Git

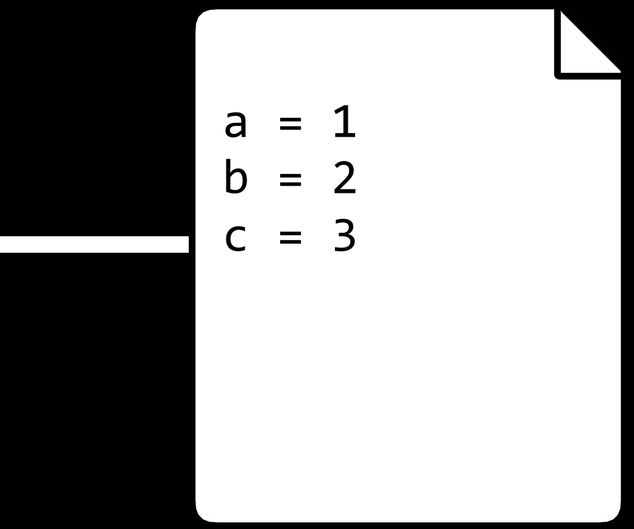
[1]

git

git (engl. Slang für Blödmann)

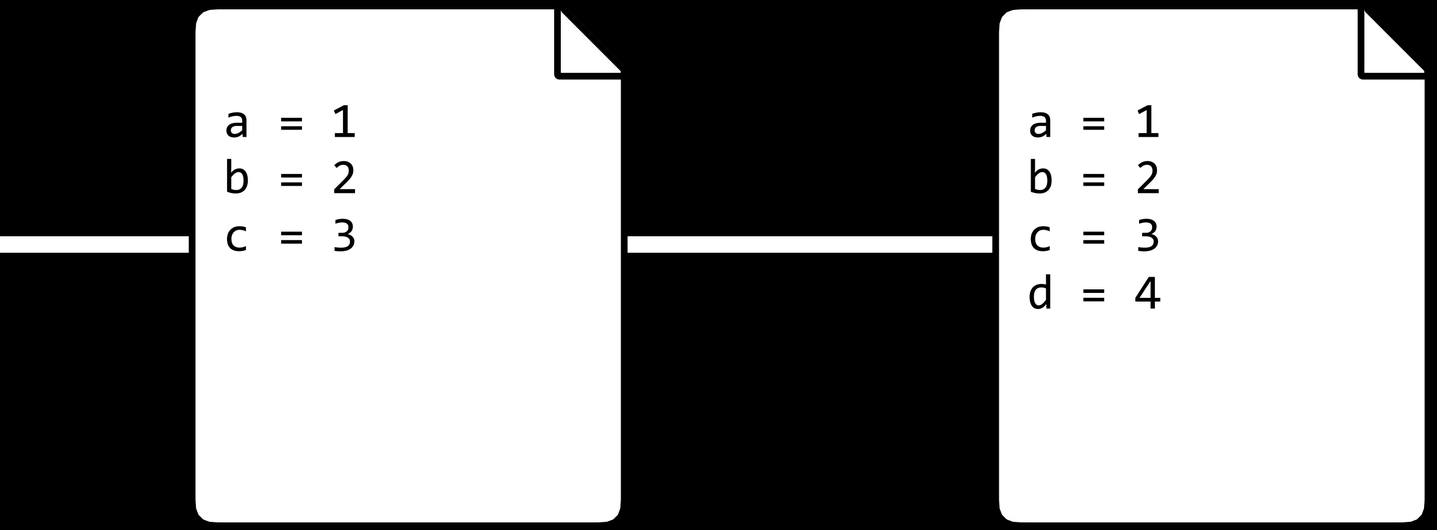
“The joke ‘I name all my projects for myself, first Linux, then git’ was just too good to pass up. But it is also short, easy-to-say, and type on a standard keyboard. And reasonably unique and not any standard command, which is unusual.”

Keep track of changes to code.



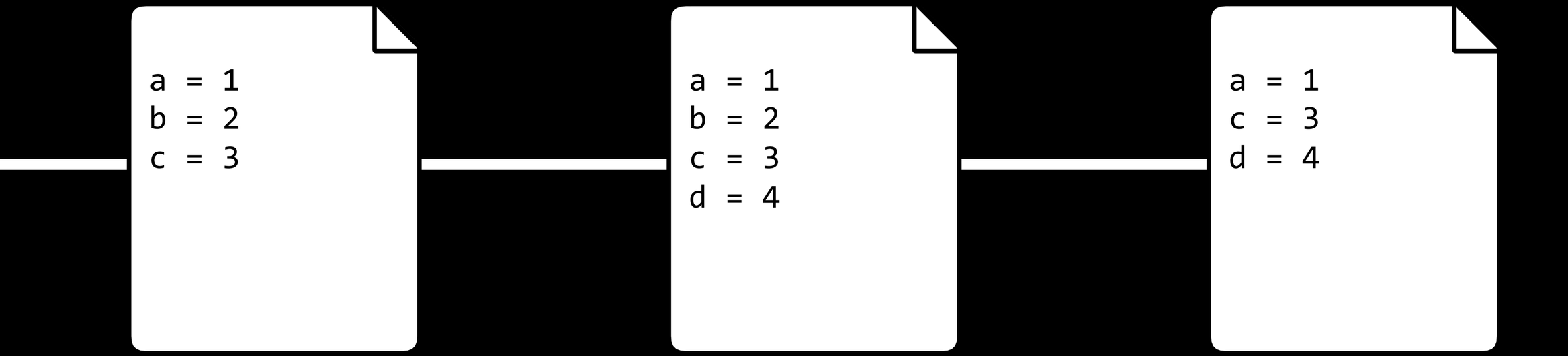
Create file

Keep track of changes to code.



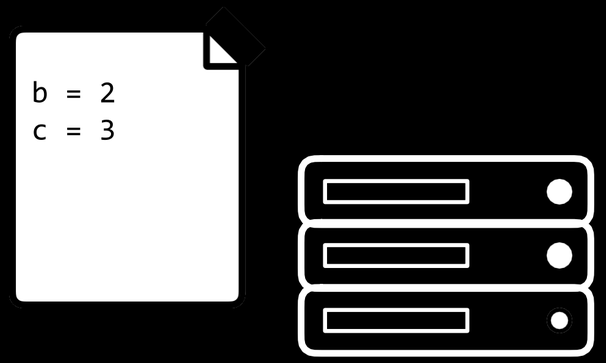
[1]

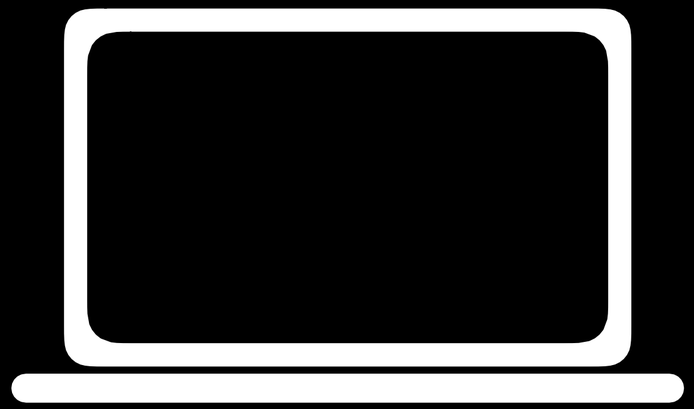
Keep track of changes to code.

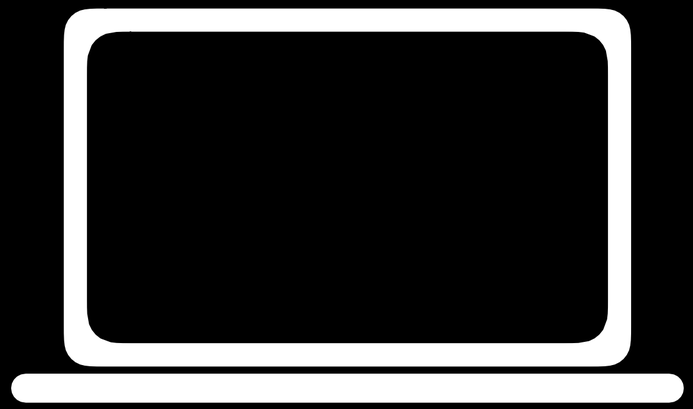


[1]

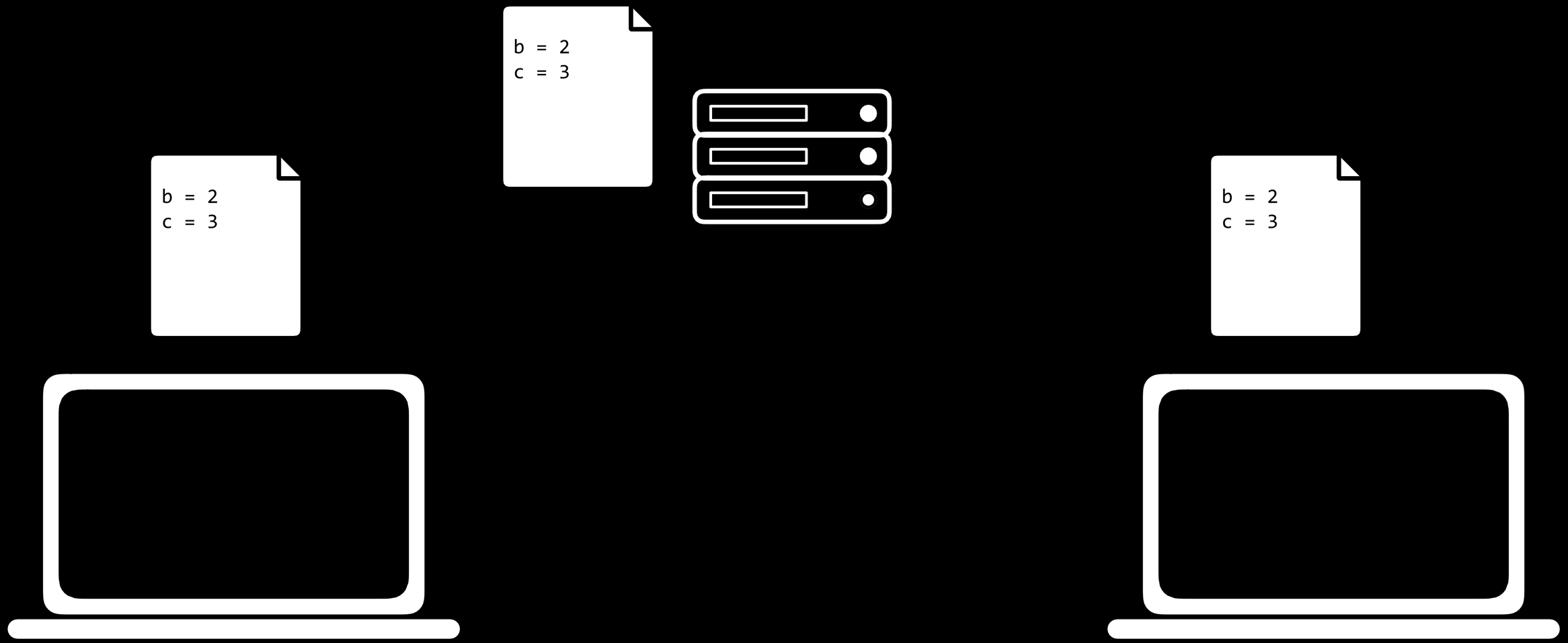
Synchronizes code between different people.





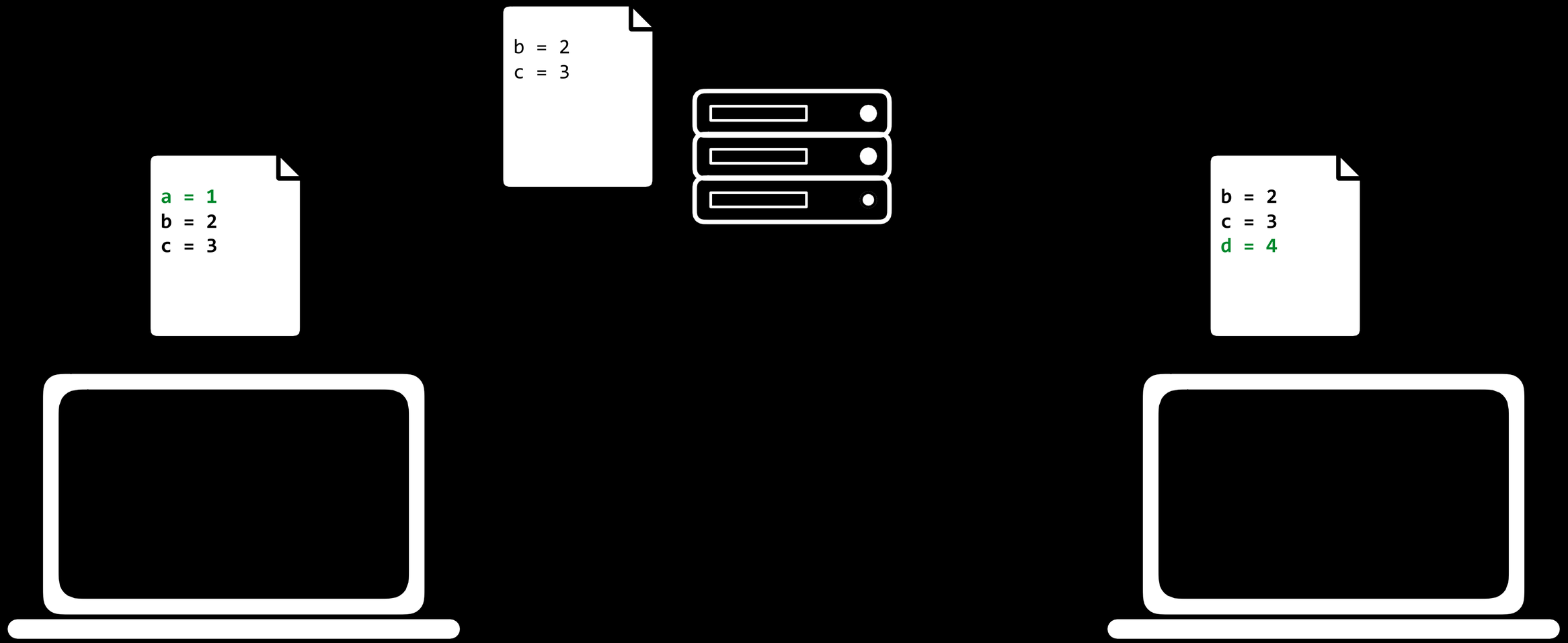


Synchronizes code between different people.



