Create and Connect to a Remote Git Repository

Exercise #1

- Initialize Local Git Repo in day-05 folder
- Add and Commit the changes

Steps to follow:

- Initialize a git repo in day 5 folder: Run git init in the terminal

```
marthavillamartin@Marthas-Air day-05 % git init
Initialized empty Git repository in /Users/marthavillamartin/Desktop/
FWDP-1000/FWDP-1000-materials/day-05/day-05/.git/
```

- Stage all the files: Run the git add . command

```
marthavillamartin@Marthas-Air day-05 % git add .
```

- Commit the content of the day 5 folder: Execute the git commit -m "your commit message" command to save your changes with a descriptive message.

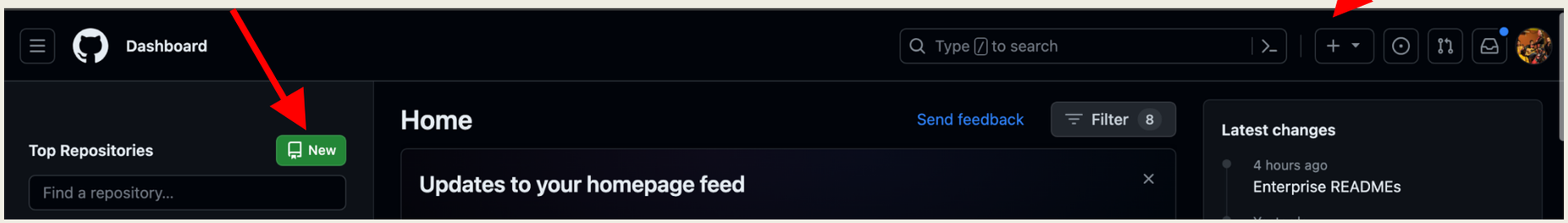
```
marthavillamartin@Marthas-Air day-05 % git commit -m "Initial commit
  of day-05 folder with lecture materials"
```

Exercise #2

- Create remote git repo on GitHub for your day 5 folder
  - Connect the remote repo to the local repo

Step 1– Create a Remote Repo

- **Sign in to GitHub**: Go to GitHub & log in to your account
- **Create New Repository:** Click the "+" icon in the upper-right corner and select "New repository" or click the green “New” button.



- **Repository Details:** Enter your repository name, description, and choose the visibility (public or private).
- **Initialize Repository:** You may choose to initialize the repository with a README, .gitignore, and license if needed.
- **Create Repository:** Click the "Create repository" button.

PS: For your landing page & country website, if your repo is public, you don’t need to add me. If you create a private repo, you will need to add me as a collaborator.

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? Import a repository.

Required fields are marked with an asterisk (*).

Owner *

piracode /

Repository name *

fwd1000-day05

fwd1000-day05 is available.

Great repository names are short and memorable. Need inspiration? How about improved-guacamole ?

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.

Choose a license

License: **None**

A license tells others what they can and can't do with your code. Learn more about licenses.

You are creating a public repository in your personal account.

Create repository

fwd1000-day05 Public

Pin | Unwatch 1 | Fork 0 | Star 0

Set up GitHub Copilot

Use GitHub's AI pair programmer to autocomplete suggestions as you code.

Get started with GitHub Copilot

Add collaborators to this repository

Search for people using their GitHub username or email address.

Invite collaborators

Make sure to select HTTPS here

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

Set up in Desktop or HTTPS SSH `https://github.com/piracode/fwd1000-day05.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 "# fwd1000-day05" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/piracode/fwd1000-day05.git
git push -u origin main
```

Click here to copy

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

