HTML & CSS

https://www.tutorialspoint.com/html/

HTML is a standard markup language, which stands for HyperText Markup Language. It is a widely used language to create webpages.

HyperText defines the internal links between web pages, and Markup language defines the layout and presentation of text and media.

Basic HTML Structure

```
<!DOCTYPE html>

<html>

<head>

<title>Page title</title>

</head>

<body>

<h1>Webpage's Heading</h1>

<p>Content (Your first paragraph).</p>

</body>

</html>
```

zElements of HTML Basic Structure

The following are the basic tags that define the basic HTML structure −

1. <!DOCTYPE html>

This element defines the document type as HTML. This element must be written before writing any HTML document.

2. <html>...</html>

The <html> tag is the parent tag for all HTML elements. Everything related to create an HTML document must be written inside the <html> tag. CSS, JavaScript, and jQuery must also be written inside this tag.

3. <head>...</head>

The <head> tag is a container tag for all those elements that are not directly displayed on the webpage but required for the page functionalities. It contains meta tags (which are used for SEO purposes), title tag, script tags, etc.

4. <title>...</title>

The <title> tag is used to define the title of the webpage that you can see in the browser's tab, bookmarks list, and search engine results. This tag is also very important for SEO purposes to help search engine to understand the content of the webpage.

5. <body>...<body>

The <body> tag is the container tag for all those elements, which represents the main content of a webpage that displays on the browser.

6. <h1>...</h1>

The <h1> tag is one of the heading tags. It is the most important heading tag, which defines the main title or headline of the webpage. Any text written inside <h1> and </h1> is a top-level heading of the content.

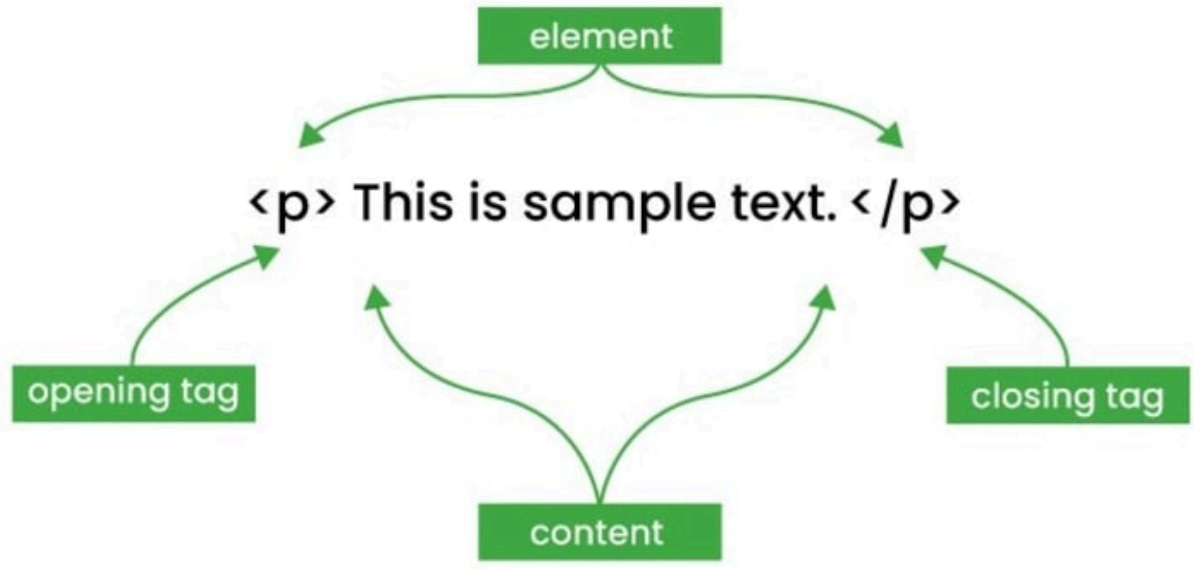
7. ...

The tag defines a paragraph, anything written inside and displays as a paragraph on the webpage. Use multiple tags to display text in different paragraphs.

What is an HTML Element?

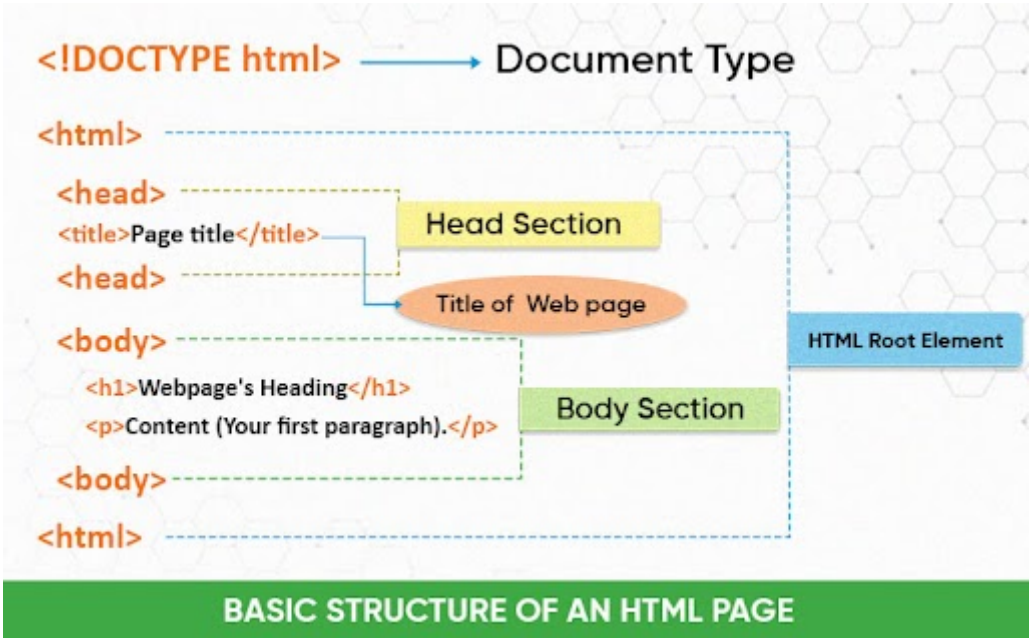
An HTML element is a basic building block to create a webpage, and It is created by a start tag, content, and end tag. In an HTML element, the content is placed between a start and end tag.

The basic syntax of an HTML element is –



<tag_name>content</tag_name>

<h1>It is top-level heading</h1>



Web Browser Role

The role of a web browser is to read HTML documents from the given path (either from the server or from a local device) and display it on the webpages. All web browsers, such as

Google Chrome, Safari, Firefox, etc., are compatible with reading HTML documents. You can use any of the web browsers to display your HTML document in web format.

The <!DOCTYPE> Declaration

The <!DOCTYPE> declaration tag is used by the web browser to understand the version of the HTML used in the document. The current version of HTML is 5 and it makes use of the following declaration −

<!DOCTYPE html>

HTML Tags Vs. Elements Vs. Attributes

HTML tags are the keywords that can be used for a specific purpose to display and format the content on the webpage.

HTML elements are the basic building blocks that are made with the help of tags and content. An HTML element is created with a start tag, a content, and an end tag.

And, HTML attributes provide additional information about HTML elements; in order to define or change their behavior. Attributes are used with an opening tag.

HTML Tags Case Sensitivity

HTML tags are not case-sensitive. They can be written in uppercase or in lowercase. But the World Wide Web Consortium (W3C) recommends using lowercase tags starting from HTML 4.

Popular HTML Editors

Notepad

TextEdit

Notepad++

Sublime

Visual Studio Code

Atom

Brackets

Adobe Dreamweaver CC

CoffeeCup

HTML - Basic Tags

HTML tags are the fundamental elements of HTML used for defining the structure of the document. These are letters or words enclosed by angle brackets (< and >).

Heading Tags	<h1>, <h2>, <h3>, <h4>, <h5>, and <h6>
Paragraph Tag	<p>
Line Break Tag	 or
Center Tag	
Horizontal Rule Tag	<hr> or <hr />
Preserve Formatting Tag	<pre>
Non-breaking Space	
Listing Tags	and • Unordered Listing: • Ordered Listing: • List Item:

HTML Basic Tags

HTML Void Elements

HTML void elements are those elements that don't require closing tags. These tags don't have any content model and don't even allow nesting of elements. The void elements are also known as empty or self-closing elements.

Some of the void elements are such as <img />, <hr />, and
 elements. The below table shows a list of void elements −

