# Tutorial: Editing code files for your portfolio!

### **Note: You will start at the “Opening the Repo” section.**

Contents

Important Links &

[Important Links & Background 2](#_Toc142506655)

[Navigating the Repository 3](#_Toc142506658)

[Opening the Repo 4](#_Toc142506659)

[Making Changes 4](#_Toc142506660)

[Step 1: Pulling 4](#_Toc142506661)

[Step 2: Open up files in VS Code 4](#_Toc142506662)

[Step 3: View files in Live Server 4](#_Toc142506663)

[Step 4: Pushing changes to the repo 5](#_Toc142506664)

[HTML Tags 7](#_Toc142506666)

## Important Links

Portfolio temporarily hosted at: <https://natsukisacks.github.io/diana-dabby-portfolio/>

All portfolio files located in this repository (repo): [https://github.com/natsukisacks/diana-dabby-portgitfolio](https://github.com/natsukisacks/diana-dabby-portfolio)

## Background

Git allows multiple people to clone a repo and edit it simultaneously if their edits do not overlap. To edit the code files, we can clone this repo to your local computer and “push” any changes from there.

Although GitHub has a code editor embedded in the website, I like to use VS Code. It separates the code with different colors, making it easier to understand. It also has an intuitive interface.

## Cloning the Repository (already done)

First, press the Windows key and search for “Git Bash”. Git Bash is a shell that allows for easy interaction with Git.

Then, go to <https://github.com/natsukisacks/diana-dabby-portfolio> and copy the HTTPS link as seen here

A screenshot of a computer

Description automatically generated

Go to Git Bash and type git clone and right click to paste in the link that you just copied. Press Enter. Now that you have a cloned repository, you should have the most up-to-date version of the website.

# Navigating the Repository

A screenshot of a computer

Description automatically generatedView the full repository folders by clicking on the stacked pages icon in the top right corner of your VS Code. You can open and collapse the File Explorer through this button.

A screenshot of a computer

Description automatically generatedThe repository is broken down as follows.

The main pages, identified with the HTML symbols <> are not in a subfolder. These pages are the basis of the navigation bar at the top of the website.

assets: contains all non-HTML code that makes the website look pretty and intuitive to interact with.

subpages: contains subpages for each section of the navigation bar

website-documents: contains all documents used in the website. This folder has more folders in it, separating supplementary material for each subpage.

website-images: contains all images used in the website. This folder has more folders in it, separating the images for each subpage.

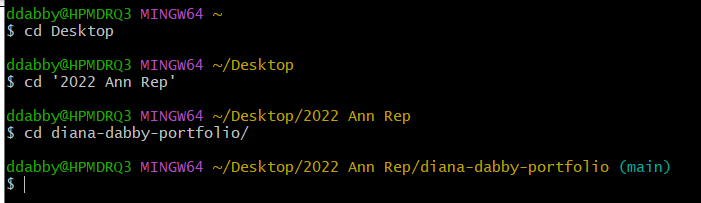
# **START HERE**

# Open Git Bash

Hit the Windows Key and search “Git Bash”. Open that up.

# Opening the Repo

Your repo is cloned to a subfolder inside of your Desktop, at the file path Desktop/2022 Ann Rep/diana-dabby-portfolio.



Enter into the directory using the commands as shown above in white.

Not applicable: You can right-click the Git Bash screen to paste any text copied to your clipboard.

# Making Changes

## Step 1: Pulling

Before you “push” your changes, you should type git pull in Git Bash to avoid potential merge conflicts. Pulling first will ensure that your repo is the most up-to-date, allowing you to accept any overlapping changes before sharing them. If it’s successful, it should look something like this:

A black screen with yellow and blue text

Description automatically generated

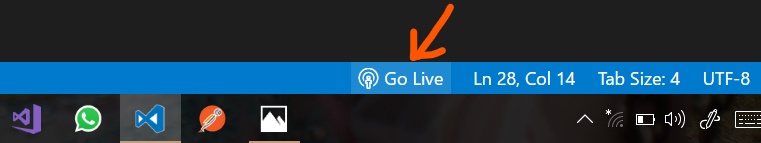
## Step 2: Open up files in VS Code

Type code . into Git Bash to open up your repository in VS Code.

## Step 3: View files in Live Server

Start reading and making any changes you feel necessary. You can turn on the Live Server mode to see your changes in action. To see your changes, you must save your file by pressing Ctrl + S.

To see the live version of your code file, you should open the code file that you’d like to see in website form. Then, go to the bottom right-hand corner and click on “Go Live”.



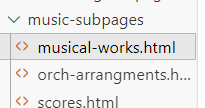
To make it easier to view the live version and code file at the same time, you should split your screen; grab your VS Code window at the very top and drag it over to the side of your screen (left or right side, doesn’t matter). It should show all of your other windows open, from which you can open the Chrome one hosting the live version of your code file.

A screenshot of a computer

Description automatically generated

Like this for example. I would select the bottom right hand corner window to split screen.

**IMPORTANT EXAMPLE:** To make changes, i.e. to Music: go to the music-subpages. For the section to be changed, go to \*-subpages. To find the text that you’d like to fix quickly, CTRL+F to search for a keyword (i.e. Olin Conductorless Orchestra).

Make the change, then ctrl s to save.

 For any given page that you‘re editing, the name of it should match the last part of the file path as shown in the search bar. Here, we are looking to edit “musical-works.html” in VS Code.

## Step 4: Pushing changes to the repo

Once you’ve done making changes for a session, you can “push” them to the main repo! Make sure to CTRL+S to save your file before pushing.

To do this, first type git add . to add all of the modified files to what you’ll push. Press Enter.

Then, type git status to check if all of your changes have been added to the staging. This is an example of the modified file not having been staged for commit. A screen shot of a computer

Description automatically generated

Then, type git commit -m ‘type commit message here’ and press Enter. Commit messages should be less than 50 characters and concisely describe any changes that you made. Commit messages make it easier to troubleshoot your code by allowing you to back track your steps. It also informs other users in the repo about the changes you’ve made.

A computer screen shot of a program

Description automatically generated

Git Once your commits have been staged, you want to finally push them to the main branch. To do this, type git push and press Enter. If successful, it should look something like this:

A computer screen shot of a program

Description automatically generated

**NOTE:** If you get any error messages that you don’t understand, you can look the message up in Google and it will give you suggestions of what to do.

If you’ve successfully pushed, the main repo should reflect that it’s been updated! You can check that [here:](https://github.com/natsukisacks/diana-dabby-portfolio) <https://github.com/natsukisacks/diana-dabby-portfolio> A screenshot of a computer

Description automatically generated

## **Remember that in order to actually ensure that your changes are relayed to the actual website, you must push your changes to the main repo!**

# HTML Tags

HTML Cheatsheet: <https://www.semrush.com/blog/html-tags-list/>

HTML utilizes tags encased in <tag>words here</tag> to signify what to display in the file. Here are some important tags:

Paragraphs: <p>example</p>

Italic: <em>example</em>

Bold: <b>example</b>

Header: <h1>example</h1>

<h2>example</h2>

<h2>example</h2>

Using CSS, I’ve styled the headers 1-3 differently, with h1 being the largest font size and h3 the smallest.

Best way to communicate any changes besides text would be w email. or you can make a word doc with questions/changes and email to suki.