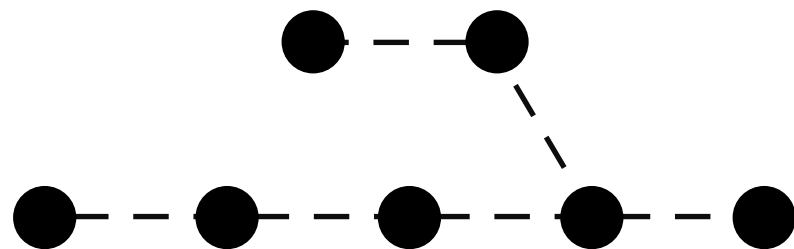


# Git workshop



# What is git ? (for a biologist)

a versioning system =  
a set of tools to track changes  
    \* in a directory...  
    \* or across computers...

a command-line tool (but has interfaces)

Why use git ?

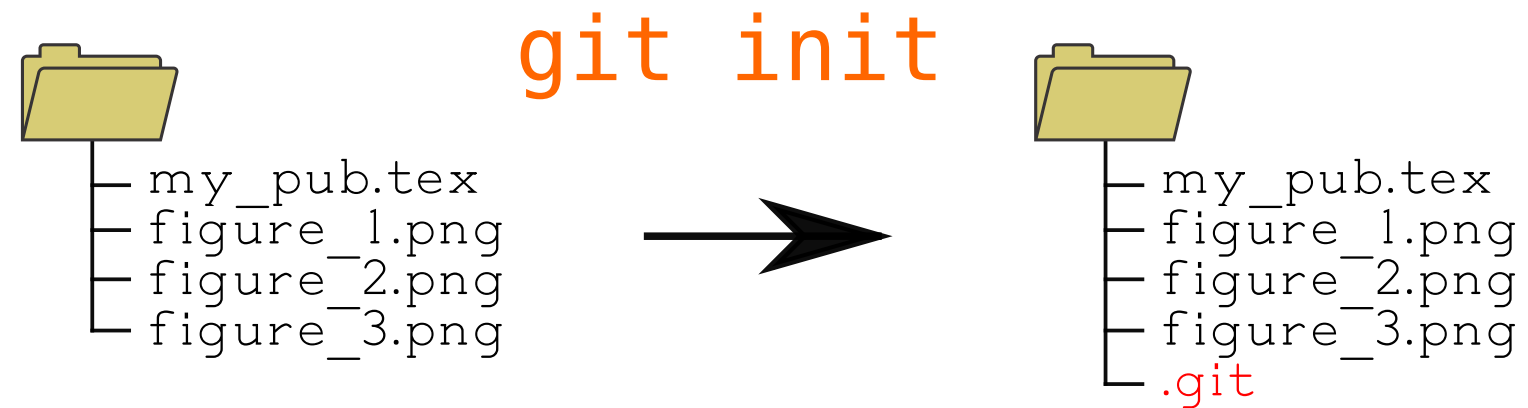
to keep track of work on a project

to work with collaborators

# git essentials

(you don't need to know everything !)

- a repository = a tracked folder



- `git status`

# git essentials - the commit loop

(you don't need to know everything !)

- a commit = a set of changes  
= a set of addition/deletion/modifs of files



**git add .** (add all files in the current directory)

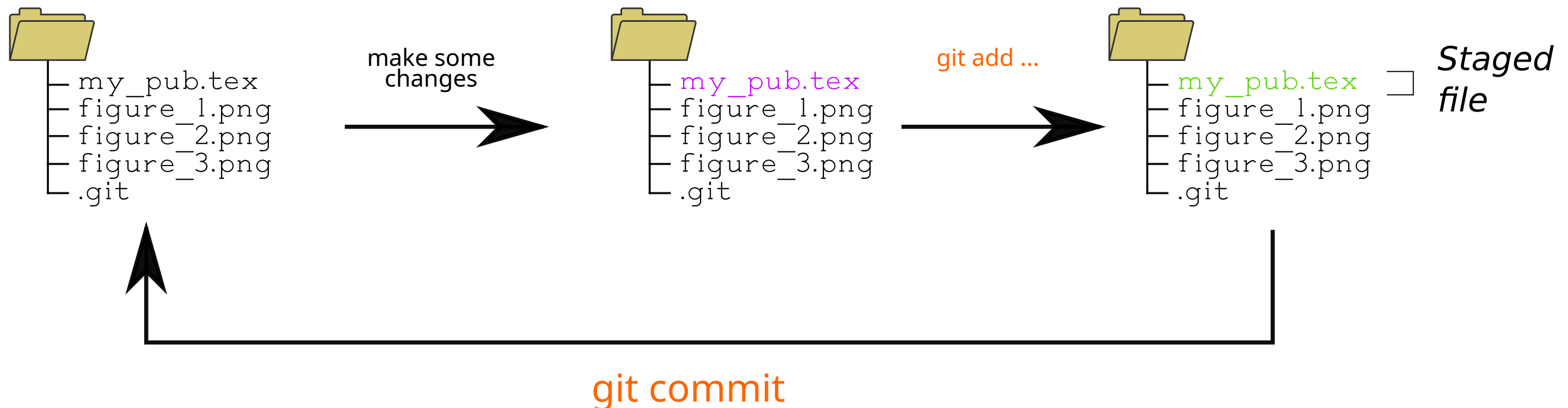
**git add my\_pub.tex figure\_1.png figure\_2.png figure\_3.png** (add specific files)

(then **git status** !)  
(then **git commit** !)