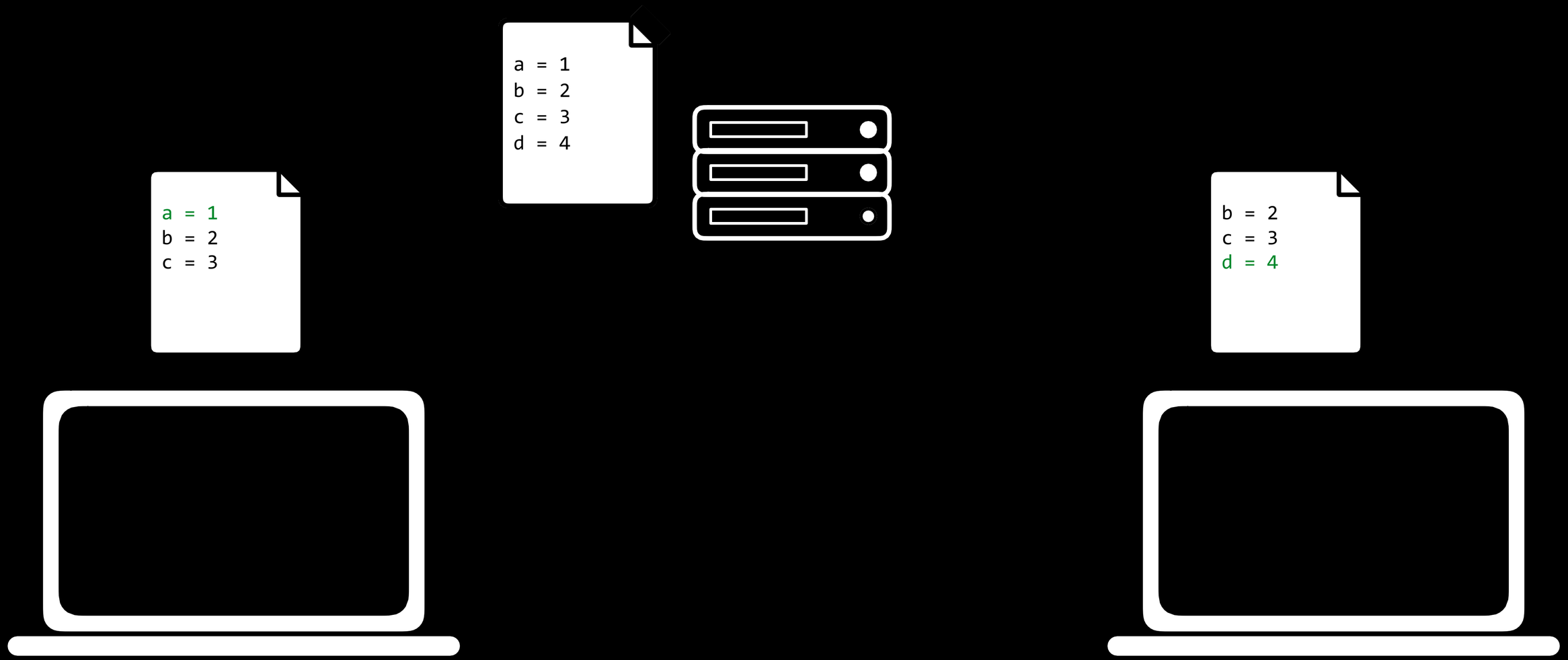
[1]

Synchronizes code between different people.



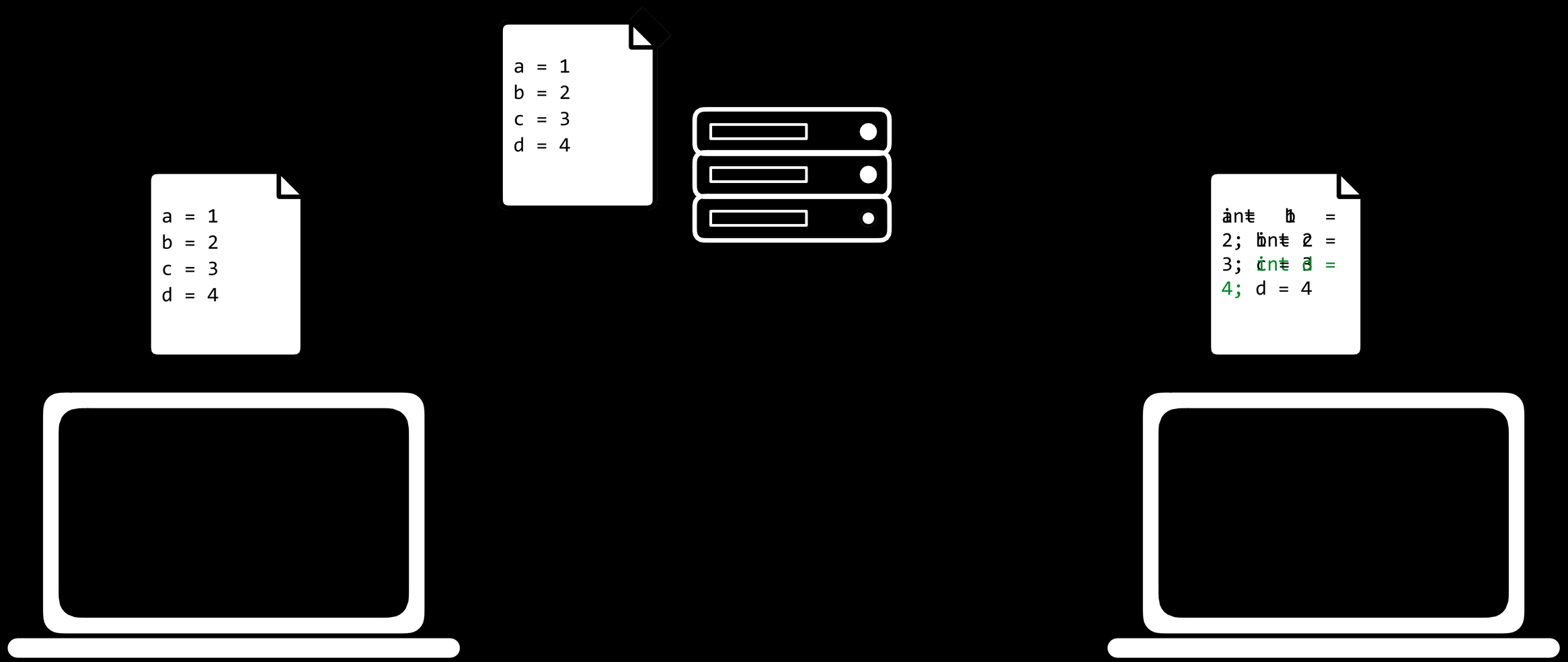
[1]

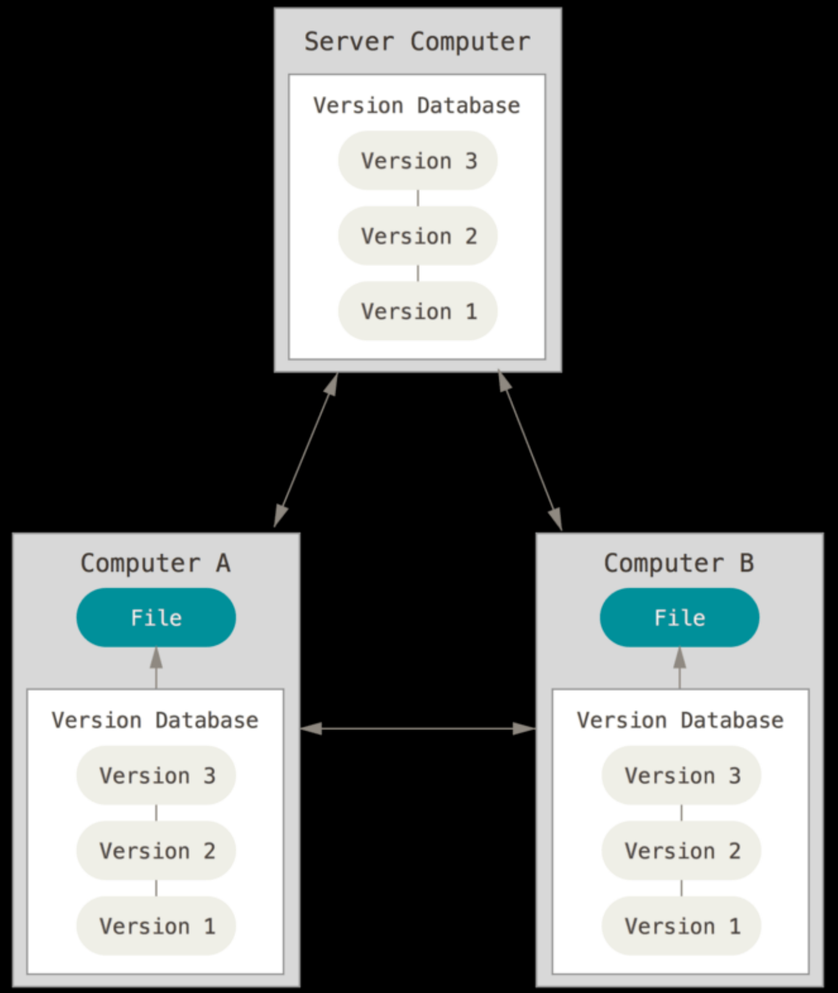
Synchronizes code between different people.



[1]

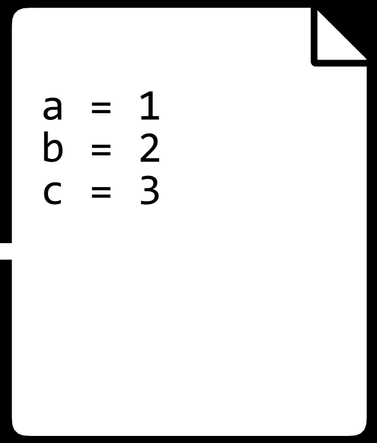
Synchronizes code between different people.



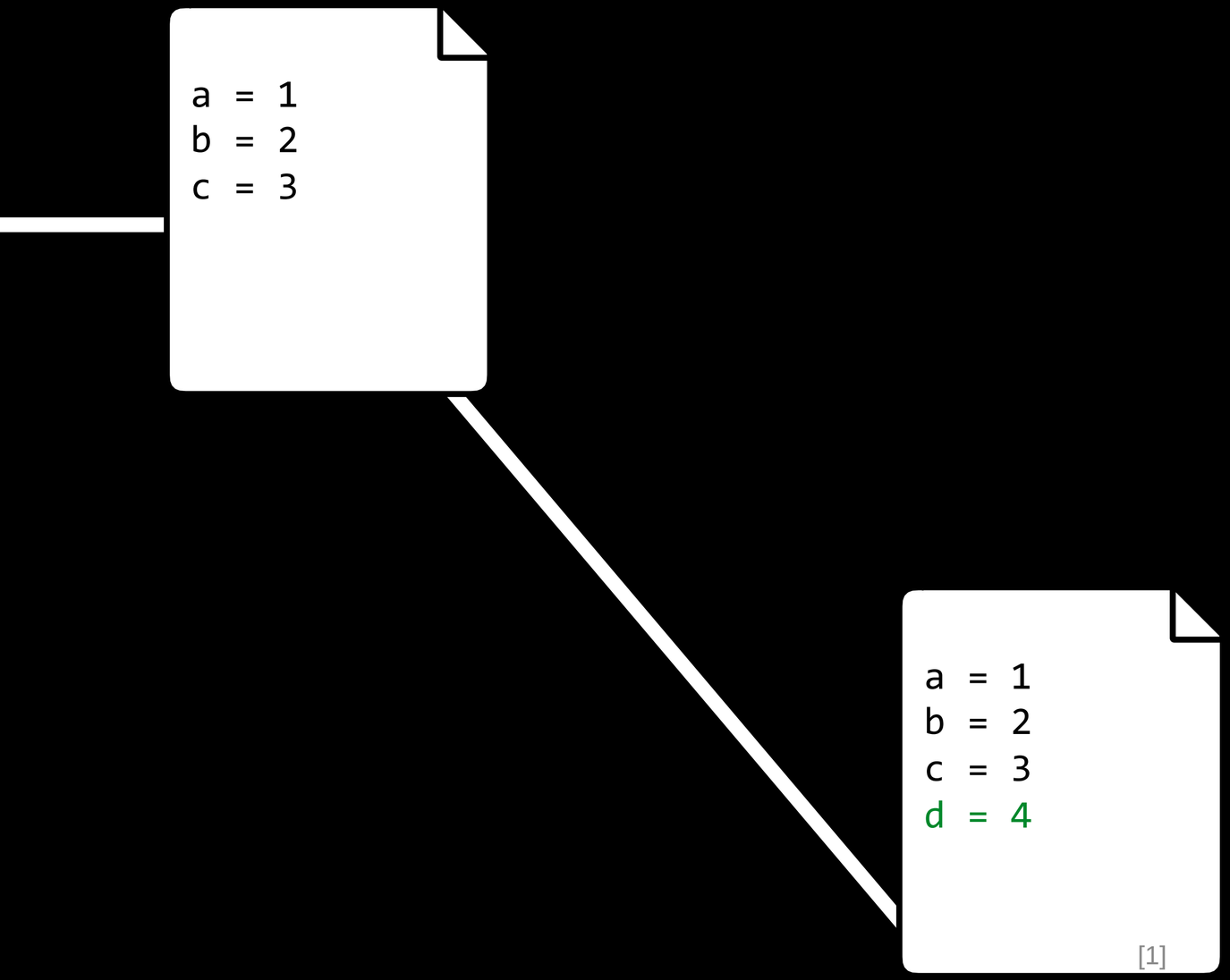

Server Computer
Version Database
Version 3
Version 2
Version 1
Computer A
File
Version Database
Version 3
Version 2
Version 1
Computer B
File
Version Database
Version 3
Version 2
Version 1

[3]

