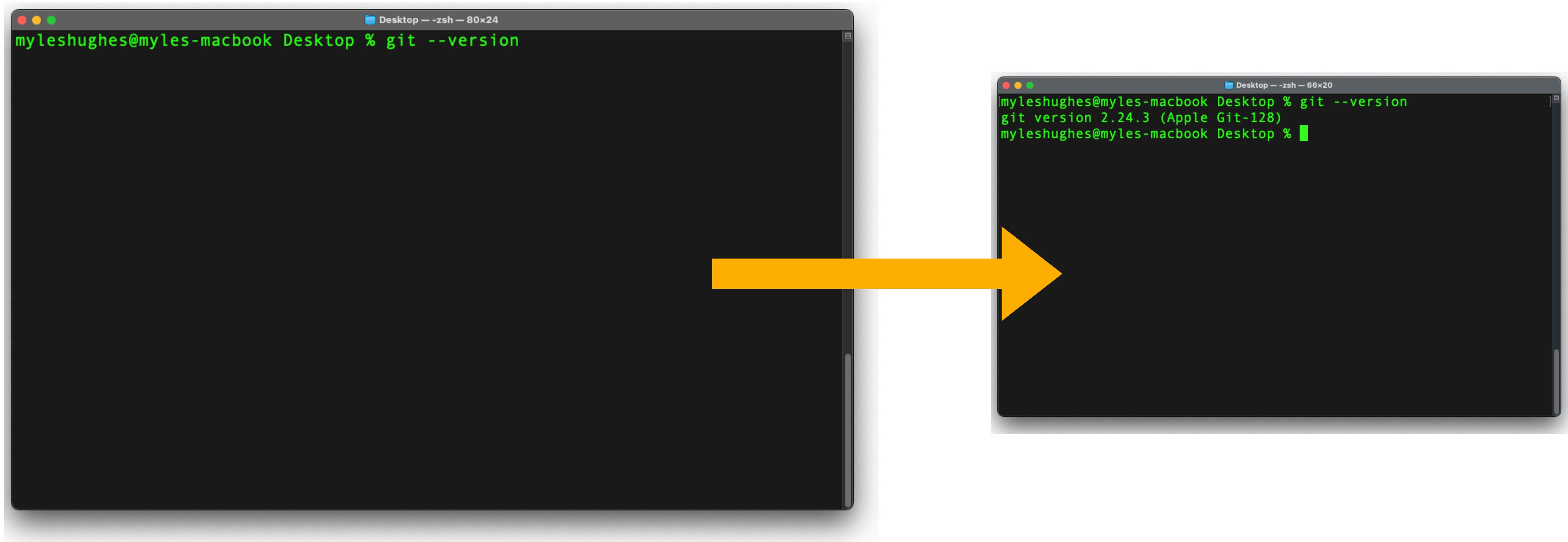


How to set up Git & GitHub

1. Check that you have git installed

Use the command: `git --version`



```
myleshughes@myles-macbook Desktop % git --version
```

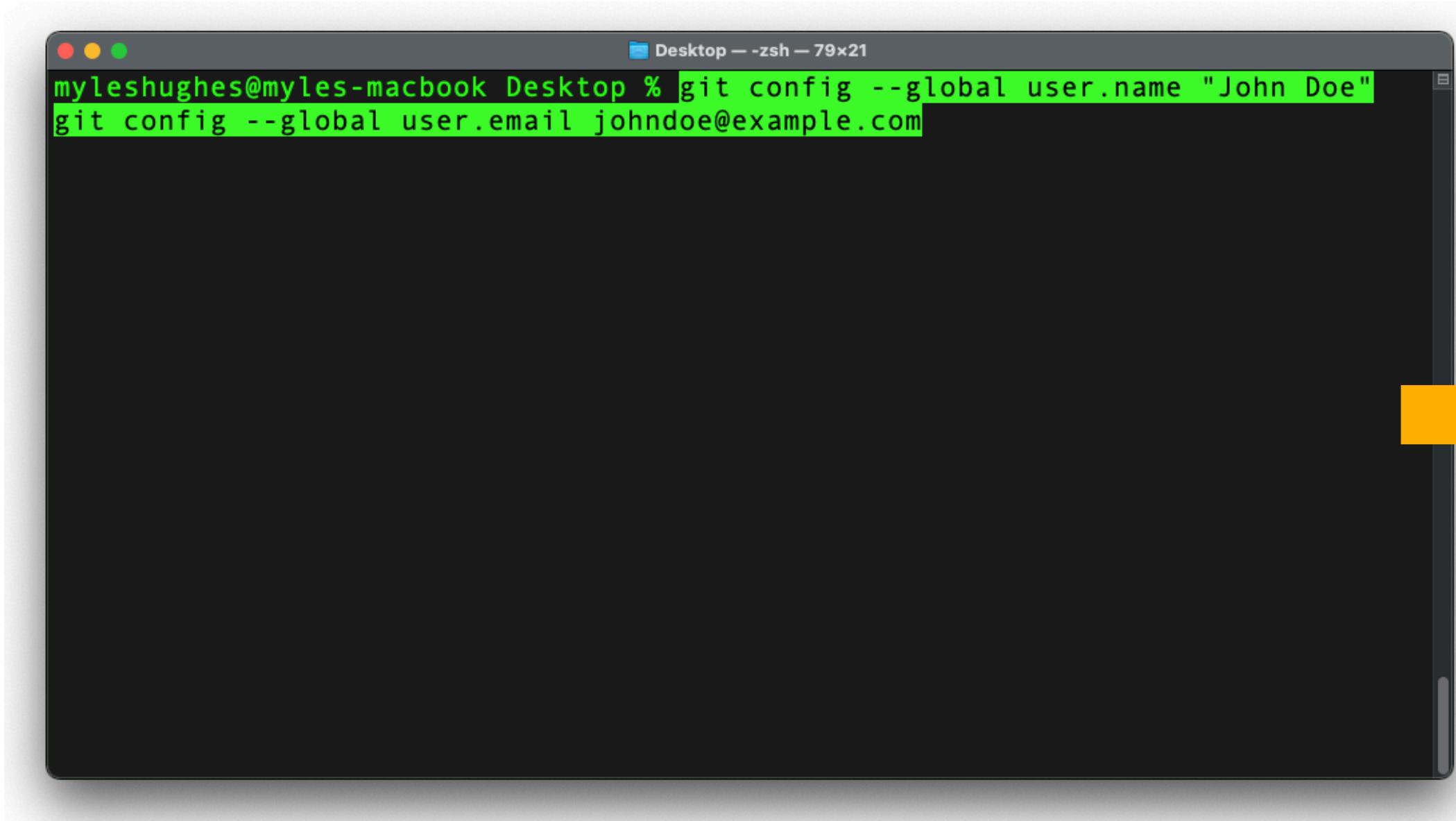
```
myleshughes@myles-macbook Desktop % git --version
git version 2.24.3 (Apple Git-128)
myleshughes@myles-macbook Desktop %
```

