How to Update the Lew Lab Website:

**Introduction**

This guide is designed to walk any user through the steps of making rudimentary changes to the content of the Lew lab website. Knowledge of HTML, CSS and JavaScript is not required, but a willingness to use Google *is*. Almost any questions can be most quickly solved by searching for the answers online. Looking for the answers yourself is also the best way to learn how web design works.

Additionally, it is strongly suggested that any beginning users **make a copy of the file you are modifying before making any changes**. It is likely that you will break some cosmetic functionality of website when making changes to the code and will be unprepared to identify the code change responsible. In such a situation, you will want to be able to restore the file to an earlier state before trying again.

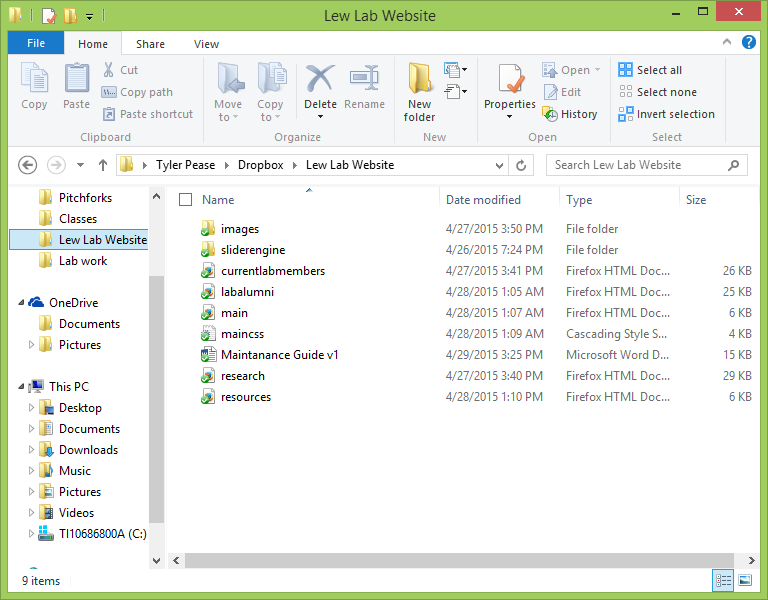
On the subject of breaking things: like in biological science, an understanding of web code is best achieved by making a single change in the code and observing the resulting change in the browser. There is a direct relationship between what you do in the code and what happens in the browser. There’s no magic involved.

**Part Zero: The Website Structure**

The Lew Lab website, and any website, is really a collection of files, including HTML files, CSS files, JavaScript files, and images, which call upon one another to create the web page that you see. HTML files have links which can direct to other HTML files within the same website, or to HTML files that make up other websites. These HTML files are stored on a server which is accessible to the rest of the world through a URL (google.com, for example, is the URL for the Google website). The practice of storing these files is called “hosting.”

At the time of writing this guide, the Lew Website consists of five web pages. These are called main.html, currentlabmembers.html, research.html, resources.html and labalumni.html.

Before we begin, ensure that all of the files and folders which make up the Lew Lab website are in a single location (a single folder). Their relationship to one another is important, as the source code points to other files based on their configuration relative to one another. For example, main.html and labalumni.html should be in the same top level folder.

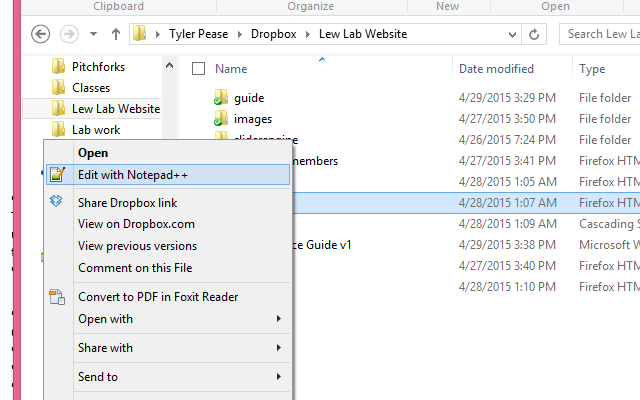


This same top level folder should contain a subfolder called *images*, which contains all of the images used on the website, as well as a subfolder called *sliderengine*, which contains the JavaScript which runs the slide show on the home page. These subfolders will further contain more subfolders.

