

# Intro Class Activity - Week 3

## Goals:

In this class activity we are going to create a new repository on github.com. We will then add the repository to our local computer into our Projects folder. Afterwards we will add a new file to that folder and update the repository online. Following that we will replicate this procedure 6 more times.

### Tools Needed

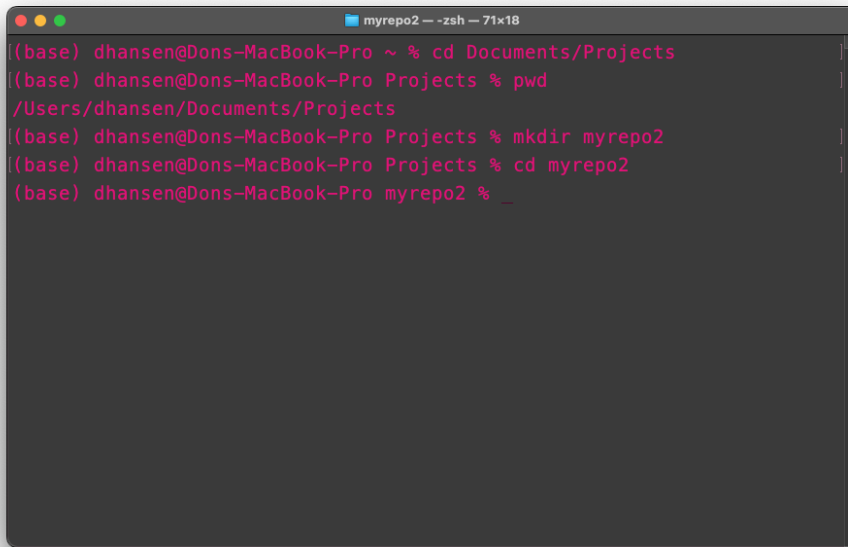
- Computer
- Console / BaSh / Terminal / PowerShell
- Text Editor
- Web Browser

*Instructions continue on the next page >>*

# Terminal / File Explorer

## Setting up our project folder

1. Create a new project folder called **“myrepo2”** using the command [ **mkdir myrepo2** ]
2. Change directory and go into the new **“myrepo2”** folder using the command [ **cd myrepo2** ]
3. ***Keep this window open!*** You will need it again.



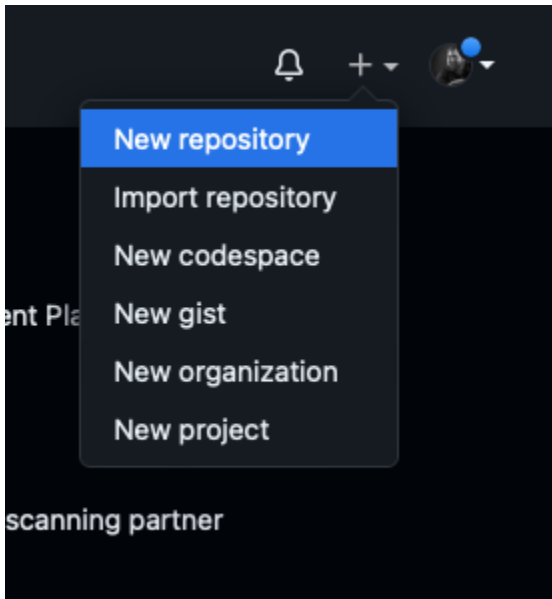
```
myrepo2 — zsh — 71x18
(base) dhansen@Dons-MacBook-Pro ~ % cd Documents/Projects
(base) dhansen@Dons-MacBook-Pro Projects % pwd
/Users/dhansen/Documents/Projects
(base) dhansen@Dons-MacBook-Pro Projects % mkdir myrepo2
(base) dhansen@Dons-MacBook-Pro Projects % cd myrepo2
(base) dhansen@Dons-MacBook-Pro myrepo2 % _
```

*Instructions continue on the next page >>*

# Github

## Creating a new repository on GitHub.com

4. Click the [ + ] icon in the top right and click on “**New repository**”



5. Name the repository “**myrepo2**” and leave the other options as the defaults.

### Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

---

**Repository template**

No template ▾

Start your repository with a template repository's contents.

