

HTML & CSS INTERVIEW QUESTIONS BY GST

What is the full form of HTML and what are the uses?

HTML stands for **Hypertext Markup Language**. It is a markup language used to create **web pages**. It tells web browsers how to display text, images, links, and other elements on the internet.

What are HTML Elements?

HTML Elements are the **building blocks** of HTML used to define the **structure** and **content** of a **web page**. They generally consist of a **start tag**, **content**, and **end tag**. **E.g.-** Heading Element, Paragraph element, Div element, List element, Form element etc...

What is the structure of an HTML document?

The basic structure of an HTML document is

A screenshot of a web browser's developer tools interface, specifically the 'HTML' tab. The code editor shows a basic HTML document structure with line numbers 1 through 11. The code includes a DOCTYPE declaration, an html root element, a head section with meta charset and title tags, and a body section with an h1 heading and a paragraph.

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <meta charset="UTF-8">
5   <title>Page Title</title>
6 </head>
7 <body>
8   <h1>Main Heading</h1>
9   <p>This is a paragraph.</p>
10 </body>
11 </html>
```

DOCTYPE Declaration: It clarifies the **document type** and **HTML version**. It is used to inform web browsers which version of HTML is being used in the document.

HTML Element: This element is the **root container** for all the content on an HTML document. It contains the entire web page and serves as the top-level container.

Head Element: This element holds **metadata** and **document-related** information. It includes things like the document's title, character encoding, linked stylesheets, and meta tags that describe the document. This information is not visible on the webpage itself but is crucial for search engines and other tools.

Body Element: This element is used to display the **main content** of a webpage, including text, images, links, videos, and other elements that users interact with when they view the webpage.

What are the important parts of an HTML document?

- A line that clarifies the **HTML version**.
- A **head** element that provides important information about the document.
- A **body** that displays the actual content of the document.

What is the difference between JavaScript and HTML?

JavaScript	HTML
JavaScript is a programming language that makes web pages more interactive and dynamic	HTML is a markup language that is used to describe the structure and layout of web pages

What are the differences between HTML and Python?

HTML	Python
HTML is a markup language that is used to describe the structure and layout of web pages	Python is an interpreted, object-oriented, high-level programming language
It is generally used to format the output (How to appear on a website)	It is generally used to instruct a computer to perform certain actions on an input and return an output

What are the uses/advantages of HTML?

- Using **HTML elements** we can organize headings, paragraphs and other content on a webpage. Without these the text on the webpage would all look the same.
- Using **HTML elements**, we can embed images and videos in the HTML document.
- Using **Hyperlinks**, we can navigate to other web pages within the HTML document.

Why HTML 5 is best compared to old versions?

HTML 5 is the latest version of **HTML**. Some of the new features added in **HTML5** that make it better than HTML are:

- audio and video elements
- semantic elements like header, footer, figure, figcaption, nav, etc.
- placeholder attribute
- progress element, etc.

What is HTML Semantic Elements?

HTML Semantic Elements are specific HTML tags that give meaning to the structure of **web content**. (`<header>`, `<nav>`, `<main>`, `<article>`, `<section>`, `<footer>`, and more.)

What are the uses of Semantic HTML?

- **Accessibility:** Makes websites usable for all, including disabilities and mobile devices.
- **Search Engine Optimization (SEO):** Improves search rankings, increasing visitor numbers.
- **Easy to understand:** Enhances code understanding for developers.

What are the differences between HTML semantic and non-semantic elements?

semantic	non-semantic
semantic elements have the meaning	non-semantic elements don't have the meaning
They describe the content they contain	They can contain anything

What are the HTML formatting elements?

The **HTML formatting elements** are used to **format the text**.

Element	Description
b	It is used to bold the text
strong	It bolds the text and is used to tell the browser that the text is important
i	It is used to make the text italic
em	It makes the text italic and is used to tell the browser that the text has to be emphasized
mark	It is used to mark or highlight a text for special interest or reference purposes
small	It is used to display the smaller text especially for copyrights and side-comments
del	It is used to display the deleted content
ins	It is used to display the additional content
sub	It is used to subscript a text
sup	It is used to superscript a text

What are the HTML media elements?

HTML media elements are Tags that allows us to put images, audio, and videos directly into web pages for interactive and engaging content. ****: For images, **<audio>**: For audio content, **<video>**: For videos, **<canvas>**: For dynamic graphics, **<iframe>**: For embedding external content

```
<video>
  <source src="movie.mp4" type="video/mp4" />
</video>
```

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

What are the HTML elements that are commonly used?

`<h1>` for headings, `<p>` for paragraphs, `<br>` for line breaks, `<hr>` for horizontal rules, `<a>` for links, `<img>` for images, `<div>` for containers, and `<span>` for inline styling.

What is the extension of an HTML file?

The extension of an HTML file is `.html`.

Can we design dynamic web pages using HTML?

No, we can't design dynamic web pages with HTML alone. By Combining **HTML** with **JavaScript**, we can create **dynamic** web pages.

What are the types of tags in HTML?

HTML Tags are like **keywords** that define how a **web browser** will format and display the content. There are **two** types of tags in html they are,

Paired Tags: Paired tags consist of two tags, **start tag** and **end tag**.

Examples: `<h1>Heading</h1>`, `<p>Paragraph</p>`, etc.

Unpaired Tags/Empty tags: Unpaired tags do not have an **end tag**.

Examples: ``, `<br/>`, etc.

What are the differences between HTML Tags and HTML Elements?

HTML Tags	HTML elements
Tags are the starting and ending parts of an HTML element. They begin with <code><</code> symbol and end with <code>></code> symbol.	It generally consists of a start tag, content, and an end tag.
Ex:- Start tag: <code><p></code> , <code><h1></code> , End tag: <code></p></code> , <code></h1></code>	Ex:- <code><p>Paragraph</p></code> , <code><h1>Heading</h1></code>

What is an HTML container element?

The HTML container element `<div>` defines a container. It is used to combine a **group of elements** and apply **CSS** styles to multiple elements **at once**.

What is Nested HTML?

HTML elements can be **nested**. Which means that we can place one HTML element inside another HTML element.