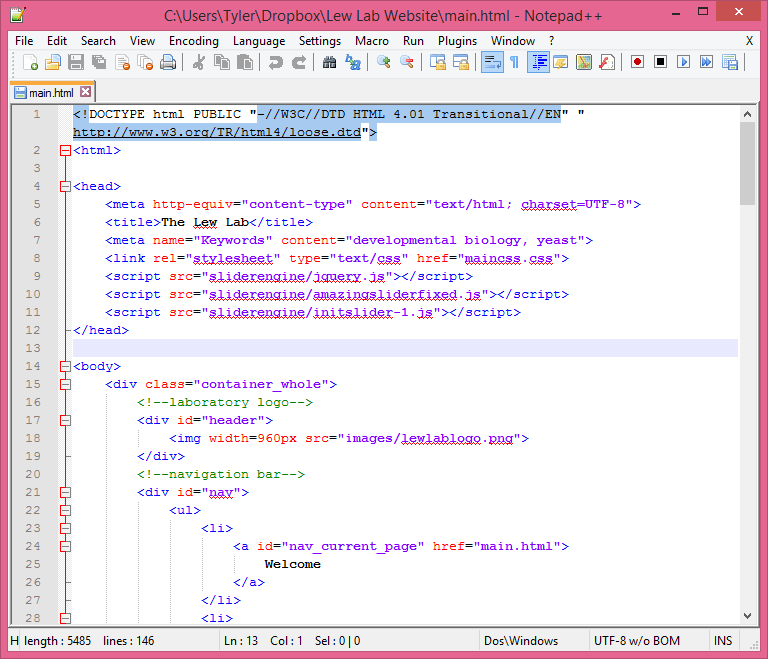
**Part One: Understanding the sections of HTML Files**

To edit any HTML, CSS, or JavaScript file, you will have to edit it with Notepad++. To do this, right click the file and choose the option, “Edit with Notepad++” from the dropdown menu. Double clicking the file will likely open the file with your web browser, as your computer has likely been set to open .html files with a web browser by default.

Open main.html in Notepad++

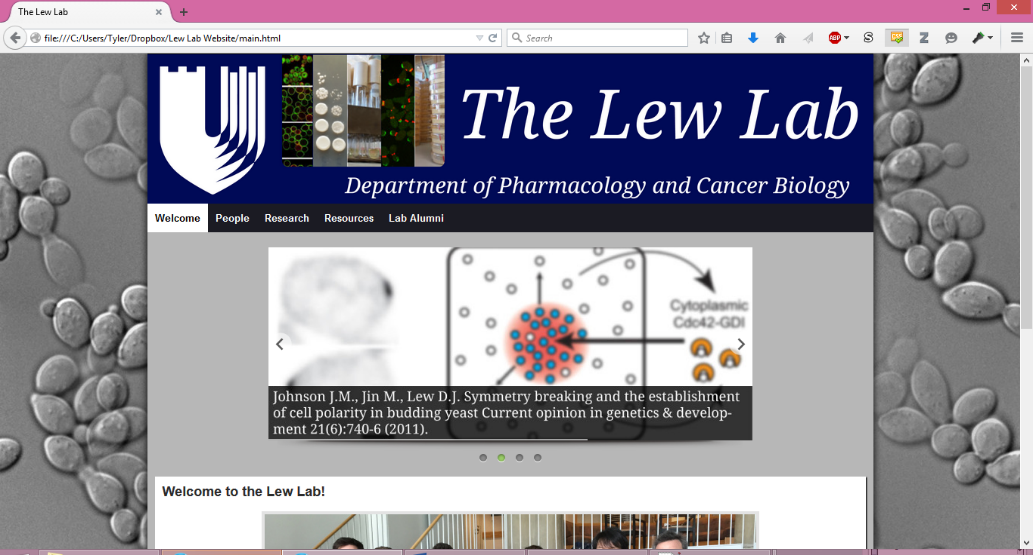


Notepad++ will show you your html file which should look somewhat like this:

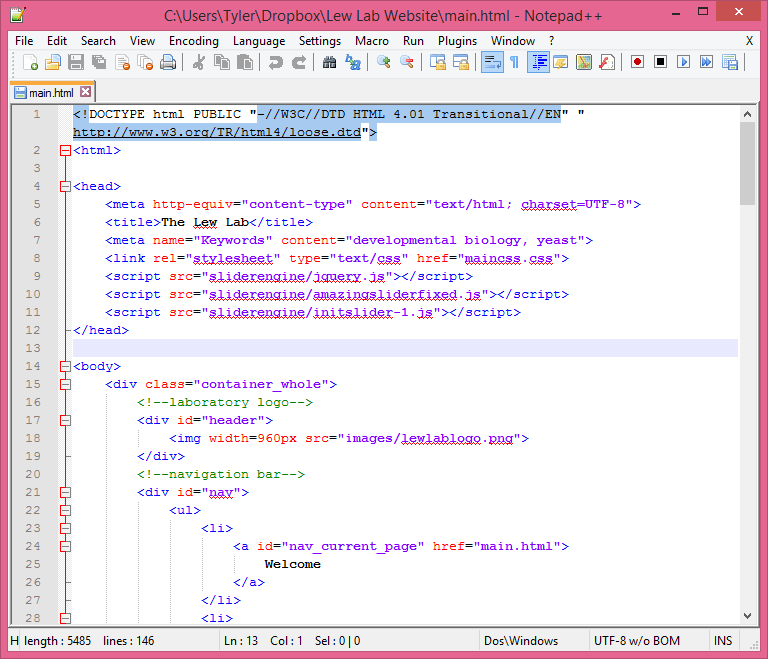


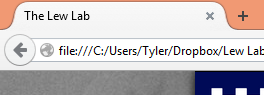
This is an HTML file, and the source code for the main page of your website. All HTML files contain *tags*, indicated by < > brackets, which indicate sections of content. Sections are “closed” by a tag with a / in it, like </ >. All HTML files open with the <html> tag, and all close with the </html> tag. Scroll to the bottom of the file in Notepad++ to find the </html> tag.

