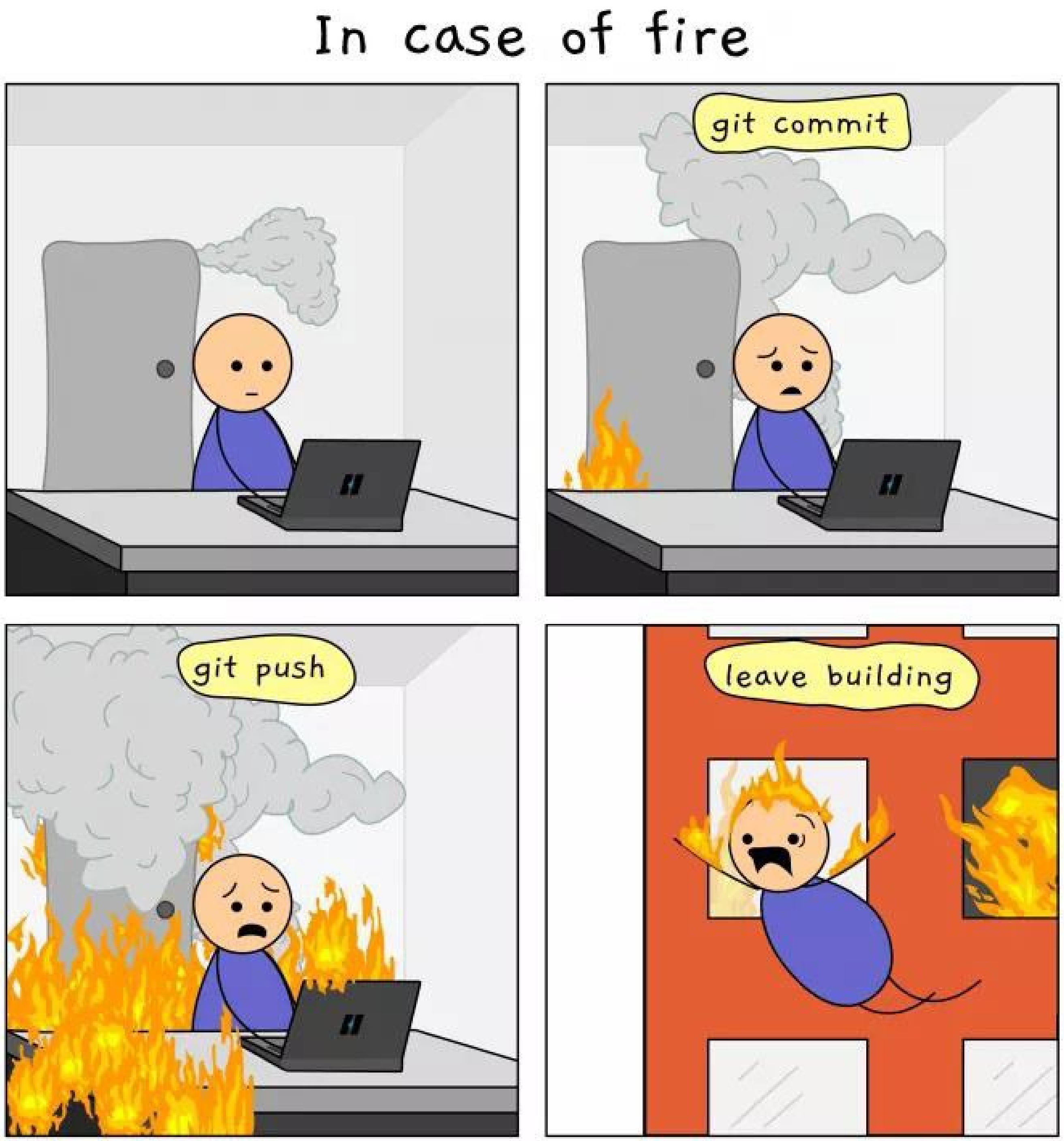
Git

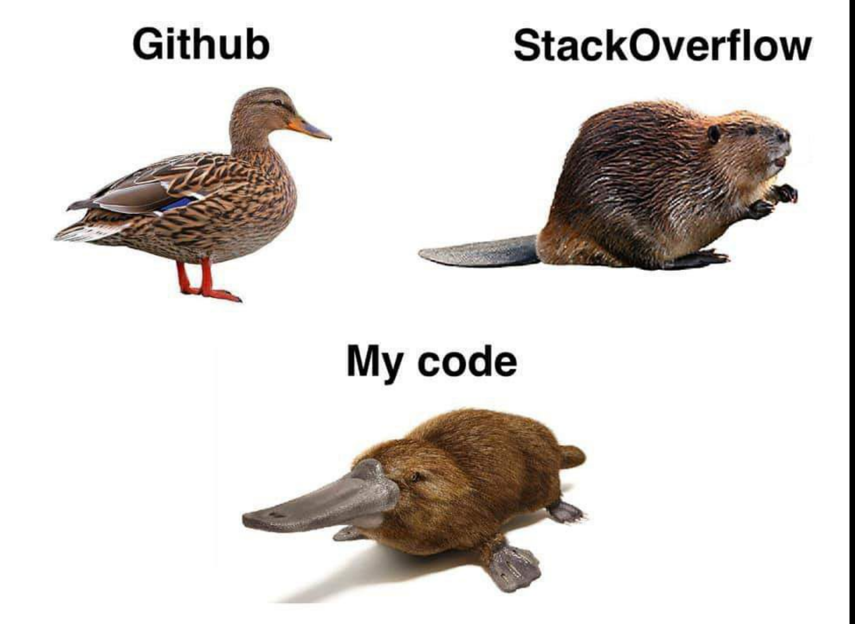
aws HEROES

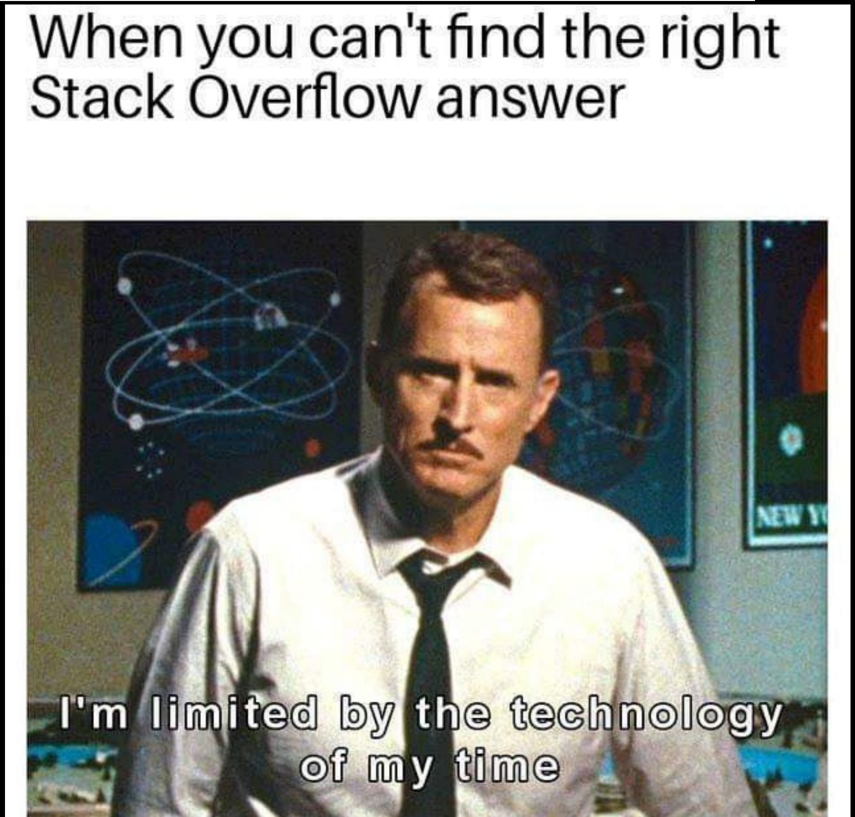
MVP Microsoft® Most Valuable Professional

Experts

Cyrus Wong


In case of fire
git commit
git push
leave building


Github
StackOverflow
My code


When you can't find the right Stack Overflow answer
I'm limited by the technology of my time

The code will be executed successfully
or
You will be executed successfully
imgflip.com

这图做的还可以

GAOXIAOGIF.COM

Create a Repository

From scratch -- Create a new local repository

```
$ git init [project name]
```

Download from an existing repository

```
$ git clone my_url
```

Observe your Repository

List new or modified files not yet committed

```
$ git status
```

Show the changes to files not yet staged

```
$ git diff
```

Show the changes to staged files

```
$ git diff --cached
```

Show all staged and unstaged file changes

```
$ git diff HEAD
```

Show the changes between two commit ids

```
$ git diff commit1 commit2
```

List the change dates and authors for a file

```
$ git blame [file]
```

Show the file changes for a commit id and/or file

```
$ git show [commit]:[file]
```

Show full change history

```
$ git log
```

Show change history for file/directory including diffs

```
$ git log -p [file/directory]
```

Working with Branches

List all local branches

```
$ git branch
```

List all branches, local and remote

```
$ git branch -av
```

Switch to a branch, my_branch, and update working directory

```
$ git checkout my_branch
```

Create a new branch called new_branch

```
$ git branch new_branch
```

Delete the branch called my_branch

```
$ git branch -d my_branch
```

Merge branch_a into branch_b

```
$ git checkout branch_b
$ git merge branch_a
```

Tag the current commit

```
$ git tag my_tag
```

Make a change

Stages the file, ready for commit

```
$ git add [file]
```

Stage all changed files, ready for commit

```
$ git add .
```

Commit all staged files to versioned history

```
$ git commit -m "commit message"
```

Commit all your tracked files to versioned history

```
$git commit -am "commit message"
```

Unstages file, keeping the file changes

```
$ git reset [file]
```

Revert everything to the last commit

```
$ git reset --hard
```

Synchronize

Get the latest changes from origin (no merge)

```
$ git fetch
```

Fetch the latest changes from origin and merge

```
$ git pull
```

Fetch the latest changes from origin and rebase

```
$ git pull --rebase
```

Push local changes to the origin

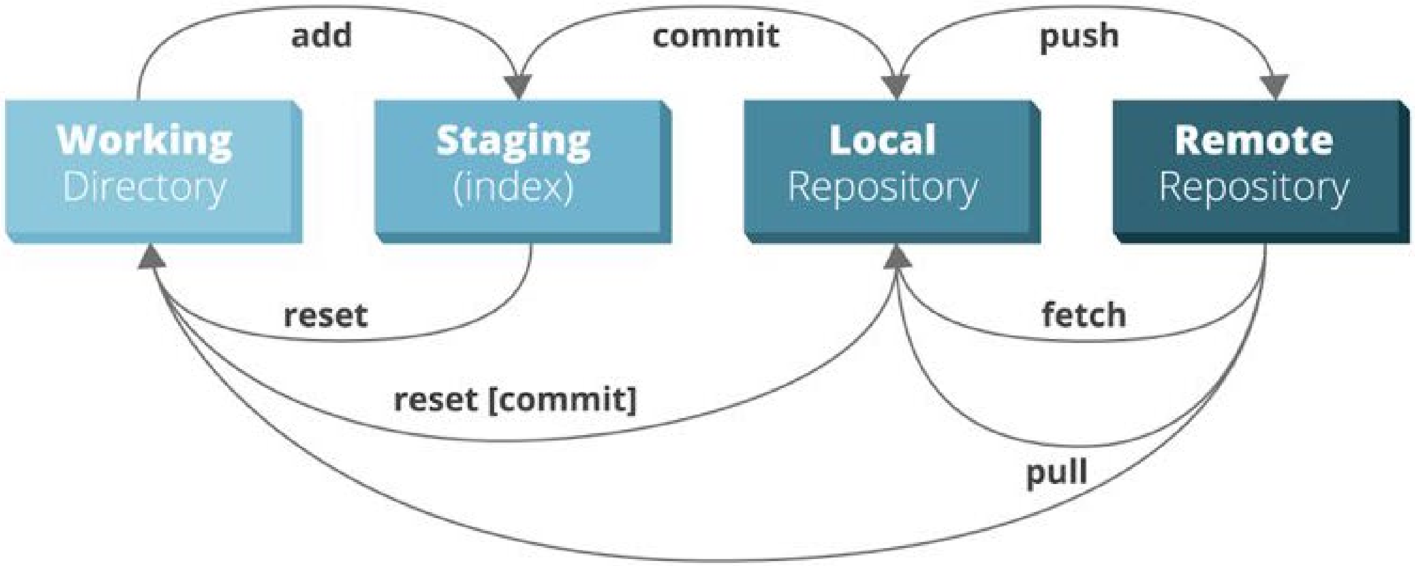
```
$ git push
```

Finally!

When in doubt, use git help

```
$ git command --help
```

Or visit https://training.github.com/ for official GitHub training.

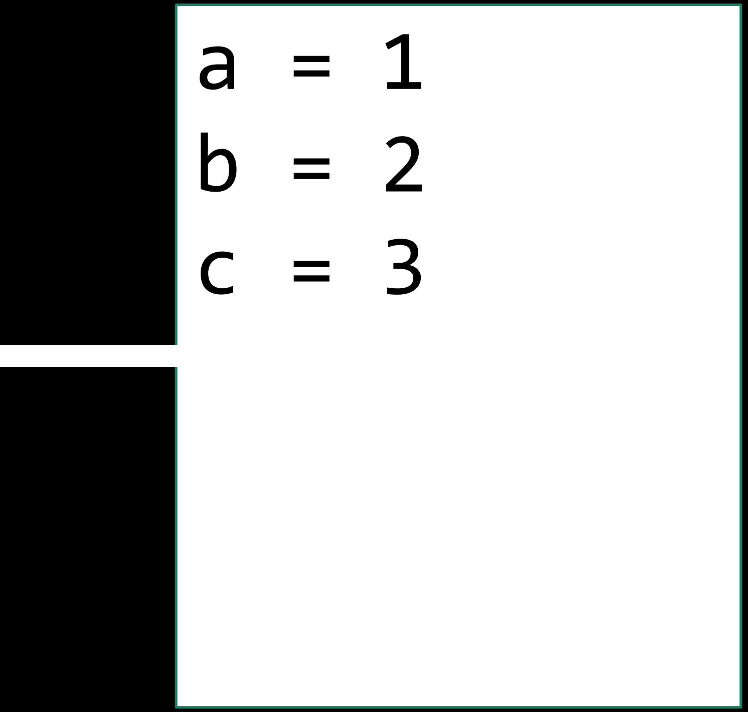


Git

- We use Git to keep track of changes to our code.

Git

- We use Git to keep track of changes to our code.



```
a = 1
b = 2
c = 3
```

Create file

Git

- We use Git to keep track of changes to our code.

```
a = 1
b = 2
c = 3
```

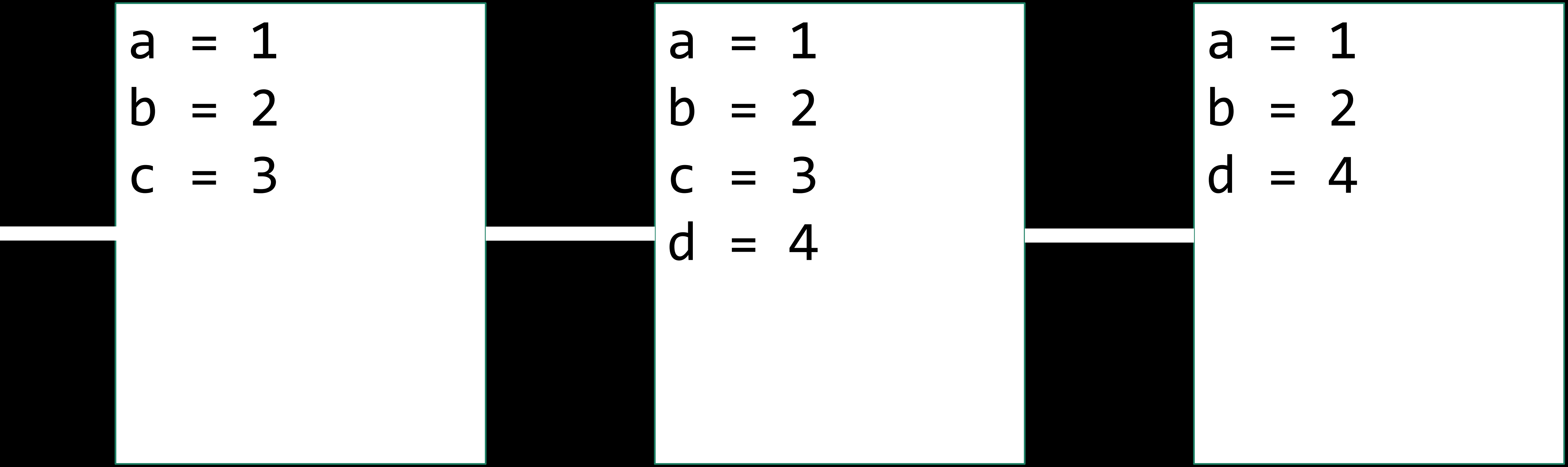
Create file

```
a = 1
b = 2
c = 3
d = 4
```

Add a line

Git

- We use Git to keep track of changes to our code.



```
a = 1
b = 2
c = 3
```

Create file

```
a = 1
b = 2
c = 3
d = 4
```

Add a line

```
a = 1
b = 2
d = 4
```

Delete a line

Git

- We also use Git to synchronize code across multiple devices.

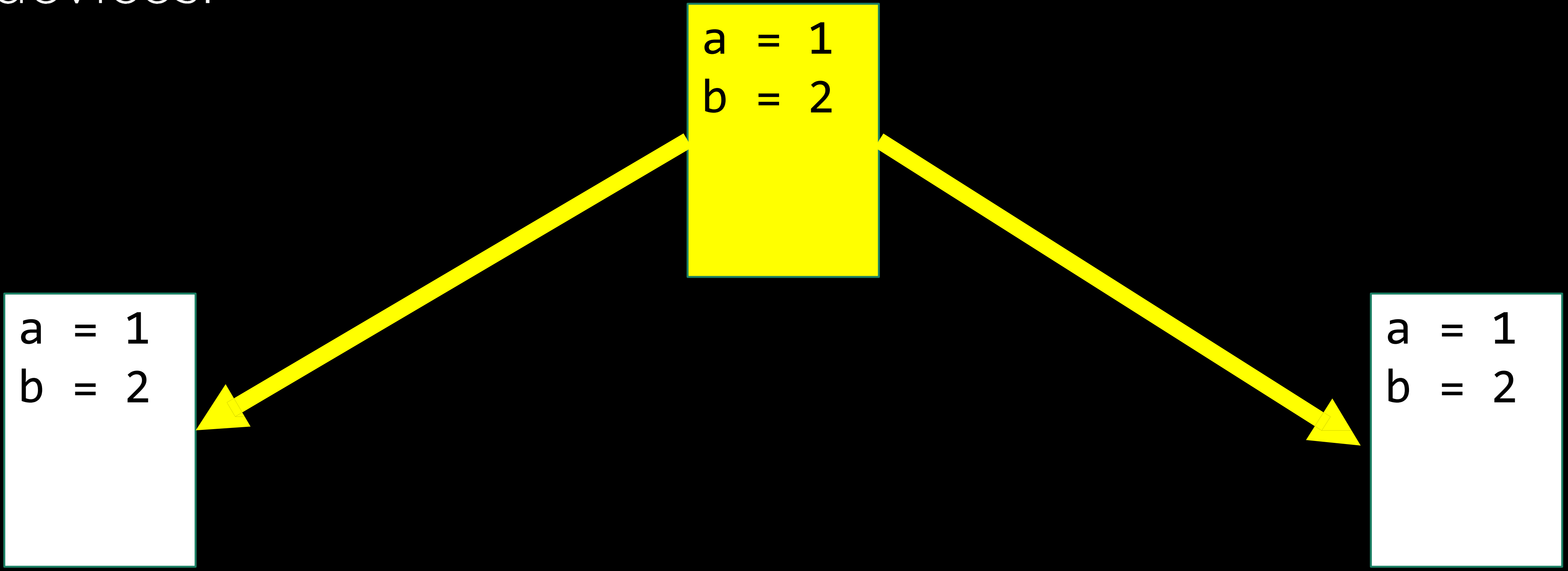
Git

- We also use Git to synchronize code across multiple devices.

```
a = 1
b = 2
```

Git

- We also use Git to synchronize code across multiple devices.



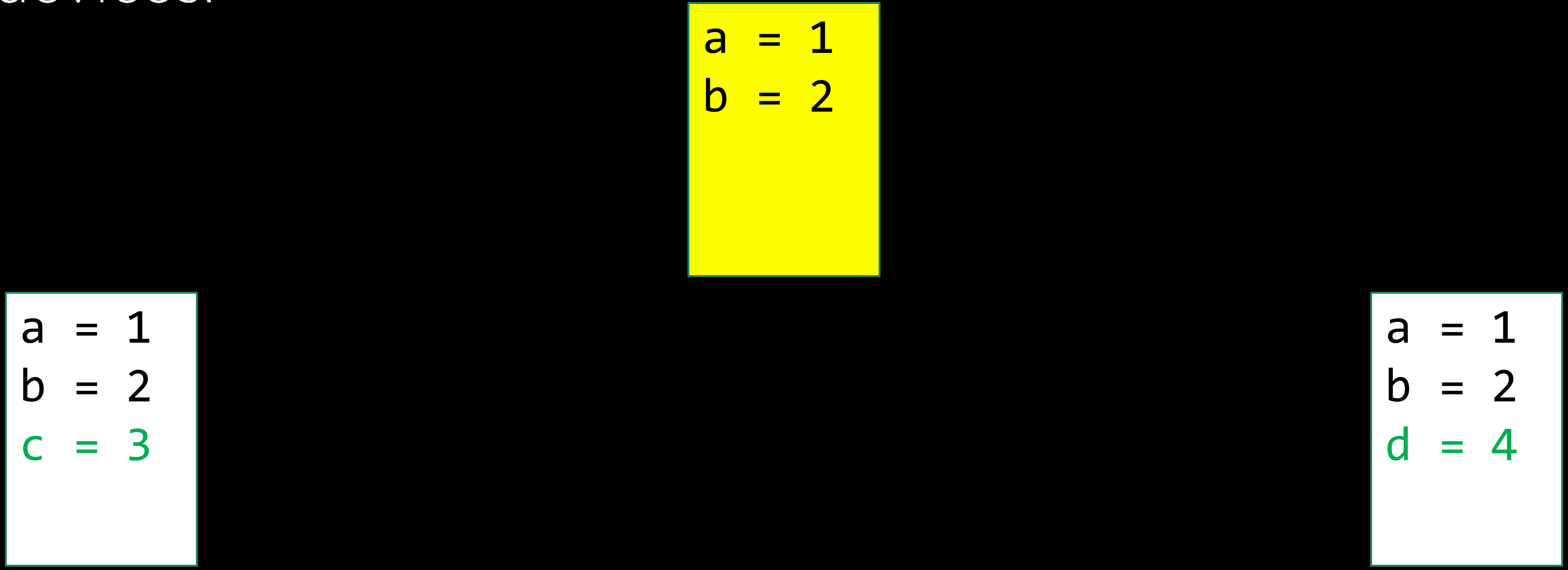
```
a = 1
b = 2
```

```
a = 1
b = 2
```

```
a = 1
b = 2
```

Git

- We also use Git to synchronize code across multiple devices.