```
html
Copy code

<div>
  <p>This is a nested HTML element.</p>
</div>
```

What are the HTML attributes?

The **HTML attributes** provide additional information about HTML elements. They are always specified in the **start tag**. They usually come in **name="value"** pairs.

Name some of the attributes used in HTML?

- **Class:** Assigns one or more classes for styling.
- **ID:** Provides a unique identifier for targeting.
- **Src:** Specifies source file (e.g., image or script).
- **Href:** Defines link URL for anchors.
- **Alt:** Offers image description for accessibility.
- **Title:** Provides hover-over tooltip text.
- **Width/Height:** Sets element dimensions.
- **Disabled:** Deactivates interactive elements.
- **Checked:** Pre-selects checkboxes/radio buttons.
- **Placeholder:** Gives input hint text in forms.

What is meant by the HTML href attribute?

The HTML **href** attribute is the most important attribute of the HTML **anchor** element. It specifies the **URL/path** of the page the link goes to.

What is the value given to the HTML href attribute?

The value of the HTML **href** attribute is **URL/path** of the page where the link goes to.

What is an HTML class attribute?

The HTML **class** attribute is used to specify a **class name** for an HTML element.

Why do we use the HTML id attribute?

Using the HTML **id** attribute, we can **organize & apply styles** to an HTML element **uniquely**.

What are the differences between the HTML id attribute and the HTML class attribute?

class	id
The HTML class attribute specifies a class name for an HTML element	The HTML id attribute specifies a unique id for an HTML element.
It is used to apply styles to multiple elements with the class name	It is used to apply styles to a single element
One or more HTML elements can have the same class name	The id must be unique within the HTML document. There shouldn't be more than one element with the same id

What are the different types of input element available and their uses?

- **type="text"**: For single-line text input.
- **type="password"**: For secure single-line password input.
- **type="submit"**: Creates a submit button for form submission.
- **type="radio"**: Generates a radio button for single-choice selection.
- **type="checkbox"**: Creates a checkbox for multiple-choice selection.

How to create a combobox in HTML?

The `<select>` element creates a **dropdown list (combobox)** in HTML. The available choices are defined within `<option>` tags.

```
1 <select>
2   <option>option1</option>
3   <option>option2</option>
4 </select>
```

HTML

What is checkbox and how to use checkbox?

A **checkbox** in HTML is a little box that users can **tick** or **untick**. It's used to choose one or more options from a list. It's created using the **type="checkbox"** attribute value in the `<input>` element.

```
1 <input type="checkbox" id="vehicle1" name="vehicle1" value="Bike1" />
2 <input type="checkbox" id="vehicle2" name="vehicle2" value="Bike2" />
```

HTML

What are the differences between the HTML checkbox and radio input elements?

Aspect	Checkbox Input	Radio Input
Purpose	Allows multiple selections	Allows single selection
`type` attribute	<code>type="checkbox"</code>	<code>type="radio"</code>
Unique `name`	Not required, but can be used	Required for related options
Selected by default	Can be pre-checked	One radio button can be pre-selected
Interaction	Independent selection	Exclusive selection in a group
Group behavior	Can be part of various groups	Belongs to a single group
Use case	Multiple options selection	Single-choice selection

How to create a single-line text box for searching queries?

The HTML **input** element with the type **search** is used to add a single line text box for searching queries.

```
i 1 <input type="search" />
```

HTML

What is an HTML checked attribute?

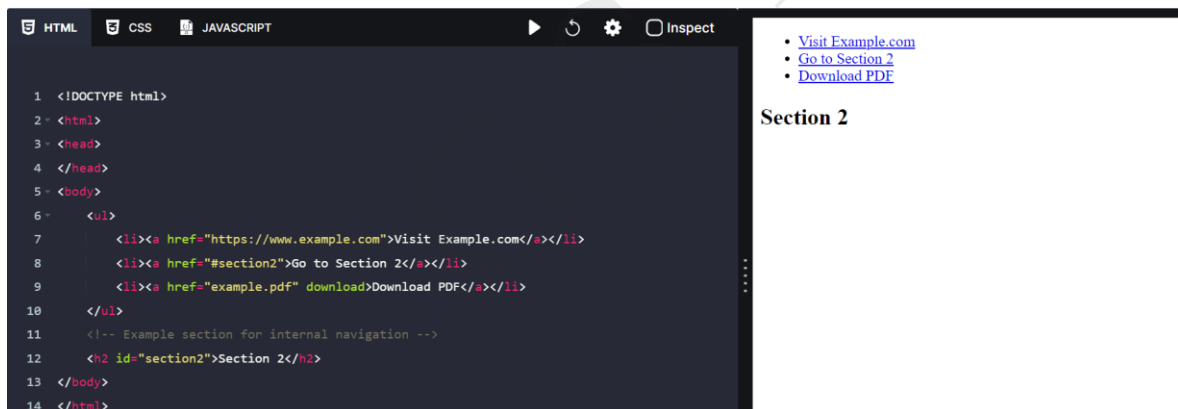
The HTML **checked** attribute specifies that an input element should be pre-selected (checked) when the page loads.

```
i 1 <input type="radio" id="genderMale" value="Male" name="gender" checked />
```

HTML

What is the use of an HTML anchor element?

Using HTML **anchor** elements, we can navigate to **other webpages** or a specific **HTML element** within the **HTML document**. They are also known as **hyperlinks** or **links**.



Does a hyperlink only apply to text?

No. We can use hyperlinks to **text** as well as **images**, **buttons**, etc.

html

Copy code

```
<a href="https://www.example.com"><button>Click here</button></a>

<a href="https://www.example.com">
  
</a>
```

Explain about HTML heading element and paragraph element.

Heading Element HTML defines **six** levels of headings.

- The HTML **<h1>** element defines the most important heading of an HTML document.
- The HTML **<h6>** element defines the least important heading of an HTML document.

Paragraph Element HTML **<p>** element defines the paragraph of an HTML document.

How to add line breaks to an HTML paragraph element?

The HTML **
** element is used to add a **line break** to an HTML paragraph element. It breaks the text and continue it in the next line. It is useful in writing poems, addresses, etc...

What is a viewport?

A **viewport** is the part of a web page that we see in a **browser** without scrolling.

How do you include comments in HTML?

HTML **comments** are written using **<!--** to start the comment and **-->** to end it.

Explain the HTML hr (horizontal line) element?

The HTML **<hr>** element inserts a horizontal line and helps to separate the content.

What are the differences between HTML div and span elements?

<code><div></code>	<code></code>
Block-level element	Inline element
Takes up the full width of its parent container	Occupies only as much width as necessary
Starts on a new line	Inline element, does not start on a new line
Used to combine a group of elements and apply CSS styles to multiple elements at once.	Used to apply CSS styles to specific parts of text or inline elements.

