

CSS BASICS: A Beginner's Guide

Table of Contents.

1. What is CSS?
2. How CSS Works
3. CSS Syntax and Selectors
4. Colors, Units, and Fonts
5. Box Model
6. Positioning and Display
7. Flexbox and Grid Basics
8. Responsive Design
9. Best Practices
10. Helpful Resources

1. What is CSS?

CSS stands for Cascading Style Sheets. It describes how HTML elements should be displayed. CSS separates content (HTML) from presentation (CSS), making websites easier to maintain.

2. How CSS Works

CSS can be added in three ways:

- Inline: `<p style="color: red;">Text</p>`
- Internal: `<style>p { color: red; }</style>`
- External: `<link rel="stylesheet" href="styles.css">`

3. CSS Syntax and Selectors

Syntax: `selector { property: value; }`

Example:

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

Common Selectors:

- Element: `p {}`

CSS BASICS: A Beginner's Guide

- Class: `.my-class {}`
- ID: `#my-id {}`
- Group: `h1, h2 {}`
- Descendant: `div p {}`

4. Colors, Units, and Fonts

Colors: names, hex, rgb, rgba, hsl

Examples:

`color: red;`

`color: #ff0000;`

`color: rgb(255, 0, 0);`

Units: px, em, rem, %, vh, vw

Fonts:

`font-family: Arial, sans-serif;`

`font-size: 1.2rem;`

`font-weight: bold;`

`line-height: 1.5;`

5. Box Model

Every HTML element is a box made of:

- Content
- Padding
- Border
- Margin

Example:

```
.box {  
  width: 200px;
```

CSS BASICS: A Beginner's Guide

```
padding: 20px;  
border: 2px solid black;  
margin: 10px;  
}
```

6. Positioning and Display

Display types: block, inline, inline-block, none

Position values:

- static (default)
- relative
- absolute
- fixed
- sticky

Example:

```
.absolute-box {  
  position: absolute;  
  top: 50px;  
  left: 100px;  
}
```

7. Flexbox and Grid Basics

Flexbox for 1D layout:

```
.container {  
  display: flex;  
  justify-content: center;  
  align-items: center;  
}
```

CSS BASICS: A Beginner's Guide

Grid for 2D layout:

```
.container {  
  display: grid;  
  grid-template-columns: repeat(3, 1fr);  
  gap: 20px;  
}
```

8. Responsive Design

Use media queries to adapt to screen sizes.

Example:

```
@media (max-width: 768px) {  
  body {  
    background: lightgray;  
  }  
}
```

Use relative units like %, em, rem.

9. Best Practices

- Use external CSS files
- Keep selectors clear and short
- Use classes for styling
- Use shorthand properties
- Validate with W3C CSS Validator

10. Helpful Resources

MDN CSS: <https://developer.mozilla.org/en-US/docs/Web/CSS>

W3Schools CSS: <https://www.w3schools.com/css/>

CSS BASICS: A Beginner's Guide

FreeCodeCamp CSS: <https://www.freecodecamp.org/learn/>