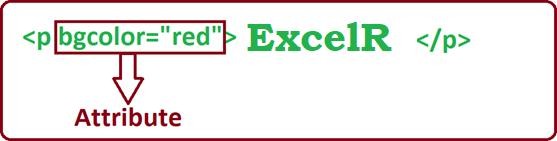
<title></title>

<head></head>

* TAGS Divided into two types
  1. container/paired tags
  2. non container tags/non paired tags
* container/paired tags contains both opening and closing tag
* non container tags/non paired tags contain only opening tag
* closing tag should contain "/"
* combination of opening tag, content and closing tag called as HTML Element



* Attributes enhances html element functionality
* Attributes are key and value pairs
* key and value separated by using "="



Features of HTML5.X

* WebSocket

WebSocket used to implement the chat applications

WebSocket available in HTML5.X

* Web Worker

Web Worker used to run JavaScript in Background Web Worker also available in HTML5.X

* Storages API

HTML5.X Supports two types of Storages

* 1. Local Storage
  2. Session Storage
* whenever we close the browser/opens the new tab we won’t lost the data from Local Storage
* whenever we close the browser/opens the new tab we will lose the data from Session Storage
* Geolocation API

it is used to find the current location of user/device Geolocation API also available in HTML5.X

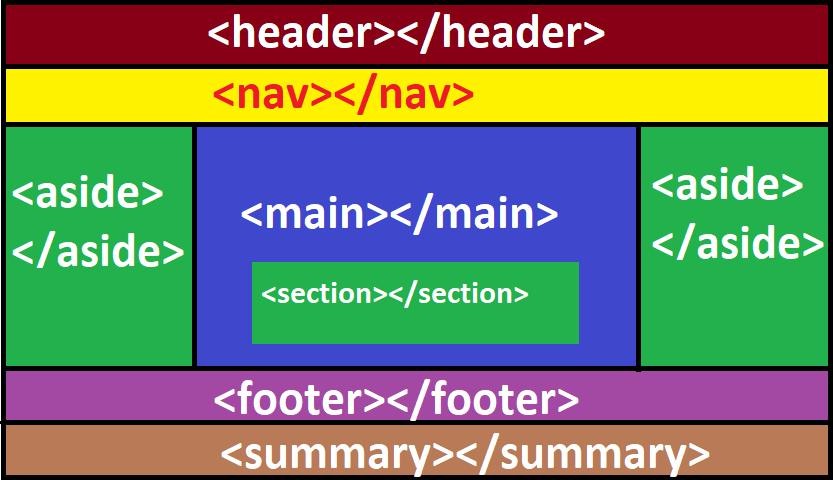
* Drag & Drop API

This API helps to Drag and Drop HTML Elements This API also available in HTML5.X

* Semantic Elements

HTML5.X Released New Elements/Semantic Elements

Semantic Elements increases application readability

Ex.

<header></header> <nav></nav>

<main></main> <section></section> <footer></footer>

<summary></summary>

<audio></audio>

<video></video> **Structure of Webpage** DOCTYPE

ROOT SECTION

HEAD SECTION

METADATA BODY SECTION

MAIN CONTENT

* DOCTYPE Representing version of HTML
* Browsers only understands the DOCTYPE
* Below DOCTYPE Representing HTML5.X

<!DOCTYPE html>

* <html></html> tag used to create the ROOT SECTION
* ROOT SECTION Divided into two sections
  1. HEAD SECTION
  2. BODY SECTION
* <head></head> tag used to create the HEAD SECTION
* <body></body> tag used to create the BODY SECTION
* Dataabout webpage called as METADATA Ex.

Author Description Title

* we will define METADATA under HEAD SECTION
* we will define Main Content under BODY SECTION Ex.

Tables Forms Headings Paragraphs Images

<!DOCTYPE html>

<html>

<head>

//METADATA

</head>

<body>

//MAIN CONTENT

</body>

</html>

# Software Installation

1. download and install Visual Studio Code
   * Visual Studio Code is the IDE.
   * Visual Studio Code provided by "Microsoft" and "open source".
   * Visual Studio Code is "recommended" to develop "web applications"

website: https://code.visualstudio.com/docs/?dv=winfile: VSCodeUserSetup-x64-1.77.3.exe

1. install Live Server plugin
   * Live Server plugin watches application changes
   * Live Server plugin gives "changes notification" to browser
   * browser will "reload/refresh" automatically

### open the "VSCode" ==> Click "Extensions" ==> Search "Live

**Server" ==>Select "Live Server" ==> Click "install"**

1. install "Dummy Text Generator" Plugin
   * "Dummy Text Generator" Plugin generates content randomly

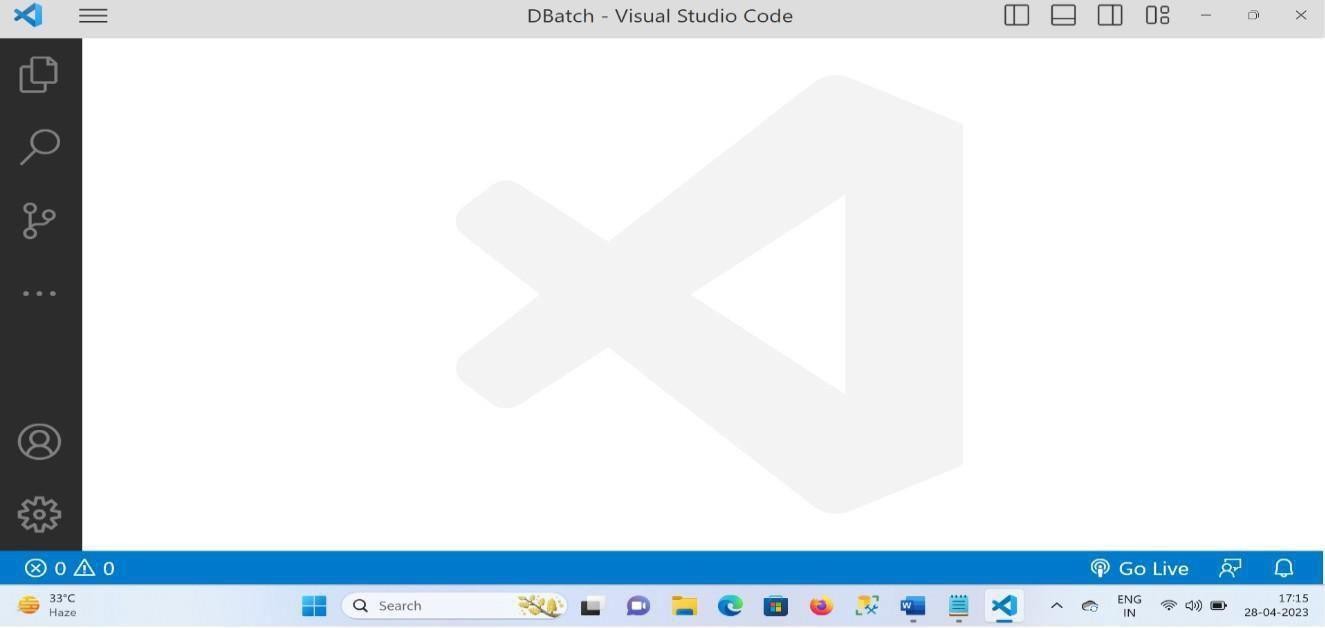
### open the "VSCode" ==> click "Extensions" ==> Search "Dummy

**Text Generator" ==> select "Dummy Text Generator" ==> click "install"**

1. Enable "AutoSave" Settings
   * Application will save Automatically without "ctrl+s"

### open the "VSCode" ==> click the "Manage" ==> click "settings"

**==> choose "AutoSave" ==> select "afterDelay"**

1. install browser
   * Google Chrome
   * Firefox

### Practice Test Paper

1. HTML Stands for? HyperText Markup Language
2. What is current version of HTML?

The current version of HTML is HTML5. HTML5 is the fifth major revision of the HTML standard, and it was published in 2014 by the World Wide Web Consortium.

1. IDE Stands for? Integrated Development Environment
2. Which IDE Is Suitable IDE to develop web applications?

There are many Integrated Development Environments (IDEs) available for web development, each with its own strengths and weaknesses. Here are some popular IDEs that are suitable for web development:

* 1. Visual Studio Code (VS Code)
  2. Sublime Text 3.Atom 4.WebStorm 5.NetBeans 6.Eclipse

1. CSS Stans for? **CSS (Cascading Style Sheets)**
2. Why CSS? Write minimum 3 points?

CSS (Cascading Style Sheets) is a crucial part of web development, and here are three reasons

* 1. Control over Layout and Design
  2. Improved User Experience
  3. Separation of Concerns

1. What is the Current version of CSS?

Current Version of CSS:-The current version of CSS is CSS3. However, it's worth noting that CSS is a living standard, and new features are being added to the language all the time.

CSS Levels:-The World Wide Web Consortium (W3C) has defined several levels of CSS, each of which adds new features and functionality to the language. The current levels of CSS are:

* CSS1: The first version of CSS, which was published in 1996.
* CSS2: The second version of CSS, which was published in 1998.
* CSS2.1: An updated version of CSS2, which was published in 2004.
* CSS3: The current version of CSS, which is a modularized version of the language.

CSS3 Modules:-CSS3 is a modularized version of the language, which means that it is divided into several modules, each of which defines a specific set of features and functionality. Some of the most commonly used

CSS3 modules include:

* CSS3 Selectors: Defines new selectors for selecting elements on a web page.
* CSS3 Box Model: Defines the box model, which is used to layout elements on a web page.
* CSS3 Color: Defines new color models and color functions.
* CSS3 Fonts: Defines new font properties and values.
* CSS3 Text: Defines new text properties and values.

CSS4:-There is no official version of CSS4, but the W3C is working on a new version of the language, which is currently known as CSS4. CSS4 is expected to include new features such as:

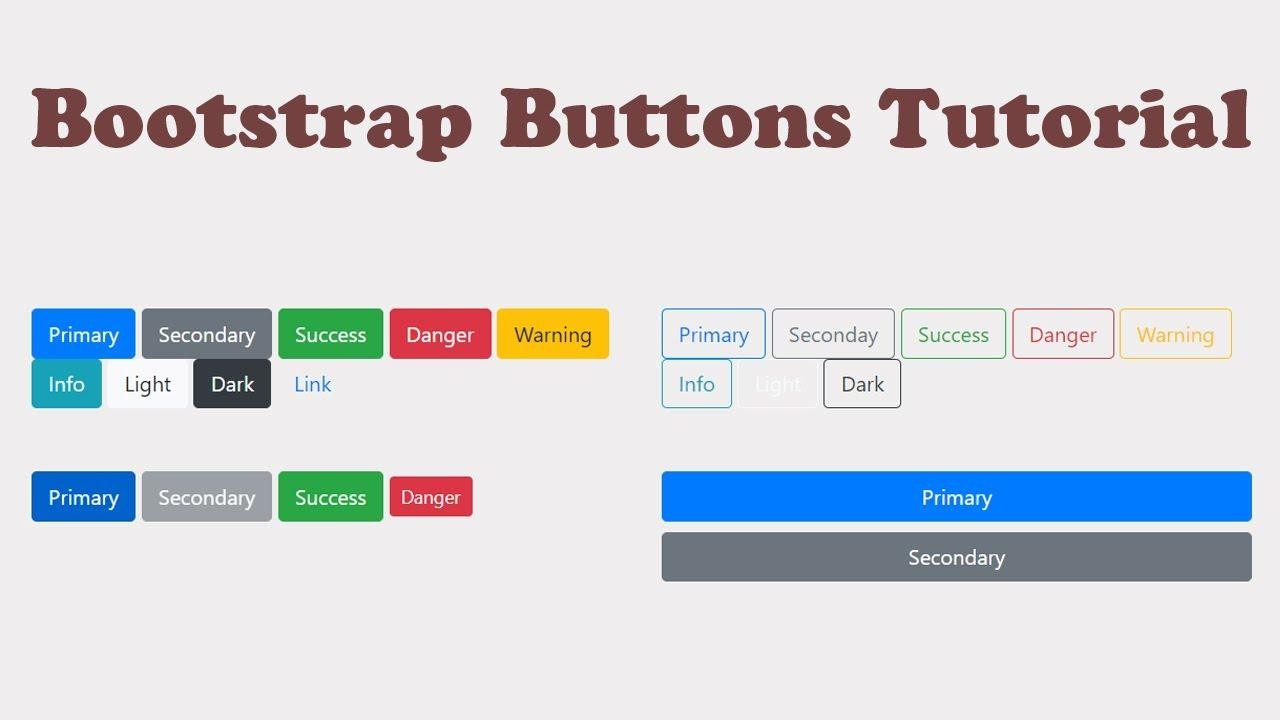
* Improved support for grid layouts
* Improved support for flexbox layouts
* New properties and values for styling elements
* Improved support for accessibility

1. What is Bootstrap?

Bootstrap is a popular, free, and open-source front-end development framework used for creating responsive and mobile-first websites and web applications. It's a collection of per-written HTML, CSS, and JavaScript code that provides a set of tools for building responsive designs , layouts, and UI

components. Bootstrap is widely used in web development due to its ease of use, flexibility, and

extensive community support.

Example:-

1. What is Angular?

Angular is a popular, open-source, and widely-used JavaScript framework used for building dynamic, single-page applications (SPAs) and complex web applications. It's a full-fledged framework that provides a robust set of tools for building scalable, maintainable, and efficient applications. Angular is built and maintained by Google and is widely adopted in the industry due to its powerful features, strong ecosystem, and large community support

`10) What is React?

React is a popular, open-source, and widely-used JavaScript library used for building user interfaces and single-page applications (SPAs). It's a view layer library that provides a efficient and flexible way to render components and manage state changes in web applications. React is maintained by Facebook and is widely adopted in the industry due to its simplicity, flexibility, and strong ecosystem

11)Explain MEAN Stack?

The MEAN Stack is a popular technology stack used for building web applications. MEAN is an acronym that stands for:

* M - MongoDB: A NoSQL database used for storing and managing data.
* E - Express.js: A lightweight and flexible Node.js web framework used for building web applications.
* A - Angular: A JavaScript framework used for building dynamic, single-page applications (SPAs) and complex web applications.
* N - Node.js: A JavaScript runtime environment used for running server-side JavaScript code.

The MEAN Stack is a full-stack solution that provides a complete set of tools for building web applications, from the database layer to the client-side user interface.

Here's a brief overview of each component:

* ​

MongoDB:

A NoSQL database that allows for flexible schema design and scalable data storage.

* Express.js: A Node.js framework that provides a lightweight and flexible way to build web applications, with features like routing, middleware, and template engines.
* Angular: A JavaScript framework that provides a robust set of tools for building dynamic, single-page applications (SPAs) and complex web applications.
* Node.js: A JavaScript runtime environment that allows developers to run server-side JavaScript code, providing a scalable and efficient way to build web applications.

The MEAN Stack is often used for building complex, data-driven applications, such as:

* Real-time analytics and monitoring systems
* Social media platforms
* E-commerce applications
* Online marketplaces
* Complex web applications

Some of the key benefits of using the MEAN Stack include:

* Flexibility: The MEAN Stack provides a flexible and scalable solution for building web applications.
* Efficiency: Node.js and Express.js provide a fast and efficient way to build web applications.
* Scalability: MongoDB provides a scalable solution for storing and managing large amounts of data.
* Easy to learn: The MEAN Stack is built on JavaScript, making it easy for developers to learn and adapt.
* Large community: The MEAN Stack has a large and active community, providing many resources for learning and troubleshooting.

12 ) Explain MERN Stack?

The MERN Stack is a popular technology stack used for building web applications. MERN is an acronym that stands for:

* M - MongoDB: A NoSQL database used for storing and managing data.
* E - Express.js: A lightweight and flexible Node.js web framework used for building web applications.
* R - React: A JavaScript library used for building user interfaces and single-page applications (SPAs).
* N - Node.js: A JavaScript runtime environment used for running server-side JavaScript code. The MEAN Stack is often used for building complex, data-driven applications, such as:
* Real-time analytics and monitoring systems
* Social media platforms
* E-commerce applications
* Online marketplaces
* Complex web applications

Some of the key benefits of using the MEAN Stack include:

* Flexibility: The MERN Stack provides a flexible and scalable solution for building web applications.
* Efficiency: Node.js and Express.js provide a fast and efficient way to build web applications.
* Scalability: MongoDB provides a scalable solution for storing and managing large amounts of data.
* Easy to learn: The MEAN Stack is built on JavaScript, making it easy for developers to learn and adapt.
* Large community: The MEAN Stack has a large and active community, providing many resources for learning and troubleshooting.