2. Use the git config commands

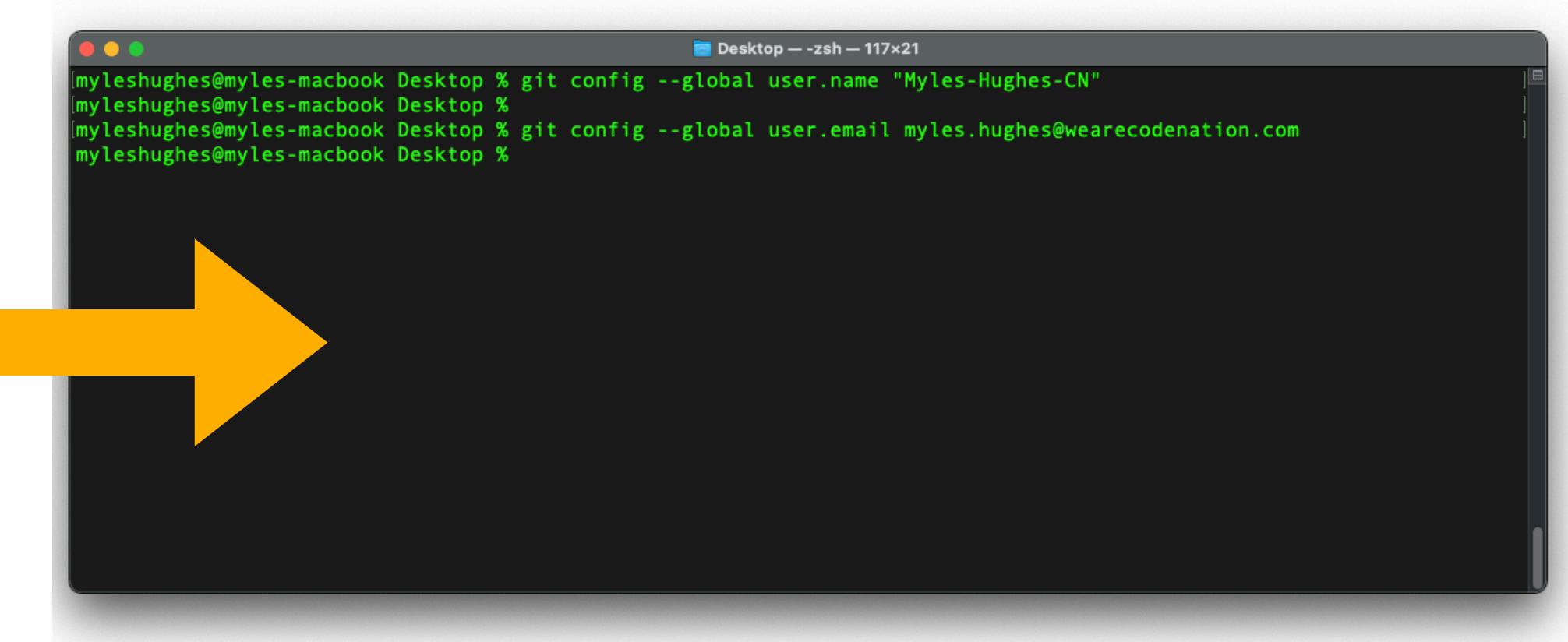
Use the commands:

git config --global user.name "John Doe"

