HTML & CSS Essential Guide for Beginners

Introduction: What You'll Learn

This guide will teach you how to create websites from scratch. Think of HTML as the structure of a house (the walls, rooms, and doors) and CSS as the interior design (the colors, furniture, and decorations).

HTML (HyperText Markup Language) - Creates the content and structure **CSS (Cascading Style Sheets)** - Makes it look beautiful and controls the layout

---

Part 1: HTML Fundamentals

Chapter 1: Understanding Structure

What is HTML?

HTML is like a set of instructions that tells a web browser (like Chrome, Firefox, or Safari) how to display a web page. Just like a newspaper has headlines, paragraphs, and pictures, a web page has similar elements that HTML helps organize.

Think of HTML as organizing information using labeled boxes. Each box (called an "element") contains content and has a label (called a "tag") that tells the browser what kind of content it is.

Your First HTML Page

Every HTML page follows the same basic structure. Let's break it down step by step:

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

Let's understand each part:

1. `<html>` - This is like the cover of a book. It tells the browser "everything inside here is HTML code"
   - Always comes first and last in your page

- Everything else goes between `<html>` and `</html>`

2. **`<head>`** - This is like the table of contents of a book
   - Contains information ABOUT the page (not shown on the page itself)
   - Includes things like the page title and links to style sheets
3. **`<title>`** - This is what appears in the browser tab
   - Shows up in search results
   - Appears when people bookmark your page
4. **`<body>`** - This is like the main content of the book
   - Everything you put here WILL be visible on the web page
   - This is where your actual content lives (text, images, links, etc.)
5. **`<h1>`** - This is a main heading (like a chapter title)
   - Makes text big and bold
   - Helps organize your content
6. **`<p>`** - This stands for "paragraph"
   - Creates a block of text
   - Automatically adds space before and after

Understanding Tags and Elements

Think of tags as labels on a box. They tell the browser what's inside.

Opening Tag: `<p>` - This says "here's where the paragraph starts" **Closing Tag**: `</p>` - This says "here's where the paragraph ends" **Element**: Everything together - `<p>This is a paragraph</p>`

Important: Most tags come in pairs (opening and closing). The closing tag has a forward slash `/` before the tag name.

Example:

```
html

<p>I love learning HTML!</p>
```

- `<p>` = "Start of paragraph"
- `I love learning HTML!` = The actual content
- `</p>` = "End of paragraph"

Self-Closing Tags

Some tags don't need a closing tag because they don't wrap around content. These are called "empty elements" or "self-closing tags."

```html
<br />  <!-- Line break - moves to next line -->
<hr />  <!-- Horizontal rule - draws a line across page -->
<img /> <!-- Image - displays a picture -->
```

Notice the `/` before the `>` - this shows the tag closes itself.

Attributes: Adding Extra Information

Attributes are like settings or properties for an element. They go inside the opening tag and provide additional information.

Basic structure:

```html
<tagname attribute="value">Content</tagname>
```

Real example:

```html
<p lang="en-us">This paragraph is in US English</p>
```

Breaking it down:

- `<p>` = Opening paragraph tag
- `lang` = The attribute name (specifies language)
- `"en-us"` = The attribute value (US English)
- `This paragraph is in US English` = The content
- `</p>` = Closing tag

Another example with multiple attributes:

```html
```

```
<img src="photo.jpg" alt="My photo" width="300" height="200" />
```

Here we have FOUR attributes:

- `src="photo.jpg"` = Where to find the image
- `alt="My photo"` = Description (shows if image doesn't load)
- `width="300"` = How wide in pixels
- `height="200"` = How tall in pixels

How Browsers Read Your Code

When you save an HTML file and open it in a browser:

1. The browser reads your code from top to bottom
2. It sees the tags and understands what type of content each part is
3. It displays the content according to the tags you used
4. Tags are invisible - users only see the content inside them

Important: Spaces and line breaks in your code don't affect how the page looks. The browser ignores extra spaces. To create space in your page, you need to use HTML tags.

Chapter 2: Working with Text

Text is the foundation of most web pages. HTML provides many ways to format and structure text to make it clear and easy to read.

Headings: Creating Hierarchy

Headings are like chapter titles and subheadings in a book. They help organize your content and show what's most important.

