

CS411 @ BU

LAB 002

GIT

THIS IS GIT. IT TRACKS COLLABORATIVE WORK ON PROJECTS THROUGH A BEAUTIFUL DISTRIBUTED GRAPH THEORY TREE MODEL.

COOL. HOW DO WE USE IT?

NO IDEA. JUST MEMORIZIZE THESE SHELL COMMANDS AND TYPE THEM TO SYNC UP. IF YOU GET ERRORS, SAVE YOUR WORK ELSEWHERE, DELETE THE PROJECT, AND DOWNLOAD A FRESH COPY.



me being really stupid



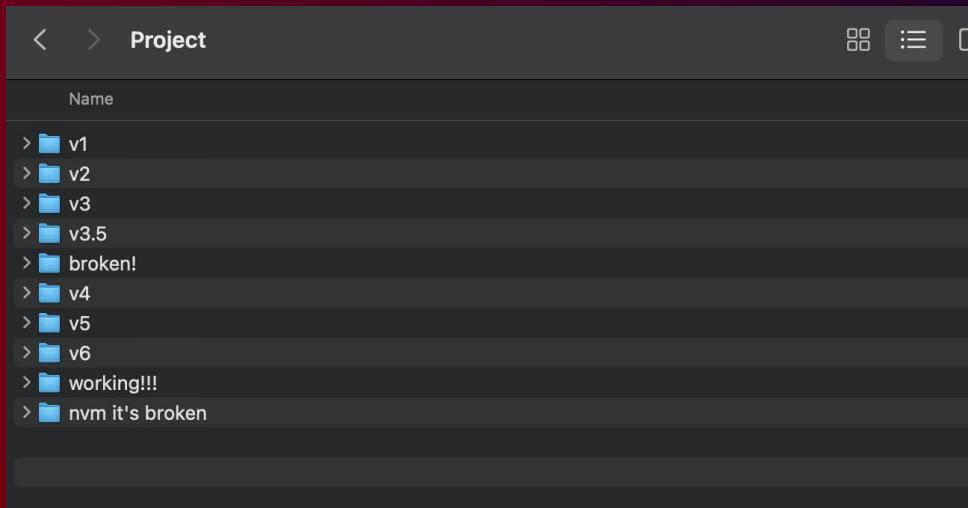
HOW DO YOU STORE FILES?



