3-1-2016

41. Introduction to HTML

Tiffani Reardon
Kennesaw State University, treardo2@kennesaw.edu

Follow this and additional works at: http://digitalcommons.kennesaw.edu/oertechcomm
Part of the Technical and Professional Writing Commons

Recommended Citation
Reardon, Tiffani, "41. Introduction to HTML" (2016). Sexy Technical Communications. 41.
http://digitalcommons.kennesaw.edu/oertechcomm/41

This Article is brought to you for free and open access by the Open Educational Resources at DigitalCommons@Kennesaw State University. It has been accepted for inclusion in Sexy Technical Communications by an authorized administrator of DigitalCommons@Kennesaw State University. For more information, please contact digitalcommons@kennesaw.edu.
Chapter Goals and Objectives

At the end of this chapter and the practice activities, readers will be able to:

- Goal 1: define HTML and identify what it stands for
- Goal 2: build a simple web page using HTML coding
  - Objective 1: identify and use common HTML tags
  - Objective 2: explain and apply basic tag rules
  - Objective 3: explain attributes and use them to stylize text and images
  - Objective 4: embed videos and other embeddable items into HTML web pages
- Goal 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 in Figure 1.

**Figure 1: Opening and closing tags**

<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 in Figure 2.

**Figure 2: Nested tags**

```html
<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>
```
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, as shown in Figure 3. Users who are including CSS information in their document might also include CSS coding in the `<head>` tag.

Figure 3: A browser tab with title information

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

**Figure 4: Ordered and Unordered Lists**

<table>
<thead>
<tr>
<th>HTML</th>
<th>Preview</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;ul&gt;</code></td>
<td></td>
</tr>
</tbody>
</table>
Figure 5: Tables

<table>
<thead>
<tr>
<th>HTML</th>
<th>Preview</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;table&gt;</td>
<td></td>
</tr>
<tr>
<td>&lt;tr&gt;</td>
<td></td>
</tr>
<tr>
<td>&lt;td&gt;Box 1&lt;/td&gt;</td>
<td>Box 1</td>
</tr>
<tr>
<td>&lt;td&gt;Box 2&lt;/td&gt;</td>
<td>Box 2</td>
</tr>
<tr>
<td>&lt;td&gt;Box 3&lt;/td&gt;</td>
<td>Box 3</td>
</tr>
<tr>
<td>&lt;/tr&gt;</td>
<td></td>
</tr>
<tr>
<td>&lt;tr&gt;</td>
<td></td>
</tr>
<tr>
<td>&lt;td&gt;Box 4&lt;/td&gt;</td>
<td>Box 4</td>
</tr>
<tr>
<td>&lt;td&gt;Box 5&lt;/td&gt;</td>
<td>Box 5</td>
</tr>
<tr>
<td>&lt;td&gt;Box 6&lt;/td&gt;</td>
<td>Box 6</td>
</tr>
<tr>
<td>&lt;/tr&gt;</td>
<td></td>
</tr>
<tr>
<td>&lt;/table&gt;</td>
<td></td>
</tr>
</tbody>
</table>
Drag the items from the bottom to the slots on the right.
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>
<thead>
<tr>
<th>Attribute</th>
<th>What it does</th>
<th>What it looks like in HTML</th>
</tr>
</thead>
<tbody>
<tr>
<td>href</td>
<td>The <code>href</code> attribute belongs to the <code>&lt;a&gt;</code> tag and defines where the link will go to.</td>
<td><code>&lt;a href=&quot;http://distanceed.hss.kennesaw.edu/technicalcommunication&quot;&gt;This is a link to the textbook website.&lt;/a&gt;</code></td>
</tr>
<tr>
<td>src</td>
<td>The <code>src</code> attribute belongs to the <code>&lt;img&gt;</code> 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.</td>
<td><code>&lt;img src=&quot;logo.jpg&quot; /&gt;</code></td>
</tr>
<tr>
<td>Attribute</td>
<td>Description</td>
<td>Example</td>
</tr>
<tr>
<td>-----------</td>
<td>-------------</td>
<td>---------</td>
</tr>
<tr>
<td><strong>width</strong></td>
<td>The attribute can be applied to images, <code>iframe</code> elements, tables, or other elements with a numerical width. It defines how wide the item will be.</td>
<td><code>&lt;img src=&quot;logo.jpg&quot; width=&quot;300px&quot; /&gt;</code></td>
</tr>
<tr>
<td><strong>height</strong></td>
<td>The attribute can be applied to images, <code>iframe</code> elements, tables, or other elements with a numerical height. It defines how tall the item will be.</td>
<td><code>&lt;img src=&quot;logo.jpg&quot; width=&quot;300px&quot; height=&quot;300px&quot; /&gt;</code></td>
</tr>
<tr>
<td><strong>style</strong></td>
<td>The attribute allows you to add inline CSS elements to your</td>
<td></td>
</tr>
</tbody>
</table>
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.

```html
<p style="color:blue;"> This is blue text. </p>
<h1 style="font-size:24px;">This is a 24pt font heading. </h1>
<a href="http://distanceed.hss.kennesaw.edu/technicalcommunication" style="font-family:Arial; font-size:12px;">This link to the textbook website is Arial 12 pt font. </a>
<h2 style="text-align:center;">This heading is centered on the page.</h2>
```

**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.

**Figure 6: A YouTube embed code**

```html
<iframe width="420" height="315" src="https://www.youtube.com/embed/zx_MsWrDfc" 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.

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?

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.
If I link to a picture on my desktop, the link will work when it gets posted online.

- True
- False

Can you see this?

See? I told you it's a language.

Control + Click, if you are using an Apple computer.

CSS stands for Cascading Style Sheet this language is an advanced styling language that can be written into the HTML document or linked to from a separate CSS document.

JavaScript is a high-complexity programming language that makes the more complicated things you see on some websites, like games.
It is common to separate each tag by a line to help the coder keep organized, however it is not required. I am separating them so that you can see the layers and nesting more clearly in this chapter.

The last version of HTML was HTML 4.01. Our current version is HTML5.

We will not study CSS in this textbook, however you can learn about it at w3Schools or Codecademy, linked at the end of this chapter.

By this, I mean each square of the table.

Notice how all of the information enclosed in this first table row tag is part of the first row in the table.

Notice how the information in this first table column tag of the second table row tag is in the first box of the second row of the table.

You'll learn about iframe elements in the next section on embeddable items.

Notice that if you need to use more than one attribute, you simply separate them with a space. There is no limit to how many attributes you use.

Notice how the base of the style attribute is just link other attributes. It starts with the attribute, followed by an equal sign, then in quotations is the value.

Inside the quotations, you then have to define the value. Start with the style element, followed by a colon, then the value, and a semicolon. No spaces.

If you want more than one style element, put them all inside the quotations, and separate them each with a space.

Most embed codes use the iframe tag now, but not all of them. It depends on the website and what kind of code was used in creating the item.
You can edit the width and height of embed codes, or add them in if it is not already in the code. This is good if the default size is not quite big or small enough for what you want to use it for. Just change the number in the quotations to change it.

The source of embed codes are usually some variation of the link to the item. If you put this link in your web address, it might not work (it also might just give you a full-screen version of the video). But if you take out the "embed/" part of it and replaced it with "watch?v=", it would take you to a video of my dog, Nutmeg, running around in her backyard on YouTube. Go ahead, try it.