



Intro to HTML and the DOM

What is HTML?

HTML (**H**yper**T**ext **M**arkup **L**anguage) is the structure language of the web. It is a *markup language* that provides the structure for all the content on a web page.

HTML written in a text document with a file extension of .htm or .html. It is comprised of tags that will contain different kinds of page content. Web browsers read the HTML code and output the content onto a user's screen. The tags tell the browser what kind of content to display. Tags also inform search engines as well.

Let's write a very simple HTML page. This is the shortest, simplest, valid HTML document you can write.

```
1. <!DOCTYPE html>
2. <html lang="en">
3.   <head>
4.     <meta charset="utf-8"/>
5.     <title>Basic HTML Page</title>
6.   </head>
7.   <body>
8.     <h1>Hello, World! :D</h1>
9.   </body>
10. </html>
```

Semantic HTML

HTML should be both well structured and *semantic*. This means that your HTML tags should be meaningful in regards to the content they contain. Every HTML tag has a specific meaning and purpose, and should be used with it's appropriate kind of content. A comprehensive list of all HTML elements and their use can be found at [MDN HTML Element Reference](https://mdn.com/html-element-reference).



Here are some very common HTML tags that you will be using (for sure):

`<html>``<p>``<head>``<section>``<body>``<div>``<link>``<h1>` - `<h6>``<script>``<a>``` and `````<hr/>` and `
````<header>````<footer>``````<main>``<table>`, `<tr>`, `<td>``<nav>`

Pay extra special attention to proper semantic usage of the following tags, which you should use in your pages where applicable.

- `<header>` - This should contain the page header, including the navigation.
- `<footer>` - This should contain the page footer, or the footer for a specific section of the page (such as a `<blockquote>`), if applicable (rare).
- `<main>` - This contains the main, core page content.
- `<nav>` - This tag is used to house links that function as page navigation.
- `<section>` - This tag defines a section in the document, that is not `<main>`.
- `<article>` - This tag is for articles that can be republished or syndicated in other locations.
- `<aside>` - Use this tag to house content that is aside, or tangentially related to the main page content.



HTML tags can also contain additional data within them. These are called **attributes**, and they contain values and data that configure the HTML elements, adjust their behavior, or provide the browser with necessary information in order for the web page to function properly.

You **must** use specific, required attributes when using the <a>, , <script>, and <link> tags.

Images require a path to the image file (**src** attribute), and alternative text (**alt** attribute).

```

```

Links require the path or URL to link to (**href** attribute).

```
<a href="http://www.somepage.com">link to some page</a>
```

CSS links require a path or URL to the CSS file (**src** attribute), and a relationship pointer (**rel** attribute)

```
<link rel="stylesheet" type="text/css" src="../css/stylesheet.css"/>
```

JavaScript links require a path or URL to the file to grab (**src** attribute).

```
<script type="text/javascript" src="../js/custom.js"></script>
```

Block & Inline Elements

In regards to appearance, there are two kinds of HTML elements: **block** and **inline**.

Block elements will create a new line of content, and will take up all of the available space to the left and right of itself. They create a solid block of content. Block elements can only be used inside the <body> tag. Some block-level elements include:

- **<p>**
- **<div>**
- **** and ****
- **<header>**
- **<footer>**
- **<h1>** - **<h6>**
- ...and more.

Inline elements will only occupy the exact amount of space that their content spans, and do not start



- ``
- ``
- ``
- ``
- `<button>`
- ...and more.

HTML Comments

Recall that well commented code is reflective of good coding style. (See the [DDC Style Guide](#)) HTML comments are created using the following syntax:

```
<!-- this is a comment -->
```

HTML Tables

Tables are created using the `<table>` tag, and they require that you adhere to a specific nested structure. Please note that tables are used for displaying tabular data **only**, and **NOT for creating layouts**.

Example Table

Please Note: This table has been STYLED using Bootstrap classes for demo purposes. Plain HTML tables do not have any default styling, and must be styled using CSS.

Column 1 Heading	Column 2 Heading
content	content
content	content

Sample Code

```
1. <table>
2.   <tr>
3.     <th>Column 1 Heading</th>
4.     <th>Column 2 Heading</th>
5.   </tr>
6.   <tr>
7.     <td>content</td>
```



```
13. </tr>
14. </table>
```

Additional Reading: [MDN: HTML Table](#)

HTML DOM

Remember that a web page is simply an HTML file that can be viewed as a web page in a browser, or you can open it in a text editor or IDE like WebStorm and view the raw HTML code. Either way, it is the same document in both cases.

When an HTML file is loaded by a program such as a web browser a **Document Object Model**, or DOM is created. The **HTML Document Object Model (DOM)** is a hierarchical map a lot like a "family tree" that outlines the data structure in the file. It is used to represent (or **model**), all of the **objects** (the HTML elements and content) in the HTML **document** so a web page can be rendered to your screen.

Technically speaking, the HTML DOM is a *programming interface* for HTML!

Web browsers build a DOM "Tree" based on the HTML document. The **parent, child, and sibling** relationships between your HTML elements are modeled in the hierarchy of this "DOM Family Tree". Web browsers use this DOM to render your web page content properly; placing the images, paragraphs, text, tables, etc., where they belong based upon how they are laid out in the HTML code. The tags in the HTML document are the **objects** in the DOM Tree that the browser displays on the page.

With the help of attributes such as classes and ids added to our HTML, we can identify and target specific HTML elements using CSS and JavaScript the modify and interact with them. We can move through the elements of a web page and "traverse the DOM" using DevTools. The term "Traversing the DOM" is generally used to refer to moving through page elements using JavaScript in order to create interactive behavior.

Additional Reading: [MDN: Document Object Model](#)

