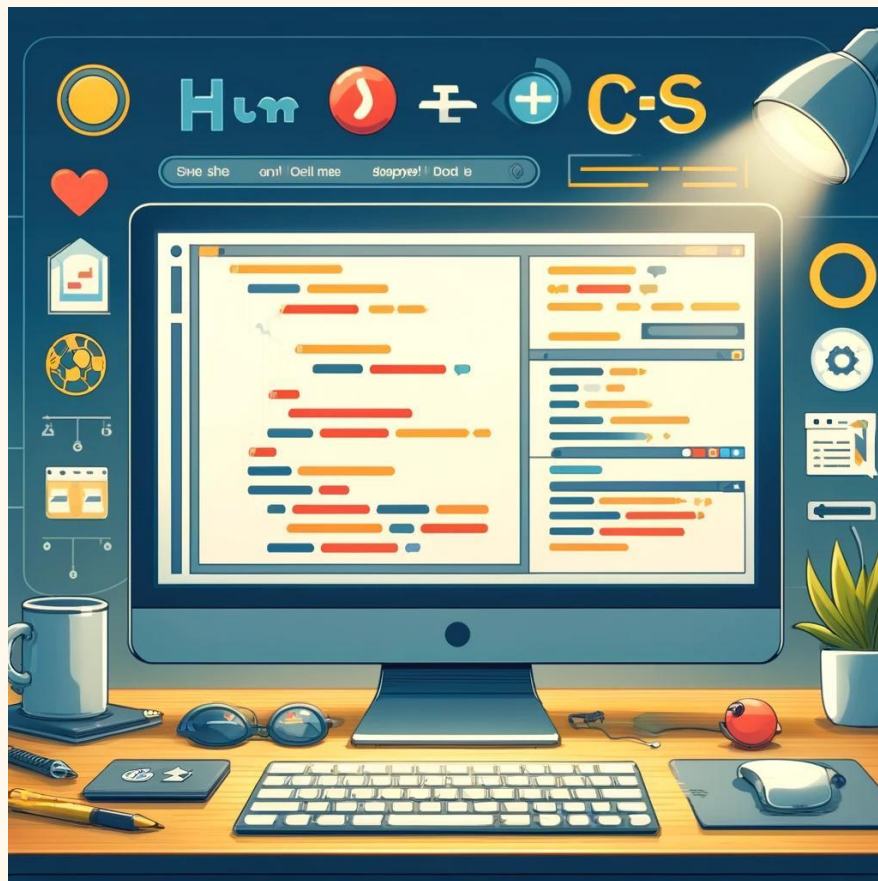


BASIC WEB DEVELOPMENT:

How to Build a Website Using HTML, CSS and JavaScript



INTRODUCTION

Web development is an essential skill in today's digital world. This tutorial will guide you through the fundamentals of building a basic website using HTML, CSS, and JavaScript. By the end of this tutorial, you will have a clear understanding of how to create and style interactive web pages.

CHAPTER 1: HTML FUNDAMENTALS

Introduction

HTML (HyperText Markup Language) is the standard language for creating and structuring web page content. It is the skeleton of any website, providing the basic structure upon which styles and additional functionalities can be applied. This chapter will cover the essential concepts of HTML, including the basic structure of an HTML document, common tags, and how to create links and images.

Basic Structure of an HTML Document

An HTML document consists of a series of elements organized in a hierarchical manner. Each element is defined by HTML tags, which are written between angle brackets (< and >).

Structure of a Typical HTML Document:

```
<!DOCTYPE html>
<html>
<head>
  <title>Mi Primera Página Web</title>
</head>
<body>
  <h1>Bienvenido a Mi Página Web</h1>
  <p>Este es un párrafo de ejemplo.</p>
</body>
</html>
```

- ★ <!DOCTYPE html>: Declares the document type and HTML version.
- ★ <html>: Root element that contains all the content of the page.
- ★ <head>: Contains metadata such as the page title and links to styles and scripts.
- ★ <title>: Defines the title of the page that appears on the browser tab.
- ★ <body>: Contains all the visible content of the webpage.

