

# Git

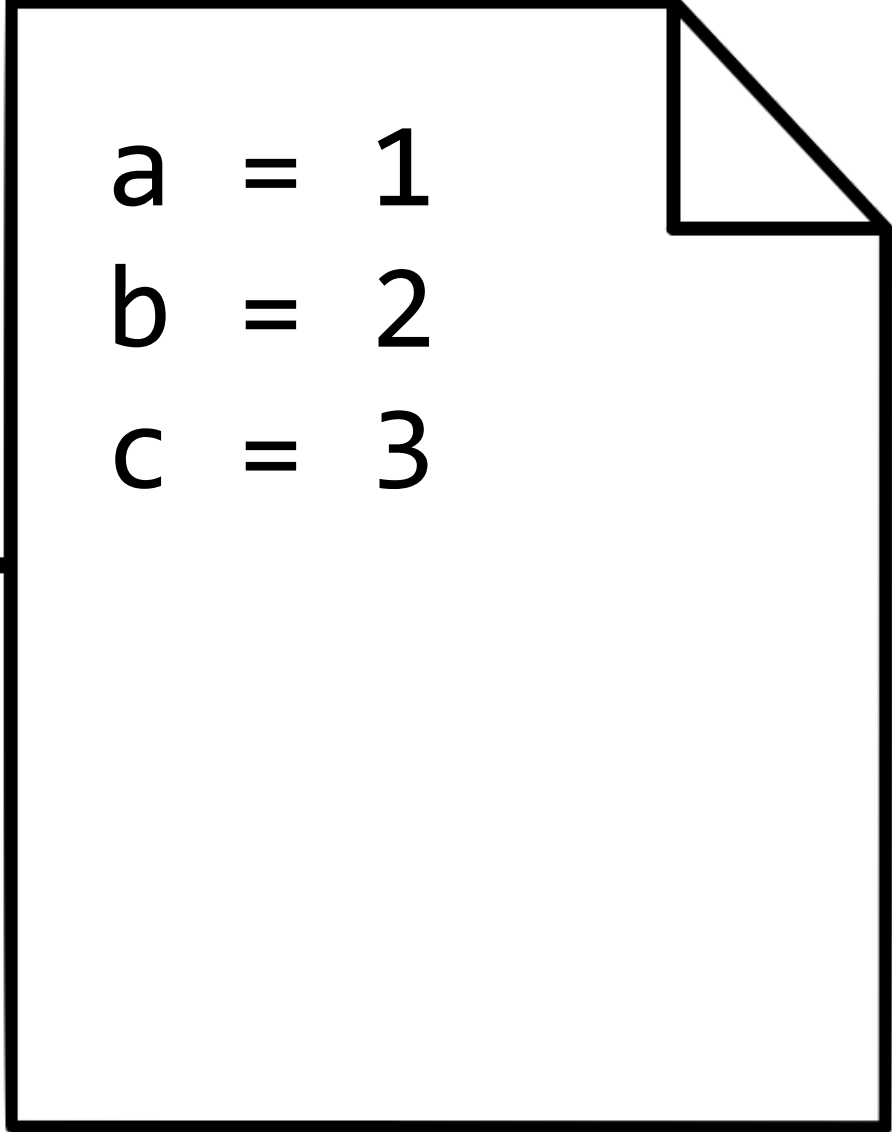
CSCI P-14300  
June 25, 2019

Brian Yu  
[brian@cs.harvard.edu](mailto:brian@cs.harvard.edu)

