

Lecture # 5&6

CSC336 Web Technologies

Credit Hours: 3(2, 1)

Course Instructor: **SAIF ULLAH IJAZ**

Lecturer CS Dept, CUI Vehari

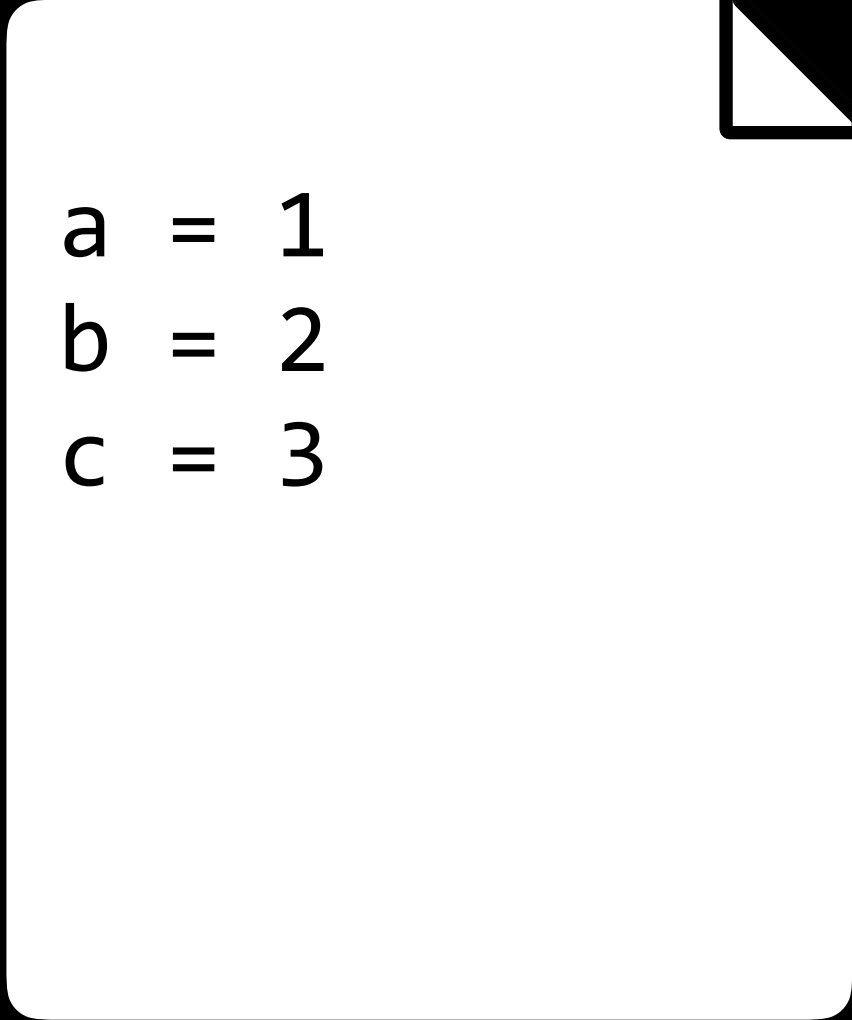
MSc University of Leicester, UK

BSc COMSATS University Islamabad

Git

git

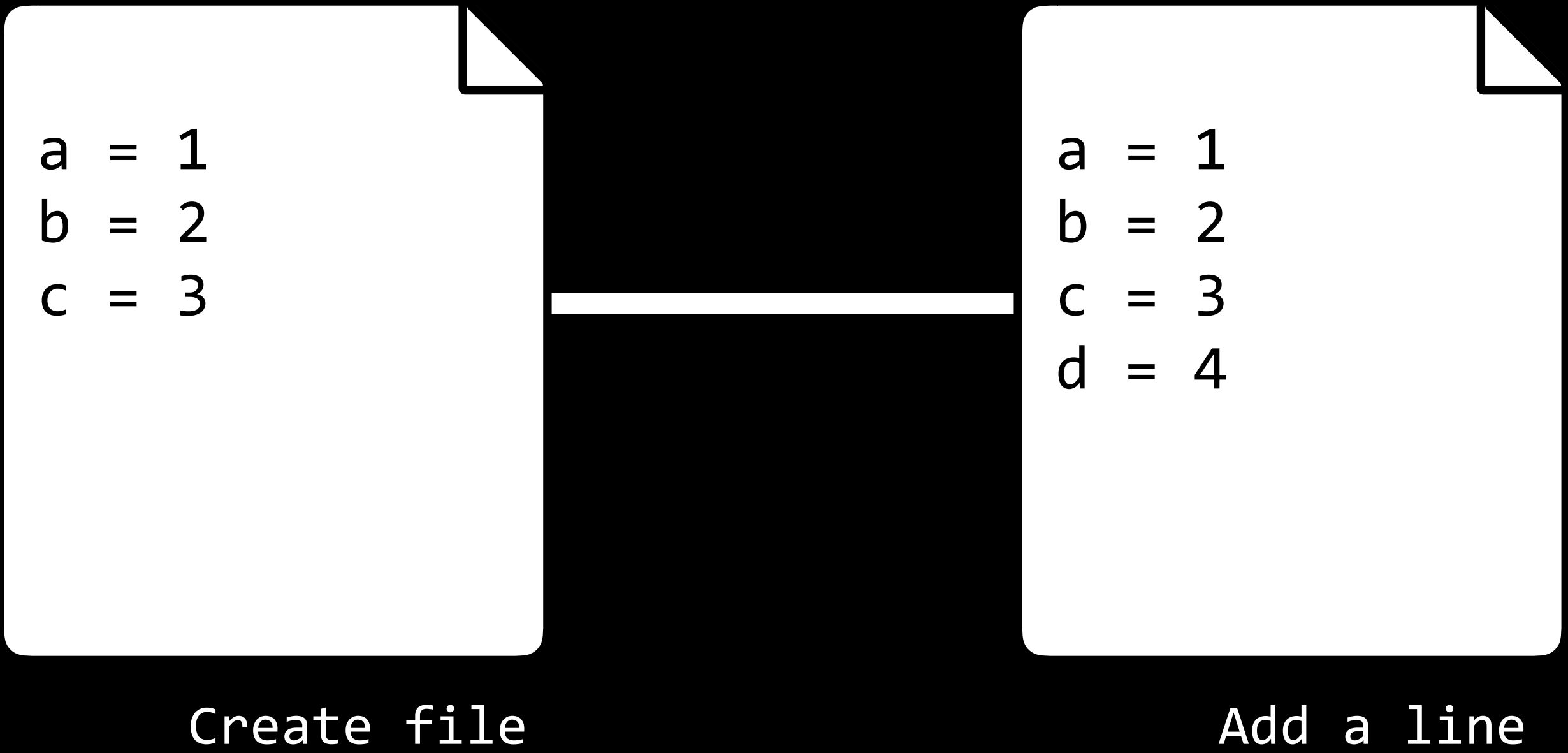
Keep track of changes to code.



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

Create file

Keep track of changes to code.



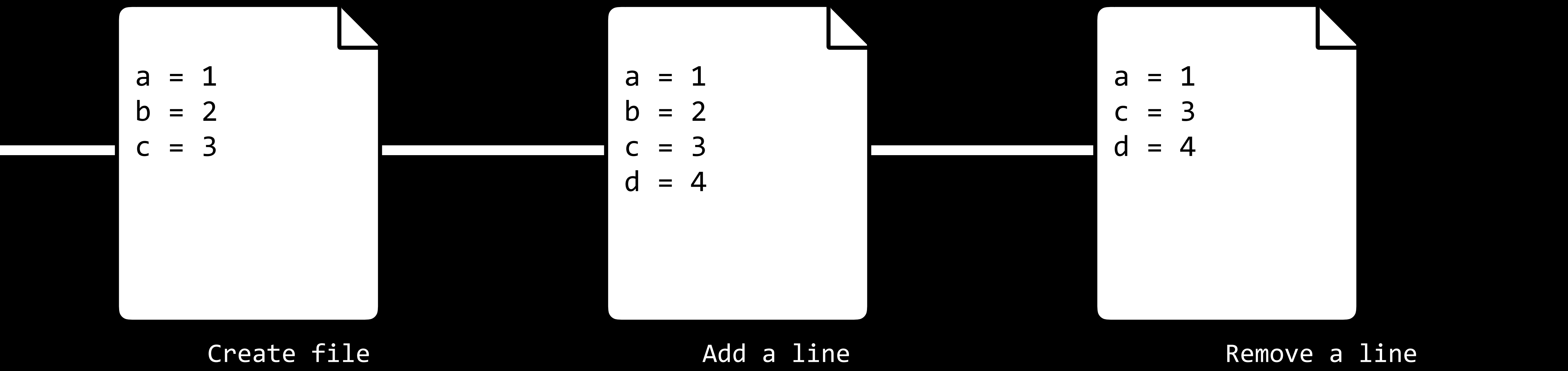
```
a = 1  
b = 2  
c = 3
```

Create file

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

Add a line

Keep track of changes to code.



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

The diagram illustrates a sequence of three code states connected by horizontal lines. The first state, labeled 'Create file', shows three lines of code: 'a = 1', 'b = 2', and 'c = 3'. The second state, labeled 'Add a line', shows the same three lines plus a new line 'd = 4'. The third state, labeled 'Remove a line', shows the first two lines 'a = 1' and 'b = 2', but the third line is now 'c = 3' and 'd = 4', indicating that the original third line has been replaced or removed and a new one added. Each state is represented by a white document icon with a folded top-right corner.

Create file

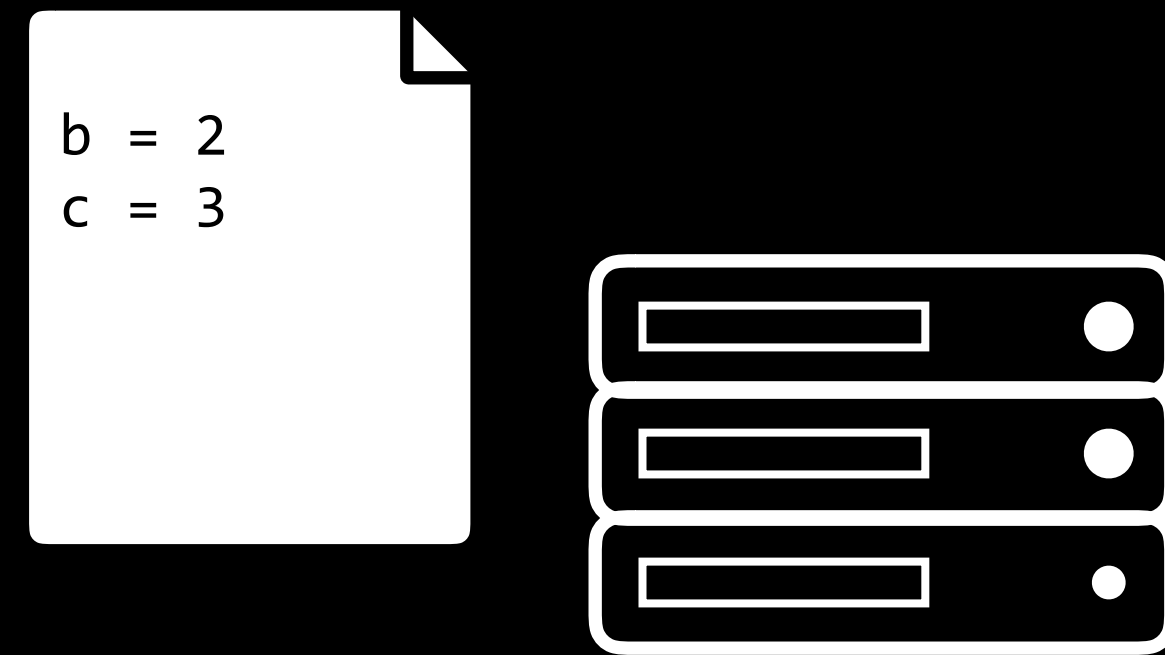
```
a = 1  
b = 2  
c = 3  
d = 4
```

Add a line

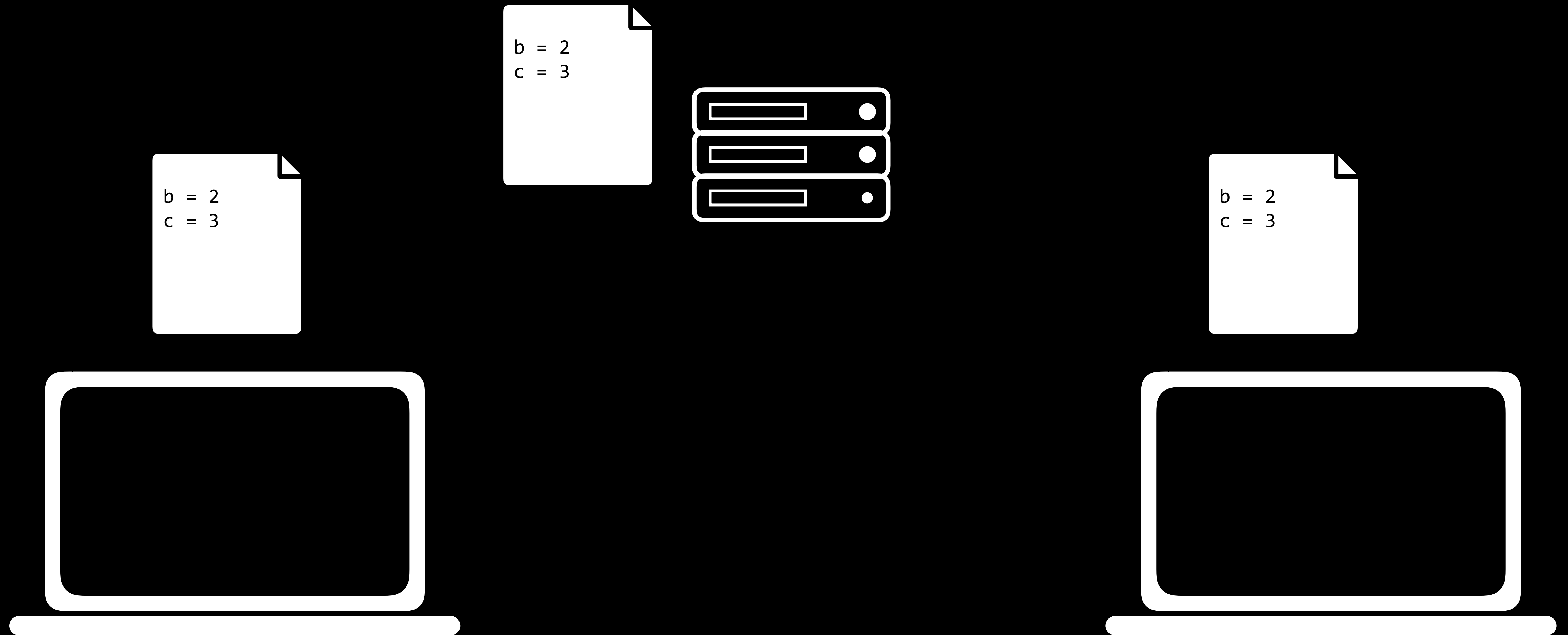
```
a = 1  
b = 2  
c = 3  
d = 4
```

Remove a line

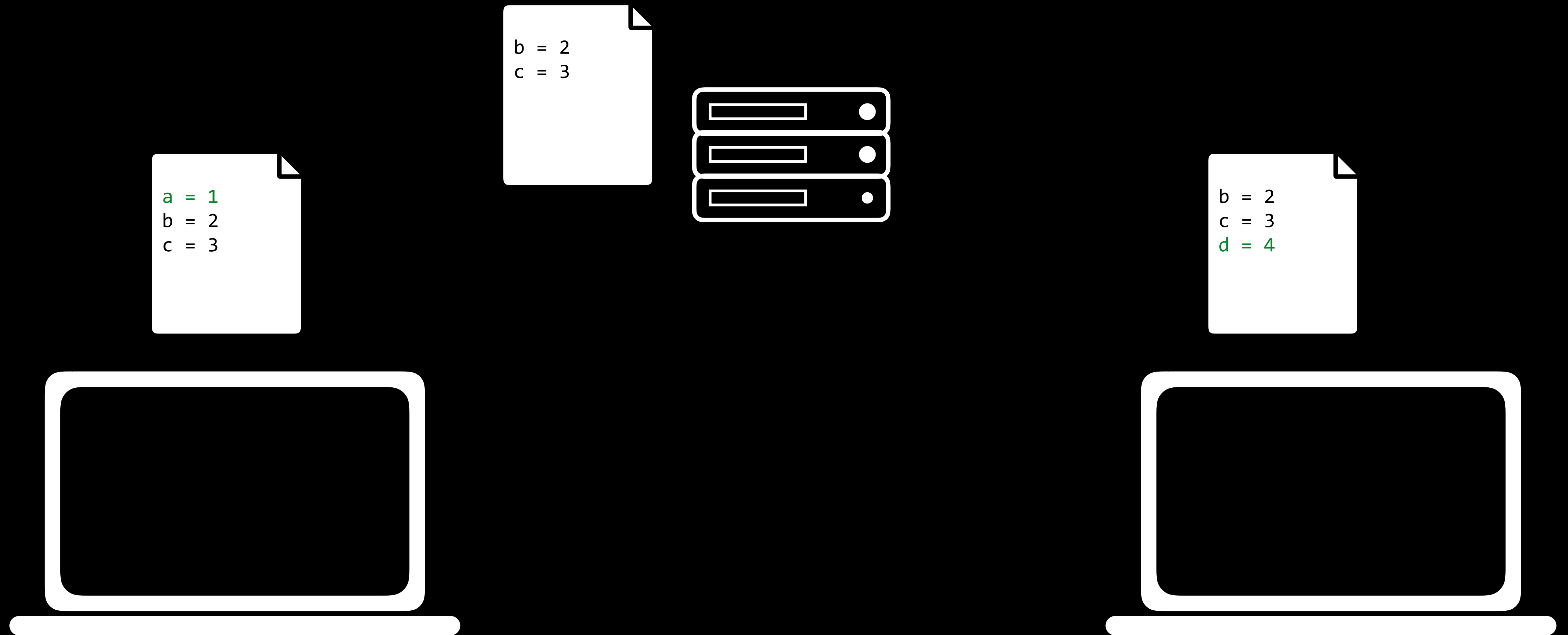
Synchronizes code between different people.



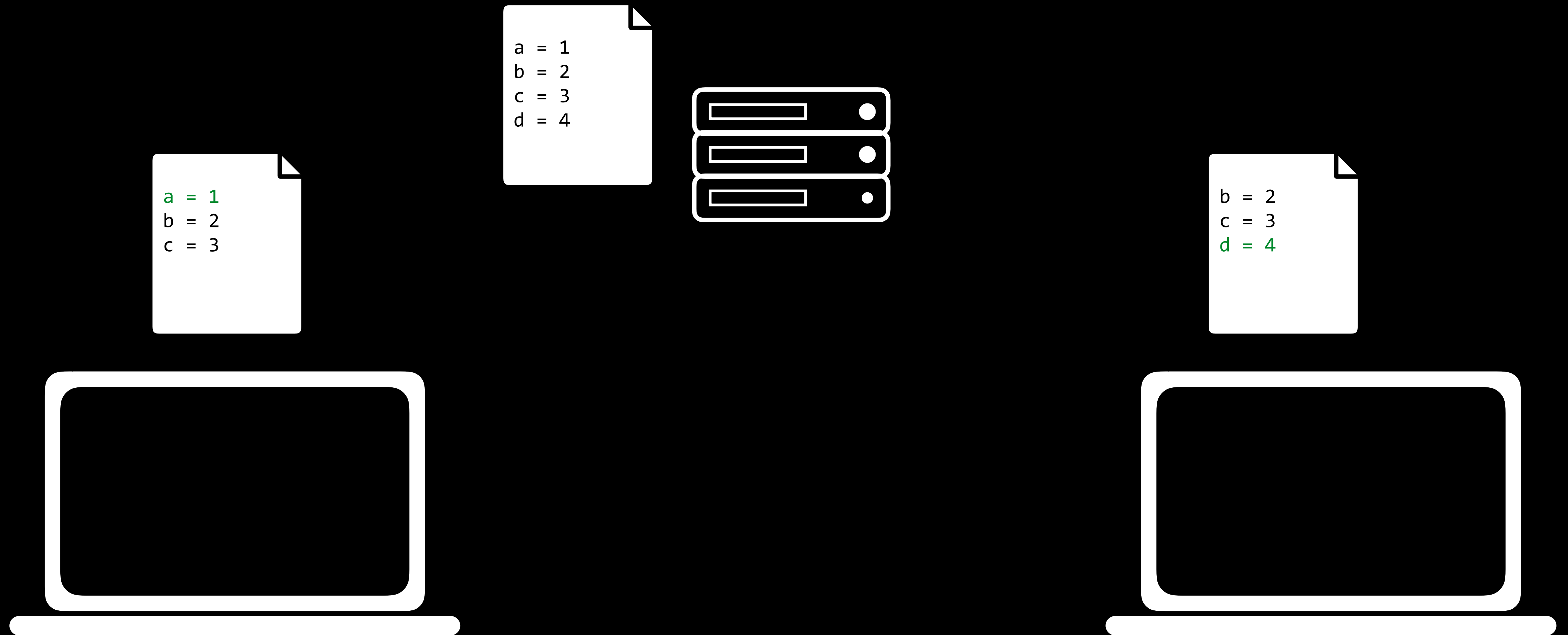
Synchronizes code between different people.



