Software Engineering in the Arts and Humanities

Introduction & Git

September 4, 2019

cs.harvard.edu/100

(This course does not use Canvas!)

Staff

Staff

- Instructors
  - At Harvard: Doug Lloyd (lloyd@cs50.harvard.edu)
  - At Yale: Benedict Brown (bjbrown@cs50.yale.edu)

- Teaching Fellows
  - Robert Roessler (robertroessler@g.harvard.edu)
  - Kareem Zidane (kzidane@cs50.harvard.edu)

- 100@cs.harvard.edu

Overview

Overview

- Who is this class for?
- How quickly will it move?
- What will we learn?

Overview

- This week and next:
    - Lectures
- Starting in two weeks:
    - Mostly lectures on Mondays, hands-on lab work and demos on Wednesdays
- Starting in mid-October:
    - No lectures or labs, weekly meetings with your project adviser

Overview

- Git
- HTML and CSS
- JavaScript
- SQL
- Node.js
- React Native
- Project Development
- Final Project Fair

Projects

Projects

- 6-7 weekly projects
  - Topics including:
    - Art
    - Literature
    - Text analysis
    - Music
  - Through mid-October
- Final Project
  - Consult with humanities faculty or design your own
  - Four milestones
  - Mid-October to early December

Office Hours

Tuesday evenings

Lottery

Enrollment limited; apply at cs.harvard.edu/100 by Saturday

Piazza

Questions?

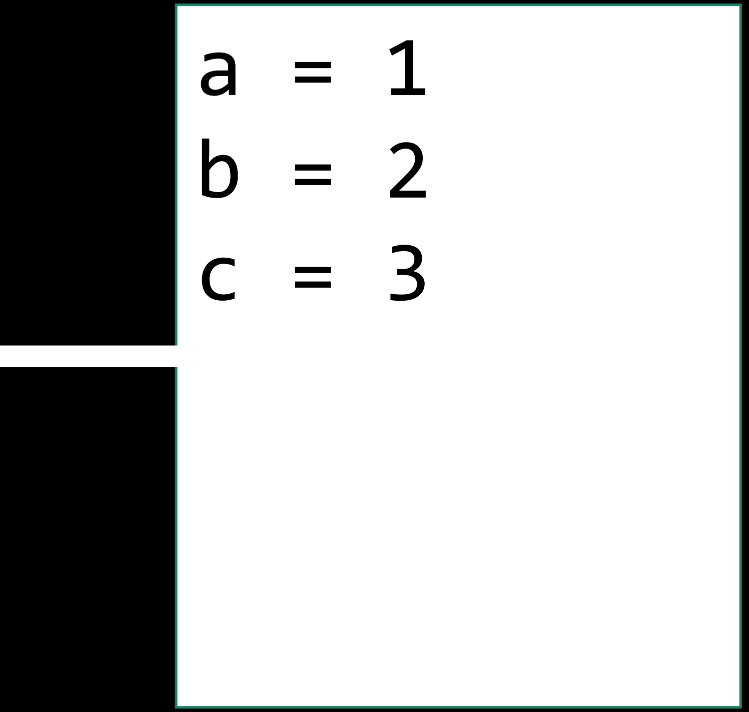
Git

Git

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

Git

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

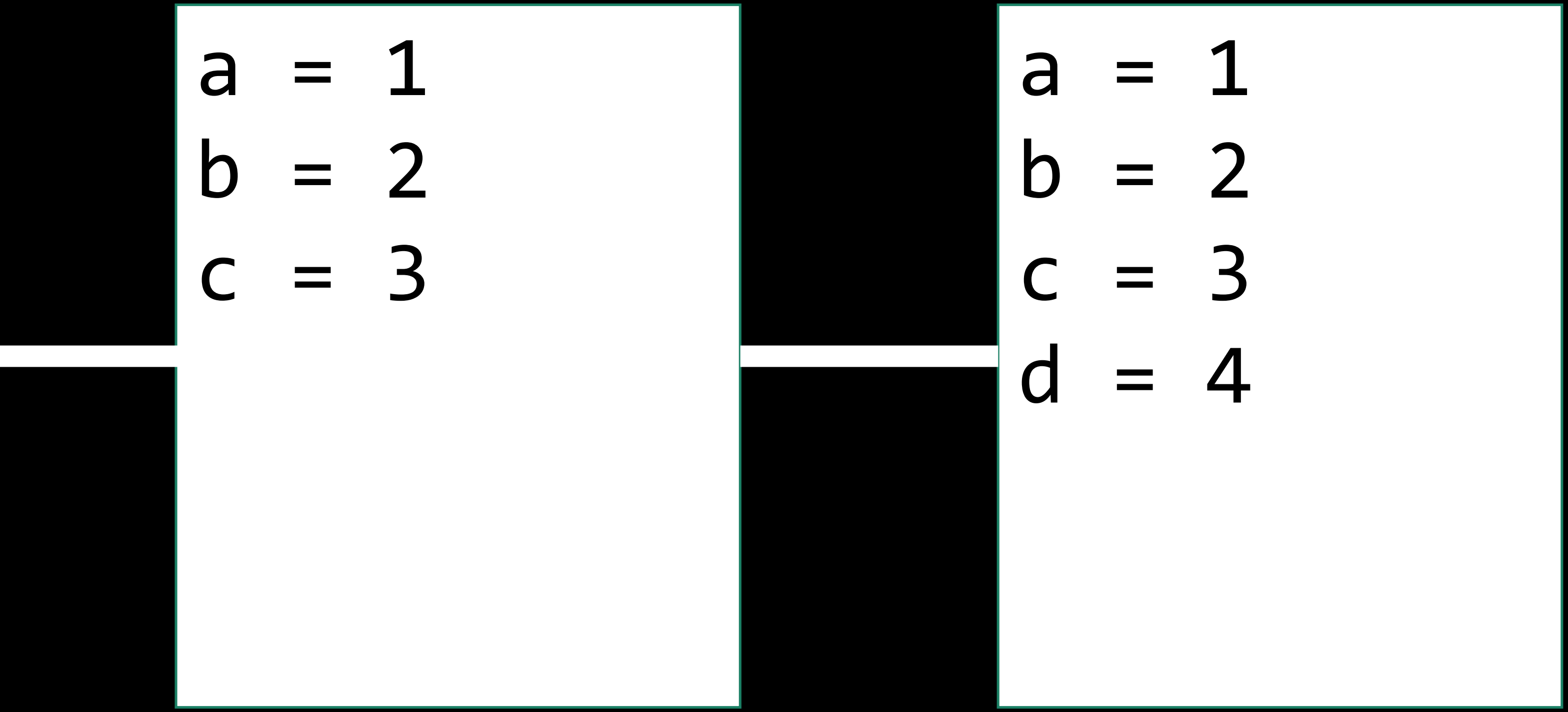


```
a = 1
b = 2
c = 3
```

`Create file`

Git

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



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

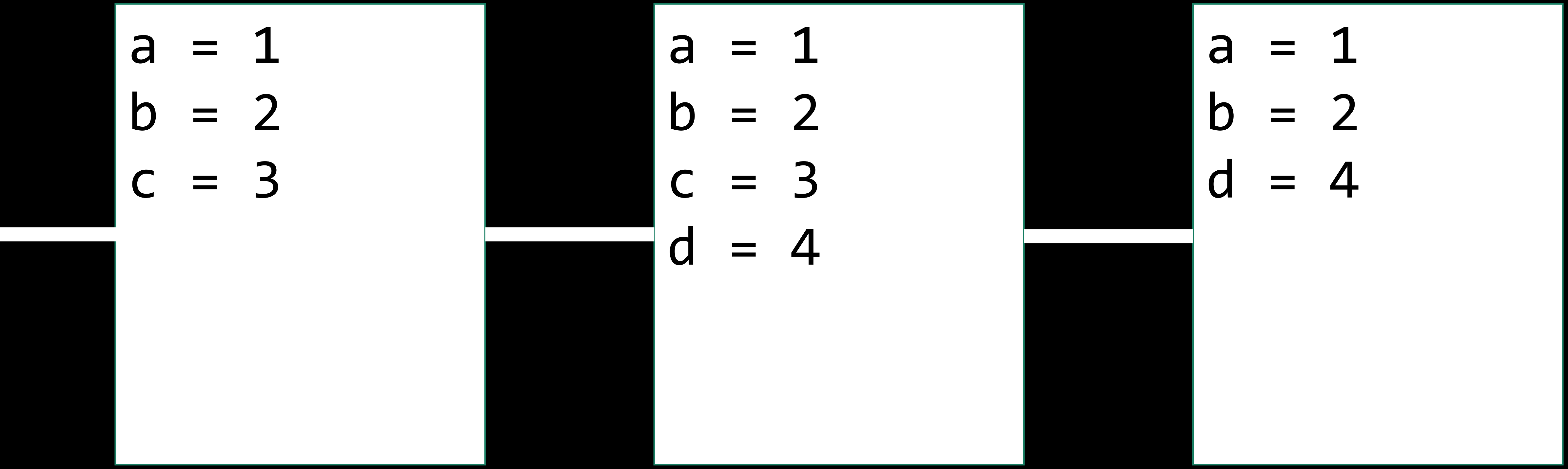
Create file

```
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
```

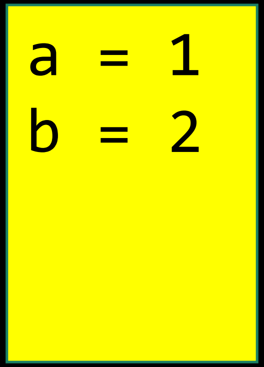
Delete a line

Git

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

Git

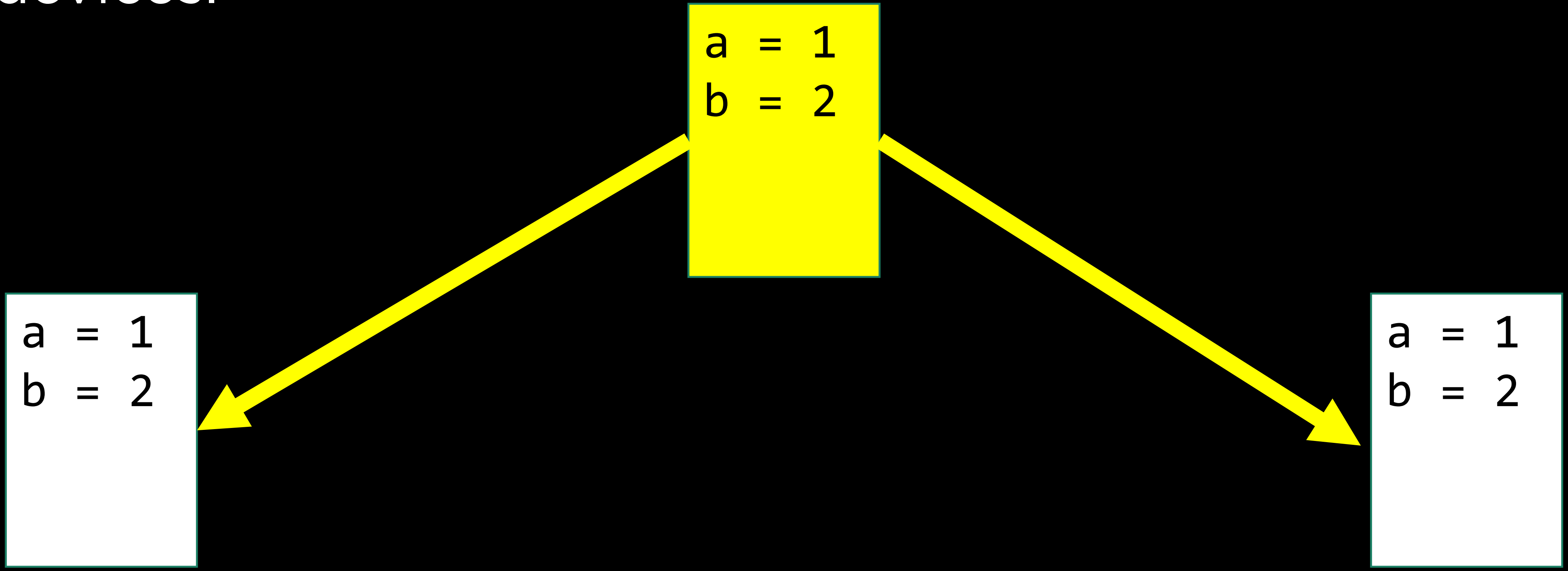
- We also use Git to synchronize code across multiple devices.



