XHTML

Communication Language of the Web

In this module, you will learn about XHTML, the language for information exchange on the web. When you have finished the module, you will:

- understand XHTML's purpose and use
- recognize the parts of an XHTML tag
- be able to identify the main parts of an XHTML document
- use the main XHTML constructs to build documents for publication on the web

What Is XHTML?

Suppose you are writing an email, say a question to your professor or a note to set up a date. We know that any email is composed of text characters—letters, numbers, spaces, and symbols. Yet the message that you write is more than just a random assortment of characters. Letters are fitted together to form words; words, in turn, are assembled into sentences, and sentences are carefully arranged to form paragraphs. All this careful construction is designed to help communicate the meaning that was the purpose of writing the email in the first place. Of course, you've been doing this for so long that you probably don't even think about it.

XHTML is the grammar of the web. When you go to a site, say Amazon or Google, their computers send a bunch of text to your browser, which turns it into the page that you see:

```
<p><em>Shop Class as Soulcraft</em> brings alive an experience
that was once quite common, but now seems to be receding from
society|the experience of making and fixing things with our hands.
Those of us ...</p>

<p>On both economic and psychological grounds, Crawford questions
the educational imperative of turning everyone into a "knowledge
worker," based on a misguided separation ...</p>
```

Exercise 1: Your First Look At XHTML

1. Open your web browser and navigate to http://www.uwg.edu
2. View the *source* of the page. The source is the plain text file your browser reads, as opposed to the pretty rendered version it shows you.
   - Mozilla Firefox: Click View → Page Source
   - Internet Explorer: Click Page → View Source, or choose the View menu and click Source
3. The browser brings up a new window showing the source of the page. Some of it will look cryptic, but if you scroll down you'll see the words and paragraphs of the page surrounded by funny symbols. These cryptic bits are *XHTML markup* that tells your browser what the page should look like.
4. Try looking at the source of some other pages, for example:
   - Google.com
   - a FaceBook / MySpace page

You should have noticed that the text of the page source does not look anything like the page you normally see.

How does the browser turn the XHTML text into web page?

Text To Web

Let's look at an example. Say we have this text file:

```
This is a heading

This is a paragraph.  Web development is fun!

This is another paragraph, brought to you by
the letters T, E, and X, and the number 2.
```

What will it look like in the browser? Try it!

Exercise 2: Try It Out

1. Open the w3schools Try It XHTML editor: http://www.w3schools.com/html/tryit.asp?filename=tryhtml_intro

   The Try It editor allows you to type XHTML snippets in the left box and see the results in the right box by clicking a button.

2. Select **all** the text in the left box and delete it

3. Select and copy the heading and two paragraphs in the example text above. Paste the text into the left box of the Try It editor

4. Click the "Edit and Click Me" button to see the results

Hmm... That doesn't look right. The browser treats all the text as one big blob: it doesn't seem to know anything about the heading and it ignored or paragraphs too!

If we had some highlighters, we could highlight the different parts of the text with different colors so the browser would know what we wanted. Maybe we'd use pink for the headings and yellow for paragraphs:

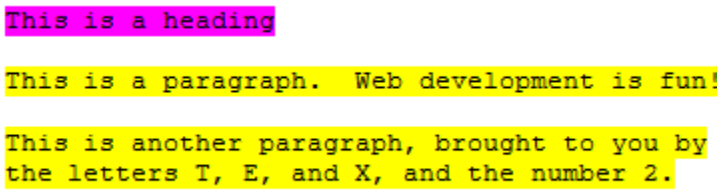


Figure 1: Plain text highlighted to show different meanings. Pink highlighting indicates the heading while yellow indicates a paragraph.

Unfortunately, text files don't know anything about highlighters. (I suppose you could try highlighting your monitor; let me know how that works out)

XHTML Markup

Instead of highlighting, XHTML uses *markup* to differentiate the various parts of the text to display (Markup is the M in xht**M**l). Think of markup as a set of instructions to the browser: if the text of the document indicates *what* to display, the markup tells the browser *how* to display it.

XHTML markup consists of *tags* the wrap around the text they modify; the tags tell the browser "this is a paragraph" or "that is a heading." Let's see an example of XHTML markup at work.

Exercise 3: Simple XHTML Markup

1. Go back to the Try It XHTML editor
2. Select **all** the text in the left box and delete it
3. Paste in the modified XHTML snippet:

```
<h2>This is a heading</h2>

<p>This is a paragraph.  Web development is fun!</p>

<p>This is another paragraph, brought to you by
the letters T, E, and X, and the number 2.</p>
```

4. Click the "Edit and Click Me" button to see the results

As trendy motivational speakers like to say, "let's unpack that." The tags in an XHTML document have several parts:

- start/end markers: each tag has a pair of markers that signal the start and end of the heading, paragraph, or other XHTML construct
- tag name: the tag name, such as p or h2, specifies what kind of tag this is, e.g. a paragraph or a heading. Both the start and the end marker must use the same tag name.