GIT
VS
RANDOM FOLDERS ON YOUR PC

NON GIT SOLUTIONS

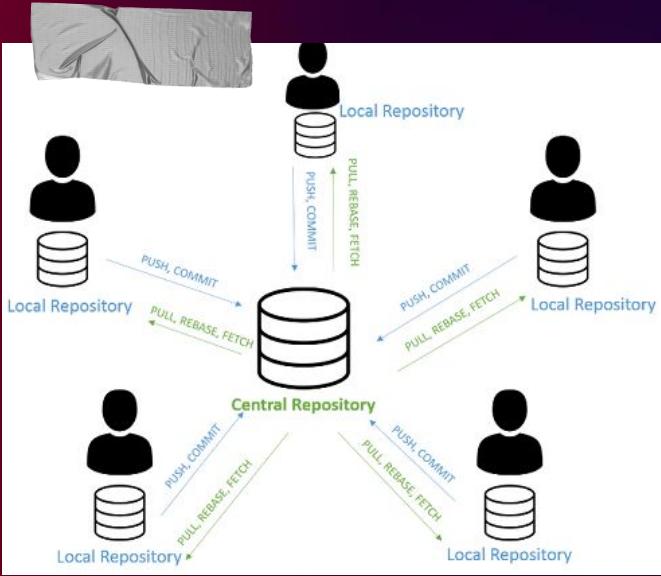
COLLABORATION, VERSION CONTROL, & 'HIGH SPEED' DATA TRANSFER



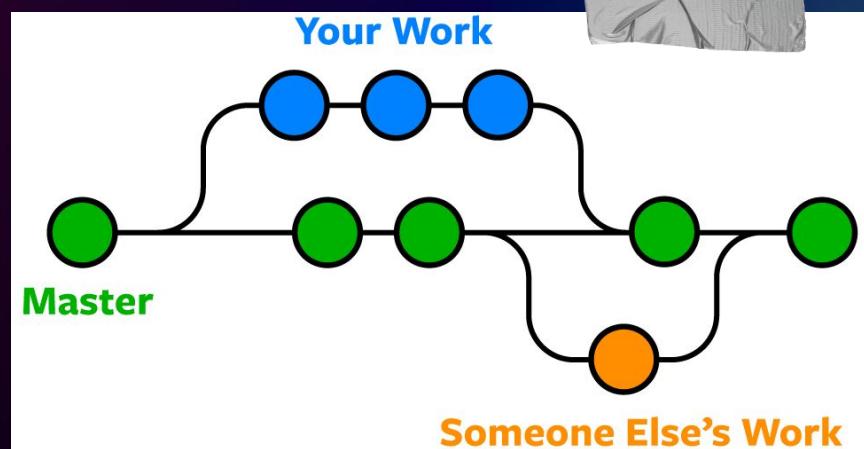
A screenshot of a file explorer window titled "Project". The left sidebar shows a list of items under "Name":

- > v1
- > v2
- > v3
- > v3.5
- > broken!
- > v4
- > v5
- > v6
- > working!!!
- > nvm it's broken





WHY GIT



GIT

VERSION CONTROL

REBASE

BRANCHING

WORKFLOWS

MERGE

DATABASES

WEBSITE HOSTING

COLLABORATION

CODE REVIEW



GIT SETUP

Do this once and forget about it



Account

<https://github.com/>
Use Your BU Email



Pro

Students get a free
GitHub Pro Account



Desktop

<https://desktop.github.com/>



CLI

<https://cli.github.com/>



DESKTOP

The easiest way to go

SETUP

The screenshot shows the GitHub Desktop landing page. At the top center is the GitHub logo (a white octocat icon inside a purple circle). Below it are three navigation links: "Overview", "Release Notes", and "Help". To the right of the logo is a stylized globe icon with a heart and a speech bubble.

GitHub Desktop

Focus on what matters instead of fighting with Git. Whether you're new to Git or a seasoned user, GitHub Desktop simplifies your development workflow.

[Download for macOS](#)

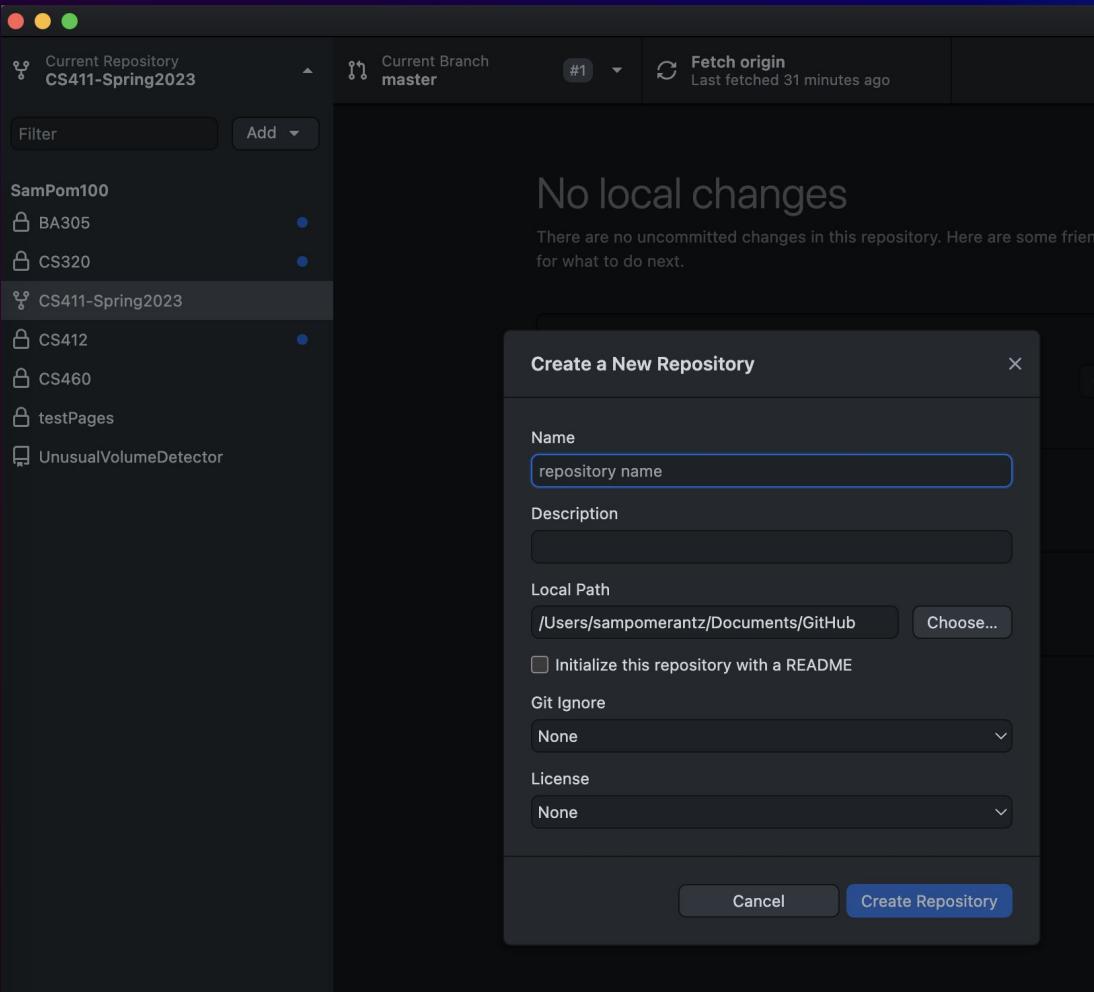
Feeling brave?
Try new features in the [Beta Channel](#) before they're released.