Common HTML Tags

HTML tags are the building blocks of a webpage. Below are some of the most commonly used tags:

- Headings (<h1> to <h6>): Used to define titles and subtitles.

```
<h1>Main Title</h1>
```

```
<h2>Subtitle</h2>
```

- Paragraph (<p>): Defines a paragraph of text.

```
<p>This is a sample paragraph.</p>
```

- Link (): Creates a hyperlink to another page or resource.

```
<a href="https://www.ejemplo.com">Visit Ejemplo.com</a>
```

- Image (): Inserts an image into the page..

```

```

- Lists:

- Unordered List (and):

```
<ul>
```

```
<li>Item 1</li>
```

```
<li>Item 2</li>
```

```
<li>Item 3</li>
```

```
</ul>
```

- Ordered List (and):

```
<ol>
```

```
<li>Item 1</li>
```

```
<li>Item 2</li>
```

```
<li>Item 3</li>
```

```
</ol>
```

- Division (<div>):Used as a container to group other elements.

```
<div>
```

```
<h2>Website Section</h2>
```

```
<p>Section content.</p>
```

```
</div>
```

- Span (): Similar to <div>, but used to group inline content.

<p>This is an inline text.</p>

HTML Tag Attributes

HTML tags can have attributes that provide additional information about the element. Attributes are written inside the opening tag.

Common Attributes:

- id: Unique identifier for the element.

<p id="intro">This is a paragraph with a unique ID.</p>

- class: Class that can be shared with other elements to apply CSS styles

<p class="highlighted-text">This is a highlighted text.</p>

- style: Defines inline styles for an element.

<p style="color: blue; font-size: 14px;">Text with inline style.</p>

- href: Defines the URL for a link.

Link to Example.com

- src y alt: Used in images to define the source and alternative text.

Forms in HTML

Forms allow users to submit data to a server. An HTML form is defined with the <form> tag and can contain various types of input fields.

Form Example:

```
<form action="/submit" method="post">
  <label for="name">Name:</label>
  <input type="text" id="name" name="name">

  <label for="email">Email:</label>
  <input type="email" id="email" name="email">

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

- `<form>`: Defines the form.
- `<label>`: Defines a label for an input field.
- `<input>`: Defines an input field (various types available).
- `<submit>`: Button to submit the form.

Links and Navigation

Links (`<a>`) are essential for navigation on the web, allowing users to move between different pages or sections.

Creating Links:

- Internal Link: `<a href="#section1">Go to Section 1</a>`
- External Link: `<a href="https://www.google.com">Visit Google</a>`
- Download Link: `<a href="file.pdf" download="file">Download PDF</a>`

HTML Best Practices

- Use Semantic Tags: Such as `<header>`, `<footer>`, `<article>`, and `<section>`.
- Validate HTML Code: Use tools like the W3C validator to ensure code complies with web standards.
- Maintain Clear Structure: Organize code into sections and use comments to describe parts of the code.

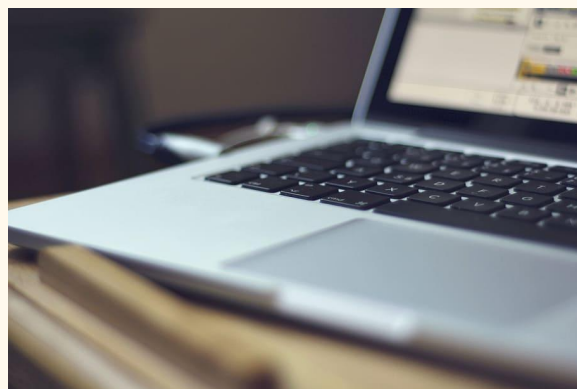
Example of Semantic Tags:

```
<!DOCTYPE html>
<html>
<head>
  <title>Semantic Page</title>
</head>
<body>
  <header>
    <h1>Main Header</h1>
  </header>
  <nav>
    <ul>
      <li><a href="#home">Home</a></li>
      <li><a href="#about">About Us</a></li>
      <li><a href="#contact">Contact</a></li>
    </ul>
  </nav>
  <main>
    <article>
      <h2>Main Article</h2>
      <p>Article content.</p>
    </article>
  </main>
  <footer>
    <p>Footer text</p>
  </footer>
</body>
</html>
```

CHAPTER 2: CSS FUNDAMENTALS

Introduction

CSS (Cascading Style Sheets) is the language used to describe the presentation of HTML documents. CSS allows the separation of content from presentation, providing greater flexibility and control over web page design. In this chapter, we will explore CSS syntax, selectors, style properties, and how to apply styles to HTML elements to create attractive and professional designs.



CSS Syntax

CSS syntax consists of selectors and declarations. Each declaration includes a property and a value, separated by a colon (:) and ended with a semicolon (;).

CSS Syntax Example:

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

Applied Example:

```
h1 {  
  color: blue;  
  font-size: 24px;  
}
```

- Selector (h1): Targets the HTML elements to be styled.
- Property (color): Indicates the specific characteristic to be styled.
- Value (blue): Defines the appearance of the property.

CSS Selectors

Selectors are used to target the HTML elements that you want to style. There are several types of selectors in CSS.

Common Selectors

- Element Selector: Targets all elements of a specific type.

