
WEB AUTHORIZING

What is a website?

- Its a collection or related webpages
- It contains three layers: structure layer (HTML), presentation layer (CSS), behaviour layer (JS).
- Hypertext Markup Language and Cascading stylesheet are two common languages used in developing static webpages.

Description of CSS Precedence

- CSS can be incorporated into a webpage in three main ways (**arranged in order of precedence**):
 - **Inline CSS:** Inline styles are applied directly to individual HTML elements using the `style` attribute. Styles defined inline override any other CSS rules.

```
<p style="color: blue;">This is a paragraph with inline CSS.</p>
```

- **Internal CSS:** Internal styles are defined within the `<style>` element in the `<head>` section of the HTML document. These styles apply to the entire document or specific elements within it.

```
<head>
  <style>
    p {
      color: red;
    }
  </style>
</head>
```

- **External CSS:** External stylesheets are separate CSS files linked to the HTML document using the `<link>` element. These styles apply globally across multiple HTML pages.

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

Explaining the Document Type Definition

The `<!DOCTYPE html>` declaration is an instruction to the web browser about which version of HTML the page is written in. It stands for "Document Type Definition" and is used to ensure that the browser renders the web page correctly according to the specified HTML version.

Specifically, `<!DOCTYPE html>` is used for HTML5 documents, which is the latest version of HTML. When a browser encounters this declaration at the beginning of an HTML document, it switches into

standards mode and renders the document according to the rules specified in the HTML5 standard. Using `<!DOCTYPE html>` is important because it helps ensure consistent rendering of web pages across different browsers and devices

In addition to `<!DOCTYPE html>` for HTML5 documents, there were various other Document Type Definitions (DTDs) used in earlier versions of HTML. These DTDs were used to specify the rules and structure of HTML documents. Example: HTML 4.01 Strict, XHTML 1.1 etc.

These different DTDs served to define the syntax and structure of HTML documents and helped browsers render them correctly. However, with the introduction of HTML5, which has a simplified and more flexible syntax, the need for different DTDs diminished, and `<!DOCTYPE html>` became the standard declaration for modern HTML documents.

The head tag

The heading tag often contains elements that are not displayed in the actual webpage. However, the elements placed within it still plays a vital role in ensuring proper functionality.

- The following are the universally accepted elements to be placed within the head tag:
 - **Title tag:** Displays text in the browser's toolbar, favourites menu, and search results. Here's how the `<title>` tag is typically used:

```
<!DOCTYPE html>
<html>
<head>
  <title>This is the Title of the Web Page</title>
</head>
<body>
  <!-- Page content goes here -->
</body>
</html>
```

The text enclosed within the `<title>` tags, in this case, "This is the Title of the Web Page," will be displayed at the top of the browser window or tab when the page is opened. It's essential to provide a descriptive and concise title that accurately reflects the content of the web page, as it helps users understand the purpose of the page and improves search engine optimization (SEO).

Search engines use the title tag to understand the topic and relevance of a webpage. A well-crafted title tag can improve a page's visibility in search engine results pages (SERPs) and attract more clicks from users.

- **Base tag:** The `<base>` tag specifies the base URL and target for all relative URLs within a document. It's placed in the `<head>` section of an HTML document.

```
<head>
  <base href="https://example.com/" target="_blank">
</head>
```

- **Link tag:** The `<link>` tag is used to link an external resource to the HTML document, such as a stylesheet, favicon, or other resources. It's commonly used to link CSS stylesheets

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

- **Meta tag:** The `<meta>` tag provides metadata about the HTML document, including character encoding, viewport settings, description and keywords for SEO purposes.

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta name="description" content="This is a description of the webpage.">
  <meta name="keywords" content="HTML, CSS, JavaScript, web development">
  <meta name="author" content="Your Name">
</head>
<body>
  <!-- Page content -->
</body>
</html>
```

- **Script tag:** The `<script>` tag is used to embed or reference JavaScript code within an HTML document. It can be placed in the `<head>` or `<body>` section.

