HAJIM SCHOOL OF ENGINEERING & APPLIED SCIENCES
UNIVERSITY of ROCHESTER

Department of
Computer Science

CSC 170 Project 2: CSS for Formatting & Layout

Due: June 11, 2015

In this assignment, you will work with the HTML document you created in Project 1 and style it using CSS. The result will be a **formatted document** that uses some version of a **columnar layout**.

Warning: Do not begin Project 2 unless your Project 1 is perfect (or as perfect as you can get it) because any points you lost in Project 1 will also be lost in Project 2 until you fix them.

Requirements

The HTML document:

- ❑ You must use the final version of your HTML document you created for Project 1.
  - Project 1 and Project 2 will be compared against each other to make sure the content is all there and similar.
  - The intent is to leave the Project 1 content as-is, as much as possible.
  - Do not add or delete **content** from your Project 1 webpage. However, in the HTML, you can add elements around existing content, e.g. wrap content in a new , or better yet, use HTML5 elements like <article>, <aside>, <main>, <section>, et cetera.
  - You can also add attributes to existing elements. For example, you can change a <p> into <p *class="lead"*>. But all the rest of the tags and content from Project 1 must remain as-is, as much as possible.
  - If need be, you can shuffle content around in the HTML document – move blocks of content from one area to another, but only if you really have to.

- ❑ The **title** in the <title> tag (in the <head>) must be updated to say, "**Project 2 -** " followed by your topic, for example: `<title>Project 1 - Marc Andreessen</title>`

The CSS document

- ❑ You must create an external CSS document and link it from your HTML document.
  - You may not use any inline or embedded CSS styles, or old fashioned HTML attributes that format content in any way.[1]
- ❑ You must clean up your CSS. There must be no unused CSS statements (selectors or declarations) cluttering up your CSS document.

---

[1] Except <table **border="1"**> is still okay.

Formatting

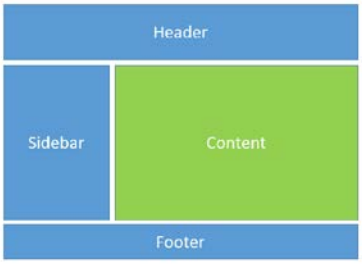
- ❑ The CSS must apply an ample amount of formatting to prove that you know how to style an HTML document using:
    - Fonts (typefaces, font sizes, font weights, and other typography settings)
    - The box model (margins, paddings and various borders)
    - Other embellishments – whatever you can find! Pseudo-classes, background images and background colors.
    - DO NOT USE the code provided from lab assignments as-is. You can refer to them and learn from them (obviously) but do not simply use that code in Project 2 without customizing it.

Container

- ❑ In addition to side-by-side content, your document must use a "container" to constrain its content in some way and keep it "floated" in the center of the web browser window. You don't have to constrain *all* your content (although that's okay if you do), but you must demonstrate good use of the trick.

Layout

A **columnar layout** is any layout where some of the content is positioned side-by-side.



A columnar layout is *more than* floated images where an image or figure is positioned left or right and text is allowed to flow around it. (That's just a float – which is okay to include – but it doesn't constitute a columnar layout.)

Note: navigation is *not* required for Project 2, but if you want to try an internal "jump" list, that's okay.

- ❑ At some point in your web page, you must break the normal document flow and position content side-by-side to create a columnar layout. (E.g. the c-clamp.) You can use any of the approved following layout techniques:
    - Inline-block layout
    - CSS table layout ← easiest
    - Flex layout

Usability and Aesthetics

The webpage appearance must be usable (readable) without strain.

- ❑ Proper contrast between foreground and background must be maintained to make the webpage easy to read
- ❑ Fonts (type, size, style and color) must be appropriate for all screen sizes.

- ❑ Ample use of the box model (margin, padding, borders) to create white space must be used to separate areas of content to into clearly delineated sections.
- ❑ Your layout must work on all normal desktop sizes
  - Part of the grading process will include shrinking and stretching the browser width to see what happens to your design. Your layout must "hold together" (make sense) at any width between 720px and 1200px.

At small browser widths (720px)…

- There must be NO horizontal scroll bars.
- The content should still be usable and look normal.
- Content must not "crash" – e.g. images, figures or tables must not hang over edges or cause nearby text to become broken or hard to read.

At larger browser widths (around 1200px or more)…

- No line of text should be so long it becomes unreadable

- ❑ Your design (formatting and layout) must have a subjectively pleasant aesthetic. Although artistic capabilities are not being graded, the webpage must not be "sloppy".[2]

Location

- ❑ Your one-page web site must be mounted on the class web server, within your **webroot** folder where you'll create a new folder that must be exactly this: **project2** …all lower case, spelled exactly like that.
- ❑ The name of the HTML document must be exactly this: **index.html**
- ❑ The image(s) you use must be in a subdirectory of the **project2** folder named **images**
- ❑ The CSS documents you use must be in a subdirectory of the **project2** folder named **css**
- ❑ Validate the HTML file (**http://validator.w3.org**)
- ❑ Validate the CSS file (**https://jigsaw.w3.org/css-validator**)

Report your work

To receive credit for this project:

- In our Blackboard section, in **Project 2**, post a link to your webpage.

---

[2] We will be very flexible when it comes to judging aesthetics. Only really, *really* ugly or sloppy designs will be penalized (unless you're being ironic, in which case you need to make it clear that you're being so.)