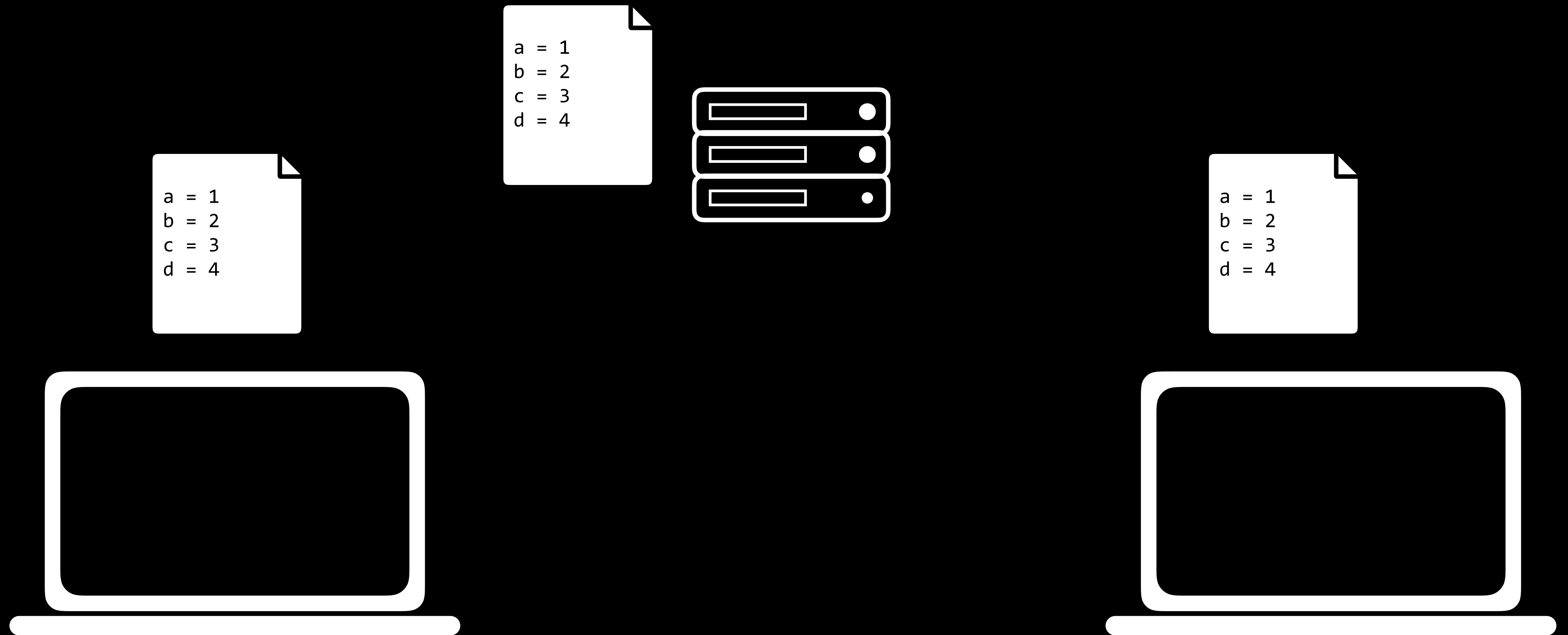
Synchronizes code between different people.



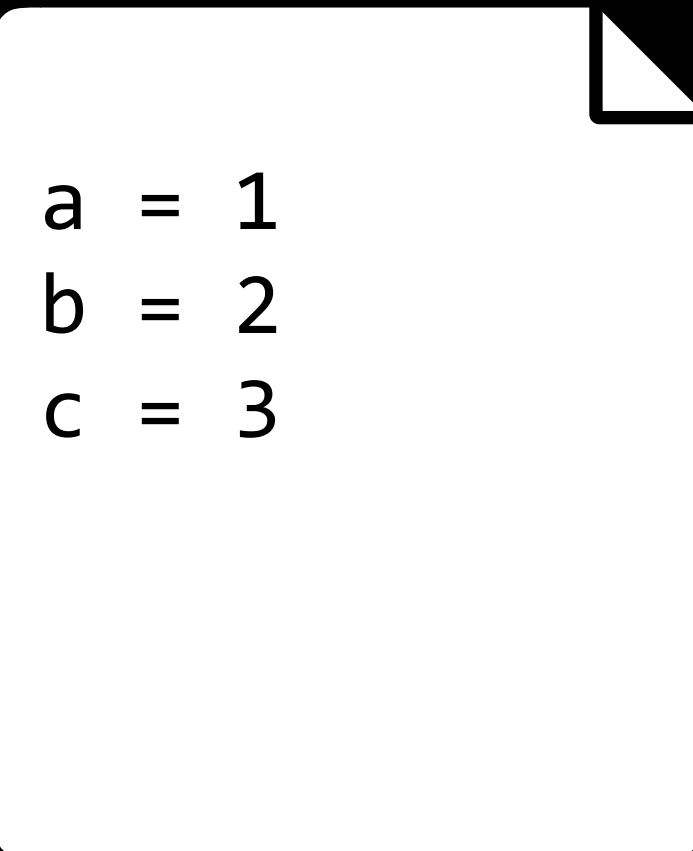
Synchronizes code between different people.



Synchronizes code between different people.

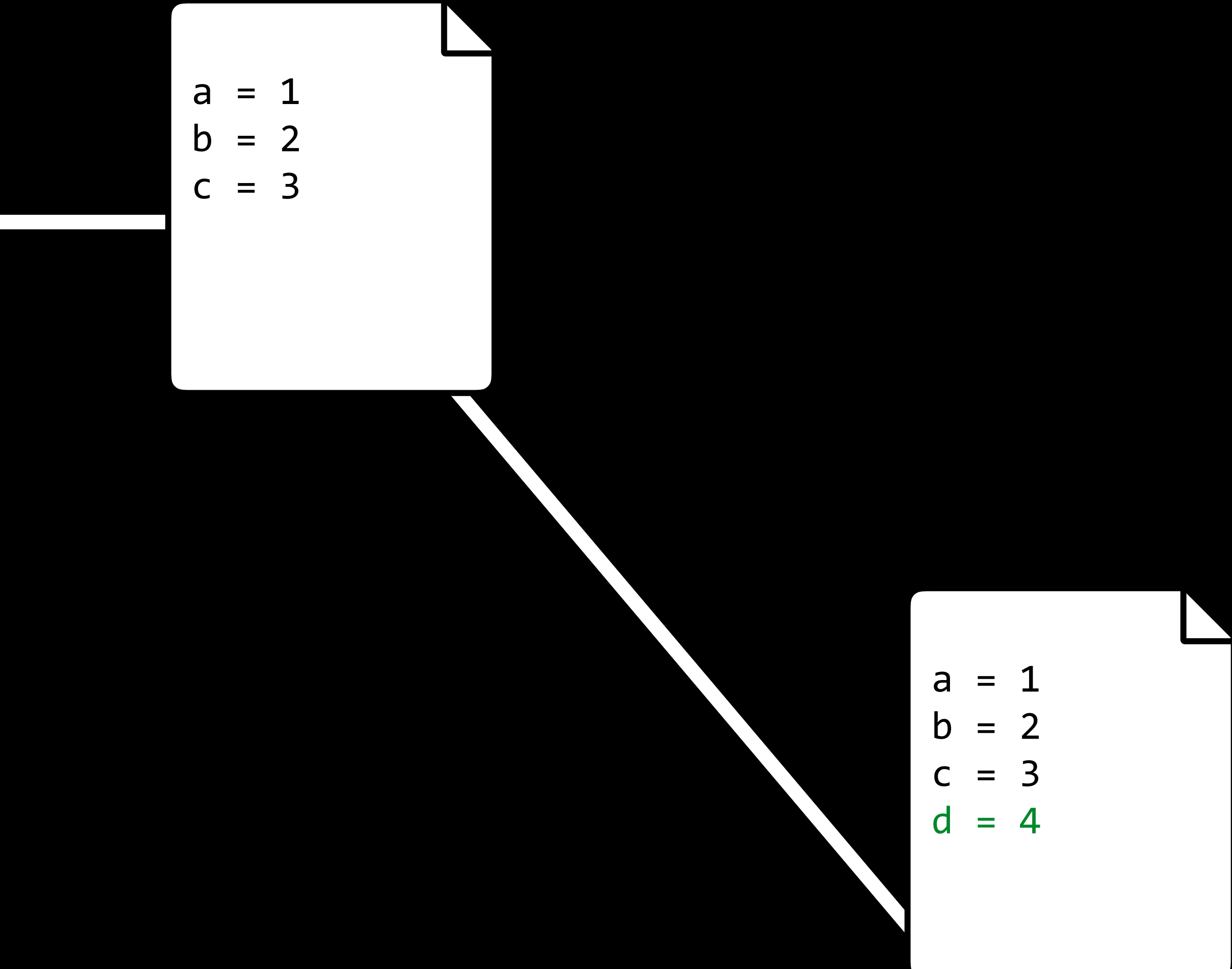


Test changes to code without losing the original.



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

Test changes to code without losing the original.

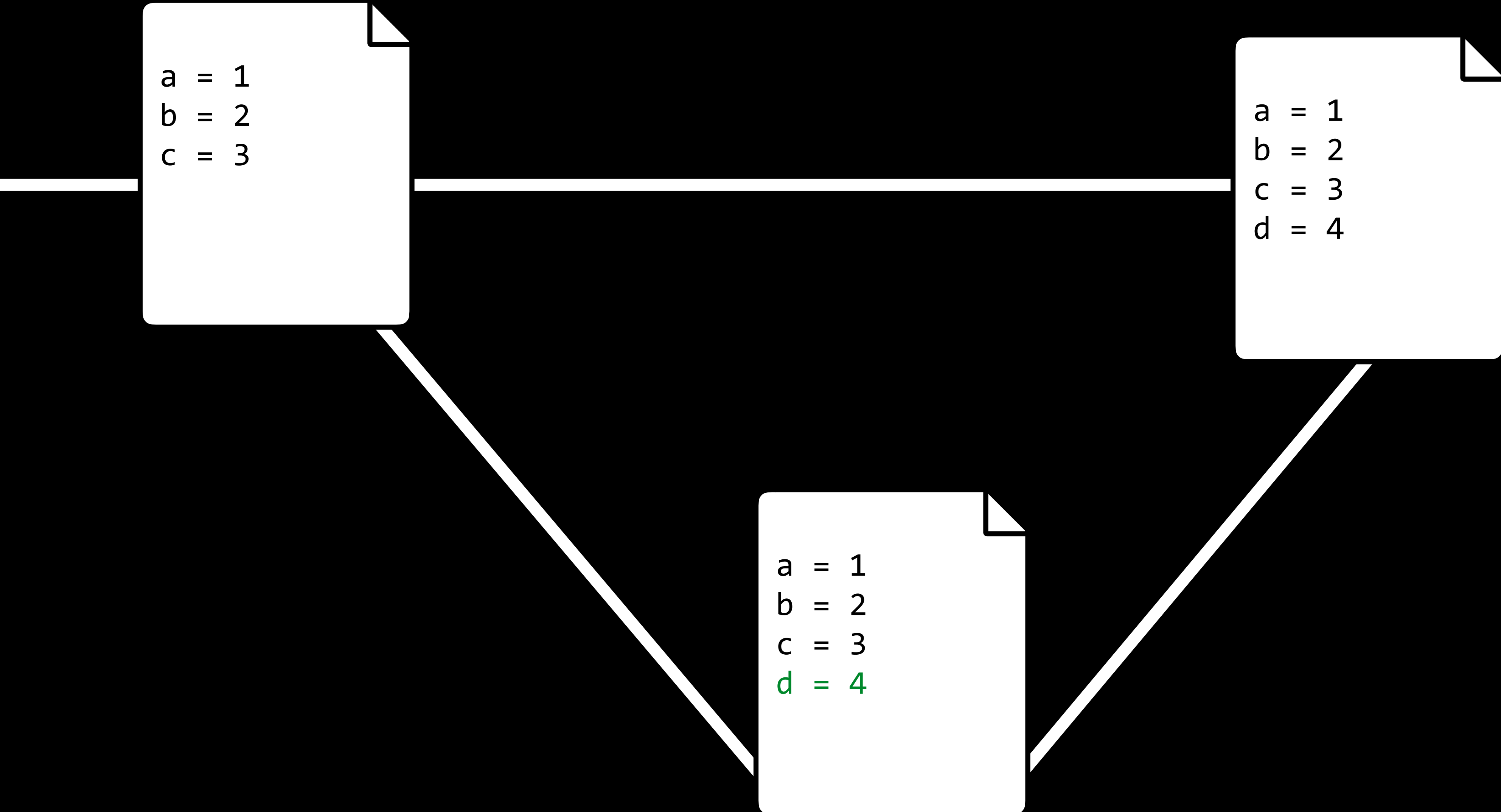


The diagram illustrates a workflow for testing code changes. It features two white rectangular boxes with a folded top-right corner, representing code files. The top-left box contains the original code: `a = 1`, `b = 2`, and `c = 3`. A horizontal line extends from the left edge of this box. A diagonal line connects the bottom-right corner of the top-left box to the top-left corner of the bottom-right box. The bottom-right box contains the same original code, but with an additional line: `d = 4`, which is highlighted in green. This represents a test change being applied to the original code.

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

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

Test changes to code without losing the original.



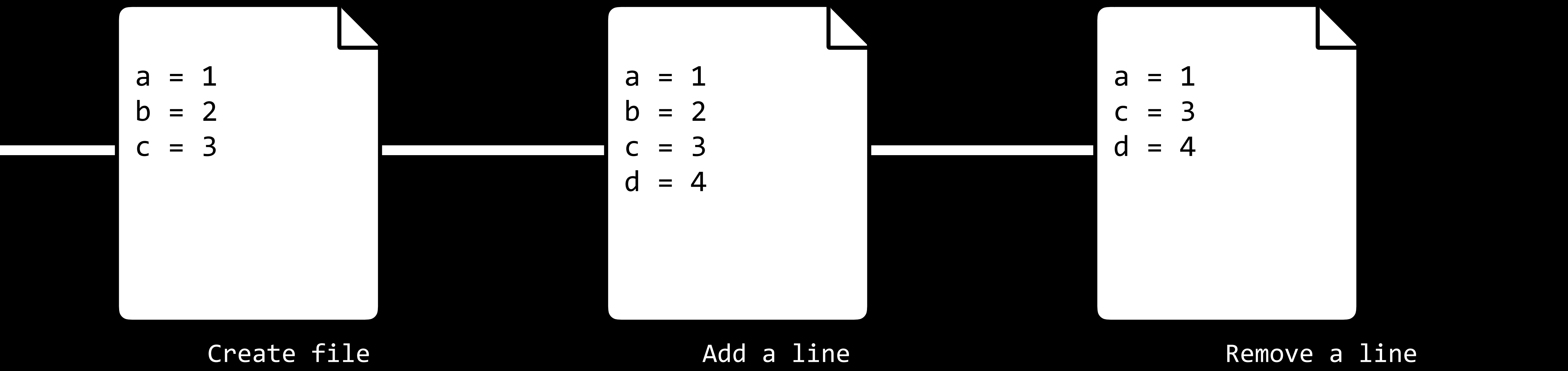
```
graph LR; A["a = 1  
b = 2  
c = 3"] --- B["a = 1  
b = 2  
c = 3  
d = 4"]; A --- C["a = 1  
b = 2  
c = 3"]; B --- D["a = 1  
b = 2  
c = 3  
d = 4"]; C --- D
```

a = 1
b = 2
c = 3

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

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

Revert back to old versions of code.



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

Create file

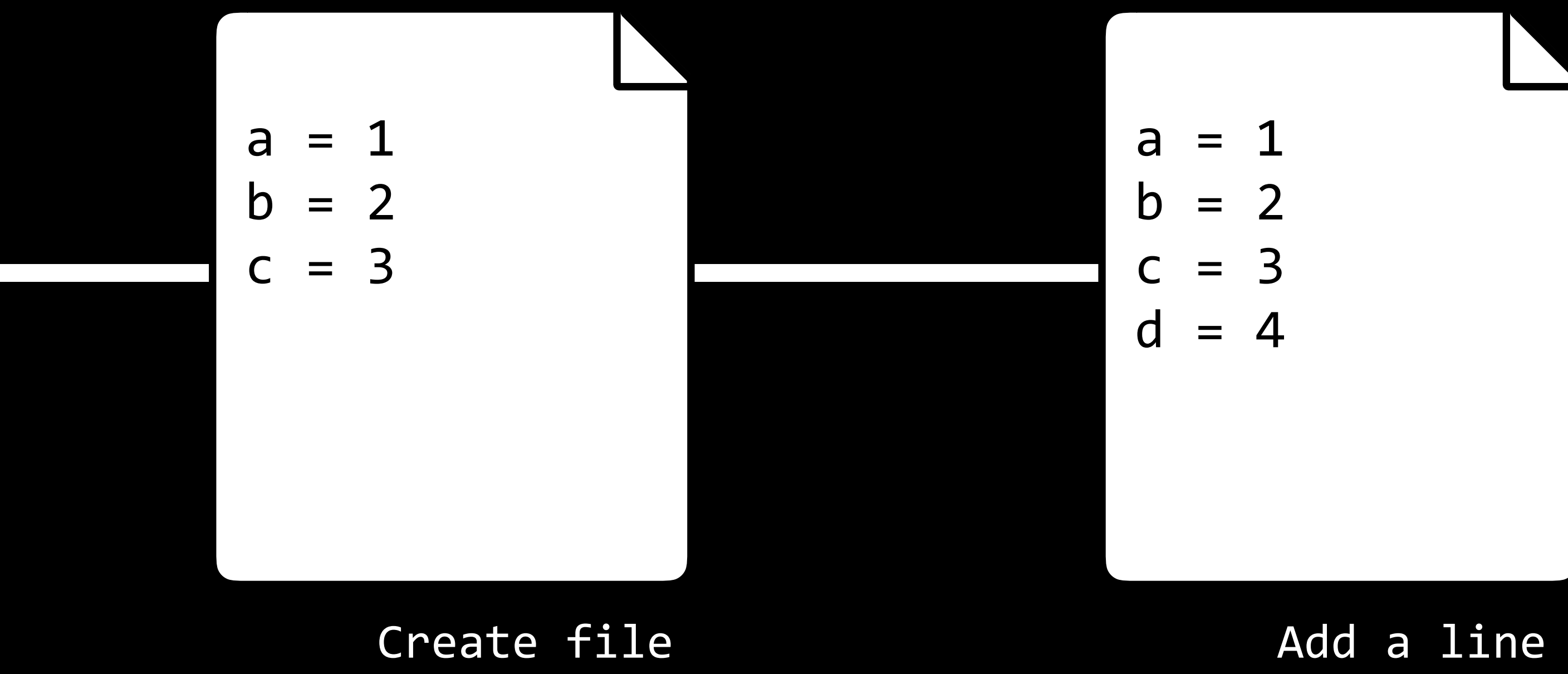
```
a = 1  
b = 2  
c = 3  
d = 4
```

Add a line

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

Remove a line

Revert back to old versions of code.



