

# HTML & CSS Comprehensive Guide (ISS LAB7)

## Table of Contents

1. HTML Fundamentals
2. CSS Essentials
3. Advanced CSS Techniques
4. Best Practices

## HTML Fundamentals

### What is HTML?

**HyperText Markup Language** (HTML) is the standard markup language for creating web pages:

- **Not a programming language** - No logic/conditional operations
- **Markup language** - Uses tags to structure content
- **Parsed by browsers** - Converts HTML into visible/interactive web pages
- **Backbone of web content** - Works with CSS (styling) and JavaScript (functionality)

### Core Concepts

#### Elements and Tags

- **Elements:** Building blocks of HTML (e.g., paragraphs, headings, images)
- **Tags:** Syntax used to define elements (`<tagname>Content</tagname>`)
- **Attributes:** Additional information within opening tags (e.g., `href`, `src`, `id`)

```
<!-- Example element with attributes -->  
<a href="https://example.com" target="_blank">Visit Example</a>
```

### DOCTYPE Declaration

```
<!DOCTYPE html>
```

- **Purpose:** Declares document type and version (HTML5)
- **Importance:** Ensures consistent rendering across browsers
- **History:** Replaced older versions like XHTML 1.0 Transitional

### Common Elements

Element	Type	Description	Example
<code>&lt;html&gt;</code>	Root	Contains all HTML elements	<code>&lt;html&gt;...&lt;/html&gt;</code>
<code>&lt;head&gt;</code>	Metadata	Holds title, meta tags, and links	<code>&lt;head&gt;&lt;title&gt;My Page&lt;/title&gt;&lt;/head&gt;</code>
<code>&lt;body&gt;</code>	Content	Visible page content	<code>&lt;body&gt;&lt;h1&gt;Hello World&lt;/h1&gt;&lt;/body&gt;</code>
<code>&lt;h1&gt;-&lt;h6&gt;</code>	Headings	Title hierarchy (h1 = most important)	<code>&lt;h1&gt;Main Title&lt;/h1&gt;</code>
<code>&lt;p&gt;</code>	Text	Paragraph container	<code>&lt;p&gt;This is text&lt;/p&gt;</code>
<code>&lt;a&gt;</code>	Hyperlink	Creates clickable links	<code>&lt;a href="/"&gt;Home&lt;/a&gt;</code>

Element	Type	Description	Example
<code>&lt;img&gt;</code>	Media	Embeds images (self-closing)	<code>&lt;img src="logo.jpg" alt="Logo"&gt;</code>
<code>&lt;div&gt;</code>	Container	Block-level grouping	<code>&lt;div class="container"&gt;...&lt;/div&gt;</code>
<code>&lt;span&gt;</code>	Inline	Inline-level grouping	<code>&lt;span class="highlight"&gt;Text&lt;/span&gt;</code>
<code>&lt;iframe&gt;</code>	Embedded Content	Embeds another HTML page	<code>&lt;iframe src="page.html"&gt;&lt;/iframe&gt;</code>
<code>&lt;br /&gt;</code>	Line Break	Inserts a line break	Text More text
<code>&lt;hr /&gt;</code>	Thematic Break	Adds a horizontal rule	<code>&lt;hr /&gt;</code>

## Comments

### What are comments?

Comments are human-readable notes added to code that are ignored by browsers. They help:

- Explain complex code sections
- Temporarily disable code snippets
- Improve team collaboration

### HTML Comment Syntax

```
<!-- This is a single-line comment -->
<!--
This is a multi-line comment
that spans multiple lines
-->
```

## Comments in CSS

### CSS Comment Syntax

```
/* Single-line comment */

/*
Multi-line comment
for longer explanations
*/
```

## Special Elements

### Forms

```
<form action="/submit" method="POST">
  <label for="username">Username:</label>
  <input type="text" id="username" name="username" required />

  <label>Password:</label>
```

```

<input type="password" name="password" />

<input type="checkbox" id="terms" name="terms" required />
<label for="terms">Accept Terms</label>

<button type="submit">Submit</button>
</form>

```

- **Input Types:** text, password, checkbox, radio, submit
- **Validation:** required attribute for mandatory fields
- **Form Data:** name attribute required for server processing

## Lists

```

<!-- Unordered list -->
<ul>
  <li>Item 1</li>
  <li>Item 2</li>
</ul>

<!-- Ordered list -->
<ol>
  <li>First item</li>
  <li>Second item</li>
</ol>

```

## Tables

```

<table>
  <tr>
    <th>Name</th>
    <th>Age</th>
  </tr>
  <tr>
    <td>John</td>
    <td>25</td>
  </tr>
</table>

