

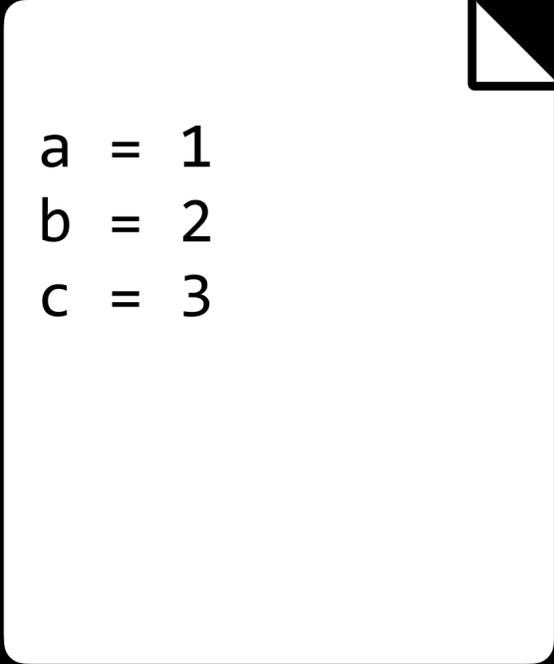
# Web Programming

with Python and JavaScript

**Git**

git

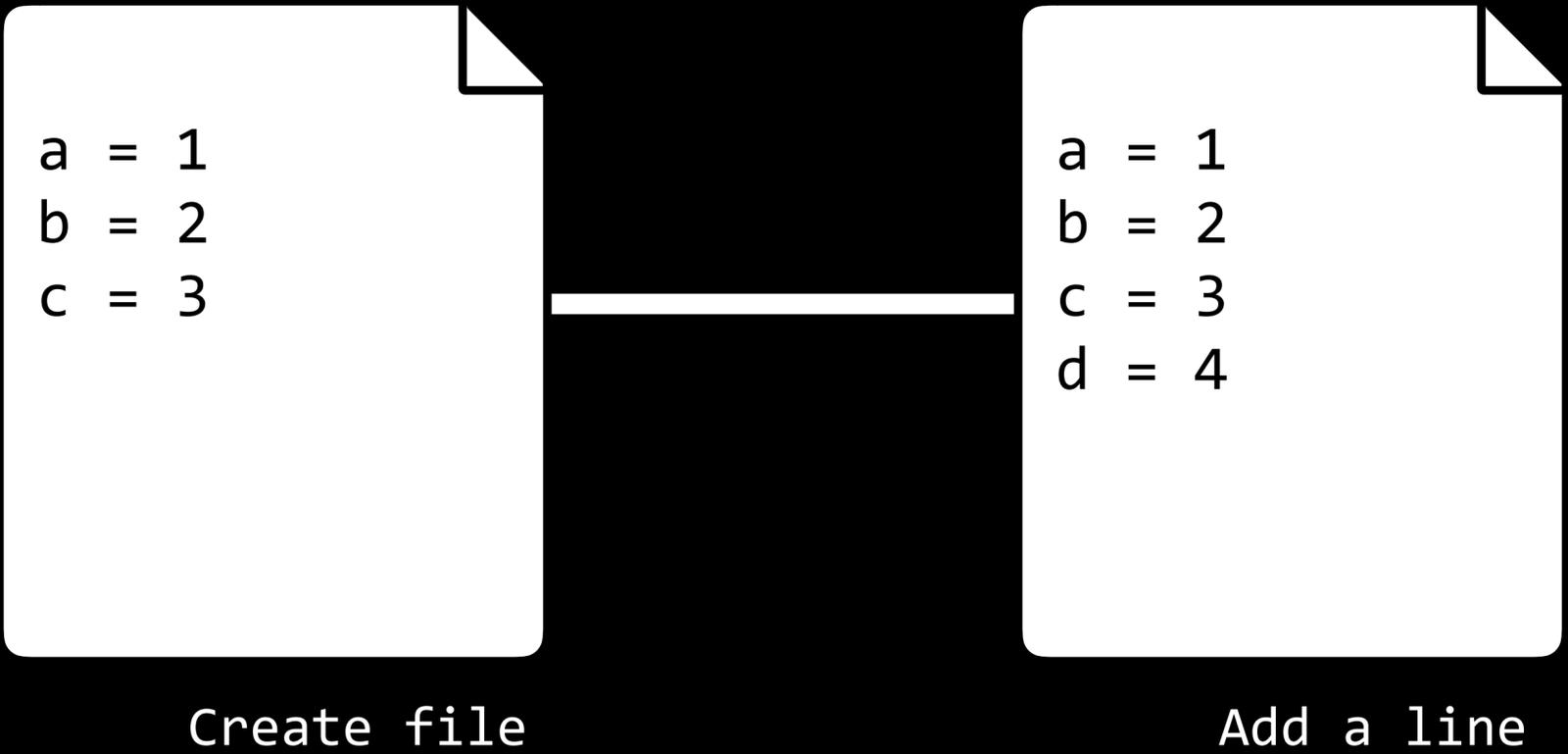
Keep track of changes to code.



```
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.

The diagram illustrates a sequence of three code files connected by horizontal lines. Each file is represented as a white document icon with a folded top-right corner. The first file contains three lines of code: 'a = 1', 'b = 2', and 'c = 3'. The second file contains four lines: 'a = 1', 'b = 2', 'c = 3', and 'd = 4'. The third file contains three lines: 'a = 1', 'c = 3', and 'd = 4'. The label 'Remove a line' is positioned below the third file, indicating that the line 'b = 2' from the second file has been removed.

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

Create file

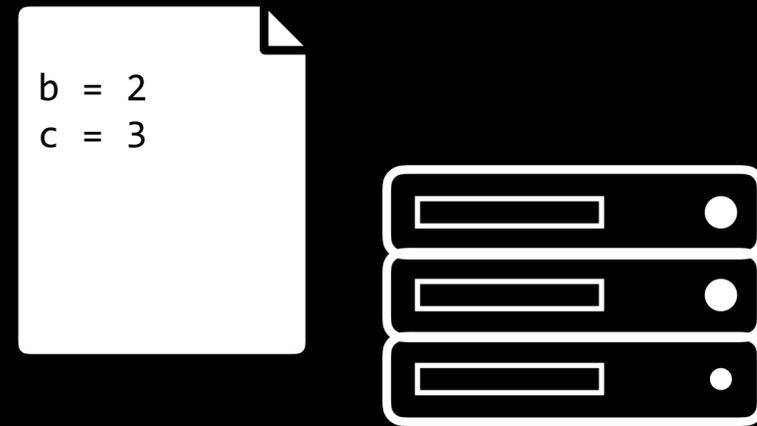
```
a = 1  
b = 2  
c = 3  
d = 4
```

Add a line

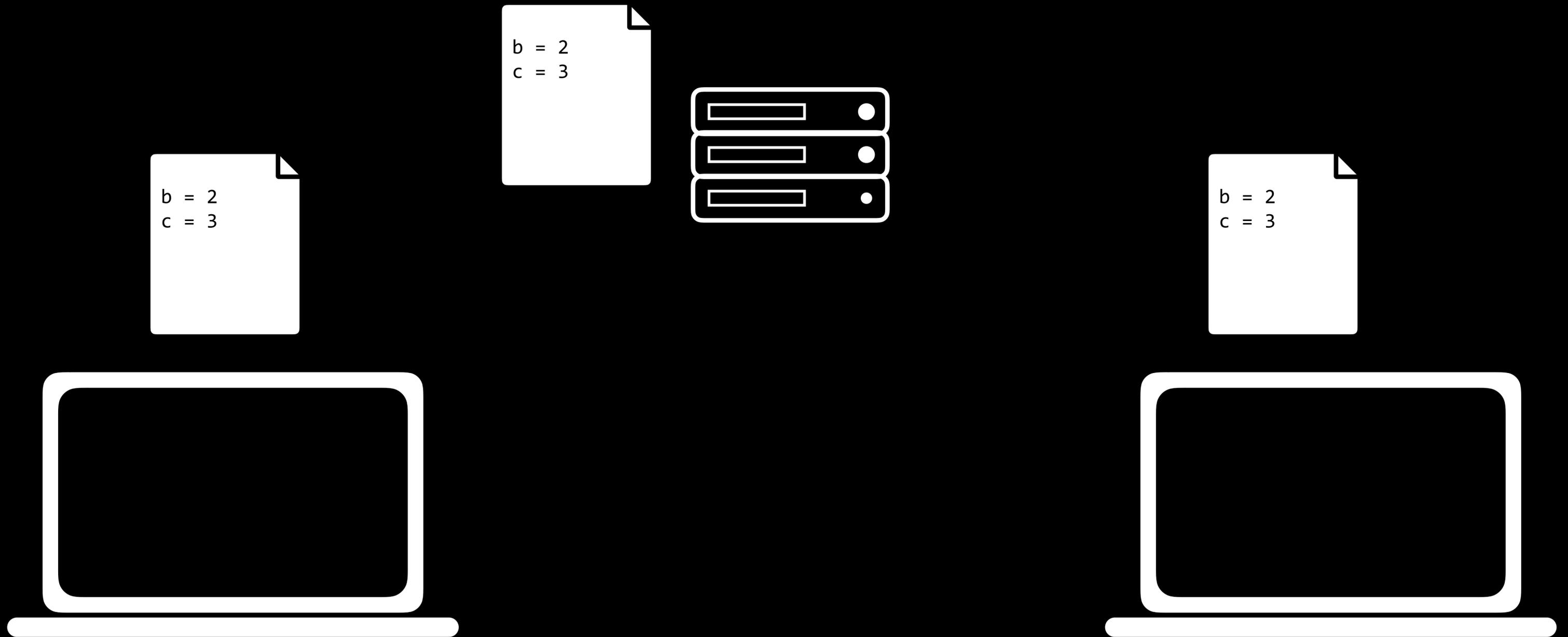
```
a = 1  
c = 3  
d = 4
```

Remove a line

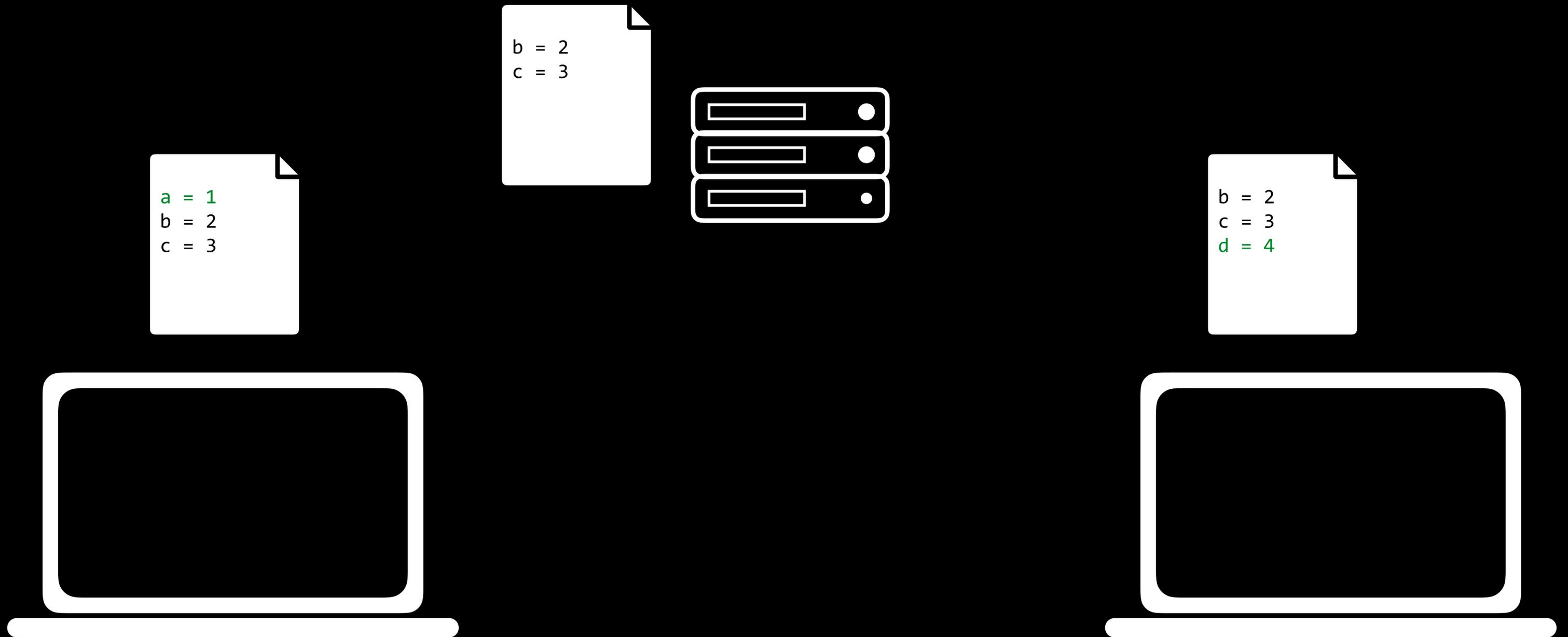
Synchronizes code between different people.



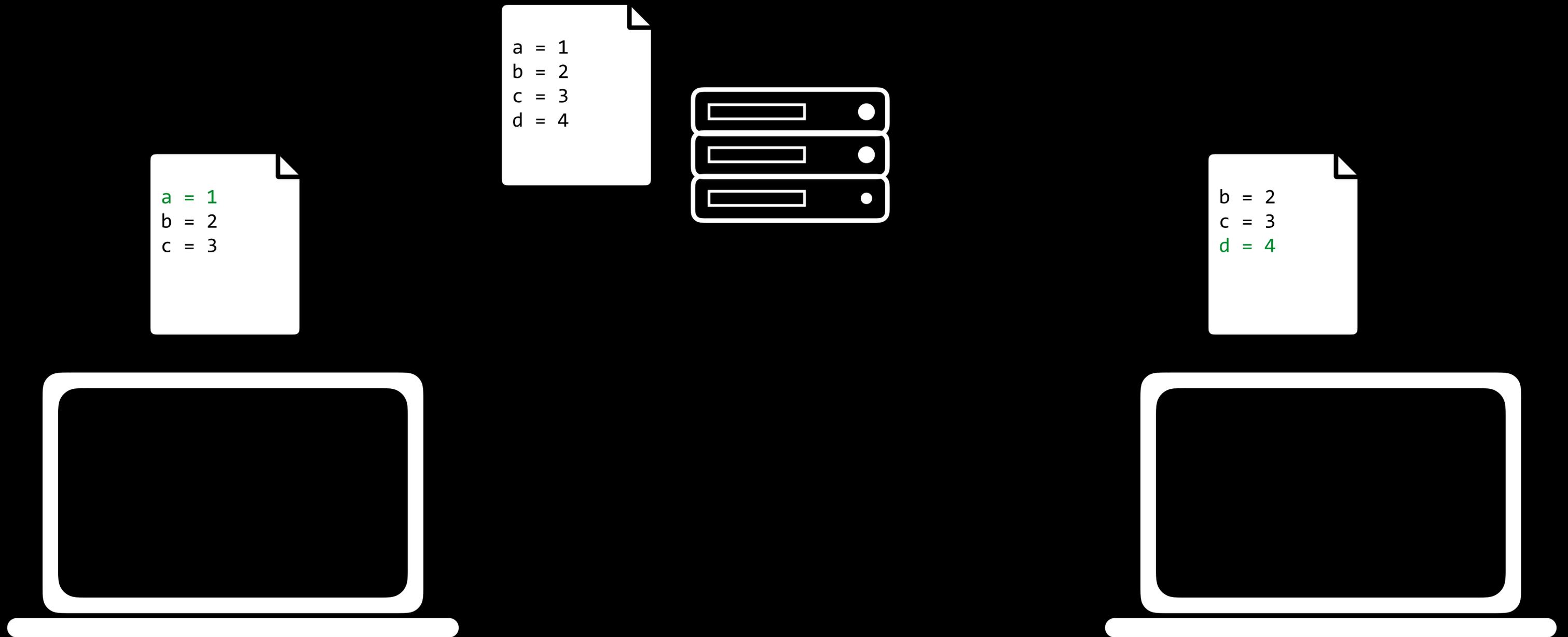
Synchronizes code between different people.



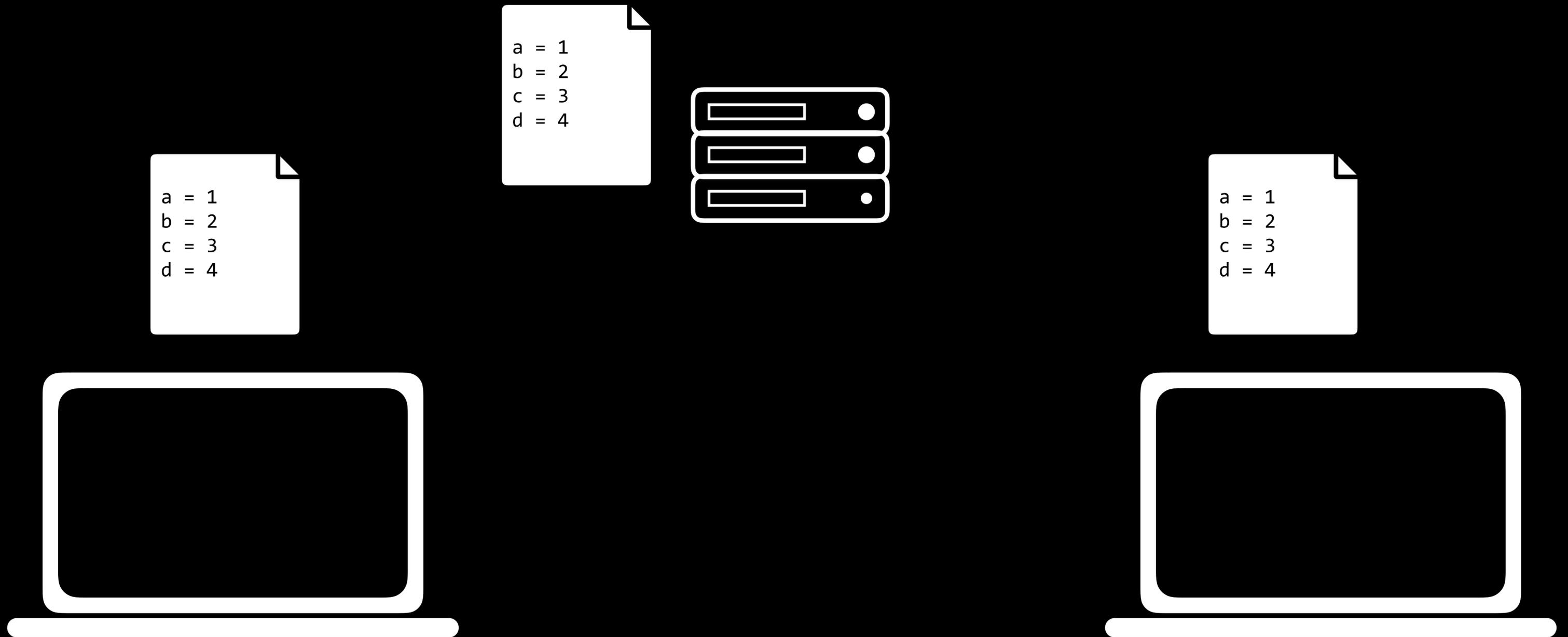
Synchronizes code between different people.



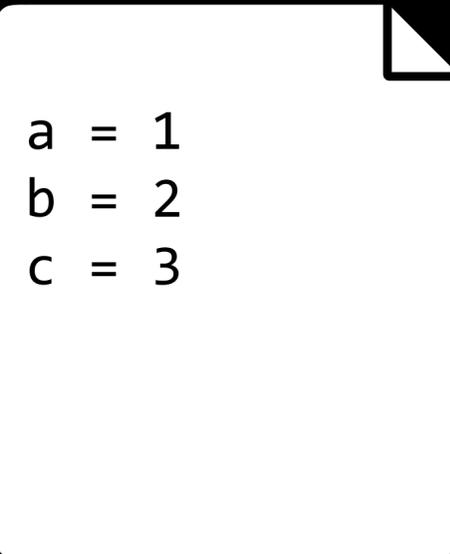
Synchronizes code between different people.



