Laboratory

First Steps in HTML

16A

Objective

- Learn some basic HTML concepts.
- Make a simple web page.

References

Software needed:

1) A basic text editor (for example, NotePad on a PC or SimpleText on a Mac) to create the web page
2) A web browser (Internet Explorer or Netscape Navigator, for example) to view the HTML page

You'll also need a disk or some other means of saving your work. Check with your lab instructor for details.

Please note: For this lab, avoid using software that automatically generates the HTML tags (such as Microsoft Word, PageMill, Dreamweaver, etc.)—you want to get a taste of working "under the hood" with the actual HTML code.

Textbook reference: pp. 507–512

Background

Chapter 16, "The World Wide Web," discusses the basics of HTML, and the Activity section below introduces the skills necessary for this lab.

Activity

For this lab you'll create a simple web page using HTML (HyperText Markup Language). HTML is the basic language of web pages. The designer of a web page uses HTML "tags" to describe the general structure of the page. These tags identify various elements of the page (titles, headings, paragraphs, etc.), along with character formatting information (bold, italic, etc.). Once the elements of the page are defined using the tags, a web browser interprets these tags to determine how to display the web page on a computer screen.

For example, let's say the browser sees this collection of text and tags in HTML:

```
<H1>Welcome to My Page!</H1>
```

Notice the first tag: <H1>. All tags are surrounded by angle brackets in this fashion. This first tag is HTML code for a heading (like a headline). There are six different levels of headings in HTML, H1 through H6. H1 is the largest and most prominent heading.

The <H1> tag "turns on" the heading formatting. All the text following this tag will be in the boldest heading format, until another tag "turns off" the formatting. Immediately following the first tag is the phrase Welcome to My Page! A browser would show that text in a bold and prominent way, due to the <H1> tag.

The text is followed by another tag: </H1>. Notice the slash before the H1. The slash is used to designate "stop" tags: This is the tag that "turns off" the heading formatting, so that subsequent text will not be formatted as a heading. Most tags function like this, with a "start" tag that designates the beginning of a section, feature, or formatting, and a "stop" tag with a slash in it that designates the end of the section, feature, or formatting.

Here, then, is how <H1>Welcome to My Page!</H1> might actually be interpreted by a browser:

Welcome to My Page!

As the biggest and boldest heading in HTML, it pretty much "screams" on the page!

To further understand how HTML works, we'll make a practice page. Start by launching your text editor, which should bring up a blank page on your screen. Carefully type the following code:

```
<HTML>
<HEAD>
<TITLE>Practice Page</TITLE>
</HEAD>
<BODY>
```

These first tags are the basic structural tags that should begin every HTML page. The <HTML> tag announces to the browser that the document is in HTML code. The <HEAD> tag is a structural tag for the header of the document. The most common element of the header is the <TITLE> of the document. Notice the text surrounded by the <TITLE> and </TITLE> tags—this text would appear in the title bar at the top of

the browser window. Notice how the <TITLE> start and stop tags are nested within the <HEAD> start and stop tags.

The main contents of the page appear after the <BODY> tag. Everything following this tag would appear in the main portion of the browser window.

Your HTML document should end with the following two stop tags—type them in now:

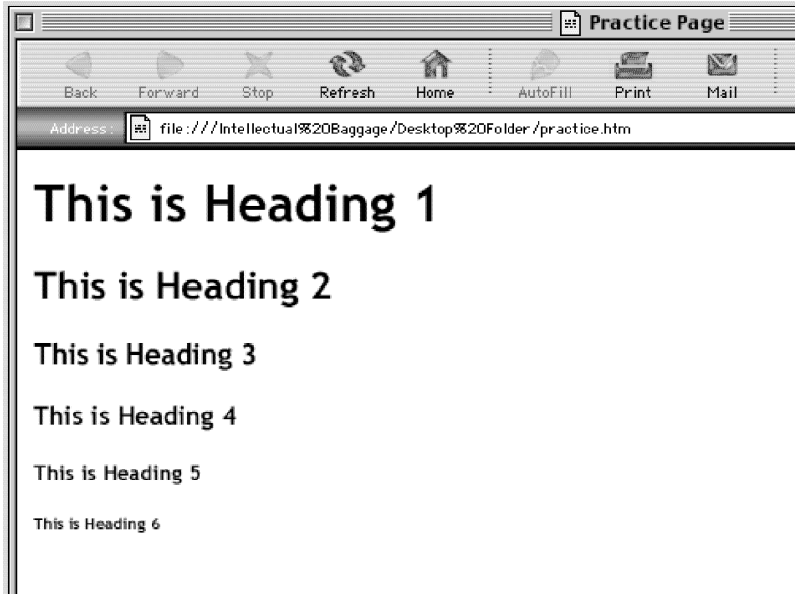
```
</BODY>
</HTML>
```

Let's return to the discussion of headings in HTML. To see how your browser handles the various heading tags, open up some space between the <BODY> and </BODY> tags in your document, and type the following code:

```
<H1>This is Heading 1</H1>
<H2>This is Heading 2</H2>
<H3>This is Heading 3</H3>
<H4>This is Heading 4</H4>
<H5>This is Heading 5</H5>
<H6>This is Heading 6</H6>
```

We are using our text editor to create our code, of course, but now we want to view our work with a browser, like Microsoft Internet Explorer or Netscape Navigator, to see what our HTML code looks like when a browser renders it as a web page. Before you can view your code in a browser, though, you need to save it to disk. We'll do that now: Go to your text editor's *File* menu and choose *Save*. Call this file `practice.htm`. (If you're not sure what disk or where on the disk you are to save your file, consult your lab instructor.)

Once you've saved your file with your text editor, launch your browser, go to the *File* menu, and choose *Open*. Find your way to the saved `practice.htm` file and open it. You should see the six sizes of headings. It will look something like the screenshot below (note the name **Practice Page** in the title bar of the browser window—this was the text that was between the start and stop <TITLE> tags):



(By the way, the exact appearance of an HTML page can vary from browser to browser. While this lack of uniformity drives serious web designers nuts, it ensures that almost any computer, regardless of what fonts are installed on it, will be able to format and display HTML pages.)

Throughout the rest of this lab, you'll go back and forth between the text editor, where you'll edit your HTML page, and the browser, where you'll view your changes. (**Important note**: Make sure to never try to save your page while viewing it in the browser—your browser may save the page in a format that your text editor can't open!)

You designate paragraphs in an HTML document with the paragraph tag: `<P>`. This tag informs the browser to leave some space in front of whatever block of text it precedes. Return to your page in the text editor and type the following code after the headers and before the `</BODY>` tag:

```
<P>This is my first paragraph.</P>
<P>This is my second paragraph.</P>
```

(It's worth noting that tags are *not* case-sensitive—you can make them lowercase if you'd like. We're going to keep ours uppercase to make them stand out more in our code, so it is easier to read. However, the new XHTML standard, which will eventually replace HTML, explicitly says they must be lowercase. Like many things in Computer Science, learn today, revise tomorrow.)

Now you'll view your changes. Save your work in the text editor, return to the browser, and click the *Reload* or *Refresh* button (or reopen the file from the browser's *File* menu if you closed it). You should see your new paragraphs in the browser window.

Return to your page in the text editor. Now let's take a quick look at lists in HTML. The two most commonly used lists in HTML are *ordered* and *unordered*. Here's a screenshot of a heading followed by an unordered list:

Here's the HTML code for the above list:

```
<H3>My Favorite Jazz Musicians</H3>
<UL>
<LI>Charles Mingus
<LI>Miles Davis
<LI>Sonny Rollins
<LI>John Coltrane
<LI>Thelonious Monk
</UL>
```

Notice from the screenshot that in an unordered list the browser puts a bullet (•) in front of each item in the list.