**Git**

**Keep track of changes to code.**

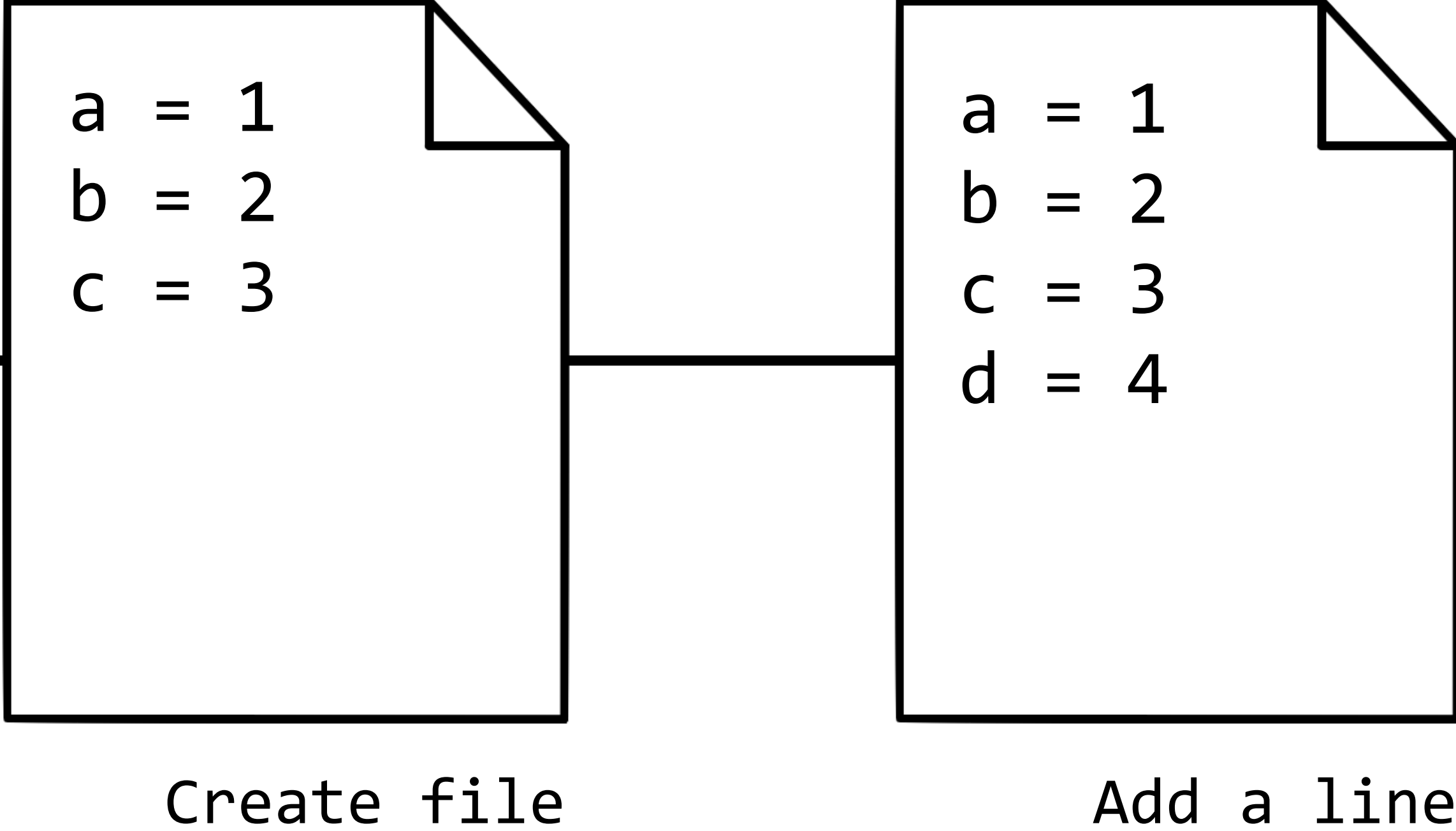
# Keep track of changes to code.



```
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, each represented as a document icon with a folded top-right corner. They are connected by horizontal lines from left to right. The first document contains three lines of code: 'a = 1', 'b = 2', and 'c = 3'. The second document contains four lines: 'a = 1', 'b = 2', 'c = 3', and 'd = 4'. The third document contains three lines: 'a = 1', 'c = 3', and 'd = 4'. Below each document is a label describing the change: 'Create file', 'Add a line', and 'Remove a line'.

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

Create file

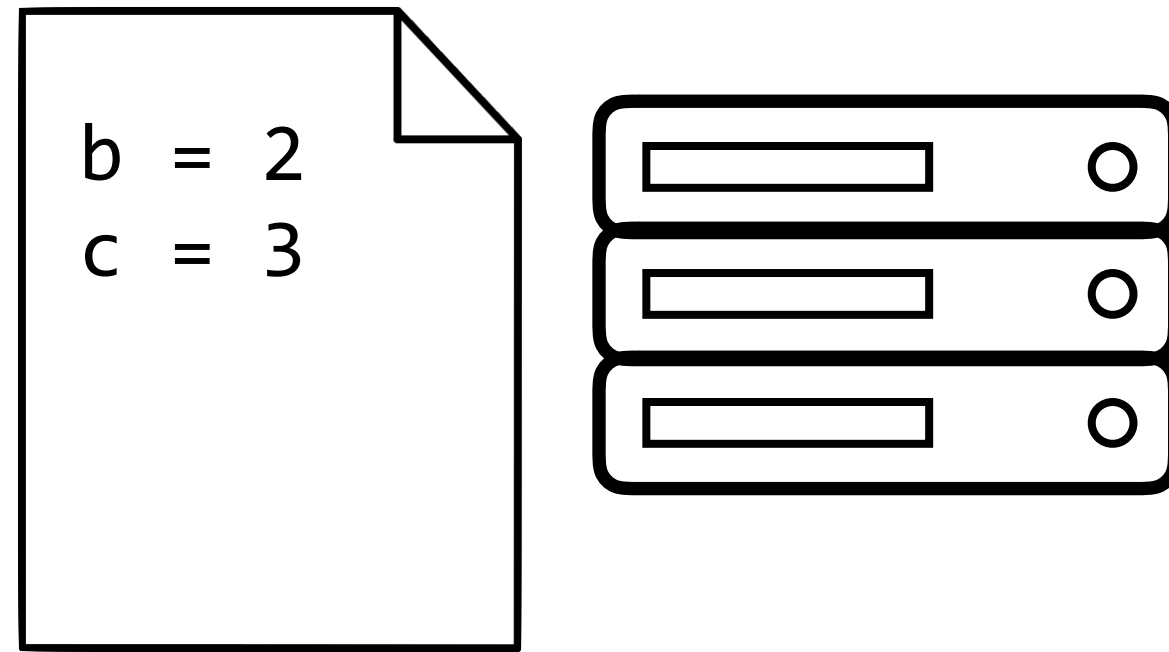
```
a = 1  
b = 2  
c = 3  
d = 4
```

Add a line

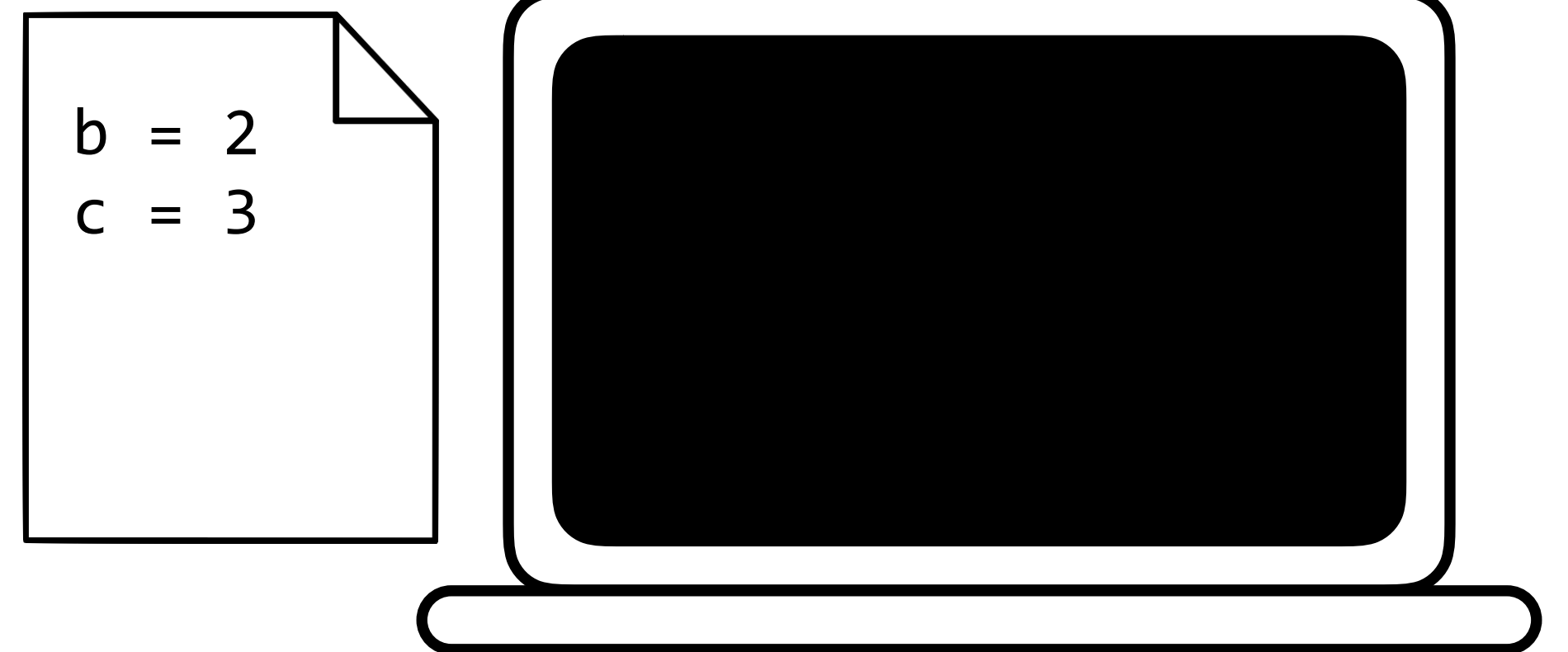
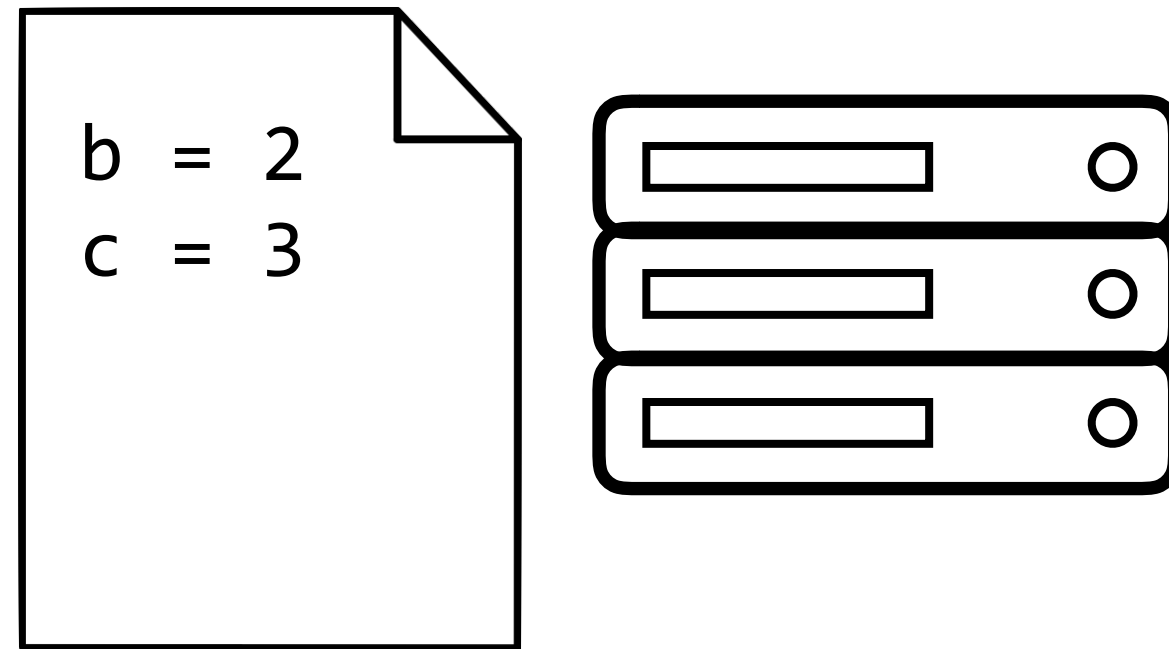
```
a = 1  
c = 3  
d = 4
```

Remove a line

# Synchronize code between collaborators.

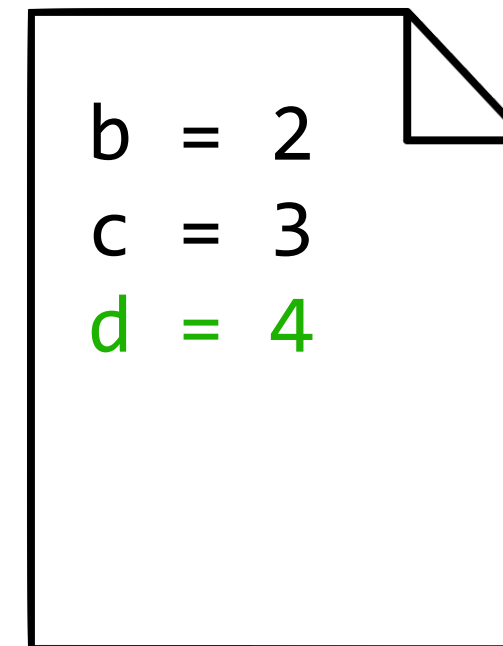
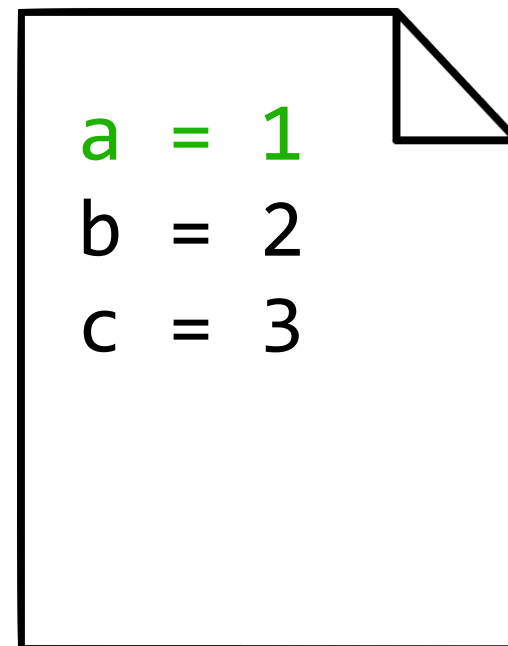
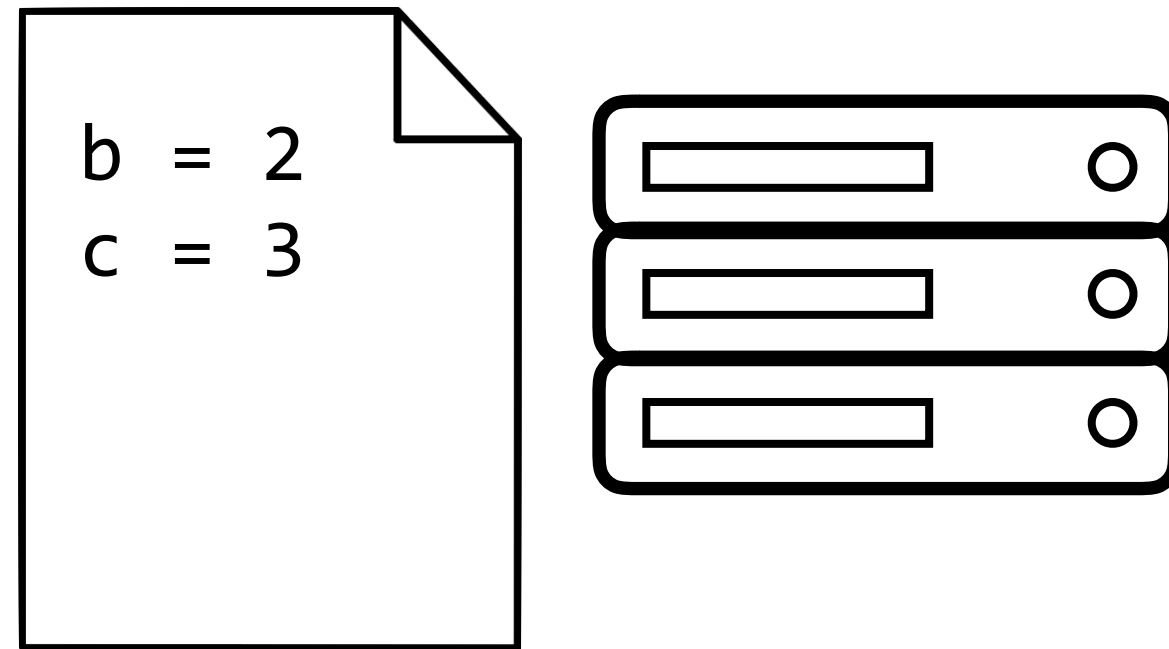


# Synchronize code between collaborators.

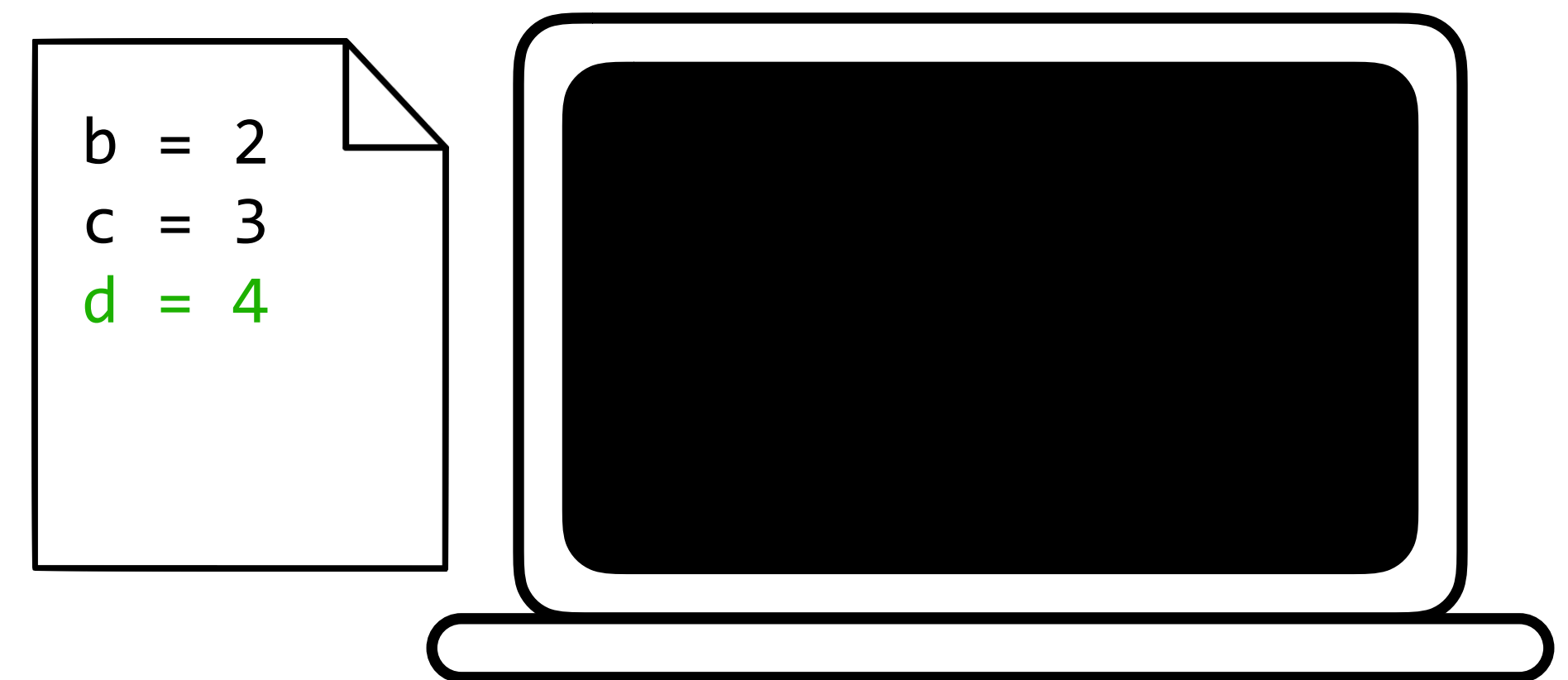
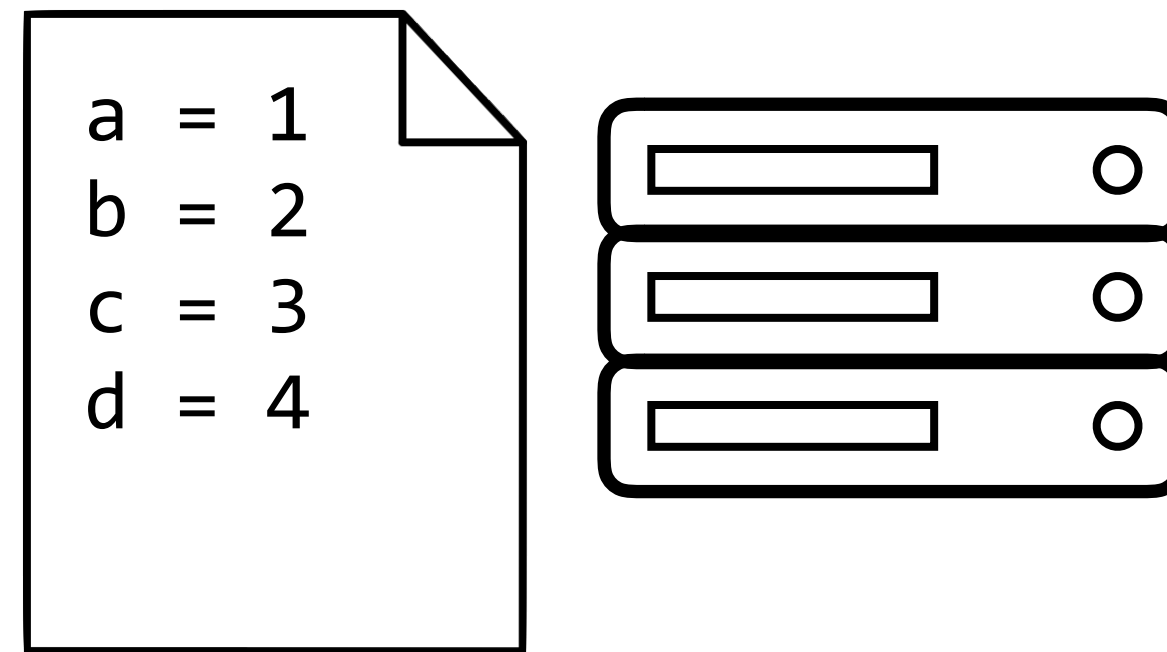




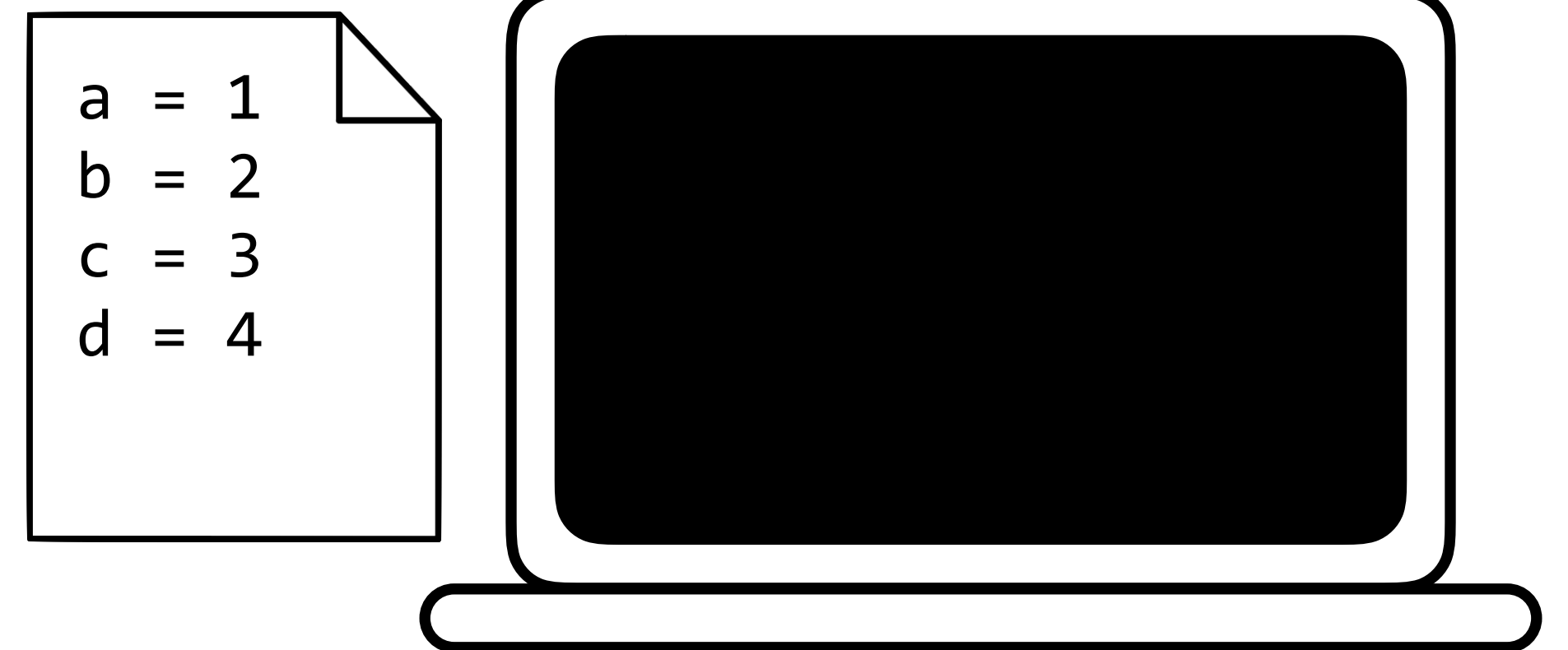
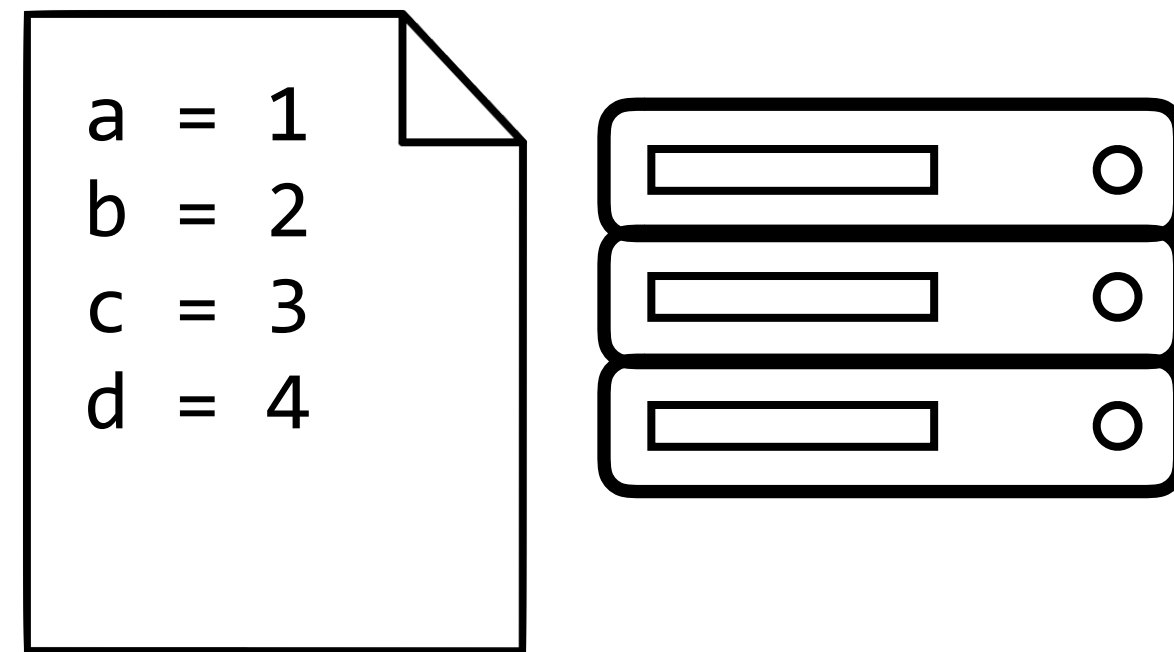
# Synchronize code between collaborators.



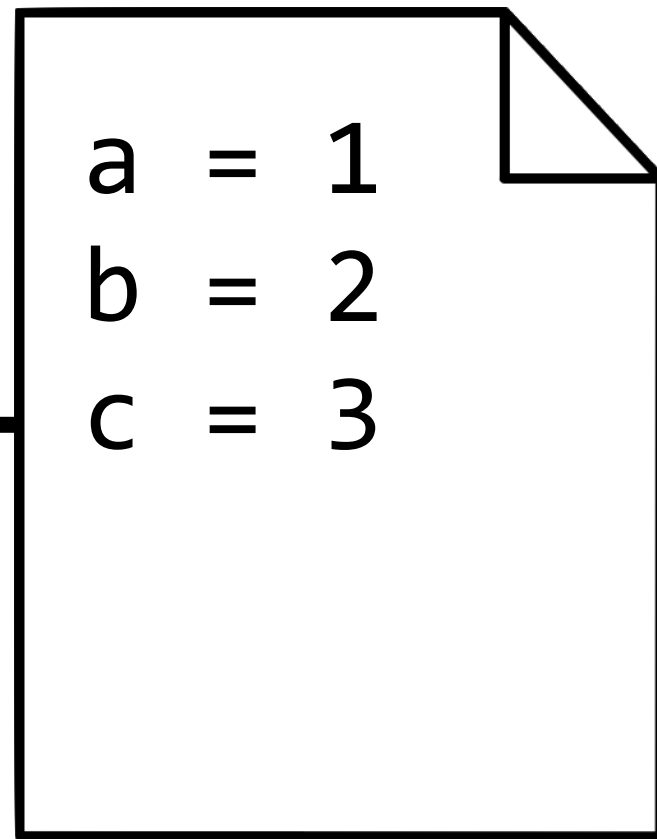
# Synchronize code between collaborators.



# Synchronize code between collaborators.



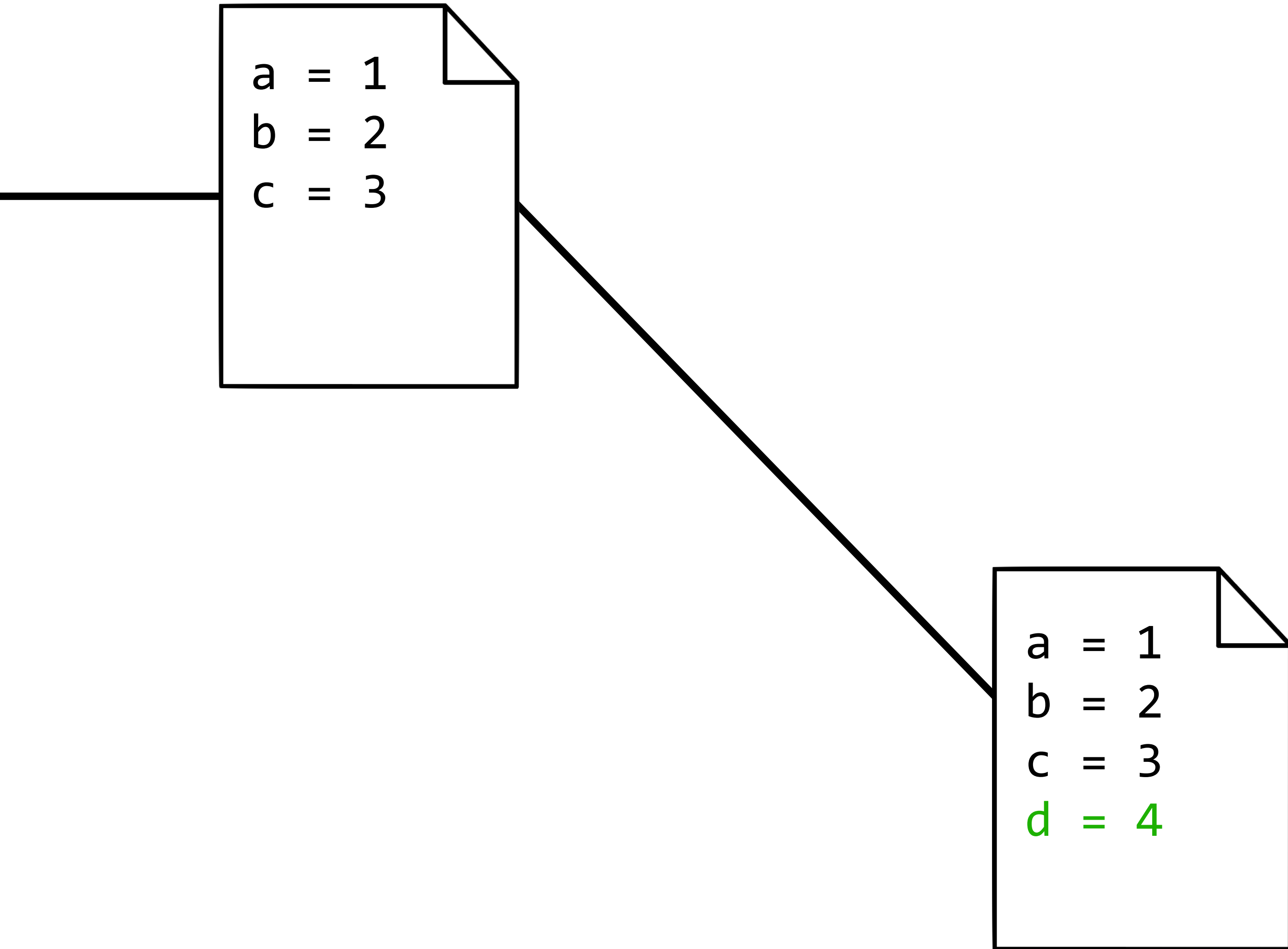