Apple silicon?
Download for an [Apple silicon Mac](#).
See the [Apple docs](#) about Apple vs Intel chips.

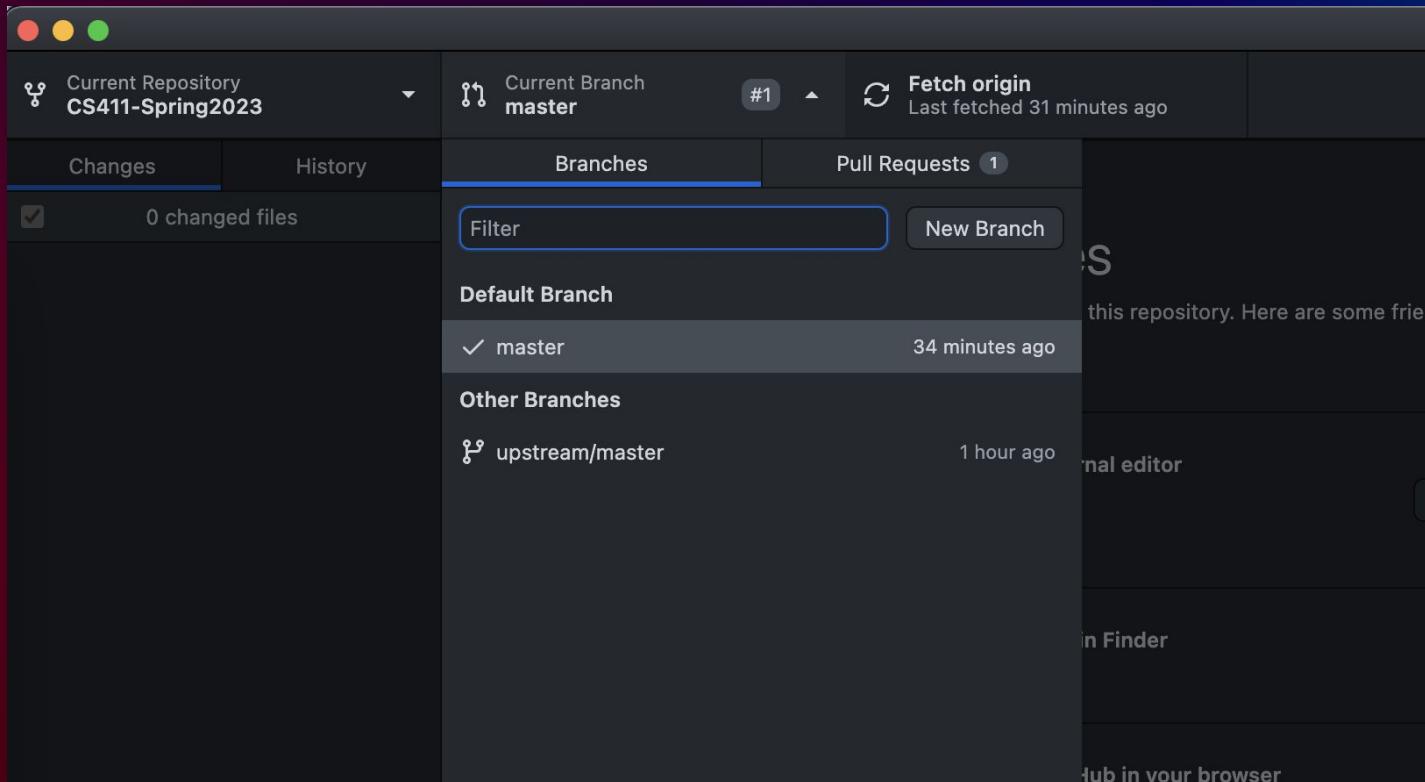
Windows?
[Download for Windows](#)