Take a look at how the code for this list works. The <UL> tag announces to the browser that an unordered list (*unordered* meaning *not numbered*) is about to begin. The <LI> tag is used to announce each individual list item. (Important note: <LI> stands for "List Item"—notice that it's the letter *L* and the letter *I*, not the letter *L* and the number *1*!) After all the items in the list have been entered, the unordered list feature is turned off using the </UL> tag. (Notice that the <LI> tag does not have a corresponding "stop" tag; it's one of only a few HTML tags that doesn't take a stop tag—this will also change in the new XHTML standard, which requires the stop tag.)

What if you wanted the list to be numbered? You'd simply tell the browser to create an ordered list instead of an unordered list, by replacing the <UL> and </UL> tags with the <OL> and </OL> tags. Note that the <LI> tags remain unchanged. Making that simple change to the preceding code would make the list look like this:

Again, note that the <LI> tags are not changed at all—the browser inserts the numbers 1 through 5 that appear in the list above. These numbers are not actually written in the HTML code.

Add a simple unordered list (list anything you'd like) on your practice page, then save it and view it in your browser. Then go back to your page in the text editor and change the <UL> and </UL> tags to <OL> and </OL>. Save your work, view your page in your browser again, and note the difference.

Finally, we'll mention some simple text formatting tags, for bold and italic text. Like the heading tags, the text formatting tags must be turned on and then turned off with the appropriate start and stop tags. To create bold or italic text, surround the text to be affected with the appropriate tags. In the case of bold text, here's a sample line of code:

```
<B>This text will appear bold.</B>.
```

Here's how your browser interprets that line of code:

This text will appear bold.

To make italicized text, use the start and stop <I> tags instead:

```
<I>This text will appear in italics.</I>.
```

Here's how it will look:

This text will appear in italics.

Experiment with these tags on your practice page.

Exercise 1

Name _____ Date _____

Section _____

While you're obviously not ready to become a freelance web designer, you *are* ready to make a simple web page. For this lab, you'll be creating a simple biography web page.

Your page should include a brief biographical sketch of yourself. Include the following information:

- Where you're from
- What your major is
- What your plans are after graduation
- What you hope to be doing 10 years from now

Also include a brief discussion of your hobbies and interests outside of school. Be creative: If you're an avid hiker or biker, a vegetarian, a world traveler, an expert knitter, a bassoonist, an excellent cook, or a national champion hog caller, tell your story (or stories...). And hey, if there are any oddities about yourself ("My eyes are two different colors" or "You should see my barbed wire collection" or "I was raised by the Borg"), mention them on your page!

You should also create a couple of lists on your page; some possible topics include:

- Your favorite musical groups
- Your favorite foods
- Your most despised foods
- Your favorite books
- Your favorite TV shows

Almost anything that can be put into a list format can be included here.

Deliverables

When you're satisfied with your page, view it with the browser and print it from the browser's *File* menu. Hand in the printout of your page as it appears in the browser. Your lab instructor might also want you to hand in an electronic copy of your file—check with your instructor for details.

Laboratory 16B

Linking & Images in HTML

Objectives

- Expand on the basic HTML skills you learned in Lab 16A.
- Work with links and images in HTML.

References

Lab 16A must be completed before working on this lab—you'll be modifying the HTML biography page created in that lab.

Software needed:

1) A basic text editor (for example, NotePad on a PC or SimpleText on a Mac) to create the web page
2) A web browser (Internet Explorer or Netscape Navigator, for example) to view the HTML page

You'll also need a disk or some other means of saving your work. Check with your lab instructor for details.

Textbook reference: pp. 507–512

Background

You already learned the basic structural, heading, list-making, and text formatting tags in HTML in Lab 16A. The Activity section below explains what you need to know for this lab.