HTML has 6 levels of headings:

```
html
```

```
<h1>This is the Main Title - Biggest</h1>
<h2>This is a Major Section</h2>
<h3>This is a Subsection</h3>
<h4>This is a Sub-subsection</h4>
<h5>This is even smaller</h5>
<h6>This is the smallest heading</h6>
```

How to think about headings:

- `<h1>` = Book title (only use ONE per page)
- `<h2>` = Chapter titles
- `<h3>` = Section titles within chapters
- `<h4>`, `<h5>`, `<h6>` = Smaller subdivisions

Why headings matter:

- Help users scan your page quickly
- Make content easier to read
- Help search engines understand your page
- Assist people using screen readers

Real-world example:

html

```
<h1>Chocolate Chip Cookie Recipe</h1>

<h2>Ingredients</h2>
<p>2 cups flour, 1 cup sugar...</p>

<h2>Instructions</h2>

<h3>Step 1: Prepare</h3>
<p>Preheat your oven to 350°F...</p>

<h3>Step 2: Mix</h3>
<p>Combine all dry ingredients...</p>
```

Paragraphs: Blocks of Text

Paragraphs create chunks of text with space before and after them.

```html
<p>This is a paragraph. It contains one or more sentences that form a complete thought. When the browser displays this, it wi

<p>This is another paragraph. Notice how it's separated from the first one.</p>
```

Key points about paragraphs:

- Each `<p>` tag creates a new block of text
- Browser adds space above and below automatically
- Text inside wraps naturally to fit the page width
- Pressing Enter in your code doesn't create a new paragraph - you need a new `<p>` tag

Bold and Italic: Adding Emphasis

There are two types of emphasis - visual and semantic (meaning-based).

Visual Bold:

```html
<b>This text is bold</b> but has no special meaning.
```

Use when you just want text to look bold, like for keywords.

Semantic Bold (Strong Importance):

```html
<strong>This text is important!</strong>
```

Use for warnings or important information. Screen readers will emphasize this differently.

Visual Italic:

```html
<i>This text is italic</i> for style only.
```

Use for technical terms, foreign words, ship names, etc.

Semantic Italic (Emphasis):

```html
<em>This text is emphasized</em> to change meaning.
```

Use when the emphasis changes the meaning, like: "I *love* HTML!"

Complete example:

```html
<p>HTML stands for <strong>HyperText Markup Language</strong>. It's used to create web pages.</p>

<p>The word <i>computer</i> comes from the Latin word <i>computare</i>.</p>

<p>I <em>really</em> think you should learn HTML!</p>
```

Superscript and Subscript

These raise or lower text, useful for mathematics, dates, and chemistry.

Superscript (raised up):

```html
<p>On the 4<sup>th</sup> of July...</p>
<p>E=MC<sup>2</sup></p>
```

Result: 4th of July, $E=MC^2$

Subscript (lowered down):

```html
<p>The chemical formula for water is H<sub>2</sub>O</p>
<p>CO<sub>2</sub> is carbon dioxide</p>
```

Result: H_2O, CO_2

White Space and Line Breaks

White Space Collapsing: The browser ignores extra spaces and line breaks in your code. This:

```html
```

```
<p>This    has    many


spaces    and    lines</p>
```

Displays as: "This has many spaces and lines" (with single spaces)

Line Breaks: To force a line break WITHIN a paragraph (not starting a new paragraph):

```
html

<p>First line<br />Second line<br />Third line</p>
```

Result:
First line
Second line
Third line

Horizontal Rule (Line Across Page):

```
html

<p>Content before the line</p>
<hr />
<p>Content after the line</p>
```

This creates a visual divider across your page.

Semantic Markup: Adding Meaning

These tags add meaning to content, not just appearance.

Quotations:

Long quotes (block quotes):

```
html

<blockquote>
  <p>To be or not to be, that is the question. Whether 'tis nobler in the mind to suffer the slings and arrows of outrageous fortu
</blockquote>
```

Use for quotes that are one or more paragraphs.

Short quotes (inline):

```html
<p>As Shakespeare said, <q>All the world's a stage</q>.</p>
```

Use for quotes within a sentence.

Abbreviations and Acronyms:

```html
<p><abbr title="Professor">Prof.</abbr> Smith teaches at <abbr title="Massachusetts Institute of Technology">MIT</abbr>.
```

When you hover over the abbreviation, the full text appears!

Citations (Referencing Works):

```html
<p>My favorite book is <cite>To Kill a Mockingbird</cite> by Harper Lee.</p>
```

Use for books, movies, songs, etc.

