

# Lab 1.3

## Basic HTML

The World Wide Web (commonly just called “the web”) is an enormous and rapidly growing collection of documents stored on computers all around the world connected by the Internet. In addition to text, the documents can contain sounds, music, video clips, 3-D animations, and more, and they are connected in such a way that readers can jump from document to document very easily. One of the many revolutionary aspects of the web is that it offers more people than ever the ability to publicly share information with an international audience. Some people use the phrase “electronic publication” to describe the process of creating and posting web pages because the web provides such wide distribution. From previous lab work, we are familiar with some of the core technologies that make the web possible, including Internet basics such as hostnames, URLs, and file transfer. This lab focuses on the last missing piece: HTML, the computer language used to produce web pages. In the procedure, you will produce your own web page from scratch, writing in basic HTML, and “publish” it to the web.

---

### For the Student

#### Vocabulary

All key vocabulary used in this lab are listed below, with closely related words listed together:

- HTML, source, tag
- link, broken link, URL
- programming language
- syntax
- host
- IP address
- hostname
- domain
- server vs. client
- web server, web browser (client)
- URL
- HTTP, FTP
- upload, download

#### Discussion and Procedure

##### Part 1. Basic HTML Syntax

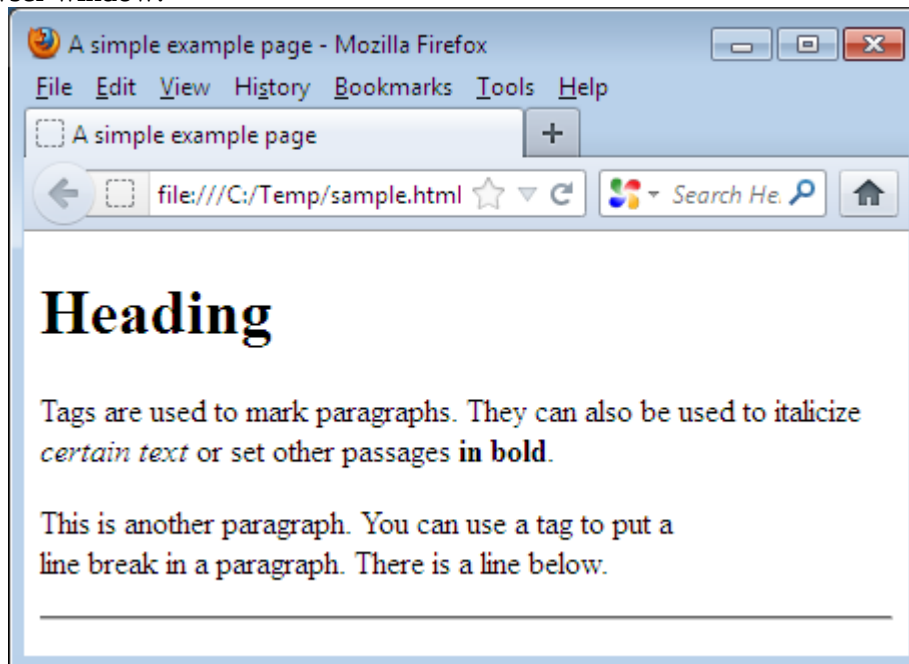
**What HTML is and isn't.** *HTML* stands for Hypertext Mark-up Language and is used to produce web pages. “Hypertext” is text viewed on a computer that includes connections called “hyperlinks” or, more commonly, just *links*, which can be selected to jump to other

documents, images, and other files. Hypertext existed before the web in the form of documents that were linked to other documents on the same computer (without using the network), but it became much more popular with the advent of the Internet and web browsers.

**How big is the web?** It is so large and grows so quickly that it is impossible to tell exactly how many pages are on the web, and there seems to be no really good estimate. In July 2008, Google claimed that its index had passed a trillion unique URLs, but an unknown number of those will be the same page appearing at different locations. The August 2014 Netcraft survey found just under a billion web sites.

HTML has some characteristics of a *programming language* but is not one, strictly speaking. The main difference is that HTML is used to specify the structure of a web page (title, paragraphs, and images, for instance) and how it should appear in a web browser, whereas a programming language is used to specify what the computer should do, in the form of instructions. Like a programming language, HTML has special words and punctuation that must be very carefully used for the computer to understand them. These special words and punctuation, combined with rules for their proper usage, are known as *syntax*, a word we will see again when we discuss programming. Syntax is a lot like the computer equivalent of grammar for a language like HTML or a programming language.