By downloading, you agree to the [Open Source Applications Terms](#).

SETUP



SETUP





CLI

A more powerful alternative



Homebrew

The Missing Package Manager for macOS (or Linux)

Search [%] [K]
English ▾

Install Homebrew

```
$ /bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"
```

Paste that in a macOS Terminal or Linux shell prompt.

The script explains what it will do and then pauses before it does it. Read about other [installation options](#).

What Does Homebrew Do?

Homebrew installs **the stuff you need** that Apple (or your Linux system) didn't.

```
$ brew install wget
```

SETUP

- 2 Paste the text below, substituting in your GitHub email address.

```
$ ssh-keygen -t ed25519 -C "your_email@example.com"
```

SETUP

```
[git@crc-dot1x-nat-10-239-226-204 ~ % ssh-keygen -t ed25519 -C "sampomer@bu.edu"
Generating public/private ed25519 key pair.
Enter file in which to save the key (/Users/git/.ssh/id_ed25519):
Created directory '/Users/git/.ssh'.
[Enter passphrase (empty for no passphrase):
[Enter same passphrase again:
Your identification has been saved in /Users/git/.ssh/id_ed25519
Your public key has been saved in /Users/git/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:uhaIAVvedLazSpTIbPMusxkl5G1No6          sampomer@bu.edu
The key's randomart image is:
+--[ED25519 256]--+
|      ....o+E+o  |
| .   ... o.oo.+  |
| B.+o+o. .       |
| .+0++=o.        |
| .==*=..oS       |
| .++ o.          |
+----[SHA256]-----+
git@crc-dot1x-nat-10-239-226-204 ~ %
```

SETUP

1

Start the ssh-agent in the background.

```
$ eval "$(ssh-agent -s)"  
> Agent pid 59566
```

SETUP

- First, check to see if your `~/.ssh/config` file exists in the default location.

```
$ open ~/.ssh/config  
> The file /Users/YOU/.ssh/config does not exist.
```

- If the file doesn't exist, create the file.

```
$ touch ~/.ssh/config
```

- Open your `~/.ssh/config` file, then modify the file to contain the following lines. If your SSH key file has a different name or path than the example code, modify the filename or path to match your current setup.

```
Host *.github.com  
AddKeysToAgent yes  
UseKeychain yes  
IdentityFile ~/.ssh/id_ed25519
```

SETUP

- ③ Add your SSH private key to the ssh-agent and store your passphrase in the keychain. If you created your key with a different name, or if you are adding an existing key that has a different name, replace *id_ed25519* in the command with the name of your private key file.

```
$ ssh-add --apple-use-keychain ~/.ssh/id_ed25519
```

SETUP

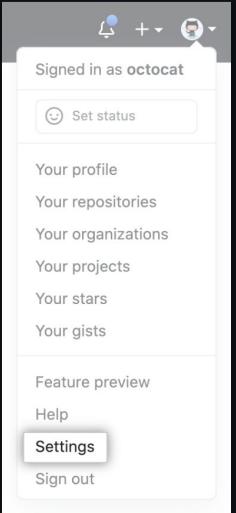
- 1 Copy the SSH public key to your clipboard.

