**Significance of CSS**

**CSS** stands for **Cascading Style Sheet.**



* **CSS** is a style sheet language used to control the **presentation** (look and layout) of web pages written in HTML.
* It defines how elements should be **displayed**, including colors, fonts, spacing, and overall layout.
* Without CSS, web pages use the **browser’s default styles**, resulting in a plain and basic appearance.

**Structure or Syntax of CSS**

* The syntax of CSS is slightly different from that of an HTML.
* CSS uses **(curly braces { } ), (colons : )** and **(semicolon ; )**.

**Syntax:**

selector

{

property : value;

property : value;

| |

property : value;

}

**Types of CSS**

| **Type of CSS** | **Description** | **Where It Is Written** | **Syntax (Example)** |
| --- | --- | --- | --- |
| **1. Inline CSS** | Used to style a **single HTML element** directly. It has the **highest priority**. | Inside the HTML tag using the style attribute. | <h1 style="color:blue; font-size:25px;">Heading</h1> |
| **2. Internal (Embedded) CSS** | Used to style elements **within the same HTML page**. | Inside the <style> tag in the <head> section of the HTML document. | <head><style> p { color: red; font-size: 18px; } </style></head> |
| **3. External CSS** | Used to apply styles to **multiple web pages** using a separate .css file. | In an external stylesheet linked using the <link> tag in the <head> section. | <head><link rel="stylesheet" href="style.css"></head>*(style.css file contains CSS rules)* |

**Various CSS Selectors**

| **Selector Type** | **Description** | **Syntax** | **Example** |
| --- | --- | --- | --- |
| **Element Selector** | Selects all elements of a specific type/tag. Lowest specificity (except universal). | **element** | p { color: black; } |
| **ID Selector** | Selects a single element with a specific id attribute. Very high specificity. | **#id** | #header { color: red; } |
| **Class Selector** | Selects elements with a specific class. Medium specificity. | **.class** | .menu { font-size: 16px; } |
| **Attribute Selector** | Selects elements based on an attribute or attribute value. | **[attr], [attr=value]** | [type="text"] { border: 1px solid #ccc; } |
| **Universal Selector** | Selects all elements. Lowest specificity. | **\*** | \* { margin: 0; padding: 0; } |
| **Descendant Selector** | Selects elements that are descendants (any level) of a specified ancestor. | **ancestor descendant** | div p { color: blue; } |
| **Child Selector** | Selects elements that are direct children of a specified parent. | **parent > child** | ul > li { list-style: none; } |
| **Grouping Selector** | Groups multiple selectors and applies the same styles. | **selector1, selector2** | h1, h2, h3 { font-family: Arial; } |

## ****CSS Specificity (Priority Order)****

When multiple CSS rules target the same element, the browser decides which one to apply based on **specificity** (priority):

| **Priority Level** | **Selector Type** | **Example** |
| --- | --- | --- |
| **1 (Highest)** | Inline Styles | <h1 style="color:pink;">Heading</h1> |
| **2** | IDs | #title { color: red; } |
| **3** | Classes, pseudo-classes, attribute selectors | .title { color: green; }, :hover, [type='text'] |
| **4 (Lowest)** | Elements and pseudo-elements | h1 { color: blue; }, ::before |

# ****Text Properties****

These properties are used to **style, format, and control the appearance** of text on a webpage, including alignment, spacing, transformation, decoration, font type, and size.

## ****Text Properties****

| **Property** | **Description** | **Syntax** | **Possible Values** |
| --- | --- | --- | --- |
| **color** | Sets the color of the text | color: color; | Named colors (e.g., red), HEX (#ff0000), RGB (rgb(255,0,0)), RGBA |
| **text-align** | Aligns text inside an element | text-align: value; | left, right, center, justify, start, end |
| **text-indent** | Adds indentation to the first line | text-indent: length; | Any length (px, em, %) |
| **text-transform** | Controls text capitalization | text-transform: value; | none, capitalize, uppercase, lowercase |
| **text-decoration** | Adds decoration like underline or line-through | text-decoration: value; | none, underline, overline, line-through, underline overline |
| **letter-spacing** | Controls space between letters | letter-spacing: length; | Normal (normal), or custom spacing (px, em). Negative values are allowed. |
| **word-spacing** | Controls space between words | word-spacing: length; | Normal (normal), or custom spacing (px, em). Negative values are allowed. |
| **line-height** | Sets space between lines | line-height: value; | normal, number (e.g., 1.5), length (px, em), %. Negative values are **not** allowed. |
| **text-shadow** | Adds shadow to text | text-shadow: h-shadow v-shadow blur color; | none or values like 2px 2px 5px gray (can add multiple shadows separated by commas) |

