GitHub Education

Module 1

Basics

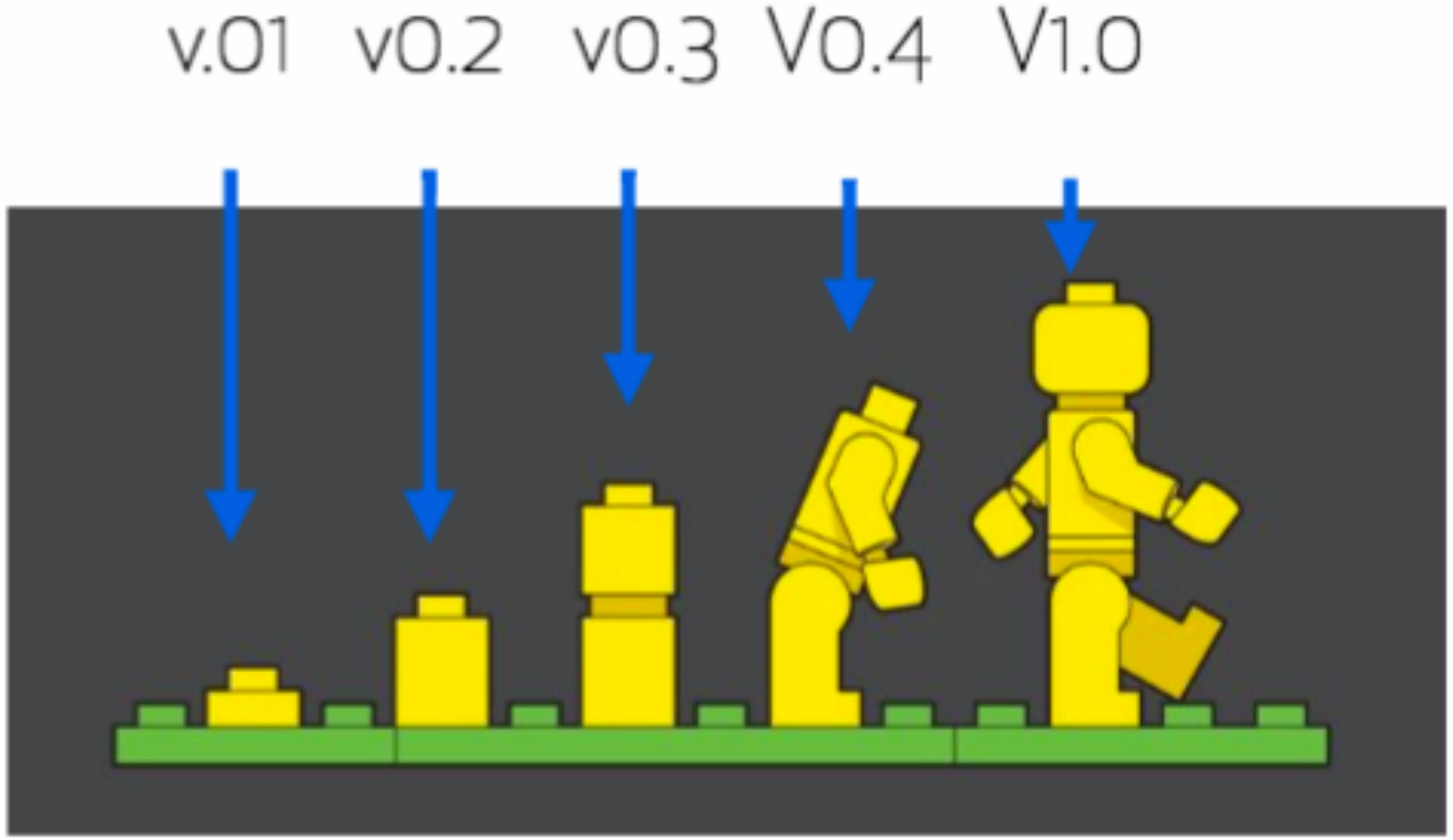
Introduction to Git
Understanding the state of your repository
Being selective with Git
Inside a commit

Questions
Exercises

Git basics
(a.k.a. ‘the internals’)

Git is a *version control system*

A tool that lets you track your progress over time.

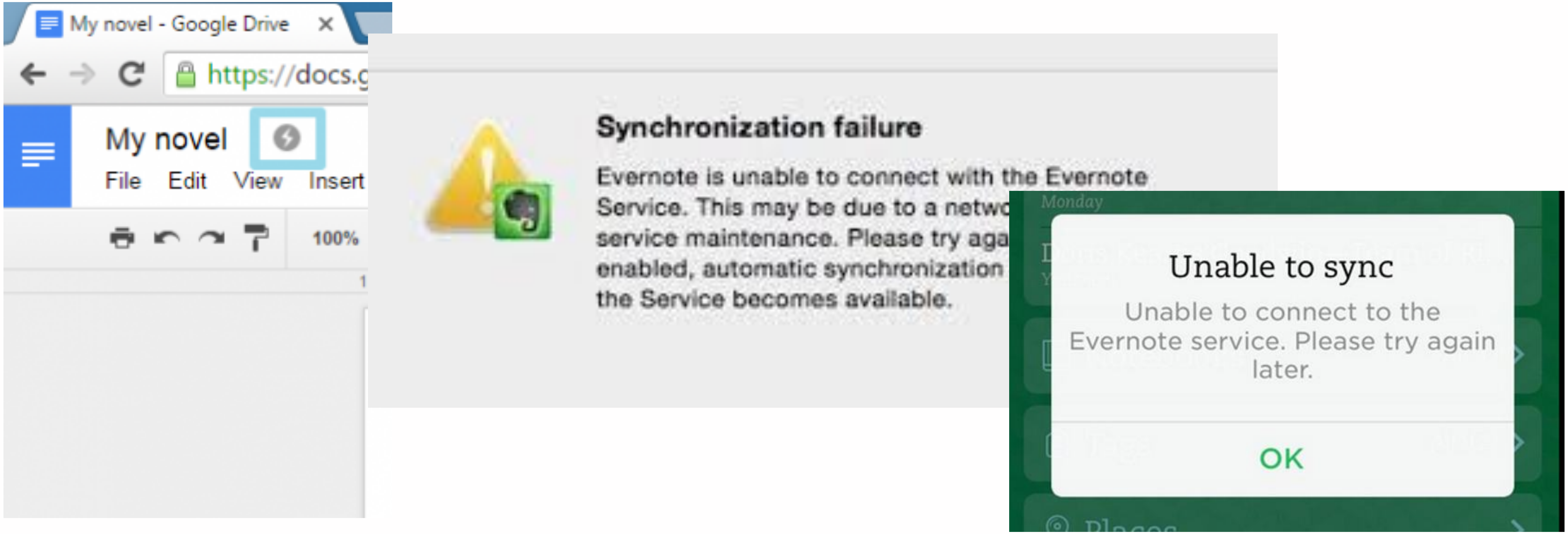


Git takes snapshots

Save snapshots to your history to retrace your steps.

Also keeps others up-to-date with your latest work.

Centralized systems require coordination…



Synchronization failure

Evernote is unable to connect with the Evernote Service. This may be due to a netwo service maintenance. Please try aga enabled, automatic synchronization the Service becomes available.

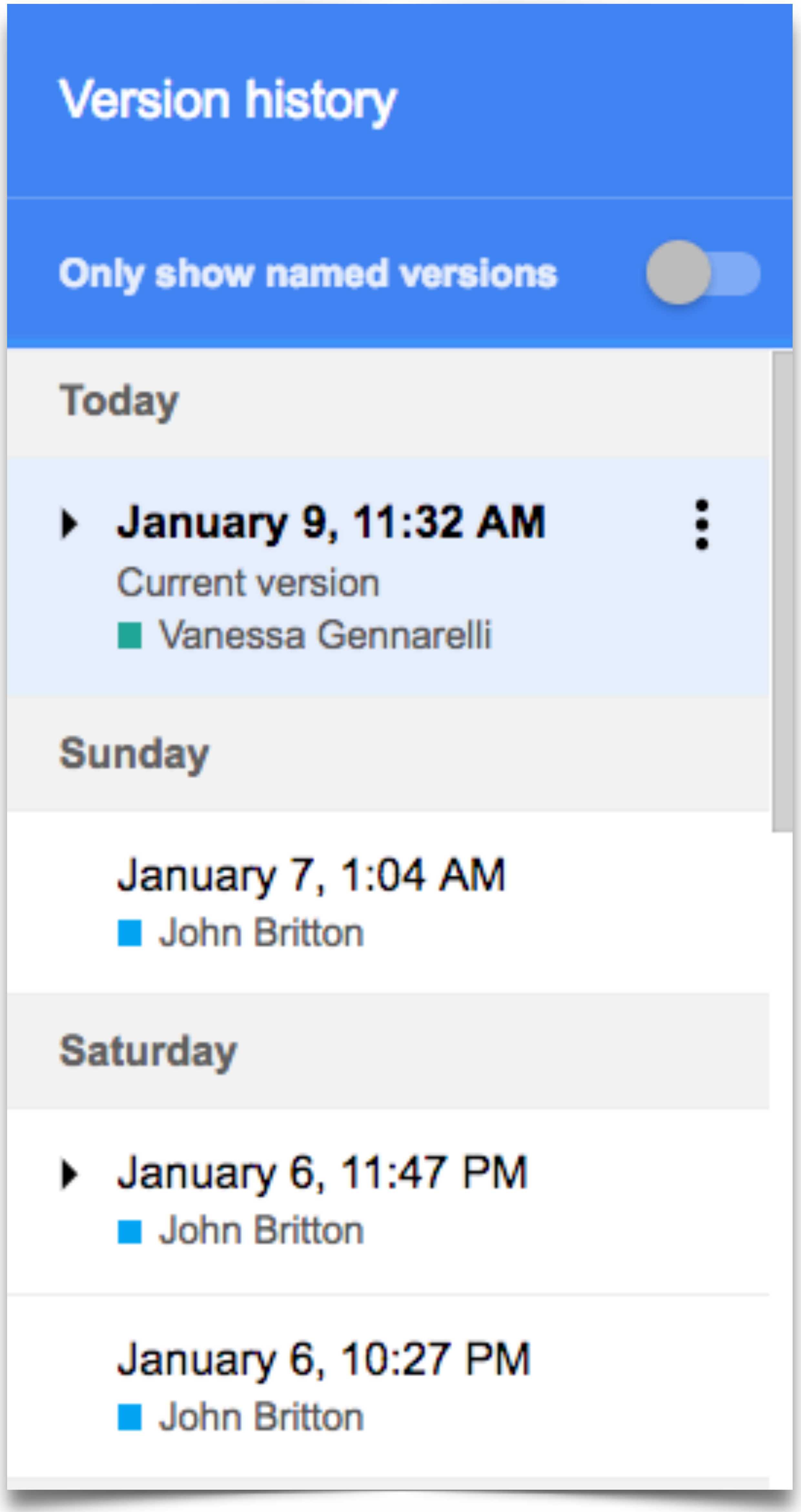
Unable to sync

Unable to connect to the Evernote service. Please try again later.

Order with coordination:

In a centralized system, you can objectively call versions a numerical progression: version 1, version 2, version 3…

Since John made a new version before Vanessa, his is n+1, and Vanessa is n+2.



Version history

Only show named versions

Today

January 9, 11:32 AM
Current version
Vanessa Gennarelli

Sunday

January 7, 1:04 AM
John Britton

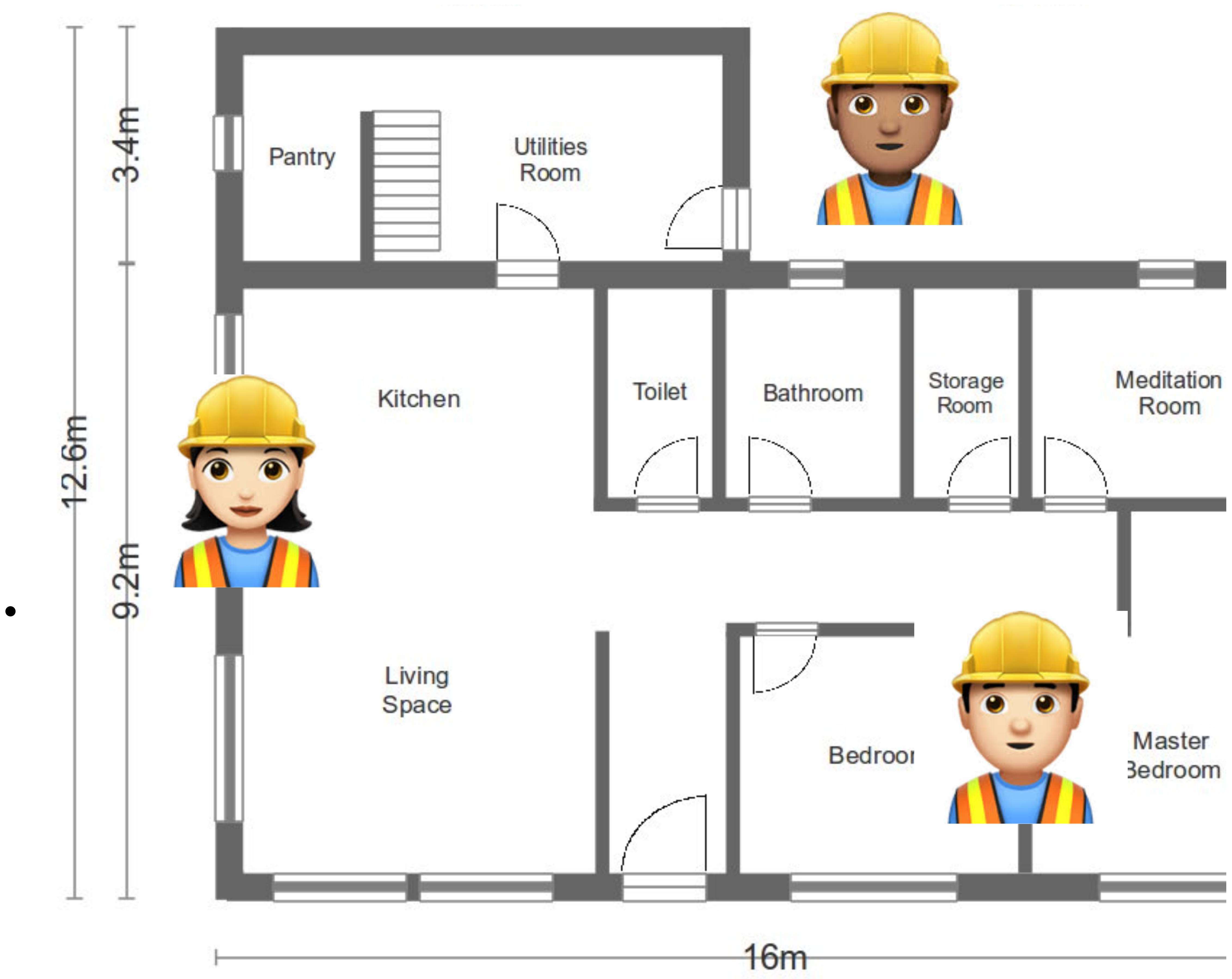
Saturday

January 6, 11:47 PM
John Britton

January 6, 10:27 PM
John Britton

Working in parallel: order without coordination

Git goes after this idea of distributed version control, so you can keep track of your versions without coordination.



In your terminal, check to see if you have Git installed.

```
git —–version
```

If it’s not installed, configure Git to recognize you:

```
git config user.name "Mona Lisa"

git config --global user.email
"email@example.com"
```

A repository holds the entire history of your project

A repository is the unit of separation between projects in Git.

Each project, library or discrete piece of software should have it's own repository.

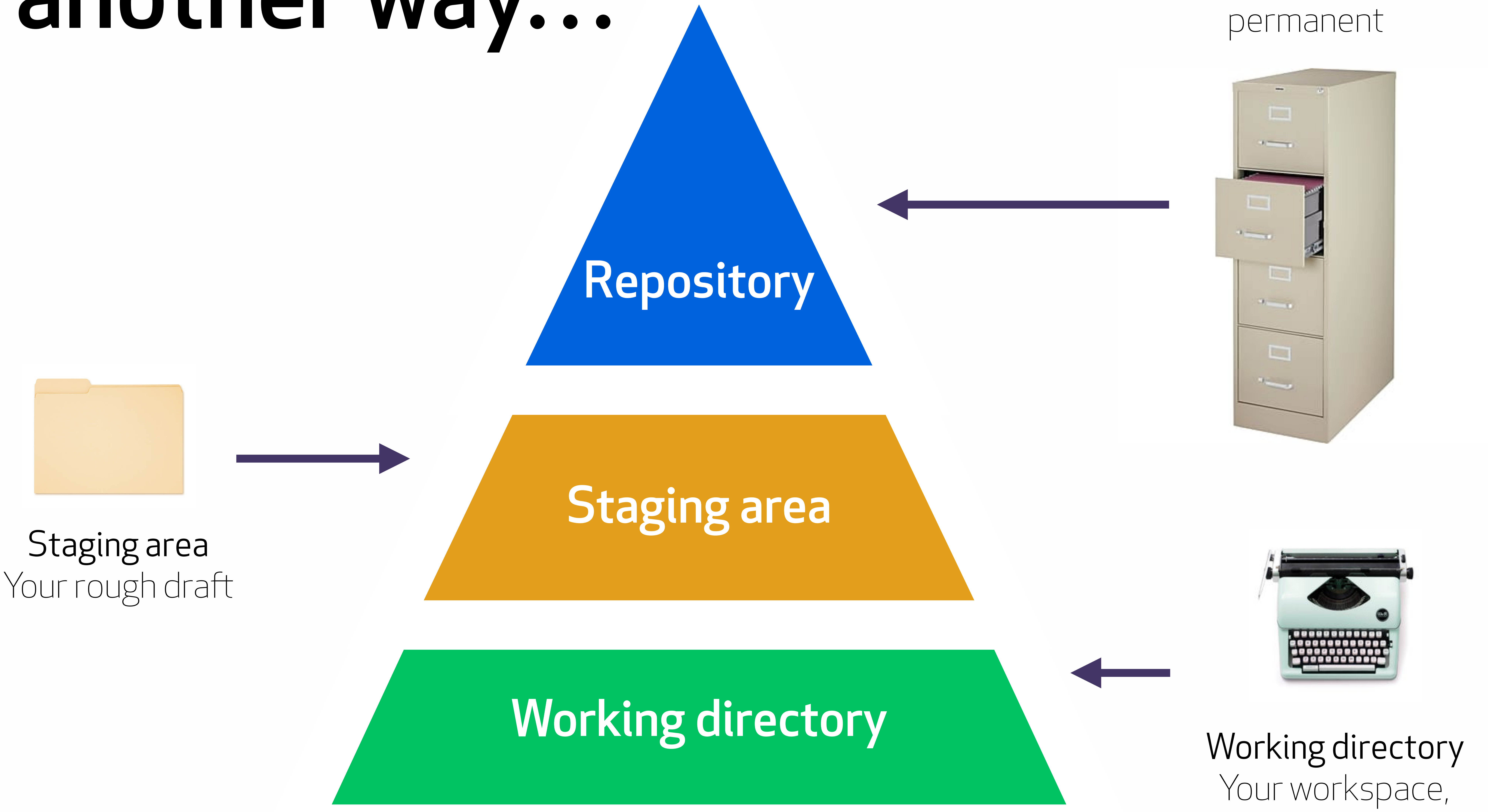
Git is like a desk

Working directory
where you write

Staging area
rough draft, in a
manila folder

Repository
final draft
in the filing cabinet

Put another way...

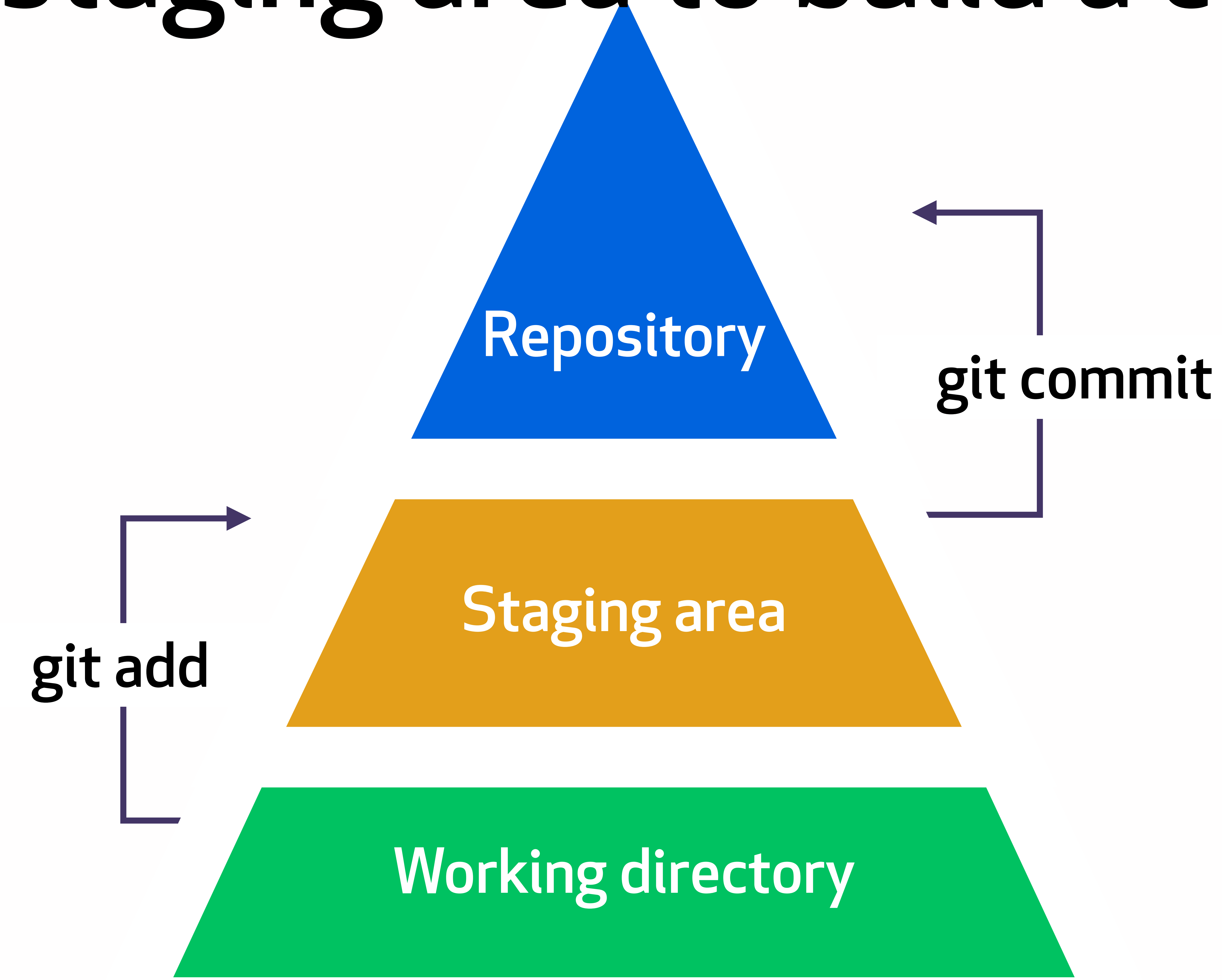


Repository
Final draft,
permanent

Staging area
Your rough draft

Working directory
Your workspace,
local on your machine

Use the staging area to build a commit



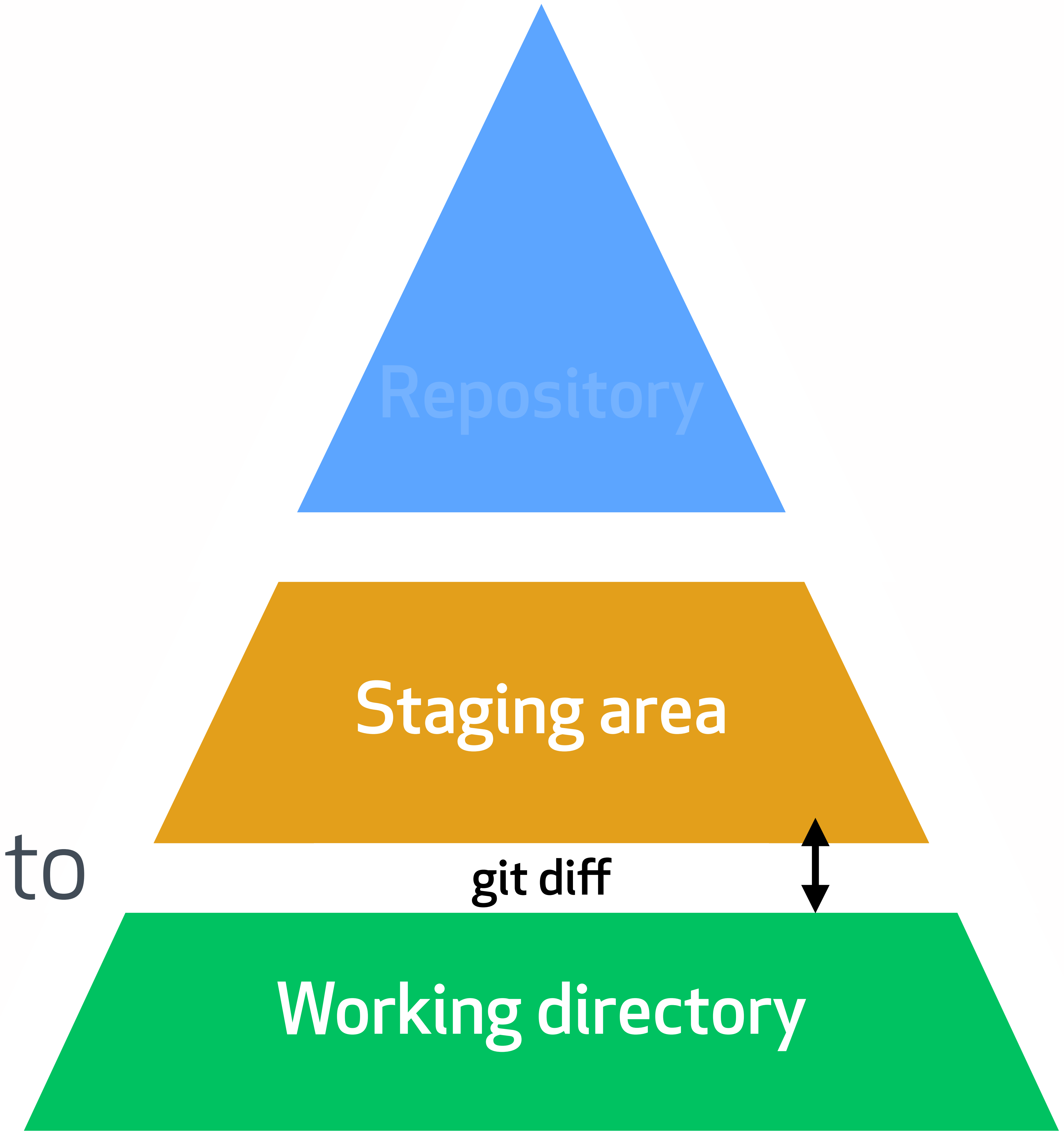
Making commits

`‘git commit’`

tells Git to save that portion of the project from the staging area into the repository history.