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



Create file

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

Add a line

GitHub

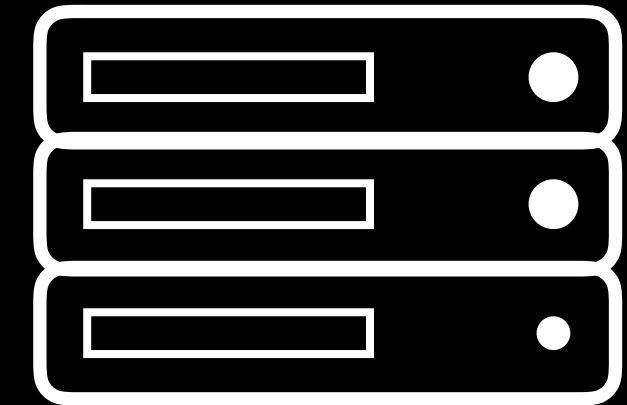
Git Vs GitHub

GIT	GITHUB
 <p data-bbox="902 1140 1316 1202">It is a software</p>	 <p data-bbox="2192 1149 2548 1211">It is a service</p>
It is installed locally on the system	It is hosted on Web
It is a tool to manage different versions of edits, made to files in a git repository	It is a space to upload a copy of the Git repository
It is a command-line tool	It provides a graphical interface

```
git clone
```

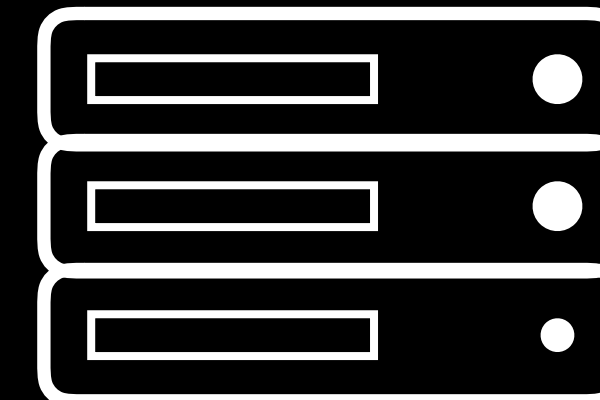
```
git clone <url>
```

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



```
git clone <url>
```

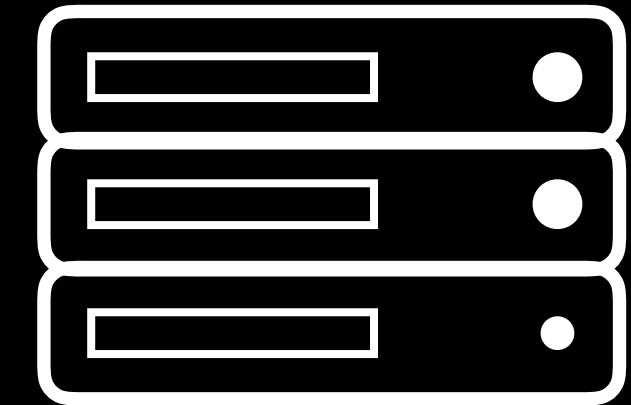
```
a = 1  
b = 2  
c = 3  
d = 4
```



```
git clone <url>
```

```
git clone <url>
```

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



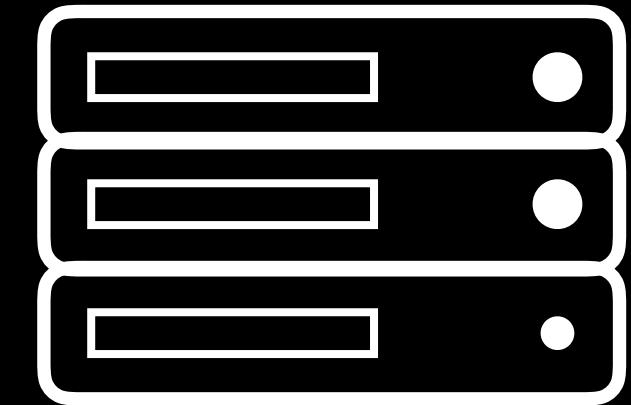
```
git clone <url>
```

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

```
git add
```

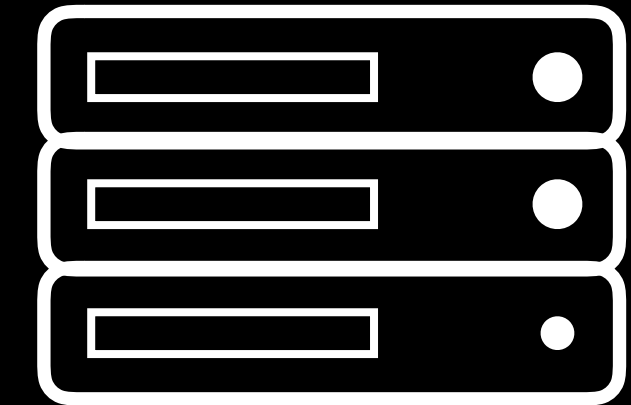
```
git add <filename>
```

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




```
git add <filename>
```

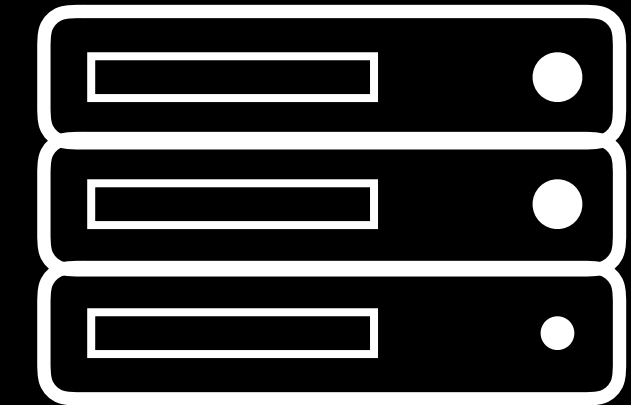
```
a = 1  
b = 2  
c = 3  
d = 4
```



```
a = 1  
b = 2  
c = 3  
d = 4  
e = 5
```

```
git add <filename>
```

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

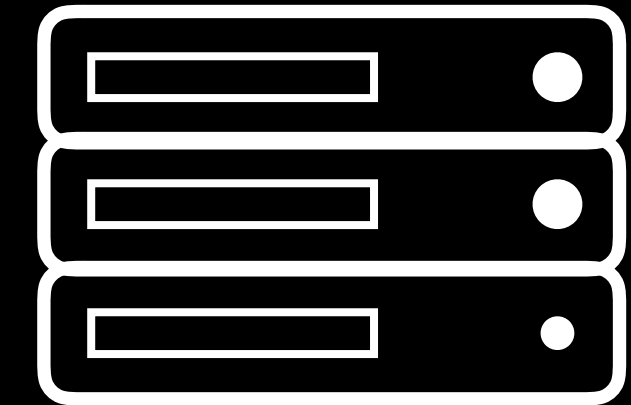


```
git add foo.py
```

```
a = 1  
b = 2  
c = 3  
d = 4  
e = 5
```

```
git add <filename>
```

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



```
git add foo.py
```

```
a = 1  
b = 2  
c = 3  
d = 4  
e = 5
```

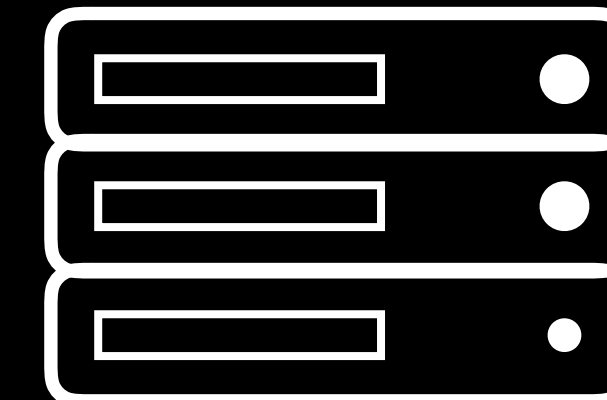
Changes to be committed:

modified: foo.py

git commit

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

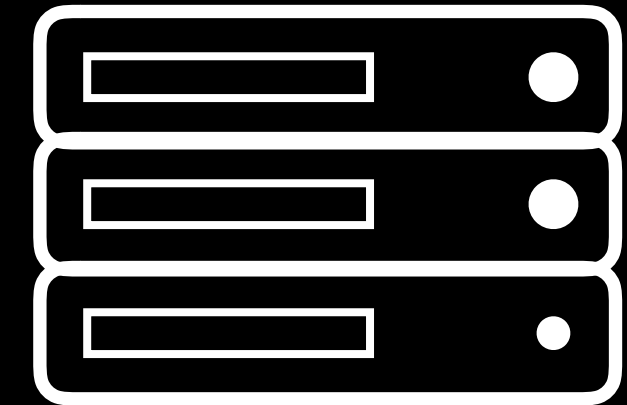
```
a = 1  
b = 2  
c = 3  
d = 4
```



```
a = 1  
b = 2  
c = 3  
d = 4  
e = 5
```

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

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

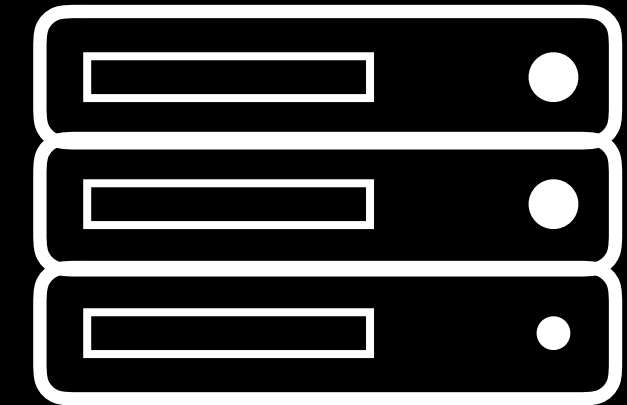


```
git commit -m  
"Add line"
```

```
a = 1  
b = 2  
c = 3  
d = 4  
e = 5
```

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

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



```
git commit -m  
"Add line"
```

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

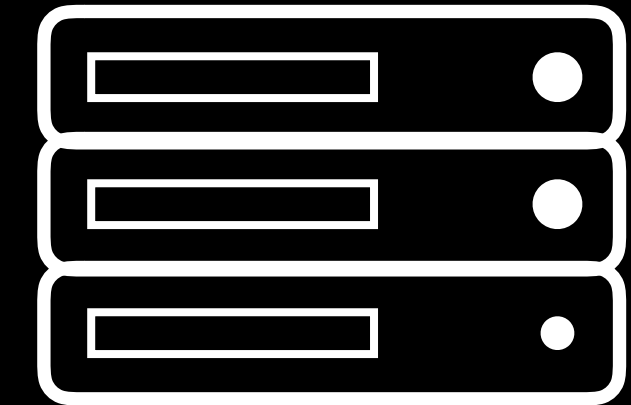
```
a = 1  
b = 2  
c = 3  
d = 4  
e = 5
```

Add line

```
git status
```


git status

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



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

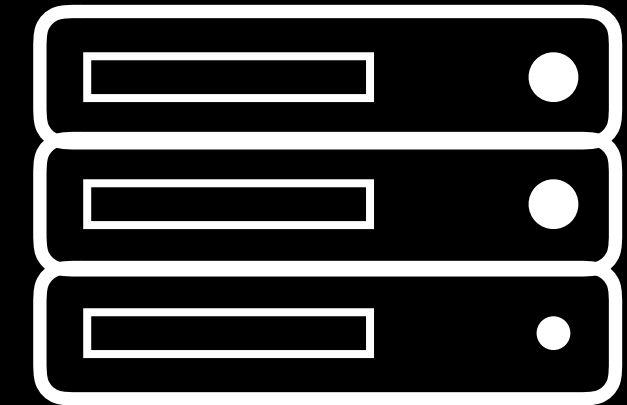
```
a = 1  
b = 2  
c = 3  
d = 4  
e = 5
```

Add line



git status

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



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

```
a = 1  
b = 2  
c = 3  
d = 4  
e = 5
```

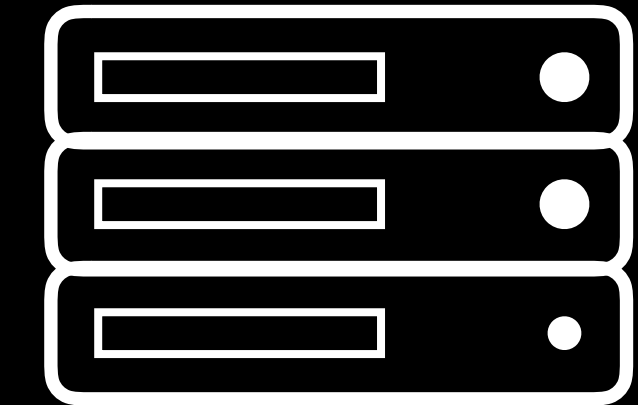
Add line



```
git status
```

git status

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



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

```
a = 1  
b = 2  
c = 3  
d = 4  
e = 5
```

Add line

git status

On branch master

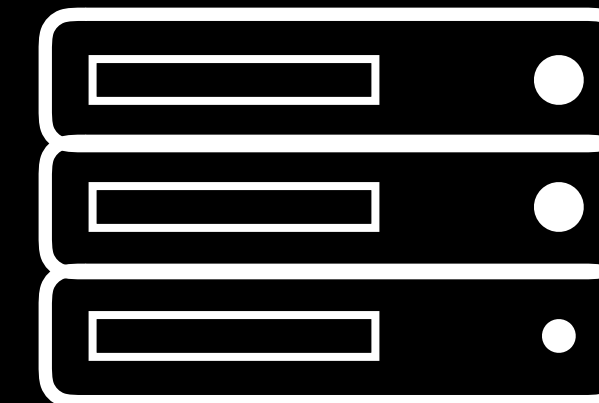
Your branch is ahead of 'origin/master' by 1 commit.

(use "git push" to publish your local commits)

```
git push
```

git push

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



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

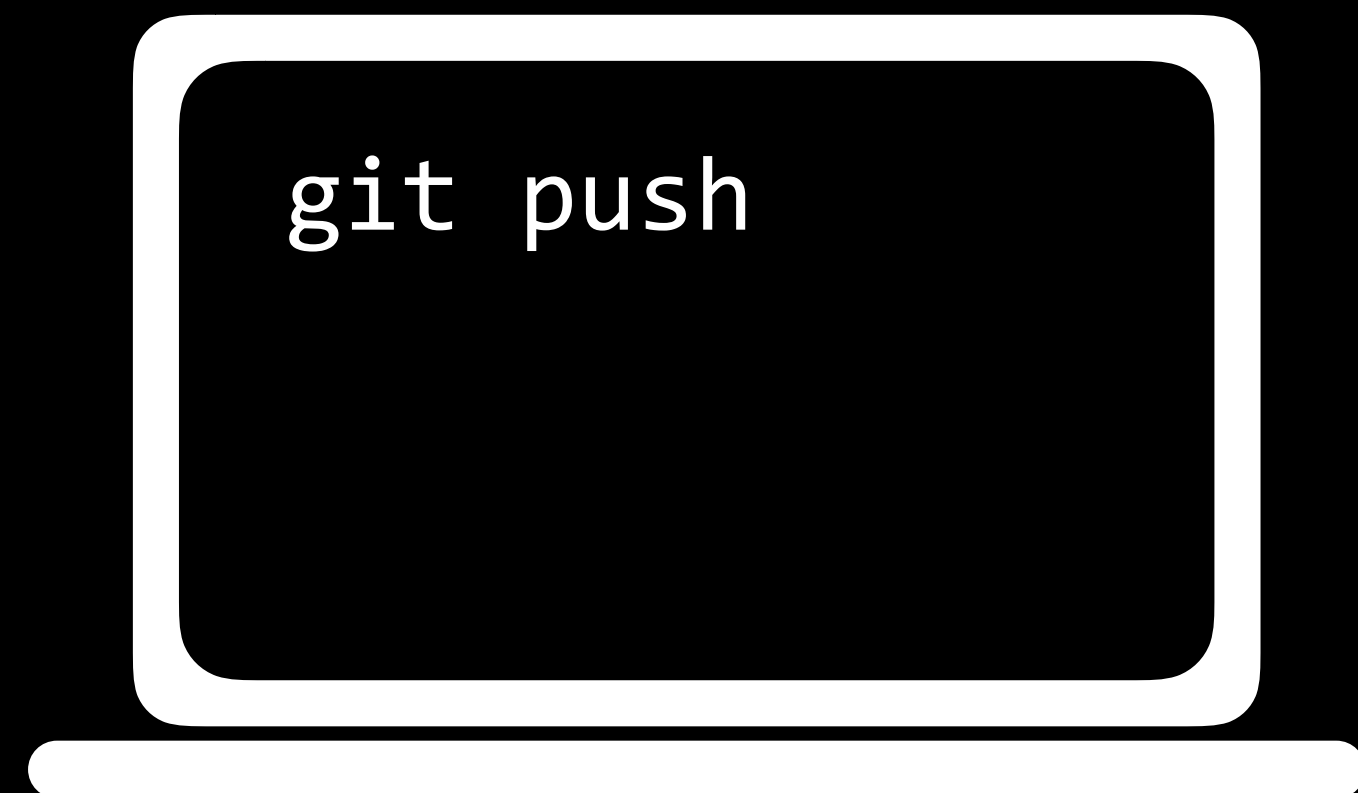
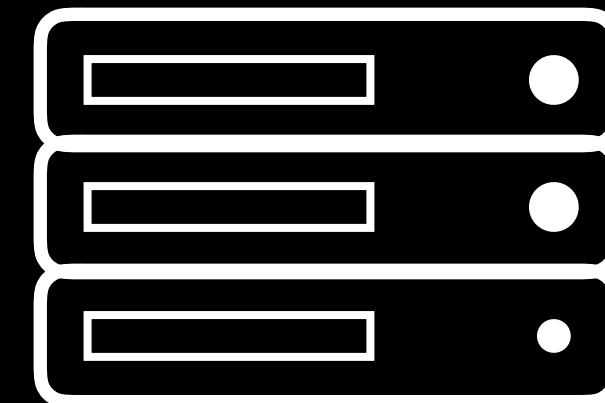


```
a = 1  
b = 2  
c = 3  
d = 4  
e = 5
```