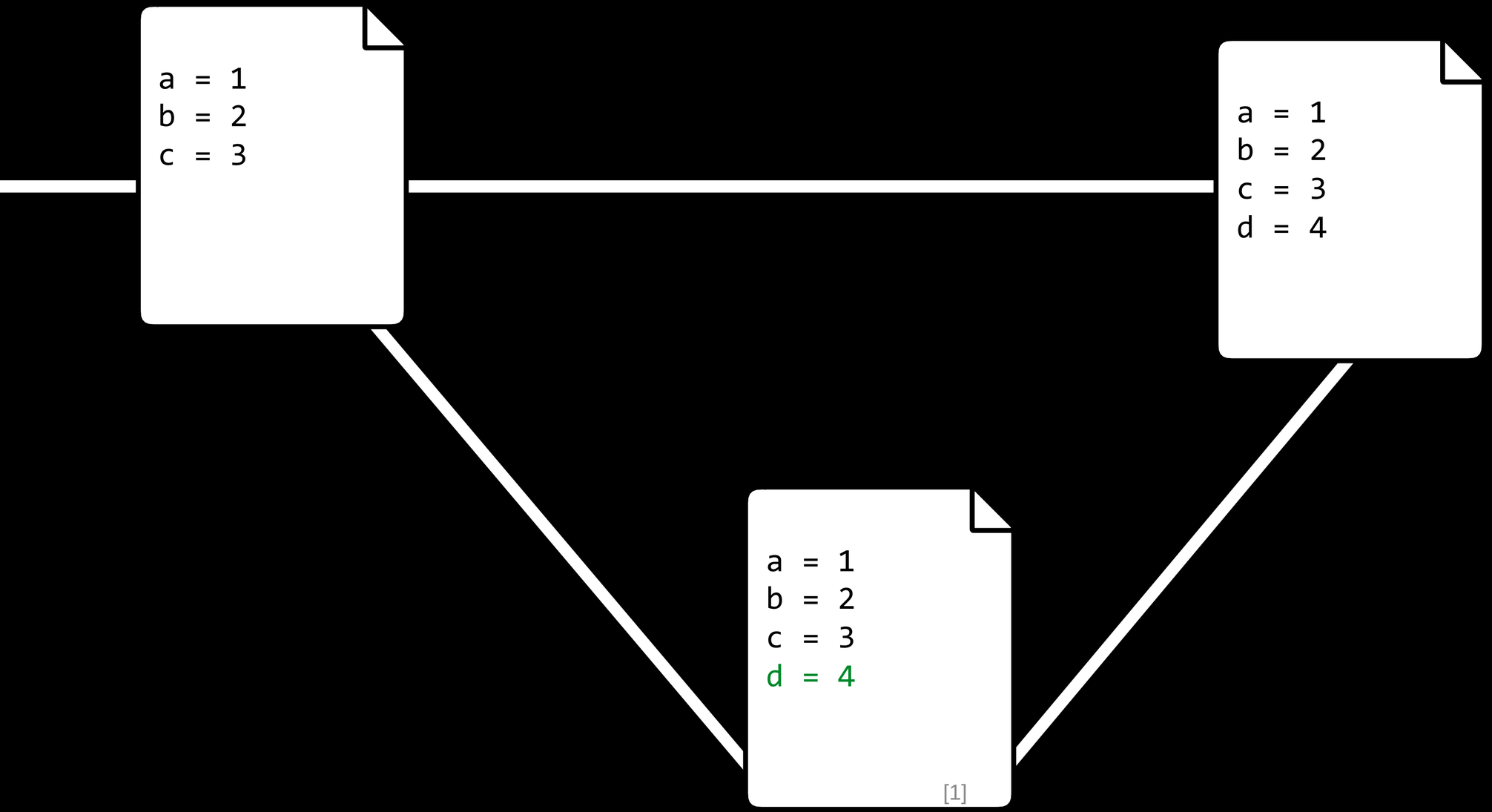
Test changes to code without losing the original.



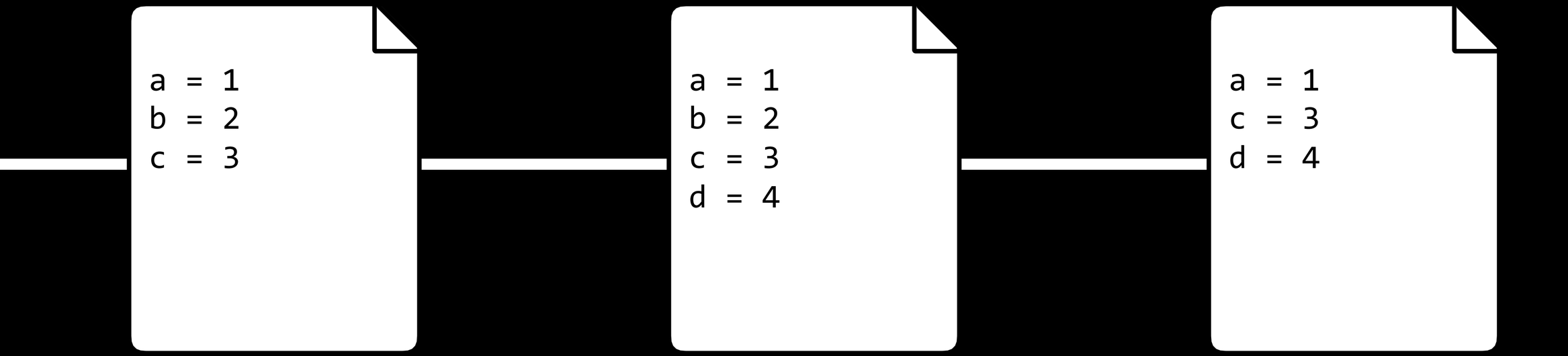
Test changes to code without losing the original.



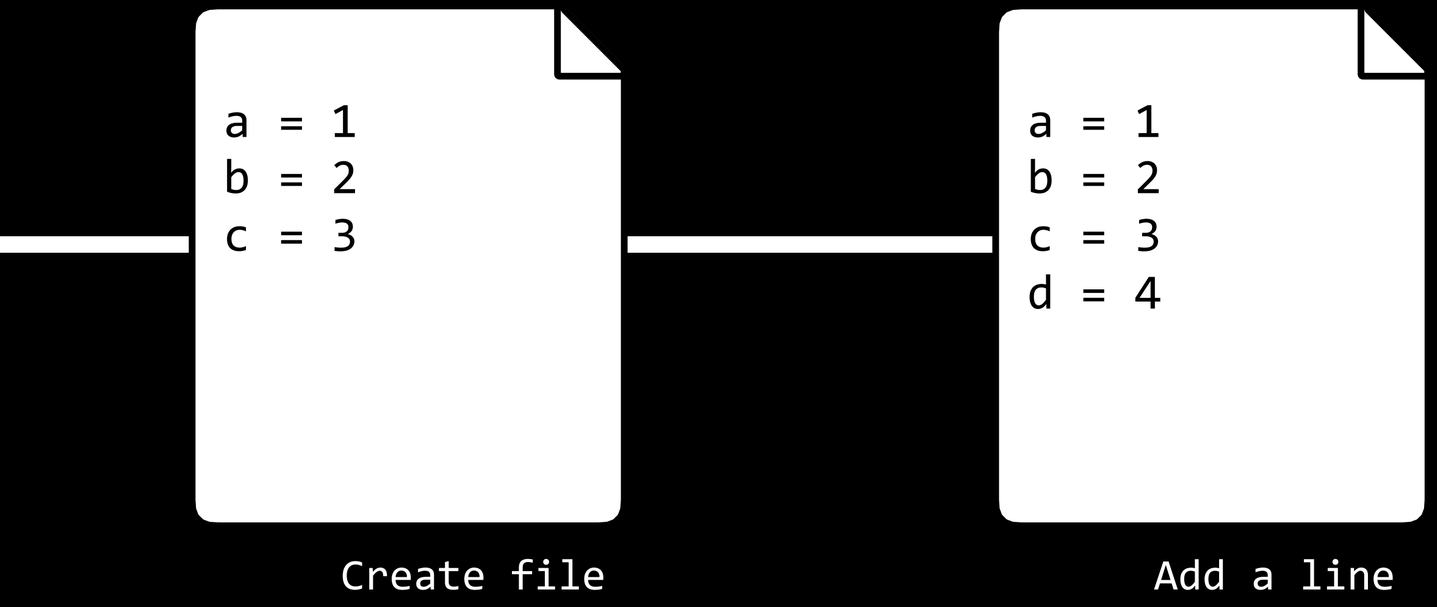
Test changes to code without losing the original.



Revert back to old versions of code.



Revert back to old versions of code.

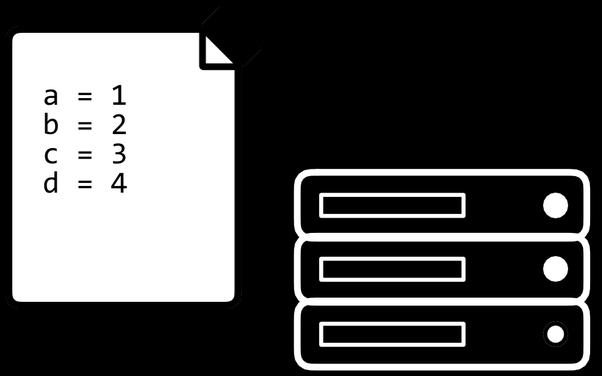


GitHub

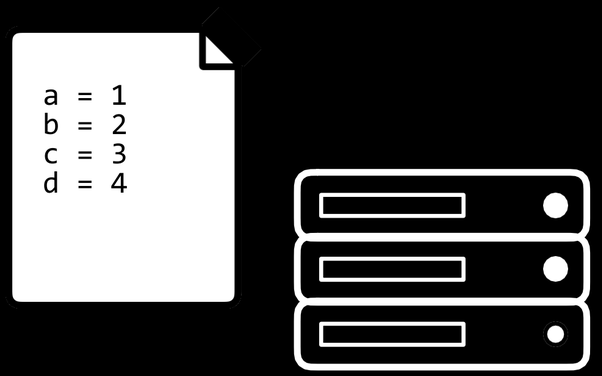
[1]