Synchronizes code between different people.

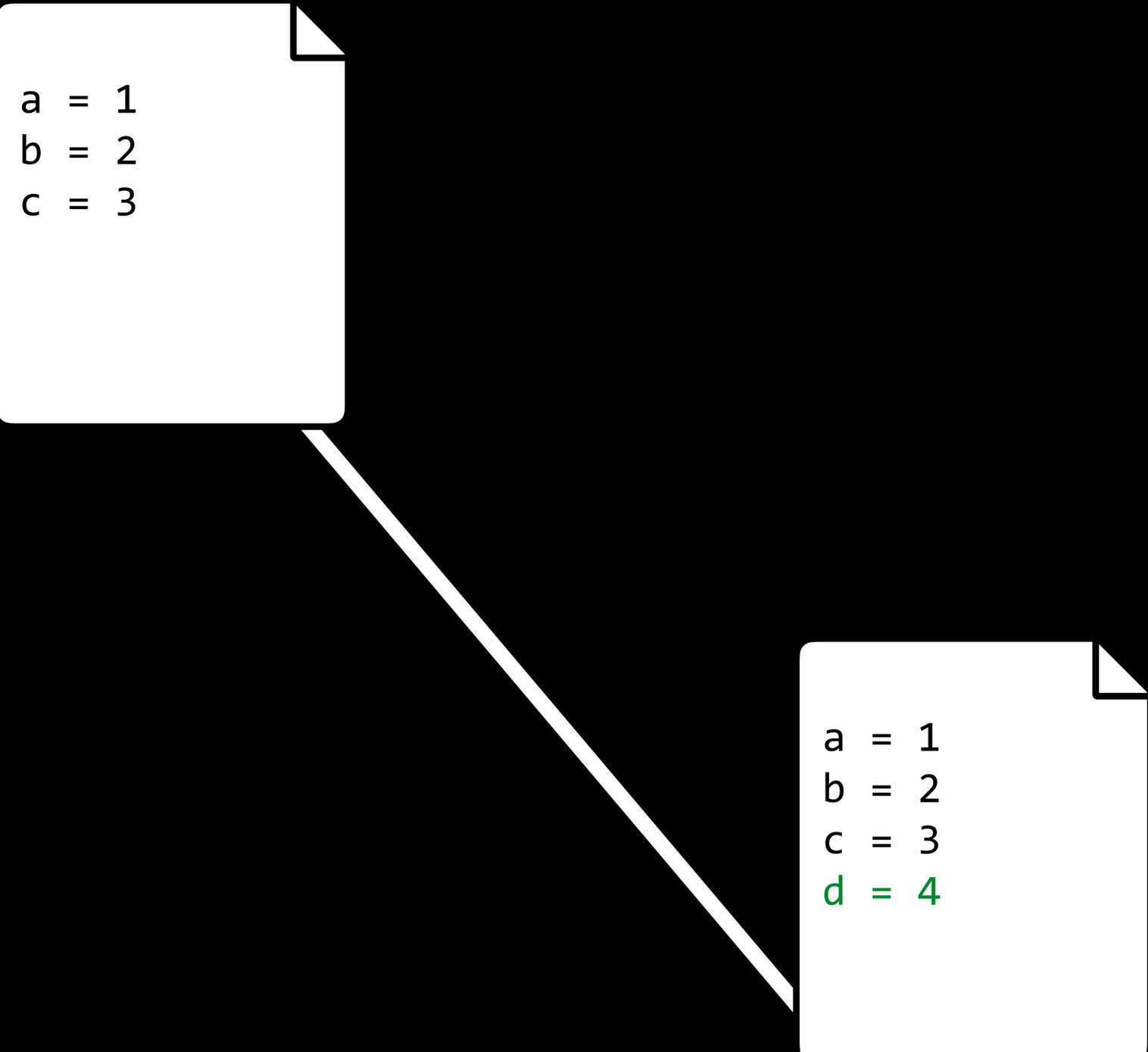


Test changes to code without losing the original.



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

Test changes to code without losing the original.

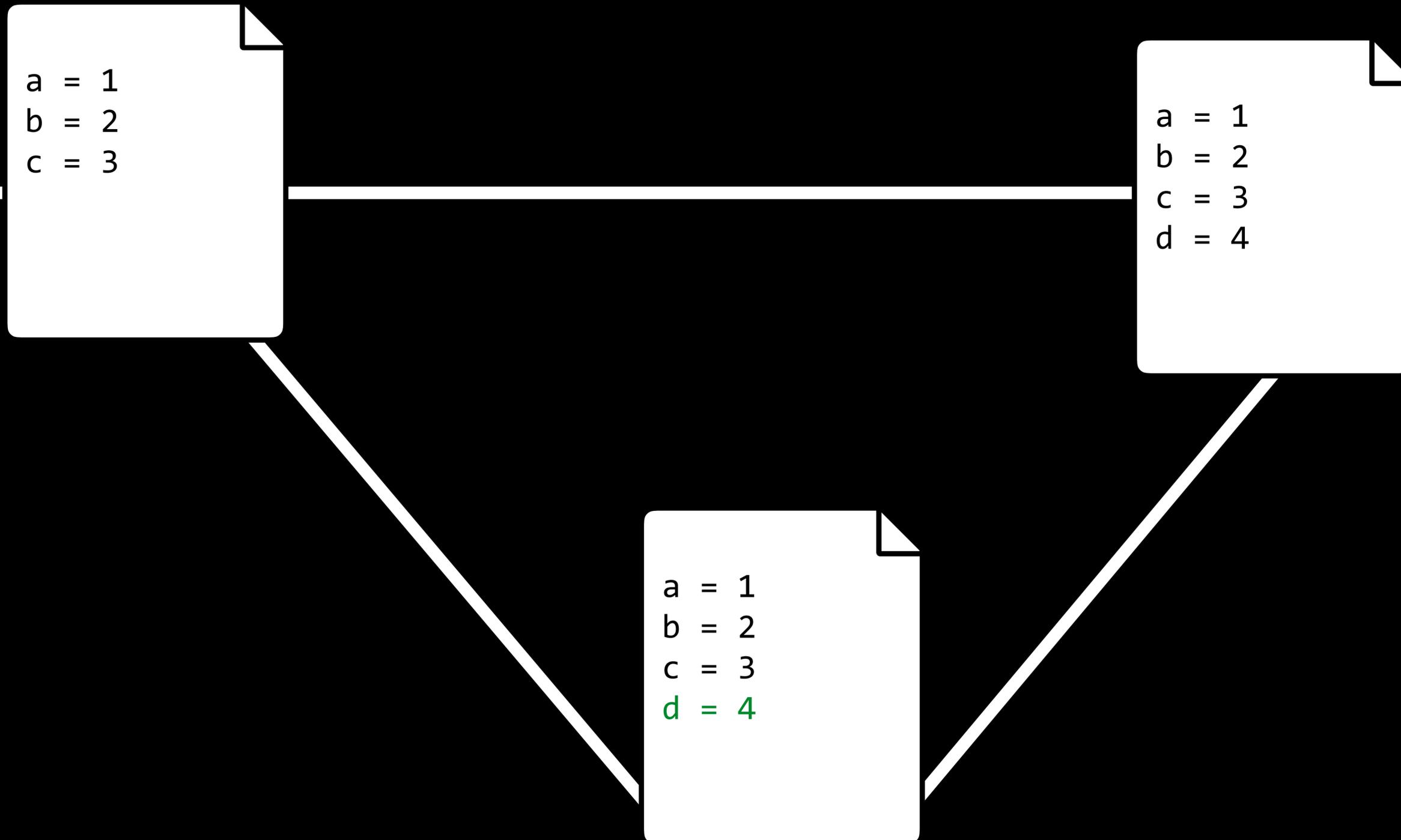


The diagram illustrates a branching strategy for testing code. It features two white document icons on a black background. The top-left icon represents the original code, containing three lines: `a = 1`, `b = 2`, and `c = 3`. A horizontal line extends from the left edge of this icon. A diagonal line branches off from the bottom-right corner of the top icon and points to the top-left corner of a second icon located below and to the right. This second icon represents a test branch, containing the same three lines as the original, plus a fourth line: `d = 4`. The fourth line is highlighted in green, indicating a change or a new addition to the code.

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

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

Test changes to code without losing the original.



Revert back to old versions of code.

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

Create file

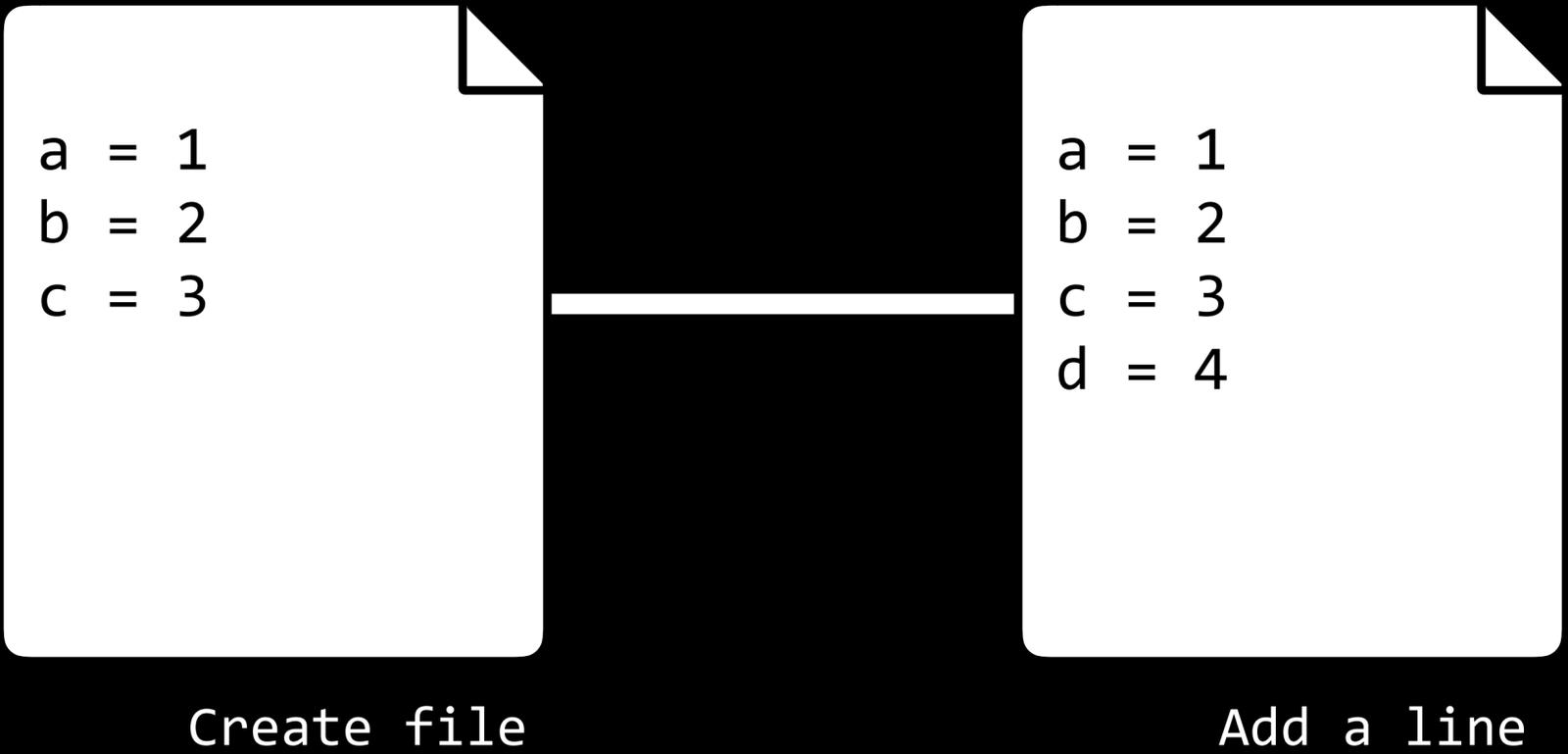
```
a = 1  
b = 2  
c = 3  
d = 4
```

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
```