Definitions:

```html
<p>A <dfn>variable</dfn> is a container for storing data values.</p>
```

Use when defining a term for the first time.

Addresses:

```html
<address>
  <p>Contact us at:</p>
  <p>123 Main Street<br />
  New York, NY 10001<br />
  Email: info@example.com</p>
</address>
```

Changes to Content:

```html
<p>My favorite color is <del>blue</del> <ins>red</ins>.</p>
<p>Original price: <s>$100</s> Now only $75!</p>
```

- `<del>` shows deleted content (strikethrough)
- `<ins>` shows inserted content (underlined)
- `<s>` shows content that's no longer accurate

Complete Example:

```html
<h1>Introduction to Web Development</h1>

<p><strong>Web development</strong> is the work involved in developing websites. It can range from developing a simple

<p>According to <cite>The Web Developer's Handbook</cite>, learning <abbr title="HyperText Markup Language">HTML

<blockquote>
  <p>The web is more a social creation than a technical one.</p>
</blockquote>

<p>A <dfn>web browser</dfn> is software that accesses and displays web pages.</p>
```

Chapter 3: Creating Lists

Lists help organize information in a clear, scannable way. HTML provides three types of lists for different purposes.

Ordered Lists: When Order Matters

Use ordered lists when the sequence is important (like steps in a recipe or rankings).

```html
<ol>
  <li>First item</li>
  <li>Second item</li>
  <li>Third item</li>
</ol>
```

What you'll see:

1. First item
2. Second item
3. Third item

Breaking it down:

- `<ol>` = "Ordered List" (numbered list)
- `<li>` = "List Item" (each individual item)
- The browser automatically adds numbers

Real-world example - Recipe steps:

```html
<h2>How to Make a Sandwich</h2>
<ol>
  <li>Get two slices of bread</li>
  <li>Spread butter on one side of each slice</li>
  <li>Add your favorite fillings</li>
  <li>Put the slices together</li>
  <li>Cut diagonally and enjoy!</li>
</ol>
```


Unordered Lists: When Order Doesn't Matter

Use unordered lists for items where sequence doesn't matter (like shopping lists or feature lists).

```html
<ul>
  <li>Milk</li>
  <li>Eggs</li>
  <li>Bread</li>
  <li>Cheese</li>
</ul>
```

What you'll see:

- Milk
- Eggs

- Bread
- Cheese

Breaking it down:

- `<ul>` = "Unordered List" (bullet list)
- `<li>` = "List Item" (same as ordered lists)
- The browser automatically adds bullet points

Real-world example - Features:

```
html

<h2>Website Features</h2>
<ul>
  <li>User-friendly navigation</li>
  <li>Mobile responsive design</li>
  <li>Fast loading speed</li>
  <li>Secure checkout</li>
</ul>
```

Definition Lists: Term and Definition Pairs

Use definition lists for glossaries, FAQs, or any term-explanation pairs.

```
html

<dl>
  <dt>HTML</dt>
  <dd>A markup language for creating web pages</dd>

  <dt>CSS</dt>
  <dd>A style sheet language for designing web pages</dd>

  <dt>JavaScript</dt>
  <dd>A programming language that makes web pages interactive</dd>
</dl>
```

What you'll see: HTML A markup language for creating web pages **CSS** A style sheet language for designing web pages **JavaScript** A programming language that makes web pages interactive

Breaking it down:

- `<dl>` = "Definition List" (the container)
- `<dt>` = "Definition Term" (the word being defined)
- `<dd>` = "Definition Description" (the explanation)

Real-world example - FAQ:

```
html

<h2>Frequently Asked Questions</h2>
<dl>
  <dt>What is HTML?</dt>
  <dd>HTML stands for HyperText Markup Language. It's the standard language for creating web pages.</dd>

  <dt>Do I need to know coding to learn HTML?</dt>
  <dd>No! HTML is beginner-friendly and is a great first step into web development.</dd>

  <dt>How long does it take to learn HTML?</dt>
  <dd>You can learn the basics in a few weeks, but mastery comes with practice over several months.</dd>
</dl>
```

Nested Lists: Lists Within Lists

You can put lists inside lists! This is useful for creating sub-items or hierarchies.