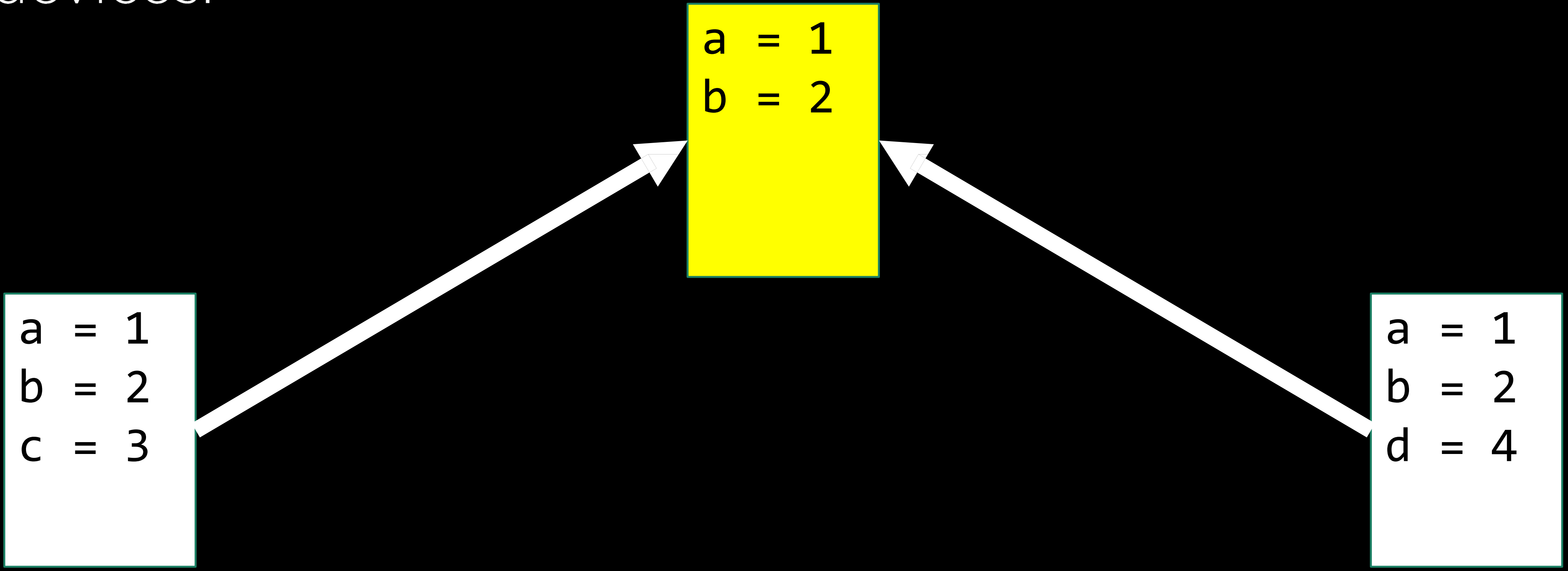
```
a = 1
b = 2
```

```
a = 1
b = 2
c = 3
```

```
a = 1
b = 2
d = 4
```

Git

- We also use Git to synchronize code across multiple devices.



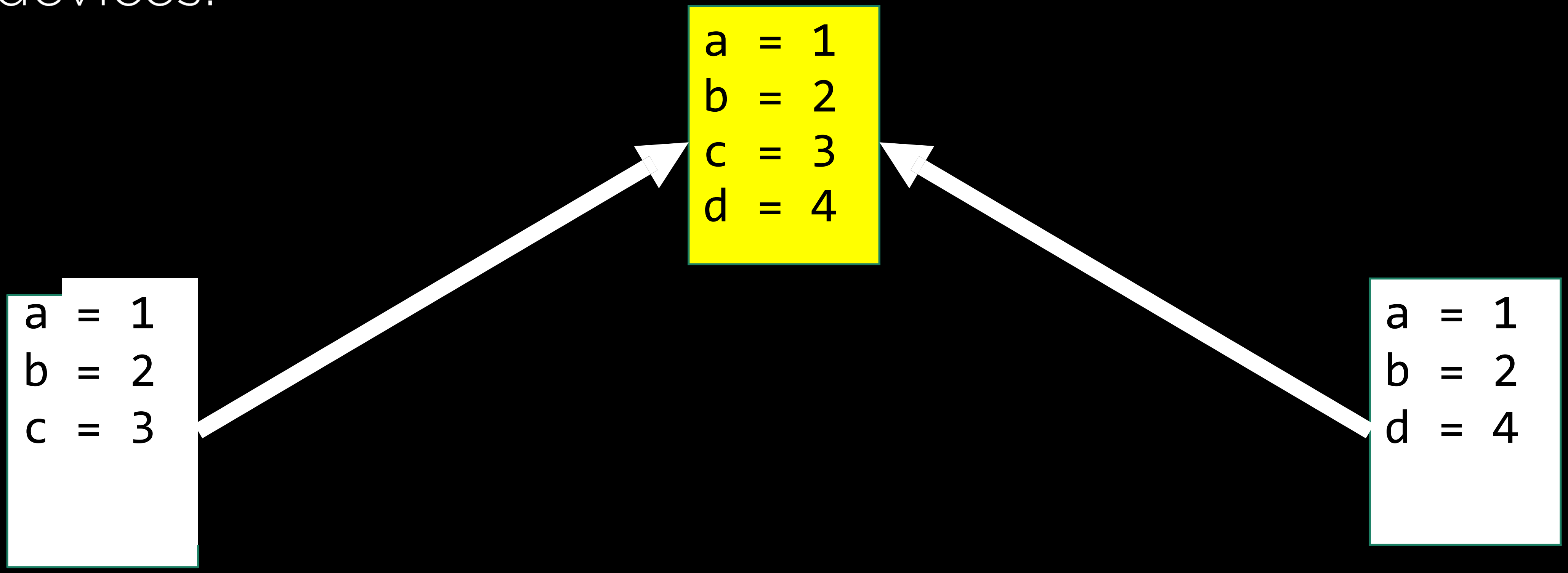
```
a = 1
b = 2
```

```
a = 1
b = 2
c = 3
```

```
a = 1
b = 2
d = 4
```

Git

- We also use Git to synchronize code across multiple devices.



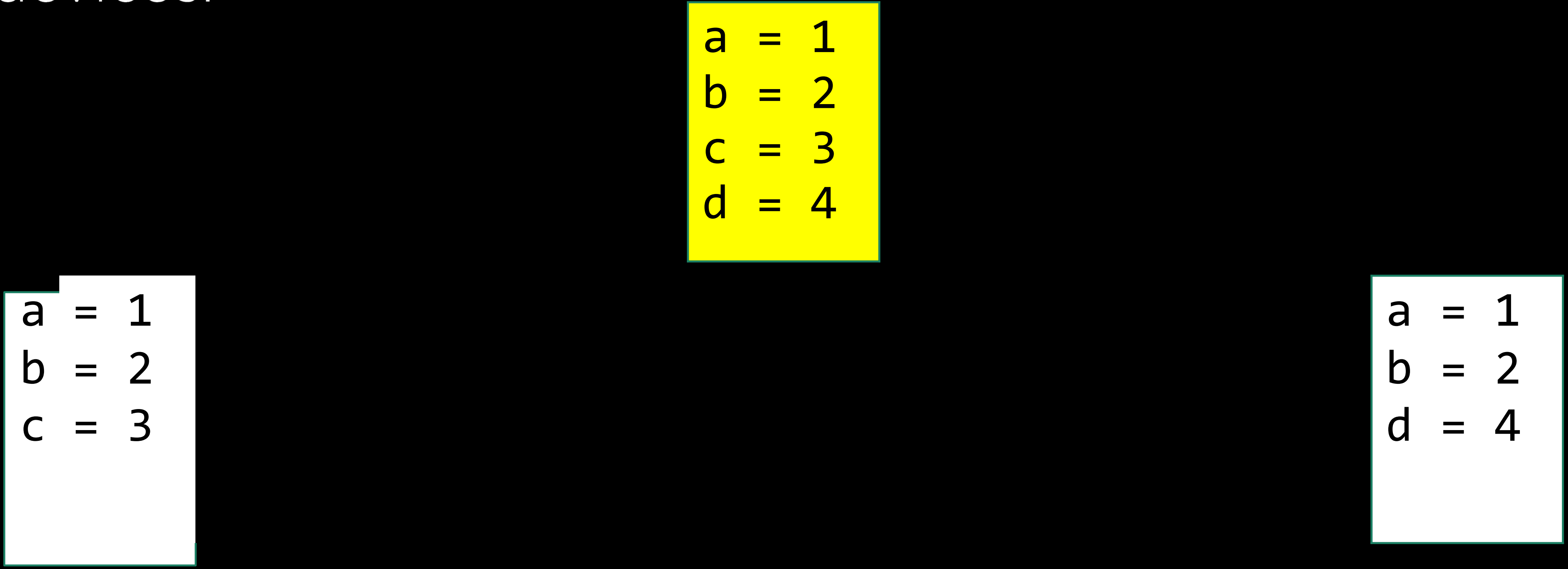
```
a = 1
b = 2
c = 3
d = 4
```

```
a = 1
b = 2
c = 3
```

```
a = 1
b = 2
d = 4
```

Git

- We also use Git to synchronize code across multiple devices.



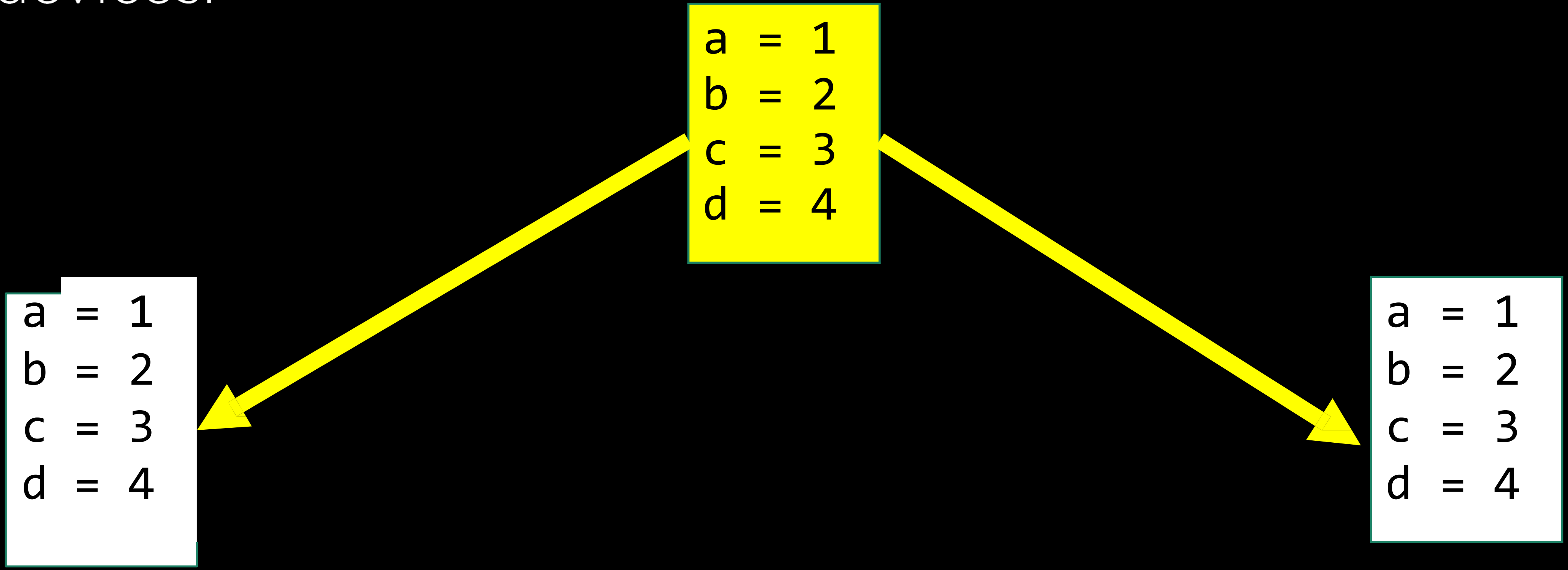
```
a = 1
b = 2
c = 3
d = 4
```

```
a = 1
b = 2
c = 3
```

```
a = 1
b = 2
d = 4
```

Git

- We also use Git to synchronize code across multiple devices.



```
a = 1
b = 2
c = 3
d = 4
```

```
a = 1
b = 2
c = 3
d = 4
```

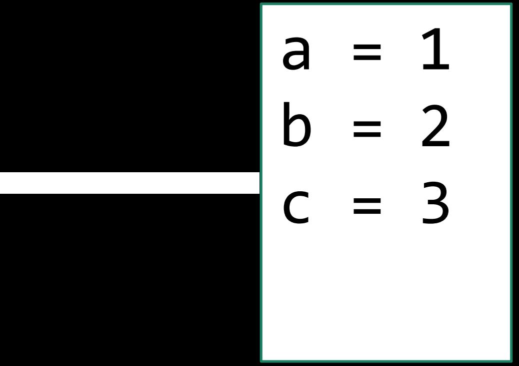
```
a = 1
b = 2
c = 3
d = 4
```

Git

- Git allows us to test changes to our code without losing the original.

Git

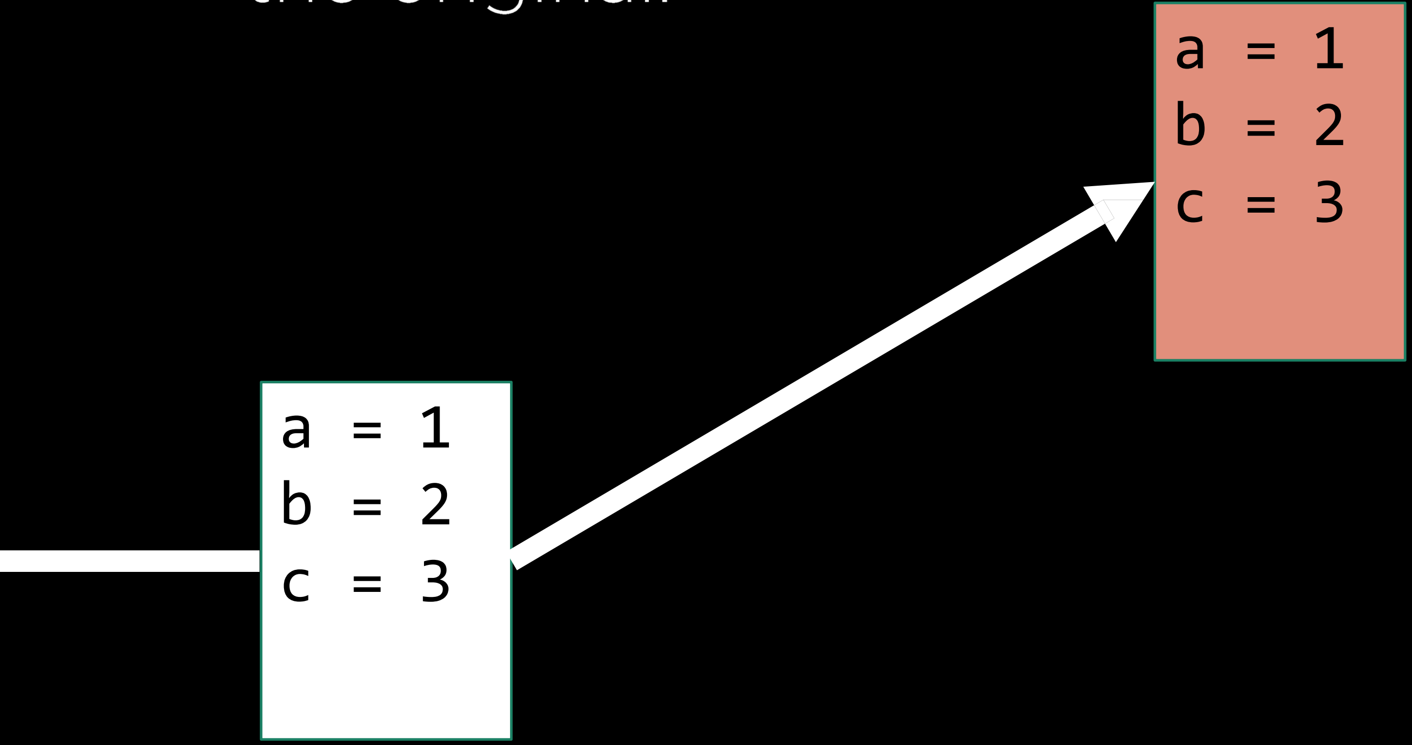
- Git allows us to test changes to our code without losing the original.



```
a = 1
b = 2
c = 3
```

Git

- Git allows us to test changes to our code without losing the original.

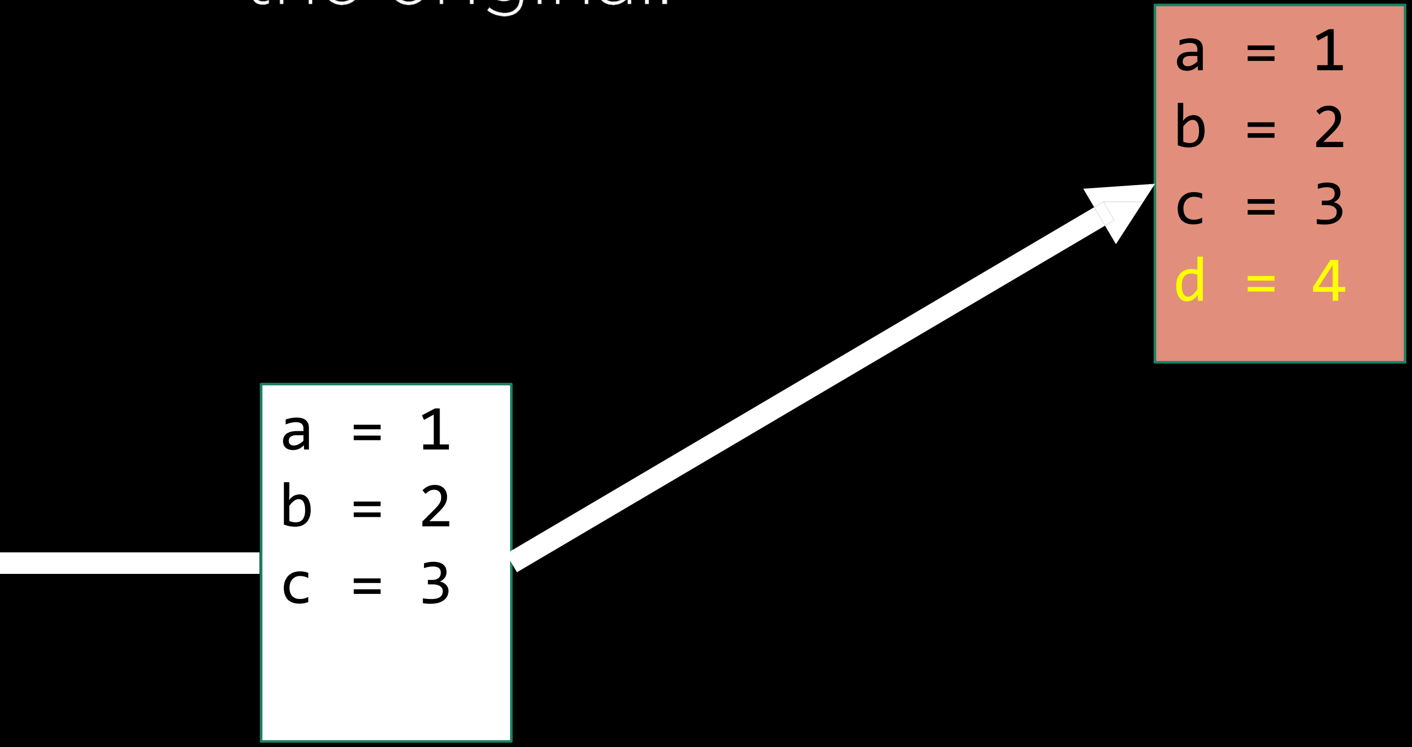


```
a = 1
b = 2
c = 3
```

```
a = 1
b = 2
c = 3
```

Git

- Git allows us to test changes to our code without losing the original.

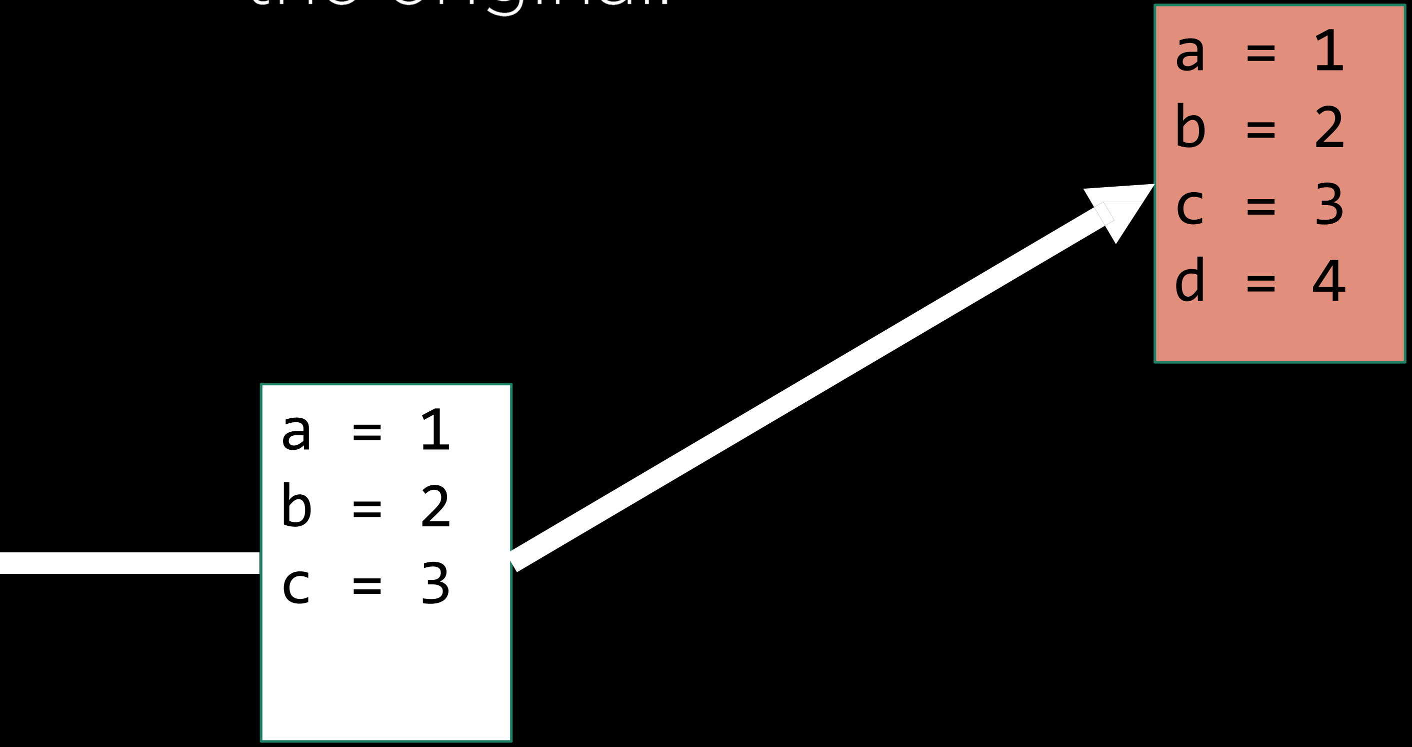


```
a = 1
b = 2
c = 3
d = 4
```

```
a = 1
b = 2
c = 3
```

Git

- Git allows us to test changes to our code without losing the original.