If all of the files and subfolders are organized properly, as described in section zero, you can run this .html file in a browser of your choice. You should see the front page of the website, as below. Notice that the URL is *actually a location on your hard drive*, not a .com or a .edu. You are looking at a local copy of the web page, and we will have to upload this file to the server which hosts our website before anyone else can see it.

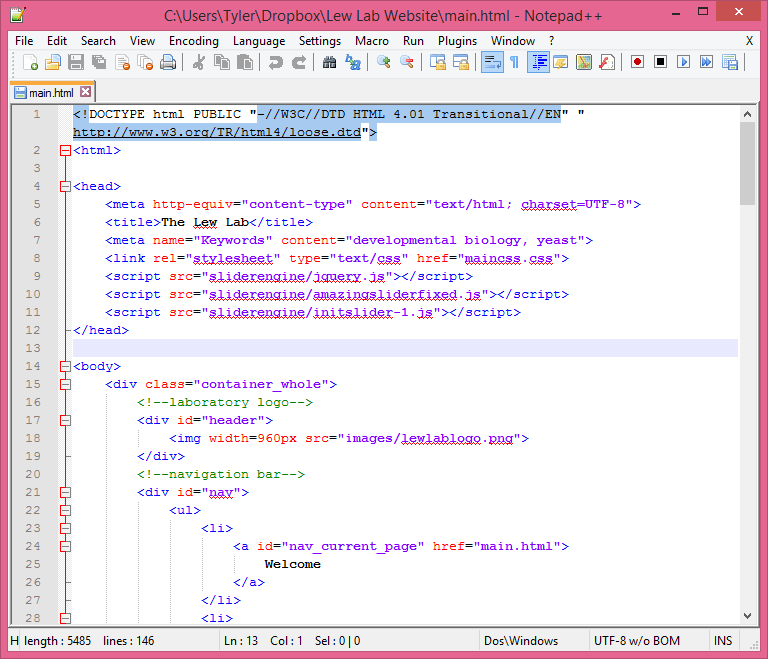
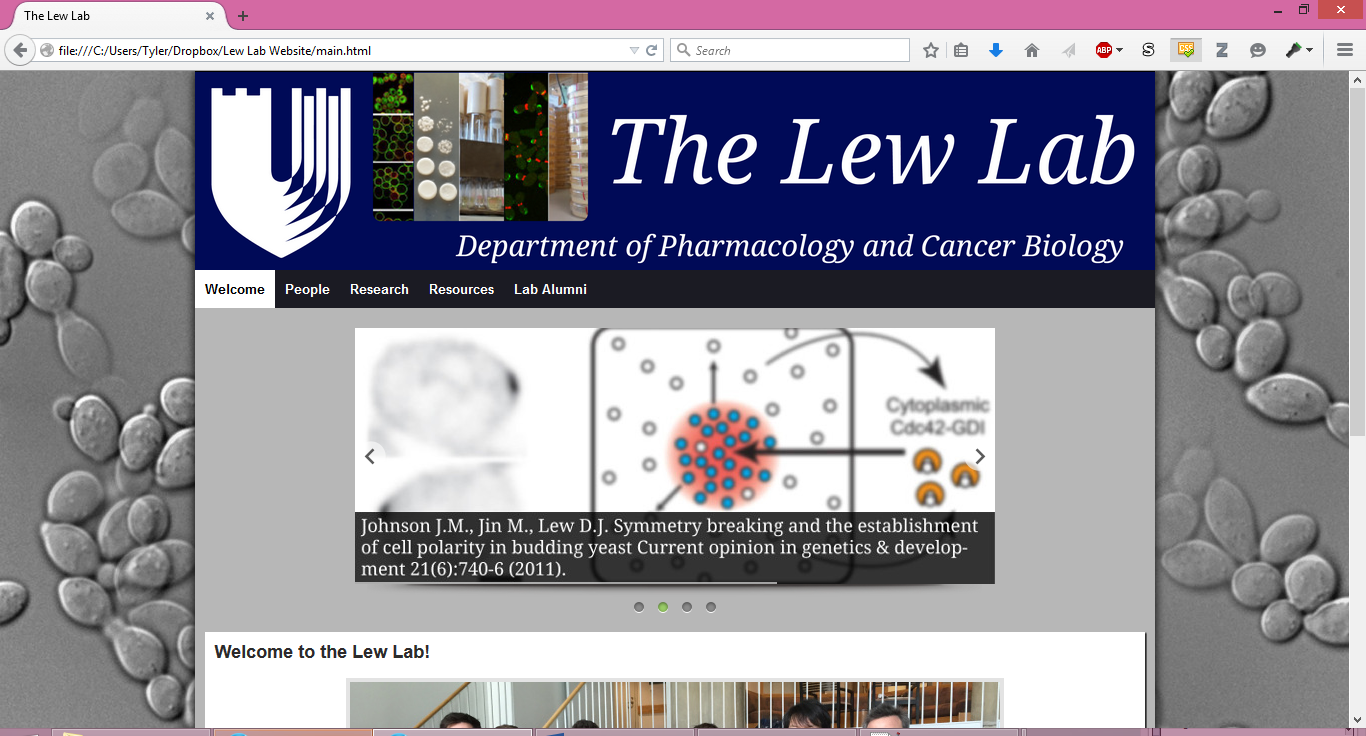