HTML - Attributes

HTML attributes are special words that provide additional information to an HTML element. Attributes are placed inside the element's opening tag, and they are used to configure or adjust the element's behavior. All attributes are made up of two parts: a name and a value −

```
<p id="html">This is a paragraph.</p>

Name: The attribute name (id, class, src, etc.).

Value: The content or value assigned to that attribute
("html", "example.jpg", etc.).
```

```
<!-- Specifies the source file for an image -->
<img src="image.jpg" alt="Description of image">

<!-- Defines a hyperlink to another page or resource -->
<a href="https://www.example.com">Visit Example</a>

<!-- Sets an alternative text for an image if the image cannot be displayed -->
<img src="logo.png" alt="Company Logo">

<!-- Specifies the text for a button -->
<button type="button">Click Me!</button>

<!-- Defines the target for a hyperlink (where the link will open) -->
<a href="https://www.example.com" target="_blank">Open in New Tab</a>

<!-- Sets a unique identifier for an element -->
<div id="header">Header Content</div>

<!-- Specifies the class for an element to apply CSS styles -->
```

```
<div class="container">This is a container</div>

<!-- Defines a style to apply CSS directly to an element -->
<p style="color: blue;">This is a blue paragraph.</p>

<!-- Specifies the width of an element -->
<img src="image.jpg" width="300">

<!-- Specifies the height of an element -->
<img src="image.jpg" height="200">

<!-- Specifies the type of input field -->
<input type="text" placeholder="Enter your name">

<!-- Indicates that an input field is required to submit a form -->
<input type="email" required>

<!-- Specifies the action for a form (where data is submitted) -->
<form action="submit_form.php" method="POST">Form Content</form>

<!-- Sets the language of the document -->
<html lang="en">

<!-- Specifies the title of the document -->
<title>My Web Page</title>

<!-- Defines a meta tag for character set -->
<meta charset="UTF-8">

<!-- Specifies the autocomplete behavior of an input field -->
<input type="text" autocomplete="on">

<!-- Specifies a placeholder text in a form input -->
<input type="text" placeholder="Enter your email">

<!-- Defines the size of an input field -->
<input type="text" size="30">

<!-- Specifies a value to pre-fill an input field -->
<input type="text" value="John Doe">
```

```
<!-- Specifies the disabled state for an element (disables it) -->
<button disabled>Disabled Button</button>

<!-- Specifies a name for form elements to group them -->
<input type="radio" name="gender" value="male"> Male
<input type="radio" name="gender" value="female"> Female

<!-- Specifies the type of the button (submit, reset, etc.) -->
<button type="submit">Submit Form</button>

<!-- Specifies the language of the document -->
<p lang="en">This is an English text.</p>

<!-- Sets a minimum value for an input element -->
<input type="number" min="1">

<!-- Specifies the maximum value for an input element -->
<input type="number" max="100">

<!-- Sets the autofocus attribute on an element -->
<input type="text" autofocus>

<!-- Specifies the maximum length of an input field -->
<input type="text" maxlength="10">

<!-- Defines a background color for an element -->
<div style="background-color: lightblue;">This is a background-colored div</div>

<!-- Specifies the source file for audio -->
<audio src="audio.mp3" controls></audio>

<!-- Specifies the source file for video -->
<video src="video.mp4" controls></video>

<!-- Specifies whether an element is visible or not -->
<div hidden>This content is hidden</div>

<!-- Specifies a custom data attribute -->
<div data-user-id="123">User Info</div>

<!-- Specifies the role of an element for accessibility -->
```

```
<div role="button">This is a button</div>

<!-- Specifies the type of list item (used for ordered lists) -->
<ol>
  <li value="10">Item 1</li>
  <li>Item 2</li>
</ol>

<!-- Defines a relationship between the current document and a linked resource -->
<link rel="stylesheet" href="styles.css">

<!-- Specifies that an input field accepts numeric values -->
<input type="number" step="0.1" value="3.14">

<!-- Specifies the form method (GET or POST) -->
<form method="POST">Form Content</form>

<!-- Specifies that an input field is of password type -->
<input type="password" placeholder="Enter your password">

<!-- Defines a list of options for a drop-down menu -->
<select>
  <option value="apple">Apple</option>
  <option value="orange">Orange</option>
</select>

<!-- Defines a textarea element for multi-line text input -->
<textarea rows="4" cols="50">Enter text here</textarea>

<!-- Specifies an element to be focused after page load -->
<input type="text" id="input1" autofocus>

<!-- Specifies whether the user can resize the textarea -->
<textarea resize="both">Resizable Textarea</textarea>

<!-- Specifies an inline style to apply a CSS rule to the element -->
<p style="color:red;">This text is red.</p>


<!DOCTYPE html>
<html>
```

```
  <body>
    <h1>This is Heading 1 (H1 Tag)</h1>
    <h2>This is Heading 2 (H2 Tag)</h2>
    <h3>This is Heading 3 (H3 Tag)</h3>
    <h4>This is Heading 4 (H4 Tag)</h4>
    <h5>This is Heading 5 (H5 Tag)</h5>
    <h6>This is Heading 6 (H6 Tag)</h6>
  </body>
</html>
```

The Tag

You can use the tag to apply inline styles or classes to specific portions of the text within a heading. This allows for custom styling of text within the heading.

```
<h2>This is a <span style="color: blue;">blue</span>
word.</h2>
```

The and Tags

These tags are used for emphasizing text within headings. The tag *italicised* the text, while makes it **bold**.

The <sup> and <sub> Tags

In heading, to include superscript or subscript text within a heading, use <sup> and <sub>.

```
<h4>The 10<sup>th</sup> floor is at the top.</h4>
   <h5>The chemical formula for water is H<sub>2</sub>O.
```

The 10th floor is at the top.

The chemical formula for water is H2O.

The <abbr> Tag for Abbreviations

When you need to include an abbreviation or acronym in a heading, use the **<abbr>** tag. It often provides a tooltip with the full meaning.

```
<h2>HTML stands for <abbr title="Hypertext Markup Language">HTML</abbr>.</h2>
```

The
 Tag for Line Breaks like :- “\n” in programming language.

```
<h3>This is the first line.<br>This is the second line.</h3>
```

This is the first line.

This is the second line.

The <mark> Tag

Use the <mark> tag to highlight specific text within a heading. It's often used to indicate search results or selected portions of text.

```
<h1>Search for "<mark>important</mark>" information here.</h1>
```

Search for "important" information here.

HTML - Block and Inline Elements

HTML block elements are used to create the logical and semantic layout of a web page. They help to organize the content into meaningful sections and make it easier for browsers, search engines, and site visitors to understand the structure and meaning of different parts

of the web page. Inline elements are used to make useful block elements, like adding anchor links.

HTML Block Elements				
<address>	**<article>**	**<aside>**	**<blockquote>**	**<canvas>**
<dd>	** **	**<dl>**	**<dt>**	**<fieldset>**
<figcaption>	**<figure>**	**<footer>**	**<form>**	**<h1> - <h6>**
<header>	**<hr>**	** **	**<main>**	**<nav>**
<noscript>	** **	**<p>**	**<pre>**	**<section>**
<table>	**<tfoot>**	** **	**<video>**	

HTML Inline Elements

Inline elements can appear within the same line and do not start a new line on their own.

HTML Inline Elements				
<a>	**<abbr>**	**<acronym>**	**<b>**	**<bdo>**
<big>	** **	**<button>**	**<cite>**	**<code>**
<dfn>	****	**<i>**	**<img>**	**<input>**
<kbd>	**<label>**	**<map>**	**<object>**	**<output>**
<q>	**<samp>**	**<script>**	**<select>**	**<small>**
****	****	**<sub >**	**<sup>**	**<textarea>**
<time>	**<tt>**	**<var>**		

```
<!-- Outer div element (Block-level) -->

<div class="outer-div">

    <!-- Content inside the outer div (Block-level) -->

    <h1>Welcome to the Outer Div</h1> <!-- Block-level: Takes full width and starts on a new line -->

    <p>This is some text inside the outer div.</p> <!-- Block-level: Paragraph that takes full width and creates space above and below -->



    <!-- Inner div inside the outer div (Block-level) -->

    <div class="inner-div">

        <!-- Content inside the inner div (Block-level) -->

        <h2>Inner Div</h2> <!-- Block-level: Another heading, similar to h1 -->

        <p>This is some text inside the inner div. It is nested inside the outer div.</p> <!-- Block-level: Paragraph inside the inner div -->



        <!-- Example of Inline Elements inside the inner div -->

        <p>This is an inline example:

            <span style="color: red;">This text is inline and appears within the paragraph.</span> <!-- Inline element: The span element only takes as much width as needed and doesn't break the flow of text. -->

        </p>

    </div>

</div>
```

HTML - Formatting

HTML formatting defines the way of content representation on the webpage to improve the readability, to give the semantic meaning, and to improve visual styling.

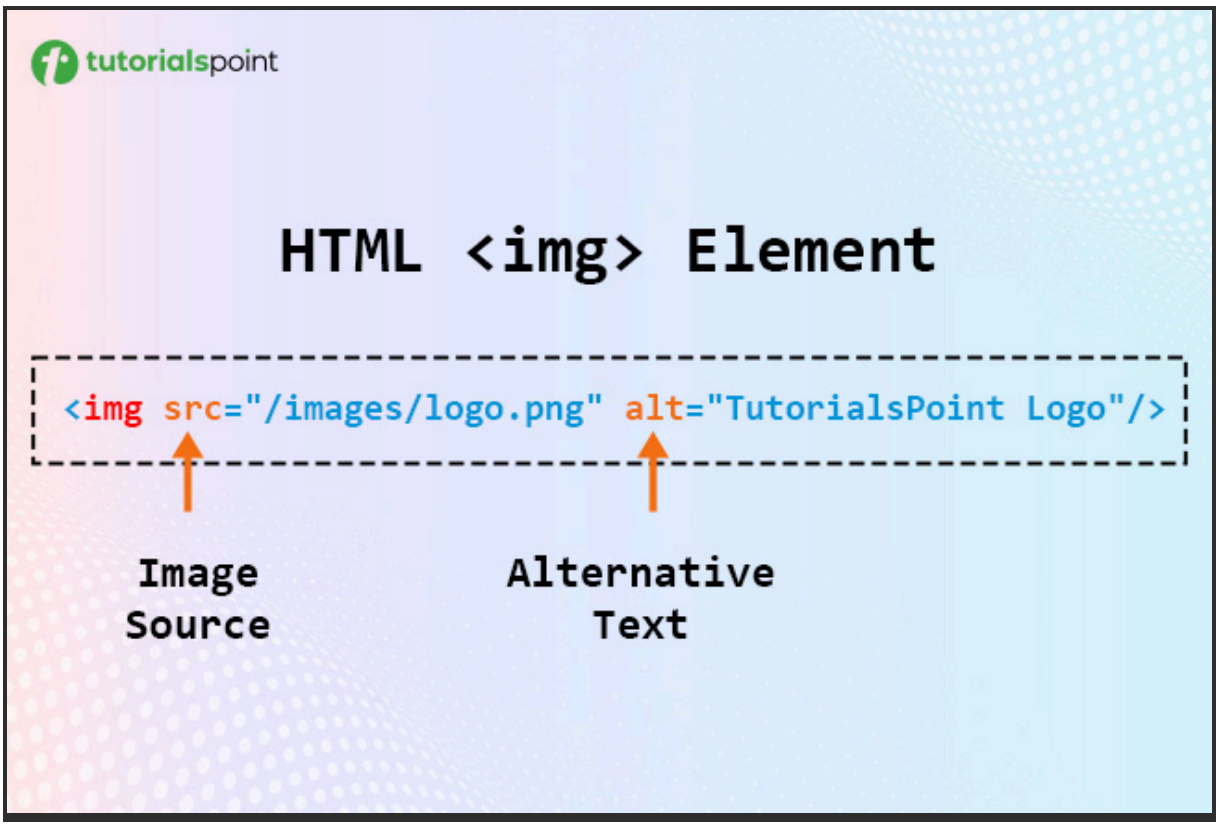
Tag	Description	Category
<b>	This tag is used to make the text **bold**.	Physical Tag
<i>	This tag is used to make the text *italic*.	Physical Tag
<big>	This tag is used to make the text bigger. It is **not supported in HTML5**.	Physical Tag
<small>	This tag is used to make the text smaller.	Physical Tag
<u>	This tag is used to underline text.	Physical Tag
<strike>	This tag is used to strike through text. It is **not supported in HTML5**.	Physical Tag
<tt>	This tag is used to make text appear in teletype (monospace font). It is **not supported in HTML5**.	Physical Tag
****	This tag is used to bold text and give it **semantic importance**.	Logical Tag
****	This tag is used to italicize text and give it *semantic emphasis*.	Logical Tag
<sup>	This tag is used to make superscript text (slightly above the normal line).	Other Tag
<sub>	This tag is used to make subscript text (slightly below the normal line).	Other Tag
<ins>	This tag is used to indicate that content has been added (typically underlined).	Other Tag
<del>	This tag is used to indicate that content has been deleted (typically struck through).	Other Tag

<mark>	This tag is used to highlight text with a yellow background.	Other Tag

HTML - Quotations

HTML quotations allow you to include and format quoted text within your web content. HTML provides tags such as <blockquote>, <q>, <cite>, <address>, <bdo>, and <abbr> to structure and style quotes.