Activity

Now that you've created a basic web page, you'll be learning a few more HTML skills:

- How to link to another page you've created.
- How to link to other sites on the Web.
- How to incorporate images into your web page.

First, let's talk about how HTML pages differ from other media, such as a novel. A best-seller provides a tried-and-true method for presenting information. A reader experiences the words in a *linear* fashion, starting on page 1 and progressing page by page through to the end of the book. If you've ever been moved by a skilled author, you know this can be a highly effective way to express ideas.

HTML permits the organization of information using a different paradigm. The presentation of information in HTML can be especially interesting and compelling because it can be experienced in a *non-linear*, user-directed fashion. While reading a web page, a reader encounters links embedded in the document, triggered by key words or phrases called *hyperlinks*. These links will take the reader to new web pages that somehow expand on or are related to the concepts implied by the key words—if the reader chooses to follow them! With HTML, readers access information as they wish, depending on their interests, following some links and ignoring others.

Words that function as links in HTML stand out from the rest of the text. The default way that most browsers display links is as underlined, blue text. While web designers can tweak the appearance of links (to appear in other colors, for instance), most at least reserve underlined text for links because users have come to expect underlined text to function in this way.

Linking in HTML is done with the `<A>` (called "anchor") tag. This tag both changes the appearance of the text it surrounds and turns that text into a link. Here's the typical code for a link:

```
<A HREF="page2.html">Go to Page 2</A>
```

Let's look it over and see what it does. The `A` specifies the anchor tag, used for links. `HREF` (short for "hypertext reference") is an *attribute* of the anchor tag, meaning it modifies the tag. This is followed by an equals sign and a value in quotes, `="page2.html"`, which specifies where the link should take the user. That's the end of the start <A> tag. This tag has "turned on" the linking feature, and any text that follows it will be affected until the stop tag turns off the feature. So the text after the start tag, `Go to Page 2`, will appear in the browser as a link—usually underlined and blue, as we've mentioned.

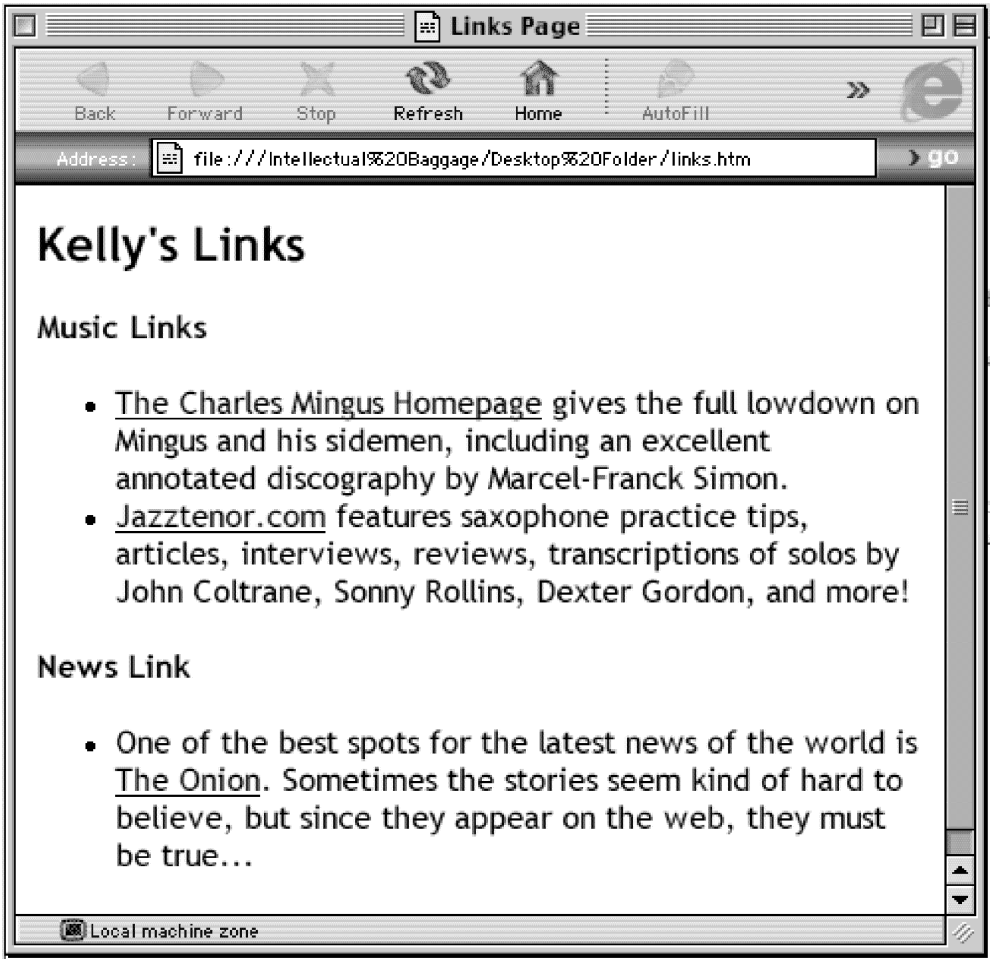
Finally, the stop tag, `</A>`, turns off the linking feature. Notice that the stop anchor tag does not include the attribute inside it—no HREF or address is listed in it. Stop tags don't take attributes.