Add line

git push

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



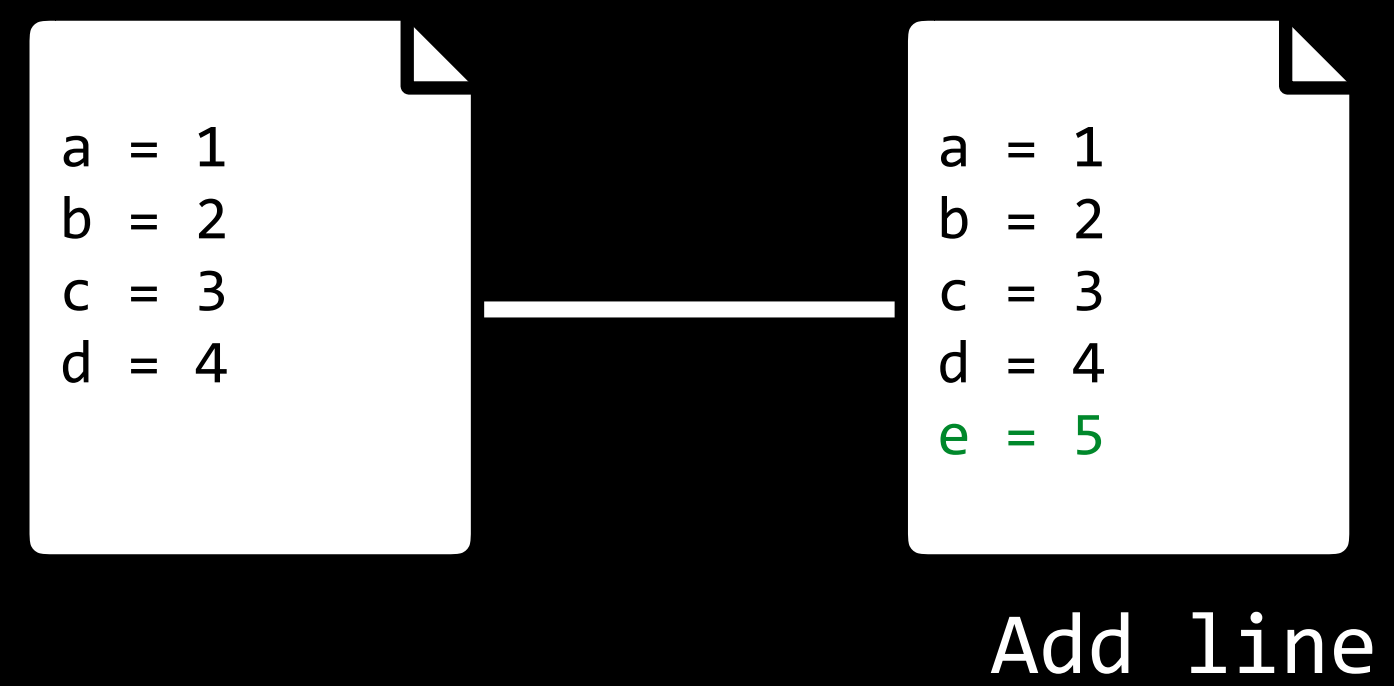
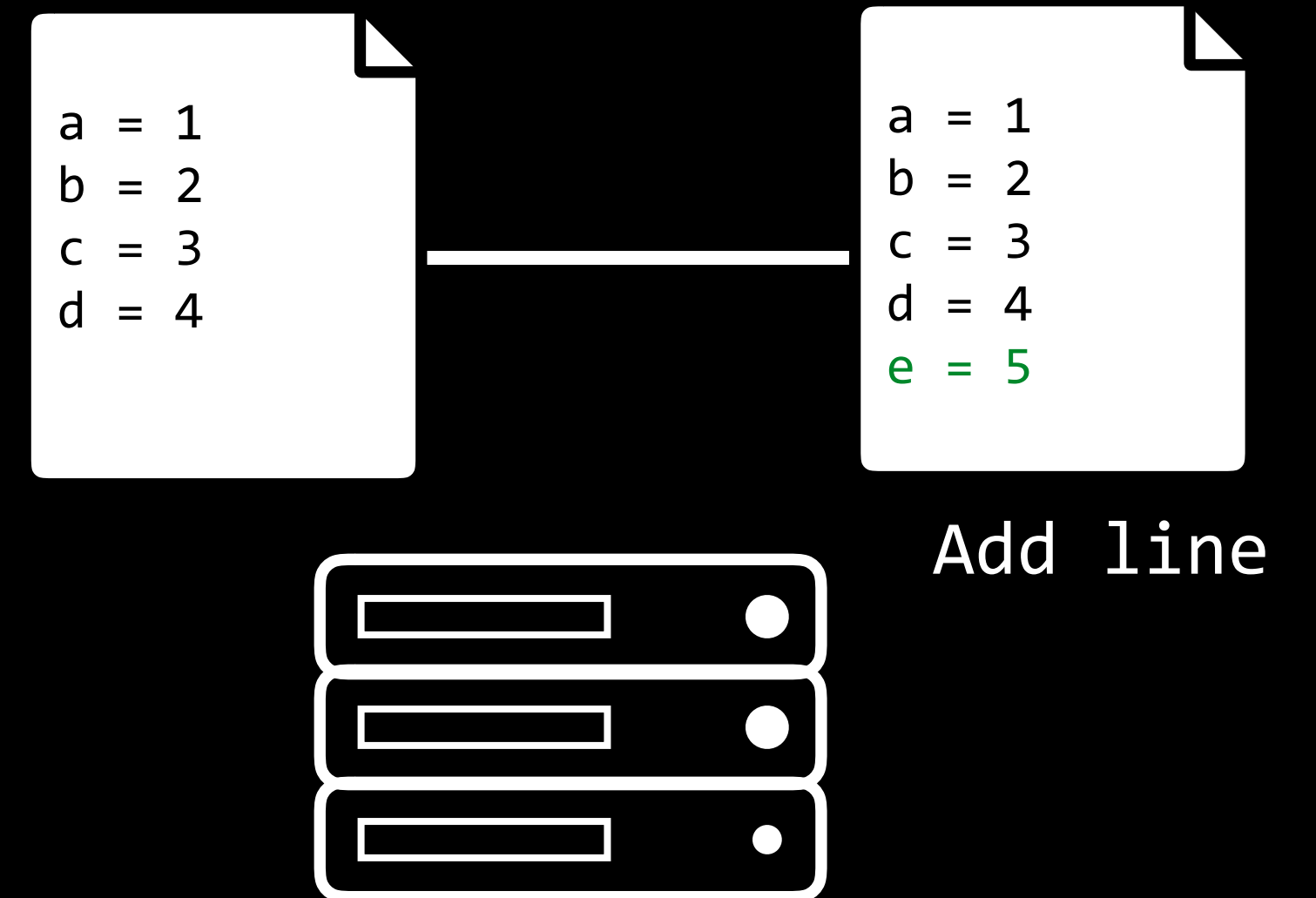
```
a = 1  
b = 2  
c = 3  
d = 4
```



```
a = 1  
b = 2  
c = 3  
d = 4  
e = 5
```

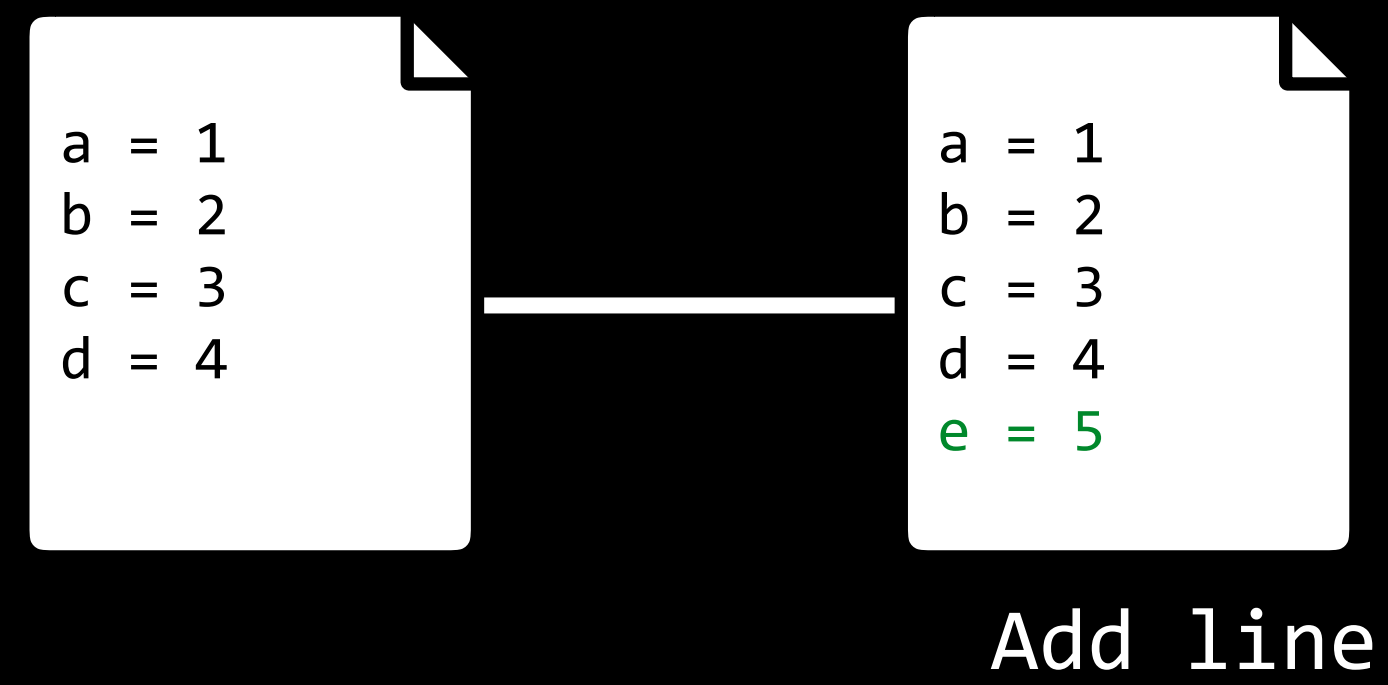
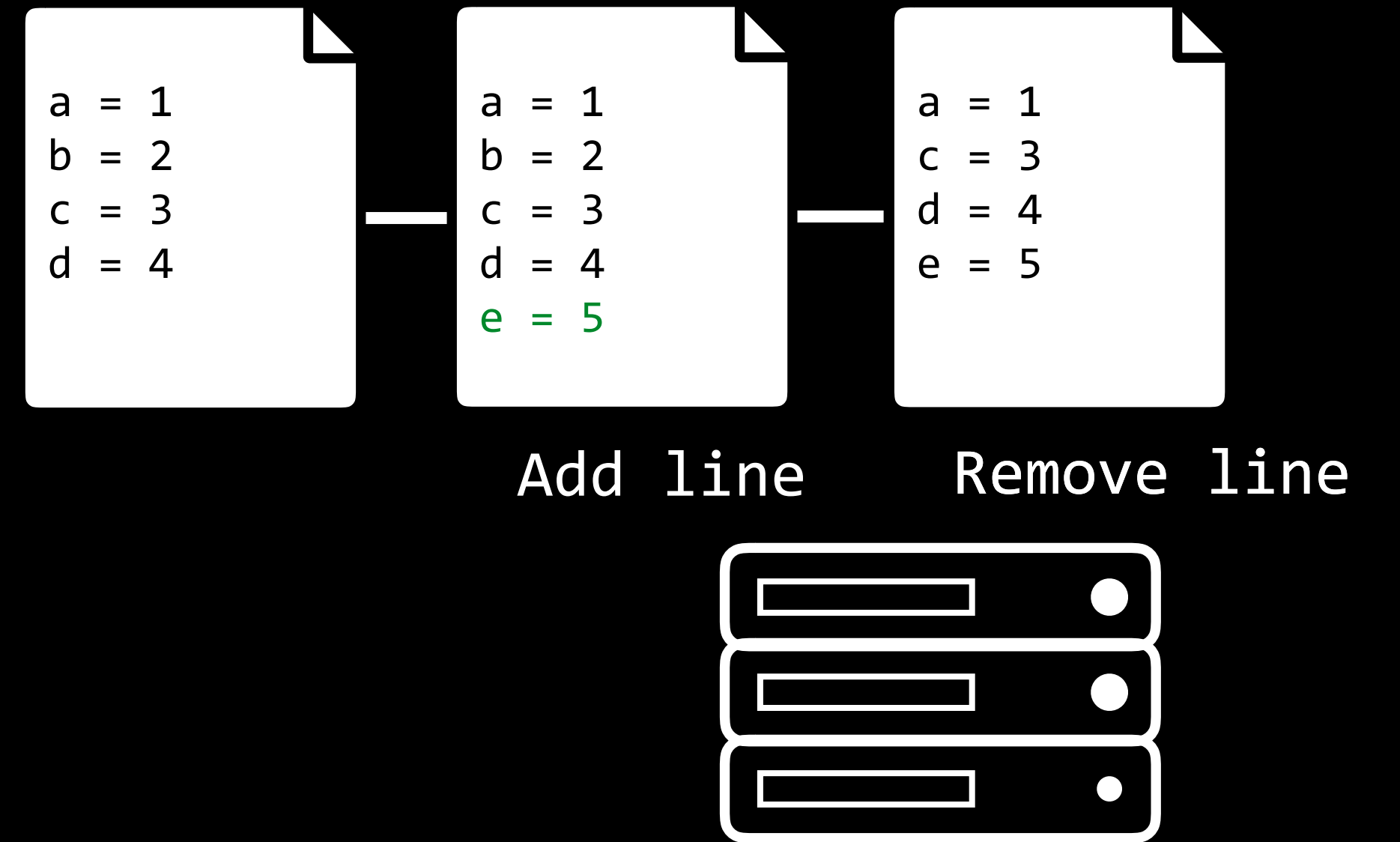
Add line

git push

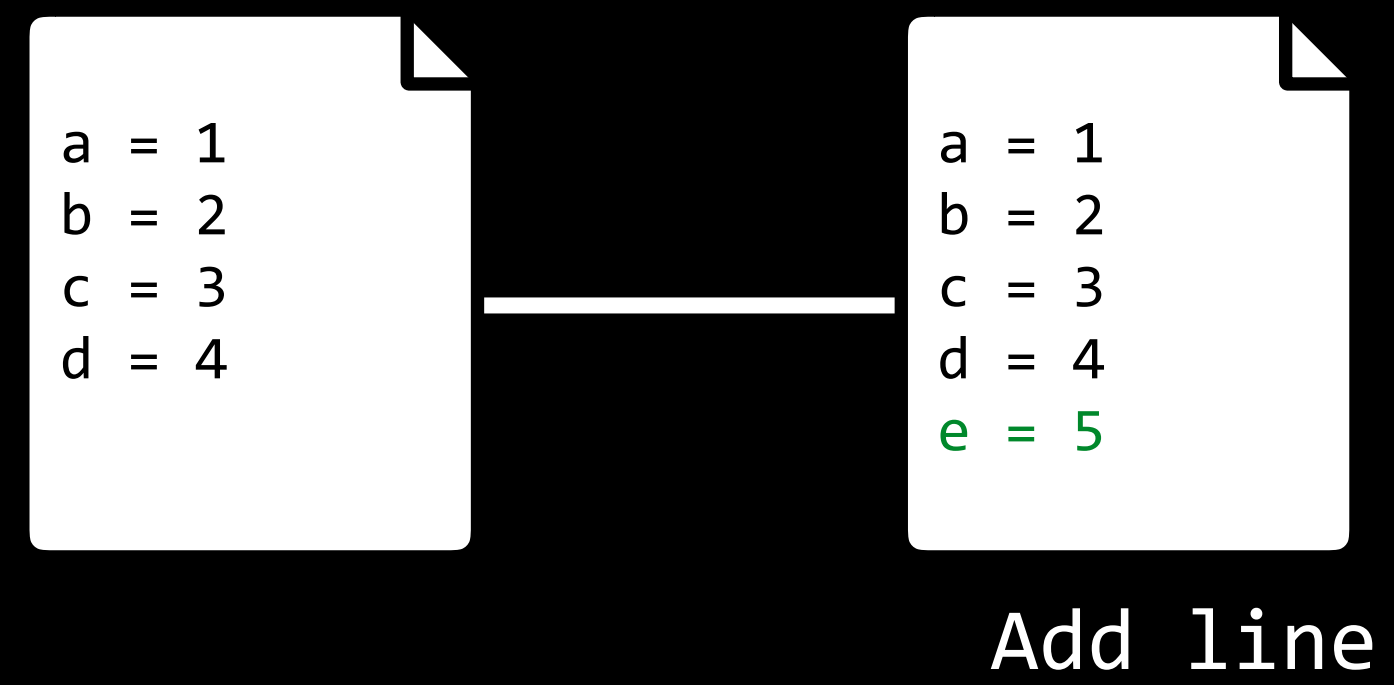
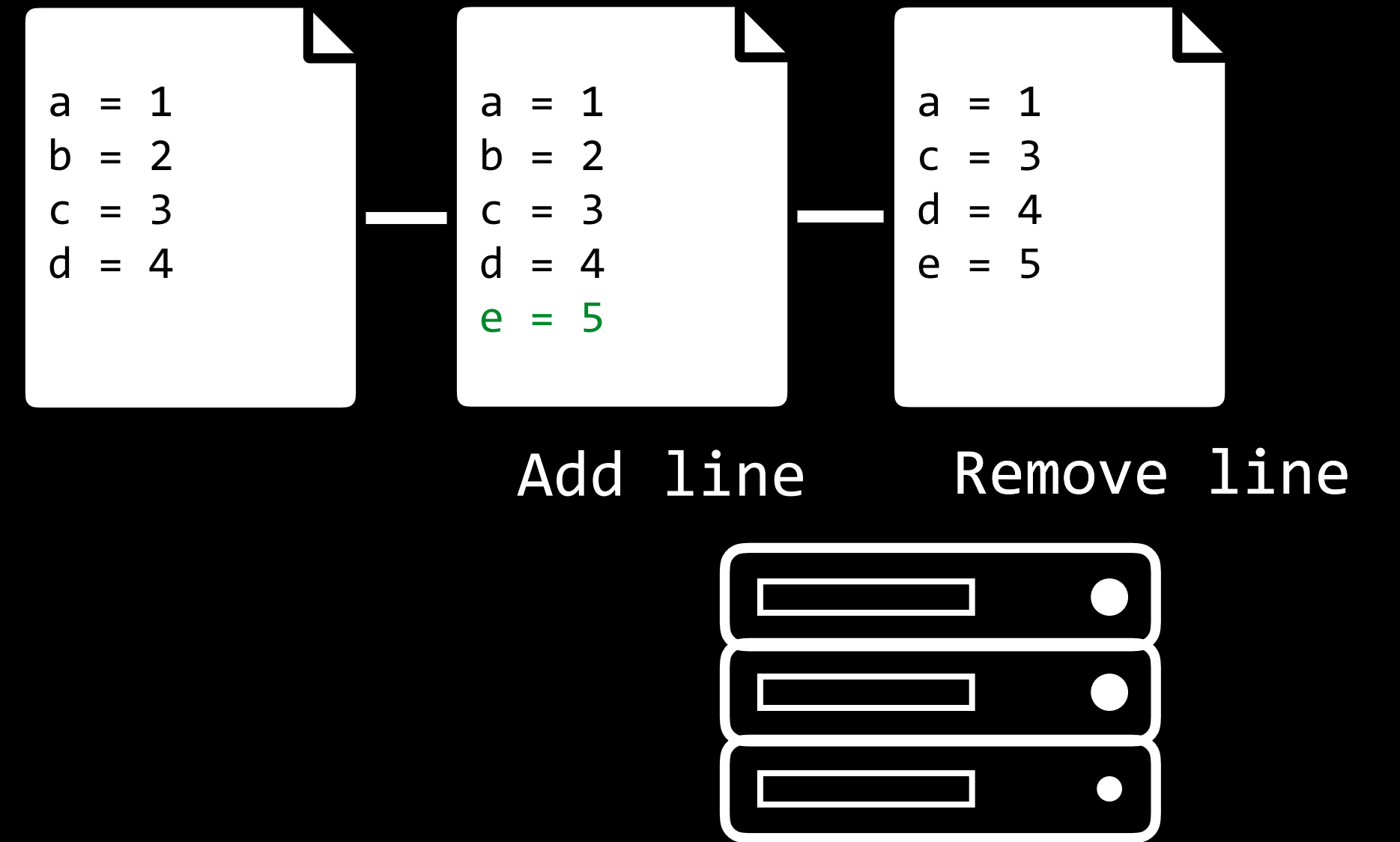


```
git pull
```