Tag	Description
<q>	Defines a short inline quotation.
<blockquote>	Defines a block-level indented quotation.
<cite>	Specifies a reference to the title of a creative work, such as books or articles.
<address>	Defines contact information.
<bdo>	Overrides text direction.
<abbr>	Defines an abbreviation or acronym.



HTML - Iframes

HTML iframe is an inline frame that allows you to embed another document within the current HTML document. Whenever you want to display another webpage within the webpage, you can use an iframe.

```
  <iframe src="/html/m.htm"> Sorry your browser does not support inline frames. </iframe>
```

HTML <kbd> Element

The <kbd> tag is used to define the keyboard input. Use this when you want the user to type on their keyboard, for example, shortcut keys Ctrl+C for copy, Esc for exit, etc.

HTML Computer Code Elements

Here is the list of the computer code tags along with their descriptions used for defining user input and computer code:

Tag	Description
<kbd>	Defines keyboard input.
<pre>	Displays preformatted text with preserved spaces and line breaks.
<code>	Defines a piece of computer code.
<var>	Represents a variable in programming or math expressions.
<samp>	Defines sample output from a program or device.

HTML <meta> tag lets us specify metadata, which is additional important information about a document, in a variety of ways. The META elements can be used to include name and content pairs describing properties of the HTML document, such as author, expiry date, a list of keywords, document author, etc.

Setting Cookies

Cookies are data stored in small text files on your computer, and it is exchanged between a web browser and a web server to keep track of various information based on your web application needs.

You can use the **<meta>** tag to store cookies on the client side, and later this information can be used by the Web server to track a site visitor. If you do not include the expiration date and time, the cookie is considered a session cookie and will be deleted when the user exits the browser.

Example

Following is an example of redirecting the current page to another page after 5 seconds. If you want to redirect the page immediately, then do not specify the **content** attribute.

```
<!DOCTYPE html>

<html>

<head>

  <title>Meta Tags Example</title>


  <!-- Meta tag for specifying keywords relevant to the page -->
```

```
  <meta name="keywords" content="HTML, Meta Tags, Metadata" />

  <!-- Meta tag for providing a short description of the page -->
  <meta name="description" content="Learning about Meta Tags." />

  <!-- Meta tag indicating when the page was last revised -->
  <meta name="revised" content="Tutorialspoint, 3/7/2014" />

  <!-- Meta tag for setting a cookie with a specific expiry date -->
  <meta http-equiv="cookie" content="userid=xyz; expires=Wednesday, 08-Aug-15 23:59:59 GMT;" />
</head>
<body>
  <p>Hello HTML5!</p> <!-- A simple paragraph displaying text -->
</body>
</html>
```

HTML - Classes

The class is an important keyword in HTML. It is an attribute that can be applied to one or more elements and is used to style and categorize elements based on common characteristics or purpose. Classes allows multiple elements to share the same styling rules. By assigning the same class to multiple elements, you can apply CSS styles or JavaScript functionality to all of them simultaneously. This promotes consistency in design and layout, making it easier to manage and update a website.

```
<element class="highlight">...</element>
```

```
/* CSS using class Attribute Selector */
.highlight {
background-color: yellow;
```

color: black;

font-weight: bold;

}

```
document.getElementsByClassName('highlight')
```

HTML – id

The id is an important keyword in HTML. HTML "id" is an attribute used to uniquely identify an element within a web page. It serves as a label for that element and enables JavaScript and CSS to target it specifically.

<!DOCTYPE html>

<html>

<head>

<script>

function showContent() {

var element = document.getElementById('content');

if (element.style.display === 'none') {

element.style.display = 'block';

} else {

element.style.display = 'none';

}

}

</script>

<style>

.interactive-button {

background-color: #007bff;

```
color: #fff;

padding: 10px 20px;

border: none;

cursor: pointer;

}

</style>

</head>

<body>

<button class="interactive-button"

onclick="showContent()">Click Me</button>

<p class="content" style="display: none;">

This content can be toggled by clicking the button.

</p>

</body>

</html>
```

Difference between id and class in HTML

In HTML, the id attribute uniquely identifies a single element on a page, making it useful for targeting with CSS and JavaScript, and it must be unique within the document. The class attribute, on the other hand, can be applied to multiple elements, allowing for the grouping of elements that share common styles or behaviors.

Within the document every id must be unique.

HTML – Tables

HTML <table> Tag: This tag is used to create the table that wrap the rows and columns within it.

HTML <tr> Tag: Stands for "table row" and is used to create a row within the table.

HTML <td> Tag: Represents "table data" and is used to create standard cells within a row.

HTML <th> Tag: Represents "table header" and is used to create header cells within a row.

```
<!DOCTYPE html>
<html>
<body>
<table border="1">
<tr>
<th>Product</th>
<th>Category</th>
<th>Price</th>
</tr>
<tr>
<td>Laptop</td>
<td>Electronics</td>
<td>$800</td>
</tr>
<tr>
<td>Bookshelf</td>
<td>Furniture</td>
<td>$150</td>
</tr>
<tr>
<td>Coffee Maker</td>
<td>Appliances</td>
```

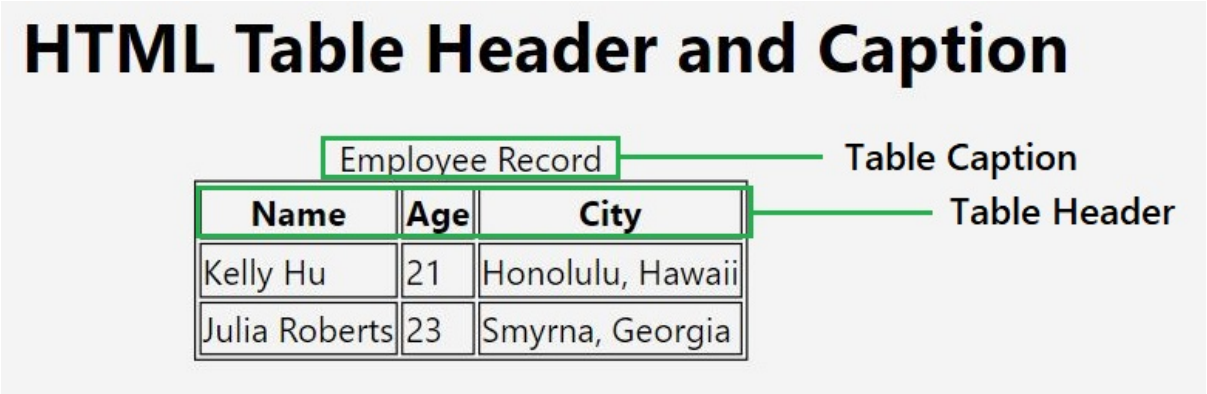
<td>$50</td>

</tr>

</table>

</body>

</html>




```
<caption>Description of table</caption>
```

HTML - Lists

HTML lists are groups or collections of items. These items can be both organized and unorganized depending on the requirement. They help in organizing, structuring, and presenting information to make it more user-friendly, readable, and accessible.

An Organized List	Numerically Organized List
▪ Rice ▪ Wheat Flour ▪ Vegetables ▪ Pulses ▪ Milk	1. Aqua Man 2. Bat Man 3. Captain America 4. Hulk 5. Iron Man

Unordered lists

Unordered lists display lists of items that are not in a specific order. The unordered lists are marked with bullet points. To create an unordered list, the <ul> tag is used along with the <li> tag. Here, the <li> tag specifies the list items.

HTML

CSS

JavaScript

Java

JavaFX

Example of HTML List

- HTML
- CSS
- JavaScript
- Java
- JavaFX

Ordered Lists

Ordered lists are lists of items that are in a specific order. The ordered lists are marked with numbers by default; you can change the numbers into alphabets, roman numbers, etc. by using the type attribute or the CSS list-style-type property.

HTML

CSS

JavaScript

Java

JavaFX

Example of HTML List

1. HTML
2. CSS
3. JavaScript
4. Java
5. JavaFX

HTML - Text Links

HTML Links (Hyperlinks) are words or buttons having a link to another page that take the user to that linked page when clicked.

A hyperlink is a specific type of link that allows users to navigate from one web page or resource to another by clicking on it. You can create hyperlinks using text or images available on a webpage. A hyperlink is created using the HTML Anchor Tag ().

```
<a href="URL" target="_target_type">Link Text</a>
```

S.No.	Option & Description
1	**_blank** Opens the linked document in a new window or tab.
2	**_self** Opens the linked document in the same frame.
3	**_parent** Opens the linked document in the parent frame.
4	**_top** Opens the linked document in the full body of the window.
5	**targetframe** Opens the linked document in a named targetframe.

Linking to a Page Section

Linking to a section on the same page allows users to navigate directly to that section. You can create a link in the same to a specific section by using the href attribute with a #id value, where the #id targets an element on the page with a corresponding id attribute.