1. Explain MEVN Stack?

The MEVN Stack is a popular technology stack used for building web applications. MEVN is an acronym that stands for:

* M - MongoDB: A NoSQL database used for storing and managing data.
* E - Express.js: A lightweight and flexible Node.js web framework used for building web applications.
* V - Vue.js: A progressive and flexible JavaScript framework used for building user interfaces and single- page applications (SPAs).
* N - Node.js: A JavaScript runtime environment used for running server-side JavaScript code.

The MEVN Stack is a full-stack solution that provides a complete set of tools for building web applications, from the database layer to the client-side user interface.

Here's a brief overview of each component:

* MongoDB: A NoSQL database that allows for flexible schema design and scalable data storage.
* Express.js: A Node.js framework that provides a lightweight and flexible way to build web applications, with features like routing, middleware, and template engines.
* Vue.js: A JavaScript framework that provides a progressive and flexible way to build user interfaces and single-page applications, with features like a robust ecosystem, robust state management, and a strong focus on developer experience.
* Node.js: A JavaScript runtime environment that allows developers to run server-side JavaScript code, providing a scalable and efficient way to build web applications.

The MEVN Stack is often used for building complex, data-driven applications, such as:

* Real-time analytics and monitoring systems
* Social media platforms
* E-commerce applications
* Online marketplaces
* Complex web applications

Some of the key benefits of using the MEVN Stack include:

* Flexibility: The MEVN Stack provides a flexible and scalable solution for building web applications.
* Efficiency: Node.js and Express.js provide a fast and efficient way to build web applications.
* Scalability: MongoDB provides a scalable solution for storing and managing large amounts of data.
* Easy to learn: The MEVN Stack is built on JavaScript, making it easy for developers to learn and adapt.
* Large community: The MEVN Stack has a large and active community, providing many resources for learning and troubleshootin

1. Write Features of HTML5.X?

HTML5 is a significant update to the HTML standard, and it has introduced many new features and improvements. Here are some of the key features of HTML5:

New Semantic Elements

* <header>: Defines the header section of a document or section
* <nav>: Defines a section of navigation links
* <main>: Defines the main content section of a document
* <section>: Defines a self-contained section of related content
* <article>: Defines an independent piece of content, such as a blog post or news article
* <aside>: Defines a piece of content that is related to the surrounding content
* <footer>: Defines the footer section of a document or section Multimedia
* <video>: Allows for the embedding of video content without the need for third-party plugins
* <audio>: Allows for the embedding of audio content without the need for third-party plugins
* <canvas>: Provides a way to create dynamic graphics and animations using JavaScript

Offline Storage

* localStorage: Allows web applications to store data locally on the client-side
* sessionStorage: Allows web applications to store data locally on the client- side for a single session

Geolocation

* geolocation: Allows web applications to access the user's location using GPSor other location services

Web Workers

* web workers: Allow web applications to run scripts in the background, improving performance and responsiveness

Web Sockets

* web sockets: Allow web applications to establish a persistent, bi-directional communication channel with the server

New APIs

* File API: Allows web applications to access and manipulate files on the client-side.
* Drag and Drop API: Allows web applications to implement drag-and-drop

functionality.

* History API: Allows web applications to manipulate the browser's history stack.
* Microdata API: Allows web applications to add semantic meaning to HTML elements.

Improved Error Handling

* Improved error handling and debugging tools, including the console API and the debugger keyword

New Attributes and Properties

* placeholder attribute: Allows developers to specify a placeholder text for

form inputs

* required attribute: Allows developers to specify that a form input is required
* pattern attribute: Allows developers to specify a regular expression pattern

for form inputs

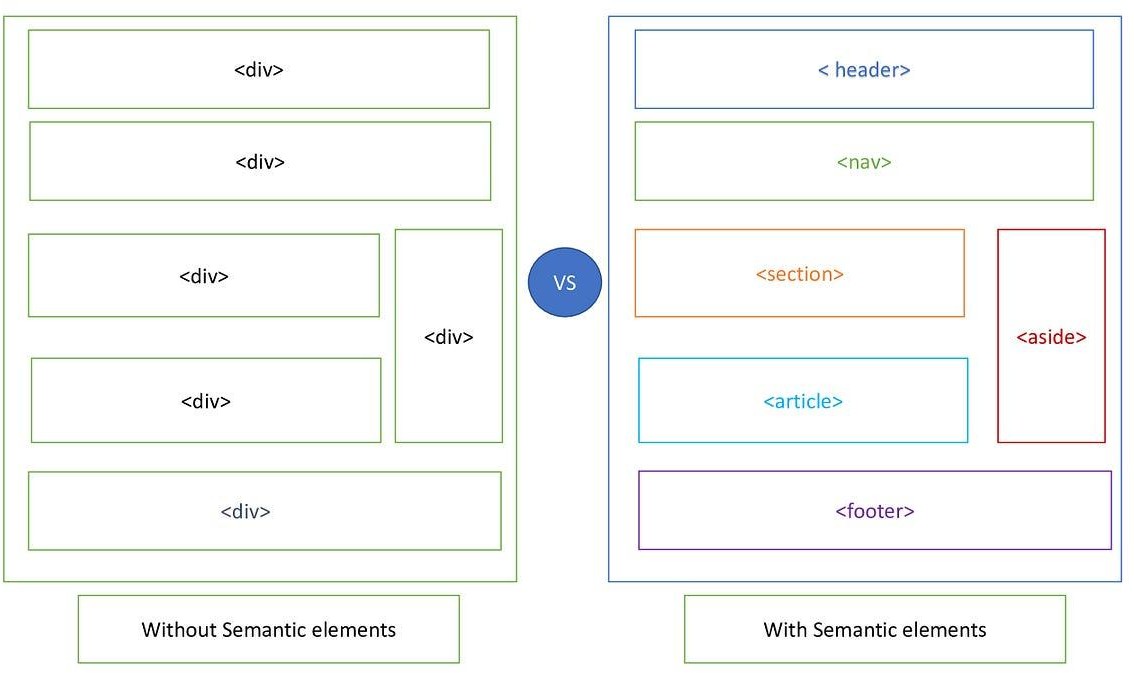
* autofocus attribute: Allows developers to specify that a form input should

receive focus automatically

Removed Elements

* <acronym>: Replaced by <abbr>
* <applet>: Replaced by <object>
* <basefont>: Removed, as it is no longer necessary
* <big>: Removed, as it is no longer necessary
* <center>: Removed, as it is no longer necessary
* <font>: Removed, as it is no longer necessary
* <strike>: Removed, as it is no longer necessary
* <tt>: Removed, as it is no longer necessary
* <u>: Removed, as it is no longer necessary

These are just some of the key features of HTML5. There are many more improvements and additions to the standard, and it continues to evolve with new features and capabilities being added all the time.

1. Explain Semantic Elements in HTML5 with Diagram?

Here's a brief description of each semantic element:

* <header>: Defines the header section of a document or section.
* <nav>: Defines a section of navigation links.
* <main>: Defines the main content section of a document.
* <section>: Defines a self-contained section of related content.
* <article>: Defines an independent piece of content, such as a blog post or news article.
* <aside>: Defines a piece of content that is related to the surrounding content.
* <footer>: Defines the footer section of a document or section.

Example Code:-

<!DOCTYPE html>

<html>

<head>

<title>Example Web Page</title>

</head>

<body>

<header>

<h1>Site Title</h1>

<nav>

<ul>

<li><a href="#">Home</a></li>

<li><a href="#">About</a></li>

<li><a href="#">Contact</a></li>

</ul>

</nav>

</header>

<main>

<section>

<h2>Section Title</h2>

<p>This is a self-contained section of related content.</p>

</section>

<article>

<h2>Article Title</h2>

<p>This is an independent piece of content, such as a blog post or news article.</p>

</article>

<aside>

<h2>Related Content</h2>

<p>This is a piece of content that is related to the surrounding content.</p>

</aside>

</main>

<footer>

<p>&copy; 2023 Example Web Page</p>

</footer>

</body>

</html>

1. How to play audio in HTML5 ?

<audio id="audio">

<source src="audio.mp3" type="audio/mp3"> Your browser does not support the audio element.

</audio>

<button onclick="playAudio()">Play</button>

<button onclick="pauseAudio()">Pause</button>

<button onclick="setVolume()">Set Volume</button>

<script>

function playAudio() {

var audio = document.getElementById("audio"); audio.play();

}

function pauseAudio() {

var audio = document.getElementById("audio"); audio.pause();

}

function setVolume() {

var audio = document.getElementById("audio"); audio.volume = 0.5;

}

</script>

1. How to play Video in HTML5 In this example:

* The <video> element is used to define the video file.
* The <controls >attribute is used to display the video controls, such as play, pause, and volume.
* The <source> element is used to specify the video file to play.
* The <src> attribute specifies the URL of the video file.
* The <type> attribute specifies the MIME type of the video file. Supported Video Formats

HTML5 supports the following video formats:

* MP4 (MPEG-4)
* WebM (WebM Video)
* OGG (Ogg Theora)

<video id="video">

<source src="video.mp4" type="video/mp4"> Your browser does not support the video element.

</video>

<button onclick="playVideo()">Play</button>

<button onclick="pauseVideo()">Pause</button>

<button onclick="setVolume()">Set Volume</button>

<script>

function playVideo() {

var video = document.getElementById("video"); video.play();

}

function pauseVideo() {

var video = document.getElementById("video"); video.pause();

}

function setVolume() {

var video = document.getElementById("video"); video.volume = 0.5;

}

</script>

1. What is attribute write few points?

Attributes in HTML

In HTML, an attribute is a piece of information that is added to a tag to provide more context or functionality to the element. Attributes are added to the opening tag of an element and are used to specify additional information about the element.

Here are a few key points about attributes:

* + Attribute syntax: Attributes are added to the opening tag of an element using the syntax attribute="value".

Attribute types: There are several types of attributes, including:

* Boolean attributes: These attributes do not require a value and are simply included in the tag to enable a particular feature. For example, disabled or readonly.
* Enumerated attributes: These attributes can only take on a specific set of values. For example, type="text" or type="password".
* Free-form attributes: These attributes can take on any value. For example, class="header" or style="color: red;".

Common attributes: Some common attributes include:

* id: Used to specify a unique identifier for an element.
* class: Used to specify a class or group of elements.
* style: Used to specify inline styles for an element.
* title: Used to specify a tooltip or title for an element.
* alt: Used to specify alternative text for an image.

Attribute values: Attribute values can be enclosed in single quotes or double quotes. For example, class='header' or class="header".

Attribute order: The order of attributes in an HTML tag does not matter. Examples of Attributes

Here are a few examples of attributes in use:

* + <p id="header" class="large">This is a paragraph of text.</p>
  + <img src="image.jpg" alt="An image of a cat" title="Cat picture">
  + <input type="text" name="username" placeholder="Enter your username">
  + <a href="https://[www.example.com](http://www.example.com/)" target="\_blank">Visit example.com</a>

19 )Write basic Structure of Web Pages

Basic Structure of Web Pages

A basic web page structure consists of several key elements that work together to create a functional and visually appealing website. Here is a breakdown of the basic structure of a web page:

HTML Document

* The HTML document is the foundation of a web page and contains all the content and structure of the page.
* It is written in HTML (Hypertext Markup Language) and is saved with a .html or .htm file extension. Head Section
* The head section is the first section of the HTML document and contains metadata about the page.
* It includes information such as the page title, character encoding, and links to external stylesheets or scripts.

Body Section

* The body section is the main content area of the HTML document and contains all the visible content of the page.
* It includes headings, paragraphs, images, links, and other elements that make up the page's content.

Header Section

* The header section is typically located at the top of the page and contains the site's logo, navigation menu, and other key elements.

Footer Section

* + The footer section is typically located at the bottom of the page and contains information such as copyright notices, contact information, and social media links.

Content Section

* + The content section is the main area of the page where the majority of the content is displayed.
  + It can include a variety of elements such as text, images, videos, and interactive elements. Sidebar Section
  + The sidebar section is a secondary content area that is typically located on the left or right side of the page.
  + It can include additional information such as navigation menus, advertisements, or other secondary content.

Footer Widgets

* + Footer widgets are small blocks of content that are typically located in the footer section of the page.
  + They can include information such as social media links, contact information, or other secondary content.

Example:- Basic HTML Structure

<!DOCTYPE html>

<html>

<head>

<title>Page Title</title>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<link rel="stylesheet" href="styles.css">

</head>

<body>

<header>

<nav>

<ul>

<li><a href="#">Home</a></li>

<li><a href="#">About</a></li>

<li><a href="#">Contact</a></li>

</ul>

</nav>

</header>

<main>

<section>

<h1>Welcome to our website!</h1>

<p>This is a sample web page.</p>

</section>

</main>

<footer>

<p>&copy; 2023 Our Company</p>

</footer>

</body>

</html>

20 ) What DOCTYPE?

DOCTYPE is a declaration that defines the document type and version of HTML, helping browsers render the page correctly. It is the first line of code required in every HTML or XHTML document and is not an HTML tag, but rather an "information" to the browser about what document type to expect.

Types of DOCTYPE Declarations

* + In HTML 4.01, the DOCTYPE declaration refers to a document type definition (DTD), which defines the structure and legal elements of an XML document.
  + In HTML5, the DOCTYPE declaration is much simpler and does not require a reference to DTDs. Examples of DOCTYPE Declarations
  + HTML5: <!DOCTYPE html>
  + HTML 4.01 Strict: <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"

"<http://www.w3.org/TR/html4/strict.dtd>">

* + HTML 4.01 Transitional: <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "<http://www.w3.org/TR/html4/loose.dtd>">
  + HTML 4.01 Frameset: <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN" "<http://www.w3.org/TR/html4/frameset.dtd>">

Importance of DOCTYPE

* The DOCTYPE declaration is important because it helps browsers interpret and render the HTML document consistently according to the specified HTML version.
* Without a DOCTYPE declaration, a web page may not render correctly across all browsers and could lead to inconsistent display issues.

Case Sensitivity

The DOCTYPE declaration is not case-sensitive, but it's common practice to use uppercase for the keyword.

21 ) Write DOCTYPE for HTML5.X

For HTML5.x, the DOCTYPE declaration is:

<!DOCTYPE html>

This is the simplest and most widely used DOCTYPE declaration, which tells the browser to render the HTML document according to the HTML5 standard.

22 ) What is METADATA?

Metadata is "data that provides information about other data". In the context of HTML, metadata refers to the information about a web page that is not displayed directly to the user but is used by search engines, browsers, and other web crawlers to understand the content and structure of the page

23 ) Write the Differences Between Container TAGS and Non-Container TAGS?

In HTML, tags can be classified into two categories: container tags and non-container tags. Container Tags

* + A container tag, also known as a paired tag, is a tag that has both an opening tag and a closing tag.
  + The content of the tag is placed between the opening and closing tags.
  + Container tags are used to wrap around content to apply styles, formatting, or other effects. Examples of container tags include:
* <p>This is a paragraph of text</p>
* <div>This is a division of content</div>
* <ul><li>Item 1</li><li>Item 2</li></ul>

Non-Container Tags

* A non-container tag, also known as a singleton tag or self-closing tag, is a tag that does not have a closing tag.
* Non-container tags are used to insert a single element or piece of content into a document. Examples of non-container tags include:
* <img src="image.jpg" alt="An image">
* <br>
* <hr>

Key Differences

* + Presence of Closing Tag: Container tags have a closing tag, while non-container tags do not.
  + Content: Container tags wrap around content, while non-container tags do not contain content.
  + Usage: Container tags are used to apply styles or formatting to content, while non-container tags are used to insert a single element or piece of content.

24 )How to develop static web pages ?