```
git remote add origin https://github.com/piracode/fwd1000-day05.git
git branch -M main
git push -u origin main
```

Copy and paste to your terminal

...or import code from another repository

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

Import code

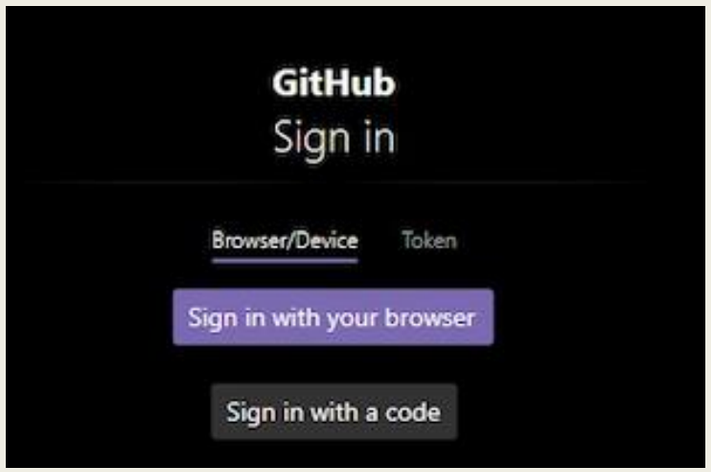
ProTip! Use the URL for this page when adding GitHub as a remote.

Step 2 – Copy and Paste GitHub commands into VSCode Terminal

- Copy & paste the code for existing repositories you got after creating a new repo:

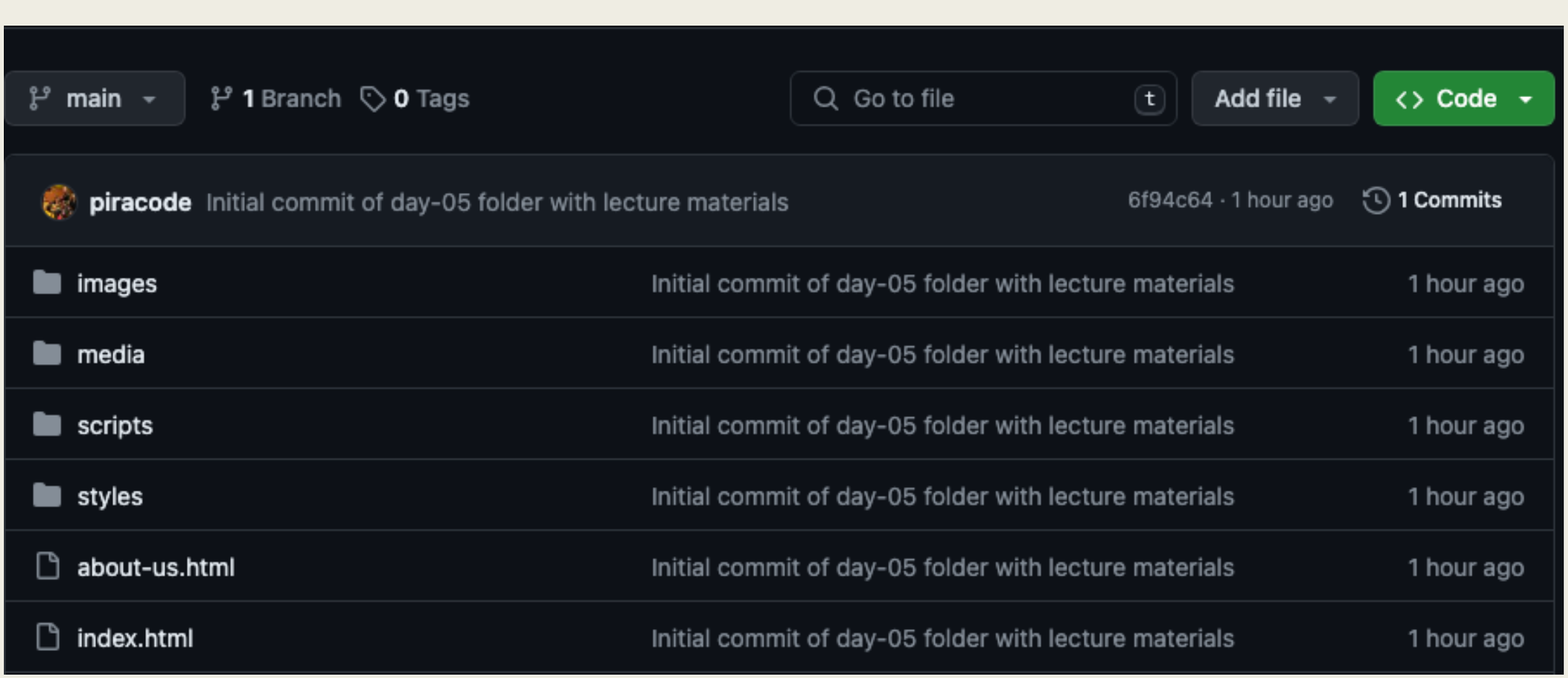
```
marthavillamartin@Marthas-Air day-05 % git remote add origin https://github.com/piracode/fwd1000-day05.git
git branch -M main
git push -u origin main
```

Note: If it is the first time you push to GitHub, you will need to authorize the connect between Git and GitHub. Make sure to follow the prompts and then authorize Git Credential Manager.