```

- <tr>: Table row
- <th>: Header cell
- <td>: Data cell

## HTML Entities

Special characters represented with `&code;` syntax:

Entity	Character	Description
&nbsp;		Non-breaking space
&lt;	<	Less than
&gt;	>	Greater than
&amp;	&	Ampersand
&quot;	"	Quotation mark

# CSS Essentials

## What is CSS?

Cascading Style Sheets (CSS) controls:

- Layout (positioning and spacing)
- Visual styling (colors, fonts)
- Responsive behavior (mobile/desktop adaptation)
- Animations and transitions

## Core Concepts

### Selectors

Type	Syntax	Description	Example
Element	<code>p</code>	Targets all instances of element	<code>p { color: blue; }</code>
Class	<code>.classname</code>	Targets elements with class attribute	<code>.alert { color: red; }</code>
ID	<code>#idname</code>	Targets unique element by ID	<code>#header { font-size: 24px; }</code>
Attribute	<code>[type="text"]</code>	Targets elements with specific attribute	<code>input[type="text"] { border: 1px solid #ccc; }</code>
Pseudo-class	<code>:hover</code>	Targets element state	<code>a:hover { color: red; }</code>

### Specificity Hierarchy

1. **Inline styles** (`style="..."`) - Highest priority
2. **ID selectors** (`#id`)
3. **Class selectors** (`.class`), attribute selectors (`[type]`)
4. **Element selectors** (`p`, `div`) - Lowest priority

Override specificity with `!important` (use sparingly):

```
.override {  
  color: red !important;  
}
```

### Box Model

- **Content**: Actual element content
- **Padding**: Space between content and border
- **Border**: Visible border around padding
- **Margin**: Space outside the border

```
.box {  
  width: 200px; /* Sets content area width to 200px */  
  padding: 20px; /* Adds 20px space inside element (total width becomes 240px without box-sizing) */  
  border: 5px solid black; /* Adds 5px solid black border around element */  
  margin: 10px; /* Creates 10px space outside the element */  
  box-sizing: border-box; /* Makes width include padding/border (total width remains 200px) */  
}
```