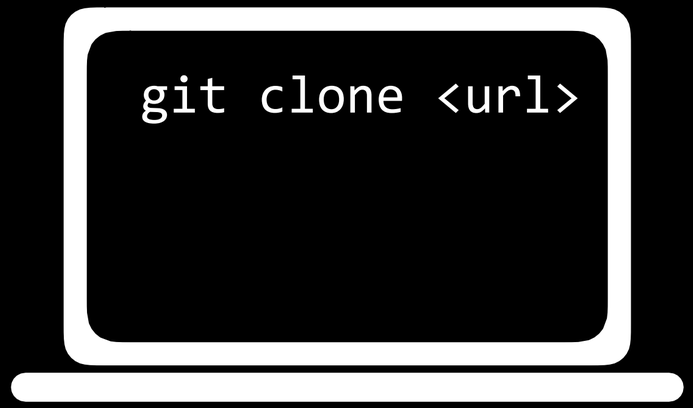
git clone

git clone <url>

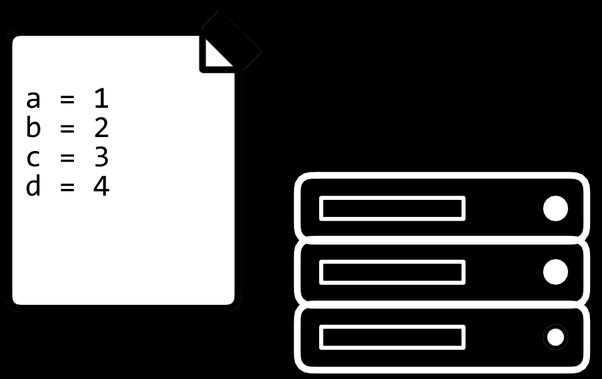


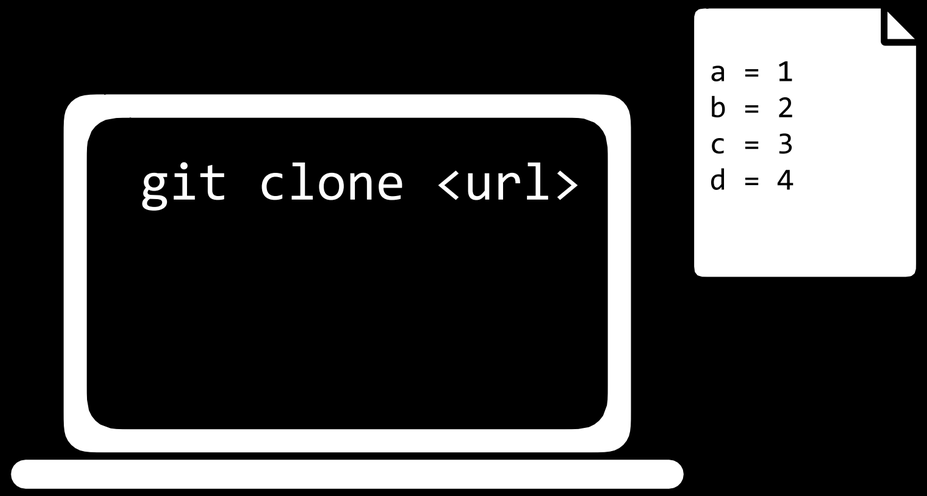
git clone <url>





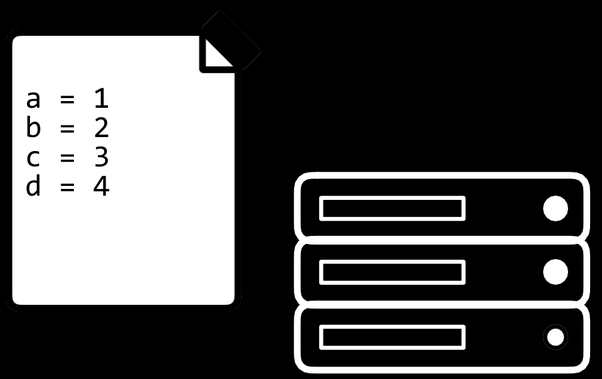
git clone <url>

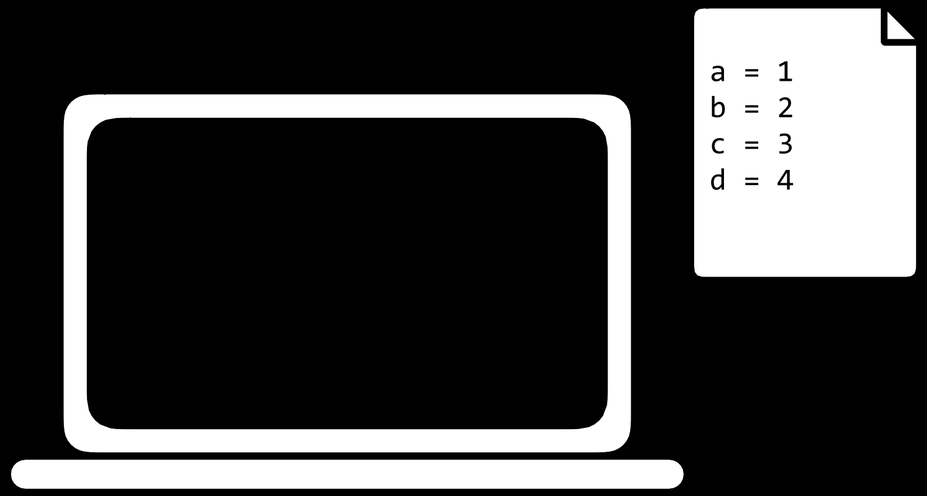




git add

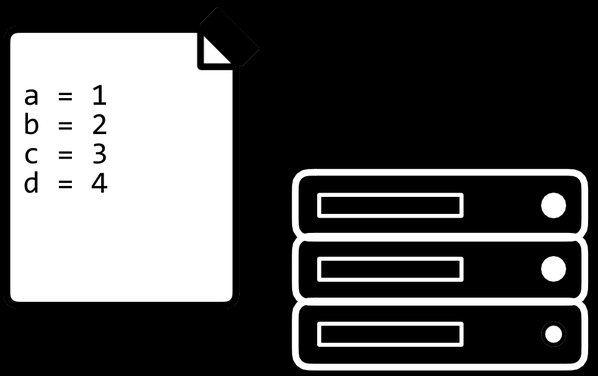
git add <filename>

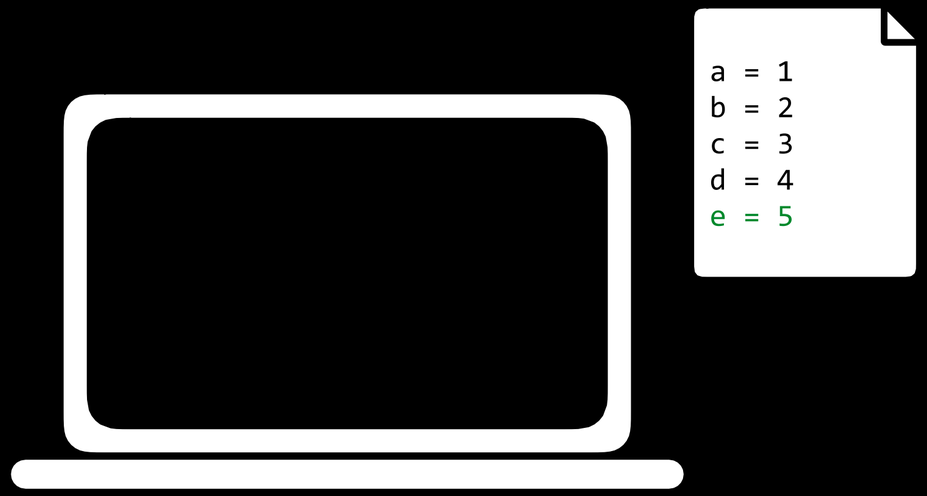




[1]

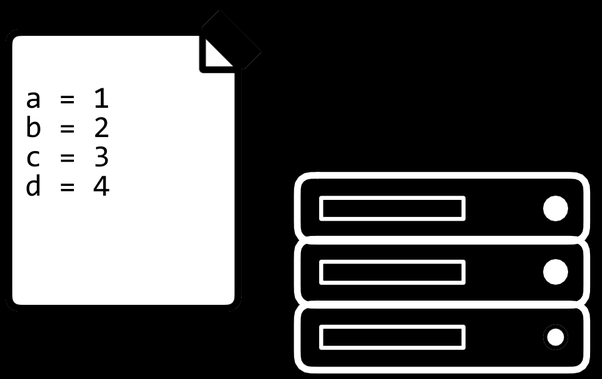
git add <filename>





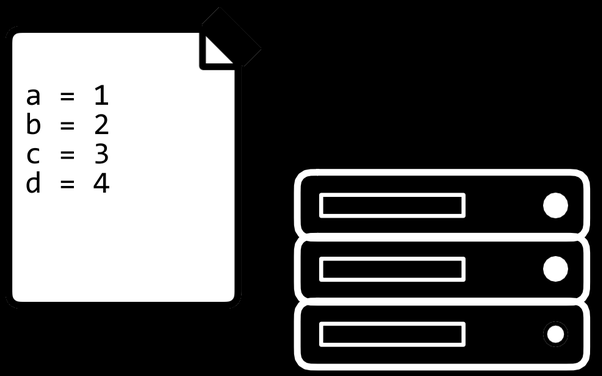
[1]

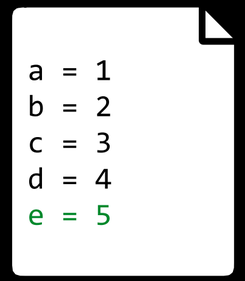
git add <filename>



[1]

git add <filename>





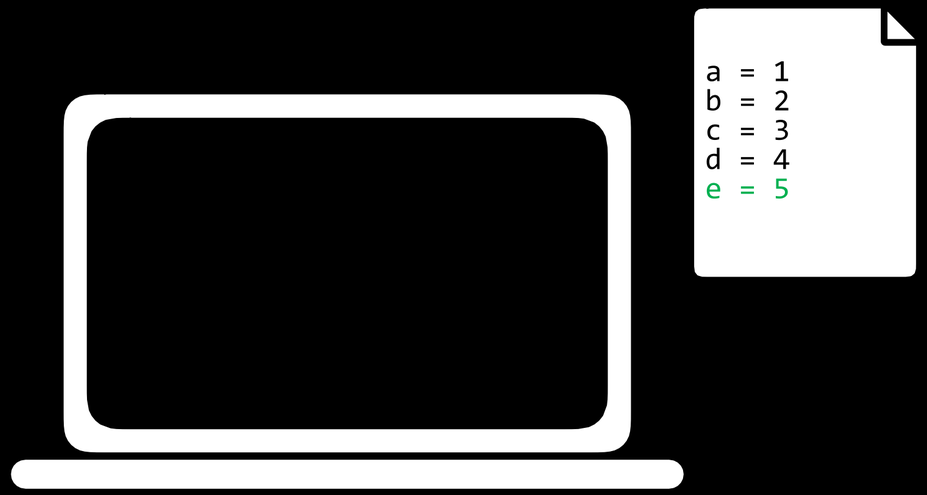
Changes to be committed:

modified: foo.py

[1]

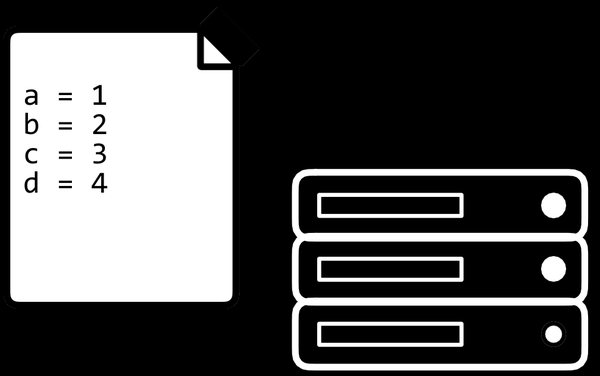
git commit

git commit -m "message"



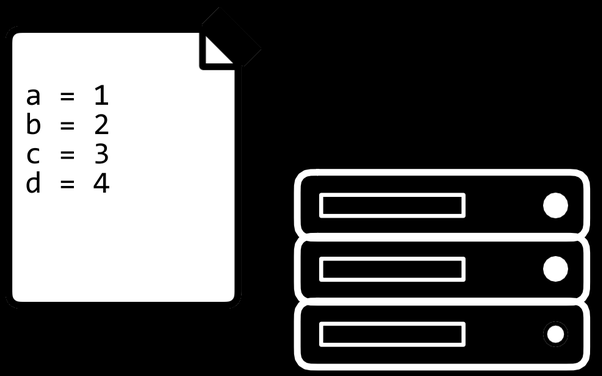
[1]

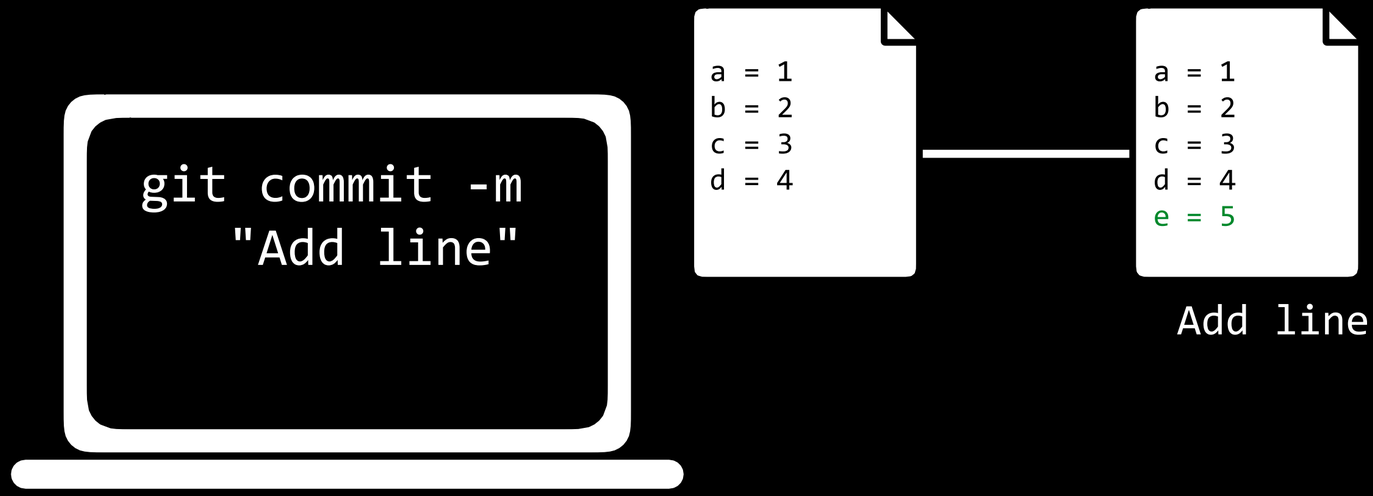
git commit -m "message"



[1]

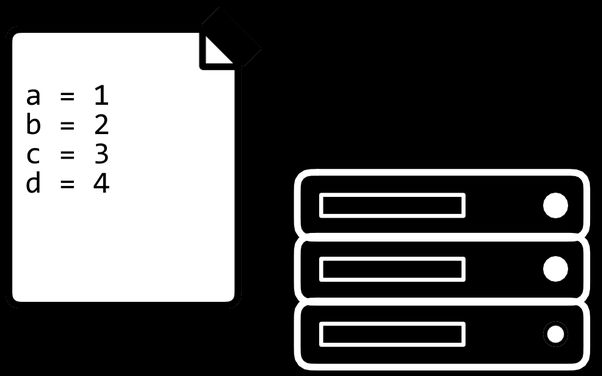
git commit -m "message"

