

HTML Basics

Tiffani Reardon

Chapter Objectives

Upon completion of this chapter, readers will be able to do the following:

1. Define HTML and identify what it stands for.
2. Build a simple web page using HTML coding.
 1. Identify and use common HTML tags.
 2. Explain and apply basic tag rules.
 3. Explain attributes and use them to stylize text and images.
 4. Embed videos and other embeddable items into HTML web pages.
3. Identify websites where readers can learn more and practice their HTML skills.

Introduction

There is a high probability that the readers of this chapter have at least heard the acronym HTML, although many may only know its most basic meaning: HTML is a code that makes web pages. You are not wrong, but there is a bit more to it. For example, did you know that HTML is actually a language with rules, just as English is? Or, perhaps, did you know that HTML actually stands for something, and is not just an all-caps name? Let's start from the beginning.

Before we begin our adventure into HTML, hover your mouse over this text. Any time you see bolded blue text, it means that there is a text-popper there. If you hover over it, you'll get more information about the indicated item.

HTML stands for HyperText Markup Language. HTML is a coded computer language that, when read by a web browser, displays the web pages you see every day when exploring the Internet.

Don't believe me? See for yourself: In Mozilla Firefox, right click on a web page and click "Inspect Element." The code of the page you are looking at should appear at the bottom of the page. This code is what Firefox reads to show you the page you are seeing in your browser. Pretty cool, huh?

HTML is the baseline coding of every web page you see out there, and when combined with CSS, JavaScript, and other languages, it can create cleanly-constructed and functional websites.

Tag Rules

HTML is a language, and it, as all languages do, has rules that make it make sense to the web browser. Just as the English language doesn't make sense when you break grammar rules, neither does HTML when you break tag rules. The difference is, most English speakers will still understand what you are trying to say when you break a grammar rule; the web browser will not understand when you break a tag rule. So let's look at some of these rules.

When you look at a line of HTML, you'll see information enclosed in brackets `< >` throughout the code. This denotes a tag. A tag is what tells web browsers what each item on the page is; it tells the web browser how to classify that information. For example, a paragraph would be classified by a `<p>` tag.

Rule #1

Most tags have an opening tag and a closing tag that surround the information it is classifying or effecting. An opening tag is just the brackets with the tag enclosed, like so: `<p>`. A closing tag is the same thing, except there is a forward slash at the front of the tag, like so: `</p>`. When tagging a paragraph with the `<p>` tag, it would look like the example.

`<p>`This is a paragraph about how to properly tag a paragraph. To tag a paragraph, you have to start with an opening paragraph tag, type the paragraph, and then end with a closing paragraph tag. This is how most sections of simple paragraph text are tagged.`</p>`

Rule #2

Rule #1 does not *always* apply. Not all tags have to have a closing tag, and you just have to remember those. We will learn about some basic tags in the next section, where we talk about a few that do not require a closing tag.

Rule #3

Opening and closing tags must be *nested* to work. That is, the last tag you open should be the first tag you close, as shown in the example.

```
<body>
```

```
<p>This is a paragraph in the body of the webpage. As you see, I opened the body tag first, and then the paragraph tag; if I am nesting the tags, then I have to close the paragraph tag first, and then close the body tag.</p>
```

```
</body>
```

Basic Tags

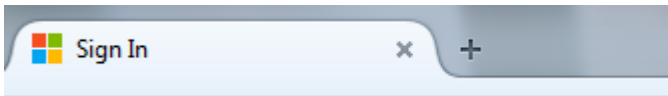
There are tons of tags, and it's almost impossible to memorize them all. Luckily, we have the Internet, which means we don't have to memorize them. In this section, you'll learn about some of the basic tags that are needed in almost every web page.

Beginning tags and the <head>

Every HTML web page starts with a **<!doctype html>** tag, with no closing tag. This is what tells the web browser what version of HTML the page is written in. The **<!doctype>** tag used to have several options in the old HTML, but in HTML5, there is only one: **<!doctype html>**.