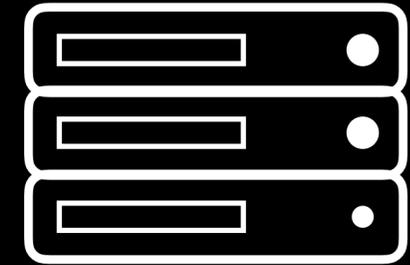
Add a line

**GitHub**

```
git clone
```

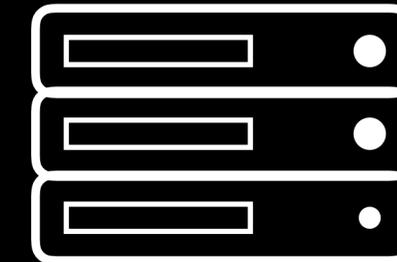
```
git clone <url>
```

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



```
git clone <url>
```

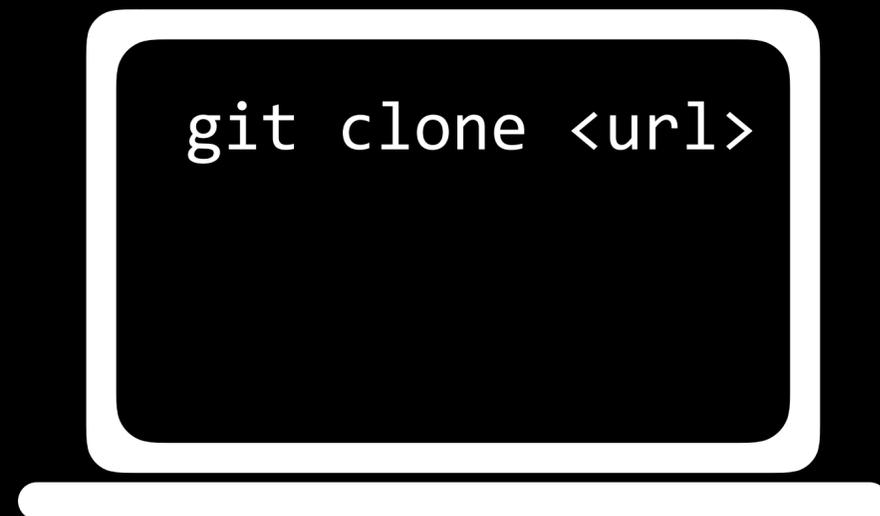
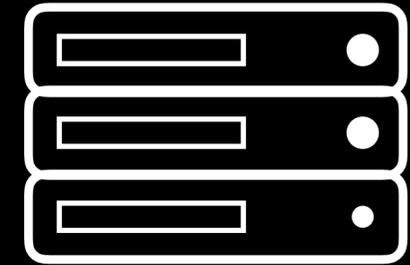
```
a = 1  
b = 2  
c = 3  
d = 4
```



```
git clone <url>
```

```
git clone <url>
```

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

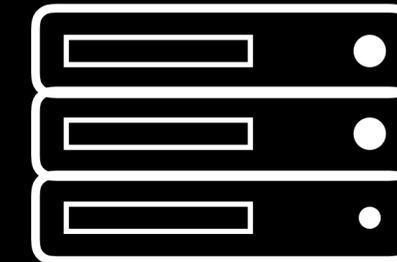


```
a = 1  
b = 2  
c = 3  
d = 4
```

```
git add
```

```
git add <filename>
```

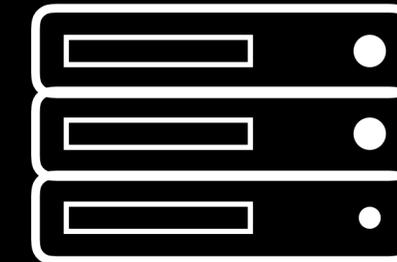
```
a = 1  
b = 2  
c = 3  
d = 4
```



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

```
git add <filename>
```

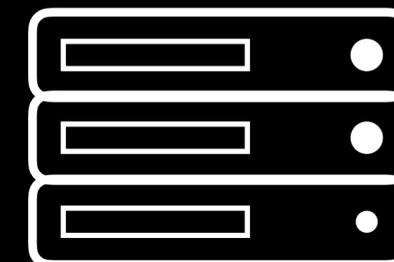
```
a = 1  
b = 2  
c = 3  
d = 4
```



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

```
git add <filename>
```

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

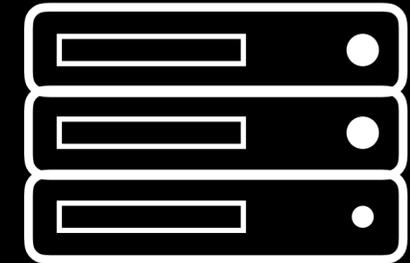


```
git add foo.py
```

```
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
```

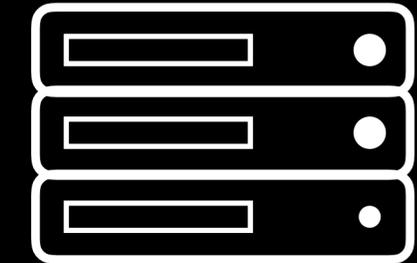
Changes to be committed:

```
modified: foo.py
```

```
git commit
```

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

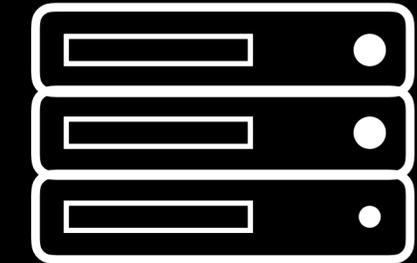
```
a = 1  
b = 2  
c = 3  
d = 4
```



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

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

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

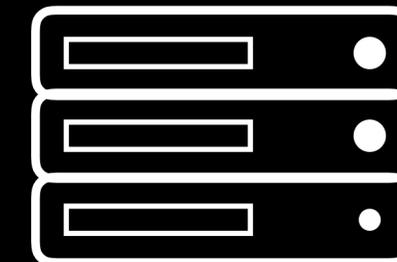


```
git commit -m  
"Add line"
```

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

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

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



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



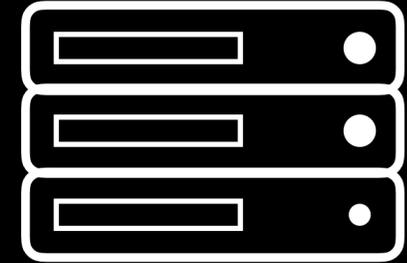
```
a = 1  
b = 2  
c = 3  
d = 4  
e = 5
```

Add line

```
git status
```

git status

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



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

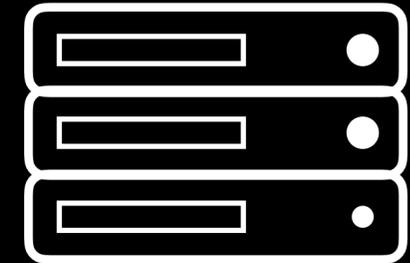
```
a = 1  
b = 2  
c = 3  
d = 4  
e = 5
```

Add line



git status

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



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

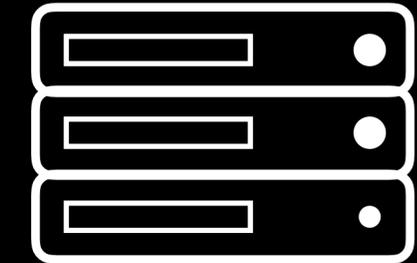
```
a = 1  
b = 2  
c = 3  
d = 4  
e = 5
```

Add line



git status

```
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

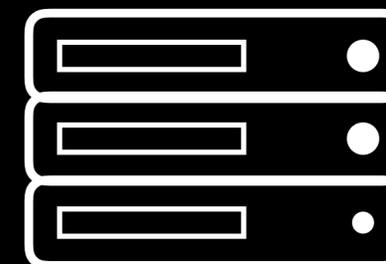
Your branch is ahead of 'origin/master' by 1 commit.

(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
```

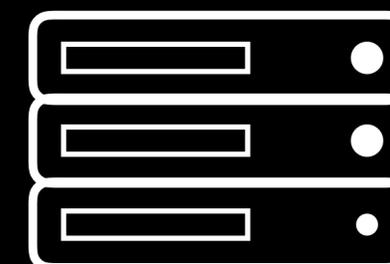


```
a = 1  
b = 2  
c = 3  
d = 4  
e = 5
```

Add line

git push

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



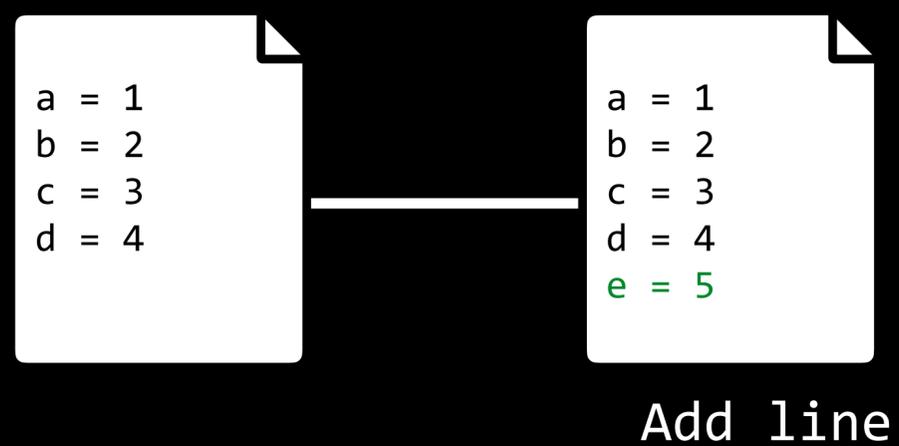
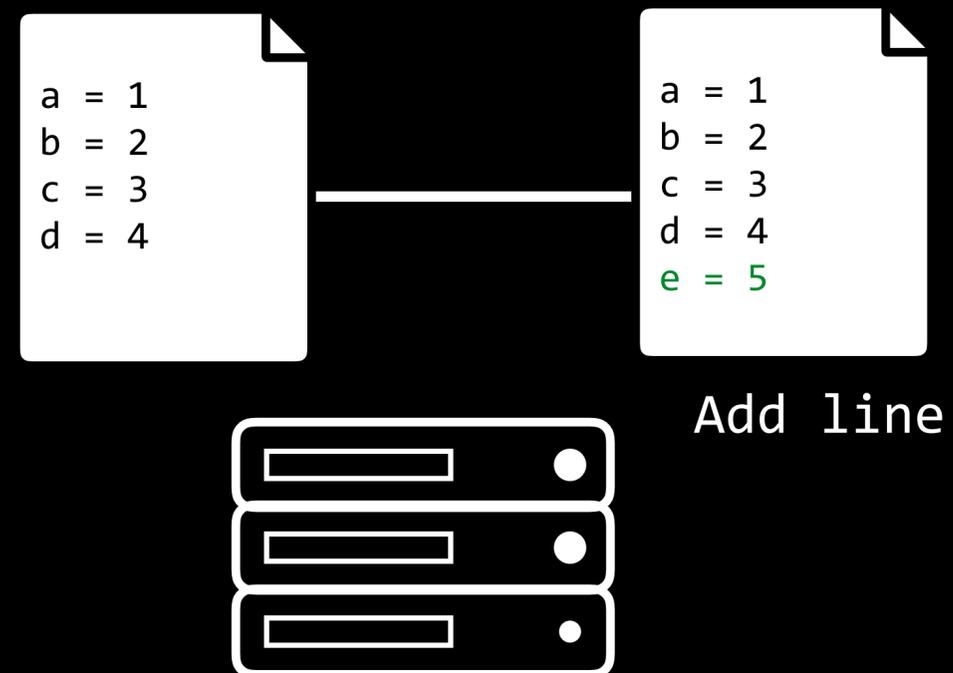
```
a = 1  
b = 2  
c = 3  
d = 4
```