If your SSH public key file has a different name than the example code, modify the filename to match your current setup. When copying your key, don't add any newlines or whitespace.

```
$ pbcopy < ~/.ssh/id_ed25519.pub
# Copies the contents of the id_ed25519.pub file to your clipboard
```

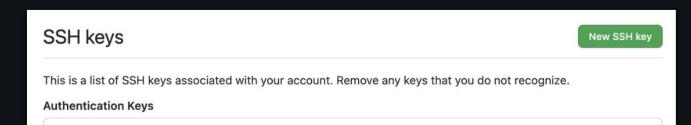
SETUP

- ② In the upper-right corner of any page, click your profile photo, then click **Settings**.



- ③ In the "Access" section of the sidebar, click **SSH and GPG keys**.

- ④ Click **New SSH key** or **Add SSH key**.



SETUP

- 5 In the "Title" field, add a descriptive label for the new key. For example, if you're using a personal laptop, you might call this key "Personal laptop".
- 6 Select the type of key, either authentication or signing. For more information about commit signing, see "[About commit signature verification](#)."
- 7 Paste your public key into the "Key" field.

The screenshot shows a user interface for managing SSH keys. At the top, there's a 'Title' input field and a 'Key type' dropdown set to 'Authentication Key'. Below that is a large text area labeled 'Key' containing placeholder text: 'Begins with 'ssh-rsa', 'ecdsa-sha2-nistp256', 'ecdsa-sha2-nistp384', 'ecdsa-sha2-nistp521', 'ssh-ed25519', 'sk-ecdsa-sha2-nistp256@openssh.com', or 'sk-ssh-ed25519@openssh.com''. At the bottom is a green 'Add SSH key' button.

- 8 Click **Add SSH key**.

Add SSH key

Troubleshooting

```
UW PICO 5.09

[core]
    repositoryformatversion = 0
    filemode = true
    bare = false
    logallrefupdates = true
    ignorecase = true
    precomposeunicode = true
[remote "origin"]
    url = git@github.com:SamPom100/GPUbot
    fetch = +refs/heads/*:refs/remotes/origin/*
[branch "main"]
    remote = origin
    merge = refs/heads/main
```

Troubleshooting

Once you have started the SSH agent with:

```
1459 eval $(ssh-agent)
```

Do either:

- 1. To add your private key to it:
ssh-add
- 2. To add and save your key permanently on **macOS**:
ssh-add -K