After the **<!doctype html>** tag, there is an **<html>** tag, which is also required for every HTML document. Unlike the **<!doctype html>** tag, the **<html>** tag does require a closing tag of **</html>**. For nesting purposes, the closing **</html>** tag will always be the very last thing in your document.

Once the **<html>** tag has been opened, the page then has its head information in the **<head>** tag, which also has a closing tag of **</head>**. The **<head>** tag includes information that does not show up on the page, but is vital to the look and feel of the page. There is always a **<title>** tag with a closing **</title>** tag that encloses the information on the tab of the web browser. Users who are including CSS information in their document might also include CSS coding in the **<head>** tag.



The <body> tag and other common tags

After the **<head>** tag has been closed, the next tag to open is the **<body>** tag. This denotes the information that you actually see on the page when looking at a web browser. Everything you see within the browser window appears after the **<body>** tag and before the closing **</body>** tag.

For common tags used within the **<body>** tag and their purpose, see Table 1. Unless otherwise noted, all tags in the table below require a closing tag.

Table 1: Common HTML tags

Tag	Function

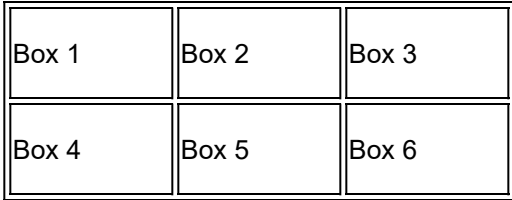
<code></code>	Defines emphasized text. Text enclosed in an <code></code> tag will appear italicized.
<code></code>	Defines important text. Text enclosed in a <code></code> tag will appear boldface.
<code></code>	Defines an image. This tag requires an attribute that defines what image it shows. This attribute is called src . The <code></code> tag does not require a closing tag, however it does end differently. Instead of the usual closing tag, an <code></code> tag is closed by ending the opening tag with a space and then <code>/></code> . For example: <code></code>
<code><a></code>	Defines a link. This tag requires an attribute that defines where the link goes to. This attribute is called href .
<code></code>	Defines an unordered list. This will start a bulleted list.
<code></code>	Defines an ordered list. This will start a numbered list.
<code></code>	Defines a line in a list. This tag applies to both unordered and ordered lists. When you start a list, you must use <code></code> tags to open and close each line separately. See Figure 4 for an example of how a list looks in HTML.
<code><table></code>	Defines a table. This will start a table.
<code><tr></code>	Defines a table row. This will start a row in a table. You must have a <code><tr></code> tag for each row of a table.
<code><td></code>	Defines a table column. This will start a column in a row of a table. You must have a new <code><td></code> tag for each column of each row of the table. See Figure 5 for an example of how a table looks in HTML.
<code>
</code>	Defines a line break. This tag does not require a closing tag.
<code><h1>- <h6></code>	Defines headings. There are six HTML heading levels: <code><h1></code> , <code><h2></code> , <code><h3></code> , <code><h4></code> , <code><h5></code> , and <code><h6></code> . They each define headings of their specified order (<code><h1></code> being the first order heading and <code><h6></code> the sixth order).
<code><p></code>	Defines a paragraph. This is how most sections of text are tagged.

Figure 4: Ordered and Unordered Lists

HTML	Preview
<pre> Item 1</pre>	<ul style="list-style-type: none"> • Item 1 • Item 2 • Item 3

<pre>Item 2 Item 3 </pre>	
<pre> Item 1 Item 2 Item 3 </pre>	<ol style="list-style-type: none"> 1. Item 1 2. Item 2 3. Item 3

Figure 5: Tables

HTML	Preview
<pre><table> <tr> <td>Box 1</td> <td>Box 2</td> <td>Box 3</td> </tr> <tr> <td>Box 4</td> <td>Box 5</td> <td>Box 6</td> </tr> </table></pre>	

Basic Attributes

HTML attributes provide more information for tags or apply stylistic features to them that are not the default. For example, an `<a>` tag requires more information to know where it should link to, and a `<p>` tag can have a different font color than the default if you want it to. Let's take a look at some common attributes and what they do in Table 2 below.