The address in quotes in the above code is called a *relative* address—it resides in the same folder as the page linking to it. As you know, web pages can also link to different sites entirely, using *absolute* addresses. Contrast the relative address in the tag above with the absolute address used below:

```
<A HREF="http://www.cnn.com">Visit the CNN web site.</A>
```

Notice that the absolute address on the preceding page is the complete URL of the referenced page. When referencing an "outside" page, it's necessary to provide a complete address so the browser knows where to find it.

To show you the use of anchor tags "in action," here's a screenshot of a page with links, followed by the actual source code for the page:




```
<HTML>
<HEAD>
<TITLE>Links Page</TITLE>
</HEAD>
<BODY>
<H2>Kelly's Links</H2>
<H4>Music Links</H4>
<UL>
<LI>
<A HREF="http://www.siba.fi/~eonttone/mingus">
```

```
The Charles Mingus Homepage</A> gives the full lowdown on Mingus and
his sidemen, including an excellent annotated discography by Marcel-
Franck Simon.

<LI><A HREF="http://www.jazztenor.com">

Jazztenor.com</A> features saxophone practice tips, articles, interviews,
reviews, transcriptions of solos by John Coltrane, Sonny Rollins, Dexter
Gordon, and more!


</UL>
<H4>News Link</H4>
<UL>
<LI>One of the best spots for the latest news of the world is
<A HREF="http://www.theonion.com">The Onion</A>. Sometimes the
stories seem kind of hard to believe, but since they appear on
the web, they must be true...
</UL>
</BODY>
</HTML>
```

Study the above code to make sure you understand how anchor tags are used in web pages.

In addition to working with links, for this lab you'll also be adding an image to a web page. To make an image appear on a web page, HTML code that tells the browser where the image can be found is added to the page. The browser then hunts down the image and displays it in the appropriate spot on the page. (The image must be in a format supported by the browser. The two most widely supported formats are JPEG and GIF. These formats are briefly discussed on p. 79 of your textbook.)

This differs from the way an image is incorporated into a word-processing document (a Microsoft Word file, for example). In a Word file, the image is actually embedded into the file itself. By contrast, in an HTML document, the browser is referred to the image's URL on the server; the browser loads this image from the server and displays it on the screen in the appropriate place in the HTML file.

The `<IMG>` tag, called the image tag, is used in the code for images. Here's an example:

```
<IMG SRC="myphoto.jpg">
```

`IMG` of course specifies the image tag, and `SRC` is the attribute that signals to the browser that what follows is the address of the image. After the equals sign, the value `"myphoto.jpg"` which is the name of the image. appears in quotation marks.