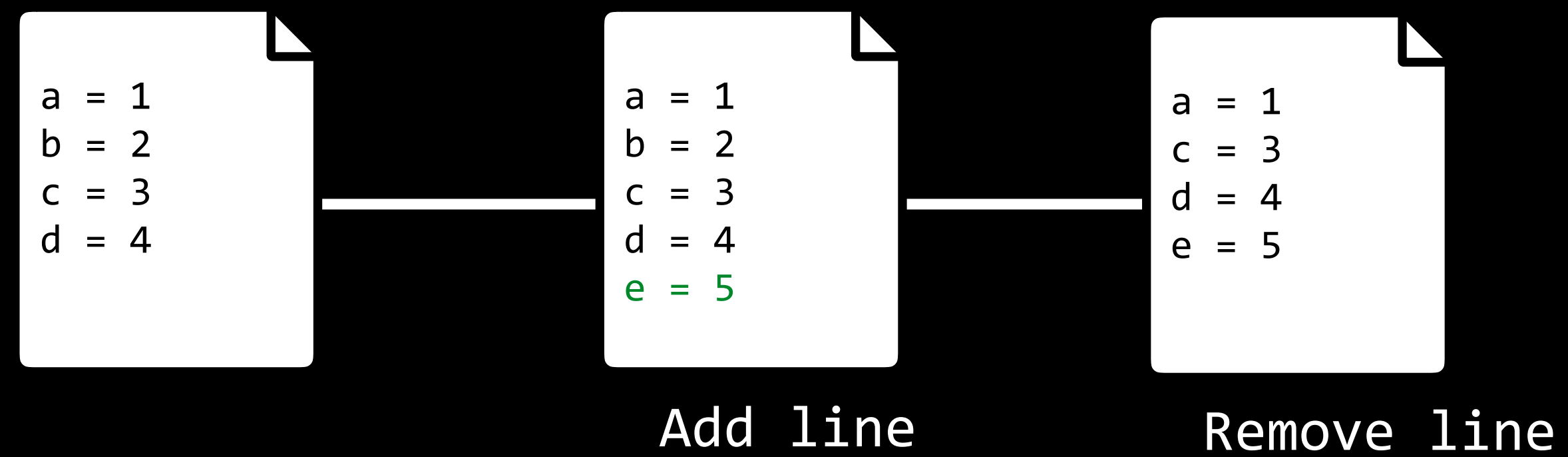
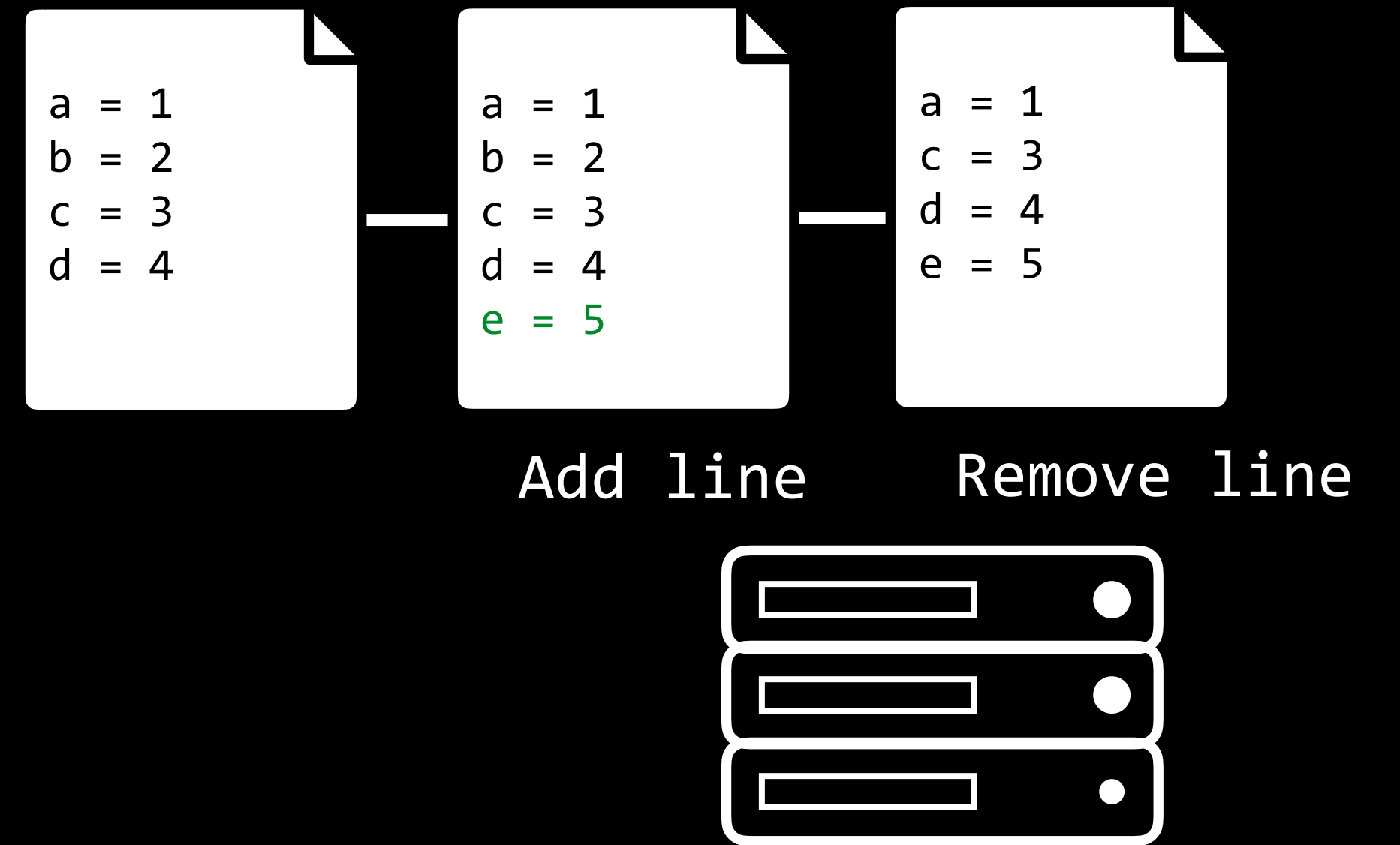

git pull



git pull

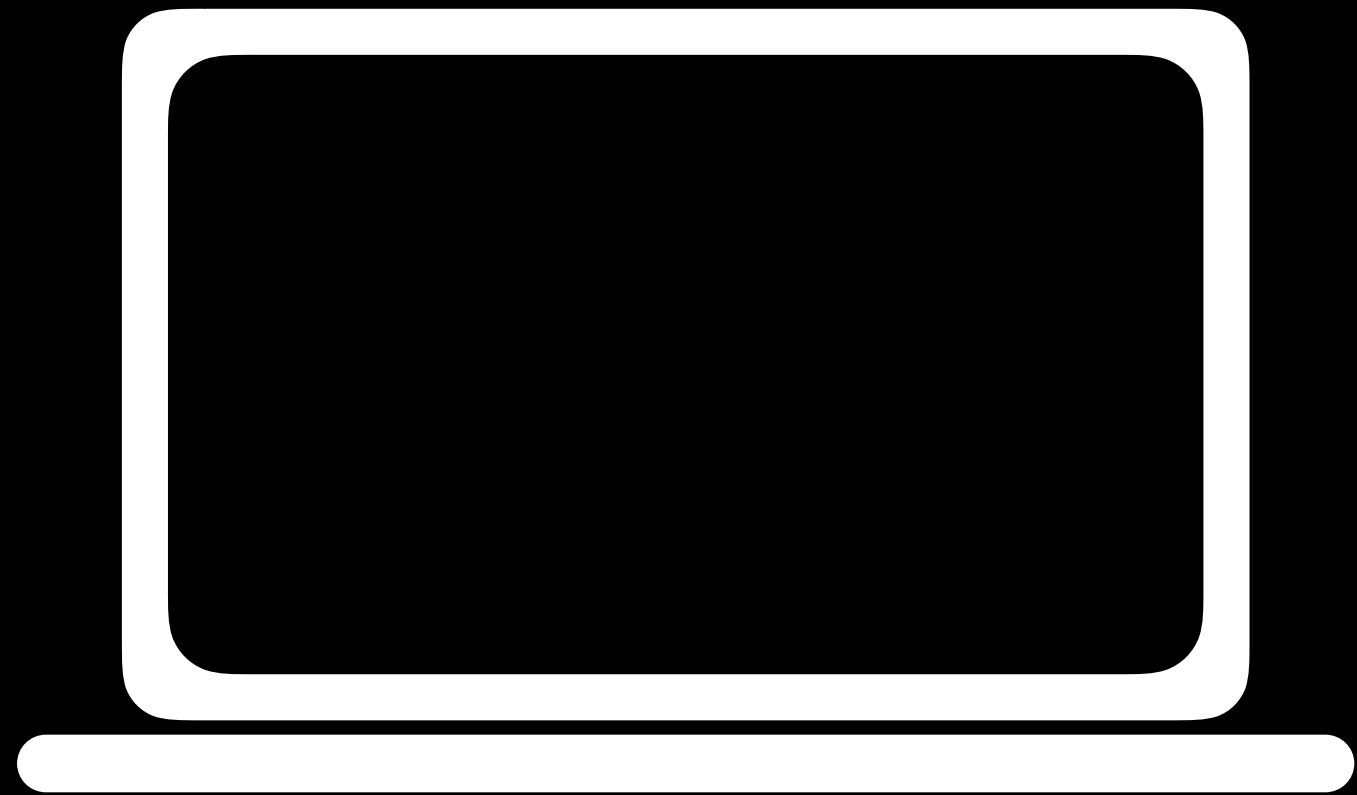


git pull



Merge Conflicts

Merge Conflicts



Merge Conflicts



```
git pull
```

Merge Conflicts



```
git pull
```

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

Merge Conflicts



```
git pull
```

```
a = 1
<<<<< HEAD
b = 2
=====
b = 0
>>>>> 57656c636f6d6520746f20576562
c = 3
d = 4
e = 5
```


Merge Conflicts




```
git pull
```

your
changes

remote
changes

```
a = 1
<<<<< HEAD
{ b = 2
  =====
  { b = 0
    >>>>> 57656c636f6d6520746f20576562
    c = 3
    d = 4
    e = 5
```

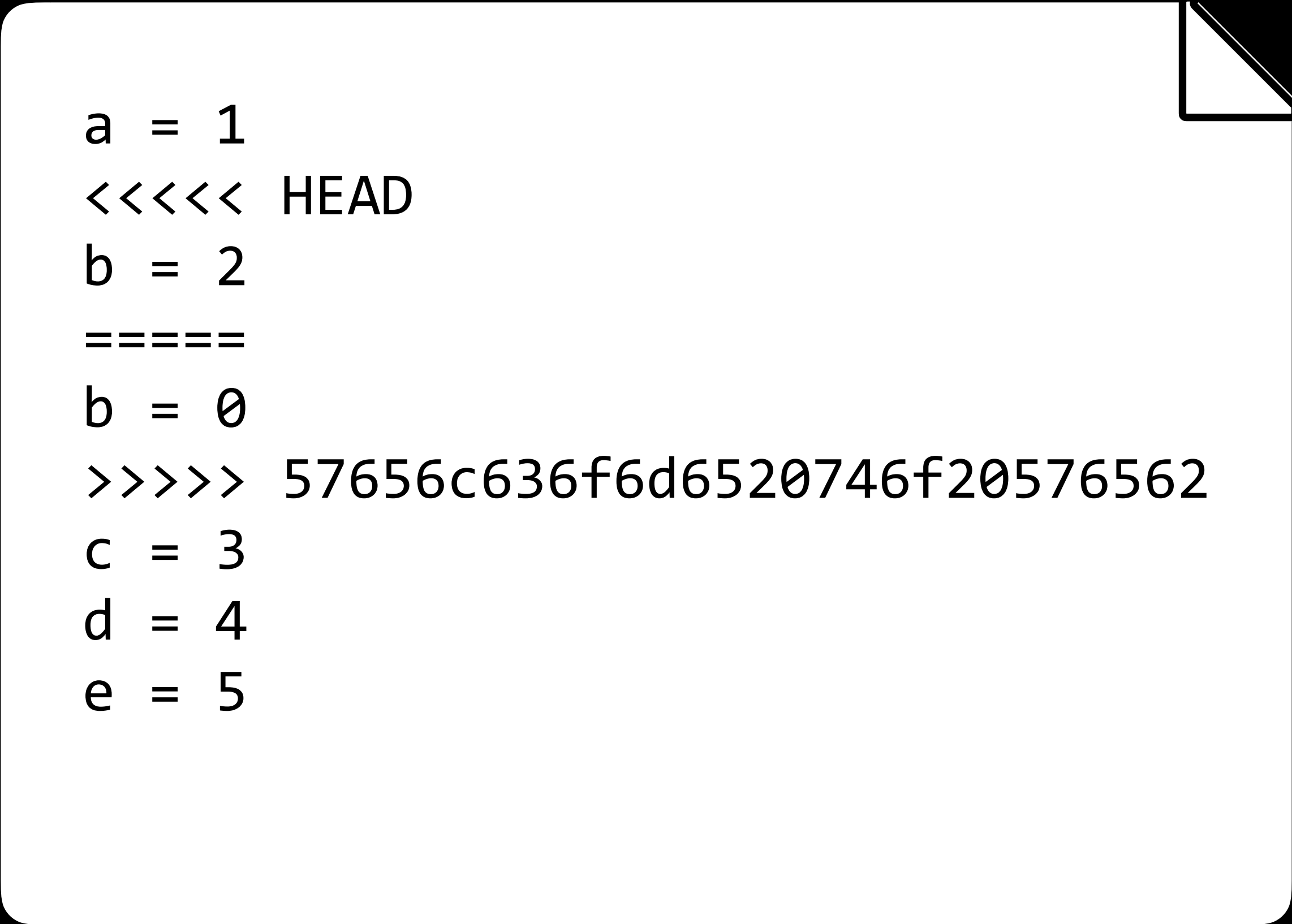
conflicting commit



Merge Conflicts



```
git pull
```



```
a = 1
<<<<< HEAD
b = 2
=====
b = 0
>>>>> 57656c636f6d6520746f20576562
c = 3
d = 4
e = 5
```

Merge Conflicts

A white-outlined icon of a terminal window with a thick border, containing the text 'git pull'.

```
git pull
```

```
a = 1
```

```
b = 2
```

```
c = 3
```

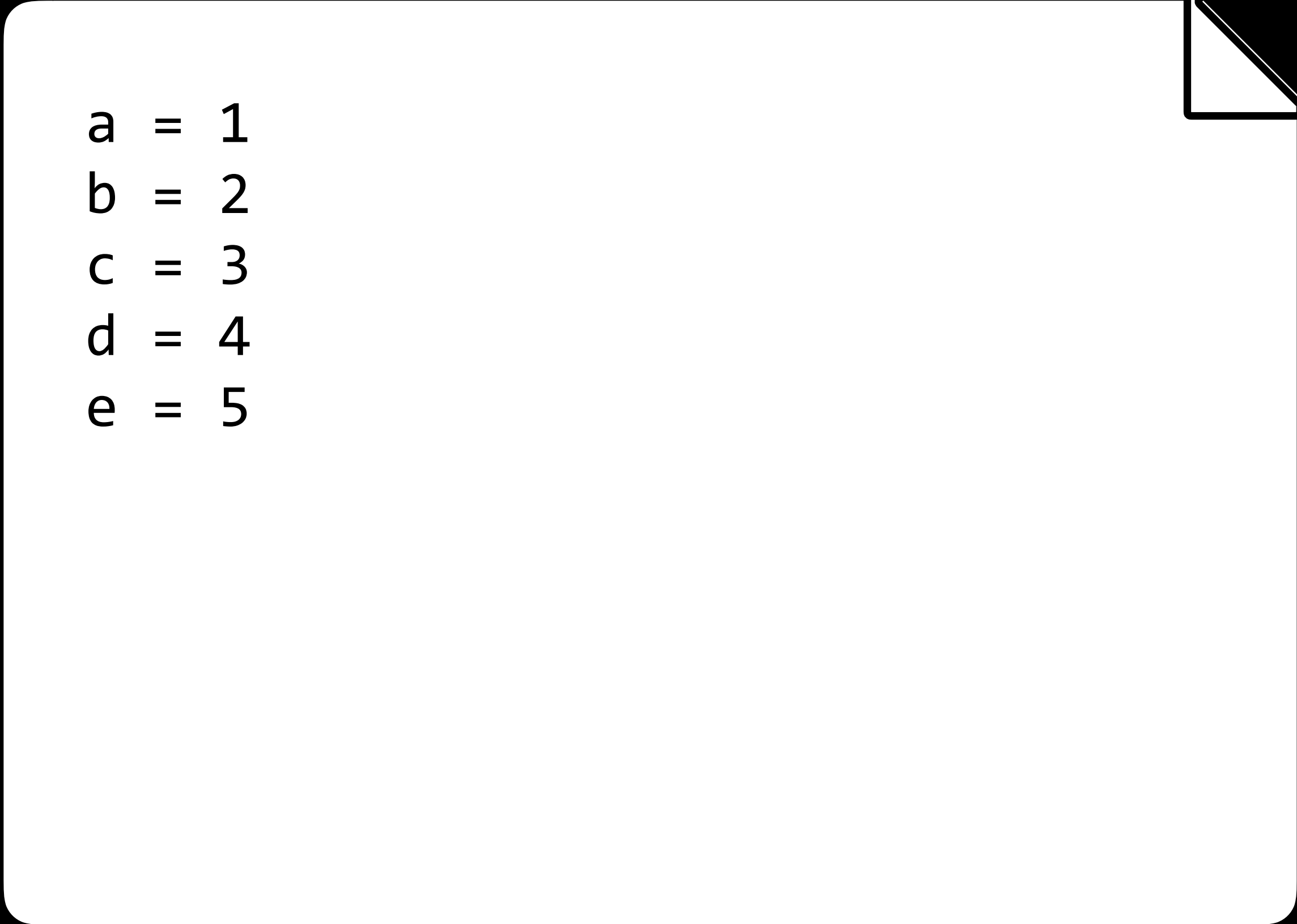
```
d = 4
```

```
e = 5
```

Merge Conflicts



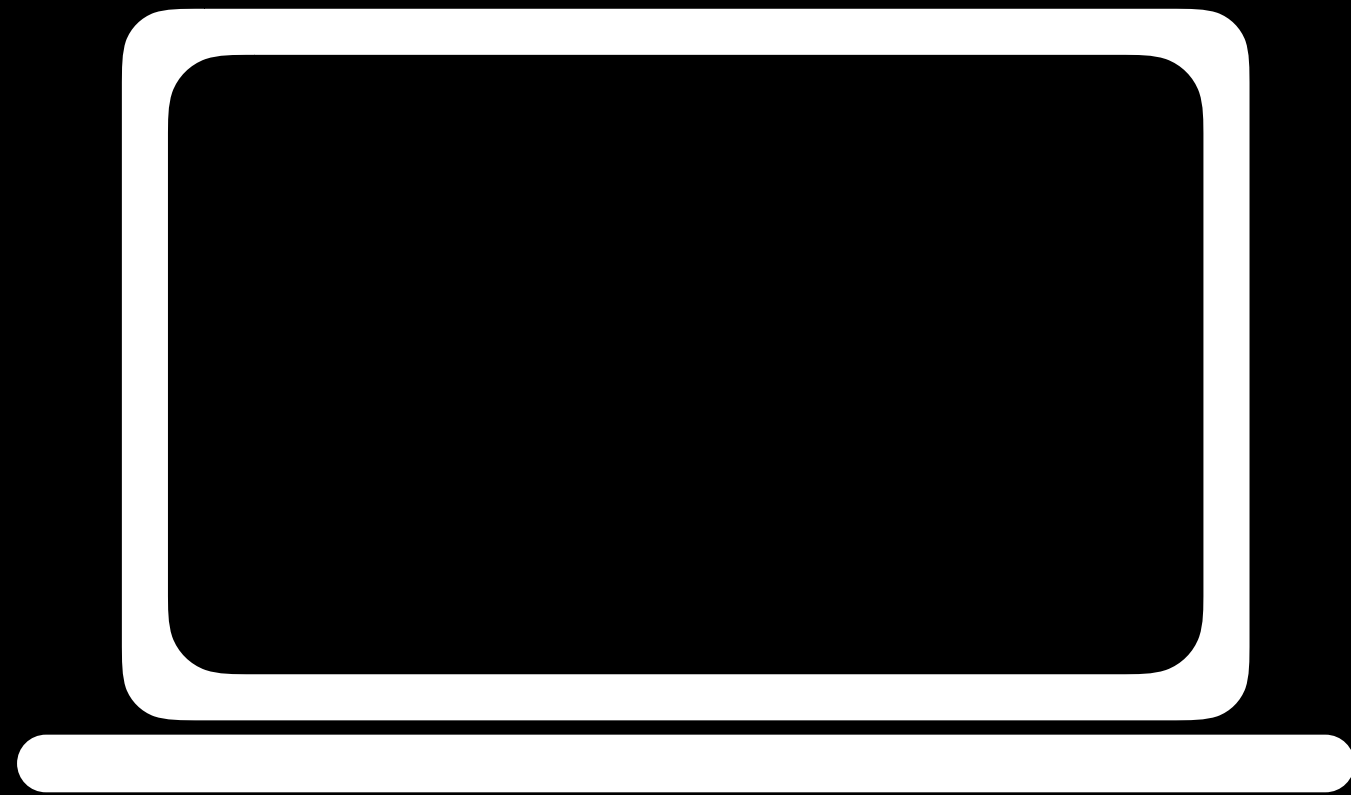
```
git pull
```



```
a = 1  
b = 2  
c = 3  
d = 4  
e = 5
```

git log

git log



```
git log
```



```
git log
```

git log

```
commit 436f6d6d6974204d73672048657265
Author: Brian Yu <brian@cs.harvard.edu>
Date:   Tue Jan 14 14:06:28 2020 -0400
```