---

**Owner \***

charger71 ▾

**Repository name \***

myrepo2

✔ myrepo2 is available.

Great repository names are short and memorable. Need inspiration? How about [fictional-octo-potato?](#)

**Description** (optional)

---

☒ **Public**  
Anyone on the internet can see this repository. You choose who can commit.

☐ **Private**  
You choose who can see and commit to this repository.

---

**Initialize this repository with:**

☐ **Add a README file**  
This is where you can write a long description for your project. [Learn more about READMEs.](#)

**Add .gitignore**

.gitignore template: None ▾

Choose which files not to track from a list of templates. [Learn more about ignoring files.](#)

6. Press the “**Create Repository**” button.



7. Copy the block of commands from the [GitHub.com](https://github.com) website and paste them into your console (BaSh, Terminal, PowerShell) and then press [ **ENTER** ] or [ **RETURN** ] on your keyboard.

Unset

```
echo "# myrepo2" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/YOUR-USERNAME/myrepo2.git
git push -u origin main
```


#### Quick setup — if you've done this kind of thing before

 Set up in Desktop or **HTTPS** **SSH** <https://github.com/charger71/myrepo2.git> 

Get started by [creating a new file](#) or [uploading an existing file](#). We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

#### ...or create a new repository on the command line

```
echo "# myrepo2" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/charger71/myrepo2.git
git push -u origin main
```



#### ...or push an existing repository from the command line

```
git remote add origin https://github.com/charger71/myrepo2.git
git branch -M main
git push -u origin main
```



#### ...or import code from another repository

You can initialize this repository with code from a Subversion, Mercurial, or TFS project.

[Import code](#)

8. In the console type [ **git status** ]. You should see the output like the following. You are looking for the line near the end to say “**nothing to commit, working tree clean**”.

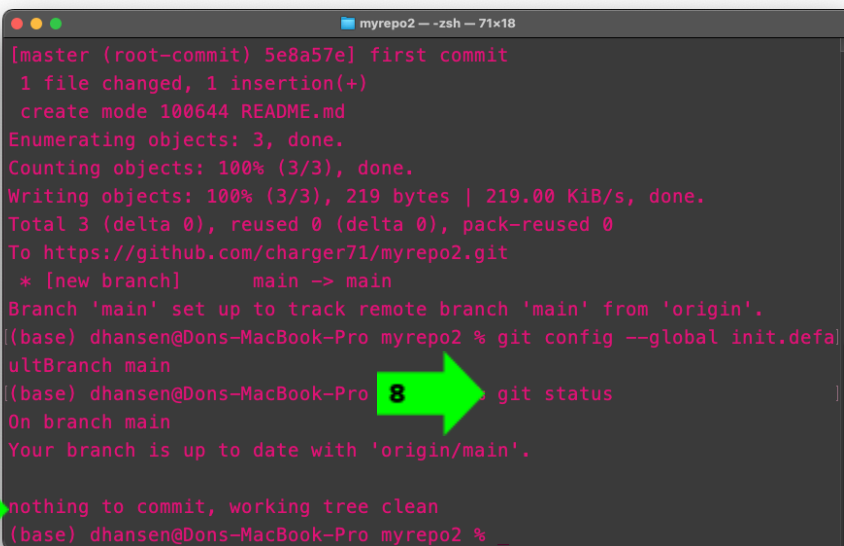
Unset

On branch main

Your branch is up to date with 'origin/main'.

