Project 1 – GitHub & Personal Profile Webpage

In this project you’ll accomplish the following:

* Build a basic professional profile web page
* Host it on the internet somewhere.
* Learn to work with git and GitHub.

Learning objectives for this assignment are to demonstrate the following skills:

* Understanding of HTML5
  + Basic formatting elements
  + Lists
  + Headings
  + Images
  + Links
* Appropriate project structure
  + Follow standard naming conventions on all source files.
  + Organize files into folders and follow proper naming
* Using git and github
  + Create a repository
  + Use basic git commands to commit your files and push them to github
* Basic understanding of what a web server does
  + Use github.io to view your finished project.

NOTE: I don’t expect a lot of fancy styling here. The main focus of this project is to learn to use git and GitHub. Keep things simple.

# Project setup

Begin by creating a profile on github.

1. Go to <https://github.com>
2. Create an account.
3. Don’t create a repository yet. I’ve got instructions for that, in a bit

## Install Git

Install the software for git on your computer.

<https://git-scm.com/downloads>

Launch the installation wizard, and use all the default settings.

Once installation is complete, open up a command prompt or terminal window

* **Windows:** Click the start menu, type cmd, then enter.
* **Apple:** (two options)
  + Click the Launchpad icon in the Dock, type Terminal in the search field, then click Terminal.
  + OR: In the Finder, open the /Applications/Utilities folder, then double-click Terminal.

Type the following command to make sure that git has successfully installed:

git --version

You should see something like this:

C:\Users\tabbott>git --version

git version 2.17.1.windows.2

C:\Users\tabbott>

Now type the following two commands, supplying your name and email address. Your name must be in double-quotes:

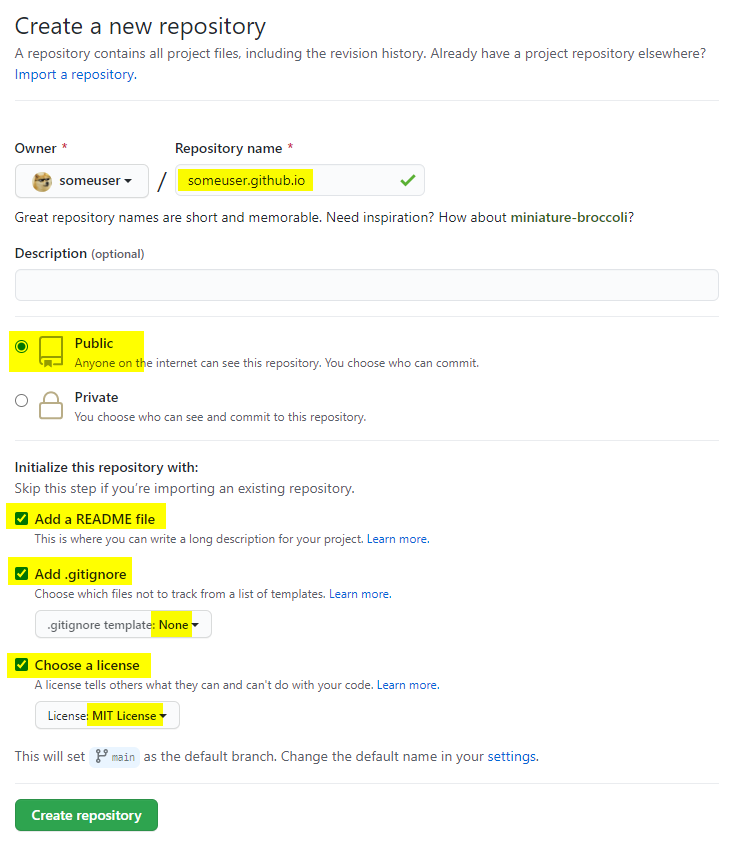
git config --global user.name "Your Name"

git config --global user.email your-email@whatever.com

Git will run both of these commands and print out no response if it was successful.