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

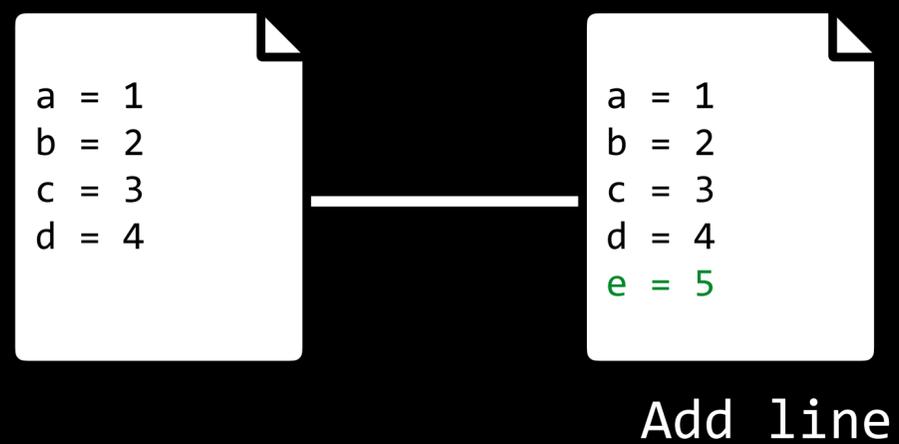
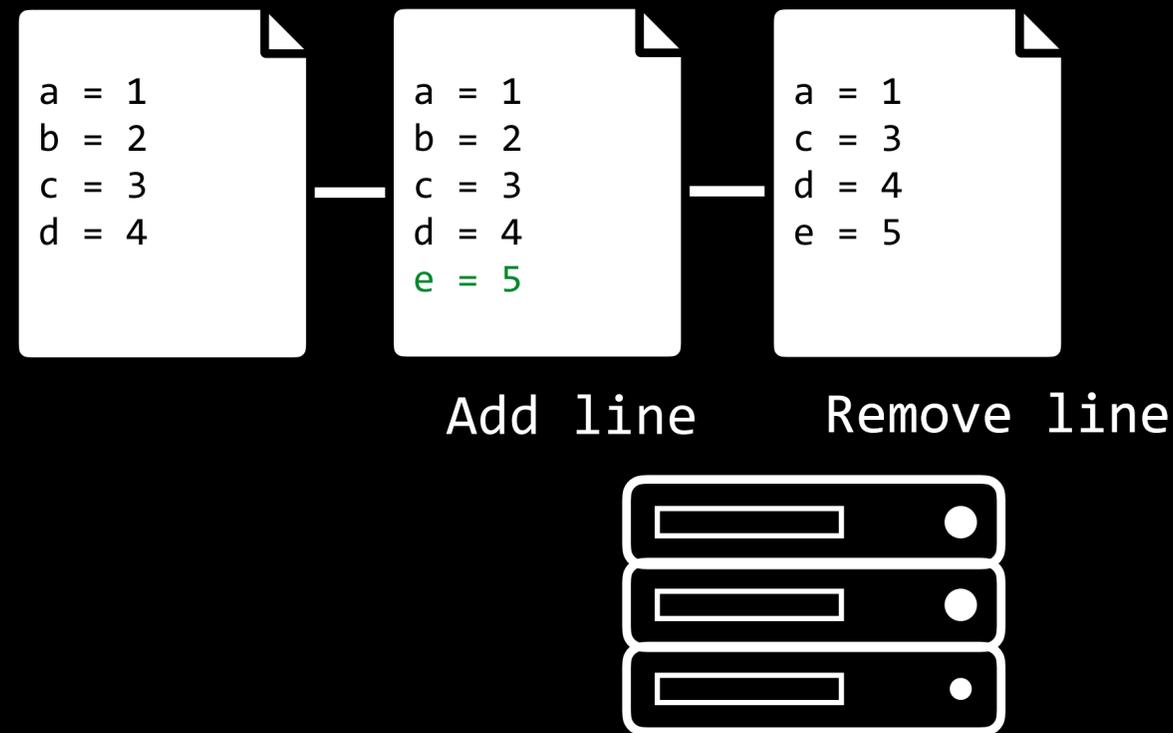
Add line

git push

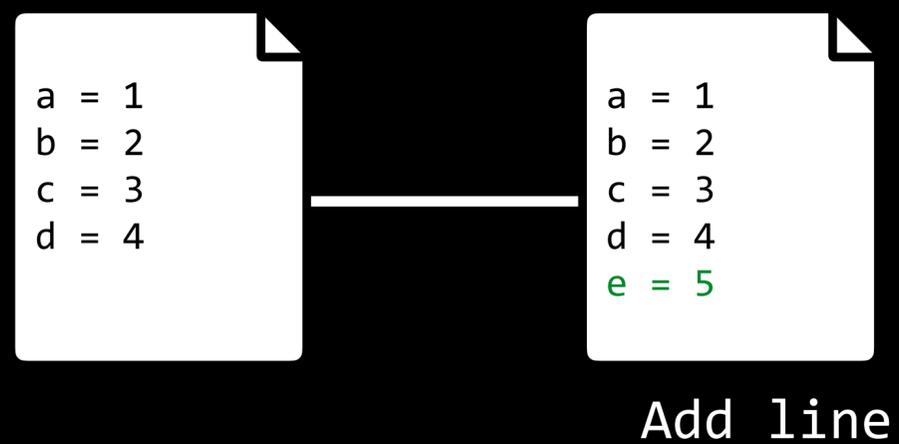
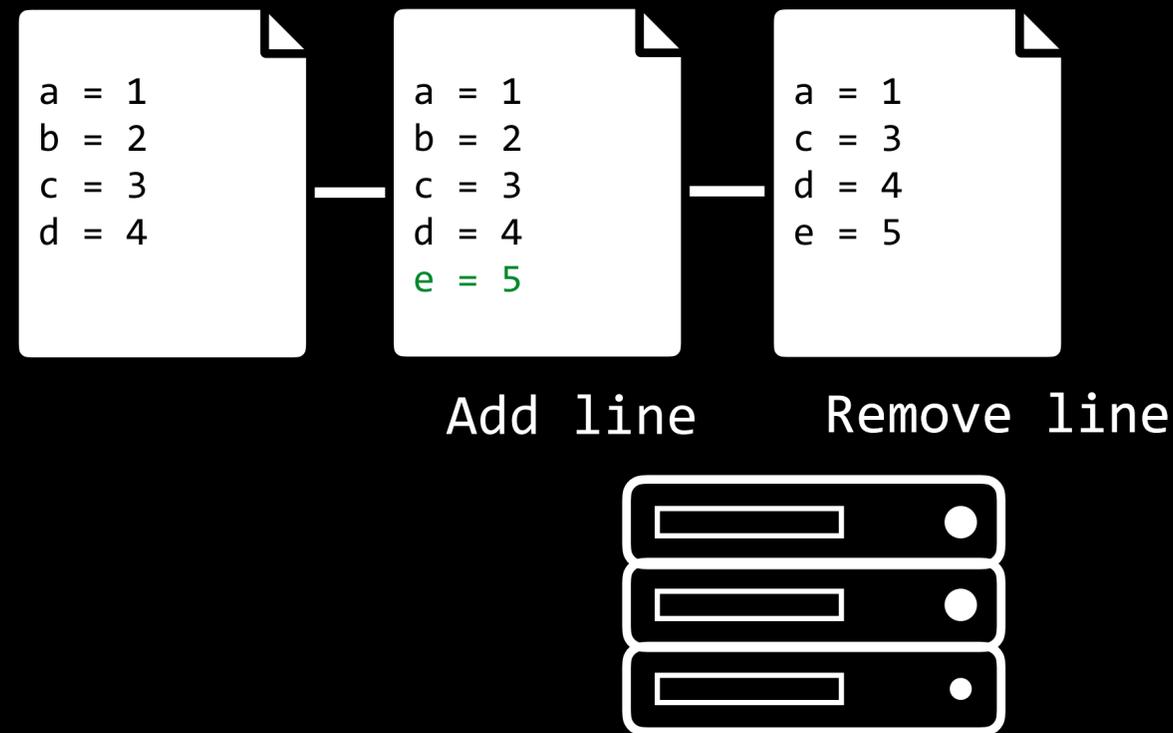


```
git pull
```

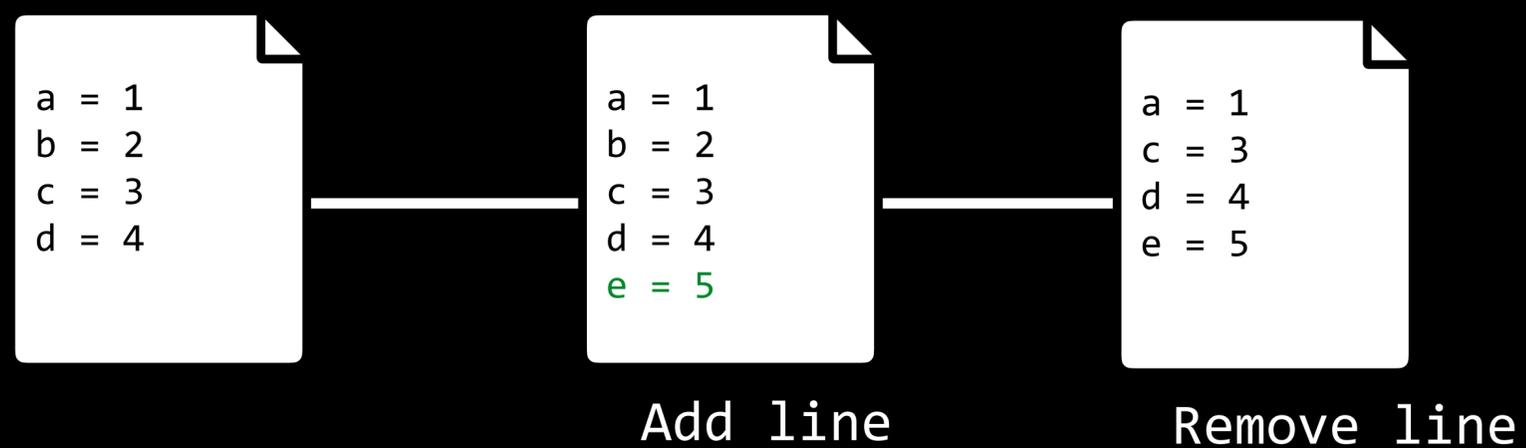
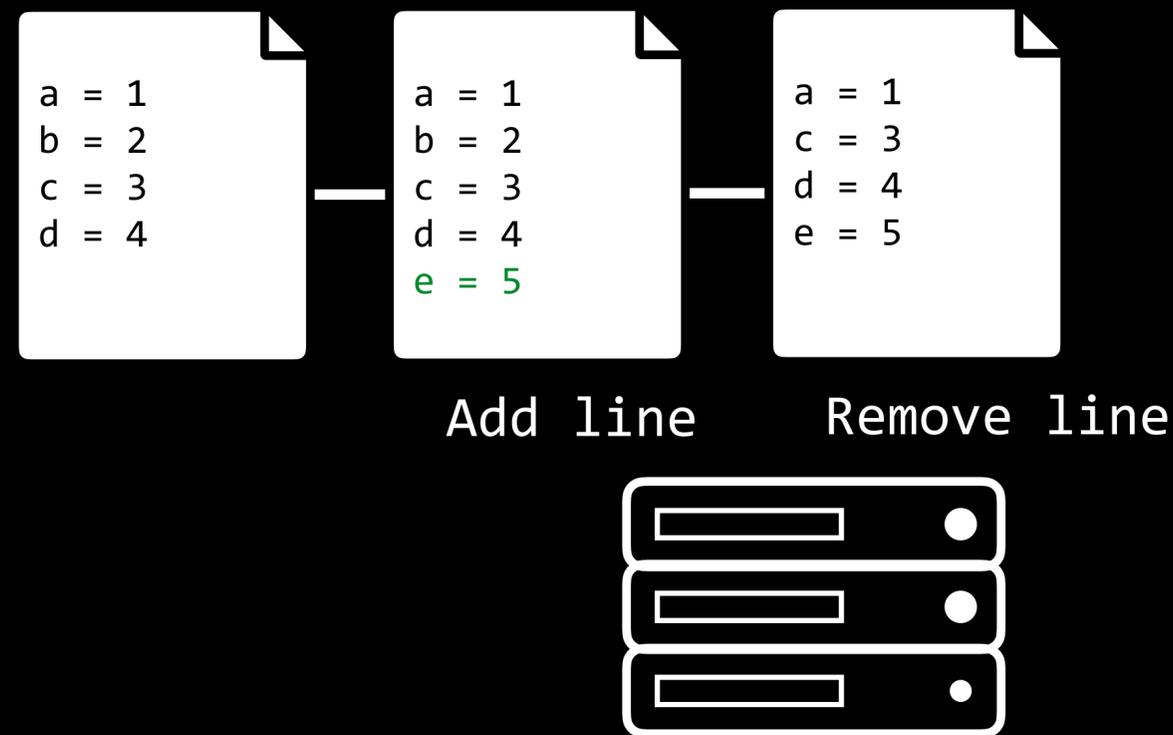
git pull



git pull



git pull



# Merge Conflicts

# Merge Conflicts

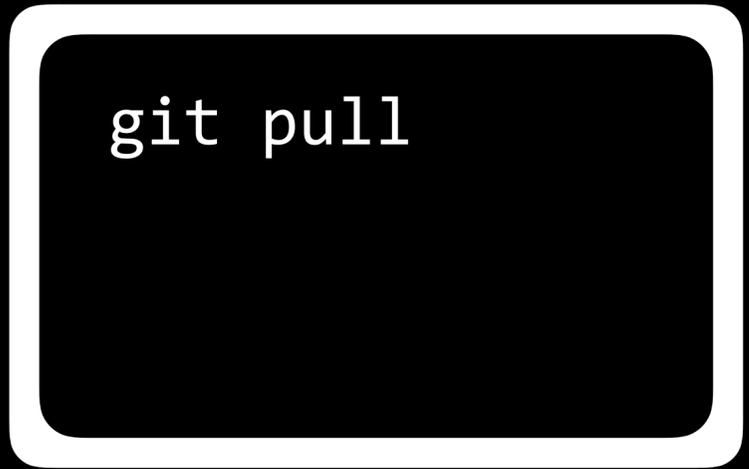


# Merge Conflicts



```
git pull
```

# Merge Conflicts



```
git pull
```

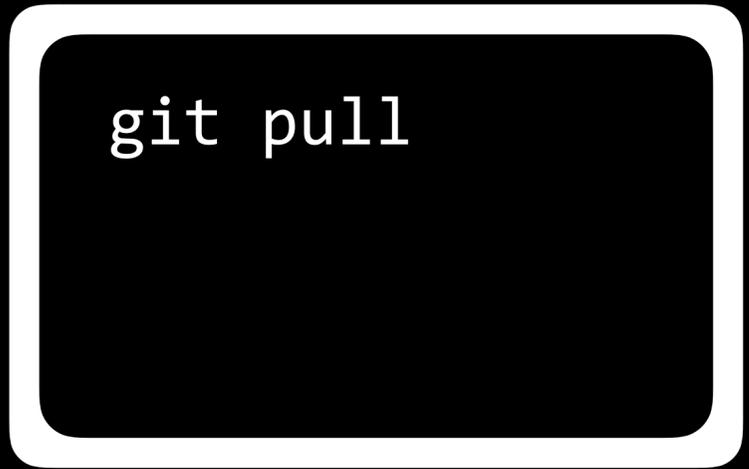
```
CONFLICT (content): Merge conflict in foo.py  
Automatic merge failed; fix conflicts and then  
commit the result.
```

