



Published on Free Software Magazine (<http://www.freesoftwaremagazine.com>)

Learning XHTML: Monty Python Style

The quickstart guide to learning standards-compliant XHTML

By [Andrew Min](#)

For reasons unknown to civilized (or uncivilized) man, all programming books are often immensely *boring*. Seriously. That is, until now. Today, Free Software Magazine presents (in conjunction with Andrew Min Writing Studios) Learning XHTML: Monty Python-Style.

Your name is King Arthur (or if you prefer, it could be Lancelot or Galahad or Bedivere or even Robin ...). Your quest is to learn XHTML. And your favorite color is... you'll get to that later. For now, all you need to know is that you need to watch the m?. They can be very dangerous around here. In fact, they can swarm together and take over sentences, even whole paragraphs. No really! (The management apologises for the fault in the introductions. Those responsible have been sacked.)

Introduction to XHTML and getting together the tools for it

You don't go out and attack castles with Holy Grails without first knowing what a Holy Grail is. So, first you'll get a little explanation. XHTML (eXtensible HyperText Markup Language) is a simple yet powerful markup language which utilizes tags. Tags are little strings of text enclosed by <> to let web browsers know what a web page contains.

Now that you know what XHTML is, you need to make sure you have all the materials you need (like coconuts and swallows and knights walking around clapping the coconuts). First, you'll need some sort of XHTML editor. Most operating systems come with Notepad for Windows or TextEdit for OS X. However, these are extremely underpowered (even for basic programming). Using them is like trying to chop down a tree with a herring: impossible. For GNU/Linux users, the choice is easy: use [Quanta Plus](#). If you absolutely /hate/ Quanta Plus (something I find very hard to understand!), try [Scribes](#), another powerful programming tool. Windows users... should switch to GNU/Linux. But if you absolutely refuse to do that then try [Notepad++](#). It's not as powerful as Quanta Plus or Scribes (think King Arthur vs. Sir Lancelot), but if you like pain... go right ahead. OS X users have three options: use [Smultron](#), a powerful editor for Mac users, [install Quanta Plus using fink](#) (a long and laborious project, but the end result is a powerful programming tool), or install GNU/Linux.

Writing a simple hello world page with paragraphs

Now that you know what you're doing (always helpful), it's time to write your first XHTML web page. Here it is:

```
<!DOCTYPE XHTML PUBLIC "-//W3C//DTD XHTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loos
```

Learning XHTML: Monty Python Style

```
<html>
<head>
<title>Hello World</title>
</head>
<body>
<p>Hello World</p>
</body>
</html>
```

That's it! Just copy that into your programming editor, save it as `helloworld.htm`, and then double click on the file to run it. It will open a new page in your web browser that simply says, `Hello World`. Congratulations! You have now completed the first step of your XHTML career.

Now, it's time to break it all up into little bite sized pieces that you could eat at that wedding feast Sir Lancelot broke up. First, look at the first line:

```
<!DOCTYPE XHTML PUBLIC "-//W3C//DTD XHTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loos
```

This is just a simple statement telling the world that this is an XHTML document. It should be at the beginning of every XHTML document.

```
<html>
```

This tells the world that this is the beginning of the XHTML document (just like meeting the Knights Who Say No? was the start of a hunt for a shrubbery). Remember that words surrounded by `<>` are tags.

```
<head>
```

This is the beginning of the "header" section of the XHTML document. This is unseen by the viewer, but relays important information to the browser.

```
<title>Hello World</title>
```

This is the title of the page. Everything located between the `<title>` and the `</title>` is part of the title. Since it's in the head section, it's unseen on the main page. However, the browser window's title will probably change to this.

```
</head>
```

This signifies the end of the head section.

```
<body>
```

This signifies the beginning of the "body" section. Virtually everything here will be shown on the web page.

```
<p>Hello World</p>
```

This is the nitty gritty of the page. The `<p>` tag is the paragraph tag. Everything inside of this will be neatly thrown into a paragraph. So if you have two paragraphs, they will automatically have whitespace between them. This is extremely handy since XHTML will ignore the file's own line breaks (you need to manually insert them, more on that later).

```
</body>
```

Learning XHTML: Monty Python Style

You should have already guessed what this does: it signifies the closing of the body section.

```
</html>
```

Can you guess this one? It's the end of the XHTML document.