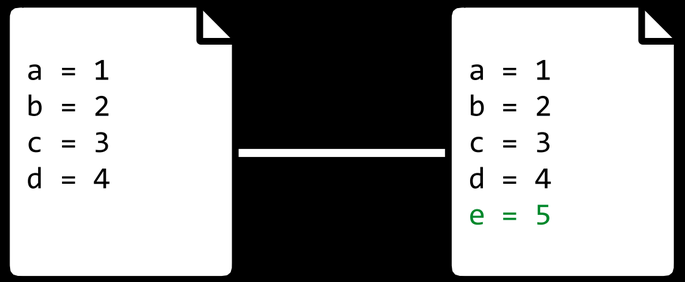




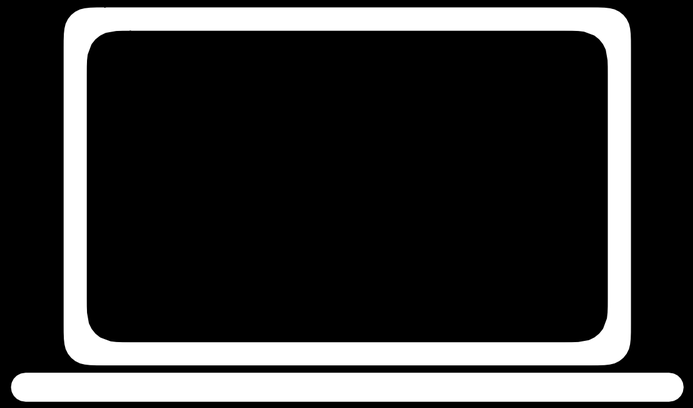
git status

git status

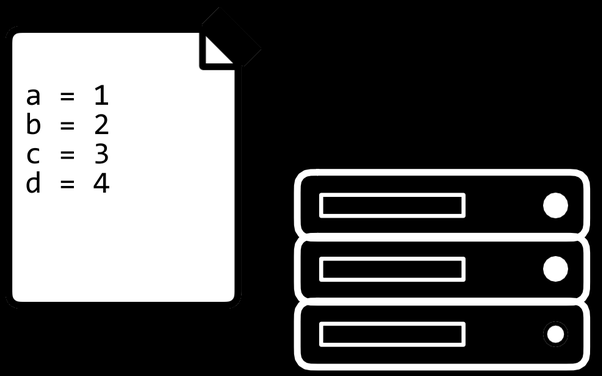


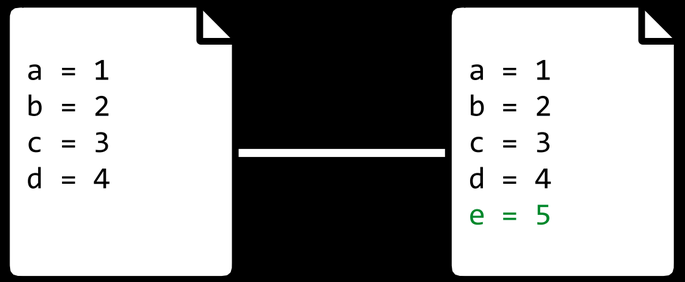


Add line



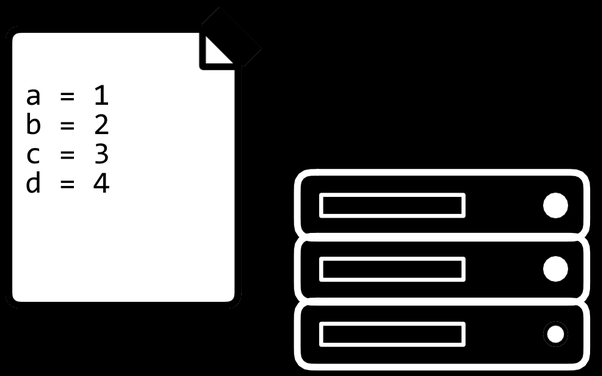
git status

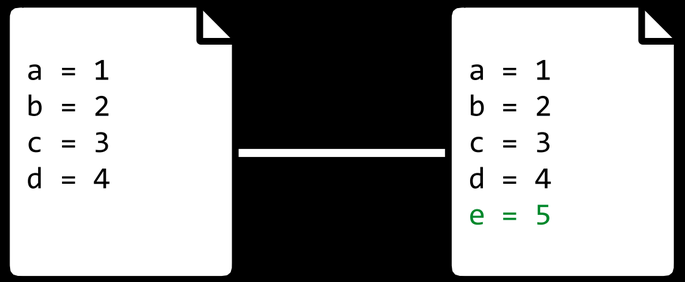




Add line

git status



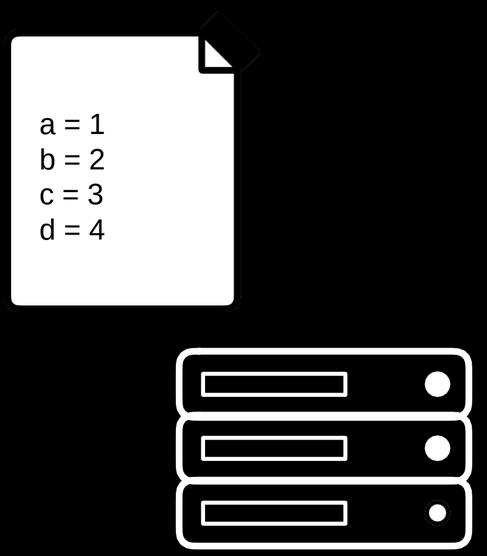


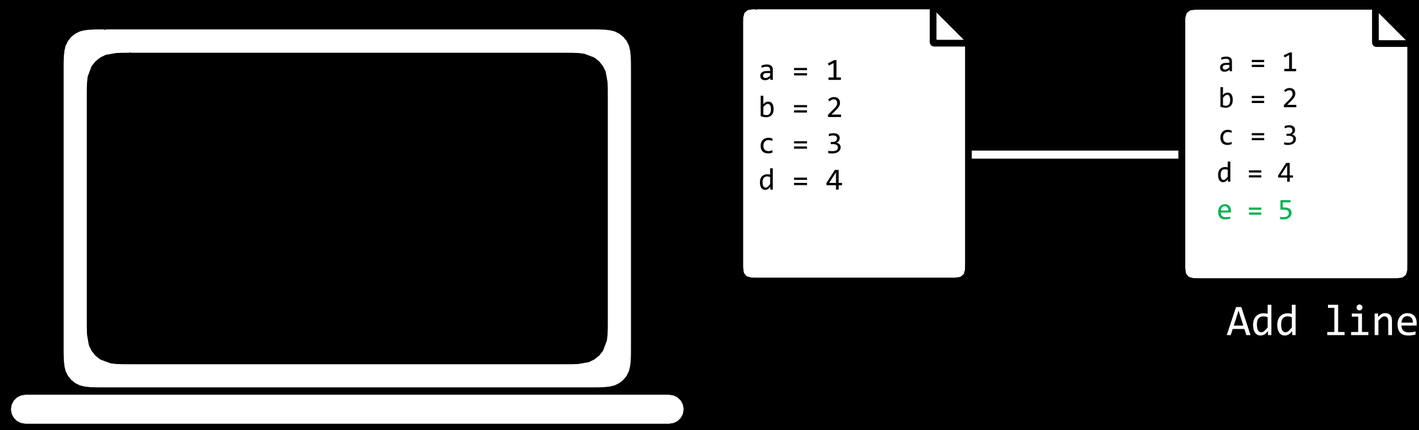
Add line


```
On branch master
Your branch is ahead of 'origin/master' by 1 commit.
  (use "git push" to publish your local commits)
```