```
html


<h1>This is a heading</h1>
  <p>This is a paragraph of text.</p>
</div>


```

```
html
<p>This is a <span style="color: blue;">blue</span> word.</p>
```

Explain about the HTML block-level elements and inline elements?

Block-level Elements: The HTML Block-level elements always **start** in a **new line** and take up the **full width** available. So, an HTML Block-level element occupies the entire horizontal space of its parent element. **Examples:** <div>, <h1>, <p> etc...

Inline Elements: The HTML Inline elements **do not start** in a **new line** and take up the content width. **Examples:** <a>, , , <button> etc...

Can we use the HTML p element inside an HTML span element?

The HTML **span** element is an **inline** element, which should contain only other inline elements. So as **span** is an **inline** element and **p** is a **block-level** element, we can't use the **p** element inside a **span** element.

Note: Not all the HTML inline elements should contain only other inline elements. There are few exceptions for it. **Example:** HTML **anchor** element, etc.

How to display images on the web page?

The HTML element is used to **display images** on a web page. It requires two main attributes: **src**, which specifies the **image file's path**, and **alt**, which provides **alternate text** for the **image**.

1. Displaying a Local Image:

html

Copy code

```

```

2. Displaying an Image from a URL:

html

Copy code

```

```

What are the attributes we can give to an HTML image element?

The **attributes** we can give to an **HTML image element** are **src**, **alt**, **height**, **width**, **class**, etc.

html

Copy code

```

```

How to create a link in HTML & what is the HTML target attribute?

- The HTML `<a>` element creates links. It uses the **href** attribute for the **destination URL**.
- The HTML **target** attribute defines where the **linked content opens** – in the **same tab**, **a new tab**, or **a new window**

Value	Description
<code>_self</code>	(Default value) It opens the document in the same window or tab as it was clicked.
<code>_blank</code>	It opens the document in a new window or tab.
<code>_parent</code>	It opens the document in a parent frame.
<code>_top</code>	It opens the document in a full-body window.

```
html Copy code  
  
<a href="https://www.example.com" target="_self">Open in the Same Tab</a>  
  
<a href="https://www.example.com" target="_blank">Open in a New Tab</a>  
  
<a href="https://www.example.com" target="_parent">Open in Parent Frame</a>  
  
<a href="https://www.example.com" target="_top">Open in Top Window</a>
```

What is the difference between HTML anchor element and HTML link element?

HTML anchor element: We use the HTML **anchor** element to **navigate** to other **web resources** or a **specific element** within the HTML document.

```
html Copy code  
  
<a href="https://www.example.com">Visit Example.com</a>
```

HTML link element: We use the HTML **link** element to **link** the **HTML** and **CSS** files in **<head>** section.

```
html Copy code  
  
<link rel="stylesheet" type="text/css" href="styles.css">
```

What is the difference between HTML `<script>` and `<link>` element?

- **HTML script element:** We use the HTML `<script>` element to insert the **JavaScript** code into a webpage.
- **HTML link element:** We use the HTML `<link>` element to link the **HTML** and **CSS** files.

What is an HTML script element?

The HTML `<script>` element is used to insert **JavaScript** code into a webpage, making the HTML document more **interactive** and **dynamic**.

```
1 <script src="counter.js"></script>
```

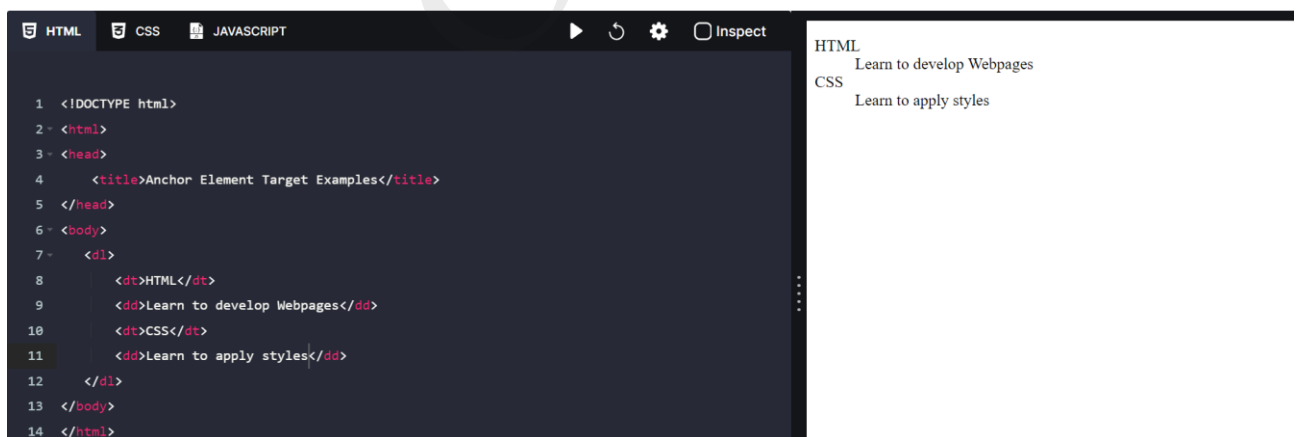
What is the `<iframe>` tag used for?

The `<iframe>` tag is used to embed **another document** or **webpage** within the **current document**. It's commonly used for embedding **videos**, **maps**, and **external content**.

```
<iframe src="https://learning.ccbp.in"></iframe>
```

Explain types of lists in HTML?

- **Ordered list:** It is used to group a set of items in a **specific order**.
- **Unordered list:** It is used to group a set of items in **no particular order**.
- **Description list:** It is used to display **name/value** pairs such as **description terms(dt)** and **description definitions(dd)**



Difference between HTML ordered List and HTML unordered List?

HTML ordered list	HTML unordered list
It is a collection of related items that follow some order or have a sequence	It is a collection of related items that have no special order or sequence
It is created using the <code>ol</code> element	It is created using the <code>ul</code> element
By default, list items in the Ordered List are marked with numbers.	By default, list items in the Unordered List are marked with bullets.

What are forms and how to create forms in HTML?

An HTML **form** is used to gather **user information**. We create forms in HTML using the `<form>` element. It acts as a container for various elements like **input fields**, **labels**, **textarea**, **selects**, **buttons**, and more.