When we run git diff
what two things are we comparing?

git diff



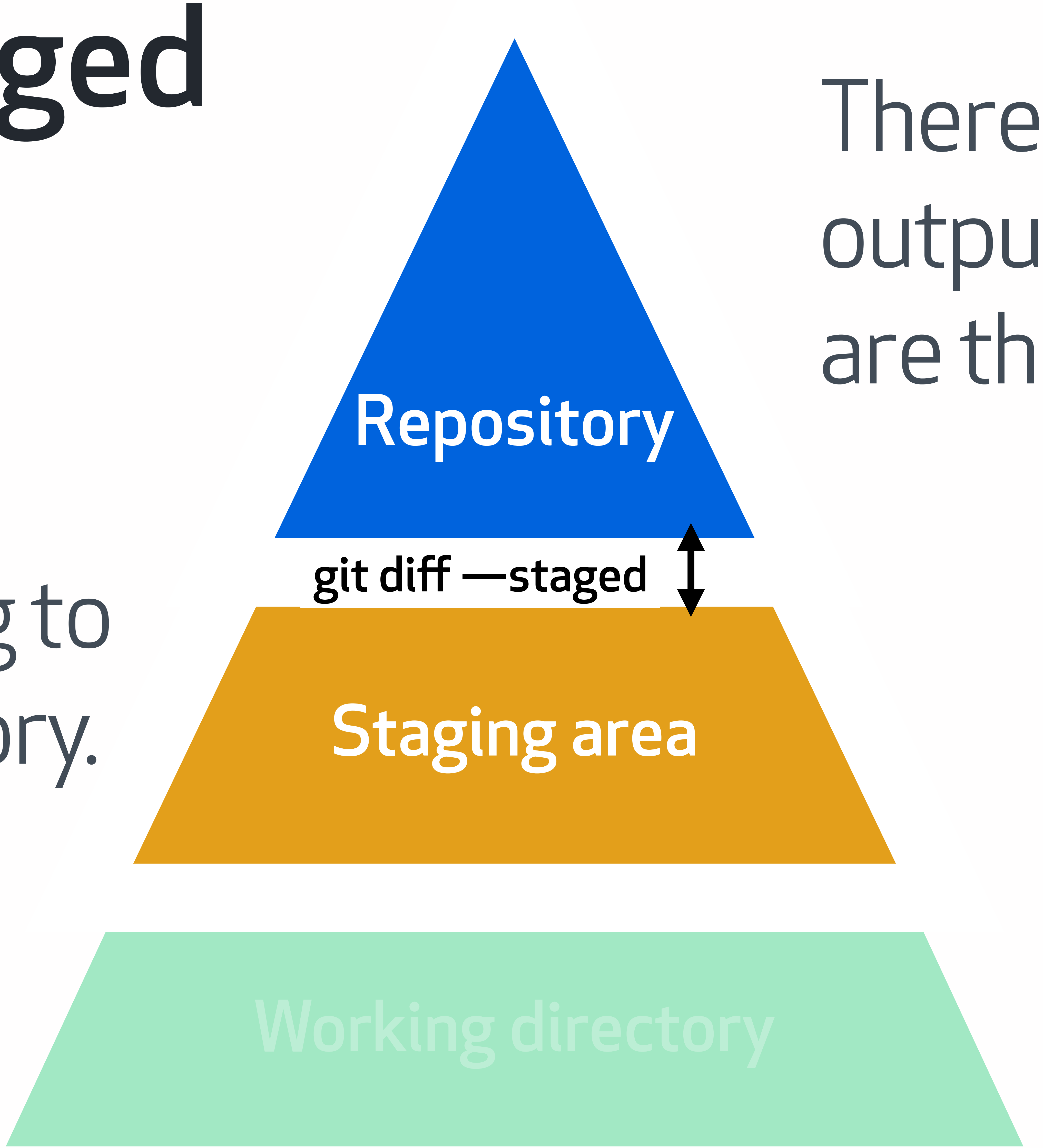
Compares staging to working directory.

There's no output if they are the same.

git diff --staged

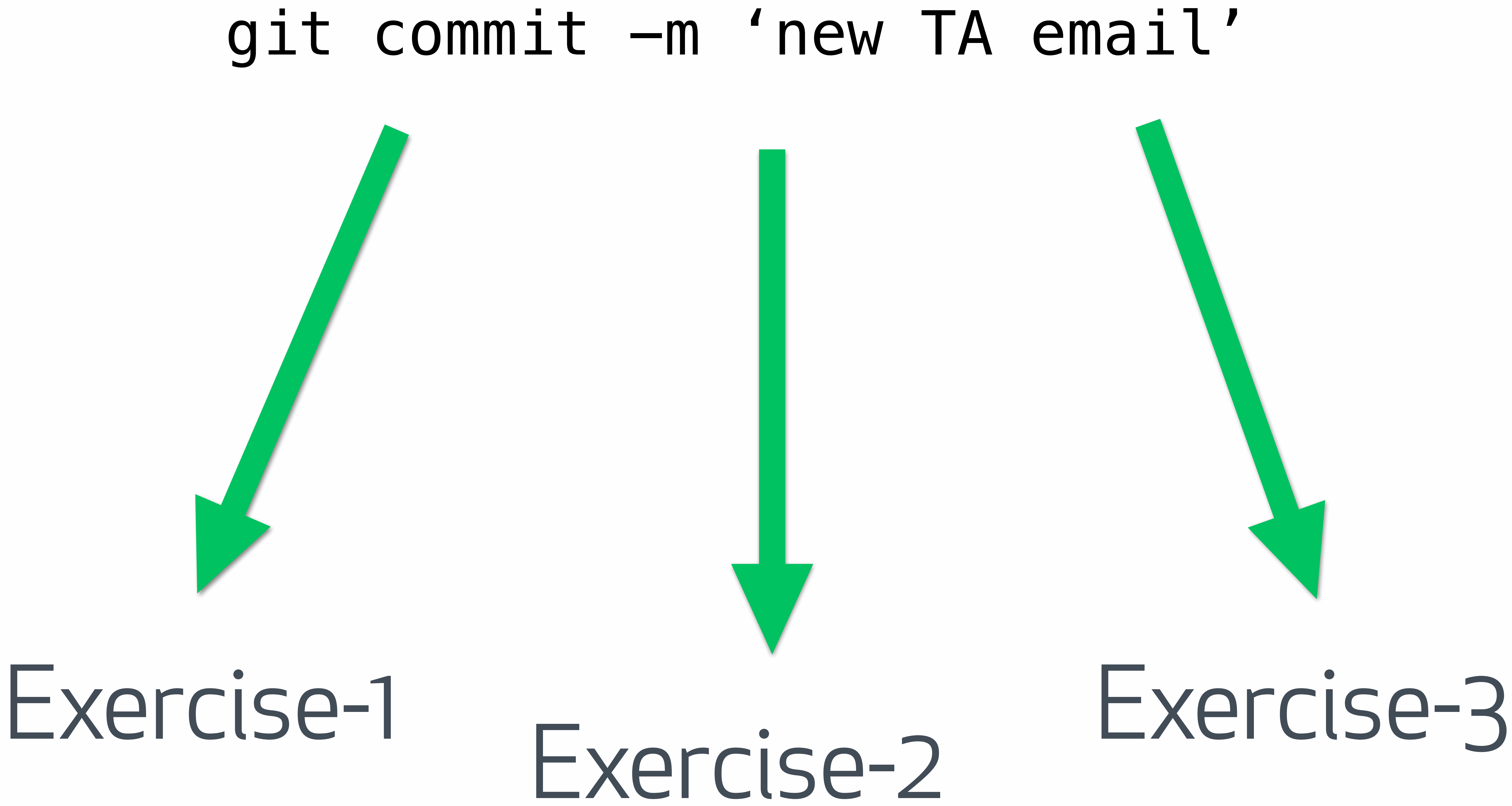
There's no output if they are the same.

Compares staging to repository directory.



Git allows you to be selective

You can fix a bug across several different files in the same commit.



```
git commit –m ‘new TA email’
```

Exercise-1

Exercise-2

Exercise-3

But commits should be logically grouped

Don’t mix typo corrections and new features.

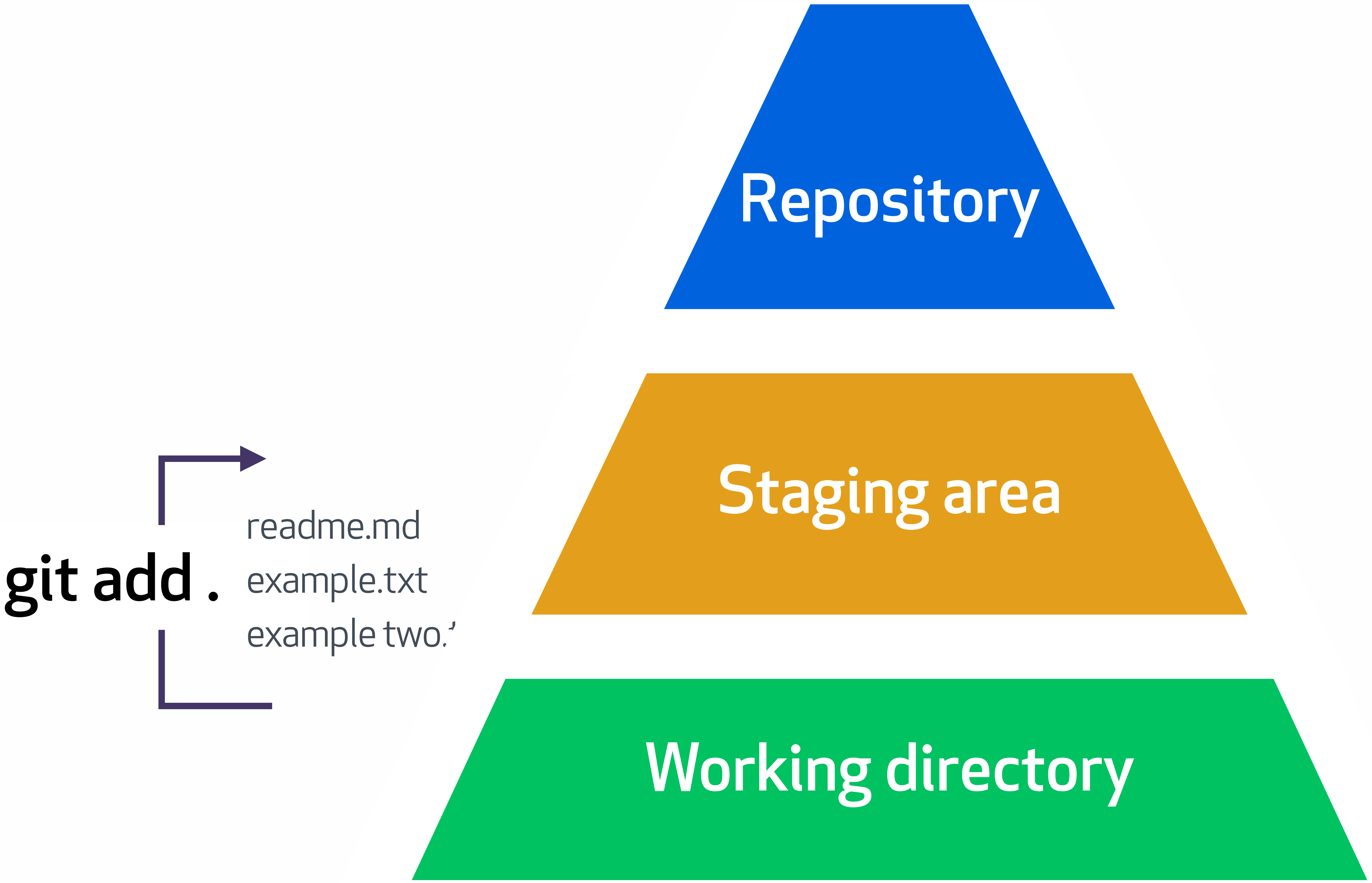
If the feature gets rolled back, you re-introduce the typo.

```
git commit –m ‘typo in readme.md’ ✅

git commit –m ‘new signup flow.’ ✅

git commit –m ‘fix typo, add
field to signup flow, create
parallax effect’ ❌
```

It’s why you should never use `git add .`



Repository

Staging area

Working directory

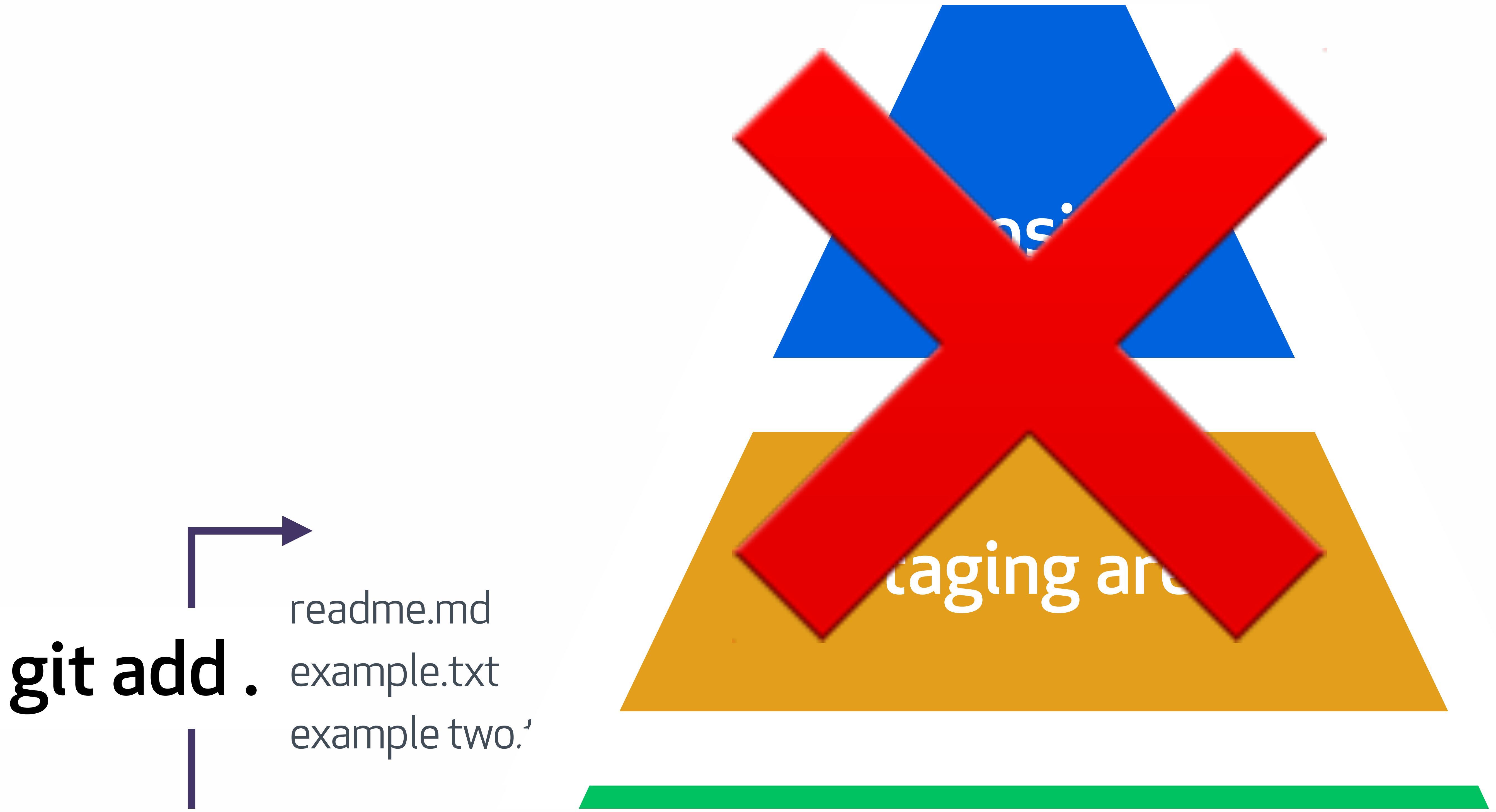
git add .

readme.md

example.txt

example two.’

It’s why you should never use `git add .`



git add .

readme.md

example.txt

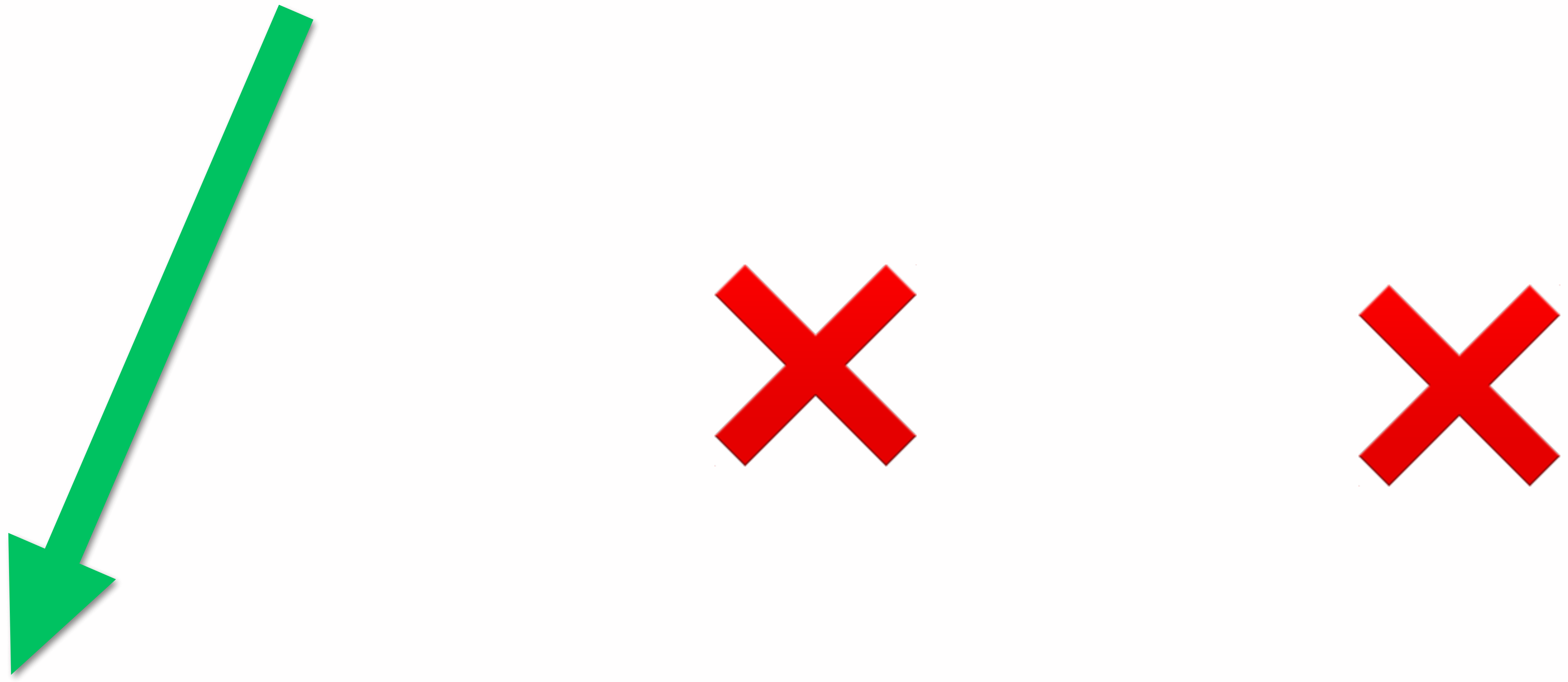
example two.’

it stages changes that aren’t logically related…

Imagine if you revealed solutions in exercise-1

You'd need to update Exercise-1, but you don't need to touch 2 or 3.

```
git commit –m ‘remove key data’
```

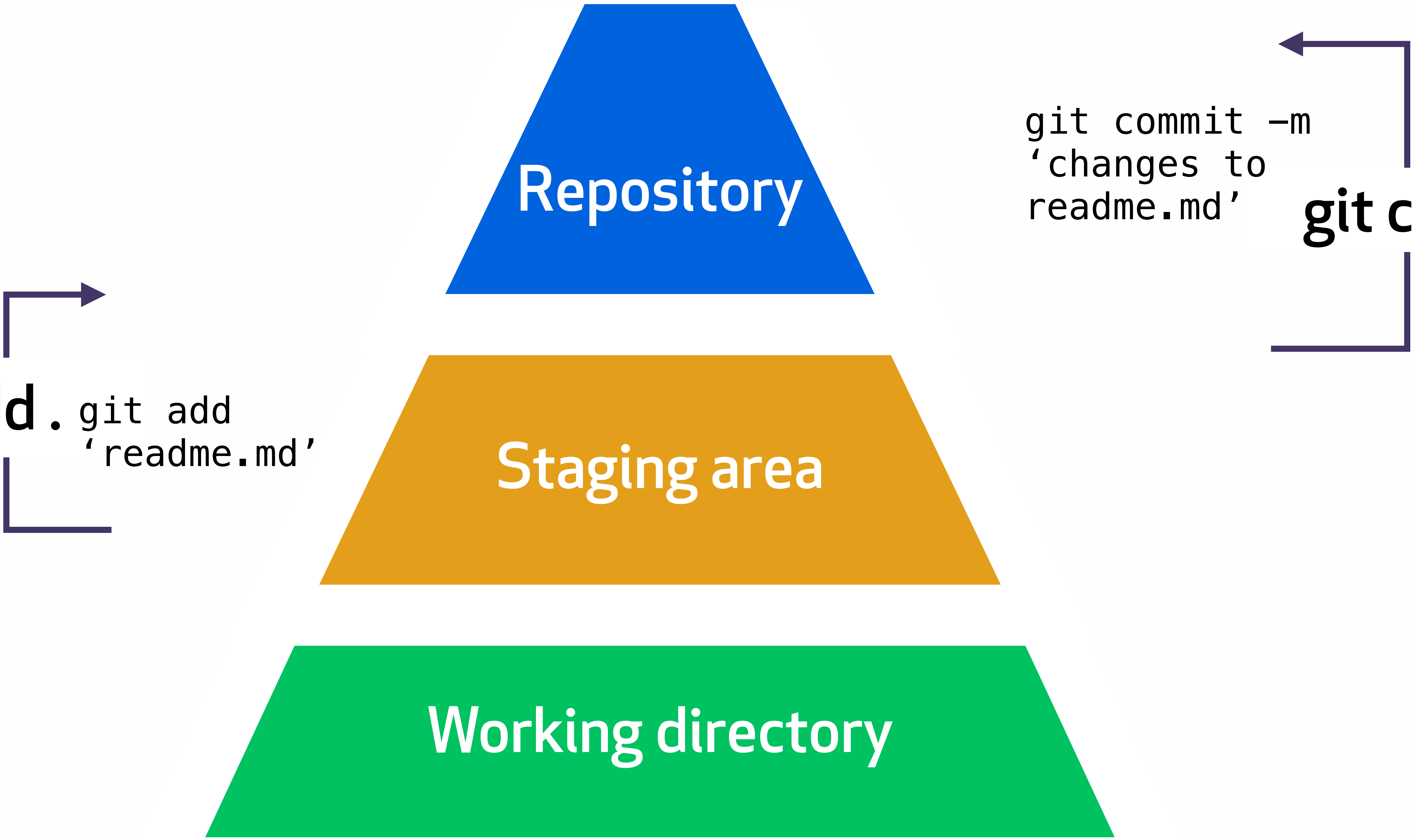


Exercise-1

Exercise-2

Exercise-3

Order of operations:



First, we stage the modifications we've made

git add .

```
git add 'readme.md'
```

Then, commit the modifications to Repository.

git commit

```
git commit -m
'changes to
readme.md'
```

Understanding the state of your repository

Create a repository

```
cd desktop
git init exercise-1
cd exercise-1
ls -al
```

Create a file in your Git repository + add it to staging.

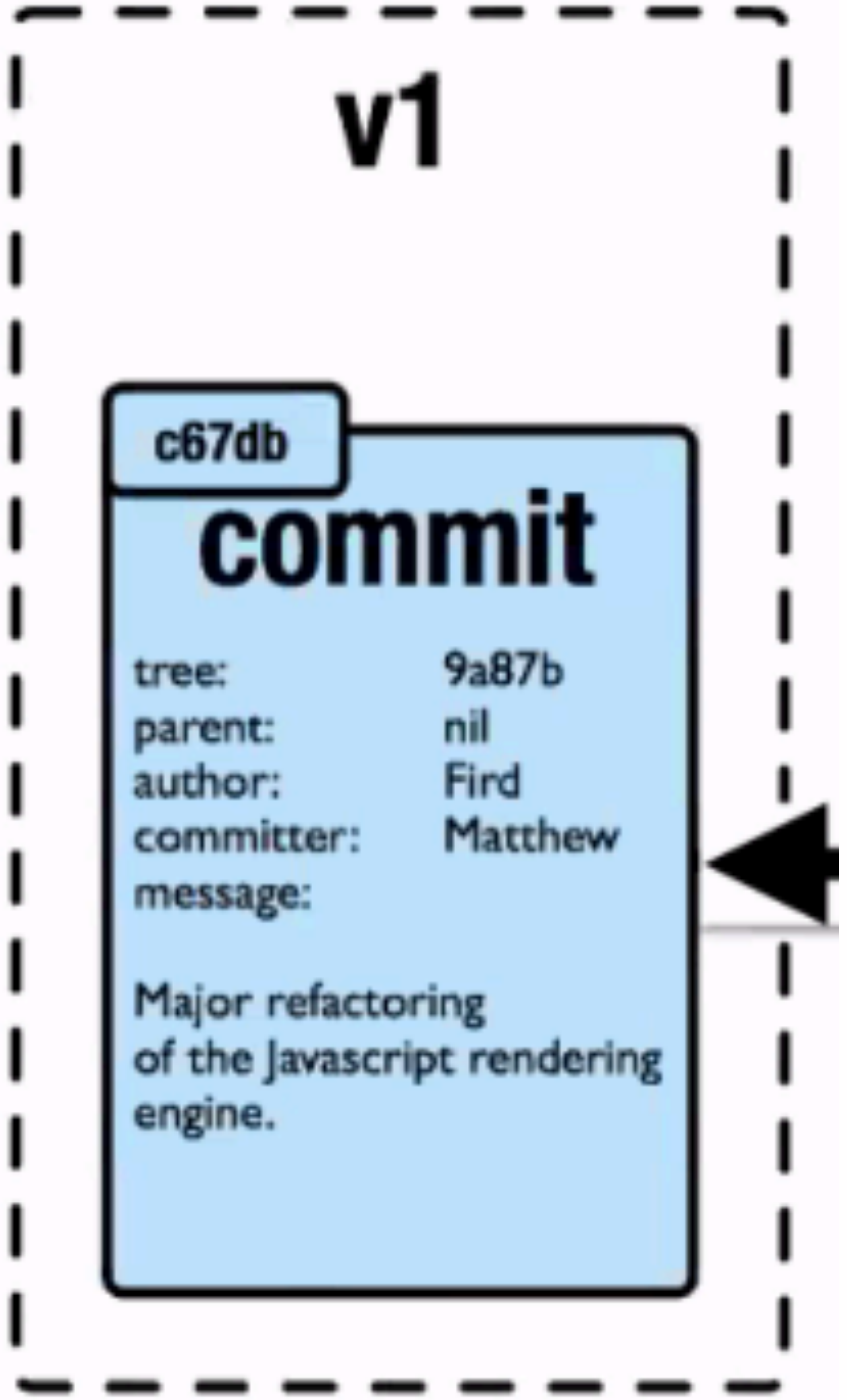
```
touch readme.md
git status
git add readme.md
git status
```

Understanding the state of your repository

```
git status
git diff
git diff --staged
```