Example - Nested unordered list:

```
html

<h2>Website Sections</h2>
<ul>
  <li>Home</li>
  <li>Products
    <ul>
      <li>Electronics</li>
      <li>Clothing</li>
      <li>Books</li>
    </ul>
  </li>
  <li>About Us</li>
  <li>Contact</li>
</ul>
```

What you'll see:

- Home
- Products ◦ Electronics ◦ Clothing ◦ Books
- About Us
- Contact

Important tips:

- The nested list goes INSIDE the `<li>` tag of the parent item
- You can nest any type of list inside any other type
- The browser automatically changes the bullet style for nested items

Example - Recipe with nested steps:

html

```
<h2>Baking a Cake</h2>
<ol>
  <li>Prepare ingredients
    <ul>
      <li>2 cups flour</li>
      <li>1 cup sugar</li>
      <li>3 eggs</li>
    </ul>
  </li>
  <li>Mix dry ingredients</li>
  <li>Add wet ingredients
    <ol>
      <li>Beat eggs</li>
      <li>Add to mixture</li>
      <li>Stir until smooth</li>
    </ol>
  </li>
  <li>Bake at 350°F for 30 minutes</li>
</ol>
```


When to Use Each List Type

Use Ordered Lists (`<ol>`) when:

- Instructions or steps
- Rankings (top 10 lists)

- Sequences that must be followed in order
- Numbered points in an argument

Use Unordered Lists (`<ul>`) when:

- Shopping lists
- Features or benefits
- Menu navigation
- Any collection where order doesn't matter

Use Definition Lists (`<dl>`) when:

- Glossaries
- FAQs
- Specifications (product features)
- Any term-definition pairs

Practice Exercise: Try creating a list for your daily routine. Use ordered lists for time-sensitive tasks and unordered lists for tasks you can do anytime!

Chapter 4: Links

Basic Link Structure

```html
<a href="http://www.example.com">Link text</a>
```

Linking to Other Sites (Absolute URLs)

```html
<a href="http://www.imdb.com">IMDB</a>
```

Linking to Pages on Same Site (Relative URLs)

```html
```

```
<a href="about.html">About</a>
<a href="products/index.html">Products</a>
```

Email Links

```
html

<a href="mailto:email@example.com">Email Jon</a>
```

Opening Links in New Window

```
html

<a href="http://www.example.com" target="_blank">Opens in new window</a>
```

Linking to Specific Part of Page

```
html

<!-- Create the target -->
<h1 id="top">Page Title</h1>

<!-- Link to the target -->
<a href="#top">Back to top</a>
```

Chapter 5: Images

Adding Images

```
html

<img src="images/photo.jpg" alt="Description of image"
    title="Additional information" />
```

Attributes:

- `src`: Path to the image file
- `alt`: Text description (essential for accessibility)
- `title`: Additional information (appears on hover)
- `width` and `height`: Dimensions in pixels

Image Sizes

Specify dimensions to help page load smoothly:

```html
<img src="images/bird.jpg" alt="Bird" width="200" height="150" />
```

Where to Place Images

Images can be placed:

- Before a paragraph (paragraph starts on new line)
- Inside start of paragraph (first row aligns with bottom of image)
- In middle of paragraph (flows with surrounding text)

Figures with Captions (HTML5)

```html
<figure>
  <img src="images/photo.jpg" alt="Description" />
  <figcaption>Caption for the image</figcaption>
</figure>
```

Chapter 6: Tables

Basic Table Structure

```html
<table>
  <tr>
    <td>Cell 1</td>
    <td>Cell 2</td>
  </tr>
  <tr>
    <td>Cell 3</td>
    <td>Cell 4</td>
  </tr>
</table>
```

Table Headings

```html
<table>
  <tr>
    <th>Heading 1</th>
    <th>Heading 2</th>
  </tr>
  <tr>
    <td>Data 1</td>
    <td>Data 2</td>
  </tr>
</table>
```

Spanning Columns and Rows

```html
<!-- Spanning columns -->
<td colspan="2">Spans 2 columns</td>

<!-- Spanning rows -->
<td rowspan="3">Spans 3 rows</td>
```


Long Tables

```html
<table>
  <thead>
    <tr><th>Header</th></tr>
  </thead>
  <tbody>
    <tr><td>Body content</td></tr>
  </tbody>
  <tfoot>
    <tr><td>Footer</td></tr>
  </tfoot>
</table>
```

Chapter 7: Forms

Basic Form Structure

```html
<form action="process.php" method="post">
  <!-- Form controls go here -->
</form>
```