You should also note that the document could have been formatted differently. Line breaks in the document's source do as much as Sir Lancelot? does to a fierce foe (absolutely nothing) as far as XHTML is concerned. So, the document could have produced the same results if it looked like this:

```
<!DOCTYPE XHTML PUBLIC "-//W3C//DTD XHTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html><head><title>Hello World</title></head><body><p>Hello World</p>
</body></html>
```

or this:

```
<!DOCTYPE XHTML PUBLIC "-//W3C//DTD XHTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html><head><title>Hello World</title>
</head>
<body>

<p>Hello World</p>
</body>
</html>
```

Also note that the tags are nested exactly in the order they are put in. This is crucial. If the `<html>` tag is before the `<body>` tag, then the `</html>` must be *after* `</body>`. So, this would be very wrong, since the `</body>` tag happens at the wrong time”

```
<body>
<p>
Hello World
</body>
</p>
```

Exercises

Yes, that's right. You can't be a good programmer without practicing. I'm told that Sir Lancelot practices throwing wooden rabbits every day.

- Write an XHTML page that says, “Free Software Magazine!” in the first paragraph, “Andrew Min is the best author ever” in the second paragraph, and “XHTML is almost as smart as Andrew” in the third paragraph.
- Try writing the above page *without* paragraphs (that is, without the `<p>` tags). See why you use them?

Basic formatting

Now that you know how to write a basic page, it's time to make it look pretty. After all, that's the appeal of XHTML (remember the appeal of Sir Lancelot's minstrel is his horrible singing). So it's time to look at adding some basic formatting.

Learning XHTML: Monty Python Style

There are several kinds of formatting in XHTML: **bold**, *italics/emphasis*, **bold/strong**, underline (Free Software Magazine doesn't render this correctly, but you should know what it looks like), deleted text (again, FSM won't render this. It's basically text with a horizontal line through it), and

blockquote

This is by no means a comprehensive list, but it should be enough to get you started.

There are two ways to add bold/strong text. The older and more popular way is to use the `` tag. The newer way is to use the `` tag. The latter is more technically correct, for reasons which are more dull than Tim the Enchanter. Don't argue. Just use the strong tag.

Next up is making your text emphasized. Again, there are two ways. You can either use the older and more popular `<i>` or the newer ``. And again, the newer `` tag is the correct answer for reasons even more boring than the bold reasons (actually, it's the same reason. If you *really* want to find out how, visit [this forum thread on `` vs ``](#)).

Now, onward to the underlined text. This you should *really* use sparingly. Why? Well, what else on a web page is underlined? Oh yeah, links (you'll get to that soon, I promise). You should use underlining about as many times as Robin attacks a dragon (virtually never). But for the times you absolutely have to have underlined text, the tag is `<u>`. Why isn't there a newer version, like it happens with `<i>` and ``? Simple: because you shouldn't be using it at all, remember?

It's time to turn to another type of formatting: deleted text. In other words, text that looks like it was crossed out (usually used for satire). But if you are a satire kind of guy (or gal), here's the tag: ``.

When you quote a source, it's nice to make it look a little different from the rest of the page. That's where the `<blockquote>` tag comes in. It allows you to quote something like this:

Howdy, I'm a quote. And swallows are much better than starlings.

Exercises

- Write some text that is strong and emphasized at the same time. *Like this*. Remember to nest your tags correctly (refer to *Hello World filled with m?, llamas, and shrubberies!* for more).
- Try adding some strong text inside a blockquote.

Using links

Refresher: what does XHTML stand for? If you say for freedom, equality, and Dennis the Constitutional Peasant, you're wrong. It stands for eXtensible HyperText Markup Language. And what does the HyperText mean? It basically means text that can be clicked on which will take the user to another page. That's what a hyperlink (commonly shortened to "link") is. So, it's time to look at adding basic links to a web page.