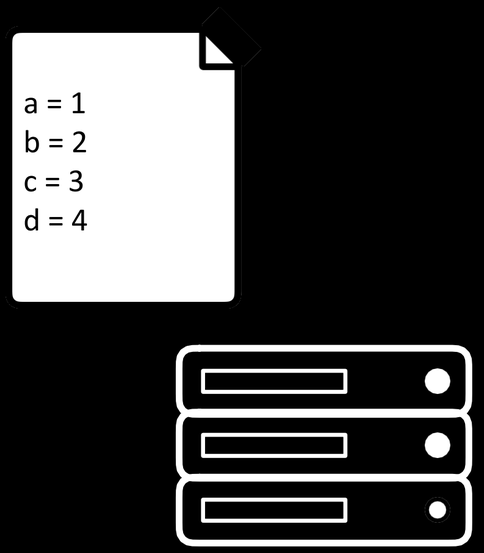
git push

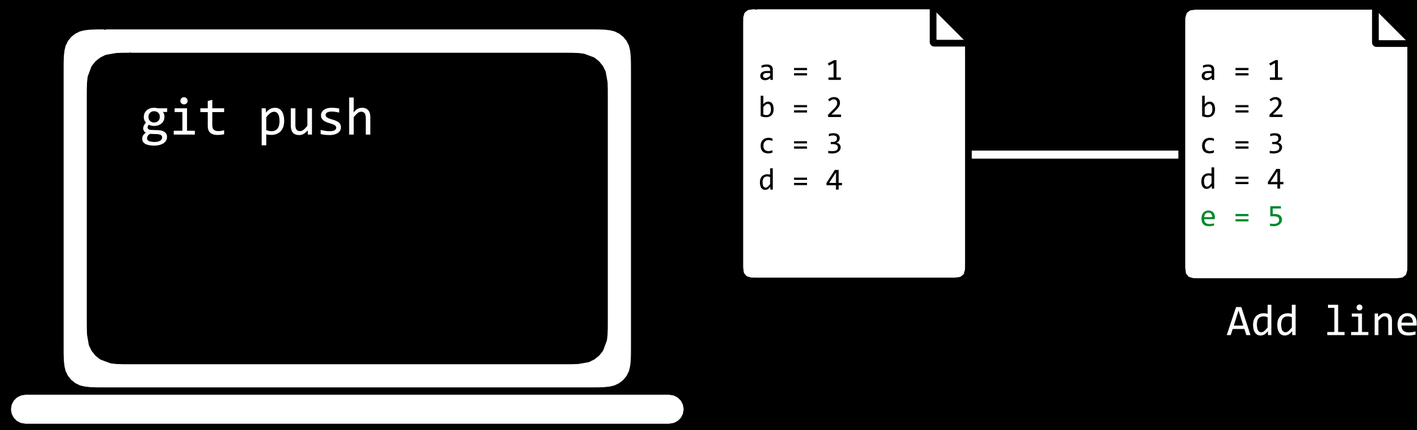
git push



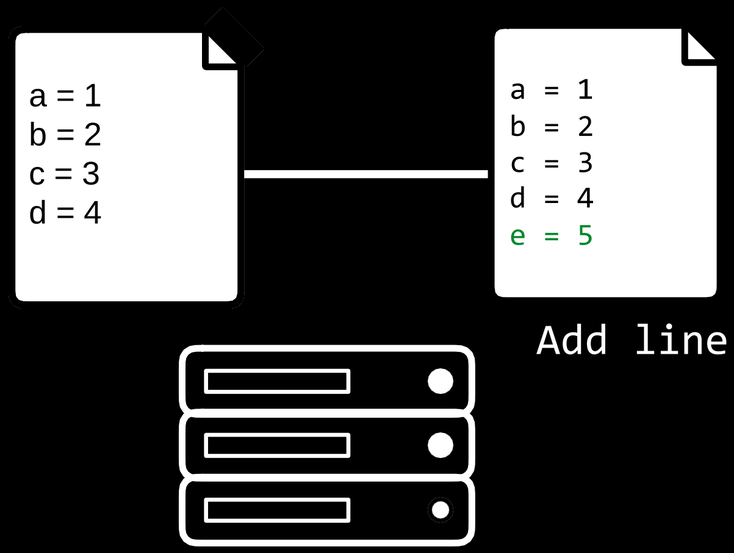


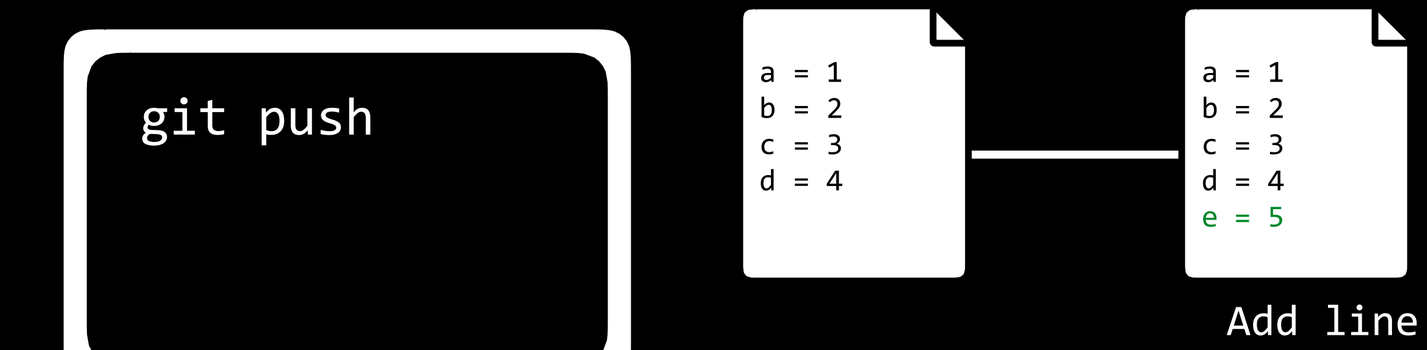
git push





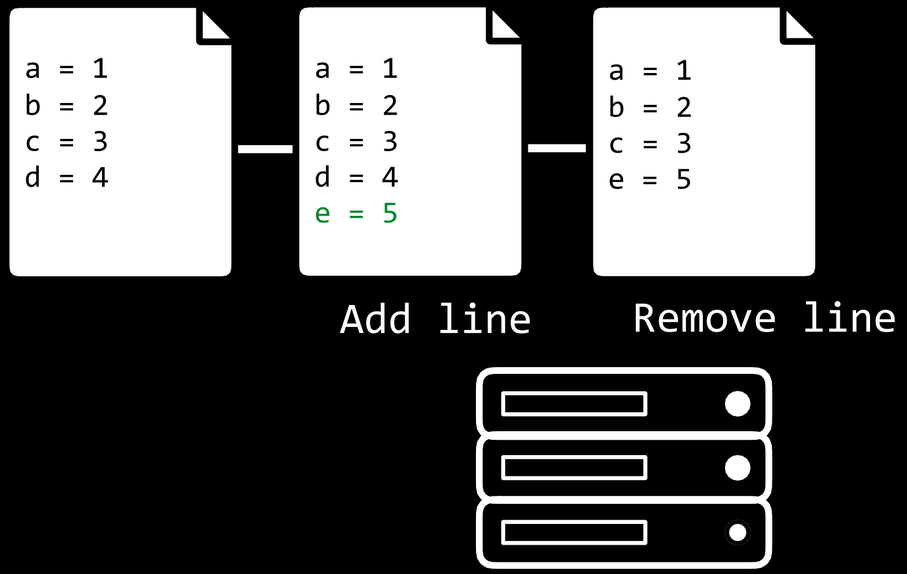
git push

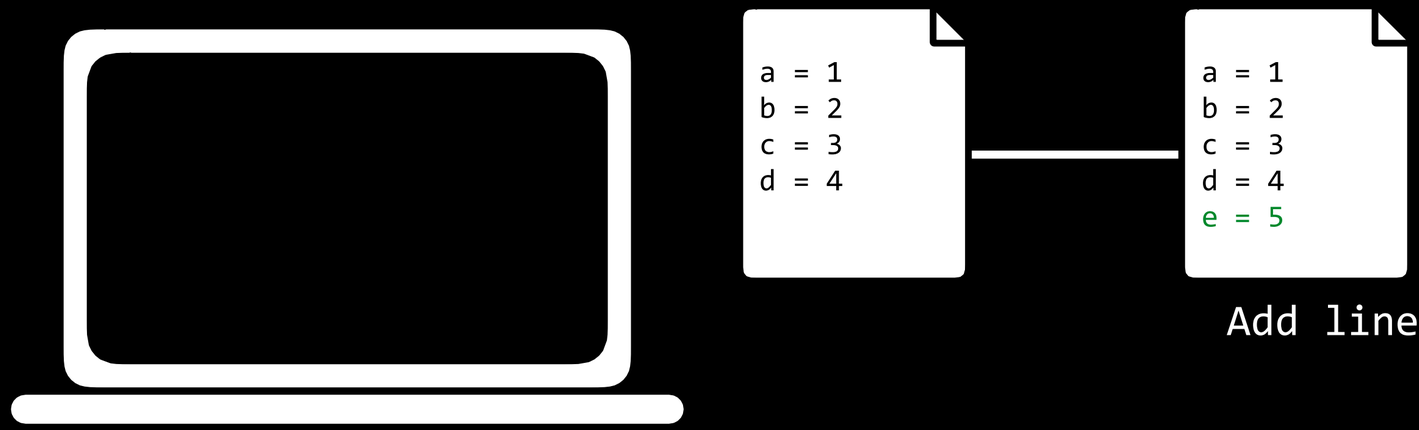

a = 1
b = 2
c = 3
d = 4
a = 1
b = 2
c = 3
d = 4
e = 5
Add line


git push
a = 1
b = 2
c = 3
d = 4
a = 1
b = 2
c = 3
d = 4
e = 5
Add line

git pull

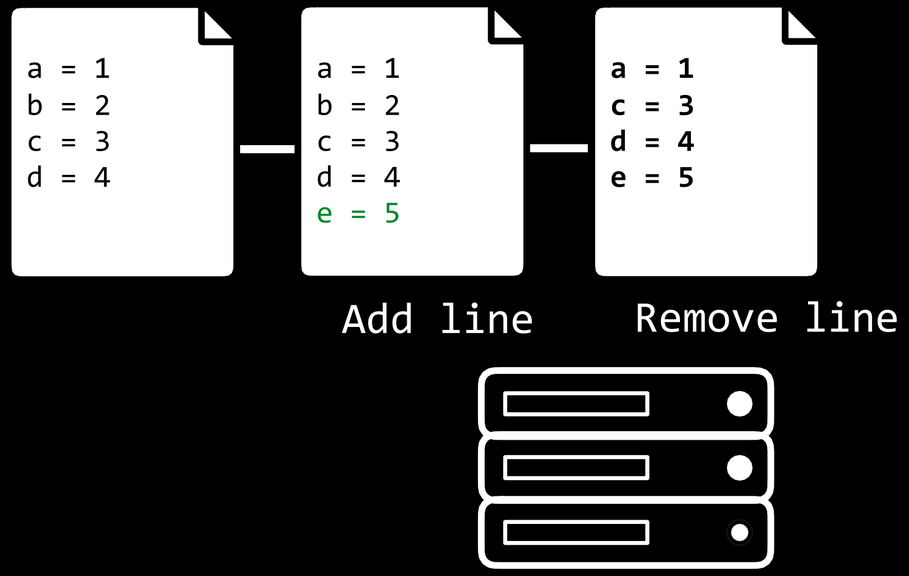
git pull

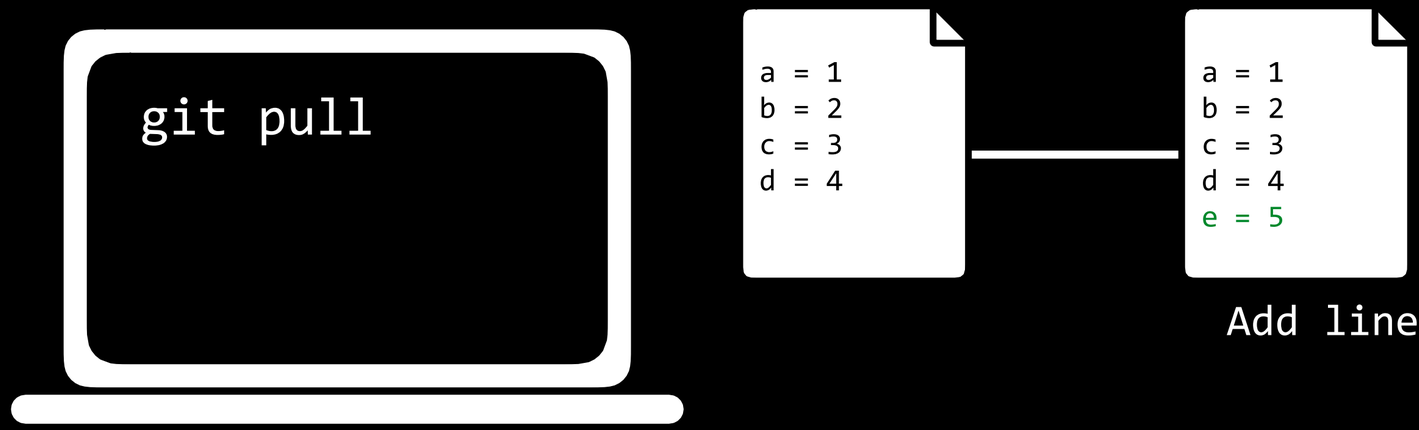

a = 1
b = 2
c = 3
d = 4
a = 1
b = 2
c = 3
d = 4
e = 5
a = 1
b = 2
c = 3
e = 5
Add line
Remove line


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

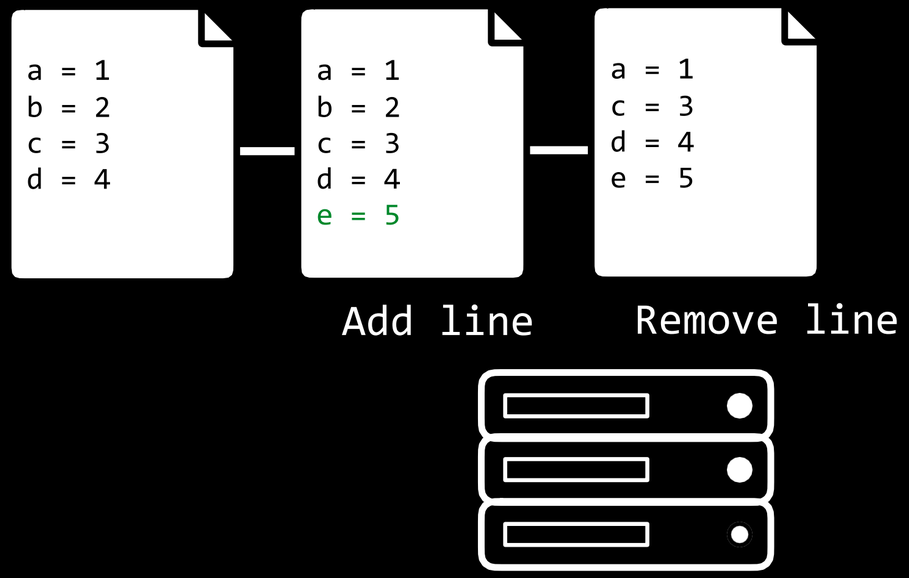
[1]

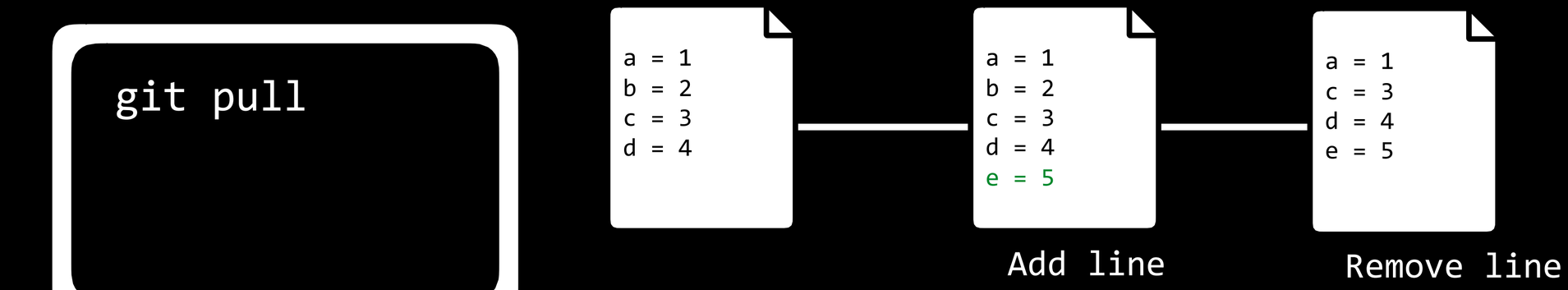
git pull


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
Remove line


git pull
a = 1
b = 2
c = 3
d = 4
a = 1
b = 2
c = 3
d = 4
e = 5
Add line

git pull


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
Remove line

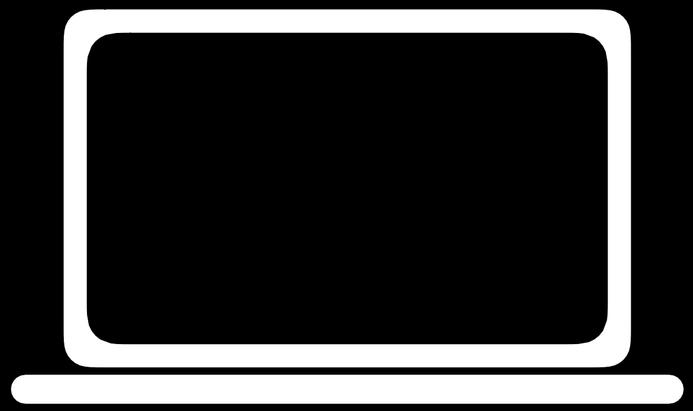

git pull
a = 1
b = 2
c = 3
d = 4
a = 1
b = 2
c = 3
d = 4
e = 5
a = 1
c = 3
d = 4
e = 5
Add line
Remove line

[1]

Merge Conflicts

[1]

Merge Conflicts



[1]

Merge Conflicts

[1]

Merge Conflicts


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

[1]

Merge Conflicts


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

[1]

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
```

[1]

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

git log

git log

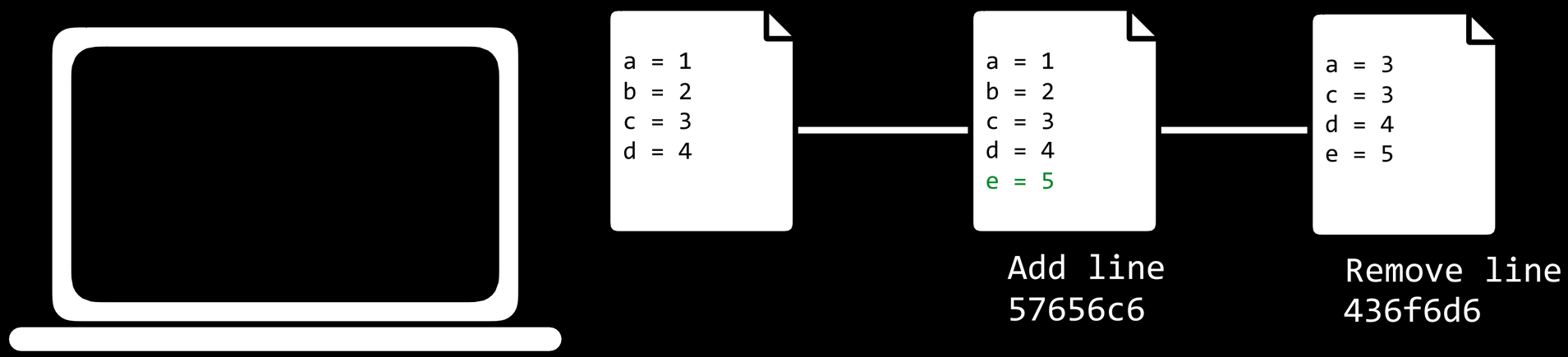

```
commit 436f6d6d6974204d73672048657265
Author: Brian Yu <brian@cs.harvard.edu>
Date:   Tue Jan 14 14:06:28 2020 -0400