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

Git

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



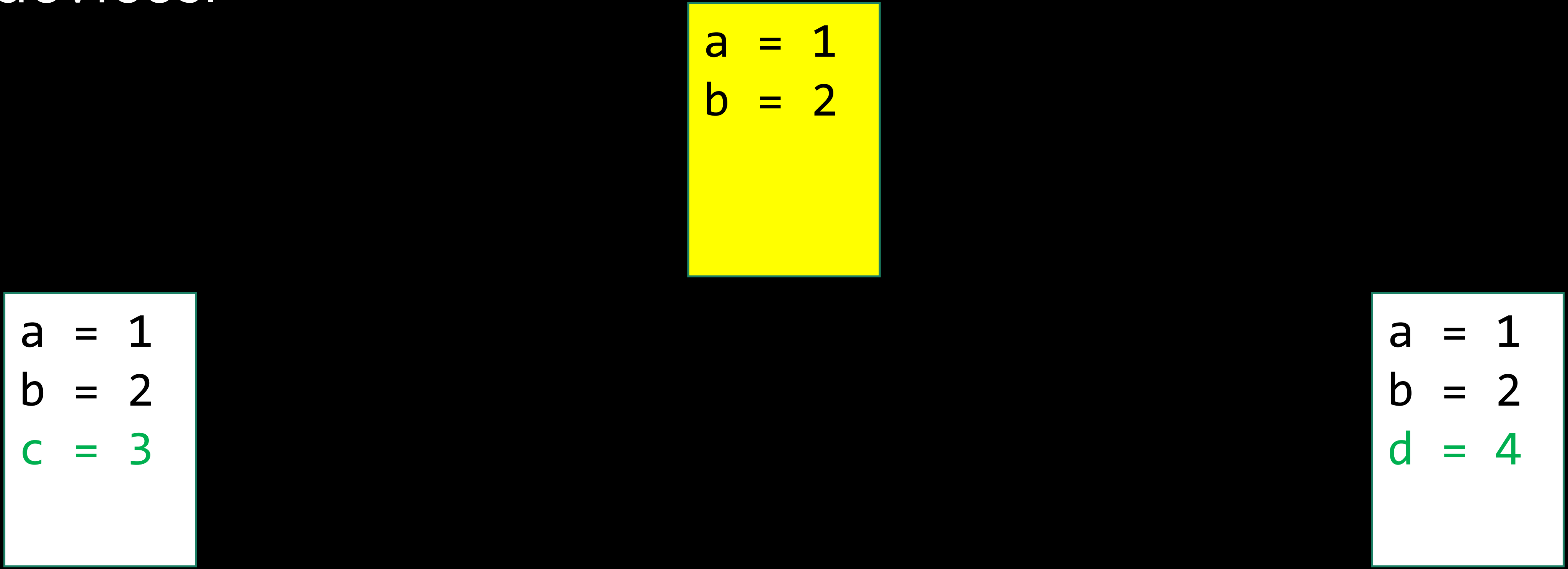
```
a = 1
b = 2
```

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

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

Git

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



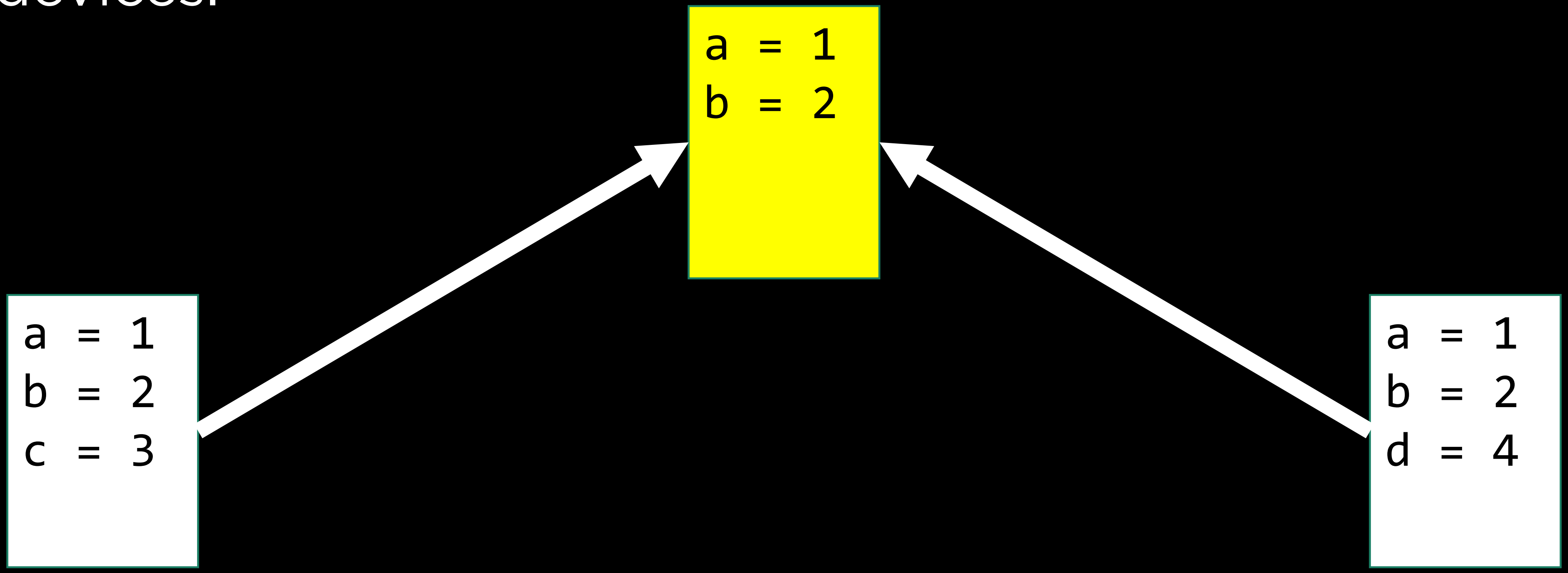
```
a = 1
b = 2
```

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

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

Git

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



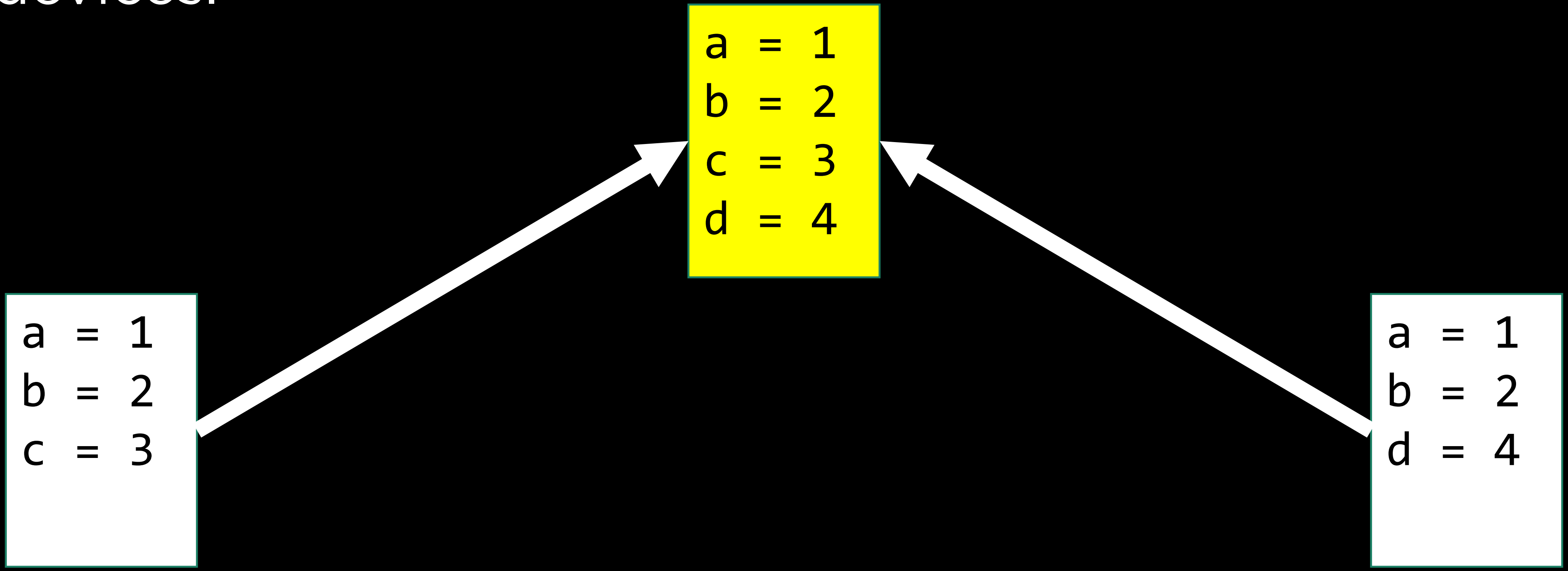
```
a = 1
b = 2
```

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

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

Git

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



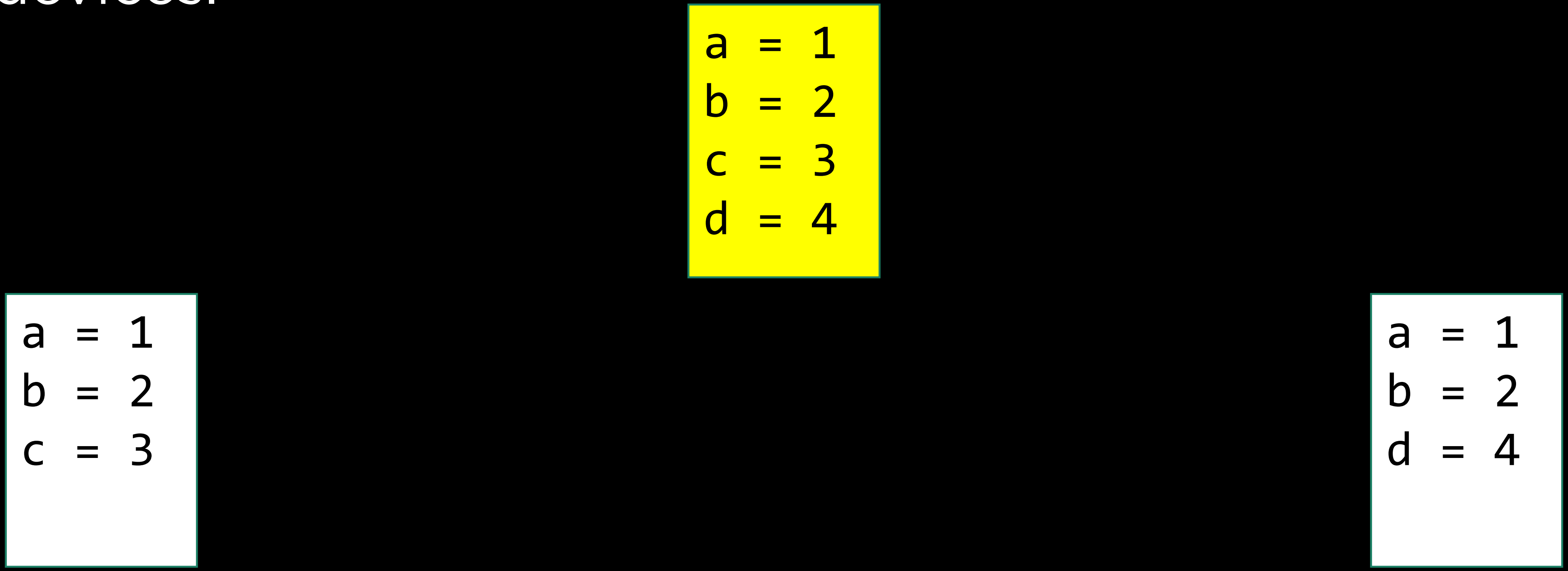
```
a = 1
b = 2
c = 3
d = 4
```

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

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

Git

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



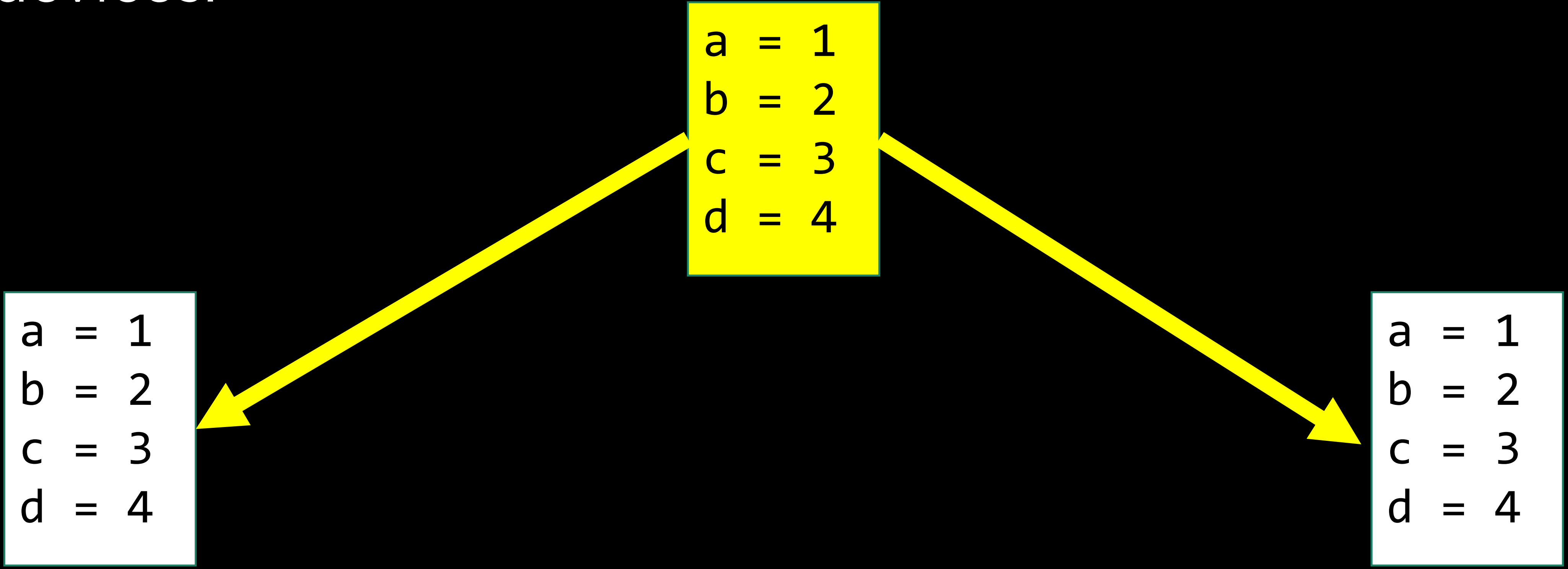
```
a = 1
b = 2
c = 3
d = 4
```

```
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
```