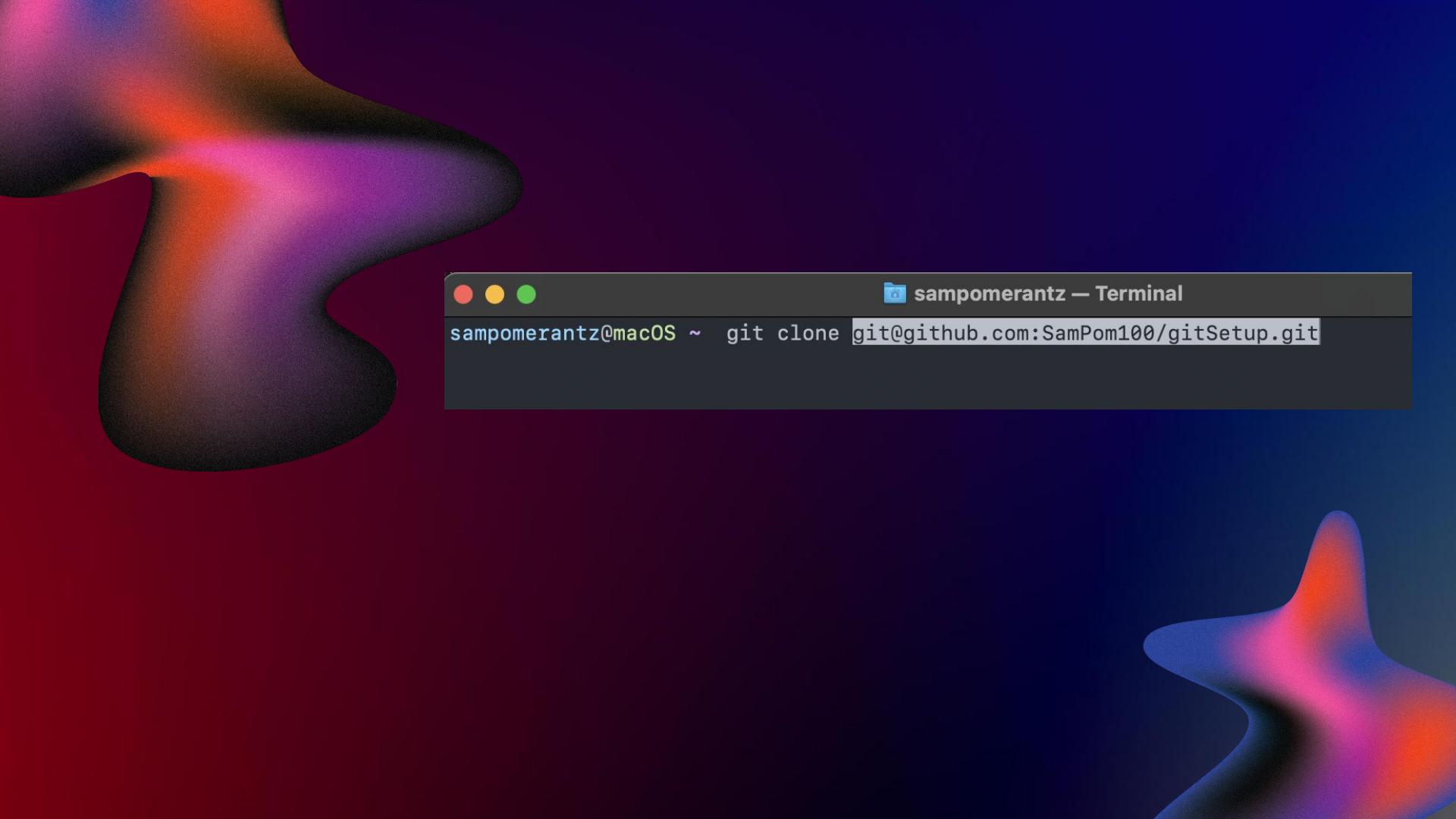
This will ask you your passphrase just once, and then you should be allowed to push, provided that you uploaded the public key to Github.

This will persist it after you close and re-open it by storing it in user's keychain.



USING GIT

Literally 3 steps



```
sampomerantz@macOS ~ git clone git@github.com:SamPom100/gitSetup.git
```

Easy as 1, 2, 3

`git add .`

`git commit -m "message"`

`git push origin`

`git commit -am "add a bunch of
important files"`

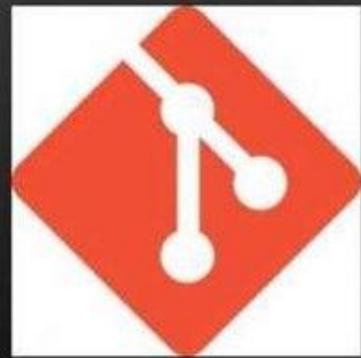


`git push -u origin HEAD`



`git add .`





PLAYLIST

Songs about GIT

Erstellt von 1124279316 • 6 Songs, 17 Min.

PLAY

...