    Remove a line

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

    Add a line
```

[1]

git reset

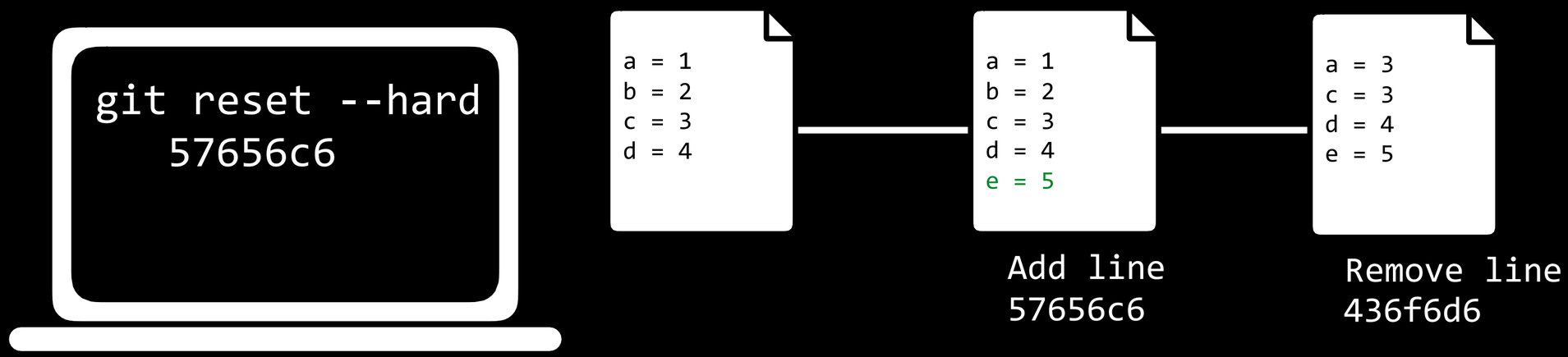
git reset

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



git reset

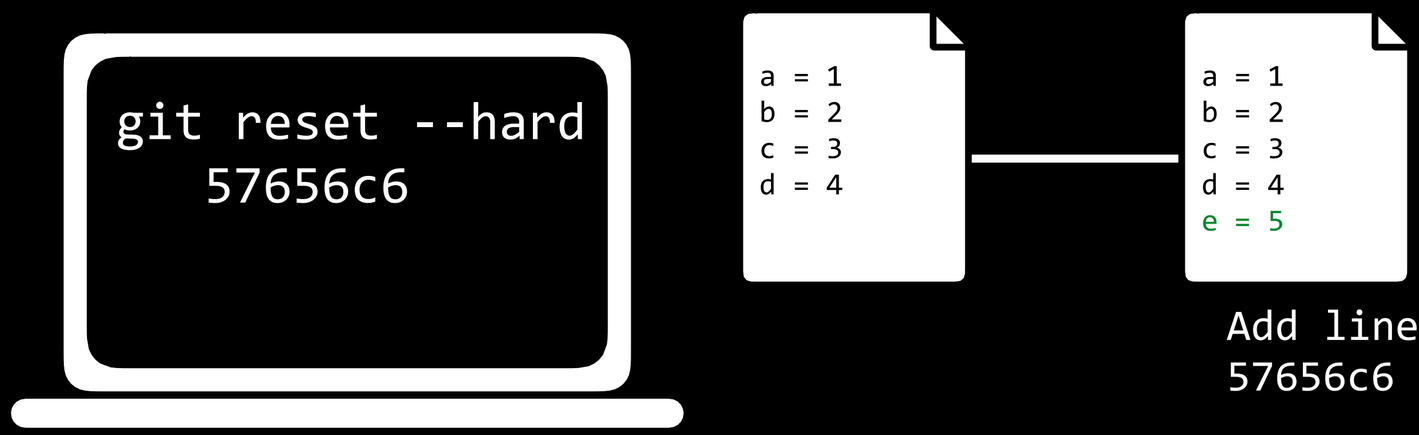
- git reset --hard <commit>
- git reset --hard origin/master



[1]

git reset

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



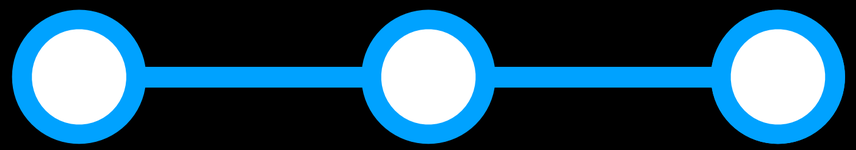
[1]

Making Changes

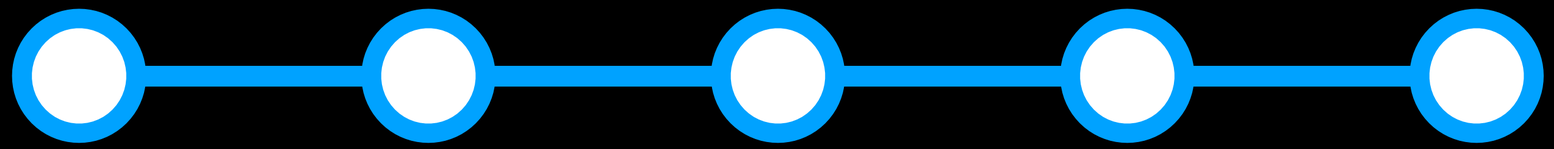
first
commit

first commit

changes


first
commit
changes
more
changes

[1]


first commit
changes
more changes
start new feature
keep working on new feature

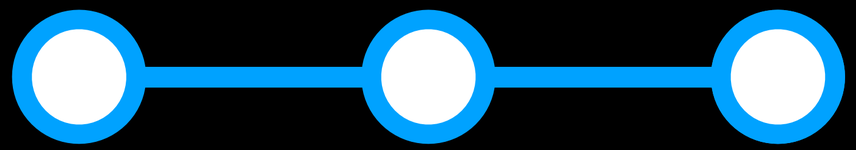
[1]

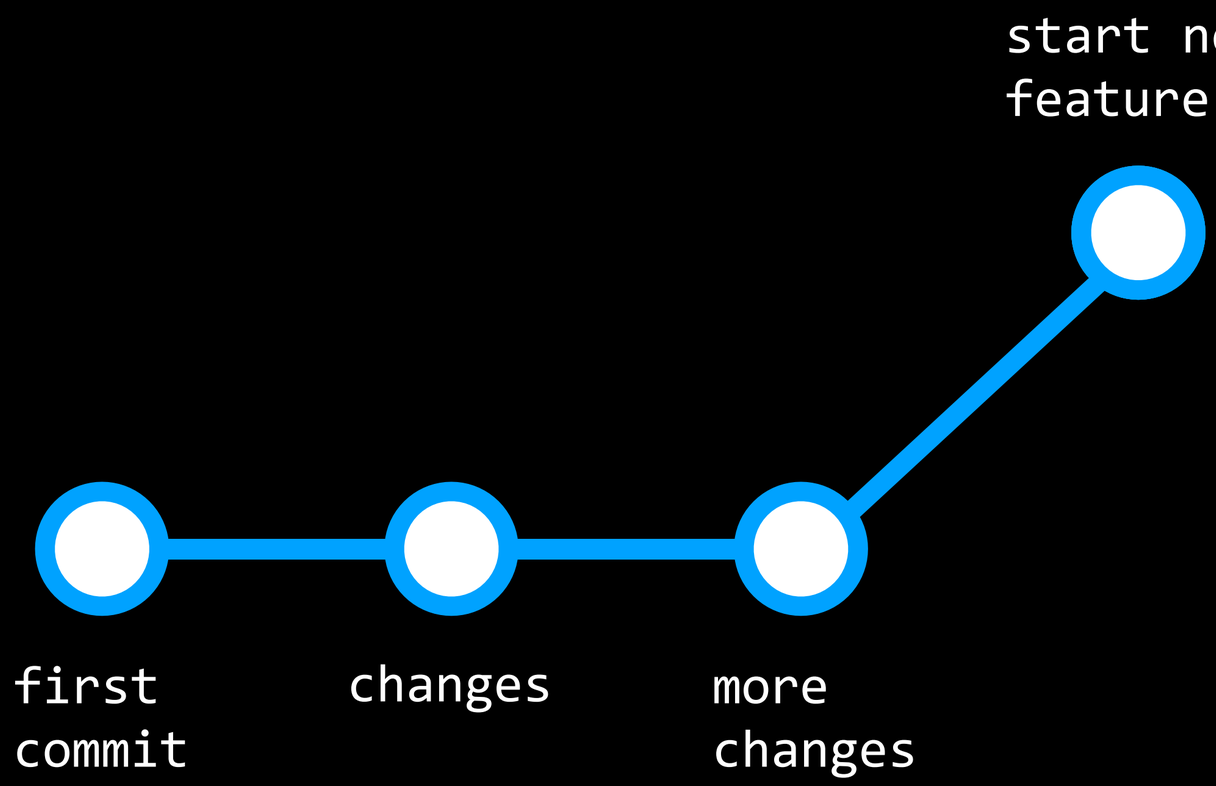
Branching

first
commit

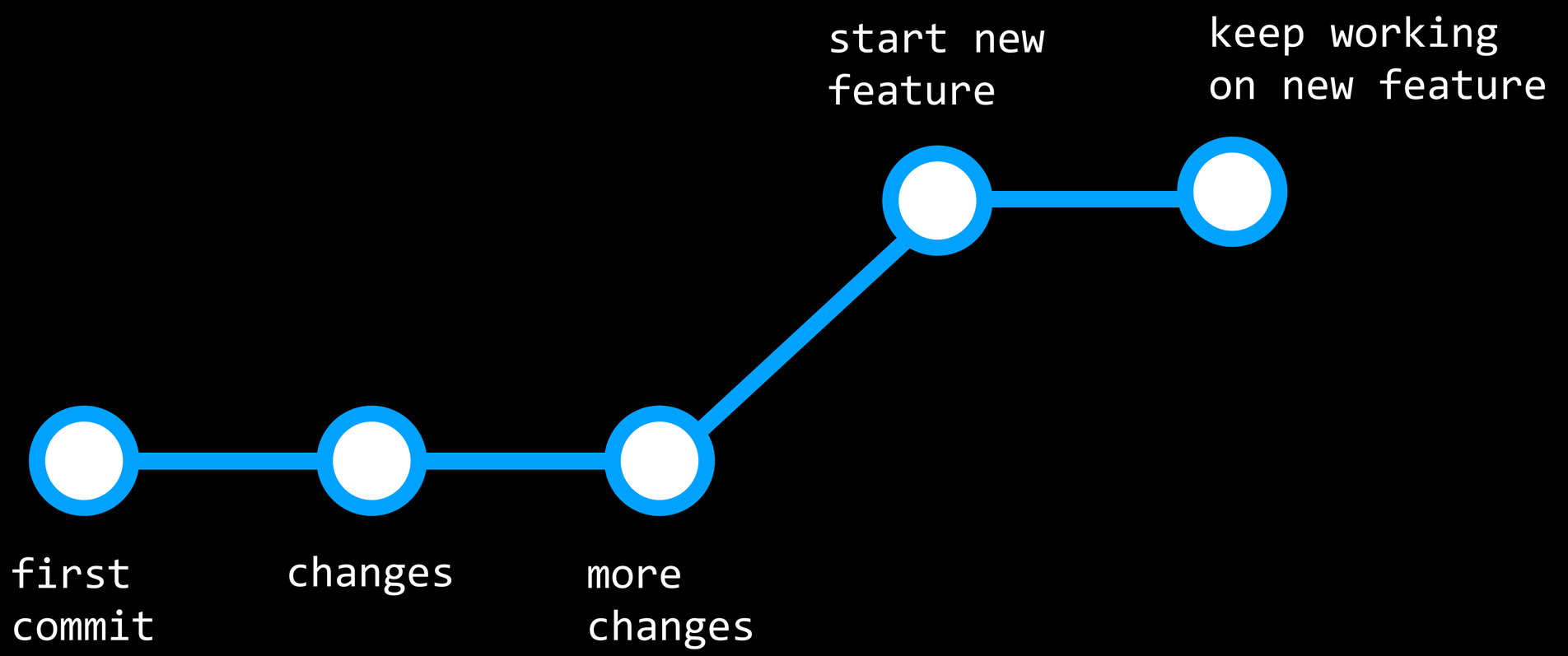
first commit

changes

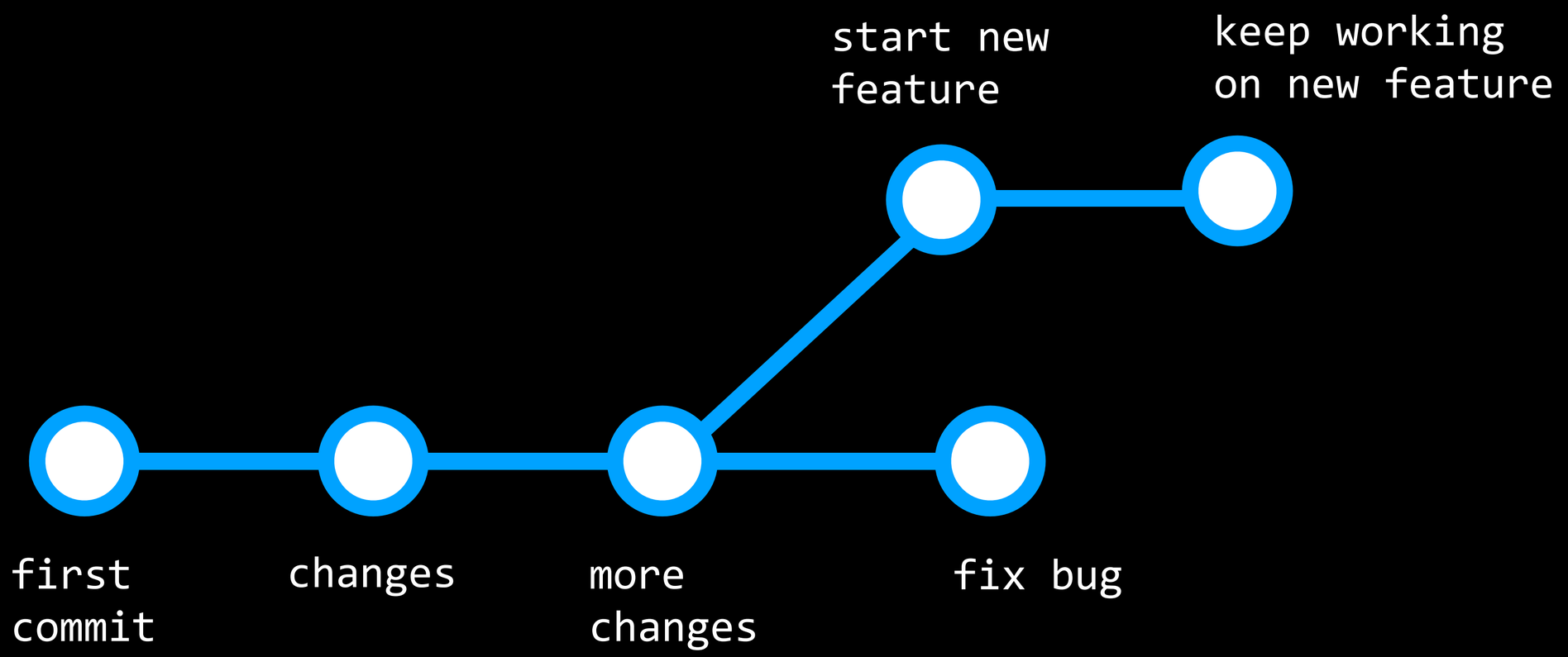

first
commit
changes
more
changes



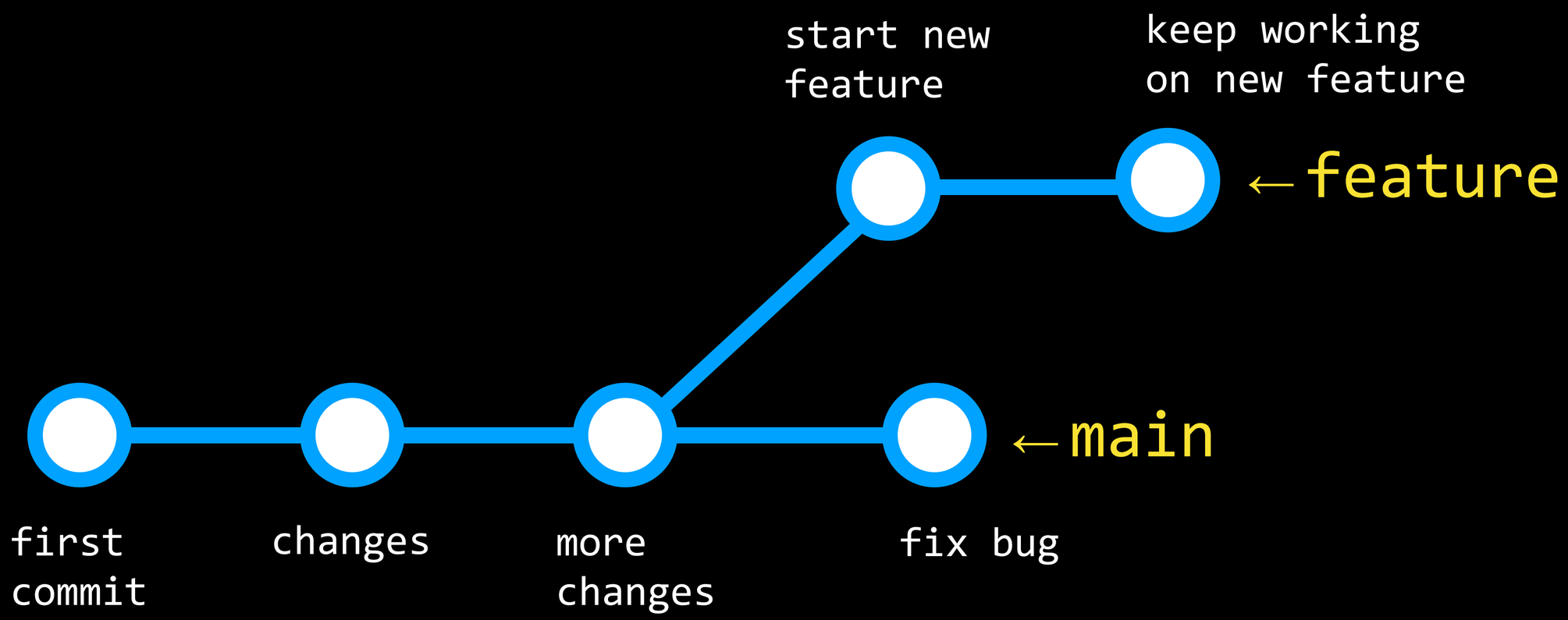
[1]


start new
feature
keep working
on new feature
first
commit
