What is CSS :  
 It is a language that is used to describe the style of a document.  
 CSS is used to **style HTML elements** — colors, fonts, layouts, animations, responsiveness, etc.

Types of CSS :

There are 3 types of Css declaration

**1.Inline CSS** – inside the HTML tag

**2.Internal CSS** – inside <style> tag in the HTML file

**3.External CSS** – linked .css file (best practice)

Inlinecss > Internal css > External css

Selectors:

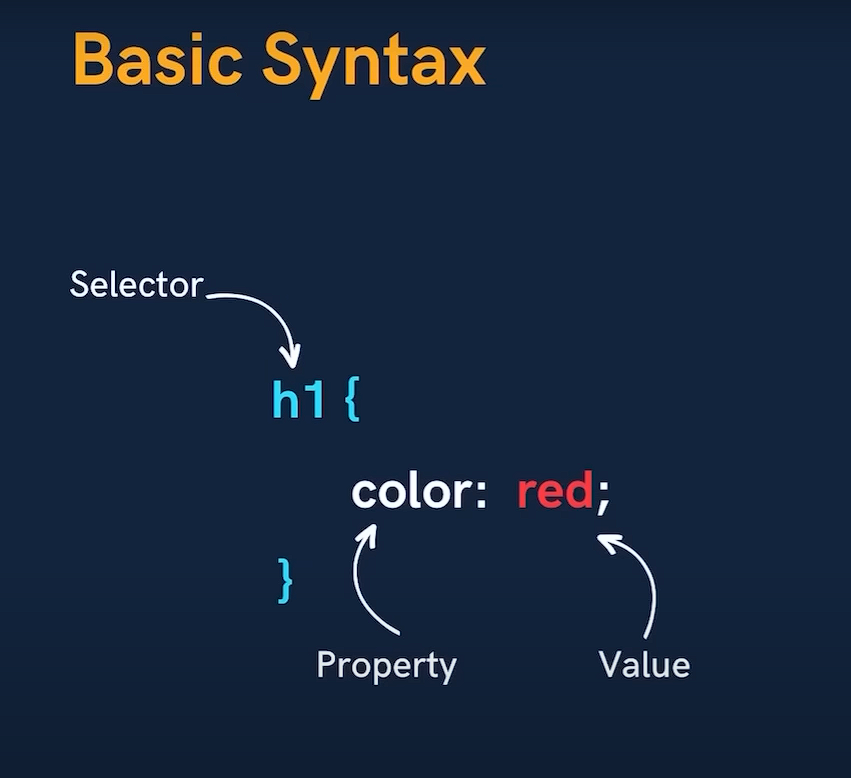
* Universal selector   
  \*{ }
* Element Selector  
  h1{ }
* Class Selector  
  .container{ }
* Id Selector  
  #About{ }

|  |  |  |
| --- | --- | --- |
| **Selector** | **Example** | **Description** |
| **Element** | p { color: red; } | Styles all <p> tags |
| **Class** | .box { border: 1px solid; } | Styles elements with class="box" |
| **ID** | #main { background: yellow; } | Styles element with id="main" |
| **Universal** | \* { margin: 0; padding: 0; } | Styles everything |
| **Group** | h1, h2, p { color: blue; } | Styles multiple elements |

selector {

  property: value;

}



How to add shadow property:

selector{

    text-shadow: horizontal-offset vertical-offset blur-radius color#5555;

}

Example:



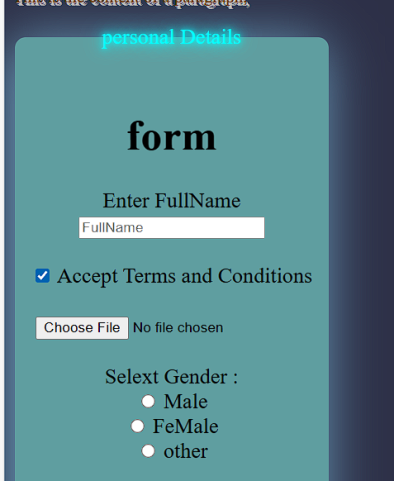
 #heading1 {

            text-shadow: 2px 2px 5px rgb(208, 208, 208);

        }

**Glow Effect :**

**For box:**

****

Glow

#personal {

            background-color: cadetblue;

            box-shadow: -5px 5px 50px rgba(140, 217, 253, 0.7);

            width: fit-content;

            border-radius: 10px;

            text-align: center;

            font-size: 20px;

            border: 0px solid black;

        }

**UNITS** in css

In **CSS**, **units** define the measurement for properties like width, height, margin, padding, font size, etc.  
They tell the browser **“how much”** of something to apply.

**1. Absolute Units**

These units are fixed and **do not change** regardless of screen size or parent element.

