Chapter 3: About HTML5

About HTML5

In this chapter, we'll discuss what HTML5 is all about. Hopefully this will be a review or refresher for most of you. But the concepts and terminology are very important, so it all bears repeating to make sure that we're all speaking the same language as we proceed through the course.

HTML stands for *Hypertext Markup Language*. Its origins go back to 1991 when developers designed it as the language for authoring Web pages for the Internet's World Wide Web.

Like all software, HTML has evolved over the years to keep up with advances in user agents. The phrase *user agents* refers collectively to all the different gadgets and programs that can access Web pages. That includes Web browsers, tablets, smartphones, assistive devices, and search engine indexing spiders. Here's a look at major versions of HTML released over the years:

Version	Year
HTML 1	1991
HTML 2	1995
HTML 3	1996
HTML 4	1997
XHTML 1	2000
XHTML 1.1	2001
HTML5	Candidate

HTML Versions

You'll notice the name changed from HTML to XHTML at the turn of the century, and the numbering went back to 1. However, *XHTML* is still *HTML*. They just added the *X* because that version borrowed some syntax conventions from the highly successful XML language. XML is a language for storing and transmitting raw data rather than published pages. You don't need to know, learn, or understand XML to create websites, so you don't have to even give that a second thought. Suffice it to say *XHTML* is just *HTML* with an *X* in front and some minor differences in how you type a few tags.

Shortly after the turn of the century, another Internet group called *WHATWG* (*Web Hypertext Application Technology Working Group*) began working on a version of HTML that would extend the language to include features you'd commonly find in application programs (apps), which tend to be more interactive than Web pages. All the different names and acronyms were becoming too confusing for most developers. So the World Wide Web Consortium (W3C), the standards body that creates all the Web languages, decided it was time to merge the WHATWG group with the XHTML group and go back to a single, standard name and numbering system. And from that decision, the current version of HTML, dubbed *HTML5*, was born. HTML5 is simply the version of HTML/XHTML that everyone will be using for the next decade or so. You don't need to worry about choosing a specific version for a specific project. It doesn't work that way. Each new version is a refinement of (and a replacement for) the previous version.

The World Wide Web Consortium is made up of members from high-tech giants like IBM, Microsoft, Apple, Google, and others. HTML5 *is* whatever the W3C says it is. There are no exceptions—no "different flavors" for different browsers, editors, or operating systems. But the W3C doesn't simply create a language in a vacuum and then force it on the rest of the world. Creating a new version of any language is a process that takes many years and many people outside the W3C.

Part of the process of defining a new version of a language involves putting it out to the public as a *candidate* recommendation or *proposed* recommendation when the W3C is reasonably confident that the spec is good and will work for everyone. But it's not an official or final recommendation. It's more of a "let's let everyone take it for a spin and kick the tires" sort of trial run that could last several years. There's still a chance that they might make some changes, based on the feedback from folks who are using the product during that trial period.

Right now, HTML5 is such a candidate spec, not a final spec. But at this stage, any changes are likely to be so minor as to be inconsequential to most folks. So development with HTML5 is well under way in the real world. And if you're going to learn HTML, Version 5 is definitely the way to go. And that's why you'll be learning HTML5 in this course.

HTML Tags

HTML is a language for creating Web pages and other electronic documents (including some kinds of applications). People who create apps, Web pages, and e-books often refer to what they do as *writing code*. The term *code* is a general term for writing any kind of computer software in any computer language. You'll be seeing the term *code* a lot in this course, because you'll be learning how to write CSS and HTML code throughout this course.

Like all of its predecessors, HTML5 consists primarily of *tags*. Each tag tends to define an element type, such as a heading, paragraph, list item, table, graphic image, or something else that shows in the document. Every tag starts with < and ends with > with one or more lowercase characters in between —usually an abbreviation for the type of element the tag creates.

Many (though not all) tags come in pairs where an *opening tag* marks the start of the element, and a *closing tag* marks the end of the element. The closing tag is the same as the opening tag, except that it contains a slash (/) right after the < symbol.