git config --global user.email johndoe@example.com



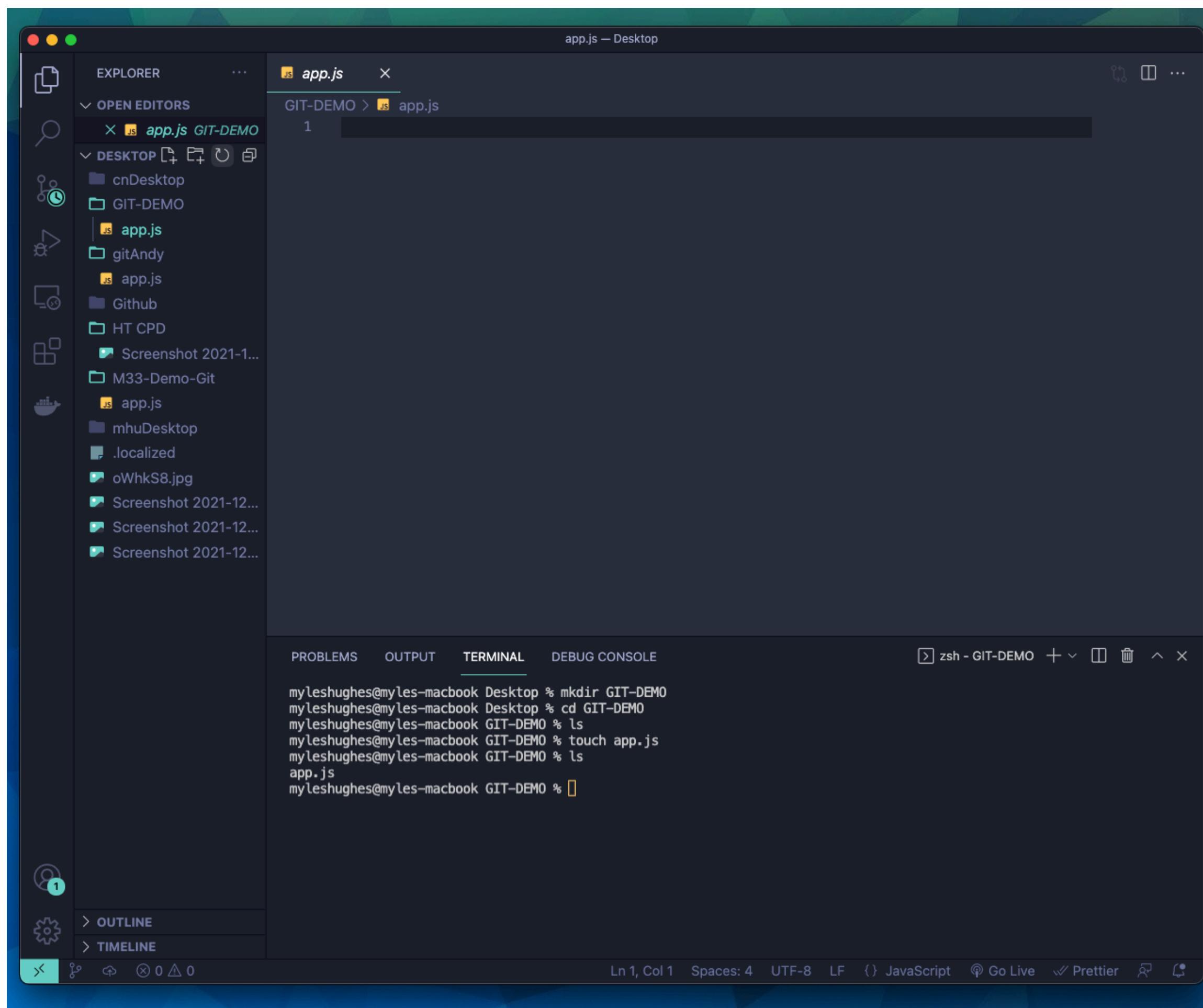
```
myleshughes@myles-macbook Desktop % git config --global user.name "John Doe"
git config --global user.email johndoe@example.com
```



```
myleshughes@myles-macbook Desktop % git config --global user.name "Myles-Hughes-CN"
myleshughes@myles-macbook Desktop %
myleshughes@myles-macbook Desktop % git config --global user.email myles.hughes@wearecodenation.com
myleshughes@myles-macbook Desktop %
```

3. Open up VS Code & set up a new folder

For this example, I am setting up a folder named “GIT-DEMO” on my desktop. I have also added a file named “app.js” into that folder.



This screenshot shows the VS Code terminal window with the "TERMINAL" tab selected. It displays the command history from the previous screenshot:

```
myleshughes@myles-macbook Desktop % mkdir GIT-DEMO
myleshughes@myles-macbook Desktop % cd GIT-DEMO
myleshughes@myles-macbook GIT-DEMO % ls
myleshughes@myles-macbook GIT-DEMO % touch app.js
myleshughes@myles-macbook GIT-DEMO % ls
app.js
myleshughes@myles-macbook GIT-DEMO %
```

4. Create a local repo using the command line

Use the commands:

git init

git status

git add app.js

git status

git commit -m “enter a comment here”