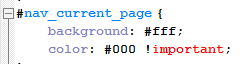
There are two primary sections to an HTML file: the head, and the body.



The **head section** (between the tags <head> and </head>) provides information for the browser which will not translate directly into content. For example, the title of the webpage is provided here (and is displayed on the tab in your browser).

The head section also contains the links to the .css file used to illustrate this document, as well as the .js files which run the slide show. You can see that the link to the .css file refers (via a tag called “href”) to *maincss.css*. The links to the .js files, on the other hand, refer to *sliderengine/jquery.js*, *sliderengine/amazingsliderfixed.js*, and *sliderengine/initslider-1.js*. This indicates to the browser that the .js files of interest will be in a folder named sliderengine, and is the reason we need to make sure that our files and subfolders are organized properly.

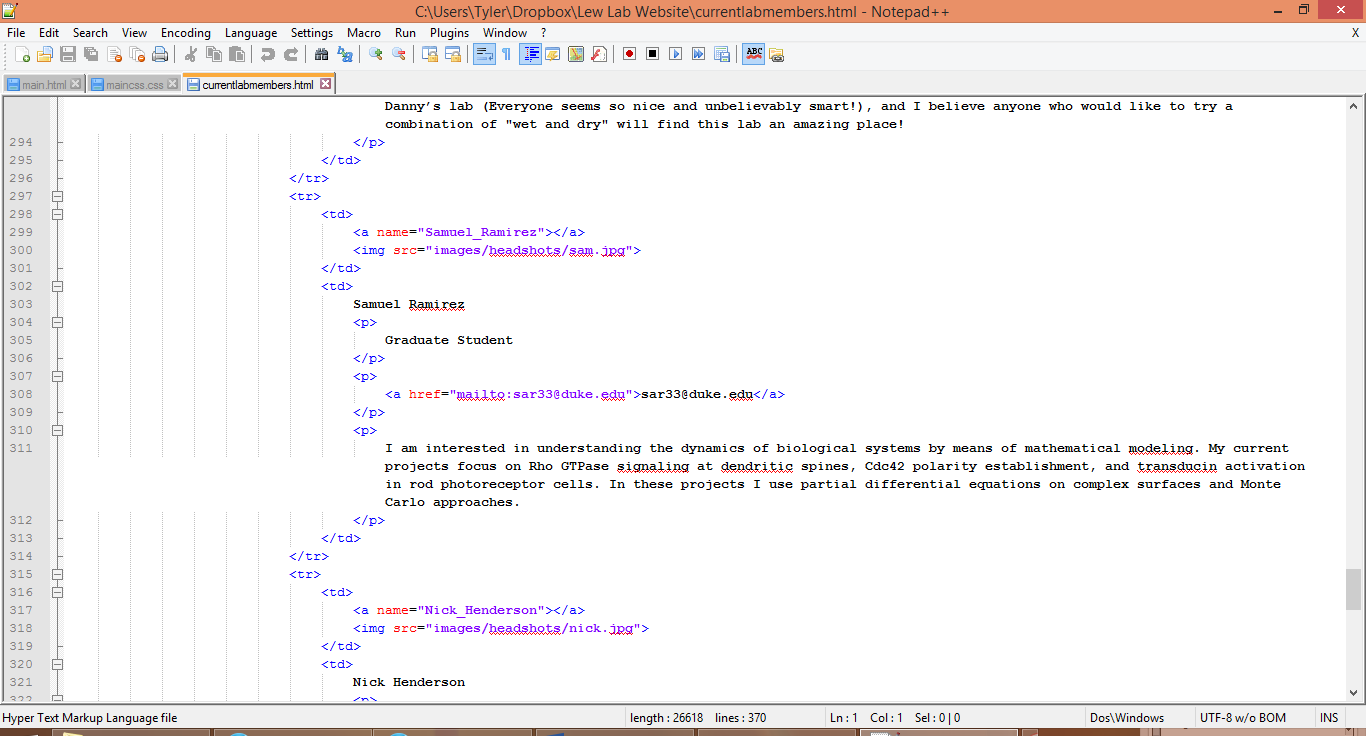
The **body section**, between <body> and </body>, contains all of the content displayed as a part of the page. It includes comments (in green; bookended by <!-- and -->) which are not read by the browser, but which provide instructions or indications to any developers about a relevant section. The image above shows the code relevant to displaying the logo at the top of the page, and the first part of the navigation bar just below it. You can also find the first piece of text on the webpage: “Welcome.” Text which will appear on the page as content is shown in black and white in Notepad++. “Welcome” corresponds to the first button on the navigation (nav) bar. You can click on any of the words on the nav bar and find that they will open the other webpages of the website. These links are created through the use of the anchor tag, <a>. Anchors are used to create hyperlinks, and have the standard format <a href=”linkedpage.html”>Click here!</a>. You can also link to web pages outside of the website by using that page’s URL: for example, <a href=”google.com”>Click here!</a>.

