

HTML Fundamentals

Introduction to HTML and the Document Structure

HTML is the essential building block of all web pages, forming the structure and content that browsers display. Learning it is crucial for anyone interested in creating websites, regardless of their technical role or experience level. This lesson serves as an introduction, offering a foundational understanding that supports further learning in web development.

What is HTML

HTML is a language used to define the structure of web pages. It connects different parts of the web through links and uses tags to guide browsers in displaying content. Unlike programming languages that perform actions, HTML focuses on organizing and presenting information on a webpage.

HTML's Role in Web Development

HTML works in conjunction with other technologies to create complete web experiences. Here's a simplified overview:

HTML: Provides the structure and content of the page.

CSS (Cascading Style Sheets): Controls the visual presentation of the page (e.g., colors, fonts, layout).

JavaScript: Adds interactivity and dynamic behavior to the page.

Web browsers read HTML files and render them visually for users to see. Without HTML, browsers wouldn't know how to display text, images, videos, or any other content.

A Brief History of HTML

HTML was invented by Tim Berners-Lee in 1990 while working at CERN. It was initially conceived as a way for scientists to share documents over the internet. Since then, HTML has undergone several revisions and improvements.

- **HTML 2.0:** The first formal specification of HTML.
- **HTML 4.01:** A widely used version that introduced many new features.
- **XHTML:** A stricter version of HTML based on XML (Extensible Markup Language).
- **HTML5:** The latest major revision, which introduced many new semantic elements, multimedia capabilities, and APIs. HTML5 is the current standard and the focus of this course.

Basic HTML Document Structure

```
<!DOCTYPE html>
```

```
<html lang="en">
```

```
<head>
```

```
  <meta charset="UTF-8">
```

```
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
```

```
  <title>Document Title</title>
```

```
</head>
```

```
<body>
```

```
  <!-- Content of the page goes here -->
```

```
</body>
```

```
</html>
```

`<!DOCTYPE html>`

Declares that the document uses HTML5, instructing the browser how to render the page.

`<html>` Tag

Root element of the HTML document; contains all other elements and defines the page's language for accessibility and SEO.

`<head>` Tag

Holds meta-information like title, charset, viewport, and stylesheet links, essential for browsers and search engines but not shown on the page.

`<meta charset="UTF-8">`

Sets the document's character encoding to UTF-8, ensuring proper text display in multiple languages.

`<meta name="viewport">`

Controls how the page scales on different devices, enabling responsive design.

`width=device-width & initial-scale=1.0`

Defines the page width as the device's screen width and sets the initial zoom level, improving readability on mobile devices.

<title> Tag

Specifies the page's title, shown in the browser tab and used by search engines for indexing.

<body> Tag

Contains all visible content of the web page, such as text, images, videos, and links.

HTML Tags and Elements

HTML documents are built using elements. AN HTML element typically consists of a start tag, some content, and an end tag.

Start Tag: Marks the beginning of the element **<p>**

Content: The actual text, images, or other elements that the tag contains.

End Tag: Markes the end of the element **</p>**. End tags are the same as starts, but with a forward slash / before the tag name.

Here's an example of a paragraph element:

```
html
<p>This is a paragraph of text.</p>
```

Some elements, called *empty elements*, don't have an end tag. These elements typically represent things like images or line breaks. For example:

```
html

<br>
```

Tag Attributes

HTML tags can have *attributes* that provide additional information about the element. Attributes are specified in the start tag and consist of a name and a value, separated by an equals sign (=).

```
html
<a href="https://www.example.com">Visit Example Website</a>
```

In this example, the `<a>` tag (which creates a hyperlink) has an `href` attribute that specifies the URL to link to.

Common attributes include:

- `id`: A unique identifier for the element.
- `class`: A CSS class that can be used to style the element.
- `style`: Inline CSS styles for the element (generally discouraged in favor of external stylesheets).
- `src`: The source URL for an image, video, or other media.
- `alt`: Alternative text for an image, used for accessibility and when the image can't be loaded.
- `title`: A tooltip that appears when the user hovers over the element.

Block-Level vs Inline Elements

HTML elements can be categorized as either block-level or inline elements.

Block-Level Elements: These elements take up the full width available to them and always start on a new line.

Examples include `<div>`, `<p>`, `<h1>`-`<h6>`, ``, ``, and ``

Inline Elements: These elements only take up as much width as necessary to contain their content and do not start on a new line. Examples include ``, `<a>`, ``, ``, and ``.

You can use CSS to change the default behavior of these elements, but it's important to understand the distinction between them.

Example of Block-Level Elements

```
html
<div>This is a div element.</div>
<div>This is another div element.</div>

<p>This is a paragraph element.</p>
<p>This is another paragraph element.</p>
```

Each `<div>` and `<p>` element will appear on a new line.

Example of Inline Elements

```
html
<span>This is a span element.</span><span>This is another span element.</span>

<a href="#">This is a link.</a><a href="#">This is another link.</a>
```

The `` and `<a>` elements will appear on the same line, as long as there's enough horizontal space.

Comments in HTML

Comments are a way to add notes to your HTML code that are not displayed in the browser. They're useful for explaining your code, leaving reminders for yourself, or temporarily disabling sections of code.

To add a comment in HTML, use the following syntax:

```
html
<!-- This is a comment. It will not be displayed in the browser. -->
```

Comments can span multiple lines:

```
html
<!--
  This is a multi-line comment.
  It can be used to explain complex sections of code.
-->
```