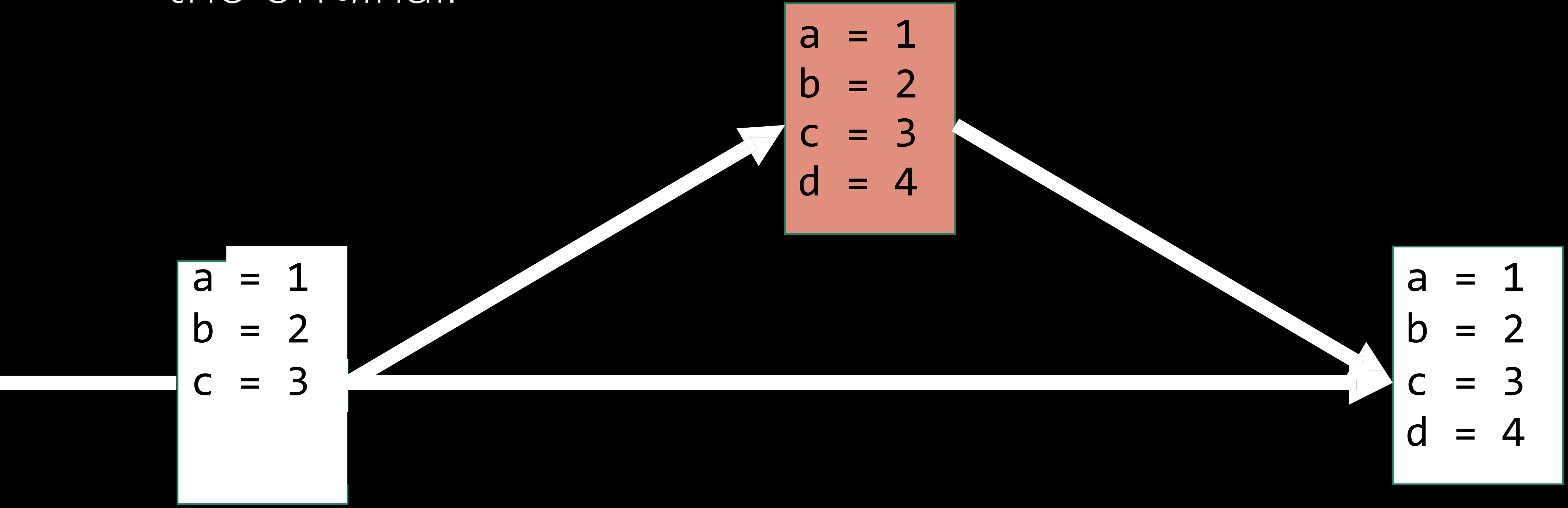


```
a = 1
b = 2
c = 3
```

```
a = 1
b = 2
c = 3
d = 4
```

Git

- Git allows us to test changes to our code without losing the original.



```
a = 1
b = 2
c = 3
```

```
a = 1
b = 2
c = 3
d = 4
```

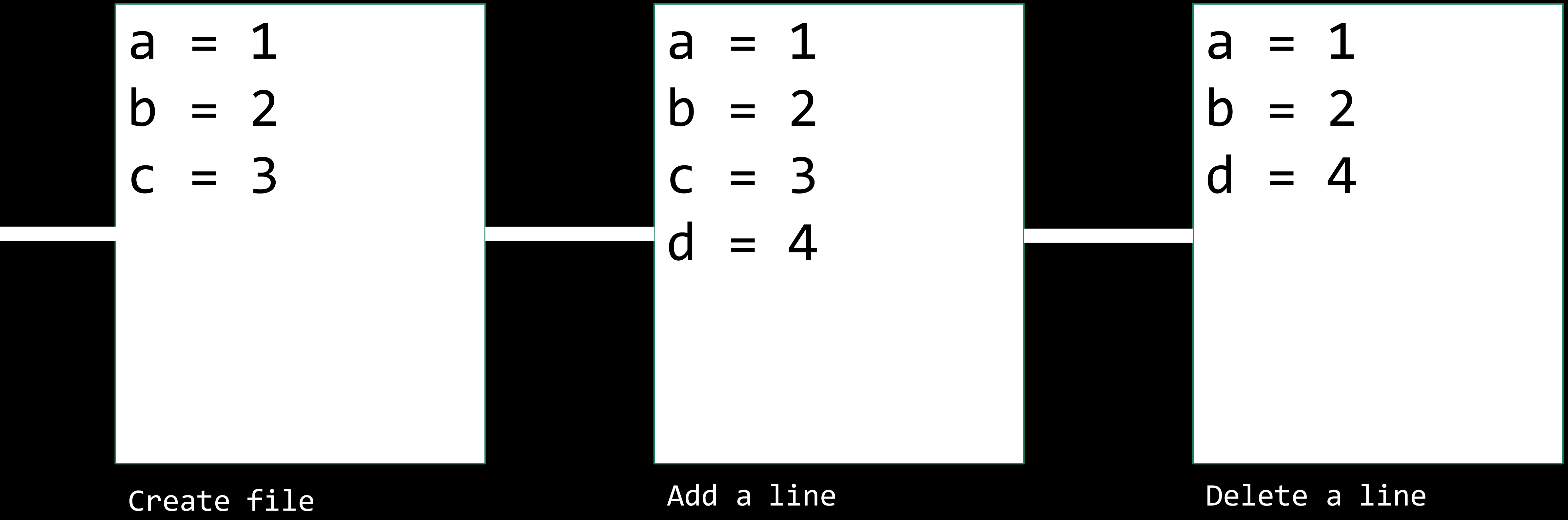
```
a = 1
b = 2
c = 3
d = 4
```

Git

- We can go back in time to prior versions of our code.

Git

- We can go back in time to prior versions of our code.



```
a = 1
b = 2
c = 3
```

Create file

```
a = 1
b = 2
c = 3
d = 4
```

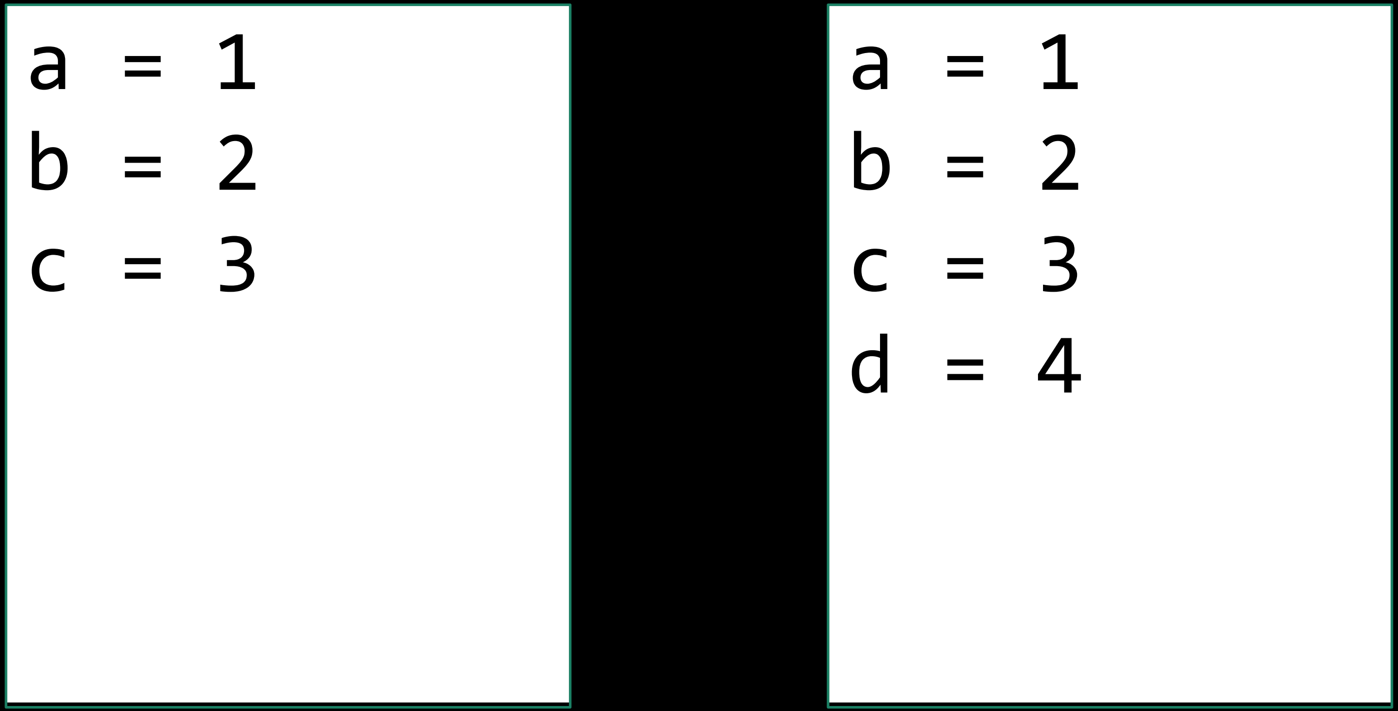
Add a line

```
a = 1
b = 2
d = 4
```

Delete a line

Git

- We can go back in time to prior versions of our code.



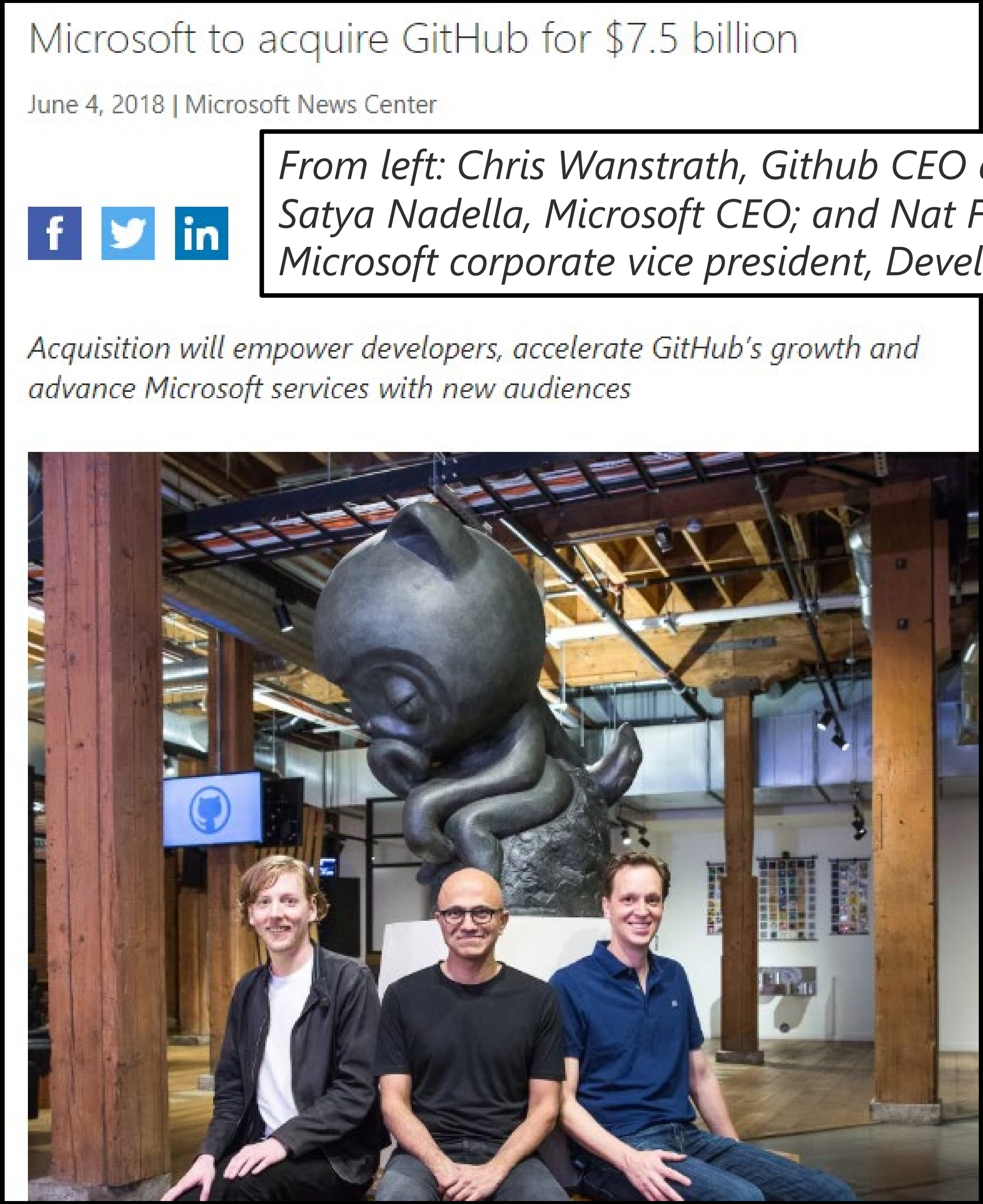
```
a = 1
b = 2
c = 3
```

Create file

```
a = 1
b = 2
c = 3
d = 4
```

Add a line

GitHub



Microsoft to acquire GitHub for $7.5 billion

June 4, 2018 | Microsoft News Center

Acquisition will empower developers, accelerate GitHub's growth and advance Microsoft services with new audiences

From left: Chris Wanstrath, Github CEO and co-founder; Satya Nadella, Microsoft CEO; and Nat Friedman, Microsoft corporate vice president, Developer Services

GitHub

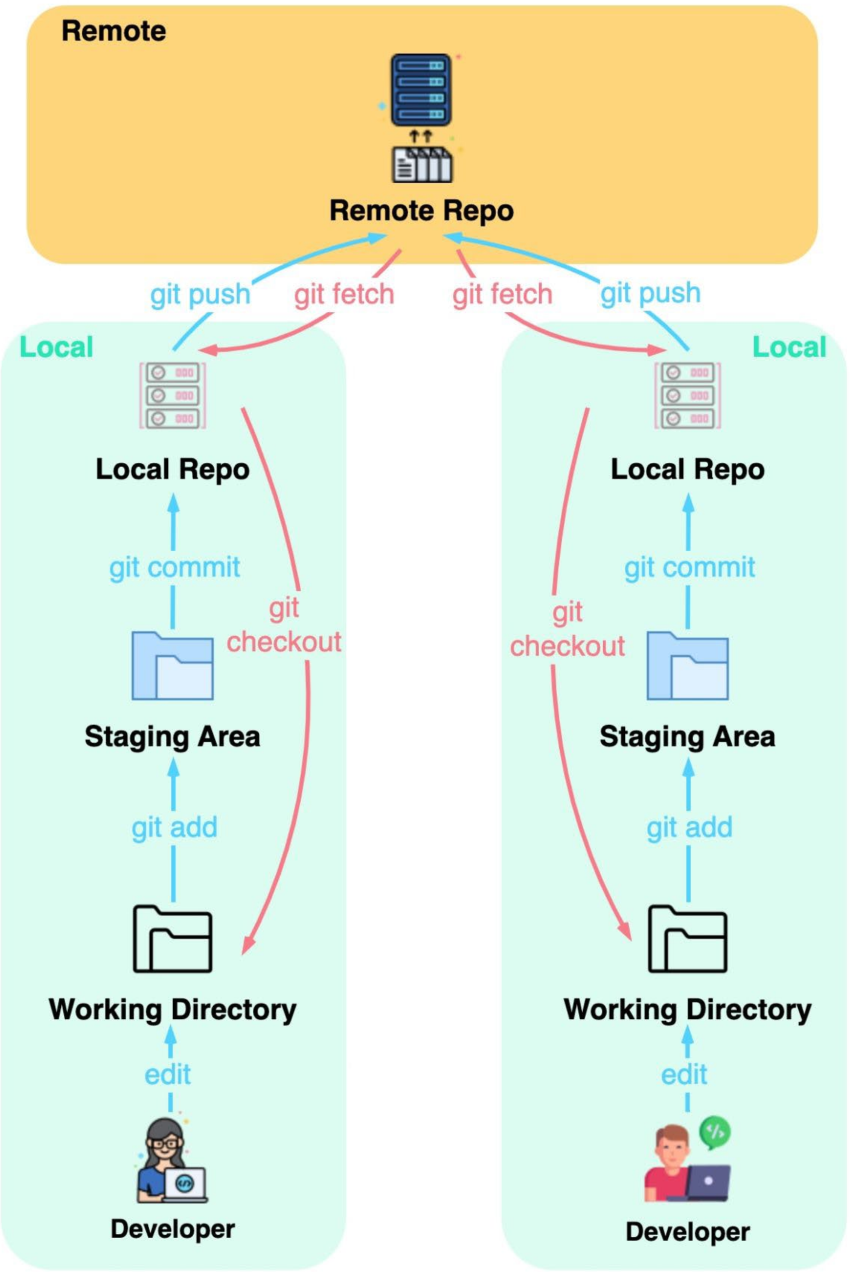
- A central hub on the internet for you to store, view, and contribute to your own and others’ repositories.

- A “repository” is Git/GitHub terminology for a collection of related code.

- You will likely find it easiest to first create a repository on GitHub’s web interface, before working on it at the command line.

How does Git Work?

blog.bytebytego.com

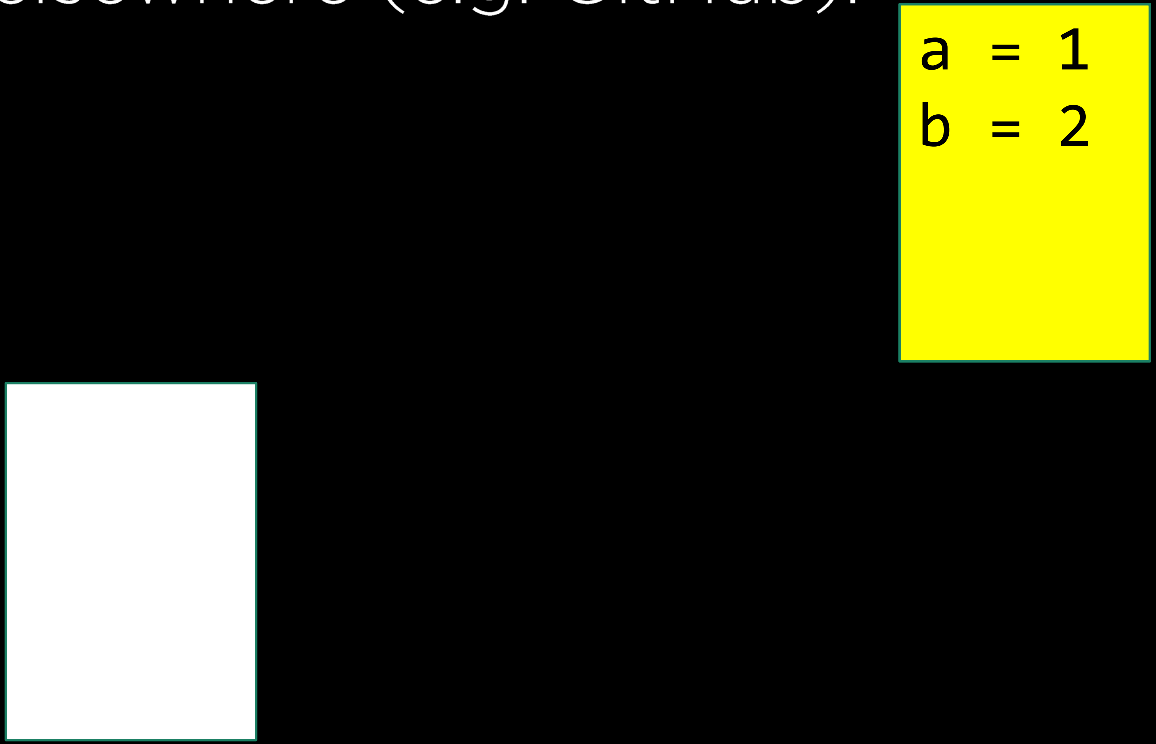


git clone

- Used to initially get a local copy of a repository stored elsewhere (e.g. GitHub).

git clone

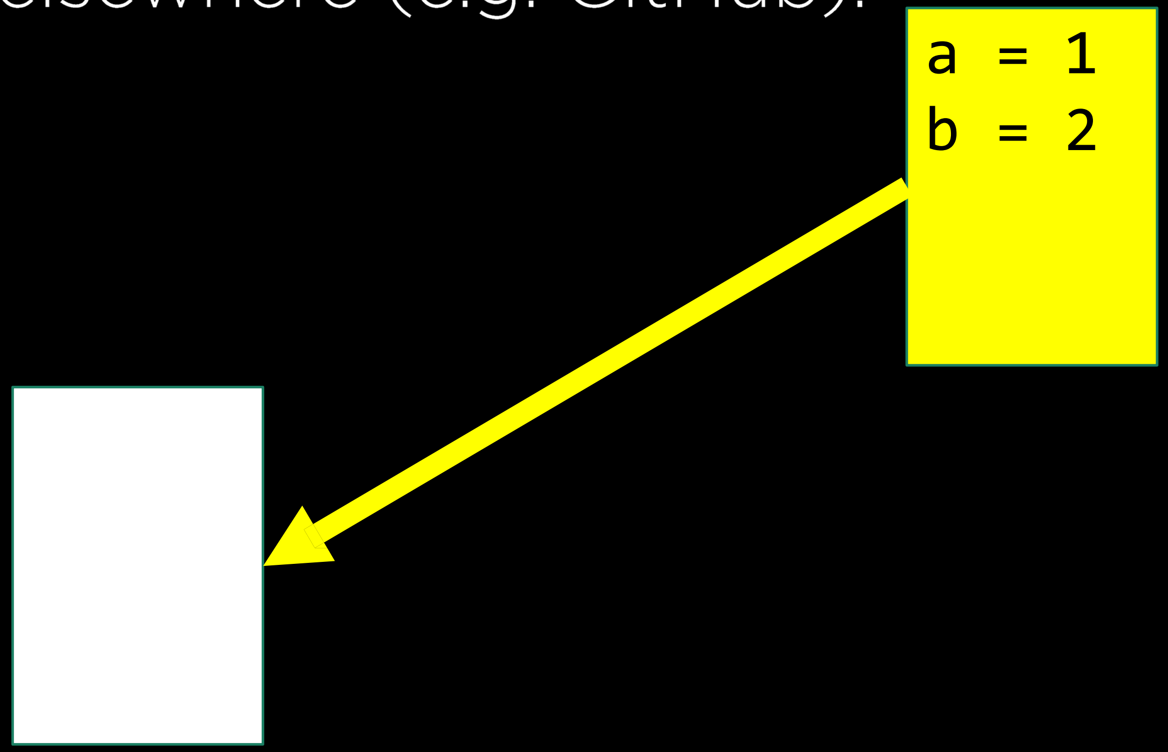
- Used to initially get a local copy of a repository stored elsewhere (e.g. GitHub).



```
a = 1
b = 2
```

git clone

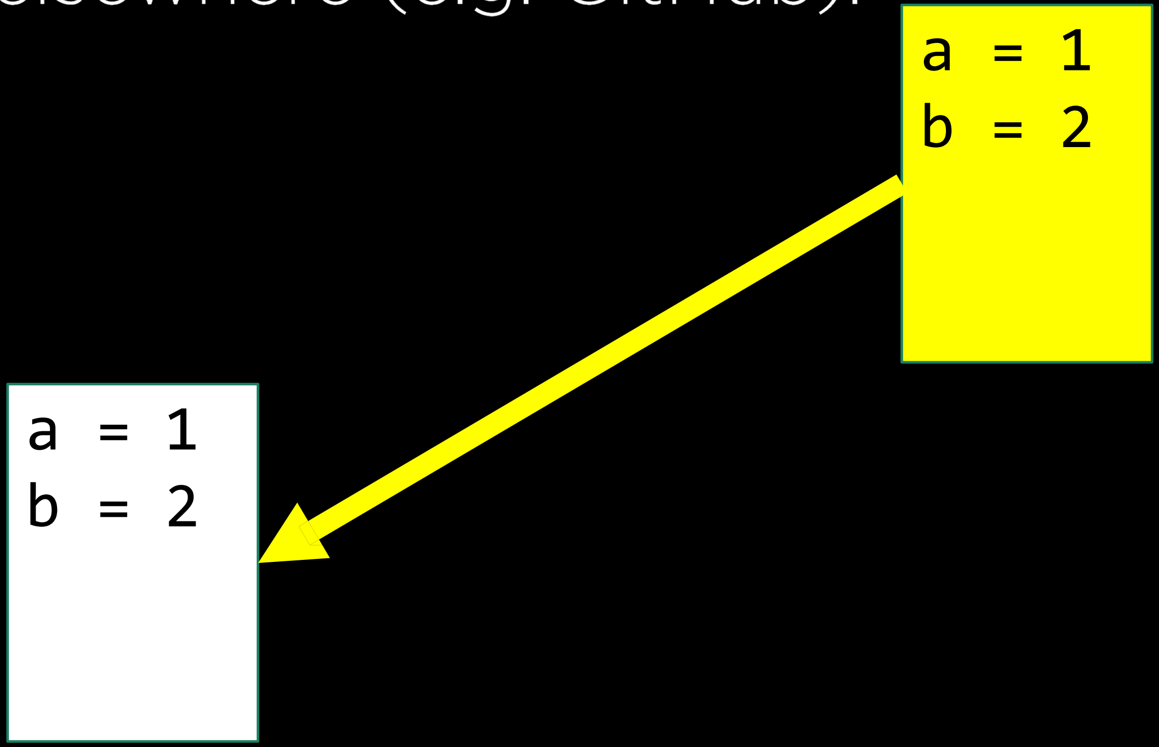
- Used to initially get a local copy of a repository stored elsewhere (e.g. GitHub).



```
a = 1
b = 2
```

`git clone`

- Used to initially get a local copy of a repository stored elsewhere (e.g. GitHub).



```
a = 1
b = 2
```

```
a = 1
b = 2
```

git clone

- Used to initially get a local copy of a repository stored elsewhere (e.g. GitHub).

