# Project No. 3

## CSS

### DUE: Friday, Nov. 17, 6 p.m.

Using a corrected version of the HTML file you created for the HTML project, style the page using CSS. You have creative freedom in your design choices, but they must adhere to the principles introduced in the readings from "Hello Web Design."

## Content Requirements

You are *not* required to use submit the exact same HTML as you did in the last project. You can change the HTML as needed to execute a design. You may even want to re-do the HTML completely (but you are not expected to).

### Images Reminder

If you add more images to your page, you still must have permission to use them. The images included on your page must be included in your GitHub repo. You may not hotlink to images on the cia.gov web site. (I *still* recommend limiting yourself to images on CIA World Factbook as using copyrighted material will earn you a zero (0) for a lapse in web ethics.)

## Grading

Points are awarded based on correct execution of the tasks below. All requirements in each lower percentage 'tier' must be met to earn points at *that level and higher levels*.

For instance, if you fail to use a basic font stack, the most you will earn is a 60 — even if everything else is correct. You must be able to demonstrate the most fundamental aspects before showing you have mastered the more challenging ones!

I try to keep the grading rubric below focused. There are often questions about the items noted with asterisks. Please read on for more detail in the **FAQ & Tips** section.

**100%**

- Exhibits strong visual design *

**95%**

- Correctly uses CSS Grid to lay out the page

**88%**

- Sets styling for *all* link states using pseudo-classes *

**80%**
- Uses a full, working font stack with two (2) Google Fonts and a "web-safe" font stack *
- Uses the box model in a logical way (`margin`, `border` and/or `padding` properties)
- Design limits text to display no more than 60 characters *

**75%**
- Uses colors in hex values, `rgb`, `rgba` and/or `hsl` format (color keywords are *not* acceptable)
- Uses a basic "web-safe" font stack *
- Files are organized using a functional directory structure *
- Uses relative paths for linking files, pages within your site
- Does not use the `<style>` or `style` attribute in html

**60%**
- Uses a working, external style sheet (change from the default browser style *must* be visible)

Failure to meet the lowest tier will earn a zero (0).

### HTML Requirements
Even though this project focuses on CSS, HTML still matters. Your project *must* meet the following requirements:

- uses a proper document structure (i.e. `doctype`, `html`, `head`, `body`, `meta`, `title`)
- validates without any errors
- file names follow proper naming convention
- images are saved in the correct file format
- no broken images or links

Failure to meet all of these requirements will result in a 5-point deduction.

If you submitted the first project on time, you have received feedback on Blackboard. You have had many live sessions and hack hours to ask questions to improve your understanding. Mistakes are expected as you learn, but it is not acceptable to repeat an error after you have been made aware it.

## Submitting Your Work
Submit the following under **Assignment Submission** portion in the "CSS" module:

- a link to your *working* web page (i.e. https://ebrown15.github.io/canada.html). Include this in your **Text Submission**.

**Do not make changes to your files after the deadline.** Commits after the deadline will be considered late. (I check logs.)

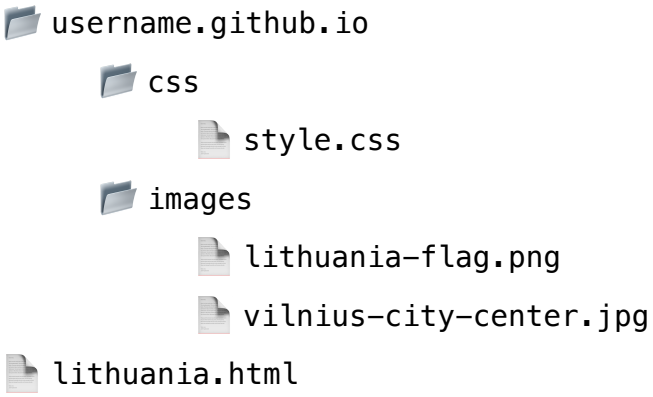
## FAQ & Tips

You are expected to have content (it's hard to create a design without content), but it may be changed. A few examples:

- *If* your design has all content "above the fold" (eliminating the need for a visitor to vertically scroll), the in-page navigation (<nav>) serves no purpose, so you may delete it.
- *If* your design has multiple rows in a grid, but the content within columns is vertically uneven. Deleting a subsection of content (i.e. "Location" in "Geography") will balance out the page. This is fine.
- *If* the images on the page would work better in a different section of the page, you can move these around to make the design work as you intend.