```
html Copy code

<form action="/submit">
  <label for="name">Name:</label>
  <input type="text" id="name" name="name" required>
  <br>

  <label for="email">Email:</label>
  <input type="email" id="email" name="email" required>
  <br>

  <input type="submit" value="Submit">
</form>
```

What is `<table>` element and how you structure rows and columns.

The `<table>` element is used to create a **table** in HTML. **Rows** are created with the `<tr>` tag, and **cells** are created with `<th>` (header cell) or `<td>` (data cell) within the **rows**.

```
HTML CSS JAVASCRIPT Inspect

1 <!DOCTYPE html>
2 <html>
3 <head> </head>
4 <body>
5 <table border="1">
6 <tr>
7 <th>Column1</th>
8 <th>Column2</th>
9 <th>Column3</th>
10 </tr>
11 <tr>
12 <td>1</td>
13 <td>2</td>
14 <td>3</td>
15 </tr>
16 </table>
17 </body>
18 </html>
19
```

Column1	Column2	Column3
1	2	3

Web

SAVE + Create New

How to convert HTML to PDF?

Converting **HTML to PDF** is generally done using **server-side** languages like **Python**, **Java**, or **Node.js** with the help of libraries, or by using specialized online services. It's not a native feature of HTML itself. **Example (If any):** In Python, we can use a library like **wkhtmltopdf** or **pdfkit** to convert HTML to PDF.

How to give space in HTML?

Using **HTML entities** or **CSS** we can give spaces in HTML. HTML collapses **multiple white spaces** into a **single space**. To add extra spaces, we can use the HTML entity ** ** or use CSS properties like **margin** and **padding**.

```
php
```

```
<p>Extra space: Hello    World!</p>
```

How to add space using `<pre>` tag in HTML?

The **<pre> (Preformatted Text)** tag is used when we want to preserve **both spaces and line breaks**, which means text appears exactly as written in the HTML.

```

php
Copy code



```

 Text with multiple spaces.

```


```

How to run HTML program?

To run an **HTML program**, **save** the HTML code in a file with a **.html** extension and then **open** that file using a **web browser**. The browser will **render** the HTML code as a webpage.

Example: Save the HTML code in a file named `index.html` and double-click to open it in your default web browser.

How to save HTML file?

We can write **HTML code** in a **text editor** like **Notepad**, **Sublime Text**, or **Visual Studio Code**, and then **save** the file with an **.html** extension. **Example:** Save the file as `my_page.html`.

How to select multiple options from a drop-down list in HTML?

To allow **multiple selections** in a drop-down list, add the **multiple attribute** to the **<select>** tag.

```
html
```

 Copy code

```
<select multiple>
  <option value="apple">Apple</option>
  <option value="banana">Banana</option>
  <option value="cherry">Cherry</option>
</select>
```

What is the difference between `<figure>` tag and `<img>` tag?

`<figure>` is for **grouping images and captions**. The `<img>` tag is used for **embedding images**, while the `<figure>` tag is a **semantic** way to group **images** with their **captions** (`<figcaption>`).

```
html Copy code

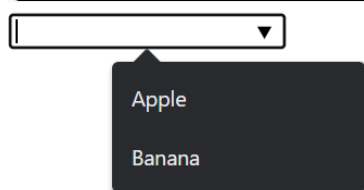
<figure>
  
  <figcaption>Caption for the image.</figcaption>
</figure>
```

Is the `<datalist>` tag and `<select>` tag same?

No, while both `<datalist>` & `<select>` are used for creating **dropdown lists**, `<datalist>` allows for an **autocomplete** feature, letting users type to **filter** options.

```
html Copy code

<input list="fruits">
<datalist id="fruits">
  <option value="Apple">
  <option value="Banana">
</datalist>
```



What is a `fieldset` tag in HTML?

The `<fieldset>` tag is used to group related elements in a form. It draws a box around the related elements and can work with the `<legend>` tag to provide a caption.

```
html Copy code

<fieldset>
  <legend>Personal Information:</legend>
  Name: <input type="text"><br>
  Email: <input type="text"><br>
</fieldset>
```



How to add a scrollbar in HTML?

We can add a **scrollbar** to an HTML element by setting the CSS `overflow` property to **auto** or **scroll**.

What is CSS?

CSS stands for **Cascading Style Sheets**, which is used to **style** and **structure** the layout of webpages.

Why do we need CSS?

CSS is used to style an **HTML** document. by choosing **fonts**, applying **colors**, defining **margins** etc. So, the HTML documents would look pretty boring without **CSS**.

What are the advantages and disadvantages of CSS?

Advantages of CSS

- **CSS** saves a lot of time.
- It helps to make **consistent** and **spontaneous** changes.
- It improves the **loading speed** of the page.
- It has better device **compatibility**.

Disadvantages of CSS

- There could be **cross-browser** issues while using **CSS**.
- There are multiple versions of **CSS** such as **CSS**, **CSS 2**, **CSS 3**. This can create a confusion for non-developers and beginners.

What is the Internal Style Sheet and External Style Sheet in CSS?

Internal Style Sheet: In Internal Style Sheet, CSS property-values are defined in the HTML **style** element inside the HTML **head** element. It is used to define the styles for a **single** HTML document.

External Style Sheet: In External Style Sheet, CSS property-values are defined in a separate **.css** file. It can be used to define the styles for **many** HTML documents.

How to link the CSS file to the HTML file?

The HTML **link** element is used to link the **CSS** file to the **HTML** file. It should be placed in the HTML **<head>** element. This element has attributes like **rel** and **href** to specify the **relationship** between the **files** and the **location** of the CSS file.



```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <link rel="stylesheet" href="tourism.css" />
5   </head>
6   <body>
7     ...
8   </body>
9 </html>
```

HTML

How to add CSS to the HTML document?

Inline CSS: Adding **CSS** directly to **HTML** elements using the **style** attribute.

```
<p style="color: blue; font-size: 16px;">This is a blue paragraph with larger text.</p>
<div style="background-color: yellow; padding: 10px;">This is a yellow box with padding.</div>
```

Internal CSS: Placing **CSS** within the **<style>** element in the **<head>** section of the **HTML** document.