# 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          conflicting commit
{ b = 2
  =====
  { b = 0
    >>>>> 57656c636f6d6520746f20576562
c = 3
d = 4
e = 5
```



# Merge Conflicts

```
git pull
```

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

# Merge Conflicts

```
git pull
```

```
a = 1
```

```
b = 2
```

```
c = 3
```

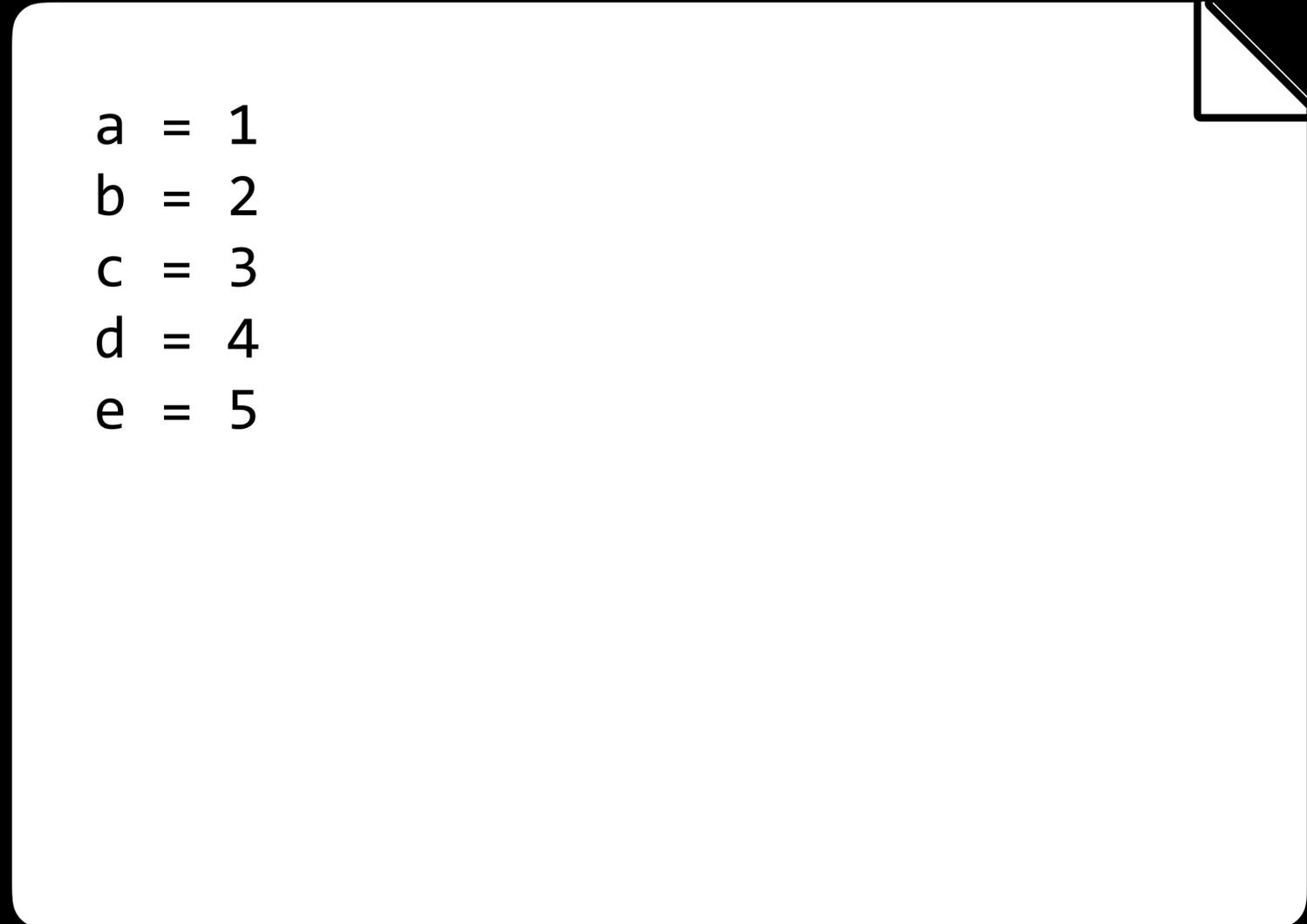
```
d = 4
```

```
e = 5
```

# Merge Conflicts



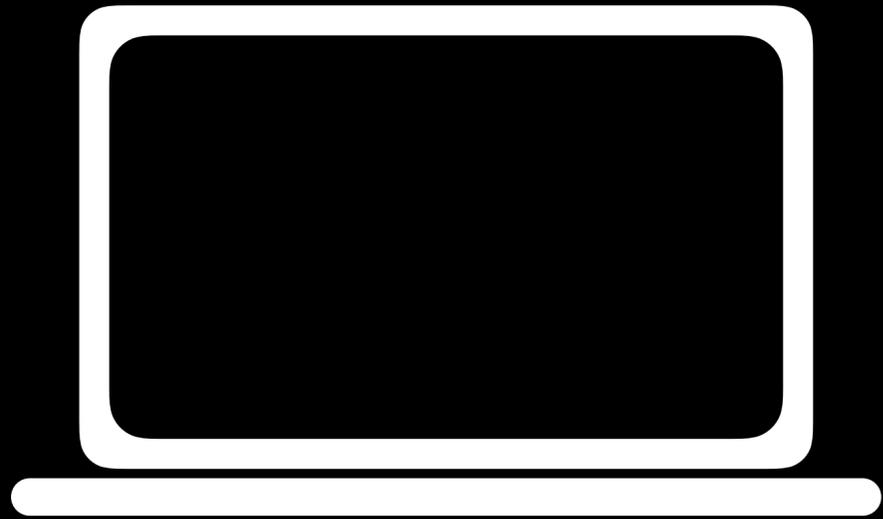
```
git pull
```



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

git log

git log



```
git log
```



git log

```
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`



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

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

Add line  
57656c6

```
a = 3  
c = 3  
d = 4  
e = 5
```

Remove line  
436f6d6

# git reset

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



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

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

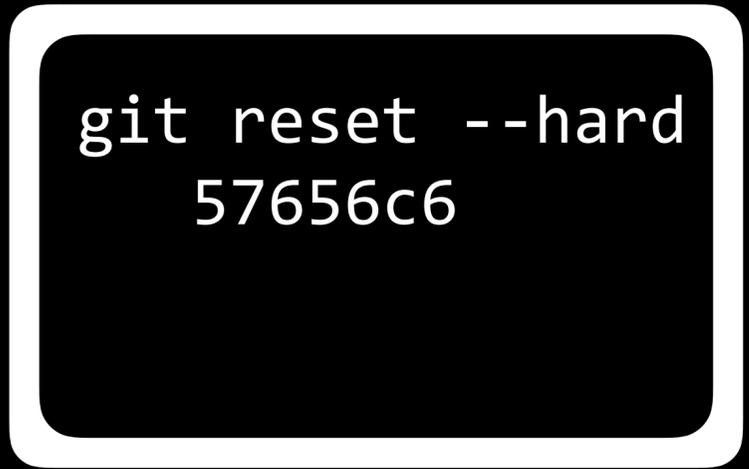
Add line  
57656c6

```
a = 3
c = 3
d = 4
e = 5
```

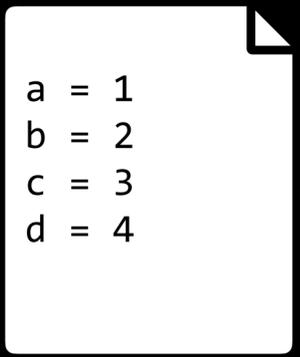
Remove line  
436f6d6

# git reset

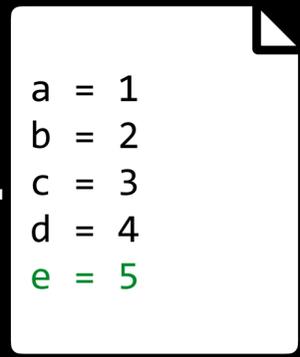
- `git reset --hard <commit>`
- `git reset --hard origin/master`