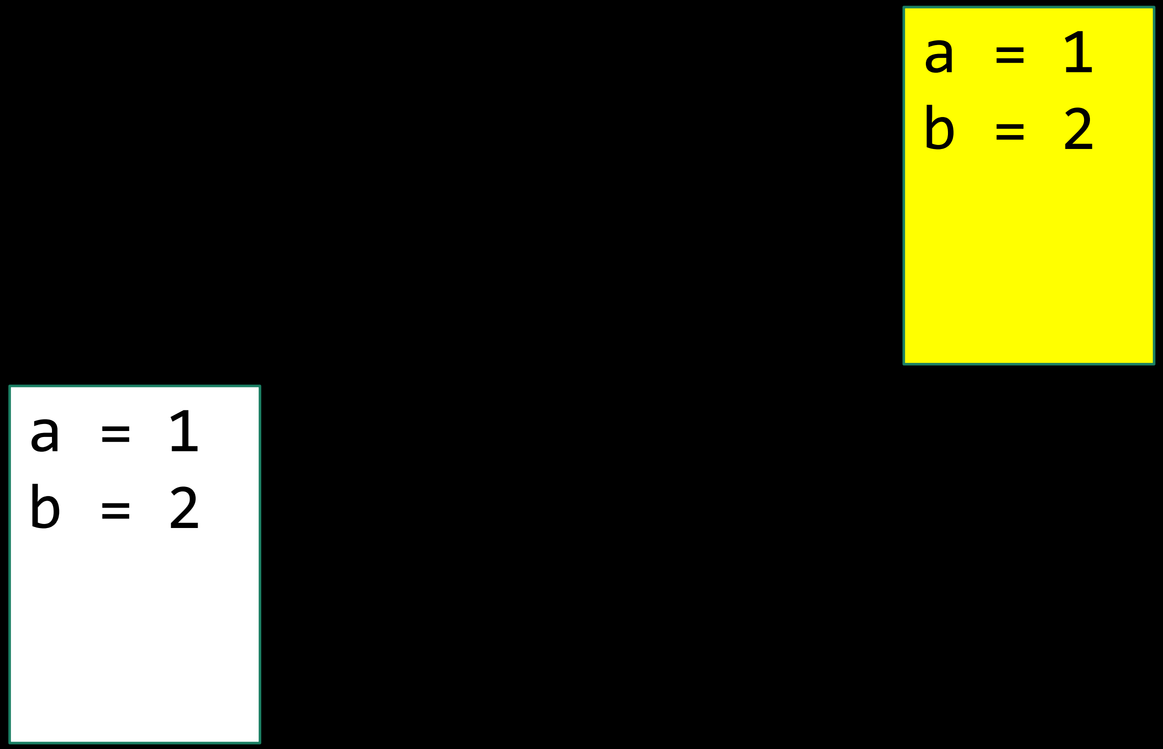
```
git clone <url>
```

`git add`

- Used to let Git know which files we want it to look at when we prepare our next “snapshot” of the repository.

git add

- Used to let Git know which files we want it to look at when we prepare our next “snapshot” of the repository.

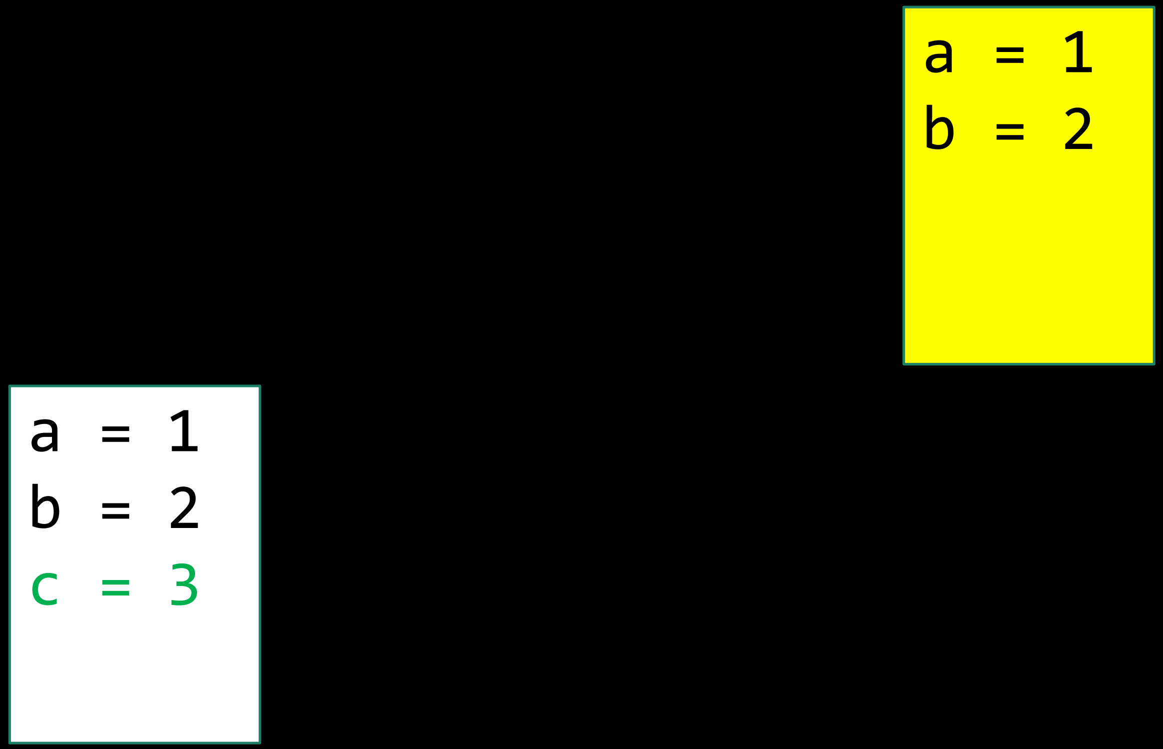


```
a = 1
b = 2
```

```
a = 1
b = 2
```

git add

- Used to let Git know which files we want it to look at when we prepare our next “snapshot” of the repository.

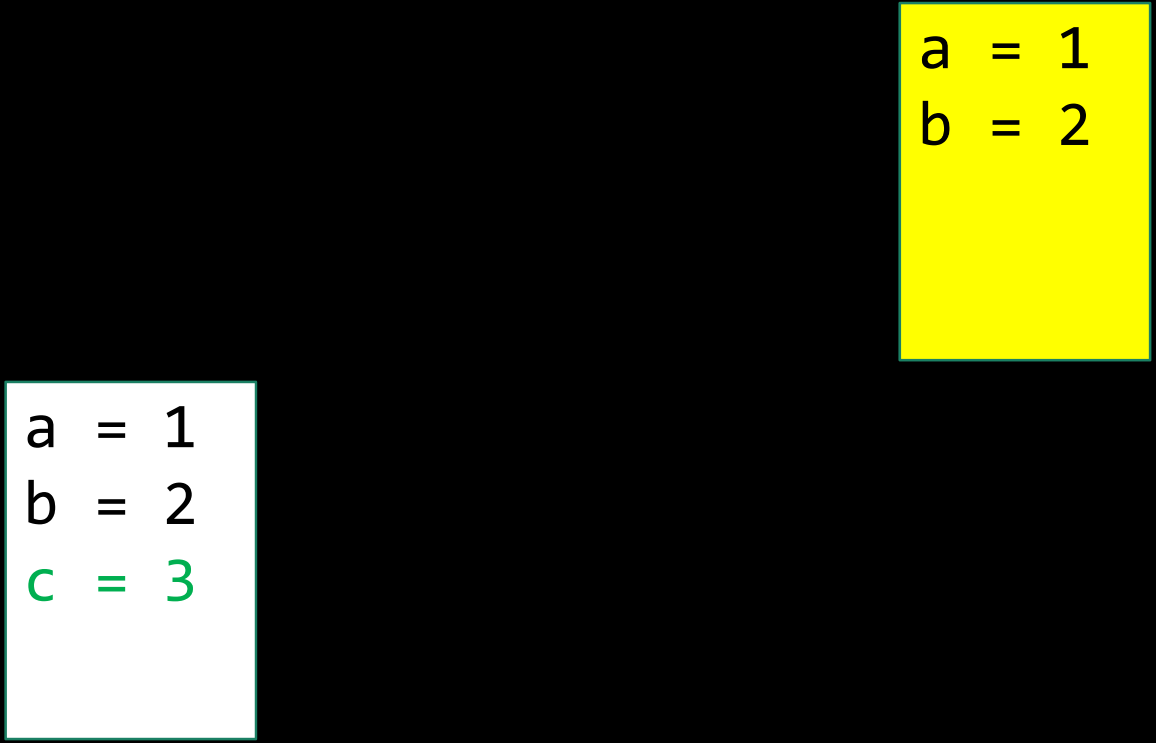


```
a = 1
b = 2
```

```
a = 1
b = 2
c = 3
```

git add

- Used to let Git know which files we want it to look at when we prepare our next “snapshot” of the repository.

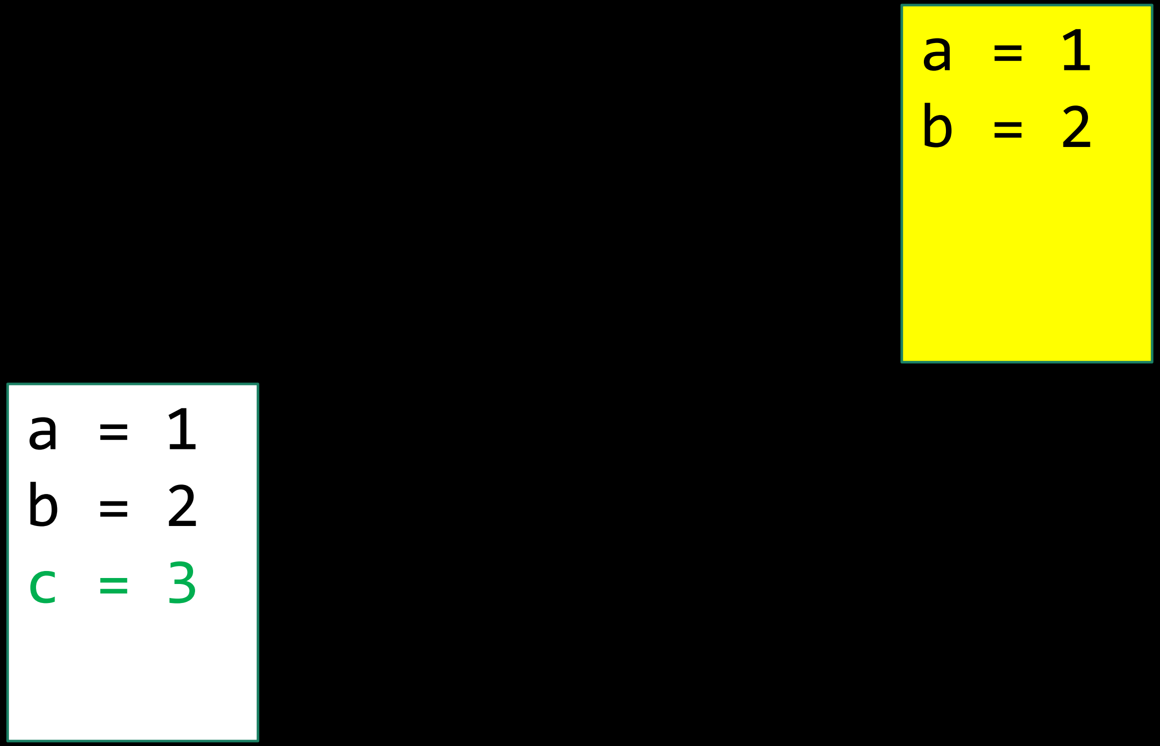


```
a = 1
b = 2
```

```
a = 1
b = 2
c = 3
```

git add

- Used to let Git know which files we want it to look at when we prepare our next “snapshot” of the repository.

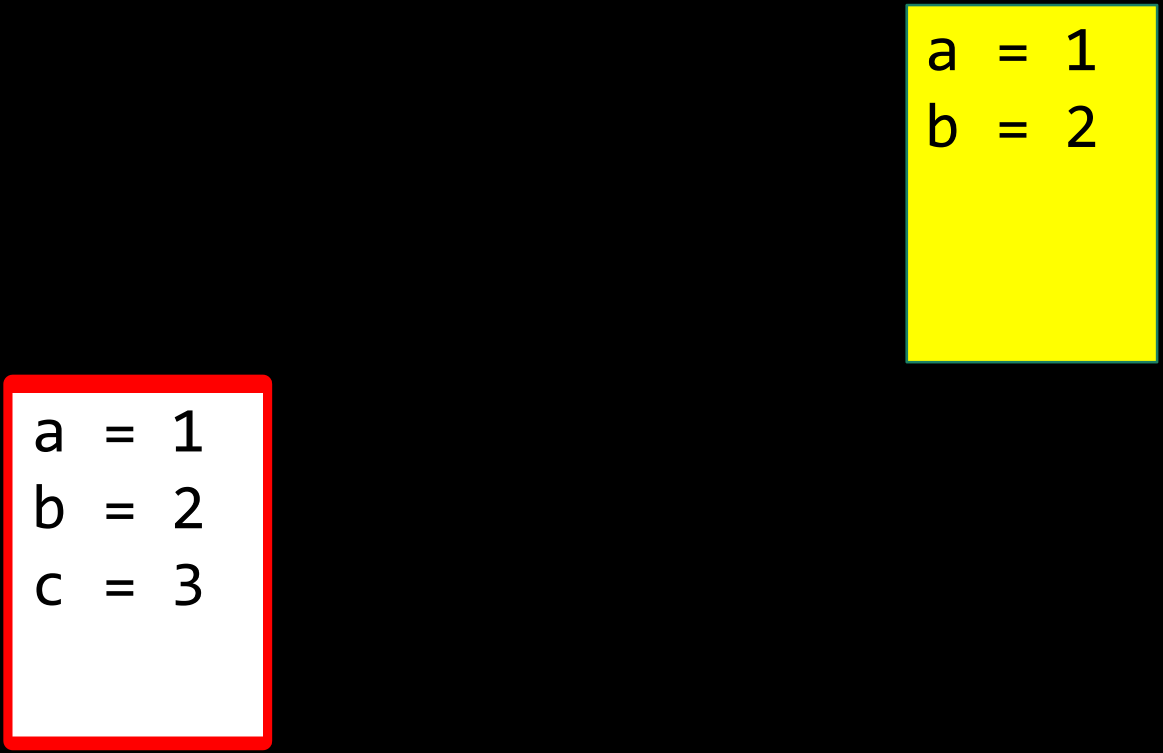


```
a = 1
b = 2
```

```
a = 1
b = 2
c = 3
```

git add

- Used to let Git know which files we want it to look at when we prepare our next “snapshot” of the repository.



```
a = 1
b = 2
```

```
a = 1
b = 2
c = 3
```

git add

- Used to let Git know which files we want it to look at when we prepare our next “snapshot” of the repository.

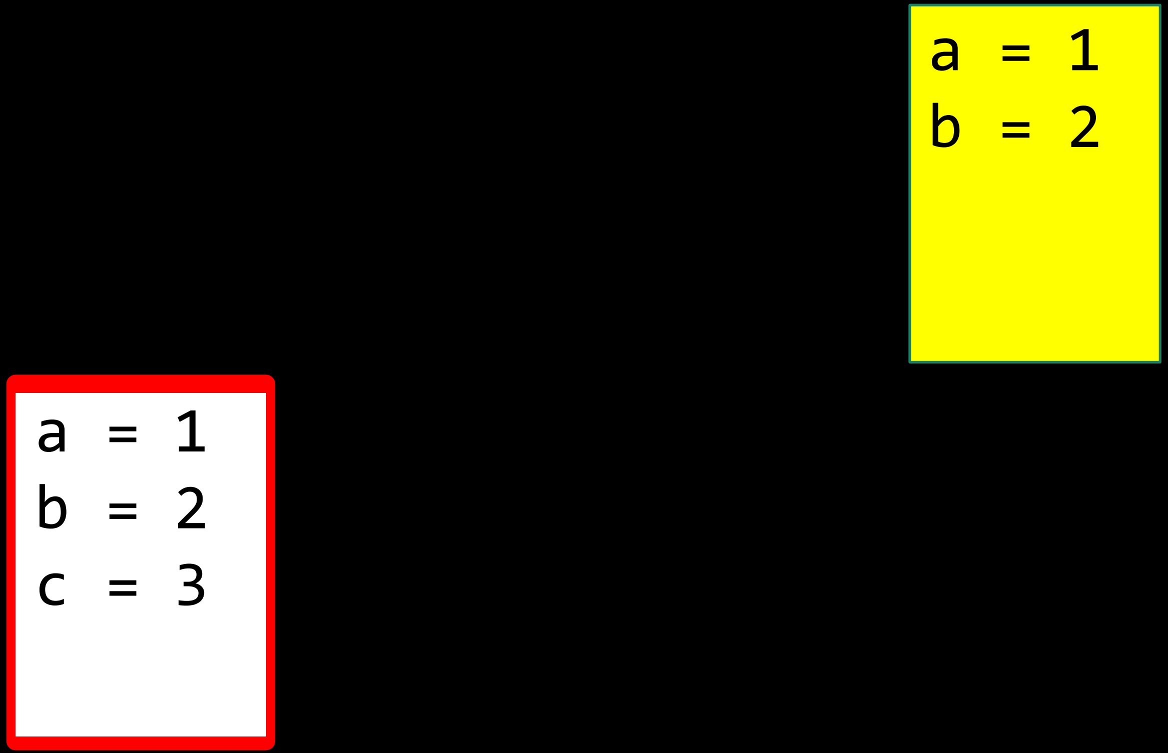
```
git add <filename>
git add -A
```

git status

- Used to let you know the current state of your directory (i.e., what’s changed)

git commit

- Used to tell Git that it’s time to take that next “snapshot” of the repository.

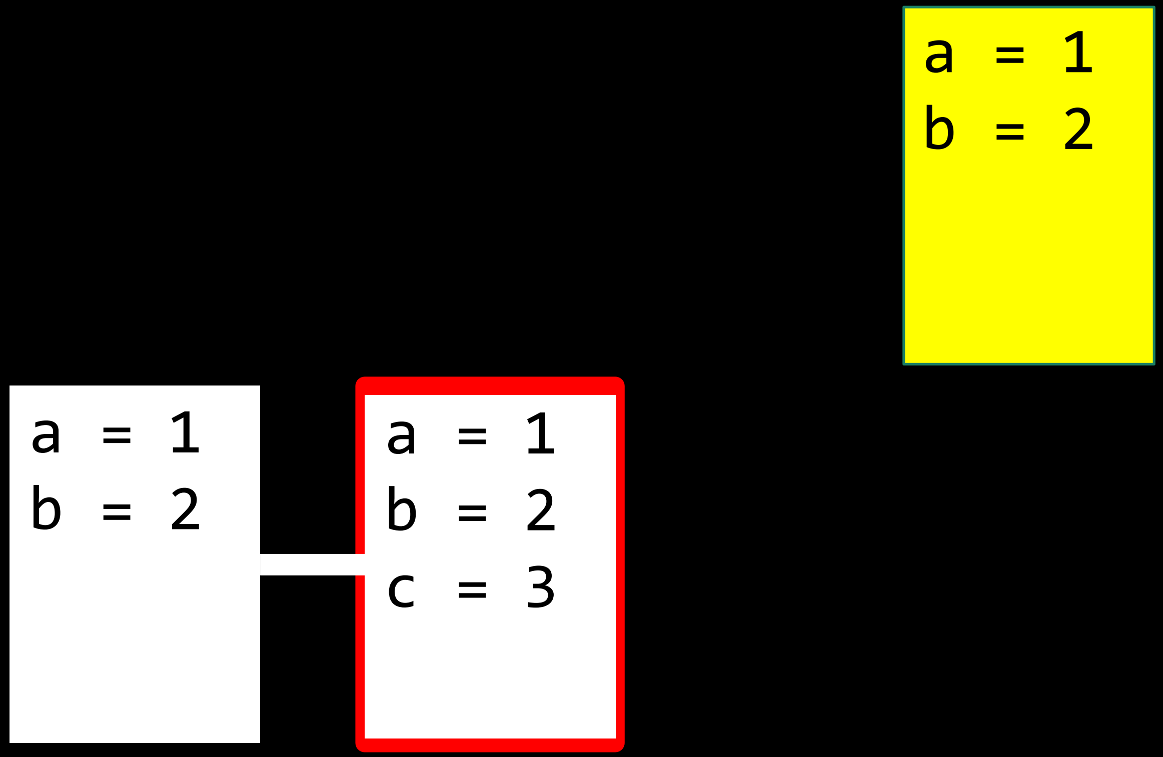


```
a = 1
b = 2
```

```
a = 1
b = 2
c = 3
```

git commit

- Used to tell Git that it’s time to take that next “snapshot” of the repository.



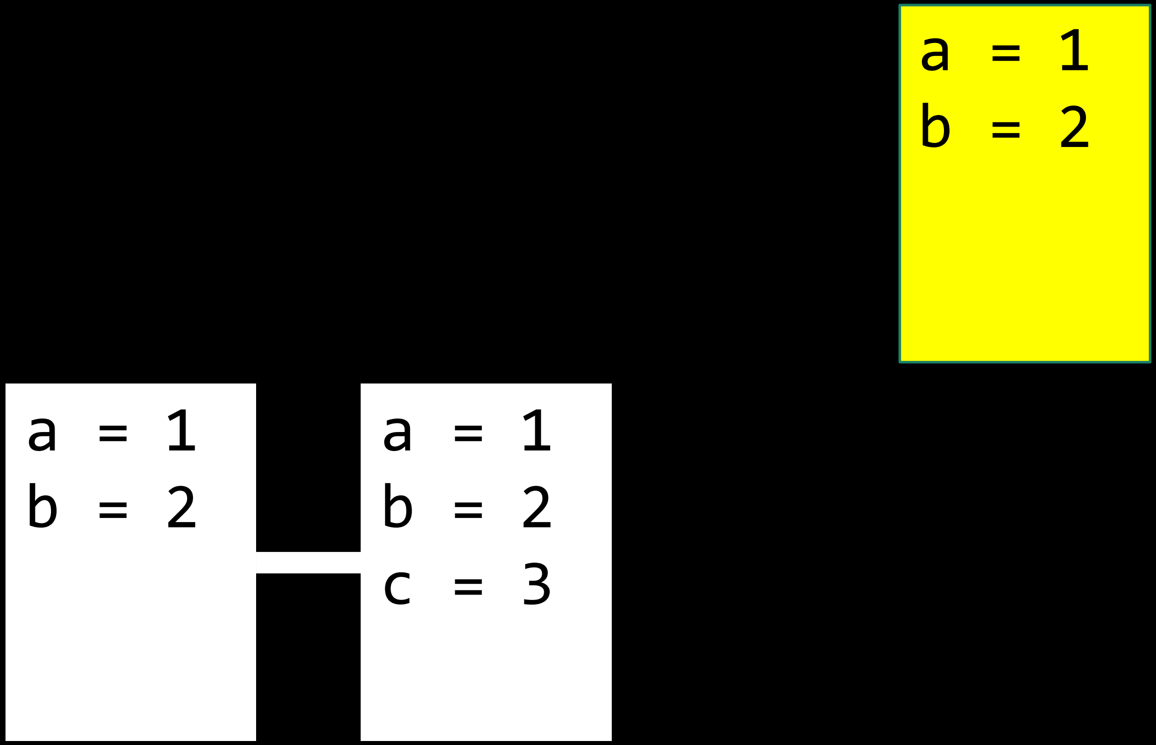
```
a = 1
b = 2
```

```
a = 1
b = 2
```

```
a = 1
b = 2
c = 3
```

git commit

- Used to tell Git that it’s time to take that next “snapshot” of the repository.



```
a = 1
b = 2
```

```
a = 1
b = 2
```

```
a = 1
b = 2
c = 3
```

Add a line

git commit

- Used to tell Git that it’s time to take that next “snapshot” of the repository.