**nothing to commit, working tree clean**

**If you see an error please try again, look for spelling errors or syntax order mistakes. If the error persists after troubleshooting please contact a mentor.**



```
myrepo2 - zsh - 71x18
[master (root-commit) 5e8a57e] first commit
 1 file changed, 1 insertion(+)
 create mode 100644 README.md
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 219 bytes | 219.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/charger71/myrepo2.git
 * [new branch]      main -> main
Branch 'main' set up to track remote branch 'main' from 'origin'.
(base) dhansen@Dons-MacBook-Pro myrepo2 % git config --global init.defaultBranch main
(base) dhansen@Dons-MacBook-Pro myrepo2 % git status
On branch main
Your branch is up to date with 'origin/main'.

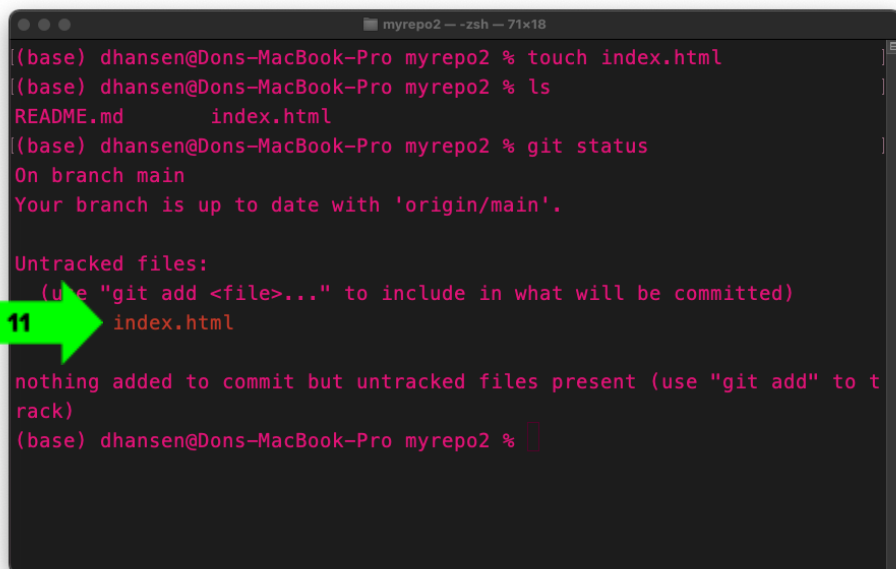
nothing to commit, working tree clean
(base) dhansen@Dons-MacBook-Pro myrepo2 %
```

## All Together Now

**Make a change to your file and push it up to the repository on GitHub.com**

9. In the console type [ **touch index.html** ] to create a new file with that name.
10. Type [ **ls** ] to make sure the file was created in the correct folder “**myrepo2**”.

11. Enter [ **git status** ] on the command line. You should see “**index.html**” highlighted in a red color. This means that git does not know about the file. We need to “**ADD**” the file to git’s database.



```
myrepo2 - zsh - 71x18
(base) dhansen@Dons-MacBook-Pro myrepo2 % touch index.html
(base) dhansen@Dons-MacBook-Pro myrepo2 % ls
README.md      index.html
(base) dhansen@Dons-MacBook-Pro myrepo2 % git status
On branch main
Your branch is up to date with 'origin/main'.

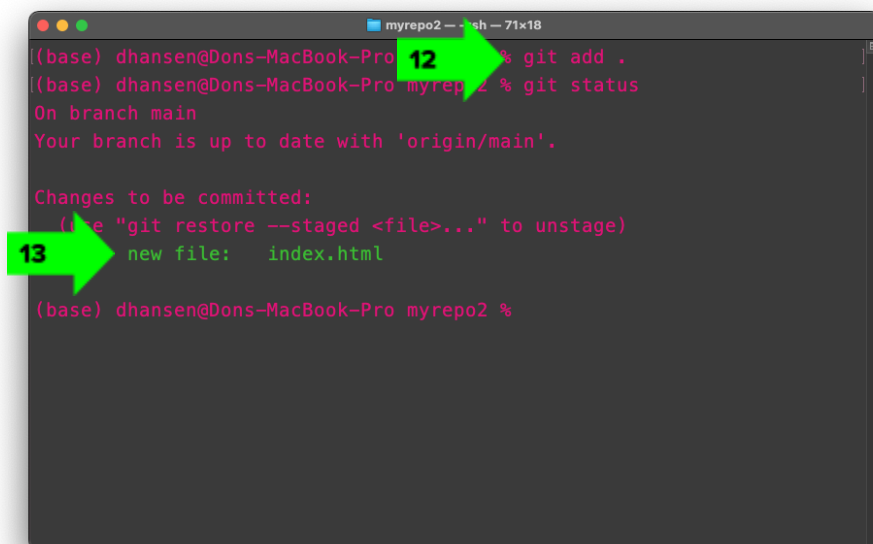
Untracked files:
  (use "git add <file>..." to include in what will be committed)
  index.html

nothing added to commit but untracked files present (use "git add" to track)
(base) dhansen@Dons-MacBook-Pro myrepo2 %
```