  XHTML defines a standard set of common tags; we'll learn a subset of these in the next few sections. The set of tags is eXtensible (the X in Xhtml).
- content: the "stuff in the middle." The content is the stuff being marked up by the tag. The content can be text, other tags, or even nothing in some cases!

XHTML Tags

As we examine the standard XHTML tags, keep in mind that XHTML's ancestor languages were designed for typesetting books and putting scientific journals online. While the web has subsequently evolved into a more interactive environment, there are still signs of this print legacy in many places. Keeping this in mind will help you grasp XHTML more quickly.

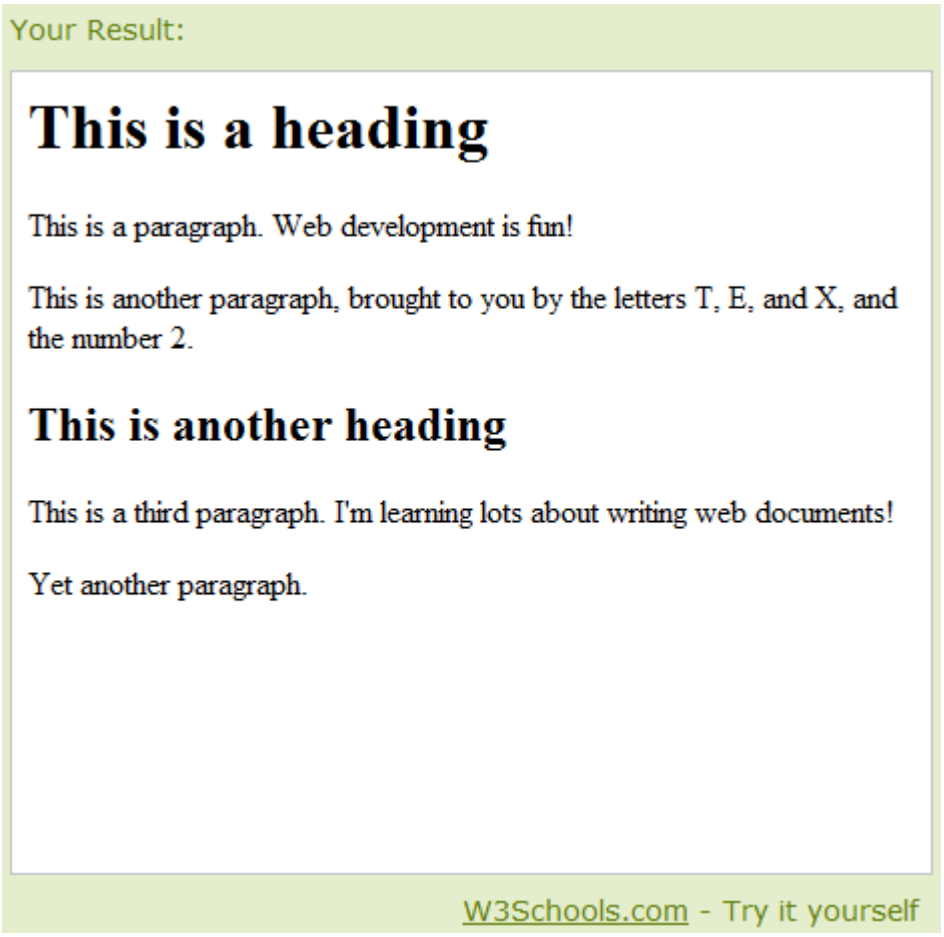
Paragraphs and Headings

You've already seen two very important XHTML tags: the paragraphs and heading tags. The `<p>` tag separates paragraphs from each other: everything between the matching paragraph start and end markers `<p>` and `</p>` is considered to be a single paragraph.

Headings supply some logical structure to a document. There are six headings, `<h1>` to `<h6>`; if your document was a book, `<h1>` would indicate a new chapter, `<h2>` would be a new section, and `<h3>` a new subsection.

Exercise 4: Try It Out

1. Go back to the Try It XHTML editor and your document from Exercise 3
2. Add two new paragraphs to the end of the document. Use the existing paragraphs as a guide
3. Add a section heading `<h2>` between the second and third paragraphs
4. Change the first heading from `<h2>` to `<h1>`
5. Click the "Edit and Click Me" button to see the results, which should look something like this:



Lists

XHTML supports two main kinds of lists: numbered and bullet. Numbered lists use the tag `<ol>` (for *ordered list*) while bullet lists use the tag `<ul>` (for *unordered list*). Both of these tags say "I want a list;" it is the stuff between the start and end markers that defines what is actually in the list.

