Beginning Web Development

So you've been surfing the net, and you have an idea for a website. Or maybe you want to get a job designing and creating web pages. Whatever reason you have for learning **HTML**, welcome. Creating a website like Facebook or Twitter is far more than coding in **HTML**, but you have to start somewhere, and **HTML** is a good place to begin. Everything else you will learn about web development will be built on your understanding of **HTML**.

What *is* HTML Anyway?

I'm not going to go into a lot of details about the history of **HTML**. You can find that information just about anywhere. Here is a quick overview:

HTML stands for **H**yper**T**ext **M**arkup **L**anguage. **HTML** is a specialized subset of **SGML** (**S**tandard **G**eneralized **M**arkup **L**anguage) and is related to **XML** (e**X**tensible **M**arkup **L**anguage). **HTML** was created in 1996 and is now on its fifth major revision.

HTML uses tags and attributes to define the structure of a document. These documents are interpreted by a web browser such as Internet Explorer, Firefox, Safari, or Edge.

HyperText refers to the links that connect a web page to another page on the same site or somewhere else on the web.

Markup refers to the tags that are used to define a web page's structure.

HTML is based on **XML**, e**X**tended **M**arkup **L**anguage, and shares many characteristics with it. (But you don't need to know **XML** to use **HTML**!)

Got it, but what is CSS?

CSS stands for **C**ascading **S**tyle**S**heets and refers to the markup that defines the fonts and layout of an **HTML** page. The styles you create can control the fonts, font sizes, colors, and page layout of your website.

Why bother with **CSS**? It's possible to apply font and color information to an **HTML** element. At least, it is for now. But the thing is, suppose that you decide that rather than bright green, you'd like to use orange for links? If you don't use **CSS**, you have to work your way through your website and change all of those explicit definitions.

Why Learn HTML and CSS?

Wouldn't it be easier to use a website generator or a blog application like **WordPress**?

Well, yes, it would be easier. You could use a website generator or a content management system like WordPress to do most of the work for you, but there are still advantages to understanding how HTML works. To me, the disadvantage of using a predefined solution, whether it is a blogging application or a website generator, is that your site will tend to look a lot like every other site that was created by that generator. If

you learn **HTML**, **CSS**, and maybe a little **JavaScript**, your imagination is less limited by the tools you are using.

Even if you decide to use an application or site generating tool after you have learned **HTML** and **CSS**, you will be able to customize the results much more easily if you understand what needs to be changed to achieve your desired result.

What you Need

Any time you teach, you must make some assumptions about your students. These are my assumptions about you:

- you know how to surf the 'net
- you have no prior experience with HTML
- you don't need to be a computer programmer
- you know how to create, edit, and save files

That's it! It doesn't matter whether you want to learn HTML because you must or because you think it would be fun. (I think it is but I'm a computer geek.)

There are two subjects you need to learn, no three, no four—yes, four subjects you need to learn to make attractive websites:

1. **HTML** - **H**yper**T**ext **M**arkup **L**anguage, the language of the web

2. **CSS** - **C**ascading **S**tyle**S**heets, supplies the "look and feel" of web pages. (Covered in Volume 2 of this series, but we'll take a quick sneak peek at it in this book.)

3. **JavaScript** - This isn't required but knowing a little programming can help you make your web pages more interactive and interesting.

4. It helps if you know a little about creating graphics and photo editing. Again, not required but knowing how to work with images will make your web pages more attractive.

There are many other languages and technologies that you may want to learn in the future, but you can do an awful lot with the four in the above list. As I said, you have to begin somewhere so let's get stared.

To follow along with these tutorials, you need:

1. **A computer**
   You can create HTML files using a computer that runs on Windows, OS X, Linux, or any other operating system that may come along in the future. If your computer is powerful enough to write email and surf the web, you can probably use it to make web pages too.