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



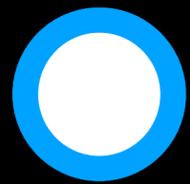
```
a = 1  
b = 2  
c = 3  
d = 4
```



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

Add line  
57656c6

# Making Changes

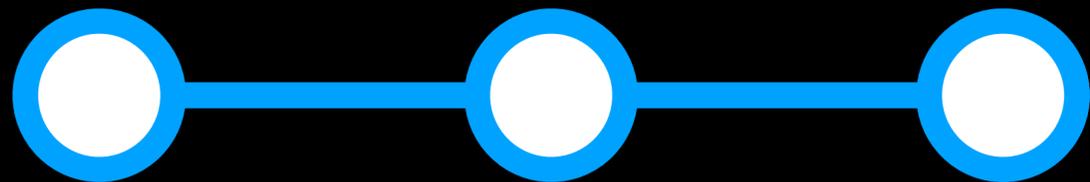


first  
commit



first  
commit

changes



first  
commit

changes

more  
changes

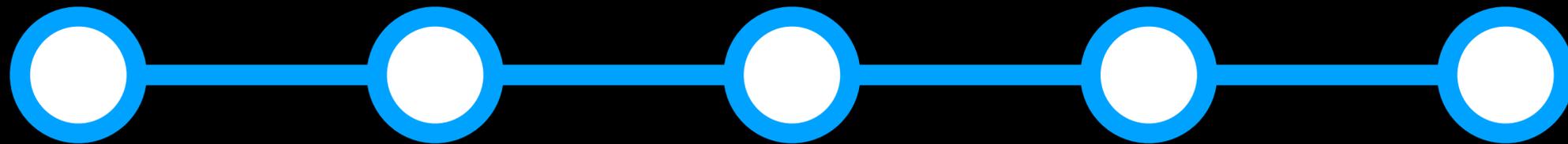


first  
commit

changes

more  
changes

start new  
feature



first  
commit

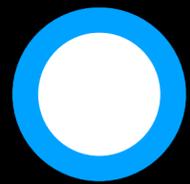
changes

more  
changes

start new  
feature

keep working  
on new feature