What you put in a list? Usually *list items*, specified using the tag `<li>`. List items in a bullet list are shown with a small bullet graphic to their left; items in a numbered list are automatically numbers starting from 1.

Exercise 5: Try It Out

1. Go back to the Try It XHTML editor and your document from Exercise 4
2. Add a section heading `<h2>` with the text "Favorite Animals" after the last paragraph
3. Start a new ordered list after the new heading. It is a good idea to create both the start and end markers at the same time see you don't forget
4. Add three list items to the list: aardvark, beaver, chipmunk

5. Click the "Edit and Click Me" button to see the results
6. **Challenge**: what happens if you put another numbered list *inside* the list?

Neat And Tidy

If you're not careful, the XHTML document you been building can quickly become a disorganized mess. One of the things about computers is that they're very picky about minute details, so you have to be very careful to make sure that you do things correctly. If you make a mistake (and we all do), it is much easier to find and correct it if your document is neat and organized.

Let's look at an example: suppose you want to insert a bullet list between these two paragraphs:

```
<p>... end of some paragraph.</p>

<p>Start of some other paragraph... </p>
```

Follow this procedure to make sure that everything stays organized and legible:

1. Go to the end of the first paragraph and press `Enter` twice:

   ```
   <p>... end of some paragraph.</p>



   <p>Start of some other paragraph... </p>
   ```

2. Type in the start marker for the tag `<ul>`, press `Enter` twice, then type the end marker `</ul>`:

   ```
   <p>... end of some paragraph.</p>

   <ul>

   </ul>

   <p>Start of some other paragraph... </p>
   ```

   Notice how the markers for the list and paragraphs line up.

3. Before you type in your first list item, press `Tab` so that your list item is indented:

   ```
   <p>... end of some paragraph.</p>

   <ul>
       <li>First item</li>
   </ul>

   <p>Start of some other paragraph... </p>
   ```

   This makes it easy to see where the list starts and ends and which items belong to the list.

4. For each additional item, press `Enter` twice and make sure you are lined up with the previous list item:

   ```
   <p>... end of some paragraph.</p>

   <ul>
   ```

```
    <li>First item</li>

    <li>Second item</li>

    <li>Third item, etc.</li>
</ul>

<p>Start of some other paragraph... </p>
```

This process works for other tags as well. Following these guidelines will ensure that your XHTML is easy to edit and maintain.

Text Styles

Sometimes you need to emphasize individual words or phrases within a paragraph or a list. The emphasis tag `<em>` will emphasize whatever text appears inside it by showing the text in *italics*. You can also use the strong tag `<strong>` to turn text **bold**.

Exercise 6: Try It Out

1. Go back to the Try It XHTML editor and your document from Exercise 5
2. Change each of the animals in the favorite animals list to bold text
3. Change the word "fun" to emphasized text
4. Click the "Edit and Click Me" button to see the results

Character Codes

XHTML allows you to insert special characters into your documents using *character codes*. Some of these, like the code `<` for <, are used for characters that mean something special in XHTML and therefore cannot be typed directly into your documents. Others, like `©` for ©, allow you to special characters that don't normally appear on your keyboard. Here is a list of some of the common XHTML character codes:

Table 1: XHTML Character Codes

XHTML Code	**Character**
<	<
>	>
&	&
©	©

Hyperlinks

Much of the power of the web stems from hypertext links, or *hyperlinks*: text in a document that links to another document on the web. Hyperlinks allow users to click parts of a web page to navigate to the link's *target*, the webpage the link takes you to.

Hyperlinks consist of two parts:

- the text that will be displayed on the page, usually underlined and colored blue
- the hyperlink target: the webpage that opens when the link is clicked

The `<a>` tag creates a hyperlink. The text of the link appears between the start and end markers (as with the emphasis, strong, and list item tags). The target of the link is placed in the tag attribute called `href`. Here is an example:

```
<a href="http://www.google.com/">Google search engine</a>
```

You now know the fully expanded XHTML acronym: XHTML is an extensible language for marking up hypertext documents, i.e. the

eXtensible HyperText Markup Language

Exercise 7: Try It Out

1. Go back to the Try It XHTML editor and your document from Exercise 6
2. At the end of the document, create a new paragraph, then copy and paste the Google link above as the new paragraph's text
3. Click the "Edit and Click Me" button to see the results

Creating XHTML Documents

While the Try It XHTML editor has been fun, I'll bet you're anxious to create real, standalone web documents.