Remove a line

```
commit 57656c636f6d6520746f20576562
Author: Brian Yu <brian@cs.harvard.edu>
Date:   Tue Jan 14 14:05:28 2020 -0400
```

Add a line

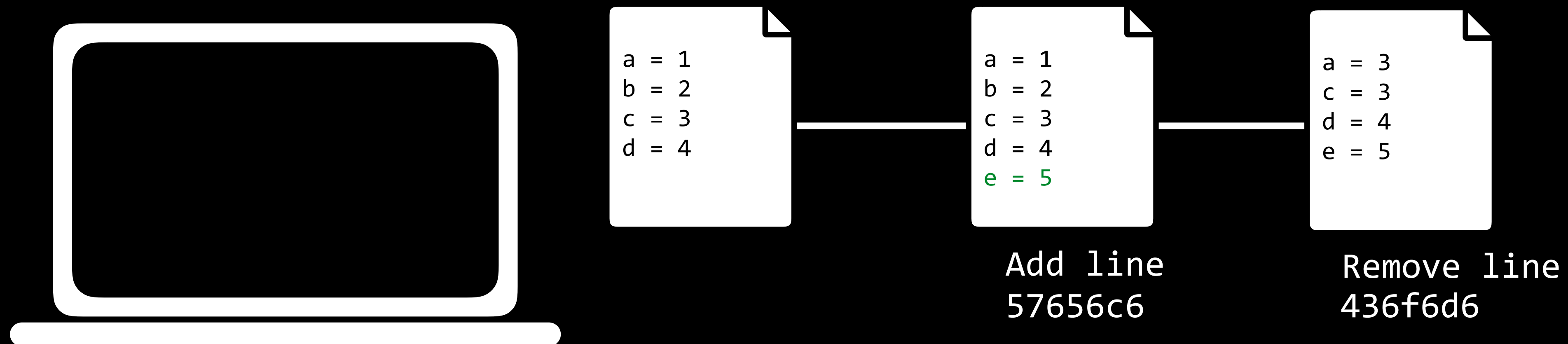


git log


```
git reset
```

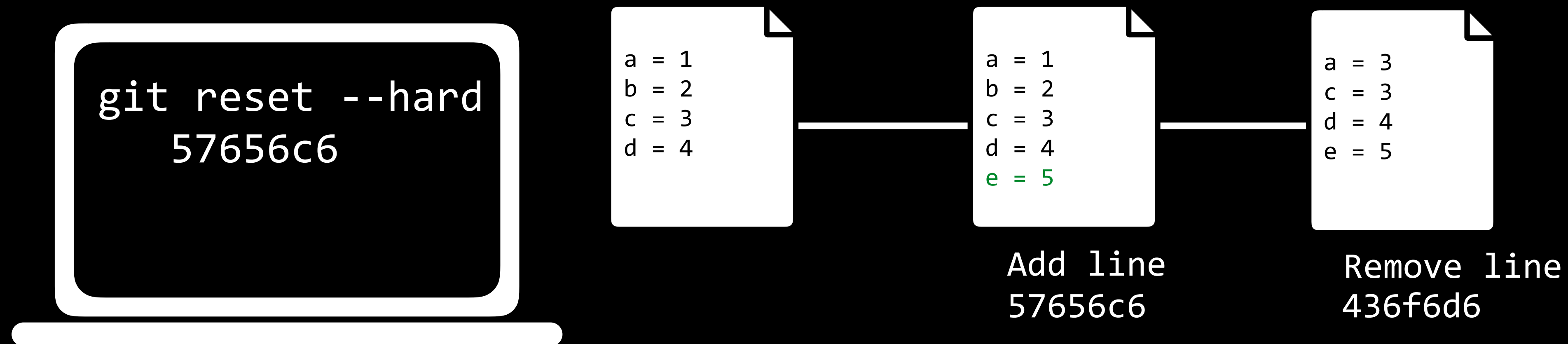
git reset

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



git reset

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

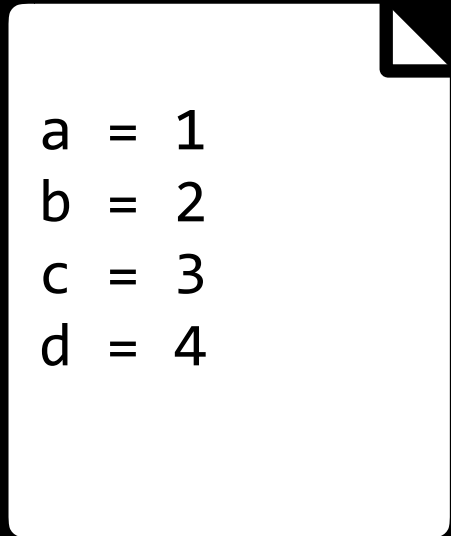


git reset

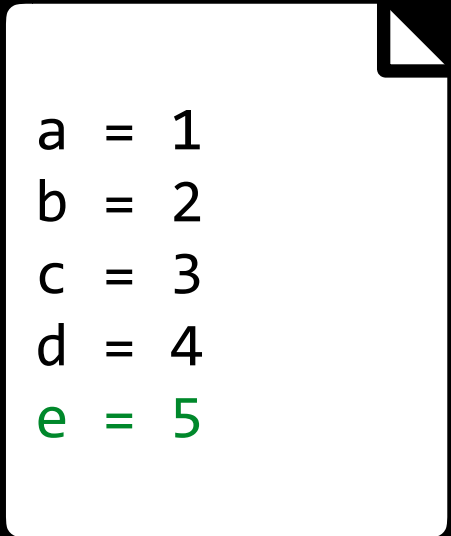
- `git reset --hard <commit>`
- `git reset --hard origin/master`



```
git reset --hard  
57656c6
```



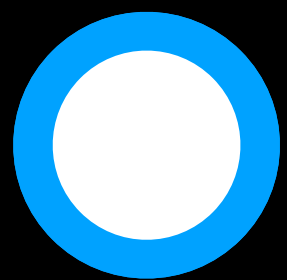
```
a = 1  
b = 2  
c = 3  
d = 4
```



```
a = 1  
b = 2  
c = 3  
d = 4  
e = 5
```