# Experiment with new changes to code.



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

# Experiment with new changes to code.

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



The diagram illustrates a code transformation. A horizontal line enters from the left and connects to a code block. A diagonal line then connects this block to a second code block positioned lower and to the right. Both code blocks are enclosed in rectangular boxes with a folded top-right corner, resembling document icons. The first block contains three lines of code: 'a = 1', 'b = 2', and 'c = 3'. The second block contains the same three lines, but with a fourth line, 'd = 4', added at the bottom. The text 'd = 4' is highlighted in green.

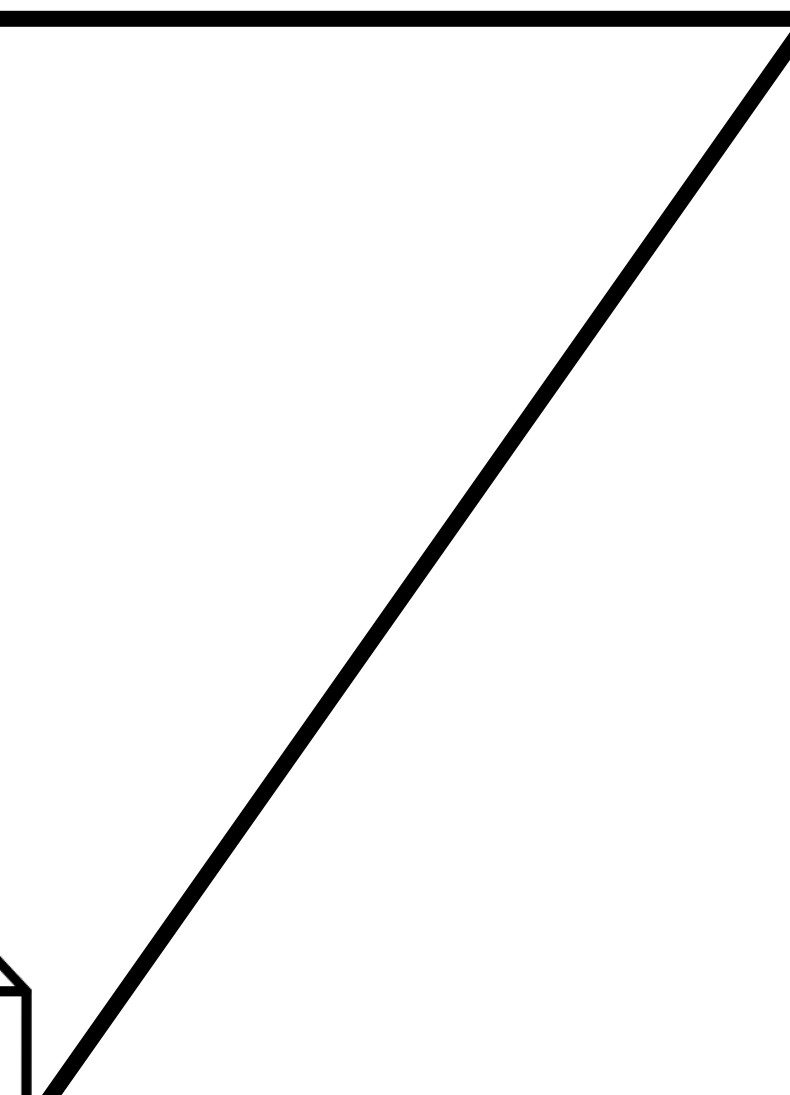
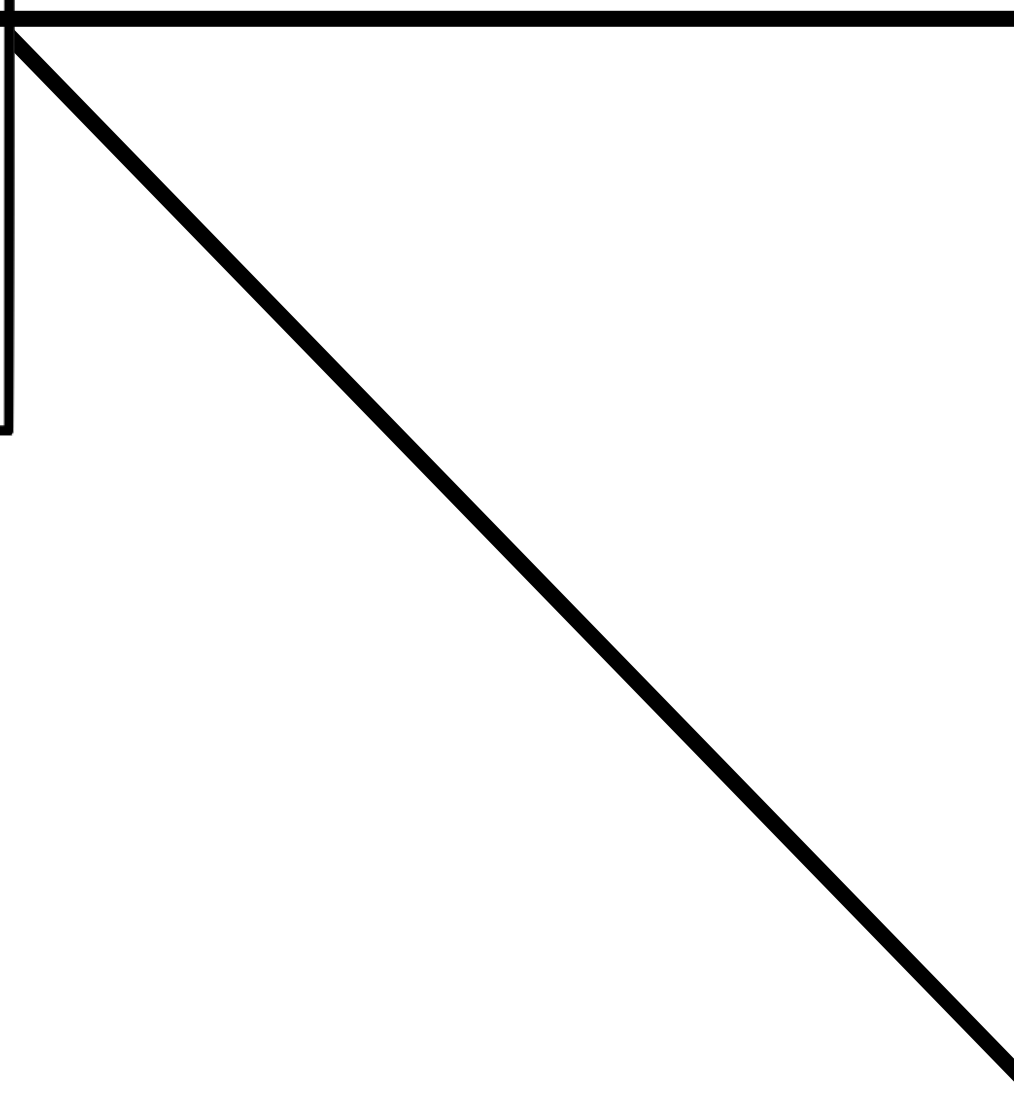
```
a = 1  
b = 2  
c = 3  
d = 4
```

# Experiment with new changes to code.

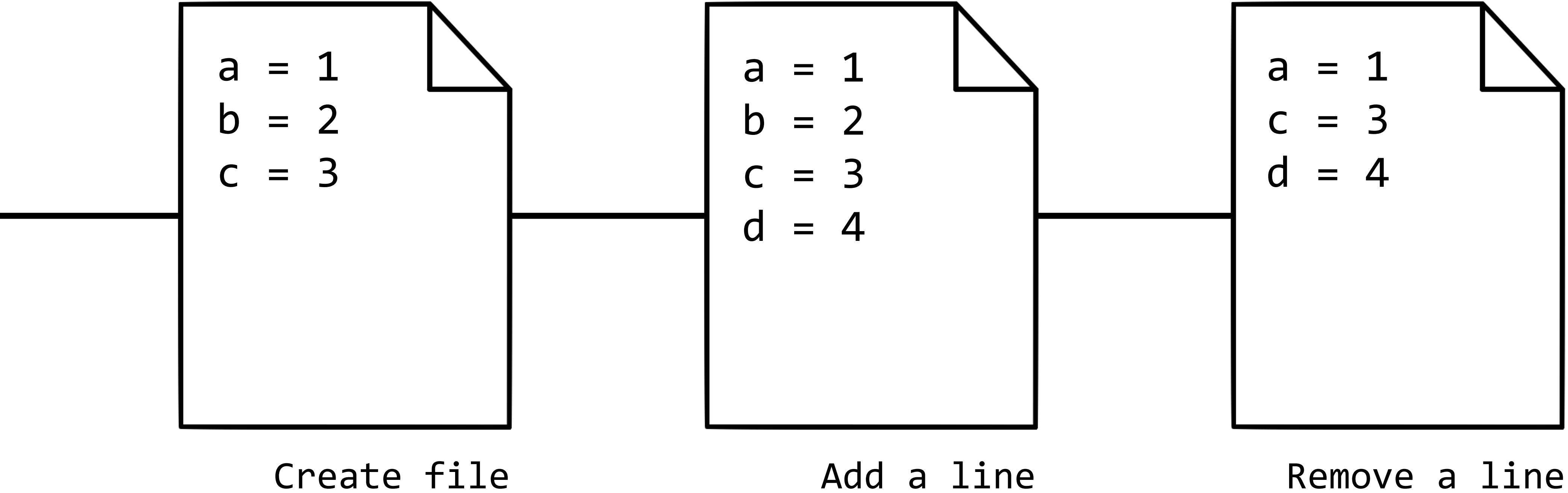
```
a = 1  
b = 2  
c = 3
```

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

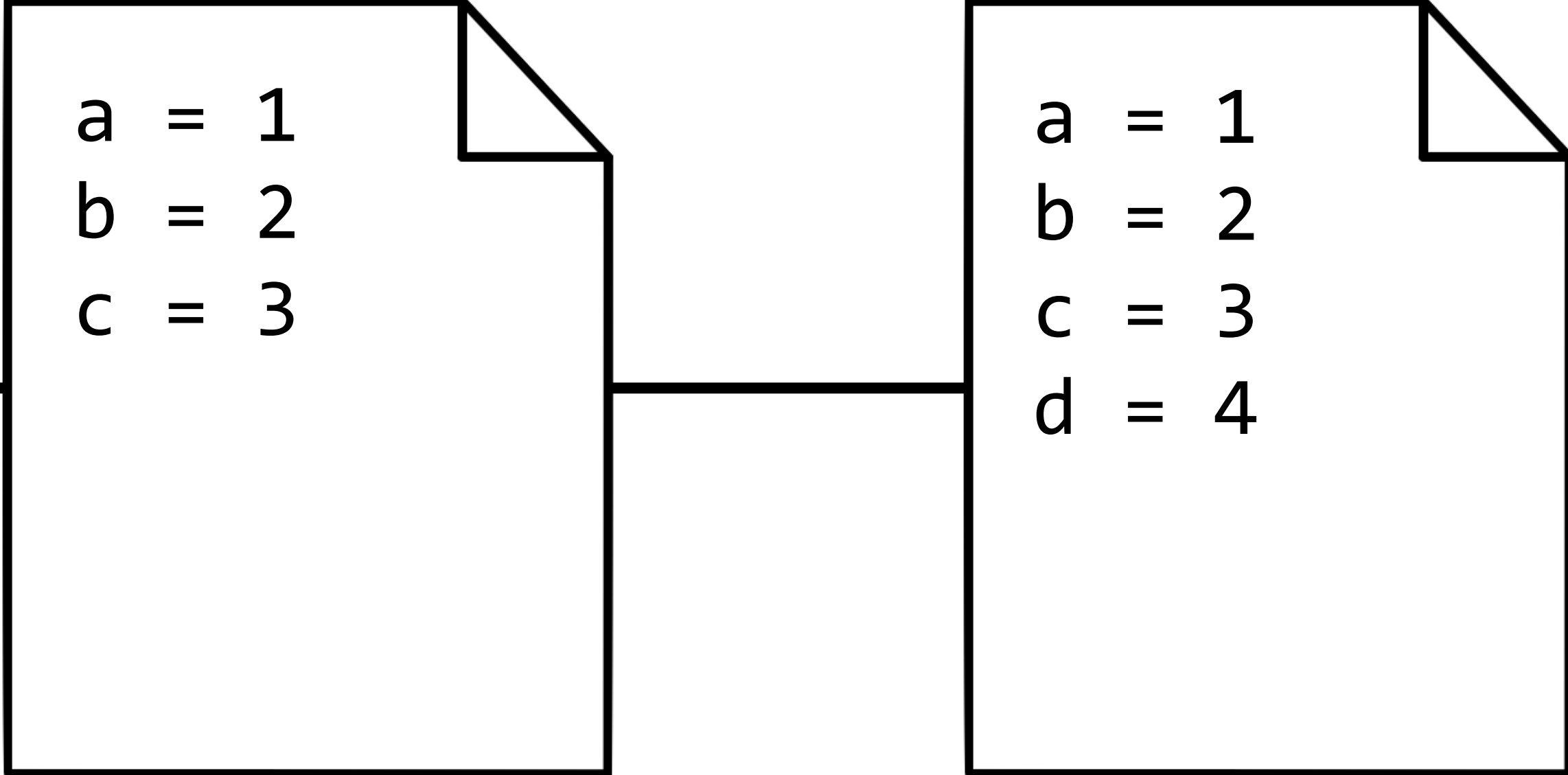
```
a = 1  
b = 2  
c = 3  
d = 4
```



# Revert back to old versions of code.



# Revert back to old versions of code.



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

Create file

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

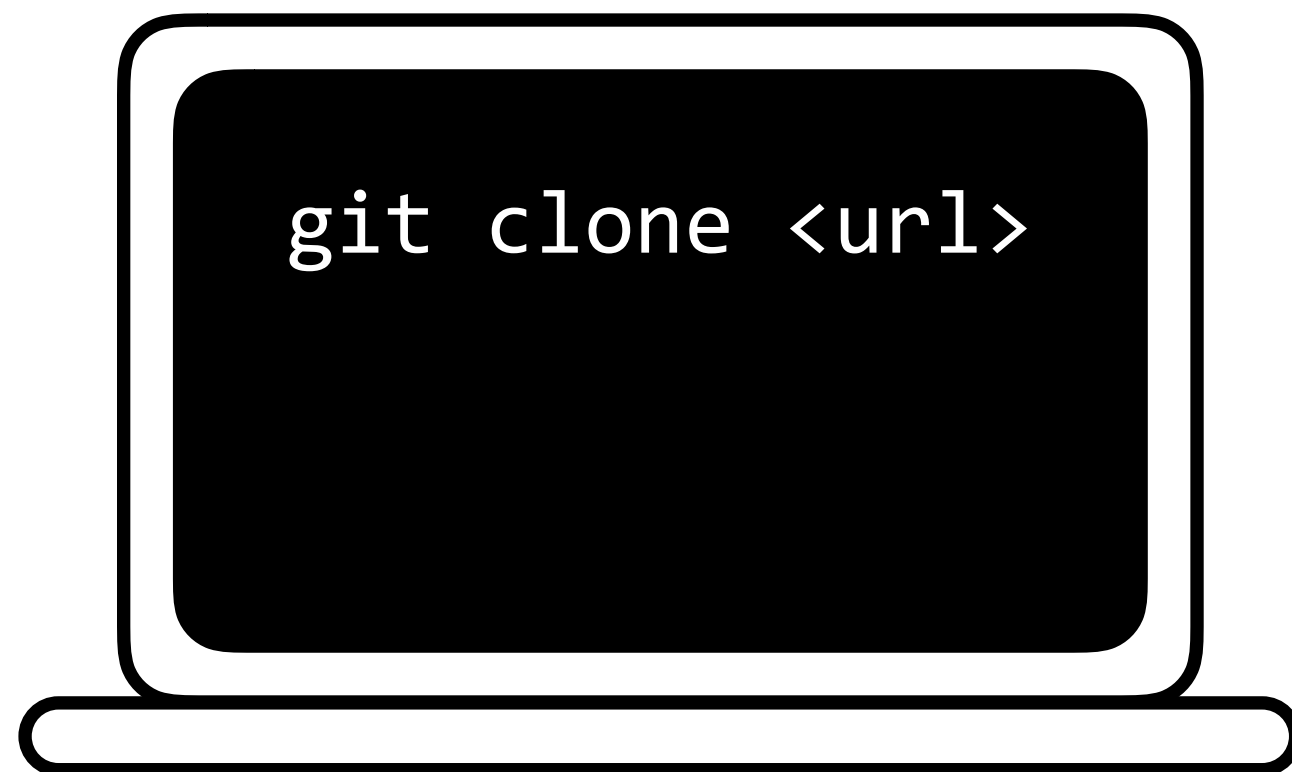
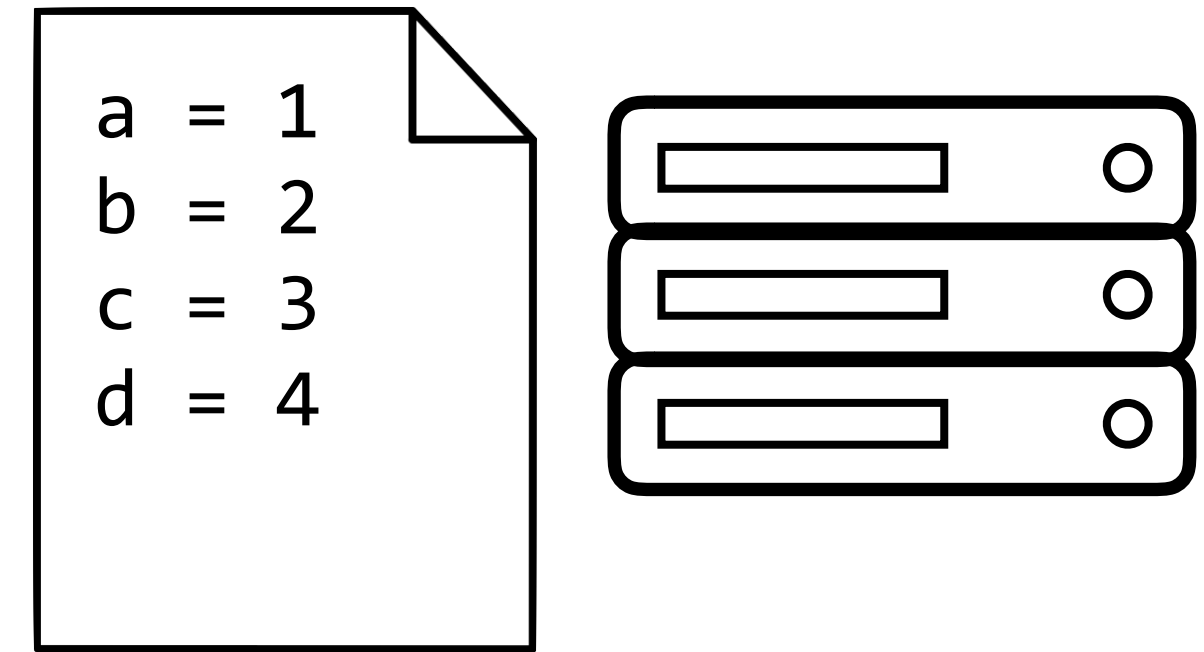
Add a line



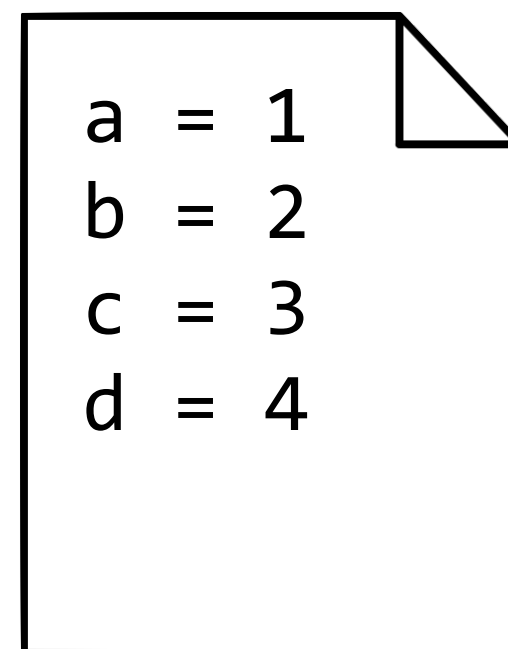
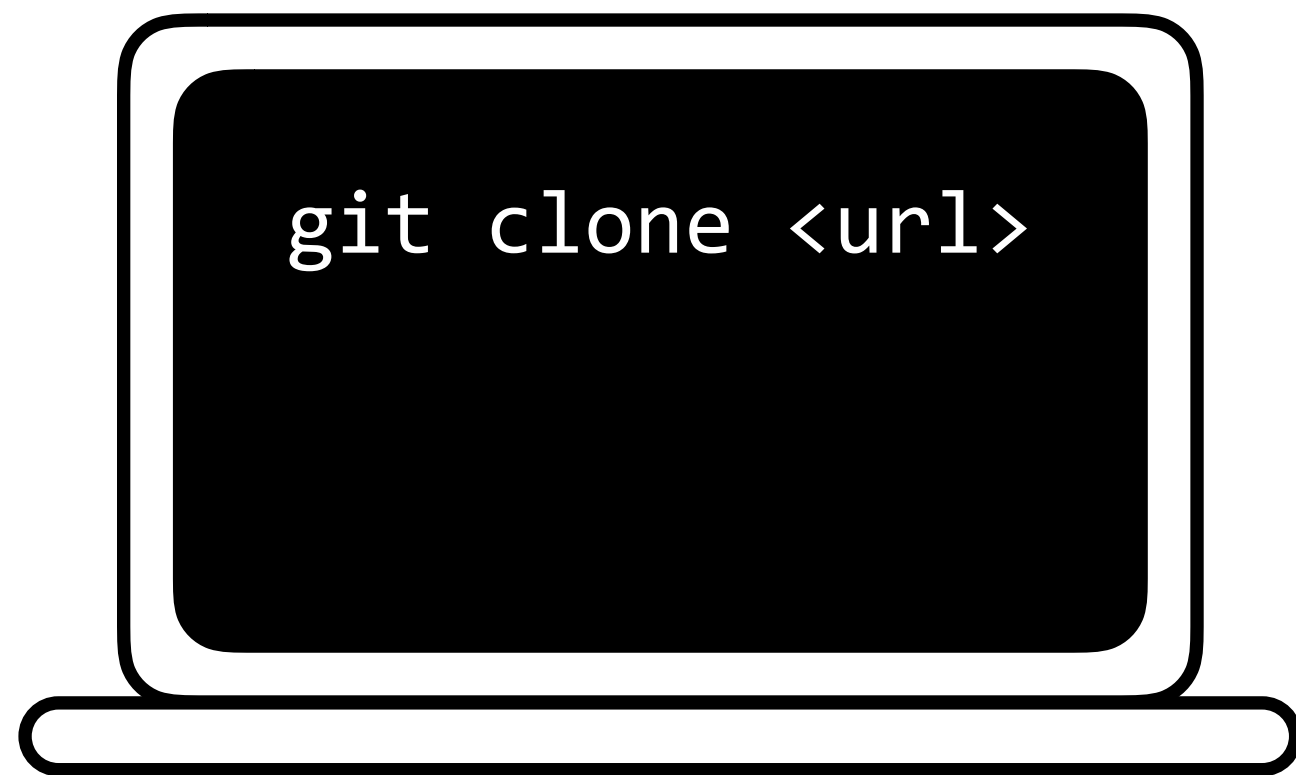
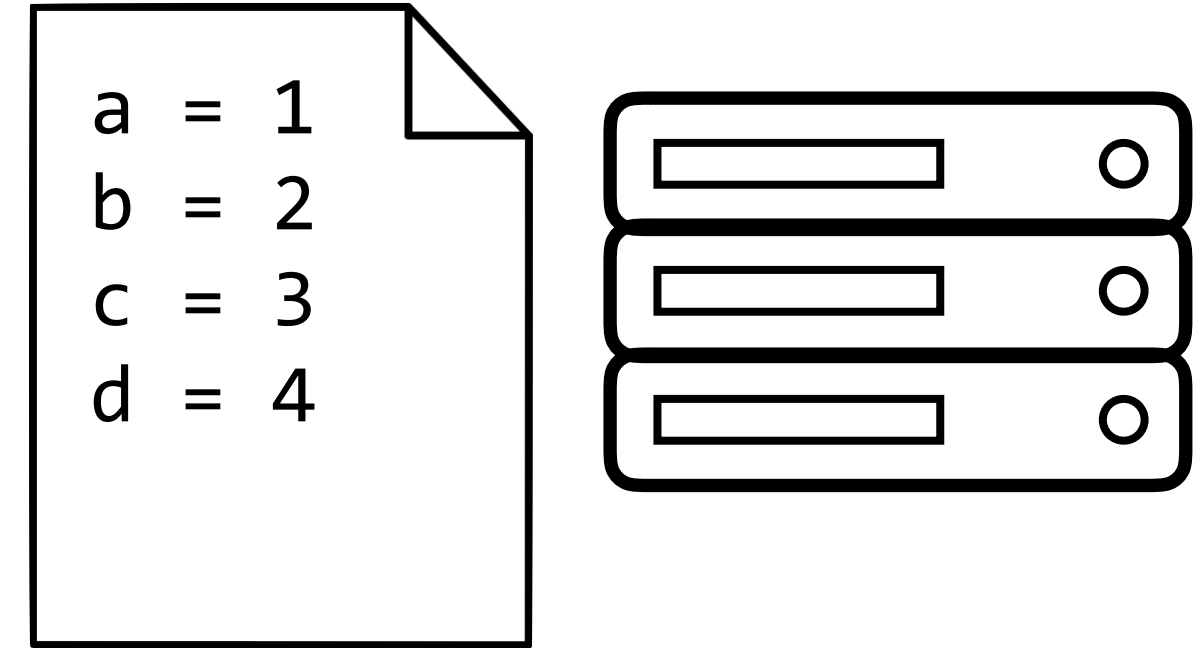
**GitHub**