Downloadable Links

HTML allows you to create downloadable links where you can create links to make your PDF, DOC, or ZIP files downloadable. To create any link downloadable, you can use the download attribute with the <a> tag and specify the downloadable file path in the href attribute.

```
<a href="/html/src/sample.txt" download>Download File</a>
```

HTML - Image Links

Images can also be used as links in HTML, which means by clicking the images we can navigate to other web pages or resources. HTML image links are very useful in creating websites like photo galleries, portfolios, online stores, and so on. In this article, we will learn how to use images to create hyperlinks. It is similar to the HTML - Text Link.

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

      <img src="/html/images/logo.png"

           alt="Tutorials Point"

           border="0" />

   </a>
```

Image Link with Tooltip

You can also define a tooltip for an image link; when someone moves the mouse over the linked image, it will display a tooltip. To set the tooltip, you can set the title attribute of the <a> tag.

```
<a href="https://www.tutorialspoint.com" title="Go to
TutorialsPoint">

      <img src="/html/images/logo.png"

           alt="Tutorials Point"

           border="0" />
```

```
    </a>
```

Mouse-Sensitive Images

The HTML and XHTML standards provide a feature that lets us embed several different links inside a single image. We can create different links on the single image based on different coordinates available on the image.

HTML - Email Links

HTML email links allow users to click on a link and automatically open their default email client with a new message composed to the specified email address.

```
<a href= "mailto:name@email.com">name@email.com</a>
```

HTML Forms

HTML forms are collections of interactive controls and various input types, such as text, numbers, email, password, radio buttons, checkboxes, buttons, etc., that collect user information. HTML forms are created by using the HTML <form> tag. All user input-related tags are placed inside the <form> tag.

Attribute	Description
action	It is used to specify a URL that processes the form submission.
method	It is used to define which HTTP method to use when submitting the form.
target	It is used to specify where to open the linked document.
autocomplete	It allows you to set whether the autocomplete for the form should be on or off.
enctype	It is used to specify how the form input data should be encoded before sending it to the server.

novalidate	It defines that while submitting the form, the form data should not be validated in an HTML document.

GET

It is the default method for form submission, which means if we don't specify the method name explicitly, the form will use the GET method to send data.

POST

It is used to send form data inside HTTP request body. It is safer than GET method.

HTML - Form Controls

<!-- Text Input Control: A single-line input field for entering text. -->

<input type="text" name="username" placeholder="Enter your username">

<!-- Checkboxes Control: A control for selecting multiple options (can be checked or unchecked). -->

<input type="checkbox" name="subscribe" value="newsletter"> Subscribe to newsletter

<!-- Radio Buttons Control: A control for selecting a single option from a list of choices. -->

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

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

<!-- Select Box Control: A dropdown list to select one or more options. -->

<select name="country">

  <option value="usa">USA</option>

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

</select>

<!-- File Select Box: A control for selecting a file from the user's device. -->

<input type="file" name="upload">

<!-- Button Control: A clickable button to perform an action. -->

<button type="button">Click Me</button>

<!-- Hidden Form Control: A hidden input field used to store data that is not visible to the user. -->

<input type="hidden" name="userId" value="12345">

<!-- Datetime Controls: A control to input a date and time (combined). -->

<input type="datetime-local" name="meeting-time">

<!-- Date Control: A control for selecting a date. -->

<input type="date" name="birthdate">

<!-- Month Control: A control for selecting a month and year. -->

<input type="month" name="month">

<!-- Week Control: A control for selecting a week (with the year). -->

<input type="week" name="week">

<!-- Time Control: A control for selecting a time (without date). -->

<input type="time" name="alarm-time">

<!-- Number Control: A control for entering a numeric value. -->

<input type="number" name="age" min="18" max="100">

<!-- Range Control: A control for selecting a value within a specified range. -->

<input type="range" name="volume" min="0" max="100">

<!-- Email Control: A control for entering an email address (validates email format). -->

<input type="email" name="email" placeholder="Enter your email">

<!-- URL Control: A control for entering a URL (validates URL format). -->

<input type="url" name="website" placeholder="Enter your website URL">

HTML - Input Attributes

The HTML input attributes define the characteristics and behavior of the <input> element. These input attributes are used with the different types of input fields, such as text, email, password, date, number, and so forth. Note that the input element is used to create interactive controls for the web-based forms so that it can accept data from the user.

The <input> element requires only an opening tag, and it will work only if we add it in between the <form> tags. In this tutorial, we are going to explore the attributes that are used with the <input> element.

<!-- type: Specifies the type of input control. -->

<input type="text" name="username" placeholder="Enter your username">

<!-- name: Specifies the name of the input element, which is used to reference form data after submission. -->

<input type="text" name="email" placeholder="Enter your email">

<!-- value: Specifies the default value of an input element. -->

<input type="text" name="username" value="JohnDoe">

<!-- size: Specifies the visible width of the input element (only for text-based inputs). -->

<input type="text" name="username" size="30" placeholder="Enter username">

<!-- maxlength: Specifies the maximum number of characters allowed in an input field. -->

<input type="text" name="username" maxlength="15" placeholder="Max 15 characters">

<!-- readonly: Specifies that an input field is read-only, preventing the user from changing the value. -->

<input type="text" name="username" value="ReadOnly" readonly>

<!-- disabled: Disables the input field, preventing user interaction. -->

<input type="text" name="username" value="Disabled" disabled>

<!-- min and max: Specifies the minimum and maximum values allowed for a number or date input. -->

<input type="number" name="age" min="18" max="100">

<!-- accept and multiple: Used in file input to specify acceptable file types and allow multiple file selection. -->

<input type="file" name="file" accept="image/*" multiple>

<!-- placeholder: Provides a short hint within the input field before the user enters data. -->

<input type="text" name="search" placeholder="Search here...">

<!-- required: Specifies that an input field must be filled out before submitting the form. -->

<input type="text" name="username" required placeholder="Enter your username">

<!-- autofocus: Automatically focuses the input field when the page is loaded. -->

<input type="text" name="username" autofocus placeholder="Autofocus on this field">

<!-- list: Specifies a list of predefined options for an input field (works with <datalist>). -->

<input type="text" name="city" list="cities">

<datalist id="cities">

  <option value="New York">

  <option value="Los Angeles">

  <option value="Chicago">

</datalist>

HTML Video

The <video> element is used to enable video playback support within a web page. It works very similarly to the <img> element, as it also requires adding the path or URL of the video within the src attribute. The HTML supports only MP4, WebM, and Ogg video formats. The <video> element also supports audio; however, the <audio> element is more suitable for that purpose.

<!-- src: Specifies the path to the video file. -->

<video src="movie.mp4" controls></video>

<!-- controls: Adds video controls (play, pause, volume, etc.) to the video player. -->

<video src="movie.mp4" controls></video>

<!-- autoplay: Specifies that the video should automatically start playing as soon as it is loaded. -->

<video src="movie.mp4" autoplay controls></video>

<!-- loop: Specifies that the video should start over again once it finishes. -->

<video src="movie.mp4" loop controls></video>

<!-- muted: Specifies that the video should be played without sound. -->

<video src="movie.mp4" muted controls></video>

```
<!-- poster: Specifies an image to be shown as a placeholder before the video starts playing. -->

<video src="movie.mp4" poster="image.jpg" controls></video>


<!-- width and height: Specifies the width and height of the video player in pixels. -->

<video src="movie.mp4" width="600" height="400" controls></video>


<!-- preload: Specifies if and how the video should be loaded when the page is loaded. -->

<!-- "auto" (default) starts downloading the video as soon as the page is loaded. -->

<!-- "metadata" downloads only metadata (like duration), and "none" doesn't load anything until the user hits play. -->

<video src="movie.mp4" preload="auto" controls></video>


<!-- type: Specifies the MIME type of the video file. -->

<video controls>

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

  <source src="movie.ogg" type="video/ogg">

  Your browser does not support the video tag.

</video>


<!-- poster (again): Specifies a placeholder image that is shown before the video starts. -->

<video src="movie.mp4" poster="image.jpg" controls></video>


<!-- autoplay and loop together: The video will start playing as soon as it's loaded, and then loop indefinitely. -->

<video src="movie.mp4" autoplay loop controls></video>
```

<!-- crossorigin: Allows cross-origin resource sharing (useful when videos are hosted on other domains). -->

<video src="movie.mp4" crossorigin="anonymous" controls></video>

<!-- height and width (alternative): Another way to set the size of the video player. -->

<video src="movie.mp4" height="400" width="600" controls></video>

HTML Audio

The <audio> element is used to enable the support of audio files within a web page. We can include multiple sources of audio; however, the browser will choose the most appropriate file automatically. Most of the attributes of <video> element is also compatible with the <audio> element. The most frequently used attributes of the HTML audio element are controls, autoplay, loop, muted, and src.

<!-- src: Specifies the path to the audio file. -->

<audio src="audio.mp3" controls></audio>

<!-- controls: Adds built-in audio controls (play, pause, volume, etc.) to the audio player. -->

<audio src="audio.mp3" controls></audio>

<!-- autoplay: Specifies that the audio should start automatically once it is loaded. -->

<audio src="audio.mp3" autoplay controls></audio>

<!-- loop: Specifies that the audio should start over once it finishes. -->

<audio src="audio.mp3" loop controls></audio>

<!-- muted: Specifies that the audio should be played without sound. -->

<audio src="audio.mp3" muted controls></audio>

<!-- preload: Specifies if and how the audio should be loaded when the page is loaded. -->

```
<!-- "auto" (default) starts downloading the audio as soon as the page is loaded. -->

<!-- "metadata" downloads only metadata (like duration), and "none" doesn't load until user plays. -->

<audio src="audio.mp3" preload="auto" controls></audio>


<!-- type: Specifies the MIME type of the audio file. -->

<audio controls>

  <source src="audio.mp3" type="audio/mp3">

  <source src="audio.ogg" type="audio/ogg">

  Your browser does not support the audio tag.