**Two views of HTML: web page and source.** A web page is a file consisting of plain text that follows the syntax rules of HTML. Before we get started writing HTML ourselves, we will look at an example. Here is a very simple example web page, as viewed in a browser window:



*a simple example HTML file as viewed in a web browser*

(Note that the address of the page is shown as a file URL, rather than the usual http. This is because the web page `sample.html` is being viewed locally, from the `temp` folder on Drive C, instead of from a web server. Although web pages are conventionally viewed via the network from a remote computer called a web server, web pages are files and can be stored on disks, just as with any other file.)

The file `sample.html` contains the following text, which is written in HTML and is called the web page's *source*, to distinguish it from the web page as seen in the browser:

```

<!doctype html>
<head>
<title>A simple example page</title>
<meta charset="utf-8">
</head>

<body>

<h1>Heading</h1>

<p>Tags are used to mark paragraphs. They can also
be used to italicize <span style="font-style:
italic">certain text</span> or set other passages
<span style="font-weight: bold">in bold</span>.</p>

<p>This is another paragraph. You can use a tag to
put a <br> line break in a paragraph. There is a
line below.</p>

<hr>

</body>
</html>

```

*HTML for the web page shown above*

The web browser's job is to read HTML source and display the page it describes.

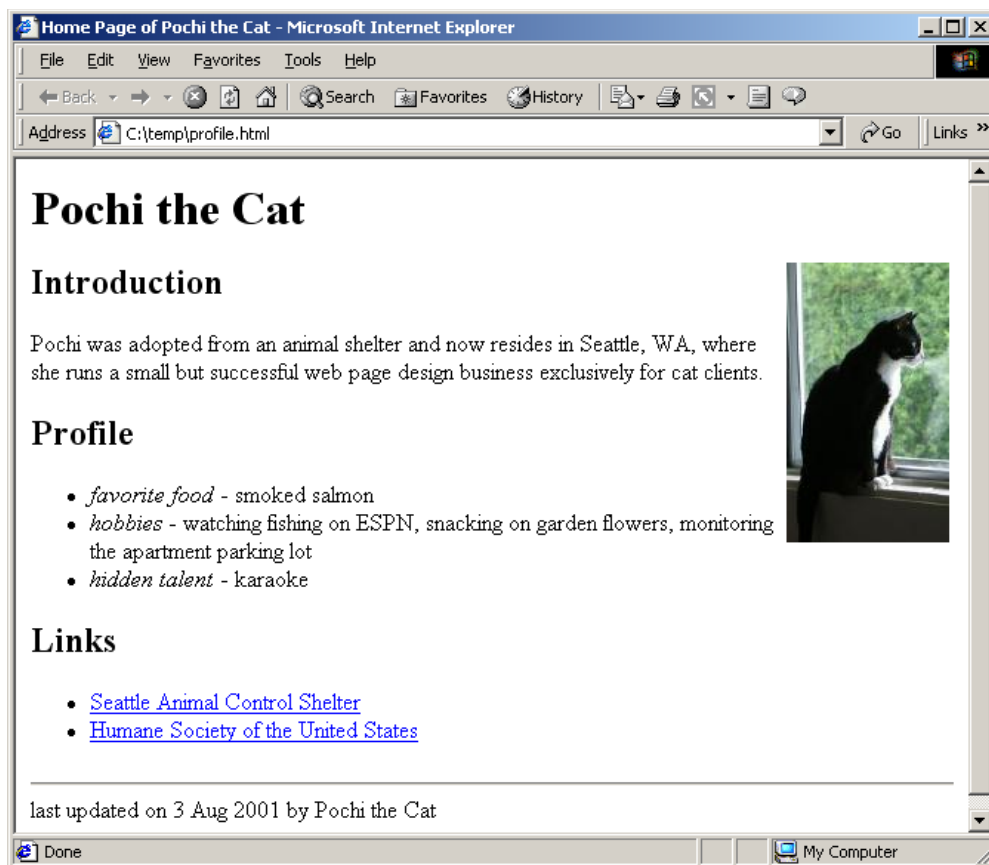
**Tags: web page building blocks.** You do not need to understand the whole HTML file now, but there are a number of important features to notice. In addition to the text to be displayed (including the page title, "a simple example page," which appears in the browser window's title bar), the source contains HTML *tags* placed carefully in the text. Tags are special markers that control text formatting (e.g., bold face, in a table, centered) or allow inclusion of non-text items such as images and links. Tags are enclosed in "pointy braces," i.e., the "less than" and "greater than" symbols `<` and `>`. These words (such as **body**, **h1**, **p**, and **span**) do not appear in the browser, but they affect how the text they surround is displayed.