“Welcome” in the code above also has the **id** “nav\_current\_page.” “Id”s are used to apply rules from the CSS file to the text. In this case, “nav\_current\_page” indicates that the content to which it applies should have a white background and black text. You can check this by opening maincss.css with Notepad++ and scrolling to the rule *#nav\_current\_page*.

Try editing any of the code in the .html, saving the file, and then opening the changed document in your web browser to see the effect of your changes.

**Part Two: Changing Pictures and Content in the Body**

Open currentlabmembers.html in Notepad++. The main content of this page is actually contained in a series of tables with two columns. In the left column is a picture of a lab member, and on the right is a profile of that person. Each table is surrounded by a shadow (accomplished through CSS). The table is indicated by <table> and by <tbody>, while each row is indicated by <tr> and each cell is indicated by <td>. Below is the row indicating Samuel Ramirez’s profile. You can see the first cell and second cell.

The first cell contains an image link, indicated by the <img> tag. The <img> tag does not need to be closed like most other tags, and it pulls up a source image by looking for the .jpg name “sam” in the subfolder headshots, which is in the subfolder images. The image can be changed by modifying the *src* to point to a new location.

First cell

Second cell

The second cell contains information about Sam in the form of text. It is separated into paragraphs, indicated by the <p> tag. This moves each paragraph onto new lines. The text can be changed by directly typing new information into the .html. Keep in mind that only the text is used by the browser. Indentations, returns, or other formatting in the source code will *not* translate into the web page you see. Instead, we must use **attribute tags**, like those used in CSS, to tell the browser how to format the text. Commonly used attribute tags include the italicize tag, <i>, and the emphasis tag, <em>.

The break tag, <br />, is used to create a line of space in between one piece of content and the next. <br />, like <img>, does not need to be closed.

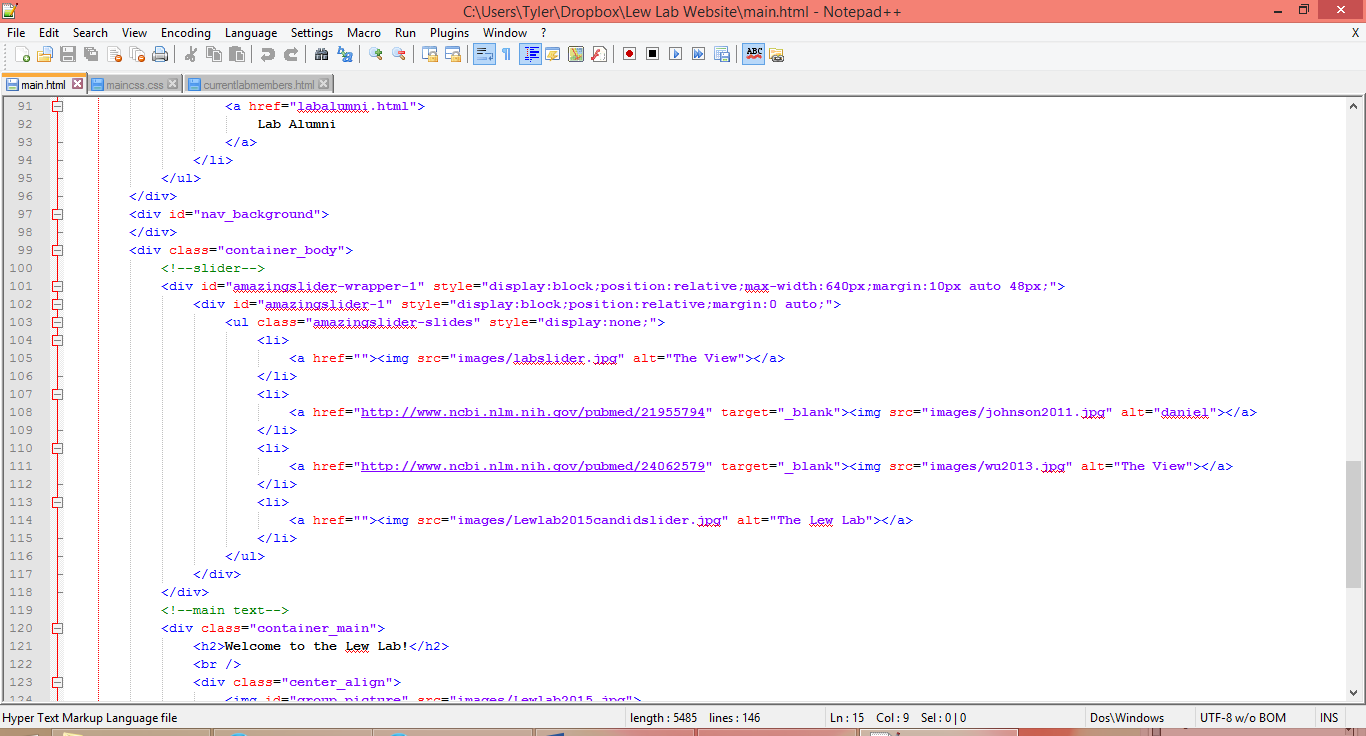
The second cell also contains a “mailto” link. When clicked in a browser, and if the user has an email service like Outlook set up, this link will open a new window to send an email directly to the address listed.