</audio>


<!-- autoplay and loop together: The audio will start playing automatically and loop indefinitely. -->

<audio src="audio.mp3" autoplay loop controls></audio>


<!-- crossorigin: Allows cross-origin resource sharing (useful when audio is hosted on other domains). -->

<audio src="audio.mp3" crossorigin="anonymous" controls></audio>


<!-- volume: Specifies the initial volume level (from 0.0 to 1.0, where 1.0 is 100%). -->

<audio src="audio.mp3" controls volume="0.5"></audio>
```

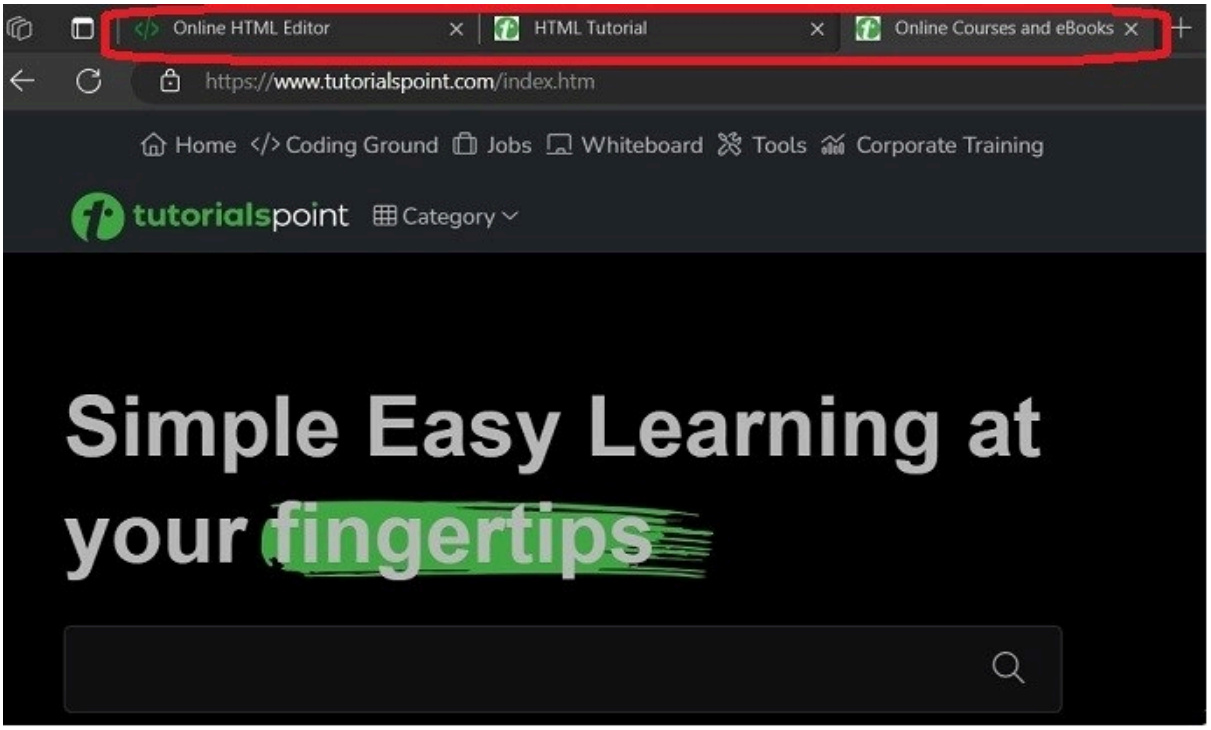
Embedding multimedia in an HTML document refers to the inclusion of content such as images, audio, video, animations, or other interactive media directly within a web page. These multimedia elements can enhance the user experience by making the content more engaging and interactive.

HTML provides several elements to embed different types of multimedia. Here's a breakdown of the most commonly used HTML elements for embedding multimedia content:

```
<head>
  <title>Page Title</title>
  <meta charset="UTF-8">
        <meta    name="viewport"    content="width=device-width,
initial-scale=1.0">
   <meta  name="description"  content="Meta  Description  of  the
page">
  <meta name="keywords" content="Keyword 1, Keyword 2, ...">
  <link rel="stylesheet" href="styles.css">
  <script src="script.js" defer></script>
  <link rel="icon" href="favicon.ico">
</head>
```

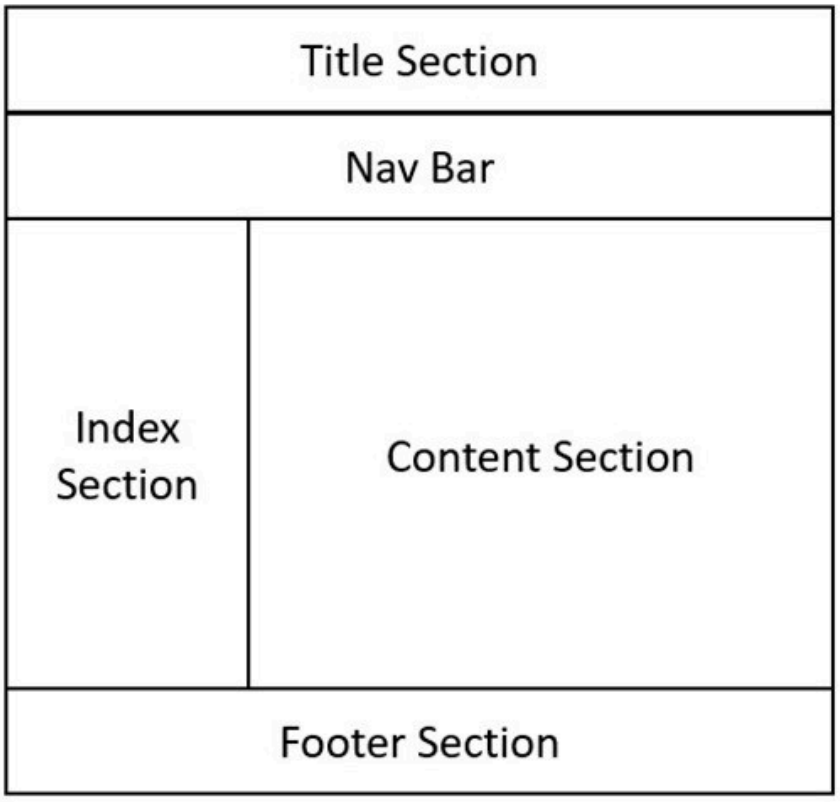
What is a HTML Favicon?

A favicon is a small image that represents your website and helps users identify it among multiple tabs, bookmarks, and search results. It can be in various formats, such as ICO, PNG, GIF, JPEG, or SVG, but ICO is the most widely supported format. If you have ever visited a website and noticed a small icon next to the page title in your browser's tab, you have seen a favicon.



```
<link rel = "icon" type = "image/png" href = "images/faviconTP.png">
```

HTML - Layout Elements



HTML `<header>` Element: Represents the header section of a document or a section, typically containing navigation links or introductory content.

HTML `<nav>` Element: Defines a navigation section with links to help users navigate through the site.

HTML `<section>` Element: Represents a thematic grouping of content, usually with a heading, to divide a page into sections.

HTML `<footer>` Element: Represents the footer section of a document or section, typically containing copyright information or contact links.

HTML `<article>` Element: Represents a self-contained piece of content that could be independently distributed, like a blog post or news article.

HTML `<aside>` Element: Defines content that is tangentially related to the surrounding content, such as sidebars or advertisements.

HTML `<div>` Element: A generic container element used to group content together for styling or layout purposes, with no inherent meaning.

Use `<div>` for generic grouping of content or layout purposes.

Use `<section>` when you want to group content that forms a distinct section of the document and carries a meaningful role.

CSS

https://www.tutorialspoint.com/css/index.htm

CSS is acronym of Cascading Style Sheets. It helps to define the presentation of HTML elements as a separate file known as CSS file having .css extension.

What is CSS?

CSS stands for Cascading Style Sheets, used to describe the presentation and design of a web pages.

Using CSS, you can control the color of the text, the style of fonts, the spacing between paragraphs, how columns are sized and laid out, what background images or colors are used.

CSS can control the layout of multiple web pages all at once.

CSS Syntax

Syntax of CSS consist of selectors and declaration used to apply styles to HTML elements.

```
selector {
    property: value;
}
```

/* Universal Selectors

* is a special selector that matches all elements in an HTML document. These are generally used to add a same length margin and padding to all the elements in the document. */

```
* {
  margin: 0;
  padding: 0;
```

```
}

/* Element Selectors
   * selects elements by their tag name. It targets all instances of the given element in the HTML document. */
h1 {
  font-size: 2rem;
}

/* Class Selectors
   * selects elements by their class name. It targets all elements with the specified class attribute. */
.button {
  background-color: blue;
  color: white;
}

/* Id Selectors
   * selects an element with a specific ID. IDs should be unique in a document and this selector targets that specific element. */
#header {
  background-color: gray;
}

/* Attribute Selectors
   * selects elements based on an attribute or its value. It allows styling elements that have a specific attribute. */
input[type="text"] {
```

```
  border: 1px solid #ccc;
}

/* Group Selectors
   * allows grouping multiple selectors together to apply the same styles to various elements at once. */
h1, h2, p {
  margin-bottom: 15px;
}

/* Pseudo-element Selectors
   * targets specific parts of an element, like the first letter or line, without needing to modify the HTML. */
p::first-letter {
  font-size: 2rem;
  font-weight: bold;
}

/* Pseudo-class Selectors
   * selects elements in a specific state, like when a link is hovered over or when an element is focused. */
a:hover {
  color: red;
}

/* Descendant Selectors
    * selects elements that are nested inside other elements (hierarchical relationship), regardless of how deeply nested. */
```

```
div p {
  color: green;
}

/* Child Selectors
   * selects direct child elements of a specific parent element. It only targets elements that are immediate children. */
ul > li {
  list-style-type: square;
}

/* Adjacent Sibling Selectors
  * selects an element that is immediately preceded by a specific sibling element. */
h1 + p {
  font-style: italic;
}

/* General Sibling Selectors
  * selects all sibling elements that follow a specific element, not necessarily immediately. */
h1 ~ p {
  color: blue;
}
```

Internal CSS: These are added inside HTML files itself.

External CSS: A separate file is made for CSS and imported to HTML file.

Inline CSS: Styles for each tag can be exclusively provided using "style" attribute.

Internal CSS

```
<!DOCTYPE html>
<html>
<head>
<style>
body {
background-color: linen;
}
h1 {
color: maroon;
margin-left: 40px;
}
</style>
</head>
<body>
<h1>This is a heading</h1>
<p>This is a paragraph.</p>
</body>
</html>
```

Inline CSS

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

<h1 style = "color:#36C;">

This is inline CSS

</h1>

</body>

</html>

External CSS

<head>

<link type = "text/css" href = "path/to/style.css" media = "..." />

</head>

Any inline style sheet takes highest priority. So, it will override any rule defined in <style>...</style> tags or rules defined in any external style sheet file.

Any rule defined in <style>...</style> tags will override rules defined in any external style sheet file.

Any rule defined in external style sheet file takes lowest priority, and rules defined in this file will be applied only when above two rules are not applicable.

Styles preference order is

External !important >> Inline >> Internal >> external

<h1 style = "color:#36C;"> >> <head><style> </style> </head> >> <link type = "text/css" href = "path/to/style.css" media = "..." />

CSS - Units

Length Units

Length units can be categorized into two types:

Absolute units: Fixed unit lengths that does not depend on screen width.