- After authorizing, ensure that Git pushes the changes to GitHub. It might take some time if there are many files (check the progress in the terminal)
- You'll only need to authorize the connection between Git and GitHub once.

- Now, check your GitHub repository, and you'll see the files that were pushed to GitHub:



What do each line mean?

- git remote add origin https://github.com/piracode/fwd1000-day05.git

This line connects the local repository to the remote repository named 'origin'.

- git branch -M main

This line renames the current branch to 'main' using the '-M' flag. We will talk about branches tomorrow.

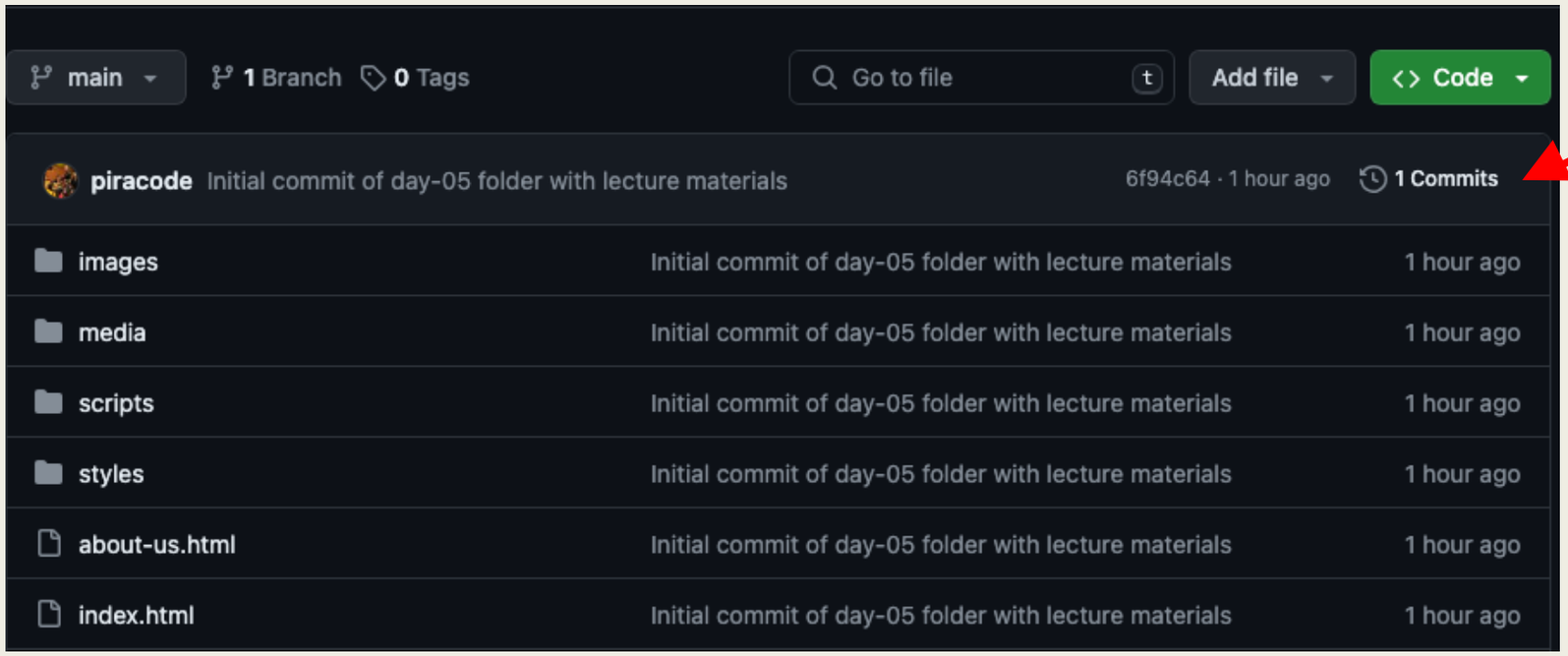
- git push -u origin main