```
git clone
```

# git clone <url>

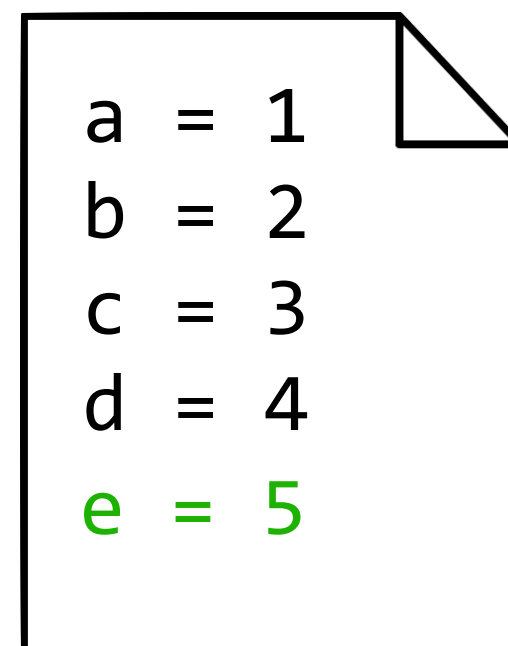
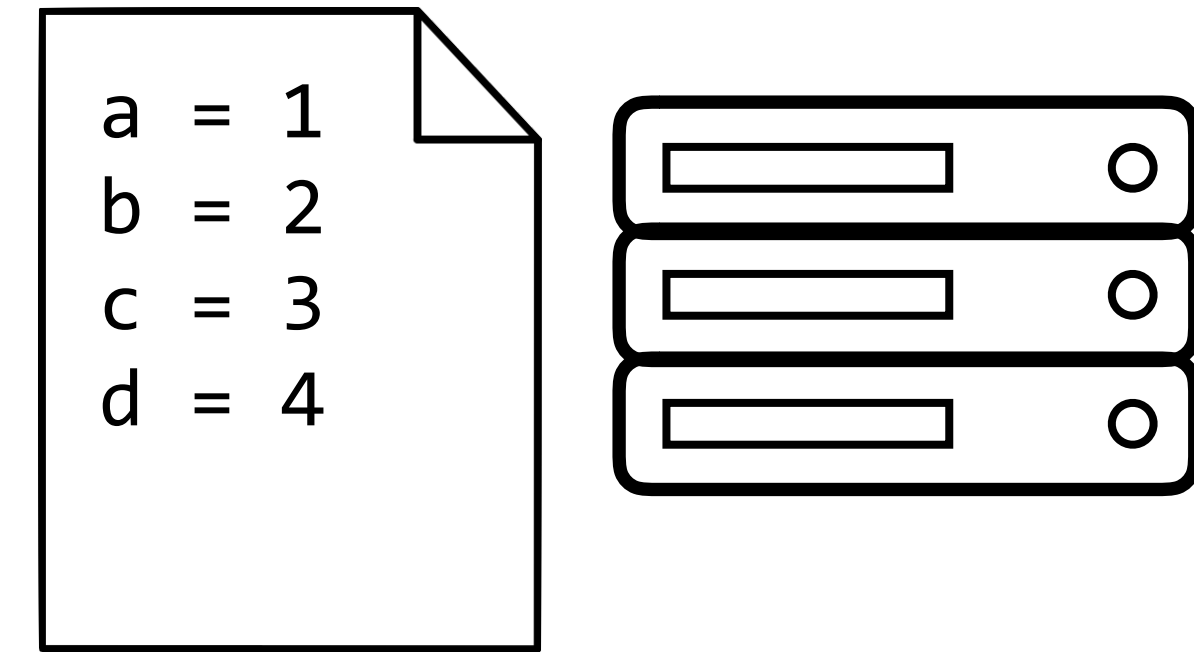


# git clone <url>