For example, the tag marks the start of a paragraph, and marks the end of a paragraph using the syntax below.

```
<p>...</p>
```

Above, the ellipsis (...) is just a placeholder. In real life, you'll put the words and sentences that make up the paragraph content between the tags. Those words and sentences, of course, can be any words and sentences you like. There's no limit to how much stuff (words, sentences) you can put between any opening and closing tag pair.

```
<p>This is a sample paragraph. A paragraph can contain any number of words and sentences.</p>
```

Not all tags come in pairs, though. Some tags stand alone. We commonly refer to such tags as *empty tags*. For example, we use the br tag to indicate a line break. In XHTML, all empty tags end with /> usually preceded by a space. For example, here's the proper way to type a br tag in XHTML:

```
<br />
```

Older versions of HTML didn't require the space or slash. In traditional HTML, the br tag is written as:

```
<br>
```

One of the goals of HTML5 is to do away with some of the unnecessary confusion and complication that comes with having too many brands on versions of the HTML language out there. Another is to make sure that bringing forth this new language doesn't break any of the pages that are already out there. So the plan right now is for HTML5 to accept and support either method of writing empty tags. So the space and slash at the end of an empty tag are optional.

Tip

XHTML has been around for over 10 years now, and many developers are accustomed to writing code that way. So you'll see plenty of people using a / in empty tags, even though it's optional, and to the beginner might seem like extra unnecessary effort.

HTML Attributes

In addition to tags, the HTML language consists of *attributes*. You can use attributes inside some tags to provide additional information about the element that the tag is creating. All attributes follow the syntax:

name="*value*"

Where *name* is an attribute name and *value* is some number or other information that's assigned to that attribute. You should always precede each attribute with a space, and you should always enclose the value in quotation marks. For example, here's an img (image) tag with two attributes, a src (source) attribute and an alt (alternate text) attribute. Don't worry about the specifics of the tag and attribute right now, though. All that's important now is syntax. Notice how there's a space before each attribute name, and the value assigned to each attribute is in quotation marks.

Text equivalent start.

Label	Tag Part
Start of tag	<
Attributes	src and alt
Values	sample.jpg and sample
End of tag	/>

Anatomy of <img src="sample.jpg" alt="sample">

Text equivalent stop.

Since img is an empty tag, we can also write it without the space and slash at the end in HTML5. In other words, the tag below is also a valid way to write the one above. It doesn't really matter which you use, since both syntaxes are equal and acceptable in HTML5, and the picture that the tag shows will look exactly the same in the Web page either way.

```
<img src="sample.jpg" alt="sample">
```

Tags that come in pairs can also use attributes. The syntax is the same, and you don't have to do anything to "close" the attributes in the closing tag. For example, the <a> tag below contains two attributes with values. The closing tag is fine the way it is. You don't need to "close" the attributes or mention them at all in the closing tags.

```
<a href="sample.htm" title="sample">

...

</a>
```

In all attributes, spaces around the equal sign are allowed but not required. If you find the code a little easier to read with spaces around the equal signs, that's fine. The code will work with or without the spaces.

```
<a href = "sample.htm" title = "sample">
```

There are about 100 tags and about 100 attributes in HTML5. That may seem like a lot, but there's no need to learn them all at once. You start with the most important and most commonly used ones and work your way up from there. You won't have to memorize them all either, because it's easy to find lists online or printed cheat sheets, as you'll learn later. For now, let's start with a few important tags—the tags that are required in all Web pages.

Note

There's no such thing as a *command* in any version of HTML. However, people occasionally misuse that term as a synonym for *tag* or *attribute*.

Required Tags

Every HTML5 document must contain some required tags that the various user agents use to identify components of the document. Here are those required tags . . . you must use them in this order.

```
<!DOCTYPE html>
<html>
 <head>
 <title></title>
 </head>
 <body>
 </body>
</html>
```

There are no required attributes, so that's why you don't see any attributes in this list.

Let's quickly summarize the purpose of each required tag:

<!DOCTYPE html>: This tag indicates that the page is written in HTML5. It may seem odd that there's no *5* in there. But doctype tags for other versions of HTML are much different and far more complex. So that simple DOCTYPE tag is sufficient to identify an HTML5 document.

<html>...</html>: The <html> tag marks the start of the HTML document, and </html> marks the end of the document.

<head>...</head>: The <head> tag marks the beginning of the head section (or *metadata* section) of the document. This section contains tags and information about the document rather than content that actually shows on the page. The </head> section marks the end of that section.

<title>...</title>: This is the document title that appears in the Web browser's title bar or tabs and in search engine results. You must place the actual text of the title between the <title> and </title> tags and keep it brief (six words or fewer).

<body>...</body>: The body section contains the actual content of the page that's visible in the Web browser. The <body> tag marks the start of that section, and </body> marks the end of that section.

Later, in the assignment for this lesson, you'll have a chance to go hands-on and create a page that contains all the required tags. You'll learn to use that page as a template for every new page you create so you don't have to memorize or retype those tags every time you create a new Web page. But first, let's head over to Chapter 4 and talk about CSS, which is the styling component of modern Web page design.