git status

**You should now have a local repo -
however you won't be able to see this on
your desktop or in your file explorer.**

The screenshot shows a terminal window in the VS Code interface. The terminal output is as follows:

```
myleshughes@myles-macbook Desktop % mkdir GIT-DEMO
myleshughes@myles-macbook Desktop % cd GIT-DEMO
myleshughes@myles-macbook GIT-DEMO % ls
myleshughes@myles-macbook GIT-DEMO % touch app.js
myleshughes@myles-macbook GIT-DEMO % ls
app.js
myleshughes@myles-macbook GIT-DEMO % git init
Initialized empty Git repository in /Users/myleshughes/Desktop/GIT-DEMO/.git/
myleshughes@myles-macbook GIT-DEMO % git status
On branch master

No commits yet

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

nothing added to commit but untracked files present (use "git add" to track)
myleshughes@myles-macbook GIT-DEMO % git add.js
git: 'add.js' is not a git command. See 'git --help'.
myleshughes@myles-macbook GIT-DEMO % git add app.js
myleshughes@myles-macbook GIT-DEMO % git stauts
git: 'stauts' is not a git command. See 'git --help'.

The most similar command is
  status
myleshughes@myles-macbook GIT-DEMO % git status
On branch master

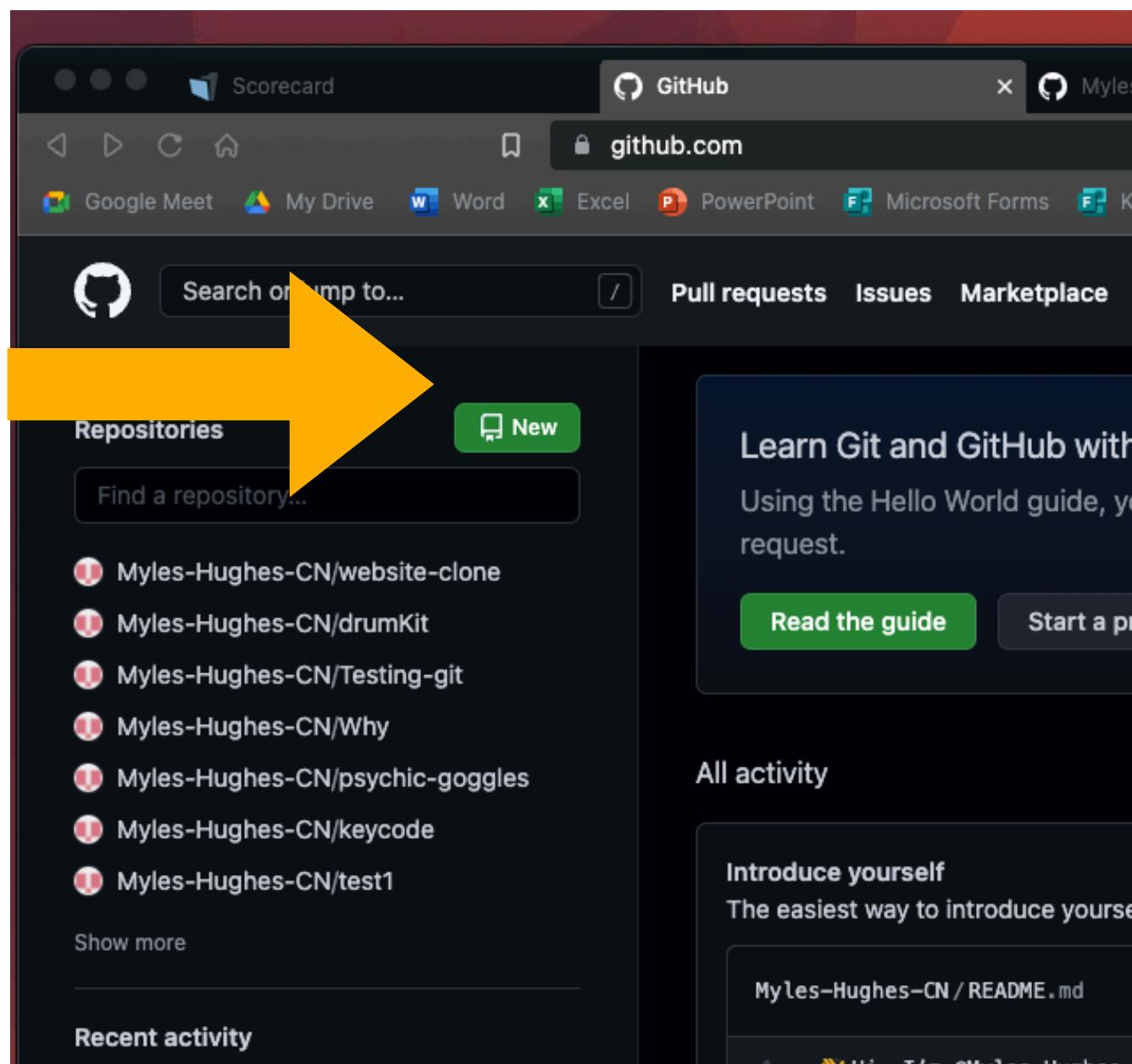
No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   app.js

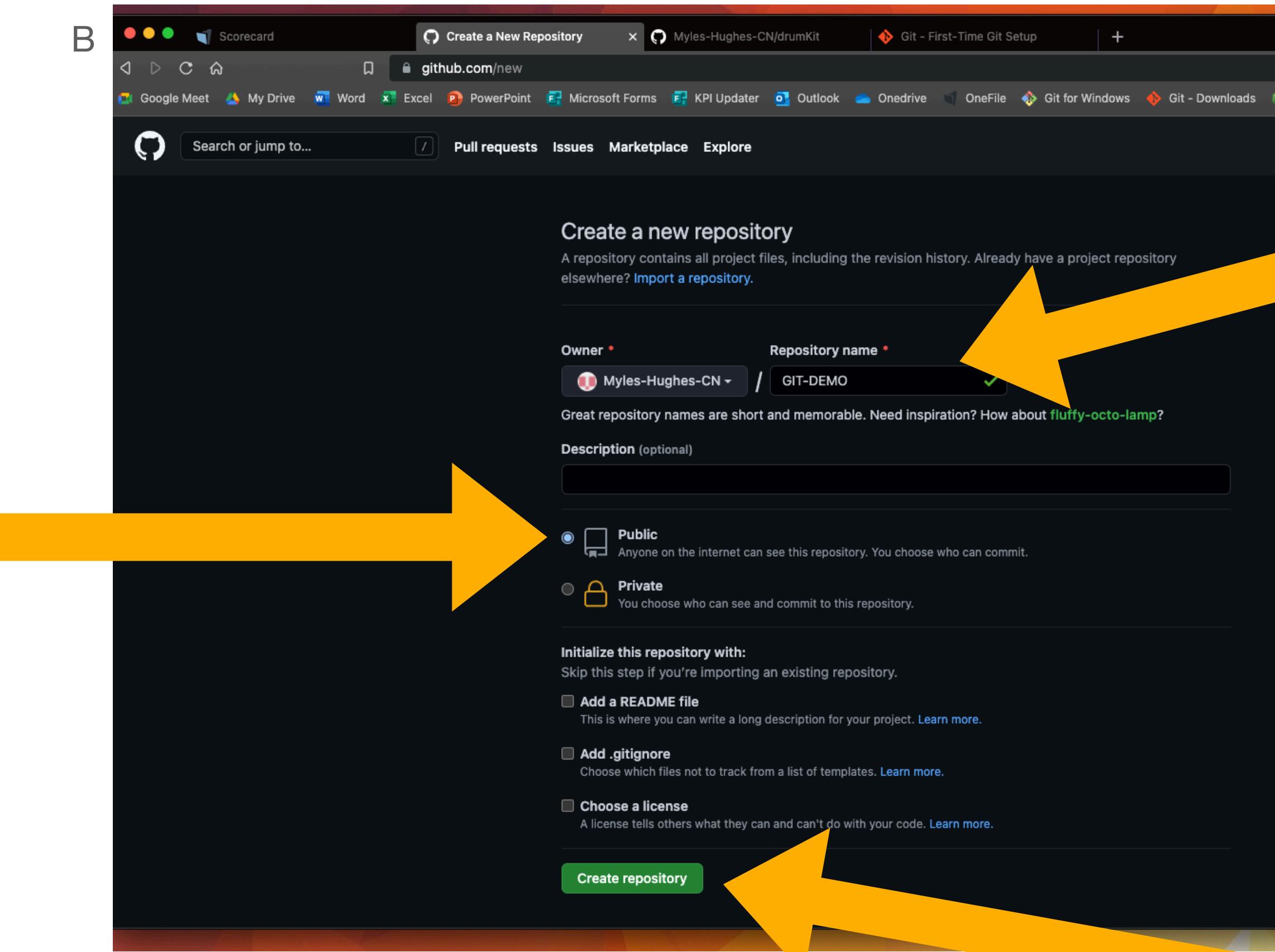
myleshughes@myles-macbook GIT-DEMO % git commit -m "Initial commit"
[master (root-commit) 87304ff] Initial commit
  1 file changed, 1 insertion(+)
  create mode 100644 app.js
myleshughes@myles-macbook GIT-DEMO % git status
On branch master
nothing to commit, working tree clean
myleshughes@myles-macbook GIT-DEMO %
```