```
git add
```

# git add <filename>

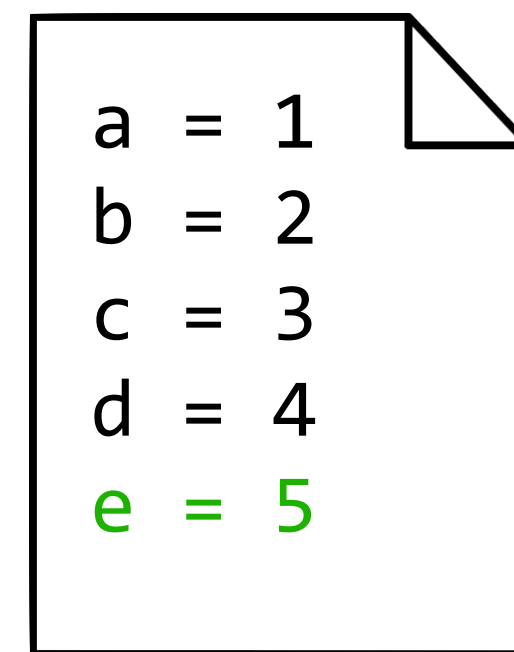
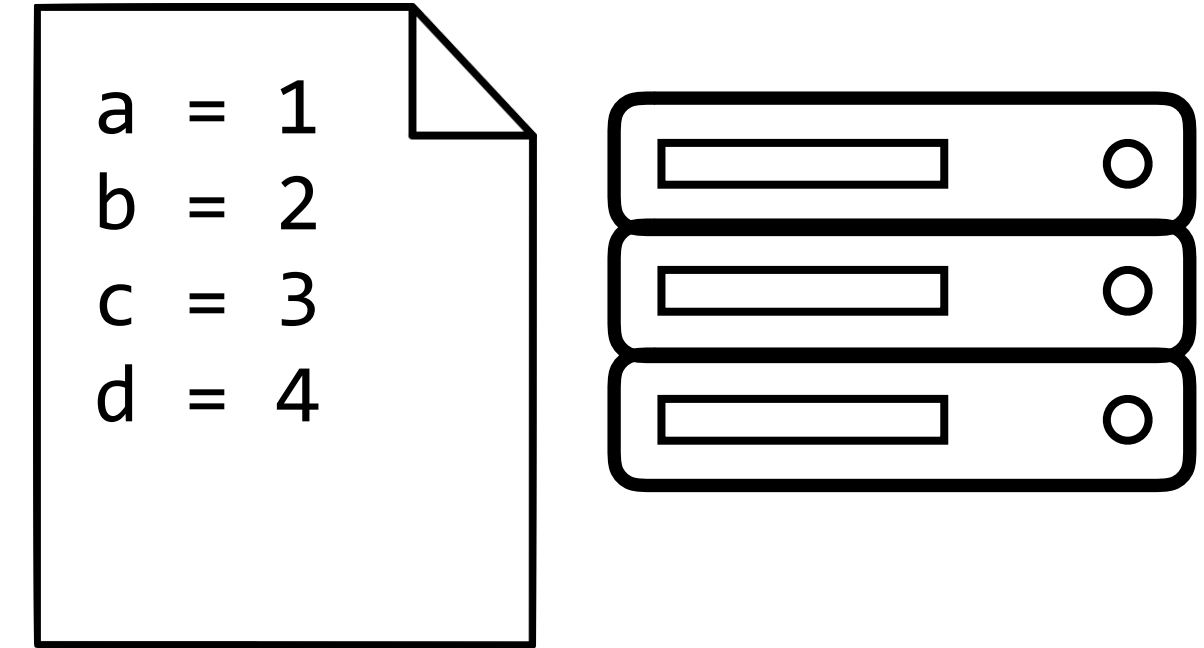


Changes to be committed:

modified: foo.py

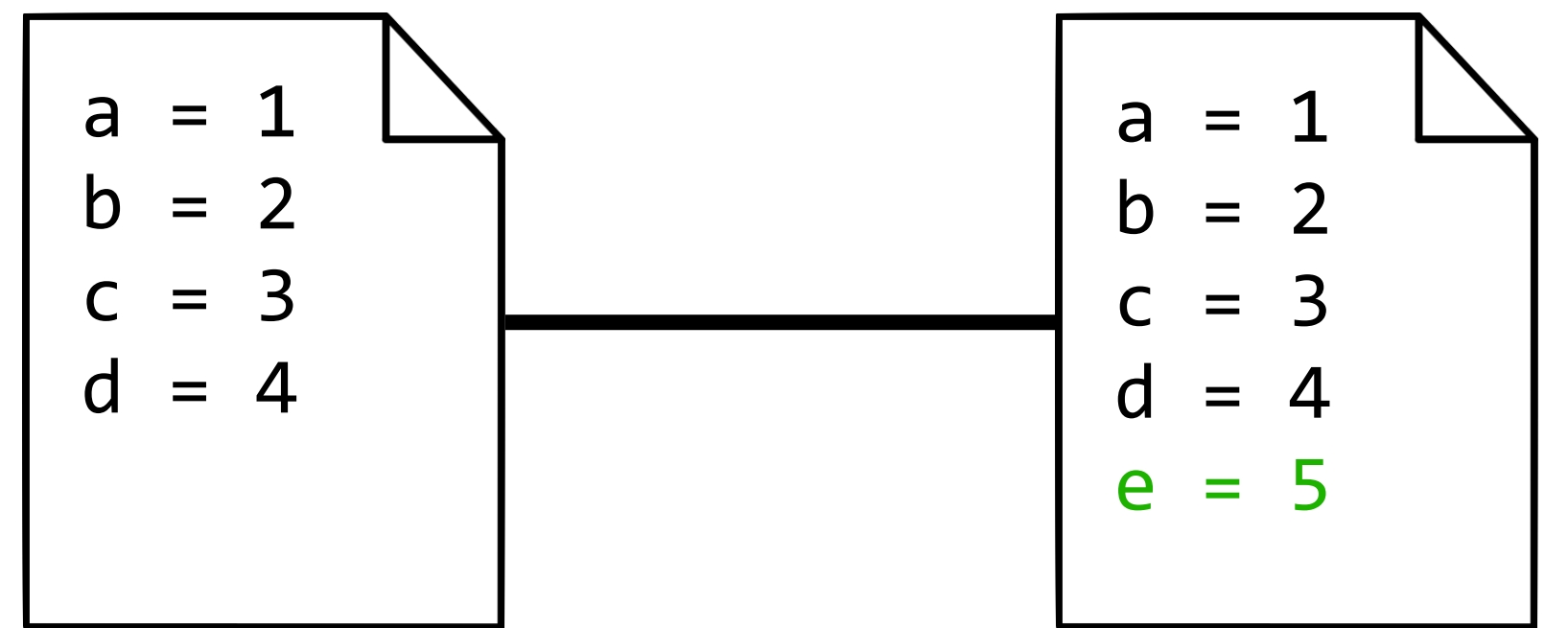
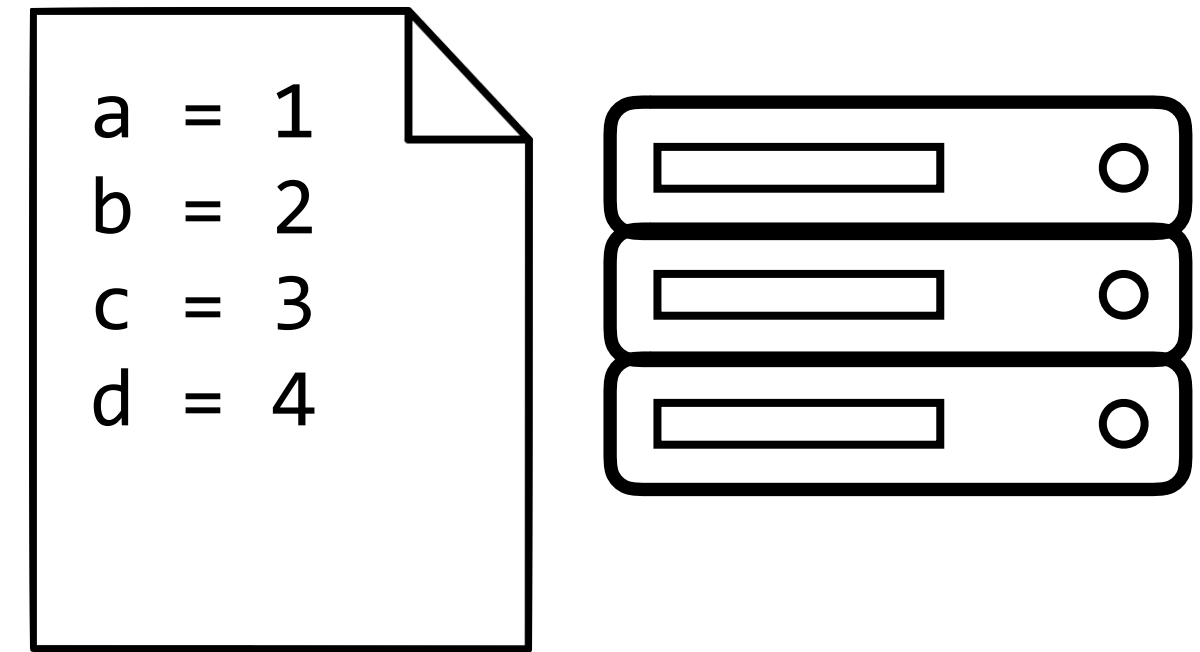
```
git commit
```

# git commit -m "message"





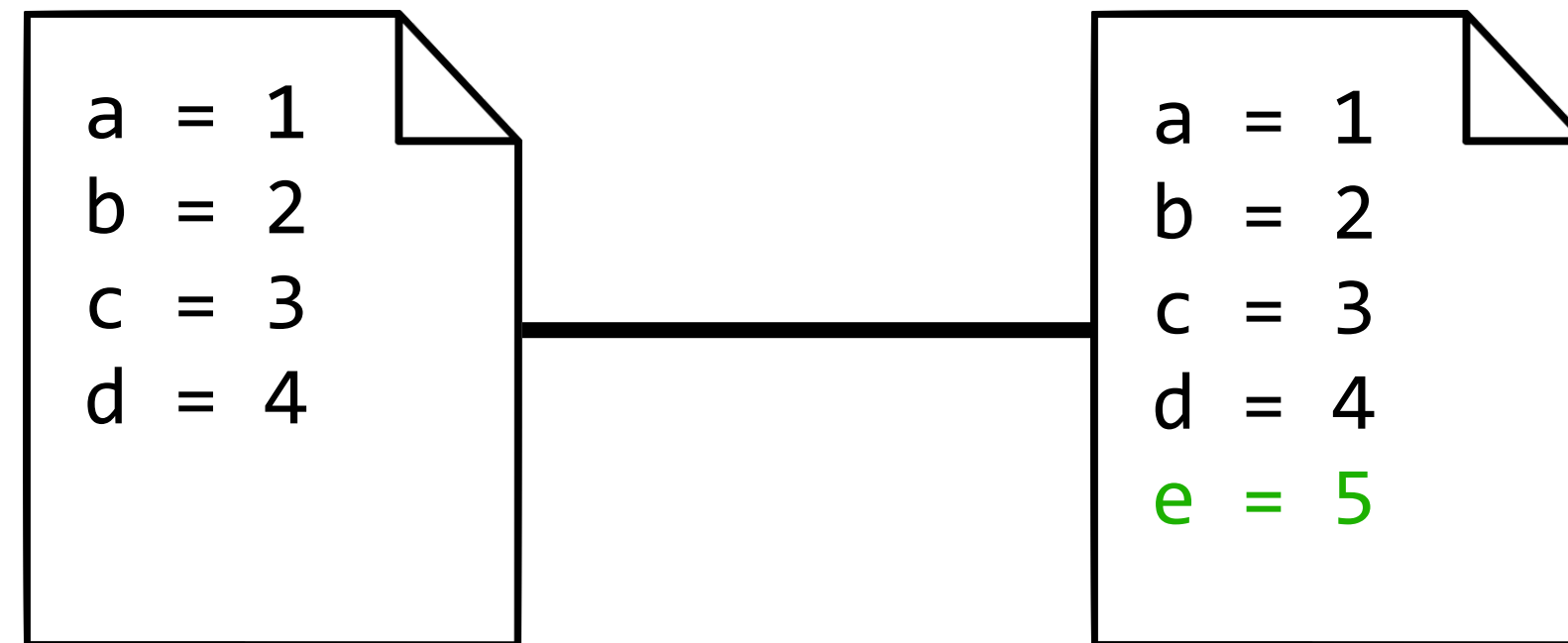
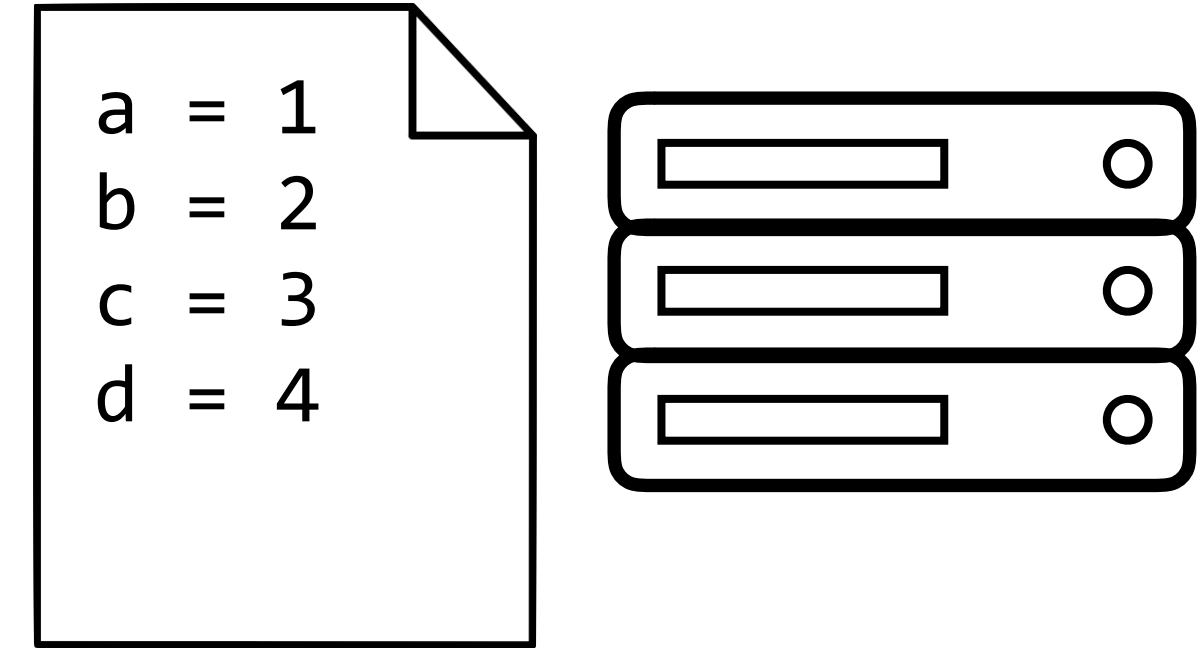
# git commit -m "message"



Add line

`git status`

# git status



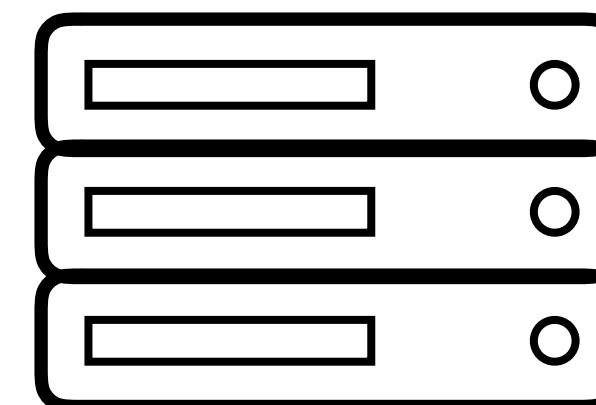
Add line

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
```