```
4 <style>
5   p {
6     color: green;
7     font-size: 18px;
8   }
9   .highlight {
10    background-color: orange;
11  }
12 </style>
```

External CSS: Linking an external **CSS** file to the **HTML** document using the **<link>** element in the **<head>** section.

```
html
Copy code

<!DOCTYPE html>
<html>
<head>
  <link rel="stylesheet" type="text/css" href="styles.css">
</head>
<body>
  <!-- Your HTML content here -->
</body>
</html>
```

What is a responsive web design?

Responsive web design is used to **create** web pages that look good on **all devices**.

What are the CSS Styles?

CSS styles are rules that determine how **HTML** elements look on a webpage.

Explain Inline styles

The **Inline styles** are applied directly to an **HTML** element. They use the **HTML style** attribute, with CSS property values defined within it.

What is CSS Selector & what are the different types of it?

The **CSS Selectors** are used to **select** the **HTML** elements that we want to **style**.

- **Type Selector:** Targets elements by their tag name (**p** for paragraphs).
- **Class Selector:** Targets elements with a specific class (**.my-class**).
- **ID Selector:** Targets a single element by its ID (**#my-id**).
- **Universal Selector:** Selects all elements (*****).
- **Attribute Selector:** Targets elements with specific attributes and values (**[type="text"]**).
- **Descendant Selector:** Targets elements that are descendants of another element (**ul li**).
- **Child Selector:** Targets direct children of an element (**ul > li**).
- **Adjacent Sibling Selector:** Targets an element immediately preceded by another element (**h2 + p**).
- **General Sibling Selector:** Targets elements that are siblings of another element (**h2 ~ p**).
- **Pseudo-classes:** Selects elements based on their state or position (**:hover**, **:nth-child()**).
- **Pseudo-elements:** Selects specific parts of an element (**::before**, **::after**).

Explain Low to high specificity in CSS Selectors?

The following list of **CSS Selectors** is in the **lowest** to **highest** order by specificity.

- Universal Selector
- Type/Tag Name Selector
- Class Selector
- ID Selector
- Inline Selector (Inline styles)

Which has more specificity among Inline Styles and Class Selectors?

Inline Styles have more specificity than **Class Selectors**.

Explain the difference b/w HTML id attribute and class attribute?

Attribute	Scope	Specificity in CSS	JavaScript Access	Global/Local Styles
<code>`id`</code>	Limited to one element	High specificity	Direct access by <code>`getElementById`</code>	Local
<code>`class`</code>	Can be applied to multiple elements	Medium specificity	Accessed through various methods	Global

What CSS property-values will be applied to an HTML element if it has both HTML id and class attributes?

When an **HTML** element has both **id** and **class** attributes, styles from **both** selectors apply. If there's a conflict, styles from the **id** selector apply due to **higher specificity**.

How do you change the style of an HTML element with and without a CSS Class Selector?

Without CSS Class Selector: Using Inline Styles: Apply styles directly to an element using the **style** attribute in HTML. `<p style="color: blue; font-size: 16px;">Blue text with inline styles</p>`

With CSS Class Selector: Defining and Applying Class: Define a class in CSS and apply it to HTML elements using the **class** attribute. `<p class="special">Highlighted and bold text</p>`

What is a Universal selector?

The **Universal selector** in CSS is denoted by the asterisk (*). It applies styles to all HTML elements on a web page.

What are the CSS text properties?

- **color:** Sets the color of the text.
- **text-align:** Specifies the alignment of the text within its container.
- **text-decoration:** Adds visual effects to text (e.g., underline, overline, line-through).
- **direction:** Sets the text direction, such as left-to-right or right-to-left.
- **letter-spacing:** Adjusts the space between characters.
- **text-shadow:** Adds a shadow effect to text.
- **text-transform:** Controls the capitalization of text (e.g., uppercase, lowercase, capitalize).
- **word-spacing:** Adjusts the space between words.
- **text-indent:** Sets the indentation of the first line of text within a block.

What are the CSS Background properties?

- **background-color:** Sets the background color of an element.
- **background-image:** Specifies an image to be used as the background.
- **background-position:** Determines the starting position of the background image.
- **background-size:** Sets the size of the background image.
- **background-repeat:** Defines how the background image should be repeated.
- **background-origin:** Determines where the background image's position is calculated from.
- **background-clip:** Specifies how the background image should be clipped within the element's borders.
- **background-attachment:** Sets whether the background scrolls with the content or remains fixed.

How to stop background image repeating for empty space?

We can use the **CSS** property **background-repeat** with the value **no-repeat** to stop repeating the background image.

Explain CSS background attachment property?

Background-attachment is a CSS property that controls how a **background image** behaves when a **webpage** is **scrolled**.

- **scroll**: The background image scrolls along with the content.
- **fixed**: The background image stays fixed in place as we scroll, creating a parallax effect

What is a CSS Gradient?

A **smooth transition** between **two** or **more colors** used as a **background image**.

There are mainly two types of gradients:

- Linear Gradient
- Radial Gradient

Linear Gradient: It is the most basic type of gradient. To create a gradient, we must have at least **two** colors. By default, **linear gradients** run from **top** to **bottom**. we can change their transition by specifying a **direction**.

```
1 <div class="linear-gradient-background"></div>
```

HTML

```
1 .linear-gradient-background {  
2   background-image: linear-gradient(to right, #2196f3, #f44336);  
3   height: 100vh;  
4 }
```

CSS

Direction	Description
to top	Colors transition (change) from bottom to top
to bottom	It is a default direction. Colors transition (change) from top to bottom
to left	Colors transition (change) from right to left
to right	Colors transition (change) from left to right

Radial Gradient: Radial Gradients **radiate** out from a **central point**.