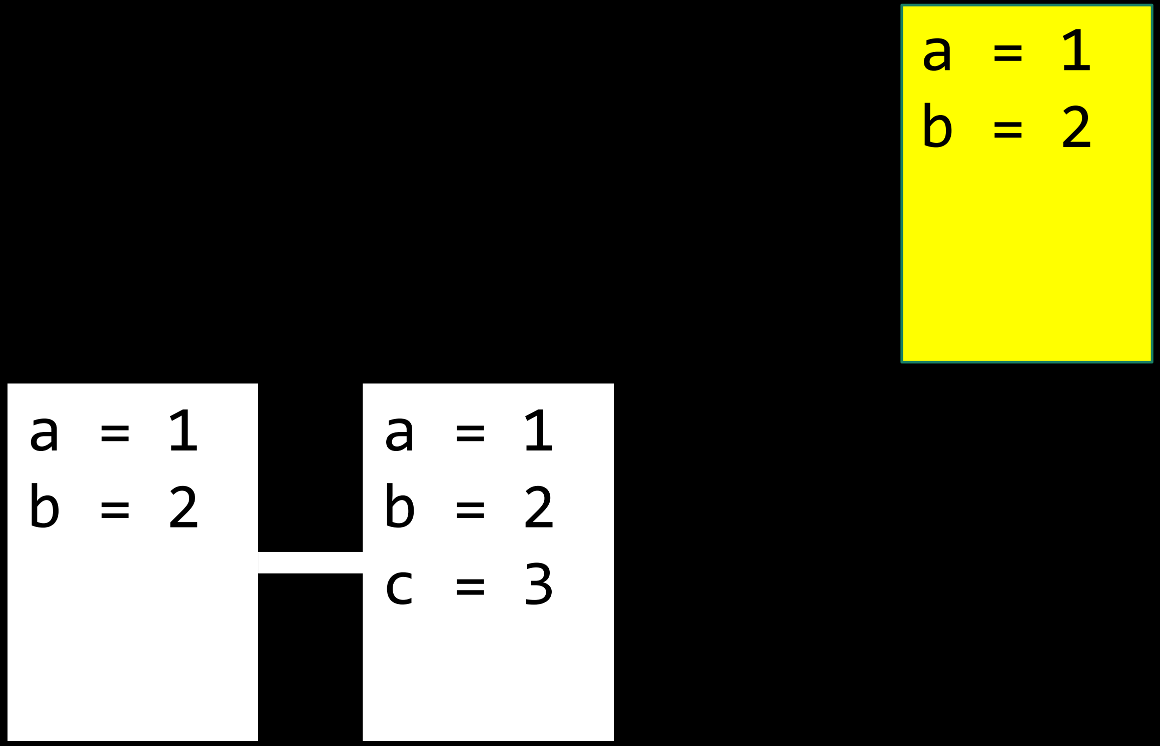
```
git commit –m "message"
git commit –m "Add a line"
```

git push

- Used to take all of the snapshots and changes made locally and send them to your “remote” (i.e., GitHub).

git push

- Used to take all of the snapshots and changes made locally and send them to your “remote” (i.e., GitHub).



```
a = 1
b = 2
```

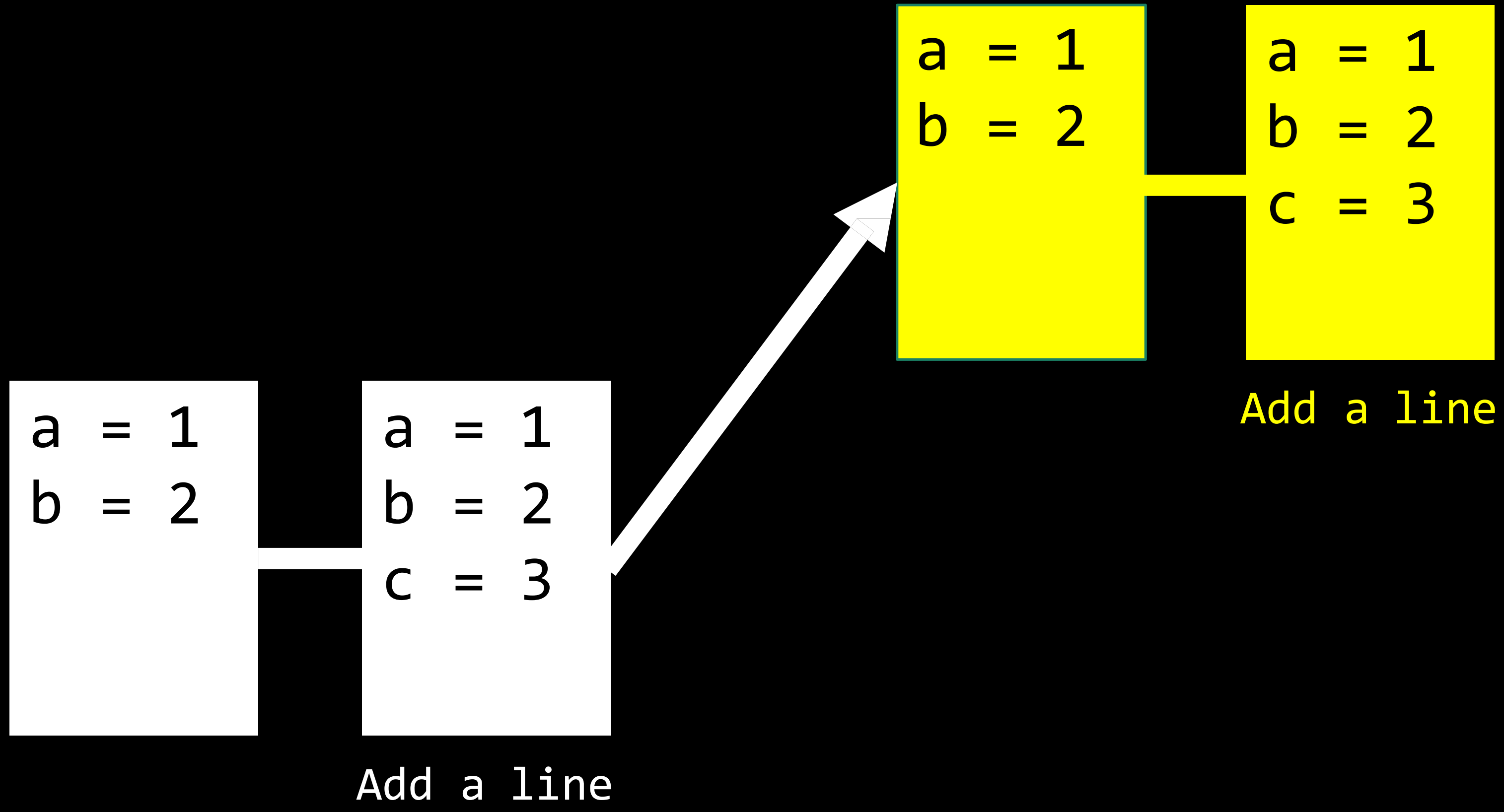
```
a = 1
b = 2
```

```
a = 1
b = 2
c = 3
```

Add a line

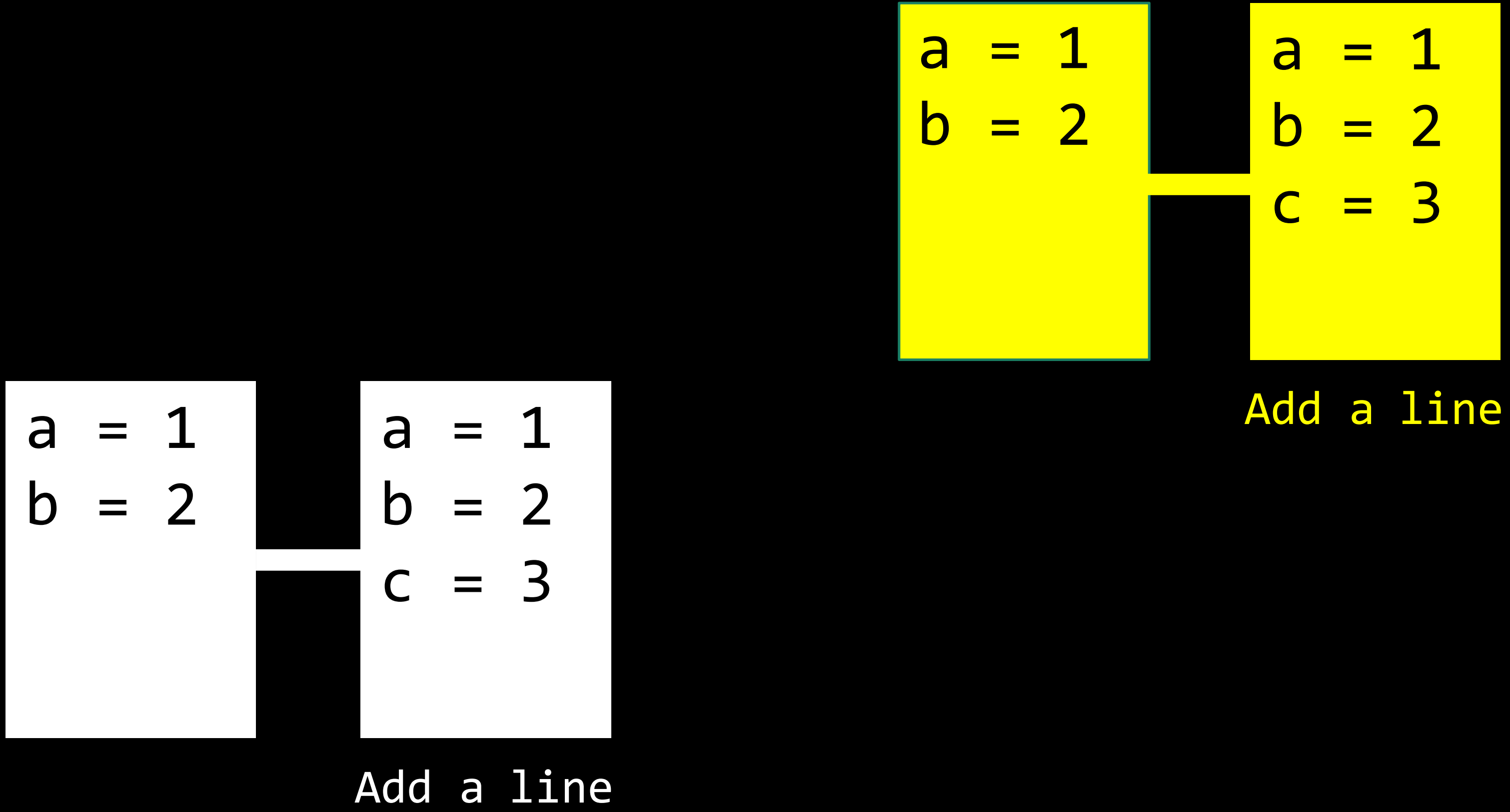
git push

- Used to take all of the snapshots and changes made locally and send them to your "remote" (i.e., GitHub).



git push

- Used to take all of the snapshots and changes made locally and send them to your “remote” (i.e., GitHub).



```
a = 1
b = 2
```

```
a = 1
b = 2
c = 3
```

Add a line

```
a = 1
b = 2
```

```
a = 1
b = 2
c = 3
```

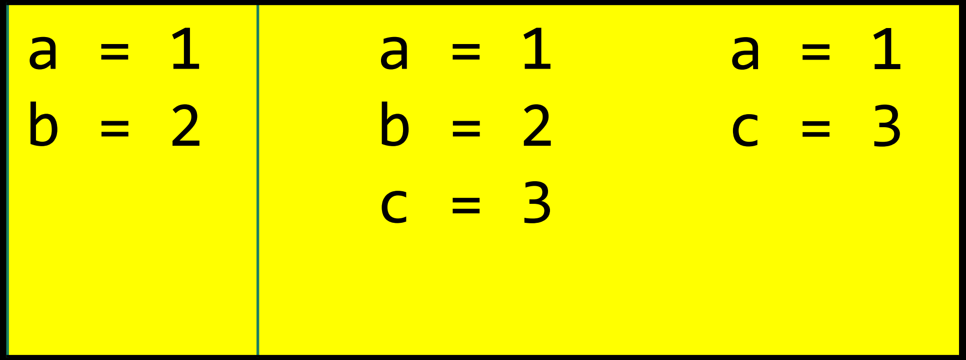
Add a line

git pull

- Used to get all of the snapshots and changes that are stored on your “remote” (i.e., GitHub) onto your local machine.

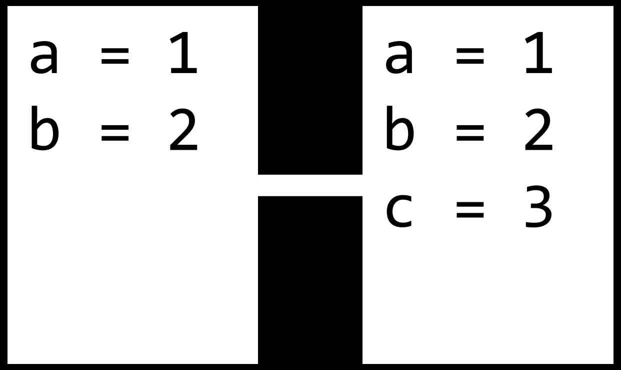
git pull

- Used to get all of the snapshots and changes that are stored on your “remote” (i.e., GitHub) onto your local machine.



```
a = 1        a = 1        a = 1
b = 2        b = 2        c = 3
             c = 3
```

Add a line   Delete line

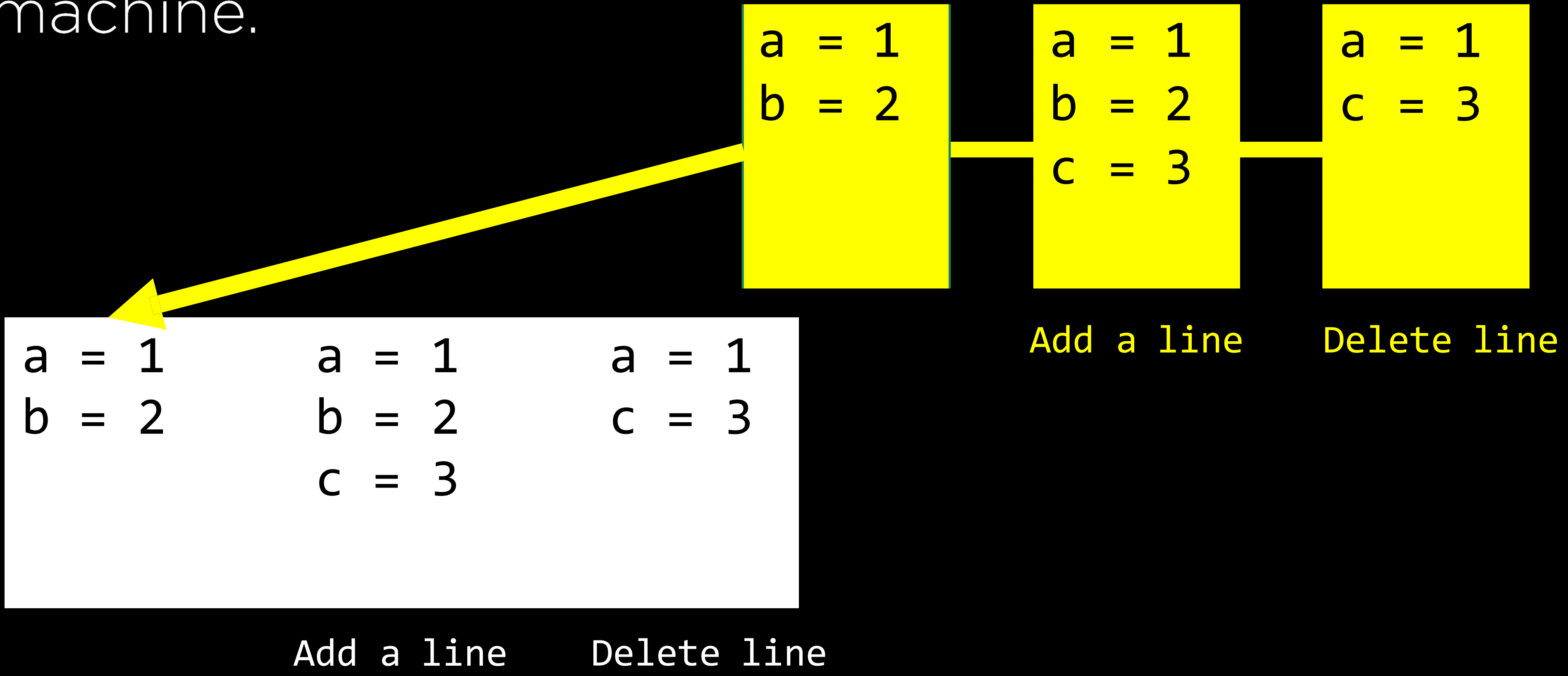


```
a = 1        a = 1
b = 2        b = 2
             c = 3
```

Add a line

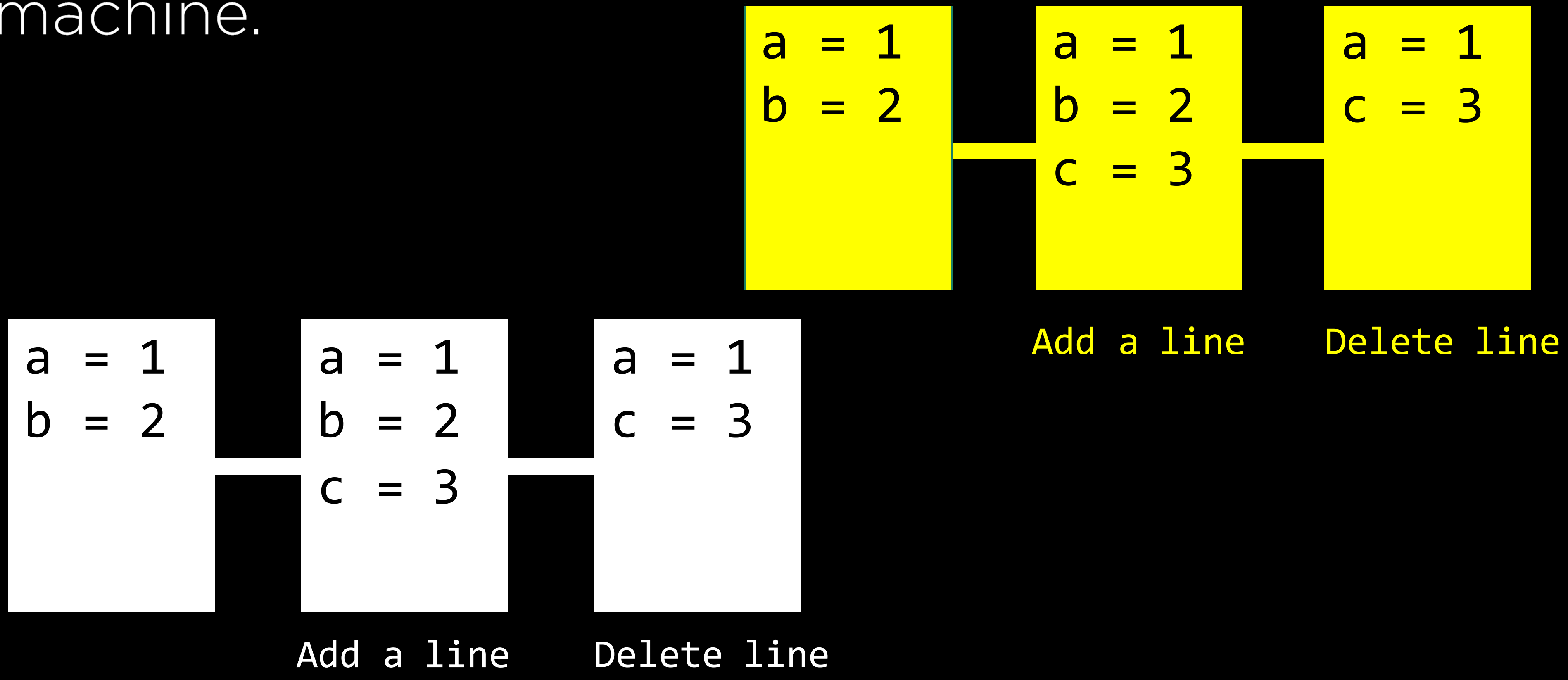
git pull

- Used to get all of the snapshots and changes that are stored on your “remote” (i.e., GitHub) onto your local machine.



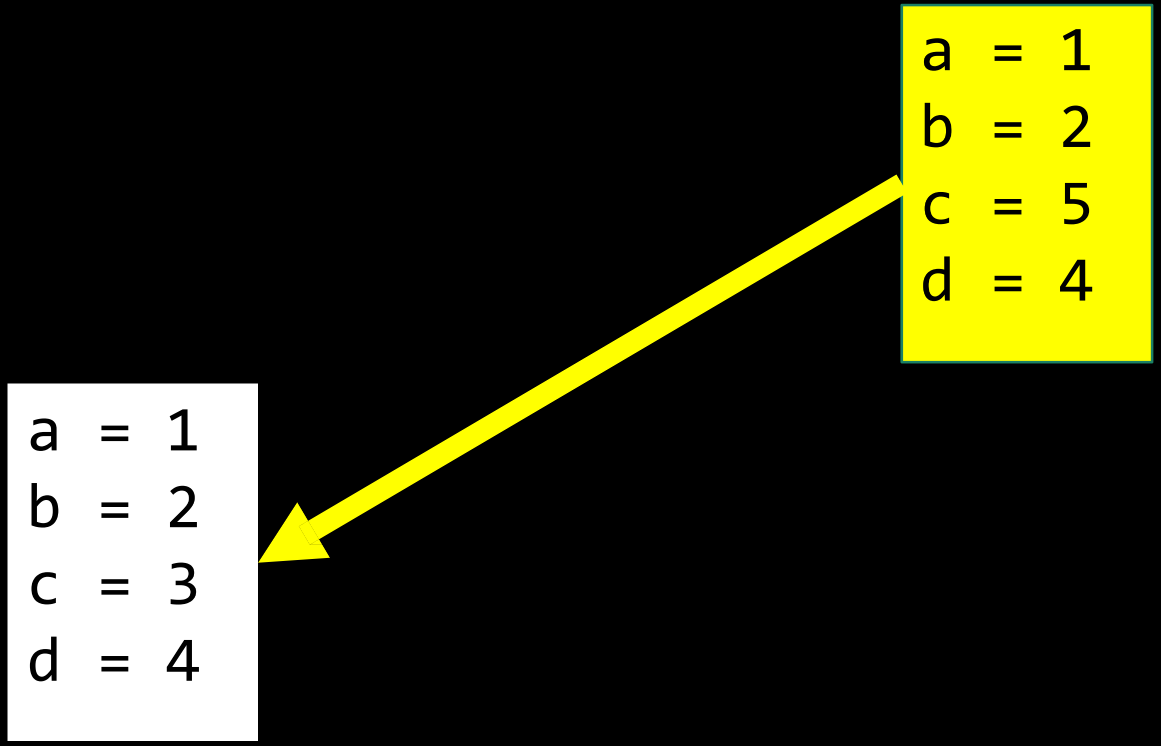
git pull

- Used to get all of the snapshots and changes that are stored on your “remote” (i.e., GitHub) onto your local machine.



Conflicts

- A conflict can occur when locally saved changes are incompatible with changes on the remote.