Now..
a bit of theory

Inside a repository



v1

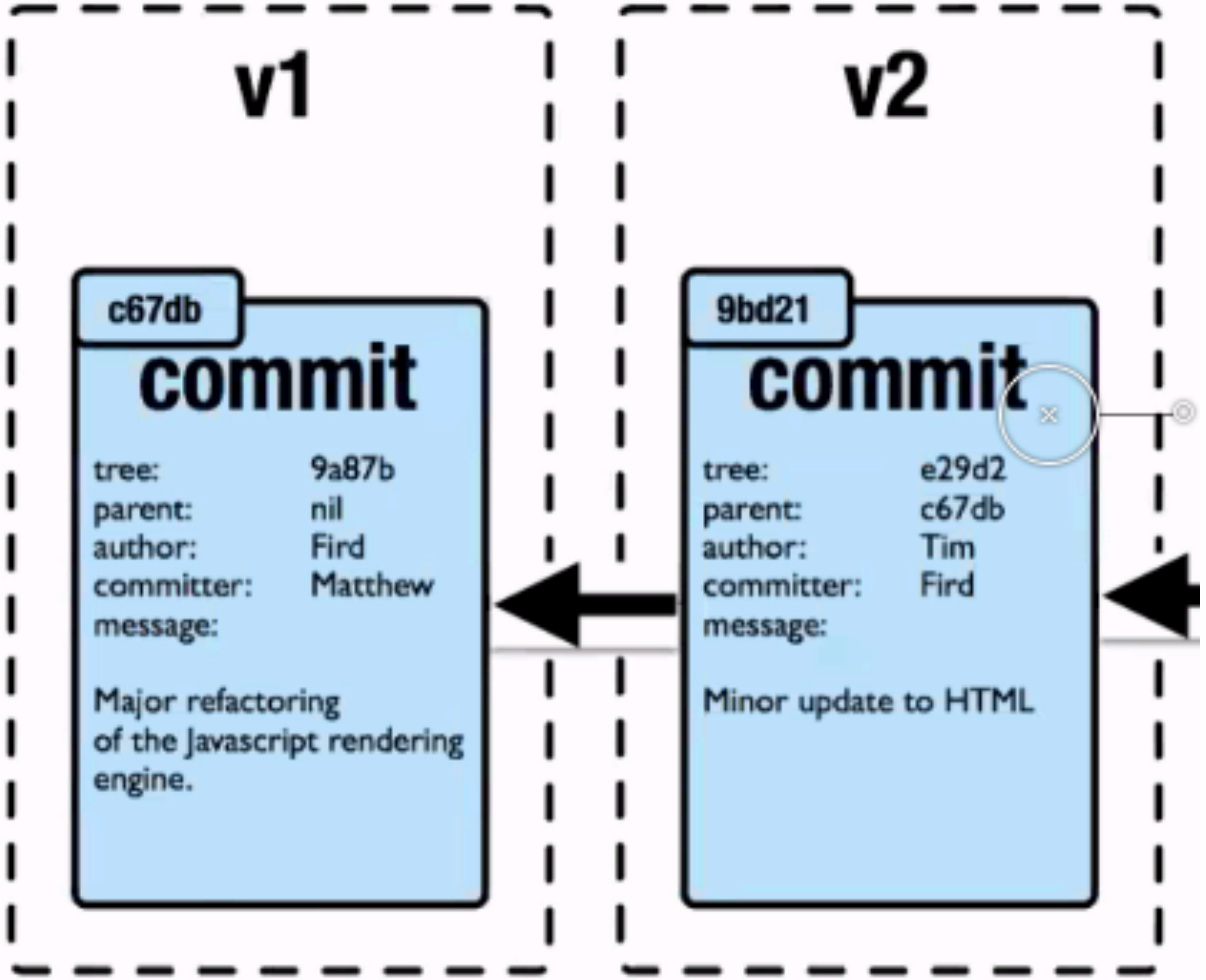
c67db

commit

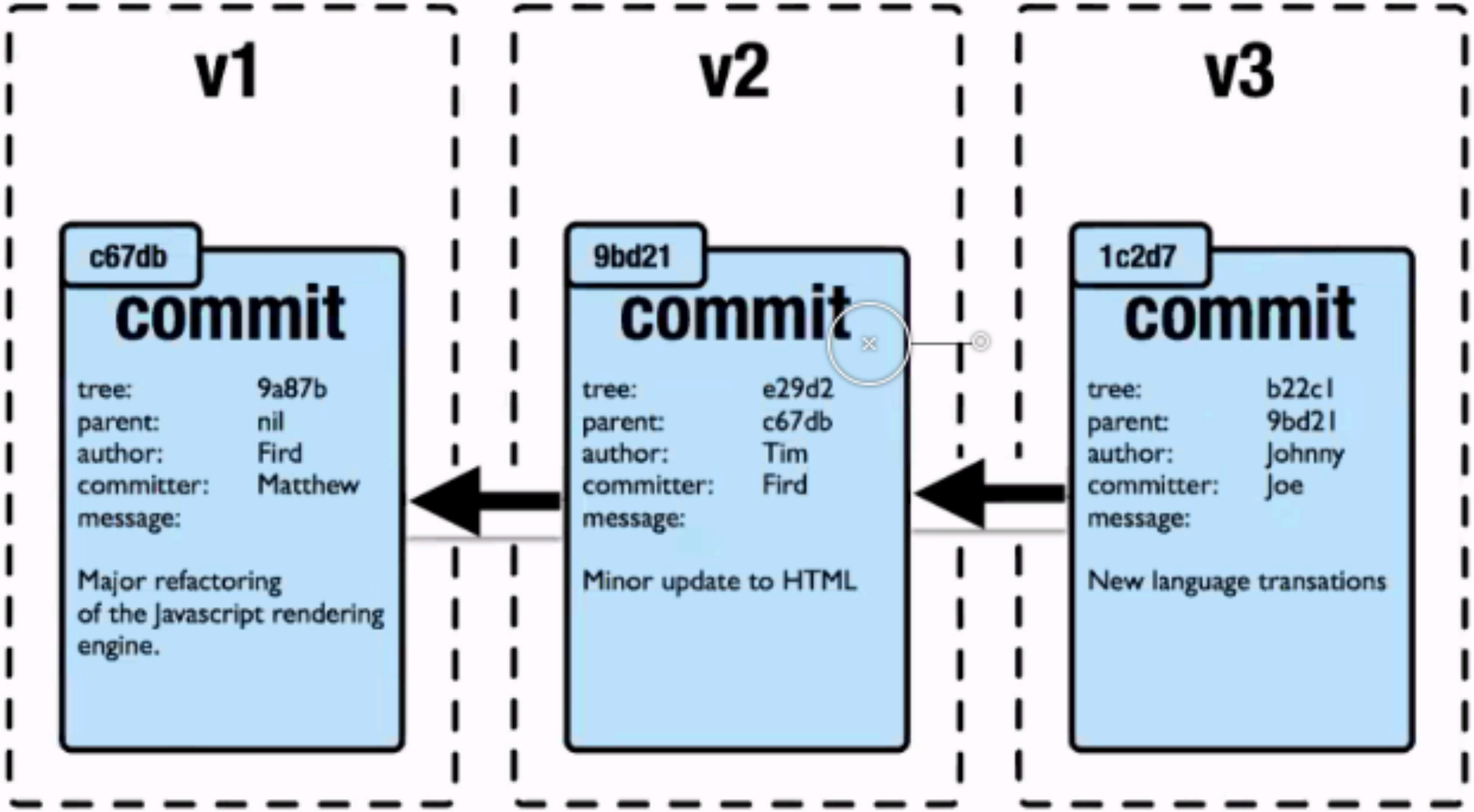
tree: 9a87b
parent: nil
author: Fird
committer: Matthew
message:

Major refactoring
of the Javascript rendering
engine.

Inside a repository



Inside a repository



v1

c67db

commit

tree: 9a87b
parent: nil
author: Fird
committer: Matthew
message:

Major refactoring of the Javascript rendering engine.

v2

9bd21

commit

tree: e29d2
parent: c67db
author: Tim
committer: Fird
message:

Minor update to HTML

v3

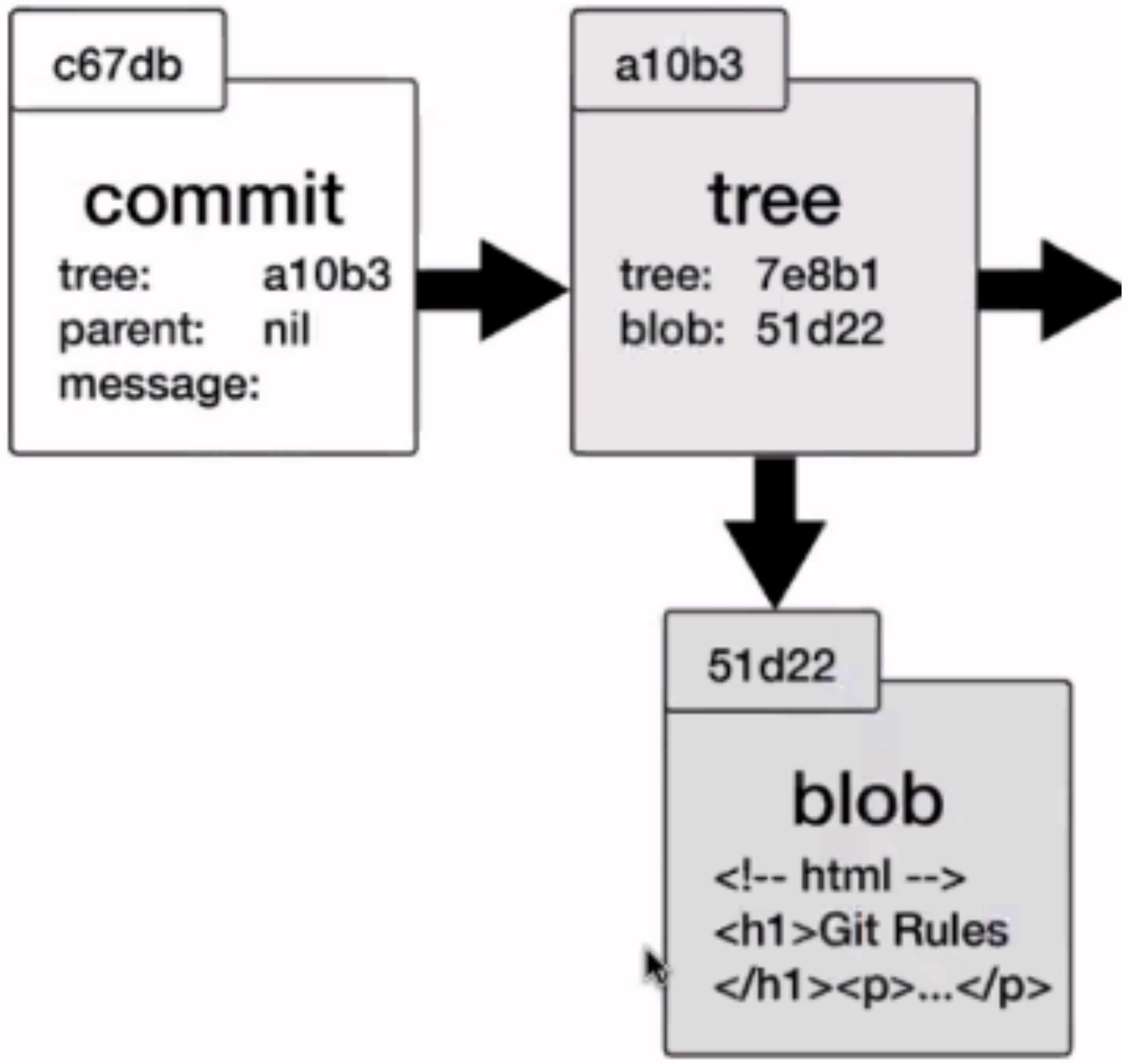
1c2d7

commit

tree: b22c1
parent: 9bd21
author: Johnny
committer: Joe
message:

New language transations

Inside the commit



c67db

commit

tree: a10b3
parent: nil
message:

a10b3

tree

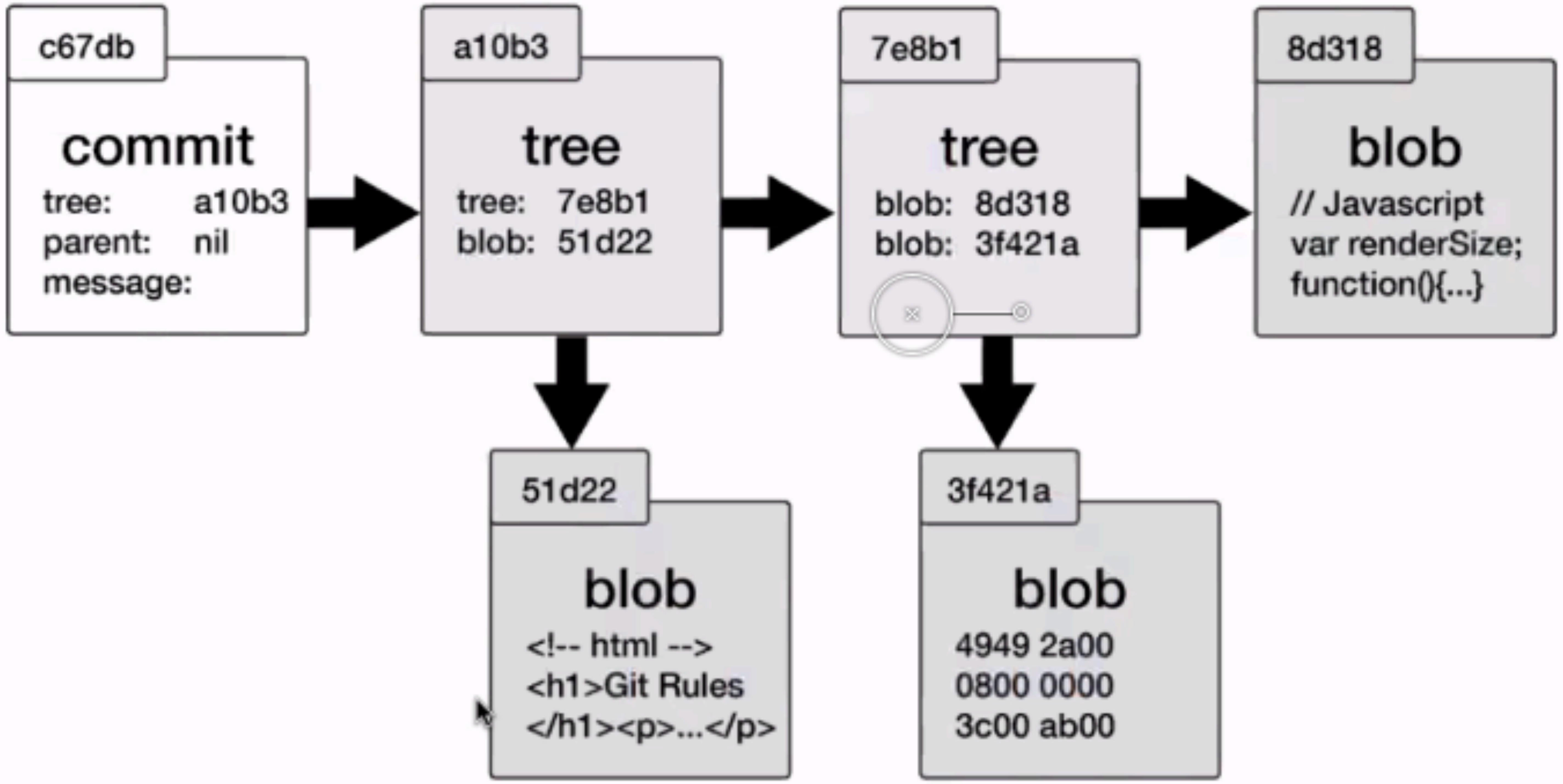
tree: 7e8b1
blob: 51d22

51d22

blob

```
<!-- html -->
<h1>Git Rules
</h1><p>...</p>
```

It’s a Merkle tree if that’s your thing



c67db

commit

tree: a10b3

parent: nil

message:

a10b3

tree

tree: 7e8b1

blob: 51d22

7e8b1

tree

blob: 8d318

blob: 3f421a

8d318

blob

```
// Javascript
var renderSize;
function(){...}
```

51d22

blob

```
<!-- html -->
<h1>Git Rules
</h1><p>...</p>
```

3f421a

blob

```
4949 2a00
0800 0000
3c00 ab00
```

GitHub Education

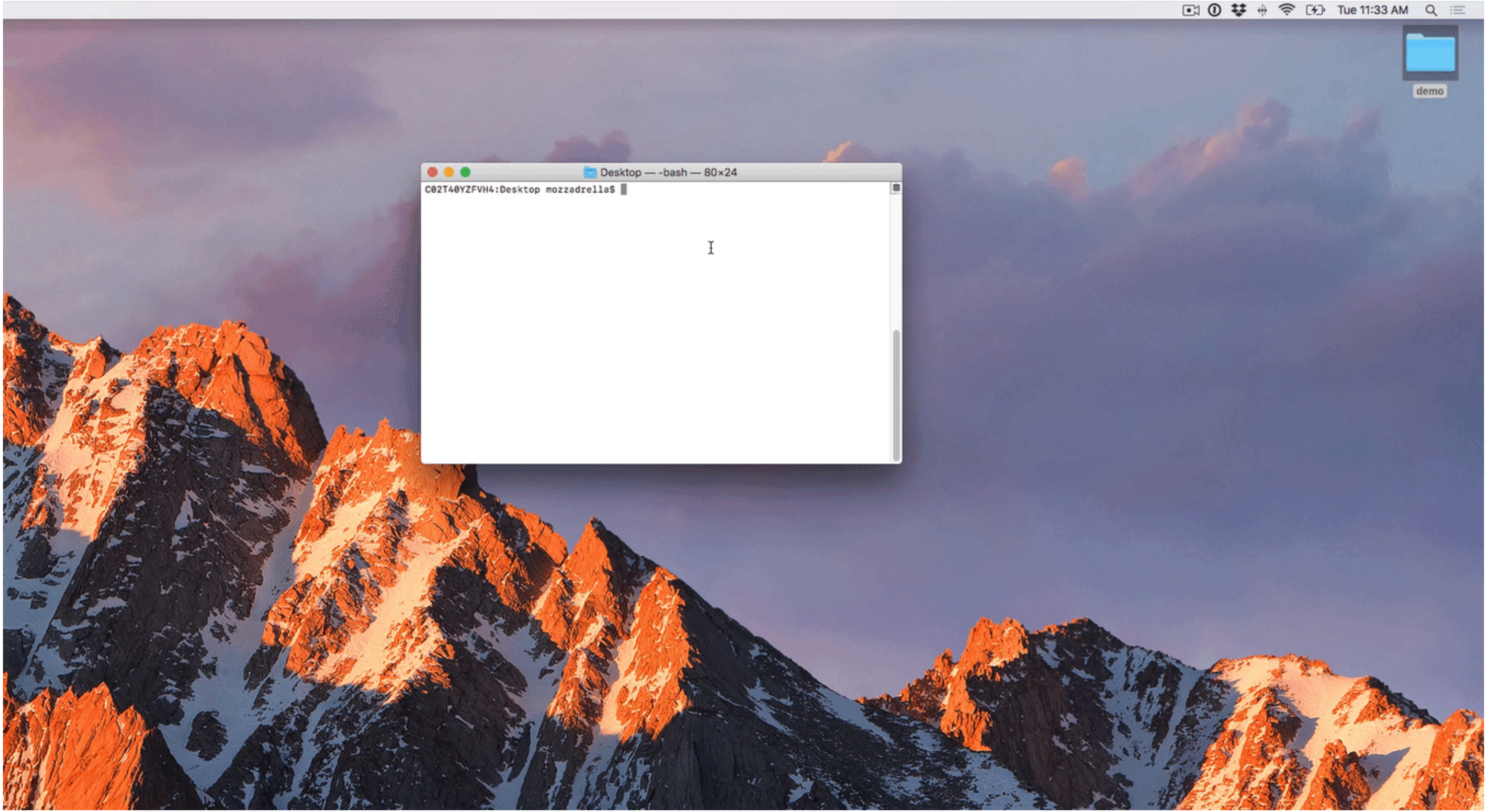
Module 2

Individual work

Network activity
Remotes
Fetch/push
Branches
Managing student repositories
GitHub Classroom
Permissions

Exercises

Network activity

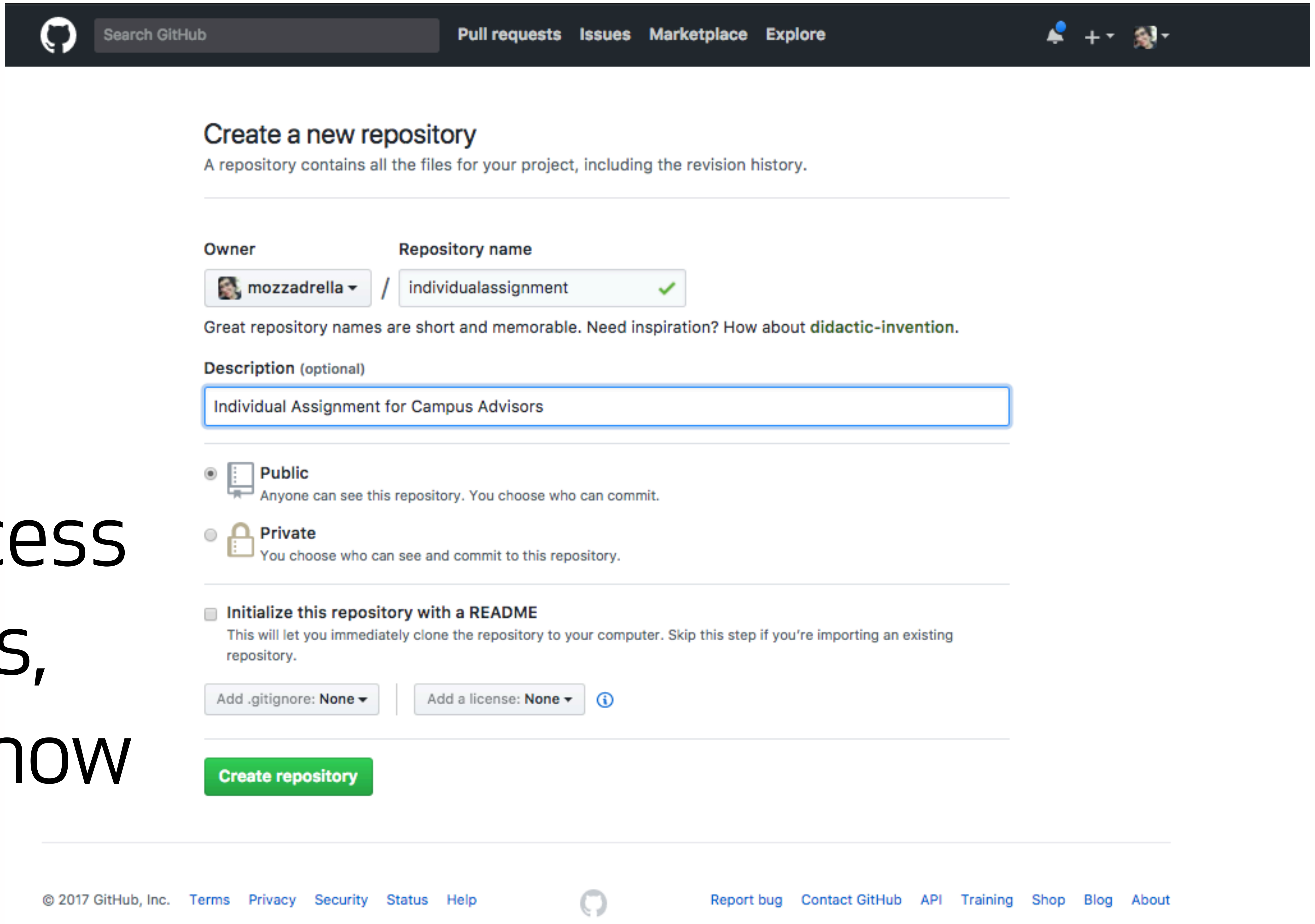


```
C02T40YZFVH4:Desktop mozzadrella$ 
```

Let’s create a repository in GitHub!

Let's set up a place to host your code

- A repository on GitHub!
- https://github.com/new
- As a teacher, you have access to free private repositories, but let's choose public for now



Create a new repository

A repository contains all the files for your project, including the revision history.

Owner: mozzadrella / Repository name: individualassignment ✓

Great repository names are short and memorable. Need inspiration? How about didactic-invention.

Description (optional): Individual Assignment for Campus Advisors

● Public
Anyone can see this repository. You choose who can commit.

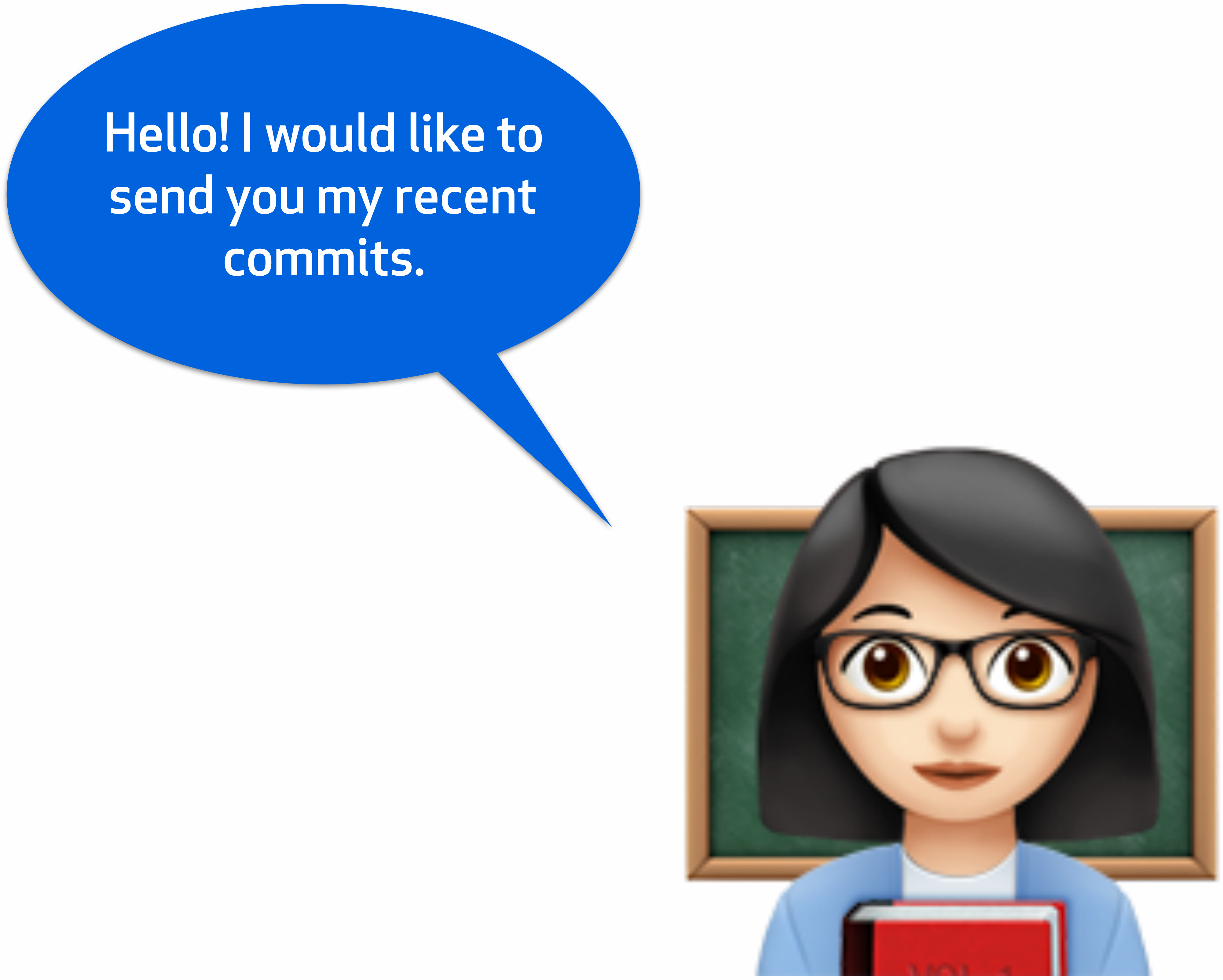
○ Private
You choose who can see and commit to this repository.