You'll start with a basic XHTML link. And the best way to explain how to add a link is to do an example:

```
<a href="http://freesoftwaremagazine.com/">Free Software Magazine</a> (http://freesoftwaremagazin
```

This `<a>` tag is much more complicated than any other tag you've learned so far. It's as complicated as Dennis the peasant. First of all, note that everything between the two tags gets turned into a link. So the above

result would output [Free Software Magazine](#). Now, look at the `href` part. Basically, anything between the two quotes (" ") is the location of the link. There are three things that can be between those hyperlinks. First, you could put a full web address like `http://freesoftwaremagazine.com/`, which is what I did above. Second, you could put in the location of a page with a URL that is relevant to the location of your page. For example, if you are editing the page `index.htm` which is on the server `http://www.freesoftwaremagazine.com` (so, the page is on `http://freesoftwaremagazine.com/index.htm`), and you wanted to link to `http://freesoftwaremagazine.com/about.htm`, you could just put down a link to `about.htm`, since both `index.htm` and `about.htm` are in the same directory: `Free Software Magazine` (`http://www.freesoftwaremagazine.com/about.htm`). If you're still trying to pick your jaw off of the floor, re-read all of section again. The third option is to put in an anchor, which is beyond the scope of this article.

Links are extremely powerful tools. You don't have to just link to web pages. You can also link to email addresses, like this: `mailto:email@emailaddress.com`. For example, `Click here to email one of the greatest actors of all time` (`mailto:john@cleese.com`). Of course, `mailto`s are much more powerful than just this (for more, see [this great article on mailto tricks](#)). Note however that this is one of the best ways for a spammer to get your email address, so you might want to consider using an obfuscator like [this one by John Haller](#) or [this by Andreas Neudecker](#).

Exercises

- Write two pages: `robin.htm` and `lancelot.htm`. Make `robin.htm` have a `mailto` link to `sirrobin@montypython.com` and `lancelot.htm` have a regular link to `robin.htm`.

Displaying images

When you're on the internet, virtually every page you see contains some sort of image. Some have them as a background image, some as a menu background, some as regular pictures. But unless you're browsing a mailing list or a directory output, it's almost guaranteed that you'll see some sort of image. So it's time to figure out how to put them in your website. First, an example:

```

```

There are several new things about this tag. First, note that *there is no closing tag*. This is *very* important. Even more important than remembering the migration patterns of African and European swallows. Also, note the `" /"` at the end of the tag. For any tag that has no closing tag, there *must* be a `/` (forward slash).

Besides these points, it should be pretty self explanatory. In the code above, `picture.jpg` refers to an image in the same directory as the page being written. Of course, you could also use something like `http://freesoftwaremagazine.com/logo.jpg`. Note however that it is often considered rude to use other images hosted by other people (It eats up their bandwidth. There's actually a pretty funny story about that [by Mike Davidson at Newsvine.com](#)).

Exercises

- Download [this](#) image, put it in the same directory as the page you're editing, and display it on your page.

- Try displaying an image that is also a link. If you can't get it, see [this article on background images](#) (but actually try /before/ looking).

Advanced formatting with stylesheets

You may have noticed that one very important thing has been left out so far: color. If you write a page right now, you'll have images and basic formatting but no color. Why? Well, XHTML actually does have support for all of this using the `` tag. However the `font` tag (along with several other tags) has been deprecated, a fancy word meaning that their use is discouraged. That's where Cascading Style Sheets, or CSS, comes into play. Now, explaining all the features of CSS is definitely out of the scope of this article (it's like trying to say the name of the Knights Who Formerly Said Ni). However, since most XHTML programmers use it you'll get a tiny taste right now. The CSS code goes in between the `<style>` tags, nested between the `<head>` tags.