How to Develop Static Web Pages:- Developing static web pages involves creating HTML, CSS, and JavaScript files that are stored on a server and served directly to clients without any server-side processing. Here's a step-by-step guide to develop static web pages:

Step 1: Plan Your Website

1. Define the purpose and scope of your website.
2. Identify your target audience and their needs.
3. Create a sitemap and wireframe to visualize your website's structure and layout. Step 2: Choose a Text Editor or IDE
   * Select a text editor (e.g., Notepad++, Sublime Text) or an Integrated Development Environment (IDE) (e.g., Visual Studio Code, Atom) to write your code
   * Familiarize yourself with the editor or IDE's features and shortcuts Step 3: Write HTML Code

* Create a new file with an .html extension (e.g., index.html)
* Write HTML code to define the structure and content of your web page
* Use semantic HTML elements to improve accessibility and search engine optimization (SEO) Step 4: Add CSS Styles
* Create a new file with a .css extension (e.g., styles.css)
* Write CSS code to define the layout, colors, and typography of your web page
* Use CSS selectors to target specific HTML elements and apply styles Step 5: Add JavaScript (Optional)
* Create a new file with a .js extension (e.g., script.js)
* Write JavaScript code to add interactivity to your web page (e.g., animations, form validation)
* Use JavaScript libraries or frameworks (e.g., jQuery, React) to simplify development Step 6: Add Images and Other Assets
* Create a folder to store images, fonts, and other assets
* Use the <img> tag to add images to your web page
* Use CSS to style and position images Step 7: Test and Debug
* Open your HTML file in a web browser to test your web page
* Use the browser's developer tools to debug and inspect your code
* Test your web page on different devices and browsers to ensure compatibility Step 8: Deploy Your Website
* Upload your HTML, CSS, and JavaScript files to a server or hosting platform (e.g., GitHub Pages, Netlify)
* Use a File Transfer Protocol (FTP) client or a deployment tool (e.g., Git) to upload your files
* Make sure your website is accessible and functional

Best Practices

* Use a consistent naming convention and folder structure
* Write clean, readable, and modular code
* Use version control systems (e.g., Git) to track changes and collaborate with others
* Optimize images and compress files to improve page load times

1. What is the Extension for HTML Pages?

The standard file extension for HTML pages is .html. This extension is used to indicate that the file contains HTML code, which is used to structure and format content on the web.

Other Extensions

While .html is the most common extension for HTML files, there are a few other extensions that can be used:

* .htm: This extension is also used for HTML files, but it's less common than .html.
* .xhtml: This extension is used for XHTML files, which are a stricter version of HTML that follows XML syntax rules.
* .shtml: This extension is used for Server-Parsed HTML files, which allow for server-side includes and other dynamic content.

1. What are the types of tags?

In HTML, tags are categorized into two main types: Structural Tags and Semantic Tags. Structural Tags

* These tags define the structure of a web page, such as headings, paragraphs, lists, and tables.
* They provide a basic layout and organization to the content. Examples of structural tags include:
* <h1>, <h2>, <h3>, etc. for headings
* <p> for paragraphs
* <ul>, <ol>, <dl> for lists
* <table> for tables Semantic Tags
  + These tags provide meaning to the content, such as defining the purpose of an element, like a navigation menu or a footer.
  + They help search engines and screen readers understand the content and its purpose. Examples of semantic tags include:
* <header> for the header section of a page
* <nav> for navigation menus
* <main> for the main content area
* <section> for a self-contained section of related content
* <article> for an independent piece of content, like a blog post
* <aside> for a piece of content that's related to the main content
* <footer> for the footer section of a page Other Types of Tags
* Inline Tags: These tags are used to format text or add functionality to a specific part of the content. Examples include <a>, <span>, <strong>, and <em>.
* Block Tags: These tags are used to define a block of content, such as a paragraph or a heading. Examples include <p>, <h1>, and <div>.
* Empty Tags: These tags do not have a closing tag and are used to add a single element to the page. Examples include <img>, <br>, and <hr>.

27 )What are container tags?

Container tags, also known as paired tags or block-level tags, are HTML tags that have both an opening- tag and a closing tag. These tags are used to wrap around content to apply styles, formatting, or other effects to the content.

Characteristics of Container Tags

* They have an opening tag and a closing tag.
* The content is placed between the opening and closing tags.
* They are used to group elements together to apply styles, formatting, or other effects.
* They can contain other HTML elements, including other container tags. Examples of Container Tags
* <p>This is a paragraph of text</p>
* <div>This is a division of content</div>
* <ul><li>Item 1</li><li>Item 2</li></ul>
* <table><tr><td>Cell 1</td><td>Cell 2</td></tr></table>
* <header>This is the header section</header>
* <main>This is the main content area</main>
* <section>This is a self-contained section of related content</section>

1. What are non-container Tags ?

Non-container tags, also known as empty tags or self-closing tags, are HTML tags that do not have a closing tag. These tags are used to add a single element to the page and do not wrap around any content. Characteristics of Non-Container Tags

* They do not have a closing tag.
* They are used to add a single element to the page.
* They do not wrap around any content.
* They are typically used for elements that do not require any content, such as images, links, or input fields.

Examples of Non-Container Tags

* <img src="image.jpg" alt="An image">
* <br>
* <hr>
* <input type="text" name="username">
* <link rel="stylesheet" type="text/css" href="style.css">
* <meta charset="UTF-8">
* <source src="video.mp4" type="video/mp4"> Commonly Used Non-Container Tags
* <img>: Used to add an image to the page.
* <br>: Used to add a line break to the page.
* <hr>: Used to add a horizontal rule to the page.
* <input>: Used to add an input field to the page.
* <link>: Used to link an external stylesheet or script to the page.
* <meta>: Used to add metadata to the page.
* <source>: Used to add a media source to the page.

1. What is HTML Element?

An HTML element is a building block of a web page, represented by a set of tags that surround and define a piece of content. HTML elements are the fundamental components of a web page, and they are used to create the structure, layout, and content of a web page.

Components of an HTML Element

An HTML element consists of three main components:

* Start Tag: The start tag is the opening tag that defines the element. It is denoted by a less-than symbol (<) followed by the element name and any attributes.
* Content: The content is the text or other elements that are contained within the element.
* End Tag: The end tag is the closing tag that marks the end of the element. It is denoted by a less-than symbol (<) followed by a forward slash (/) and the element name.

1. What is the extension of CSS files ?

The standard file extension for CSS (Cascading Style Sheets) files is .css. This extension is used to indicate that the file contains CSS code, which is used to style and layout web pages.

Other Extensions

While .css is the most common extension for CSS files, there are a few other extensions that can be used:

* + **.scss:** This extension is used for Sass (Syntactically Awesome StyleSheets) files, which are a preprocessor for CSS.
  + **.sass:** This extension is also used for Sass files.
  + **.less:** This extension is used for Less files, which are another preprocessor for CSS.
  + **.styl:** This extension is used for Stylus files, which are a preprocessor for CSS.

Example:- To link a CSS file to an HTML file, you can use the <link> tag in the <head> section of the HTML file.

**<head>**

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

**</head>**

1. What is the extension of JavaScript files ?

The standard file extension for JavaScript files is .js. This extension is used to indicate that the file contains JavaScript code, which is used to add interactivity and dynamic effects to web pages.

Other Extensions

While .js is the most common extension for JavaScript files, there are a few other extensions that can be used:

* + **.mjs:** This extension is used for ES6 modules, which are a new way of organizing and loading JavaScript code.
  + **.jsx:** This extension is used for JSX files, which are a syntax extension for JavaScript that allows you to write HTML-like code in your JavaScript files.
  + **.ts:** This extension is used for TypeScript files, which are a superset of JavaScript that adds optional static typing and other features.

External JavaScript Files

You can also link to external JavaScript files, such as libraries or frameworks, using the <script> tag.

**<head>**

**<script src="https://code.jquery.com/jquery-3.6.0.min.js" type="text/javascript"></script>**

**</head>**

How to Link a JavaScript File to an HTML File

To link a JavaScript file to an HTML file, you can use the <script> tag in the <head> or <body> section of the HTML file.

**<head>**

**<script src="script.js" type="text/javascript"></script>**

**</head>**

1. How to implement Forms Validations

Form validation is the process of checking the data entered by a user into a form to ensure that it meets certain criteria. This can include checking that the data is in the correct format, that it is within a certain range, or that it matches a specific pattern.

Here are the steps to implement form validations:

Step 1: Define the Validation Rules

* + Determine what data needs to be validated and what the validation rules are.
  + Decide whether the validation should be done on the client-side (using JavaScript) or on the server- side (using a programming language such as PHP or Python).

Step 2: Choose a Validation Method

* + There are several methods for validating forms, including:
  + Client-side validation: This involves using JavaScript to validate the form data before it is submitted to the server.
  + Server-side validation: This involves using a programming language to validate the form data after it has been submitted to the server.
  + Hybrid validation: This involves using a combination of client-side and server-side validation. Step 3: Implement Client-Side Validation
* Use JavaScript to validate the form data before it is submitted to the server.
* Use the onsubmit event to trigger the validation when the form is submitted.
* Use the preventDefault() method to prevent the form from being submitted if the validation fails. Step 4: Implement Server-Side Validation
* Use a programming language to validate the form data after it has been submitted to the server.
* Use a framework or library to simplify the validation process.
* Return an error message to the client if the validation fails. Step 5: Display Error Messages
  + Use JavaScript to display error messages to the user if the validation fails.
  + Use a library or framework to simplify the process of displaying error messages
  + How to execute HTML, CSS and JavaScript Example of Client-Side Validation

**// Get the form element**

**const form = document.getElementById('myForm');**

**// Add an event listener to the form's submit event form.addEventListener('submit', (e) => {**

**// Get the input elements**

**const nameInput = document.getElementById('name'); const emailInput = document.getElementById('email');**

**JavaScript**

**// Validate the input data**

**if (nameInput.value === '') { alert('Please enter your name'); e.preventDefault();**

**} else if (emailInput.value === '') { alert('Please enter your email address'); e.preventDefault();**

**}**

**});**

1. Write the Examples for Browsers? Desktop Browsers
   * **Google Chrome:** Developed by Google, Chrome is one of the most widely used browsers in the world. It is known for its speed, security, and extensive library of extensions.
   * **Mozilla Firefox:** Developed by Mozilla, Firefox is a popular browser that is known for its speed, security, and customization options.
   * **Microsoft Edge:** Developed by Microsoft, Edge is a browser that is designed to be fast, secure, and easy to use.
   * **Safari:** Developed by Apple, Safari is a browser that is designed for Mac and iOS devices. It is known for its speed, security, and integration with other Apple devices.
   * **Opera:** Developed by Opera Software, Opera is a browser that is known for its speed, security, and innovative features such as a built-in VPN and ad blocker.

Mobile Browsers

* **Google Chrome:** Chrome is also available on mobile devices and is one of the most widely used mobile browsers in the world.
* **Safari:** Safari is the default browser on Apple devices and is known for its speed, security, and integration with other Apple devices.
*  Firefox is also available on mobile devices and is known for its speed, security, and customization options.
* **Opera Mini:** Opera Mini is a mobile browser that is designed to be fast and efficient, even on slow internet connections.
* **UC Browser:** UC Browser is a popular mobile browser in Asia and is known for its speed, security, and innovative features such as a built-in download manager.

1. Write the Examples for IDE’S? Mobile App Development IDEs
   * **Xcode:** A free, proprietary IDE developed by Apple for iOS and macOS development. It's known for its comprehensive set of features, including code completion, debugging, and project management.
   * **Android Studio:** A free, open-source IDE developed by Google for Android development. It's known for its comprehensive set of features, including code completion, debugging, and project management.
   * **React Native:** A popular framework for building cross-platform mobile apps using JavaScript and React. 35 ) Who given VSCode?

Visual Studio Code (VS Code) was developed by Microsoft. Specifically, it was created by a team at Microsoft led by Erich Gamma, who is a Technical Fellow at Microsoft. Gamma is a well-known software engineer and researcher who has made significant contributions to the field of software development.

History of VS Code

VS Code was first released in 2015 as a preview version, and it was officially released in April 2016. Since then, it has become one of the most popular code editors in the world, known for its speed, flexibility, and extensibility.

Key Features of VS Code

VS Code is a lightweight, open-source code editor that is designed to be highly customizable and extensible. Some of its key features include:

* + **Syntax highlighting:** VS Code supports syntax highlighting for a wide range of programming languages.
  + **Code completion:** VS Code provides code completion suggestions as you type.
  + **Debugging:** VS Code has built-in debugging tools that allow you to set breakpoints, inspect variables, and step through your code.
  + **Extensions**: VS Code has a large collection of extensions that can be used to add new features

and functionality to the editor.

* + **Customization:** VS Code is highly customizable, allowing you to change the layout, theme, and behavior of the editor to suit your needs.

35 ) Which IDE Recommended to develop web applications?



The choice of IDE (Integrated Development Environment) for web application development depends on several factors such as the programming language, framework, and personal preferences. Here are some popular IDEs that are commonly used for web application development:

* + **Visual Studio Code (VS Code):** A lightweight, open-source code editor that is widely used for front-end development. It has a large collection of extensions available for various programming languages and frameworks.

For Front-end Development

* + **Sublime Text:** A popular, feature-rich text editor that is widely used for front-end development. It has a large collection of plugins available for various programming languages and frameworks.
  + **Atom:** A free, open-source text editor that is widely used for front-end development. It has a large collection of packages available for various programming languages and frameworks.

For Back-end Development

* **IntelliJ IDEA:** A commercial IDE that is widely used for back-end development. It supports a wide range of programming languages and frameworks, including Java, Python, and Ruby.
* **Eclipse:** A free, open-source IDE that is widely used for back-end development. It supports a wide range of programming languages and frameworks, including Java, Python, and Ruby.
*  A free, open-source IDE that is widely used for back-end development. It supports a

wide range of programming languages and frameworks, including Java, Python, and Ruby.

1. How to give line break?

There are several ways to give a line break, depending on the context and the formatting requirements. Here are a few common methods:

**HTML:**



**<p>This is a paragraph of text.<br>This is a new line of text.</p> CSS:-**



**p {**

**white-space: pre-wrap;**

**}**

1. How to draw Horizontal line in webpages ?

Drawing Horizontal Lines in Webpages

There are several ways to draw horizontal lines in webpages, depending on the desired appearance and the HTML and CSS versions being used. Here are a few common methods:



The <hr> tag is a simple way to draw a horizontal line in a webpage. It is a self-closing tag, meaning that it does not require a closing tag.

<hr>

By default, the <hr> tag will draw a horizontal line that spans the full width of the container element. You can customize the appearance of the line by adding attributes to the tag, such as width, size, and color.

1. Are comments executed by browsers? (yes/no)

No,Comments are not executed by browsers. They are ignored by the browser and are used to provide information to developers about the code.

1. Write the Syntax for HTML Comments?

HTML comments are not displayed in the browser, but they can help document your HTML source code.