XHTML documents are text files with a special format. The two main parts of an XHTML document are the header and the body. The *header* contains some bookkeeping information about the document itself, such as its title and links to the style information for the document. The *body* is the part of the document that gets displayed in your browser. When you are creating an XHTML document, the tags and content you've worked with in this module would be part of the document body.

Here is a template XHTML document you can use as the starting point for your own pages:

```
<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE html
  PUBLIC "-//W3C//DTD XHTML 1.1//EN"
  "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
<head>
  <title>DOCUMENT TITLE</title>
</head>
<body>



</body>
</html>
```

Save a copy of the template in your documents folder on your hard drive or USB drive so you can find it when you need to create a new web page. (A name like `template.html` will help you remember what the file is for.)

Exercise 8: First Webpage

Let's create our first webpage!

1. In the folder where you're keeping the files for this class, create a new folder for this assignment. Call the folder `webpage_assignment`
2. Copy the XHTML template to a new file called `index.html`, which is the special name used for the main web page in a folder
3. Open the file in your text editor. If you're using Notepad++, right-click the file and choose "Edit with Notepad++" to edit the file. DON'T double-click the file yet, since this will open it in the browser.
4. Find the document title in the header. Replace the placeholder text with a descriptive title for your document. For this exercise, be sure to include your name in the title.
5. Copy the XHTML from the Try It Editor and paste it as the body of the document (between the start and end markers of the `body` tag)
6. Save your work, then double-click the file to see it in your browser!

Linking To Other Documents

We've seen how to link to other web sites with the `<a>` tag, but you can also went to other documents in your web site simply by placing the name of the document in the href attribute of the `<a>` tag. For example, another document could link to the index file you just created with this hyperlink:

```
<a href="index.html">Home</a>
```

Exercise 9: Create and Link A New Document

1. Create a new document called `about.html` by copying and renaming the XHTML template file
2. Open the file with your text editor
3. Change the title of the document to "About Me"
4. We're going to create a navigation aid called a *breadcrumb* that identifies where a visitor is on your site and provides links to other parts of the site. Breadcrumbs usually look something like this:

   Home > About Me

   a. Create a new empty paragraph
   b. Create the Home part of the breadcrumb using a hyperlink. Make sure the link points back to `index.html`
   c. Create the rest of the breadcrumb by inserting the ">" and location text "About Me"
5. Save the file, then double-click it to open it in your browser. Click the link to make sure that it connects to your original page correctly
6. On your own, add a link from `index.html` to `about.html`

Images

Web pages can display all kinds of images and graphics using the `<img>` tag:

```
<img src="picture.jpg" alt="Picture of Me" />
```

This image link specifies the image source (src), which is just the file name or location of the image. The `alt` attribute specifies an alternate description of the image, which is useful if the image can't load or for accessibility (e.g. folks using screen reading software).

What's with the /> at the end? Well, image tanks don't have any content, so the /> is a shortcut meaning "this tag has no content." In other words, the above in the link is the same as:

```
<img src="picture.jpg" alt="Picture of Me" ></img>
```

You can type the long version if you prefer, but in practice almost everybody uses the short form.

Exercise 10: Image

1. Find (or draw) a small picture of yourself. Copy it to the folder for this assignment
2. Add a new paragraph at the end of about.html
3. In the new paragraph, insert a new image link that links to your picture
4. Save your work and double-click the file to see your changes

Editing Web Pages

While you are editing web pages, make sure that you keep your work organized and neat to make it easier to read and understand your document. Another important point is to **work incrementally**: do your work as a series of small changes, saving and checking your progress often.

When you are editing a web page:

1. Double-click the file to open it in your browser
2. Open the file in your text editor to begin your work
3. Edit the file to make your changes, being sure to keep your changes small. Save your work
4. Switch back to your browser, then reload the page to see your changes
5. Repeat steps 3 & 4 for each change you need to make, remembering to work incrementally and keep your changes small.

Be especially careful with links: click the links in your document to make sure that your links actually go where you expect.

Summary

In this module you learned the basics of creating web pages with XHTML. This was a long module with lots of details, but don't worry: the more you practice, the easier creating web pages will become.

Key Points

- web pages that we browse on the Internet are XHTML documents, which are simply text files with a special format
- XHTML documents use *markup tags* to tell the browser how to display the content of the page
- tags consist of a start and end marker, which both have the same tag name
- XHTML supports many kinds of tags, including paragraphs, headings, lists, links, and images
- when writing or editing XHTML documents, it is important to **work incrementally**, making small changes and checking your work often
- keeping your XHTML documents neat and organized will help make your documents more understandable and easier to change

Additional Resources

- w3schools HTML tutorials: http://www.w3schools.com/html/default.asp
- W3C, the standards body of the web: http://www.w3.org/