## Create a repository on GitHub

Log into GitHub and create a new repository. On your main homescreen, in the upper left, you’ll see a green button to create a new repository. Click it, and fill out the form, as follows:



**NOTE: If you have another hosting service that you’d like to use besides github.io, you may use that instead. To be clear, you do not have to host your new page on github.io. That said, the purpose of this assignment is to create a webpage and upload it to a server. Your submission for this assignment will be a URL to the page that you’ve created.**

Once you’ve created the repository, open up a command window

1. change directory to the folder where you keep all your projects on your hard drive,
2. type the git clone command
   1. supply your github username
   2. supply the name of the repo that you just created.

Here is an example:

cd <your-projects-folder>

git clone https://github.com/your-username/your-username.github.io.git

Here is an example of what I did for my repo. I’ve highlighted the two lines that I typed in yellow:

C:\>cd users\tabbott\projects

C:\Users\tabbott\projects>git clone https://github.com/twabbott/twabbott.github.io.git

Cloning into 'twabbott.github.io'...

remote: Enumerating objects: 23, done.

remote: Counting objects: 100% (23/23), done.

remote: Compressing objects: 100% (17/17), done.

remote: Total 23 (delta 6), reused 21 (delta 4), pack-reused 0

Unpacking objects: 100% (23/23), done.

C:\Users\tabbott>\_

You should see something that looks like the output above. My new repo now lives on my hard drive at

c:\users\tabbott\twabbott.github.io

NOTE: When you clone a repo, git will always create a new folder that has the same name as the repo you’re cloning.

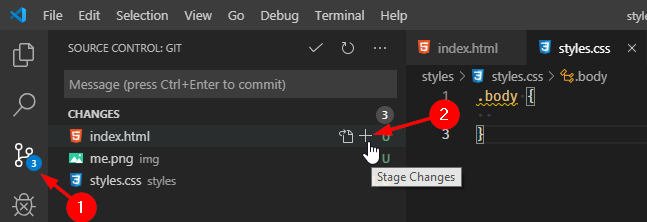
If all this is still confusing, I made a video. Watch the video and it should make sense.

## Commit often

As you work, remember to commit often. This is a good habit to get into, as it will allow you to roll back your changes if you mess up.

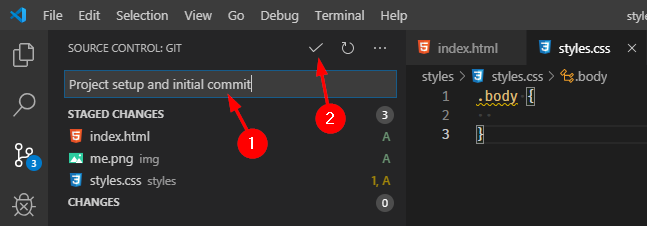
If you don’t like working with the command line, you can commit your work using VS Code. Click the source control icon on the left. You will see all the files that you’ve changed.

As you mouse over each file, a + icon will appear. Click the icon to stage each file that you want to commit.



Once you’ve staged all your files for commit, type a message. This is a note to your future self so that you can keep a journal of what you did in each commit. This can be as short or as long as you like, but it should describe what you did. Don’t write a novel; one sentence will suffice.

When you’re done, click the check-mark icon at the top of the panel. This will commit your changes locally. The source control panel will clear out and reset to a blank list.

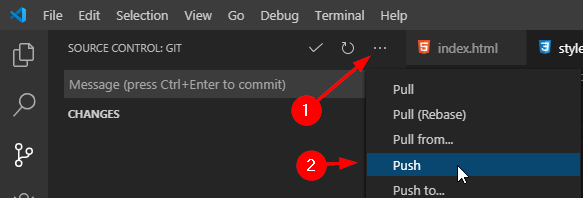


You should do this often while you work. You don’t need to commit every 5 minutes, but it’s definitely a good idea to commit every time you complete a section of work.