A green arrow labeled '11' points to the `index.html` file listed under 'Untracked files'.

12. Use the [ **git add .** ] command to add the files to git’s database or “staging”, and press [ **enter** ].

13. Run [ **git status** ] again. We should see the **index.html** file now in green. Meaning that it has been added to the database and is in “**staging**”.



```
myrepo2 - zsh - 71x18
(base) dhansen@Dons-MacBook-Pro % git add .
(base) dhansen@Dons-MacBook-Pro myrepo2 % git status
On branch main
Your branch is up to date with 'origin/main'.

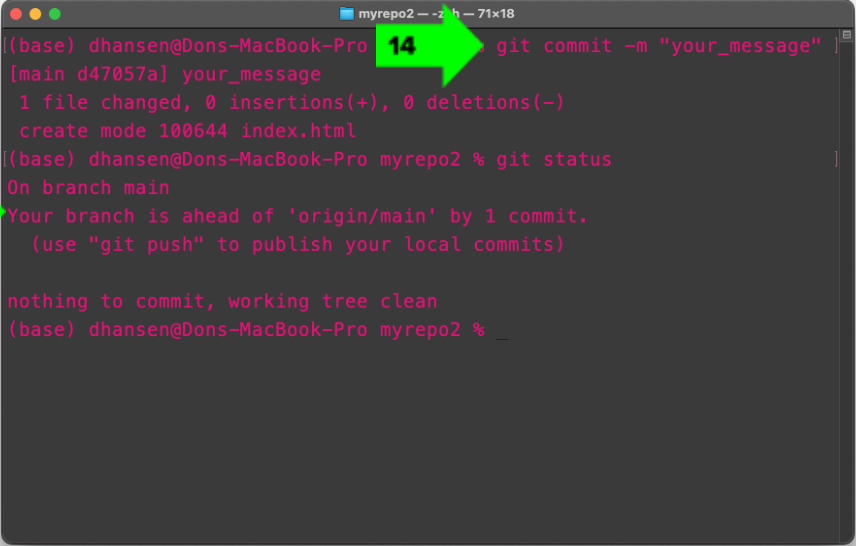
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
  new file:   index.html

(base) dhansen@Dons-MacBook-Pro myrepo2 %
```

A green arrow labeled '12' points to the `git add .` command, and another green arrow labeled '13' points to the `new file: index.html` line in the output.

14. Next run [ **git commit -m "your message"** ]. This command tells git’s database why this file or files were added to the source code.

15. Run [ **git status** ] You should see a message stating that [ **Your branch is ahead of 'origin / main by 1 commit'** ]. We now need to “**PUSH**” the files up to the [GitHub.com](https://github.com) server.



