

Module 10: Intro to Git and Markdown

October 29, 2025

Forking a Repo

Forking a repo allows for you to create your own copy of a repository. You can modify the main branch, you can add as many feature branches, and you can manage your own git repo on Github in your own namespace.

Let's all take a moment to fork the lecture notes to our own Github namespace

Adding a new remote

Now that we have forked our repo on Github, we now have a copy of the repository stored in our own namespace there. To have access to it, we need to create a new remote to pull and push data to it.

Some general conventions with repos, the main repo, the one we forked from is usually referred to as `upstream` and it is where we would pull new features and commits from. Let's rename our remote from `origin` to `upstream`

```
git remote rename origin upstream
```

Now let's add our new remote. Repositories by default are named

Contributing to a git repository

`git` allows for you to share content to a repository. Often, what you want to do is contribute your changes or `patches` to a project and have them reviewed and eventually merged into the project.

Let's create our first patch together.

```
git checkout -b yearbook-teon-brooks
```

Contributing to the Yearbook

Let's all add an entry to this class yearbook. We will all submit a markdown file with our name `firstname_lastname.md` and it will contain a markdown message in its body.

As a reminder, this is the full path for our local copy of the git repo:

```
cd ~/workspace/mskcc-python/mskcc-python-lecture-notes
```

First, let's install a new shortcut for our Terminal

Using the Command Palette (Command+Shift+P on macOS/Ctrl+Shift+P on Windows)

```
>Install 'code' command in PATH
```

We can use the command `touch` to create an empty file.

We can use the command `code` to launch our VS Code session from the terminal.

```
cd yearbook/2025  
touch teon_brooks.md  
code teon_brooks.md
```

Creating a commit (patch)

In terminal, we can stage the file we just created to git. This lets git know that a file was created and added to the project.

```
git add teon_brooks.md
```

To create a commit, we use the `commit` command and we add a message to go along with it:

```
git commit -m "Adding Teon Brooks to the yearbook"
```

Pushing our Commit

To share our commit with the project, we need to push our branch with its new commit.

We will use the branch that we created earlier, for me, that was `yearbook-teon-brooks`

```
git push origin yearbook-teon-brooks
```

Github

Let's see where all these branches were just pushed to:

<https://github.com/teonbrooks/mskcc-python-lecture-notes>

Pulling down updates

For the lecture today, I will create a commit and we will all sync our repos to accept the updated commits.

```
git pull upstream main
```

Markdown

<https://docs.github.com/en/get-started/writing-on-github/getting-started-with-writing-and-formatting-on-github/basic-writing-and-formatting-syntax>