changes
more
changes

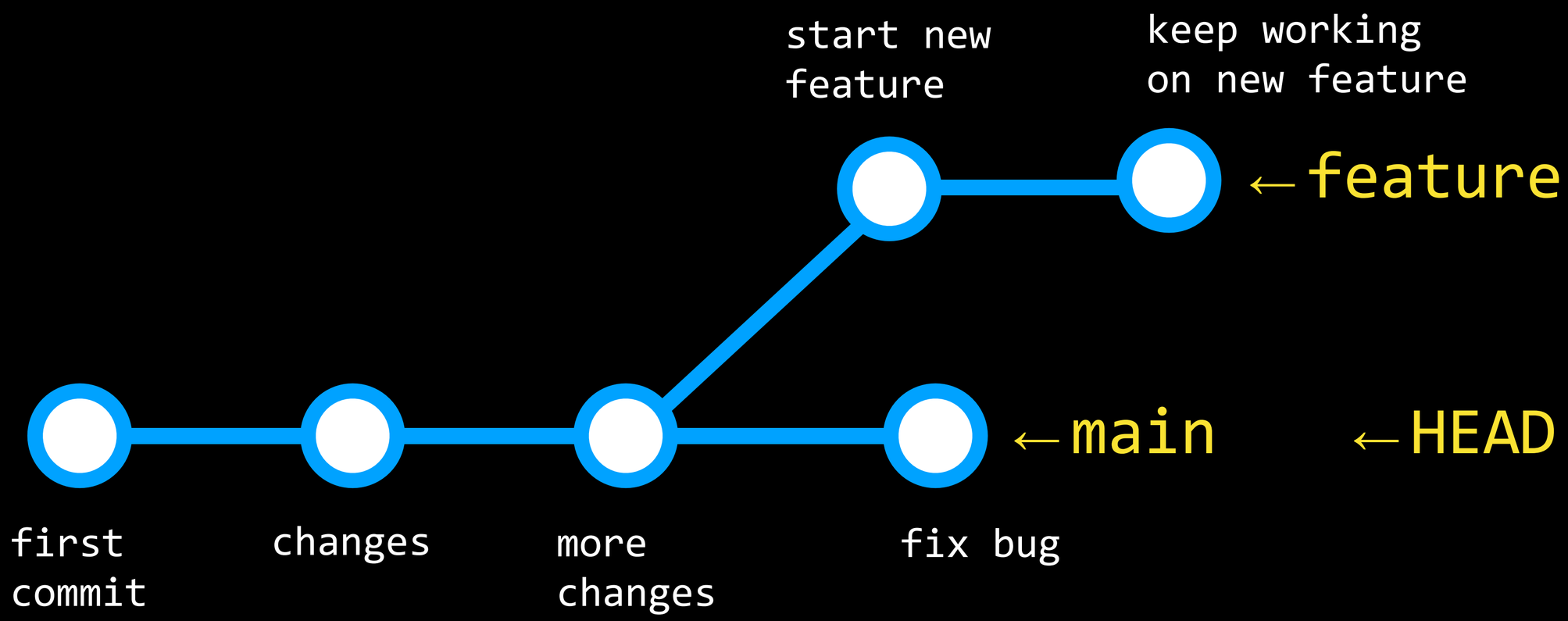
[1]


start new feature
keep working on new feature
first commit
changes
more changes
fix bug

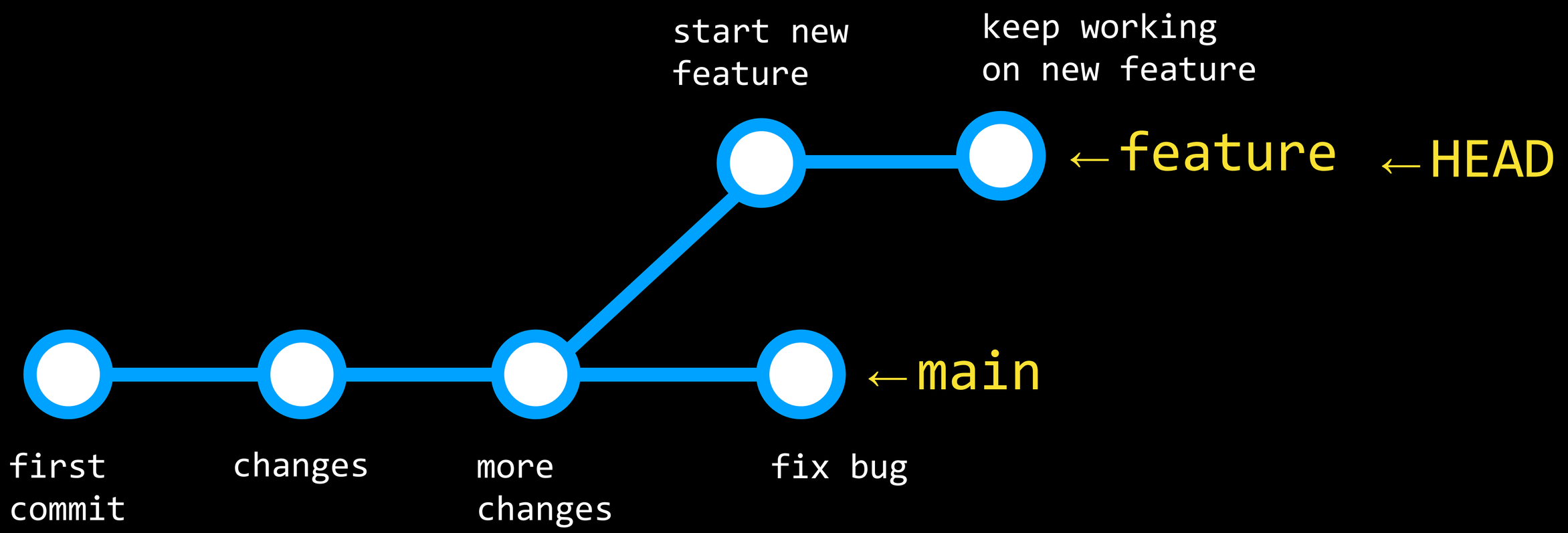
[1]


start new feature
keep working on new feature
←feature
←main
first commit
changes
more changes
fix bug

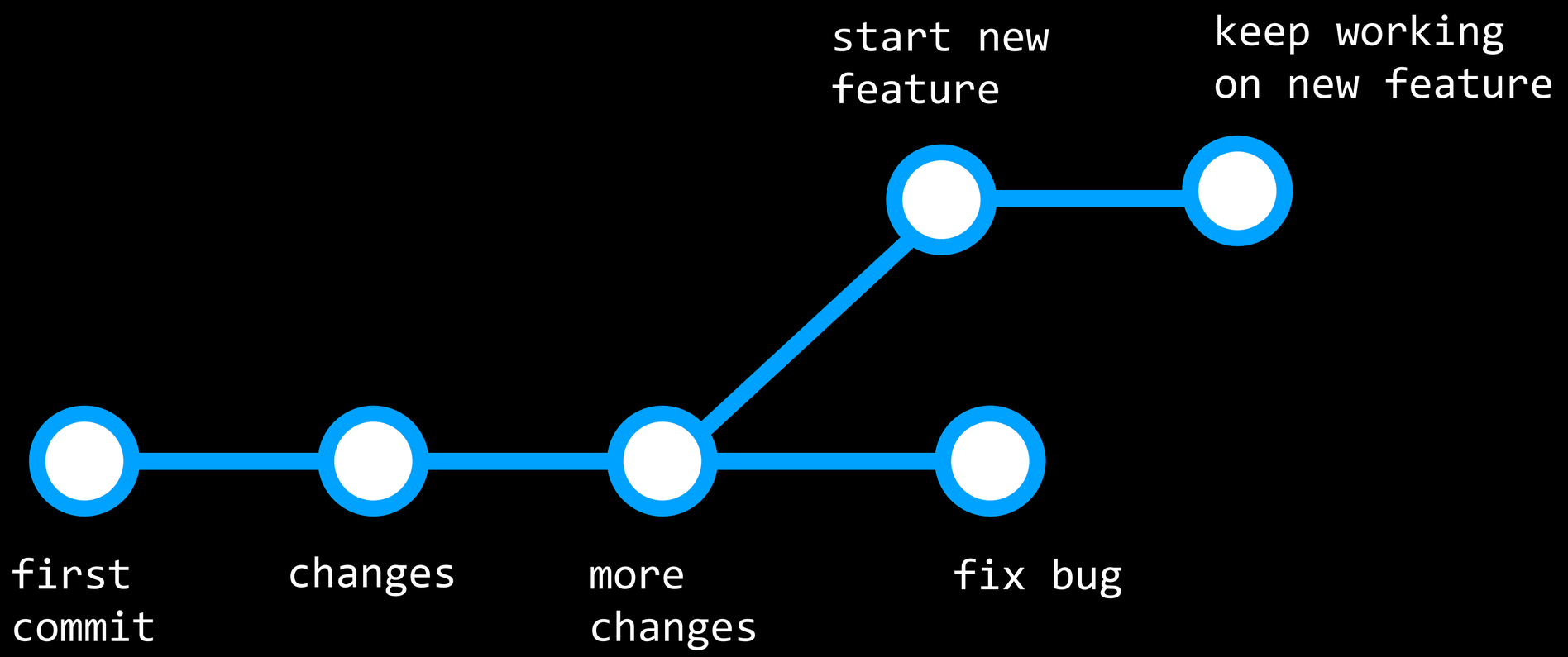
[1]


start new feature
keep working on new feature
←feature
←main
←HEAD
first commit
changes
more changes
fix bug

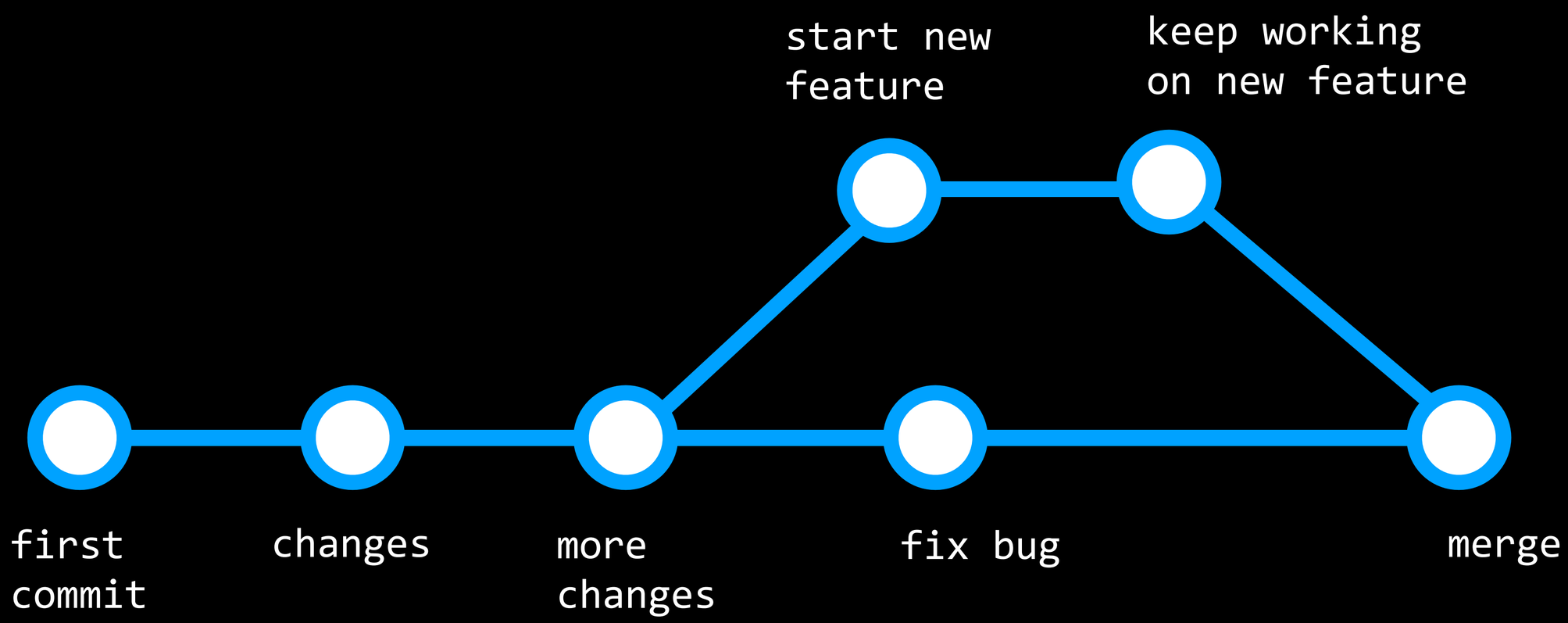
[1]


start new feature
keep working on new feature
←feature
←HEAD
←main
first commit
changes
more changes
fix bug

[1]


start new
feature
keep working
on new feature
first
commit
changes
more
changes
fix bug

[1]


start new feature
keep working on new feature
first commit
changes
more changes
fix bug
merge

[1]

Branching

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

[1]

Lizenz und Quellen

[2] https://www.wired.com/2012/02/github-2/

[3] https://git-scm.com/book/de/v2/Erste-Schritte-Was-ist-Versionsverwaltung%3F