# Introduction to CSS

## What is CSS?

- **CSS (Cascading Style Sheets)** is used to style and format the layout of HTML elements.
- It separates **content (HTML)** from **presentation (CSS)**.
- Helps make web pages **attractive, responsive, and consistent**.

## Why use CSS?

1. Improves design (colors, fonts, layout).
2. Makes websites responsive across devices.
3. Reusable styles (write once, use multiple times).
4. Faster page loading (less inline styling).

## Types of CSS

1. **Inline CSS** – written inside an HTML element using the `style` attribute.

```
<p style="color: blue; font-size: 18px;">This is inline CSS</p>
```

2. **Internal CSS** – written inside `<style>` tag in the HTML `<head>`.

```
<head>
  <style>
    p {
      color: green;
      font-size: 20px;
    }
  </style>
</head>
<body>
  <p>This is internal CSS</p>
</body>
```

3. **External CSS** – written in a separate `.css` file and linked with `<link>`.

```
<!-- index.html -->
<head>
  <link rel="stylesheet" href="style.css">
</head>

<!-- style.css -->
p {
  color: red;
  font-size: 22px;
}
```

**Best Practice:** Always use **External CSS** for large projects.

## CSS Syntax

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

**Example:**

```
h1 {
  color: blue;
  font-size: 30px;
}
```

- **Selector** → selects HTML element (`h1`)
- **Property** → defines what you want to change (`color`, `font-size`)
- **Value** → actual styling value (`blue`, `30px`)

### Output Example

```
<!DOCTYPE html>
<html>
<head>
  <style>
    h1 {
      color: purple;
      text-align: center;
    }
    p {
      color: gray;
      font-size: 18px;
    }
  </style>
</head>
<body>
  <h1>Hello CSS</h1>
  <p>This is styled with CSS.</p>
</body>
</html>
```

# CSS Selectors

## What are Selectors?

- A **selector** is used to target HTML elements so that we can apply styles to them.
- They define **"which elements"** the CSS rules will apply to.

## Types of CSS Selectors

### 1. Universal Selector (`*`)

Selects **all elements** on the page.

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

Useful for resetting default browser styles.

### 2. Element Selector

Selects all elements of a given type.

```
p {
  color: blue;
}
```

All <p> elements will be blue.

### 3. ID Selector (#)

Selects an element with a specific **id**. (Unique for each element)

```
#main-heading {
  color: red;
}
<h1 id="main-heading">Welcome</h1>
```

### 4. Class Selector (.)

Selects elements with a specific **class**. (Can be reused multiple times)

```
.highlight {
  background-color: yellow;
}
<p class="highlight">This is highlighted</p>
```

### 5. Grouping Selector (,)

Apply same style to multiple elements.

```
h1, h2, p {
  font-family: Arial;
}
```

### 6. Combinators

Used to define relationships between elements.

- **Descendant (`space`)** → selects all elements inside another element

  ```
  div p {
    color: green;
  }
  ```
  All <p> inside  will be green.

- **Child (>)** → selects direct children only

  ```
  div > p {
    color: orange;
  }
  ```

- **Adjacent sibling (+)** → selects element immediately after another

  ```
  h1 + p {
    font-weight: bold;
  }
  ```

- **General sibling (~)** → selects all siblings after an element

```
h1 ~ p {
  color: purple;
}
```

### 7. Attribute Selectors

Select elements based on attributes.

```
input[type="text"] {
  border: 2px solid blue;
}
a[href^="https"] {
  color: green;
}
a[href$=".pdf"] {
  color: red;
}
```

Note: `^=` = starts with, `$=` = ends with, `*=` = contains

### 8. Pseudo-classes

Select elements based on **state**.

```
a:hover {
  color: red;
}
input:focus {
  border-color: green;
}
li:nth-child(2) {
  color: orange;
}
```

### 9. Pseudo-elements

Select **parts of elements**.

```
p::first-letter {
  font-size: 30px;
  color: blue;
}
p::before {
  content: "Before Elment";
}
p::after {
  content: "After Element";
}
```

# Colors and Background in CSS:

CSS supports multiple ways to define colors:

**1. Color Names**

```
h1 {
  color: red;
}
```

2. **Hexadecimal (`#RRGGBB`)**

```
p {
 color: #008000; /* green */
}
```

3. **RGB (Red Green Blue)**

```
p {
 color: rgb(255, 0, 0); /* red */
}
```

4. **RGBA (Red Green Blue Alpha)** → last value = transparency (0 to 1)

```
p {
 color: rgba(0, 0, 255, 0.5); /* semi-transparent blue */
}
```

5. **HSL (Hue Saturation Lightness)**

```
p {
 color: hsl(120, 100%, 40%); /* green shade */
}
```

6. **HSLA (Hue Saturation Lightness Alpha)**

```
p {
 color: hsla(200, 100%, 50%, 0.6);
}
```

# Background Properties

1. Background Color

```
body {
  background-color: lightblue;
}
```

2. Background Image

```
body {
 background-image: url("bg.jpg");
}
```

3. Background Repeat

```
body {
 background-repeat: no-repeat; /* default is repeat */
}
```

4. Background Position

```
body {
 background-position: center top;
```

}

5. Background Size

```
body {
 background-size: cover; /* cover entire screen */
 /* other values: auto, contain, specific px/% */
}
```

6. Background Attachment

```
body {
 background-attachment: fixed; /* background stays fixed while scrolling */
}
```

7. Background Shorthand

Instead of writing separately, you can write in one line:

```
body {
 background: url("bg.jpg") no-repeat center center/cover;
}
```

## Example

```
<!DOCTYPE html>
<html>
<head>
  <style>
    body {
      background: url("https://picsum.photos/800/400") no-repeat center
center/cover;
      color: white;
      font-family: Arial;
      text-align: center;
      height: 100vh;
    }
    h1 {
      background-color: rgba(0, 0, 0, 0.6);
      padding: 20px;
    }
    p {
      color: hsl(50, 100%, 50%);
    }
  </style>
</head>
<body>
  <h1>CSS Background Example</h1>
  <p>This text has color using HSL.</p>
</body>
</html>
```