```
<head>
  <script src="script.js"></script>
</head>
```

- **Style tag:** The `<style>` tag is used to define internal CSS styles within an HTML document. It's placed within the `<head>` section.

```
<head>
  <style>
    body {
      font-family: Arial, sans-serif;
      background-color: #f0f0f0;
    }
    /* Other CSS styles */
  </style>
</head>
```

Understanding comments

Comments in HTML are used to add explanatory notes or annotations within the code that are not displayed in the browser when the page is rendered. They are helpful for developers to document their code, provide reminders, or temporarily disable specific sections of code without deleting them entirely.

HTML comments are written using the `<!--` opening tag and the `-->` closing tag. Anything between these tags is treated as a comment and is ignored by the browser during rendering.

The body tag

The body tag contains elements whose contents will be rendered on the browser as instructed by the elements themselves. The rest of the tags described in this document must only be placed in the body tag.

Preset Styles in HTML

In HTML, there are no "preset" styles inherent to the language itself. HTML is a markup language used for structuring content, while styles are applied using CSS (Cascading Style Sheets). However, browsers do have default stylesheets, which are applied to HTML elements unless overridden by specific CSS rules.

These default styles vary slightly between different browsers, but they generally aim to ensure that web pages are displayed consistently and are readable even if no CSS styles are explicitly applied.

Some common default styles applied by browsers to HTML elements include:

- **Text styling:** Browsers typically set default font sizes, font families, line heights, and margins for text elements like `<p>`, `<h1>`, `<h2>`, `<h3>`, etc.
- **Lists:** Default styles for `` (unordered lists) and `` (ordered lists) include padding, margins, and list item markers (bullets or numbers). Each list item is added by ``.
- **Links:** Default link styles typically include underlining and changing the color to indicate visited and unvisited links. However, these styles can vary depending on the browser and the state of the link.
- **Tables:** Default table styles include border spacing, padding, and border styles for `<table>`, `<tr>`, `<td>`, and related elements.
- **Form elements:** Basic styling is applied to form elements like `<input>`, `<select>`, `<textarea>`, etc., to ensure they are visually distinguishable and accessible.

Text Enhancements

The following tags can be used to enhance texts in a html document.

- `` tag: This stands for "bold." It is used to make the enclosed text bold.

```
<b>This text will be bold</b>
```

- `<i>` tag: This stands for "italic." It is used to make the enclosed text italicized.

```
<i>This text will be italicized</i>
```

- `<sup>` tag: This stands for "superscript." It is used to render text in a smaller size and slightly above the normal line of text, typically used for exponents or footnotes.

```
x<sup>2</sup> represents x squared.
```

- `<sub>` tag: This stands for "subscript." It is used to render text in a smaller size and slightly below the normal line of text, often used for chemical formulas or mathematical expressions.

H₂O represents water.

HTML Tables

`<table>` - Used to define the outline of a table. It is somewhat graphically responsible for the external grid lines. The `<table>` has an attribute called **border** which can be set to "0"/"" for no visible gridlines and "1" for visible gridlines. The following are tags that can be placed within `<table>` (**arranged in the order expected by HTML parsers**).

- `<caption>` - The first item within the table. Items placed within this tag is centred and appears above the table it self.
- `<thead>` - Stands for "Table header". Defines the first section of a table. Must be the second tag within the table
- `<tfoot>` - Stands for "Table footer". Defines the last section of a table. Must be the third tag within the table
- `<tbody>` - Stands for "Table body". Defines the second section of a table. Must be the last tag within the table.
 - `<tr>` - Defines a row. Can go inside `<thead>`, `<tfoot>`, `<tbody>` or just within the table itself.
 - `<td>` - Defines a **normal** cell. Goes inside `<tr>`. There can be any tag within `<td>`.
 - `<th>` - Defines a **header** cell. Goes inside `<tr>` like `<td>`. Header cells appear emboldened and centred. There can be any tag within `<th>`.

This is a caption	
This is a heading cell in the header section.	
This is a row heading using heading cell.	This is a normal cell.
This is a row heading using heading cell.	This is a normal cell

The General form of HTML Attributes and CSS Style Rules

In HTML, attributes provide additional information about an element and are typically defined within the opening tag of an element. The general form of an **HTML attribute** is:

The general form of HTML attributes		
attribute	=	"value"

In CSS, **style rules** consist of one or more selectors followed by a block of declarations enclosed in curly braces. The general form of a style rule is:

The general form of style rules							
selector	CSS Combinators (' ', + >)	{	property	:	"value"	; (Seperator)	Other Declarations }

The Width and Height of a Table

The width of a column can be adjusted by changing exactly the width one cell in the required column.

Example:

```
<tr>
  <th colspan="2" style="width: 1000px;" <!-- Take up two columns -->
    This is a heading cell in the header section.
  </th>
</tr>
```

This is a caption	
This is a heading cell in the header section.	
This is a row heading using heading cell.	This is a normal cell.
This is a row heading using heading cell.	This is a normal cell

The height of the rows needs to be changed manually for each row. Changing the height of a single row only affects it, and not any other row in the table.

Example:

```
<tfoot>
  <tr>
    <th style="height: 100px;"
      This is a row heading using heading cell.
    </th>
    <td>
      This is a normal cell
    </td>
  </tr>
</tfoot>
```

This is a caption	
This is a heading cell in the header section.	
This is a row heading using heading cell.	This is a normal cell.
This is a row heading using heading cell.	This is a normal cell

Note: The column width **has to be** modified in the `<td>` or `<th>` elements; changing the width of the `<tr>` (row) **will not** affect the width of the column -- in fact, it has no effect in this context. On the other hand, the height of the row **can be** changed by either adjusting the height of a cell (`<td>` or `<th>`) in the row or by modifying the height of the row itself (`<tr>`).

Centre Alignment using Margins

Setting the left and right margins of an element to "auto" in HTML/CSS is a common technique used to horizontally center that element within its containing parent element. When applied to a block-level element like a table, setting both margins to "auto" effectively distributes the remaining space equally on both sides, thus centering the table horizontally within its container.

Here's how it works:

- **Setting both margins to "auto":** By setting both the left and right margins to "auto", you're telling the browser to calculate equal margins on both sides of the table, effectively centering it within its containing element.
- **Block-level element behavior:** Tables are block-level elements by default, which means they occupy the full width of their containing parent element unless specified otherwise.
- **Automatic margin calculation:** When you set a margin to "auto", the browser automatically calculates the margin value to distribute the available space equally. In the case of setting both left and right margins to "auto", the browser calculates and applies equal margins on both sides, thus centering the table horizontally.

This technique is not specific to tables; it can be applied to other block-level elements like divs, paragraphs, and headings to center them horizontally within their containing elements as well.

Collapsing Borders in tables

In HTML tables, the `<table>` contain borders called **table borders**, and the cell contains its own border called **cell border**. However, conventional tables have a single **external border** and **internal gridlines**. To give an HTML table the conventional table look, you should use the `border-collapse` property in CSS.

Adding this `<table border="1" style="border-collapse: collapse;">` changes the table from the left to the right

This is a caption

This is a heading cell in the header section.	
This is a row heading using heading cell.	This is a normal cell.
This is a row heading using heading cell.	This is a normal cell

This is a caption

This is a heading cell in the header section.	
This is a row heading using heading cell.	This is a normal cell.
This is a row heading using heading cell.	This is a normal cell

Styling Cell and Table borders

Styling to cell borders must be added to the `<td>` or `<th>` tags. And styling to the table borders must be added to `<table>` tag. Adding styling to the row (`<tr>`), aiming to style the cells within the rows, will be futile. It may be important to note that, adding styling to a `<td>` tag only modifies the border around that tag. To style the borders for all the cells within a table, all the `<td>` and `<th>` tags within the table must be individually styles, either by redundantly styling the individual cells using **inline CSS**, or by using an **external stylesheet**.

Combinators in CSS

In CSS, you can use the **descendent combinator**, represented by the ' ' (space) character, to select the descendant elements of a specifies ancestor element. **Note** that this is different from the **grouping/multiple selector** which groups elements that match selector to apply the same styles to them.

