□ CSS Complete Introduction - Comprehensive Lecture Notes

□ What is CSS?

CSS = **Cascading Style Sheets**

- **Purpose**: Makes HTML websites beautiful and visually appealing
- **Current Problem**: Plain HTML looks boring - no colors, fonts, or styling
- **Solution**: CSS adds styling, layout, colors, and visual design to web pages

□ The 3 Ways to Apply CSS

1. □ Inline CSS

```
<h1 style="background-color: aqua; font-size: 30px;">Gym Website</h1>
```

How it works:

- Add `style` attribute directly to HTML element
- Write CSS properties inside the style attribute
- Applies only to that specific element

When to use: Quick testing and small changes only

2. □ Internal CSS

```
<head>
    <style>
        h1 {
            background-color: blue;
            font-size: 100px;
        }
        h2 {
            background-color: antiquewhite;
        }
    </style>
</head>
```

How it works:

- Write CSS inside `<style>` tag in the `<head>` section

- Select elements first, then apply properties
- Applies to all matching elements in that HTML page only

3. ⭑ External CSS (Most Popular!)

```
<!-- In HTML file -->
<head>
    <link rel="stylesheet" href="style.css">
</head>
```

```
/* In style.css file */
h1 {
    background-color: pink;
}
h2 {
    background-color: aquamarine;
}
```

How it works:

- Create separate `.css` file
- Link it to HTML using `<link>` tag
- Write all CSS in the external file

▯ Why External CSS is BEST:

- **Reusable**: Same CSS file can be used for multiple HTML pages
- **Clean Code**: Keeps HTML and CSS separate
- **Maintainable**: Easy to manage and update styles
- **Professional**: Industry standard approach

▯ CSS Selectors - The Heart of CSS

1. Element/Tag Selector

```
h1 {
    background-color: pink;
}
ul {
    background-color: aquamarine;
}
```

- **Targets**: All elements of that tag type
- **Usage**: When you want to style ALL h1s, h2s, etc.

2. ID Selector

```
<h2 id="first">Exercise You Have To Do</h2>
```

```
#first {
    background-color: blue;
}
```

- **Symbol**: `#` (hash)
- **Rule**: IDs must be unique in entire HTML page
- **Usage**: When you want to style ONE specific element

3. Class Selector

```
<h1 class="third">Gym Website</h1>
<ul class="third">Exercise list</ul>
```

```
.third {
    background-color: green;
}
```

- **Symbol**: `.` (dot)
- **Rule**: Classes can be repeated/reused
- **Usage**: For grouping elements with same styling

4. Group-Based Selection

```
h1, .fourth, #first {
    background-color: aqua;
}
```

- **Symbol**: `,` (comma) separates selectors
- **Usage**: Apply same styles to multiple different elements
- **Very Common**: You'll use this A LOT!

CSS Colors - Complete Guide

1. Named Colors

```
background-color: red;
background-color: blue;
background-color: aquamarine;
```

- Simple color names
- Limited options but easy to remember

2. RGB Colors □□□

```
background-color: rgb(255, 0, 0);     /* Pure Red */
background-color: rgb(0, 255, 0);     /* Pure Green */
background-color: rgb(0, 0, 255);     /* Pure Blue */
background-color: rgb(127, 140, 209); /* Custom Mix */
```

□ RGB Concept:

- **R**ed + **G**reen + **B**lue combination
- Each value: **0 to 255**
- Memory usage: **8 bits per color = 24 bits total = 3 bytes**
- **Why 8 bits?** 2^8 = 256 possible values (0-255)

3. RGBA Colors (RGB + Transparency)

```
background-color: rgba(159, 19, 161, 0.5);
```

- **A** = Alpha (transparency)
- **Range**: 0 (fully transparent) to 1 (fully opaque)
- **0.5** = 50% transparent

4. HSL Colors

```
background-color: hsl(110, 50%, 30%);
```

- **H**ue: 0-360 degrees (color wheel)
  - 0 = Red, 120 = Green, 240 = Blue
- **S**aturation: 0-100% (color intensity)
- **L**ightness: 0-100% (brightness)
  - 0% = Black, 50% = Normal, 100% = White

5. HSLA Colors (HSL + Transparency)

```
background-color: hsla(110, 50%, 30%, 0.6);
```

6. Hexadecimal Colors □

```
background-color: #FF0000;  /* Red */
background-color: #00FF00;  /* Green */
background-color: #0000FF;  /* Blue */
```

- **Format**: `#RRGGBB`
- First 2 digits = Red, Next 2 = Green, Last 2 = Blue
- Each pair represents 0-255 in hexadecimal

□ Important CSS Concepts

CSS Rule Anatomy

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

Font Properties

```
h1 {
    font-size: 30px;
    font-family: Arial;
    font-weight: bold;
    font-style: italic;
}
```

Multiple Properties

```
h1 {
    background-color: blue;
    font-size: 100px;
    color: white;
    border: 2px solid black;
}
```

⚡ Pro Tips & Best Practices

✅ DO:

- **Always use External CSS** for real projects
- Use **classes for grouping** similar elements
- Use **IDs for unique** elements only
- Use **RGB/RGBA** for precise colors
- Keep CSS **organized and clean**

❌ DON'T:

- Avoid inline CSS for large projects (makes code messy)
- Don't use internal CSS for multi-page websites
- Don't repeat IDs in same page
- Don't mix different color formats unnecessarily

Key Memory Points

1. **External CSS** is the professional choice - **most used!**
2. **Classes** can repeat, **IDs** are unique
3. **RGB values**: 0-255 for each color
4. **Transparency**: 0 (invisible) to 1 (solid)
5. **Group selectors** with commas for efficiency
6. **Selectors** = way to target HTML elements
7. **Properties** = what you want to change
8. **Values** = how you want to change it

Practical Example

```
<!DOCTYPE html>
<html>
<head>
    <link rel="stylesheet" href="style.css">
</head>
<body>
    <h1 id="main-title" class="header">Gym Website</h1>
    <h2 class="section-title">Exercises</h2>
    <ul class="exercise-list">
        <li>Biceps</li>
        <li>Chest</li>
        <li>Legs</li>
    </ul>
```

```
</body>
</html>
```

```
/* style.css */
#main-title {
    background-color: rgba(255, 100, 150, 0.8);
    font-size: 48px;
}

.section-title {
    background-color: #00FF7F;
    color: white;
}

.exercise-list {
    background-color: hsl(200, 70%, 80%);
}

h1, .section-title {
    font-family: Arial, sans-serif;
    text-align: center;
}
```

▯ What's Next?

After mastering these CSS basics, you'll learn:

- **Advanced Selectors** (child, descendant, pseudo)
- **CSS Box Model** (margin, padding, border)
- **Flexbox & Grid** (modern layouts)
- **Responsive Design** (mobile-friendly)
- **Animations & Transitions**
- **CSS Frameworks** (Bootstrap, Tailwind)

▯ Quick Reference

Selector Type	Symbol	Example
Element	none	`h1 { }`
ID	#	`#header { }`
Class	.	`.menu { }`
Group	,	`h1, .menu { }`
Color Type	Format	Example
------------	--------	---------

Selector Type	Symbol	Example
Named	name	`red`
RGB	rgb(r,g,b)	`rgb(255,0,0)`
RGBA	rgba(r,g,b,a)	`rgba(255,0,0,0.5)`
HSL	hsl(h,s%,l%)	`hsl(0,100%,50%)`
Hex	#rrggbb	`#FF0000`

□ **Remember**: Practice makes perfect! Start with simple styling and gradually build complex designs.