

Cover Page

Course Title: CSS – Styling the Web

Prepared By: AetherCode Team

Website: www.aethercode.com

"Code. Notes. Clarity."

Table of Contents

1. [Introduction](#)
 2. [CSS Syntax and Selectors](#)
 3. [Types of CSS](#)
 4. [Colors and Units](#)
 5. [Box Model](#)
 6. [Margins, Padding, and Borders](#)
 7. [Fonts and Text Styling](#)
 8. [Backgrounds](#)
 9. [Flexbox](#)
 10. [Grid Layout](#)
 11. [Positioning](#)
 12. [Pseudo-classes and Elements](#)
 13. [Transitions and Animations](#)
 14. [Media Queries and Responsiveness](#)
 15. [Best Practices](#)
 16. [Summary](#)
-

Introduction

CSS (Cascading Style Sheets) is used to style HTML elements, controlling layout, colors, fonts, and responsiveness.

CSS Syntax and Selectors

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

- Example:

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

- Types: universal (*), element (h1), class (.name), ID (#id), attribute, descendant, etc.

Types of CSS

1. **Inline CSS** – Inside the tag: `<h1 style="color:red">`
 2. **Internal CSS** – Inside `<style>` in `<head>`
 3. **External CSS** – Linked via `<link rel="stylesheet">`
-



Colors and Units

- **Colors:** red, #ff0000, rgb(255,0,0), hsl(0, 100%, 50%)
 - **Units:** px, %, em, rem, vw, vh
-

Box Model

Each HTML element is a box made of: - **Content** - **Padding** - **Border** - **Margin**

```
.box {  
  padding: 10px;  
  border: 1px solid #000;  
  margin: 20px;  
}
```

Margins, Padding, and Borders

```
div {  
  margin: 10px;  
  padding: 15px;  
  border: 2px dashed blue;  
}
```

Fonts and Text Styling

```
body {  
  font-family: 'Lato', sans-serif;  
  font-size: 16px;  
  color: #333;  
  text-align: center;  
  text-transform: uppercase;  
}
```



Backgrounds

```
body {  
  background-color: #f0f0f0;  
  background-image: url('bg.jpg');  
  background-repeat: no-repeat;  
  background-size: cover;  
}
```

Flexbox

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

- Flexible layout for 1D arrangements (row/column)

Grid Layout

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

- 2D layout system for complex UI

Positioning

- static, relative, absolute, fixed, sticky

```
.box {  
  position: absolute;  
  top: 20px;  
  left: 50px;  
}
```

Pseudo-classes and Elements

- :hover, :focus, :nth-child()
- ::before, ::after

```
a:hover {  
  color: red;  
}
```

Transitions and Animations

```
.box {  
  transition: all 0.3s ease;  
}  
  
@keyframes slide {  
  0% { left: 0; }  
  100% { left: 100px; }  
}
```

Media Queries and Responsiveness

```
@media (max-width: 768px) {  
  body {  
    font-size: 14px;  
  }  
}
```

- Helps in making responsive websites

Best Practices

- Use external stylesheets
 - Follow naming conventions (BEM)
 - Avoid inline styles
 - Use shorthand properties
 - Use variables with preprocessors like SASS
-

Summary

CSS makes web pages visually appealing and responsive. Combined with HTML and JS, it's essential for front-end development.

Next: Add interactivity with JavaScript.

© **AetherCode** – Code. Notes. Clarity.