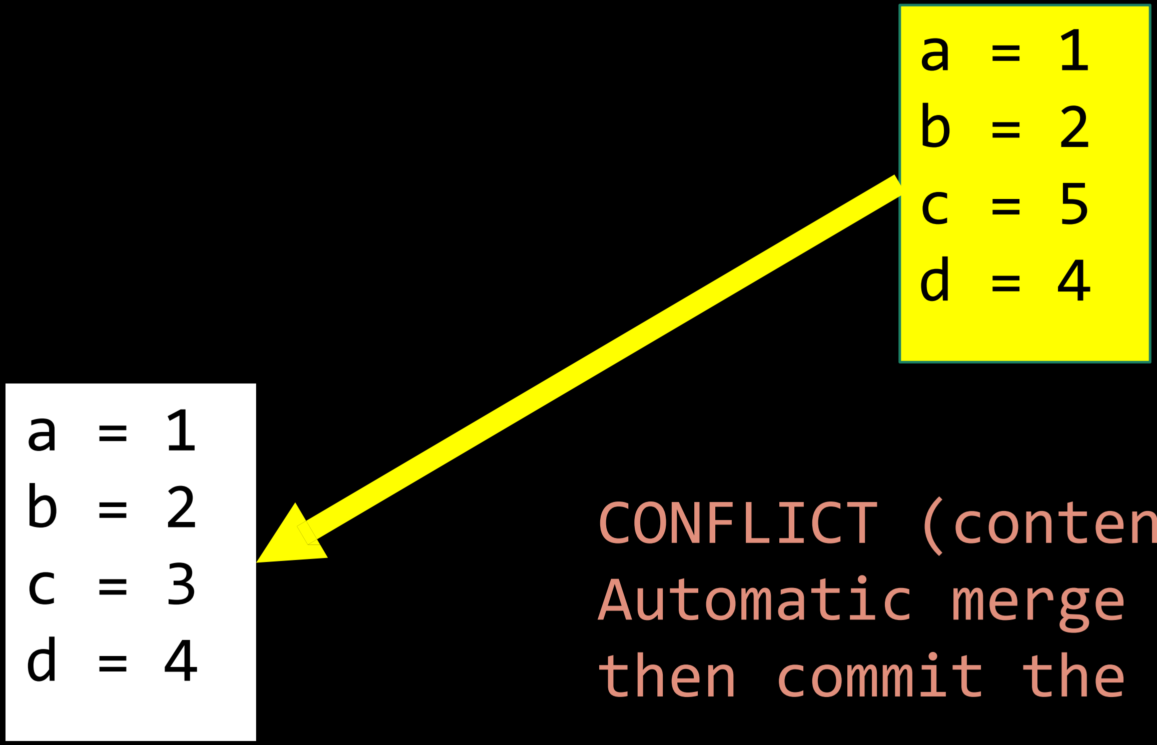


```
a = 1
b = 2
c = 5
d = 4
```

```
a = 1
b = 2
c = 3
d = 4
```

Conflicts

- A conflict can occur when locally saved changes are incompatible with changes on the remote.



```
a = 1
b = 2
c = 5
d = 4
```

```
a = 1
b = 2
c = 3
d = 4
```

```
CONFLICT (content): Merge conflict in foo.py
Automatic merge failed; fix conflicts and
then commit the result.
```

Conflicts

- A conflict can occur when locally saved changes are incompatible with changes on the remote.

```
a = 1
b = 2
<<<<< HEAD
c = 3
=====
c = 5
>>>>> 2828abcdef0123456789
d = 4
```

Conflicts

- A conflict can occur when locally saved changes are incompatible with changes on the remote.

```
a = 1
b = 2
<<<<< HEAD
c = 3
=====
c = 5
>>>>> 2828abcdef0123456789
d = 4
```

Conflicts

- A conflict can occur when locally saved changes are incompatible with changes on the remote.

```
a = 1
b = 2
<<<<< HEAD
c = 3
=====
c = 5
>>>>> 2828abcdef0123456789
d = 4
```

`git log`

- Used to get a history of all of the commits you’ve made locally.

`git reset`

- Used to jump back in time to a prior version of your code (perhaps because the current version is broken).

git reset

- Used to jump back in time to a prior version of your code (perhaps because the current version is broken).

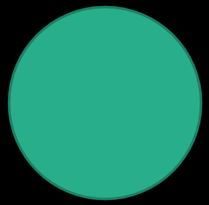
```
git reset --hard <commit>
git reset --hard origin/master
```

Git

- We use Git commits to create a history of how our code evolves.

Git

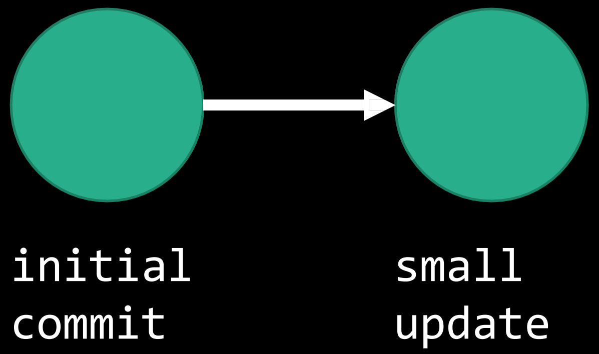
- We use Git commits to create a history of how our code evolves.



```
initial
commit
```

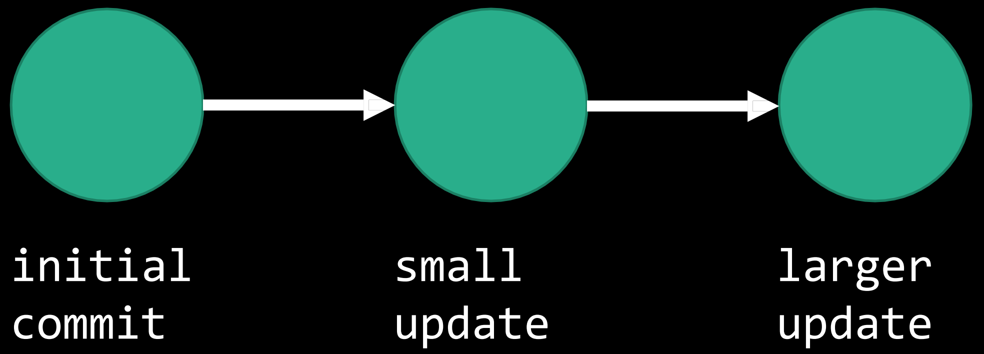
Git

- We use Git commits to create a history of how our code evolves.



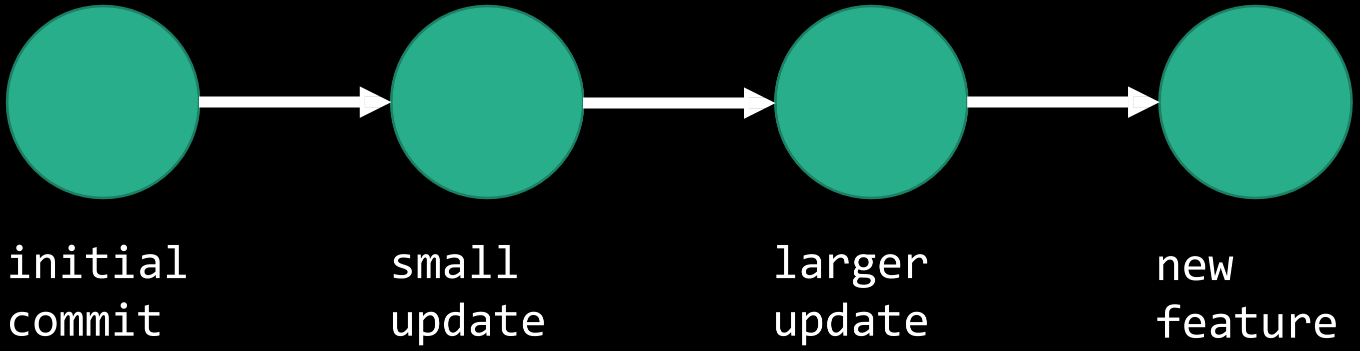
Git

- We use Git commits to create a history of how our code evolves.



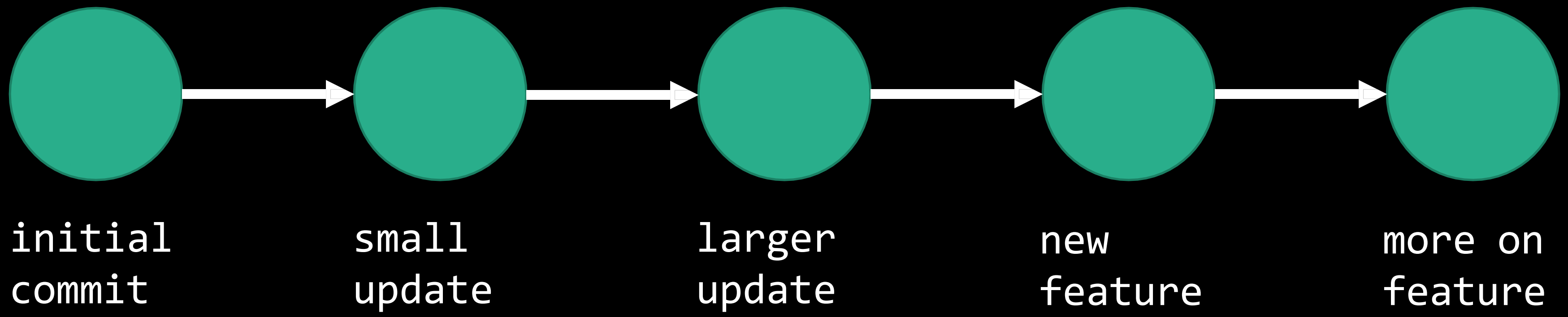
Git

- We use Git commits to create a history of how our code evolves.



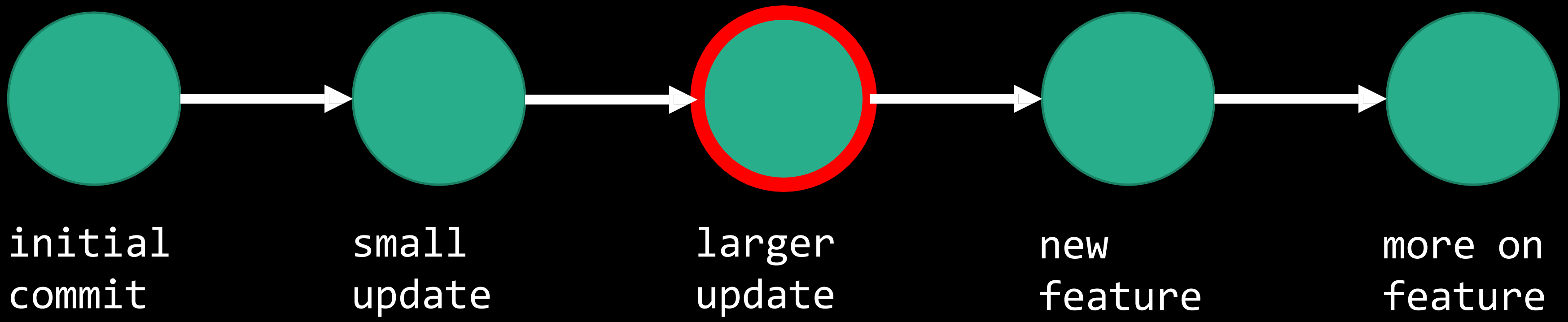
Git

- We use Git commits to create a history of how our code evolves.



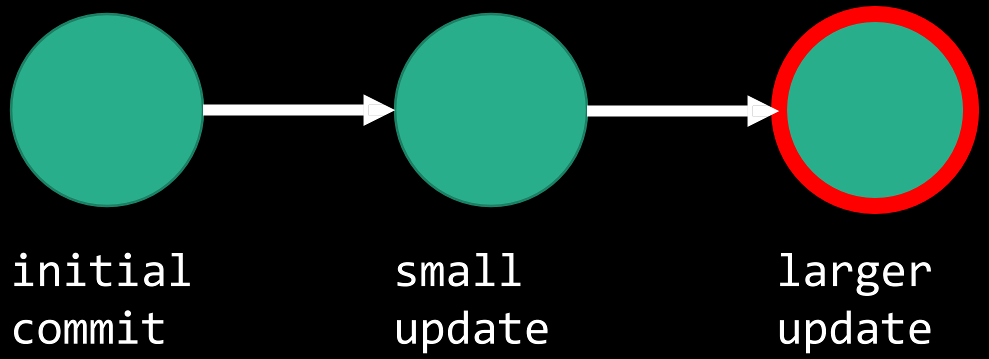
Git

- We use Git commits to create a history of how our code evolves.



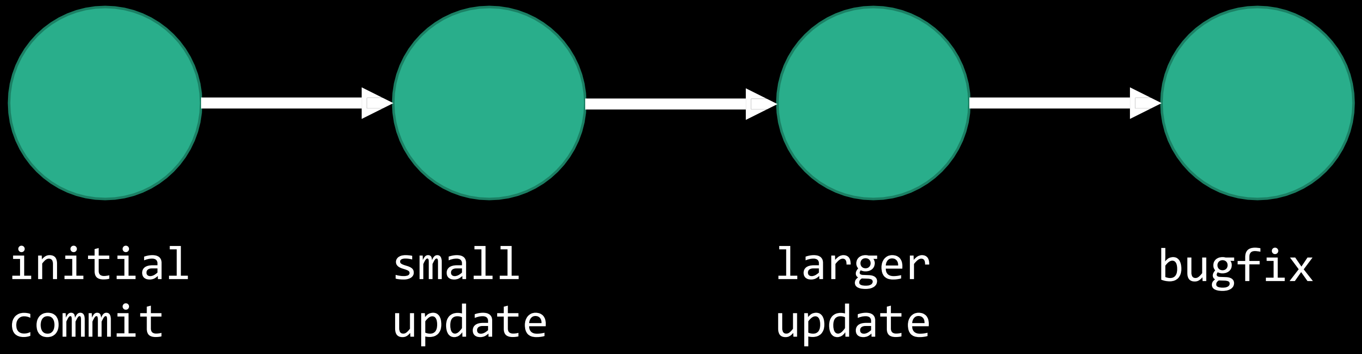
Git

- We use Git commits to create a history of how our code evolves.



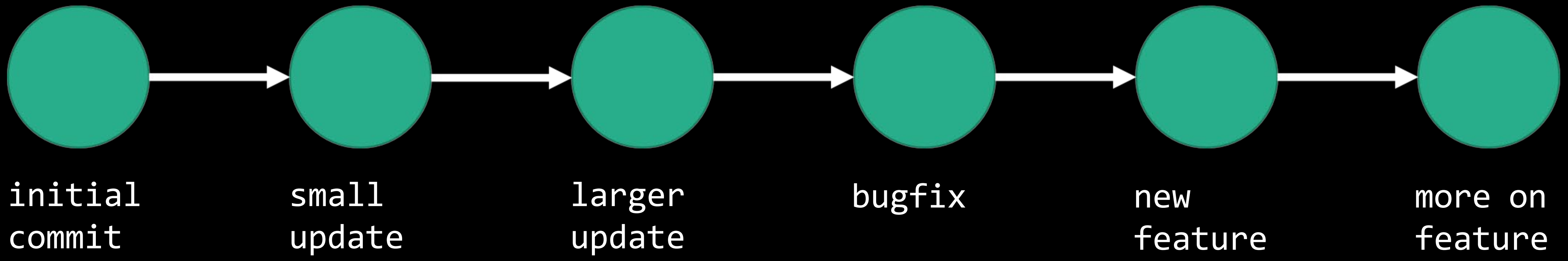
Git

- We use Git commits to create a history of how our code evolves.



Git

- We use Git commits to create a history of how our code evolves.

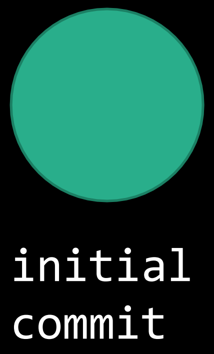


git branch

- Used to get an identical copy of your repository that can be edited and tested separately without impacting production environments.

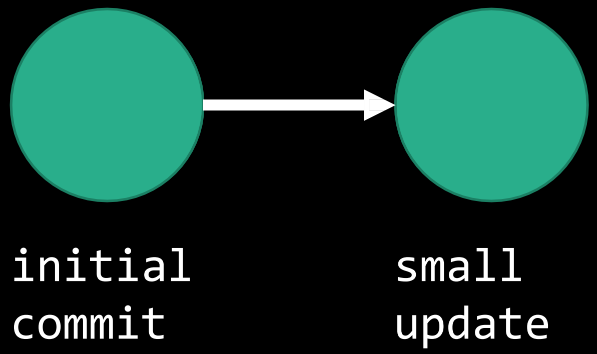
git branch

- Used to get an identical copy of your repository that can be edited and tested separately without impacting production environments.



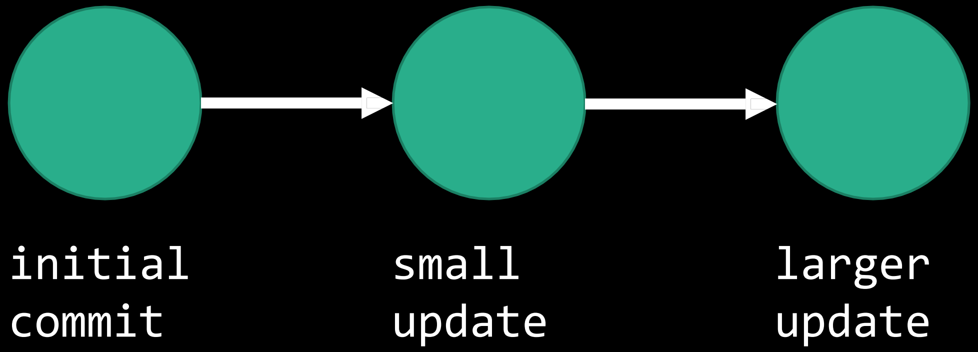
git branch

- Used to get an identical copy of your repository that can be edited and tested separately without impacting production environments.



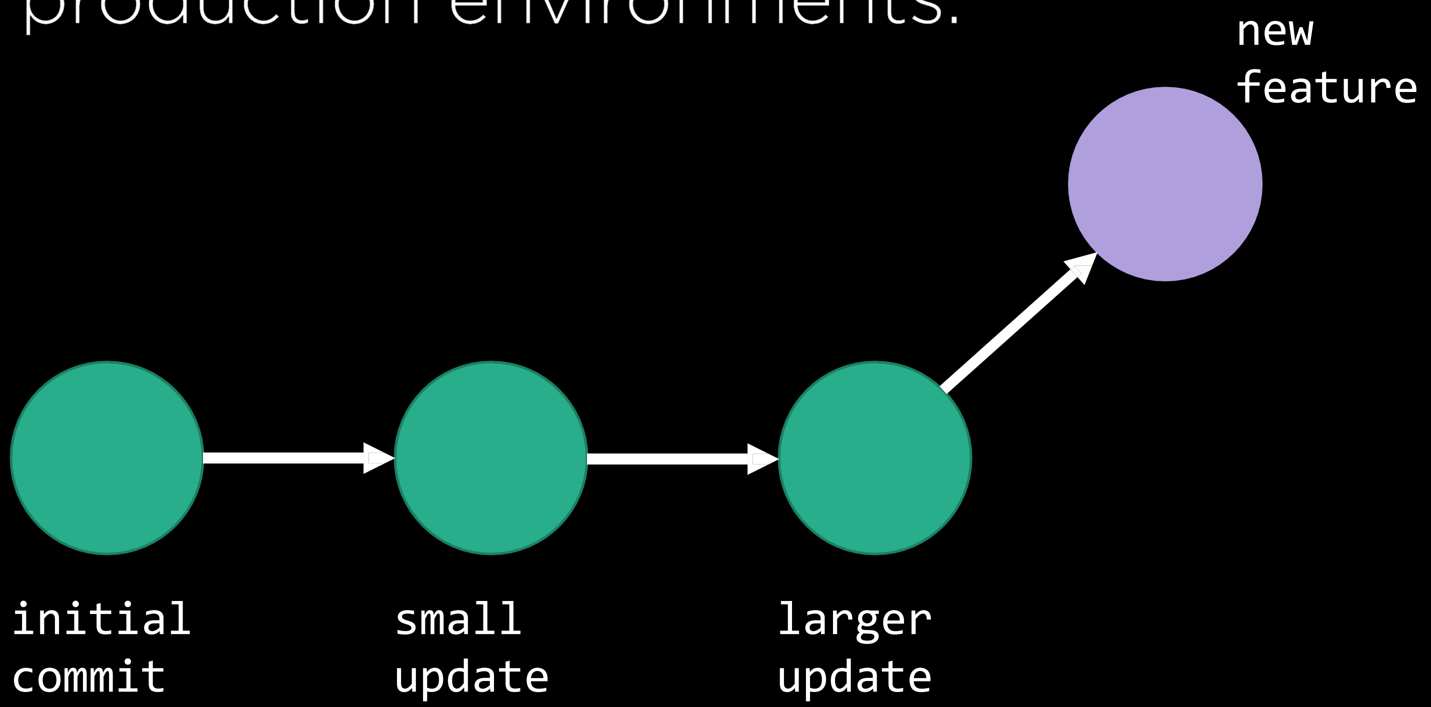
git branch

- Used to get an identical copy of your repository that can be edited and tested separately without impacting production environments.



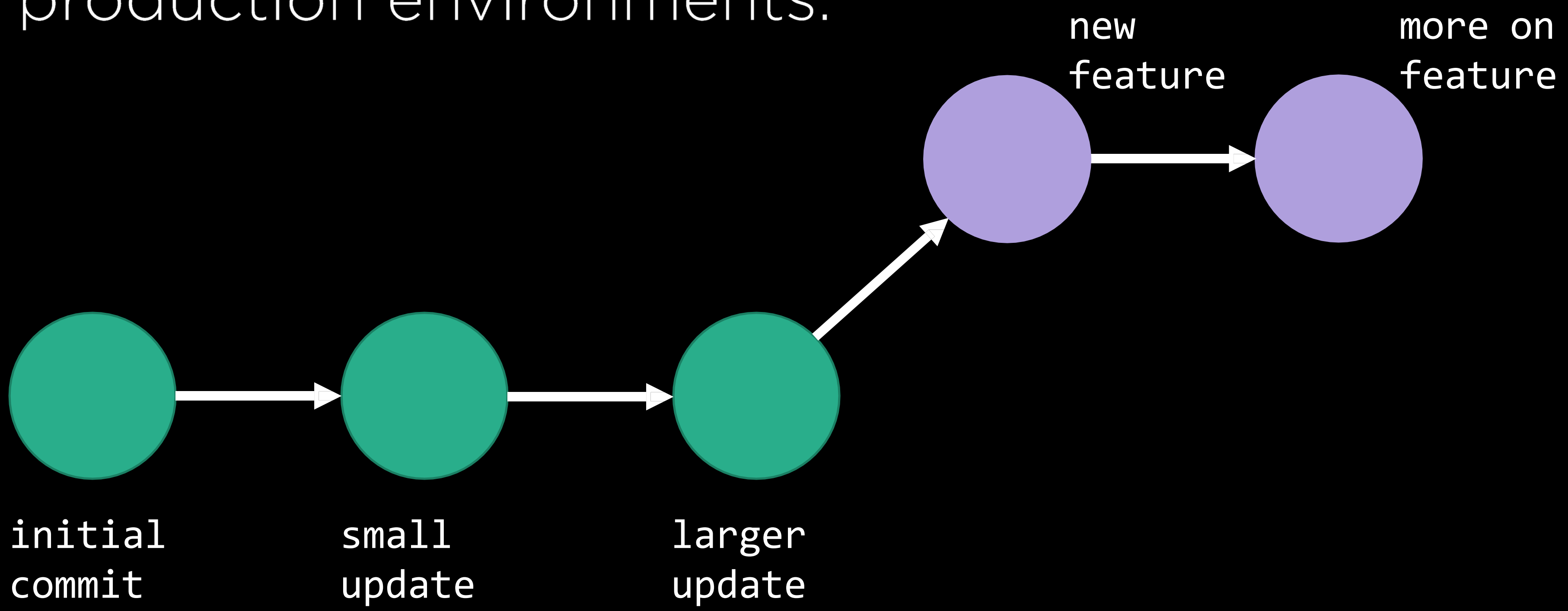
git branch

- Used to get an identical copy of your repository that can be edited and tested separately without impacting production environments.



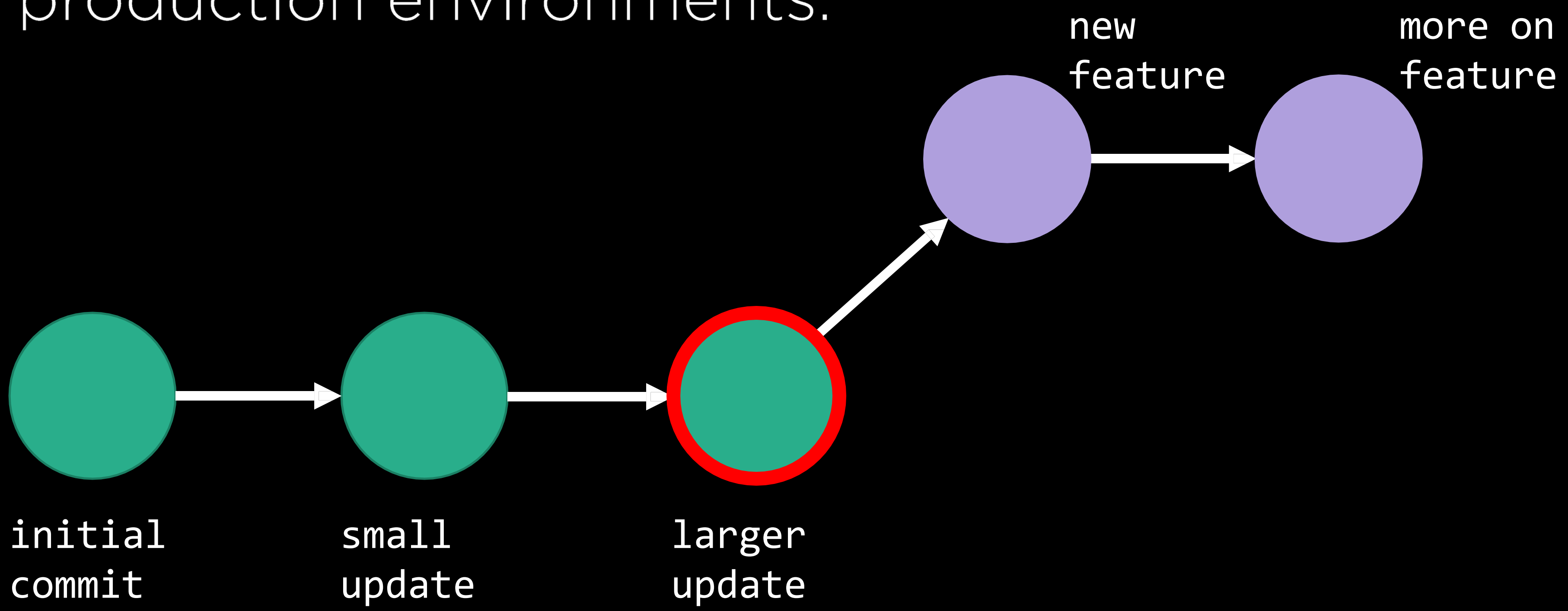
git branch

- Used to get an identical copy of your repository that can be edited and tested separately without impacting production environments.