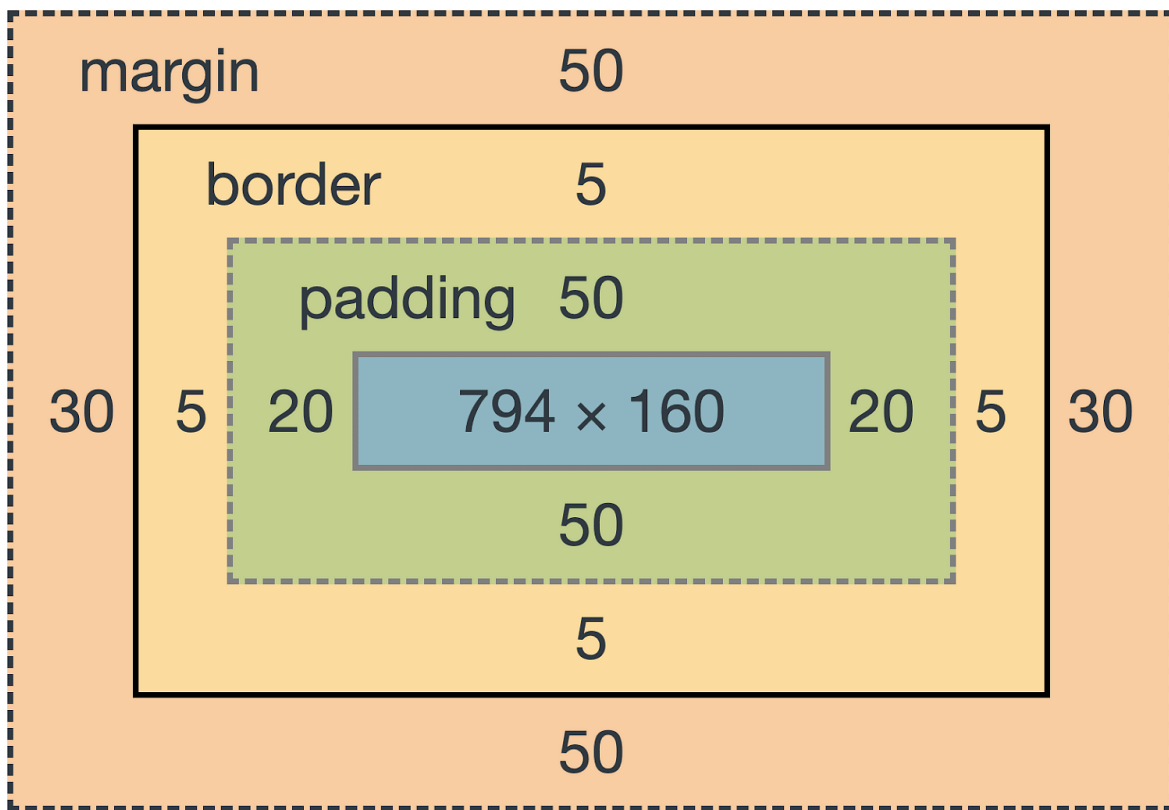


Figure 1: Box Model Diagram

## Positioning

Value	Description	Example
static	Default positioning (no special behavior)	position: static;
relative	Positioned relative to itself	position: relative; top: 10px;
absolute	Positioned relative to nearest positioned ancestor	position: absolute; right: 0;
fixed	Fixed position relative to viewport	position: fixed; bottom: 0;
sticky	Toggles between relative and fixed	position: sticky; top: 0;

## Advanced CSS Techniques

### Flexbox Layout

```
.container {  
  display: flex; /* Enables Flexbox container */  
  justify-content: space-between; /* Distributes items evenly along main axis */  
  align-items: center; /* Vertically centers items */  
  flex-direction: row; /* Horizontal main axis (default) */  
  flex-wrap: wrap; /* Allows items to wrap to new lines */  
}  
  
.item {  
  flex: 1 1 200px; /* flex-grow: 1, flex-shrink: 1, flex-basis: 200px */  
  order: 2; /* Changes visual order of items */  
}
```

### CSS Grid

```
/* CSS Grid */  
.grid {  
  display: grid; /* Enables Grid layout */  
  grid-template-columns: 1fr 2fr 1fr; /* Three columns with ratio 1:2:1 */  
  grid-template-rows: auto 300px; /* First row auto-height, second fixed 300px */  
  grid-gap: 20px; /* 20px spacing between grid items */  
  grid-template-areas:  
    "header header" /* Header spans two columns */  
    "sidebar content" /* Sidebar and content columns */  
    "footer footer"; /* Footer spans two columns */  
}  
  
.sidebar {  
  grid-area: sidebar; /* Places element in 'sidebar' grid area */  
}
```

### Responsive Design

```
/* Responsive Design - Mobile-first approach */  
.container {  
  padding: 20px; /* Base mobile padding */  
}  
  
@media (min-width: 768px) {  
  /* Tablet breakpoint */  
}
```

```

.container {
  padding: 40px; /* Increased padding for larger screens */
  max-width: 1200px; /* Limits container width */
}

@media (min-width: 1024px) {
  /* Desktop breakpoint */
  .container {
    display: grid; /* Switches to grid layout */
    grid-template-columns: 3fr 1fr; /* 3:1 column ratio */
  }
}

```

## Transitions & Animations

```

.button {
  transition: transform 0.3s ease; /* Smooth transform transition */
}

.button:hover {
  transform: scale(1.1); /* Scales element on hover */
}

@keyframes slideIn {
  /* Defines slide animation */
  0% {
    transform: translateX(-100%);
  } /* Starts off-screen left */
  100% {
    transform: translateX(0);
  } /* Ends at original position */
}

.animated {
  animation: slideIn 1s ease-out; /* Applies slide animation */
}

```

---

## Best Practices

### File Structure

```

project/
|-- index.html      # Entry point
|-- css/
|   |-- styles.css  # Main stylesheet
|-- images/         # Image assets
|-- js/
|   |-- main.js     # JavaScript files

```

### Accessibility

- **Semantic HTML:** Use `<nav>`, `<main>`, `<article>`
- **Alt text:** Describe images with `alt="Description"`

- **ARIA attributes:** Use `aria-label` for screen readers

## Performance

- **Minify files:** Reduce CSS/JS file sizes
- **Optimize images:** Use WebP format and compression
- **Lazy loading:** `loading="lazy"` for images
- **Reduce HTTP requests:** Combine files where possible

## Project Example: Responsive Blog Layout

### HTML Structure:

```

<!--
DOCTYPE declaration: Specifies document type and version (HTML5)
Ensures consistent rendering across browsers
-->
<!DOCTYPE html>

<!--
Root element: Contains all HTML content
lang="en" specifies document language (English)
-->
<html lang="en">
  <head>
    <!--
    Viewport meta tag: Controls layout on mobile devices
    width=device-width: Matches screen width
    initial-scale=1.0: Sets default zoom level
    -->
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />

    <!--
    External stylesheet link: Connects to CSS file for styling
    Best practice for separating content and presentation
    -->
    <link rel="stylesheet" href="styles.css" />
  </head>
  <body>
    <!--
    Header section: Contains website branding/navigation
    Semantic element for introductory content
    -->
    <header>
      <!-- Main heading (h1): Primary page title -->
      <h1>My Blog</h1>

      <!--
      Navigation container: Holds site links
      Semantic <nav> element for accessibility
      -->
      <nav>
        <ul>
          <!-- List items with anchor tags for navigation links -->
          <li><a href="#">Home</a></li>

