CSS selectors are patterns used to select and style HTML elements. They provide a mechanism to apply styles selectively rather than applying them universally. Types include:

* Element Selector – selects elements by name (p { })
* Class Selector – selects elements with a class (.highlight { })
* ID Selector – selects elements with a unique id (#header { })
* Attribute Selector – styles elements with specific attributes (input[type="text"] { })
* Pseudo-classes – styles elements in a particular state (a:hover { })
* Pseudo-elements – styles a part of an element (p::first-line { })

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>CSS Selectors Example</title>

<style>

/\* Element Selector \*/

p { color: blue; font-size: 16px; }

/\* Class Selector \*/

.highlight { background-color: yellow; }

/\* ID Selector \*/

#special { font-weight: bold; color: red; }

/\* Pseudo-class \*/

a:hover { color: orange; }

/\* Pseudo-element \*/

p::first-line { text-transform: uppercase; }

</style>

</head>

<body>

<p>This is a normal paragraph.</p>

<p class="highlight">This paragraph is highlighted.</p>

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

<a href="#">Hover over me</a>

</body>

</html>

**Output / Analysis:**

* Paragraph text appears in blue.
* Highlighted paragraph has yellow background.
* Special paragraph is bold and red.
* Link changes color when hovered.
* First line of each paragraph is in uppercase.

**Colors & Backgrounds**

**Description:**  
CSS allows you to style colors using named colors, HEX, RGB, and HSL. Backgrounds can be solid, images, or gradients. Gradients create smooth color transitions and are widely used in modern UI design.

**Example (HTML + CSS):**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Colors & Backgrounds</title>

<style>

body {

background: linear-gradient(to right, red, yellow);

color: white;

font-family: Arial, sans-serif;

}

.pattern-box {

width: 200px;

height: 100px;

background-image: url('https://via.placeholder.com/200x100');

background-size: cover;

margin-top: 20px;

}

</style>

</head>

<body>

<h1>CSS Colors & Backgrounds</h1>

<div class="pattern-box"></div>

</body>

</html>

**Box Model**

**Description:**  
The CSS box model consists of four layers: content, padding, border, and margin. It is the foundation for spacing and layout.

**Example (HTML + CSS):**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Box Model Example</title>

<style>

.box {

width: 200px;

padding: 20px;

border: 5px solid black;

margin: 30px;

background-color: lightblue;

}

</style>

</head>

<body>

<div class="box">Box Model Example</div>

</body>

</html>

**Positioning & Display**

**Description:**  
CSS positioning controls element placement.

* static: default
* relative: relative to its normal position
* absolute: relative to nearest positioned ancestor
* fixed: fixed to viewport
* sticky: toggles between relative and fixed

**Example (HTML + CSS):**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Positioning Example</title>

<style>

.static { background: lightgray; }

.relative { position: relative; top: 20px; left: 20px; background: lightgreen; }

.absolute { position: absolute; top: 50px; left: 50px; background: lightcoral; }

</style>

</head>

<body>

<div class="static">Static Box</div>

<div class="relative">Relative Box</div>

<div class="absolute">Absolute Box</div>

</body>

</html>

**Flexbox Layout**

**Description:**  
Flexbox simplifies alignment and distribution in one-dimensional layouts. Commonly used for navigation bars, cards, and rows.

**Example (HTML + CSS):**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Flexbox Example</title>

<style>

nav {

display: flex;

justify-content: space-around;

background-color: #333;

padding: 10px;

}

nav a {

color: white;

text-decoration: none;

}

</style>

</head>

<body>

<nav>

<a href="#">Home</a>

<a href="#">About</a>

<a href="#">Services</a>

<a href="#">Contact</a>

</nav>

</body>

</html>

**Grid Layout**

**Description:**  
CSS Grid is a two-dimensional layout system, allowing precise control over rows and columns. Suitable for dashboards and complex page layouts.

**Example (HTML + CSS):**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Grid Example</title>

<style>

.container {

display: grid;

grid-template-columns: 1fr 2fr;

grid-gap: 10px;

}

.box { background: lightblue; padding: 20px; }

</style>

</head>

<body>

<div class="container">

<div class="box">Sidebar</div>

<div class="box">Main Content</div>

</div>

</body>

</html>

**Typography & Fonts**

**Description:**  
CSS controls typography: font-family, size, weight, style, line-height, letter-spacing. Google Fonts offer modern typefaces.

**Example (HTML + CSS):**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Typography Example</title>

<link href="https://fonts.googleapis.com/css2?family=Roboto&display=swap" rel="stylesheet">

<style>

body { font-family: 'Roboto', sans-serif; line-height: 1.6; }

h1 { text-transform: uppercase; }

</style>

</head>

<body>

<h1>Typography Example</h1>

<p>This paragraph demonstrates Google Fonts and line-height spacing.</p>

</body>

</html>

**Transitions**

**Description:**  
Transitions create smooth changes between CSS property states, enhancing UI feedback.

**Example (HTML + CSS):**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Transition Example</title>

<style>

button {

background: blue;

color: white;

padding: 10px 20px;

border: none;

cursor: pointer;

transition: background 0.5s, transform 0.5s;

}

button:hover {

background: green;

transform: scale(1.2);

}

</style>

</head>

<body>

<button>Hover Me</button>

</body>

</html>

**Animations**

**Description:**  
CSS animations use @keyframes for motion effects, allowing continuous or finite transitions.

**Example (HTML + CSS):**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Animation Example</title>

<style>

.ball {

width: 50px;

height: 50px;

background: red;

border-radius: 50%;

position: relative;

animation: bounce 2s infinite;

}

@keyframes bounce {

0%, 100% { top: 0; }

50% { top: 200px; }

}

</style>

</head>

<body>

<div class="ball"></div>

</body>

</html>

**Transformations**

**Description:**  
CSS transforms allow elements to rotate, scale, translate, or skew for visual effects.

**Example (HTML + CSS):**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Transform Example</title>

<style>

.box {

width: 100px;

height: 100px;

background: orange;

transition: transform 0.5s;

}

.box:hover {

transform: rotate(45deg) scale(1.5);

}

</style>

</head>

<body>

<div class="box"></div>

</body>

</html>