```
<style>
.unit-mm {
font-size: 5mm;
}
.unit-cm {
font-size: 1cm;
}
.unit-inch {
font-size: 0.5in;
}
.unit-quarter {
font-size: 40Q;
}
</style>
```

Relative units: Responsive unit lengths that changes according to screen width.

```
<style>
.unit-em {
font-size: 2em;
}
.unit-rem {
font-size: 1.5rem;
}
.unit-vw {
```

```
font-size: 5vw;

}

.unit-vh {

font-size: 5vh;

}

.unit-percent {

font-size: 150%;

}

</style>
```

CSS - Colors

CSS Colors can be specified using predefined name of colors, RGB, RGBA, HSL, HSLA and Hexadecimal values. CSS allow us to change background color, text color, border color and caret color of any elements in HTML documents.

Colors are very important aspects of web design, as they not only enhance the visual appeal but also influence user behavior.

```
background-color: grey;

background-color: #FF0000; /* Red Color */
```

Short Hexadecimal code: Shorter version of hexadecimal format where each of the RGB components is represented by a single digit, and the value is duplicated.

```
background-color: #F00; /* Red Color */

background-color: rgb(0, 0, 255); /* Blue Color */

/* Half Intense Blue Color */

background-color: rgba(0, 0, 255, 0.5);

/* Half Intense Green Color */

background-color: hsla(120, 100%, 50%, 0.5);
```

CSS - Backgrounds

CSS background properties are used to set colors, gradients, or images behind HTML contents. There are several CSS properties associated with backgrounds for setting color, size, image, repeat behavior, and position.

The background is a shorthand for the following properties:

background-color: Sets the background color of an element.

background-image: Sets one or more background image(s) on an element.

background-position: Sets the initial position of each image in a background.

background-size: Controls the size of the background image.

background-repeat: Controls the repetition of an image in the background.

background-origin: Sets the origin of the background.

background-attachment: Specifies the position of the background relative to the viewport, either fixed or scrollable.

background-clip: Controls how far a background image extends beyond the element's padding or content box.

CSS - Fonts

Property	Description
font-family	Specifies the font for an element.
font-size	Specifies the size of the font.
font-weight	Specifies the boldness of the font.
font-style	Specifies the style of the font (normal, italic, oblique).

font-variant	Specifies whether or not a text should be displayed in a small-caps font.
line-height	Specifies the line height.

CSS - Styling Text

Property	Description
color	**Sets the color of the text.**
text-align	**Sets the alignment of the text.**
text-align-last	**Sets the alignment of the last line of a block of text.**
vertical-align	**Sets the vertical alignment of the text.**
direction	**Sets the direction of the text.**
text-indent	**Sets the indentation of the first line of the text.**
letter-spacing	**Specifies the space between the letters of a word.**
word-spacing	**Specifies the space between the words in a block of text.**
white-space	**Controls the white space flow inside the text in an element.**
text-decoration	**A shorthand property for setting the text decoration.**

text-transform	**Transforms the text in either uppercase, lowercase or capitalize.**
text-emphasis	**Applied emphasis marks to text.**
text-shadow	**Adds shadow to the text.**
line-break	**Controls how to set the rule for a line break.**
word-break	**Controls how to set the rule for a word break.**
text-combine-upright	**Combines multiple typographic character units into the space of a single typographic character unit.**
text-orientation	**Sets the orientation of the text characters in a line.**
text-underline-offset	**Adds special visual effects to the text.**
text-overflow	**Controls how hidden overflow content is displayed to users.**

CSS - Images

1. **Round Cornered Images** - `border-radius: 15px;`
2. **Circular And Elliptical Images** - `border-radius: 50%;`
3. **Bordered Images** - `border: 5px solid #000;`
4. **Image Filters** - `filter: grayscale(100%);`
5. **Image as a Card** - `box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1);`
6. **Center an Image** - `display: block; margin: 0 auto;`
7. **Text Inside Image** - `position: absolute; color: white;`
8. **Image Fade In Overlay** - `background: rgba(0, 0, 0, 0.5);`

CSS - Links

1. **States of Link** - `a:link, a:visited, a:hover, a:active { }`
2. **Default Styles of Links** - `color: blue; text-decoration: underline;`
3. **CSS Text Link** - `color: #3498db; text-decoration: none;`
4. **Styling States of Link** - `a:hover { color: red; }`
5. **CSS Button Links** - `display: inline-block; padding: 10px 20px; background-color: #008CBA; color: white; border-radius: 5px;`
6. **CSS Image Links** - `display: inline-block; width: 100px; height: 100px; background-size: cover;`

CSS - Styling Tables

CSS Table Border Styling - `border: 1px solid #ccc;`

CSS Table Border Collapse - `border-collapse: collapse;`

CSS Table Border Spacing - `border-spacing: 10px;`

CSS Tables Caption Side - `caption-side: top;`

CSS Tables Empty Cells - `empty-cells: hide;`

CSS Table Layout - `table-layout: fixed;`

CSS Table Contents Alignment - `text-align: center; vertical-align: middle;`

CSS Tables Background Color - `background-color: #f4f4f4;`

CSS Table Text Font Styles - `font-family: Arial, sans-serif; font-size: 14px;`

CSS Table Dividers - `border-right: 1px solid #ddd;`

CSS Striped Table - `tr:nth-child(even) { background-color: #f2f2f2; }`

CSS Responsive Table - `overflow-x: auto; display: block;`

CSS - Borders

Property	Description
border	A shorthand property for setting all the border properties in one declaration
border-color	A shorthand property for setting border color of an element.
border-style	A shorthand property for setting style (solid / dashed) of border of an element
border-width	A shorthand property for setting border width of an element.
border-bottom	A shorthand property for setting bottom border of an element.
border-bottom-color	Sets the color of bottom border of an element.
border-bottom-width	Sets the width of bottom border of an element.
border-bottom-style	Sets the style of bottom border of an element.
border-left	A shorthand property for setting left border of an element.
border-left-color	Sets the color of left border of an element.
border-left-width	Sets the width of left border of an element.
border-left-style	Sets the style of left border of an element.
border-right	A shorthand property for setting right border of an element.

border-right-color	Sets the color of right border of an element.
border-right-width	Sets the width of right border of an element.
border-right-style	Sets the style of right border of an element.
border-top	A shorthand property for setting top border of an element.
border-top-color	Sets the color of top border of an element.
border-top-width	Sets the width of top border of an element.
border-top-style	Sets the style of top border of an element.
border-image	A shorthand property for setting border image.
border-image-outset	Sets the image outset i.e how much the border image area extends beyond the border box.
border-image-repeat	This property determines whether the border image should be repeated, rounded, spaced or stretched.
border-image-source	Sets the source/path of an image to be passed as a border to an element.
border-image-slice	This property shows how to slice up an image in a border.
border-image-width	Sets the width of the image to be set as a border.

CSS - border-block Property

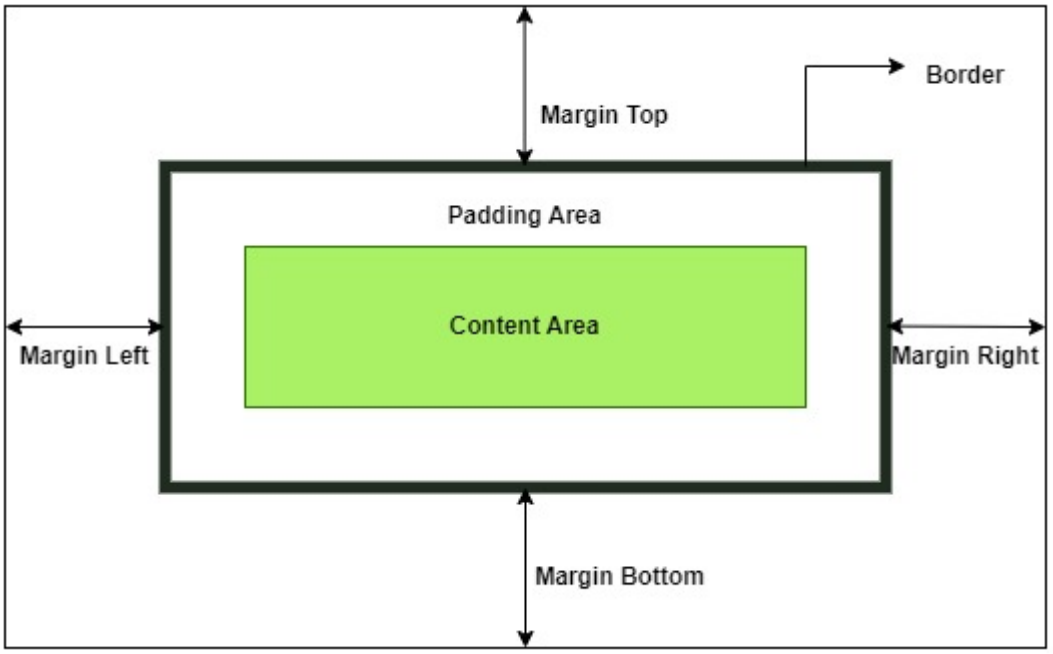
Value	Description
border-block-width	It specifies the width of the border in the block direction. Default is medium.
border-block-style	It specifies the style of the border in the block direction. Default is none.
border-block-color	It specifies the color of the border in the block direction. Default is text's color.
initial	This sets the property to its default value.
inherit	This inherits the property from the parent element.

CSS - border-inline Property

Value	Description
border-inline-width	It specifies the width of element borders in inline direction. Default value is medium.
border-inline-style	It specifies the style of element borders in inline direction. Default value is none.
border-inline-color	It specifies the color of element borders in inline direction. Default value is color of the text.
initial	It sets the property to its default value.
inherit	It inherits the property from the parent element.

CSS - Margins

CSS Margins are used to create space around outer part of an element. In this tutorial we will learn how to add different types of margins to HTML elements and properties associated with it.



margin - A shorthand property that is used for setting all the margin properties in one declaration.

margin-top Sets the top margin of an element.

margin-right Sets the right margin of an element.

margin-bottom Sets the bottom margin of an element.

margin-left Sets the left margin of an element.

margin: "value" — All four sides (top, right, bottom, left) use the same value.

margin: "value value" — First value: top & bottom, second value: left & right.

margin: "value value value" — First value: top, second value: left & right, third value: bottom.

margin: "value value value value" — First value: top, second value: right, third value: bottom, fourth value: left.

`px`, `em`, `rem`, `pt`, `cm`, `mm`, `in`, `%`, `vw`, `vh`, `vmin`, `vmax`, `auto`, `ch`, and `ex` as units for margins.

CSS - Styling Lists

UL

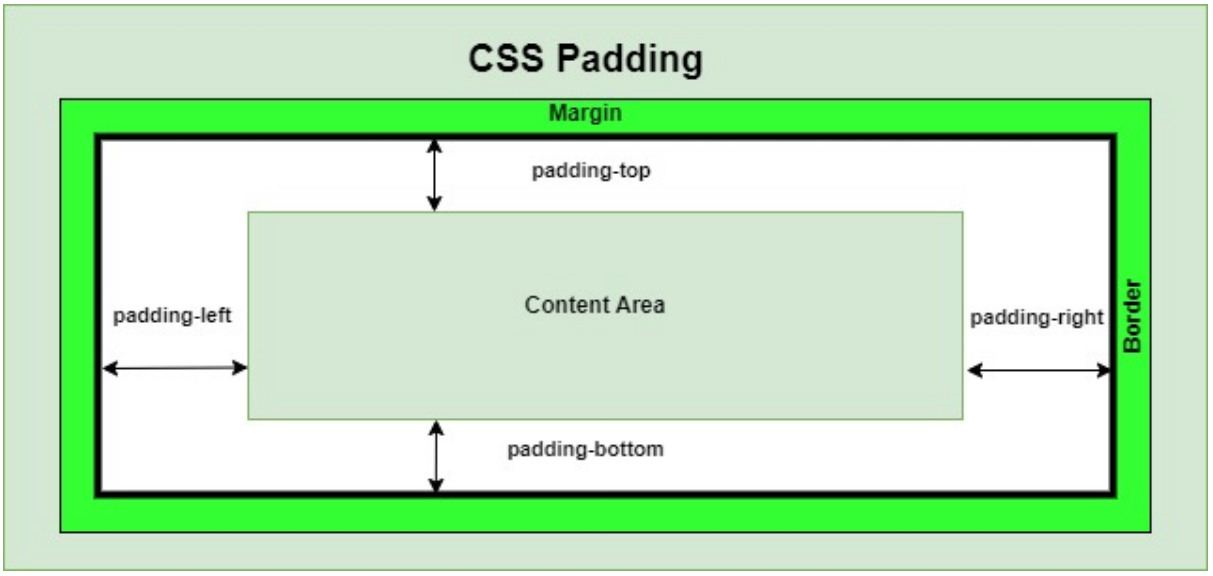
- `list-style-type: disc;`
- `list-style-type: circle;`
- `list-style-type: square;`
- `list-style-type: none;`

OL

- `list-style-type: decimal;`
- `list-style-type: upper-roman;`
- `list-style-type: lower-roman;`
- `list-style-type: upper-alpha;`
- `list-style-type: lower-alpha;`
- `list-style-type: decimal-leading-zero;`
- `list-style-type: hebrew;`
- `list-style-type: armenian;`
- `list-style-type: georgian;`

CSS - Paddings

In CSS, padding is a property that is used to create additional spacing inside the boundary of an element. The default value for padding is zero. A padding value zero indicate that content (mostly text contents) will start from the border of element itself.



padding - A shorthand property that is used for setting all the padding properties in one declaration.

Padding-top

Sets the top padding of an element.

padding-right
Sets the right padding of an element.

padding-bottom
Sets the bottom padding of an element.

padding-left
Sets the left padding of an element.

---

padding: "value" — All four sides (top, right, bottom, left) use the same value.
padding: "value value" — First value: top & bottom, second value: left & right.
padding: "value value value" — First value: top, second value: left & right, third value: bottom.
padding: "value value value value" — First value: top, second value: right, third value: bottom, fourth value: left.

---

px, em, rem, pt, cm, mm, in, %, vw, vh, vmin, vmax, auto, ch, and ex as units for padding.

CSS - cursor Property

CSS cursor property determines the appearance of the mouse cursor when hovering over an element to which this property is applied. Its main purpose is to improve usability by visually representing certain functions.

CSS - float Property - float: none | left | right | initial | inherit;

CSS - Resize -

CSS resize is a property that allows users to adjust the size of an element, either vertically, horizontally, both, or none, based on the specified value.

Resize property adds a handle at the bottom-right corner of an element on a webpage. This handle allows users to click and drag to change the size of an elements, making it larger or smaller according to their preference.