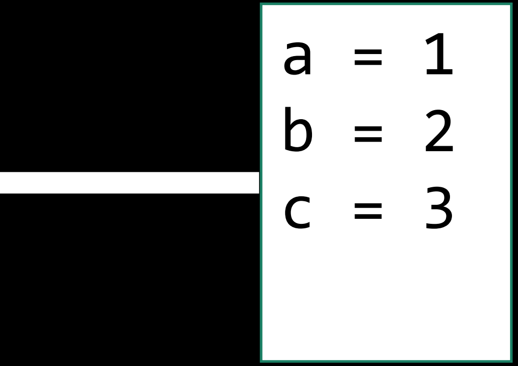
```
a = 1
b = 2
c = 3
d = 4
```

Git

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

Git

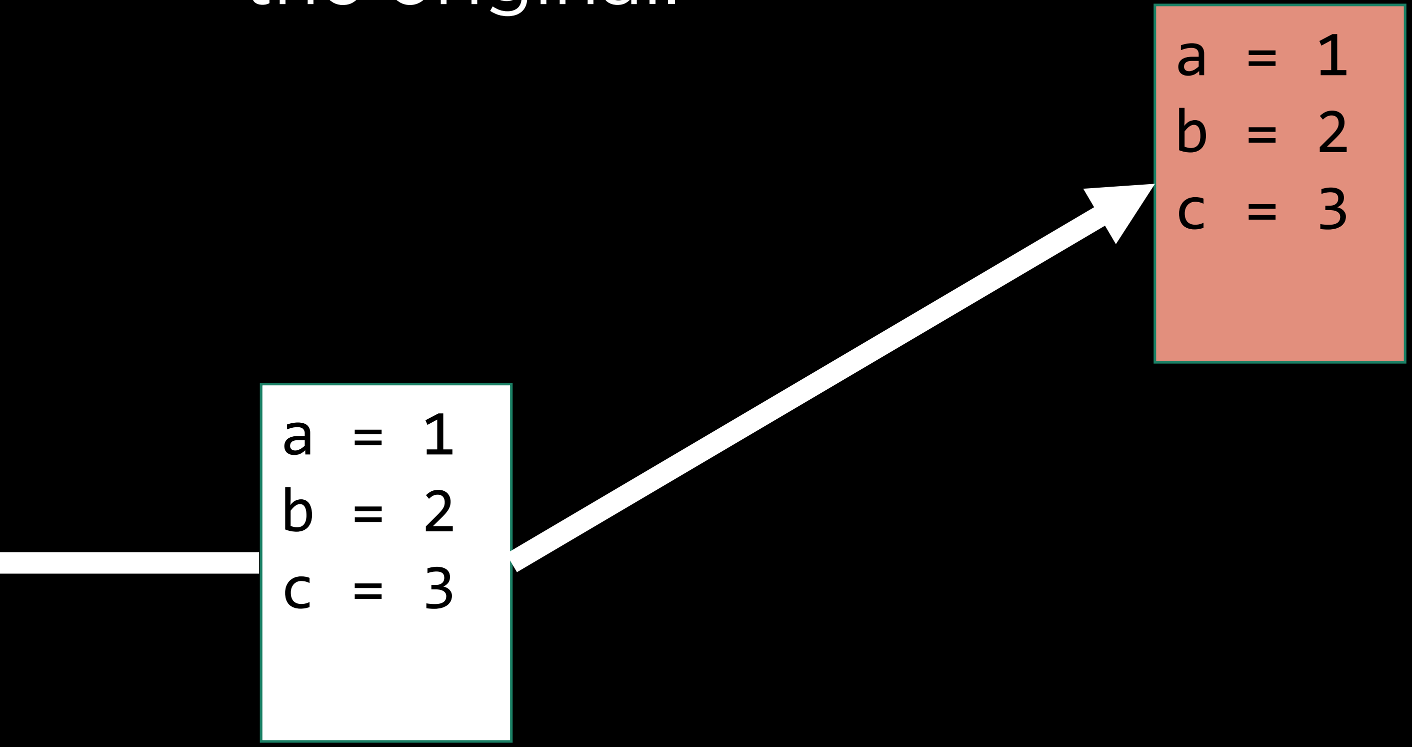
- Git allows us to test changes to our code without losing the original.