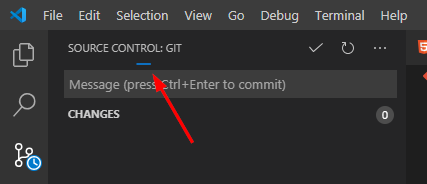
## Push your code to GitHub

Once your project is finished, you’ll need to perform one final step and push to github.

1. On the source control panel, click the elipses menu (The … icon).
2. From the menu, select Push.



You’ll see a busy-indicator slide along the top of the source control panel while VS Code does the push. When your code has been pushed up to GitHub, it will go away.



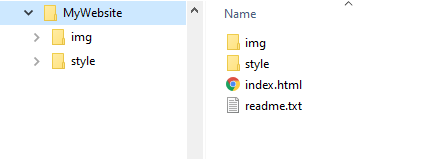
# Create an about-me webpage

## Project folder and file structure

Follow industry-standard conventions for creating a folder structure for your project and naming your files:

1. Your project’s main HTML page should be called index.html and it should live in your project’s root folder.
2. If you have any images, put them in a sub-folder, called images, or img, or pictures, or something along those lines.
3. If you have any external css files, put them in a folder called styles, or css, or anything along those lines.

When finished, your project’s folder and file makeup should look something like this:



## Related files

You need to have the following project related files. GitHub will create these files for you if you followed my instructions above

### .gitignore file

Your .gitignore file tells git which files or file-types it should ignore. This file should support multiple platforms (Windows and Mac).

Here is a basic template. You can add anything else you like, but it should include this:

# Windows

Thumbs.db

Desktop.ini

# Mac

.DS\_Store

# VS Code

.vs

.vscode

# Webstorm

.idea

### README.md

This file should contain your name and a general description of what styling items you decided to add to your project. See section 2.4 (Styling and appearance) for more info.

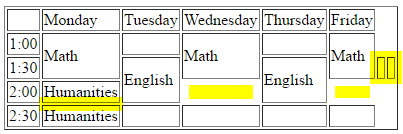
DO NOT mention this course name or course number (CS 2550)

## Content

Your web page must have the following, in this order:

1. **Folder structure (10 points)**
   1. You must call your file index.html. It must live in your root folder
   2. Put all images in a folder. Name it something that makes sense.
   3. Put all CSS files in a folder. Name it something that makes sense.
2. **HTML header (5 points)**
   1. Make sure you set it to tell the browser you’re using HTML5
   2. Set the title, which appears on the browser’s title bar. The title should read: About <your-name>
3. **At the top of the page (5 points)**
   1. Put your name in a header, level 1.
   2. Put a recognizable picture of you.
      1. Please use a paint program to resize and crop your image so that it is no larger than 500 by 500px (It does not have to be square).
      2. If the image looks squashed or stretched I’m going to take off points.
      3. The image should live in the same folder as your HTML page.
4. **Your personal info (5 points)**
   1. Have a header (level 2) that says: “Personal Info:”
   2. Next, use an unordered list (bullet points) and give me the following pieces of info:
      1. Your major (spell it out. Don’t abbreviate).
      2. Your high school, and the year you graduated.
      3. What year are you in, currently, at UVU.
      4. Your favorite flavor of ice cream (mmmm, ice cream).
   3. Make use bold text for each label like in the example that I provided below.
5. **Five favorite books (5 points)**
   1. Have a level-2 header titled “My five favorite books”
   2. Choose five books that you like the best, and list them in the order of your preference.
   3. Each title should be italicized. Each author’s name should be in regular text.
   4. Add hyperlinks. For each of the five books you listed, make the author’s name a hyperlink to the author’s website.
6. **Compose a haiku or a limerick, or other short poem (3 or 4 lines) (5 points). Please keep your** poem PG-rated (no potty humor, no sexually suggestive stuff).
   1. Have a level-2 header with a title of your poem
   2. Have a sentence or two explaining your inspiration for the poem.
   3. Put your poem after the header.
   4. Each line should use line-breaks, not paragraph breaks.
7. **Brief essay (5 points)**
   1. Make a level-2 header titled something like: “My personal interests”
   2. Give me three paragraphs, with three or more sentences each. In each paragraph talk about something you’re interested in.
8. **Class schedule (10 points):** I want you to create a table that shows your weekly class schedule. This is easy to do using table elements, but it does require some attention to detail.
   1. Make a level-2 header titled “class schedule”
   2. Create a table with 6 columns.
   3. Set the border size to 1, so I can see the borders for each cell. Set the cell spacing to 3, and set the cell padding to 3.
   4. The header row should have a blank cell, followed by cells for Monday, Tuesday, Wednesday, Thursday, and Friday. If you have a Saturday class (you poor soul), you can add a sixth column (or not, I don’t care, really).
   5. Going down the left side of the table, I want you to list out the hours of your week-day in 30-minute increments. You can start as early as you feel necessary.
   6. List the classes you have as cells. Use the row-span element to block out as many 30-minute cells as you need for each item. If you have stuff that starts at an odd time, then guess. I won’t care, as long as it doesn’t look all wonky.
   7. Fill up your schedule. Make stuff up if you have to. I want to see at least four classes that run M-W-F or Tu-Th or every weekday. In addition to classes, You can put your employment, your work-out time, your commute time, or anything else you like.
   8. Use the rowspan attribute, so if you have an hour-long class, the cell should take up two rows, without a line between them.