# Branching

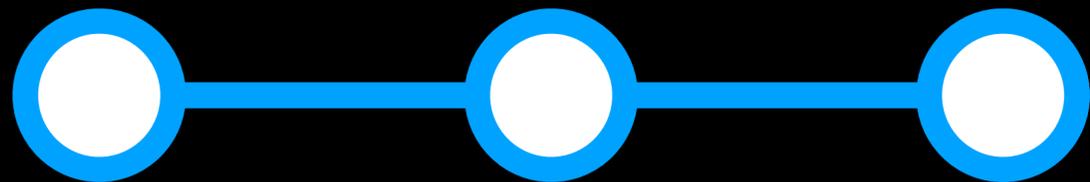


first  
commit



first  
commit

changes

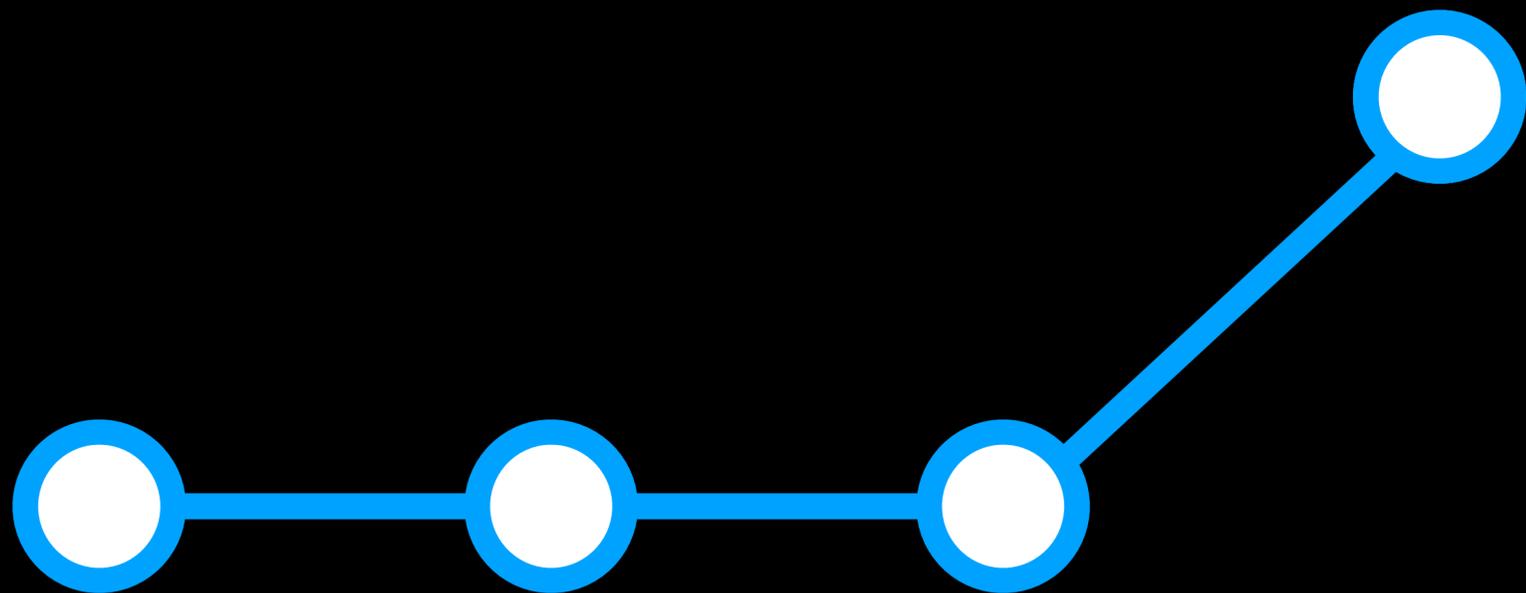


first  
commit

changes

more  
changes

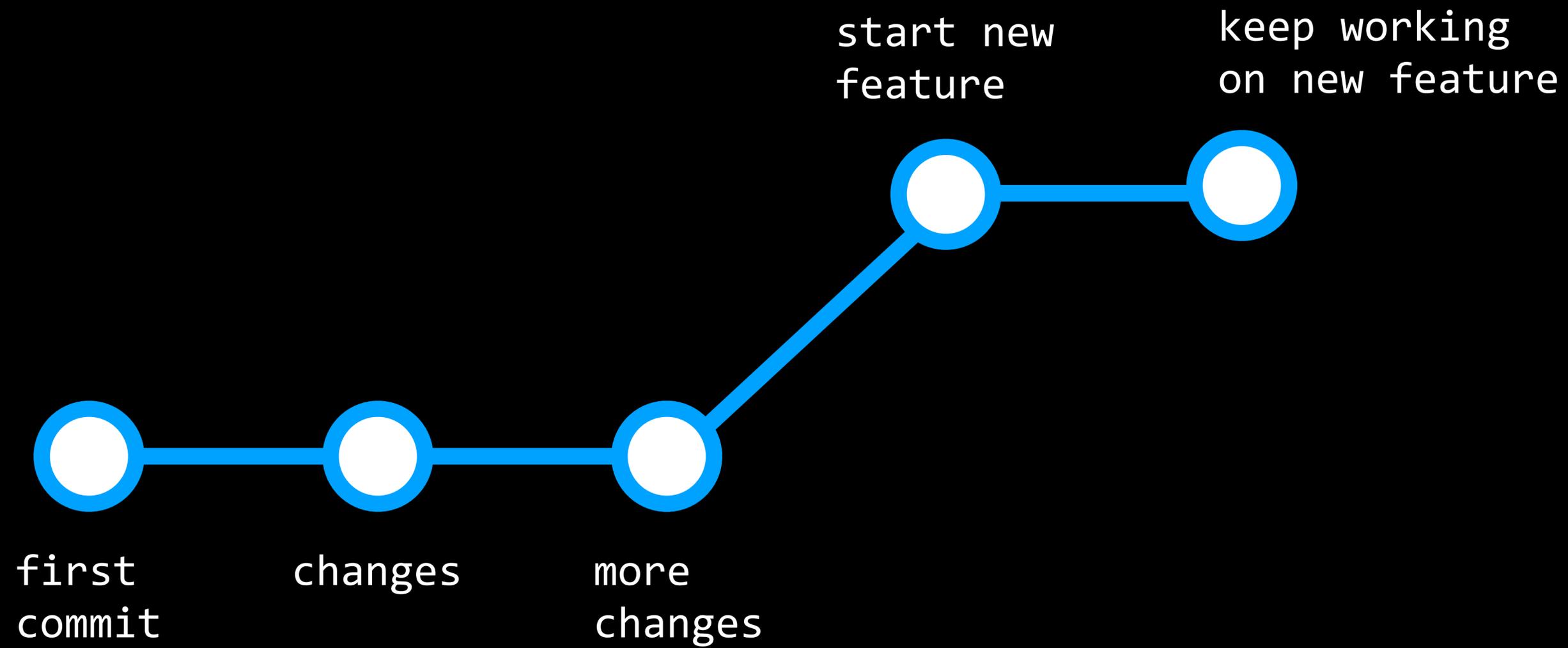
start new  
feature

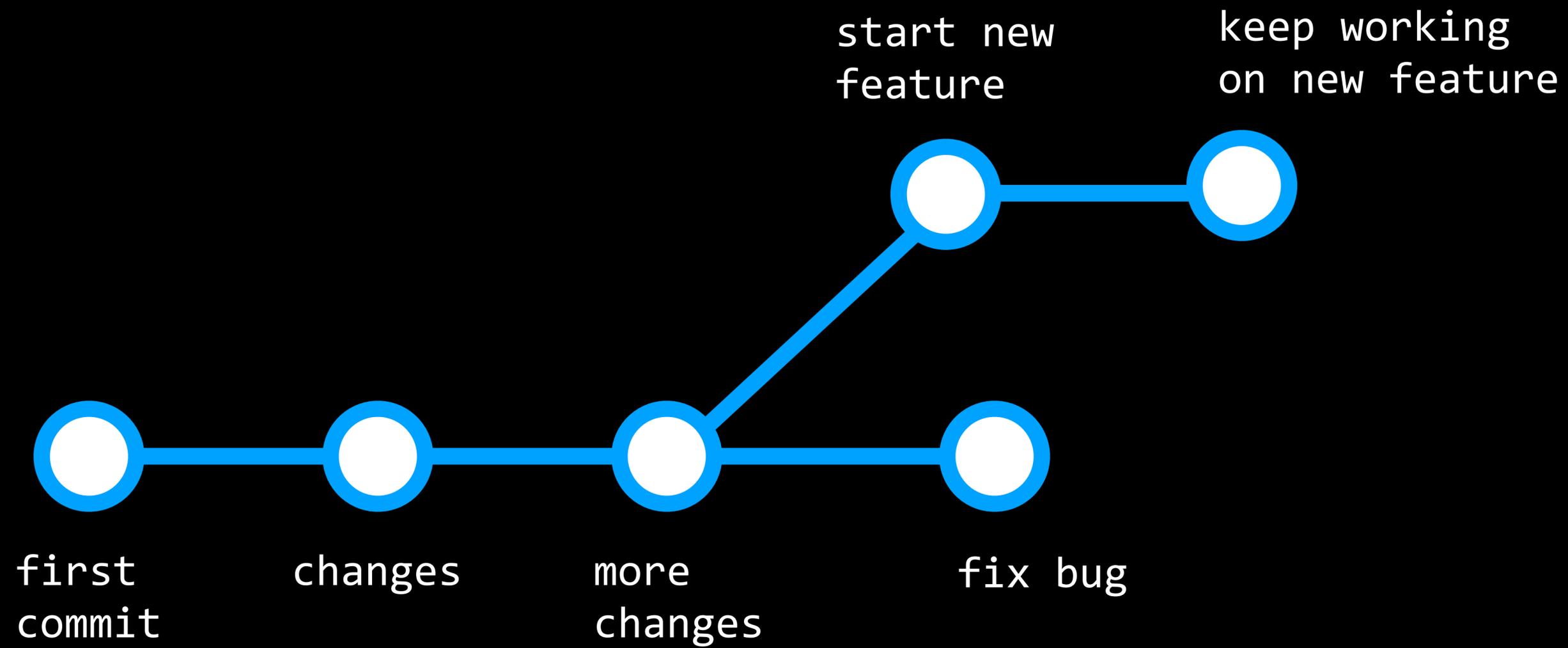


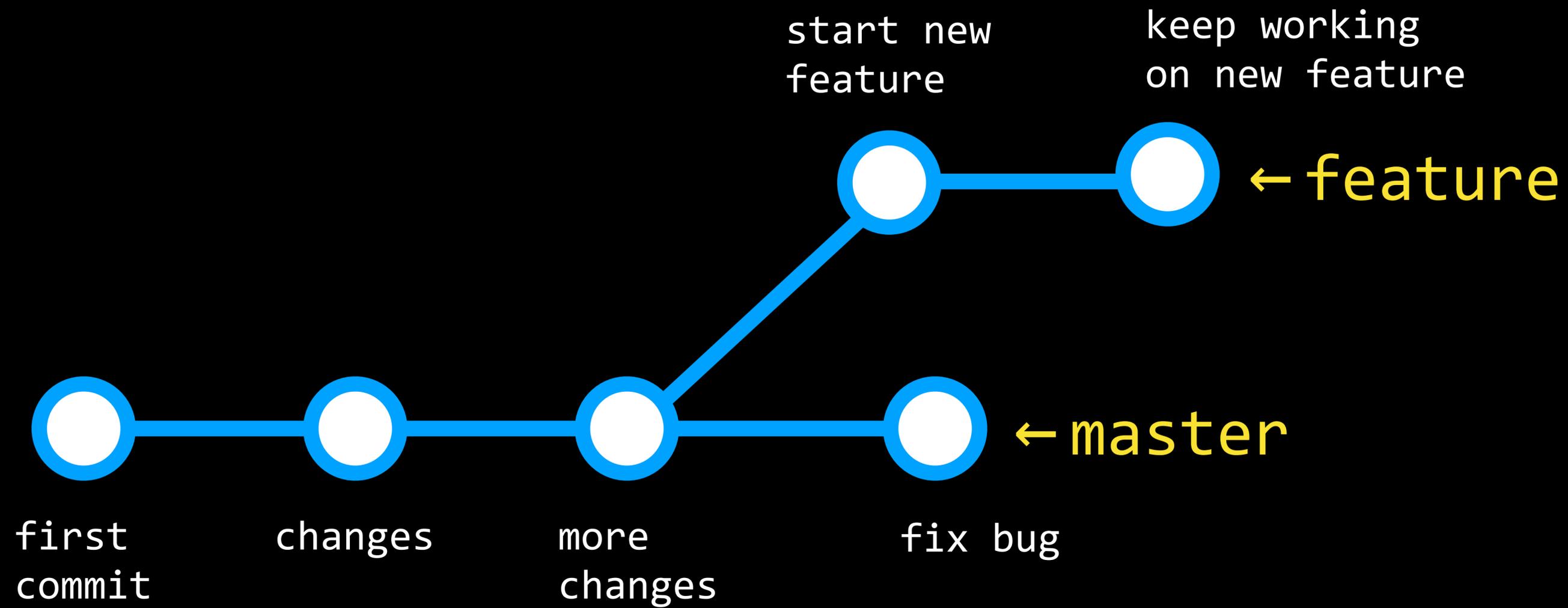
first  
commit

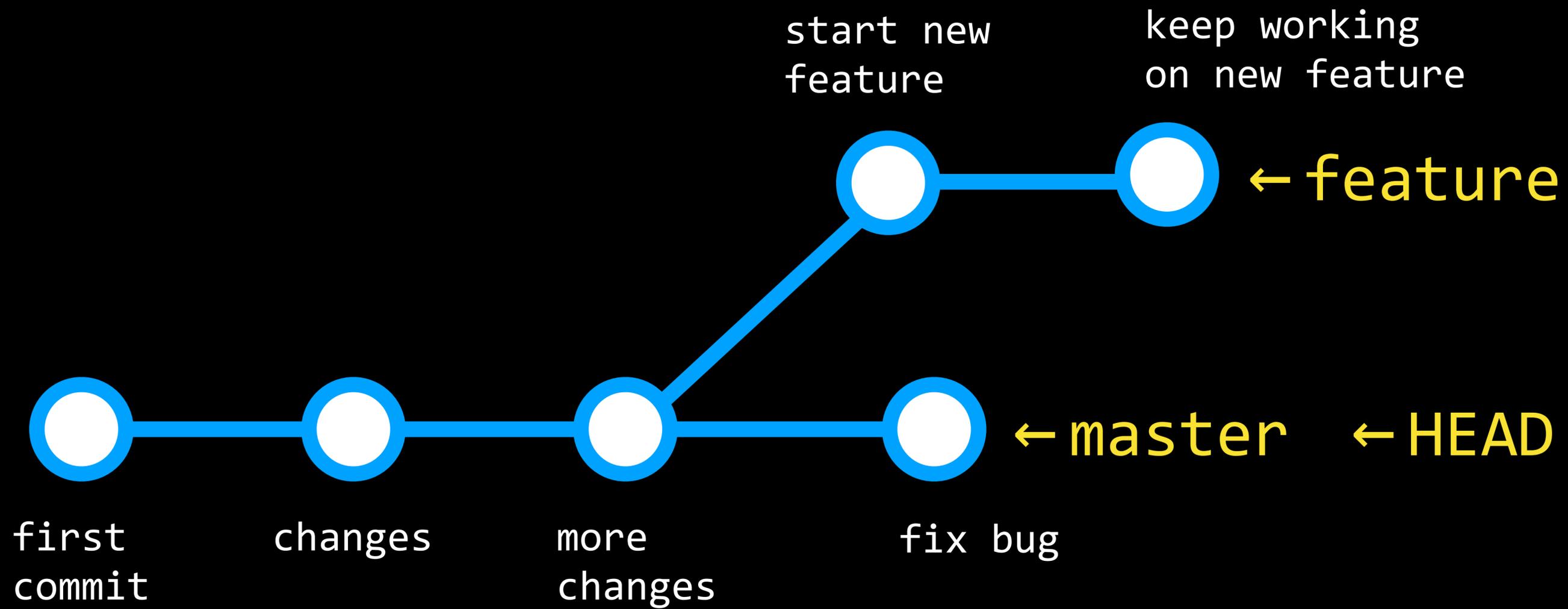
changes

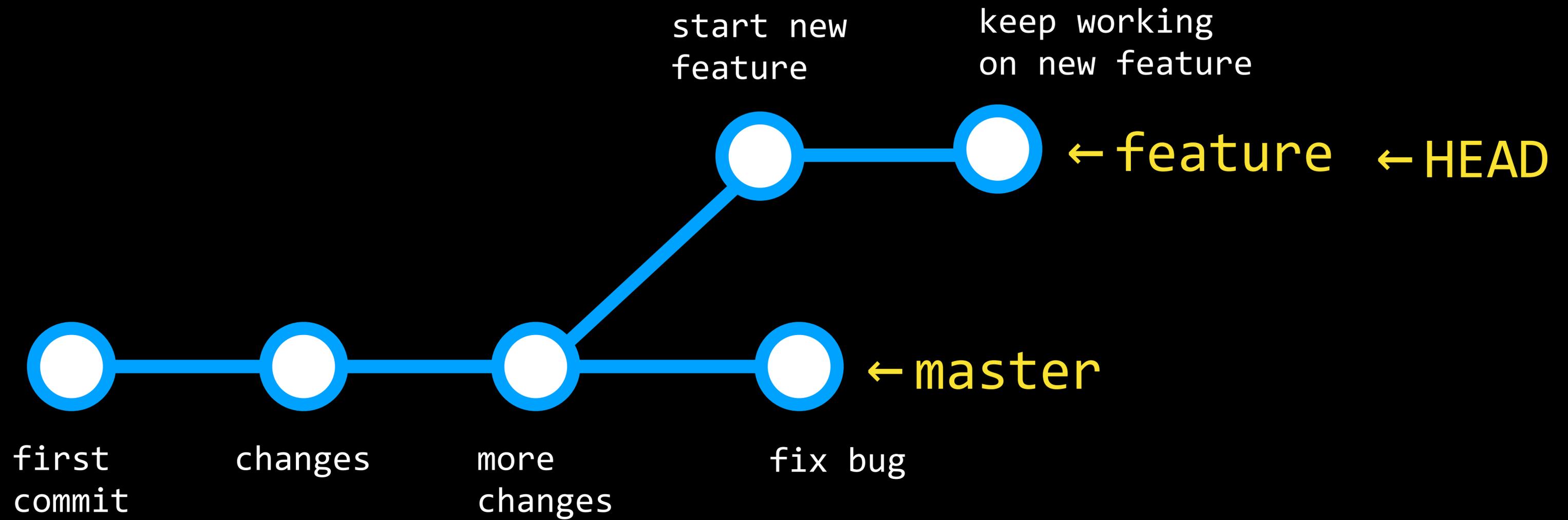
more  
changes

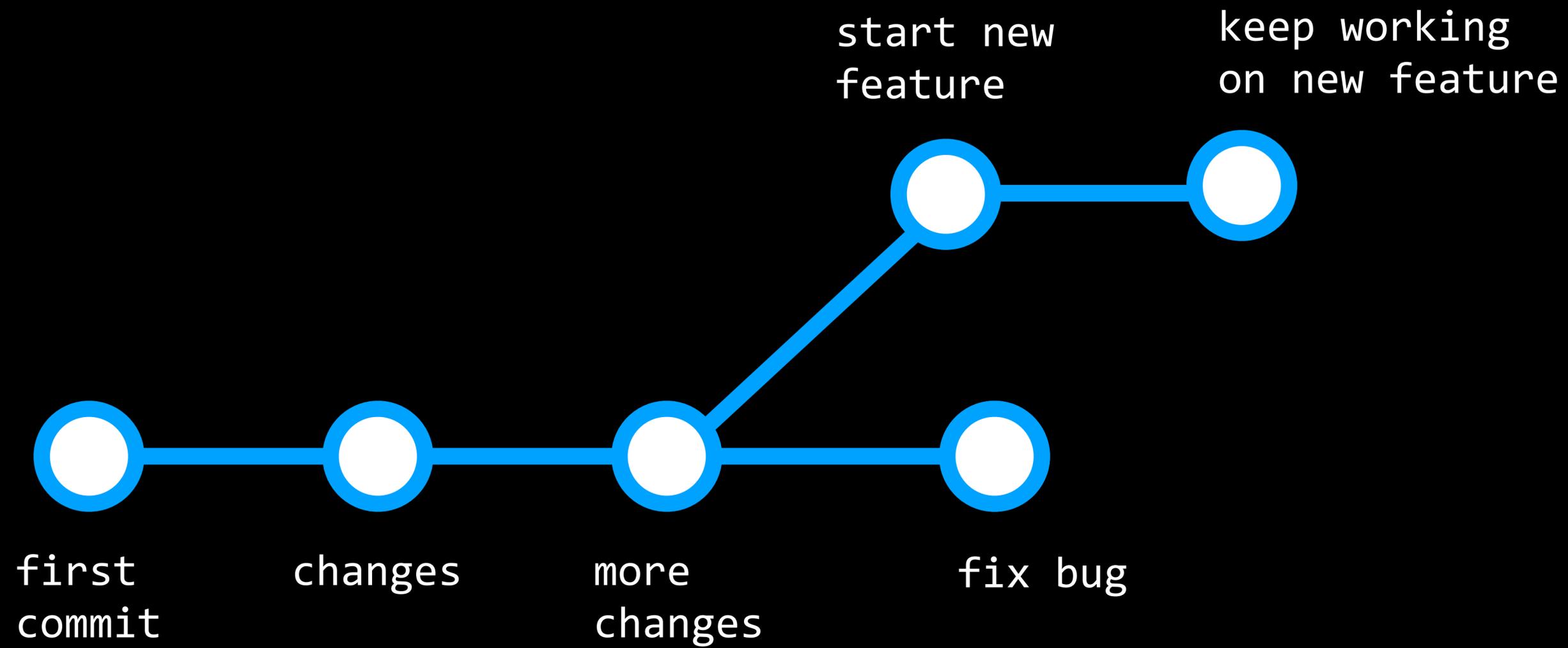


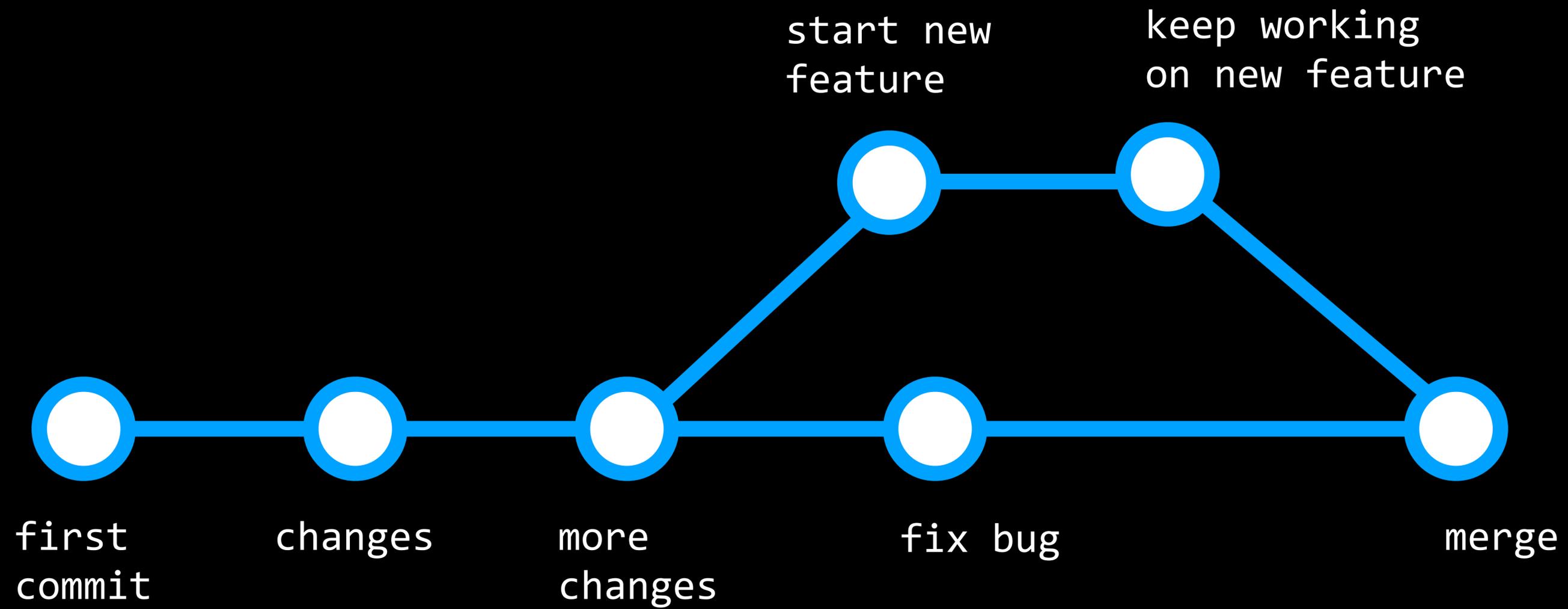












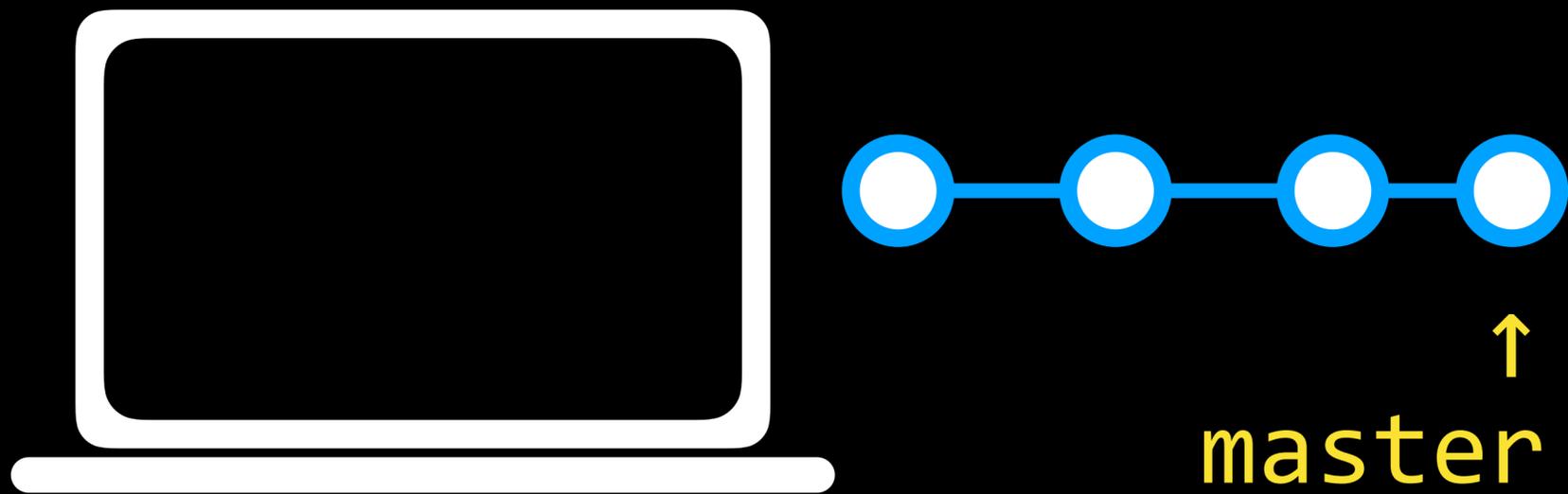
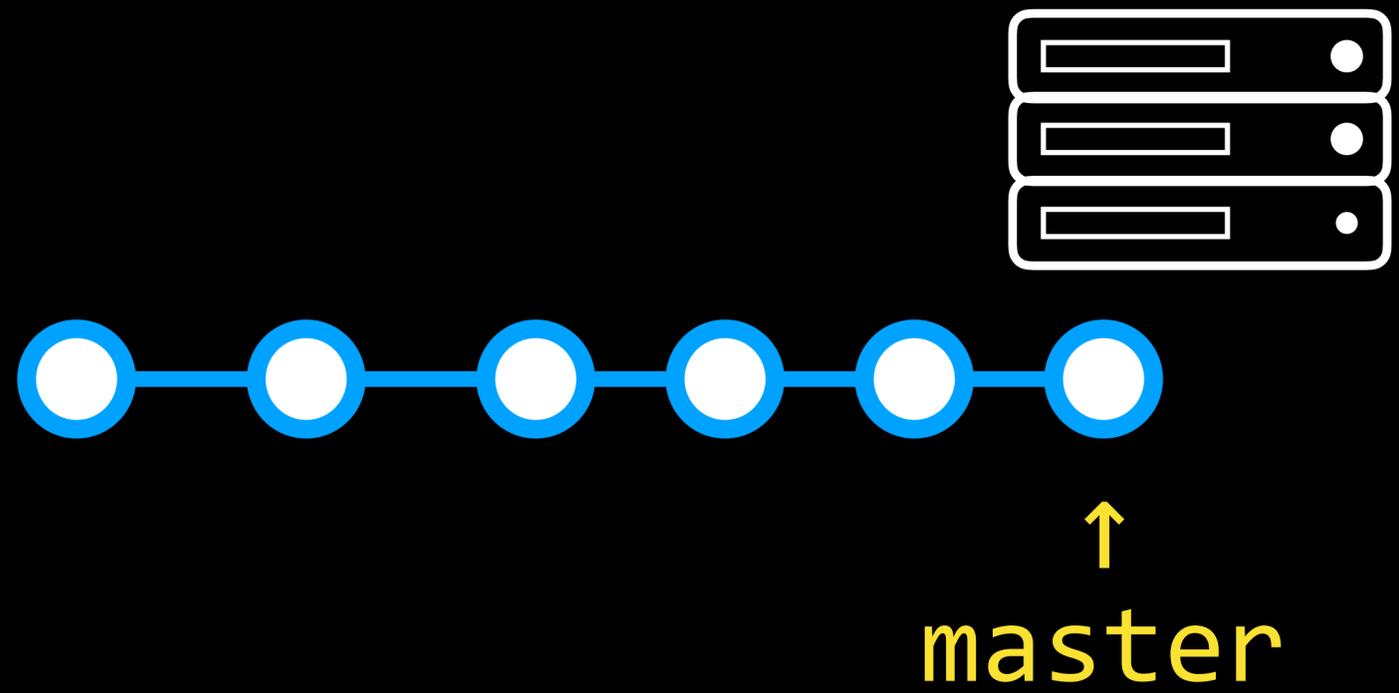
# Branching