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

Git

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

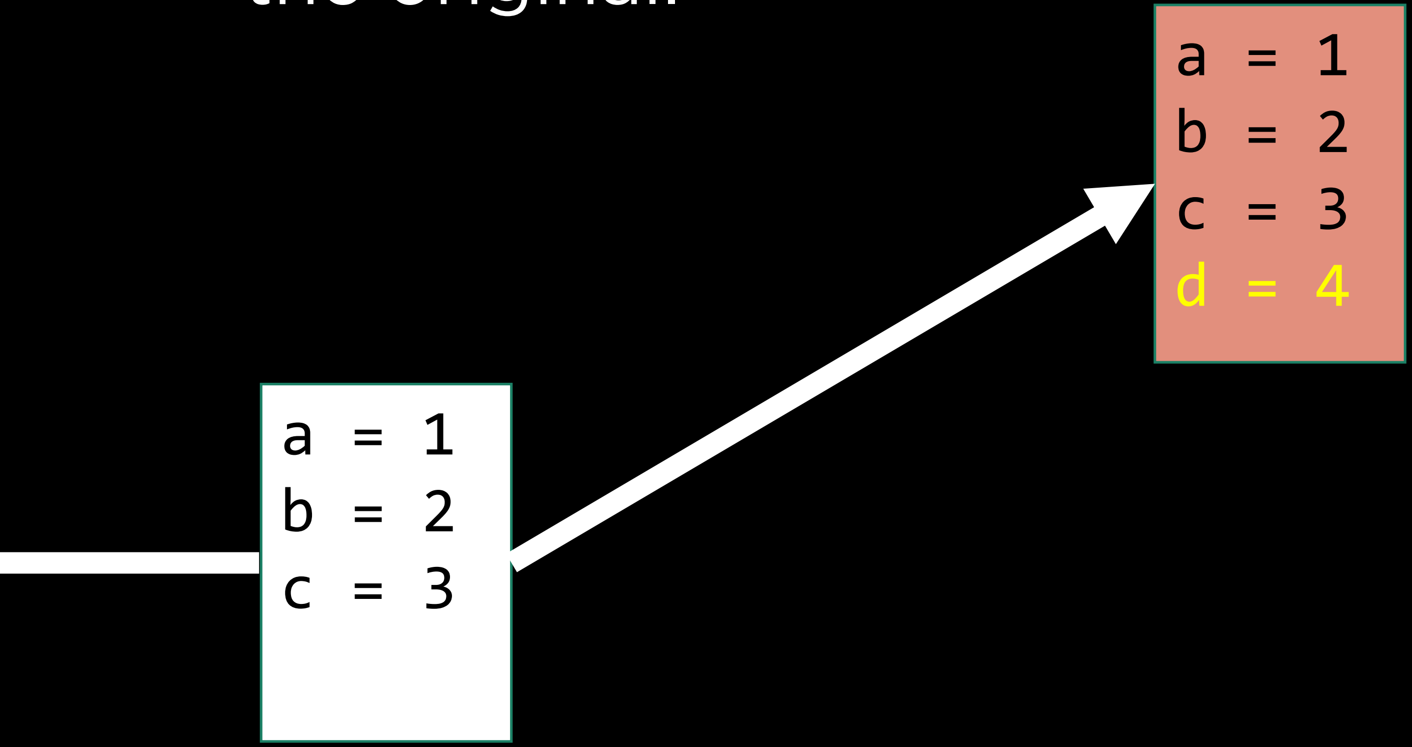


```
a = 1
b = 2
c = 3
```

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

Git

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

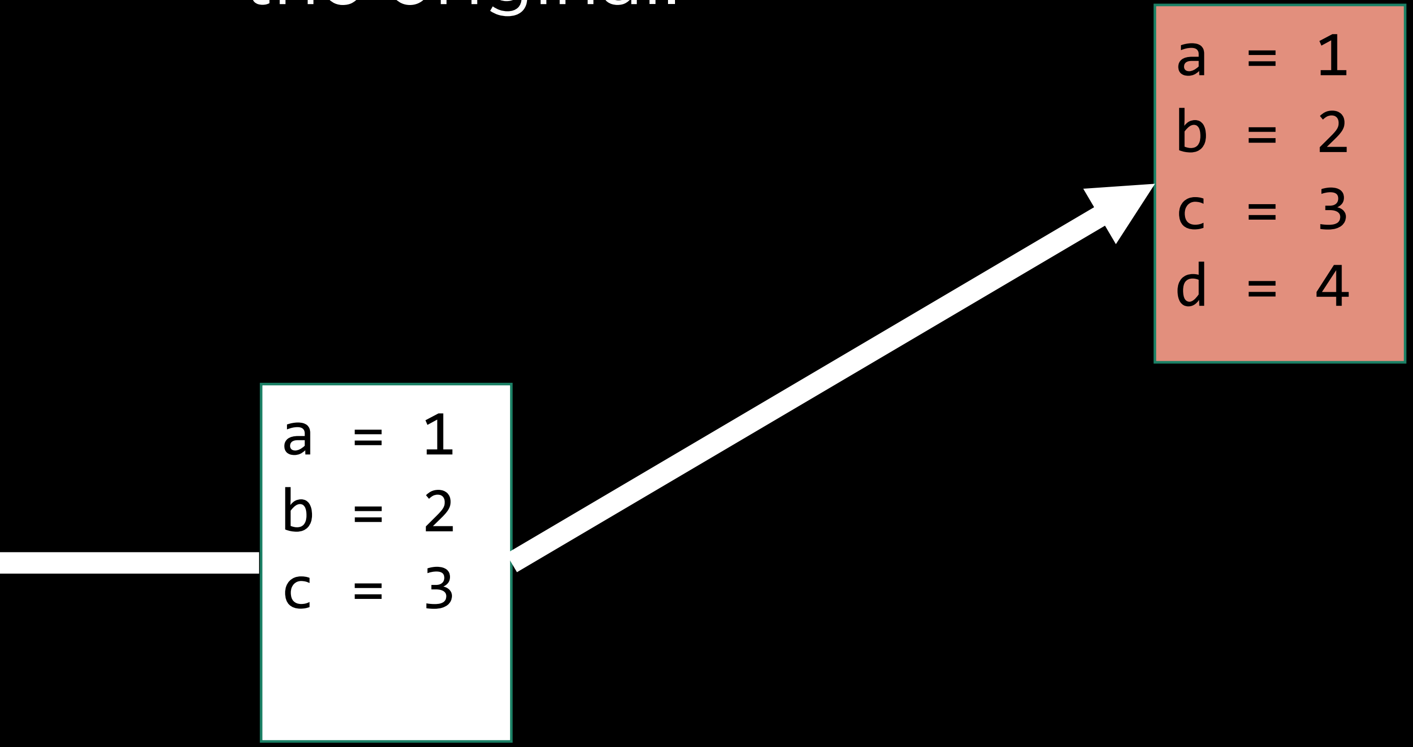


```
a = 1
b = 2
c = 3
```

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

Git

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

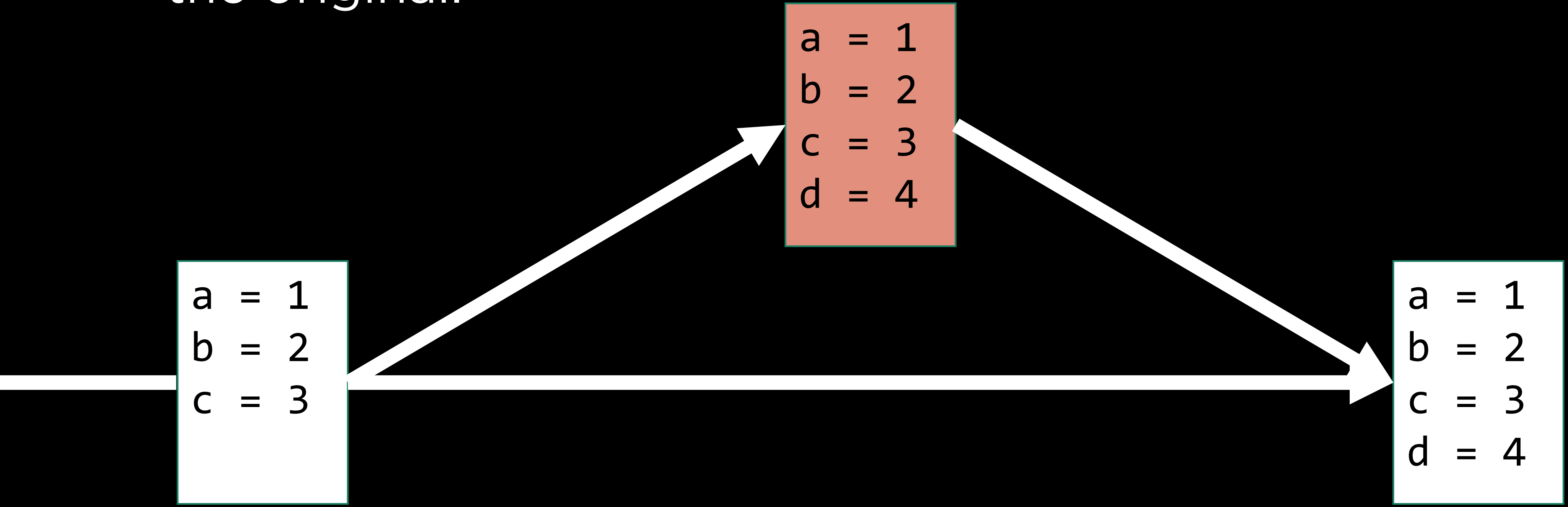


```
a = 1
b = 2
c = 3
```

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

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
```

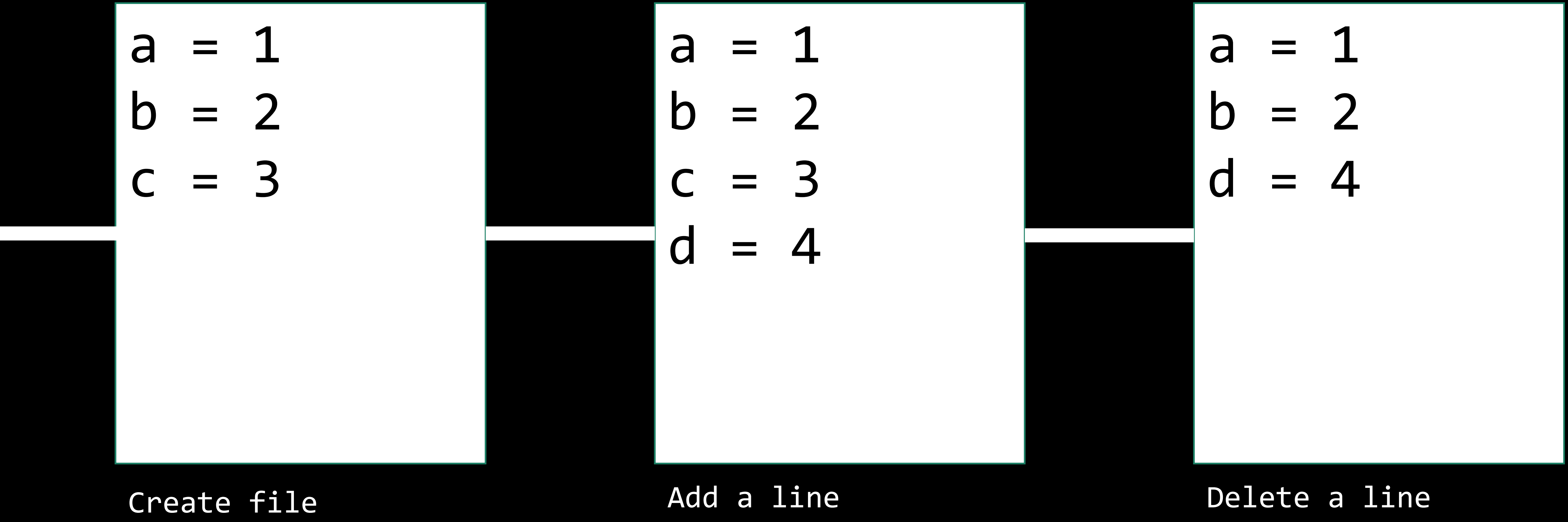
```
a = 1
b = 2
c = 3
d = 4
```

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
```

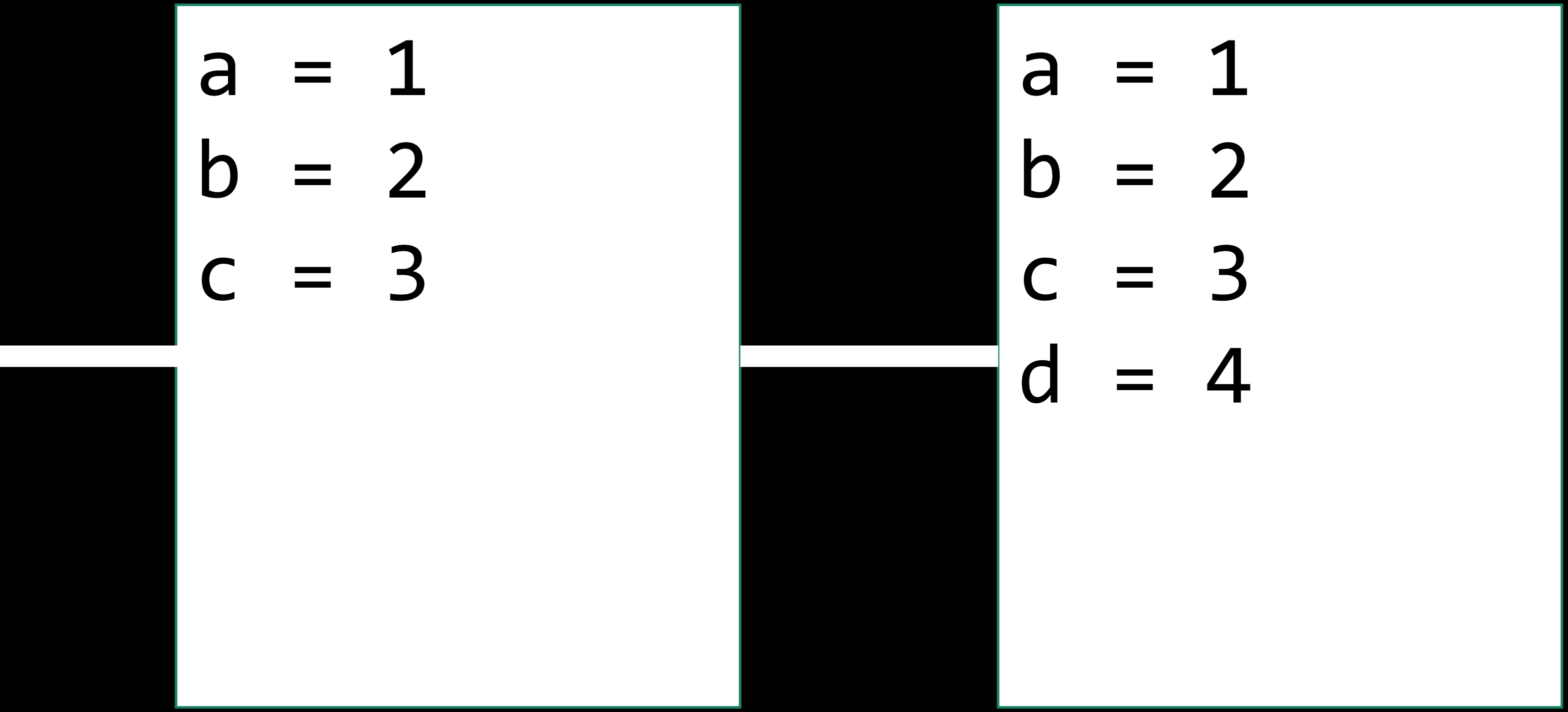
Add a line

```
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

GitHub

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

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.

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.

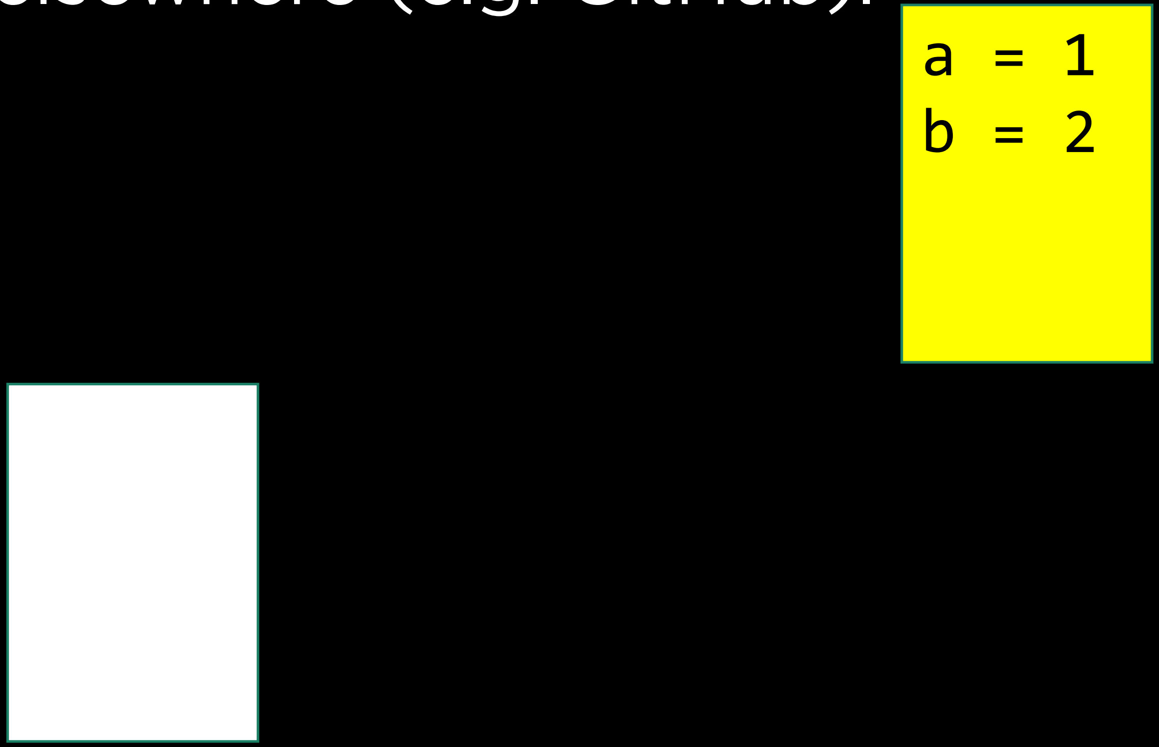
git clone

git clone

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

git clone

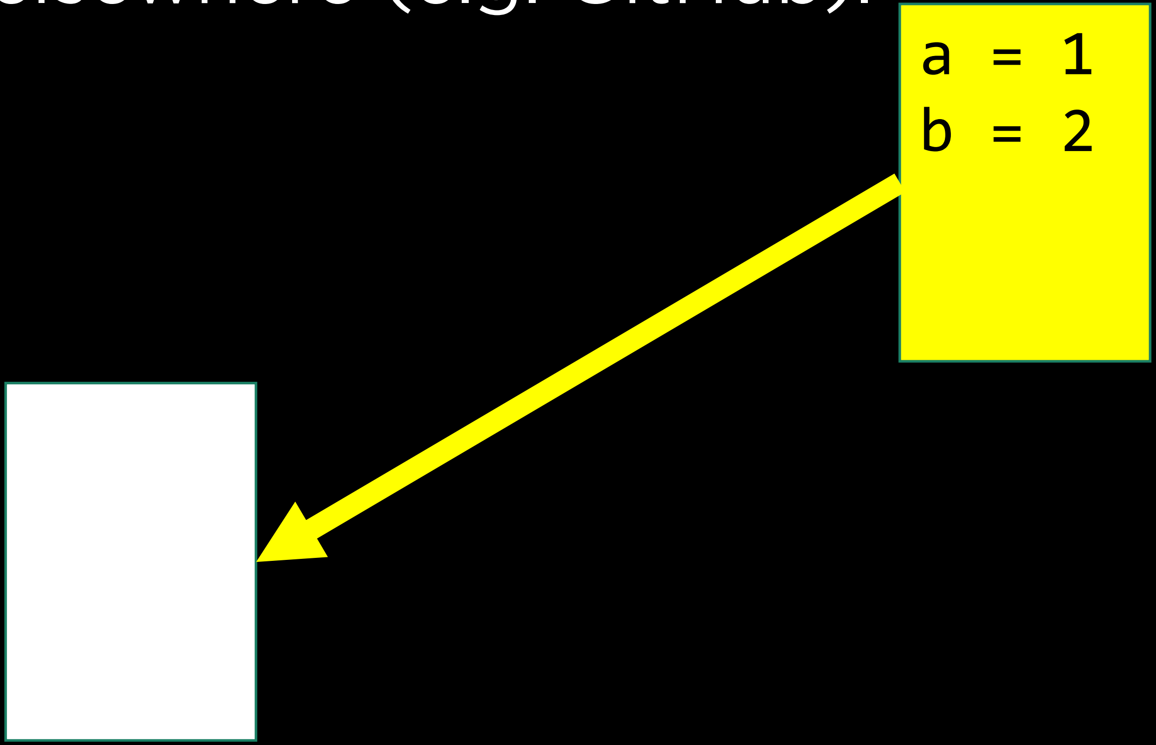
- Used to initially get a local copy of a repository stored elsewhere (e.g. GitHub).



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

git clone

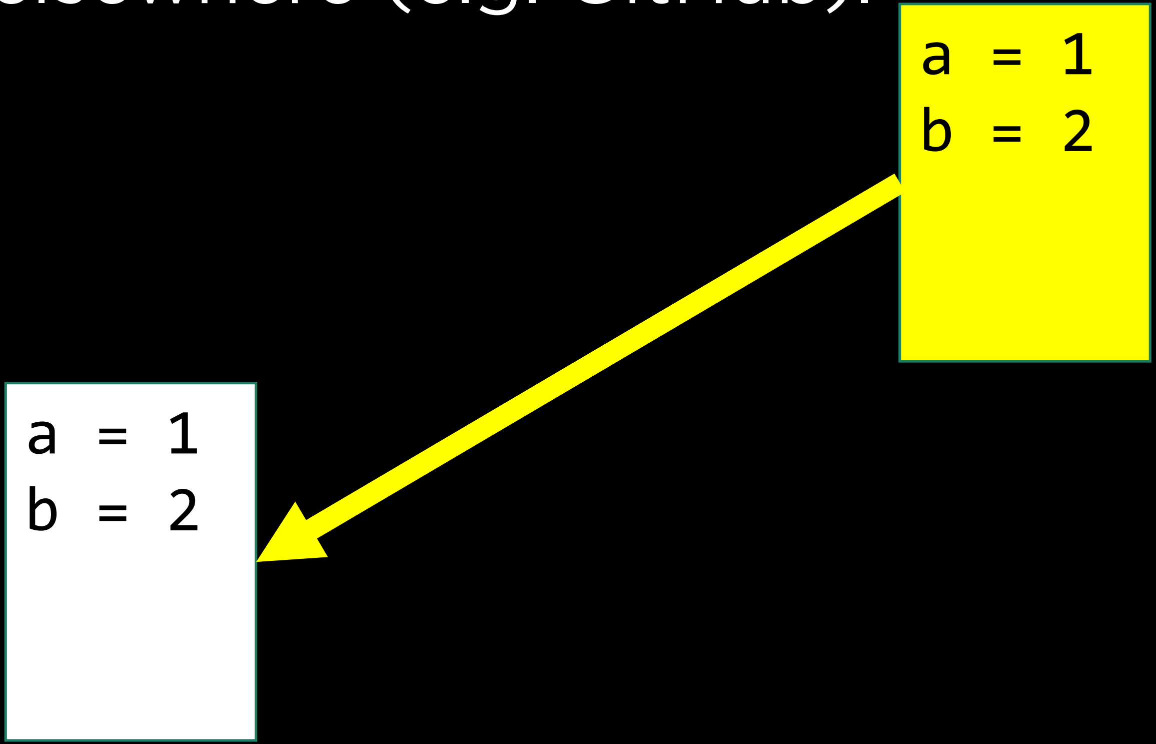
- Used to initially get a local copy of a repository stored elsewhere (e.g. GitHub).