5. Create a remote repo on GitHub

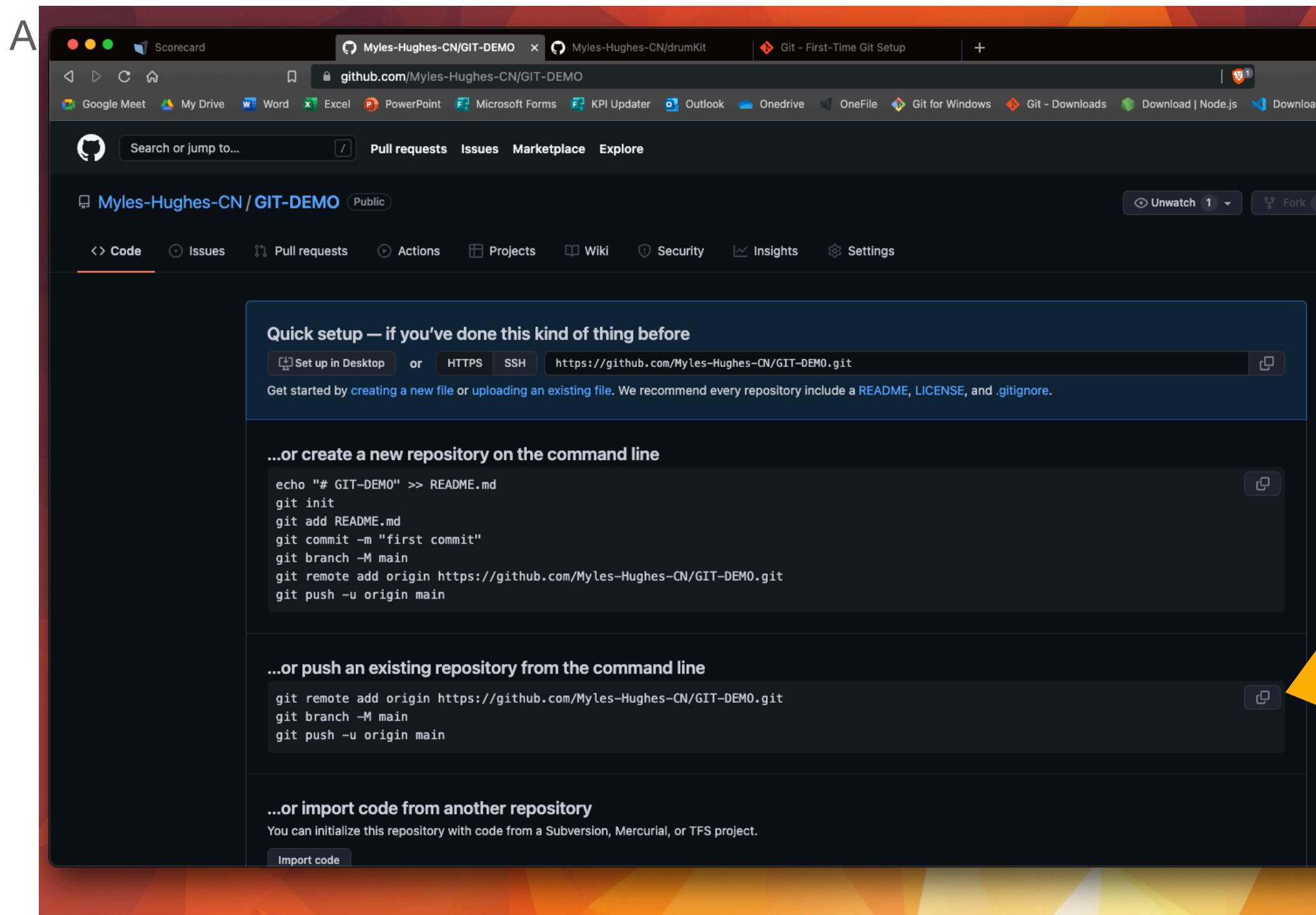
A



B



5. Create a remote repo on GitHub cont...



B

```
The most similar command is
status
myleshughes@myles-macbook GIT-DEMO % git status
On branch master
No commits yet
Changes to be committed:
(use "git rm --cached <file>..." to unstage)
  new file: app.js

myleshughes@myles-macbook GIT-DEMO % git commit -m "Initial commit"
[master (root-commit) 87304ff] Initial commit
  1 file changed, 1 insertion(+)
    create mode 100644 app.js
myleshughes@myles-macbook GIT-DEMO % git status
On branch master
nothing to commit, working tree clean
myleshughes@myles-macbook GIT-DEMO % git remote add origin https://github.com/Myles-Hughes-CN/GIT-DEMO.git
git branch -M main
git push -u origin main
```

C

```
new file: app.js
myleshughes@myles-macbook GIT-DEMO % git commit -m "Initial commit"
[master (root-commit) 87304ff] Initial commit
  1 file changed, 1 insertion(+)
    create mode 100644 app.js
myleshughes@myles-macbook GIT-DEMO % git status
On branch master
nothing to commit, working tree clean
myleshughes@myles-macbook GIT-DEMO % git remote add origin https://github.com/Myles-Hughes-CN/GIT-DEMO.git
git branch -M main
git push -u origin main
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 250 bytes | 250.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://github.com/Myles-Hughes-CN/GIT-DEMO.git
 * [new branch] main -> main
Branch 'main' set up to track remote branch 'main' from 'origin'.
myleshughes@myles-macbook GIT-DEMO %
```

6. Push to your remote repo

Use the command: `git push`

A

```
on branch master
nothing to commit, working tree clean
myleshughes@myles-macbook GIT-DEMO % git remote add origin https://github.com/Myles-Hughes-CN/GIT-DEMO.git
git branch -M main
git push -u origin main
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 250 bytes | 250.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://github.com/Myles-Hughes-CN/GIT-DEMO.git
 * [new branch]    main -> main
Branch 'main' set up to track remote branch 'main' from 'origin'.
myleshughes@myles-macbook GIT-DEMO % git push
Everything up-to-date
myleshughes@myles-macbook GIT-DEMO %
```

B

Check that your file(s) have been pushed.