You may have noticed that the first cell also contains an anchor. This anchor “names” this section of the webpage *Samuel\_Ramirez*. This can be used to link directly to this section of the web page. If you hover over the “People” button on the nav bar, a dropdown menu appears. It is possible to select individual lab members by name and go directly to their profile on currentlabmembers.html. This is because the button “Samuel Ramirez” links to *currentlabmembers.html#Samuel\_Ramirez*. The # sign indicates a named anchor on the page.

To add another profile, it is easiest to copy another row, from <tr> to </tr>, pasting the row in the new location, and then modifying the image and the text to reflect the new profile. This way, the CSS will apply in the same manner. Deleting a profile is as simple as deleting everything from <tr> to </tr>.

**Part Three: Changing the Slide Show**

The slide show is a series of images which cycles sequentially, due to some JavaScript. Changing an image is as simple as changing the file referenced by the *src* part of the <img> tag. It is possible to make an image link to another page, as has been done in the example below, by surrounding the <img> tag with an <a href=”externalwebsite.com”> anchor.

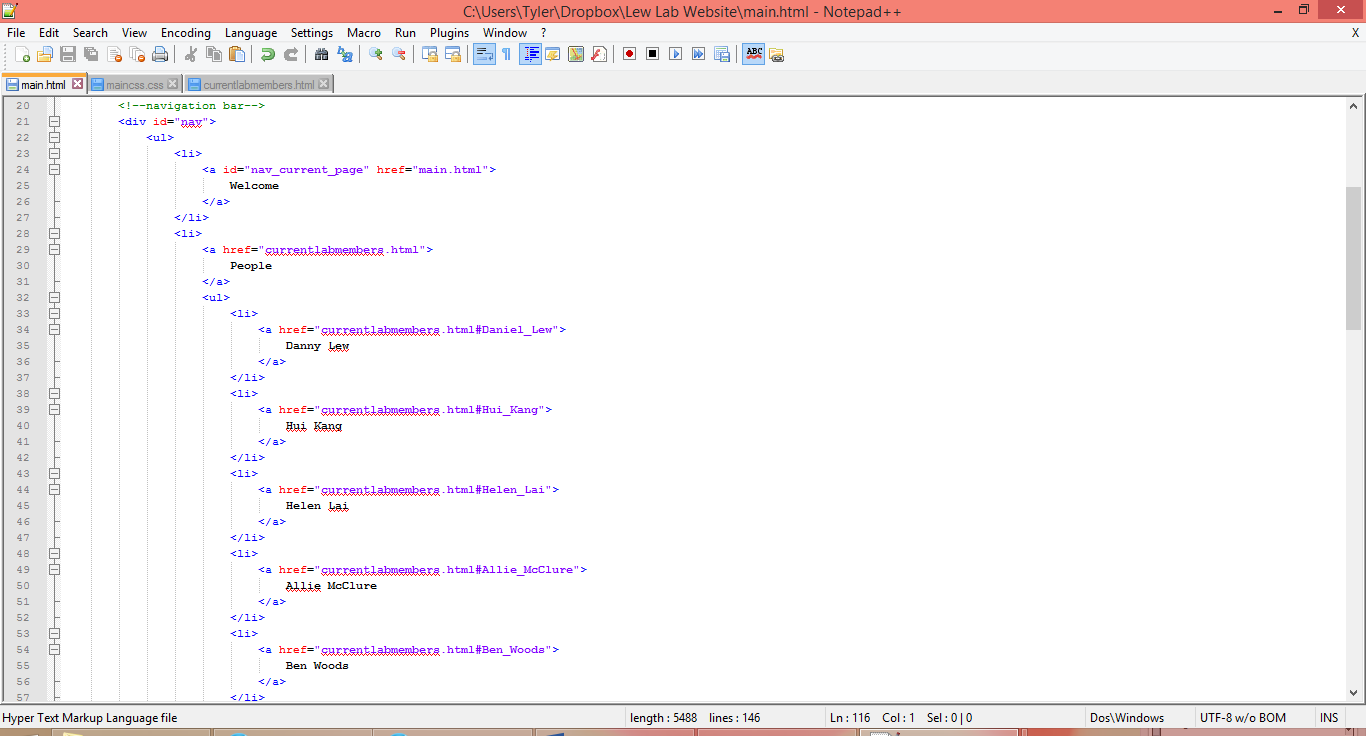


You will notice that each image is bookended by the tag <li>, and that all of the images are bookended by the tag <ul>. This is because these images are part of a “list.” Lists are a useful way of organizing information in HTML. To add more items to the slide show, it will be necessary to add another list item. This is most easily done by copying an existing item, from <li> to </li>, pasting that item in the desired location, and modifying its contents to point to a new picture and a new URL.