### What's a "functional" directory structure?

Within your root directory (`username.github.io`, where `username` is *your* GitHub username), you will have your html file, a css folder and an images folder. Your css file will go inside the css folder. Your image files — gif, png or jpg files — will go inside the images folder. Your directory structure will look similar to:




(Of course, this *requires* use of relative paths in `link` and `img` tags in your html file for everything to correctly work!)

### What is a basic font stack? What is a full font stack?

A *basic* font stack should include web-safe fonts and a generic fallback. A *full* font stack will include web fonts (from Google Fonts in this case) along with web-safe and generic fallbacks. In order to earn credit for a full font stack, the web fonts you have chosen *must* appear in your web page.

### What is meant by, "design limits text to display no more than 60 characters"?

No, I do not count every character. Anywhere there is text, it should not exceed a width of about 60 characters. For instance: if we use 16 pixels for our font size, the text should not be in a block wider than 960 pixels (60 characters x 16 pixels = 960 pixels).

The objective here is to keep you mindful about screens other than your own! If you only design for your 12" laptop screen without putting some constraint on your text, when a visitor with a 27" desktop screen visits, it will be very uncomfortable to read. The text might look fine on your screen, but it won't on another.

### What do you mean by, "uses box model properties in a logical way"?

If you are trying to purposely trying to demonstrate good design practices this will not be an issue. Logical uses of box model properties include (but are not limited to) keeping text aligned; adjusting whitespace to establish stronger relationships among content; centering page content; using borders to better define the boundaries of images, etc.

### What link states do I have to style?

*All of them*. You are expected to style your links in their default, unvisited, visited, hover and active states.

### What qualifies as "strong visual design”?

The reading from “Hello Web Design” went over a number of simple principles which can be applied with just the CSS we have learned. Your project is expected to meet these requirements in full to earn a perfect score. The things I consider as I review your work are:

*Alignment/Grid*

- Do the elements of the content line up with one another?

*Color*

- Is the color palette appealing and effective?
- Is there enough contrast for readable, accessible text?

*Typography*

- Are the fonts legible?
- Is the line height comfortable?
- Does the design avoid using centered/justified text?
- Is the line length comfortable? (i.e. 60 characters)

*Whitespace*

- Is there clutter?
- Is the space among like elements smaller?
- Is the space among differing elements larger?

It is okay to keep your design *very simple* in order to demonstrate these basic principles. Keep in mind the CSS Grid exercises, how we always finished up by adjusting margins and padding to clean up the whitespace on the page. These are the kind of steps you would take to fine-tune your design to make it presentable.

### May I delete the contents of my username.github.io repository?

You may delete the contents of the repository, *do not delete the username.github.io repository*. Some students prefer to start with a clean directory, especially now that directories are being added.

### Wow, this is a lot to remember.

I realize that. This is why I recommend printing the rubric and using it as a checklist. Don't feel overwhelmed. When you finish your first attempt at the project, go through the sections and mark each requirement you have met. Then, fix your project to meet the requirements you have missed.

If you are not familiar with a requirement (maybe you don't remember how to write an rgb value?), it is perfectly acceptable to look up the answer. Use your notes, your textbook or Mozilla Developers Network as resources!

### Any other tips?

Pay attention to detail.
Pay attention to detail.
Pay attention to detail.

Seriously.

We have to strive for perfection because computers demand it. Whether in a simple web page or a massive software application running a data center, one mistake in code can cause everything to break. For example: Facebook's hours-long outage on October 5, 2021 — caused by an error in a configuration file — cost Mark Zuckerberg an estimated $6 billion to $9 billion dollars.

Attention to detail is very important. 🙂