Text Input

```html
<input type="text" name="username" maxlength="30" />
```

Password Input

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

Text Area

```html
<textarea name="comments" cols="20" rows="4">
  Default text here
</textarea>
```

Radio Buttons

```html
<input type="radio" name="genre" value="rock" checked="checked" /> Rock
<input type="radio" name="genre" value="pop" /> Pop
```

Checkboxes

```html
<input type="checkbox" name="service" value="email" checked="checked" /> Email
<input type="checkbox" name="service" value="sms" /> SMS
```

Drop-down List

```html
<select name="devices">
  <option value="ipod">iPod</option>
  <option value="radio">Radio</option>
  <option value="computer">Computer</option>
</select>
```

File Upload

```html
<input type="file" name="user-file" />
```

Submit Button

```html
<input type="submit" value="Submit" />
```

Labeling Form Controls

```html
<label>Name: <input type="text" name="name" /></label>

<!-- Or separately -->
<label for="email">Email:</label>
<input type="text" id="email" name="email" />
```

Grouping Form Elements

```html
<fieldset>
  <legend>Contact Details</legend>
  <label>Email: <input type="text" name="email" /></label>
  <label>Mobile: <input type="text" name="mobile" /></label>
</fieldset>
```

HTML5 Form Features

```html
<!-- Date input -->
<input type="date" name="birthday" />

<!-- Email input -->
<input type="email" name="email" />

<!-- URL input -->
<input type="url" name="website" />

<!-- Search input -->
<input type="search" name="search" placeholder="Enter keyword" />
```

Part 2: CSS Fundamentals

Chapter 10: Introducing CSS

What is CSS?

CSS (Cascading Style Sheets) allows you to control the appearance and layout of your web pages. CSS rules specify how HTML elements should be displayed.

CSS Rule Structure

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

Example:

```css
p {
  font-family: Arial;
  color: blue;
}
```

Using External CSS

In HTML file:

```html
<head>
  <link href="styles.css" rel="stylesheet" type="text/css" />
</head>
```

In styles.css:

```css
body {
  font-family: Arial;
  background-color: #f0f0f0;
}
```

Using Internal CSS

```html
<head>
  <style type="text/css">
    body {
      font-family: Arial;
      color: #333;
    }
  </style>
</head>
```


CSS Selectors

Universal Selector:

```css
* {
  margin: 0;
  padding: 0;
}
```

Type Selector:

```css
```

```
h1, h2, h3 {
  font-weight: normal;
}
```

Class Selector:

```
css

.note {
  color: red;
}

p.note {
  /* Only p elements with class="note" */
}
```

ID Selector:

```
css

#introduction {
  font-weight: bold;
}
```


Child Selector:

```
css

li>a {
  /* a elements that are children of li */
}
```

Descendant Selector:

```
css

p a {
  /* a elements that sit inside p */
}
```

How CSS Rules Cascade

When multiple rules apply to the same element:

1. **Last Rule**: If selectors are identical, the latter takes precedence
2. **Specificity**: More specific selectors take precedence
3. **Important**: Rules with `!important` override others

```
css

p {
  color: blue !important;
}
```

Inheritance

Some properties (like `font-family` and `color`) are inherited by child elements, while others (like `border` and `padding`) are not.

Chapter 11: Color

Specifying Colors

Color Names:

```
css

h1 {
  color: DarkCyan;
}
```


Hex Codes:

```
css

h2 {
  color: #ee3e80;
}
```

RGB Values:

```
css

p {
  color: rgb(100, 100, 90);
}
```

Background Color

```css
body {
  background-color: #f0f0f0;
}

h1 {
  background-color: rgb(200, 200, 200);
}
```

Understanding Color

HSL (Hue, Saturation, Lightness):

```css
body {
  background-color: hsl(0, 0%, 78%);
}
```

RGBA and HSLA (with Opacity):

```css
p {
  background-color: rgba(0, 0, 0, 0.5); /* 50% opacity */
}

div {
  background-color: hsla(0, 100%, 100%, 0.5);
}
```

Opacity

```css
p {
  opacity: 0.5; /* 50% transparent */
}
```

Chapter 12: Text

Font Family

```css
body {
  font-family: Georgia, Times, serif;
}

h1, h2 {
  font-family: Arial, Verdana, sans-serif;
}
```

Font Size

```css
/* Pixels */
body {
  font-size: 12px;
}

/* Percentages */
h1 {
  font-size: 200%;
}

/* Ems */
h2 {
  font-size: 1.3em;
}
```


