



tasks

GitHub for Individuals

[GitHub](#) is one of the most popular and important development tools you will use. It keeps a history of your development activity and can be used as a tool to show potential employers your code and coding habits.



GitHub can also be used to help multiple developers work on the same code set without overwriting each other's additions. However, this task will focus on using the tool as an individual. In this task you will use GitHub to:

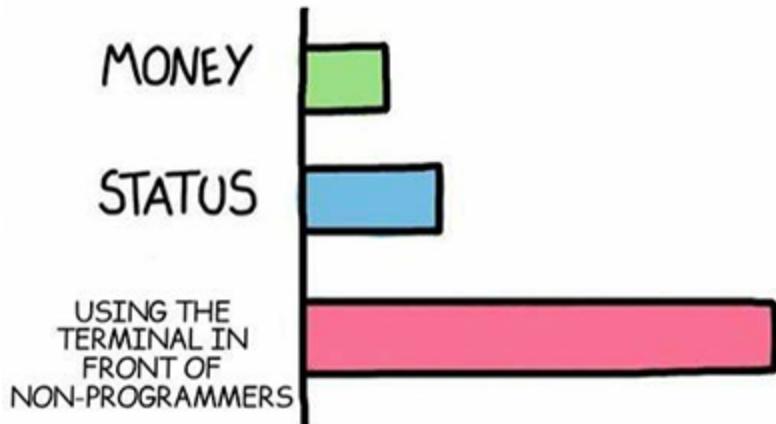
- Store and share your code
- Create backups and version control
- Practice using the Git and GitHub CLI

Remember that [Git](#) and [GitHub](#) are two different things. Just like Gmail is one of many providers of email. GitHub is one of many providers for storing Git repositories.

Git and GitHub can be hard at first. And if you haave never used the Command Line, it can be a bit intimidating. However, it might be the most important and widely used development tool.



WHAT GIVES PEOPLE FEELINGS OF POWER



Git CLI

In this class we will use Git through the command line. Start by following the [Getting Started - Git Installation](#). Once you're done run the following command in your Terminal (Mac/Linux) or Git Bash (Windows) to confirm it has been installed:

```
git --version
```

A screenshot of a Mac OS X terminal window titled "tasks -- zsh -- 80x24". The window shows the command "git --version" being run and its output: "git version 2.24.3 (Apple Git-128)".

```
git --version
git version 2.24.3 (Apple Git-128)
```



We will also use the [GitHub Command Line Interface \(CLI\) Tool](#). Please download and install the proper version. Once you're done, run the following command in your Terminal (Mac/Linux) or Git Bash (Windows) to confirm it has been installed:

```
gh --version
```

Once you have the GitHub CLI installed, use it to login to your GitHub account:

1. Run the following command:

```
gh auth login
```

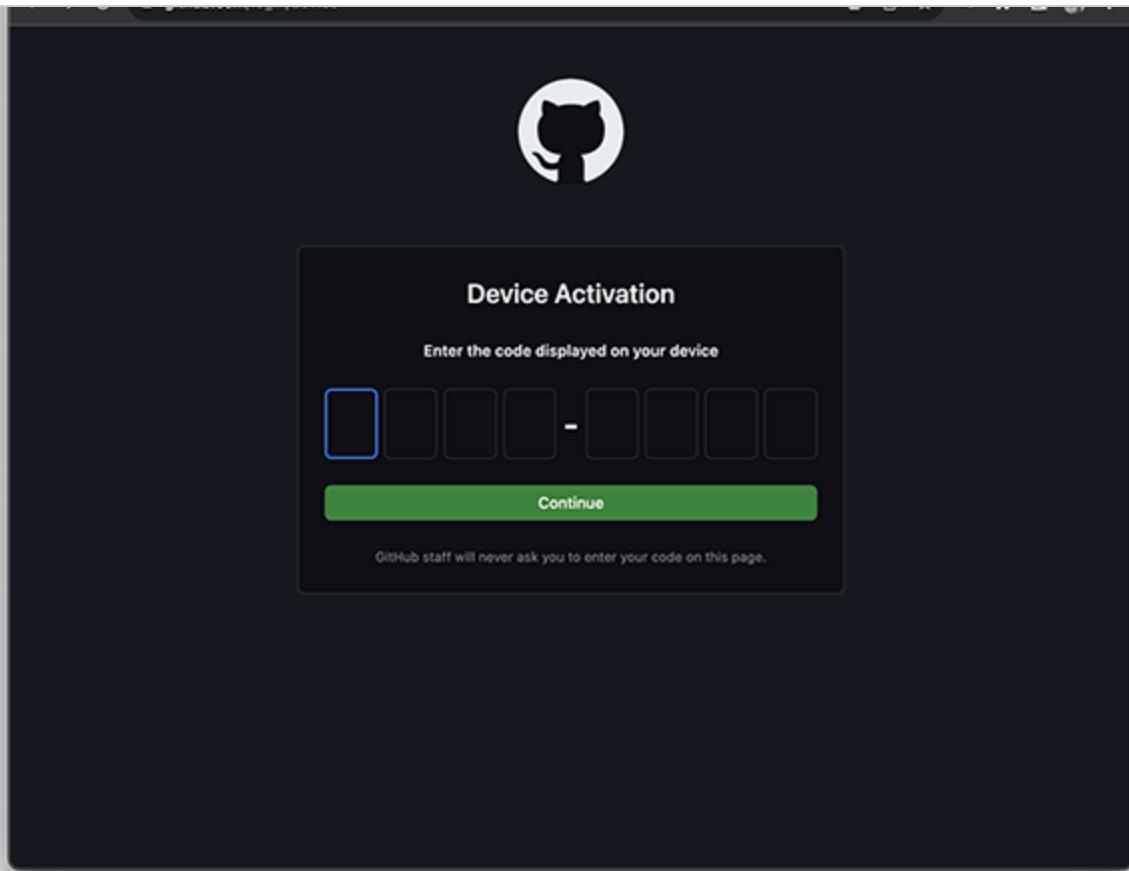
2. Select `GitHub.com`.
3. Select `HTTPS`.
4. Answer `Y` to authenticate Git with your GitHub credentials.
5. Select `Login with a web browser`.
6. Press `ENTER`.
7. Enter the eight digit code from the Terminal into the browser and then click `Authorize github`.

A screenshot of a terminal window titled "tasks — gh auth login — 80x24". The window shows the following text:

```
[Adam@NJ-00116-loaner074 tasks ~]$ gh auth login
? You're already logged into github.com. Do you want to re-authenticate? Yes
? What is your preferred protocol for Git operations? HTTPS
? Authenticate Git with your GitHub credentials? Yes
? How would you like to authenticate GitHub CLI? Login with a web browser

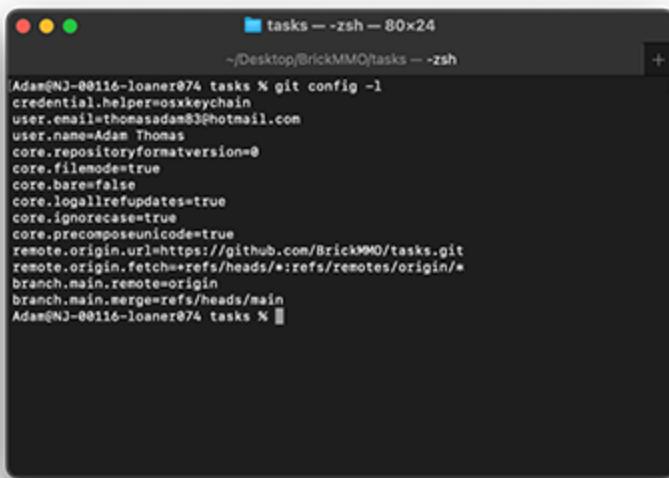
! First copy your one-time code: 103C-4491
Press Enter to open github.com in your browser...
✓ Authentication complete.
- gh config set -h github.com git_protocol https
✓ Configured git protocol
✓ Logged in as c0deadma
[Adam@NJ-00116-loaner074 tasks ~]$ gh auth logout
✓ Logged out of github.com account 'c0deadma'
[Adam@NJ-00116-loaner074 tasks ~]$ gh auth login
? What account do you want to log into? GitHub.com
? What is your preferred protocol for Git operations? HTTPS
? Authenticate Git with your GitHub credentials? Yes
? How would you like to authenticate GitHub CLI? Login with a web browser

! First copy your one-time code: 454C-2DC7
Press Enter to open github.com in your browser...
```



