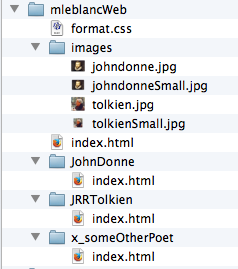
(X)HTML/CSS/JavaScript – “My Site” Due: Wed, February 5th

Create a (very) small website. The objective is not about the content on your website, nor is it about the graphics design on your pages. Rather, this assignment’s goal is for you to remove some of the mystery of how web pages are created, organized, linked, served, and ultimately rendered by a browser[[1]](#footnote-1). **You must pick a topic for your site as discussed in class.**

**Requirements**

*Your website* ***file structure*** *should look “like” the following:*



This is my CSS style for all files

This is my (default) home directory

Note: Since your website can be on a topic of your choosing, you obviously do *not* have to have this exact same structure, for example, you do *not* have to have a directory called “JohnDonne/”. However, you *do* need to create a similar directory structure with your top-level directory (folder) named with *YourLastName*Web/ and contains:

1. index.html – your “homepage” (Thus, on a server, any internet-enabled user should be able to reach your site at the URL:

http://**server**.wheatoncollege.edu/*YourLastName*Web/ ),

*where parts of the URL, e.g., server and YourLastName*Web *will obviously change.*

1. format.css – your cascading style sheet; all special formatting styles are done here; You must use at least two CSS features in your .css page. You should store your .css file in your top-level directory (your “home” directory); notice my format.css file above. Each page should refer to this one .css file. Extra credit: Display some content using the CSS Box Model (see Scott Kleinman’s tutorial for this feature of CSS as well as other items:

http://www.csun.edu/~sk36711/WWW/tutorials/css.html#box ).

1. inside your top level (sometimes called “home”) directory an images/ directory that holds *all* your images; all links to images must use relative addresses in the source (src) attribute (source file attribute of your <img> elements). Attributes are often referred like this: @src.
2. at least three (3) subdirectories, in my case one for each poet: JohnDonne/ and JRRTolkien/ and *x\_someOtherPoet/* . Thus a browser can directly reach my JRRTolkien poems at the URL:

http://**server**.wheatoncollege.edu/*YourLastName*Web/JRRTolkien/ ).

*Note: on an Apache web server, “index.html” is the default name for each directories’ “home”.*

1. each of the subdirectories (e.g., JohnDonne/ ) must contain an index.html file that contains some content.
2. As described more fully on page 3 of this specification (“spec”), *your* subdirectories should focus on at least two questions that your ask of Google’s Ngram viewer that are in some way related to your final project topic (see page 3 for more details on this item).
3. Your files should use only XHTML rules. I am assuming that each page has been validated at: **http://validator.w3.org/**

**Navigation** (moving around your website)

1. Your homepage (the top level) should contain at least one picture and link to second level directories.
2. Each of the second level content pages must contain links to the other pages (e.g., to jump directly from one poet to another) but of course should not contain a link to the (poet) page you are currently on. Each of these pages should also link back to your home page:

Go back to my **<a href="../index.html">**MYHOMEPAGE**</a>**

All hyperlinks within your site *must* use relative URLs, as shown above. In short, **the user *must* be able to jump between all your pages in an easy fashion**.As discussed in lecture, the URL need not contain the index.html filename, but that will depend on your group’s convention.

1. Again, *all* images, no matter on what page where they appear, must be stored in your /images folder. (Note: on larger sites, you would probably have multiple /images folders; however, for this assignment, all URL references to images must use relative references to your top-level /images folder).
2. Every .html file must contain JavaScript at the bottom to indicate a “Date Last Modified” message.

Your website (top-level folder, e.g., mleblancWeb/) must be zipped-up and submitted to onCourse by Wed, February 5th (so technically, Thursday morning, 4am).

See the next page for more specifics of what content should appear on *your* webpage. Again, your website will *not* be anything like my topic of John Donne *vs.* J.R.R. Tolkien.

(X)HTML/CSS/JavaScript – “My Site” **Due: Wed, February 5th**

The content of your website should be focused on a preliminary survey of the topic that you will explore for your final project in this course. As discussed in class, you will be working on your final project *in parallel, that is at the same time*, with other work. One way to think about your final project is that you will be continually working on it all semester. You should stay keenly aware of new content that cover, all the while asking yourself if this new material could be applied to your project.

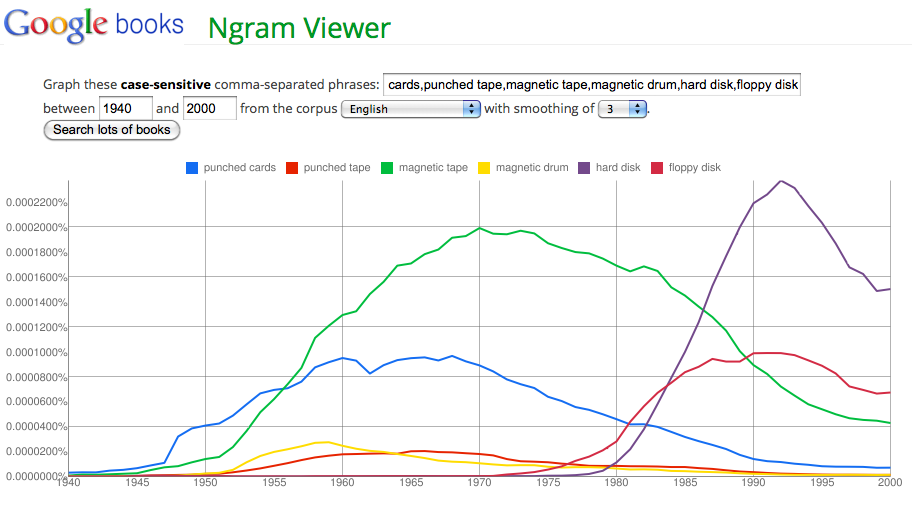
**This sheet is an amendment to the general instructions for your website on pages 1 and 2.**

Use Google’s Ngram viewer ( <http://ngrams.googlelabs.com> ) that explores two “culturomics” questions that pertain to your (potential) final project topic.

1. Spend some time reviewing the Culturomics website: <http://www.culturomics.org/>
2. Read the following article (found on the Culturomics site) introducing google’s ngram viewer and the term *culturomics*:

Michel, J.B. *et al.* (2011). Quantitative Analysis of Culture Using Millions of Digitized Books. *Science*, v331, Jan. 14, 2011, 176-182.

1. Ask two questions to the ngram viewer and capture/save the graphical output as .png or .jpg files. These two questions can serve as two of your directories in your website. More specifically, your questions could form the basis of your directories, replacing my generic example of Donne and Tolkien.
2. Store a description and rationale for each question that you posed to the Ngram viewer, the graphical output (.jpg), and your commentary of the output on your webpages.
3. Your site should have the directory structure as described on the first two pages. That is, your submission should have index.html and format.css pages, you should an images/ directory where *all* your images (such as screen shots of the Ngram viewer output) are stored, and your commentary of your two Ngram queries stored in separate, unique directories with of course accompanying .html pages.



1. A side effect of of learning “raw” XHTML is that you will own a level of confidence that later enables you to edit pages (say in WordPress) and automatically generate XHTML “on the fly” via a Python program. [↑](#footnote-ref-1)