Font Weight and Style

```css
.credits {
  font-weight: bold;
  font-style: italic;
}
```

Text Transformation

```css
h1 {
  text-transform: uppercase;
}

h2 {
  text-transform: lowercase;
}

.credits {
  text-transform: capitalize;
}
```

Text Decoration

```css
.credits {
  text-decoration: underline;
}

a {
  text-decoration: none;
}
```


Line Height and Letter Spacing

```css
p {
  line-height: 1.4em;
  letter-spacing: 0.2em;
}
```

Text Alignment

```css
```

```
h1 {
  text-align: left;
}

p {
  text-align: justify;
}

.credits {
  text-align: right;
}
```

Text Indentation

css

```
.credits {
  text-indent: 20px;
}
```

Text Shadow

css

```
p {
  text-shadow: 1px 1px 3px #666666;
}
```

Styling Links

css

```css
/* Unvisited link */
a:link {
  color: deeppink;
  text-decoration: none;
}

/* Visited link */
a:visited {
  color: black;
}

/* Mouse over link */
a:hover {
  color: deeppink;
  text-decoration: underline;
}

/* Selected link */
a:active {
  color: darkcyan;
}
```

Chapter 13: Boxes

Box Dimensions

```css
div {
  height: 300px;
  width: 300px;
}
```

Limiting Width and Height

```css
```

```css
td {
  min-width: 450px;
  max-width: 650px;
  min-height: 10px;
  max-height: 30px;
}
```

Overflow

```css
p {
  overflow: hidden;  /* Hides overflow */
  overflow: scroll;  /* Adds scrollbar */
}
```

Border, Margin, and Padding

Border:

```css
p {
  border-width: 2px;
  border-style: solid;
  border-color: #0088dd;
  /* Shorthand */
  border: 3px dotted #0088dd;
}
```

Padding:

```css
p {
  padding: 10px;  /* All sides */
  padding: 10px 5px 3px 1px;  /* Top, Right, Bottom, Left */
}
```

Margin:

```css
```

```css
p {
  margin: 20px;
  /* Centering a box */
  margin: 10px auto;
}
```

Display Property

```css
li {
  display: inline;  /* Inline element */
  display: block;   /* Block-level element */
  display: none;    /* Hidden */
}
```

Visibility

```css
li {
  visibility: hidden;  /* Hidden but space remains */
  visibility: visible;
}
```


Border Radius (Rounded Corners)

```css
p {
  border-radius: 10px;
}
```

Box Shadow

```css
p {
  box-shadow: 5px 5px 5px #777777;
}
```

Chapter 15: Layout

Positioning Schemes

Normal Flow (default):

```
css

body {
  width: 750px;
}
```

Relative Positioning:

```
css

p {
  position: relative;
  top: 10px;
  left: 100px;
}
```

Absolute Positioning:

```
css

h1 {
  position: absolute;
  top: 0px;
  left: 500px;
}
```

Fixed Positioning:

```
css

h1 {
  position: fixed;
  top: 0px;
  left: 50px;
}
```

Z-Index (Overlapping Elements)

```css
h1 {
  position: fixed;
  z-index: 10;
}

p {
  position: relative;
  z-index: 5;
}
```

Floating Elements

```css
blockquote {
  float: right;
  width: 275px;
}

/* Clearing floats */
.clear {
  clear: left;   /* or right, both, none */
}
```


Creating Multi-Column Layouts

```css
.column1of2 {
  float: left;
  width: 620px;
  margin: 10px;
}

.column2of2 {
  float: left;
  width: 300px;
  margin: 10px;
}
```

Fixed Width Layouts

```css
body {
  width: 960px;
  margin: 0 auto;
}
```

Liquid Layouts

```css
body {
  width: 90%;
  margin: 0 auto;
}

.column {
  width: 31.3%;
  float: left;
  margin: 1%;
}
```

Chapter 16: Images

Controlling Image Sizes

```css
img.large {
  width: 500px;
  height: 500px;
}

img.medium {
  width: 250px;
  height: 250px;
}
```

Aligning Images

```css
```

```css
img.align-left {
  float: left;
  margin-right: 10px;
}

img.align-right {
  float: right;
  margin-left: 10px;
}
```

Centering Images

css

```css
img.align-center {
  display: block;
  margin: 0px auto;
}
```

Background Images

css

```css
body {
  background-image: url("images/pattern.gif");
}

p {
  background-image: url("images/texture.jpg");
}
```