```
1 <div class="radial-gradient-background"></div>
```

HTML

```
1 .radial-gradient-background {  
2   background-image: radial-gradient(#2196f3, #f44336);  
3   height: 100vh;  
4 }
```

CSS

What is the CSS box shadow property?

CSS **box-shadow** is a property that adds a **shadow effect** around an **element's box**.

Value	Description
none	Default value. No shadow is displayed
h-offset	(Required) It specifies the horizontal distance. A positive value puts the shadow on the right side of the box, a negative value puts the shadow on the left side of the box
v-offset	(Required) It specifies the vertical distance. A positive value puts the shadow below the box, a negative value puts the shadow above the box
blur	(Optional) It specifies the blur radius. The higher the number, the more blurred the shadow will be
spread	(Optional) The spread radius. A positive value increases the size of the shadow, a negative value decreases the size of the shadow
color	(Optional) It specifies the color of the shadow
inset	(Optional) If not specified (default), the shadow is a drop shadow (box seems raised above the content). If specified, the shadow is an inset shadow (shadows are drawn inside the border, above the background, but below content)

What is the CSS border radius property?

The CSS **border-radius** property specifies the **roundness** of the **four corners** of an **HTML element**.

What is the sequence of CSS margin properties?

The sequence of CSS margin properties is **margin-top**, **margin-right**, **margin-bottom**, and **margin-left**.

How to create a circular HTML button element using CSS?

We can create a circular HTML button element using the CSS property **border-radius**.

- Apply the **same width** and **height** to the HTML **button** element
- Apply the CSS **border-radius** property with the value **50%** to the HTML button element.

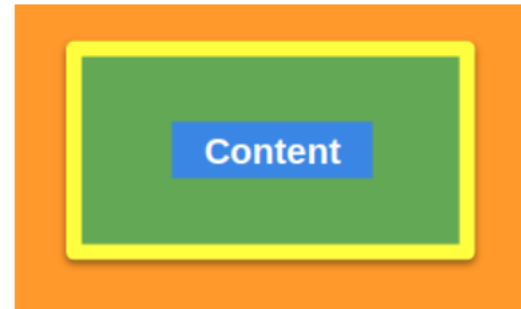
```
1 .button {  
2   background-color: skyblue;  
3   width: 100px;  
4   height: 100px;  
5   border-radius: 50%;  
6 }
```

CSS

What is the CSS box model?

The **CSS box model** is a concept where each **HTML** element is enclosed in a box with four layers: **content**, **padding**, **border**, and **margin**.

- **Content (Blue):** It holds the actual content like text and images.
- **Padding (Green):** Provides space around the content.
- **Border (Yellow):** It is a line around the padding and content.
- **Margin (Orange):** Represents the space outside the border.



What are the CSS Box properties?

- **height:** Controls the vertical size of an element.
- **width:** Manages the horizontal size of an element.
- **border-style:** Determines the style of the element's border (dashed, solid, etc.).
- **border-width:** Sets the thickness of the element's border.
- **border-color:** Defines the color of the element's border.
- **margin:** Creates space outside the element's border.
- **padding:** Generates space between the content and the element's border.

What is the CSS box sizing property?

The CSS **box-sizing** property sets how an element's total width and height are calculated, with values **content-box** (default, includes only content) and **border-box** (includes content, padding, border, but not margin).

How to create a responsive website without Bootstrap?

We can create a **responsive website** using **media queries** or **flexbox** without **Bootstrap**.

Explain media queries and why are they used?

Media queries play a crucial part while developing **Responsive Layouts**. Using Media queries, we can conditionally apply styles based on the **device type** (e.g., screen, print, tv, all, etc.) and media features (e.g., width, height, orientation.).

```
1 @media media-type and (media-feature-expression) {  
2   /* CSS rules go here */  
3 }
```

What is a media type in a media query?

Media types describe the general category of devices. If the media type is not declared, it will default to **all**.

screen: For all screened devices like mobiles, laptops, tablets, etc., **print:** For printers, **tv:** For televisions, **all:** Matches all types of devices and media.

What is the syntax for a CSS Media Query?

```
@media media-type and (media-feature-expression) {  
    /* CSS rules go here */  
}
```

- **Media types** describe the general category of devices. Examples: **screen**, **print**, **tv**, **all**.
- **Media Features** include features like **width**, **height**, **orientation**, etc.

We can combine multiple Media Features in a single Media Query using **Logical Operators** and the **comma operator**.

```
1 @media (orientation: landscape), (min-width: 600px) {  
2     .bg-container {  
3         background-color: yellow;  
4     }  
5 }
```

CSS

How to align the two HTML container (div) elements in a single row?

Making the HTML container elements as inline elements:

Apply CSS display property with the value **inline** to both the HTML container (div) elements.

```
<body>  
  <div class="inline-element">Element 1</div>  
  <div class="inline-element">Element 2</div>  
</body>
```

```
2 .inline-element {  
3     display: inline;  
4 }
```

Making the HTML container elements as flex-items:

Wrap the two HTML container elements in another container element. Apply CSS display property with the value **flex** to the parent element of two HTML container elements

```
<body>  
  <div class="flex-container">  
    <div class="flex-element">Element 1</div>  
    <div class="flex-element">Element 2</div>  
  </div>  
</body>
```

```
2 .flex-container {  
3     display: flex;  
4 }
```

How to apply background color for an HTML element in only small devices?

We can apply **background color** for an HTML element in only **small devices** using **media queries** with the **media feature max-width**.

CSS

```
1 @media (max-width: 576px) {  
2   h1 {  
3     background-color: blue;  
4   }  
5 }
```

How to hide the container (div) element on mobiles and show it on iPad and desktop?

We can hide the container (div) element in mobile and show it on iPad and desktop using **Media Queries** or **Bootstrap**.

Using Media Queries

```
1 div {  
2   display: none;  
3 }  
4  
5 @media (min-width: 576px) {  
6   /*Assuming the tablet screen width is 576px.*/  
7   div {  
8     display: block;  
9   }  
10 }
```

Using Bootstrap.