Add line
57656c6

Making Changes

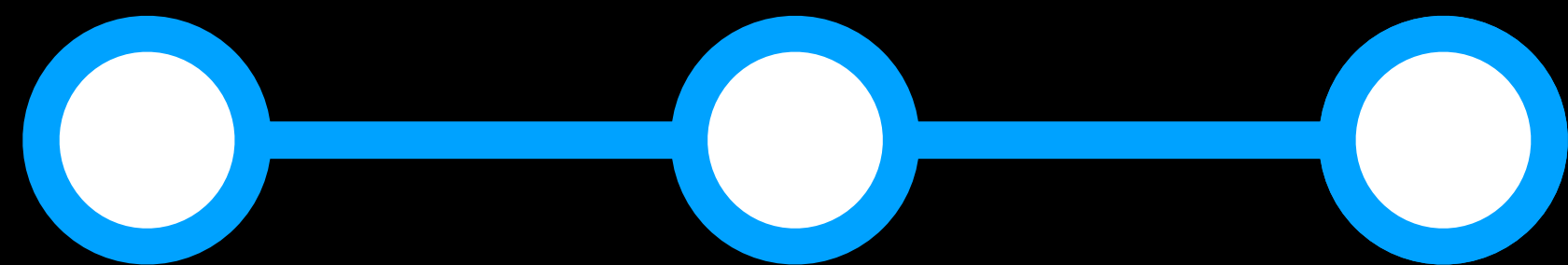


first
commit



first
commit

changes



first
commit

changes

more
changes

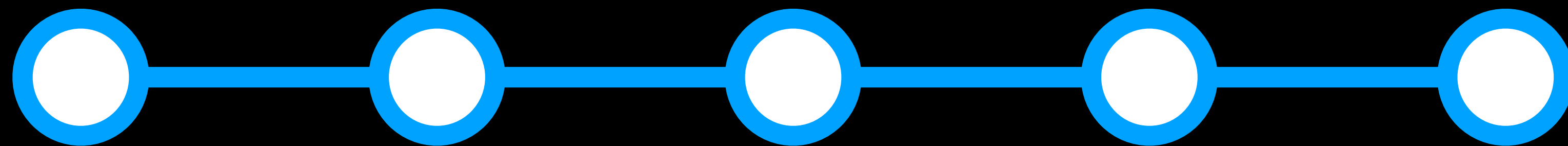


first
commit

changes

more
changes

start new
feature



first
commit

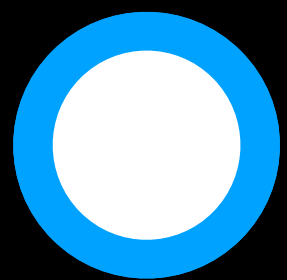
changes

more
changes

start new
feature

keep working
on new feature

Branching

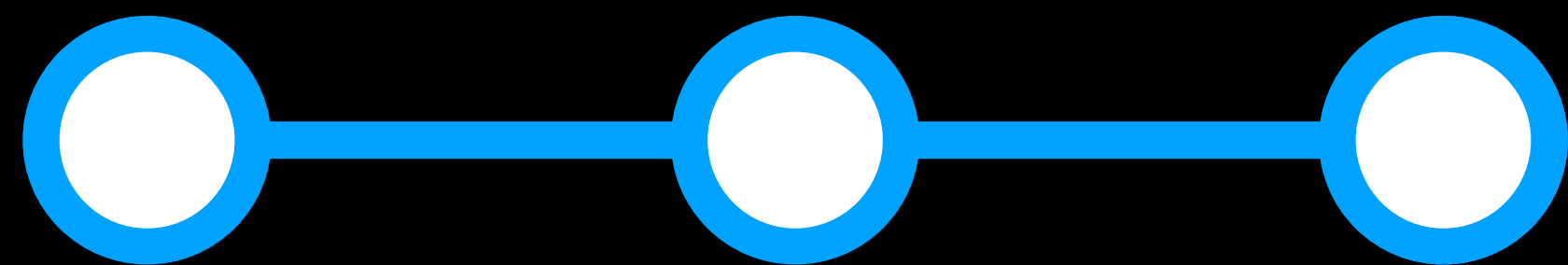


first
commit



first
commit

changes



first
commit

changes

more
changes

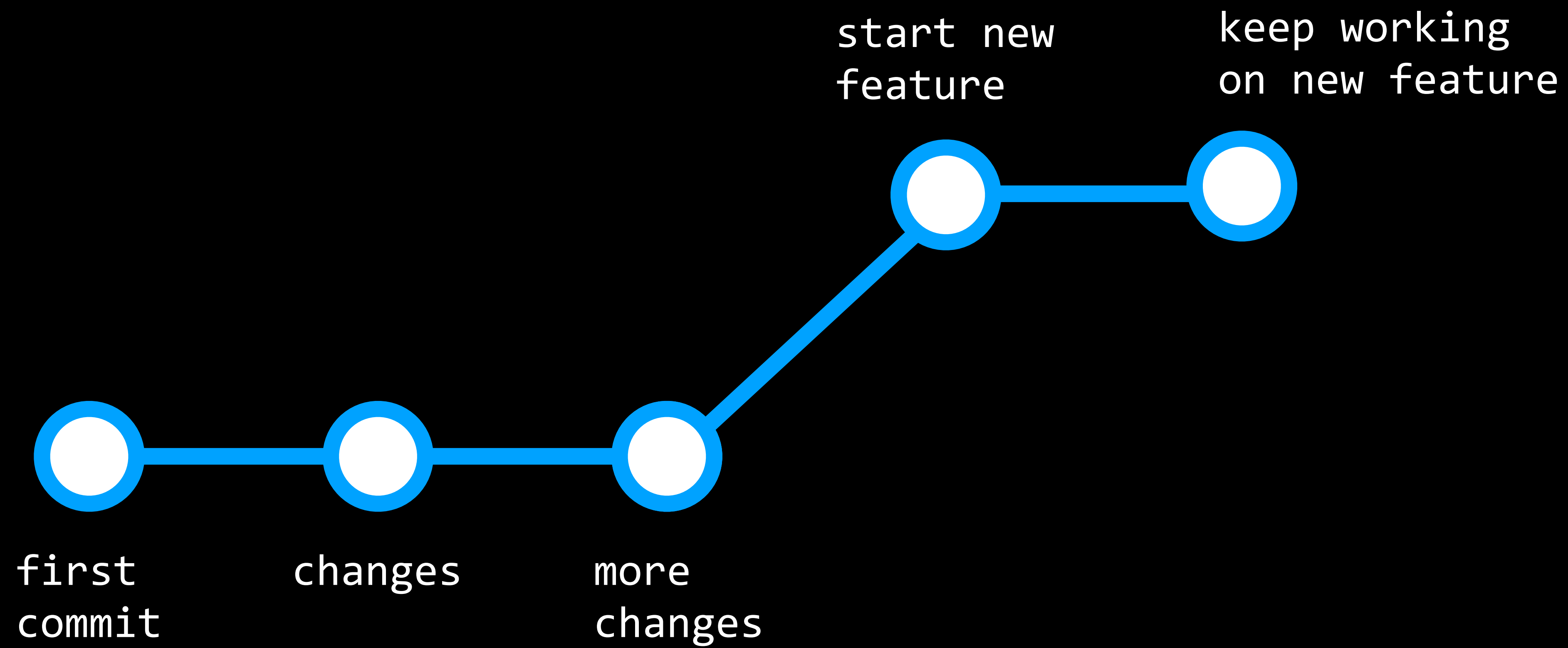
start new
feature

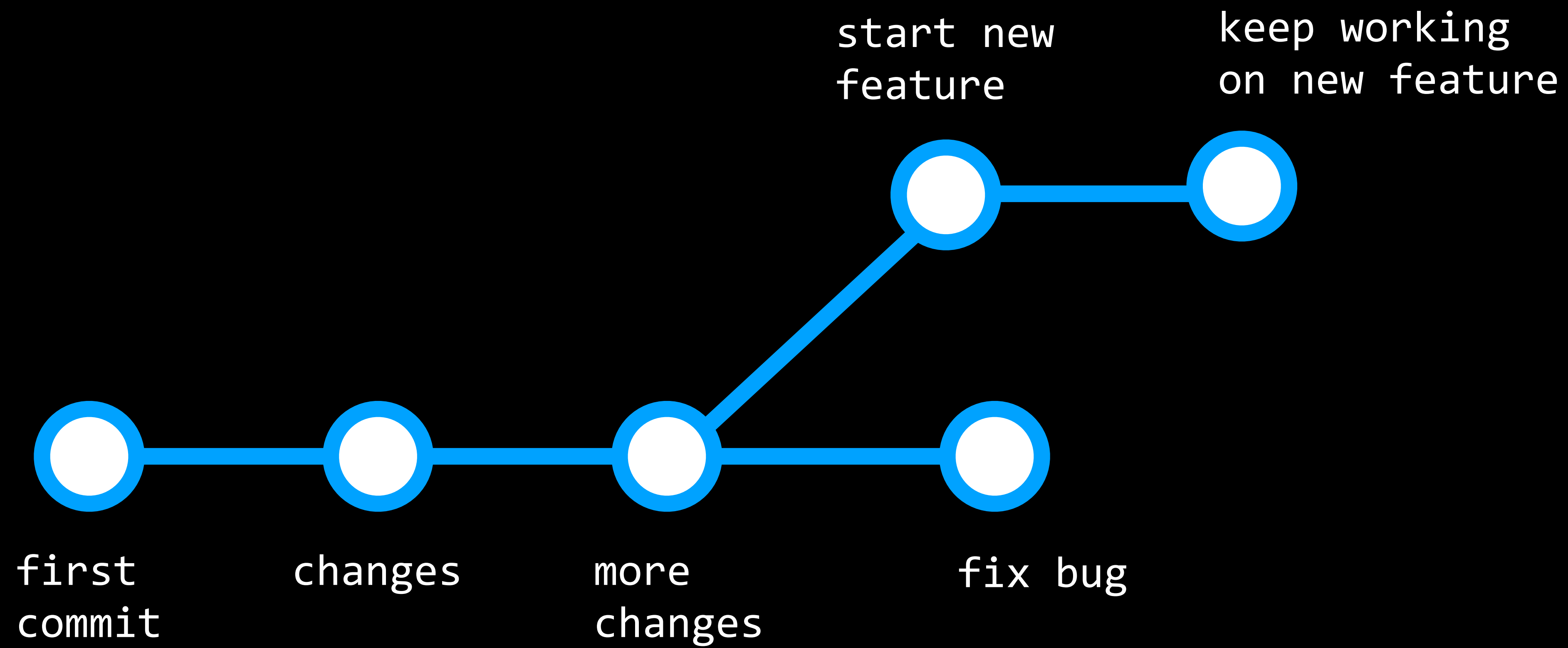


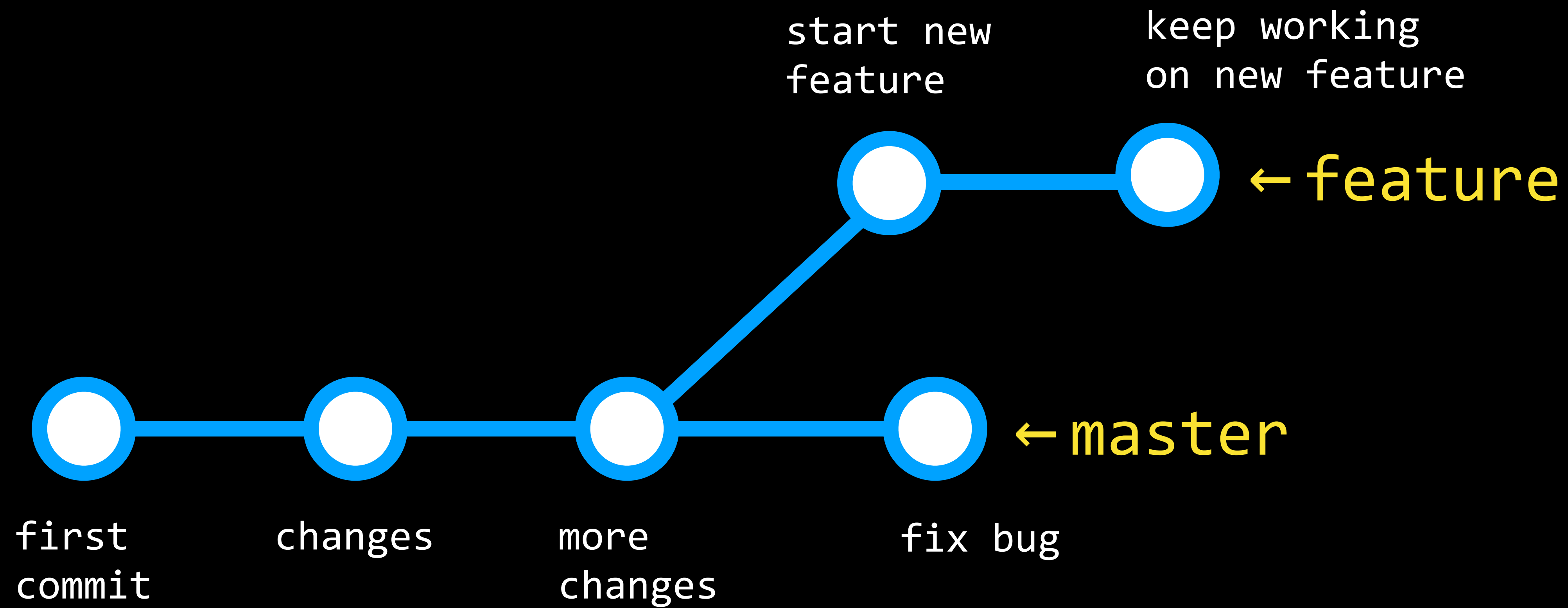
first
commit

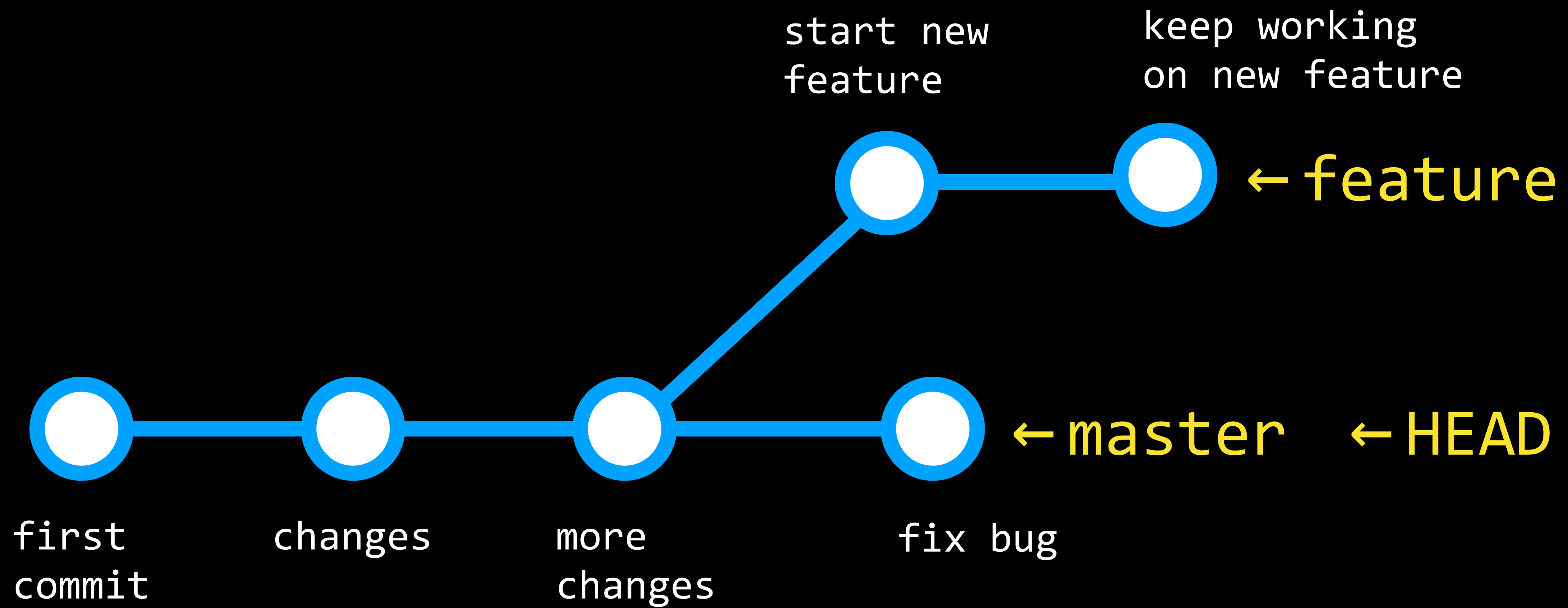
changes

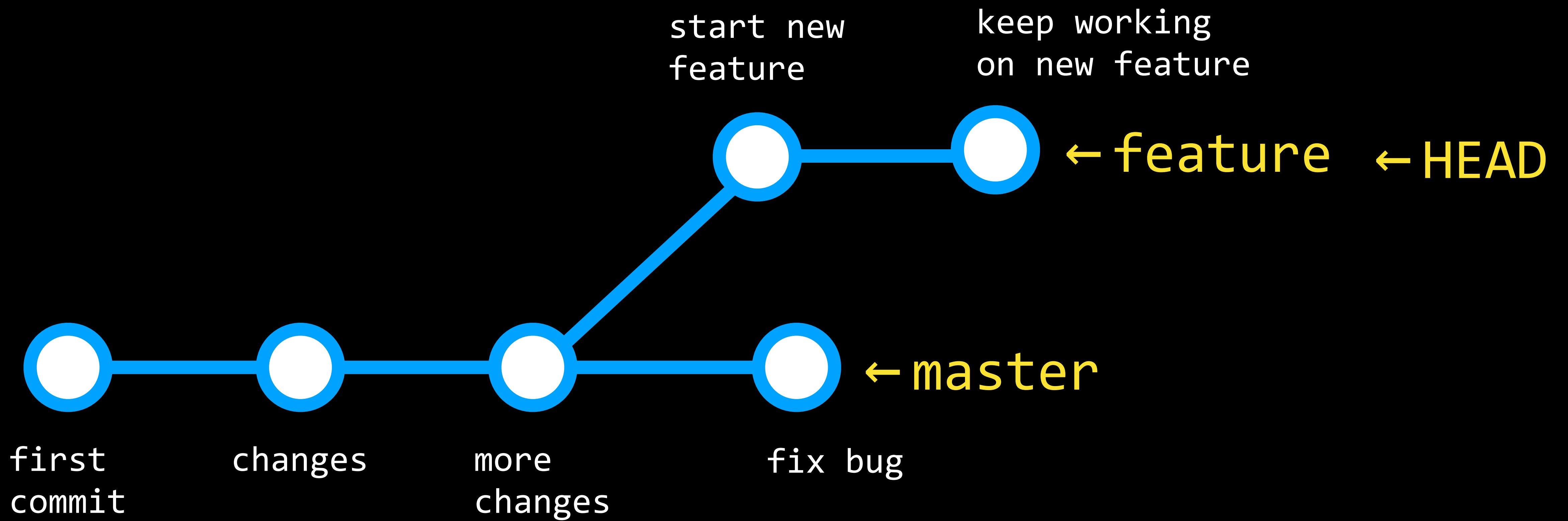
more
changes

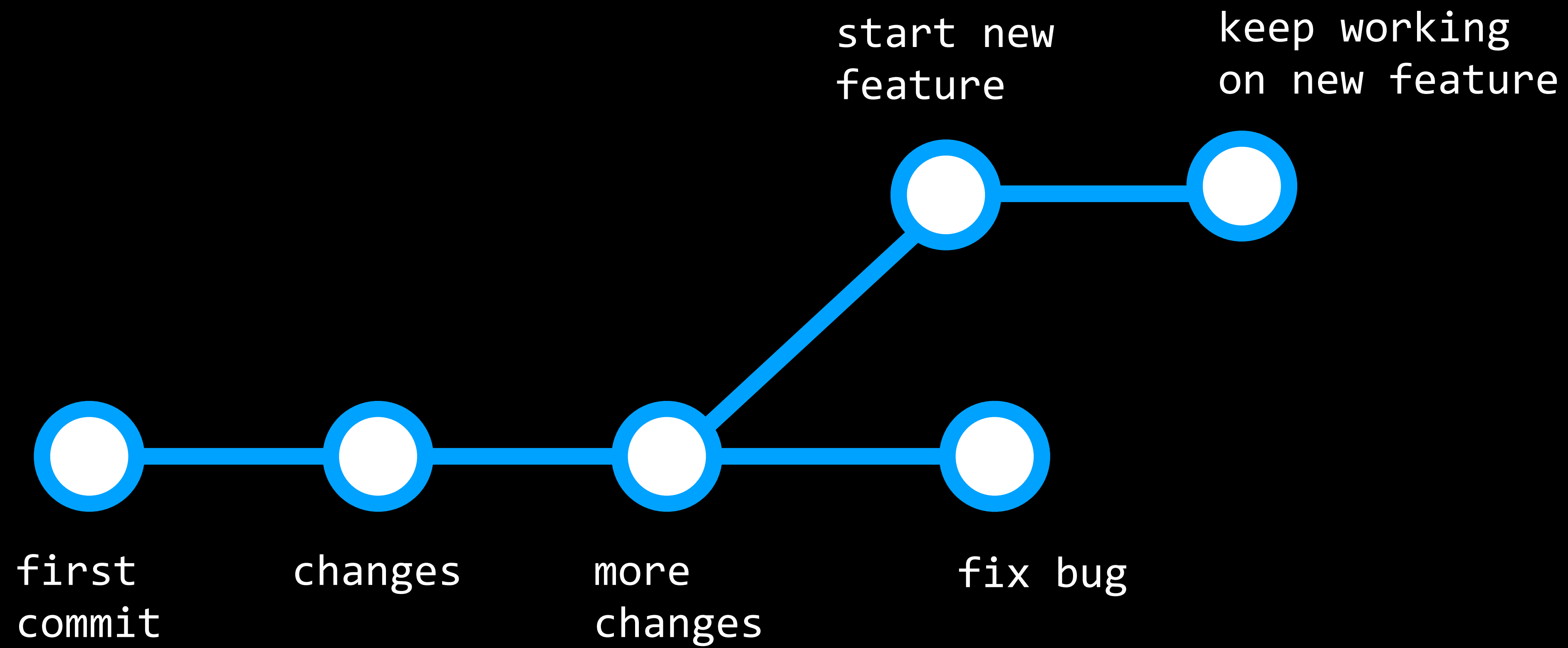


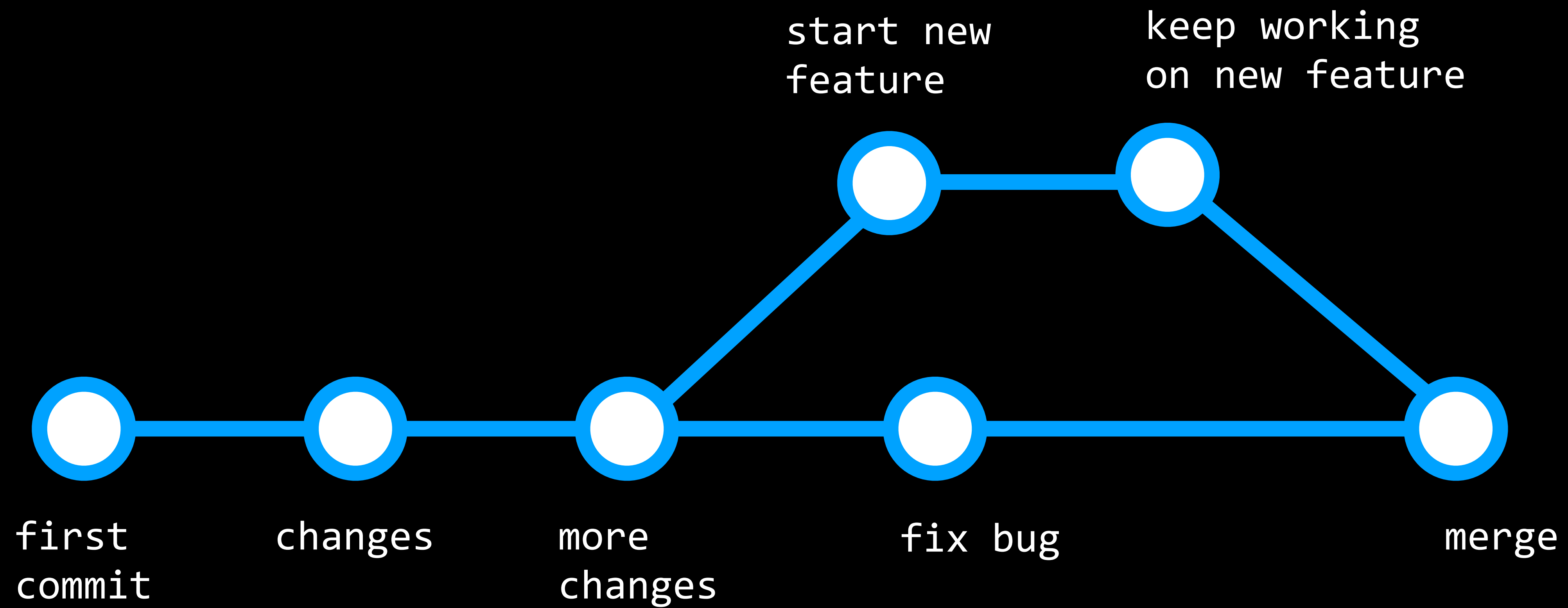












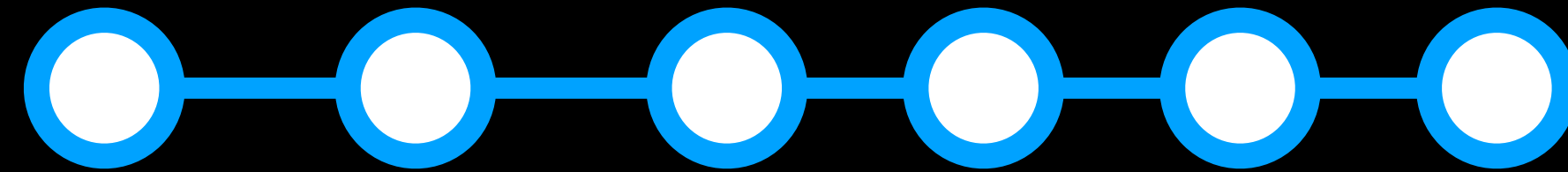
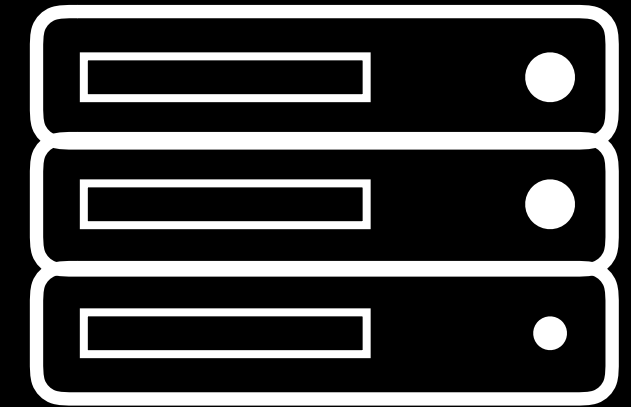
Branching

- `git branch`
- `git checkout`
- `git merge`

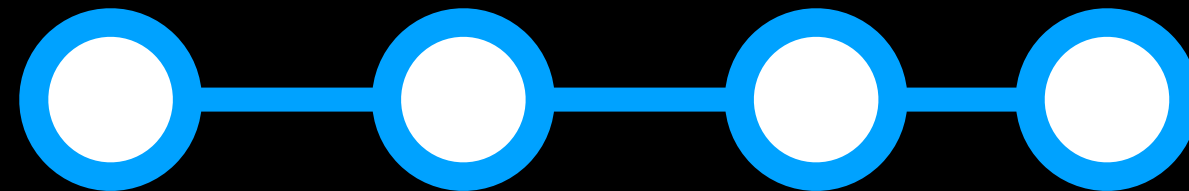
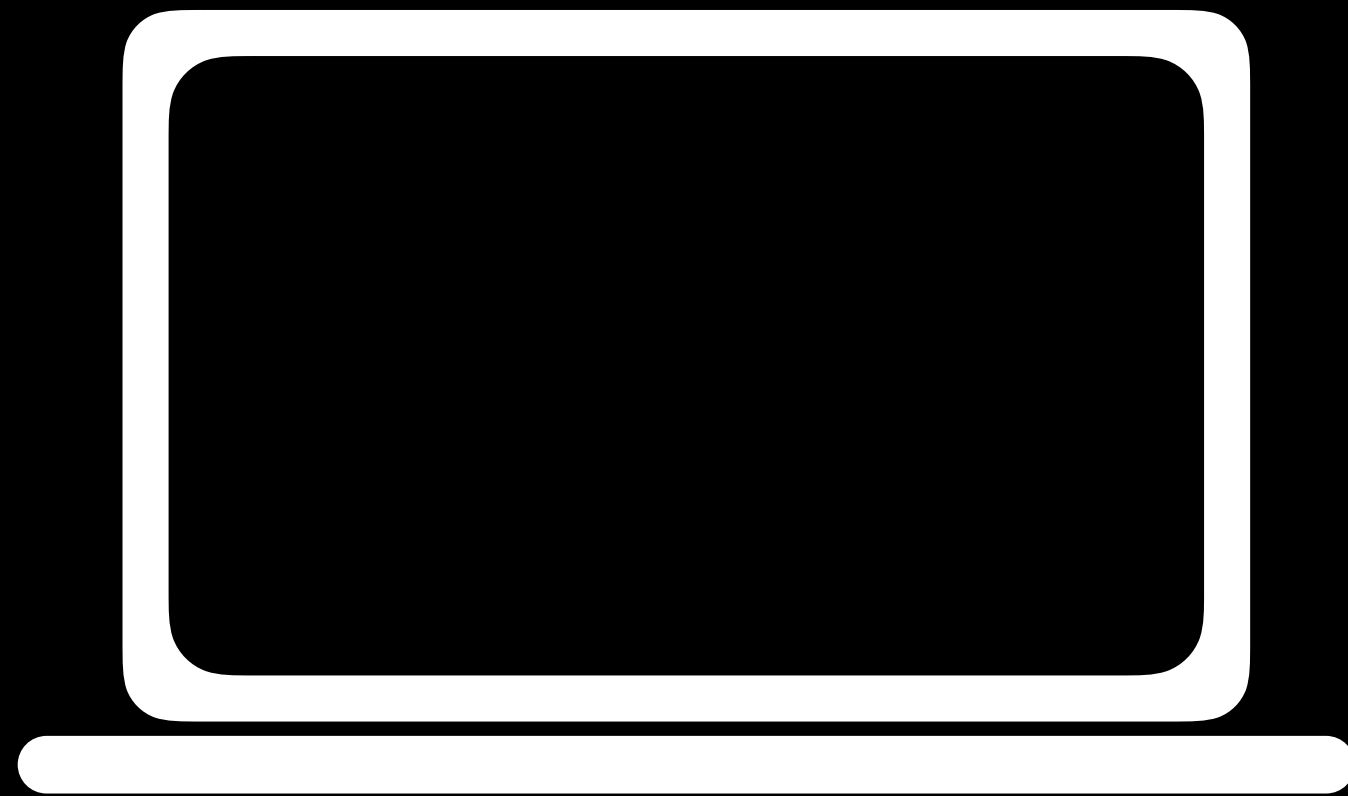
Remotes