Background Properties

css

```
body {
  background-image: url("images/header.gif");
  background-repeat: repeat-x;      /* or repeat-y, no-repeat */
  background-attachment: fixed;     /* or scroll */
  background-position: center top;  /* or 50% 50%, etc. */
}

/* Shorthand */
body {
  background: #ffffff url("images/tulip.gif") no-repeat top right;
}
```

Gradients

```
css

#gradient {
  background-color: #66cccc; /* Fallback */
  background-image: linear-gradient(#336666, #66cccc);
}
```

Chapter 17: HTML5 Layout

New HTML5 Layout Elements

Header and Footer:

```
html

<header>
  <h1>Site Title</h1>
  <nav><!-- navigation --></nav>
</header>

<footer>
  <p>&copy; 2023 Company Name</p>
</footer>
```

Navigation:

```
html
```

```
<nav>
  <ul>
    <li><a href="/">Home</a></li>
    <li><a href="/about">About</a></li>
    <li><a href="/contact">Contact</a></li>
  </ul>
</nav>
```

Articles:

```
html

<article>
  <h2>Article Title</h2>
  <p>Article content...</p>
</article>
```

Sections:

```
html

<section>
  <h2>Section Heading</h2>
  <p>Section content...</p>
</section>
```


Asides:

```
html

<aside>
  <h2>Related Links</h2>
  <ul>
    <li><a href="#">Link 1</a></li>
    <li><a href="#">Link 2</a></li>
  </ul>
</aside>
```

Helping Older Browsers

CSS for older browsers:

```
css
```

```
header, section, footer, aside, nav, article, figure, figcaption {
  display: block;
}
```

HTML5 Shiv (for IE8 and older):

```
html

<!--[if lt IE 9]>
  <script src="http://html5shiv.googlecode.com/svn/trunk/html5.js"></script>
<![endif]-->
```

Essential Concepts

Box Model

Every element is treated as a box with:

- **Content**: The actual content (text, images)
- **Padding**: Space between content and border
- **Border**: Edge of the box
- **Margin**: Space between boxes

CSS Specificity

Priority order (highest to lowest):

1. Inline styles
2. IDs
3. Classes, attributes, pseudo-classes
4. Elements and pseudo-elements

Responsive Design Considerations

- Use relative units (%, em) for flexible layouts
- Consider min-width and max-width
- Test on different screen sizes
- Use media queries for different devices

File Organization

```
website/
  index.html
  about.html
  css/
    styles.css
  images/
    logo.png
    photo.jpg
  js/
    scripts.js
```

Best Practices

1. **HTML**:
   - Use semantic elements
   - Include alt text for images
   - Validate your HTML
   - Use proper nesting
2. **CSS**:
   - Use external stylesheets
   - Keep selectors simple
   - Use shorthand properties
   - Comment your code
   - Group related rules
3. **General**:
   - Test in multiple browsers
   - Optimize images
   - Use consistent naming conventions
   - Keep files organized

Quick Reference

Common HTML Elements

- `<div>`: Generic container (block)
- `<span>`: Generic container (inline)
- `<header>`, `<footer>`, `<nav>`, `<article>`, `<section>`, `<aside>`: HTML5 semantic elements
- `<img>`: Image
- `<a>`: Link
- `<ul>`, `<ol>`, `<li>`: Lists
- `<table>`, `<tr>`, `<td>`, `<th>`: Tables
- `<form>`, `<input>`, `<textarea>`, `<select>`: Forms

Common CSS Properties

- `color`: Text color
- `background-color`: Background color
- `font-family`: Typeface
- `font-size`: Text size
- `font-weight`: Bold/normal
- `text-align`: Alignment
- `width`, `height`: Dimensions
- `margin`, `padding`: Spacing
- `border`: Border styling
- `display`: Display type
- `position`: Positioning scheme
- `float`: Float elements

Color Formats

- **Named**: `red`, `blue`, `white`
- **Hex**: `#ff0000`, `#0000ff`
- **RGB**: `rgb(255, 0, 0)`

- **RGBA**: `rgba(255, 0, 0, 0.5)`
- **HSL**: `hsl(0, 100%, 50%)`
- **HSLA**: `hsla(0, 100%, 50%, 0.5)`

This guide covers the essential concepts from "HTML and CSS: Design and Build Websites" to help you start creating web pages effectively.