You can add comments to your HTML source by using the following syntax:

<!-- Write your comments here -->

* + Headings in HTML
  + External CSS
  + Padding in CSS
  + Block Level Elements

**Headings in HTML**

# HEADINGS

* + - HTML Supports 6 types of Headings

<h1></h1> main heading

<h2></h2> sub heading

<h3></h3> ""

<h4></h4> ""

<h5></h5> ""

<h6></h6> ""

**External CSS**

* + - <link> tag, used to include the external CSS file
    - <link> tag is non paired tag
    - <link> tag supports two attributes
      1. href
      2. rel

**Padding**

* space around the content called as padding Ex.

padding:20px;

**Block Level Element**

* + Block Level Elements Starts with new line
  + Headings are Example for Block Level Elements

### headings.html

<!DOCTYPE html>

<html>

<head>

<title>Headings</title>

<link rel="stylesheet" href="headings.css">

</head>

<body>

<h1>ExcelR</h1>

<h2>ExcelR</h2>

<h3>ExcelR</h3>

<h4>ExcelR</h4>

<h5>ExcelR</h5>

<h6>ExcelR</h6>

</body>

</html>

**headings.css**

h1,h2,h3,h4,h5,h6{ color: red;

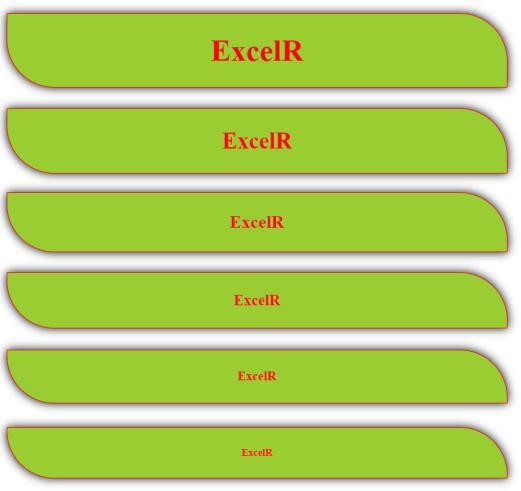
background-color: yellow; text-align: center; padding: 20px;

border: 1px solid red; width: 50%;

border-radius: 0px 60px;

box-shadow: 0px 0px 20px red;

}



**PRACTICE PAPER**

1. Write the Tags to create Headings in HTML Ans: HTML Heading Tags

In HTML, headings are created using the following tags:

* + h1: Main heading
  + h2: Subheading
  + h3: Sub-subheading
  + h4: Sub-sub-subheading
  + h5: Sub-sub-sub-subheading
  + h6: Sub-sub-sub-sub-subheading Example:-

<h1>Main Heading</h1>

<h2>Subheading</h2>

<h3>Sub-subheading</h3>

<h4>Sub-sub-subheading</h4>

<h5>Sub-sub-sub-subheading</h5>

<h6>Sub-sub-sub-sub-subheading</h6>

1. How to apply external CSS?

Ans: There are three ways to apply external CSS to an HTML document:

Applying External CSS

There are three ways to apply external CSS to an HTML document:

Linking to an External Stylesheet

You can link to an external stylesheet using the <link> tag in the <head> section of your HTML document. Example:-

**undefined**

<head> <link rel="stylesheet" type="text/css" href="styles.css"> </head> ``` In this example, `styles.css` is the name of the external stylesheet file.

1. What is padding in CSS?

In CSS, padding refers to the space between the content of an element and its border. It is the area between the content and the border of an element. Padding is used to create space between the content of an element and its border, making it look more visually appealing and easier to read.

Example: padding: 10px; Individual Padding Properties

You can also use individual padding properties to set the padding for each side of an element separately.

* + padding-top: sets the padding for the top side of an element
  + padding-right: sets the padding for the right side of an element
  + padding-bottom: sets the padding for the bottom side of an element
  + padding-left: sets the padding for the left side of an element padding-top: 10px;

padding-right: 20px; padding-bottom: 30px; padding-left: 40px;

Shorthand Padding Properties

You can also use shorthand padding properties to set the padding for multiple sides of an element at once.

* padding: top right bottom left;
* padding: top right bottom; (left padding is the same as right padding)
* padding: top bottom; (right and left padding are the same)
* padding: all; (all sides have the same padding) Example: padding: 10px 20px 30px 40px;

Padding Units

Padding values can be specified in different units, including:

* px (pixels)
* % (percentage of the parent element's width)
* em (ems, relative to the font size of the element)
* rem (root ems, relative to the font size of the root element)

Example:- CSS:- padding: 10px; /\* 10 pixels \*/

padding: 10%; /\* 10% of the parent element's width \*/

padding: 1em; /\* 1 em, relative to the font size of the element \*/ padding: 1rem; /\* 1 rem, relative to the font size of the root element \*/

1. What are Block Level Elements?

In HTML, block level elements are elements that occupy the full width of their parent element and start on a new line. They are also known as "block elements" or "block-level tags." Block level elements are typically used to define the structure and layout of a web page.

Characteristics of Block Level Elements

Block level elements have the following characteristics:

* They occupy the full width of their parent element.
* They start on a new line.
* They can contain other block level elements or inline elements.
* They can have margins, borders, and padding.
* Examples of Block Level Elements

Here are some examples of block level elements:

* + div
  + p
  + h1-h6
  + ul
  + ol
  + li
  + table
  + tr
  + td
  + blockquote
  + pre
  + form
  + header
  + footer
  + nav
  + section
  + article
  + Aside

Here are some best practices for using block level elements:

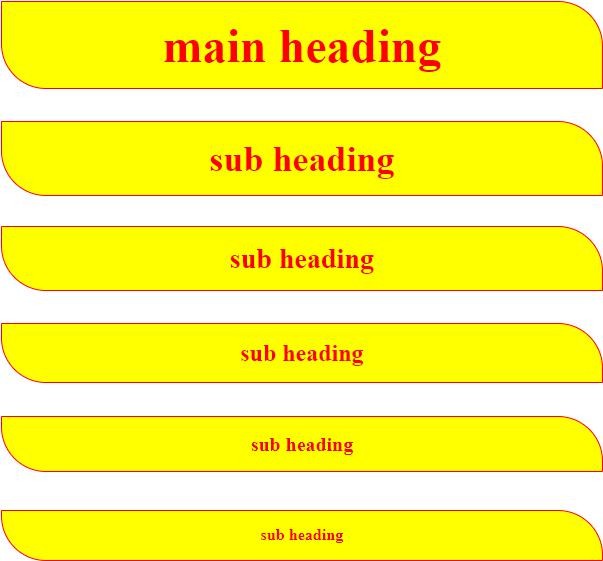
* + Use block level elements to define the structure and layout of a web page.
  + Use div elements to group other elements together.
  + Use p elements to define paragraphs of text.
  + Use h1-h6 elements to define headings.
  + Use ul and ol elements to define lists.
  + Use table elements to define tables.
  + Use form elements to define forms.
  + Use header and footer elements to define the header and footer of a web page.
  + Use nav elements to define navigation menus.
  + Use section and article elements to define sections and articles.
  + Use aside elements to define sidebars

1. Are Headings Block Level Elements? (yes/no)

Yes

Headings (h1-h6) are block level elements. They occupy the full width of their parent element and start on a new line. They are used to define the structure and hierarchy of a web page, and are typically displayed in a larger font size than regular text.

1. write the html code to develop below application with headings



## Text Formatting Tags

1. <b></b>------------------------ bold text
2. <strong></strong> strongtext
3. <i></i> italic text
4. <em></em> emphasizedtext
5. <u></u> underlined text
6. <del></del> deleted text
7. <strike></strike> strike text
8. <mark></mark> marked text
9. <sup></sup> superscript
10. <sub></sub> subscript
11. <big></big> big text
12. <small></small> smalltext
13. <code></code> represent the computer code
14. <var></var> mathematicalformulas
15. <br> line break
16. <hr> horizontalline
17. <address></address> displayaddress
18. <abbr></abbr> abirritations

title attribute used to define abbreviation

1. <bdo dir=""></bdo>---------bi directional override
2. <blockquote></blockquote> represent the quotations

**Example**

<!DOCTYPE html>

<html>

<head>

<title>Text Formatting Tags</title>

</head>

<body>

This is <b>Bold</b> Text <br>

This is <strong>strong</strong> text <br> This is <i>Italic</i> Text <br>

This is <em>Emphasized</em> Text <br> This is <u>underlined</u> Text <br>

This is <del>Deleted</del> Text <br> This is <strike>Striked</strike> Text <br>

This is <mark>Marked</mark> Text <br> a<sup>2</sup>+b<sup>2</sup> <br> H<sub>2</sub>O <br>

This is <big>Big</big> Text <br>

This is <small>small</small> Text <br>

<code>for (int i=0; i<10; i++) {} </code> <br>

<var>a2+b2</var> <br>

<hr>

<address>

ExcelR <br>Ameerpet

<br> Hyderabad

</address>

<br>

<abbr title="Hyper Text Markup Language"> HTML

</abbr>

<br>

<bdo dir="rtl">Welcome to UI Technologies</bdo>

<br>

<blockquote>

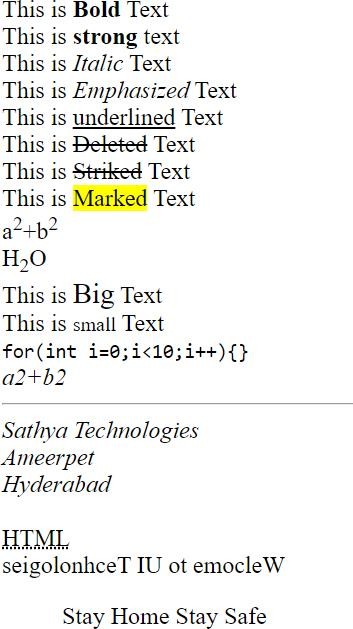
Stay Home Stay Safe

</blockquote>

</body>

</html>

**Output**



* Paragraphs in HTML
* Inline Elements

**PARAGRAPHS**

* Converting Block Level Elements to Inline Elements
* Converting Inline Elements to Block Level Elements
* Margin in CSS
* Box Model in CSS

### Paragraphs

* + <p></p> tag, used to display paragraphs
  + <p></p> tag also block level element

### Inline Elements

* + each element never starts with the new line (Inline elements displays horizontally)

### Converting Block Level Elements to Inline Elements in CSS

* + Below snippet used to convert block level elements to inline elements

Ex.

### display: inline-block;

**Converting Inline Elements to Block Level Elements in CSS**

* + Below snippet used to convert inline elements to block level elements

Ex.

### display: block;

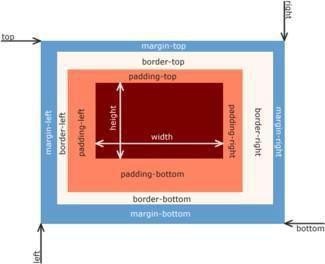
**Margin in CSS**

* + space around the element called as margin

Ex.

### box-model in CSS

margin: 20px

* + combination of padding, border and margin called as box-model in CSS

Example:

paragrahs.html

<!DOCTYPE html>

<html>

<head>

<title>Paragraphs</title>

<link rel="stylesheet" href="paragraphs.css">

</head>

<body>

<p>

Lorem ipsum dolor sit amet consectetur adipisicing elit. Magnam cum numquam, ab blanditiis laudantium nemo odit perspiciatis officia itaque tenetur ut. Natus praesentium dolore magnam.

</p>

<p>

Lorem ipsum dolor sit amet consectetur adipisicing elit. Magnam cum numquam, ab blanditiis laudantium nemo odit perspiciatis officia itaque tenetur ut. Natus praesentium dolore magnam.

</p>

<p>

Lorem ipsum dolor sit amet consectetur adipisicing elit. Magnam cum numquam, ab blanditiis laudantium nemo odit perspiciatis officia itaque tenetur ut. Natus praesentium dolore magnam.

</p>

</body>

</html> paragraphs.css p{

border: 1px solid red; width: 20%;

display: inline-block; padding: 20px; margin: 20px; border-radius: 20px; text-align: justify;

font-family: Comic Sans MS; color: white;

background: linear-gradient(45deg,red,black);

}

### Output

**PRACTICE PAPER**

1. which tag is used to create the Paragraphs The <p> tag \_
2. Are Paragraphs Block Level Elements? (yes/no) Yes
3. how to convert block to inline To convert a block-level element (like a <div> or <p>)
4. how to convert inline to block To convert an inline element (like <span> or <a>)
5. complete the below application with paragraphs



Ans:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Gradient Blocks with Text</title>

<style>

.container {

display: grid;

grid-template-columns: repeat(4, 1fr);

gap: 20px;

padding: 20px;

}

.block {

background: linear-gradient(to right, red, green);

border-radius: 15px;

padding: 15px;

color: white;

font-family: Arial, sans-serif;

box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);

}

p {

margin: 0;

line-height: 1.5;

}

</style>

</head>

<body>

<div class="container">

<div class="block">

<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Unde quia nulla nam. Voluptatum quis, recusandae commodi repellat, a sequi, porro pariatur tenetur quia aspernatur perspiciatis?</p>

</div>

<div class="block">

<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Unde quia nulla nam. Voluptatum quis, recusandae commodi repellat, a sequi, porro pariatur tenetur quia aspernatur perspiciatis?</p>

</div>

<div class="block">

<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Unde quia nulla nam. Voluptatum quis, recusandae commodi repellat, a sequi, porro pariatur tenetur quia aspernatur perspiciatis?</p>

</div>

<div class="block">

<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Unde quia nulla nam. Voluptatum quis, recusandae commodi repellat, a sequi, porro pariatur tenetur quia aspernatur perspiciatis?</p>

</div>

<div class="block">

<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Unde quia nulla nam. Voluptatum quis, recusandae commodi repellat, a sequi, porro pariatur tenetur quia aspernatur perspiciatis?</p>

</div>

<div class="block">

<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Unde quia nulla nam. Voluptatum quis, recusandae commodi repellat, a sequi, porro pariatur tenetur quia aspernatur perspiciatis?</p>

</div>

<div class="block">

<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Unde quia nulla nam. Voluptatum quis, recusandae commodi repellat, a sequi, porro pariatur tenetur quia aspernatur perspiciatis?</p>

</div>

<div class="block">

<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Unde quia nulla nam. Voluptatum quis, recusandae commodi repellat, a sequi, porro pariatur tenetur quia aspernatur perspiciatis?</p>

</div>

</div>

</body>

</html>

* + divisions in html
  + internal CSS

### DIVISIONS

* + differences between internal CSS and external CSS
  + selectors
    1. class selector
    2. id selector
    3. element selector
    4. universal selector
  + differences between class selector and id selector

## divisions in html

* Divisions are used to divide webpage into multiple sections
* <div></div> tag used to create divisions in webpages
* <div></div> tag also block level element



## internal CSS

* + <style></style> tag used to write internal CSS

## differences between internal CSS and external CSS

|  |  |
| --- | --- |
| **Internal CSS** | **External CSS** |
| <style></style> tag used to write the internal CSS | <link> tag, used to include the external CSS |
| <style></style> tag is paired tag | <link> tag is non paired tag |
| we can achieve CSS reusability through external CSS | we can't reuse CSS through internal CSS |

**selectors**

* + Selector "Selects" Particular "HTML Elements" to apply CSS

### Types of Selectors

* + 1. class selector
    2. id selector
    3. element selector / tag selector
    4. universal selector

### class selector

* + class selector should start with "." (dot)

Syntax

HTML CSS

\*\*\*\*\* \*\*\*

<div class="c1"> .c1{

</div> }

### id selector

* + id selector should start with "#"

Syntax

HTML CSS

\*\*\*\*\* \*\*\*

<div id="id1"> #id1{

</div> }

### element selector

* + element selector starts with "tag" name

Syntax

HTML CSS

\*\*\*\*\* \*\*\*

<div> div {

</div> }

### universal selector

* + we will represent "universal selector" with "\*"
  + CSS applied to "all elements" through "universal selector"

**differences between class selector and id selector**

|  |  |
| --- | --- |
| **Class selector** | **Id selector** |
| class selector starts with . (dot) | id selector starts with "#" |
| we may duplicate class | Id should be unique |
| we can apply more than one class Ex.  <div class="c1 c2 c3">  </div> | we can apply only one id Ex.  <div id="id1">  </div> |
| class selector has less priority compared to id selector  (specificity) | id selector has more priority compared to class selector  (specificity) |
| if a greater number of teams working  on same project, then class selector is suggested | to override 3rd party CSS then we will use id selector |
|  |  |

1. How to create universal selector use the asterisk symbol (\*)
2. Write the code for below application



Ans:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Card Layout</title>

<style>

body {

font-family: Arial, sans-serif;

display: flex;

justify-content: center;

align-items: center;

height: 100vh;

margin: 0;

}

.container {

display: grid;

grid-template-columns: repeat(3, 1fr);

gap: 20px;

padding: 20px;

}

.card {

border: 2px solid #ddd;

border-radius: 10px;

box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);

text-align: center;

padding: 20px;

background-color: #f9f9f9;

}

.card-header {

background: linear-gradient(to right, red, black);

color: white;

font-size: 18px;

padding: 10px 0;

border-top-left-radius: 8px;

border-top-right-radius: 8px;

}

.card-content {

padding: 10px;

color: #333;

}

</style>

</head>

<body>

<div class="container">

<div class="card">

<div class="card-header">HTML</div>

<div class="card-content">

Lorem ipsum, dolor sit amet consectetur adipiscing elit. Doloribus nulla, eos dolores quisquam mollitia.