**Part Four: The Navigation Bar**

The navigation bar allows users to move from one webpage to another. The nav bar is actually present in all five .html files with only minor differences.

Like the slider images discussed before, the nav bar is actually a list. It has been modified through CSS to look like a series of buttons. The nav bar also contains a sub-list, also referred to as a “nested” list, under the item “People.” This nested list provides dropdown functionality, so that when a user hovers over the People button, a list of the lab’s members will appear.



To change any part of the nav bar, it is easiest to copy and paste a previously existing section at the same level. The content can then be changed in the pasted section. Once a change is made, be sure to copy and paste that change into all of the webpages on the website, so that the nav bar on every page is updated.

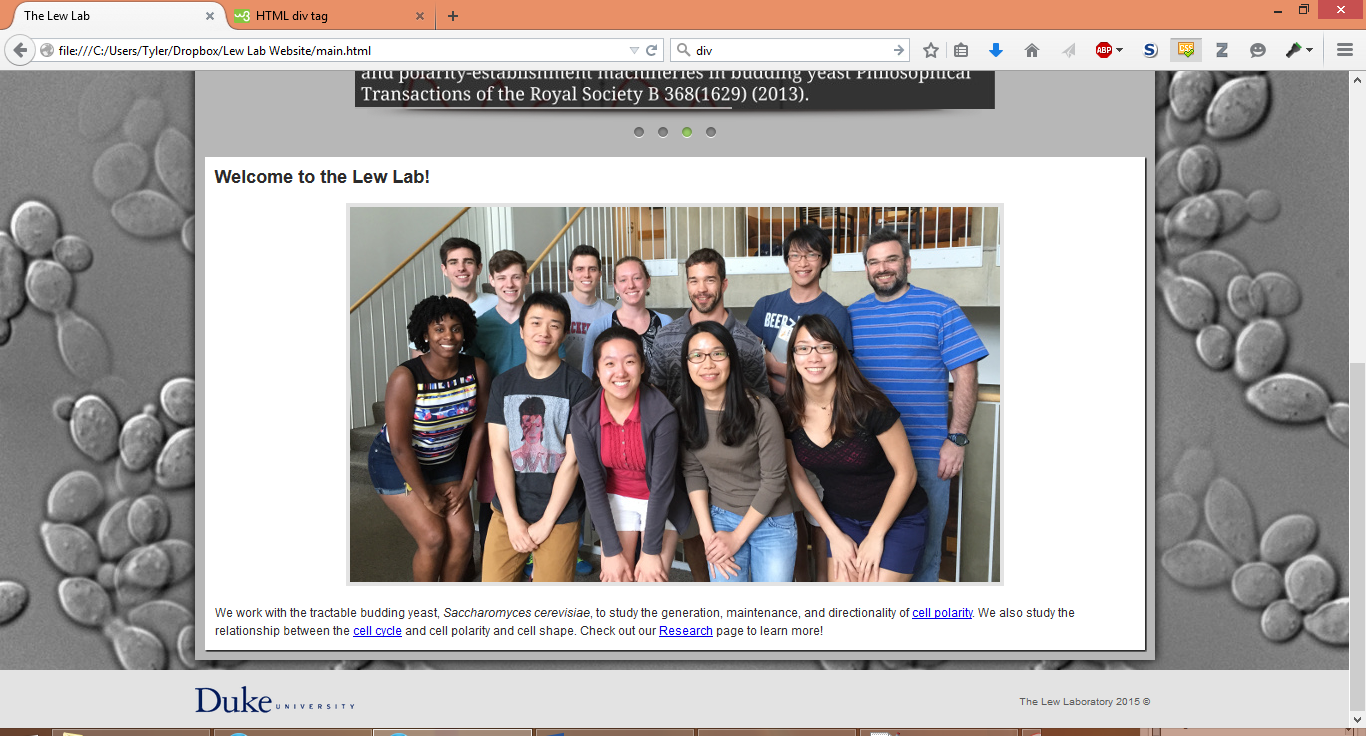
It is important to note that the nav bar of each webpage is *slightly* different. This is because the button which corresponds to the current page has a white background with black text, rather than a dark grey background with white text. For example, on the main page, the “Welcome” button is white, while on the lab alumni page, the “Lab Alumni” button is white:

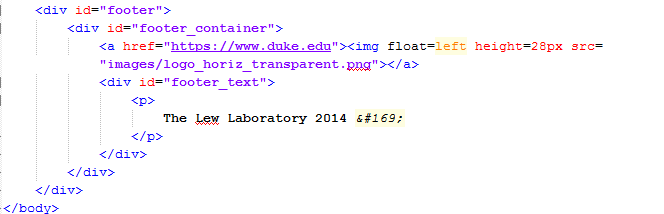
This is defined by the id “nav\_current\_page,” as discussed in section one. Make sure that this id is only included in the appropriate anchor on the page.