First up, you'll change the color of the font on your web page. A usability note: /always/ make sure your text is readable. That means no light colors on light colors (e.g. yellow on white) or dark on dark (e.g. navy on blue). Doing that makes the text barely readable, just like last words in the Cave of Caerbannog. There's a good [resource on colors](#) at Mashable.com. Anyway, back to CSS. Here's an example of changing the font color:

```
<style type="text/css">
body {color: red}
</style>
```

First, note that everything between the `<style>` tags is CSS code. On the second line, the CSS code starts (and ends). The `body` tells the browser that all the text between the `<body>` tags should be colored red. The `{` and `}` are similar to the `<` and `>` of XHTML. Finally, the `color: red` makes the text color red. This brings us to an important point about web color. There are two types of web color: hexadecimal (a six digit code preceded by a # like #000000) or named colors (in plain English, like `red` or `blue`). These can get complicated so refer to [Wikipedia's great article on web colors](#) for reference.

Another thing you can do with CSS is set a background color. Here's the sample code:

```
<style type="text/css">
body {background-color: blue}
</style>
```

As you can see, this code is very similar to the previous code: instead of defining the `color`, it defines the `background-color`. You can also set a background image:

```
<style type="text/css">
background-image:
url('bg.jpg')
</style>
```

Exercises

- Create a page with a background color of #BC2EE4 and text color #F8C042.
- Figure out how to change link colors. Change it to #CAC5D0, and add it to the other page. If you have trouble, see [this page on link colors](#).
- Create a page with [this](#) background image, #CCDDEE as the text, and #000000 as the link color.

Publishing your web pages

Now that you have a web page, you'll want to put it on the world wide web so that King Arthur and his wife can see it. First, you'll need a web host with FTP support (there are tons of free ones around—at this point Google is your friend. Once you've registered with a host, connect to their FTP server using an FTP client. Quanta Plus has this built in, see [Quanta Plus's documentation](#) for more information. If you don't use Quanta, use [FileZilla](#) (there's a good tutorial [at the FileZilla wiki](#)). Make sure your main page is named index.htm.

Now, go to your web site that your host set up (e.g. <http://freedomain.com/andrewmin/>). You should see your index.htm page come up immediately (if not, you may have to wait a few hours for your site set up to be completed).

There's only one thing left to do: check your site in web browsers. There are two types of browsers: standards compliant browsers like the free software [Mozilla Firefox](#) or [Konqueror](#), and non-standards compliant browsers like Internet Explorer. If your page is valid XHTML according to [the W3C Validator](#), it will show up perfectly in Firefox or Konqueror. Unfortunately, you're also going to have to make sure your page works in Internet Explorer 6, well known for being one of the worst browsers for standards compliance on the net (you can use the free [Browsershots](#) service to see what your page looks like). Internet Explorer 7 improves on this, but is still pretty bad. It's like saying that Sir Not-Appearing-In-This-Film is better than Sir Robin the Not-Quite-So-Brave-As-Sir-Lancelot.

Exercises

- Create a webpage! Make a page with nice colors (use [Color Inspirator](#) if you're as uncreative as Sir Lancelot) detailing your experience with XHTML. Then, upload it to a host of your choice.

Where to go from here

Well, you're just about done. Now, you're probably wondering what to do next. If you want to learn more about XHTML or CSS, visit the [W3Schools](#). There are also a host of programming languages for the web, including PHP, JavaScript, Perl, ASP, Ruby on Rails, and Cold Fusion. They will allow you to generate XHTML pages on the spot. W3Schools has tutorials for these, as does our sponsor [O'Reilly Media](#). You may also want to learn about content management systems which are powerful web tools that give you more power for less work. Graham Oakes has a great article on this called [Introduction to Content Management Systems](#).

Good luck!

Biography

[Andrew Min](#) (/user/37372" title="View user profile.): Definition: Andrew Min (n): a non-denominational, Bible-believing, evangelical Christian. (n): a Kubuntu Linux lover (n): a hard core geek (n): a journalist for several online publications including Free Software Magazine, Full Circle Magazine, and Mashable.com

Copyright information

This article is made available under the "Attribution-Sharealike" Creative Commons License 2.5 available from <http://creativecommons.org/licenses/by-sa/2.5/>.

Learning XHTML: Monty Python Style

Source URL:

http://www.freesoftwaremagazine.com/articles/learning_html_monty_python_style