```
<!DOCTYPE html>  
<html>  
<head>  
  <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css">  
</head>  
<body>  
  <div class="container d-none d-md-block">  
    <!-- Content of the container goes here -->  
  </div>  
</body>  
</html>
```

What is CSS Flexbox?

The **CSS Flexbox** is a layout method that helps to arrange the HTML elements in rows (horizontally) or columns (vertically).

What are the uses of CSS Flexbox?

CSS Flexbox helps control how things **align** and **behave** when **screens change**.

Some ways it's useful:

1. **Flexible Size:** Elements adjust size as needed.
2. **Easy Centering:** Elements can easily be centered.
3. **Change Order:** We can change element order without changing the code.

What is the CSS display property in Flexbox?

The **CSS display** property in **Flexbox** sets the layout of the **children** elements. Providing display property with the value **flex** converts the element to a **Flexbox Container**. All the HTML elements that are direct children of Flexbox Container are called **flex items**.

What are the CSS Flexbox properties?

- **display**: Specifies the display type, with **flex** creating a Flexbox container.
- **flex-direction**: Sets the direction of flex items within the container.
- **justify-content**: Aligns flex items along the main axis.
- **align-items**: Aligns flex items along the cross axis.
- **flex-wrap**: Determines whether flex items should wrap or stay in a single line.
- **flex-flow**: Shorthand for **flex-direction** and **flex-wrap**.
- **align-content**: Aligns flex lines within the container along the cross axis.
- **align-self**: Overrides **align-items** for a specific flex item.
- **flex-grow**: Defines the ability of a flex item to grow.
- **flex-basis**: Specifies the initial size of a flex item.
- **flex-shrink**: Defines the ability of a flex item to shrink.
- **order**: Specifies the order of a flex item within the container.

What is meant by the CSS justify content property?

The CSS **justify-content** property specifies the alignment of **flex-items** along the **main axis**.

Value	Description
flex-start (default)	flex-items will arrange to the start of the flexbox container
center	flex-items will arrange to the center of the flexbox container
flex-end	flex-items will arrange to the end of the flexbox container
space-between	Left-over space will be arranged in between the flex-items
space-around	Left-over space will be arranged around each flex-item

When do we use the CSS percentages?

CSS percentages used for sizing child elements relative to their parent's dimensions in responsive designs.

```
css Copy code

.parent {
  width: 80%; /*The parent container takes up 80% of its parent's width*/
}
.child {
  width: 50%; /*The child div takes up 50% of the parent's width*/
}
```

What is CSS flex property?

The CSS **flex** property is a shorthand that combines **flex-grow**, **flex-shrink**, and **flex-basis**. It controls how a flex item grows, shrinks, and its initial size within a Flexbox container.

```
1 flex: none /* value 'none' case */
2 flex: <'flex-grow'> /* One value syntax, variation 1 */
3 flex: <'flex-basis'> /* One value syntax, variation 2 */
4 flex: <'flex-grow'> <'flex-basis'> /* Two values syntax, variation 1 */
5 flex: <'flex-grow'> <'flex-shrink'> /* Two values syntax, variation 2 */
6 flex: <'flex-grow'> <'flex-shrink'> <'flex-basis'> /* Three values syntax */
```

CSS

Example: Using three values syntax

```
1 .container {
2   flex: 0 1 1;
3 }
```

CSS

Common Values for CSS flex property are:

Value	Description
flex: 0 auto; (Default)	Same as flex: initial; and is shorthand for flex: 0 1 auto;
flex: auto;	Same as flex: 1 1 auto;
flex: none;	Same as flex: 0 0 auto;

Explain CSS Flexbox properties?

Flex Direction:

The CSS **flex-direction** property specifies the direction of the **flex-items** in the Flexbox Container.

Value	Description
row	Direction of the flex-items is horizontal
column	Direction of the flex-items is vertical

Justify Content:

The CSS **justify-content** property specifies the alignment of **flex-items** along the **main axis**.

Value	Description
flex-start (default)	flex-items will arrange to the start of the flexbox container
center	flex-items will arrange to the center of the flexbox container
flex-end	flex-items will arrange to the end of the flexbox container
space-between	Left-over space will be arranged in between the flex-items
space-around	Left-over space will be arranged around each flex-item

Align Items:

The CSS **align-items** property specifies the alignment of **flex-items** along the **cross axis**.

Value	Description
stretch (default)	If the flex-direction is row , flex-items will stretch to their available height.
flex-start	flex-items will be at the start of the flexbox container
center	If the flex-direction is row , flex-items will be at the center of the available height in the flexbox container
flex-end	If the flex-direction is row , flex-items will be at the end of the available height in the flexbox container

What is the CSS display property?

The CSS **display** property is crucial for managing layout in web design. With this property, we can:

1. Convert HTML elements into **inline** or **block-level** elements.
2. Control the layout of **child elements**.

Here are some common values of the **display** property:

- **block**: Generates a block-level box.
- **inline**: Generates an inline-level box.
- **flex**: Turns an element into a flex container.
- **inline-flex**: Makes an inline-level flex container.
- **grid**: Creates a grid container.
- **none**: Element is hidden.

Explain CSS display property and its value inline-block?

The CSS **display** property with the value **inline-block** displays the HTML element as an **inline-block container**. The inline-block container doesn't start on a **new line** but height and width can be applied to it.

How to convert inline elements to block-level elements?

The CSS property **display** with the value **block** converts any HTML element to a **block-level element**.

```
1 .container {  
2   display: block;  
3 }
```

Explain CSS display property with the values inline and block?

display: block:

- Converts an HTML element to a **block-level element**.
- **Starts on a new line**.
- Occupies the **full** available width horizontally.

display: inline:

- Converts an HTML element to an **inline element**.
- **Doesn't start on a new line**.
- Takes up only the **necessary** width horizontally.

Difference between “display: none” and “visibility: hidden”

display: none	visibility: hidden
1. It doesn't display the HTML element	1. It hides the HTML element (i.e cannot be seen in the layout/web page)
2. Space will not be allocated for an HTML element in the layout	2. Space will be allocated for an HTML element in the layout

What is the CSS visibility property?

The CSS **visibility** property specifies the visibility of an HTML element.

Value	Description
visible	shows the element
hidden	hides the element but space will be allocated to the element in the HTML document

Example:

CSS

```
1 .container {  
2   visibility: hidden;  
3 }
```

How can we layout the page for large screens and small screens?

We can build different layouts for **large screens** and **small screens** using:

- Bootstrap Grid System

```
<div class="container">  
  <div class="row">  
    <div class="col-md-6 col-sm-12">  
      <!-- Content for large screens (≥768px) -->  
    </div>  
    <div class="col-md-6 col-sm-12">  
      <!-- Content for large screens (≥768px) -->  
    </div>  
  </div>  
</div>
```

- Media Queries

```
@media (min-width: 768px) {  
  .content {  
    width: 50%;  
    float: left;  
  }  
}
```

```
<div class="content">  
  <!-- Content that adjusts based on screen size -->  
</div>
```

- Flexbox

```
.container {  
  display: flex;  
}  
  
.item {  
  flex: 1;  
  /* Additional styles */  
}
```

```
<div class="container">  
  <div class="item">  
    <!-- Content for both large and small screens -->  
  </div>  
  <div class="item">  
    <!-- Content for both large and small screens -->  
  </div>  
</div>
```

How to style the visited and unvisited links using CSS?

We can style the **visited** and **unvisited** links with the **CSS pseudo-classes** **:link** and **:visited**.

Unvisited links: Using `:link` or other selectors like tag, class name, etc.

