

Complete HTML & CSS Notes - Scrimba Course Curriculum

Course Overview

This comprehensive guide covers all HTML and CSS topics from Scrimba's renowned web development course^[1] ^[2]. The course features **106 interactive lessons** spread across **6 modules** with over **75 coding challenges** and **5 hands-on projects**^[1] ^[2]. Designed for absolute beginners, this curriculum takes you from zero knowledge to building and deploying professional websites^[2].

Course Projects

- [Google.com Clone](#) - Master CSS fundamentals^[2]
- **Digital Business Card** - Learn advanced layouts^[2]
- **Space Exploration Landing Page** - Modern UI techniques^[2]
- **Birthday GIFt Site** - Interactive elements and animations^[2]
- **Hometown Homepage** - Solo project to showcase skills^[2]

Module 1: Build & Deploy Your First Website (HTML Fundamentals)

Duration: 16 lessons, 1 hour^[2]

What is HTML?

HTML (HyperText Markup Language) is the standard markup language for creating web pages^[3] ^[4]. It defines the structure and content of web pages using elements and tags^[5] ^[6].

Basic HTML Structure:

```
<!DOCTYPE html>
<html>
<head>
  <title>My First Web Page</title>
</head>
<body>
  <h1>Welcome to My Website</h1>
  <p>This is my first paragraph.</p>
</body>
</html>
```

HTML Tags and Elements

Tags are HTML components enclosed in angle brackets that define content structure^{[5] [7]}. **Elements** consist of opening tags, content, and closing tags^{[6] [7]}.

Common HTML Tags:^{[8] [6]}

- **Headings:** <h1>, <h2>, <h3>, <h4>, <h5>, <h6>
- **Text:** <p>, , ,
- **Structure:** <div>, <header>, <main>, <footer>, <section>
- **Lists:** , ,
- **Links & Media:** <a>, , <video>, <audio>
- **Forms:** <form>, <input>, <textarea>, <button>

HTML Attributes

Attributes provide additional information about HTML elements and appear in the opening tag^{[9] [7]}. They follow the syntax: <tag attribute="value">content</tag>^{[10] [11]}.

Essential Attributes:^{[9] [8]}

- **Global:** id, class, style, title, lang
- **Links:** href, target, download
- **Images:** src, alt, width, height
- **Forms:** type, name, value, placeholder, required

Document Structure

Every HTML document follows a standard structure^[8]:

- <!DOCTYPE html> - Declares HTML5 document type^[7]
- <html> - Root element containing all content^[6]
- <head> - Contains metadata, title, links to CSS/JS^[8]
- <body> - Contains visible page content^[6]

Module 2: Build a Google.com Clone (CSS Fundamentals)

Duration: 26 lessons, 1 hour 20 min^[2]

What is CSS?

CSS (Cascading Style Sheets) controls the visual presentation of HTML elements^{[3] [4]}. It separates content from design, allowing you to style multiple pages with one stylesheet^[12].

CSS Syntax:^{[13] [14]}

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

CSS Selectors

Selectors target HTML elements for styling^[15] ^[14]. There are five main categories^[14]:

Basic Selectors:^[15] ^[16]

- **Element:** `p { color: blue; }` - Selects all `<p>` elements^[14]
- **Class:** `.className { font-size: 16px; }` - Selects elements with class^[14]
- **ID:** `#idName { background: yellow; }` - Selects element with specific ID^[14]
- **Universal:** `* { margin: 0; }` - Selects all elements^[15]

Attribute Selectors:^[17] ^[15]

- `[type="text"]` - Elements with specific attribute value^[17]
- `[href^="https"]` - Elements whose attribute starts with value^[17]
- `[class*="nav"]` - Elements whose attribute contains value^[17]

The CSS Box Model

Every HTML element is treated as a rectangular box with four components^[8]:

1. **Content** - The actual content (text, images)^[8]
2. **Padding** - Space between content and border^[8]
3. **Border** - Line around padding and content^[8]
4. **Margin** - Space outside the border^[8]

Box Model Properties:^[8]

```
.box {  
  width: 300px;  
  height: 200px;  
  padding: 20px;  
  border: 2px solid black;  
  margin: 10px;  
}
```

CSS Layout Properties

Display Property: ^[8]

- `block` - Takes full width, stacks vertically
- `inline` - Takes only necessary width, flows horizontally
- `inline-block` - Combines features of both
- `flex` - Creates flexible layouts
- `grid` - Creates grid-based layouts

Positioning: ^[8]

- `static` - Default positioning
- `relative` - Positioned relative to normal position
- `absolute` - Positioned relative to nearest positioned ancestor
- `fixed` - Positioned relative to viewport

Module 3: Build a Digital Business Card (Intermediate CSS)

Duration: 21 lessons, 1 hour 5 min^[2]

Advanced Flexbox

Flexbox is a CSS layout method for arranging items in rows or columns^[8]. It provides powerful alignment and distribution capabilities^[18].