```
resize: none | vertical | horizontal| both;
```

CSS - quotes

CSS quotes property allows browser to render quotation marks for the content.

Quotes can be added to any element. They seek the benefit of pseudo-elements ::before and ::after to insert the quotation marks at the beginning and at the end of a quote. These pseudo-elements are defined by the content property.

This CSS quotes specifies how the browser should render quotation marks that are added using the open-quote and close-quote values of the content property.

CSS - order Property

CSS order property is used to specify the order in which flex items appear within a flex container. The order of the flex items is determined by the values of their order property. The flex items with the lower order value will be displayed first.

CSS - Position Property

static − The element is positioned according to the default or normal flow of the page. So if we set left/right/top/bottom/z-index, then there will be no effect on that element.

relative − The element's orginial position is according to normal flow of the page just like static value. But now left/right/top/bottom/z-index will work. The positional properties push the element from the original position in that direction.

absolute − The element is completely removed from the document flow. It is then positioned with respect to its containing block, and its edges are placed using the side-offset properties. An absolutely positioned element may overlap other elements, or be overlapped by them.

fixed − The element's fixed positioning is just like absolute positioning, except the containing block of a fixed element is always the viewport. Here the element is totally removed from the document's flow and does not have a position relative to any part of the document.

sticky − The element sticks to the top of its nearest positioned ancestor that has a "scrolling mechanism" .

```
position: static | relative | absolute | fixed | sticky;
```

CSS - Hover Effects

CSS hover effects are used to make interactive elements like buttons and links more dynamic and engaging for users browsing the webpage.

The :hover pseudo-class in CSS is used to target an element when the user hovers over it with the mouse cursor. Its purpose is to apply styles or trigger specific behaviors to enhance the user experience or provide additional visual feedback.

CSS - display Property

CSS display property is used to specify how an element should be displayed on the webpage. It controls the layout and visibility of an element. The display property is useful in setting the inner and outer display types of an element.

```
display: value;
```

Value	Description
inline	It displays the element as an inline element on which width and height properites do not have any effect. Default.
block	It displays the element as a block element which starts on a new line and takes up the whole width.
contents	It make an element disappear from the layout while keeping its child elements visible and in their original positions within the layout.
flex	It displays an element as a block-level flex container.
grid	It displays an element as a block-level grid container.
inline-block	It allows an element to flow along with other inline elements along with having block-level characteristics such as width and height.
inline-flex	It displays an element as an inline-level flex container.
inline-grid	It displays an element as an inline-level grid container.

inline-table	It displays the element as an inline-level table.
run-in	It displays an element depending on context as either block or inline.
table	It enables the element to behave like a <table> element.
table-caption	It enables the element to behave like a <caption> element.
table-column-group	It enables the element to behave like a <colgroup> element.
table-header-group	It enables the element to behave like a <thead> element.
table-footer-group	It enables the element to behave like a <tfoot> element.
table-row-group	It enables the element to behave like a <tbody> element
table-cell	It enables the element to behave like a <td> element.
table-column	It enables the element to behave like a <col> element.
table-row	It enables the element to behave like a <tr> element.
none	It removes the element completely.
initial	This sets the property to its default value.
inherit	This inherits the property from the parent element.

CSS - Focus Effects

CSS focus effects are used to make form elements like input fields, buttons, and links more dynamic and engaging for users interacting with the webpage.

The :focus pseudo-class in CSS is used to target an element when it receives focus (by clicking on it or by pressing tab). Its purpose is to apply styles or trigger specific behaviors to enhance the user experience or provide additional visual feedback.

CSS - translate Property

The translate property of CSS allows you to move an element along the X axis (horizontal), Y axis (vertical) and Z axiz (depth).

CSS - height Property

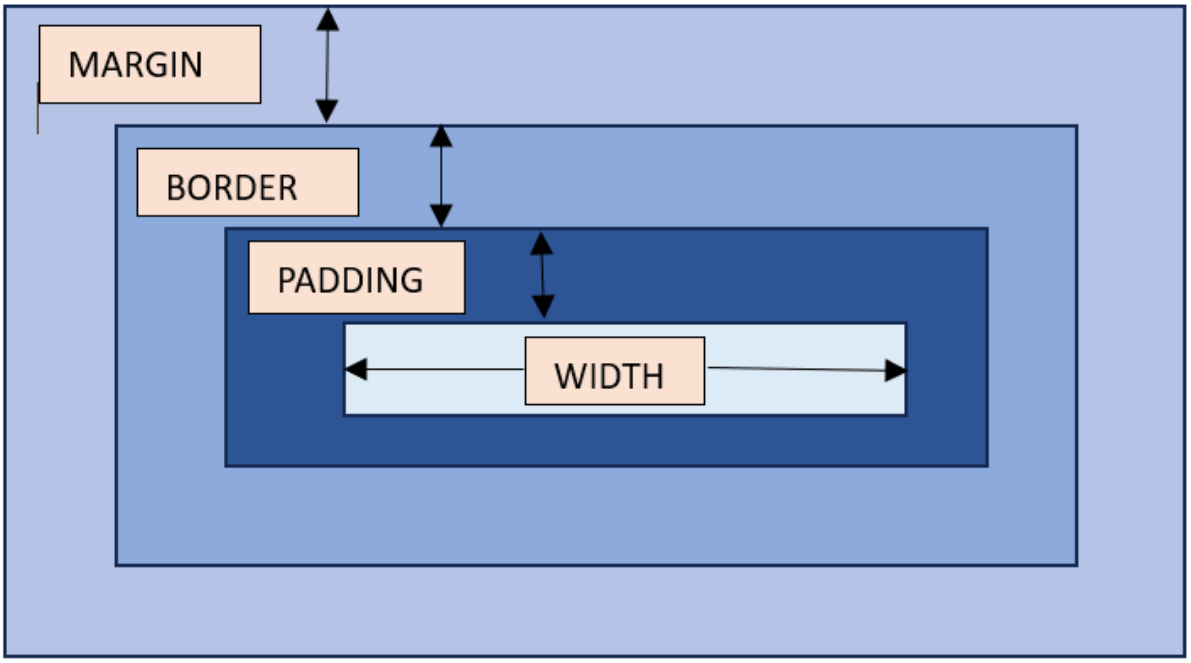
CSS height property specifies the height of an element. It determines how tall an element will be, affecting its layout and positioning within the document. The property can be applied to block-level elements, inline-block elements, and replaced elements like images.