```
p {  
  color: green;  
}
```
- Class Selector: Targets all elements with a specific class. Denoted by a period (.) followed by the class name.

```
.clase-ejemplo {  
  font-size: 18px;  
}
```
- ID Selector: Targets an element with a specific ID. Denoted by a hash sign (#) followed by the ID name.

```
#example-id {  
  background-color: yellow;  
}
```
- Universal Selector (*): Targets all elements.

```
* {  
  margin: 0;  
  padding: 0;  
}
```
- Attribute Selector: Targets elements with a specific attribute.

```
input[type="text"] {  
  border: 1px solid #ccc;  
}
```

CSS Style Properties

CSS allows you to style various aspects of HTML elements. Below are some of the most commonly used style properties.

- Text Properties:
 - color: Defines the text color. `p { color: #333;}`
 - font-size: Defines the font size. `h1 { font-size: 32px;}`
 - text-align: Aligns the text. `h2 { text-align: center;}`
 - text-decoration: Adds decorations to the text. `a { text-decoration: none;}`
 - font-family: Defines the font family. `body { font-family: Arial, sans-serif;}`
- Background Properties:
 - background-color: Defines the background color. `div { background-color: #f0f0f0;}`
 - background-image: Defines a background image. `body {background-image: url('fondo.jpg');}`
 - background-size: Defines the size of the background image.. `header { background-size: cover;}`
- Border and Spacing Properties:
 - border: Defines the border of an element. `img { border: 2px solid #000;}`
 - margin: Defines the outer space around an element. `.contenedor { margin: 20px;}`
 - padding: Defines the inner space within an element. `.contenido { padding: 15px;}`

CSS Layout Models

CSS offers various layout models to organize and align elements on a web page.

CSS Box Model:

Every element in CSS is represented as a rectangular box that contains properties such as margin, border, padding, and content.

```
.box {
  width: 200px;
  padding: 10px;
  border: 1px solid #ccc;
  margin: 20px;
}
```

Flexbox Model:

Flexbox is a layout model that allows creating flexible and efficient layouts.

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

Grid Model:

Grid is a layout model that provides a way to create complex layouts with rows and columns.

```
.container-grid {
  display: grid;
  grid-template-columns: 1fr 1fr 1fr;
  gap: 10px;
}
```

Media Queries and Responsive Design

Media Queries allow applying specific CSS styles based on the screen size, making the design responsive.

- Media Query Example:


```
@media (max-width: 600px) {
  .container {
    flex-direction: column;
  }
}
```

Integrating CSS with HTML

CSS can be integrated with HTML in three ways: inline, internal, and external.

- Inline CSS: Uses the style attribute directly in the HTML element.

```
<p style="color: red;">Red text</p>
```

- Internal CSS: Uses a <style> tag within the <head> of the HTML document.

```
<head>  
  <style>  
    body {  
      background-color: #e0e0e0;  
    }  
  </style>  
</head>
```

- External CSS: Uses a separate CSS file, linked with the <link> tag..

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

Practical Examples

1. Styling a Basic Web Page:

a. index.html:

```
<!DOCTYPE html>
<html>
<head>
  <title>Styled Page</title>
  <link rel="stylesheet" type="text/css" href="styles.css">
</head>
<body>
  <header>
    <h1>My Web Page</h1>
  </header>
  <nav>
    <ul>
      <li><a href="#home">Home</a></li>
      <li><a href="#about">About Us</a></li>
      <li><a href="#contact">Contact</a></li>
    </ul>
  </nav>
  <main>
    <section id="home">
      <h2>Home</h2>
      <p>Welcome to my website.</p>
    </section>
    <section id="about">
      <h2>About Us</h2>
      <p>Information about us.</p>
    </section>
  </main>
  <footer>
    <p>Contact: contact@example.com</p>
  </footer>
</body>
</html>
```

b. styles.css:

```
body {
  font-family: Arial, sans-serif;
  background-color: #f4f4f4;
  margin: 0;
  padding: 0;
}
```

```
header {
  background-color: #333;
  color: #fff;
  padding: 10px 0;
  text-align: center;
}

nav ul {
  list-style: none;
  padding: 0;
}

nav ul li {
  display: inline;
  margin-right: 10px;
}

nav ul li a {
  text-decoration: none;
  color: #333;
}

main {
  padding: 20px;
}

footer {
  background-color: #333;
  color: #fff;
  text-align: center;
  padding: 10px 0;
  position: absolute;
  width: 100%;
  bottom: 0;
}
```