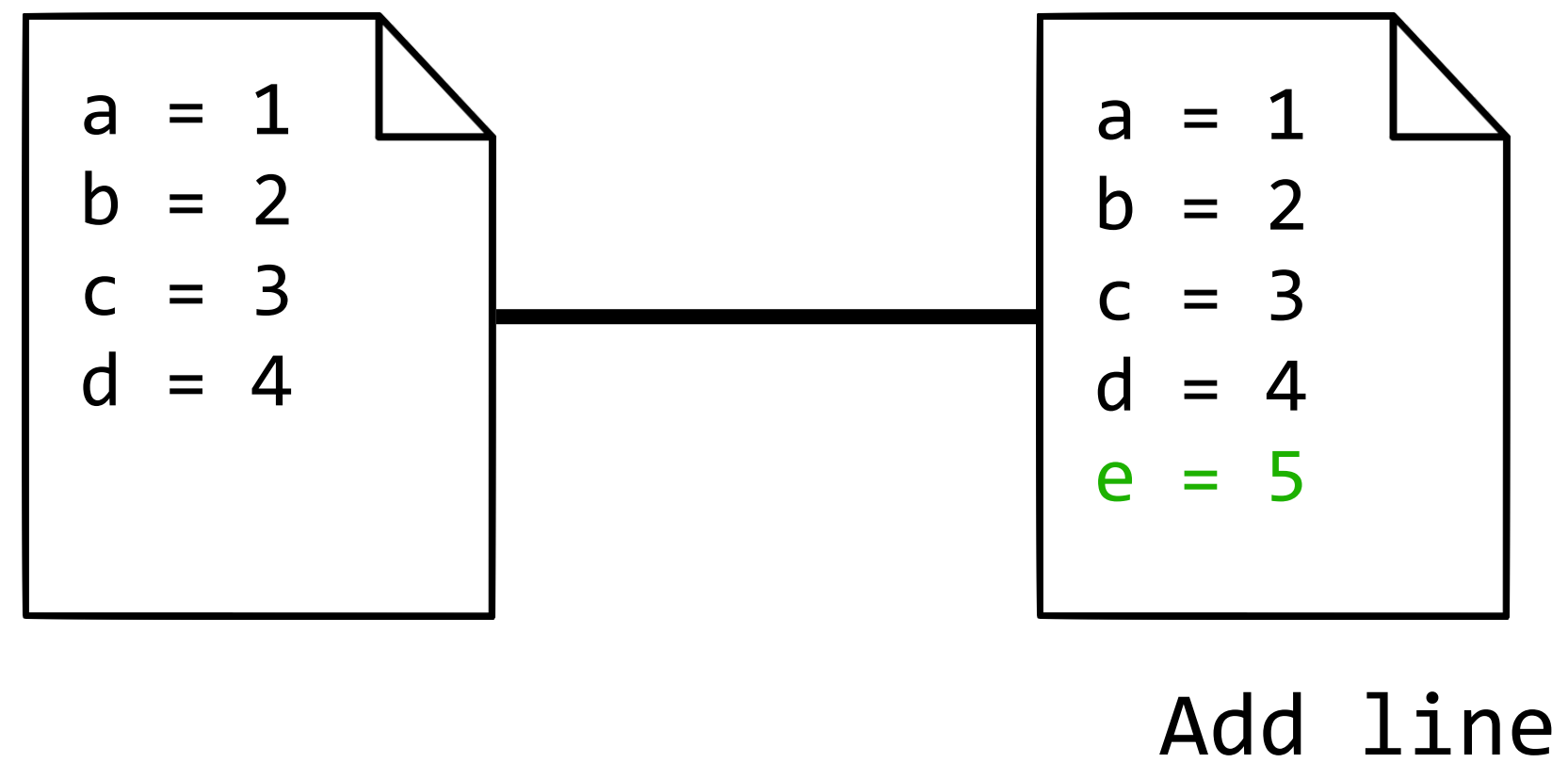
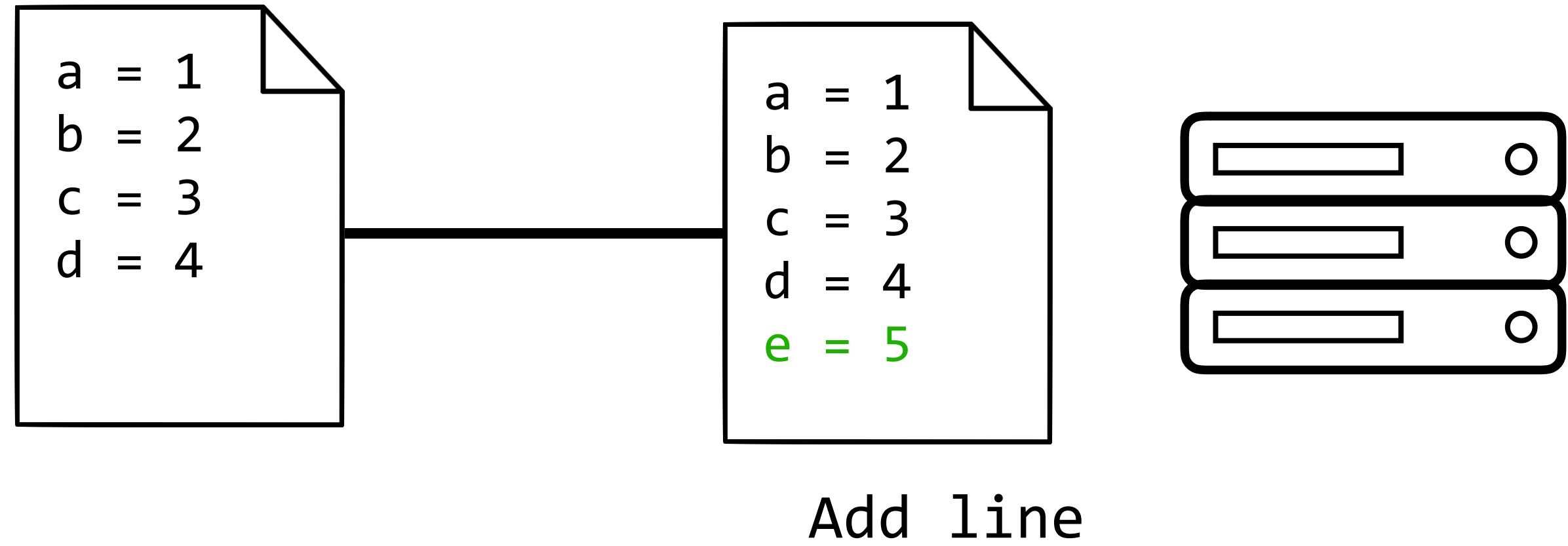


```
a = 1  
b = 2  
c = 3  
d = 4
```

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

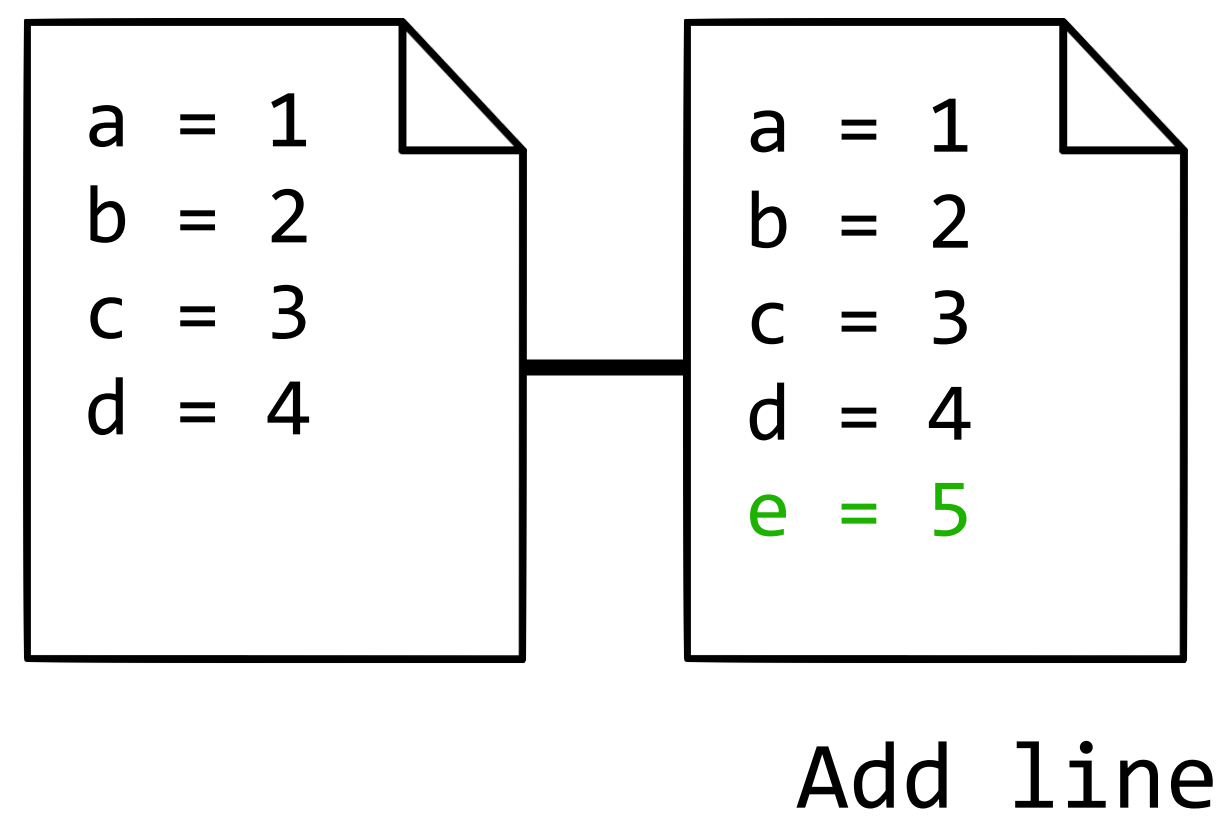
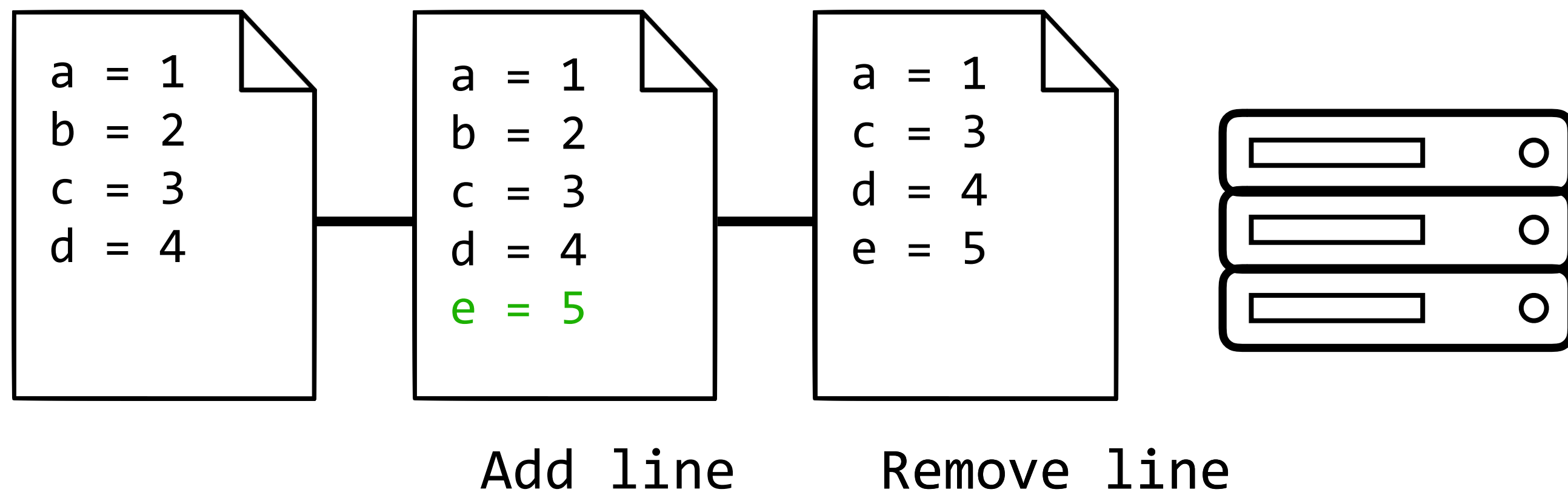
Add line

# git push



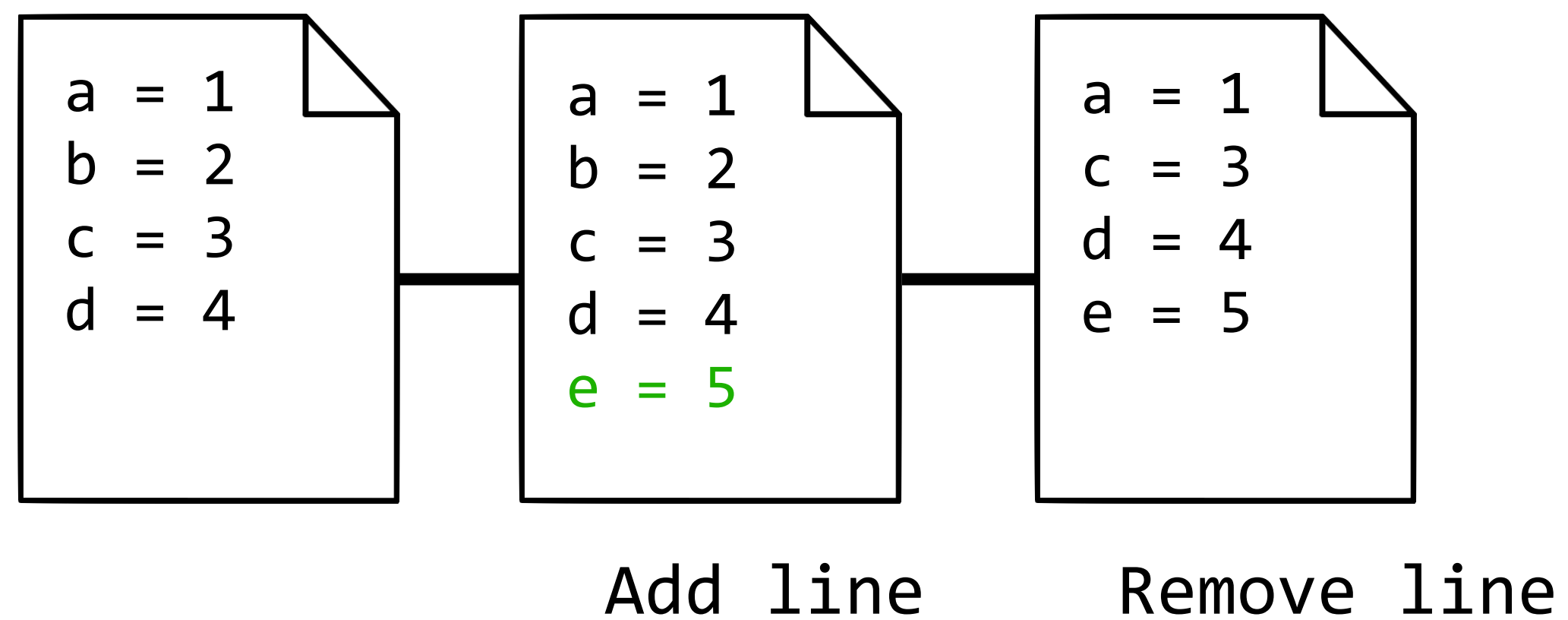
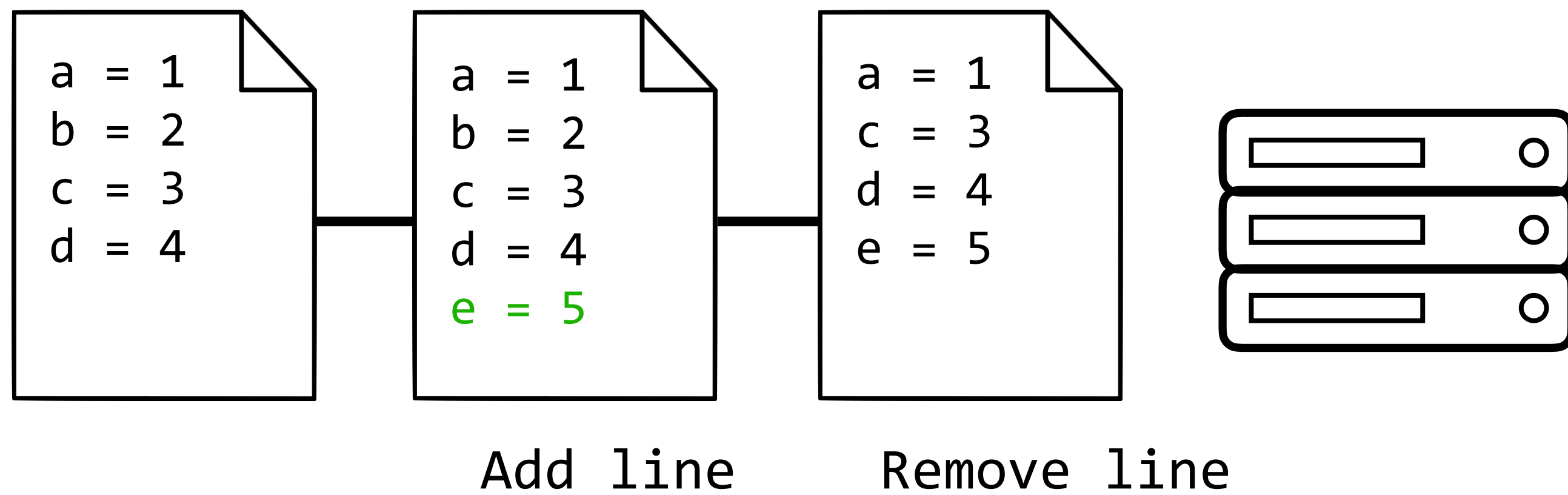
```
git pull
```