Just as with links, image tags can use relative or absolute addresses. Because the address in our image tag above includes only the name of the file, with no additional navigational information, our browser will assume that the file is in the same directory as the HTML page referring to it.

Here's an example of an image tag referring to an outside, absolute address:

```
<IMG SRC="http://www.mysite.com/photos/myphoto.jpg">
```

Please note that there is no stop tag for the `<IMG>` tag.

Exercise 1

Name _____ Date _____

Section _____

You will be modifying a copy of your biography web page from Lab 16A. Before you begin, you need to:

1) Make a new directory (folder) on your disk, called `YourLastNameLab16b`. (Of course, substitute *your* last name for `YourLastName`! For example, Meyer Lab16b.)
2) Make a copy of your web page from Lab 16A. The copy of the file should be called `index.htm`.
3) Put `index.htm` into your new directory.

   Note: From this point on, we will refer to the `index.htm` page as your **home page**.
4) You'll also need to put a JPEG image into your Lab 16b directory. If you have your own JPEG image, you can use that; otherwise, a sample JPEG image (called `lab16b.jpg`) is available on the CD-Rom. Your lab instructor can show you how to copy this practice image into your Lab 16b directory.

Exercise 2

Name _____ Date _____

Section _____

1) Open your home page with your text editor.
2) Open a new line immediately after the <BODY> tag near the top of your HTML document, and add a line of code that will make the JPEG image appear on your web page. (If you're using the image provided on the CD-ROM, it's called `lab16b.jpg`; if you're using your own image, you'll need to use the name of your image as part of the tag.)
3) Save your home page in your text editor. (It should still be called `index.htm`.)
4) View your home page in your browser (as described in Lab 16A).

   (Please note: As long as your JPEG image and your home page are in the same folder, you should see the image appear at the top of your page when viewing it with your browser.)

Exercise 3

Name _____ Date _____

Section _____

1) From within your text editor, go to the *File* menu and choose *New*. A blank text document will appear.
2) Go to the *File* menu and choose *Save As*.
3) In the *Save* dialog box that appears on your screen, route this document so that it is saved in your Lab 16B directory.
4) Name this new document `links.htm` (make sure there are no spaces in the name, and use lowercase letters).
5) Click the *Save* button. From now on, we'll call this new page your **links page**.
6) Add the appropriate structural tags to set up your new links page, starting with `<HTML>`. (Refer to your home page, or to the Activity section of Lab 16A, if you need a reminder of what the structural tags look like.)
7) Once you've got the appropriate structural tags in place, add a heading that says: YourName's Favorite Links. (Replace YourName with *your* name.)
8) Using the screenshot and code demonstrating the use of the `<A>` tag as a model, create your own simple links page showing a few sites you enjoy.
9) Test your new links page in the browser to make sure that it works. If you try clicking on a link in the browser and get an error message, check to make sure you have no typos in your code. (By the way, sometimes you'll get an error message when trying to visit another site because the server for that site is down. It doesn't happen often, but it is a possibility. If you're sure that there are no errors in your code and that you're online, and you're still getting an error message when you click on a link, it's possible that's the source of the problem.)

Exercise 4

Name _____ Date _____

Section _____

1) Open your home page with the text editor.
2) On your home page, add a new paragraph that says "Check out some of my favorite links" on a line just above the `</BODY>` tag toward the bottom of your document. Make the last three words of that sentence ("my favorite links") into a link that, when clicked on, will take you to your links page.
3) Save your work, and view it in your browser.

Exercise 5

Name _____ Date _____

Section _____

1) Add a link to your links page that, when clicked on, will take you to your home page.

Deliverables

Open your two web pages with your text editor and print them out, and then view them with your browser and print them again, for a total of four printouts. Your lab instructor might also want you to hand in an electronic copy of your file—check with your instructor for details.