2. **A text editor**
   Windows comes with notepad.exe, the Mac comes with TextEdit, and Linux systems use an application called Vim. If you have another editing tool you like, feel free to use it. However, if you use a Word Processor (like Microsoft Word) you must be very careful that you save your file as plain text.
3. **A web browser**
   You will want to see how your web pages look, after all. If you are serious about doing web development, you might want to consider downloading multiple browsers (such Firefox, Safari, and Opera in addition to whatever is the standard web browser for your operating system) so you can see how your pages will look to visitors using different software. Every web browser has its own quirks when it comes to displaying HTML and you want to make sure that whoever visits your website they will have the same experience or, at least, a comparable experience.

That's it. For simple HTML files, you don't need anything more and even when we begin using CSS to style our web pages, you can still use a simple text editor.

Let's Start with Some HTML

The source for a page on a website a file with the extension *.htm* or *.html* whose content is made up of plain text.

Your First Web Page

Let's get started by creating our first web page. Then we'll go back and I'll explain what all that markup stuff means. Are you ready?

Open your favorite text editor and enter the following text:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8" />
    <title>My First Web Page</title>
  </head>
  <body>
    <h1>Welcome to the World Wide Web!</h1>
    <p>This is my first ever web page.</p>
    <address>
      <!-- Replace the following -->
      Your Name<br>
      you@example.com
    </address>
  </body>
</html>
```

Save the file into a folder on your hard drive. A good choice for the location of the file on Microsoft Windows is a new folder in the *"My Documents"* folder. Perhaps you could call it "MyFirst.html" or something similar. Give yourself a round of applause. You have created your first web page.

NOTE: Mac users can create a directory under the `\Users\UserName\Documents` and Linux users can create a new directory under `/home/UserName/Documents`.

Now open the file in whatever web browser you use. In most cases, *Ctrl+O* (*Command+O* on a Mac) allows you to browse to a file on your local computer and open it. Your web page should look something like Figure 2. *(NOTE: I only included the actual page contents here to save space.)*

Figure 1 - Your first HTML file in the browser.

What does it all mean?

Let's go through the contents of your first HTML page bit by bit. The first line tells the browser that this document will be using HTML 5. Earlier versions of HTML required a more complicated DOCTYPE tag but unless you have a compelling reason to use an earlier version of HTML, use this DOCTYPE and HTML 5. By including this line, we tell the browser that it doesn't have to try to work around the quirky behaviors that were part of earlier versions of HTML.

The second and last line of the document work together to define a container for the rest of the page. `<html lang="en">` is the opening tag. The part that follows "html" is an *attribute* that tells the browser that the language for this page is English. Other possibilities include: fr (French), de (German), it (Italian), and es (Spanish). If you want to know more about the lang attribute, you can check out the definition on the w3c website, Specifying the language of content: the lang attribute (https://www.w3.org/TR/html5/dom.html#attr-lang).

Next, in lines 3 to 6, we define the head section of the document. The head section contains *metadata* for the current page. (Metadata is data *about* the page, as opposed to data that will be displayed in the browser.) In our example, I have included two bits of information. The tag on line 4 tells the web browser how to interpret the text on the page. The character set used in this case is "utf-8". If you include the character set information, it should be within the first 1,024 characters of the page definition. The best way to ensure that is to place it first in the head section of the page, right after the opening `<head>` tag.

The title tag, which is the only *required* child tag for the head section of a web page. The text between `<title>` and `</title>` will usually be displayed either in the title bar of the web browser or the tab for the page.

In case you're curious, UTF stands for Unicode Transformation Format and the 8 means that each character in a string is represented by 8 bits or one byte.

The next section, the body, is where all the visible content of the page is placed. In the case of our sample document, we have a few bits of information. We have a heading, the information between `<h1>` and `</h1>`, a paragraph, and finally a section that contains my name and email address.

The `<address>` tag is usually formatted as italic text, but that depends, of course, on what formatting the web designer may have applied using CSS, but also on how the browser you are using chooses to interpret the tag.

Notice that each line of the address ends with `<br />`? That tag indicates that the browser should begin a new line.

Browsers and white space -- When the web browser interprets an HTML page, it ignores whitespace. Characters such as spaces and carriage return/line feed pairs are removed. You cannot format text using extra spaces to produce indentations, or extra paragraph marks to create new lines.

*If you want a new line, you have to use the
 tag. We'll talk about spaces later when we get to character entities.*

This would be a good time to play around a little bit. See what you can do with the few tags I've already shown you. It won't be fancy, but the goal is to be able to type any one of the tags you learn from this book without having to look up what it means or how to use it.

Working with Tags

Now that we have created a page, let's back up a bit and talk about some of the most commonly used tags. Tags define HTML elements.

Creating Paragraphs

The first tag you need to know how to use is the paragraph element, defined by the `<p>` tag.

With the tags I have shown you so far, you could actually create a web page. It would be boring to look at, but it would be a page that people could read. Let's add a little bit of interest. I'm going to give you a few more tags to play with. And that is exactly what you should do, by the way, *PLAY*!

Headings

HTML defines headings with levels from one to six. You use the heading tags the same way as the paragraph tag. The heading tags look like this:

```
<h1>This is heading level 1</h1>
<h2>This is heading level 2</h2>
<h3>This is heading level 3</h3>
<h4>This is heading level 4</h4>
<h5>This is heading level 5</h5>
<h6>This is heading level 6</h6>
```

If you were to look at that code as it is rendered in the browser, it would look something like this:

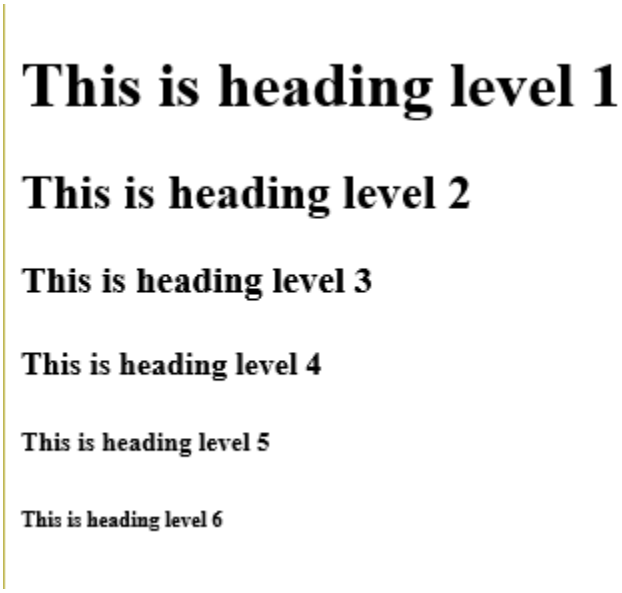

Figure 2: HTML Heading Tags in the browser.

To tell the truth, I've seldom gone any further than heading level three, but all six of them are there for your use. Remember that the look of each of these headings can be modified through the use of CSS. What we are looking at here is the default formatting supplied by my browser.

Text formats

Here are some tags you can use to control the look of your text. These tags are used on sections of text within a paragraph or other container. So far you've seen the heading tags and paragraph tags; we'll talk about other containers later.

Here are the text formatting tags:

**** - The em tag defines text that should be emphasized in some way. You might use it to set off a technical term, for example. Most browsers will render text between the `<em>` and `</em>` tag using italics, but once we start working with CSS, you'll see that you can change that to suit the content of the page.

**** - The strong tag indicates text that should stand out. In most cases the text marked as strong will be display in boldface type but this isn't necessarily so because, as with , you can define other formatting for this tag using CSS. All of the text between the opening and the closing will be formatted so that it stands out from the surrounding text.

In case you are wondering why I didn't tell you about the <i> or <b> tags, that is because HTML 5 is all about meaning rather than formatting. In other words, <i> used to indicate that a run of text should be formatted in italics, while <b> indicated that the text should be bold.

The , emphasis, and , strong, tags indicate that text should be made to stand out from the rest of the surrounding text without indicated what formatting should be applied to accomplish that task.

<small> - Text formatted with the small tag is rendered in a smaller font than the surrounding text. It can be useful for a subtitle, or for information that is less important than the surrounding text.

<cite> - Used to mark the name of a work, such as a song, movie, book, or play. This text is usually rendered as italic.

That's about it for chapter one.

In the next chapter, we'll talk about using pictures and links in your web page.

Let's finish by creating a slightly more interesting web page. Let's create a page about you! Once you've entered the following text, save it. The name "aboutme.html" is a good choice for this one. Don't forget to replace my placeholder text with your own information! I've set off the text you need to change with { and }, the curly braces.

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>About {Your Name}</title>
  </head>
  <body>
    <h1>All About {Your Name} <small>{your email}</small></h1>
    <p>Hi, my name is <strong>{Your first name}</strong> and I have created
       this website to show my skills with HTML. I am a {your job title} and
       am learning HTML in order to expand my skills.</p>
    <h2>My Family</h2>
    <p>{Tell us about your family here.}</p>
    <h2>My Job</h2>
    <p>{Put stuff about your job here!}</p>
  </body>
</html>
```

