

Beyond Rote - Guide Section 2

The last time, you memorized the basic elements of a well-formed web page. This part of the guide will try to explain what you memorized. Fit this new information into your brain wherever you think it works best to help you move forward.

```
<!doctype html>
<html>
<head>
<charset="utf-8">
</head>
<body>
Hello World!
</body>
</html>
```

Line by line

```
<!doctype html>
```

The first line of the template identifies the file as being coded as HTML5. Later, when you examine the pages written by others, you will see some other ways the coding is identified. HTML stands for "Hypertext Markup Language" and was initially designed by Tim Berners-Lee in the early 1990s. Earlier versions of HTML followed most of the same rules, but HTML version 5 is the most modern and powerful.

The individual lines in this file are mostly "tags". Each word in brackets like `<html>` or `<head>` is an instruction to the web browser software (Internet Explorer, Firefox, Chrome, etc.) and directs the browser to do something. Each tag means something different to the browser. The browser interprets your tags to display a web page.

Some tags stand alone. Others must come in pairs, one to open a task and the other to close it. The tag `<html>` is the beginning tag of a pair. It tells the browser that what follows is a regular web page, up until the closing `</html>` tag. The closing one effectively says: "This page is done."

(There are other bracket options like `[]` and `{ }`, for example, but don't use them here.)

```
<html>
```

HTML is not the only coding option for the web, just the most well-known. HTML may be written in capital letters like the acronym/abbreviation it is, but many times it is written in lower case and nobody complains. For your actual web coding, **always** use lower case. You could run into trouble if you did <HTML> to open and <html> to close. They don't match. Tags which come in pairs must match.

Most browsers would actually do okay if the web page instructions started with just the <html> tag, but don't do it that way. Doing things using accepted standards is part of becoming an effective developer of web pages.

The next three tags are a standard section of code called the "head" and gradually we will expand it to contain many useful pieces of information as we move forward. Especially notice that this cluster has an opening tag <head> and a closing tag </head>.

```
<head>
<charset="utf-8">
</head>
```

```
<charset="utf-8">
```

The "charset" information identifies the nature of the work on the page. Way back, there was a coding character set called the "American Standard for Computer Information Interchange" (ASCII) It contained only the capital and lower case letters of English along with basic punctuation and the digits for numbers. Over the years a whole bunch of international languages have been added. The most useful international coding system is the "Unicode Transformation Format", a fancy name for a large set of characters beyond the English language. It allows web pages to display Arabic, Hebrew, Russian, and many, many more.

```
</head>
```

The closing tag contains a slash to show it is closing the head section of code. Most browsers will try to show the page if you forget to close the section, but work hard to always do it right. Close all tags which need to be closed. Also, be careful to put the tag </head> to finish the head before starting the body section.

If you did this:

```
<body>
</head>
```

... closing the head after starting the body of the page, things could go very wrong.

We will cover all sorts of useful head elements as we move ahead.

Next up is the main section of a web page. Between the pair of tags, `<body>` and `</body>`, is the content we want to show when we design a web page. In this page body, we simply say "Hello World!", but all sorts of other kinds of information can be put into the body to be displayed: words in big sizes, words in tiny sizes, pictures, animations, videos, etc.

```
<body>
Hello World!
</body>
```

When all the fun is over, we need to shut down the page. The `</html>` tag is the final closing tag on the page.

```
</html>
```

Do think about what each tag does. Don't attempt to memorize anything you do not use regularly. Expand your list of tags as you actually use them. Review as needed.

Exploring Further

Let's add another line in our page between the body tags.

```
<body>
Hello World!
<p>This is a paragraph.</p>
</body>
```

The `<p>` tags tell the browser to treat the stuff between the tags as a paragraph. In browsers, a paragraph has spacing above and below the content. Try removing the paragraph tags and look at the page again.

What happens when the text has no paragraph tags? What happens if you add spaces before a line of text (inside or outside the paragraph tag, even without paragraph tags)? What happens if you add spaces between words in a line of text? Does the browser still display a paragraph if you forget (or intentionally leave off) the closing paragraph tag? (Learn your browser's abilities to overlook errors.)

Ask yourself questions and do tests to see what happens as we continue through this work. Trying and testing are important while you learn to code in html. Remember, understanding mistakes is good. When you figure out how to fix mistakes, you enhance your learning. When you code, you **will** make mistakes. It happens all the time to all levels of coders. Don't be ashamed to make the mistakes. But, be alert to fix the mistakes, too. You will learn more rapidly if you fix mistakes after making them. Even if you don't figure out the correction right away, think. Come back later. Think some more.

Ask for help, too. Look for help, preferably in the form of someone who will see your mistake, will point it out, but will prefer if you fix it yourself. That may not always be possible, but you should advance in skill by getting nudges ahead instead of being pulled ahead by someone who just shows you the "right" way to fix your mistake. Learning isn't necessarily efficient. Being efficient is a piece of the later stages of being a web designer. It's called "productivity" but isn't something to expect while you are learning. Work hard and it will come.

Enhancing Text

Next, let's experiment with some tag pairs which cause some of the text to look fancier.

The `` tags display the text as **bold** which often just makes the letter darker, but might be handled in other, fancier ways as we advance. "Strong" is a replacement for the earlier "bold" tags `` and ``. In the language of the designers of html, the bold tags have been **deprecated**. It's a fancy way to say, "Going forward, avoid using these obsolete tags." They still work, but are not as capable as the newer strong tags.

- Put a `` tag before the word(s) you want bold and the `` tag after the word or phrase or sentence, etc.
- Try italics with `` and `` (These replace the deprecated pair `<i>` and `</i>`)
- Try an advanced one: `` and be sure to use the closing tag after what you want in red. (note, I didn't tell you what the closing tag for a span is...but I'm hoping you will figure it out. If not, try doing a web search for html `` and read what you find there. There are many online guides for web coding. While there are lots of links from a search like this, I recommend following the link for **** W3C Schools**** which is free and generally even lets you try out code right online!)

Oh, by the way, just because something is in brackets does not mean that the browser will know what to do. Most browsers will just ignore tags it does not recognize. Try adding this to the body and see what you get.

```
<pr>hot</pr>
```

...until next time.

Though I'll send these guide sections by email, They are also going to be available on my website. Look for guide sections here: <http://runeman.org/classes/webclass/>