```



```

        <li><a href="#">About</a></li>
    </ul>
</nav>
</header>

<!--
Main content container: Uses class for styling
Semantic <main> element for primary content
-->
<main class="container">
    <!--
    Blog posts section: Groups article elements
    Semantic <section> for thematic content grouping
    -->
    <section class="posts">
        <!--
        Individual blog post: Self-contained content
        Semantic <article> element for standalone content
        -->
        <article class="post">
            <!-- Post title (h2): Secondary heading -->
            <h2>Post Title</h2>

            <!--
            Metadata paragraph: Contains publication date
            <time> element for machine-readable dates
            -->
            <p class="meta">Posted on <time>2023-10-01</time></p>

            <!-- Excerpt content: Brief post summary -->
            <p>Content excerpt...</p>

            <!-- Read more link: Uses class for styling -->
            <a href="#" class="read-more">Read more</a>
        </article>
    </section>

    <!--
    Sidebar section: Secondary content area
    Semantic <aside> element for tangential content
    -->
    <aside class="sidebar">
        <!-- Sidebar heading (h3): Section title -->
        <h3>Categories</h3>

        <!-- Unordered list of category links -->
        <ul>
            <li><a href="#">Web Dev</a></li>
            <li><a href="#">CSS</a></li>
        </ul>
    </aside>
</main>

<!--

```

```

    Footer section: Contains copyright information
    Semantic <footer> element for closing content
-->
<footer>
  <p>&copy; 2023 My Blog</p>
</footer>
</body>
</html>

```

## CSS Styling:

```

/*
CSS Variables (Custom Properties)
Define reusable colors for theme consistency
*/
:root {
  --primary: #2c3e50; /* Primary brand color (dark blue) */
  --secondary: #3498db; /* Secondary accent color (bright blue) */
}

/*
Body styling
Set base typography and remove default margins/padding
*/
body {
  font-family: "Georgia", serif; /* Fallback to serif if Georgia unavailable */
  line-height: 1.7; /* Improved readability with 1.7em line height */
  margin: 0; /* Remove default browser margin */
  padding: 0; /* Remove default browser padding */
}

/*
Responsive grid container
Mobile-first grid layout with 1 column base
*/
.container {
  display: grid; /* Enable CSS Grid layout */
  grid-template-columns: 1fr; /* Single flexible column */
  gap: 2rem; /* 32px spacing between grid items */
  padding: 20px; /* Base padding for mobile view */
}

/*
Tablet breakpoint (768px+)
Switches to 2-column layout with 3:1 ratio
*/
@media (min-width: 768px) {
  .container {
    grid-template-columns: 3fr 1fr; /* Main content (75%) + Sidebar (25%) */
  }
}

/*
Blog post styling
Card-like container with subtle elevation

```

```

/*
.post {
  background: #fff; /* White background */
  border-radius: 8px; /* Rounded corners */
  box-shadow: 0 2px 4px rgba(0, 0, 0, 0.1); /* Soft shadow for depth */
  padding: 20px; /* Internal content spacing */
}

/*
Read more link styling
Uses secondary color with hover state
*/
.read-more {
  color: var(--secondary); /* Use CSS variable for color */
  text-decoration: none; /* Remove underline */
  font-weight: bold; /* Emphasize link text */
}

.read-more:hover {
  text-decoration: underline; /* Add underline on hover */
}

/*
Sidebar styling
Uses primary color with contrasting text
*/
.sidebar {
  background: var(--primary); /* Brand color background */
  color: white; /* White text for contrast */
  padding: 20px; /* Match post padding */
  border-radius: 8px; /* Consistent with post styling */
}

/*
Footer styling
Centered content with brand color background
*/
footer {
  text-align: center; /* Center align content */
  padding: 1rem; /* Vertical spacing */
  background: var(--primary); /* Brand color background */
  color: white; /* White text */
  margin-top: 2rem; /* Space above footer */
}

```

## Conclusion

### References

- FULL HTML TUTORIAL
- HTML TAGS GEEKSFORGEEKS
- HTML TAGS W3SCHOOLS
- CSS STYLES TUTORIAL
- CSS PROPERTIES W3SCHOOLS

For best results:

1. Practice each example in a code editor
2. Use browser developer tools for debugging
3. Validate code using W3C Validator
4. Build small projects to reinforce concepts

[Back to top](#)