```
height: auto | length | percentage | min-content | max-content | initial | inherit;
```

CSS - Width Property

The width property sets the width of an element's content area. In case, the box-sizing is set to border-box, the property width sets the width of the border area.

The value specified by the width property remains within the values defined by min-width and max-width properties.



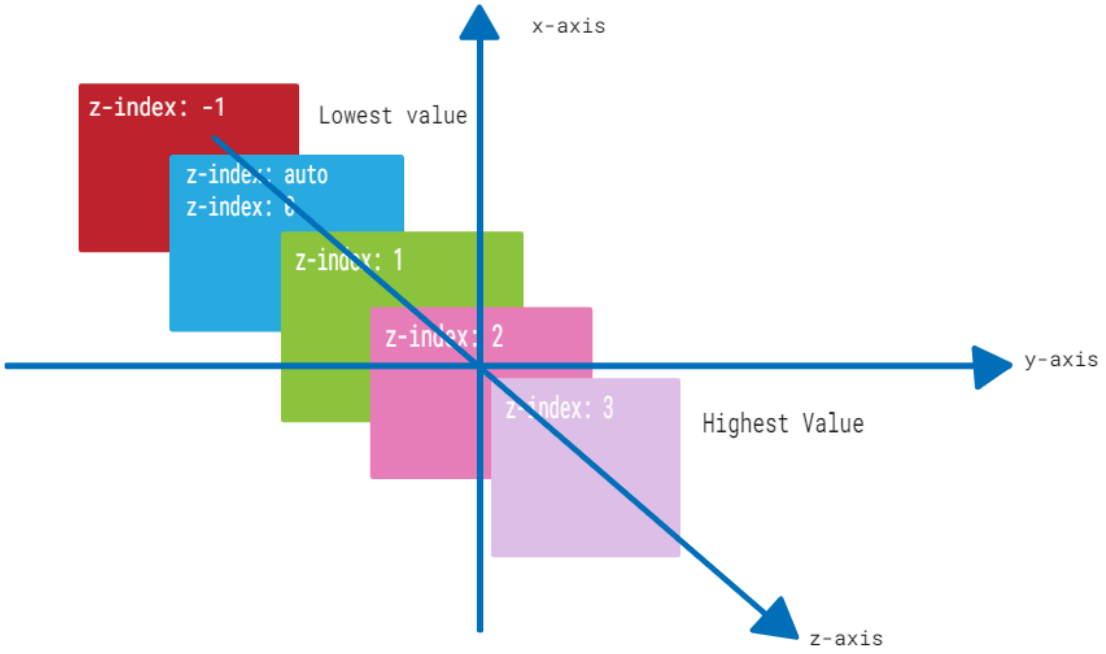
CSS - opacity Property

CSS opacity property controls the transparency of an element. It determines how much of a hidden element's content is visible. The property can be used on various elements, whether they contain text, images, or serve as backgrounds.

```
opacity: number | percentage | initial | inherit;
```

CSS - z-index Property

CSS z-index property is used to control the stacking order of elements in a web page when they overlap in the same stacking context. Elements with a higher z-index value appear in front of elements with lower values.



CSS - Navigation Bar

Navigation bar is a section of a graphical user interface (GUI) that helps users navigate through a website, app, or other software. It is essential for users to quickly and easily navigate to the content they are looking for.

The navigation bar can be horizontal or vertical, that contains links to important pages or features.

CSS - Forms

Forms are required, when you want to collect some data from the site visitor. They have input fields for users to enter information, labels to identify the input fields, and buttons to submit the form or perform an action.

We can style HTML forms by changing the appearance of form elements, such as text fields, checkboxes, radio buttons, select menus, and submit buttons.

Use Containers: Containers can be used to wrap all the form elements like text input, radio buttons, submit buttons in a single container and apply common style to it.

For Text Input: For text input you can add padding, margin and background color according to your color theme. Also you can use focus pseudo-class to style selected state of an input element.

Hover Effect: By using pseudo-class :hover, you can style mouse over the element state of elements like submit button. This gives user a dynamic experience.

Responsive Design: By using Media Queries you can adjust form style for different screen widths

Transition Effect Further more you can use transition effect for smooth interaction with input tags and buttons

CSS - Align

The term alignment, in context of web design and CSS, refers to the positioning and arrangement of elements or content within a layout, typically with respect to specific guidelines or reference points. Alignment is used to create visually pleasing and organized designs by ensuring that elements are positioned relative to each other or to the layout structure in a consistent and harmonious way.

Alignment can be applied to various types of elements, including text, images, buttons, and more, to create a cohesive and polished design. CSS provides various properties that can be used to align the elements.

Property	Description
align-content	Aligns the content of a flex container along the cross-axis or a grid's block axis.
align-items	Controls the alignment of items of a flex container along the cross-axis.
align-self	Controls the alignment of an individual item within a container.
vertical-align	Determines the vertical alignment of inline, inline-block or a table cell text.
line-height	Sets the distance between lines of text.

text-align	Sets the horizontal alignment of inline, inline-block or a table cell text.
margin	Shorthand for margin values that can modify the alignment.

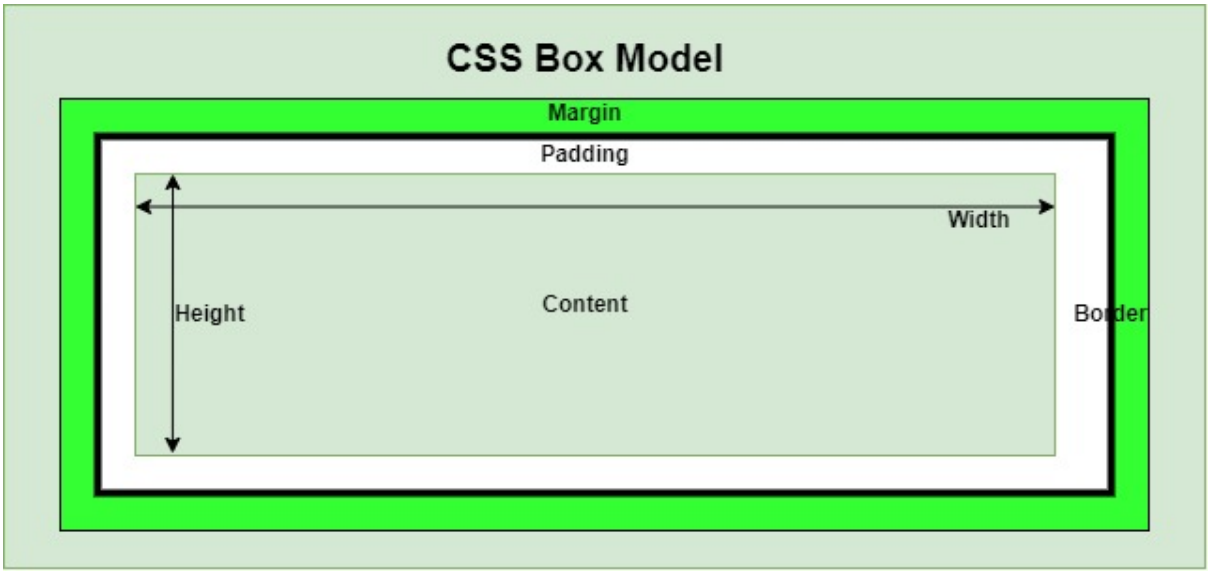
CSS - Image Gallery

CSS Image gallery is used to organize and display images in responsive and visually appealing format. CSS properties can be used to control the layout of images, size, shape, spacing, spanning and lot of other visual effects. In this tutorial we will learn all of those.

CSS image galleries are commonly used on websites to display products, portfolios, or other visual content in a visually appealing way.

CSS - Box Model

The CSS box model is a container that used to structure the elements in a webpage so the element can be displayed visually good. It consists of four essential components content, padding, border, and margin, as shown in the following diagram.



CSS - inset Property

CSS inset property is used to control the position of an element relative to its containing block. It is a shorthand property for defining values to the properties top, right, bottom, and / or left in a single statement. The position property must be declared in order for the property to show its effect.

```
inset: auto | length | percentage | initial | inherit;
```

CSS - justify-items Property

CSS justify-items property is used to align grid-items within their grid area along the inline direction (horizontal). It controls how items are placed within their grid cells, effectively setting their alignment within the container's grid.

```
justify-items: legacy | normal | stretch | start | left | center | end | right | overflow-alignment | baseline alignment | initial | inherit;
```

CSS pagination is a technique of creating page numbers for a website. This help users to easily navigate between large amounts of content. In this chapter, we will learn how to setup and style pagination using CSS.