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

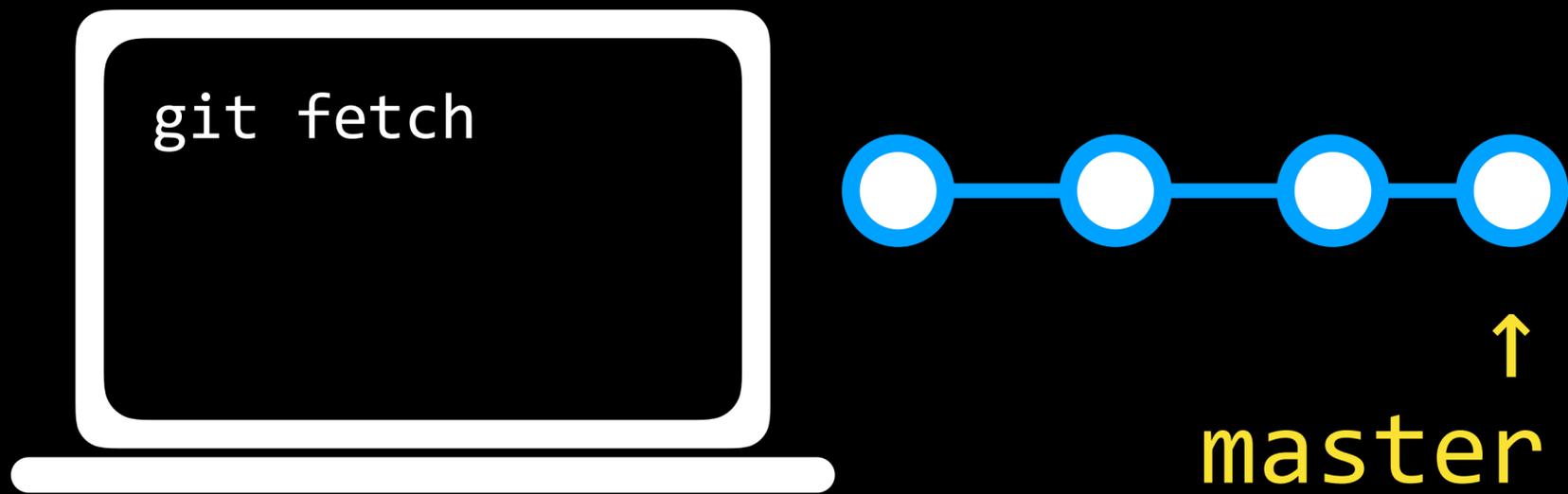
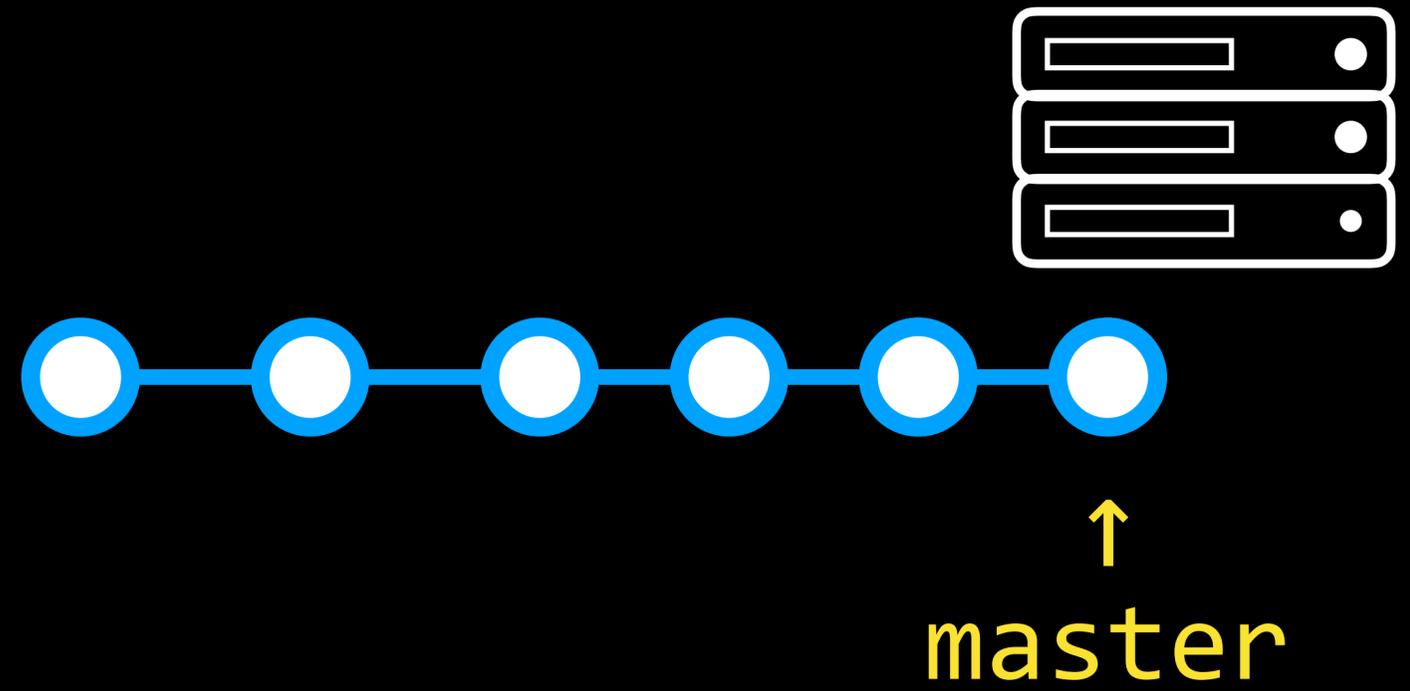
**Remotes**

# Remotes

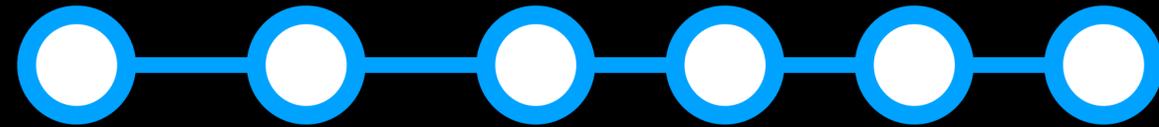
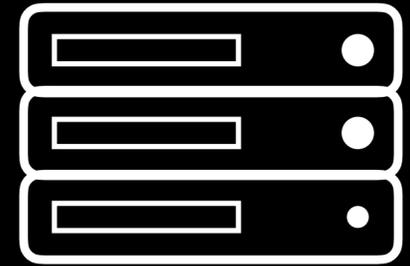
# Remotes



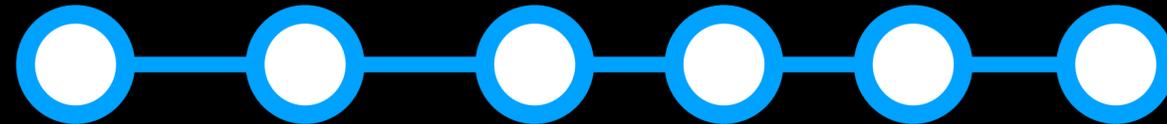
# Remotes



# Remotes



master

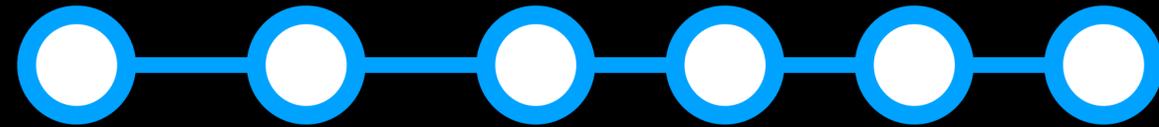
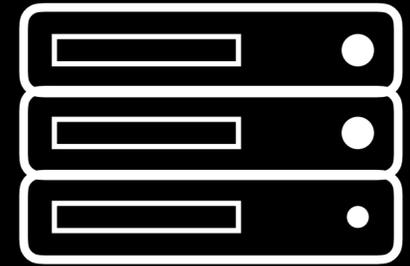


master

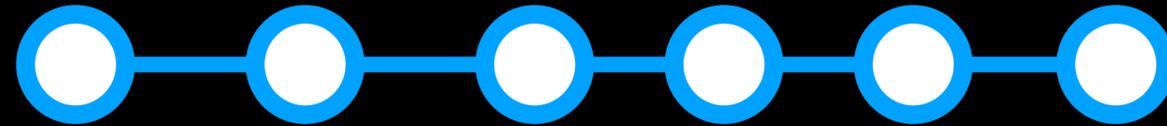


origin/master

# Remotes



↑  
master

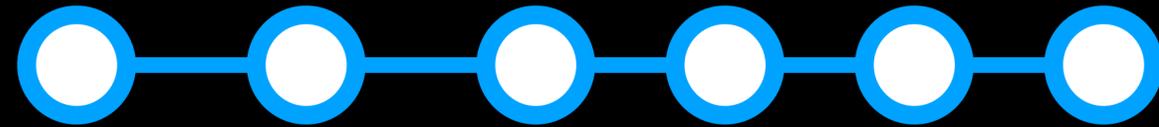
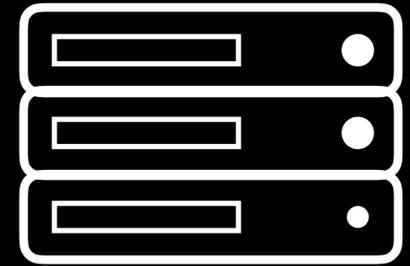


↑  
master

↖  
origin/master

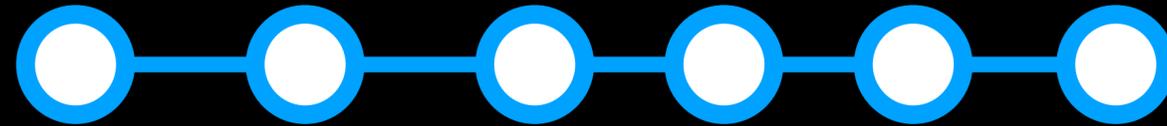
```
git fetch  
git merge  
origin/master
```

# Remotes



master

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

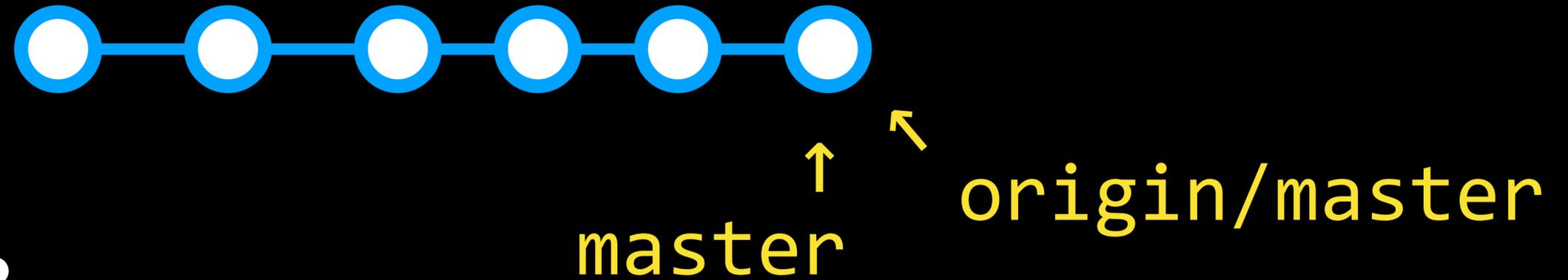
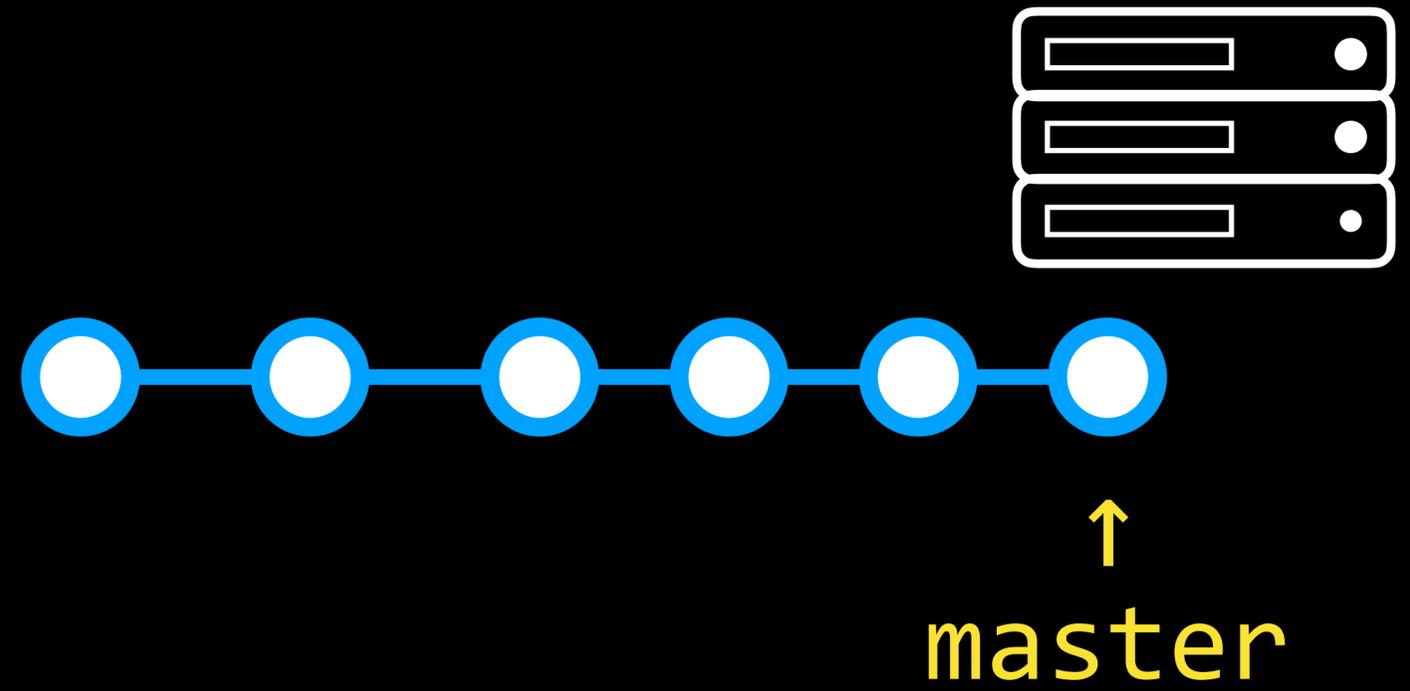


master



origin/master

# Remotes



**Forks**

# Pull Requests

# Web Programming

with Python and JavaScript