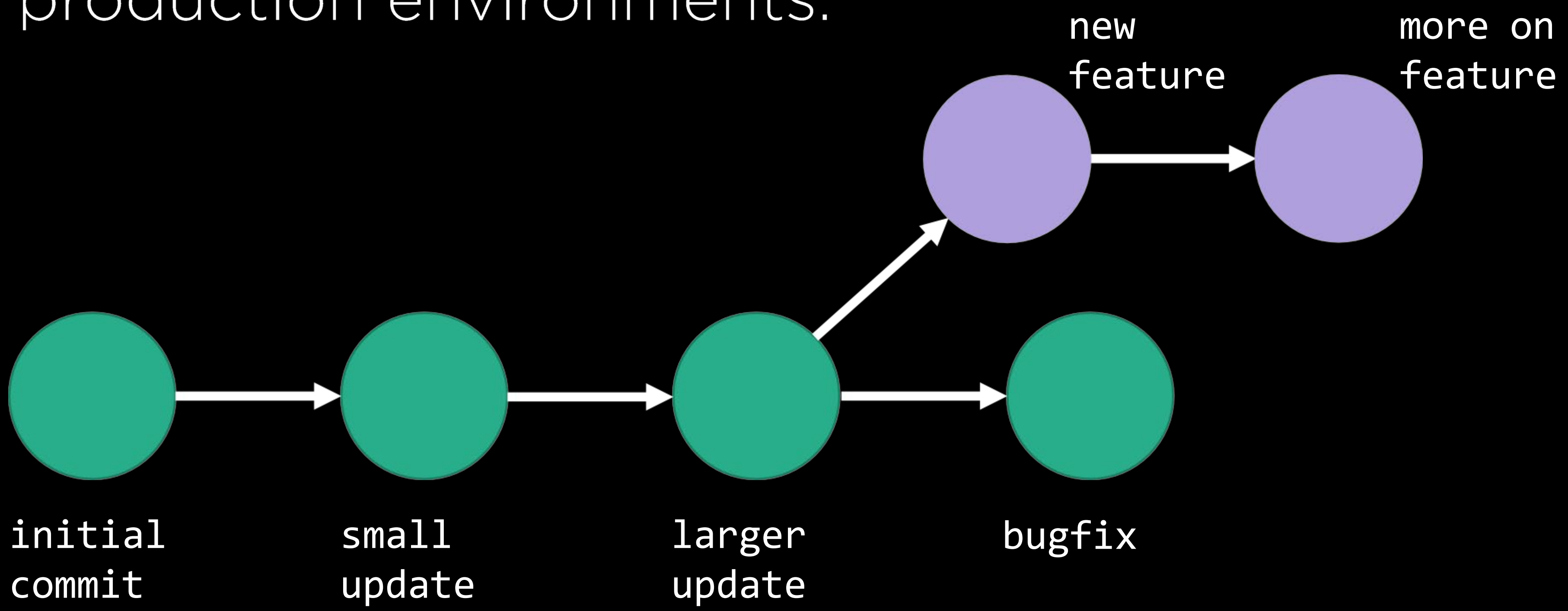
git branch

- Used to get an identical copy of your repository that can be edited and tested separately without impacting production environments.



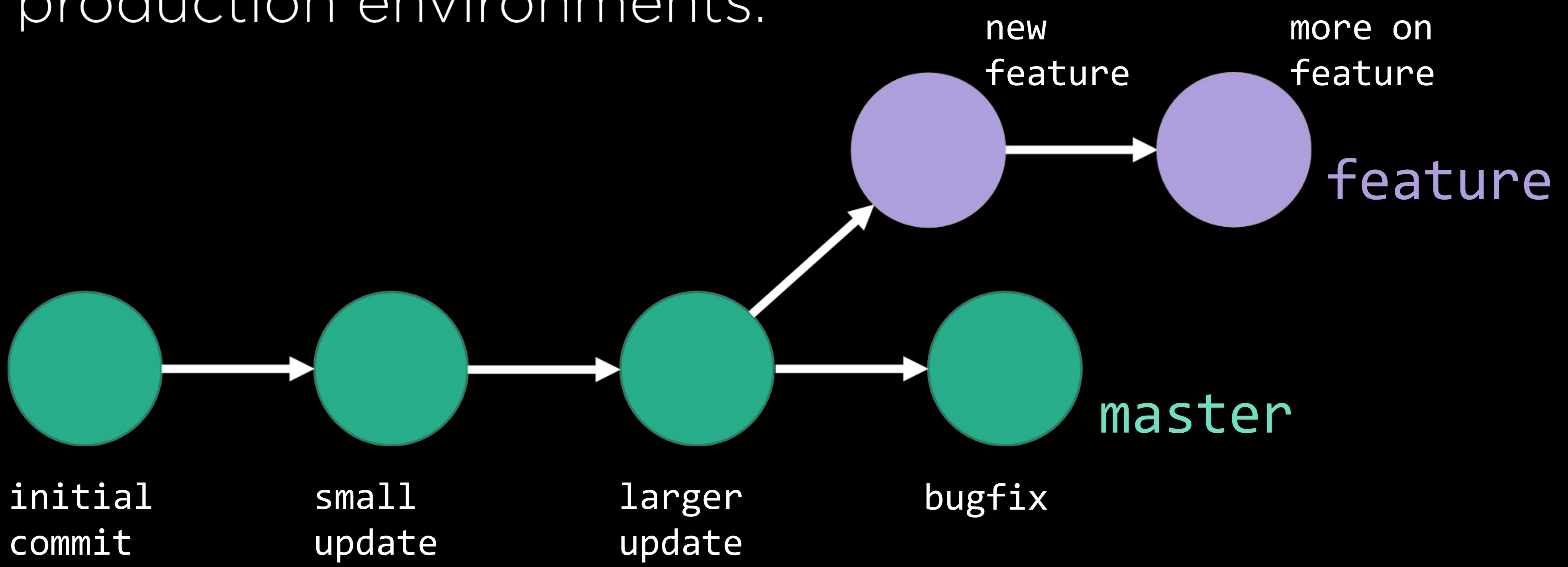
git branch

- Used to get an identical copy of your repository that can be edited and tested separately without impacting production environments.



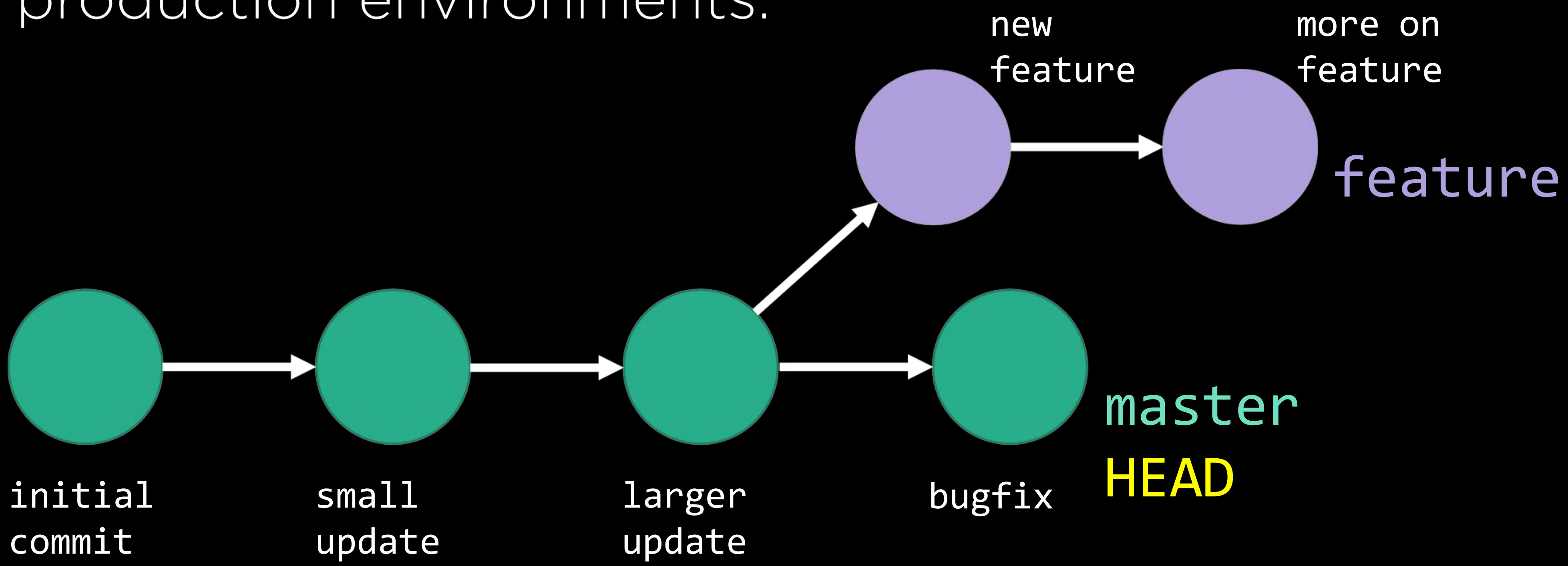
git branch

- Used to get an identical copy of your repository that can be edited and tested separately without impacting production environments.



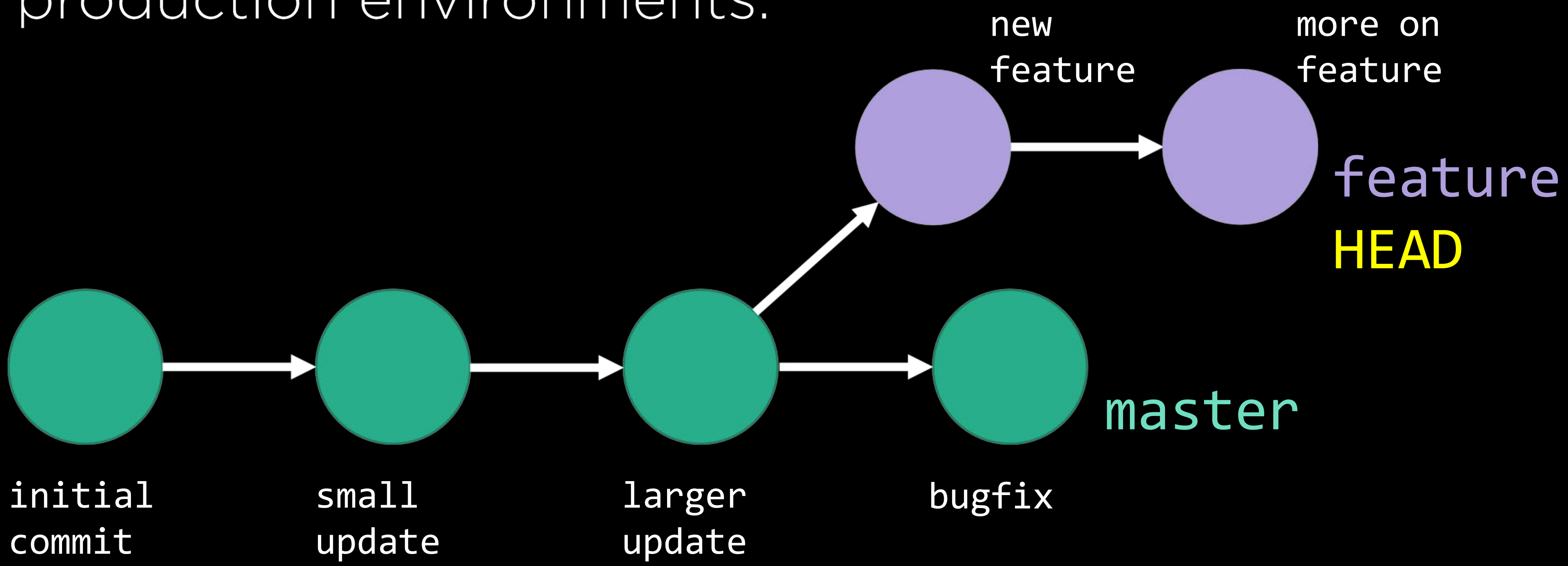
git branch

- Used to get an identical copy of your repository that can be edited and tested separately without impacting production environments.



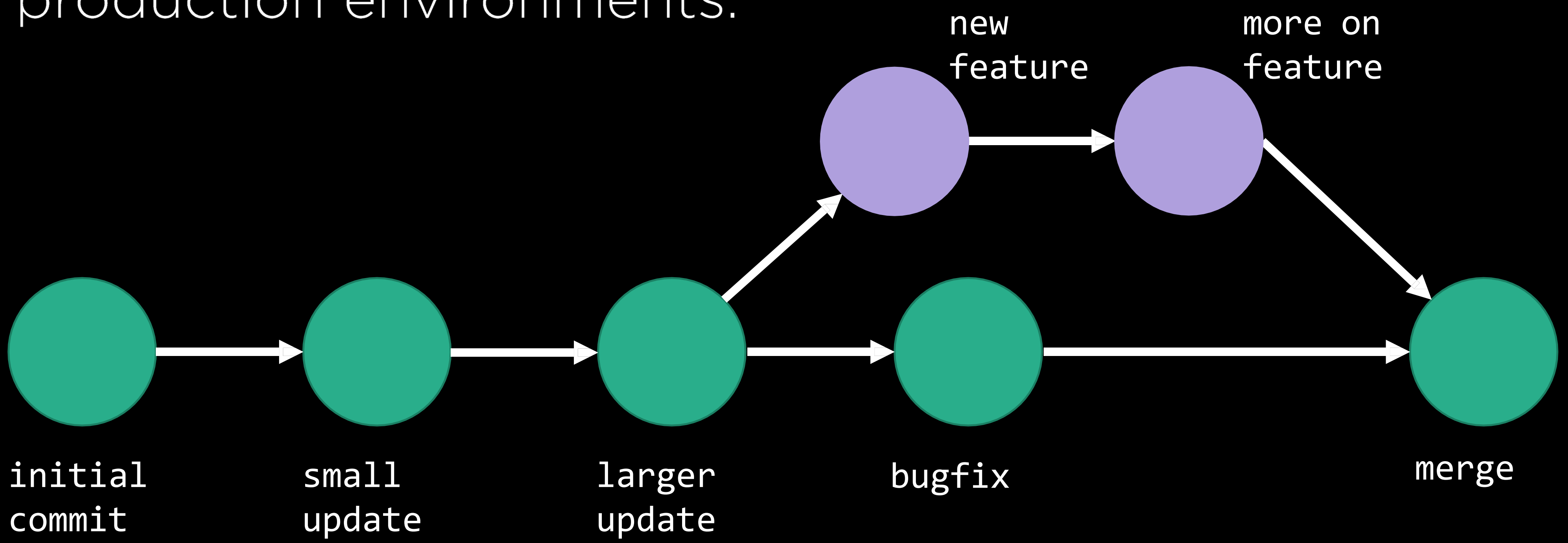
git branch

- Used to get an identical copy of your repository that can be edited and tested separately without impacting production environments.



git branch

- Used to get an identical copy of your repository that can be edited and tested separately without impacting production environments.



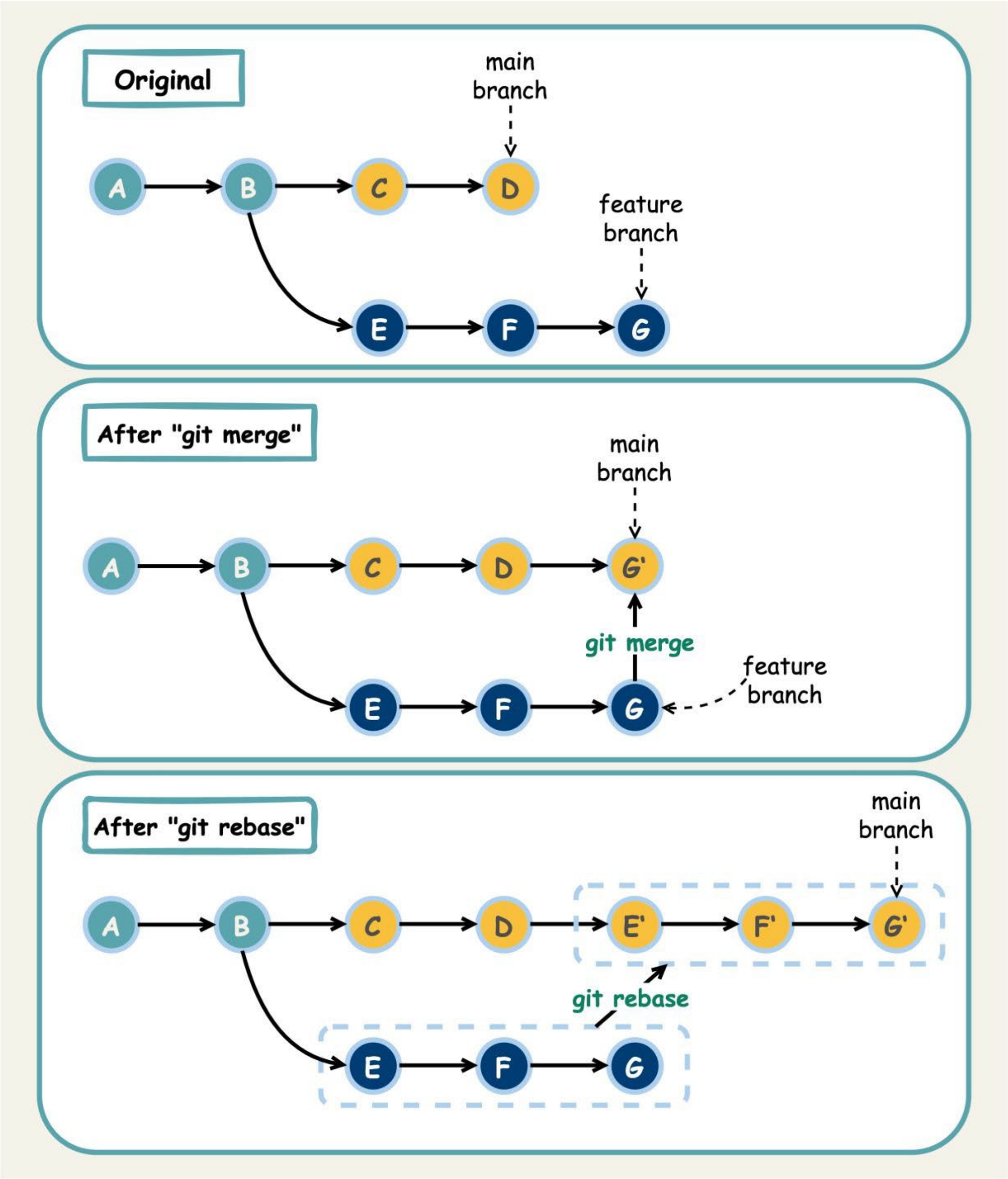
git branch

- Used to get an identical copy of your repository that can be edited and tested separately without impacting production environments.

```
git branch
git branch <name>
git checkout <name>
git merge <name>
```

Git Merge vs. Git Rebase

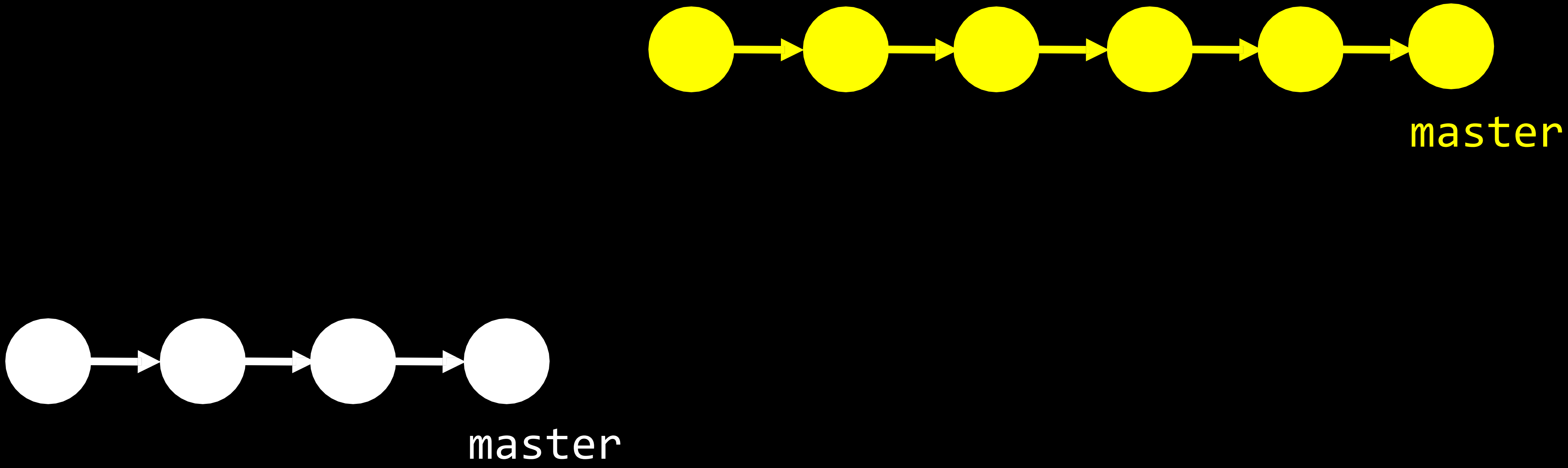
blog.bytebytego.com



Remotes

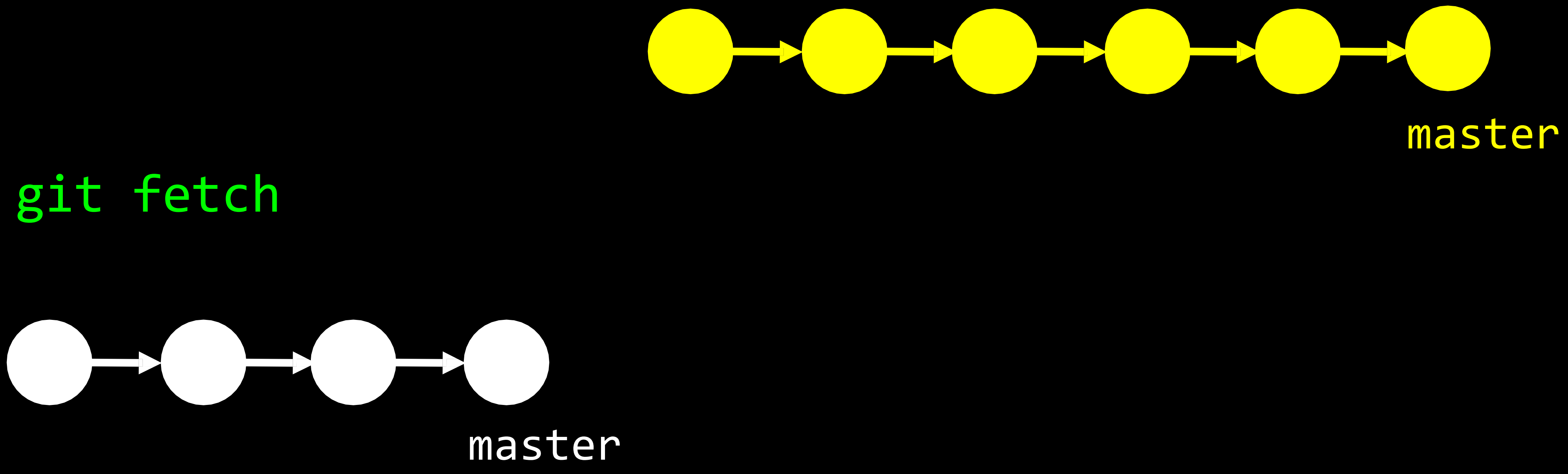
Remotes

- Recall that there's a distinction between code on your local device and code on the "remote", e.g. GitHub.



