cs002 Lab 2

Introduction to HTML

Assigned: October 24, 2016 Due: November 4, 2016

Introduction

Have you ever wondered how websites are made? They're all written in a language called **HTML** (HyperText Markup Language), which is the building block and structure of the webpage. We can use HTML to create headings, paragraphs, lists, tables, links, and much more.

This tutorial will introduce you to the very basics of HTML. By the end of it, you should know enough to create your own basic web page.

This tutorial will require:

- A web browser (such as Chrome, Firefox, Opera, Internet Explorer or Safari)
- A text editor (such as Sublime, Notepad, Notepad++, TextEdit, Vim or Emacs)

Text Editors

At various points in this course, you will be asked to write files containing either text or code using a text editor. While there are many free text editors to choose from (vim, emacs, Notepad, Notepad++, etc...), the TAs recommend using Sublime Text. If you do not have Sublime Text installed on the computer on which you are working, you can go to http://www.sublimetext.com/3 and choose the correct download link. If you are working on a Brown computer, you will want to choose the portable version instead of the full installation.

When using Sublime Text, you may occasionally encounter a popup asking you to purchase a subscription, which you should feel free to ignore.

Please note that for this course, *only simple text editors are allowed*. What You See Is What You Get (WYSIWYG) editors such as Dreamweaver or FrontPage are NOT allowed. Also, we recommend editors like Sublime (or Notepad++), which have nice features that highlight your HTML nicely and handle indentation levels.

Lab Goals

- Learn how to set up an HTML document
- Use common HTML tags to add headings, paragraphs, links, and images
- Make lists and tables
- Understand divs, classes, and IDs

Lab Assignment

Writing HTML In A Text Editor

You will be writing your HTML in a simple text editor. Once you have finished, you will open the document in a web browser, which will take care of interpreting what you have written and rendering it into a web page.

The following steps are a high-level overview of your workflow:

1. Open the text editor from the Start menu (or equivalent). For example, click on *Start* and search for 'Notepad' or 'Sublime'.
2. Once you have your text editor open, create your web page (which you will soon learn how to do).
3. Once you are done, make sure to save. In Notepad and Sublime, you can do this by going to File → Save As. Make sure that whatever you name your file ends in ".html", so that the operating system will know to open it with your web browser.
4. You can go back to editing your "*.html" file by right-clicking on it and selecting Open With → Notepad or double click the file.
5. To preview your web page, just right-click the file and Open With → Firefox, Chrome, etc.

Tip: In Sublime, you can right-click anywhere in the editor and select "Open in browser" to preview your web page.

Setting up an HTML file

Always start by putting `<!DOCTYPE html>`, which informs the browser what language it is reading. To start the the document, put `<html>` on the next line and `</html>` on the last line. Your entire document will look like this:

```
<!DOCTYPE html>
<html>
</html>
```

There are two parts to an HTML file, a header and a body. The header is where you will put information like the title (what it says at the top of your browser and its tab), information about the document (don't worry about this yet), and links to other files (don't worry about this yet either). Add the header (`<head></head>`), title (`<title></title>`), and body (`<body></body>`) tags to your document. There should only be ONE set each of html, head, title, and body tags. All other tags may be repeated.

Sample HTML file:

```
<!DOCTYPE html>
<html>
     <head>
          <title>My first HTML page</title>
     </head>
     <body>
     </body>
</html>
```

When you make web pages, you should always set up your document to look like this—specifying the DOCTYPE and adding the html, head, title, and body tags.

TASK: Set up an HTML document with DOCTYPE, head, title, and body tags.

HTML Tags

HTML Tags are the <> ("brackets") you saw. There almost always is an opening tag and a closing tag. `<html>` was the opening tag and `</html>` was the closing tag of the document. Tags represent different ways of structuring your text and logically separating it into different sections.

TASK: Add a paragraph tag <p></p> to your document between the body tags and write a message between the opening and closing tags (if you add another paragraph tag, this will put the text on a new line).

Note that the order of closing tags is important. Whatever the most recently opened tag is must be closed before you can close any other tag. For example, consider the following:

```
<p>Click<a href="other.html">here!</a></p>
```

Note that the closing anchor tag (`</a>`) came before the closing paragraph tag (`</p>`), instead of the other way around. But wait! What's that "href" inside of the opening tag? There's all of this extra text! This is called an **attribute**. Opening tags can have attributes, or specific properties. There are some attributes that all tags have, and some that are specific to certain tags. Here, we have the "href" attribute of the anchor, which describes a destination for a link that you can click on.

