

CSS — Cascading Style Sheets

1. What is CSS?

CSS is used to style and layout web pages — controlling colors, fonts, spacing, positioning, and more. CSS is applied to HTML elements using selectors and property-value pairs.

Ways to Apply CSS

Applying CSS

```
<!-- 1. Inline CSS -->
<p style="color: red; font-size: 18px;">Hello</p>

<!-- 2. Internal CSS (in <head>) -->
<style>
  p { color: blue; }
</style>

<!-- 3. External CSS (recommended) -->
<link rel="stylesheet" href="styles.css">
```

NOTE: Always use external CSS for real projects. It keeps HTML clean and CSS reusable.

2. CSS Selectors

CSS Selectors

```
/* Element Selector */
p { color: black; }

/* Class Selector (.) */
.card { background: white; }

/* ID Selector (#) */
#header { height: 80px; }

/* Group Selector */
h1, h2, h3 { font-family: Arial; }

/* Descendant Selector */
nav a { text-decoration: none; }

/* Child Selector (>) */
ul > li { color: gray; }

/* Pseudo-class */
a:hover { color: red; }
li:nth-child(2) { background: yellow; }
input:focus { border: 2px solid blue; }

/* Pseudo-element */
p::first-line { font-weight: bold; }
h1::before { content: "→ "; }

/* Attribute Selector */
```

```
input[type="email"] { border: 1px solid green; }

/* Universal Selector */
* { box-sizing: border-box; margin: 0; padding: 0; }
```

3. Colors & Units

```
Colors & Units
/* Colors */
color: red; /* Named color */
color: #FF5733; /* Hex */
color: rgb(255, 87, 51); /* RGB */
color: rgba(255, 87, 51, 0.5); /* RGBA (with opacity) */
color: hsl(9, 100%, 60%); /* HSL */

/* Units */
font-size: 16px; /* Pixels (absolute) */
font-size: 1.5em; /* Relative to parent font-size */
font-size: 1.5rem; /* Relative to root font-size */
width: 50%; /* Percentage of parent */
height: 100vh; /* Viewport height */
width: 100vw; /* Viewport width */
```

4. The Box Model

Every HTML element is a box. The box model defines the space an element takes up.

```
Box Model
div {
  width: 300px; /* Content width */
  padding: 20px; /* Space INSIDE border */
  border: 2px solid; /* Border around padding */
  margin: 30px; /* Space OUTSIDE border */
}

/* Individual sides */
padding: 10px 20px 10px 20px; /* top right bottom left */
margin: 10px 20px; /* top-bottom left-right */

/* Fix total sizing (recommended) */
* { box-sizing: border-box; }
```

NOTE: With box-sizing: border-box, width includes padding and border, preventing layout surprises.

5. Typography & Text

```
Typography
p {
  font-family: 'Arial', sans-serif;
  font-size: 18px;
  font-weight: bold; /* normal | bold | 100-900 */
  font-style: italic;
  line-height: 1.6; /* Spacing between lines */
  letter-spacing: 1px;
```

```

text-align: center; /* left | right | justify */
text-decoration: underline;
text-transform: uppercase; /* lowercase | capitalize */
color: #333;
}

/* Google Fonts (add in <head>) */
<link href="https://fonts.googleapis.com/css2?family=Roboto&display=swap"
rel="stylesheet">
body { font-family: 'Roboto', sans-serif; }

```

6. Backgrounds

Backgrounds

```

div {
  background-color: #f0f0f0;
  background-image: url('bg.jpg');
  background-size: cover; /* contain | auto | 100% */
  background-position: center;
  background-repeat: no-repeat;
  background-attachment: fixed; /* Parallax effect */

  /* Shorthand */
  background: #fff url('bg.jpg') no-repeat center / cover;

  /* Gradient */
  background: linear-gradient(to right, #FF5733, #6C63FF);
  background: radial-gradient(circle, #fff, #000);
}

```

7. Display & Positioning

Display Property

Display Values

```

display: block; /* Takes full width, new line */
display: inline; /* Only takes content width */
display: inline-block; /* Inline but respects width/height */
display: none; /* Hides element (removed from flow) */
display: flex; /* Flexbox layout */
display: grid; /* Grid layout */

```

Position Property

Positioning

```

/* Static (default) */
position: static;

/* Relative – moves from its normal position */
position: relative;
top: 10px; left: 20px;

/* Absolute – positioned relative to nearest positioned parent */

```

```

position: absolute;
top: 0; right: 0;

/* Fixed – stays fixed on screen when scrolling */
position: fixed;
top: 0; width: 100%;

/* Sticky – sticks when scrolled to */
position: sticky;
top: 0;

```

8. Flexbox (Intermediate)

Flexbox is a layout method for arranging items in a row or column with powerful alignment options.

```

Flexbox
/* Parent (Flex Container) */
.container {
  display: flex;
  flex-direction: row;          /* row | column | row-reverse */
  justify-content: center;     /* Main axis: flex-start | flex-end | space-between
| space-around */
  align-items: center;         /* Cross axis: stretch | flex-start | flex-end */
  flex-wrap: wrap;            /* Allow items to wrap */
  gap: 16px;                  /* Space between items */
}

/* Child (Flex Items) */
.item {
  flex: 1;                    /* Grow to fill space equally */
  flex-grow: 2;              /* Grow twice as much as others */
  flex-shrink: 0;           /* Don't shrink */
  flex-basis: 200px;        /* Starting size */
  align-self: flex-end;     /* Override align-items for this item */
}

```

9. CSS Grid (Intermediate)

CSS Grid is a 2D layout system — it works with both rows and columns simultaneously.

```

CSS Grid
.grid {
  display: grid;
  grid-template-columns: 1fr 2fr 1fr; /* 3 columns */
  grid-template-rows: auto;
  gap: 20px;

  /* Shorthand */
  grid-template-columns: repeat(3, 1fr);

  /* Responsive with auto-fill */
  grid-template-columns: repeat(auto-fill, minmax(200px, 1fr));
}

/* Placing items */
.item {
  grid-column: 1 / 3; /* Span from column 1 to 3 */
}

```

```
grid-row: 1 / 2;
}
```

10. Responsive Design & Media Queries

Media Queries

```
/* Mobile first approach */
body { font-size: 14px; }

/* Tablet: 768px and up */
@media (min-width: 768px) {
  body { font-size: 16px; }
}

/* Desktop: 1024px and up */
@media (min-width: 1024px) {
  body { font-size: 18px; }
  .container { max-width: 1200px; margin: 0 auto; }
}

/* Max-width (desktop first) */
@media (max-width: 600px) {
  nav { flex-direction: column; }
}
```

11. CSS Variables & Transitions

Variables & Transitions

```
/* Define CSS Variables */
:root {
  --primary: #6C63FF;
  --secondary: #FF6584;
  --font-size: 16px;
}

/* Use them */
button {
  background: var(--primary);
  font-size: var(--font-size);
}

/* Transitions (smooth changes) */
button {
  background: var(--primary);
  transition: background 0.3s ease, transform 0.2s;
}
button:hover {
  background: var(--secondary);
  transform: scale(1.05);
}

/* Keyframe Animations */
@keyframes fadeIn {
  from { opacity: 0; transform: translateY(20px); }
  to { opacity: 1; transform: translateY(0); }
}
```

```
.card { animation: fadeIn 0.5s ease forwards; }
```