</div>

</div>

<div class="card">

<div class="card-header">CSS</div>

<div class="card-content">

Lorem ipsum, dolor sit amet consectetur adipiscing elit. Doloribus nulla, eos dolores quisquam mollitia.

</div>

</div>

<div class="card">

<div class="card-header">JavaScript</div>

<div class="card-content">

Lorem ipsum, dolor sit amet consectetur adipiscing elit. Doloribus nulla, eos dolores quisquam mollitia.

</div>

</div>

<div class="card">

<div class="card-header">ReactJS</div>

<div class="card-content">

Lorem ipsum, dolor sit amet consectetur adipiscing elit. Doloribus nulla, eos dolores quisquam mollitia.

</div>

</div>

<div class="card">

<div class="card-header">Angular</div>

<div class="card-content">

Lorem ipsum, dolor sit amet consectetur adipiscing elit. Doloribus nulla, eos dolores quisquam mollitia.

</div>

</div>

</div>

</body>

</html>

### PRACTICE PAPER

1. How to Create Button and Write Basic Example

Ans:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Basic Button Example</title>

<style>

.my-button {

background-color: #4CAF50; /\* Green background \*/

border: none; /\* No borders \*/

color: white; /\* White text \*/

padding: 15px 32px; /\* Button padding \*/

text-align: center; /\* Center the text \*/

text-decoration: none; /\* No underline \*/

display: inline-block; /\* Align the button \*/

font-size: 16px; /\* Text size \*/

margin: 4px 2px; /\* Margins \*/

cursor: pointer; /\* Pointer cursor \*/

border-radius: 8px; /\* Rounded corners \*/

}

.my-button:hover {

background-color: #45a049; /\* Darker green on hover \*/

}

</style>

</head>

<body>

<button class="my-button" onclick="buttonClicked()">Click Me!</button>

<p id="output"></p>

<script>

function buttonClicked() {

document.getElementById("output").innerText = "Button was clicked!";

}

</script>

</body>

</html>

1. Explain <span> tag in HTML?

Ans:

The <span> tag in HTML is an inline container used to group and style small portions of text or other inline elements within a document. Unlike the <div> tag, which is a block-level element, <span> does not start on a new line and only takes up as much width as necessary.

The <span> tag is most often used in combination with CSS and JavaScript to apply specific styles or to manipulate parts of text dynamically.

Syntax: <span>Content here</span>.

1. Are Button Span and img tags are inline (yes/no) YES
2. How to write inline CSS The style attribute is used inside the HTML tag
3. In How Many ways we can apply CSS?

Ans:

1. Inline CSS

2. Internal CSS

3. External CSS

1. Explain CSS Specificity?

Ans:

CSS Specificity is a set of rules that determines which CSS style is applied to an element when there are conflicting declarations. It’s essentially a ranking system that calculates the importance of different CSS selectors. The higher the specificity, the more likely the style rule will be applied over others.

1. How to display images in HTML by using <img> tag
2. Is <img> tag paired tag? NO
3. Write <img> tag attributes

Ans:

**src:** Specifies the path to the image.

**alt:** Provides alternate text for the image, useful for informing users about the image and displaying in case of network issues.

**Crossorigin:** Allows importing images from third-party sites with cross-origin access for use with canvas.

**height:** Specifies the height of the image.

**width:** Specifies the width of the image.

**ismap:** Specifies an image as a server-side image map.

**loading:** Specifies whether a browser should defer loading of images until certain conditions are met or load an image immediately.

**longdesc:** Specifies a URL to a detailed description of an image.

**referrerpolicy:** Specifies which referrer information to use when fetching an image (e.g., no-referrer, no-referrer-when-downgrade, origin).

**size:** Specifies image sizes for different page layouts.

**srcset:** Specifies a list of image files to use in different situations.

**usemap:** Specifies an image as a client-side image map.

1. Explain hover in CSS?

Ans:

In CSS, :hover is a pseudo-class that applies styles to an element when the user hovers their cursor over it. This is commonly used to enhance user interaction by changing the appearance of an element in response to mouse movements. When you define styles using the :hover pseudo-class, they take effect when the mouse pointer is positioned over the selected element. This can be applied to various HTML elements, including links, buttons, images, and more.

**Syntax:**

selector:hover {

property: value;

}

### PRACTICE PAPER

1. How to create Hyper Links in HTML? Ans:

To create hyperlinks in HTML, you use the <a> (anchor) tag. The <a> tag is used to define a hyperlink that links to another webpage, file, email address, or any other resource.

**Syntax:** <a href="URL">Link Text</a>

1. Write Attributes of Anchor Tag? Ans:

1. href

2. name

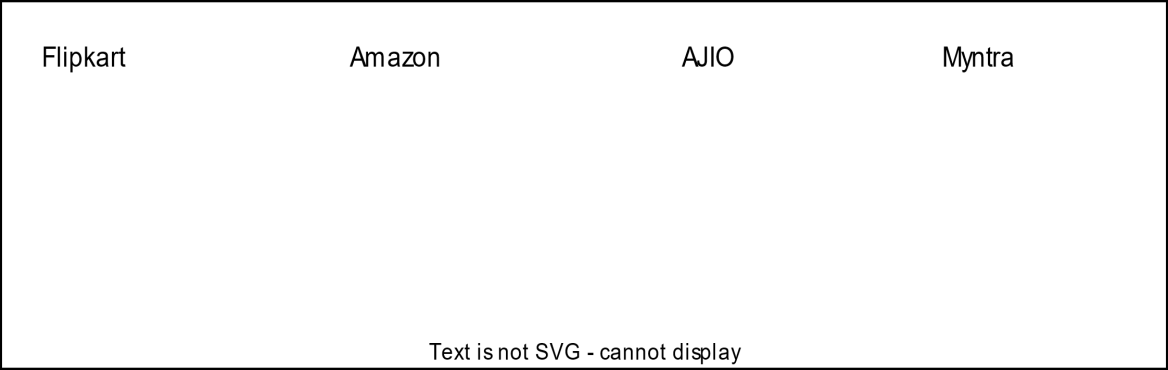
3. target

4. title

5. rel

1. implement the below application with anchor tag Flipkart -> https:[//www.flipkart.com/](http://www.flipkart.com/) Amazon -> https://[www.amazon.in/](http://www.amazon.in/)

AJIO -> https://[www.ajio.com/](http://www.ajio.com/) Myntra -> https://[www.myntra.com/](http://www.myntra.com/) Load above web sites in new tab



Ans:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Shopping Websites</title>

</head>

<body>

<div style="text-align: center;">

<a href="https://www.flipkart.com/" target="\_blank">Flipkart</a> &nbsp; &nbsp;

<a href="https://www.amazon.in/" target="\_blank">Amazon</a> &nbsp; &nbsp;

<a href="https://www.ajio.com/" target="\_blank">AJIO</a> &nbsp; &nbsp;

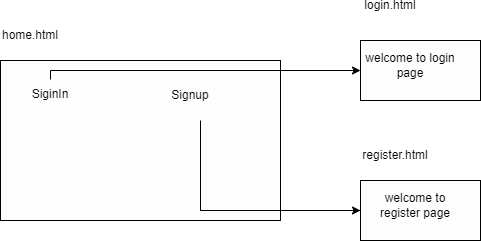
<a href="https://www.myntra.com/" target="\_blank">Myntra</a>

</div>

</body>

</html>

1. implement the below application with anchor tag



Ans:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>SignIn and SignUp</title>

</head>

<body>

<div style="width: 300px; height: 150px; border: 1px solid black; display: flex; justify-content: space-around; align-items: center;">

<a href="login.html" style="text-decoration: none;">SignIn</a>

<a href="register.html" style="text-decoration: none;">Signup</a>

</div>

<div style="margin-top: 50px;">

<h3>Login Page Content</h3>

<p>To implement the content of the login page, create a file named <code>login.html</code> with the following content:</p>

<pre>

&lt;!DOCTYPE html&gt;

&lt;html lang="en"&gt;

&lt;head&gt;

&lt;meta charset="UTF-8"&gt;

&lt;meta name="viewport" content="width=device-width, initial-scale=1.0"&gt;

&lt;title&gt;Login&lt;/title&gt;

&lt;/head&gt;

&lt;body&gt;

&lt;h1&gt;Welcome to Login Page&lt;/h1&gt;

&lt;/body&gt;

&lt;/html&gt;

</pre>

</div>

<div style="margin-top: 50px;">

<h3>Register Page Content</h3>

<p>To implement the content of the register page, create a file named <code>register.html</code> with the following content:</p>

<pre>

&lt;!DOCTYPE html&gt;

&lt;html lang="en"&gt;

&lt;head&gt;

&lt;meta charset="UTF-8"&gt;

&lt;meta name="viewport" content="width=device-width, initial-scale=1.0"&gt;

&lt;title&gt;Register&lt;/title&gt;

&lt;/head&gt;

&lt;body&gt;

&lt;h1&gt;Welcome to Register Page&lt;/h1&gt;

&lt;/body&gt;

&lt;/html&gt;

</pre>

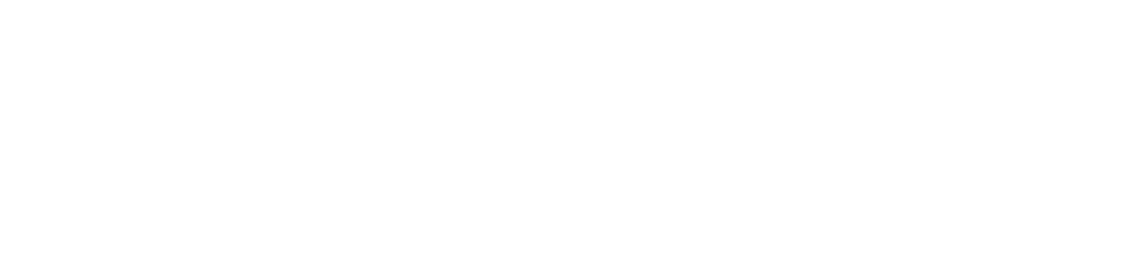
</div>

</body>

</html>

### PRACTICE PAPER

1)



Design the Below Application with fieldset and legend

Ans:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Fieldset with Legend Example</title>

<style>

fieldset {

width: 80%;

margin: 20px auto;

border: 2px solid #000;

}

legend {

font-weight: bold;

}

h1, h2, h3, h4 {

margin: 10px 0;

}

</style>

</head>

<body>

<fieldset>

<legend>Login</legend>

<h1>Welcome to UI</h1>

<h2>Welcome to UI</h2>

<h3>Welcome to UI</h3>

<h4>Welcome to UI</h4>

</fieldset>

</body>

</html>

### 

### PRACTICE PAPER

1. which tag used to create tables <table>
2. write the tags to create tables with diagram Ans:

<!-- index.html -->

<!DOCTYPE html>

<html>

<body>

<table>

<tr>

<th>Firstname</th>

<th>Lastname</th>

<th>Age</th>

</tr>

<tr>

<td>Priya</td>

<td>Sharma</td>

<td>24</td>

</tr>

<tr>

<td>Arun</td>

<td>Singh</td>

<td>32</td>

</tr>

<tr>

<td>Sam</td>

<td>Watson</td>

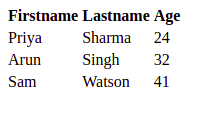
<td>41</td>

</tr>

</table>

</body>

</html>



1. write table attributes with explanation

Ans:

**Attribute** **Use**

width width of table or table cell

height height of table or table cell

align align text in table

valign vertically align text in table cell

border border width of table in px

bgcolor background color of table

cellspacing gap between table cells

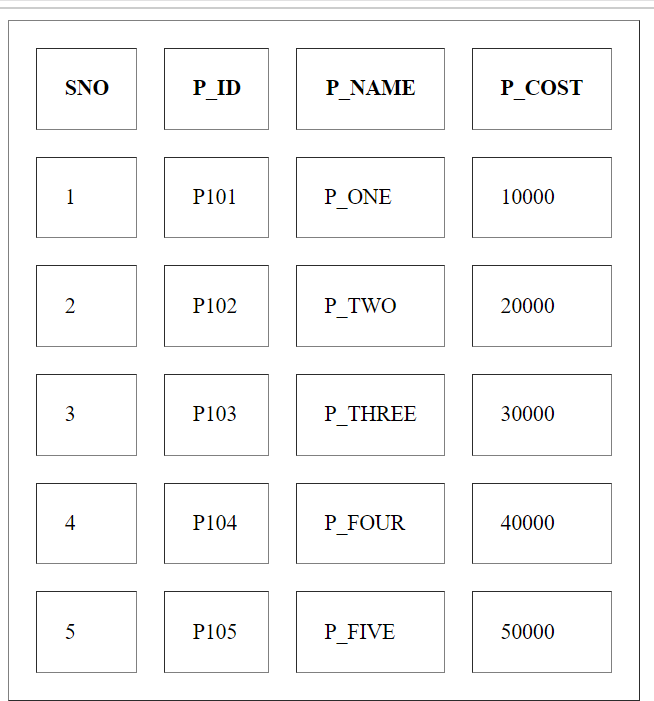
cellpadding gap inside table cells

colspan used to group columns in same row.

rowspan used to group columns in next row.

span used to span colgroup cols.

1. write the code for below diagram in practice paper and execute in laptop



Ans:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Product Table</title>

<style>

table, th, td {

border: 1px solid black;

border-collapse: collapse;

padding: 10px;

text-align: center;

}

th, td {

width: 100px;

}

</style>

</head>

<body>

<table>

<tr>

<th>SNO</th>

<th>P\_ID</th>

<th>P\_NAME</th>

<th>P\_COST</th>

</tr>

<tr>

<td>1</td>

<td>P101</td>

<td>P\_ONE</td>

<td>10000</td>

</tr>

<tr>

<td>2</td>

<td>P102</td>

<td>P\_TWO</td>

<td>20000</td>

</tr>

<tr>

<td>3</td>

<td>P103</td>

<td>P\_THREE</td>

<td>30000</td>

</tr>

<tr>

<td>4</td>

<td>P104</td>

<td>P\_FOUR</td>

<td>40000</td>

</tr>

<tr>

<td>5</td>

<td>P105</td>

<td>P\_FIVE</td>

<td>50000</td>

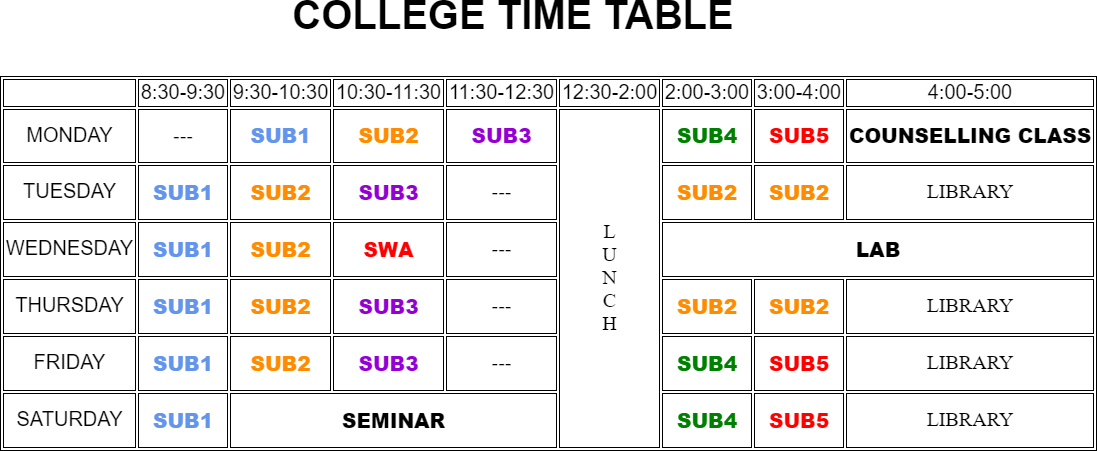
</tr>

</table>

</body>

</html>

1. write the code for below diagram in practice paper and execute in laptop



Ans:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>College Time Table</title>

<style>

table, th, td {

border: 1px solid black;

border-collapse: collapse;

text-align: center;

padding: 10px;

font-size: 16px;

}

th, td {

width: 100px;

}

th {

font-weight: bold;

}

.title {

font-size: 24px;

font-weight: bold;

text-align: center;

margin-bottom: 20px;