# ****Font Properties****

| **Property** | **Description** | **Syntax** | **Example** | **Possible Values** |
| --- | --- | --- | --- | --- |
| **font-family** | Specifies the font type | font-family: "FontName", fallback; | font-family: "Arial", sans-serif; | Any font name: "Arial", "Times New Roman", "Courier New", generic families: serif, sans-serif, monospace, cursive, fantasy |
| **font-size** | Sets the font size | font-size: size; | font-size: 20px; | Length units: px, em, rem, %; Keywords: xx-small, x-small, small, medium, large, x-large, xx-large, smaller, larger |
| **font-style** | Defines style of text | font-style: value; | font-style: italic; | normal, italic |
| **font-weight** | Sets font thickness | font-weight: value; | font-weight: bold; | normal, bold, bolder, lighter, 100, 200, 300, 400, 500, 600, 700, 800, 900 |
| **font-variant** | Displays text in small-caps | font-variant: value; | font-variant: small-caps; | normal, small-caps |
| **font** (shorthand) | Sets all font properties in one line | font: [style] [variant] [weight] [size]/[line-height] [family]; | font: italic small-caps bold 16px/1.5 "Times New Roman", serif; | Combines all above properties in one; values are the same as individual properties: <style> <variant> <weight> <size>/<line-height> <family> |

## ****Google Fonts****

## ****How to Embed Google Fonts****

### ****Method 1: Using**** <link> ****tag (Recommended)****

### ****Method 2: Using**** @import ****in CSS****

**CSS Borders**

A **border** is a line that wraps around an HTML element’s **content and padding**. It visually separates elements or highlights them on a web page.

| **Property** | **Description** | **Why to Use** | **Possible Values / Syntax** |
| --- | --- | --- | --- |
| border-style | Defines the style of the border | To give different visual effects like solid, dashed, dotted, etc. | none, solid, dashed, dotted, double |
| border-width | Sets the thickness of the border | To adjust border size | length (px, em, rem) |
| border-color | Sets the color of the border | To visually match design or highlight elements | Named colors (red), hex (#FF0000), RGB (rgb(255,0,0)) |
| border (shorthand) | Combines style, width, color in one line | To quickly define a complete border | border: 2px solid red; |
| border-top | Sets border properties for the top side | To style only the top edge | border-top: 3px dashed blue; |
| border-right | Sets border properties for the right side | To style only the right edge | border-right: 2px solid green; |
| border-bottom | Sets border properties for the bottom side | To style only the bottom edge | border-bottom: 4px dotted orange; |
| border-left | Sets border properties for the left side | To style only the left edge | border-left: 5px double purple; |

**Box Properties**

| **Property** | **Description** | **Why to Use** | **Possible Values / Syntax / Examples** |
| --- | --- | --- | --- |
| **margin** | Creates **space outside** the element (outside border). | To **separate elements** on the page. | **Values:**  • auto → browser calculates margin (useful for centering)  • length → px, em, %, negative values allowed  **Shorthand examples:**  • 4 values → margin: 25px 50px 75px 100px; → top=25px, right=50px, bottom=75px, left=100px  • 3 values → margin: 25px 50px 75px; → top=25px, right/left=50px, bottom=75px  • 2 values → margin: 25px 50px; → top/bottom=25px, right/left=50px  • 1 value → margin: 25px; → all sides=25px |
| **padding** | Creates **space inside** the element (between content & border). | To **add inner spacing** inside the element. | **Values:**  • length → px, em, %  **Shorthand examples:**  • 4 values → padding: 25px 50px 75px 100px; → top=25px, right=50px, bottom=75px, left=100px  • 3 values → padding: 25px 50px 75px; → top=25px, right/left=50px, bottom=75px  • 2 values → padding: 25px 50px; → top/bottom=25px, right/left=50px  • 1 value → padding: 25px; → all sides=25px |
| **width** | Specifies the **width** of an element’s content area. | To **control horizontal size** of the element. | **Values:** auto, length (px, em, %), |
| **height** | Specifies the **height** of an element’s content area. | To **control vertical size** of the element. | **Values:** auto, length (px, em, %), |
| **box-sizing** | Defines **how width and height are calculated** — whether padding and borders are included. | To **control layout behavior** when adding padding/borders. | **Values:**  • content-box (default): width/height exclude padding & border.  • border-box: width/height include padding & border. |
| **box-shadow** | Adds **shadow effects** around an element’s frame. | To **add depth**, **hover effects**, or **visual highlights**. | **Syntax:** box-shadow: h-offset v-offset blur spread color;  **Example:** box-shadow: 2px 2px 5px gray; |