Flex Container Properties: ^[8]

```
.container {
  display: flex;
  flex-direction: row; /* or column */
  justify-content: center; /* main axis alignment */
  align-items: center; /* cross axis alignment */
  flex-wrap: wrap; /* allows wrapping */
}
```

Flex Item Properties: ^[8]

```
.item {
  flex-grow: 1; /* how much to grow */
  flex-shrink: 0; /* how much to shrink */
  flex-basis: 200px; /* initial size */
}
```

CSS Colors and Typography

Color Systems:^[8]

- **Named:** red, blue, green
- **Hex:** #FF0000 (red), #00FF00 (green)
- **RGB:** rgb(255, 0, 0) for red
- **RGBA:** rgba(255, 0, 0, 0.5) for semi-transparent red
- **HSL:** hsl(0, 100%, 50%) for red

Typography Properties:^[8]

```
.text {  
  font-family: Arial, sans-serif;  
  font-size: 16px;  
  font-weight: bold;  
  text-align: center;  
  line-height: 1.5;  
}
```

Module 4: Build a Space Exploration Site (Modern CSS)

Duration: 15 lessons, 44 min^[2]

Background Images and Styling

Background Properties:^[8]

```
.hero {  
  background-image: url('space.jpg');  
  background-size: cover;  
  background-position: center;  
  background-repeat: no-repeat;  
  background-attachment: fixed;  
}
```

CSS Gradients:^[8]

```
.gradient {  
  background: linear-gradient(to right, #ff6b6b, #4ecdc4);  
  background: radial-gradient(circle, #ff6b6b, #4ecdc4);  
}
```

Google Fonts Integration

Method 1 - HTML Link: ^[8]

```
<link href="https://fonts.googleapis.com/css2?family=Roboto:wght@300;400;700&display=swap"
```

Method 2 - CSS Import: ^[8]

```
@import url('https://fonts.googleapis.com/css2?family=Roboto:wght@300;400;700&display=swap');

body {
  font-family: 'Roboto', sans-serif;
}
```

Text Effects

Text Shadows: ^[8]

```
.text-shadow {
  text-shadow: 2px 2px 4px rgba(0,0,0,0.5);
  text-shadow: 1px 1px 2px #000, 0 0 1em #ff0000;
}
```

Module 5: Build a Birthday GIFT Site (Advanced CSS)

Duration: 23 lessons, 1 hour 14 min ^[2]

Advanced Flexbox Techniques

Flex Direction and Alignment: ^[8]

```
.card-container {
  display: flex;
  flex-direction: column;
  align-items: center;
  justify-content: space-between;
  gap: 20px;
}

.card-row {
  display: flex;
  flex-direction: row;
  flex-wrap: wrap;
  justify-content: space-around;
}
```

Hover Effects and Pseudo-classes

Interactive Elements:^[8]

```
.card {  
    transition: transform 0.3s ease, box-shadow 0.3s ease;  
}  
  
.card:hover {  
    transform: scale(1.05);  
    box-shadow: 0 10px 20px rgba(0,0,0,0.2);  
    cursor: pointer;  
}  
  
.button:active {  
    transform: translateY(2px);  
}
```

CSS Animations

Basic Animations:^[8]

```
@keyframes fadeIn {  
    from { opacity: 0; }  
    to { opacity: 1; }  
}  
  
.fade-in {  
    animation: fadeIn 1s ease-in-out;  
}  
  
@keyframes bounce {  
    0%, 20%, 53%, 80%, 100% {  
        transform: translate3d(0,0,0);  
    }  
    40%, 43% {  
        transform: translate3d(0, -20px, 0);  
    }  
    70% {  
        transform: translate3d(0, -10px, 0);  
    }  
    90% {  
        transform: translate3d(0, -4px, 0);  
    }  
}
```

Module 6: Web Development Essentials & Solo Projects

Duration: 5 lessons, 14 min + projects^[2]

Developer Tools and Workflow

Chrome Developer Tools:^[8]

- **Elements Tab** - Inspect and modify HTML/CSS
- **Console Tab** - View errors and run JavaScript
- **Network Tab** - Monitor resource loading
- **Sources Tab** - Debug JavaScript code

Version Control with Git:^[8]

```
git init
git add .
git commit -m "Initial commit"
git push origin main
```