For example, look at Figure 3 to see my “About Irene” page. Notice that the text between <small> and </small> is slightly smaller than the rest of the line. Also notice the extra space between the headings and the paragraphs. That can be changed. When you get to volume two, CSS for Fun and Profit, you will learn exactly how to do that.

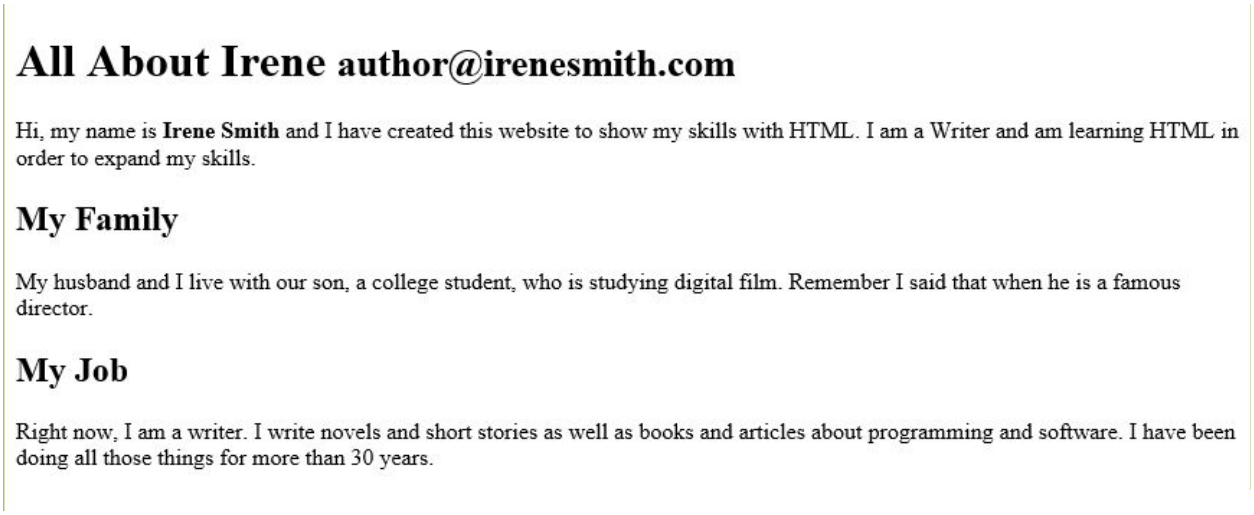

Figure 3: My "About Irene" page shown in the Edge browser.

Conclusion

So, you’ve seen how to create a web page and how to use a handful of HTML tags. In coming chapters, well expand your knowledge of tags, and show you a little bit about how to make your web pages more attractive.