# git pull





# git pull



# Merge Conflicts

# Merge Conflicts



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

# Merge Conflicts



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

# Merge Conflicts



Your Changes  
Remote Changes

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

Conflicting Commit  
↓

# Merge Conflicts



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

# Merge Conflicts



```
a = 1
```

```
b = 2
```

```
c = 3
```

```
d = 4
```

```
e = 5
```

# Merge Conflicts

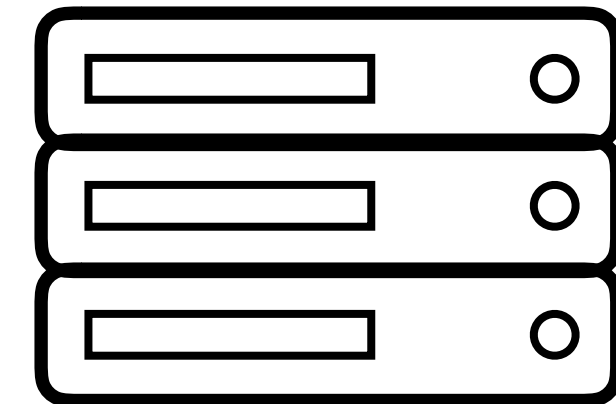


```
a = 1  
b = 2  
c = 3  
d = 4  
e = 5
```



```
git log
```

# git log



```
commit 436f6d6d6974204d73672048657265
Author: Brian Yu <brian@cs.harvard.edu>
Date:   Tue Jun 25 14:06:28 2019 -0400
```

Remove a line

```
commit 57656c636f6d6520746f20576562
Author: Brian Yu <brian@cs.harvard.edu>
Date:   Tue Jun 25 14:05:28 2019 -0400
```