```
1 a:link {  
2   color: red;  
3 }
```

CSS

Visited links: Using `:visited`

```
1 a:visited {  
2   color: green;  
3 }
```

CSS

All the visited links will be in green color whereas all unvisited links are in red color.

What are the differences between static, relative, fixed, absolute and sticky positioning?

Property	Description	Positioning Behavior	Affects Document Flow	Scroll Behavior	Relative to
Static	Default positioning in normal document flow.	No	Yes	Scrolls with page	None (normal flow)
Relative	Positioned relative to its normal position.	Yes	Yes	Scrolls with page	Itself (relative to normal)
Absolute	Positioned relative to nearest positioned ancestor or viewport if none.	No	No	Scrolls with page	Ancestor or viewport
Fixed	Positioned relative to the viewport.	No	No	Does not scroll	Viewport
Sticky	Initially relative but becomes fixed on scrolling past a specified point.	Yes	Yes	Scrolls with page	Ancestor or viewport

What is CSS Positioning?

The CSS **position** property defines the position of an HTML element in an HTML document. There are **five** values the CSS position property can take. They are:

- *static*
- *relative*
- *absolute*
- *fixed*
- *sticky*

1. **Static:** Default positioning, elements appear in normal flow.

CSS

 Copy code

```
position: static;
```

2. **Relative:** Positioned relative to its normal position.

CSS

 Copy code

```
position: relative;
top: 10px; /* Moves 10px down from normal position */
left: 20px; /* Moves 20px right from normal position */
```

3. **Absolute:** Positioned relative to nearest positioned ancestor.

CSS

 Copy code

```
position: absolute;
```

4. **Fixed:** Positioned relative to the viewport, doesn't scroll.

CSS

 Copy code

```
position: fixed;
top: 0; /* Stays at the top of the viewport */
left: 0; /* Stays at the left of the viewport */
```

5. **Sticky:** Initially like relative, becomes fixed on scrolling.

CSS

 Copy code

```
position: sticky;
top: 20px; /* Becomes fixed 20px from the top */
```

How to wrap a block of text around the HTML image element?

We can **wrap** a block of text around the HTML image element using the CSS **float** property.

Value	Description
none	(Default value) The element does not float (will be displayed just where it occurs in the text)
left	The element floats to the left of its container
right	The element floats to the right of its container

CSS

```
1 img {  
2   float: right;  
3 }
```

What is the CSS direction property?

The CSS **direction** property sets the direction of **text**, **table columns**, and **horizontal overflow** in an HTML **block-level element**.

Value	Description
ltr	(Default value) Text and other elements go from left to right
rtl	Text and other elements go from right to left

Example:

CSS

```
1 .card {  
2   direction: rtl;  
3 }
```

Explain what is Bootstrap and its importance?

Bootstrap is a large collection of **predefined reusable Code Snippets** written in **HTML**, **CSS**, and **JavaScript**. It is developed by **Twitter**. Bootstrap can be used to develop desktop and mobile applications.

What are the advantages of Bootstrap?

Bootstrap saves time with reusable code, ensures consistency across browsers, allows customization, and is beginner-friendly.

- **.container**
- **.row**
- **.col-*** (e.g., **.col-12**, **.col-md-6**)
- **.btn**
- **.btn-primary**
- **.text-center**
- **.text-danger**

What are the different button styles in Bootstrap 4?

Button Styles	Class names
Buttons with different colors	btn-primary , btn-secondary , etc
Outline buttons with different colors	btn-outline-primary , btn-outline-secondary , etc.
Buttons with different sizes (Small, large)	btn-sm , btn-lg
Block-level buttons	btn-block

Design two HTML buttons, success and danger, using Bootstrap class names.

HTML

```
i 1 <button class="btn btn-success">Get Started</button>
  2 <button class="btn btn-danger">Get Started</button>
```

How to add an HTML button and style it using Bootstrap to the existing HTML document?

- Add the HTML button element in the HTML **<body>** element.
- Include the Bootstrap CDN in the HTML **<head>** element.
- Add the Bootstrap button class names to the HTML **button** element.

HTML

```
i 1 <button class="btn btn-success">Get Started</button>
```

What is a Carousel?

The **Carousel** is a slideshow for cycling through **images, text, etc.** Slides will change for every few seconds.

What is the Bootstrap grid system?

Bootstrap Grid System is a collection of reusable code snippets to create responsive layouts. It is made up of **containers, rows, and columns**.

It uses a **12-column** system for the lay outing. We can create up to twelve columns across the page.

- **Container:** The purpose of a container is to hold rows and columns.
- **Row:** The purpose of a row is to wrap all the columns.
- **Column:** We should place the columns inside a row and the content inside a column.

We can specify the number of columns our content should occupy in any device. The number of columns we specify should be a number in the range of 1 to 12.

The Bootstrap class names **col-*** indicates the number of columns we would like to use out of the possible twelve columns per row. **For example**, col-1, col-5, etc.

Explain the Bootstrap class names d-none and d-block?

Class Name	Description
d-none	Doesn't display an HTML element and doesn't take any space in the HTML document.
d-block	Displays an HTML element

What is Flexbox Utility in Bootstrap?

- The `.d-flex` class name enables the flex behaviours.

Flexbox property	Class Names
flex-direction	flex-row , flex-column , etc.
justify-content	justify-content-start , justify-content-end , justify-content-center , justify-content-between , justify-content-around , etc.
align-items	align-items-start , align-items-end , align-items-center , align-items-baseline , align-items-stretch , etc.
align-self	align-self-start , align-self-end , align-self-center , align-self-baseline , align-self-stretch , etc.

What are the typical screen sizes for building a responsive web design?

Device	Device Size (Width)
Extra small devices	<576px
Small devices	>=576px
Medium devices	>=768px
Large devices	>=992px
Extra large devices	>=1200px

What is the difference between Frontend and Backend Development?

Frontend Development: Frontend development, also known as **client-side** development, involves designing and creating the **user interface** and **user experience (UI/UX)** of a website or web application using **HTML, CSS, and JavaScript**.

Backend Development: Backend development, also known as **server-side** development, involves managing functionality, including data storage, processing user requests, and executing business logic using **programming languages** and **databases**.