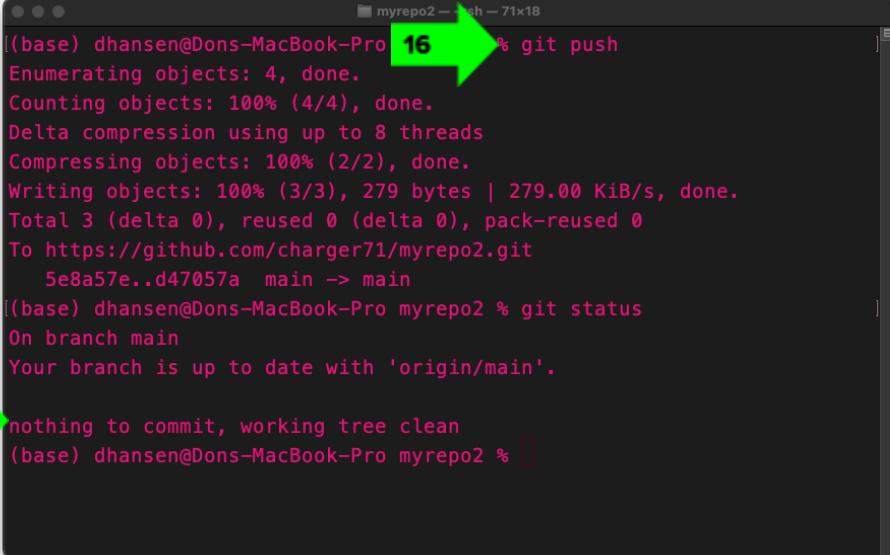
A terminal window titled 'myrepo2' showing the execution of two Git commands. A green arrow labeled '14' points to the 'git commit' command. The output shows a commit on the 'main' branch. Another green arrow labeled '15' points to the 'git status' command. The output indicates the branch is ahead of the origin by one commit.

```
myrepo2 --zsh -- 71x18
(base) dhansen@Dons-MacBook-Pro % git commit -m "your_message"
[main d47057a] your_message
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 index.html
(base) dhansen@Dons-MacBook-Pro myrepo2 % git status
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
(use "git push" to publish your local commits)

nothing to commit, working tree clean
(base) dhansen@Dons-MacBook-Pro myrepo2 % _
```

16. Use [ **git push** ] to send the files to our remote repository on [GitHub.com](https://github.com).

17. In the console type [ **git status** ]. You should see the output like the following. You are looking for the line near the end to say “**nothing to commit, working tree clean**”.