**Part 5: The Footer**

The footer is the grey strip on the bottom of the page which has a link to duke.edu and a declaration of copyright. The footer indicates to a user that he has reached the end of the content on the page.



The footer has its own section in the body, indicated by the division with the id footer.



Like the nav bar, the footer is present on every page. Consequently, any changes to the footer should be copied and pasted into all of the .html files, to keep the web pages consistent.

**Appendix A: Glossary of Terms**

**Source Code** – Instructions for a computer (in a human-readable, text-based language, rather than in a machine-oriented language like binary) which is transformed into the webpage you see. Source code may be written in any number of languages, but web development at a basic level will usually work only with the languages HTML, CSS, and JavaScript.

**Web** **Browser** – An application, like Firefox, Google Chrome, Internet Explorer or Safari, which retrieves and presents information resources, usually on the World Wide Web, and allows for the navigation from one information resource to another. An *information resource* is identified by a Uniform Resource Identifier (URI/URL) and may be a web page, image, video or other piece of content. These four browsers are the most commonly used, and each processes web pages slightly differently; as a consequence, it is good practice to run a web page in each browser before uploading the web page to the internet, to ensure that it looks and functions the same way for all users.

**HTML** (HyperText Markup Language) – The standard language in which source code for websites is written. Files of this type are instructions to a web browser in the form of tags (indicated by < > brackets which bookend source code). Browsers do not display HTML tags, but use them to interpret the content of the page. HTML describes the structure of a website semantically along with cues for presentation. HTML files can also synthesize instructions from other HTML files, CSS files and JavaScript files to save space in the primary HTML file.

**CSS** (Cascading Style Sheets) – The standard language used for describing the look and formatting of a document written in a markup language (like HTML). CSS is designed primarily to enable the separation of document content from document presentation, including elements such as the layout, colors, and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple HTML pages to share formatting by specifying the relevant CSS in a separate .css file, and reduce complexity and repetition in the structural content.

**JavaScript** – The standard and most basic programming language used in web design. JavaScript allows for the alteration of web page content dynamically, either autonomously or with user input. The slide show on the front page of the website, for example, operates through a JavaScript script.

**Appendix** **B: Standard content dimensions**

***Content Boxes***

The content of the page is 960 pixels wide. This includes the header (which contains the logo and the navigation bar) and the content below.

Each sub-content box is 20 pixels narrower than its parent content box. So the secondary content box is 940 pixels wide, the tertiary content box is 920 pixels wide, etc.

***Images***

Images in the slide show are 640 pixels wide and 272 pixels high

The logo is 960 pixels wide and 200 pixels high.

The standard headshot picture size (for currentlabmembers.html and labalumni.html) is 225 pixels wide and 300 pixels wide.

**Appendix C: Useful (Mostly Free) Software Packages**

[Notepad++](http://notepad-plus-plus.org/)

Notepad++ is one of many free source code editors which enables a user to easily edit source code. It will automatically color code different tags and instructions in any number of languages (including HTML, CSS and JavaScript), allow for simple modification of multiple pages at once, and quickly let a developer run an HTML file in any browser of their choosing. Notepad++ has many additional features and substantial online support via Google.

[GNU Image Manipulation Program (GIMP)](http://www.gimp.org/downloads/)

GIMP is a free image manipulation software package. It functions *very* similarly to Adobe Photoshop, and has all of the features one would need to crop, scale, and otherwise manipulate photos for use on a website.

Adobe Illustrator

Adobe Illustrator is a vector graphics editor used for making and editing images. Adobe Illustrator is unfortunately very expensive and is not necessary for most cases; however, *if* it is available, Illustrator provides a number of functions which a developer might find useful.

[Firefox](https://www.mozilla.org/en-US/firefox/new/)

A commonly used web browser.

[Chrome](http://www.google.com/chrome/)

A commonly used web browser.

Safari

A commonly used web browser. This will not be easily available to PC users, so try to test on an Apple computer.

Internet Explorer

A commonly used web browser. This will not be easily available to Mac users, so try to test on a PC.

**Appendix D: Online Web Development Tutorials**

<w3schools.com> offers an excellent series of tutorials for HTML and CSS and I strongly suggest that any aspiring developer work through at least the first HTML lessons to understand the basic workings of HTML. Even once a basic level of knowledge has been established, w3schools is the best online resource for information on the specific function and syntax of HTML tags and CSS attributes. You will return there repeatedly.