Table 2: Common HTML attributes

Attribute	What it does	What it looks like in HTML

href	The href attribute belongs to the <a> tag and defines where the link will go to.	<code>This is a link to the textbook website.</code>
src	The src attribute belongs to the tag and defines where the tag will pull the image from. This can be an image site or a file path from within the server.	<code></code>
width	The width attribute can be applied to images, iframe elements, tables, or other elements with a numerical width. It defines how wide the item will be.	<code></code>
height	The height attribute can be applied to images, iframe elements, tables, or other elements	<code></code>

	with a numerical height. It defines how tall the item will be.	
style	<p>The style attribute allows you to add inline CSS elements to your tag. A few common ones are color, font-size, font-family, and text-align. You define a style element the same way as other attributes, however you then have to define the value of the style element. View the examples to the right to see how these style elements are defined.</p>	<pre><p style="color:blue;"> This is blue text.</p> <h1 style="font-size:24px;">This is a 24pt font heading.</h1> This link to the textbook website is Arial 12 pt font. <h2 style="text-align:center;">This heading is centered on the page.</h2></pre>

Embeddable Items

YouTube is the most popular video repository in America, if not the world. What if you want those videos on your web page? Perhaps you posted a video of a project you've been working on, and you want to show that video on your online portfolio. The answer is simple: embed codes. Most sites with hard-to-code items have an embeddable feature to them, and it essentially puts the information in a frame of one website into a frame on your own website. You can usually find the embed codes in the share functions of items. The nice (and easy!) thing about embed codes is that they don't require you to add anything to them. You simply paste the code into your HTML page, and in that spot you will see the video or other item when you preview it. However, you can edit them. Let's analyze a YouTube embed code in Figure 6 below. Just hover over the words to see the different comments.

```
<iframe width="420" height="315" src="https://www.youtube.com/embed/zx_M-sWrDfc" frameborder="0" allowfullscreen></iframe>
```

Watch the video below to see tags, attributes, and embed codes in action!

Organizing Your Web Page

Web pages are a series of pages connected by hyperlinks, and they have unique needs. Let's think about it this way: you have a small child whom you keep in a playpen. In the playpen is a tiny toy box with a ball, a block, and a fire truck. You play a game where you tell her to show you the ball, and she points to the ball. You tell her to show you the block, and she points to the block. You tell her to point to the fire truck, and she points to the fire truck. This game works very well.

But then you take her into another room and put her in another playpen, and there's a pillow, a rattle, and a dolly. You tell her to point to the fire truck, and she looks around and shrugs her shoulders. Her mind is not advanced enough to understand that you wish her to point through the wall into the other playpen. And she does not have the physical ability to get out of the playpen, walk to the other room, and retrieve your fire truck for you. This game will not work.

You then try again. You put her back in the original playpen, but you take the small box of toys and scatter them around the house. You put the ball under the couch. You put the block in the bathtub. And you put the fire truck in the trunk of your car. You return to the playpen and ask the child to show you the fire truck. Again, she shrugs her shoulders. This game will not work.

Why does the first game work, but the other two do not?

What if you change the game by taking the child, playpen, toy box, and all into the next room and again ask the child for the block? Unless something distracts the child, the game will work! The child knows the block is in the box, and all functions as normal.

Your web page is the child, and its name is index. Your main folder is the playpen, and your subfolders are the toy boxes.

To be more clear, when you make the homepage of a website, you'll want to name it "index.html" and put it in a folder. In that folder, you can then put any pictures, documents, videos, or other HTML files that you might link to. You can also organize further by putting those things into subfolders. But no matter what, everything that is not already online must be put inside that main folder to be then posted online.

Where to Practice and Learn More HTML Skills

- Codecademy
- W3Schools
- Lynda