□ Initialize this repository with a README
This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

Add .gitignore: None | Add a license: None

Create repository
© 2017 GitHub, Inc. Terms Privacy Security Status Help
Report bug Contact GitHub API Training Shop Blog About

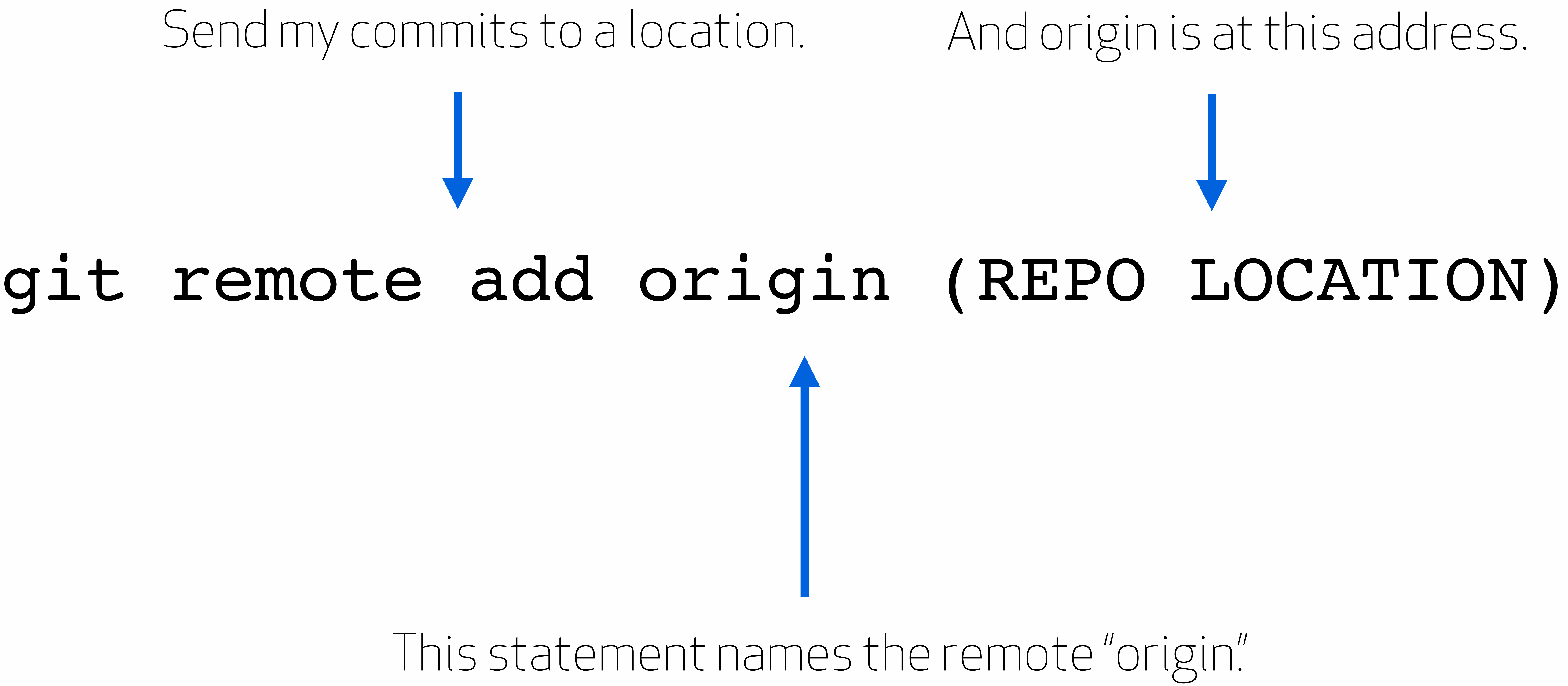
Adding a remote allows the transfer of your commits to another machine.



The bookmarked location is referred to as a “remote.”

```
git remote add origin (REPO LOCATION)
```

Add origin



Send my commits to a location.

And origin is at this address.

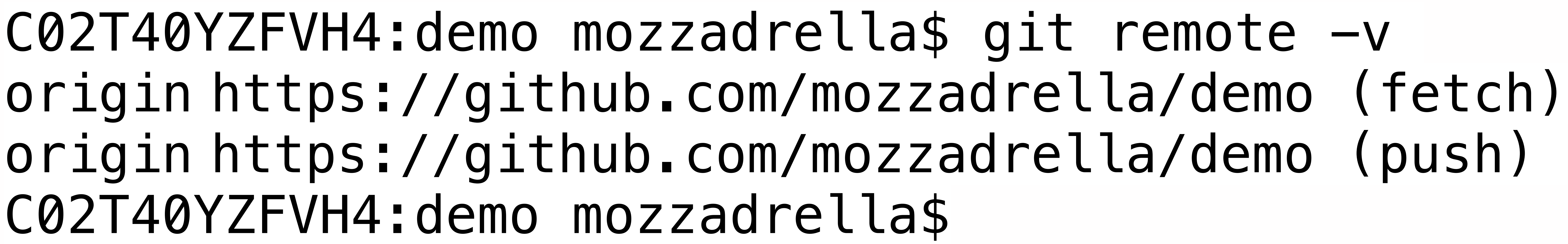
```
git remote add origin (REPO LOCATION)
```

This statement names the remote "origin."

Back to the terminal!

Check remotes with -v

Where am I sending and receiving commits from?



```
C02T40YZFVH4:demo mozzadrella$ git remote -v
origin	https://github.com/mozzadrella/demo (fetch)
origin	https://github.com/mozzadrella/demo (push)
C02T40YZFVH4:demo mozzadrella$
```

Now: editing in the GitHub UI (not recommended)

Tue 11:36 AM
Create a New Repository
Vanessa
GitHub, Inc. [US] | https://github.com/new
remove-ar-inheritan…
Switch to GitHub:lab
Rails 3.2.22.5
Search GitHub
Pull requests
Issues
Marketplace
Explore
Desktop/individual
jobs.md
al Assignments: Learning Objectives
LF
UTF-8
GitHub Markdown
master
0 files
individual
demo

Create a new repository

A repository contains all the files for your project, including the revision history.

Owner: mozzadrella ▾ / Repository name: |

Great repository names are short and memorable. Need inspiration? How about didactic-invention.

Description (optional)

Individual Assignment for Campus Advisors

● **Public**
Anyone can see this repository. You choose who can commit.

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

□ **Initialize this repository with a README**
This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

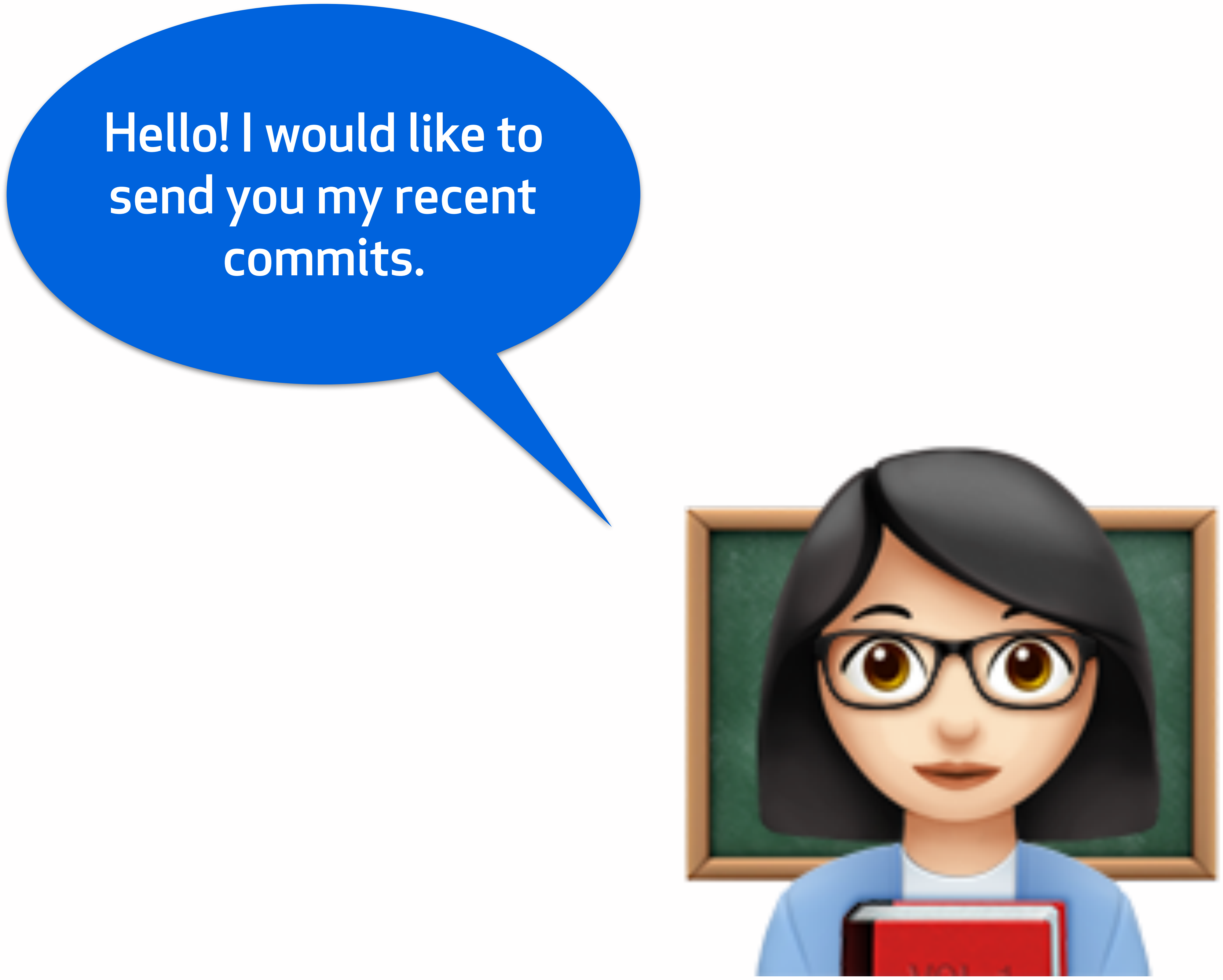
Add .gitignore: **None** ▾ | Add a license: **None** ▾

Create repository

Fetch

I'd like the commits I don't already have, please.

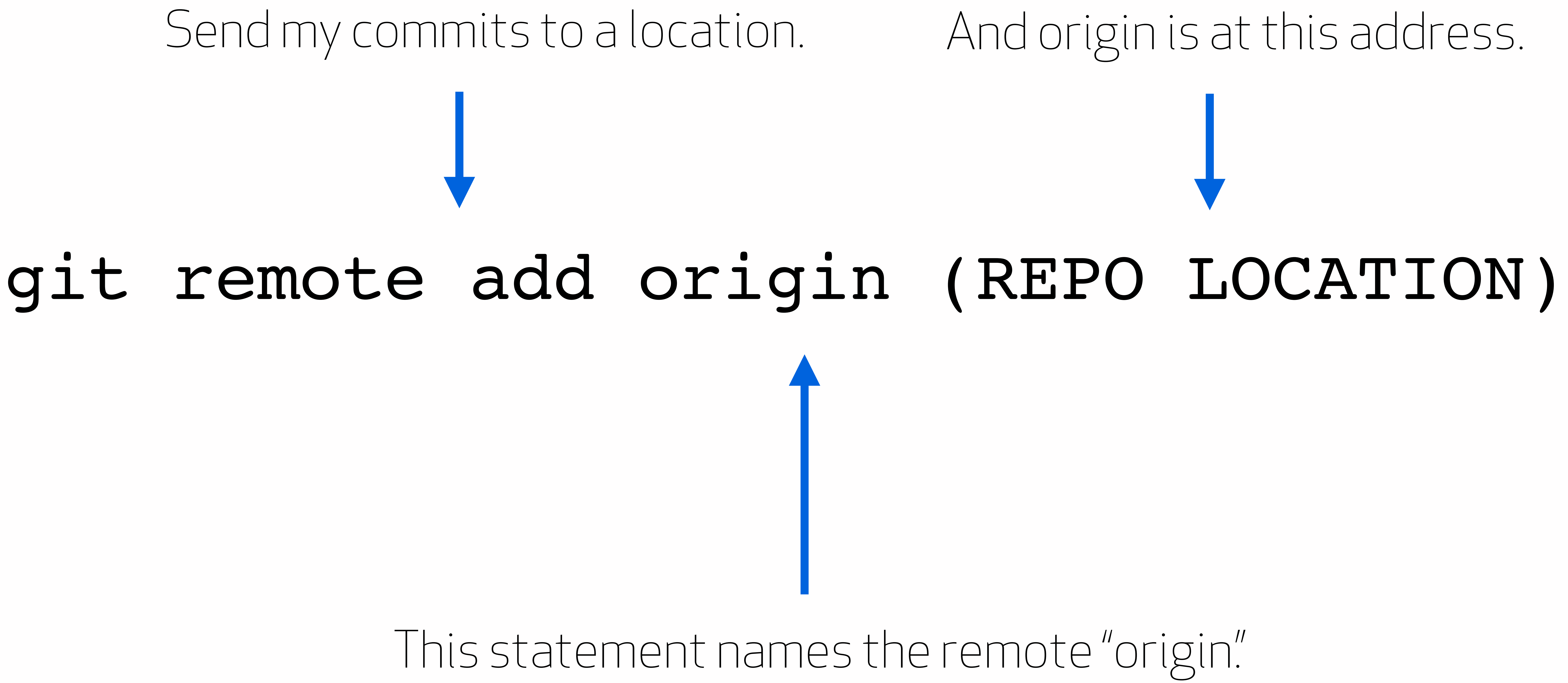
Adding a remote allows the transfer of your commits to another machine.



The bookmarked location is referred to as a "remote."

```
git remote add origin (REPO LOCATION)
```

Add origin



Send my commits to a location.

And origin is at this address.

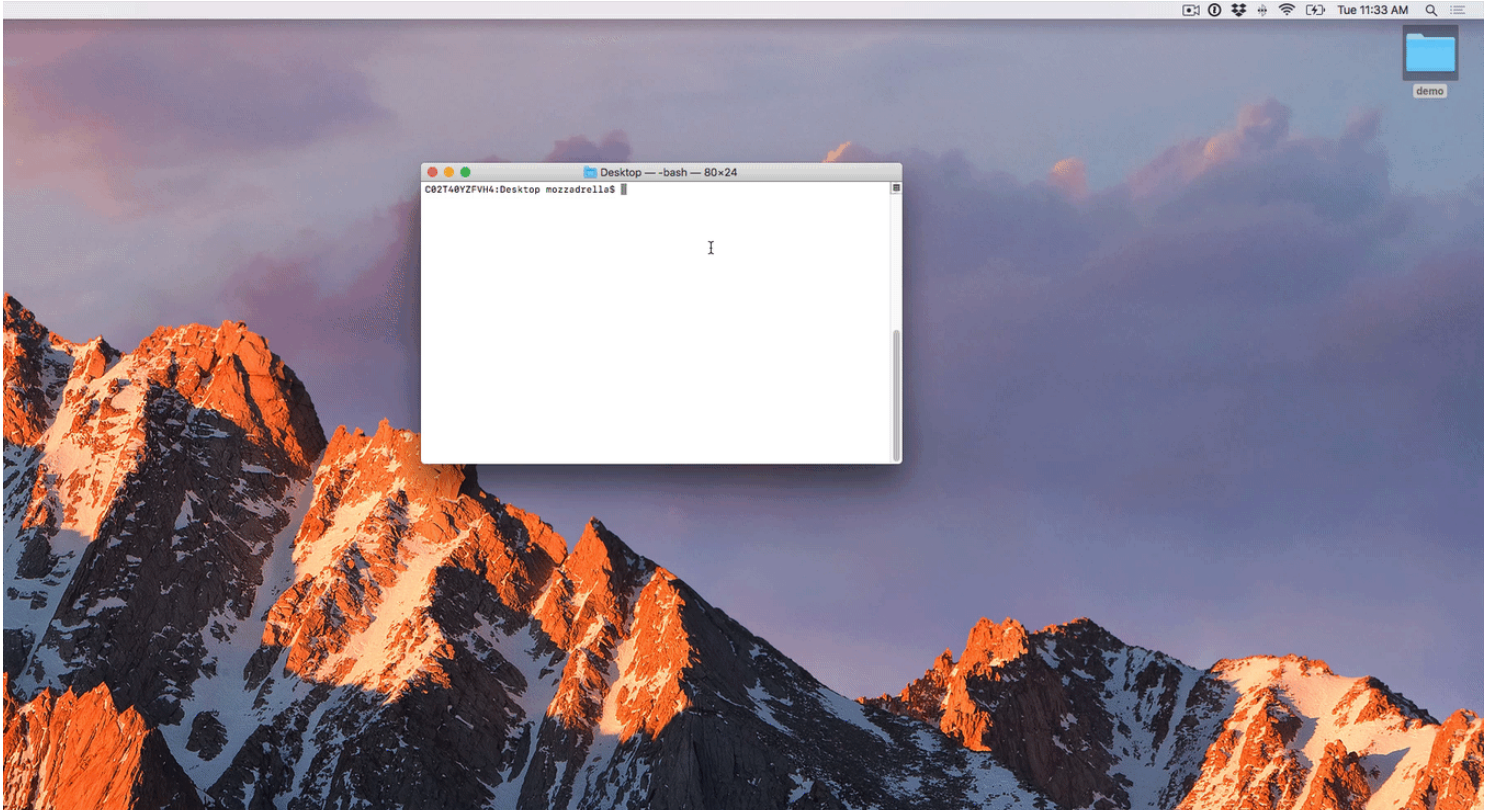
```
git remote add origin (REPO LOCATION)
```

This statement names the remote "origin."

Back to the terminal!

Pushing to a remote

Network activity



```
C02T40YZFVH4:Desktop mozzadrella$ 
```

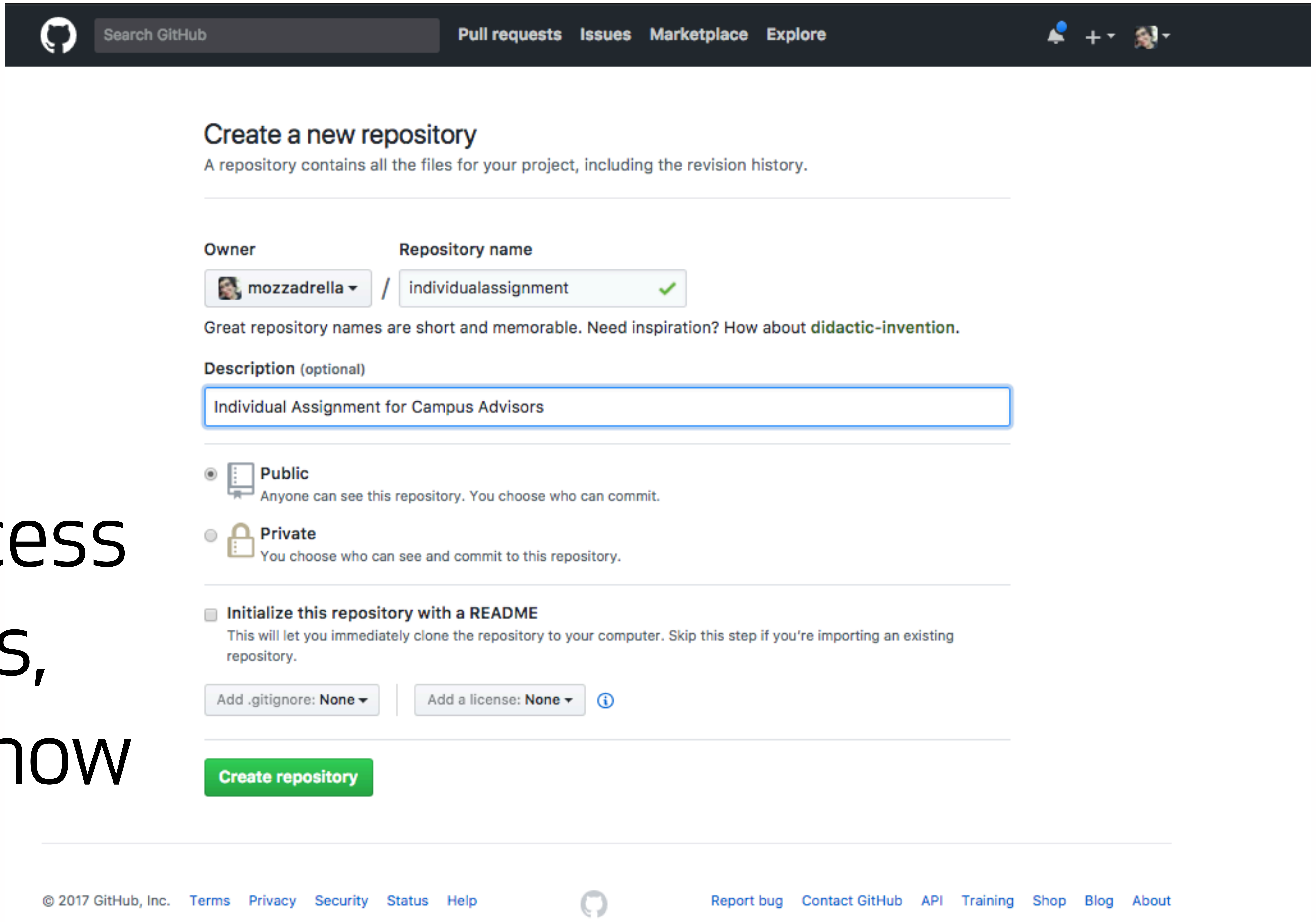
GitHub

- Hosts your repositories
- Track student progress
- Social features to enable collaboration

Let's create a repository in GitHub!

Let’s set up a place to host your code

- A repository on GitHub!
- https://github.com/new
- As a teacher, you have access to free private repositories, but let’s choose public for now



Search GitHub   Pull requests   Issues   Marketplace   Explore

Create a new repository

A repository contains all the files for your project, including the revision history.

Owner: mozzadrella ▾ / Repository name: individualassignment ✓

Great repository names are short and memorable. Need inspiration? How about didactic-invention.

Description (optional): Individual Assignment for Campus Advisors

● Public
Anyone can see this repository. You choose who can commit.

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

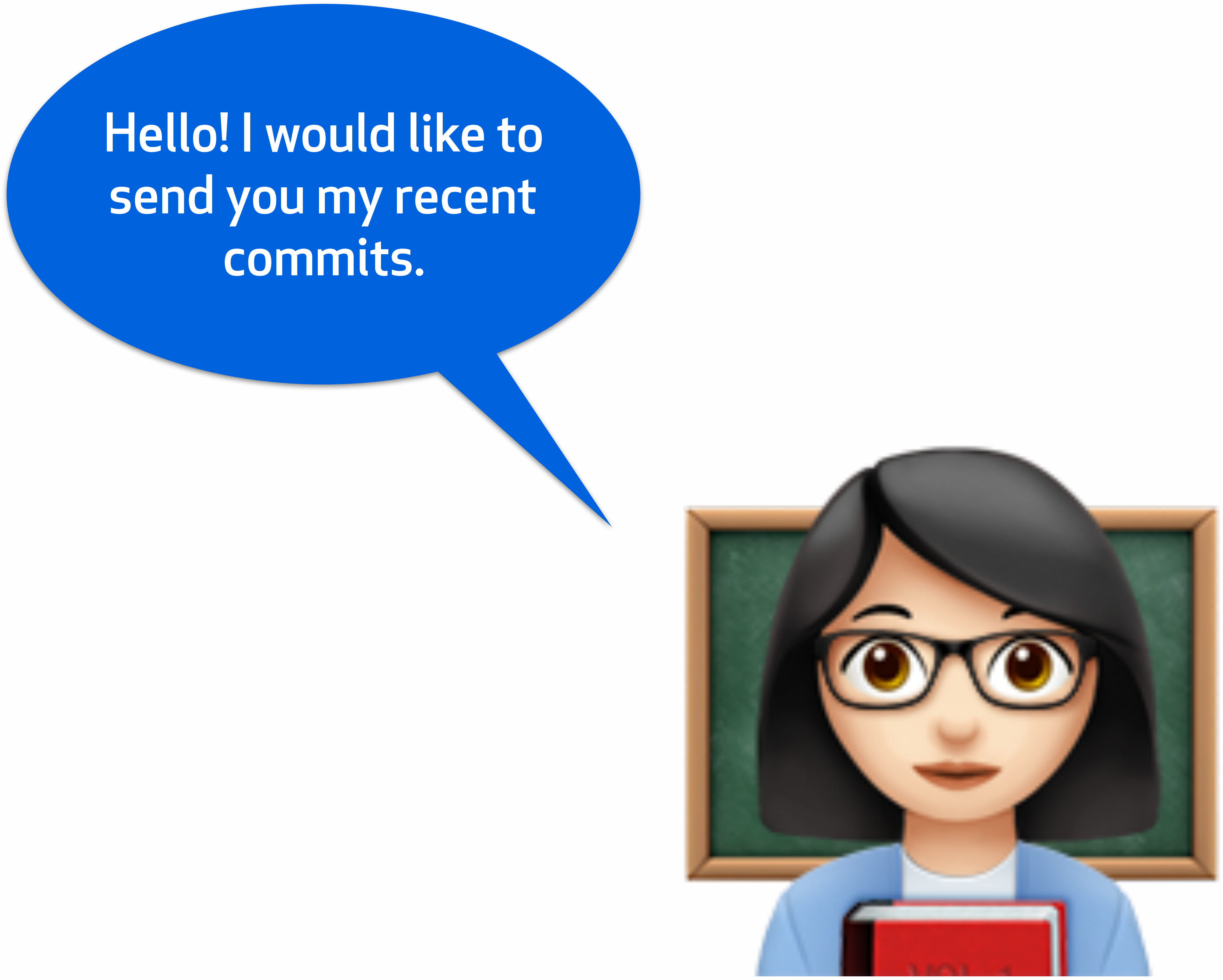
□ Initialize this repository with a README
This will let you immediately clone the repository to your computer. Skip this step if you’re importing an existing repository.

Add .gitignore: None ▾ | Add a license: None ▾ ⓘ

Create repository

© 2017 GitHub, Inc. Terms Privacy Security Status Help   Report bug Contact GitHub API Training Shop Blog About

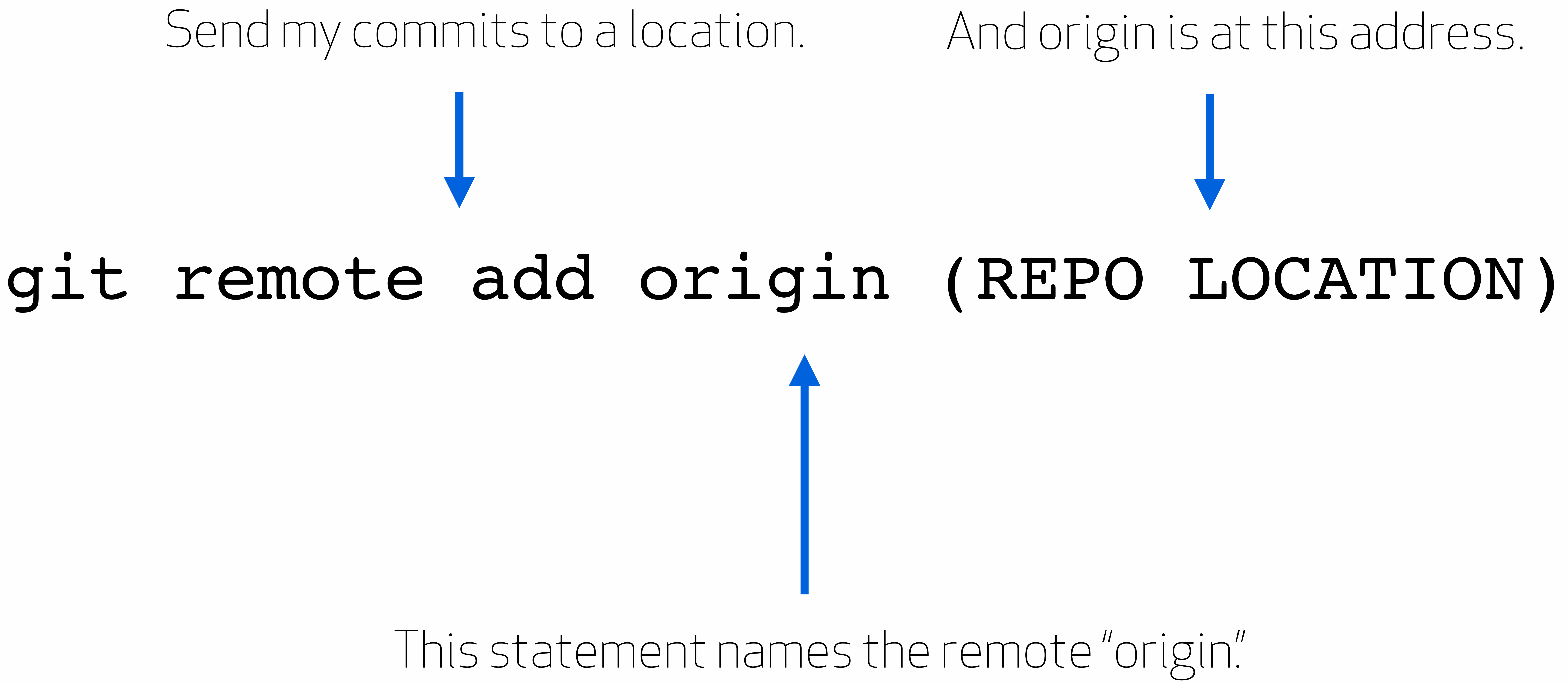
Adding a remote allows the transfer of your commits to another machine.