Adding Headings

Headings (not to be confused with the header, `<head></head>`) help break up and emphasize sections of the page. There are six heading tags, `<h1>` to `<h6>`. `<h1>` is the biggest heading, and `<h6>` is the smallest.

Example:

```
<h1>Big Heading!</h1>
```

```
<h6>Little Heading!<h6>
```

TASK: Add two different-sized headings to your document.

Remember, this goes between the body tags.

Adding links

The anchor tag is used to create links to other web pages. Try adding an anchor tag between the body tags of your document, just like in the above example. You’ll notice that "here!" becomes a link, but clicking on it creates an error. That’s because you probably don’t have an "other.html" file. If we did, it would go to that page. You can also link to a page online:

```
<a href="http://google.com">Link</a>
```

Clicking on "Link" will bring you to Google. This is called an absolute link.

If you have several HTML pages and want to link them together:

```
<a href = "home.html">Homepage</a>
```

The HTML above would make the "Homepage" text a link to your home.html page.

TASK: Add a link to your HTML page.

Adding Images

We can also add images to our webpage using an `<img>` tag. For example:

```
<img src="http://example.com/about/logo.jpg/">
```

TASK: Find an image and add it to your webpage.

We can also save an image to the computer and link it to your page. Right click the image you found online and press Save As to save it to your computer. Now when you link this image to your page, you will have something like this:

```
<img src="logo.jpg" />
```

TASK: Add an image that is saved to your computer to your webpage.

This link is *relative to where you are in the file system*. For example, if your image is saved to your Downloads folder, you will need to have something like this:

```
<img src = "Downloads/logo.jpg"/>
```

Notice that image tags can be closed within the same tag by adding the "/" at the end.

Important things to remember:

- End tags have a slash before the tag name
- Text doesn't need to be within tags to appear on the page.
- Tags must be closed, either with a separate closing tag or self-closing, and closed in the proper order. If you fail to do this, the browser will try to guess how to show your page, which could be wrong. If your page looks strange, *you may have forgotten to properly close a tag*.
  - Follow this rule: Always set the beginning and end tags at the same time, always placing them on the farthest end of the item being affected.

  Ex: `<tag3><tag2><tag1>Multiple Tags</tag1></tag2></tag3>`

- Indenting your code according to levels can make your life much easier, and is industry standard. For example:

```
<html>
    <head>
        <title> Some title </title>
   </head>
   <body>
       <p>Some text</p>
   </body>
</html>
```

Making Lists

In HTML, there are two types of lists: ordered (numbered) and unordered (bullets). An ordered list begins with the `<ol>` tag and ends with `</ol>`. An unordered list begins with `<ul>` and ends with `</ul>`.

Each item in the list must be surrounded with `<li>` and `</li>`, which stands for List Item.

Here's an example of an ordered list:

`<ol>`
`   <li>Frozone</li>`
`   <li>Edna Mode</li>`
`   <li>Syndrome</li>`
`</ol>`
	1. Frozone
	2. Edna Mode
	3. Syndrome

You can also nest multiple lists together, like so:

`<ul>`
`   <li>Mr. Incredible</li>`
`   <li>Elastigirl`
`       <ol>`
`           <li>Violet</li>`
`           <li>Dash</li>`
`           <li>Jack-Jack</li>`
	● Mr. Incredible
	● Elastigirl
	1. Violet
	2. Dash
	3. Jack-Jack

```
        </ol>
    </li>
</ul>
```

Notice that the above code indents "Violet", "Dash", and "Jack-Jack" (sublist) under "Elastigirl". Also notice that an ordered list lists items numerically, starting with 1. An unordered list lists items in bullet points.

TASK: Make an unordered list with three bullet points and a sublist with two bullet points.

Making Tables

You can organize data on your page using a table.

Here’s a breakdown of the tags used in a table, in the order used to declare a table:

1. `<table>` starts the table. It can have attributes:
    - `border` tells how thick the table borders are, in pixels
    - `cellspacing` is the amount of spacing between cells
    - `cellpadding` is the amount of space/padding between the cell border and the cell
2. `<caption>` and`</caption>` adds a caption over the table. This is optional.
3. `<tr>` starts a new table row.
4. `<td></td>` adds a piece of table data (text, image, or link) as a cell in the row (like a column).
5. `</tr>` ends the current table row.
6. `</table>` ends the table declaration.