**When to use:**

* For fixed layouts or printing.
* For screen elements, **px** is most common.

|  |  |  |
| --- | --- | --- |
| **Unit** | **Meaning** | **Example** |
| px | Pixels (most commonly used) | font-size: 16px; |
| cm | Centimeters (rarely used in web) | width: 10cm; |
| mm | Millimeters | height: 50mm; |
| in | Inches (1in = 96px) | width: 1in; |
| pt | Points (1pt = 1/72 inch) | font-size: 12pt; |
| pc | Picas (1pc = 12pt) | width: 2pc; |

1rem = 16px

## ****2. Relative Units****

These units **adjust based on context**, like the parent element size, root font size, or viewport.

|  |  |  |
| --- | --- | --- |
| **Unit** | **Relative To** | **Example** |
| % | Parent element size | width: 50%; (half of the parent width) |
| em | Font size of the parent element | font-size: 2em; (2× parent font size) |
| rem | Font size of the root (html) element | font-size: 1.5rem; |
| vw | 1% of the viewport width | width: 50vw; (half the screen width) |
| vh | 1% of the viewport height | height: 100vh; (full screen height) |
| vmin | 1% of the smaller side of viewport | font-size: 3vmin; |
| vmax | 1% of the larger side of viewport | font-size: 3vmax; |
| ch | Width of the character 0 in current font | width: 30ch; |
| ex | Height of the letter x in current font | line-height: 2ex; |

**Flex :**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Aspect** | **Property** | **Values** | **Description / Usage** | **Example** |
| **Container Setup** | display | flex / inline-flex | Enables flexbox on a container. | display: flex; |
| **Main Axis Direction** | flex-direction | row (default), row-reverse, column, column-reverse | Sets the direction of items. | flex-direction: column; |
| **Wrapping** | flex-wrap | nowrap (default), wrap, wrap-reverse | Controls if items wrap to the next line. | flex-wrap: wrap; |
| **Combine Direction & Wrap** | flex-flow | <flex-direction> <flex-wrap> | Shorthand for direction + wrapping. | flex-flow: row wrap; |
| **Main Axis Alignment** | justify-content | flex-start, flex-end, center, space-between, space-around, space-evenly | Aligns items along the main axis. | justify-content: space-between; |
| **Cross Axis Alignment** | align-items | flex-start, flex-end, center, baseline, stretch | Aligns items on the cross axis. | align-items: center; |
| **Multi-line Alignment** | align-content | flex-start, flex-end, center, space-between, space-around, stretch | Aligns multiple rows when wrapped. | align-content: space-around; |
| **Individual Item Alignment** | align-self | same as align-items values | Overrides alignment for a single item. | align-self: flex-end; |
| **Item Flexibility** | flex | flex-grow flex-shrink flex-basis | Shorthand for controlling size. | flex: 1 0 auto; |
| **Growth** | flex-grow | number (0 default) | How much the item grows. | flex-grow: 1; |
| **Shrink** | flex-shrink | number (1 default) | How much the item shrinks. | flex-shrink: 0; |
| **Initial Size** | flex-basis | auto, px, %, etc. | Sets the initial size. | flex-basis: 200px; |
| **Order** | order | number (0 default) | Changes the visual order of items. | order: 2; |

**Grid:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Aspect** | **Property** | **Values** | **Description / Usage** | **Example** |
| **Container Setup** | display | grid / inline-grid | Enables CSS Grid layout. | display: grid; |
| **Columns** | grid-template-columns | e.g. 200px 1fr 1fr, repeat(3, 1fr) | Defines column sizes. | grid-template-columns: 1fr 1fr; |
| **Rows** | grid-template-rows | e.g. 100px auto, repeat(2, minmax(100px, auto)) | Defines row sizes. | grid-template-rows: auto 100px; |
| **Gap** | gap / column-gap / row-gap | px, %, etc. | Space between grid items. | gap: 20px; |
| **Item Placement** | grid-column / grid-row | start / end line numbers or spans | Positions items in specific tracks. | grid-column: 1 / 3; |
| **Auto Placement** | grid-auto-flow | row, column, dense | Controls how items auto-place. | grid-auto-flow: column; |
| **Auto Rows/Cols** | grid-auto-rows / grid-auto-columns | size values | Sets size for implicit rows/cols. | grid-auto-rows: 100px; |
| **Justify in Container** | justify-items | start, end, center, stretch | Aligns items horizontally in cell. | justify-items: center; |
| **Align in Container** | align-items | start, end, center, stretch | Aligns items vertically in cell. | align-items: center; |
| **Overall Alignment** | place-items | align-items justify-items | Shorthand for aligning items. | place-items: center; |
| **Content Alignment** | justify-content / align-content | same as flex values | Aligns the whole grid in container. | justify-content: space-between; |
| **Named Areas** | grid-template-areas | custom area names | Creates named layout areas. | grid-template-areas: "header header" "sidebar content"; |
| **Item Area Assignment** | grid-area | matches template name | Places an item in a named area. | grid-area: header; |
| **Span Items** | grid-column / grid-row | span X | Makes an item span multiple cells. | grid-column: span 2; |