```
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>
```

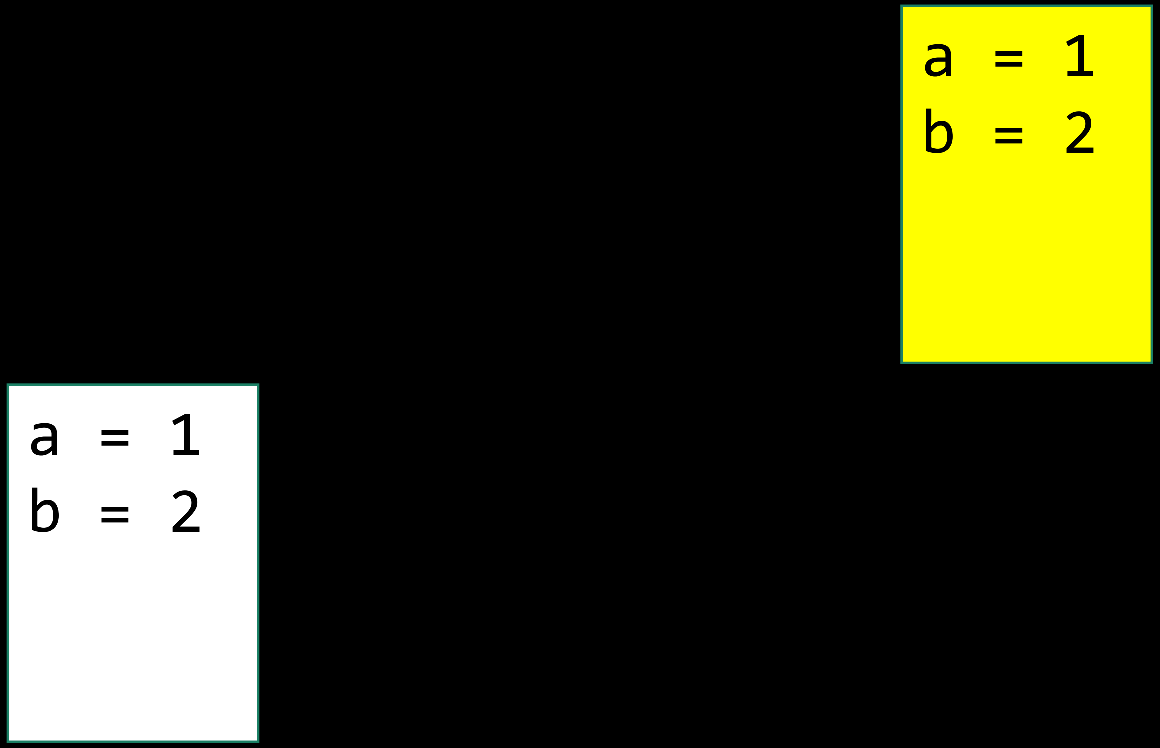
git add

`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.

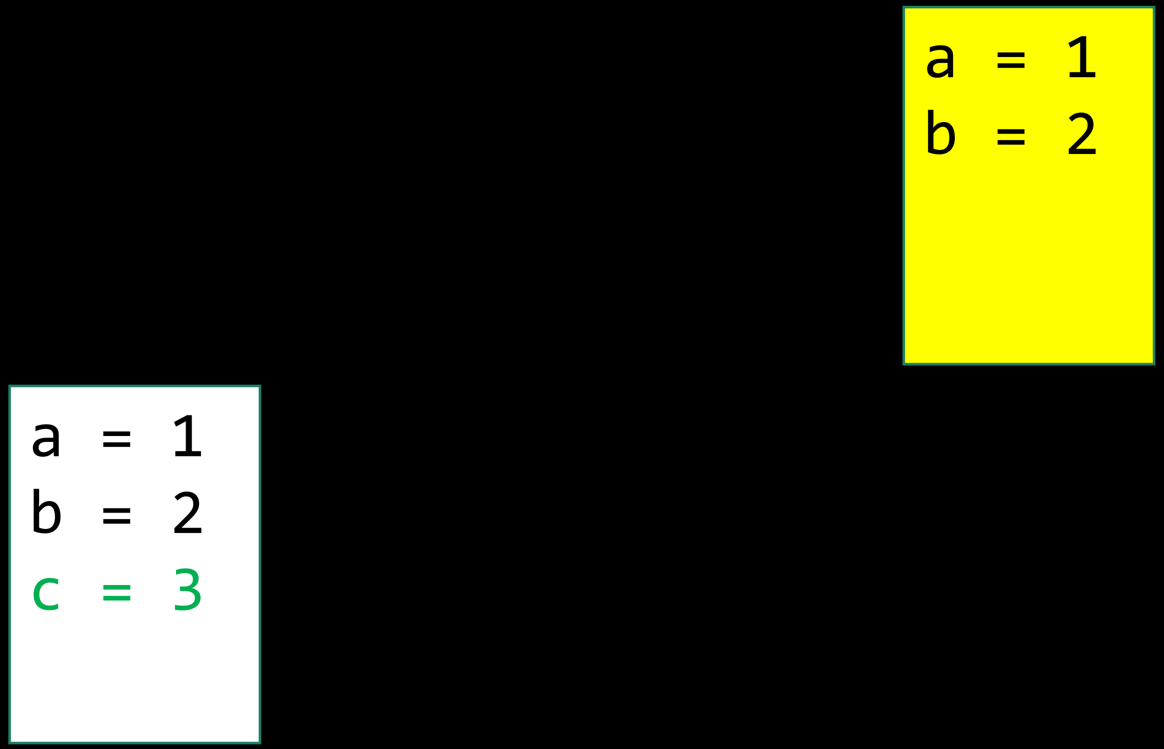


```
a = 1
b = 2
```

```
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.

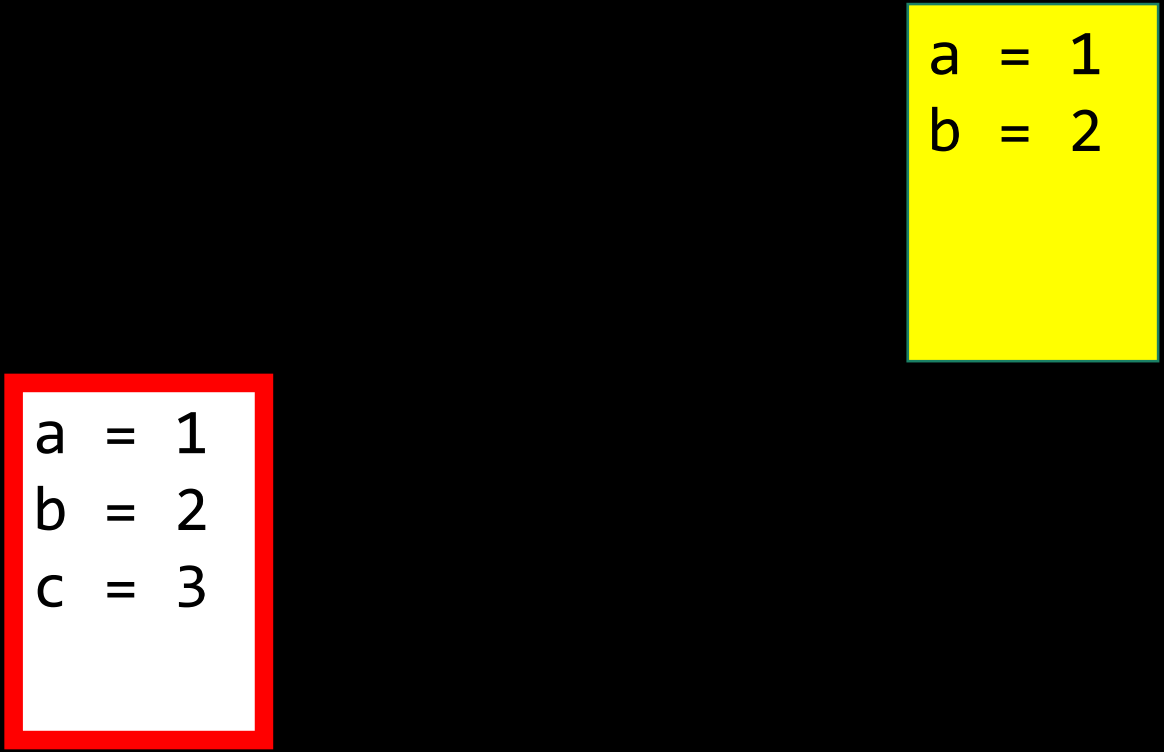


```
a = 1
b = 2
```

```
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.

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

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
```