6. Commit every time you make a change

Try to commit as often as possible.

This really helps when having to roll back broken apps.

A

The screenshot shows the VS Code interface with the Explorer sidebar open, displaying a file tree. In the center, there are two code editors for the file `app.js`. The first editor shows the initial commit:

```
// This is my first edit.
```

The second editor shows the second commit:

```
// This is a second edit.
```

Below the editors, the terminal window shows the command-line history of the git operations:

```
git commit -m "Initial commit"
[master (root-commit) 87304ff] Initial commit
 1 file changed, 1 insertion(+)
 create mode 100644 app.js
git status
On branch master
nothing to commit, working tree clean
git remote add origin https://github.com/Myles-Hughes-CN/GIT-DEMO.git
git branch -M main
git push -u origin main
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 250 bytes | 250.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://github.com/Myles-Hughes-CN/GIT-DEMO.git
 * [new branch]  main -> main
Branch 'main' set up to track remote branch 'main' from 'origin'.
git push -u origin main
Everything up-to-date
git status
On branch main
Your branch is up to date with 'origin/main'.
```

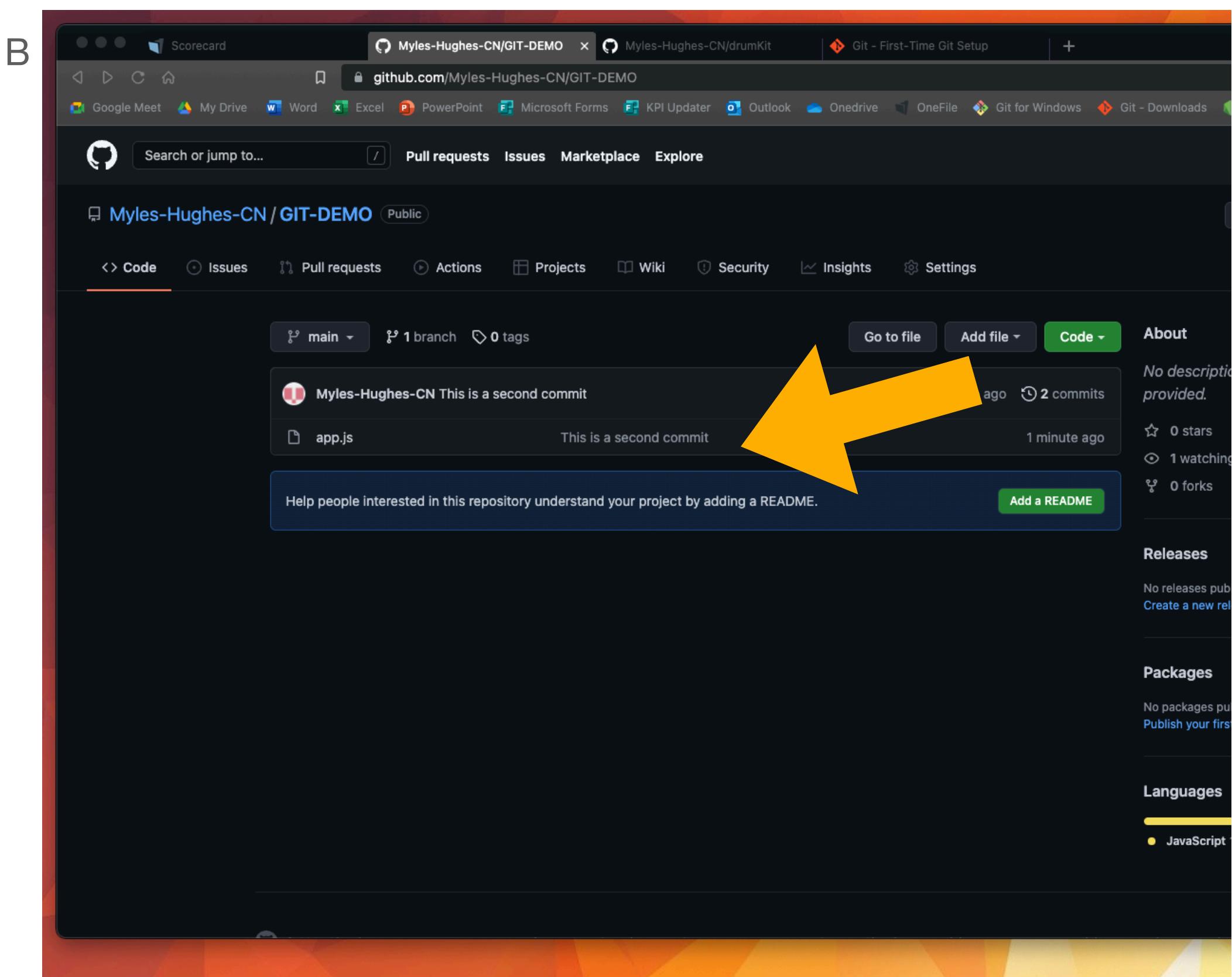
Changes not staged for commit:
(use "git add <file>..." to update what will be committed)
(use "git restore <file>..." to discard changes in working directory)
modified: app.js

no changes added to commit (use "git add" and/or "git commit -a")

Changes to be committed:
(use "git restore --staged <file>..." to unstage)
modified: app.js

```
git commit -m "This is a second commit"
[main 843a1be] This is a second commit
 1 file changed, 2 insertions(+), 1 deletion(-)
git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 304 bytes | 304.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://github.com/Myles-Hughes-CN/GIT-DEMO.git
 87304ff..843a1be  main -> main
myleshughes@myles-macbook GIT-DEMO %
```