}

.blue { color: blue; }

.orange { color: orange; }

.purple { color: purple; }

.red { color: red; }

.green { color: green; }

.black { color: black; }

.lunch {

writing-mode: vertical-lr;

transform: rotate(180deg);

}

</style>

</head>

<body>

<div class="title">COLLEGE TIME TABLE</div>

<table>

<tr>

<th></th>

<th>8:30-9:30</th>

<th>9:30-10:30</th>

<th>10:30-11:30</th>

<th>11:30-12:30</th>

<th>12:30-2:00</th>

<th>2:00-3:00</th>

<th>3:00-4:00</th>

<th>4:00-5:00</th>

</tr>

<tr>

<td>MONDAY</td>

<td>---</td>

<td class="blue">SUB1</td>

<td class="orange">SUB2</td>

<td class="purple">SUB3</td>

<td rowspan="6" class="lunch">L U N C H</td>

<td class="green">SUB4</td>

<td class="red">SUB5</td>

<td>COUNSELLING CLASS</td>

</tr>

<tr>

<td>TUESDAY</td>

<td class="blue">SUB1</td>

<td class="orange">SUB2</td>

<td class="purple">SUB3</td>

<td>---</td>

<td class="orange">SUB2</td>

<td class="orange">SUB2</td>

<td>LIBRARY</td>

</tr>

<tr>

<td>WEDNESDAY</td>

<td class="blue">SUB1</td>

<td class="orange">SUB2</td>

<td class="red">SWA</td>

<td>---</td>

<td colspan="3">LAB</td>

</tr>

<tr>

<td>THURSDAY</td>

<td class="blue">SUB1</td>

<td class="orange">SUB2</td>

<td class="purple">SUB3</td>

<td>---</td>

<td class="green">SUB4</td>

<td class="red">SUB5</td>

<td>LIBRARY</td>

</tr>

<tr>

<td>FRIDAY</td>

<td class="blue">SUB1</td>

<td class="orange">SUB2</td>

<td class="purple">SUB3</td>

<td>---</td>

<td class="green">SUB4</td>

<td class="red">SUB5</td>

<td>LIBRARY</td>

</tr>

<tr>

<td>SATURDAY</td>

<td class="blue">SUB1</td>

<td colspan="3" class="black">SEMINAR</td>

<td>LIBRARY</td>

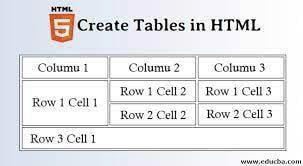
</tr>

</table>

</body>

</html>

1. write the code for below diagram in practice paper and execute in laptop



Ans:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Create Tables in HTML</title>

<style>

table, th, td {

border: 1px solid black;

border-collapse: collapse;

padding: 10px;

text-align: center;

}

th {

background-color: lightgray;

}

</style>

</head>

<body>

<h2>Create Tables in HTML</h2>

<table>

<tr>

<th>Column 1</th>

<th>Column 2</th>

<th>Column 3</th>

</tr>

<tr>

<td>Row 1 Cell 1</td>

<td>Row 1 Cell 2</td>

<td>Row 1 Cell 3</td>

</tr>

<tr>

<td>Row 2 Cell 1</td>

<td>Row 2 Cell 2</td>

<td>Row 2 Cell 3</td>

</tr>

<tr>

<td>Row 3 Cell 1</td>

<td colspan="2"></td>

</tr>

</table>

</body>

</html>

1. write the code for below diagram in practice paper and execute in laptop



Ans:

<!DOCTYPE html>

<html>

<head>

<style>

table {

border-collapse: collapse;

width: 100%;

}

th, td {

text-align: left;

padding: 8px;

border: 1px solid #ddd;

}

tr:nth-child(even) {

background-color: #f2f2f2;

}

th {

background-color: #4CAF50;

color: white;

}

</style>

</head>

<body>

<h2>Seminar Schedule</h2>

<table>

<tr>

<th>Day</th>

<th>Schedule</th>

<th>Topic</th>

</tr>

<tr>

<td>Monday</td>

<td>8:00 a.m. - 5:00 p.m.</td>

<td>Introduction to XML, Validity: DTD and Relax NG</td>

</tr>

<tr>

<td>Monday</td>

<td>8:00 a.m. - 11:00 a.m.</td>

<td>XPath</td>

</tr>

<tr>

<td>Tuesday</td>

<td>11:00 a.m. - 2:00 p.m.</td>

<td>XSL Transformations</td>

</tr>

<tr>

<td>Tuesday</td>

<td>2:00 p.m. - 5:00 p.m.</td>

<td></td>

</tr>

<tr>

<td>Wednesday</td>

<td>8:00 a.m. - 12:00 p.m.</td>

<td>XSL Formatting Objects</td>

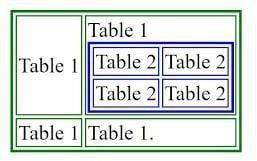
</tr>

</table>

</body>

</html>

1. write the code for below diagram in practice paper and execute in laptop



Ans:

<!DOCTYPE html>

<html>

<head>

<style>

table {

border: 1px solid black;

border-collapse: collapse;

}

td {

border: 1px solid black;

padding: 5px;

}

</style>

</head>

<body>

<table>

<tr>

<td>Table 1</td>

<td><table border="1">

<tr>

<td>Table 2</td>

<td>Table 2</td>

</tr>

<tr>

<td>Table 2</td>

<td>Table 2</td>

</tr>

</table></td>

<td>Table 1</td>

</tr>

<tr>

<td colspan="3">Table 1</td>

</tr>

</table>

</body>

</html>

1. write the code for below diagram in practice paper and execute in laptop



Ans:

<!DOCTYPE html>

<html>

<head>

<style>

.calculator {

width: 200px;

border: 1px solid #ccc;

padding: 10px;

text-align: center;

}

.display {

font-size: 24px;

height: 40px;

line-height: 40px;

text-align: right;

padding: 0 10px;

border: 1px solid #ccc;

margin-bottom: 10px;

}

.buttons {

display: grid;

grid-template-columns: repeat(4, 1fr);

gap: 5px;

}

.button {

width: 100%;

height: 40px;

font-size: 18px;

line-height: 40px;

background-color: #f0f0f0;

border: 1px solid #ccc;

cursor: pointer;

}

.button.operator {

background-color: #ff6666;

}

.button.clear {

background-color: #ccc;

}

</style>

</head>

<body>

<div class="calculator">

<div class="display">0</div>

<div class="buttons">

<button class="operator">+</button>

<button class="operator">-</button>

<button class="operator">x</button>

<button class="operator">÷</button>

<button>7</button>

<button>8</button>

<button>9</button>

<button class="orange">=</button>

<button>4</button>

<button>5</button>

<button>6</button>

<button>1</button>

<button>2</button>

<button>3</button>

<button>0</button>

<button>.</button>

<button class="clear">AC</button>

</div>

</div>

<script>

const display = document.querySelector('.display');

const buttons = document.querySelectorAll('.button');

let currentInput = '';

let operator = '';

buttons.forEach(button => {

button.addEventListener('click', () => {

const value = button.textContent;

if (value === '+' || value === '-' || value === 'x' || value === '÷') {

operator = value;

currentInput += value;

} else if (value === '=') {

if (operator) {

const result = eval(currentInput);

currentInput = result.toString();

operator = '';

}

} else if (value === 'AC') {

currentInput = '';

operator = '';

} else {

currentInput += value;

}

display.textContent = currentInput;

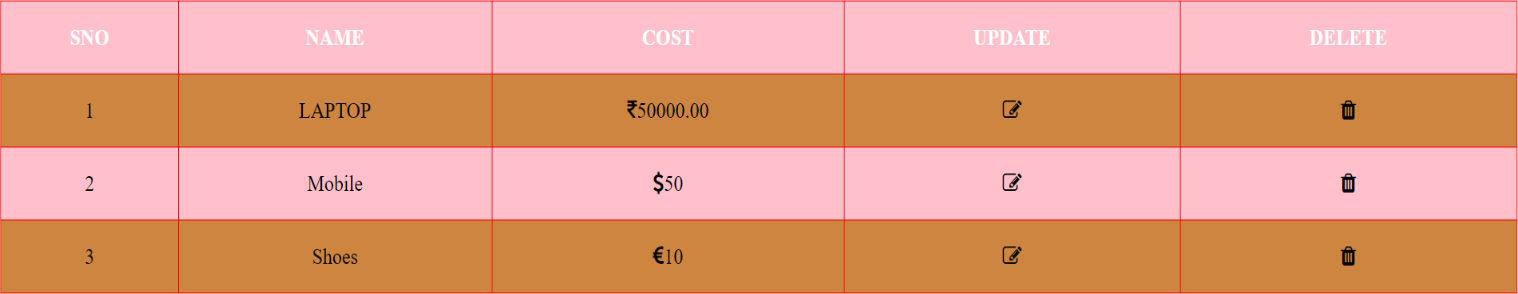
}); });

</script>

</body>

</html>

1. write the code for below diagram in practice paper and execute in laptop



Ans:

<!DOCTYPE html>

<html>

<head>

<style>

table {

border: 1px solid black;

border-collapse: collapse;

}

th, td {

border: 1px solid black;

padding: 5px;

text-align: center;

}

th {

background-color: #f2f2f2;

}

.sno-column {

width: 5%;

}

.name-column {

width: 20%;

}

.cost-column {

width: 20%;

}

.update-column {

width: 10%;

}

.delete-column {

width: 10%;

}

</style>

</head>

<body>

<table>

<tr>

<th class="sno-column">SNO</th>

<th class="name-column">NAME</th>

<th class="cost-column">COST</th>

<th class="update-column">UPDATE</th>

<th class="delete-column">DELETE</th>

</tr>

<tr>

<td>1</td>

<td>LAPTOP</td>

<td>₹80000</td>

<td><img src="edit\_icon.png" alt="Edit"></td>

<td><img src="delete\_icon.png" alt="Delete"></td>

</tr>

<tr>

<td>2</td>

<td>Mobile</td>

<td>$50</td>

<td><img src="edit\_icon.png" alt="Edit"></td>

<td><img src="delete\_icon.png" alt="Delete"></td>

</tr> <tr>

<td>3</td>

<td>Shoes</td>

<td>€10</td>

<td><img src="edit\_icon.png" alt="Edit"></td>

<td><img src="delete\_icon.png" alt="Delete"></td>

</tr>

</table>

</body>

</html>

1. write few points related to font-awesome

Ans:

Font Awesome is a popular icon library that provides a vast collection of scalable vector graphics (SVGs) for use in web projects. These icons can be easily integrated into your HTML code, offering a convenient and customizable way to add visual elements to your website.

**Key Features and Benefits:**

Scalability: Font Awesome icons are vector-based, meaning they can be resized without losing quality. This ensures crisp and clear rendering at any size.

Customizability: You can change the color, size, and style of icons to match your website's design.

Accessibility: Icons are accessible to screen readers, making your website more inclusive for users with disabilities.

Performance: Font Awesome icons are lightweight, improving your website's loading speed.

Cross-Browser Compatibility: The library is designed to work seamlessly across different web browsers.

1. cart symbol <i class="fa fa-shopping-cart"></i>
2. trash symbol <i class="fa fa-trash"></i>
3. edit symbol <i class="fa fa-edit"></i>
4. rupee symbol Unicode entity: &#8377
5. dollar symbol Unicode entity: &#36
6. user profile profile-container
7. how to use font-awesome link the Font Awesome CSS file from a CDN
8. CDN Stands for Content Delivery Network
9. is Font Awesome CSS library? YES

# Practice Paper

1. Types of Lists in HTML?

Ans:

1. Unordered Lists (<ul>):

Items are displayed with bullet points.

The default bullet style is a filled circle, but you can customize it using the type attribute (type="circle", type="disc", or type="square").

2. Ordered Lists (<ol>):

Items are displayed with numbers or letters.

The default numbering style is Arabic numerals, but you can change it using the type attribute (type="1" for Arabic numerals, type="A" for uppercase letters, type="a" for lowercase letters, type="I" for uppercase Roman numerals, or type="i" for lowercase Roman numerals).

3. Definition Lists (<dl>):

Used to define terms and their descriptions.

Each definition list item has a <dt> element for the term and a <dd> element for the description.

1. Write the Syntax for Ordered List?

Ans:

<ol>

<li>Item 1</li>

<li>Item 2</li>

</ol>

1. Write the Attributes of Ordered List?

Ans:

1. type:

Specifies the type of numbering to use for the list items.

Possible values: 1 (default), A, a, I, i.

2. start:

Sets the starting number for the list items.

The default value is 1.

3. reversed:

Reverses the order of the list items.

4. numbered:

Deprecated attribute (use type instead).

1. Write the Syntax for Unordered List?

Ans:

<ul>

<li>Item 1</li>

<li>Item 2</li>

<li>Item 3</li>

</ul>

1. Write the Unordered List Attributes?

Ans:

1. type:

Specifies the type of bullet style to use for the list items.

Possible values: disc (default), circle, square.

2. style:

Allows you to apply custom CSS styles to the list, such as changing the color, font, or spacing.

3. class:

Assigns a class name to the list, allowing you to style it using CSS selectors.

4. id:

Gives the list a unique identifier, which can be used for referencing the list in CSS or JavaScript.

1. Write the Syntax for Definition List

Ans:

<dl>

<dt>Term 1</dt>

<dd>Description of Term 1</dd>

<dt>Term 2</dt>

<dd>Description of Term 2</dd>

</dl>

1. <dt></dt> stands for data term
2. <dd></dd> stands for data description
3. <li></li> stands for list
4. <dl></dl> stands for Definition List
5. Write the code for below application and execute in laptop



Ans:

<!DOCTYPE html>

<html>

<head>

<style>

.container {

display: flex;

flex-wrap: wrap;

gap: 10px;

}

.box {

background-color: red;

padding: 20px;

text-align: center;

}

.box.html {

background-color: blue;

}

</style>

</head>

<body>

<div class="container">

<ul>

<li class="box html">HTML</li>

<li class="box">CSS</li>

<li class="box">JavaScript</li>

<li class="box">Angular</li>

<li class="box">ReactJS</li>

</ul>

<ul>

<li class="box html">HTML</li>

<li class="box">CSS</li>

<li class="box">JavaScript</li>

<li class="box">Angular</li>

<li class="box">ReactJS</li>

</ul>

<ul>

<li class="box html">HTML</li>

<li class="box">CSS</li>

<li class="box">JavaScript</li>

<li class="box">Angular</li>

<li class="box">ReactJS</li>

</ul>

<ul>

<li class="box html">HTML</li>

<li class="box">CSS</li>

<li class="box">JavaScript</li>

<li class="box">Angular</li>

<li class="box">ReactJS</li>

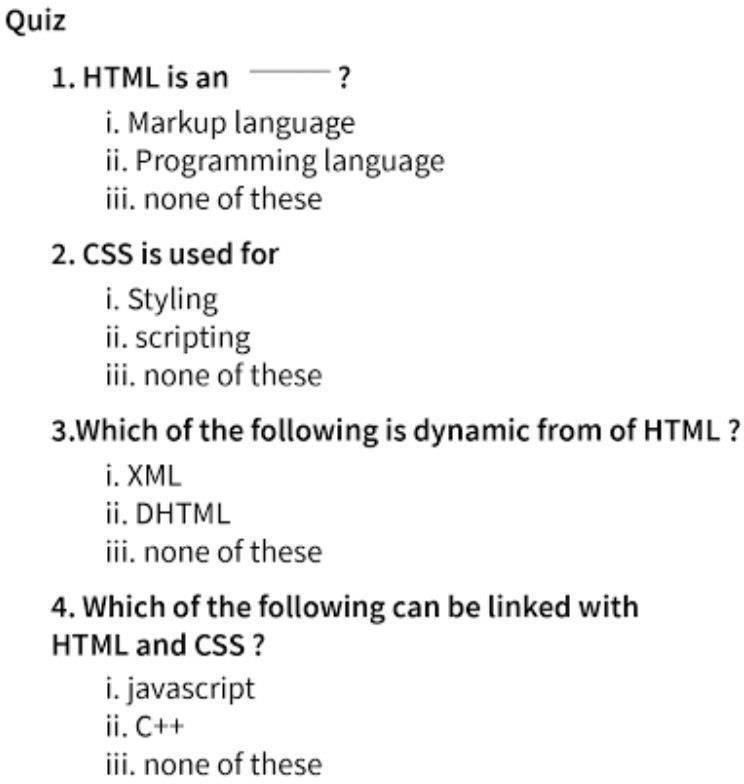
</ul>

</div>

</body>

</html>

1. Write the code for below application and execute in laptop



Ans:

<!DOCTYPE html>

<html>

<head>

<style>

.quiz-container {

width: 80%;

margin: 0 auto;

padding: 20px;

}

.question {

font-weight: bold;

margin-bottom: 10px;

}

.options {

margin-left: 20px;

}

input[type="radio"] {

margin-right: 5px;

}

</style>

</head>

<body>

<div class="quiz-container">

<h2>Quiz</h2>

<form>

<ol>

<li class="question">

1. HTML is an \_\_\_\_\_?

<ul class="options">

<li><input type="radio" name="q1" value="markup language"> Markup language</li>

<li><input type="radio" name="q1" value="programming language"> Programming language</li>

<li><input type="radio" name="q1" value="none of these"> none of these</li>

</ul>

</li>

<li class="question">

2. CSS is used for

<ul class="options">

<li><input type="radio" name="q2" value="styling"> Styling</li>

<li><input type="radio" name="q2" value="scripting"> scripting</li>

<li><input type="radio" name="q2" value="none of these"> none of these</li>

</ul>

</li>

<li class="question">

3. Which of the following is dynamic from of HTML?

<ul class="options">

<li><input type="radio" name="q3" value="XML"> XML</li>

<li><input type="radio" name="q3" value="DHTML"> DHTML</li>

<li><input type="radio" name="q3" value="none of these"> none of these</li>

</ul>

</li>

<li class="question">

4. Which of the following can be linked with HTML and CSS?

<ul class="options">

<li><input type="radio" name="q4" value="javascript"> javascript</li>

<li><input type="radio" name="q4" value="C++"> C++</li>

<li><input type="radio" name="q4" value="none of these"> none of these</li>

</ul>

</li>

</ol>

<button type="submit">Submit</button>

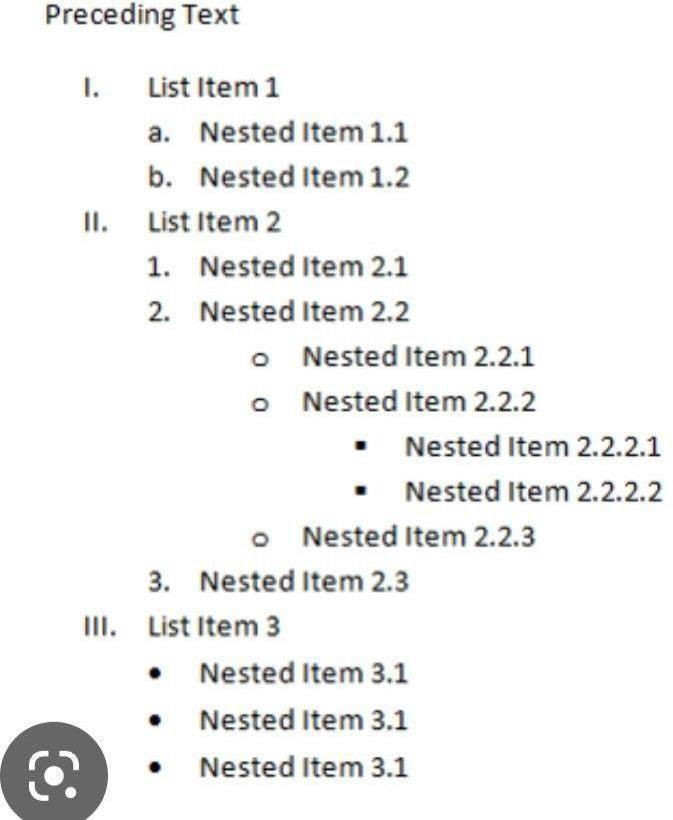
</form>

</div>

</body>

</html>

1. Write the code for below application and execute in laptop



Ans:

<!DOCTYPE html>

<html>

<head>

<style>

ol, ul {

margin-left: 20px;

padding-left: 0;

}

li {

margin-bottom: 10px;

}

</style>

</head>

<body>

<p>Preceding Text</p>

<ol>

<li>List Item 1

<ol>

<li>Nested Item 1.1</li>

<li>Nested Item 1.2</li>

</ol>

</li>

<li>List Item 2

<ol>

<li>Nested Item 2.1</li>

<li>Nested Item 2.2

<ol>

<li>Nested Item 2.2.1</li>

<li>Nested Item 2.2.2

<ul>

<li>Nested Item 2.2.2.1</li>

<li>Nested Item 2.2.2.2</li>

</ul>

</li>

<li>Nested Item 2.2.3</li>

</ol>

</li>

<li>Nested Item 2.3</li>

</ol>

</li>

<li>List Item 3

<ul>

<li>Nested Item 3.1</li>

<li>Nested Item 3.1</li>

<li>Nested Item 3.1</li>

</ul>

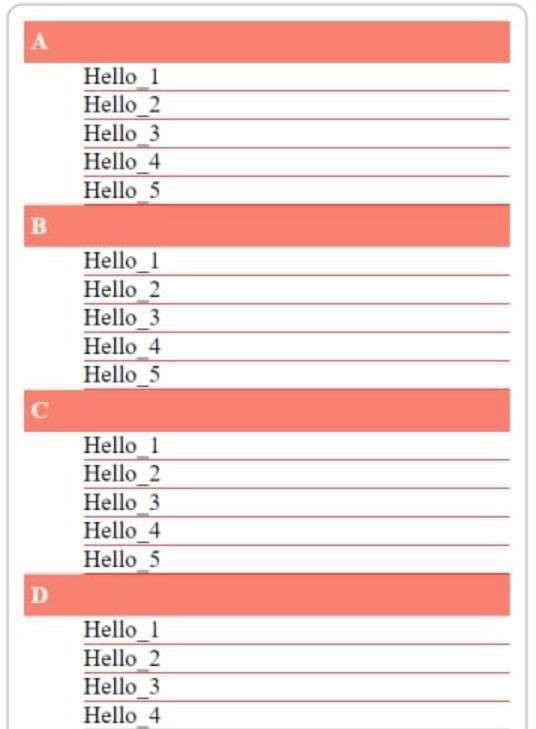
</li>

</ol>

</body>

</html>

1. Write the code for below application and execute in laptop



Ans:

<!DOCTYPE html>

<html>

<head>

<style>

.container {

display: flex;

flex-direction: column;

gap: 20px;

}

.section {

border: 1px solid #ccc;

padding: 10px;

}

.section-header {

font-weight: bold;

}

.items {

list-style: none;

padding: 0;

}

.items li {

margin-bottom: 5px;

}

</style>

</head>

<body>

<div class="container">

<div class="section">

<div class="section-header">A</div>

<ul class="items">

<li>Hello 1</li>

<li>Hello 2</li>

<li>Hello 3</li>

<li>Hello 4</li>

<li>Hello 5</li>

</ul>

</div>

<div class="section">

<div class="section-header">B</div>

<ul class="items">

<li>Hello 1</li>

<li>Hello 2</li>

<li>Hello 3</li>

<li>Hello 4</li>

<li>Hello 5</li>

</ul>

</div>

<div class="section">

<div class="section-header">C</div>

<ul class="items">

<li>Hello 1</li>

<li>Hello 2</li>

<li>Hello 3</li>

<li>Hello 4</li>

<li>Hello 5</li>

</ul>

</div>

<div class="section">

<div class="section-header">D</div>

<ul class="items">

<li>Hello 1</li>

<li>Hello 2</li>

<li>Hello 3</li>

<li>Hello 4</li>

<li>Hello 5</li>

</ul>

</div>

</div>

</body>

</html>

### PRACTICE PAPER

1. How to create Forms in HTML by using the <form> tag
2. Write form attributes

Ans:

**Attribute**  **Description**

**accept-charset** Specifies the character encodings used for form submission

**action** Specifies where to send the form-data when a form is submitted

**autocomplete** Specifies whether a form should have autocomplete on or off

**enctype** Specifies how the form-data should be encoded when submitting it to the server (only for method="post")

**method**  Specifies the HTTP method to use when sending form-data

**name** Specifies the name of the form

**novalidate** Specifies that the form should not be validated when submitted

**rel** Specifies the relationship between a linked resource and the current document

**target** Specifies where to display the response that is received after submitting the form

1. Why placeholder attribute in <input></input> tag is used to display a short, descriptive text inside the input field

1. How to create dropdown in Forms write basic example

Ans.

To create a dropdown menu in an HTML form, you use the <select> tag along with multiple <option> tags inside it. Each <option> represents an item in the dropdown list that users can select.

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Form with Dropdown</title>

</head>

<body>

<h1>Dropdown Form Example</h1>

<form action="/submit-form" method="POST">

<label for="cars">Choose a car:</label>

<select id="cars" name="car">

<option value="volvo">Volvo</option>

<option value="bmw">BMW</option>

<option value="audi">Audi</option>

<option value="tesla">Tesla</option>

</select>

<br><br>

<button type="submit">Submit</button>

</form>

</body>

</html>

1. How to create address field in HTML Forms

Ans.

To create an address field in an HTML form, you typically use a combination of input elements for different parts of the address, such as street, city, state, postal code, and country. There isn't a single <input> element for an address; instead, you'll break it down into logical parts using multiple input fields. Here's an example of how to create a form for collecting an address:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Address Form</title>

</head>

<body>

<h1>Address Form Example</h1>

<form action="/submit-address" method="POST">

<label for="street">Street Address:</label>

<input type="text" id="street" name="street" placeholder="1234 Main St" required>

<br><br>

<label for="apartment">Apartment/Suite (optional):</label>

<input type="text" id="apartment" name="apartment" placeholder="Apartment or Suite">

<br><br>

<label for="city">City:</label>

<input type="text" id="city" name="city" placeholder="City" required>

<br><br>

<label for="state">State/Province:</label>

<input type="text" id="state" name="state" placeholder="State/Province" required>

<br><br>

<label for="postal">Postal Code:</label>

<input type="text" id="postal" name="postal" placeholder="Postal Code" required>

<br><br>

<label for="country">Country:</label>

<select id="country" name="country" required>

<option value="">Select a country</option>

<option value="us">United States</option>

<option value="ca">Canada</option>

<option value="uk">United Kingdom</option>

<option value="in">India</option>

</select>

<br><br>

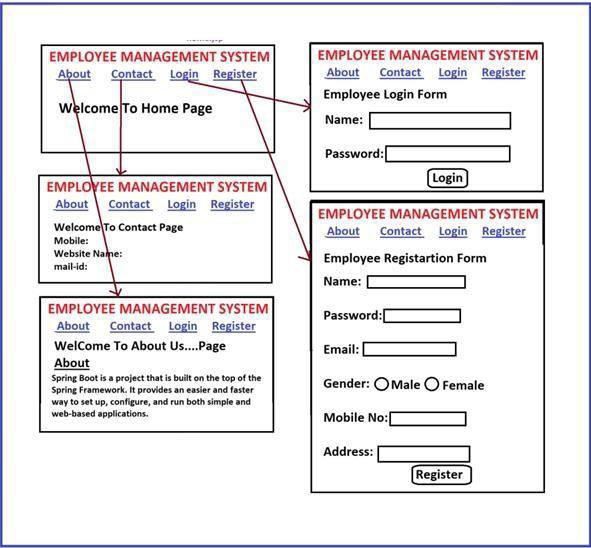
<button type="submit">Submit</button>

</form>

</body>

</html>

1. Single selection controls called as radio button
2. Multi selection controls called as check boxes
3. implement the below form with text field



Ans.

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Employee Management System</title>

</head>

<body>

<nav>

<a href="#home">Home</a> |

<a href="#about">About</a> |

<a href="#contact">Contact</a> |

<a href="#login">Login</a> |

<a href="#register">Register</a>

</nav>

<section id="home">

<h2>Welcome To Home Page</h2>

</section>

<hr>

<section id="about">

<h2>Welcome To About Us Page</h2>

<p>Spring Boot is a project that is built on the top of the Spring Framework. It provides an easier and faster way to set up, configure, and run both simple and web-based applications.</p>

</section>

<hr>

<section id="contact">

<h2>Welcome To Contact Page</h2>

<form>

<label for="mobile">Mobile:</label>

<input type="tel" id="mobile" name="mobile" placeholder="Enter mobile number"><br><br>

<label for="website">Website Name:</label>

<input type="url" id="website" name="website" placeholder="Enter website"><br><br>

<label for="mail">Mail-ID:</label>

<input type="email" id="mail" name="mail" placeholder="Enter email"><br><br>

<button type="submit">Submit</button>

</form>

</section>

<hr>

<section id="login">

<h2>Employee Login Form</h2>

<form>

<label for="login-name">Name:</label>

<input type="text" id="login-name" name="name" placeholder="Enter your name" required><br><br>

<label for="login-password">Password:</label>

<input type="password" id="login-password" name="password" placeholder="Enter your password" required><br><br>

<button type="submit">Login</button>

</form>

</section>

<hr>

<section id="register">

<h2>Employee Registration Form</h2>

<form>

<label for="register-name">Name:</label>

<input type="text" id="register-name" name="name" placeholder="Enter your name" required><br><br>

<label for="register-password">Password:</label>

<input type="password" id="register-password" name="password" placeholder="Enter your password" required><br><br>

<label for="register-email">Email:</label>

<input type="email" id="register-email" name="email" placeholder="Enter your email" required><br><br>

<label>Gender:</label>

<input type="radio" id="male" name="gender" value="male">

<label for="male">Male</label>

<input type="radio" id="female" name="gender" value="female">

<label for="female">Female</label><br><br>

<label for="register-mobile">Mobile No:</label>

<input type="tel" id="register-mobile" name="mobile" placeholder="Enter your mobile number" required><br><br>

<label for="register-address">Address:</label>

<input type="text" id="register-address" name="address" placeholder="Enter your address" required><br><br>

<button type="submit">Register</button>

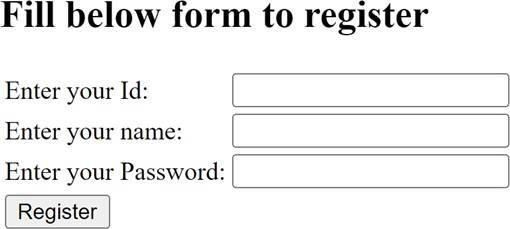
</form>

</section>

</body>

</html>

1. implement the below form



Ans.

<!DOCTYPE html>

<html>

<head>

<title>Registration Form</title>

<style>

form {

width: 300px;

margin: 0 auto;

padding: 20px;

border: 1px solid #ccc;

}

label {

display: block;

margin-bottom: 5px;

}

input[type="text"], input[type="password"] {

width: 100%;

padding: 10px;

border: 1px solid #ccc;

}

button {

display: block;

margin-top: 10px;

padding: 10px;

background-color: #4CAF50;

color: white;

border: none;

cursor: pointer;

}

</style>

</head>

<body>

<h2>Registration Form</h2>

<form>

<label for="userId">Enter your Id:</label>

<input type="text" id="userId" name="userId" required><br><br>

<label for="name">Enter your name:</label>

<input type="text" id="name" name="name" required><br><br>

<label for="password">Enter your Password:</label>

<input type="password" id="password" name="password" required><br><br>

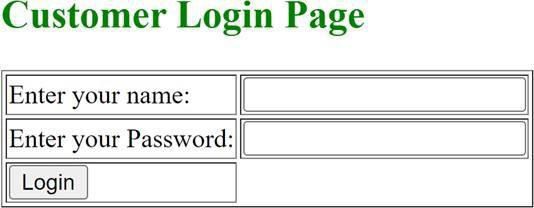
<button type="submit">Register</button>

</form>

</body>

</html>

1. implement the below form



Ans.