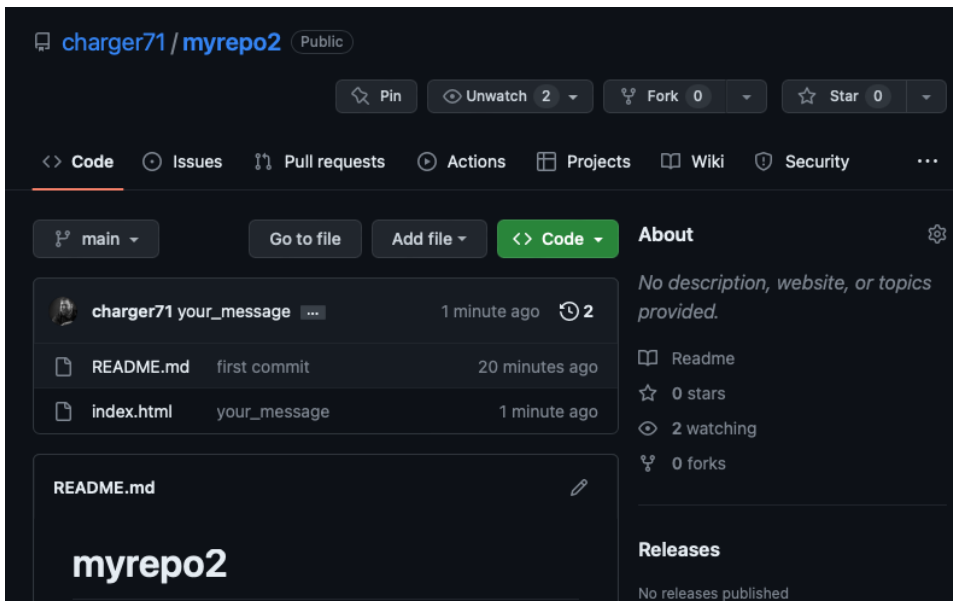
A terminal window titled 'myrepo2' showing the execution of two Git commands. A green arrow labeled '16' points to the 'git push' command. The output shows the files being pushed to the remote repository. Another green arrow labeled '17' points to the 'git status' command. The output indicates the branch is up to date with the origin.

```
myrepo2 --zsh -- 71x18
(base) dhansen@Dons-MacBook-Pro % git push
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 279 bytes | 279.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/charger71/myrepo2.git
5e8a57e..d47057a main -> main
(base) dhansen@Dons-MacBook-Pro myrepo2 % git status
On branch main
Your branch is up to date with 'origin/main'.

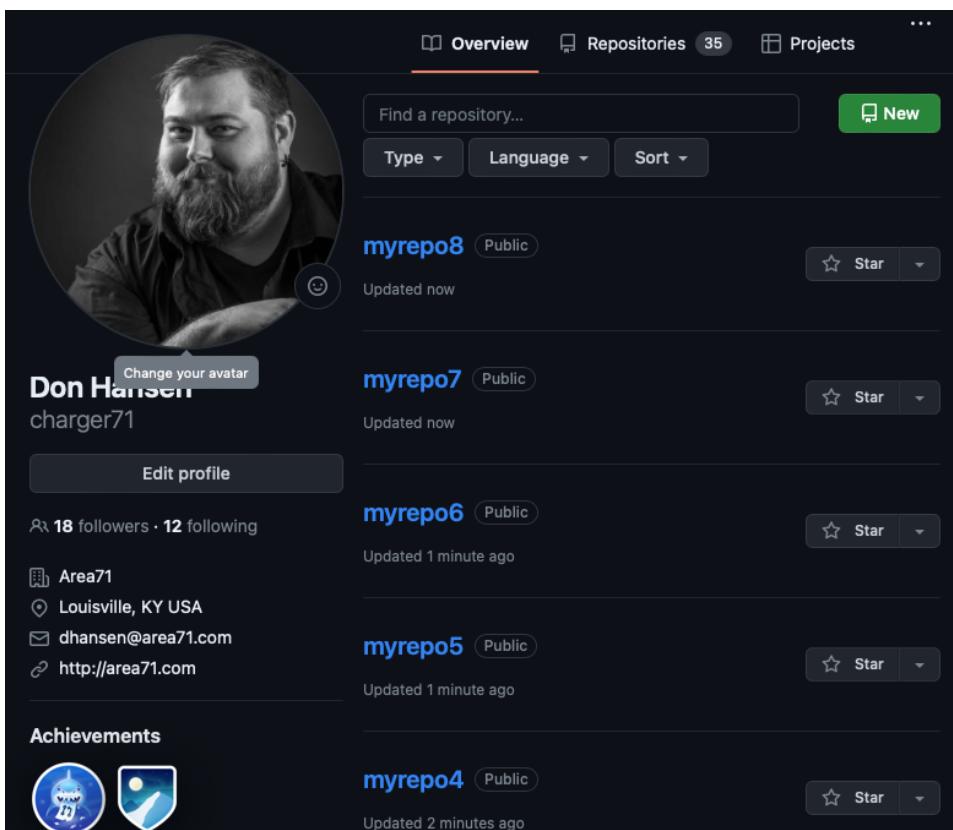
nothing to commit, working tree clean
(base) dhansen@Dons-MacBook-Pro myrepo2 %
```

*Instructions continue on the next page >>*

18. Go back to the browser window with the [GitHub.com](https://github.com) repo creation window in it. **REFRESH** the web page, and check out your new repo with files.



19. Repeat the process steps for 6 more repos. You should have “myrepo2, myrepo3, myrepo4, myrepo5, myrepo6, myrepo7, myrepo8” all with **index.html** files inside them and loaded into individual repositories in your [GitHub.com](https://github.com) account. Your repository overview page should look similar to the following screenshot.





20. Your Projects folder should look similar to the following screenshot.