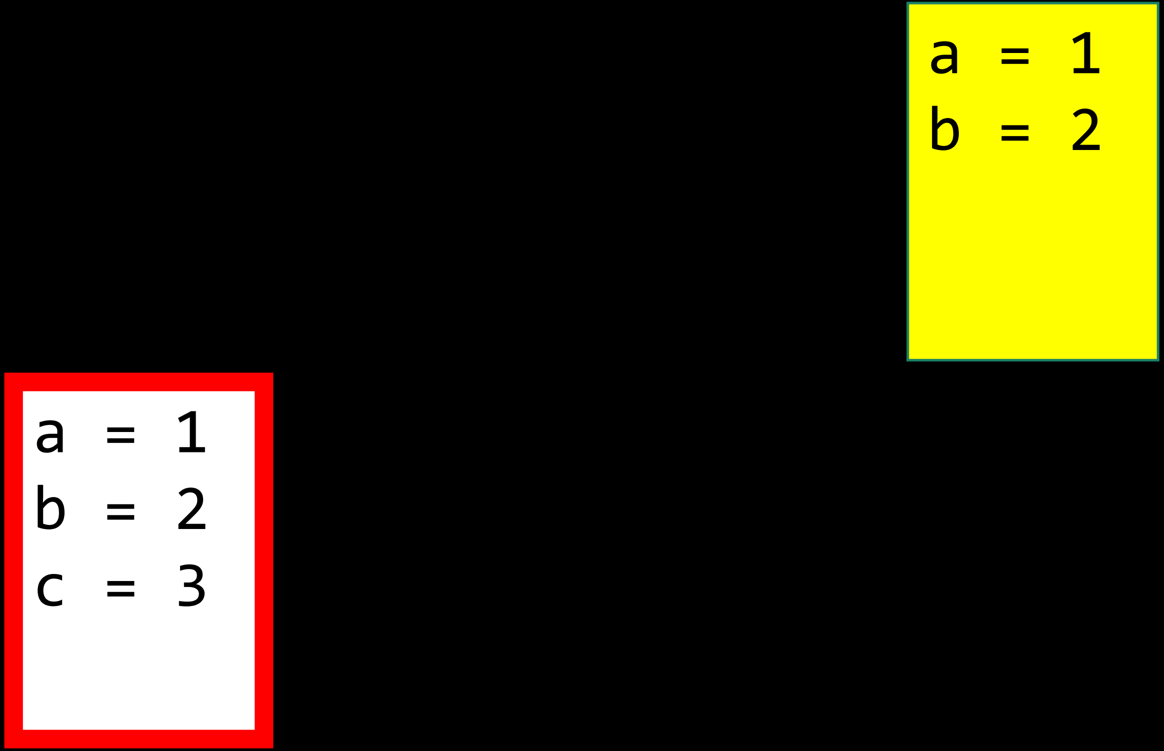
git status

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

```
git commit
```

git commit

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

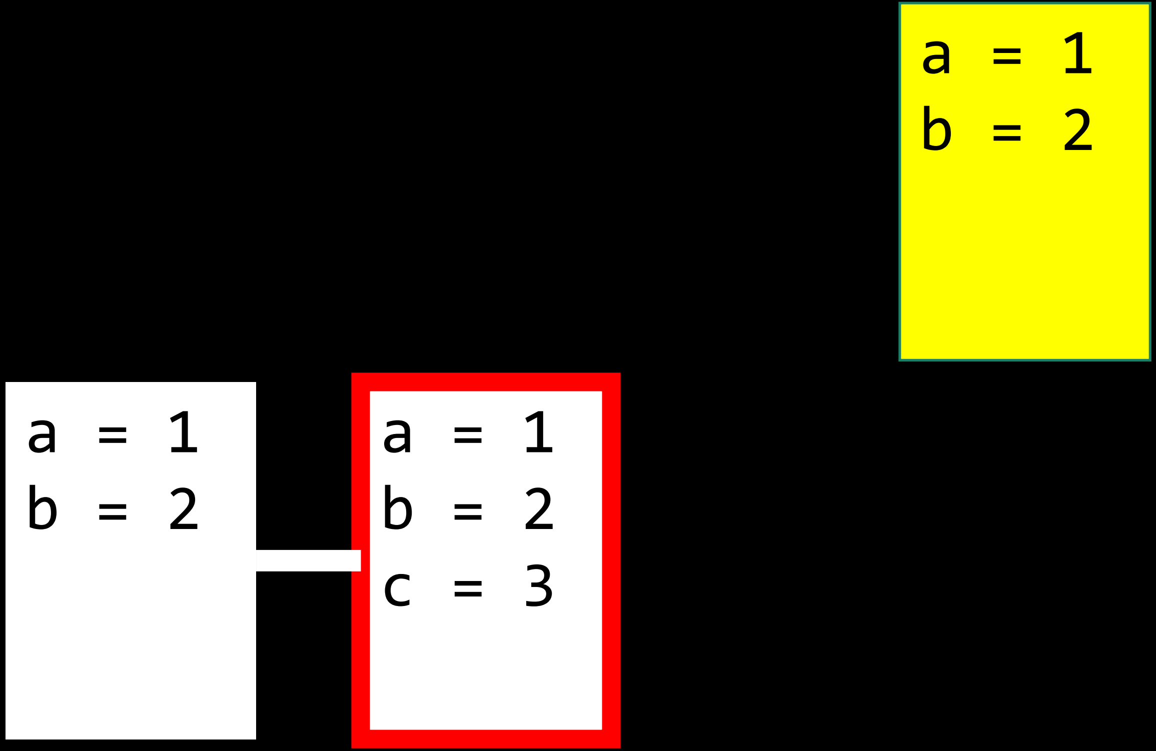


```
a = 1
b = 2
```

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

git commit

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



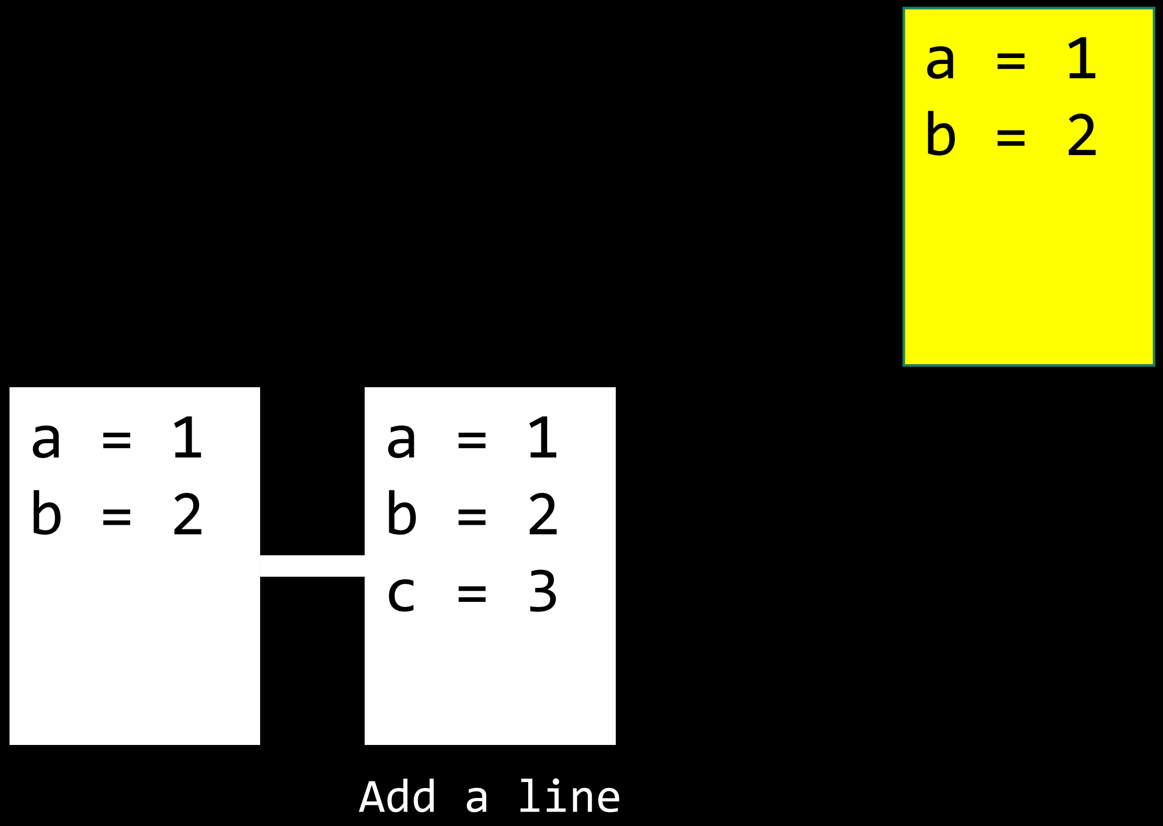
```
a = 1
b = 2
```

```
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.

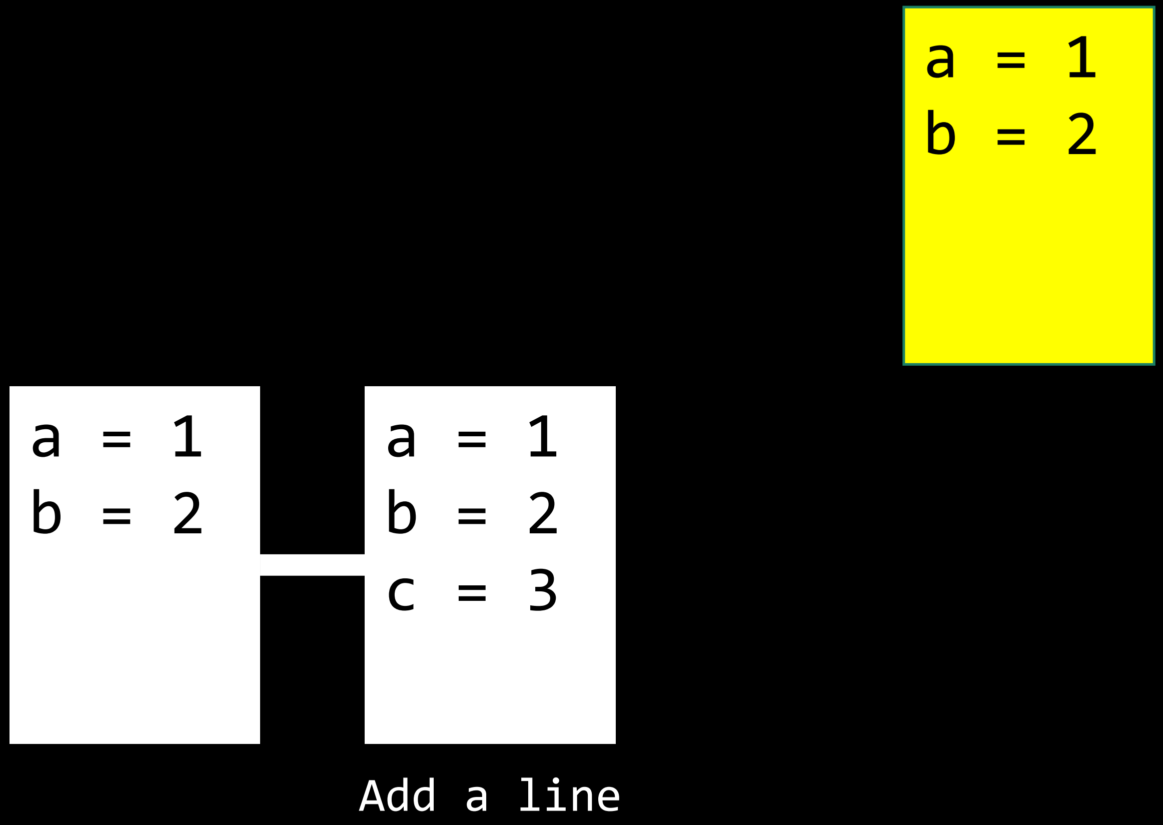
```
git commit –m "message"
git commit –m "Add a line"
git push
```

`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
```