<!DOCTYPE html>

<html>

<head>

<title>Customer Login Page</title>

<style>

form {

width: 300px;

margin: 0 auto;

padding: 20px;

border: 1px solid #ccc;

}

label {

display: block;

margin-bottom: 5px;

}

input[type="text"], input[type="password"] {

width: 100%;

padding: 10px;

border: 1px solid #ccc;

}

button {

display: block;

margin-top: 10px;

padding: 10px;

background-color: #4CAF50;

color: white;

border: none;

cursor: pointer;

}

</style>

</head>

<body>

<h2>Customer Login Page</h2>

<form>

<label for="username">Enter your name:</label>

<input type="text" id="username" name="username" required><br><br>

<label for="password">Enter your Password:</label>

<input type="password" id="password" name="password" required><br><br>

<button type="submit">Login</button>

</form>

</body>

</html>

11) Write HTML code to get following text box?



Ans.

<!DOCTYPE html>

<html>

<head>

<style>

input[type="text"] {

width: 200px;

padding: 10px;

border: 1px solid #ccc;

}

</style>

</head>

<body>

<input type="text" placeholder="User Name">

</body>

</html>

1. Write HTML Code for following output



Ans.

<!DOCTYPE html>

<html>

<head>

<style>

input[type="text"] {

width: 200px;

padding: 10px;

border: 1px solid #ccc;

}

</style>

</head>

<body>

<input type="text" placeholder="User Name">

</body>

</html>

1. Write code for following output



Ans.

<!DOCTYPE html>

<html>

<head>

<style>

input[type="password"] {

width: 200px;

padding: 10px;

border: 1px solid #ccc;

}

</style>

</head>

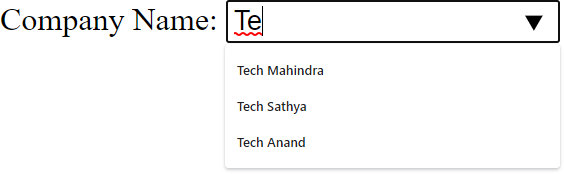
<body>

<input type="password" placeholder="Password">

</body>

</html>

1. Write Code for following output



Ans.

<!DOCTYPE html>

<html>

<head>

<style>

.autocomplete-container {

position: relative;

}

input[type="text"] {

width: 200px;

padding: 10px;

border: 1px solid #ccc;

}

.autocomplete-options {

position: absolute;

top: 100%;

left: 0;

right: 0;

background-color: white;

border: 1px solid #ccc;

z-index: 1;

display: none;

}

.autocomplete-option {

padding: 10px;

cursor: pointer;

}

.autocomplete-option:hover {

background-color: #f0f0f0;

}

</style>

</head>

<body>

<div class="autocomplete-container">

<input type="text" id="company-name" placeholder="Company Name" oninput="filterOptions()">

<ul id="company-options" class="autocomplete-options">

<li class="autocomplete-option">Tech Mahindra</li>

<li class="autocomplete-option">Tech Sathya</li>

<li class="autocomplete-option">Tech Anand</li>

</ul>

</div>

<script>

function filterOptions() {

const input = document.getElementById('company-name');

const options = document.getElementById('company-options').children;

const filter = input.value.toUpperCase();

for (let i = 0; i < options.length; i++) {

const option = options[i];

const text = option.textContent.toUpperCase();

if (text.indexOf(filter) > -1) {

option.style.display = '';

} else {

option.style.display = 'none';

}

}

}

</script>

</body>

</html>

1. Write code for following output



Ans.

<!DOCTYPE html>

<html>

<head>

<style>

.product-input {

display: flex;

align-items: center;

border: 1px solid #ccc;

padding: 5px;

}

.product-name {

margin-right: 10px;

}

.product-value {

flex-grow: 1;

}

.close-button {

cursor: pointer;

}

</style>

</head>

<body>

<div class="product-input">

<span class="product-name">Product Name</span>

<input type="text" class="product-value" value="Keyboard">

<span class="close-button">x</span>

</div>

</body>

</html>

1. Write HTML code to for following Output



Ans.

<!DOCTYPE html>

<html>

<head>

<style>

.whatsapp-button {

display: flex;

align-items: center;

padding: 10px;

border: 1px solid #ccc;

cursor: pointer;

}

.whatsapp-icon {

margin-right: 5px;

}

.whatsapp-number {

flex-grow: 1;

}

</style>

</head>

<body>

<div class="whatsapp-button">

<img class="whatsapp-icon" src="whatsapp-icon.png" alt="WhatsApp">

<span class="whatsapp-number">Whatsapp Number 8074864650</span>

</div>

</body>

</html>

1. Write HTML Code for following output



Ans.

<!DOCTYPE html>

<html>

<head>

<style>

.checkbox-container {

display: flex;

align-items: center;

}

input[type="checkbox"] {

margin-right: 5px;

}

</style>

</head>

<body>

<div class="checkbox-container">

<input type="checkbox" id="accept-license" name="accept-license">

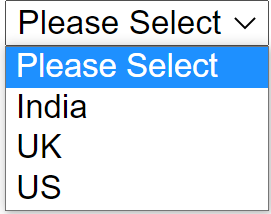
<label for="accept-license">I accept license agreement</label>

</div>

</body>

</html>

1. Write HTML code for following output



Ans.

<!DOCTYPE html>

<html>

<head>

<style>

.select-container {

position: relative;

}

select {

width: 200px;

padding: 10px;

border: 1px solid #ccc;

}

.options {

position: absolute;

top: 100%;

left: 0;

right: 0;

background-color: white;

border: 1px solid #ccc;

z-index: 1;

display: none;

}

.option {

padding: 10px;

cursor: pointer;

}

.option:hover {

background-color: #f0f0f0;

}

</style>

</head>

<body>

<div class="select-container">

<select id="country-select">

<option value="">Please Select</option>

<option value="India">India</option>

<option value="UK">UK</option>

<option value="US">US</option>

</select>

<ul id="country-options" class="options">

<li class="option">Please Select</li>

<li class="option">India</li>

<li class="option">UK</li>

<li class="option">US</li>

</ul>

</div>

<script>

const select = document.getElementById('country-select');

const options = document.getElementById('country-options');

select.addEventListener('click', () => {

options.style.display = 'block';

});

options.addEventListener('click', (event) => {

const selectedOption = event.target;

select.value = selectedOption.textContent;

options.style.display = 'none';

});

</script>

</body>

</html>

1. implement the below form



Ans.

<!DOCTYPE html>

<html>

<head>

<title>Student Registration Form</title>

<style>

form {

width: 80%;

margin: 0 auto;

padding: 20px;

}

label {

display: block;

margin-bottom: 5px;

}

input[type="text"], input[type="number"], select {

width: 100%;

padding: 10px;

border: 1px solid #ccc;

}

button {

display: block;

margin-top: 10px;

padding: 10px;

background-color: #4CAF50;

color: white;

border: none;

cursor: pointer;

}

</style>

</head>

<body>

<h2>Student Registration Form</h2>

<form>

<label for="firstName">First Name:</label>

<input type="text" id="firstName" name="firstName" required maxlength="50"><br><br>

<label for="lastName">Last Name:</label>

<input type="text" id="lastName" name="lastName" required maxlength="50"><br><br>

<label for="email">Email ID:</label>

<input type="email" id="email" name="email" required><br><br>

<label for="mobileNumber">Mobile Number:</label>

<input type="number" id="mobileNumber" name="mobileNumber" required maxlength="10"><br><br>

<label for="gender">Gender:</label>

<input type="radio" id="male" name="gender" value="Male">

<label for="male">Male</label>

<input type="radio" id="female" name="gender" value="Female">

<label for="female">Female</label><br><br>

<label for="dob">Date of Birth (DOB):</label>

<input type="date" id="dob" name="dob" required><br><br>

<h4>Address</h4>

<label for="city">City:</label>

<input type="text" id="city" name="city" required maxlength="50"><br><br>

<label for="pincode">Pin Code:</label>

<input type="number" id="pincode" name="pincode" required maxlength="6"><br><br>

<label for="state">State:</label>

<input type="text" id="state" name="state" required maxlength="50"><br><br>

<label for="country">Country:</label>

<input type="text" id="country" name="country" required value="India"><br><br>

<label for="hobbies">Hobbies:</label><br>

<input type="checkbox" id="drawing" name="hobbies" value="Drawing">

<label for="drawing">Drawing</label>

<input type="checkbox" id="singing" name="hobbies" value="Singing">

<label for="singing">Singing</label>

<input type="checkbox" id="dancing" name="hobbies" value="Dancing">

<label for="dancing">Dancing</label>

<input type="checkbox" id="sketching" name="hobbies" value="Sketching">

<label for="sketching">Sketching</label>

<input type="checkbox" id="others" name="hobbies" value="Others">

<label for="others">Others:</label>

<input type="text" id="otherHobbies" name="otherHobbies" maxlength="50"><br><br>

<label for="qualification">Qualification:</label><br>

<input type="radio" id="highSchool10th" name="qualification" value="High School(10th)">

<label for="highSchool10th">High School(10th)</label><br>

<input type="radio" id="higherSchool12th" name="qualification" value="Higher School(12th)">

<label for="higherSchool12th">Higher School(12th)</label><br>

<input type="radio" id="graduation" name="qualification" value="Graduation(Bachelors)">

<label for="graduation">Graduation(Bachelors)</label><br>

<input type="radio" id="postGraduation" name="qualification" value="Post Graduation(Masters)">

<label for="postGraduation">Post Graduation(Masters)</label><br>

<input type="radio" id="phd" name="qualification" value="Phd">

<label for="phd">Phd</label><br>

<input type="radio" id="bca" name="qualification" value="BCA(Bachelor of Computer Applications)">

<label for="bca">BCA(Bachelor of Computer Applications)</label><br>

<input type="radio" id="bcom" name="qualification" value="B.Com(Bachelor of Commerce)">

<label for="bcom">B.Com(Bachelor of Commerce)</label><br>

<input type="radio" id="bsc" name="qualification" value="B.Sc(Bachelor of Science)">

<label for="bsc">B.Sc(Bachelor of Science)</label><br>

<input type="radio" id="ba" name="qualification" value="BA(Bachelor of Arts)">

<label for="ba">BA(Bachelor of Arts)</label><br>

<input type="radio" id="mca" name="qualification" value="MCA(Master of Computer Applications)">

<label for="mca">MCA(Master of Computer Applications)</label><br>

<input type="radio" id="mcom" name="qualification" value="M.Com(Master of Commerce)">

<label for="mcom">M.Com(Master of Commerce)</label><br>

<input type="radio" id="msc" name="qualification" value="M.Sc(Master of Science)">

<label for="msc">M.Sc(Master of Science)</label><br>

<input type="radio" id="ma" name="qualification" value="MA(Master of Arts)">

<label for="ma">MA(Master of Arts)</label><br><br>

<label for="coursesAppliedFor">Courses Applied For:</label><br>

<input type="checkbox" id="course1" name="coursesAppliedFor" value="Course 1">

<label for="course1">Course 1</label><br>

<input type="checkbox" id="course2" name="coursesAppliedFor" value="Course 2">

<label for="course2">Course 2</label><br>

<input type="checkbox" id="course3" name="coursesAppliedFor" value="Course 3">

<label for="course3">Course 3</label><br>

<input type="checkbox" id="course4" name="coursesAppliedFor" value="Course 4">

<label for="course4">Course 4</label><br>

<input type="checkbox" id="course5" name="coursesAppliedFor" value="Course 5">

<label for="course5">Course 5</label><br><br>

<button type="submit">Submit</button>

</form>

</body>

</html>

**PRACTICE PAPER**

1. Explain Positions in CSS?

Ans.

CSS Positions

CSS positions allow you to control the placement of elements on a web page. There are five main positioning properties:

**static:** (default) Elements are positioned according to their normal flow. They are placed in the order they appear in the HTML document.

**relative:** Elements are positioned relative to their normal position. You can use the top, right, bottom, and left properties to offset the element from its original position.

**absolute:** Elements are positioned relative to their nearest positioned ancestor. If there is no positioned ancestor, they are positioned relative to the initial containing block (usually the <html> element).

**fixed:** Elements are positioned relative to the viewport. They remain fixed in place even when the page is scrolled.

**sticky:** Elements are positioned relative to their nearest positioned ancestor. However, once the element reaches a certain scroll threshold, it becomes fixed relative to the viewport.

1. What are the Differences between Absolute and Relative Positions?

Ans.

Absolute vs. Relative Positioning in CSS

Both absolute and relative positioning allow you to move elements from their default position within a document. However, they differ in how they reference their positioning context:

Relative Positioning:

Reference point: The element's normal position within the document flow.

Offsets: You can use the top, right, bottom, and left properties to offset the element from its original position.

Does not affect other elements: Relative positioning doesn't affect the layout of other elements on the page.

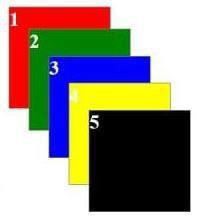
Absolute Positioning:

Reference point: The nearest positioned ancestor element (or the initial containing block if there are no positioned ancestors).

Offsets: You can also use top, right, bottom, and left to position elements absolutely.

Overlaps other elements: Elements positioned absolutely can overlap other elements on the page.

1. Implement The Below Application Absolute Position



Ans.

<!DOCTYPE html>

<html>

<head>

<style>

.container {

position: relative;

width: 400px;

height: 200px;

}

.box {

position: absolute;

width: 100px;

height: 100px;

}

.box1 {

background-color: red;

top: 0;

left: 0;

}

.box2 {

background-color: green;

top: 0;

left: 100px;

}

.box3 {

background-color: blue;

top: 0;

left: 200px;

}

.box4 {

background-color: yellow;

top: 100px;

left: 0;

}

.box5 {

background-color: black;

top: 100px;

left: 100px;

}

</style>

</head>

<body>

<div class="container">

<div class="box box1"></div>

<div class="box box2"></div>

<div class="box box3"></div>

<div class="box box4"></div>

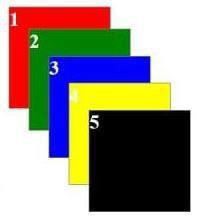
<div class="box box5"></div>

</div>

</body>

</html>

1. Implement the Below Application with CSS Relative Position



Ans.

<!DOCTYPE html>

<html>

<head>

<style>

.container {

width: 400px;

}

.box {

position: relative;

width: 100px;

height: 100px;

float: left;

margin: 10px;

}

.box1 {

background-color: red;

}

.box2 {

background-color: green;

}

.box3 {

background-color: blue;

}

.box4 {

background-color: yellow;

}

.box5 {

background-color: black;

}

</style>

</head>

<body>

<div class="container">

<div class="box box1"></div>

<div class="box box2"></div>

<div class="box box3"></div>

<div class="box box4"></div>

<div class="box box5"></div>

</div>

</body>

</html>

1. What is Z-Index in CSS?

Ans.

Z-index in CSS is a property used to control the stacking order of overlapping elements. It determines which element appears on top of others when they overlap.

Elements with a higher z-index value appear on top of elements with a lower z-index value.

If two elements have the same z-index, their stacking order is determined by their source order in the HTML document (the element listed later will appear on top).

The default z-index value is 0.

1. Which one is the default position in CSS static
2. How to display Element in center of webpage Ans.

Here are the methods to center an element in the middle of a webpage using CSS:

1. Using margin: auto;:

This is the most common and straightforward method.

Set the margin property of the element to auto.

This will automatically distribute the remaining space equally on both sides of the element, centering it horizontally.

2. Using text-align: center; (for inline elements):

If the element is inline (like a <span> or <p>), you can use text-align: center; on its parent element to center the content horizontally.

3. Using flexbox:

Set the container element's display property to flex and use justify-content: center; to center the child element horizontally.

4. Using grid layout:

Set the container element's display property to grid and use place-items: center; to center the child element both horizontally and vertically.

1. What is border-box in CSS?

Ans.

border-box is a CSS box model that specifies how the width and height of an element are calculated. When using border-box, the width and height values you set include the padding, border, and content areas of the element.

**Width and height include borders and padding:** Unlike the default content-box model, border-box ensures that the specified width and height encompass the entire box, including its borders and padding.

**No overflow:** If the content of the element exceeds its specified dimensions, the content will not overflow. Instead, the content will be resized or clipped to fit within the box.

**Calculation:** The browser calculates the content area based on the specified width and height, then subtracts the padding and border sizes to determine the actual content width and height.