Add a line



`git reset`

# git reset



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

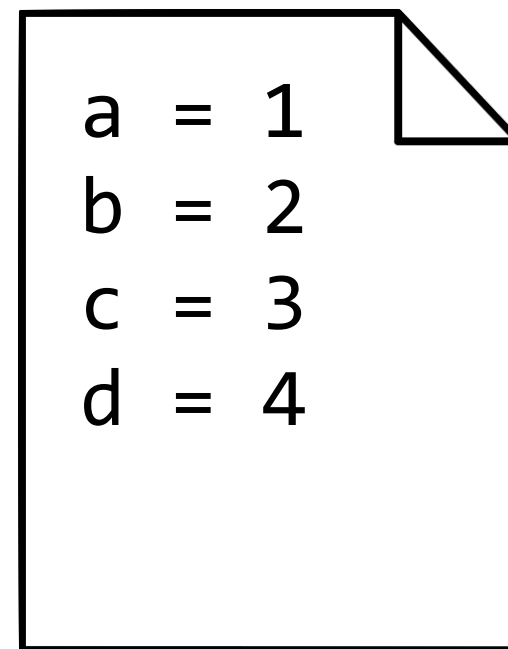
a = 1  
b = 2  
c = 3  
d = 4  
e = 5

a = 1  
c = 3  
d = 4  
e = 5

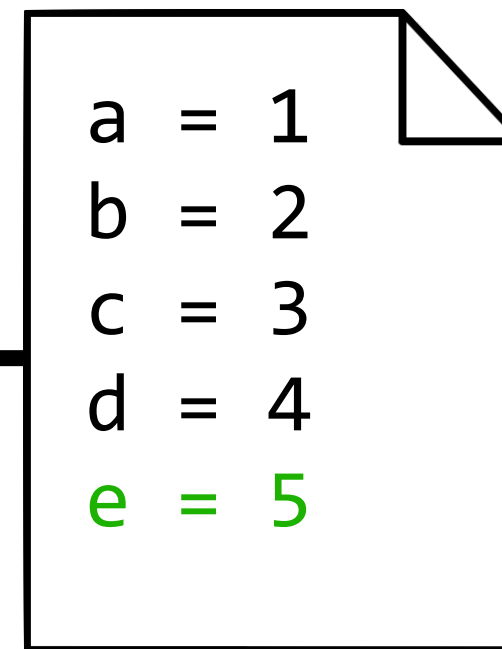
Add line  
57656c6

Remove line  
436f6d6

# git reset



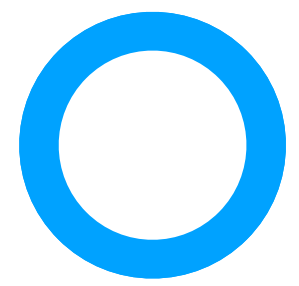
a = 1  
b = 2  
c = 3  
d = 4



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

Add line  
57656c6

# Making Changes



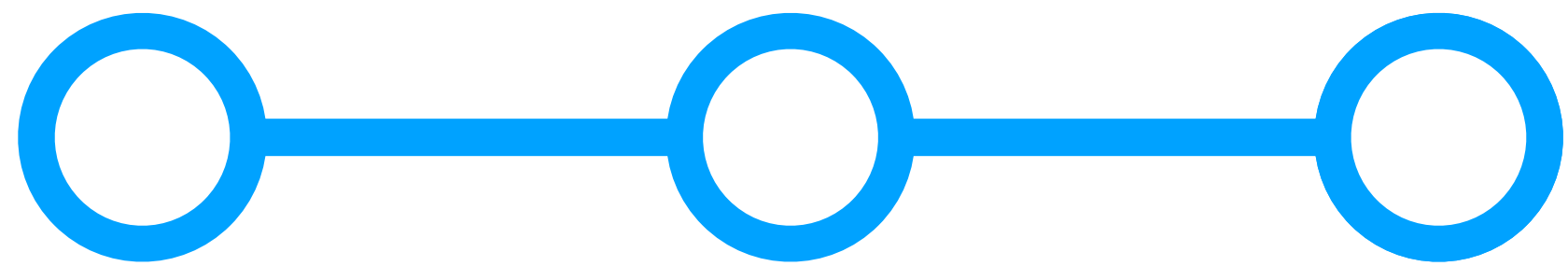
first  
commit



first  
commit

changes





first  
commit

changes

more  
changes

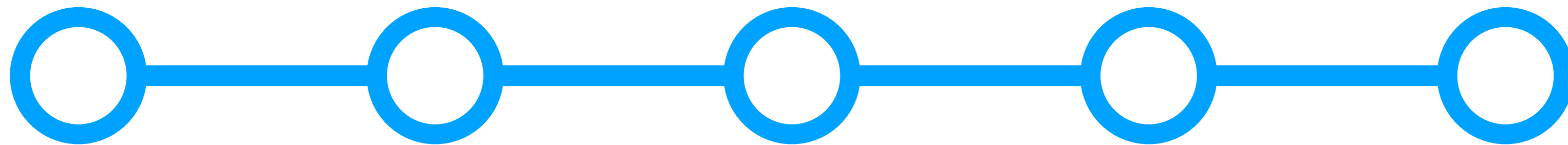


first  
commit

changes

more  
changes

start new  
feature



first  
commit

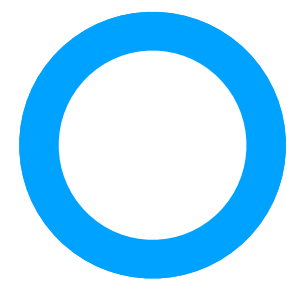
changes

more  
changes

start new  
feature

keep working  
on new feature

# Branching

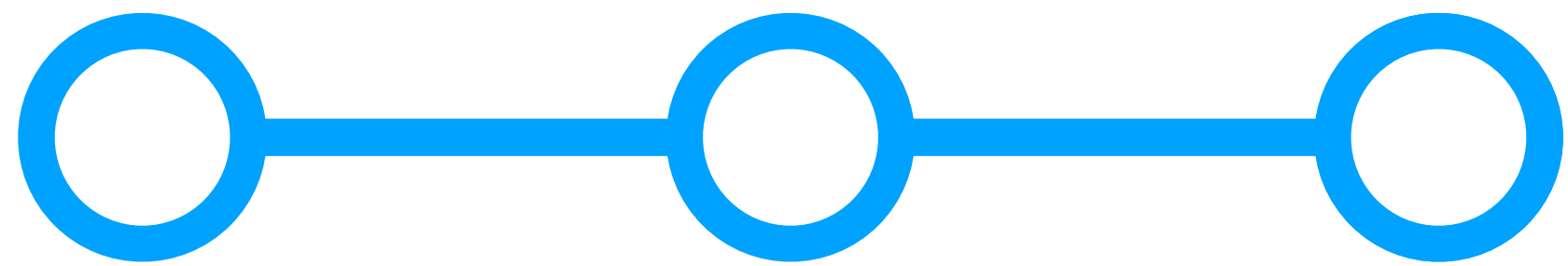


first  
commit



first  
commit

changes

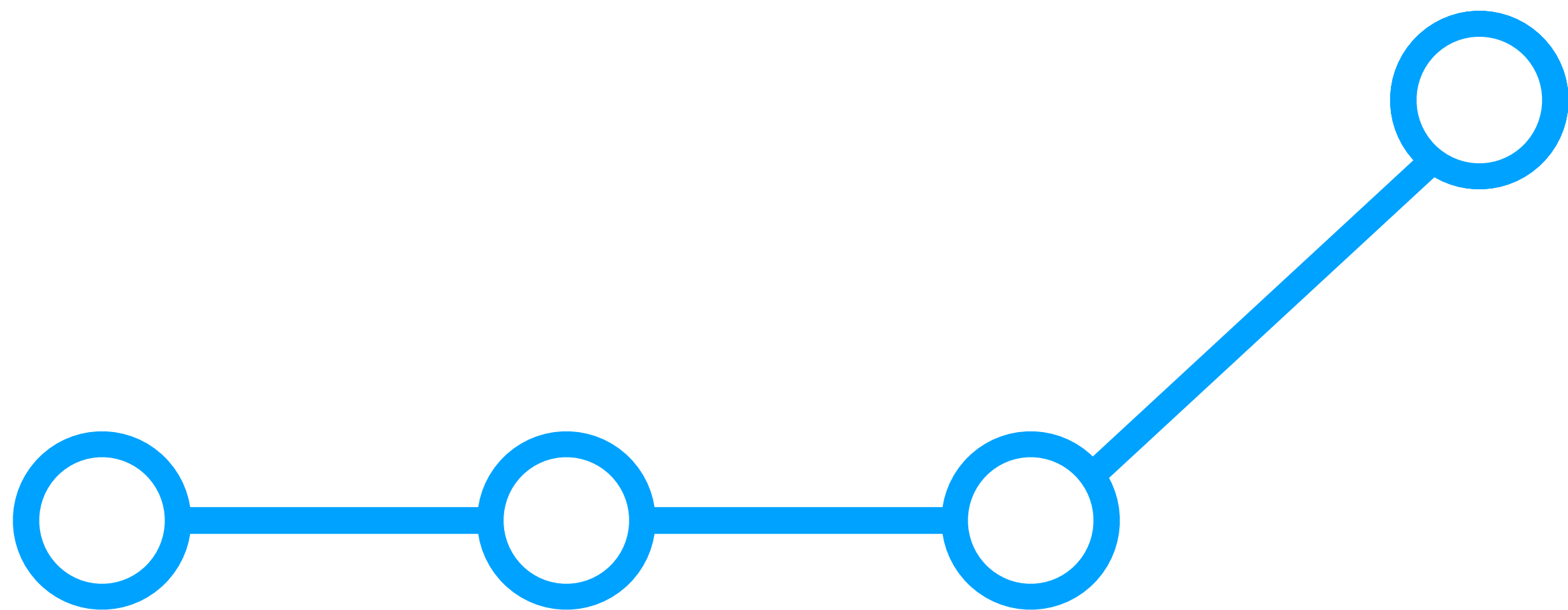


first  
commit

changes

more  
changes

start new  
feature

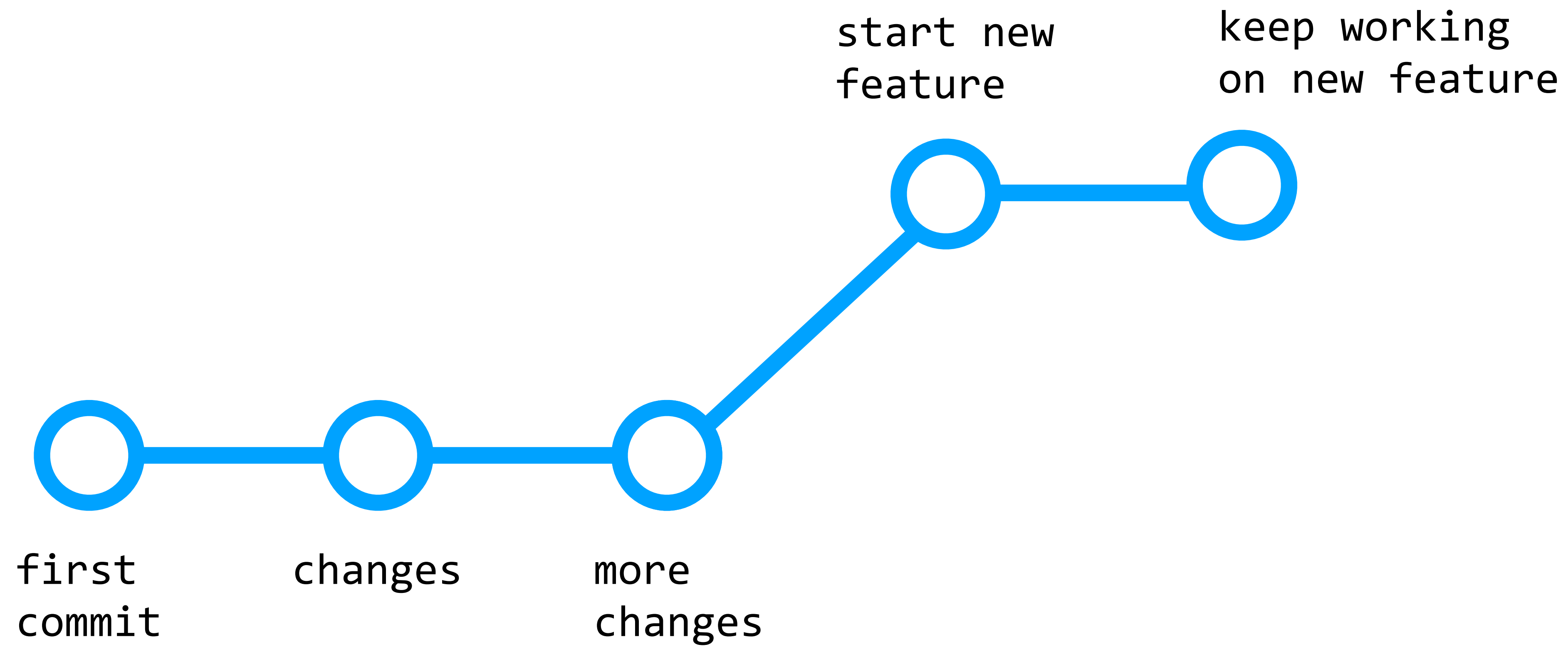


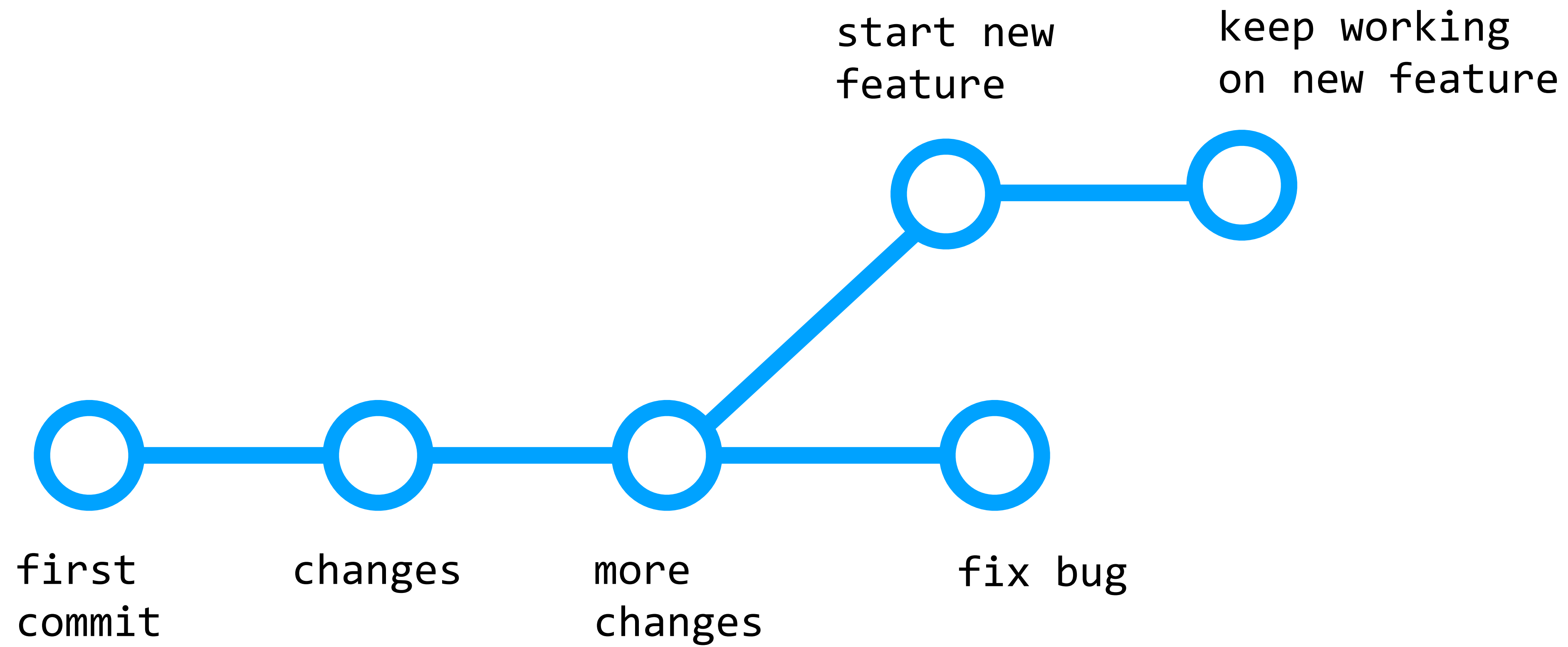
first  
commit

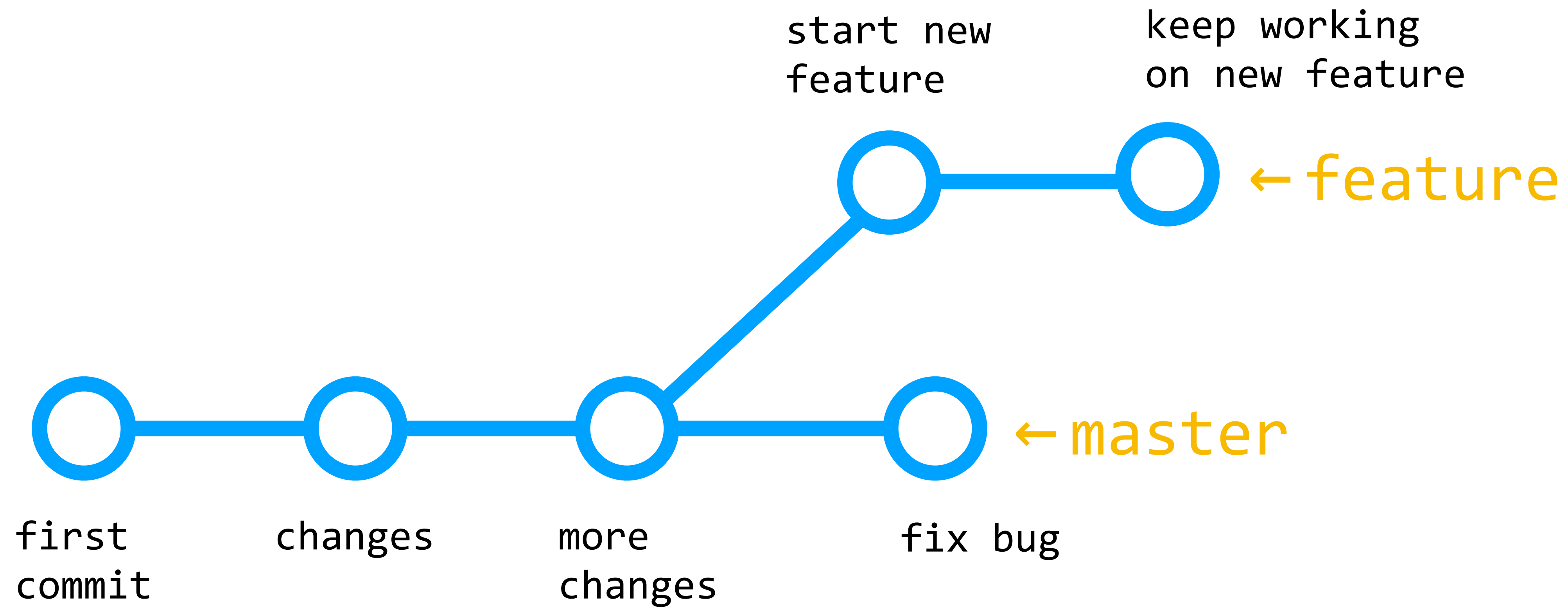
changes

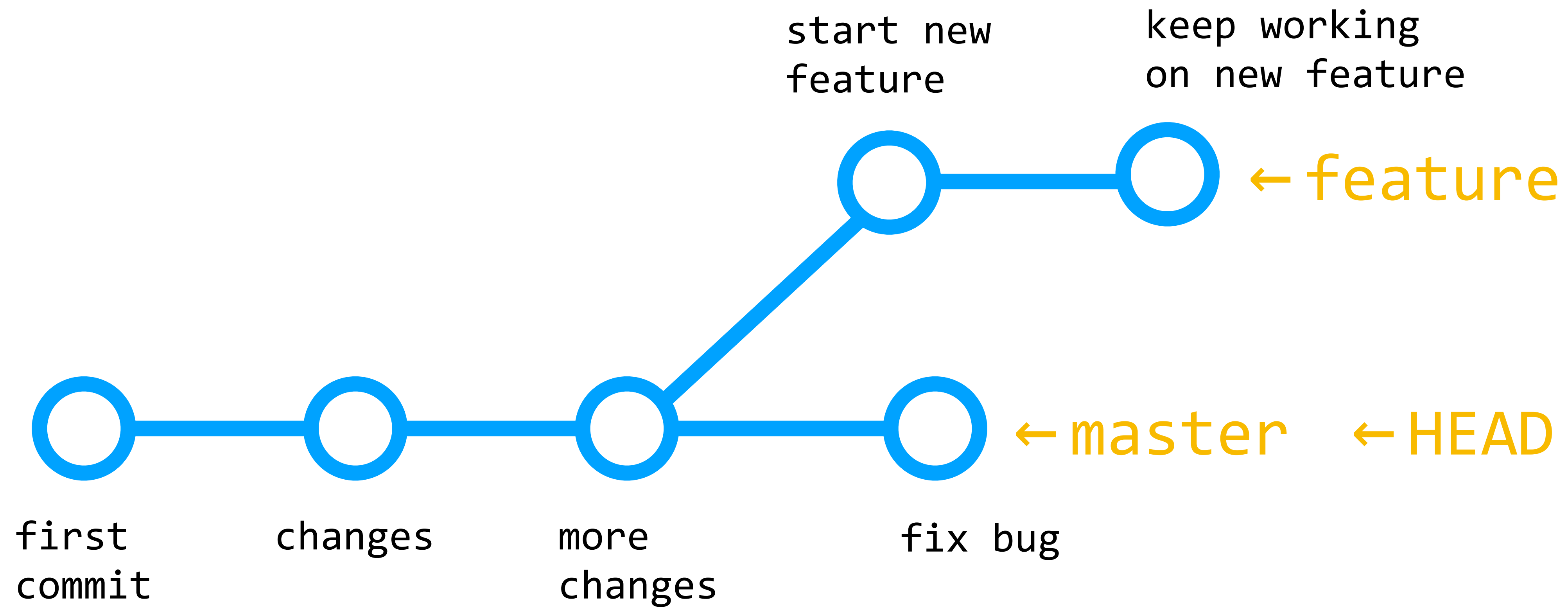
more  
changes



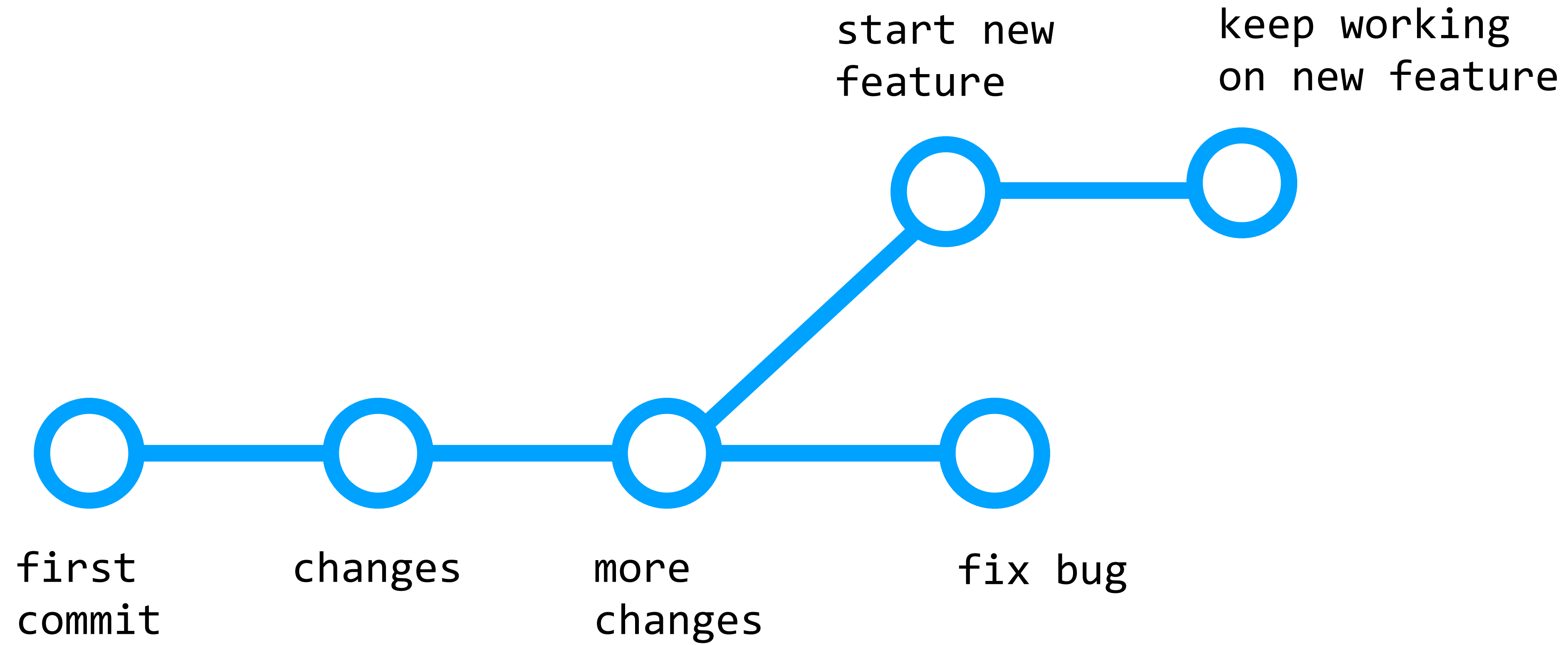


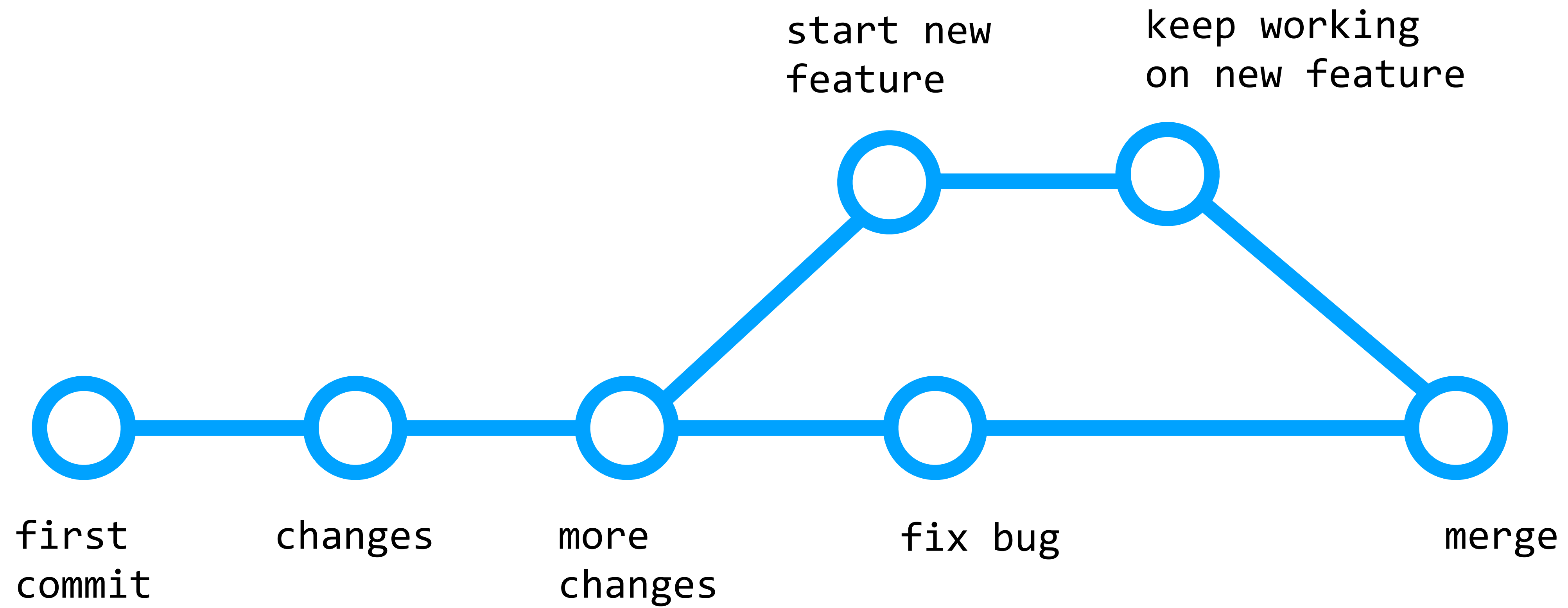












# Branching

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



# GitHub Features

- **Issues**
- **Forks**
- **Pull Requests**

# GitHub Pages

# Git

CSCI P-14300  
June 25, 2019

Brian Yu  
[brian@cs.harvard.edu](mailto:brian@cs.harvard.edu)