The bookmarked location is referred to as a “remote.”

```
git remote add origin (REPO LOCATION)
```

Add origin



Send my commits to a location.

And origin is at this address.

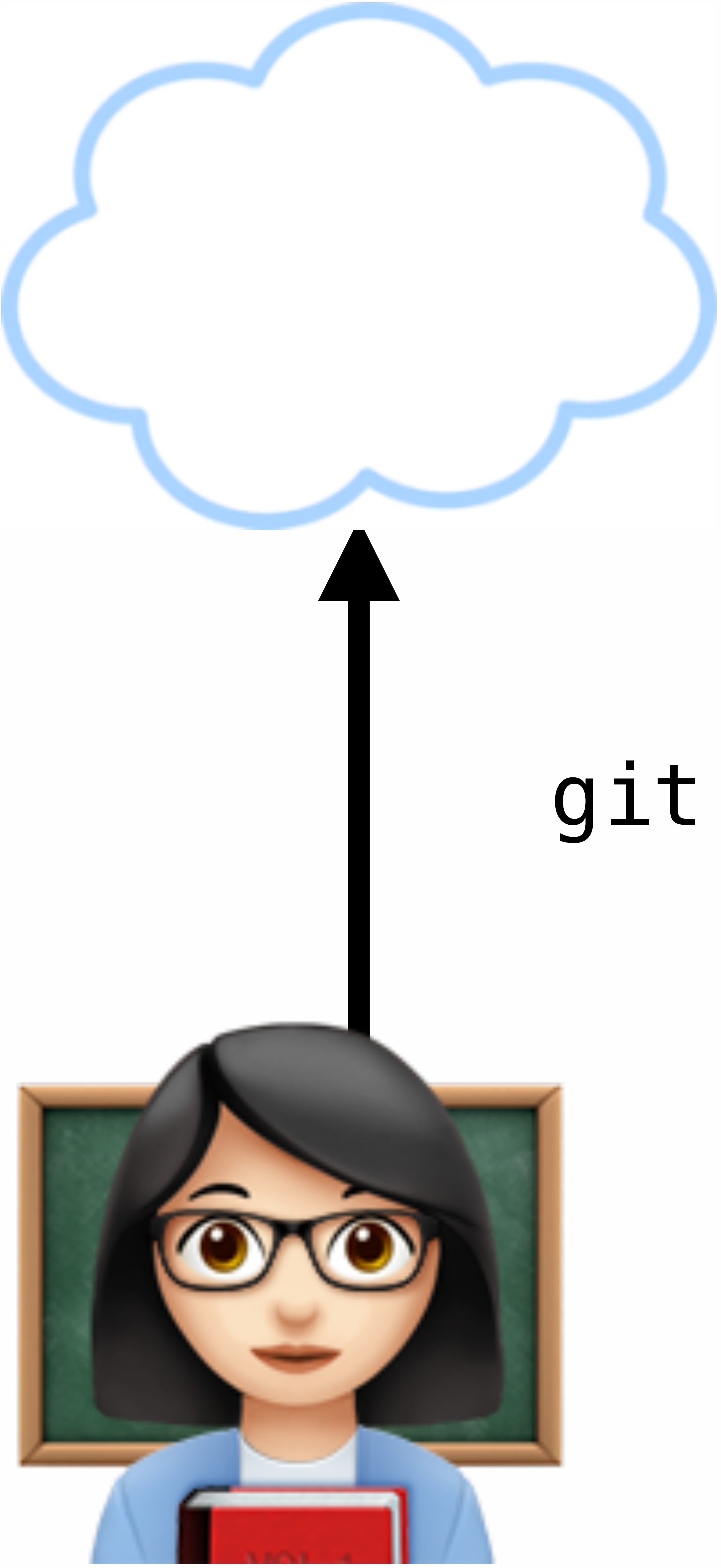
```
git remote add origin (REPO LOCATION)
```

This statement names the remote "origin."

Back to the terminal!

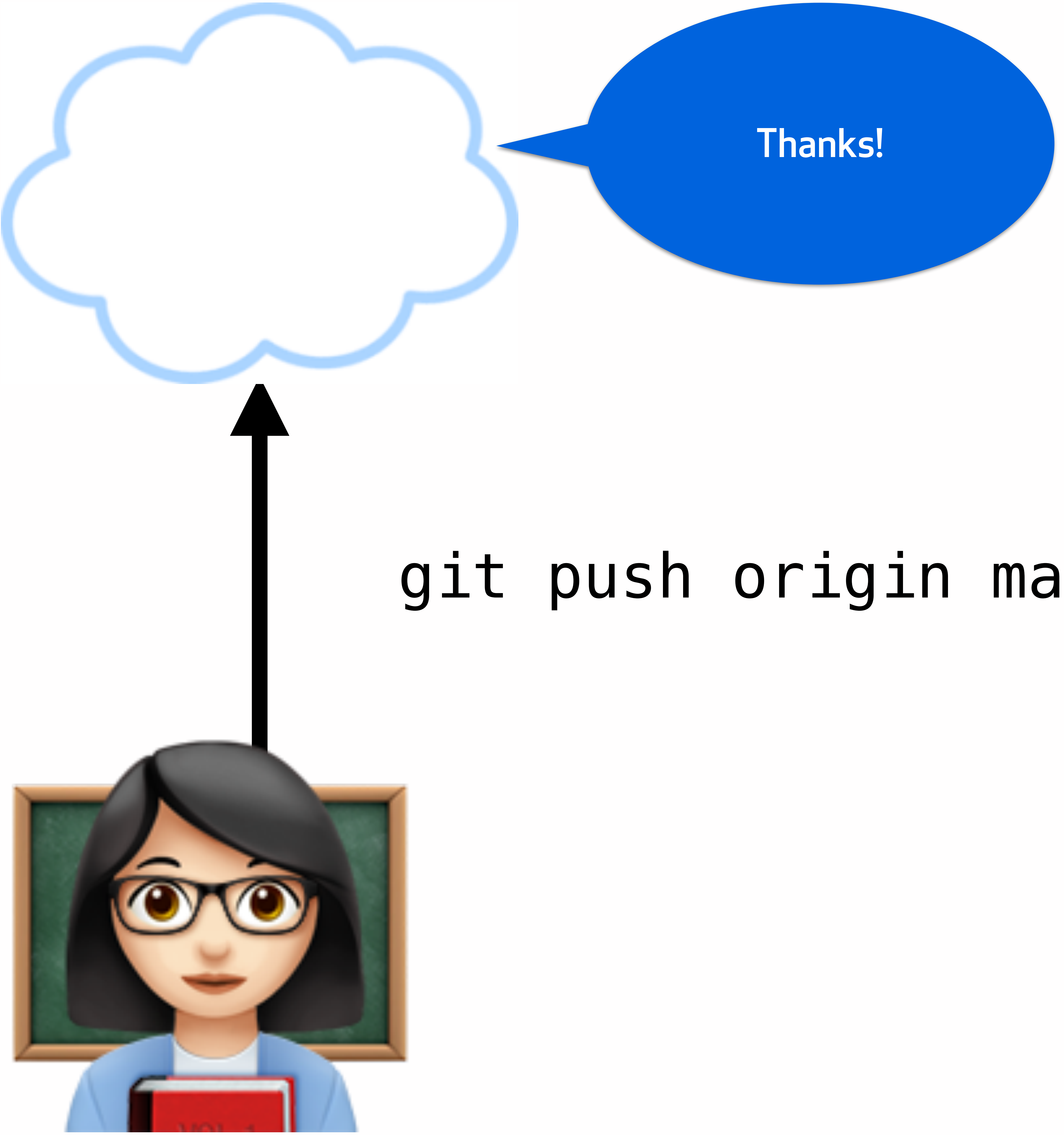
Pushing to a remote

How do you get your commits up to the remote?



```
git push origin master
```

How do you get your commits up to the remote?



```
git push origin master
```

Link remote with local.

-u is short for --set-upstream

```
git push –u origin master
```

Useful because you can just write “git push” when you want to push future commits.

Types of remote addresses

- HTTP/HTTPS urls
- Git protocol over SSH and use the file path
- GitHub Desktop client (clone repository and open in Desktop)

Check remotes with -v

Where am I sending and receiving commits from?

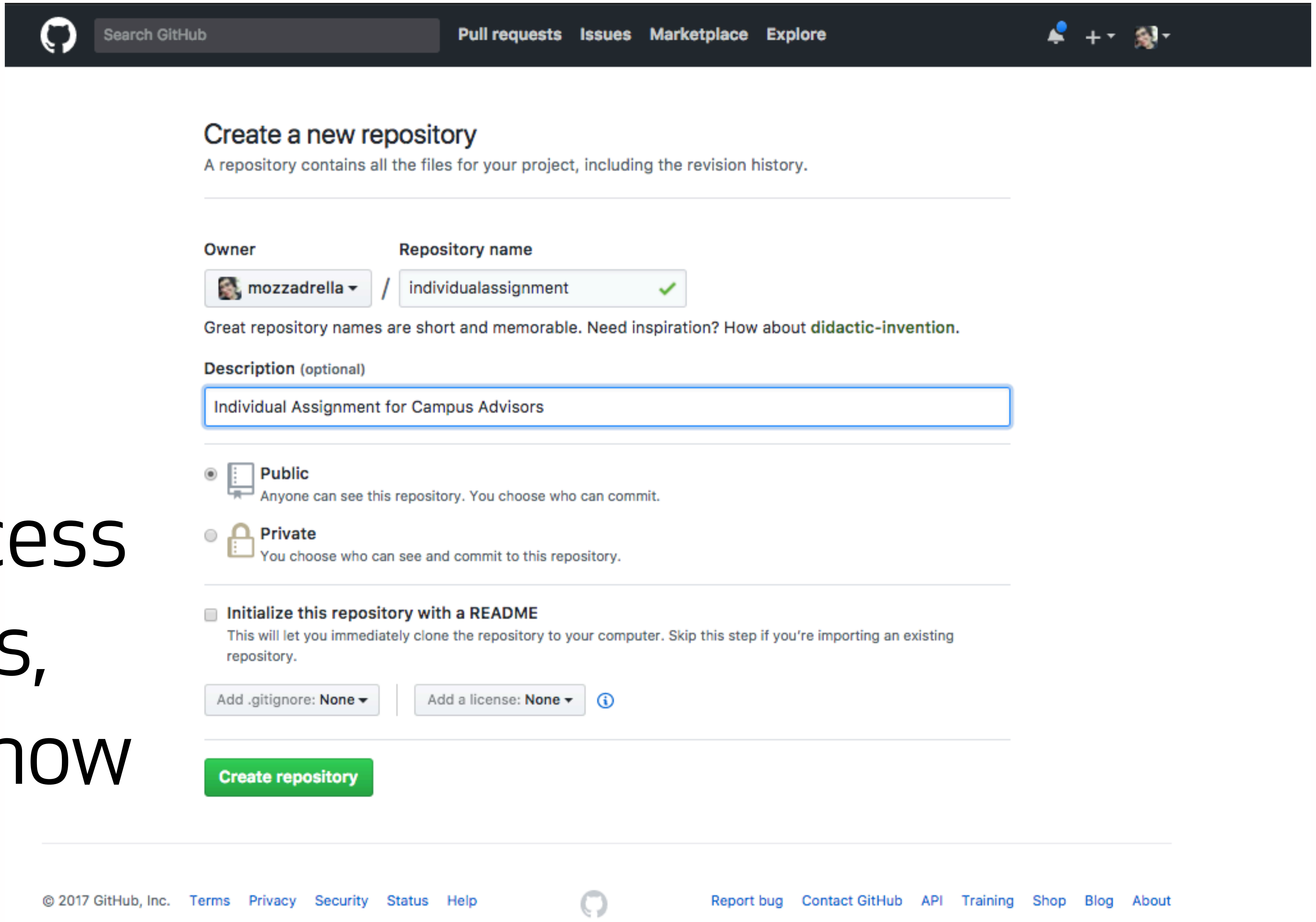
```
C02T40YZFVH4:demo mozzadrella$ git remote -v
origin https://github.com/mozzadrella/demo (fetch)
origin https://github.com/mozzadrella/demo (push)
C02T40YZFVH4:demo mozzadrella$
```

Fetch

I'd like the commits I don't already have, please.

Let's set up a place to host your code

- A repository on GitHub!
- https://github.com/new
- As a teacher, you have access to free private repositories, but let's choose public for now



Create a new repository

A repository contains all the files for your project, including the revision history.

Owner: mozzadrella / Repository name: individualassignment ✓

Great repository names are short and memorable. Need inspiration? How about didactic-invention.

Description (optional): Individual Assignment for Campus Advisors

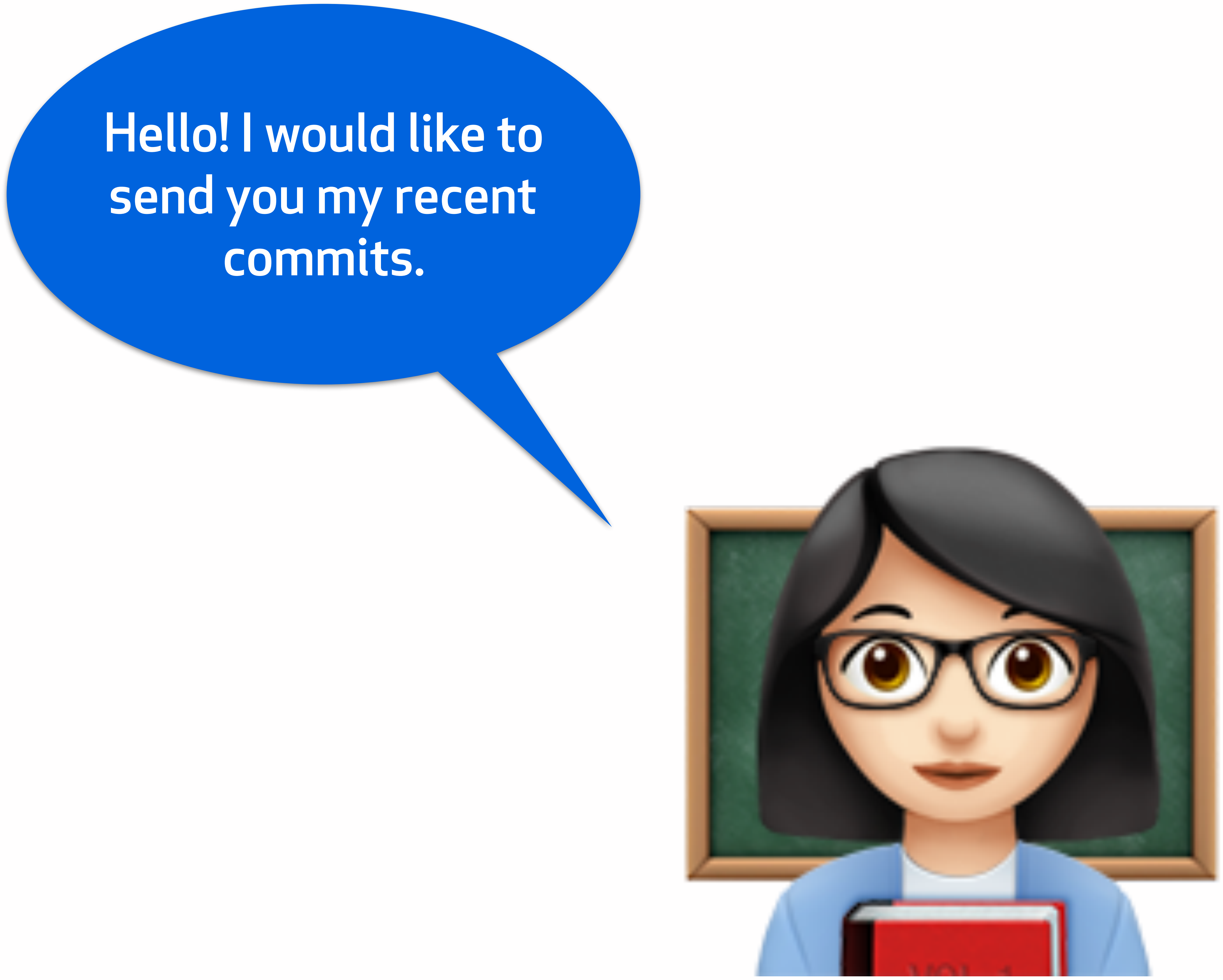
● Public
Anyone can see this repository. You choose who can commit.

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

☐ Initialize this repository with a README
This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

Add .gitignore: None | Add a license: None

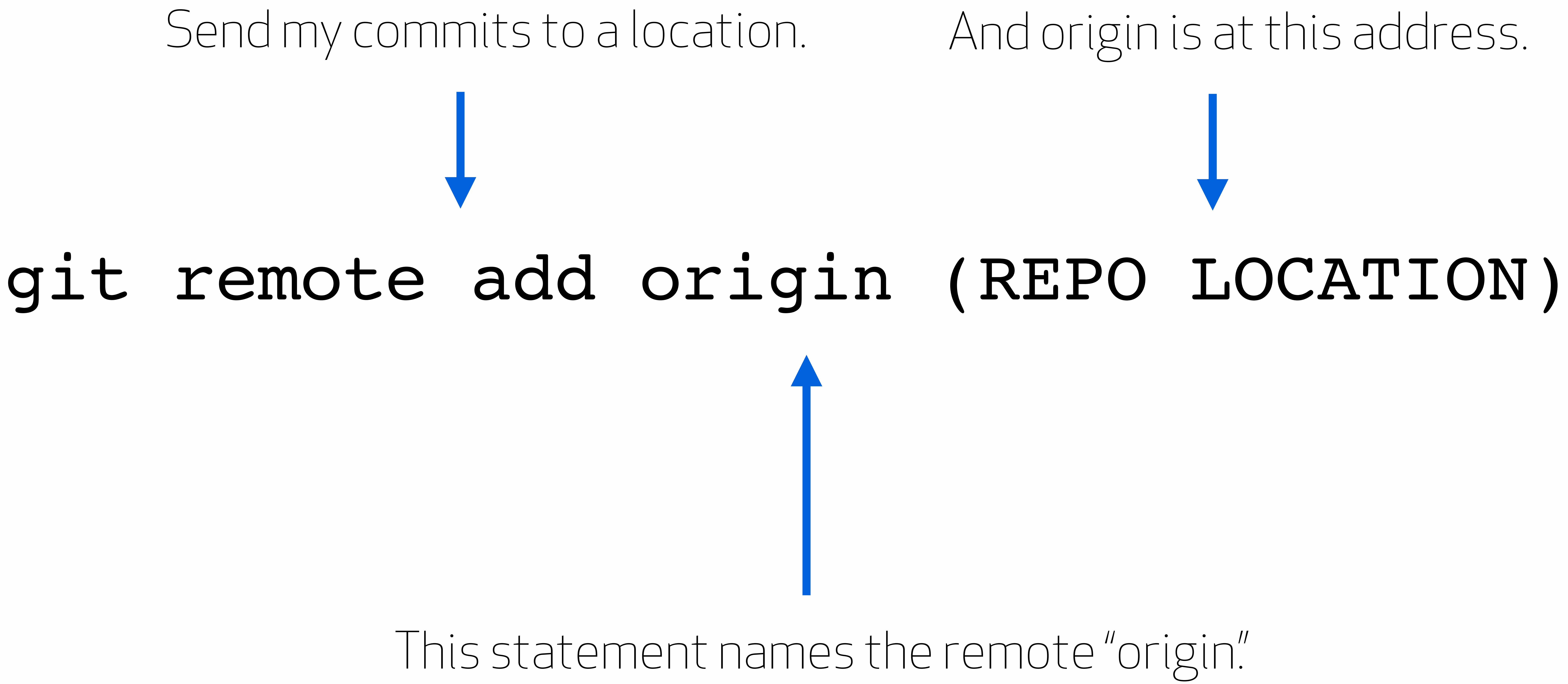
Adding a remote allows the transfer of your commits to another machine.



The bookmarked location is referred to as a “remote.”

```
git remote add origin (REPO LOCATION)
```

Add origin



Send my commits to a location.

And origin is at this address.

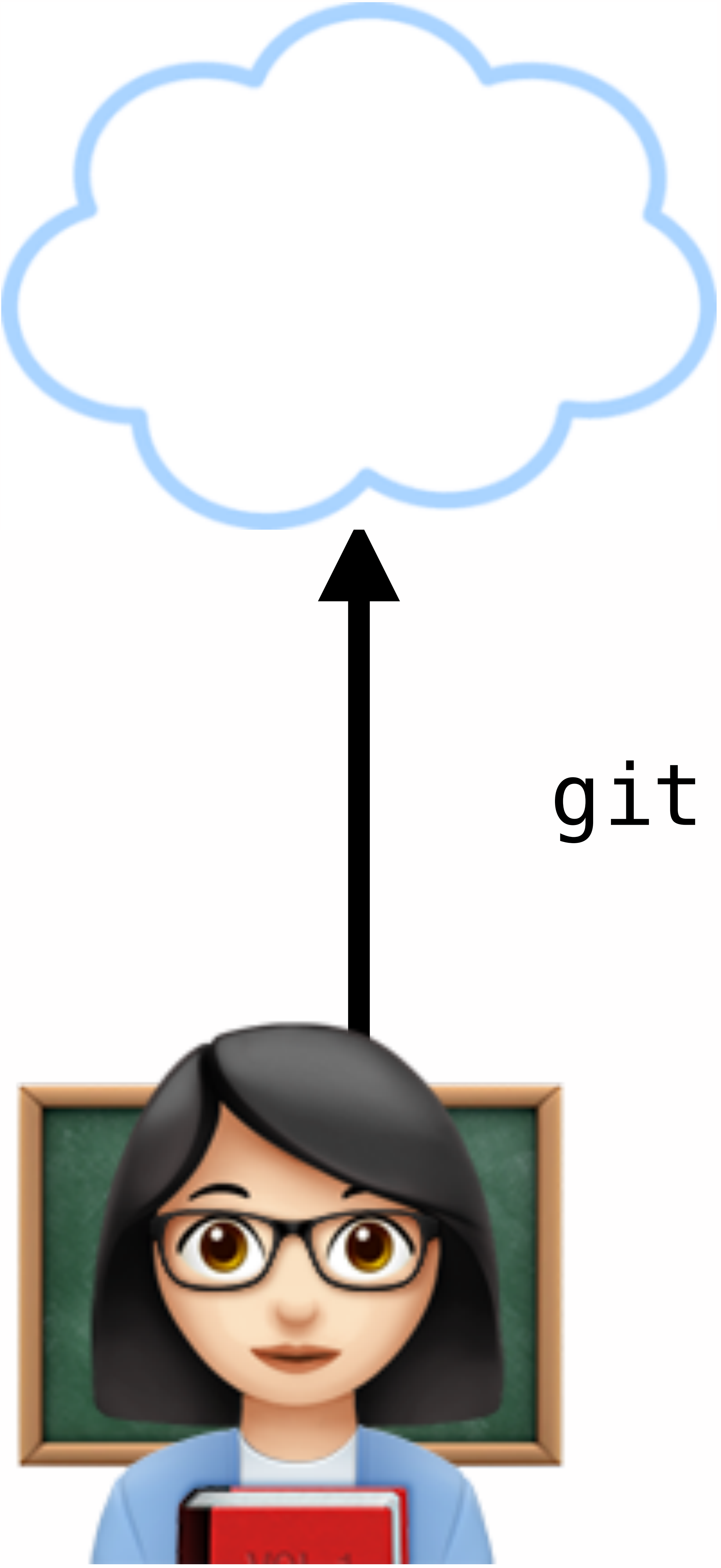
```
git remote add origin (REPO LOCATION)
```

This statement names the remote “origin.”

Back to the terminal!

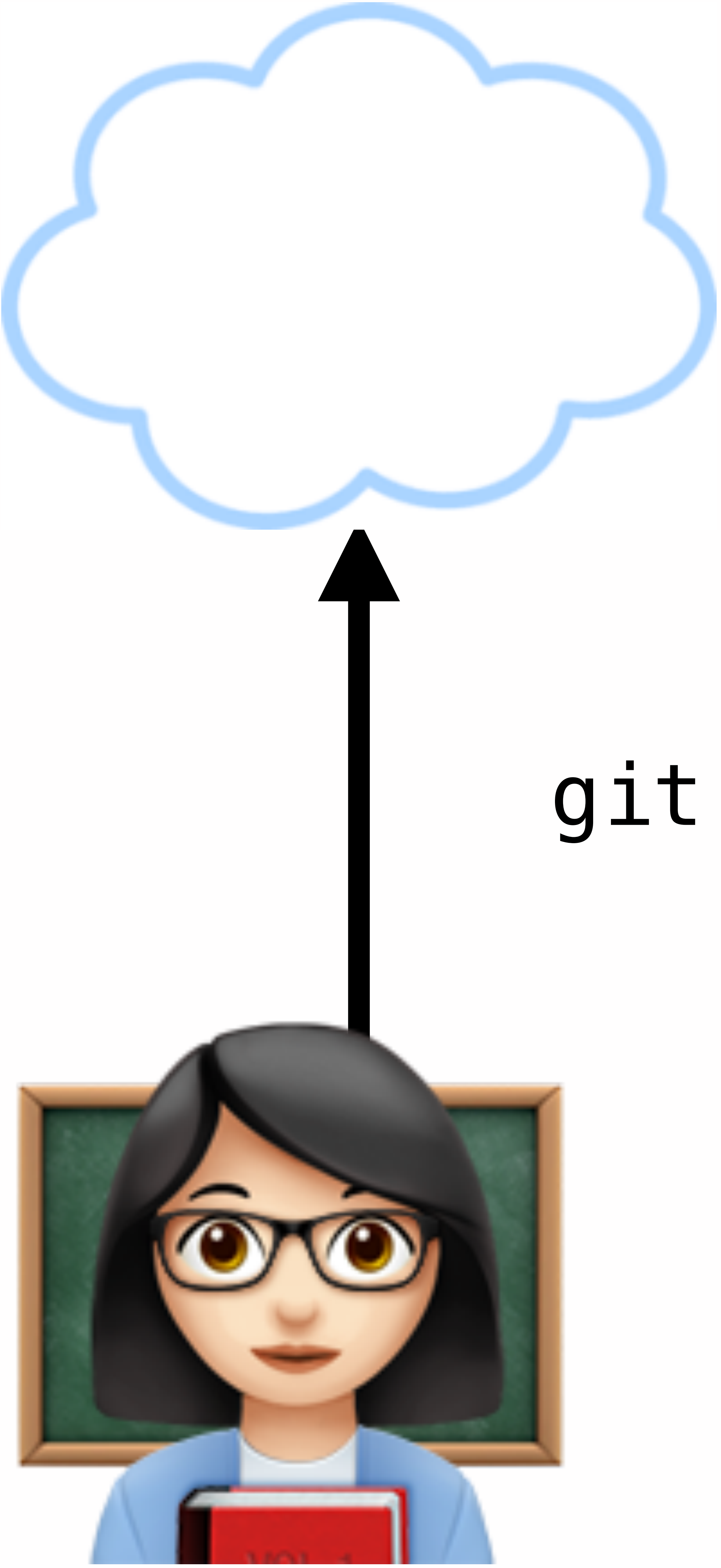
Pushing to a remote

How do you get your commits up to the remote?