If you look carefully at the tags in this example, you will see the two kinds of tags that exist in HTML. Some tags are "paired tags" and are used to mark sections of text, as opposed to just a specific point in the text. For example, text between a `<p>` tag and the

next `</p>` tag is treated as a single paragraph. Since the first tag (without the slash) denotes the start of a paragraph and the other tag (with the slash) denotes the end of a paragraph, in general, we call the first a “start tag” and the other an “end tag.” Other paired tags used above include **span** and **title**, which also apply their meaning to the text they enclose.

The other kind of tag is the “unpaired tag,” which is generally used to denote the insertion of some non-text item in the web page. In the example above, the `<hr>` tag is used to insert a “horizontal rule,” a thin horizontal bar stretching the full width of the window, in the page. There is no `</hr>` tag, because it does not make sense for a horizontal rule to “contain” text, the way it does for a heading or paragraph.

**Building your web page.** In the next steps, starting from a very simple HTML file, you will gradually add tags and enhance a personal home page. This page could be about you, your pet, someone you know, or an imaginary person. (See the example below.)



*example page you will construct in this lab*

Although web pages are usually viewed from a web server via the network, your web browser can also view local web pages—pages stored on the computer you are using now. Until your web page is basically complete, you should view it locally.

A good habit to develop early is to work on any project in small steps. The page you will construct in this lab will look something like this when you are finished, but we will add one feature at a time and check the result in a web browser after each step.

1. *Start Notepad.* Since HTML files consist of plain text, you can use a text editor like Notepad for a simple web page. In this first step, we will set up the basic framework that every standard HTML document has. Start by typing the **html** start and end tags, **<html>** and **</html>** tags, each on its own line. These tags will enclose the file's entire contents and instruct the browser to interpret the contents as HTML, rather than display it directly as plain text (as you see it in Notepad).
2. *Continue adding HTML framework.* Inside the **html** section (between the **<html>** and **</html>** tags), add a **head** section and a **body** section, each with paired tags. The **head** section will contain the web page title and other information that the browser does not display in the web page itself. The **body** section is where you put the text you want to appear on the page. Your source should now look something like this:

```
<!doctype html>
<html>

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

<body>
</body>

</html>
```

3. *Add a title.* In the **head** section, add a **title** section using paired tags, and type a title for your web page between the **title** tags. Choose something concise that describes your page well, such as "Home Page of [Your Name Here]."
4. *Save your document.* It might not seem like you have done very much, but this is already a good place to stop, save, and check your work. Choose the filename `homepage.html` and save the file somewhere, perhaps in a new folder called `Lab 1.3`. The `html` extension distinguishes this file from plain text files, which have the `txt` extension.

ALTERNATIVE: You will be saving your document very often in this lab. Rather than using the **File** menu, you might prefer to just press **Ctrl-S** to save.

5. *View your page in a web browser.* Minimize Notepad and a run a web browser. Select **File \ Open...** and click **Browse...** to select a local file. Find and open `homepage.html` on disk. Although your HTML file clearly contains text, it consists entirely of tags, so you should see a completely blank page. However,

your page title should appear in the browser window's title bar (top stripe). If not, double-check that your tags are properly typed (with pointy braces) and paired.

6. *Add headings to your page and its three sections.* Restore Notepad and in the **body** section, add a heading for the whole page in paired **h1** tags. Make the heading the name of the person (or pet) this page will feature. HTML offers multiple levels of headings, and **h1** is the highest level heading, so it will be displayed as large, bold text. Create subheadings for the “Introduction,” “Profile,” and “Links” sections using three pairs of **h2** tags. Your HTML file should now look something like this:

```
<!doctype html>
<html>

<head>
<title>Home Page of Pochi the Cat</title>
<meta charset="utf-8">
</head>

<body>

<h1>Pochi the Cat</h1>

<h2>Introduction</h2>

<h2>Profile</h2>

<h2>Links</h2>

</body>

</html>
```

Double-check that your heading tags are properly paired, then save your document and switch to your browser window. To force the browser to load the updated version of the page, you will have to click the **Refresh** button or press **Ctrl-R**. You should see your headings, with the first **h1** heading larger than the rest.