Deployment with Netlify

Steps for Deployment:^[8]

1. Create account at netlify.com
2. Connect your GitHub repository
3. Configure build settings
4. Deploy and get live URL
5. Set up custom domain (optional)

Using Figma for Design

Key Figma Features:^[8]

- **Design System** - Consistent colors, fonts, components
- **Prototyping** - Interactive design previews
- **Developer Handoff** - CSS code generation
- **Collaboration** - Real-time design feedback

Responsive Design Principles

Mobile-First Approach

Viewport Meta Tag: ^[19]

```
<meta name="viewport" content="width=device-width, initial-scale=1">
```

Media Queries: ^[8]

```
/* Mobile styles (default) */
.container { width: 100%; }

/* Tablet styles */
@media (min-width: 768px) {
  .container { width: 750px; }
}

/* Desktop styles */
@media (min-width: 1024px) {
  .container { width: 1000px; }
}
```

Flexible Images and Media

```
img {
  max-width: 100%;
  height: auto;
}

video {
  width: 100%;
  height: auto;
}
```

Best Practices and Guidelines

HTML Best Practices ^[8] ^[19]

- Use semantic HTML elements (<header>, <nav>, <main>, <footer>)
- Include alt attributes for all images
- Use proper heading hierarchy (h1 → h2 → h3)
- Validate HTML markup regularly
- Keep code clean and properly indented
- Use meaningful class and ID names

CSS Best Practices^[8] ^[19]

- Use external stylesheets instead of inline styles
- Follow consistent naming conventions (BEM methodology)
- Group related styles together
- Use shorthand properties when possible
- Comment complex code sections
- Optimize for performance (minimize CSS files)

Development Workflow^[8] ^[19]

- Test in multiple browsers and devices
- Optimize images for web (use WebP format when possible)
- Use version control (Git) for all projects
- Validate and test for accessibility (WCAG guidelines)
- Keep learning new techniques and frameworks
- Build projects to practice skills

Advanced Topics for Further Learning

CSS Grid Layout

```
.grid-container {  
  display: grid;  
  grid-template-columns: repeat(3, 1fr);  
  grid-gap: 20px;  
  grid-template-areas:  
    "header header header"  
    "sidebar main main"  
    "footer footer footer";  
}
```

CSS Custom Properties (Variables)

```
:root {  
  --primary-color: #3498db;  
  --secondary-color: #2ecc71;  
  --font-size: 16px;  
}  
  
.button {  
  background-color: var(--primary-color);  
  font-size: var(--font-size);  
}
```

Accessibility Considerations

- Use sufficient color contrast ratios
- Provide keyboard navigation support
- Include ARIA labels for screen readers
- Test with accessibility tools
- Follow WCAG 2.1 guidelines

This comprehensive guide covers all essential HTML and CSS concepts from Scrimba's course curriculum ^[1] ^[2] ^[8]. The structured approach takes you from basic HTML tags to advanced CSS layouts and modern web development practices, providing the foundation needed to build professional websites and continue your web development journey.

✱

1. <https://www.freecodecamp.org/news/learn-html-and-css-from-the-ceo-of-scrimba/>
2. <https://scrimba.com/learn/htmlandcss>
3. <https://www.geeksforgeeks.org/web-development/>
4. <https://www.monocubed.com/blog/fundamentals-of-web-development/>
5. <https://www.geeksforgeeks.org/html/tags-vs-elements-vs-attributes-in-html/>
6. https://www.w3schools.com/html/html_elements.asp
7. <https://www.htmldog.com/guides/html/beginner/tags/>
8. <https://www.classcentral.com/course/scrimba-htmlandcss-86270>
9. https://www.w3schools.com/html/html_attributes.asp
10. <https://www.almabetter.com/bytes/tutorials/html/html5-attributes>
11. <https://www.dynamicwebtraining.com.au/blog/element-tag-attributes-in-html/>
12. https://northell.design/wp-content/uploads/2021/11/A_Complete_Web_Development_Guide_For_Non_Technical_Startup_Founder.pdf
13. <https://codecademy.com/en/blog/entry/css-selectors-properties-and-values>
14. https://www.w3schools.com/css/css_selectors.asp
15. <https://www.geeksforgeeks.org/css/css-selectors/>
16. https://developer.mozilla.org/en-US/docs/Learn_web_development/Core/Styling_basics/Basic_selectors
17. https://www.w3schools.com/css/css_attribute_selectors.asp
18. <https://v1.scrimba.com/learn/frontend>
19. <https://brainstation.io/career-guides/how-to-become-a-web-developer>