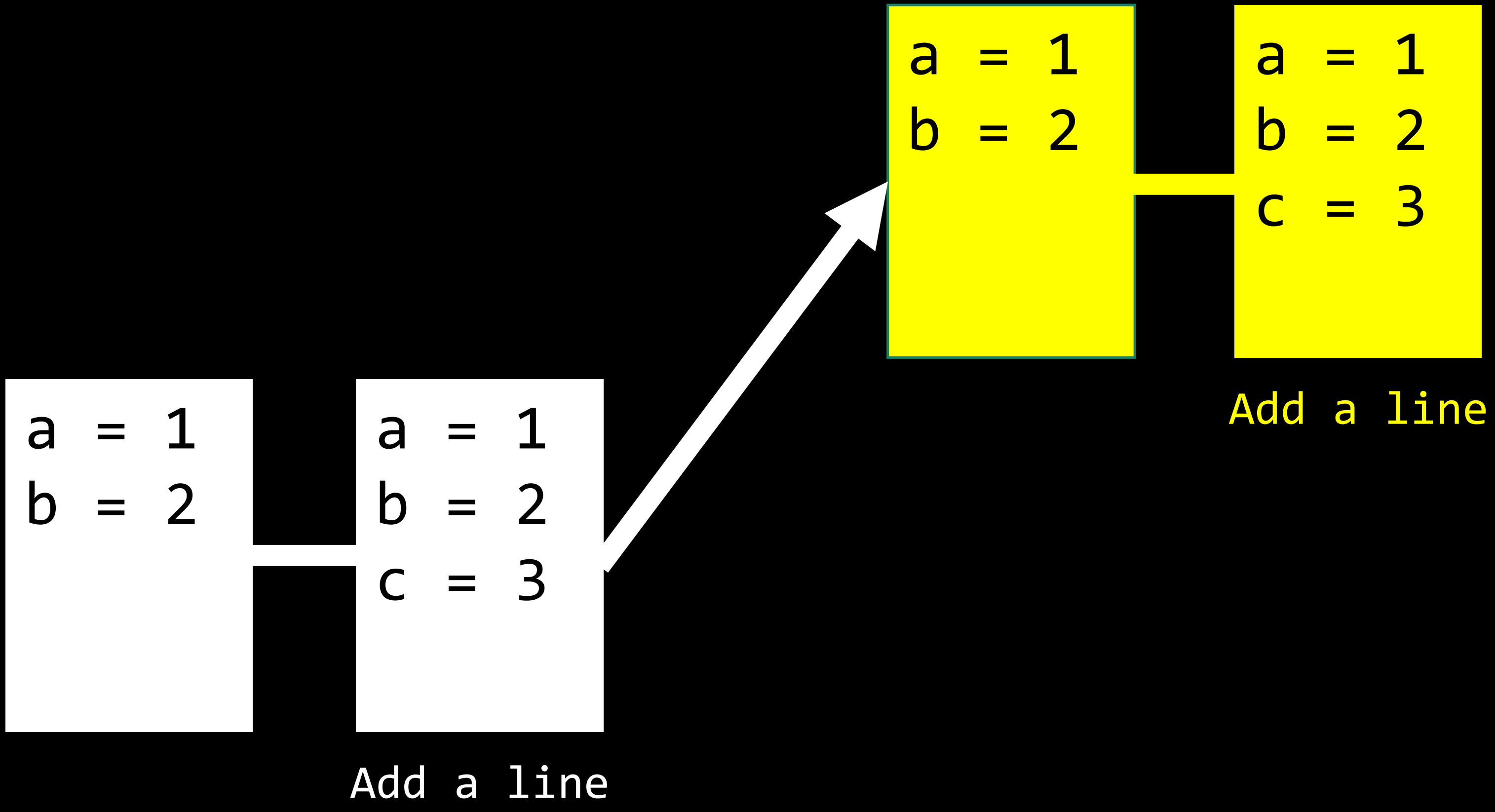
```
a = 1
b = 2
```

```
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).



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

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

Add a line

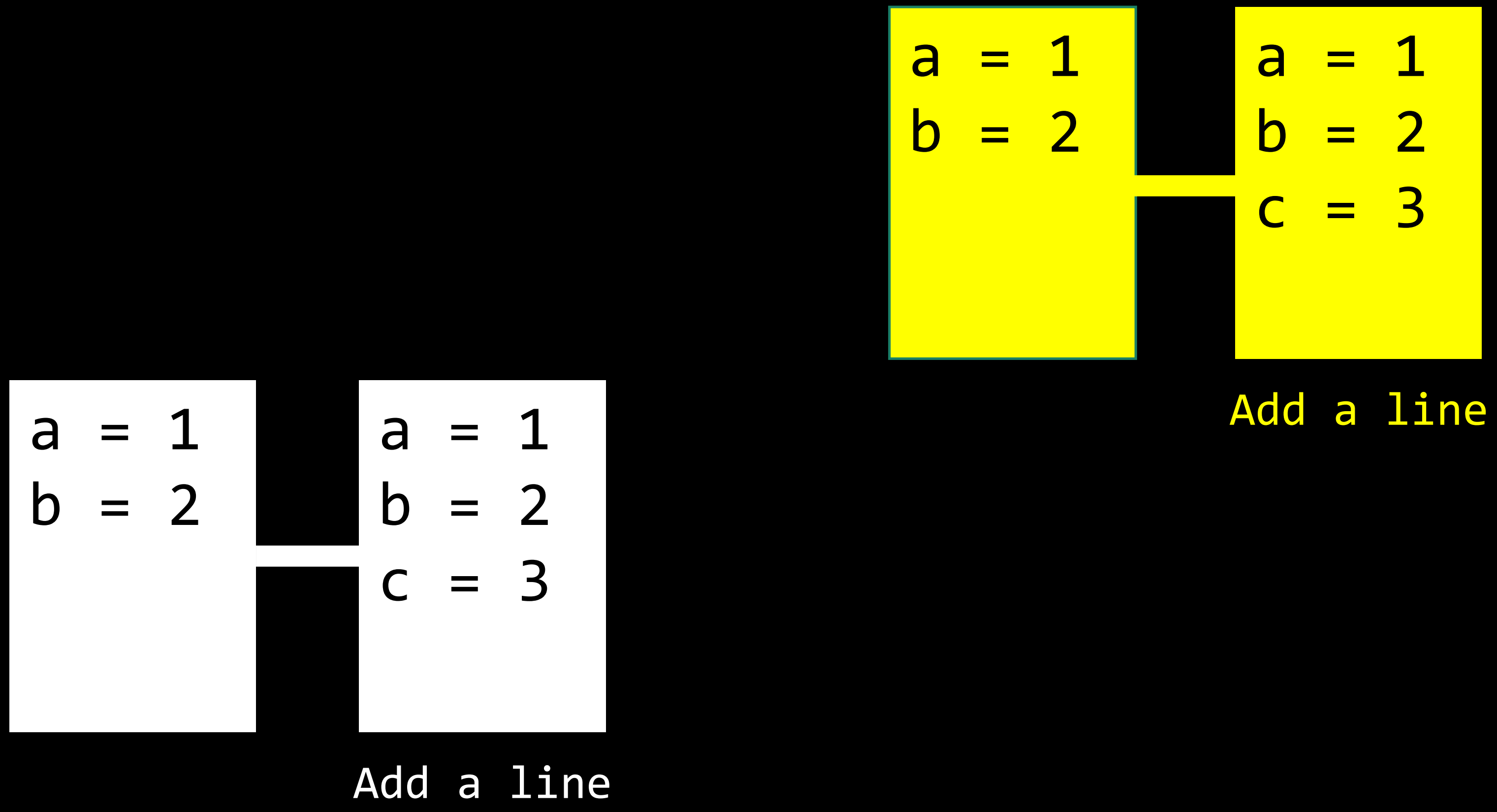
```
a = 1
b = 2
```

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

Add a line

git 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

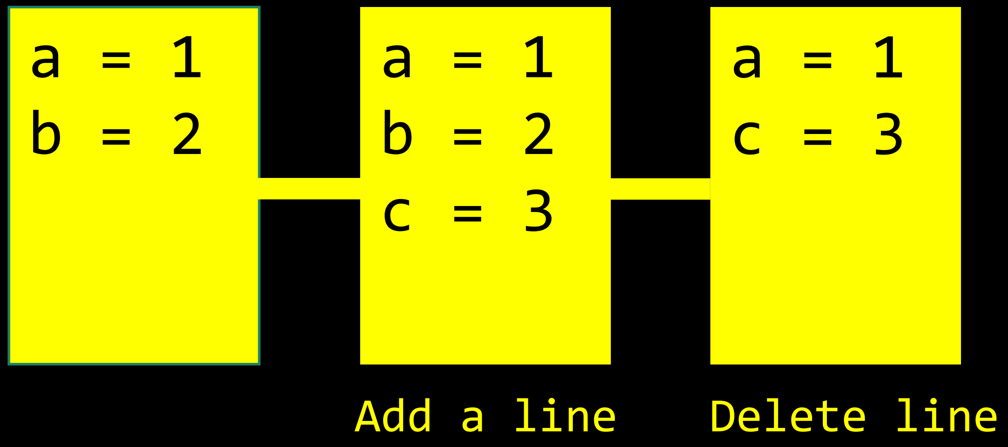
```
git pull
```

`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.



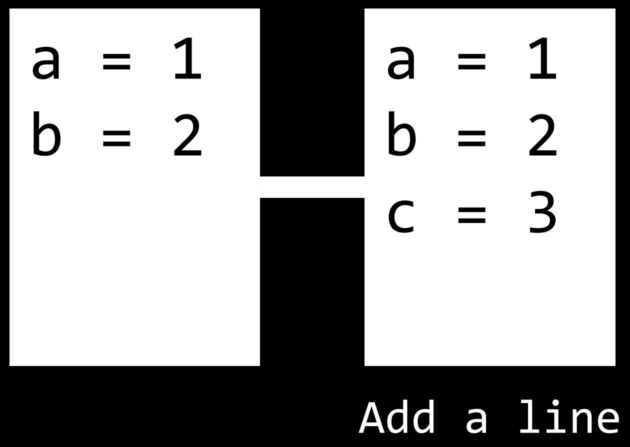
```
a = 1
b = 2
```

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

Add a line

```
a = 1
c = 3
```

Delete line



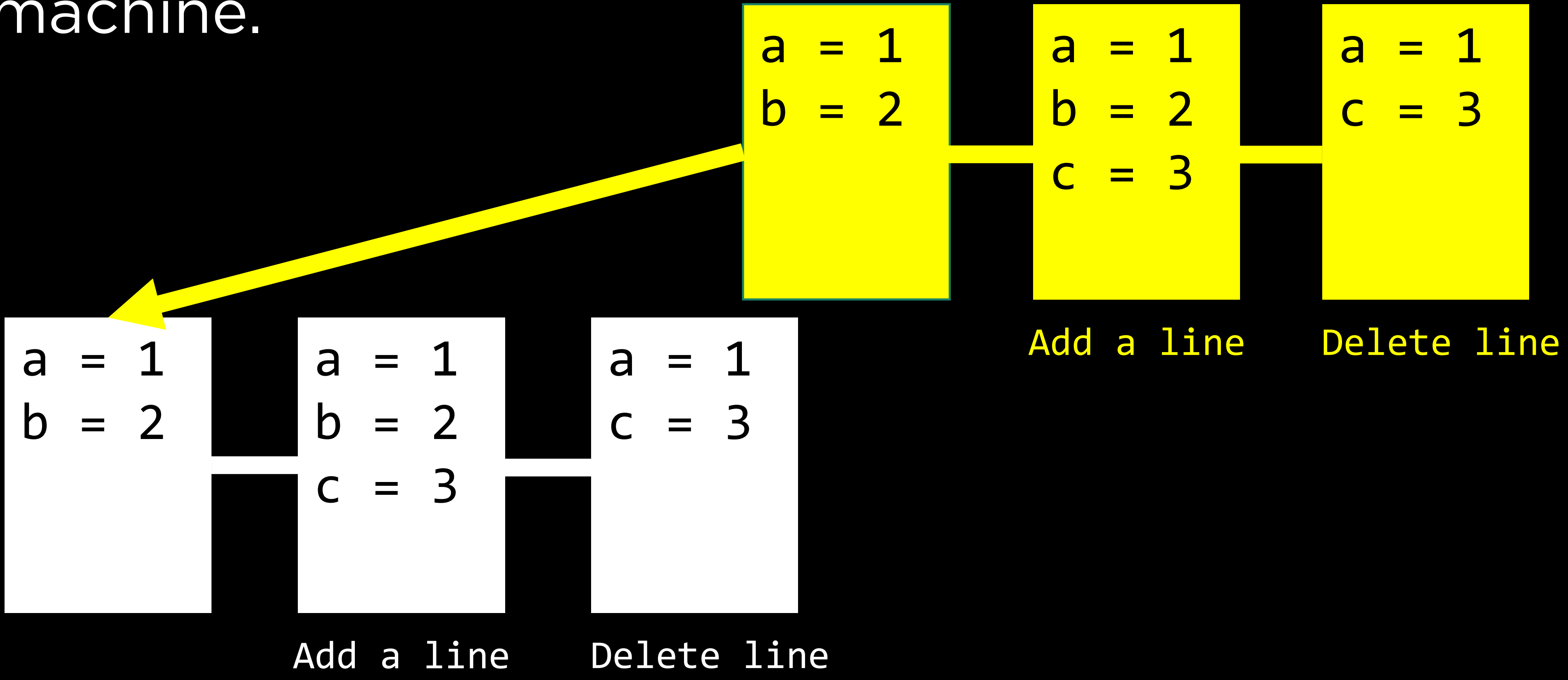
```
a = 1
b = 2
```

```
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.