Example:

```
<table border="3" cellspacing="1" cellpadding="1">
   <caption>The Incredibles’ Superpowers</caption>
   <tr>
       <td>Mr. Incredible</td>
       <td>Elastigirl</td>
       <td>Violet</td>
```

```
            <td>Dash</td>
            <td>Jack-Jack</td>
        </tr>
        <tr>
            <td>Strength and durability</td>
            <td>Extreme flexibility</td>
            <td>Invisibility</td>
            <td>Super speed</td>
            <td>Shapeshifting and fire powers</td>
        </tr>
     </table>
```

TASK: Make a table with three rows and three columns.

Adding Divs

Divs are useful because they allow you to divide your page into different containers. You will need this when you learn CSS in the next lab, so that you can style different parts of your website individually.

Try this:

```
<div style="width:30px; height:30px;background-color:red">
</div>
<div style= "width:30px; height:30px;background-color:yellow">
</div>
```

Now, you can visually see the two different divs as two different containers.

TASK: Make three divs with different colors and sizes. (hint: change the size of the div by modifying the width and height)**.**

Classes and IDs

All tags can take a "class" or "ID" attribute, which is used for CSS. Adding a class to an element labels these elements into one category, which you can style at the same time using CSS. You can label multiple elements with the same class.

```
     <div class = "Incredibles"></div>
     <p class = "Incredibles"></p>
```

Adding an ID to an element labels this one individual element, so that you can style it separately from other elements in CSS. You cannot label multiple elements with the same ID.

```
    <div id = "Jack-Jack"></div>
```

This might not make a lot of sense now, but classes and IDs are very important when you learn how to style your webpage with CSS next week.

Creating Email Links From Your Page

To let your visitors email you (or somebody else) directly from your website, you can use the `mailto`: command. It follows the same coding scheme as a standard hypertext link and appears in a similar manner, but clicking it should prompt the visitor to send you a piece of e-mail using an email application on their computer.

For example: `<a href="mailto:president@brown.edu">Click Here To Email Christina Paxson</a>`

Notice it's the same format as the links you should be used to writing except you write "mailto:" in place of the “http://” and your email address in place of the page address. You still need the `</a>` tag at the end. Note there is *no space* between the colon and the e-mail address.

Some commonly used tags to remember:

Tag/Closing Tag	Description	Common Attributes
`<html></html>`	This starts an entire web page. Everything goes inside of it.	
`<head></head>`	This declares the head section, which contains information about the page (such as the title).	

`<title></title>`	This contains the title for the page.	
`<body></body>`	This declares the body section, which contains the content that will actually be on the page.	
`<p></p>`	This creates a paragraph.	
`<a></a>`	This creates a HyperText link (Hyperlink).	`href` (link to another page)
`<br />`	This tag creates a line break.	
`<hr />`	This creates a horizontal line across the page.	
`<img />`	This inserts an image.	`src` (URL of the image)
`<span></span>`	This declares a section of text that you want to be logically separated from its surrounding text.	
`<div></div>`	Declares a division of the page for grouping together other HTML elements.	
`<strong></strong>`	This tag bolds the text within it.	

`<em></em>`	This tag italicizes the text within it.	

Check-off Requirements

- Have a correctly set up HTML document with DOCTYPE, head, title, and body tags
- Have one paragraph tag, two different sized headings, and a link
- Add two images, one from an online source and one that you downloaded
- Make an unordered list with three bullet points and a sublist with two bullet points
- Make a table with three rows and three columns
- Make three different divs of different colors/sizes

Submission

To submit this lab, please raise your hand so a TA can come check your work. Make sure you tell the TA to check you off their list, or else you will not receive credit. If you do not finish your lab in the allotted time, please either come to office hours or another lab section to finish your work. If you are unable to complete this lab due to sickness or injury, please contact **BOTH** Don and the cs002 TAs via email. Please contact the cs002 TA's if you have any further questions.

cs002 TA Email: cs002tas@cs.brown.edu

Don Stanford's Email: don.stanford@gmail.com