Remotes

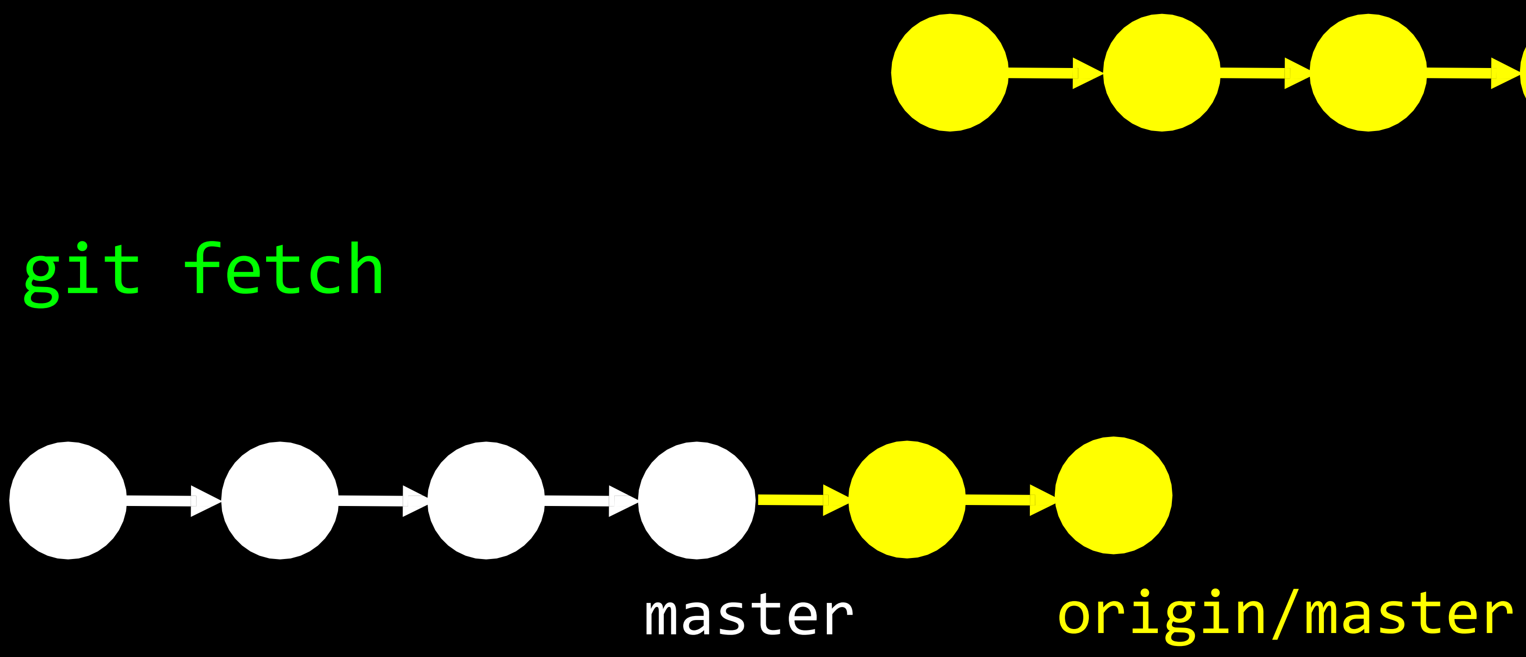
- Recall that there’s a distinction between code on your local device and code on the “remote”, e.g. GitHub.



```
git fetch
```

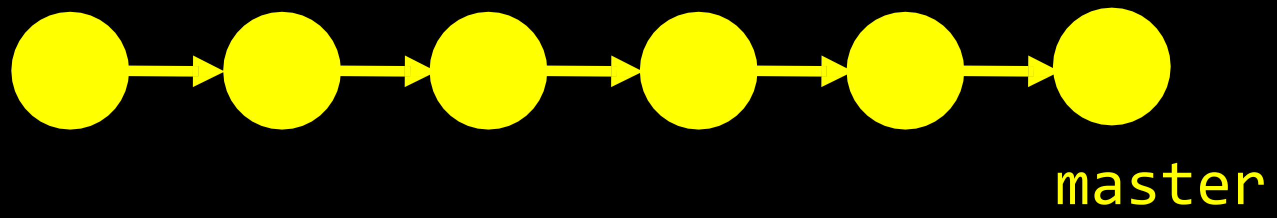
Remotes

- Recall that there’s a distinction between code on your local device and code on the “remote”, e.g. GitHub.

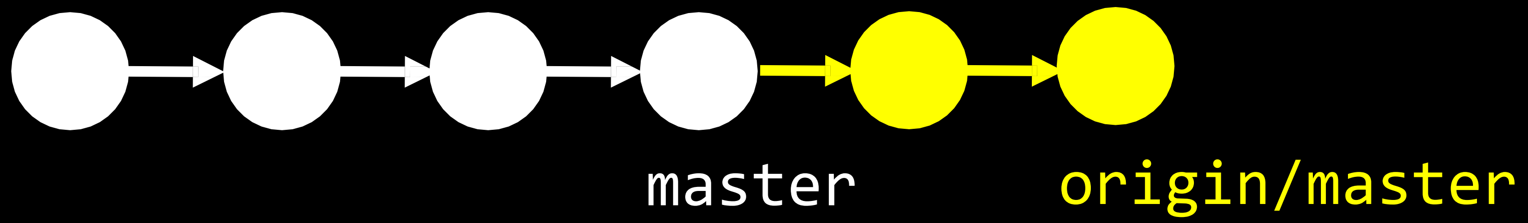


Remotes

- Recall that there's a distinction between code on your local device and code on the "remote", e.g. GitHub.

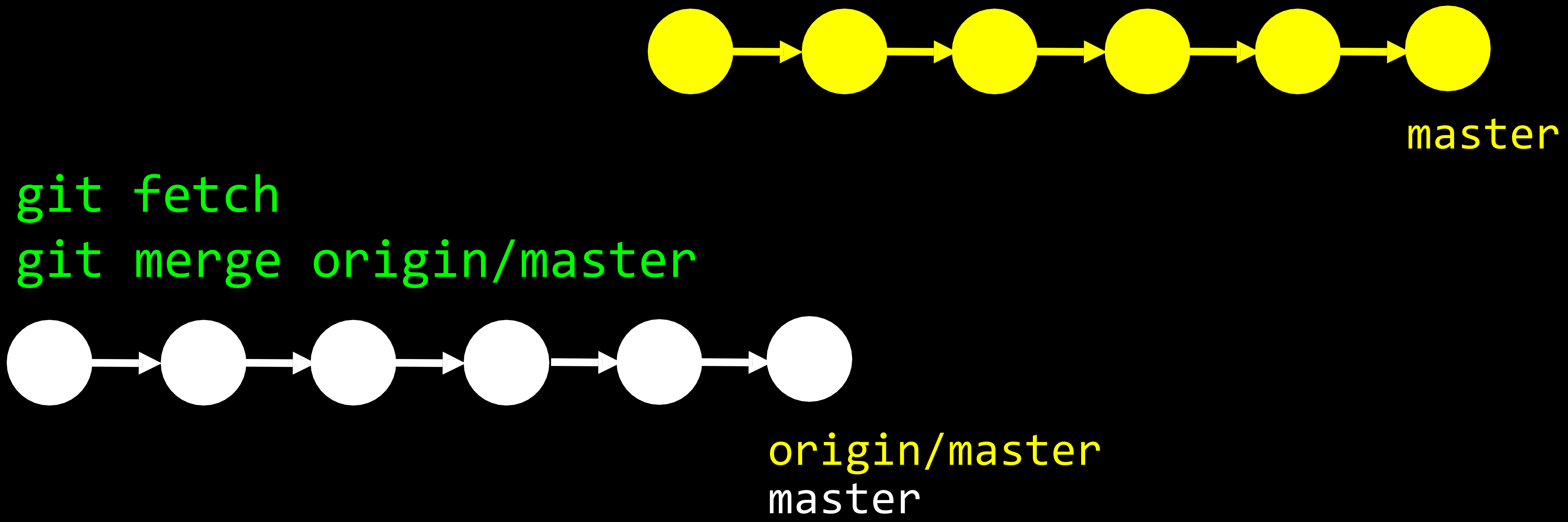


```
git fetch
git merge origin/master
```



Remotes

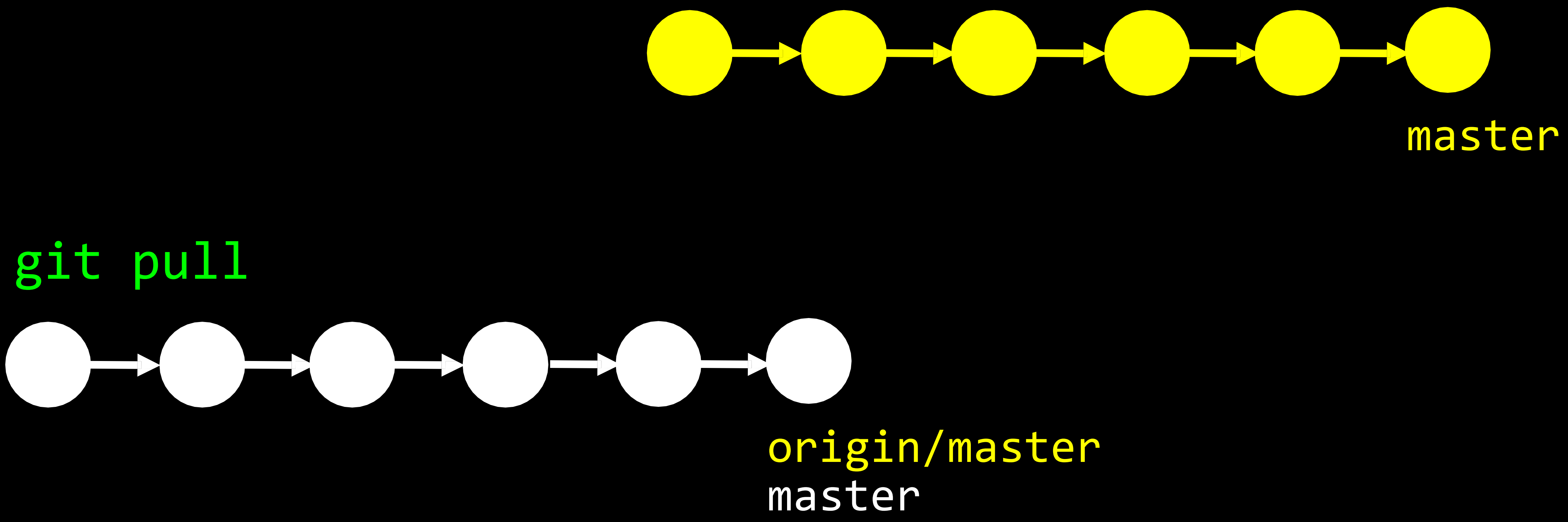
- Recall that there's a distinction between code on your local device and code on the "remote", e.g. GitHub.

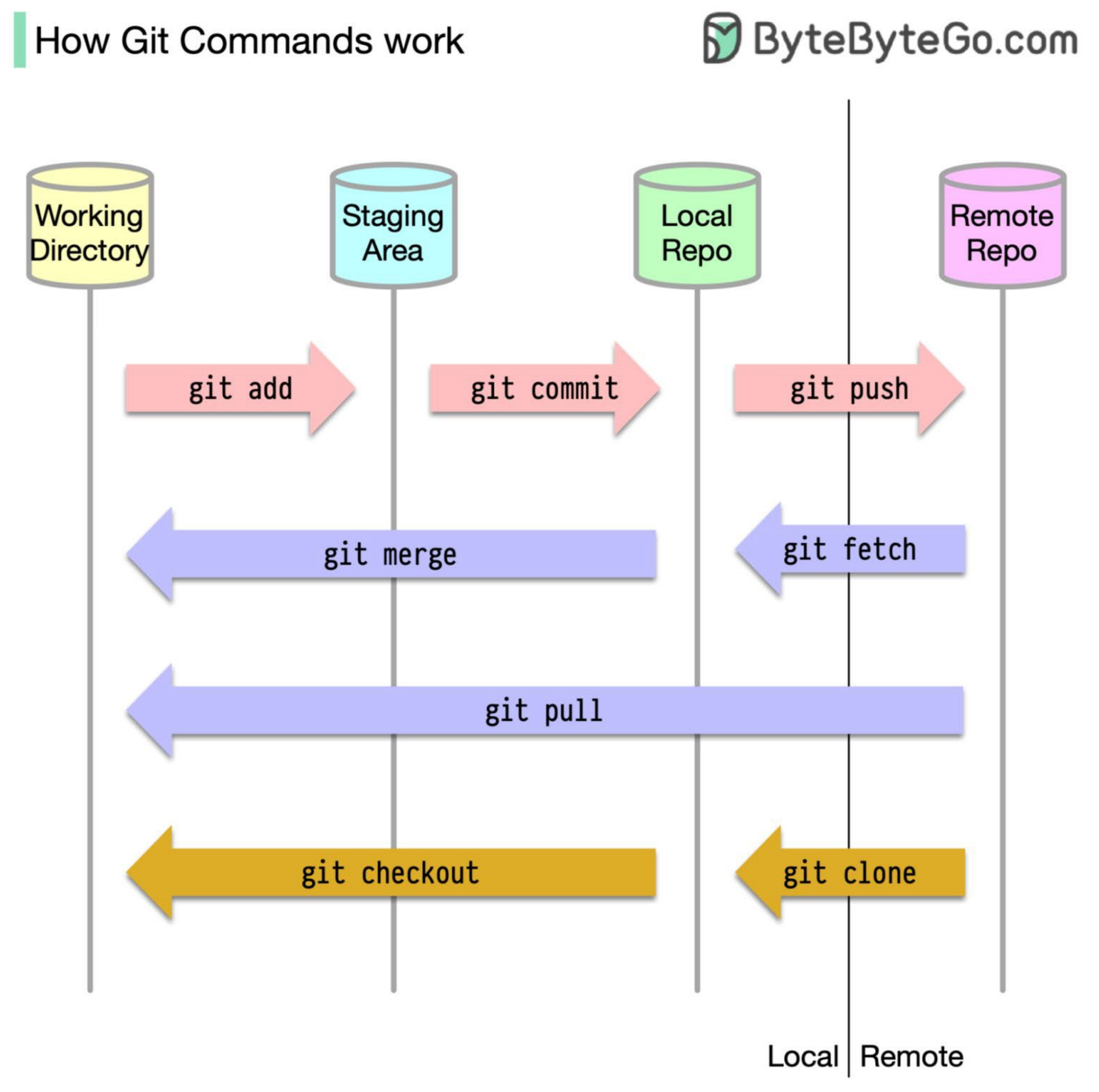


```
git fetch
git merge origin/master
```

Remotes

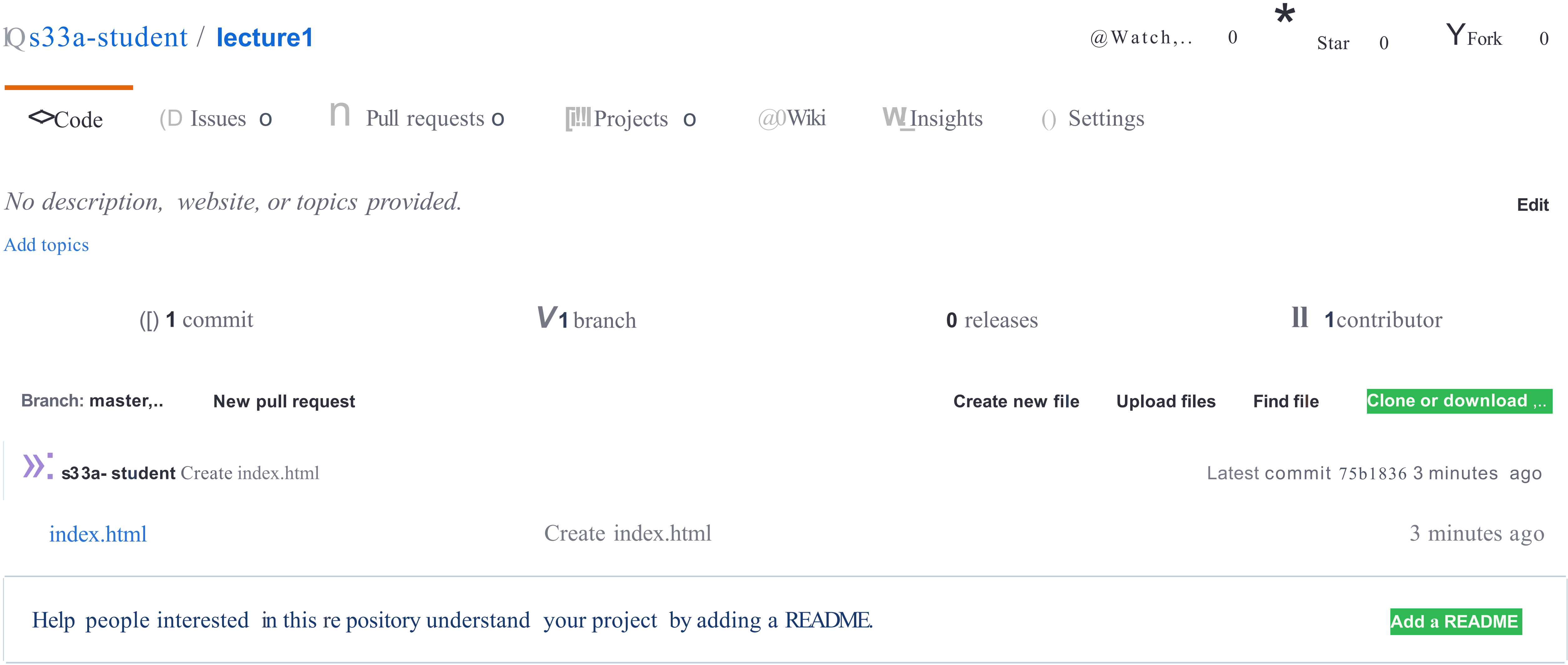
- Recall that there’s a distinction between code on your local device and code on the “remote”, e.g. GitHub.





How Git Commands work
ByteByteGo.com
Working Directory
Staging Area
Local Repo
Remote Repo
git add
git commit
git push
git merge
git fetch
git pull
git checkout
git clone
Local
Remote

Forks



IQ s33a-student / **lecture1**

@Watch,.. 0 * Star 0 Y Fork 0

<>Code (D Issues o n Pull requests o [!!! Projects o @0Wiki W Insights () Settings

No description, website, or topics provided. **Edit**

Add topics

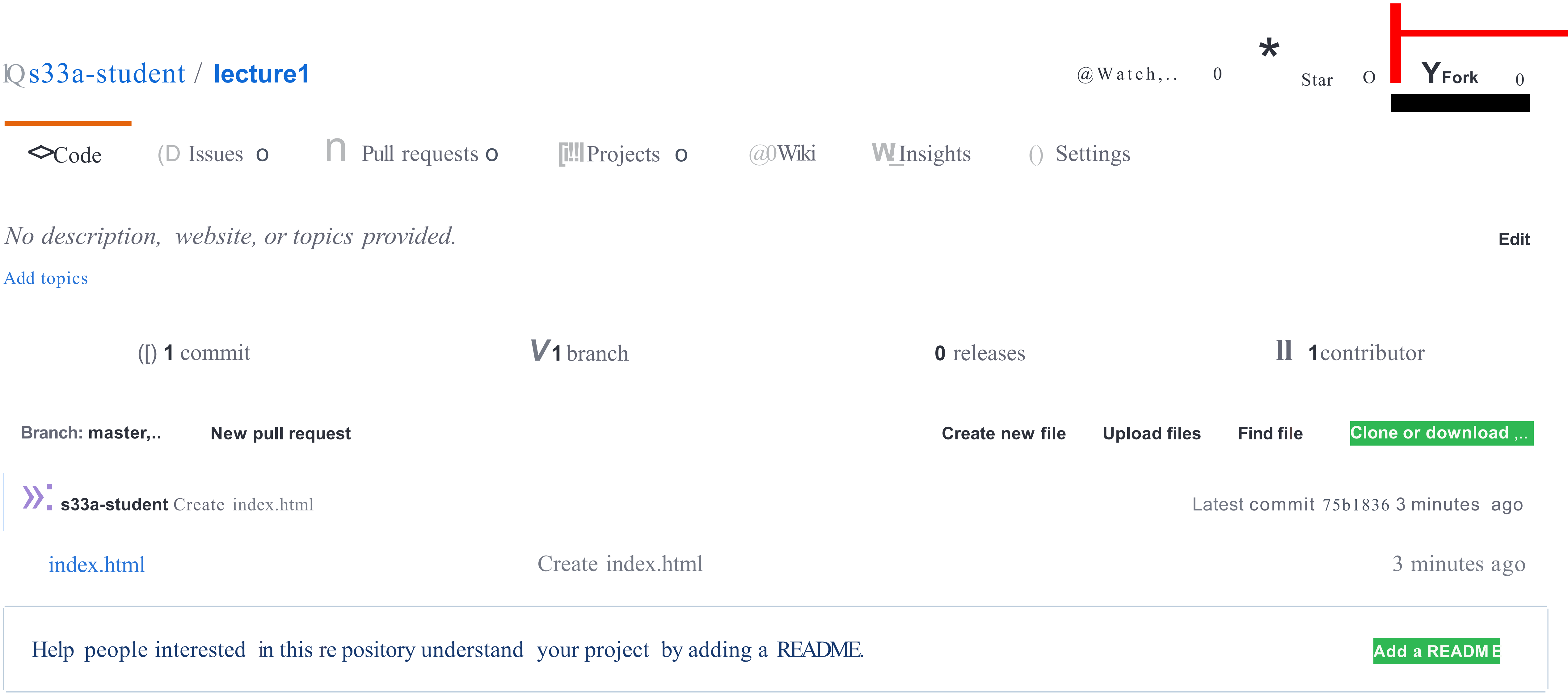
([) **1** commit V **1** branch **0** releases ll **1** contributor

Branch: master,.. **New pull request** **Create new file** **Upload files** **Find file** **Clone or download ,..**

s33a- student Create index.html Latest commit 75b1836 3 minutes ago

index.html Create index.html 3 minutes ago

Help people interested in this re pository understand your project by adding a README. **Add a README**



IQ s33a-student / **lecture1**

@Watch,.. 0 * Star O YFork 0

<>Code (D Issues o n Pull requests o [!!!Projects o @0Wiki W Insights () Settings

No description, website, or topics provided.

Edit

Add topics

([) **1** commit V **1** branch **0** releases ll **1** contributor

Branch: **master**,.. **New pull request** **Create new file** **Upload files** **Find file** **Clone or download ,..**

s33a-student Create index.html Latest commit 75b1836 3 minutes ago

index.html	Create index.html	3 minutes ago

Help people interested in this re pository understand your project by adding a README.

Add a READM E

Pull Requests

Pull Requests

- A ***pull request*** is GitHub-specific terminology that arises when someone other than the original contributor wishes to propose modifications to someone else's repository.

Pull Requests

- A ***pull request*** is GitHub-specific terminology that arises when someone other than the original contributor wishes to propose modifications to someone else's repository.

- They typically arise after someone has forked their own copy of a repository, tested out a change, and wants to incorporate it back in to the original codebase.

Pull Requests

- A ***pull request*** is GitHub-specific terminology that arises when someone other than the original contributor wishes to propose modifications to someone else's repository.

- They typically arise after someone has forked their own copy of a repository, tested out a change, and wants to incorporate it back in to the original codebase.

- It's also possible to use PRs to merge your own branches.

GitHub Pages

GitHub Pages

- So far, all of the code we've written has only been "run" locally, on my own machine or IDE.

- GitHub Pages is a feature of GitHub that effectively allows you to deploy the contents of a repository to live on the internet.