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

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

Add a line

```
a = 1
c = 3
```

Delete line

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

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

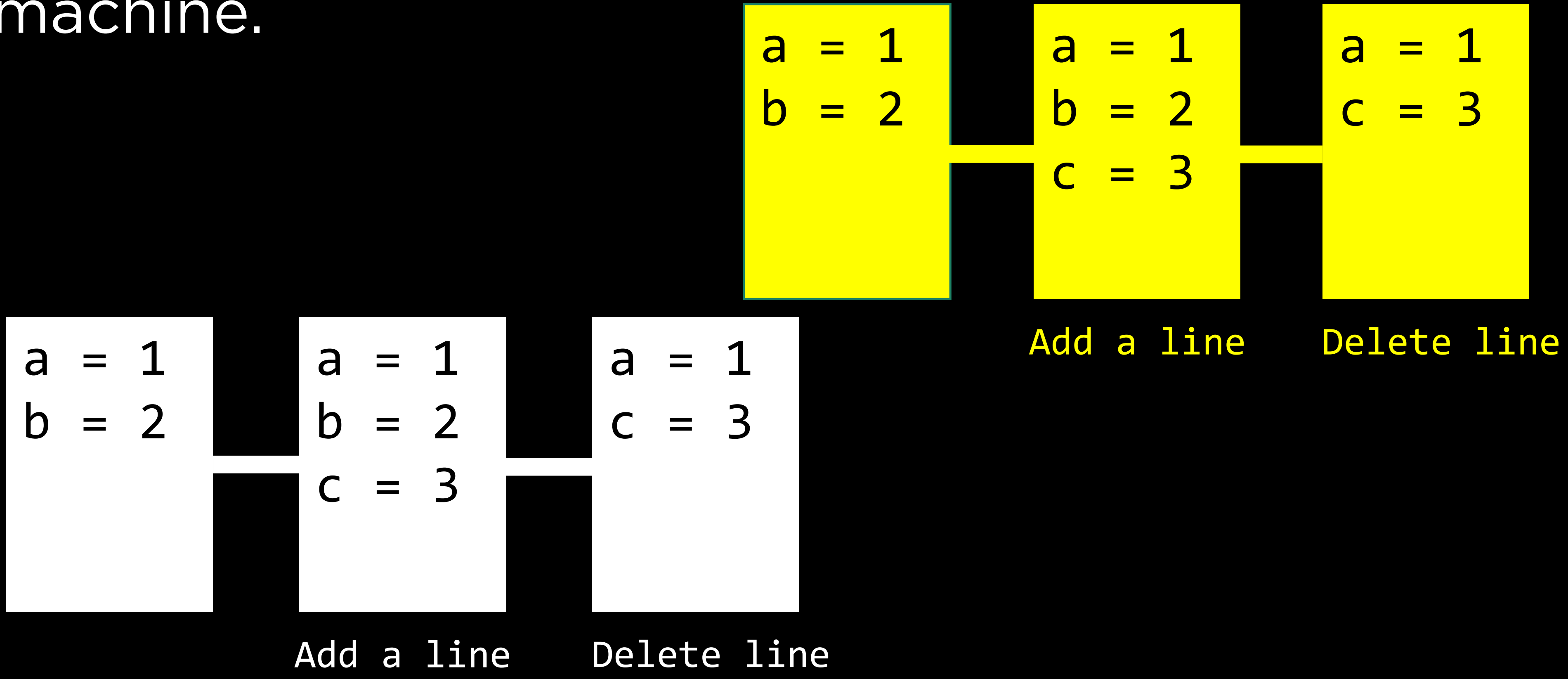
Add a line

```
a = 1
c = 3
```

Delete 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.



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

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

Add a line

```
a = 1
c = 3
```

Delete line

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

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

Add a line

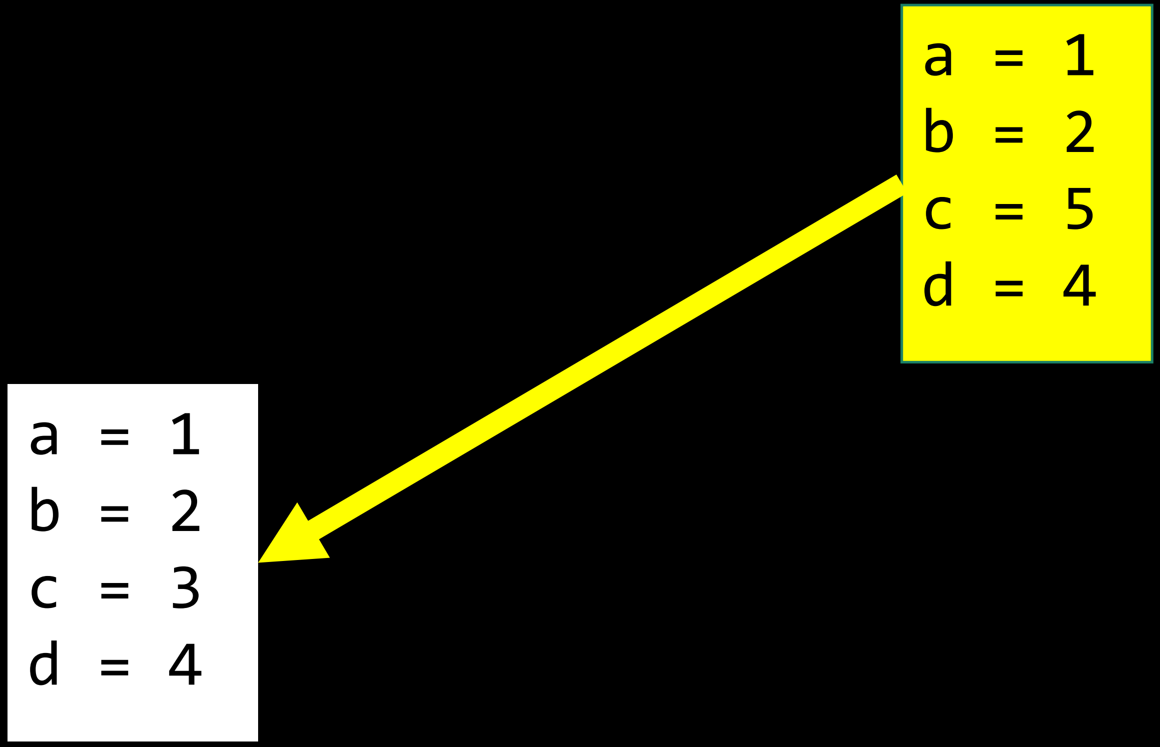
```
a = 1
c = 3
```

Delete line

Conflicts

Conflicts

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

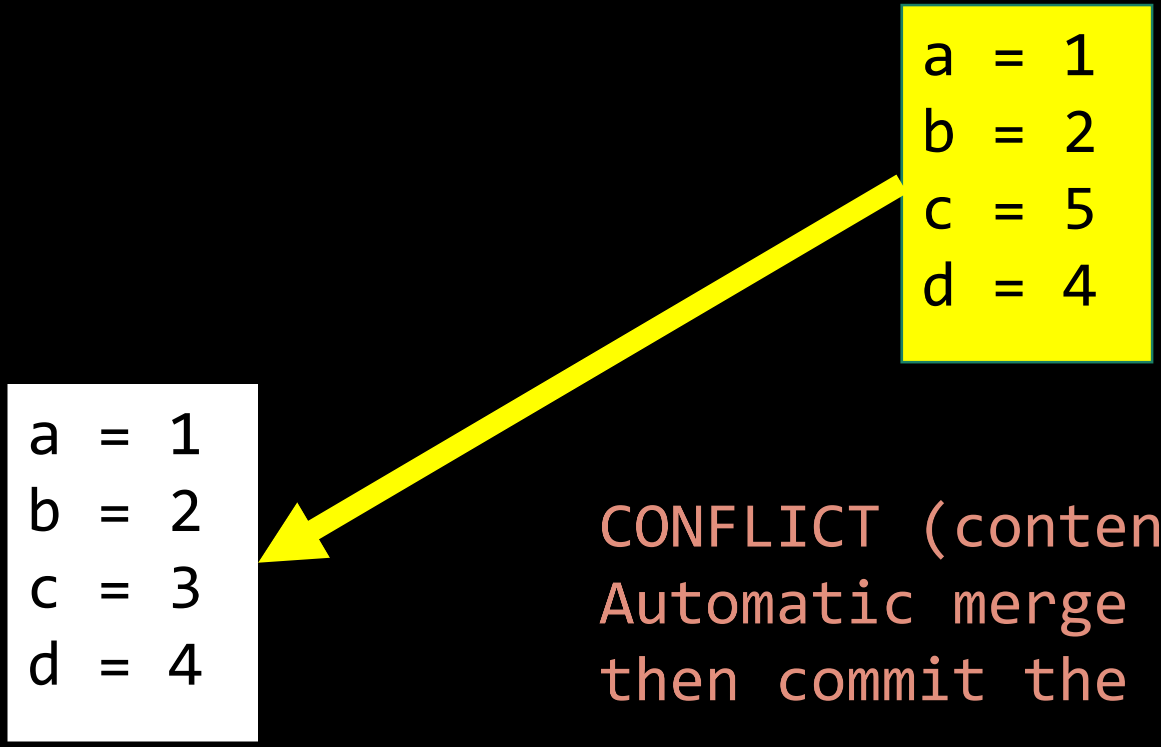


```
a = 1
b = 2
c = 5
d = 4
```

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

Conflicts

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



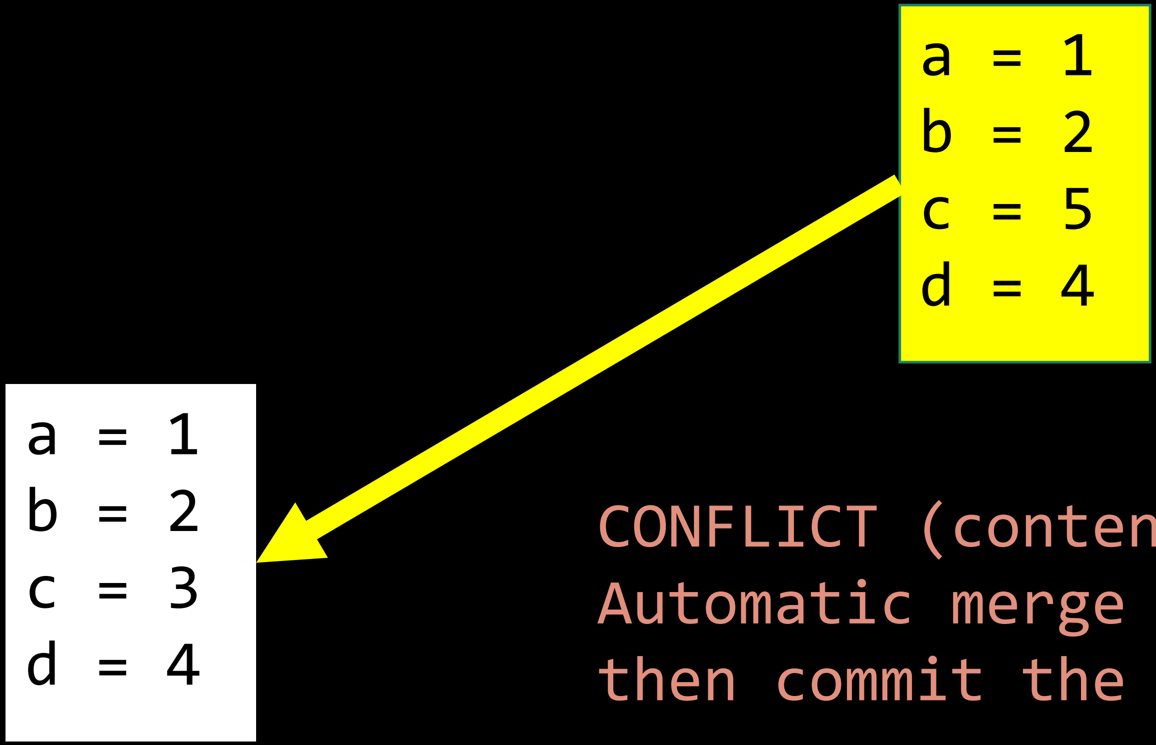
```
a = 1
b = 2
c = 5
d = 4
```

```
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
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
```

HTML

HTML Tags

- `<h1>, <h2>, ..., <h6>`
- `<ul>, <ol>, <li>`
- `<img>`
- `<a>`
- `<table>`
- `<form>`
- `<b>, <i>`
- `<p>`

```
<html>
   <head>
      <title>Hello, world</title>
   </head>
   <body>
      <h2>Here's my page</h2>
      <p>World, hello</p>
   </body>
</html>
```

CSS

CSS

- If HTML is the content of our web pages, then CSS (*Cascading Style Sheets*) are how we will style our sites to make them more aesthetically pleasing.

CSS Properties

- `color`
- `text-align`
- `width`
- `height`
- `margin`
- `padding`
- `font-family, font-size, font-weight`
- `border`

Abstract Elements and Attributes

- `<div></div>`
- `<span></span>`

- `id`
- `class`

GitHub Pages

GitHub Pages

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

GitHub Pages

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

- If that repository's contents happen to describe a website, then you now have a URL friends and family can visit.

GitHub Pages

- GitHub Pages is a feature of GitHub that effectively allows you to deploy the contents of a repository to live on the internet.
- If that repository's contents happen to describe a website, then you now have a URL friends and family can visit.
- This is your objective for Lab 0, this week and early next.

Lab 0

cs.harvard.edu/labs/0

Due Wed 9/11 at 11:59am

Lottery

Enrollment limited; apply at
cs.harvard.edu/100 by Saturday

Software Engineering in the Arts and Humanities