When doing your weekly schedule, make sure your table doesn’t show any of the following errors or I will deduct points:



Extra blank cells at the end of a row

Block of time that takes a whole hour, but uses two separate cells

Missing cells

## Styling and appearance

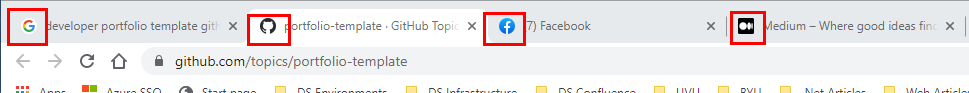
This is as much an art project as it is a programming project. Web programming (especially “front-end development”) is very visual and requires a strong artistic sense.

That said, this project needs to be done in a week, so I don’t expect a whole lot.

However, I do want you to spend some time styling your webpage and making it look attractive. If you do nothing more than the basics, you will only get 70% on this assignment.

**You must list out which items you did in the README.md file for this project.**

1. **(5 points) CSS3 Web Fonts.** W3Schools has an in-depth article on this.
   1. Use a sans-serif font. I don’t want to see Times New Roman anywhere on your page.
      1. **This is Times New Roman.** Serif fonts are hard to read on computer screens, and cause eye-strain on anything less than a 4K monitor. Fonts like Times New Roman are great on a printed document, but not so good on an LCD display.
   2. You can use fancy fonts for your name at the top of the page, and for the section headings.
   3. If you do a special font for your body text, it must be a font that is easy to read. Please don’t make my eyes bleed. No fancy script fonts, gothic-style fonts, etc.
2. **(5 points) artistic portrait**
   1. Hero image. Google this; there are tons of websites that explain how to do this. Your hero image should take up at least 50% of the page, and should scroll up (i.e., it is NOT just a page background. I will not give credit if you make your portrait image the background for your whole page.
   2. Have your photo offset in some way using positioning or float or flexbox. Here are some attractive ideas: <https://colorlib.com/wp/html5-resume-templates/>
   3. At the very least, can you right-align your photo and have the text flow around it?
3. **(5 points) background parallax effect.**
   1. This is easier to explain with an example. Go to W3Schools for more info. You can see a demo [here](https://www.w3schools.com/howto/tryhow_css_parallax_demo.htm).
   2. You can apply this effect with your portrait at the top of your page, or anywhere else on your page.
4. **(5 points) Put a nav menu somewhere on your page.**
   1. Clicking on the nav menu should scroll the page to that particular section.
   2. If you don’t know what to put, you can have the following items:
      1. Home: takes you to the top
      2. Reading: takes you to the favorite books and the poem
      3. About: takes you to the blurb about yourself.
      4. Schedule: takes you to your class schedule.
   3. W3Schools has tons of examples on how to do this. Go read up.
5. **(5 points) nav menu with highlights**
   1. (for additional points, if you do a nav menu)
   2. Clicking an item on the nav menu should make that item highlighted.
6. **(10 points) sticky nav with auto-highlight**
   1. As you scroll through your document, your sticky nav menu should highlight the section that is topmost in your viewport.
   2. W3Schools has code to do this. Look it up.
7. **(5 points) sticky nav menu**
   1. (for additional points, if you do a nav menu)
   2. Nav menu should always be visible on the page.
   3. You can incorporate this behavior into your design however you like. Here are some ideas:
      1. Just use position: fixed, and set it at the top of the page.
      2. Put the nav menu below the hero image, then use “position: sticky; top: 0px;” to make it stick to the top of the page
8. **(5 points) sticky footer**
   1. Add a footer at the bottom of your page. The footer should say something like “Copyright © 2020 – Your Name”
   2. Use an HTML entity for the ©
   3. Use an HTML entity for the em-dash
9. **(5 points) center your document**
   1. Embed your document in a centered <div> that floats over a fixed background, like so: <http://forgefire.blogspot.com/>. (this is easier than you think).
10. **(5 points) Set a repeating background image**
    1. The text must be readable. The image must not distract from the rest of the page or call undue attention to itself.
    2. **(2 points) on all hyperlinks,** can you make it so that the underline only appears when you mouse-over the link?
    3. **(2 points) find a way to indent your poem**, or center it on the page.
11. **(5 points) Style your weekly Schedule –** Style your weekly schedule using two or more ideas from below
    1. **Text alignment:** have all the text vertically aligned to the top and horizontally aligned to the left (including the headers).
    2. **Color-coded blocks:** each item on your schedule is color-coded in its own color (English is yellow, calculus is light red, work is light blue, etc)?
    3. **Equal column widths.** On your schedule, can you find a way to have the column widths fixed so that they’re all the same size
12. **(5 points) striped table rows**
    1. in your class schedule, do alternating striped colors for each row, or for each column (not both). You should have a third color for the table header.
    2. Here’s an article on how to do that: <https://www.w3.org/Style/Examples/007/evenodd.en.html>.
13. **(10 points) add a browser tab icon (“favicon”)**
    1. A favicon is a little icon that appears on the left, on the tab for your page.



* 1. There are a number of ways to do this for a static webpage. All of them involve creating a manifest.
  2. Google the instructions on how to do this.

1. **(15 points) bring up your portrait in a modal**
   1. If you click on your portrait, have a larger version appear in a modal <div> (like what Facebook or Twitter does). You’ll need JavaScript to do this, but if you search the web you can find lots of examples on how to do this.
   2. There should be a close-box in the upper left.
   3. Clicking the close box or clicking anywhere outside the image should make the modal <div> disappear.
   4. While the modal <div> is visible, the page in the background should appear darkened or blurred.
2. **(30 points) use an online template.** 
   1. Yes, you can copy someone else’s code, as long as you spend the effort to personalize it, and you do it well. This is a little harder than it sounds.
   2. I will waive all the requirements listed in section 2.2 Content
   3. There are developer templates all over the web—probably thousands, and they’re all free. Most of them are on GitHub. All you have to do is download one and then customize it for your own use.
   4. Here are some lists of templates I found:
      1. <https://colorlib.com/wp/html5-resume-templates/>
      2. <https://colorlib.com/wp/bootstrap-portfolio-website-templates/>
      3. <https://github.com/topics/portfolio-template>
   5. You MUST
      1. Give credit to the original author
      2. Give a link to the source code for the template that you used.