B

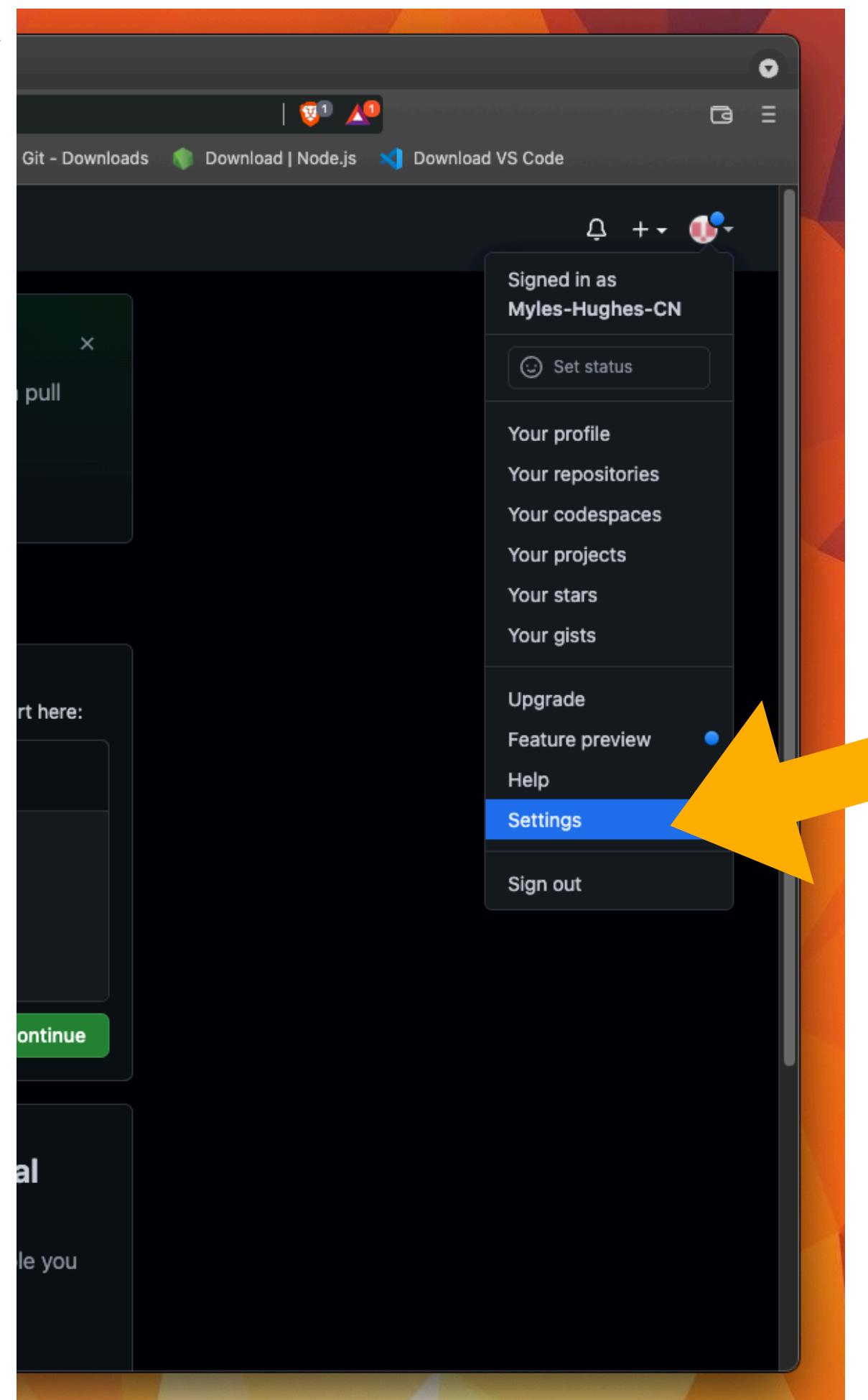


Further information

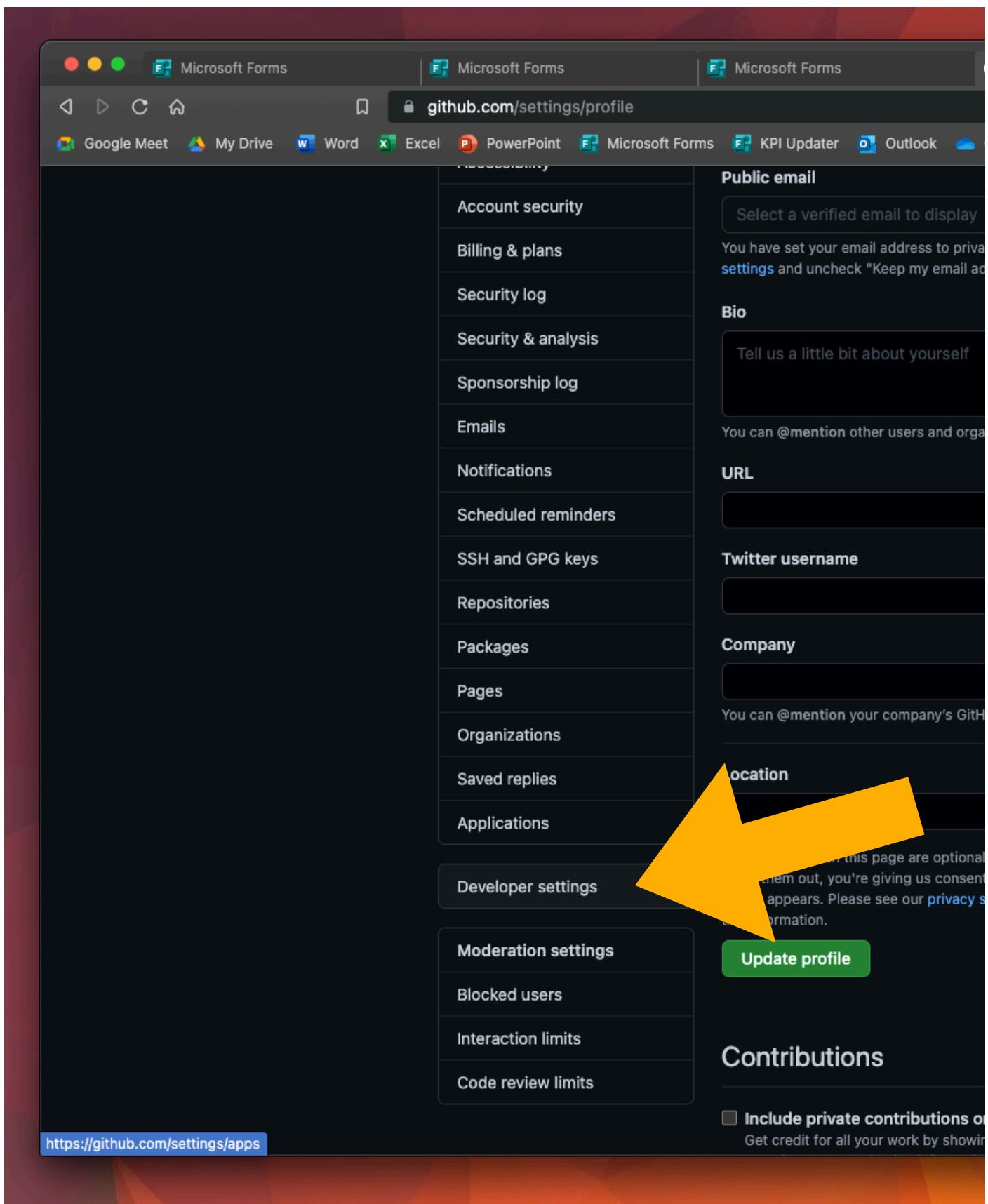
- At any point, git may ask you for a password or Personal Access Token.
These can be set up on GitHub (see PAT GUIDE on the next slide).
- <https://git-scm.com/book/en/v2/Getting-Started-First-Time-Git-Setup>
- <https://www.youtube.com/watch?v=USjZcfj8yxE>
- <https://www.youtube.com/watch?v=nhNq2klvi9s>