7. *Add a paragraph of text.* Return to Notepad and add a short paragraph of text between the “Introduction” and “Profile” headings, making sure to put your paragraph in paired **p** tags. You can break your lines anywhere you want, because when the web browser displays the page, it decides where to place line breaks so that the paragraph fits in the current window size. Again, save and check your updated page with the browser.

## Part 2. Syntax Experiments

When it comes to correct syntax, the computer is very sensitive—with HTML and, as you will discover later, with programming languages, too. If your tags are not exactly right,

the resulting web page might not look anything like you intend. To investigate exactly what aspects of HTML syntax are crucial, in the next steps, we will make slight changes to the HTML source and observe its effects on the web page as viewed in the browser. In some cases, we will intentionally “break” the HTML source by introducing errors.

8. *Change the spacing of your HTML source.* Modify the source by adding at least three blank lines between the **h1** heading section and the **h2** subheading section for “Introduction.” Save your changes, switch to your browser window, but before refreshing it, make note of how many lines of space appear to be between your **h1** heading and the “Introduction” subheading in the web page. Refresh the page. Did the number of lines of space change with your modification to the source?

Modify the source by adding at least ten extra space characters (not lines) between any two words in your first paragraph. Again, save changes and watch carefully as you refresh your browser, focusing on the two words you added space between in the source. Did the word spacing on the web page change with your modification to the source?

9. *“Break” an end tag.* Find the `</title>` end tag and remove the leading `<` character, leaving just `/title>`. Informally, we call this “breaking” the tag, because the resulting tag is not properly enclosed in pointy braces. Save changes and refresh your browser. What, if anything, changes in the page?

Put the `<` back, save, and refresh your browser. What happens?

10. *Change the spelling of a tag.* Change the `<h1>` start tag in your file by adding an extra h to the tag name to get `<hh1>`. Save changes and refresh your browser. What, if anything, changes?

Change the tag back and save changes before proceeding.

### Part 3. Lists of Items Using the `ol` and `ul` Tags

In the next step, you will add a list to your document. HTML supports two kinds of lists, “ordered” and “unordered.” Each item in an ordered list is numbered and displayed on a new line. An unordered list is similar, except that, instead of numbers, each item has a bullet (a solid dot, in HTML’s case) in front of it. The general format of a list is a section marked by paired `ol` or `ul` tags (for ordered or unordered list, respectively). The items go inside this section, with each item in paired `li` tags (where `li` stands for list item), as in this ordered list example:

```
<ol>
  <li>apples</li>
```

```
<li>peaches</li>
<li>lemons</li>
<li>oranges</li>
</ol>
```

The above HTML, viewed in a browser, looks like this:

1. apples
2. peaches
3. lemons
4. oranges

Note that in the HTML source, you do not have to explicitly number the items. Instead, the web browser numbers them for you in the order you provide them in. (For an example of what an unordered list looks like in a browser, see the “Profile” section in the example web page, “Pochi the Cat” above.)

11. *Add a list to your “Profile” section.* Using list tags (either of ordered or unordered is fine) as described above, add a list of “fun facts” about you under the “Profile” subheading. Be aware that you can use other tags inside each list item (**li**) section, e.g., **span** can be used to apply styles.
12. *Add author and modification date to the bottom of the page.* Before you add more to your page, add author’s credits and today’s date at the bottom of the page (properly placed in paired paragraph tags). So far, you have been working with paired tags, so this is an opportunity to add an unpaired tag. Use the **<hr>** tag for a horizontal rule line to separate the author and date from the rest of the page.

#### Part 4. Linking to Another Page with the a Tag

The web’s key feature is the ability to link documents together across networks with ease. In the “Pochi the Cat” example page, there are two links under the “Links” subheading. Clicking on the second link (“Humane Society of the United States”) brings you to the page at URL <http://www.hsus.org/>. The paired **a** tag is used to mark the link and specify the link URL. The HTML to for the Humane Society link is as follows:

```
<a href="http://www.hsus.org/">
Humane Society of the United States</a>
```

When you use the **a** tag to make links, make sure that the URL is in quotes and in the start tag, you have a space between the tag name, **a** (“anchor”), and the **href** (“hyperlink reference”) URL section.