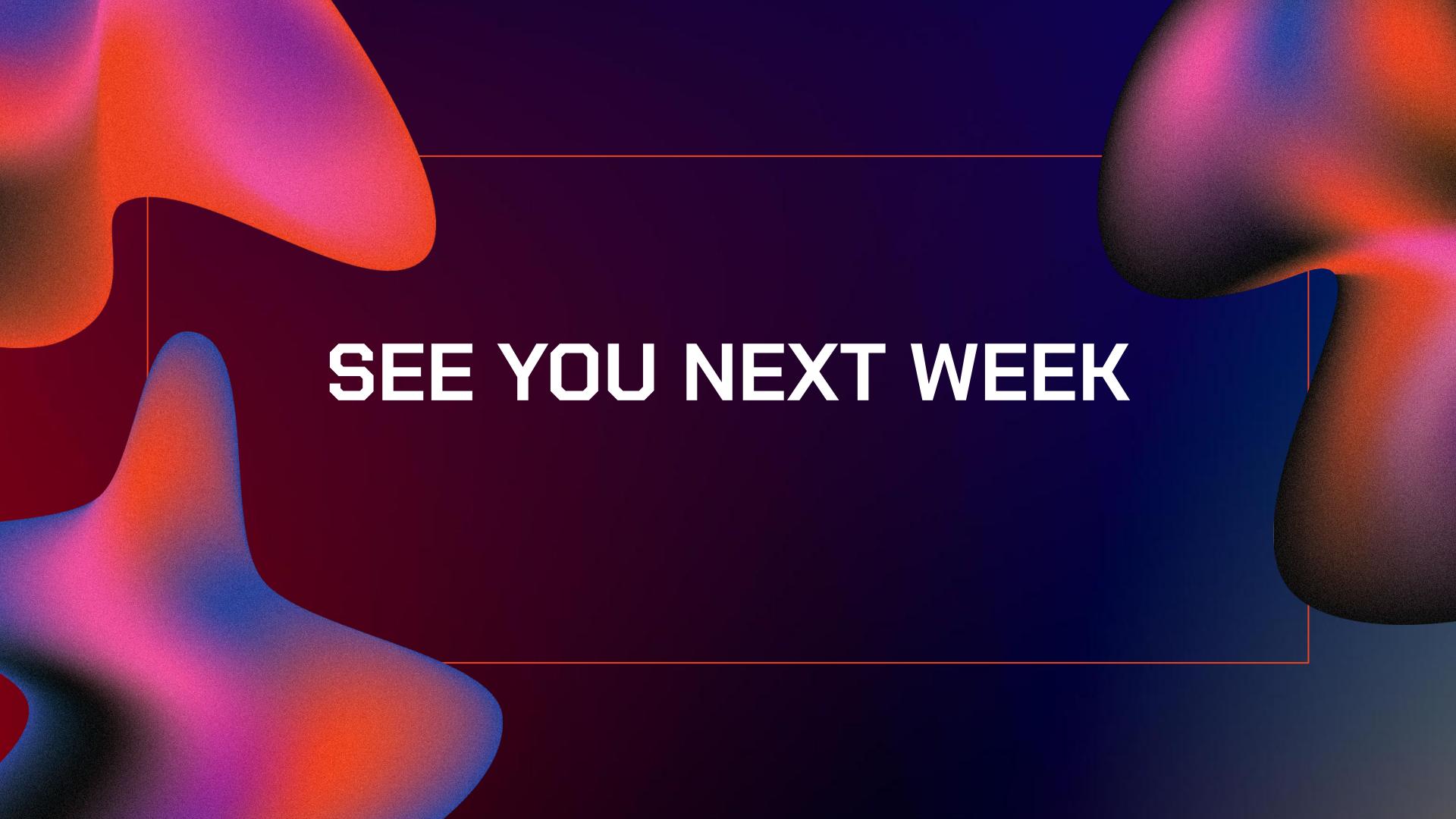
Filter:

TITEL

KÜNSTLER

- | | |
|--------------------------|----------------|
| + Pull It | Jeff Beck |
| + Push It | Salt-N-Pepa |
| + Committed | Unsane |
| + My computer is dying | GRISFX |
| + Catastrophic Failure | Syncretia |
| + Fuck This Shit I'm Out | The Theme Song |

EXPLICIT



SEE YOU NEXT WEEK