# Sample web-page

Here is an example of a minimum effort, which will get you 70 points. If you want a better score, you’ll have to make it look nicer (which shouldn’t be hard):



Tom Abbott



Personal Info

* **Major:** Computer Science
* **High School:** Kadena High, Okinawa, Japan. 1987
* **Current year at UVU:** Sophomore
* **Favorite ice cream:** Bear Claw

Five favorite fantasy novels

1. *Lord of the Rings*, J. R. R. Tolkien
2. *Harry Potter*, J. K. Rowling
3. *Chronicles of Narnia*, C. S. Lewis
4. *A Wizard of Earthsea*, Ursula K. LeGuin
5. *Son of the Black Sword*, Larry Corriea

A Haiku I Wrote

Ode to my co-worker, whose desk is too close to mine:

*The slurping man slurps*

*herba mate all day long*

*I crank my headphones*

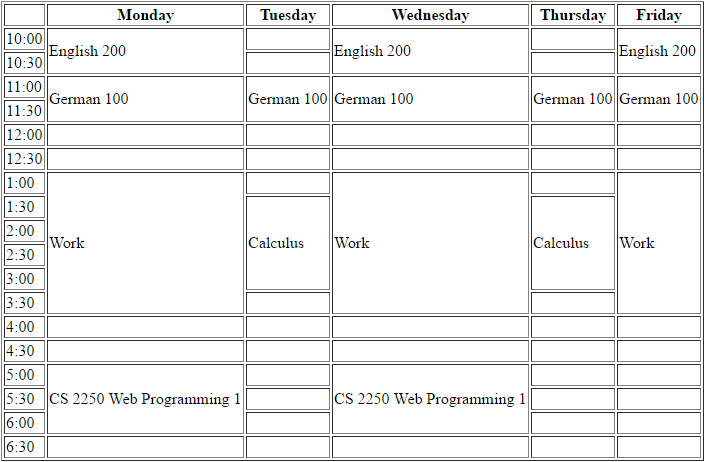
Personal interests

I like writing. I started about 11 years ago, after I moved to Utah. I’ve won a few minor awards, but right now I don’t have a lot of time. I’ve completed one novel and three short stories. Two of my short stories took Honorable Mention with the L. Ron Hubbard Writers of the Future contest.

I also like historical european martial arts (HEMA). When I have time I study with a group of HEMA enthusiasts in Provo. We study old manuscripts from the 1300s and 1400s and learn fighting techniques with two-handed longswords.

I also teach college as an adjunct professor for both BYU and UVU. This was an unexpected opportunity, which fell into my lap. It takes a lot of time and it is tough, but I get to work with a lot of people and make friends, and share the things that I’ve learned in my career.

Class schedule



# Submitting your work

Push your work up to GitHub. If you’ve created a github.io repo, you should be able to go to the following URL and view your webpage:

<https://your-username.github.io>

If you want to host your webpage on Netlify, or Heroku, or your friend’s Minecraft server, or a Raspberry PI running under your desk, that’s fine.

You are responsible for making sure your webpage is viewable on the internet before submitting your assignment.

When you’re ready to submit, go to Canvas and enter the following:

* Your GitHub username
* URL to the repository for your code on GitHub.
* URL to your website.
* A bulleted list of items you did for Styling and Appearance (section 2.4).

## What you may not do

You may not use any kind of ready-made website, or website generator. The purpose of this assignment is to learn how to code using HTML, CSS, and some light JavaScript.

This includes the following:

* Wordpress
* Wix
* Blogspot.com / blogger.com
* Medium.com
* Any kind of social media page

## Grade breakdown

This project will be graded as follows

* 10 points: your GitHub account with a repository where this code lives.
* 10 points: your webpage hosted on a server. This can be github.io or anywhere.
* 50 points: your website
* 30 points: styling your website.