PAT GUIDE

A



B



PAT GUIDE

A screenshot of a web browser showing the GitHub 'Personal Access Tokens' settings page. The browser has three tabs open, all titled 'Personal Access Tokens'. The main content area shows a list of existing tokens and a button to generate a new one.

The page title is 'Personal Access Tokens' and the URL is github.com/settings/tokens.

The sidebar on the left lists 'GitHub Apps', 'OAuth Apps', and 'Personal access tokens', with 'Personal access tokens' being the active tab, indicated by a red border.

The main content area displays three tokens:

- 133-CN** — admin:enterprise, admin:gpg_key, admin:org, admin:org_hook, admin:public_key, Never used
admin:repo_hook, delete:packages, delete_repo, gist, notifications, repo, user, workflow, write:discussion, write:packages
⚠️ This token has no expiration date.
Expires on Sat, Jan 15 2022.
- Test** — admin:enterprise, admin:gpg_key, admin:org, admin:org_hook, admin:public_key, Never used
admin:repo_hook, delete:packages, delete_repo, gist, notifications, repo, user, workflow, write:discussion, write:packages
- Myles-MBA** — admin:enterprise, admin:gpg_key, admin:org, Last used within the last week
admin:org_hook, admin:public_key, admin:repo_hook, delete:packages, delete_repo, gist, notifications, repo, user, workflow, write:discussion, write:packages
⚠️ This token has no expiration date.

At the bottom of the page, a note states: "Personal access tokens function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to authenticate to the API over Basic Authentication."

The bottom navigation bar includes links for GitHub, Inc., Terms, Privacy, Security, Status, Docs, Contact GitHub, Pricing, API, Training, Blog, and About.

The status bar at the bottom shows the URL <https://github.com/settings/tokens/new>.

PAT GUIDE

Choose No Expiration

A

Settings / Developer settings

New personal access token

Personal access tokens function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to authenticate to the API over Basic Authentication.

Note

Add a name

What's this token for?

Expiration *

No expiration The token will never expire!

GitHub strongly recommends that you set an expiration date for your token to help keep your information secure. [Learn more](#)

Select scopes

Scopes define the access for personal tokens. [Read more about OAuth scopes](#).

<input type="checkbox"/> repo	Full control of private repositories
<input type="checkbox"/> repo:status	Access commit status
<input type="checkbox"/> repo_deployment	Access deployment status
<input type="checkbox"/> public_repo	Access public repositories
<input type="checkbox"/> repo:invite	Access repository invitations
<input type="checkbox"/> security_events	Read and write security events
<input type="checkbox"/> workflow	Update GitHub Action workflows
<input type="checkbox"/> write:packages	Upload packages to GitHub Package Registry
<input type="checkbox"/> read:packages	Download packages from GitHub Package Registry
<input type="checkbox"/> delete:packages	Delete packages from GitHub Package Registry
<input type="checkbox"/> admin:org	Full control of orgs and teams, read and write org projects
<input type="checkbox"/> write:org	Read and write org and team membership, read and write org projects
<input type="checkbox"/> read:org	Read org and team membership, read org projects
<input type="checkbox"/> admin:public_key	Full control of user public keys
<input type="checkbox"/> write:public_key	Write user public keys
<input type="checkbox"/> read:public_key	Read user public keys
<input type="checkbox"/> admin:repo_hook	Full control of repository hooks
<input type="checkbox"/> write:repo_hook	Write repository hooks

Check every box

B

write:discussion Read and write team discussions
read:discussion Read team discussions

admin:enterprise Full control of enterprises
manage_runners:enterprise Manage enterprise runners and runner-groups
manage_billing:enterprise Read and write enterprise billing data
read:enterprise Read enterprise profile data

admin:gpg_key Full control of public user GPG keys (Developer Preview)
write:gpg_key Write GPG keys
read:gpg_key Read GPG keys

Generate token Cancel

Once your PAT has been generated, copy and paste it to somewhere safe.
This can be a word document or similar.

If you lose this token, you will have to create a new one.