Getting started with source control

Download GIT

- Git is free and open source software for distributed version control: tracking changes in any set of files, usually used for coordinating work among programmers collaboratively developing source code during software development.

Starting your own repo

- Create a local git repository using

```
git init
```

- Create a long description for your project

```
git add README.md
```

- Add your repository to be committed with

```
git add .
```

- Create your own github repository to link to the local repository
- *git commit* creates a commit, which is like a snapshot of your repository. These commits are snapshots of your entire repository at specific times.

```
git commit -m "first commit"
```

- The git branch command **lets you create, list, rename, and delete branches**

```
git branch -M main
```

- To add a new remote, **use the git remote add command on the terminal, in the directory your repository is stored at**. The git remote add command takes two arguments: A unique remote name, for example, “my_awesome_new_remote_repo” A remote URL, which you can find on the Source sub-tab of your Git repo.

```
git remote add origin https://github.com
/AustinHood7/PortfolioWebsiteV2.git
```

- The git push origin master command says "**push the commits in the local branch named master to the remote named origin**". Once this is executed, all the stuff that we last synchronized with origin will be sent to the remote repository and other people will be able to see them there. Main can be any of the declared remote origin branches.

```
git push -u origin main

git push -u origin austinBranch
```

- Daily you can pull from any remote origin using git pull. This fetches data from the specified branch and merges it with the existing local repository

```
git pull https://github.com/AustinHood7
/CoinAnalyticsWebApp.git austinBranch
```

Working off another repository

- Clone the project with `git clone https://github.com/AustinHood7/CoinAnalyticsWebApp.git`
  - This gets all branches and recreates the entire repository within your local system
- **Optional:** Make a branch for the created ticket on GitHub using `git checkout -b tix5`
  - This creates a new branch in your local system to work on the specific ticket you checked out
- Stage the changes that were made in this branch using `git add .`
- Commit and save those changes to the local repository using `git commit -m "Message"`
- Make sure everything you have is up to date with main my making a pull using `git pull origin main`
- To push these changes into GitHub use `git push -u origin tix5`
  - This is done to make sure that you don't have any merge conflicts when you try to merge with main
- Create a new pull request to merge the newly created branch with main using the GitHub UI
- After we can delete the local branch using `git branch -D tix-5`
  - We do this to save space on the local drive and still have the branch in the distributed repository