8. Confirm you are logged in by checking the local configuration settings:

```
git config -l
```



A screenshot of a terminal window titled "tasks -- zsh -- 80x24" running on a Mac OS X system. The command "git config -l" has been run, and the output shows the following configuration settings:

```
Adam@NJ-00116-loaner074 tasks % git config -l
credential.helper=osxkeychain
user.email=thomasadam3@hotmail.com
user.name=Adam Thomas
core.repositoryformatversion=0
core.filemode=true
core.bare=false
core.logallrefupdates=true
core.ignorecase=true
core.precomposeunicode=true
remote.origin.url=https://github.com/BrickMMO/tasks.git
remote.origin.fetch=+refs/heads/*:refs/remotes/origin/*
branch.main.remote=origin
branch.main.merge=refs/heads/main
Adam@NJ-00116-loaner074 tasks %
```



When using Git and GitHub as an individual, you will usually follow these steps:

1. Create a new repo for your project.
2. Clone the repo to your computer:

```
git clone https://github.com/BrickMMO/tasks.git
```

You can find the URL of a GitHub repo here:

A screenshot of a GitHub repository page for "BrickMMO / tasks". The page shows a list of files and a "Clone" button. The "Clone" button has three options: "HTTPS", "SSH", and "GitHub CLI". The "HTTPS" option is selected, and the URL "https://github.com/BrickMMO/tasks.git" is displayed. Other repository details shown include 1 branch, 0 tags, 0 stars, 1 watching, 0 forks, and a description "A repo to store BrickMMO tasks".

3. Make changes using your code editor.

4. Add current changes to the main copy:

```
git add .
```



```
git commit -am "Describe latest changes"
```

6. Push the changes to your GitHub repo:

```
git push origin main
```

When you are working as an individual, you will not need to create any additional branches. You can simply add and commit all your changes directly to the `main` branch like in the examples above.

You can get detailed instructions of this process on my [github-demo-branching](#) repo.

Recommendations

Use Git and GitHub as much as possible! It will force you to practice the skill, you will have a backup of all your work, and your GitHub contributions chart will start turning green!

I would recommend the following:

1. Create a GitHub repo for each class. Put all your class files in that repo. Push the files from class to your GitHub account at the end of each class (or even more often).
2. Create a GitHub repo for each assignment. You can use these repos to hand assignments in. And you can also use a GitHub repo to host a project (we cover that later in the [Deployment Task](#)).
3. If you're learning a new skill, create a repo and store the code there in case you need it for later.

Consistency

Be consistent! As a developer you should be obsessed with consistency!

- Use consistent naming conventions.
- Give your files nice clean names like `logo-github.png`. Not `IMG-20170407-WA0007_resized.jpg`, `IMG-20170407-WA0007.jpg`, or `Copy of IMG-20170407-WA0007.jpg`.
- Organize your files into classes, and weeks, and topics:

```
http5111
|   README.md
|
```



```
└ week2
    └ images.html
    └ tables.html
    └ images
        └ logo.jpg
```

I have written a set of rules for myself (and all BrickMMO development) to follow when working with GitHub repos and README.md files. These rules are called [_readme](#). You can use these as a starting point.

Steps

To complete this task, follow these steps:

1. Create a repo for each of your classes.
2. Clone each of those repos to your computer.
3. Put your class files into those repos.
4. Add, commit, and push those files to your GitHub account.

Submitting this Task

Submit your class GitHub repo URLs to the **Individuals** assignment in [Blackboard](#).

[→ Back to Task List](#)