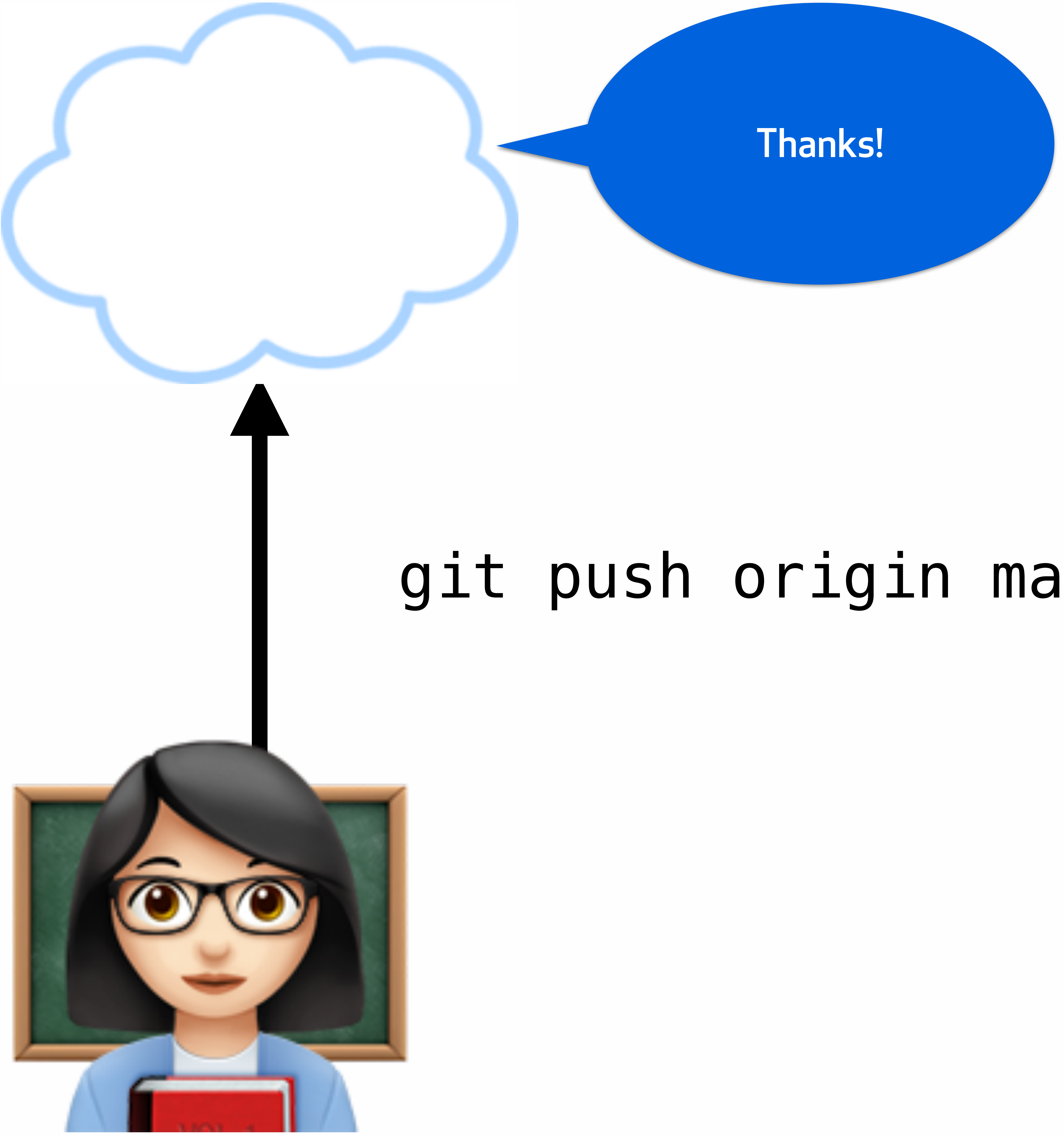
```
git push origin master
```

How do you get your commits up to the remote?



```
git push origin master
```

How do you get your commits up to the remote?



```
git push origin master
```

Link remote with local.

-u is short for --set-upstream

```
git push –u origin master
```

Useful because you can just write “git push” when you want to push future commits.

Types of remote addresses

- HTTP/HTTPS urls
- Git protocol over SSH and use the file path
- GitHub Desktop client (clone repository and open in Desktop)

Now: editing in the GitHub UI (not recommended)

Tue 11:36 AM
Create a New Repository
Vanessa
GitHub, Inc. [US] | https://github.com/new
remove-ar-inheritan…
Switch to GitHub:lab
Rails 3.2.22.5
Search GitHub
Pull requests
Issues
Marketplace
Explore
individual
demo
Desktop/individual
jobs.md
al Assignments: Learning Objectives
LF
UTF-8
GitHub Markdown
master
0 files

Create a new repository

A repository contains all the files for your project, including the revision history.

Owner: mozzadrella / Repository name:

Great repository names are short and memorable. Need inspiration? How about didactic-invention.

Description (optional): Individual Assignment for Campus Advisors

● Public
Anyone can see this repository. You choose who can commit.

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

□ Initialize this repository with a README
This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

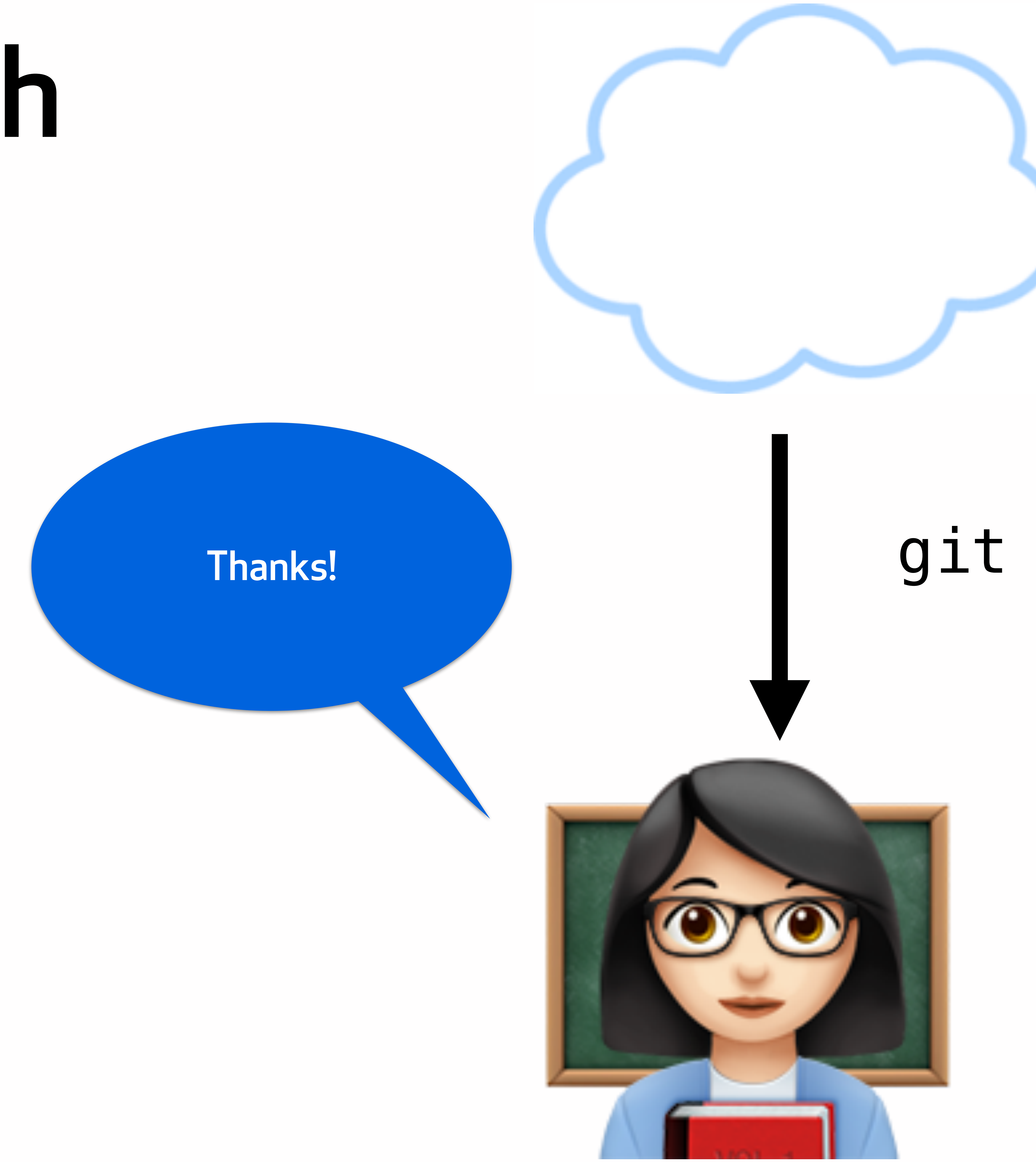
Add .gitignore: None

Add a license: None

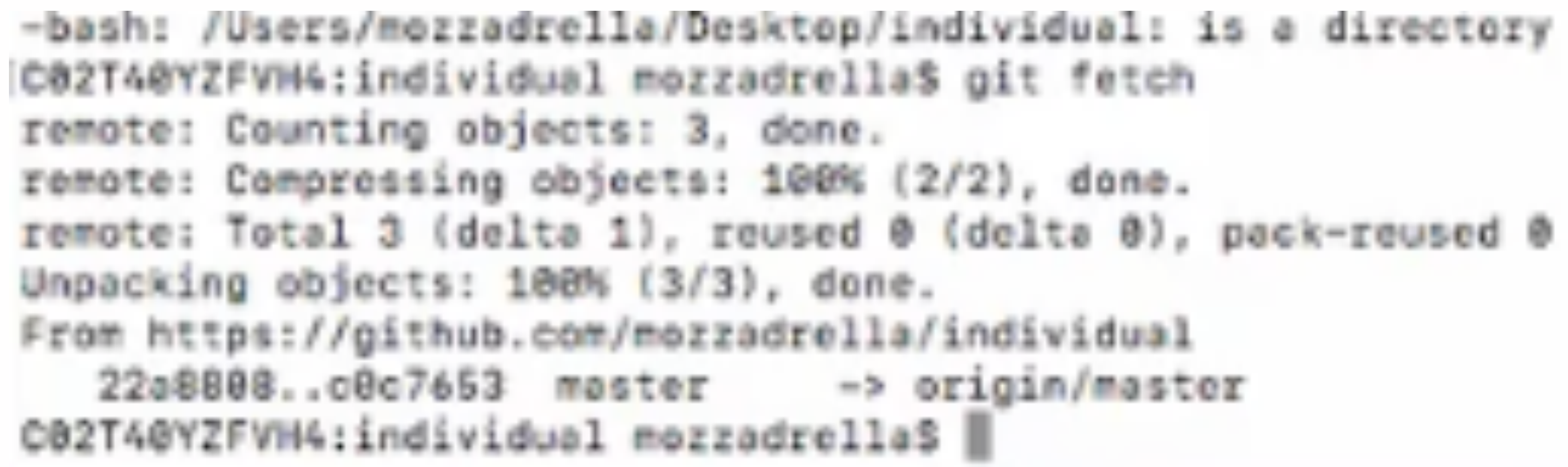
Fetch

I'd like the commits I don't already have, please.

Fetch

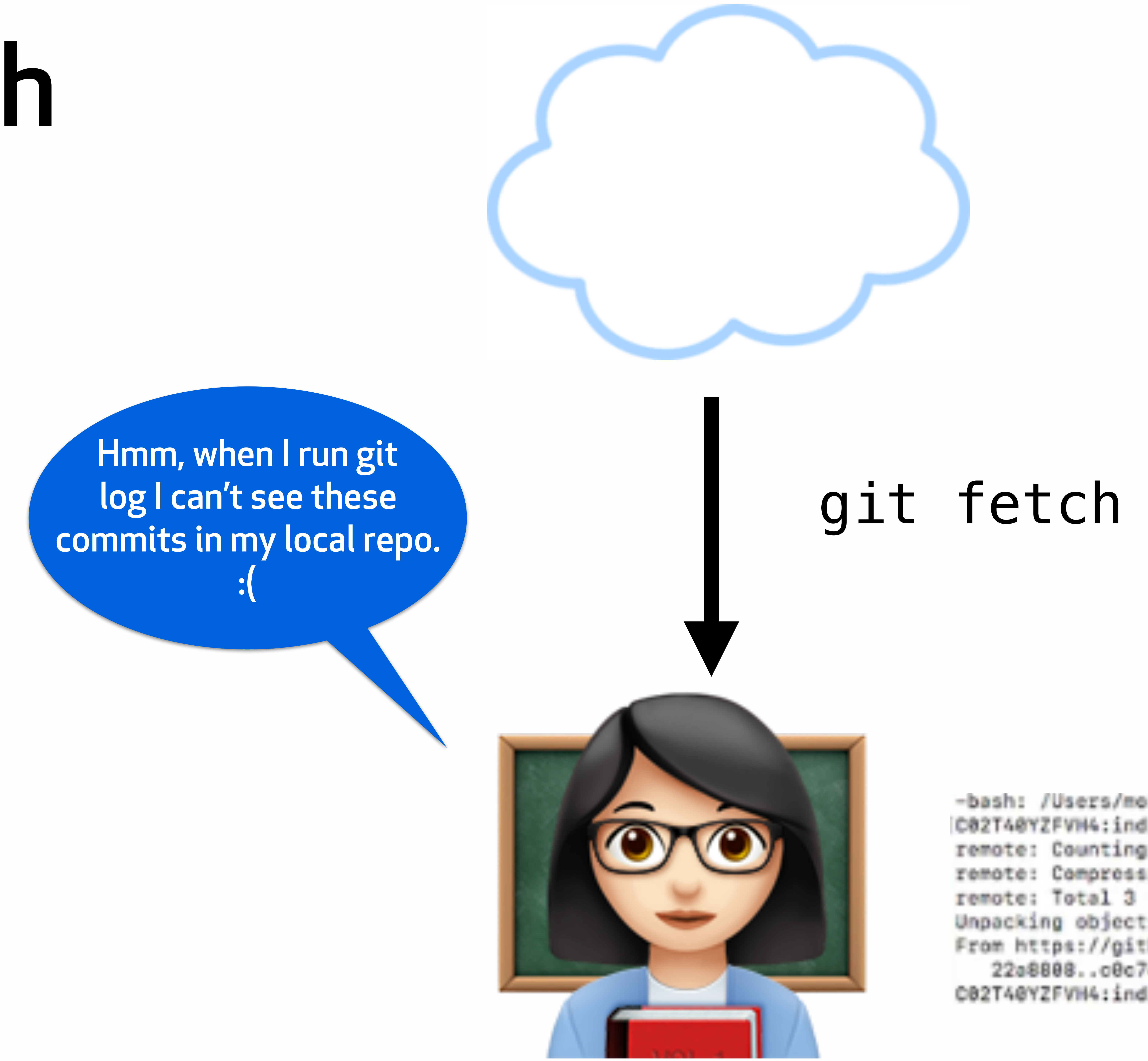


git fetch

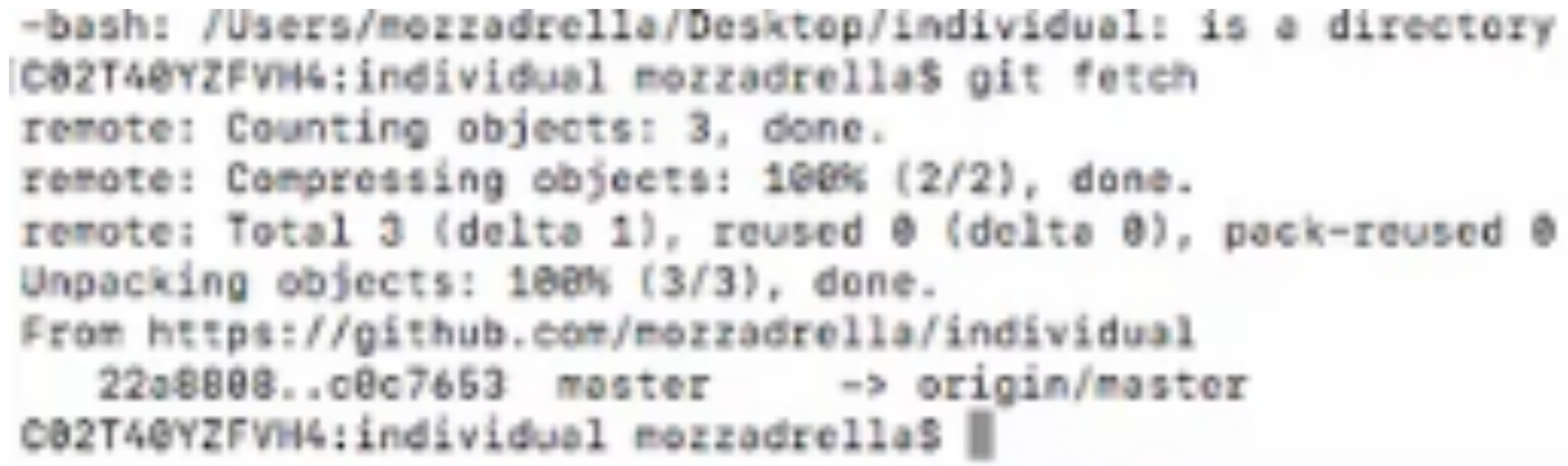


```
-bash: /Users/mozzadrella/Desktop/individual: is a directory
C02T40YZFVH4:individual mozzadrella$ git fetch
remote: Counting objects: 3, done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), done.
From https://github.com/mozzadrella/individual
   22a8808..c0c7653  master     -> origin/master
C02T40YZFVH4:individual mozzadrella$ 
```

Fetch

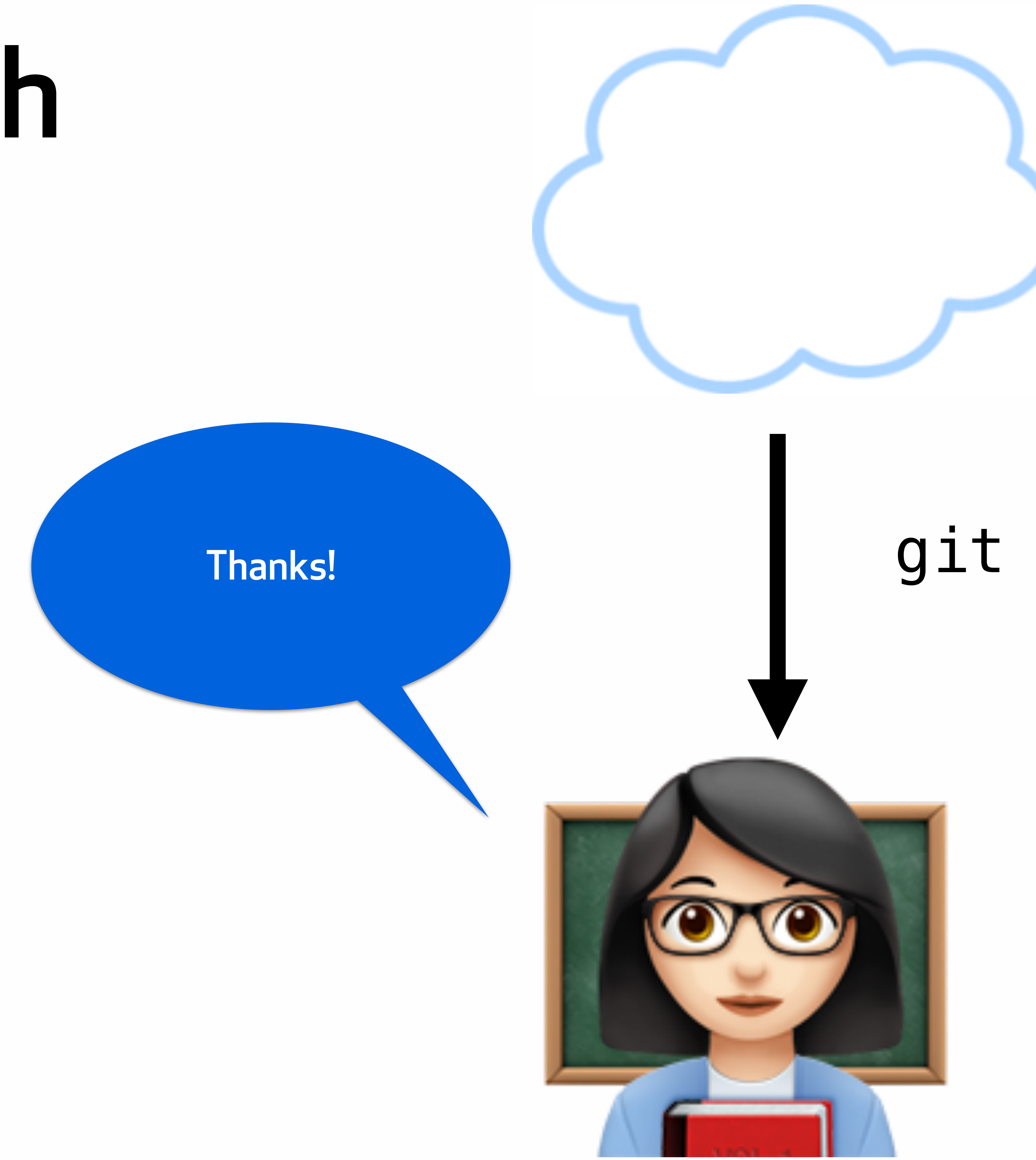

Hmm, when I run git log I can't see these commits in my local repo. :(

git fetch

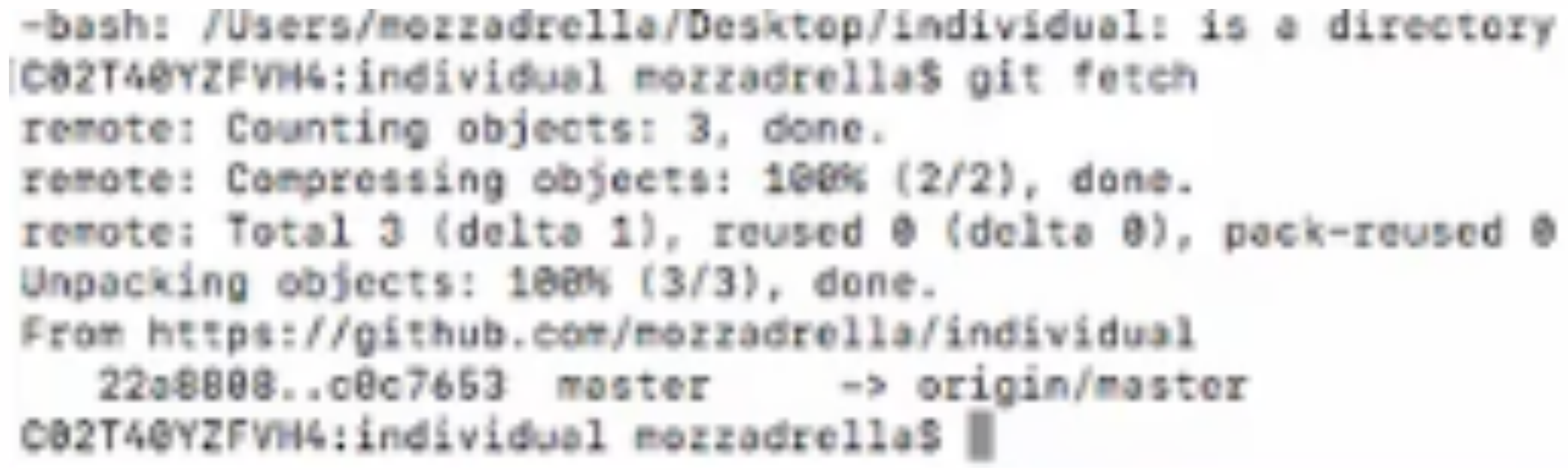


```
-bash: /Users/mozzadrella/Desktop/individual: is a directory
C02T40YZFVH4:individual mozzadrella$ git fetch
remote: Counting objects: 3, done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), done.
From https://github.com/mozzadrella/individual
   22a8808..c0c7653  master     -> origin/master
C02T40YZFVH4:individual mozzadrella$
```

Fetch

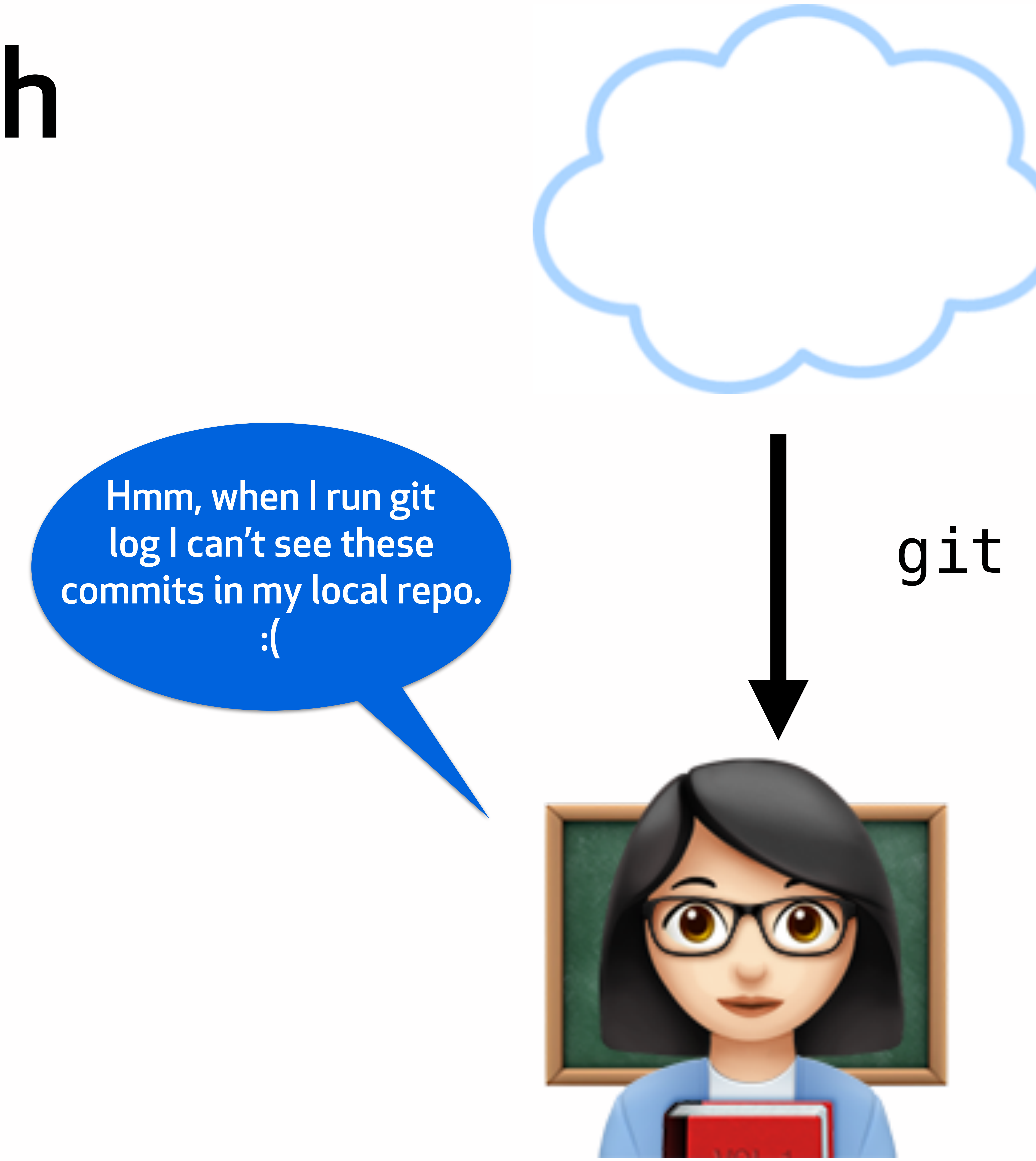


git fetch



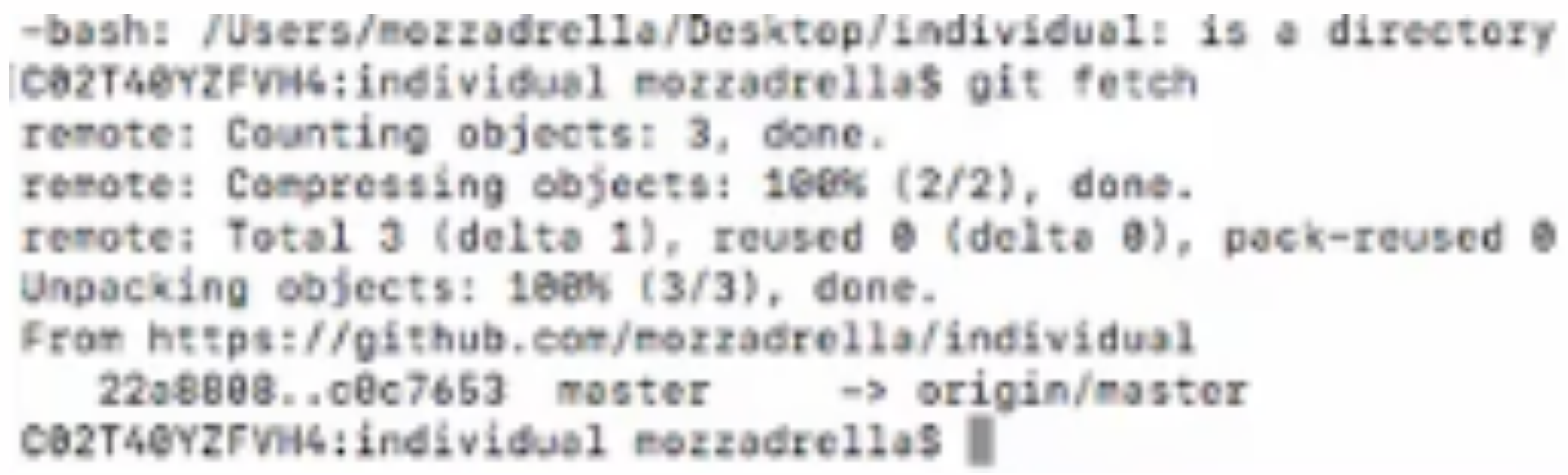
```
-bash: /Users/mozzadrella/Desktop/individual: is a directory
C02T40YZFVH4:individual mozzadrella$ git fetch
remote: Counting objects: 3, done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), done.
From https://github.com/mozzadrella/individual
   22a8808..c0c7653  master     -> origin/master
C02T40YZFVH4:individual mozzadrella$ 
```