* 1. **CSS background properties**

| **Property** | **Description** | **Syntax** | **Key Values / Notes** |
| --- | --- | --- | --- |
| **background-color** | Sets the background color of an element | background-color: color; | Any valid color (red, #ffc0cb, rgb(255,0,0)) |
| **background-image** | Sets an image as the background | background-image: url("image.jpg"); | Use image URL; |
| **background-repeat** | Controls how background image repeats | background-repeat: repeat; | repeat, repeat-x, repeat-y, no-repeat  default repeats if not specified |
| **background-position** | Sets the starting position of a background image | background-position: top; | top, bottom, left, right, center, x% y% |
| **background-size** | Specifies the size of the background image | background-size: auto; | auto, cover, contain, width height |
| **background-attachment** | Sets whether background scrolls with content | background-attachment: scroll; | scroll, fixed, local |
| **background-clip** | Determines how far the background extends | background-clip: border-box; | border-box, padding-box, content-box |
| **background (shorthand)** | Combines color, image, repeat, position, size, attachment | background: [color] [image] [repeat] [position] [size] [attachment]; | Example: background: pink url("scenary.jfif") no-repeat top right cover fixed; |

**Pseudo classes**

| **Pseudo-class** | **What It Does** |
| --- | --- |
| :hover | Changes color when mouse hovers |
| :active | Changes color when clicked |
| :focus | Highlights input field when clicked |
| :checked | Styles label when checkbox is checked |
| :link / :visited | Colors for unvisited/visited links |
| :nth-child(2) | Targets the 2nd list item |
| :not(.special) | Styles all list items except .special |

**Pseudo Elements**

| **Pseudo-element** | **Description** | **Syntax** | **Example** | **Result / Effect** |
| --- | --- | --- | --- | --- |
| **::first-line** | Styles the **first line** of a block-level element. | p::first-line { property: value; } | p::first-line { color: red; font-weight: bold; } | First line of the paragraph appears red and bold. |
| **::first-letter** | Styles the **first letter** of a block-level element. | p::first-letter { property: value; } | p::first-letter { font-size: 50px; color: red; } | First letter of the paragraph becomes large and red. |
| **::before** | Inserts content **before** an element’s actual content. | selector::before { content: "..."; } | p::before { content: "Note: "; color: red; } | Adds “Note:” before each paragraph. |
| **::after** | Inserts content **after** an element’s actual content. | selector::after { content: "..."; } | p::after { content: " ✔"; color: green; } | Adds a green checkmark after the paragraph. |
| **::marker** | Styles the **marker (bullet or number)** of list items. | li::marker { property: value; } | li::marker { color: red; font-size: 20px; } | List bullets or numbers appear red and large. |
| **::selection** | Styles the **highlighted text** selected by the user. | ::selection { property: value; } | ::selection { background: blueviolet; color: white; } | Selected text appears white on a blue-violet background. |