If you do not have any favorite web sites, for the purposes of this lab, just add links to popular and generally useful sites like Google web search and the Yahoo web directory:

```
http://www.google.com/
http://www.yahoo.com/
```



13. *Add an unordered list of links in the “Links” section.* Recall that you can put tags inside list items (**li** sections). Using paired **ol** tags for an unordered list, make a list of at least two links. Just as in the example link above, make sure to put the title of the web page you are linking to between the **a** tags. As you add each link, save changes, refresh your browser, and click the link to make sure it works.

**Moving targets.** A *broken link* is a link whose URL is incorrect. The URL might have a typo in it, or the page at the URL might have been renamed, moved to another folder or server, or deleted altogether. The web changes so frequently that links often go out of date. Some sites, such as Yahoo ([www.yahoo.com](http://www.yahoo.com)), are stable and are unlikely to move or otherwise change URL, but pages that private individuals maintain frequently come and go without warning. If you do end up running into a broken link, a good first strategy for checking if the page has moved somewhere is to use a search engine. Search using as many words and phrases as you can think of that you are sure will appear on the page.

## Part 5. Putting Your Page on the Web

Before we add an image to your page, we will “publish” the version you have now—put it on the web. Before you go on, proofread your page, because what you put on the web is viewable by anyone in the world with web access.

To publish your page, you will be transferring a copy of your HTML file to your web folder by FTP, just as you did with a plain text file in the previous lab. Refer to your Lab 1.2 instructions if you need the full instructions. In summary:

14. *Upload the page to your web folder.* Open the WinSCP program, log in to the server `sandbox.mc.edu` using your mc account credentials.
15. After a successful login, drag the copy of your home page from wherever it is stored on your desktop machine to the server.
16. *Access your page on the web server.* Your page will be available at the URL `http://sandbox.mc.edu/users/userid/filename`, where *userid* is still your MC id, and *filename* is `homepage.html` if you've been following the instructions.
17. Open this URL in your web browser and verify that the page is the same as when you viewed it from a local disk. If you encounter problems, see the Troubleshooting section of the previous lab.

If you are on campus, you may open the URL `http://sandbox.mc.edu/users` to show the list of users, and click on your own account, then click on the file name name. This does not work off-campus.

Minimize FTP once you are done checking your page on the web server, then proceed to the next part of the lab.

## Part 6. Adding an Image Using the `img` Tag

An image is the last difference between your web page and the sample page presented earlier. There are many different image file formats, but the more common ones are called JPEG and PNG, both of which apply special techniques for storing image data using a relatively small amount of disk space. JPEG filenames usually end with extension `jpg` or `jpeg`, and PNG filenames end with extension `png`. The older GIF format is still quite common also. Most of the images that you see on the web are files in one of these three formats.

You first need to find an image file to add to your page. One way is to use an image you have taken with your digital camera or cell phone. Another way is to copy an image you find on the web already, but you should make sure only to do this with images that you have permission to use. The first lab page (from which this and the quiz are linked) includes some suggested image sources.

18. *Save an image from your camera or an existing web page.* For a camera, you're mostly one your own, but once you manage it plug it into the computer, the software should prompt you appropriately for transferring and saving images. To copy an image from a web page, first see if there is a link expressly for downloading, and use that. If not, you can **right-click** on the image to pop up a menu. Select **Save Image As...** and save it on the hard drive.

Save your image into the same folder where your `homepage.html` file is stored. Before clicking **Save**, you might want to change the name of the file so that it concisely describes the image. Your image will not appear in your web page simply because it is in the same folder as the HTML file. In the next step, you will add a special tag to your HTML source to display the image.

19. *Add the image to your web page.* The unpaired `img` tag is used to add an image to an HTML document. Here is an example usage of the `img` tag:

```

```

The **src** (“image source”) setting specifies the name of the image file to be inserted in the page. The **alt** (“image alternative”) setting is used to provide a brief text substitute for the image, commonly as a descriptive phrase. Strictly speaking, the **alt** setting is optional, but it is very important for web browsers that are not capable of displaying images. For example, there are special web browsers for visually impaired people that use computerized speech to “read aloud” web pages. Text substitutes for images are essential for browsers like these. In conventional browsers, if you leave your mouse pointer over an image, the associated **alt** text will pop up in a small label, like a caption for the image.

Follow the example above and add an `img` tag to your HTML source before your “Introduction” heading. Make sure the image filename you specify in the **src** setting is exactly right, including capitalization. Save your HTML and open the HTML file from local disk to verify that the page includes the image.