Fetch



Hmm, when I run git log I can’t see these commits in my local repo. :(

`git fetch`



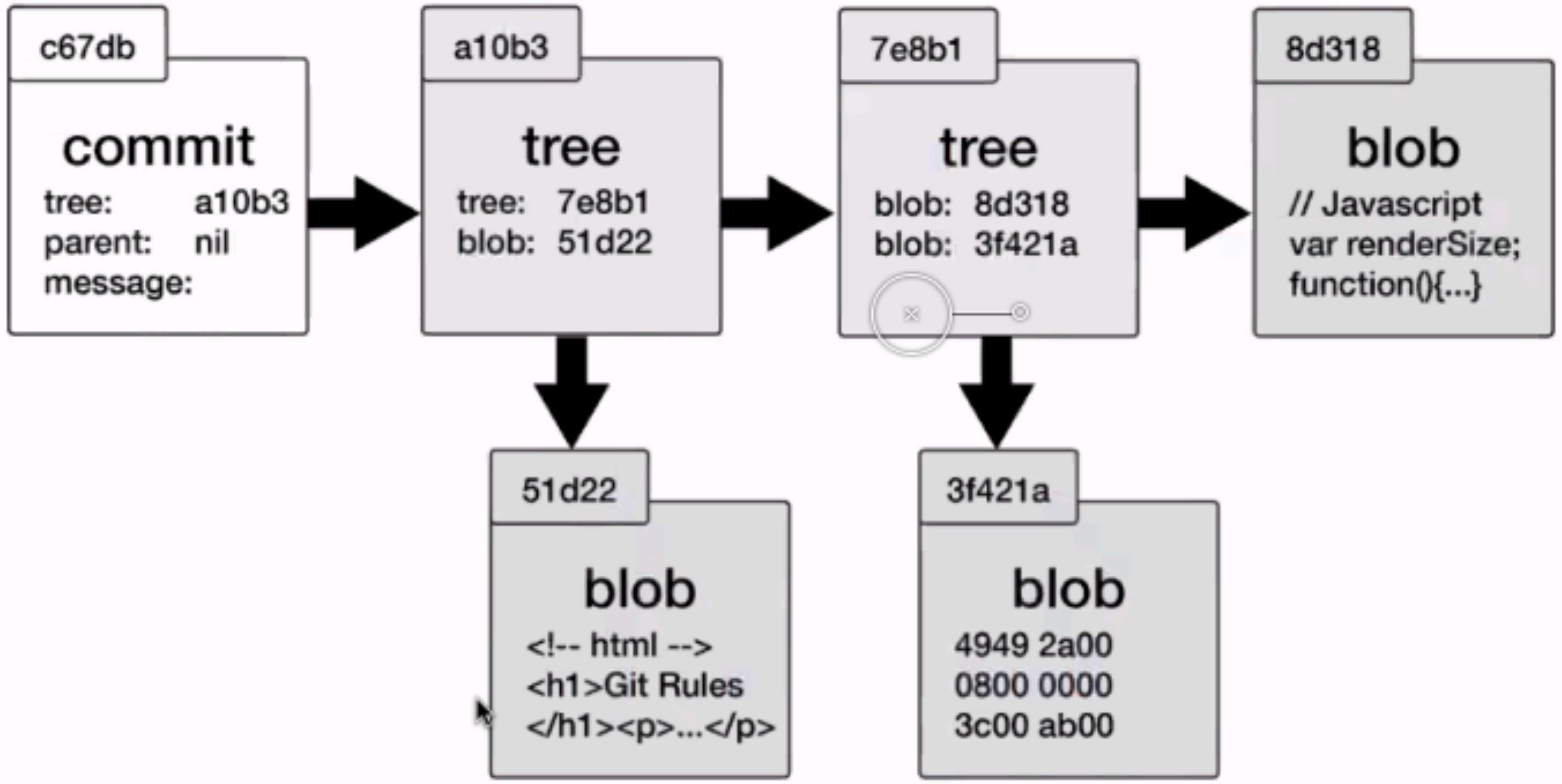
```
-bash: /Users/mozzadrella/Desktop/individual: is a directory
C02T40YZFVH4:individual mozzadrella$ git fetch
remote: Counting objects: 3, done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), done.
From https://github.com/mozzadrella/individual
   22a8808..c0c7653  master     -> origin/master
C02T40YZFVH4:individual mozzadrella$ 
```

Counting objects

Git **only** transmits the necessary objects.

Push: sends objects the remote doesn't have.

Fetch: receives objects we don't have locally.



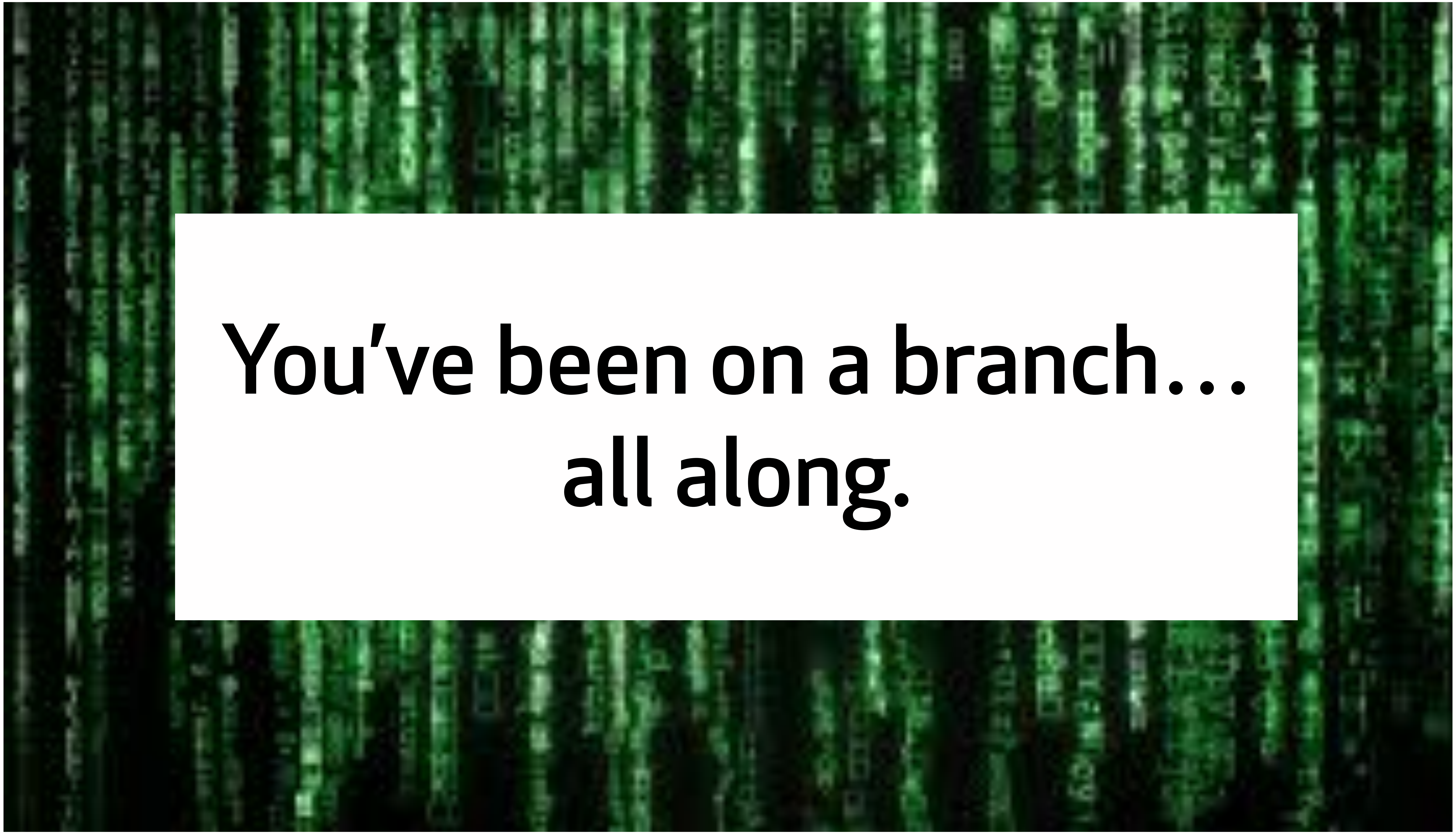
Which branch do you want to push?

```
git push —u origin master
```

You want to push master.
To origin, the remote.

But what is “master”?

```
git push –u origin master
```



You've been on a branch…
all along.

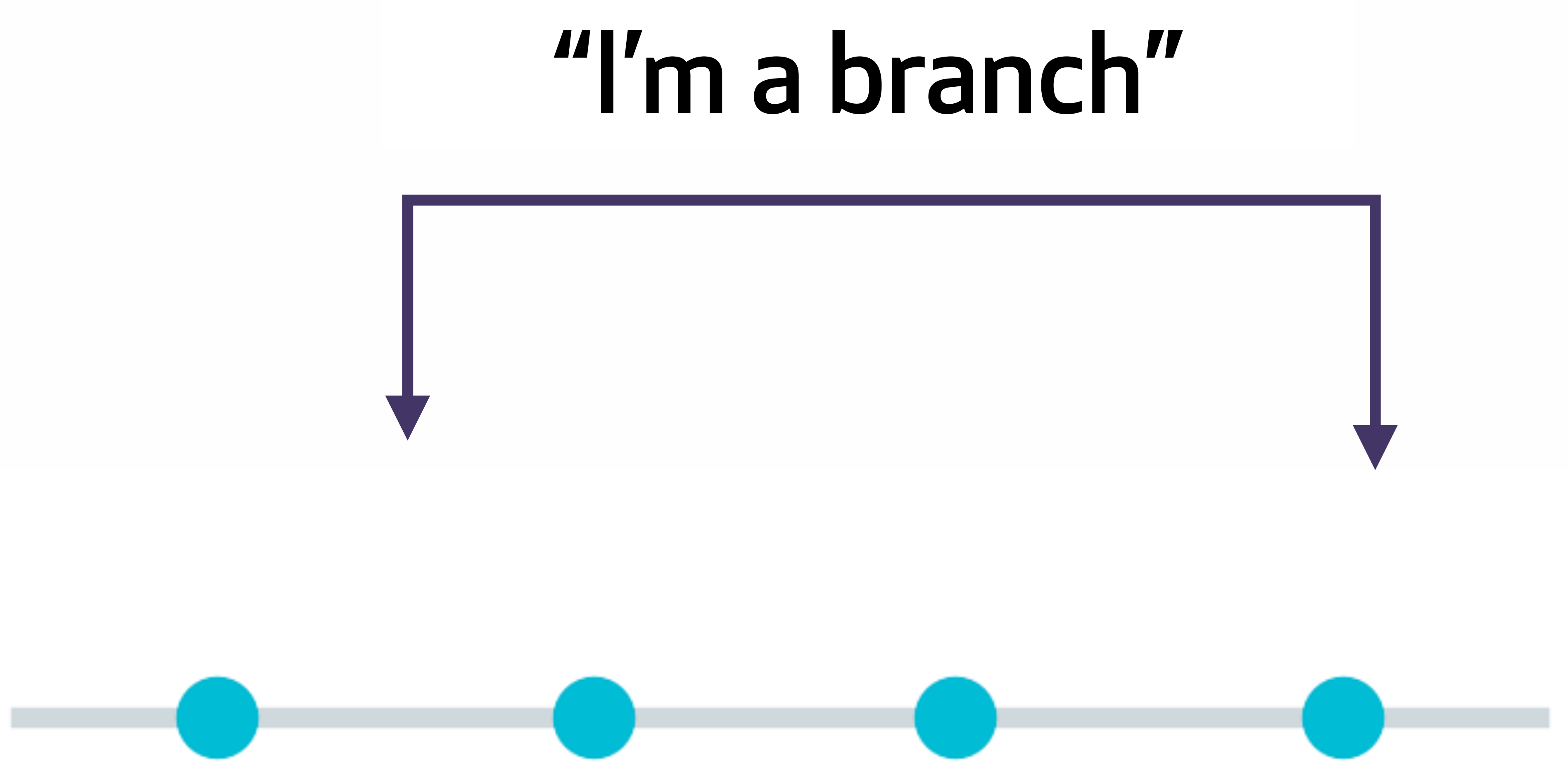
An aside to discuss branches.

Branches are bookmarks to commits

"Master" is the default, it's a naming convention.

Can think about branches as either a bookmark or a pointer for commits.

As we add commits, the active branch updates to point to the newest commit (HEAD).



demo : master
Search
Filter
Local branches
master
origin branches
origin/master
demo
Changes
All Commits
Branches
master 3↑
Remotes
origin
master
Tags
Stashes
Submodules
Filter
Commit
Changes
File Tree

master add info about Fork, m	rella	cc30f26	Today at 1:28 PM
add info about branches	rella	6c8bfc8	Today at 1:25 PM
add learning objectives to readme.	rella	83203d0	Today at 1:23 PM
origin/master readme.md	rella	bce46da	Today at 12:36 PM

Author: Mozzadrella <mozzadrella@github.com>

Date: December 16, 2017 at 1:28:05 PM EST

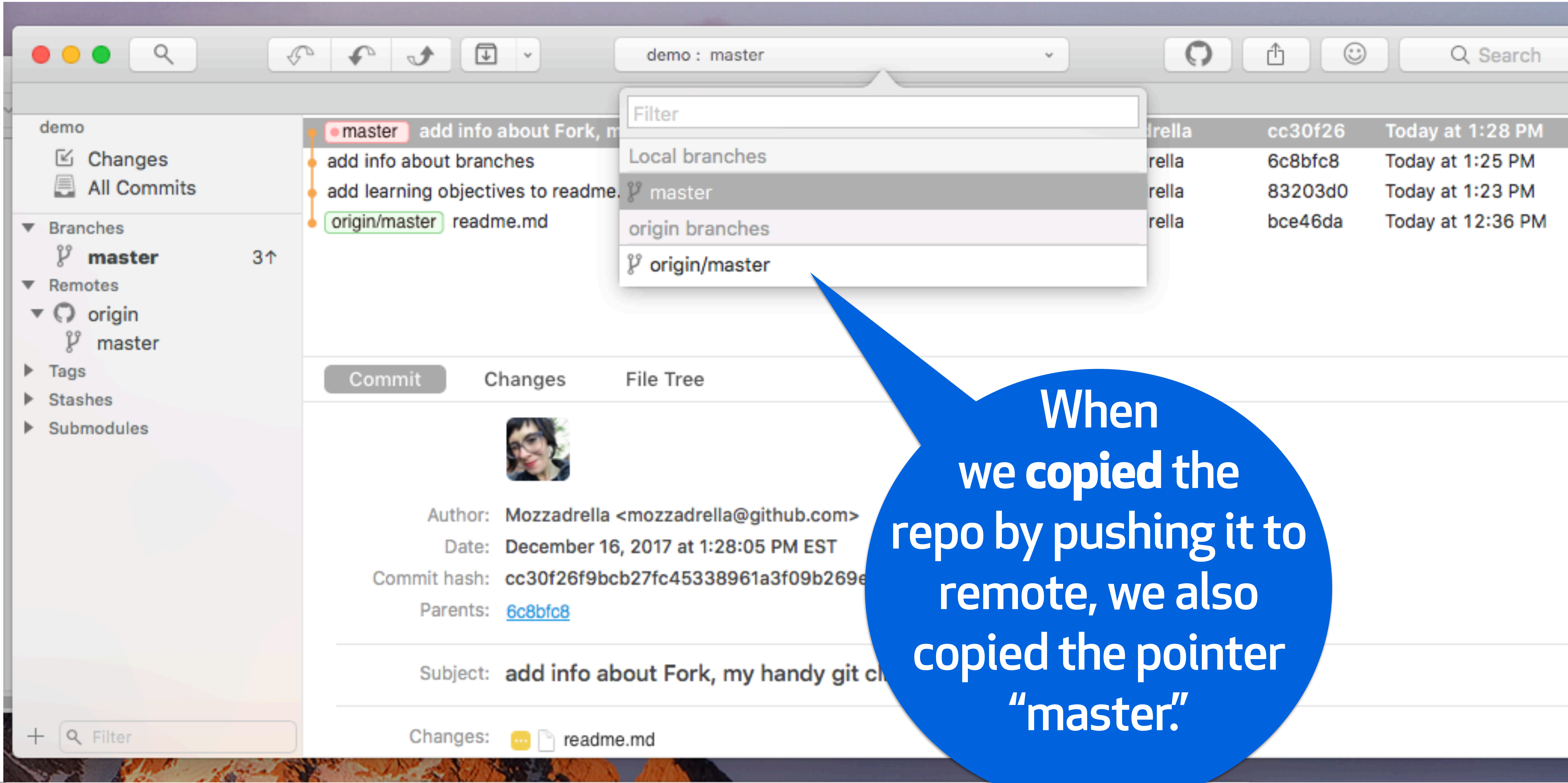
Commit hash: cc30f26f9bcb27fc45338961a3f09b269ecd0931

Parents: 6c8bfc8

Subject: add info about Fork, my handy git client

Changes: readme.md

Individual assignments



master add info about Fork, m	rella	cc30f26	Today at 1:28 PM
add info about branches	rella	6c8bfc8	Today at 1:25 PM
add learning objectives to readme.	rella	83203d0	Today at 1:23 PM
origin/master readme.md	rella	bce46da	Today at 12:36 PM

Author: Mozzadrella <mozzadrella@github.com>

Date: December 16, 2017 at 1:28:05 PM EST

Commit hash: cc30f26f9bcb27fc45338961a3f09b269e

Parents: 6c8bfc8

Subject: add info about Fork, my handy git cl

Changes: readme.md

When we **copied** the repo by pushing it to remote, we also copied the pointer "master."

demo : master

Filter

Local branches

master

origin branches

origin/master

demo

Changes

All Commits

Branches

master 3↑

Remotes

origin

master

Tags

Stashes

Submodules

master add info about Fork, m	rella	cc30f26	Today at 1:28 PM
add info about branches	rella	6c8bfc8	Today at 1:25 PM
add learning objectives to readme.	rella	83203d0	Today at 1:23 PM
origin/master readme.md	rella	bce46da	Today at 12:36 PM

Changes File Tree

ozzadrell

ecemb

c30f26f9bcb2

6c8bfc8

ct: add info about For

Changes: readme.md

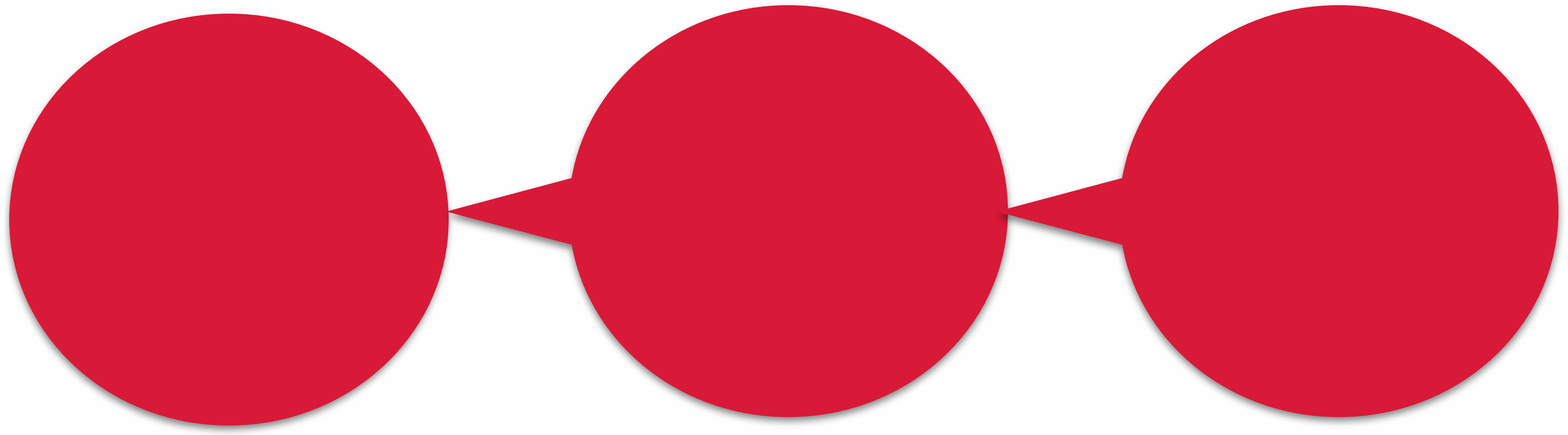
Filter

Since branches are bookmarks to commits…

in effect, every commit could be a branch.

When you commit , the active branch (bookmark) is updated.

Using branches in your terminal



Remember, branches are pointers to commits.

```
git show master
```

If we say ‘git show master’ we’ll see the commit master points to.

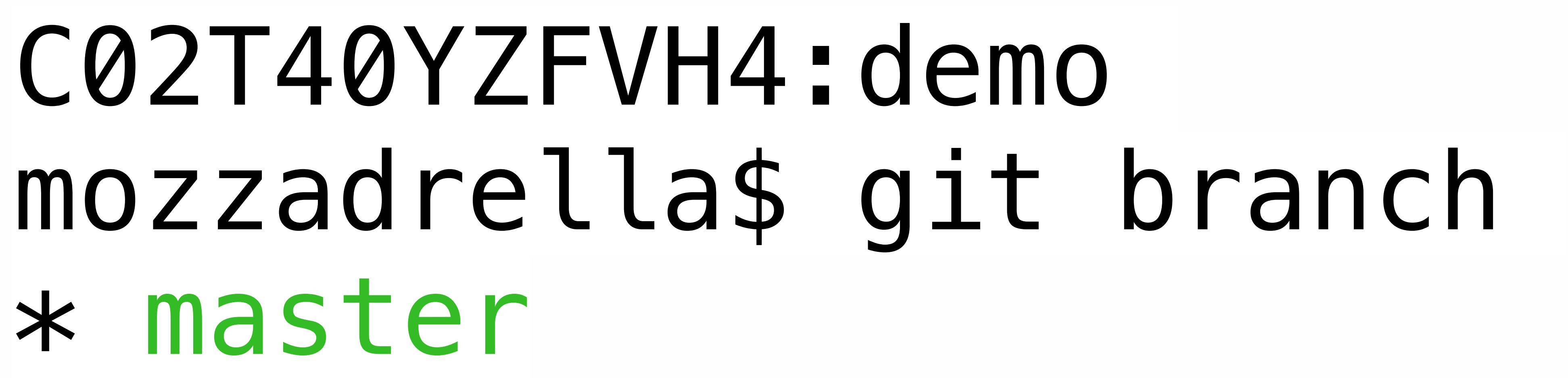
```
commit cc30f26f9bcb27fc45338961a3f09b269ecd0931
(HEAD -> master)
Author: Mozzadrella <mozzadrella@github.com>
Date:   Sat Dec 16 13:28:05 2017 -0500
```

Finding the active branch

‘Git branch’ will show you the branches in your project…

and the “*” indicates your currently active branch.

