

Lab 2

Part (-1): Before you begin

REMEMBER:

1. You know more than you think you do. Everything you need to complete the lab can be found either in your head, in your notes, in the slides (may also include Friday's slides, already available on LMS), or a very quick Google search away.
2. **BREAK EVERYTHING**. Break everything so badly you have no idea how to fix it. Be bold and boisterous and loud in your mistakes. More so than when you're 100% sure of something. After you've broken everything and experimented as much as you can, then go ahead and do the lab for real.
3. Don't overthink the directions. But do make sure you follow the directions as asked. Part of your grade for the lab is based on whether or not what you submit matches the specifications.
4. Don't forget about office hours! Both mine and the TA's.
5. You are working in groups. One submission per group. That also means that you should be producing a respectable amount of work for the number of people in your group.
6. Are you remembering to log into LMS every class to submit attendance?

Part 0: Setup

Create a new directory for Lab 2 if you have not done so already. It should be available at <http://lab2.websys>. Ensure that you have a text editor or other integrated development environment (IDE) ready. Which one you use doesn't matter, so long as you are comfortable and efficient in its use (or are willing to invest the time to learn it). If you don't know what text editor/IDE to use, Microsoft Visual Studio Code is popular (and free!).

Part 1: Semantic HTML

For this part of the lab, you will be marking up a playlist of your ten (10) favorite songs. Each entry must include the following information:

- * Track Name
- * Artist - with a link to the artist's website
- * Album Name - with a link to its page (if any) on Allmusic.com (or other similar site)

- * Album Cover (Image)
- * Release Date
- * Genre(s) - list multiple if appropriate

Think about the semantic correctness of the elements you're using, keeping in mind that your document might not always be rendered by a browser (how would you communicate this information to a blind person using a screen reader?).

To receive credit for Part 1: Name your HTML file *groupname-lab2-part1.html* and save it in your lab2 directory. Create a README file, and in it explain the reasoning for the markup you chose and explain how your HTML is semantically correct. Additionally, add a screenshot of your browser with the html page and save it as *groupname-lab2-part1.xxx* (jpg, png, etc.) in your lab2 directory.

Part 1: 3 files - HTML, Readme, and Screenshot (jpg, png, etc.)

Part 2: XML

Yes, I know we will focus on HTML for a lot of things, but you should have XML under your fingertips (who knows--you may want to use it in your projects or the future).

Create a new file, *groupname-lab2-part2.xml*, and mark up the same content that you did in Part1, this time in XML. Although you may choose whichever labels and attributes for your tags that make sense, take care to structure the document in a hierarchical, extensible way. Recall that XML may be valid, but must be well-formed.

If you are having difficulty with this part (and want your XML to neatly display on your browser), please look up XSLT:

https://www.w3schools.com/xml/xml_xslt.asp

Make a new file called "style.xml" and link to it by putting the following after the XML prologue:

```
<?xml-stylesheet type="text/xsl" href="style.xml"?>
```

NOTE: You are **not required** to have your XML for this part display neatly in a browser (but you can use XSLT to do it). Hence why no screenshot is required here (but... see part 4). You can put all the usual HTML tags you need for this in your XSLT stylesheet ``, ``, etc.

To receive credit for Part 2: Place your XML file in your lab2 directory. In the Readme file, explain the reasoning for the markup you chose and be prepared to explain how your XML is semantically correct.

Part 2: 1 file - XML

Part 3: HTML and CSS

Create a new file, *groupname-lab2-part3.css*. Edit your HTML file to reference the CSS externally using the `<link>` tag in the `<head>` of your document and save this as *groupname-lab2-part3.html*. This is always the preferred way to add CSS to an HTML document, as it further separates the presentation from semantic markup and promotes the reuse of CSS between documents. Make sure that song title is in small-caps, the artist is in **bold** and in a 20 percent larger font, the album is *italicized*, the release date and genres are in a proportionally reduced font size, and that the genres appear as a list with dashes. The album cover should appear to the right of each listing, and the title of your favorite song should appear in green. There should be a border around your list as well as an entry centered at the bottom indicating the end of the list, in a smaller, italicised font. Make sure that you add a header for each column indicating the content for each. You may not use tables for part 3. You can of course make a copy of your HTML file in part 1 as the base for part 3, but do make sure you don't overwrite the part 1 HTML file with the part 3 HTML file.

To receive credit for Part 3: Place your HTML and CSS files in your lab2 directory. Using comments in your file(s), and in your Readme file, explain the logic for the markup you have chosen and make sure your markup is well documented. Additionally, add a screenshot of your browser with the html page and save it as *groupname-lab2-part3.xxx* (jpg, png, etc.) in your lab2 directory.

Part 3: 3 files - HTML, CSS, and Screenshot (jpg, png, etc.)

Part 4: XML and CSS

Turns out you can use CSS in your XML documents!

According to the base specification, XML documents may also refer to CSS.

After the XML prologue:

```
<?xml-stylesheet type="text/css" href="x"?>
```

Create a new file called *groupname-lab2-part4.css*, and modify your XML document to refer to this new stylesheet. Edit the stylesheet such that the information should be displayed with each item on a separate line. The Song title should be first and should be slightly smaller than the ArtistName which should also be in blue. The Album name should remain in *italics*. The hyperlink should work, and the image should be last, not be part of the hyperlink, and must be forced to be the same size as the other images. Finally, there should be at least one line separating each entry in your list. This time, in order to be able to display your image and hyperlink using the HTML tags, you will need to add the html namespace to your XML file by adding an attribute to the root_element tag--which is done by adding

```
<root_element xmlns:html="http://www.w3.org/1999/xhtml">
```

to your XML file after the stylesheet directive. Alternatively you can accomplish this without using the img element. However, you will need some other means of rendering the image on the page using CSS (hint: look into how backgrounds are displayed).

Take note of what happens with the margins: one might assume that two elements vertically separated by opposing margins, the one with the greater value will apply, and the other will collapse to zero. This rendering behavior is known as “collapsing margins,” and is used to maintain vertical rhythm when elements are in the normal flow of the page. Elements outside of the normal page flow (those that are floated, positioned absolutely, etc.) will not have their margins collapsed.

To receive credit for Part 4: Show that the appropriate changes were made using your CSS. Add the two *groupname-lab2-part4.css* and *xml* files to your lab2 directory. Add a screenshot to the directory named *groupname-lab2-part4.xxx* (jpg, png, etc) as well. As above add any comments to your Readme file.

Part 4: 3 files - XML, CSS, and Screenshot (jpg, png, etc.)

Grading rubric:

10 points - Part 1

10 points - Part 2

10 points - Part 3

10 points - Part 4

10 points - Creativity

(What is creativity? Creativity is going above and beyond what is being asked of you in *some meaningful way*. Cater to your strengths. For some of you, creativity may be wowing us with HTML page design. For others, it will be interesting usage of CSS. Others might go as far as to begin to define their own mini-ML in XML. It doesn't, nor should it, need to be everything. It just has to be something, and it has to be enough of that something to show us how you are beginning to grow and develop into your own unique web developer. It would be good to tell us about those things in your Readme file.)