Remotes

Remotes

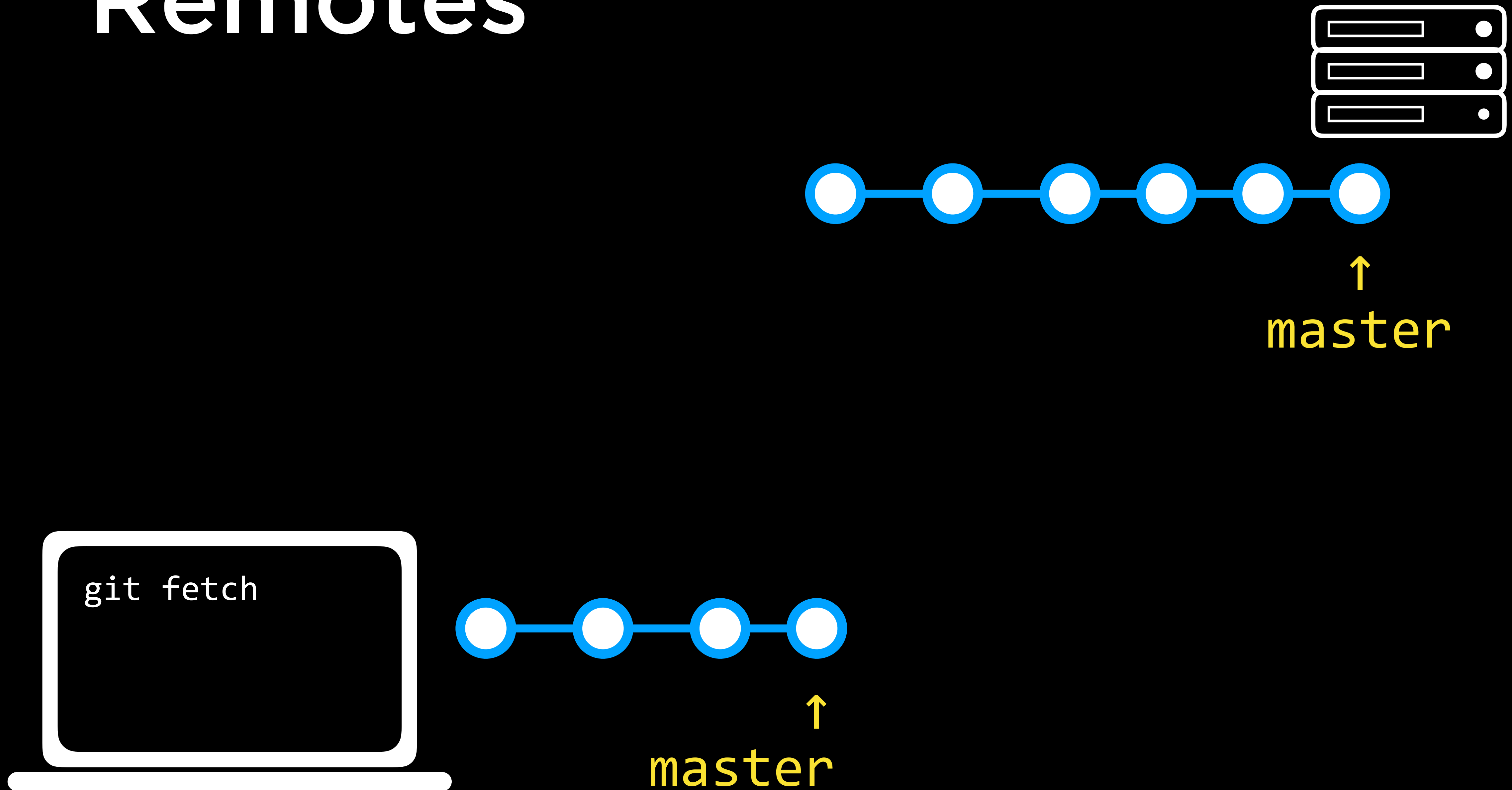


master

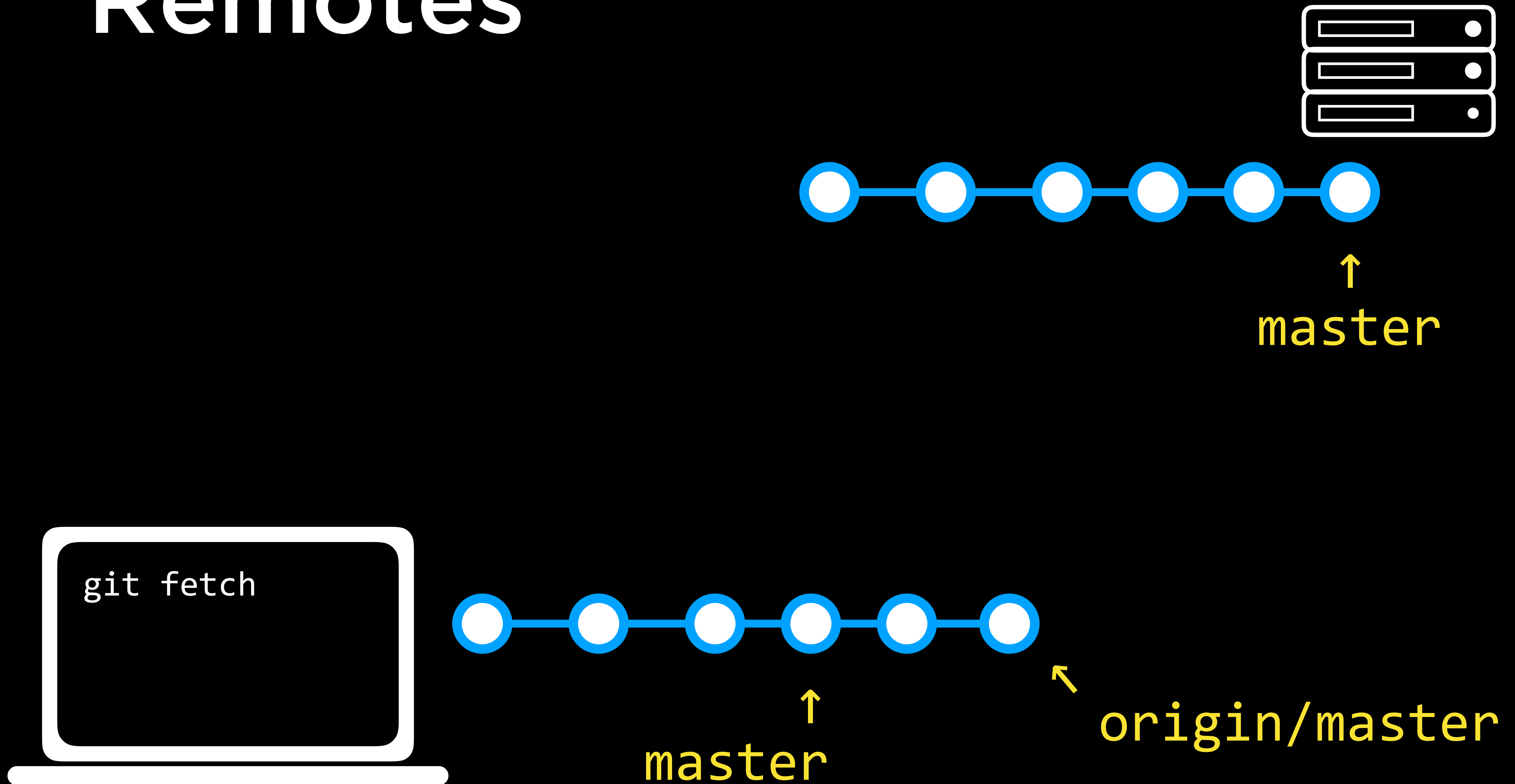


master

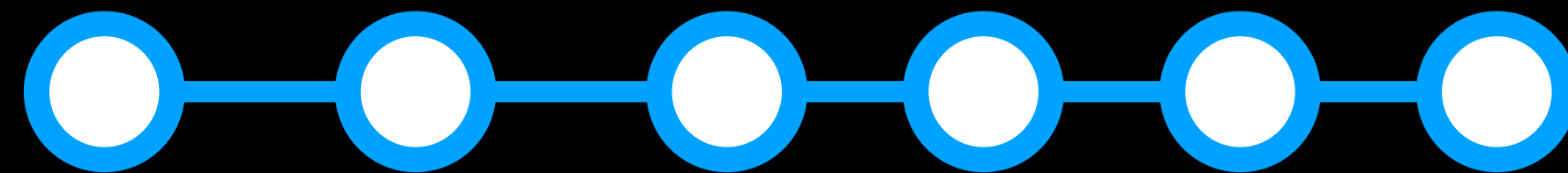
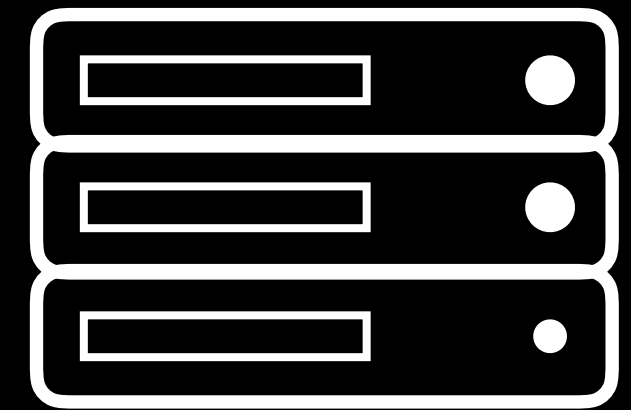
Remotes



Remotes

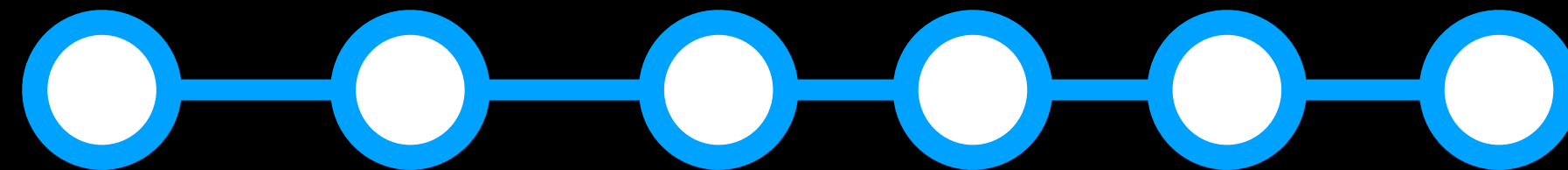


Remotes



master

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

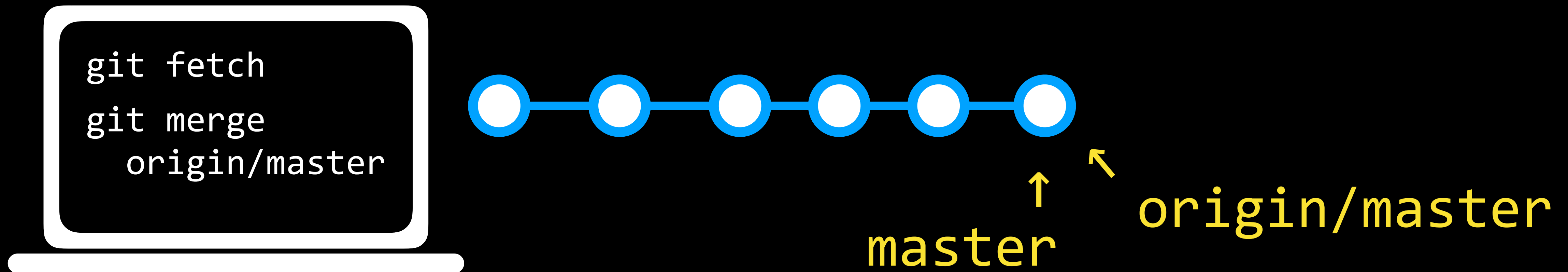
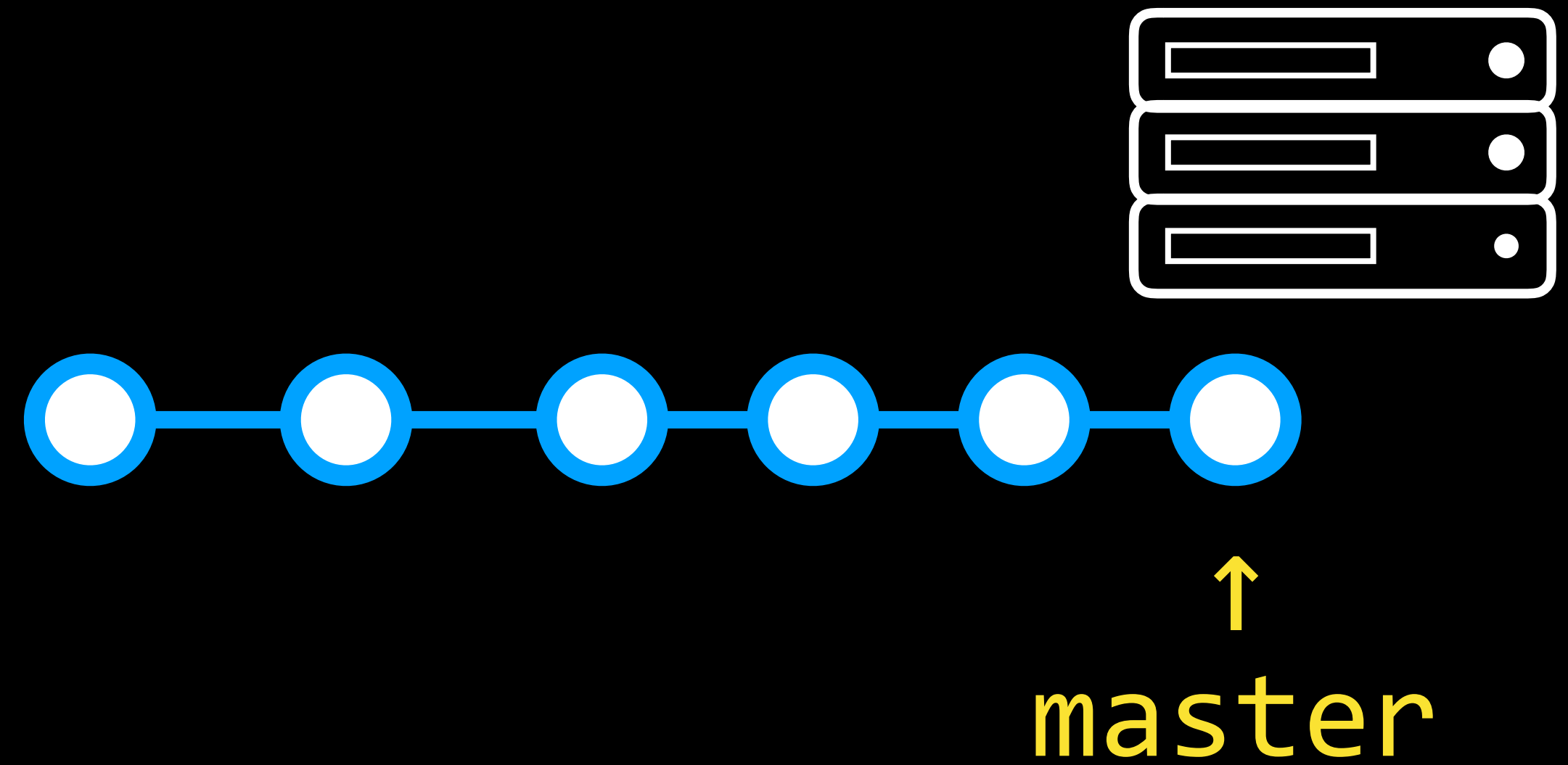


master

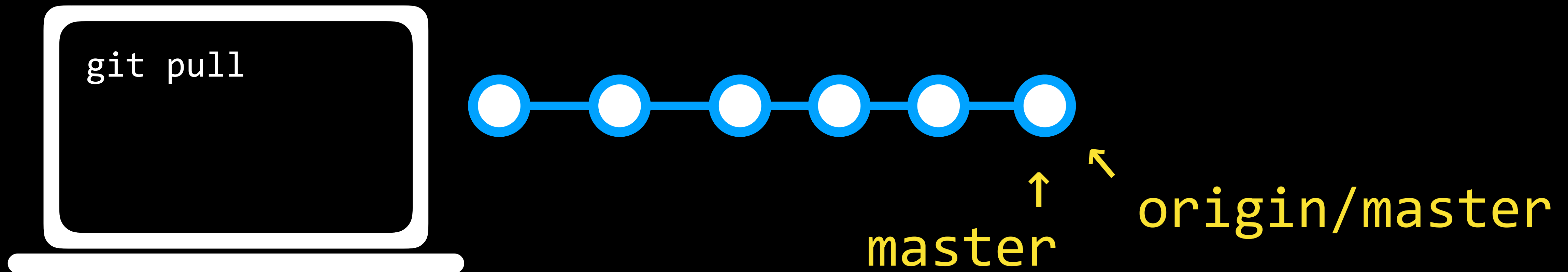
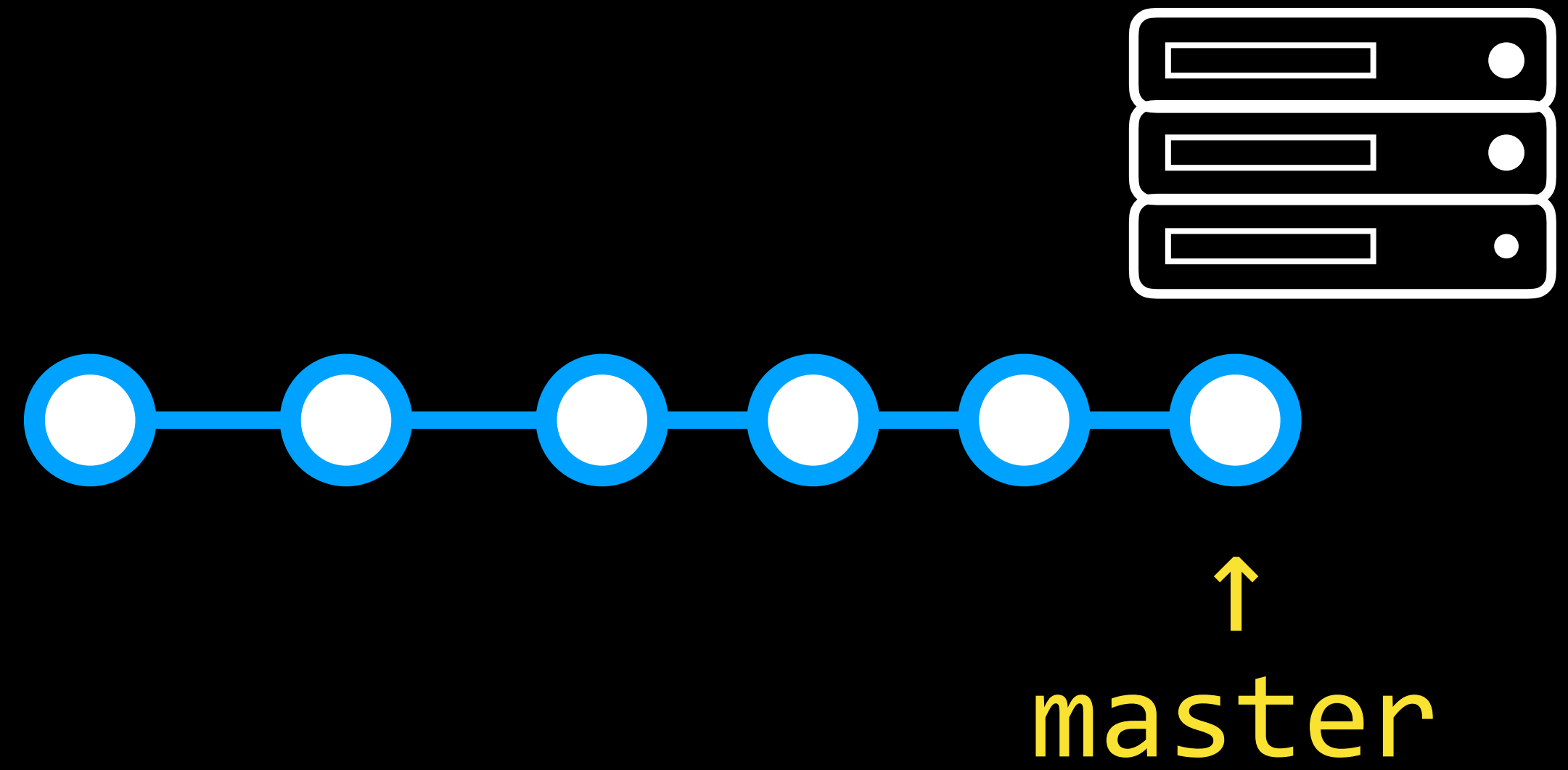


origin/master

Remotes



Remotes



Forks

Pull Requests

Q & A