If you made commits at that moment, the active branch would be updated to point to the new commit.



```
C02T40YZFVH4:demo
mozzadrella$ git branch
* master
```

Creating a new branch

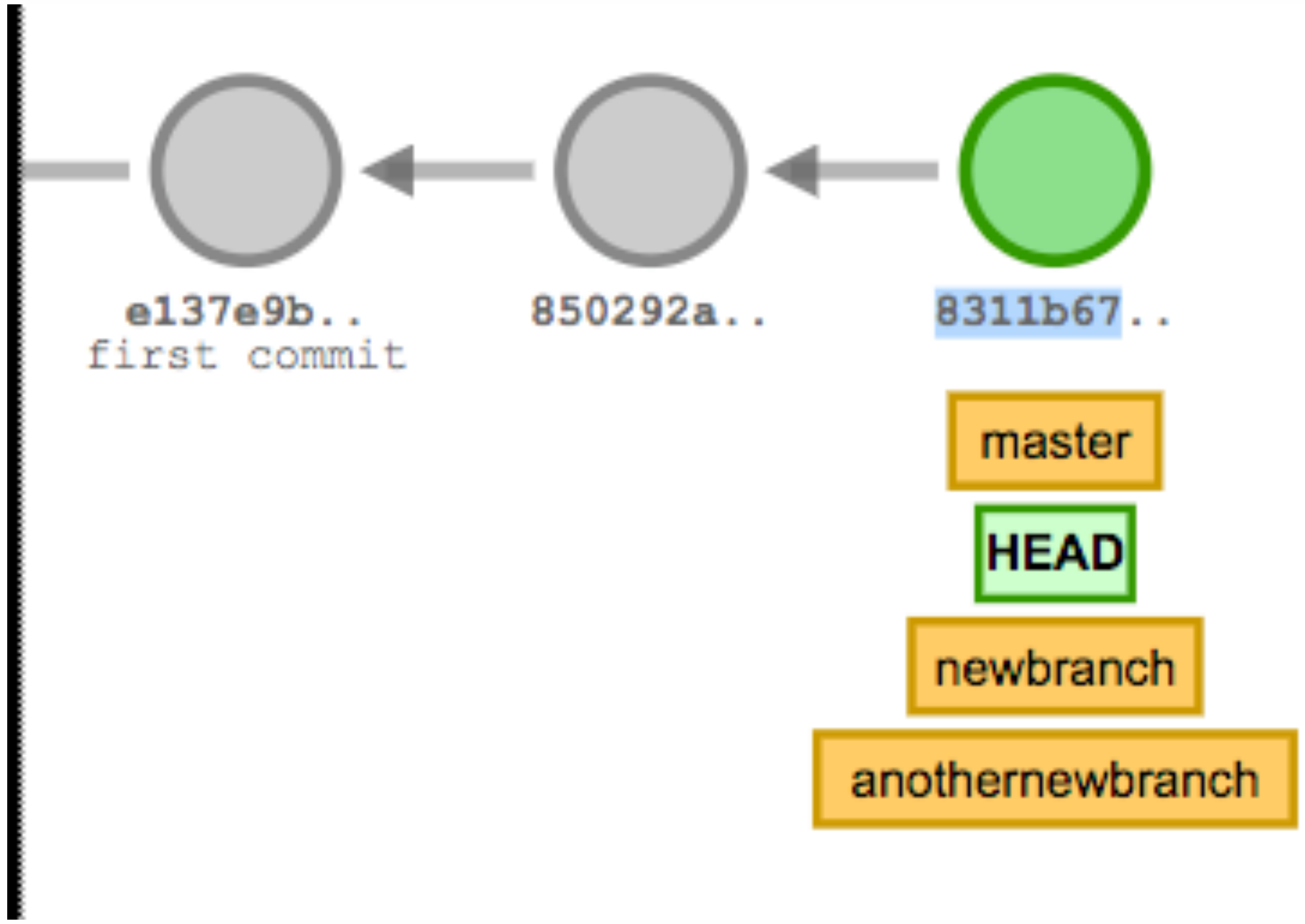
To summon a new branch, use ‘git branch’ and the new branch name. We’ll call ours ‘newbranch’

```
C02T40YZFVH4:demo mozzadrella$ git branch newbranch
C02T40YZFVH4:demo mozzadrella$ git branch
* master
  newbranch
```

Branches point back to the currently active commit

If we created 2 new branches from

8311b67



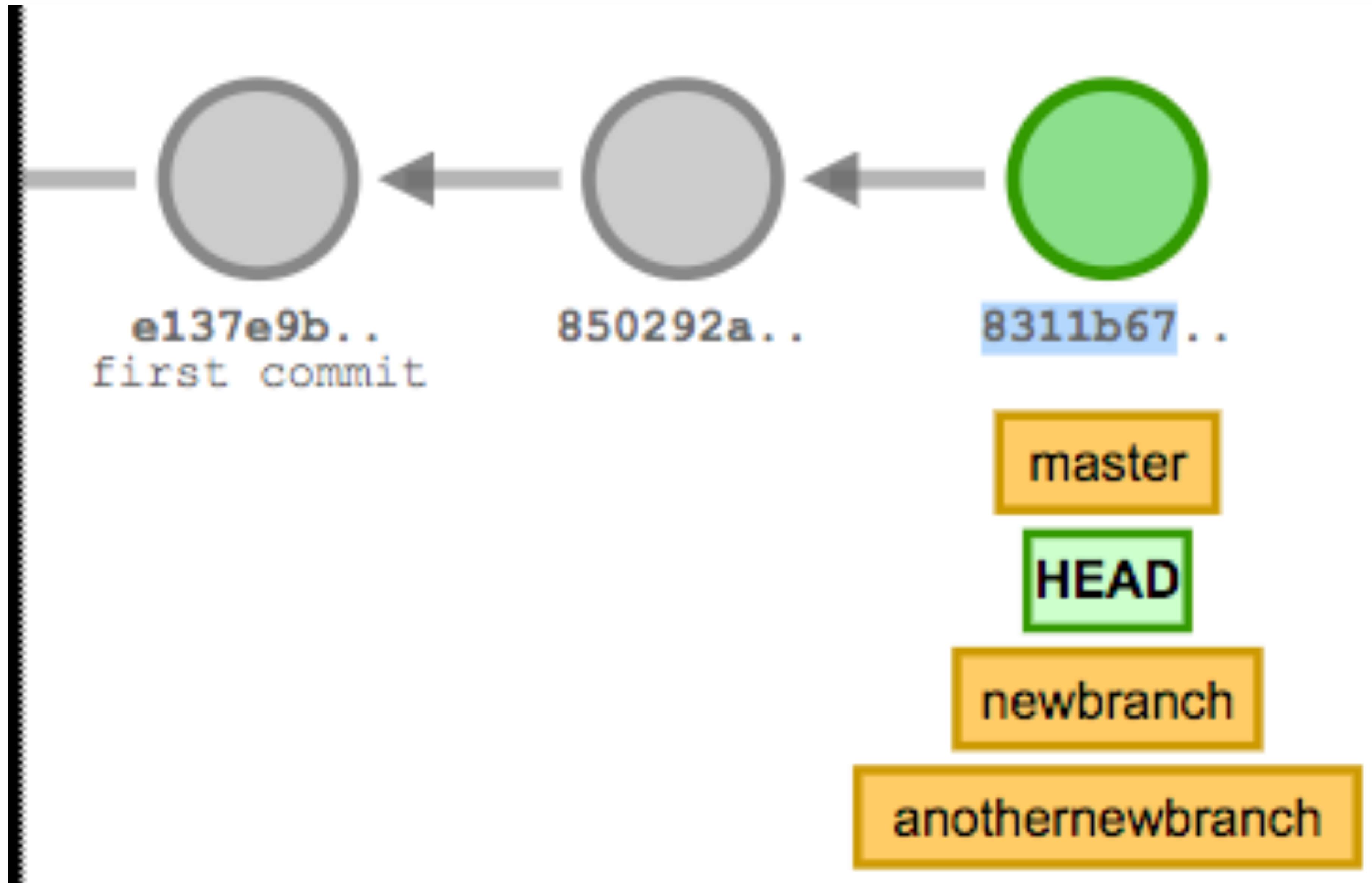
Branches point back to the currently active commit

If we created 2 new branches from

8311b67…

they would both point to

8311b67

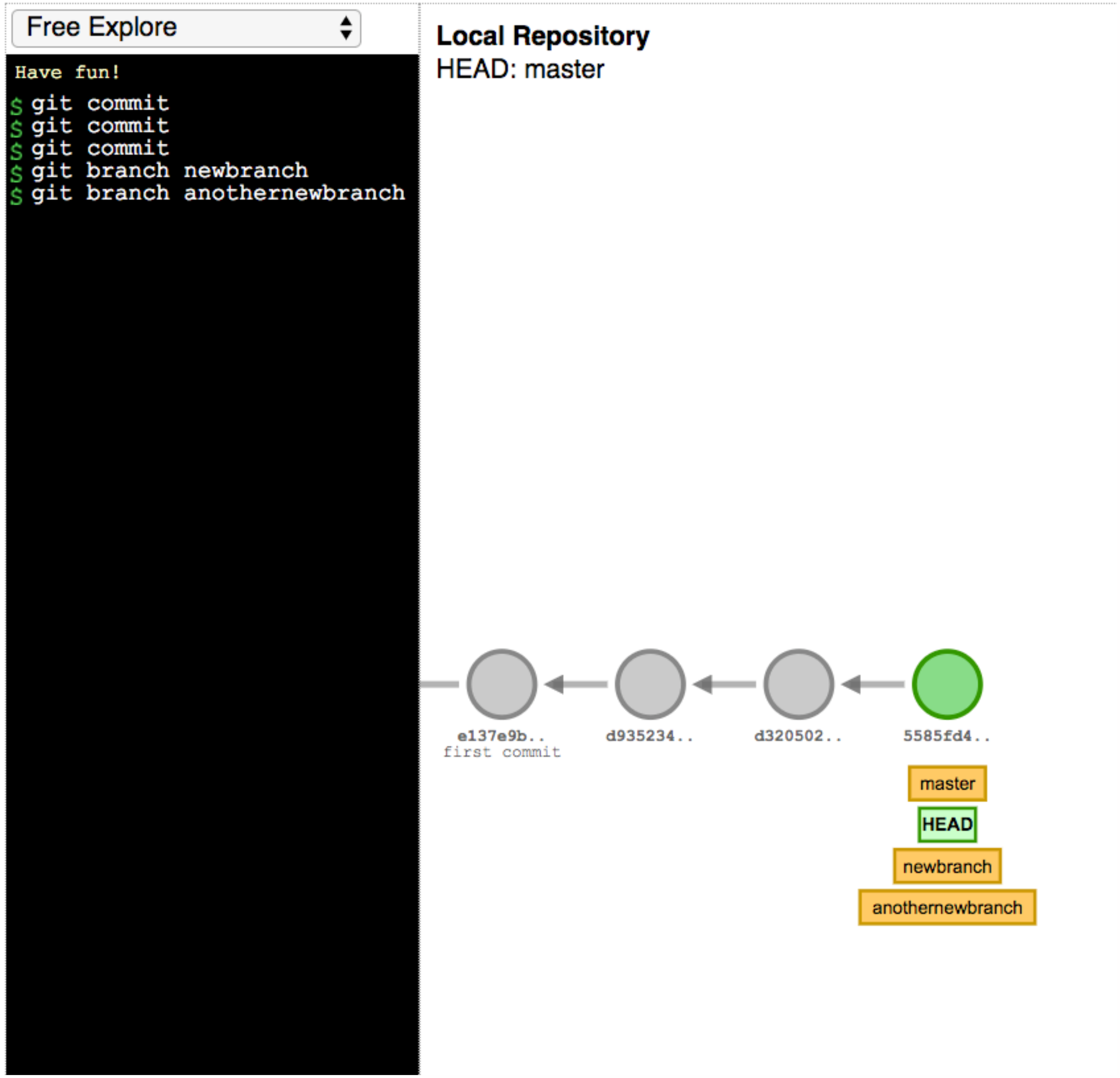


Branching from commits using references

`git branch <name>` creates a branch at HEAD

`git branch <name> <ref>` creates a branch at `<ref>`

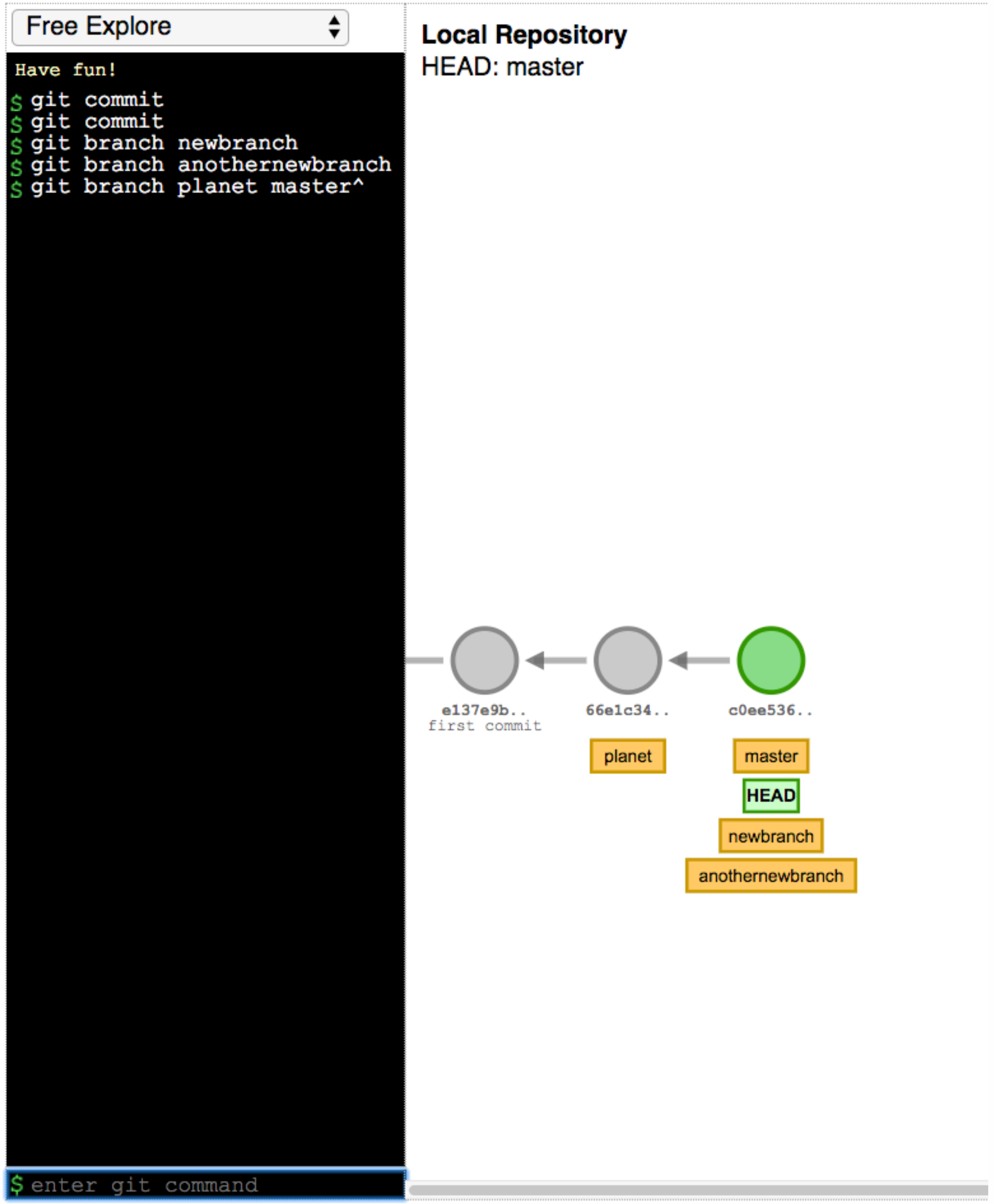
`<ref>` can be HEAD, a branch name, a commit, or a commit-ish (e.g. HEAD^ or master~3)



```
Have fun!
$ git commit
$ git commit
$ git commit
$ git branch newbranch
$ git branch anothernewbranch
```

Or branch from previous commits

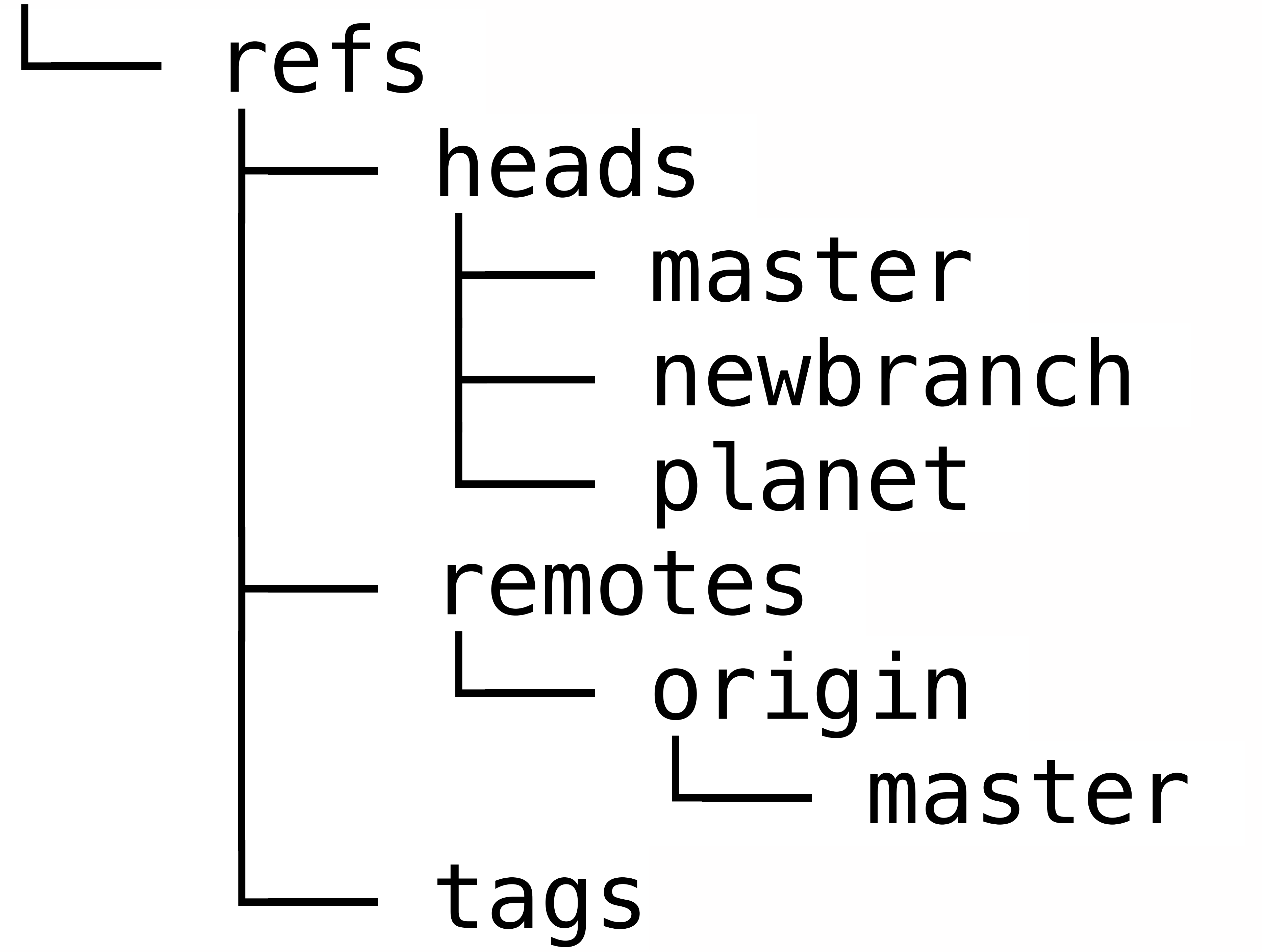
```
git branch planet master^
```



```
Have fun!
$ git commit
$ git commit
$ git branch newbranch
$ git branch anothernewbranch
$ git branch planet master^
$ enter git command
```

A final word on branches…

```
tree .git/refs
```



```
└── refs
    ├── heads
    │   ├── master
    │   ├── newbranch
    │   └── planet
    ├── remotes
    │   └── origin
    │       └── master
    └── tags
```

Those files contain the commit ID…

```
C02T40YZFVH4:demo mozzadrella$ cat .git/refs/heads/planet
6c8bfc88bb440844f18a5e0a6ca885998b461bb7
```

So the implementation for branches is a file with a hash in it.

Back to the world of network activity.

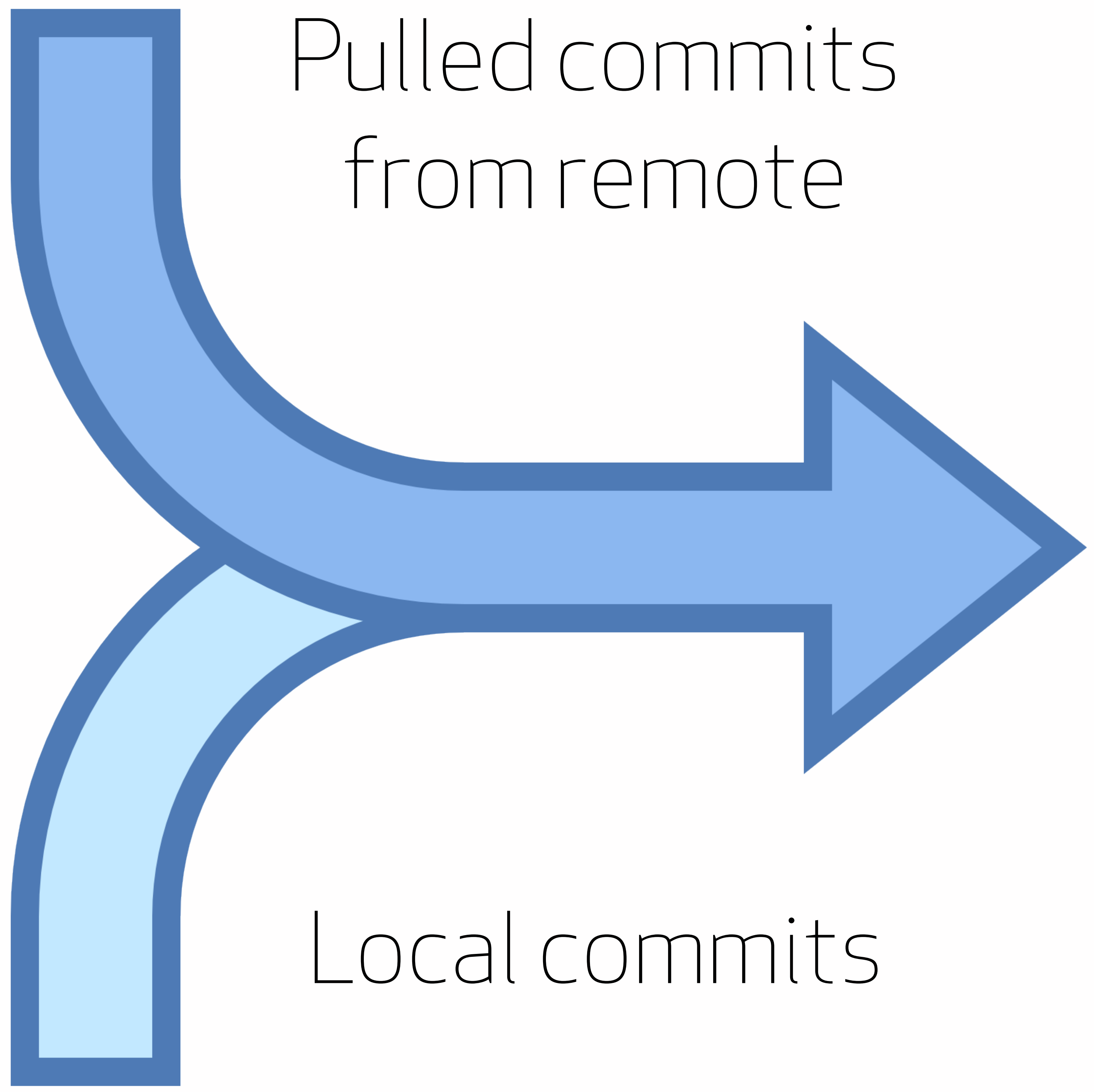
Pull = fetch + merge

Pull first fetches the commits and stores them locally.

Merge takes the two divergent commits, puts them together in the staging area and makes a new commit with two parents.

Merge updates the active branch to point to the new merge commit

You'll see the new commits reflected in your local project when you run "git log."

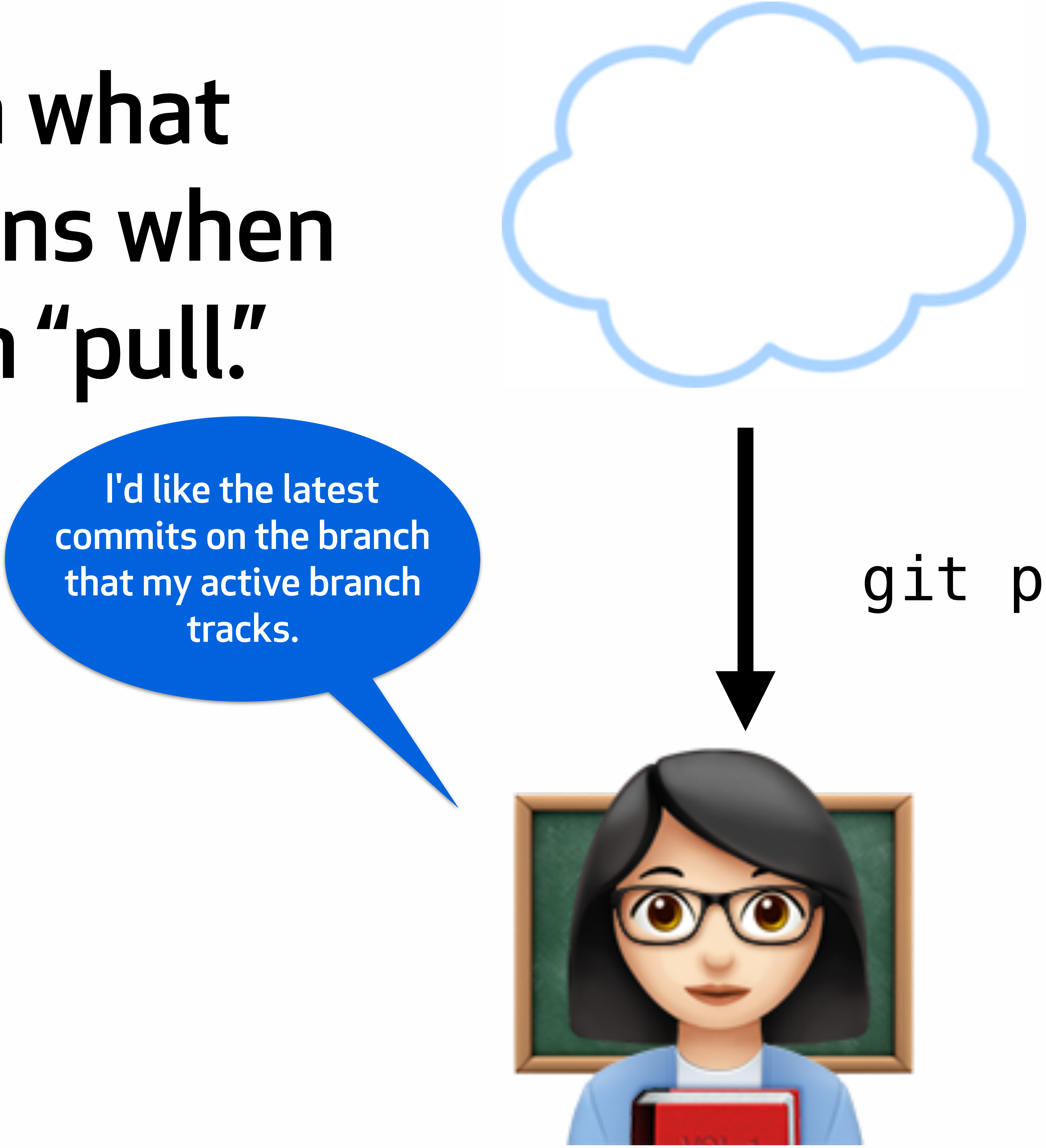


Pulled commits from remote

Local commits

Merged!

Watch what happens when we run “pull.”

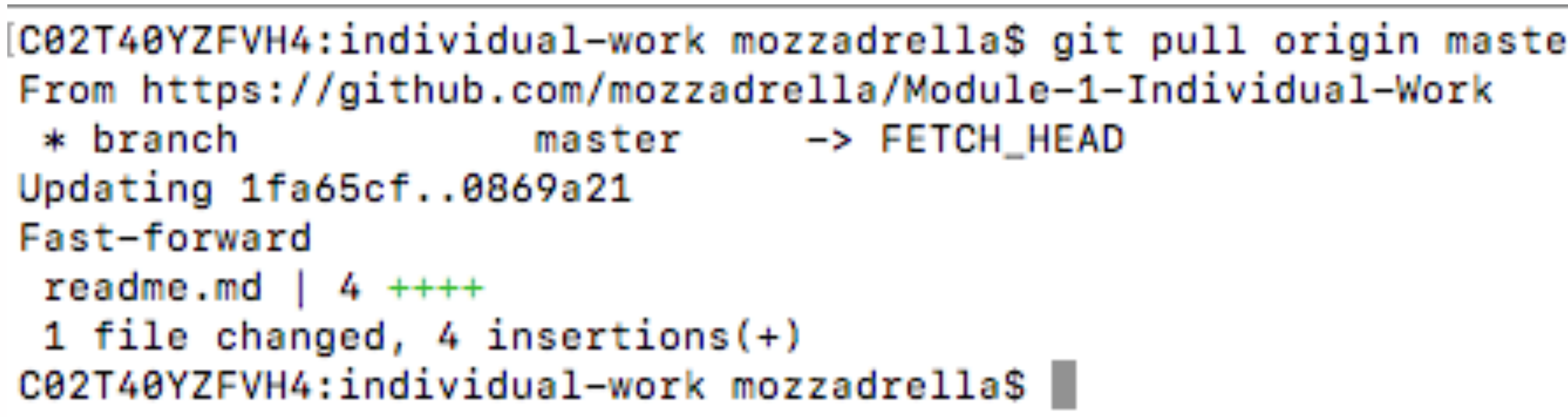


I'd like the latest commits on the branch that my active branch tracks.

```
git pull origin master
```

Watch what happens when we run "pull."

git pull origin master



```
C02T40YZFVH4:individual-work mozzadrella$ git pull origin maste
From https://github.com/mozzadrella/Module-1-Individual-Work
 * branch            master     -> FETCH_HEAD
Updating 1fa65cf..0869a21
Fast-forward
 readme.md | 4 ++++
 1 file changed, 4 insertions(+)
C02T40YZFVH4:individual-work mozzadrella$ 
```

To sum up, here are the commands with network activity:

```
git push
git fetch
git pull (fetch part, not merge)
```