**TROUBLESHOOTING:** If your image does not appear in the page, check to make sure that the image file is in the same folder as your HTML file and that the **src** filename is exactly right, paying attention to case and extension.

Try resizing the browser window and watch the image carefully. Does it change size when you resize the window?

20. *Adjust image size, if necessary.* Image files vary widely in size, so you might find that your image is too large on the page when viewed in a browser. One good idea is to style the image tag to force the displayed image width to be a particular portion of the page width. For instance, adding this setting to the **img** tag example above forces the browser to resize the image so that it always takes 20% (one fifth) of the browser window width:

```

```

Alternatively, you can set a specific width, like `style="width: 350pt"`, but the percentage allows the picture width to adjust as the window size changes.

Try adding a width setting to your **img** tag, making sure not to forget the percent symbol. Check results in your browser as you adjust the percentage. Try resizing the browser window and watch the image carefully. Does it change size when you resize the window?

21. *Adjust image alignment, if desired.* If your image is tall, like the cat photograph in the example page, you might want it displayed side-by-side with the text. You can easily do this by adding an floating alignment style to your **img** tag. Adding **"float: right"** to the example above puts the image on the right side of the window, allowing text to appear to its left.

```

```

22. *Upload the updated web page.* Using FTP, upload the updated HTML file to your web folder. Do not close FTP. Your page is not quite ready to view from the web server yet, because it includes an **img** tag whose source image file is not yet in your web folder. In your web browser, open the URL for your page (as recorded in Step 15) and refresh the page to make sure you are seeing the latest version. What do you see where the image belongs?
23. *Upload the image file.* Back in FTP, upload the image file to your web folder. Switch to the browser window displaying your page from the web server and refresh the page. The browser should now properly display the image.
24. **Troubleshooting:** If your page looks fine viewed from the hard drive, but not from the server, check these things.

- Make sure you have saved the latest version of your HTML page on the server. After adding the `img` tag to your page, you must send the modified version to the server using FTP; the server still has the old version. Be sure you did this.
- Make sure you didn't miss the previous step: the image must be sent to the server just like the HTML page.
- Windows generally ignores case, but the Linux server we use does not. Make sure the file name given in the `img` tag matches the file name exactly in both case and extension.
- Windows often hides known file extensions. (This supposedly simplifies things, but it generally just creates confusion.) Your `src` attribute must give the file name with its proper extension, and the image will not display properly if you do not. On windows, right-click on the file and choose properties to see the extension (shown just after giving the file type). You can also list your server files in a web browser to see their true names. Make sure your `src` name has the correct extension.

Congratulations! Your very own corner of the World Wide Web is now complete. Remember, your page is accessible to anyone in the world with a web browser, as long as they have your page's URL, perhaps through a link. Consider sending e-mail to a friend or family member with your new page's URL.

### Optional Additional Activities

- Most HTML tags can be combined freely, as you saw with **li** sections containing text formatting tags and link tags. Try other combinations in your web page, e.g., lists nested inside lists, list of small images, links that are images instead of text (`img` tag inside an `a` section).
- Make a second page on another topic (e.g., a friend of yours, a family member, your favorite book, a band you like). Save the page with a filename other than `homepage.html`. In your `homepage.html`, add a link to this page by using the new HTML file's name as the link reference (`href` setting). To link a web page to another page in the same folder, instead of using a full URL, you can just use the filename.
- Most web browsers have an interesting feature that allows you to view the HTML source of the currently displayed page. Not surprisingly, HTML as used in commercial web sites is much more complicated than the basic tag usage we covered in this lab. Try visiting your favorite pages and, from the **View** menu, select **Source**. Although you will find many tags that we have not presented here, virtually every page has links and images, so you should be able to recognize those tags.
- Add a `<style>` section in the header to control the appearance of certain tags. Perhaps you would like [a sans-serif font](#) for your headers, or [indented paragraphs](#).

## Further Reading

- The organization responsible for setting and publicizing standards (including those defining the HTML language) is the World Wide Web Consortium (W3C). Their home page includes links to resources about the latest web technologies.  
<http://www.w3.org/>
- Make sure your web pages are accessible to everyone, including the many users out there who have disabilities. In addition to the W3C's resources on accessibility, check out this page for guidelines and information:  
<http://www.washington.edu/doit/Resources/web-design.html>