**Display Property**

| **Display Type** | **Description** | **Behavior / Use Case** | **Visual Behavior** |
| --- | --- | --- | --- |
| **inline** | Displays elements **in a line**, without starting on a new line. | Does **not accept width/height**. Common for <span>, <a>, <strong>. | Elements sit **side by side** in a single line. |
| **block** | Displays element as a **block**, starting on a new line. | Takes **full width** available and allows **width/height** to be set. | Each element appears **on a new line**. |
| **inline-block** | Combines features of **inline** and **block**. | Appears **inline**, but allows **width and height**. | Boxes are **side by side**, but **size-controllable**. |
| **none** | **Hides** the element completely (removed from layout). | Element takes **no space** on the page. | The element is **invisible** and **does not occupy space**. |
| **flex** | Displays element as a **flex container**. | Allows flexible alignment and distribution of child elements. | Items are **arranged in a row** (or column) with flexible spacing. |
| **grid** | Displays element as a **grid container**. | Divides layout into **rows and columns** for advanced control. | Elements are placed in a **grid layout** (rows and columns). |

**Flexbox**

| **Property** | **Description** | **Possible Values** | **Syntax / Example** |
| --- | --- | --- | --- |
| **display: flex** | Defines a flex container to arrange items flexibly. | flex, inline-flex | div { display: flex; } |
| **flex-direction** | Defines the direction of flex items. | row (default), row-reverse, column, column-reverse | flex-direction: row; |
| **justify-content** | Aligns items horizontally (along the main axis). | flex-start, flex-end, center, space-between, space-around, space-evenly | justify-content: space-between; |
| **align-items** | Aligns items vertically (along the cross axis). | stretch (default), flex-start, flex-end, center | align-items: center; |
| **flex-wrap** | Determines whether flex items wrap onto multiple lines. | nowrap (default), wrap, wrap-reverse | flex-wrap: wrap; |
| **gap** | Defines the space between flex items. | Any CSS length unit (px, em, %) | gap: 15px; |
| **order** | Specifies the display order of flex items. | Integer values (0 default, can be positive or negative) | order: 2; |
| **flex** *(shorthand)* | Sets how an item grows, shrinks, and defines its base size. | flex: grow shrink basis; e.g., flex: 1 0 100px; | flex: 1; or flex: 1 0 200px; |

**Grid Layout Properties**

| **Property** | **Description** | **Possible Values** | **Default Value** | **Syntax / Example** |
| --- | --- | --- | --- | --- |
| **grid-template-columns** | Defines the number and width of columns in a grid layout. | Fixed units (px, em, %) | none | grid-template-columns: 200px 1fr 2fr; |
| **grid-template-rows** | Defines the number and height of rows in a grid layout. | Fixed units (px, em, %), | none | grid-template-rows: 100px auto 100px; |
| **gap (grid-gap)** | Defines the space between rows and columns. | Any CSS length (px, em, %) | 0 | gap: 10px; or gap: 20px 40px; |
| **grid-column** | Specifies the horizontal position of an item within the grid columns (start / end lines). | <start-line> / <end-line> | auto | grid-column: 1 / 3; (spans across 2 columns) |
| **grid-row** | Specifies the vertical position of an item within the grid rows (start / end lines). | <start-line> / <end-line> | auto | grid-row: 1 / 2; (spans across 1 row) |

**CSS Positioning Properties**

| **Property** | **Description** | **Possible Values** | **Default Value** | **Syntax / Example** |
| --- | --- | --- | --- | --- |
| **position** | Defines how an element is positioned in the document. | static, relative, absolute, fixed, sticky | static | position: absolute; |
| **top** | Distance between element and top edge of container. | Any CSS length (px, %, auto) | auto | top: 20px; |
| **right** | Distance between element and right edge of container. | Any CSS length (px, %, auto) | auto | right: 10px; |
| **bottom** | Distance between element and bottom edge of container. | Any CSS length (px, %, auto) | auto | bottom: 15px; |
| **left** | Distance between element and left edge of container. | Any CSS length (px, %, auto) | auto | left: 30px; |
| **z-index** | Controls stack order of overlapping elements. | Integer (auto, positive or negative values) | auto | z-index: 2; (Higher = on top) |

* **static: Default, element follows normal document flow.**
* **relative: Moved relative to its normal position.**
* **absolute: Positioned relative to the nearest positioned ancestor.**
* **fixed: Stays fixed in place even when scrolling.**
* **z-index: Controls which element appears on top of others.**