“git push -u origin main” is like saving your project in the cloud. The “-u”(upstream flag) makes it easier next time, telling Git to remember this place. This way later you can just say “git push” to update your project in the cloud. (instead of saying “git push origin main” every time you need to push to the cloud)

“origin” is your cloud's address, and “main” is the project's main folder.

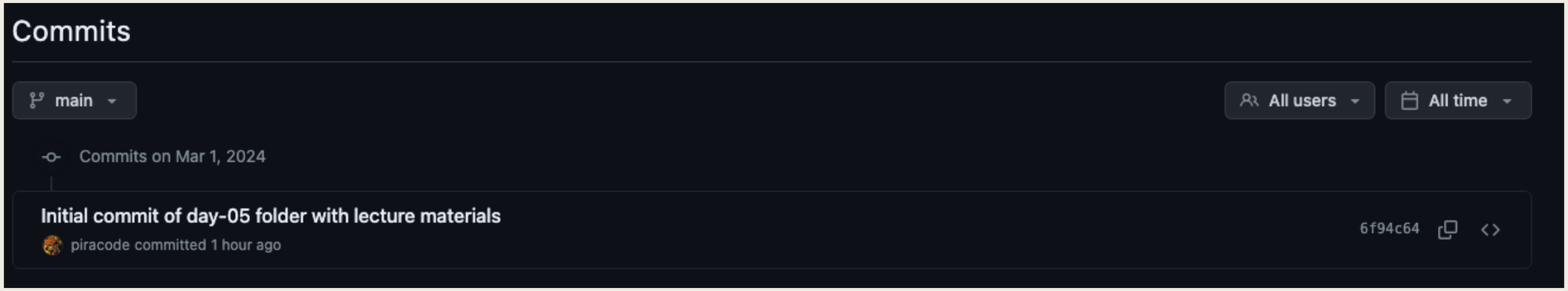
Optional: Check Your Commits on GitHub

- You can view detailed commit information, including additions, removals, and changes.



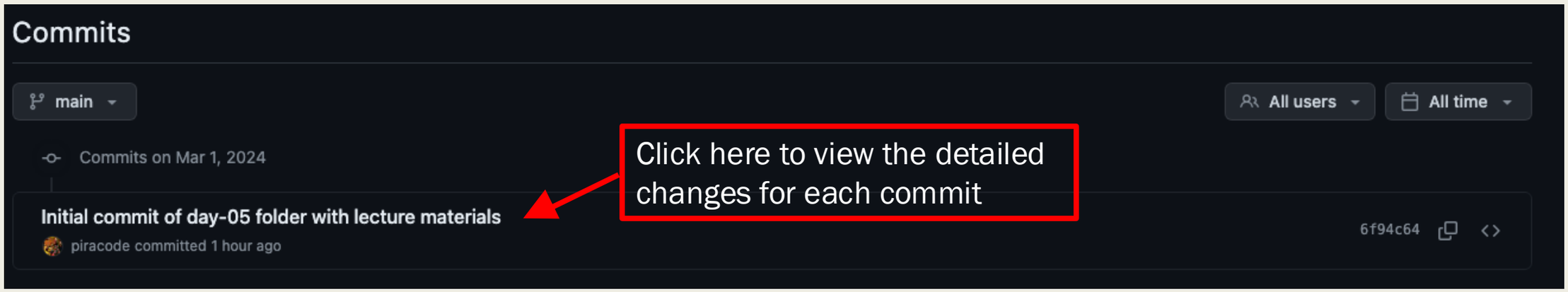
Click here to see the list of commits

- We've made only one commit so far. The list will expand with more commits pushed to the remote repository.

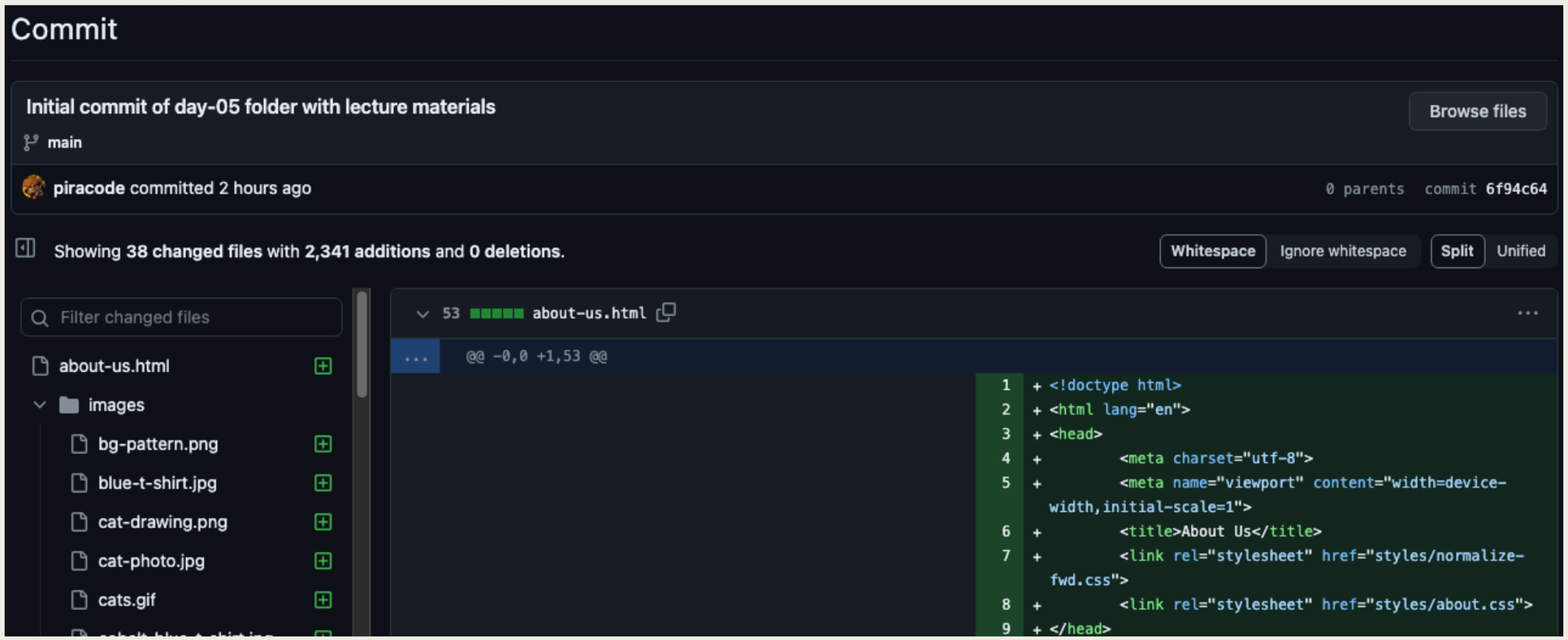


Optional: View Commits Details on GitHub

- After following the previous slide and accessing the list of commits you've made, click on the commit message. This action will direct you to a new page where you can view the specific changes made, including additions, deletions, and more.



- Here, for instance, we can see the additions we made.



Recap

- Initialize Local Repository: Start with git init.
- Create GitHub Repo: Create a new repository on GitHub.
- Link Local to GitHub: Use git remote add origin [repo URL].
- Rename & Push: Change the branch name if necessary with git branch -M main, then push using git push -u origin main.

These commands can be copied directly from GitHub after creating a repo

- Workflow after GitHub repo creation: Add changes (git add .), then commit (git commit -m ”your message"). Then Push the changes using git push