```
<style>

  table td /* Select all the <td> tags that are present within the <table> tags */
  {
    border-color: red;
  }

  table, td /* Select all the <td> and <table> tags */ {
    border-style: dotted;
  }

</style>
```

The following are a general descriptions of CSS combinators:

- **Descendant combinator:** Selects all elements that are descendants of a specified ancestor. Selects all elements that are descendants of a specified ancestor element, regardless of how deeply nested they are.

```
ancestor descendant {
  /* CSS rules */
}
```

- **Child combinator:** Selects all direct children of the specified parent. It targets elements that are immediately nested within the parent, excluding any deeper descendants.

```
parent > child {
  /* CSS rules */
}
```

- **Adjacent sibling combinator:** Selects an element that is directly preceded by a sibling element. The following rule selects the `adjacent` element that immediately follows `element`.

```
element + adjacent {
  /* CSS rules */
}
```

- **General sibling combinator:** Selects all sibling elements that follow a specified element. It targets all elements that come after the specified element, not just the immediately following one unlike the **Adjacent sibling combinator**. The following selects all `sibling` elements that come after `element` in the document structure.

```
element ~ sibling {
  /* CSS rules */
}
```


Text Alignment in Table cells

Text can be aligned in two ways – horizontally and vertically.

- To **horizontally align** text, use the `text-align` property.
- To **vertically align** text, use the `vertical-align` property.

Merging HTML cells

Like text alignment, html cells can be merged in two ways – horizontally (merging columns) vertically (merging rows).

- To merge **rows**, use the `rowspan` attribute. The result will be similar to the picture below.

- To merge **columns**, use the `colspan` attribute. The result will be similar to the picture below.

Understanding Padding and Border-spacing in HTML

In HTML, "padding" is a term used to describe the space between the content of an element and its border. It allows you to control the amount of space between the content and the border of an element. Padding is commonly used to improve the visual appearance of elements by adding space around them.

Padding can be applied to various HTML elements such as `<div>`, `<p>`, ``, etc. It is typically defined using CSS (Cascading Style Sheets), either inline or in an external stylesheet

On the other hand, `border-spacing` is a CSS property used only with the `<table>` element in HTML. `border-spacing` adds spacing between the borders of adjacent cells within a table. In addition, it also adds spacing between the cells and the outer border of the table itself.

Note: Both `border-spacing` and `padding` **require** `border-collapse: separate` to be set to separate. If this requirement is not met, `border-spacing` and `padding` will have no effect on the table.

Despite the similarities in the ability of the **selectors** the effect of one cannot effectively be recreated by the other.

1. For example, when styled for **padding** like this:

```
<table style="border-collapse: separate; width: 1000px; padding: 100px" border="1">
```

The result is:

1	2	3
4		5
6	7	

2. And, when styled for **border-spacing** like this:

```
<table style="border-collapse: separate; width: 1000px; border-spacing: 100px 10px" border="1">
```

The result is:

1	2	3
4		5
6	7	

Horizontal Rule and Line Break

Horizontal Rule: The `<hr>` element is used to create a thematic break or horizontal rule in the content. It typically renders as a horizontal line or rule across the width of the containing element.

```
<p>This is paragraph one.</p>
<hr>
<p>This is paragraph two.</p>
```

Line Break: The `
` element is used to insert a line break or newline within text content. It forces the text or content following it to start on a new line. It's useful for creating line breaks within paragraphs, addresses, or other text where you want to control the line breaks manually.

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