# git essentials - the commit loop

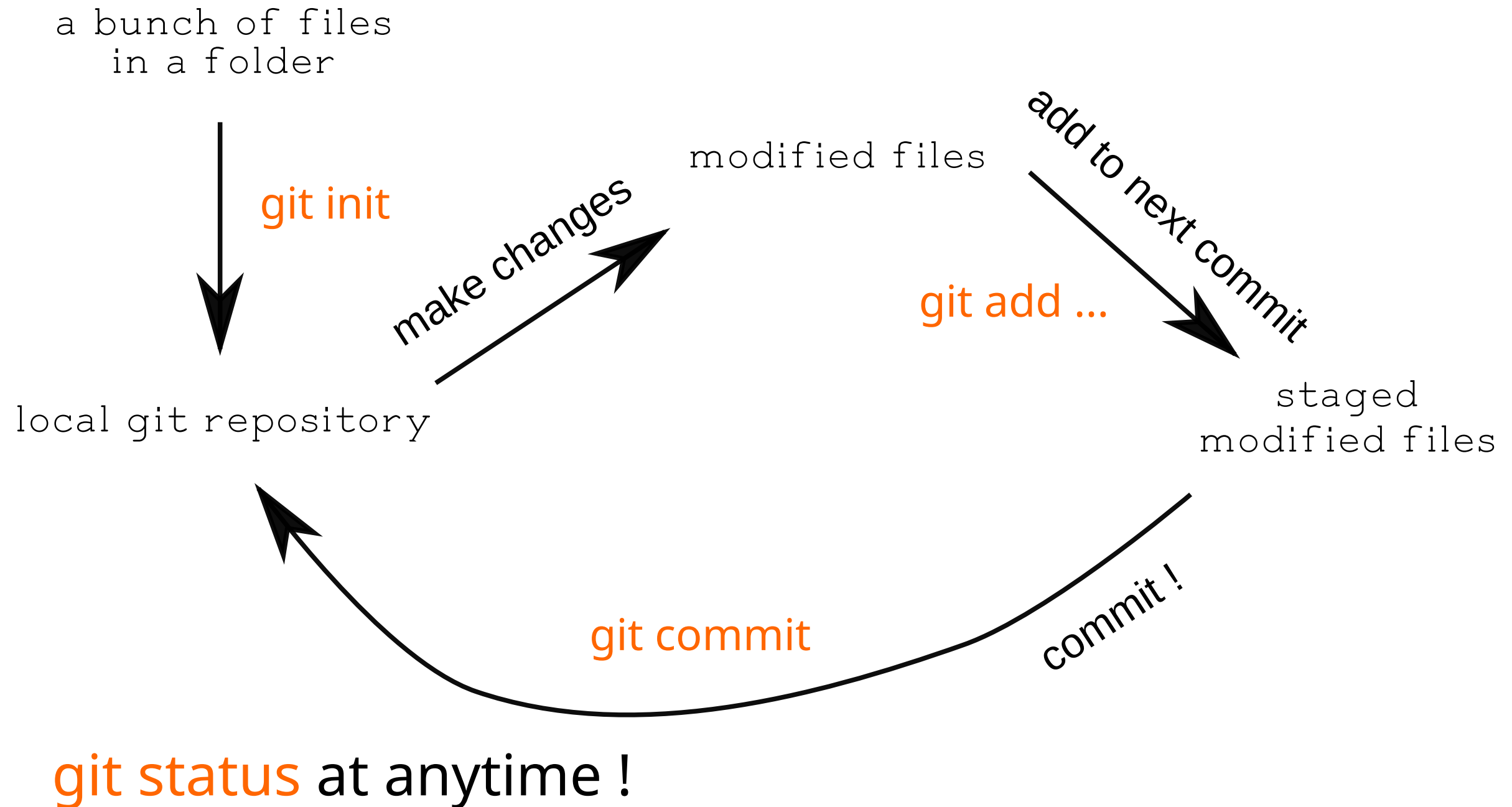
(you don't need to know everything !)

- a commit = a set of changes  
= a set of addition/deletion/modifs of files



git status at anytime !

# tracking changes locally



# tracking changes locally

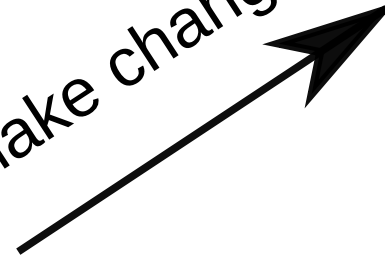
a bunch of files  
in a folder

**git init**



local git repository

make changes



modified files

**git add ...**

add to next commit

staged  
modified files

**git commit**

commit !

**git status** at anytime !

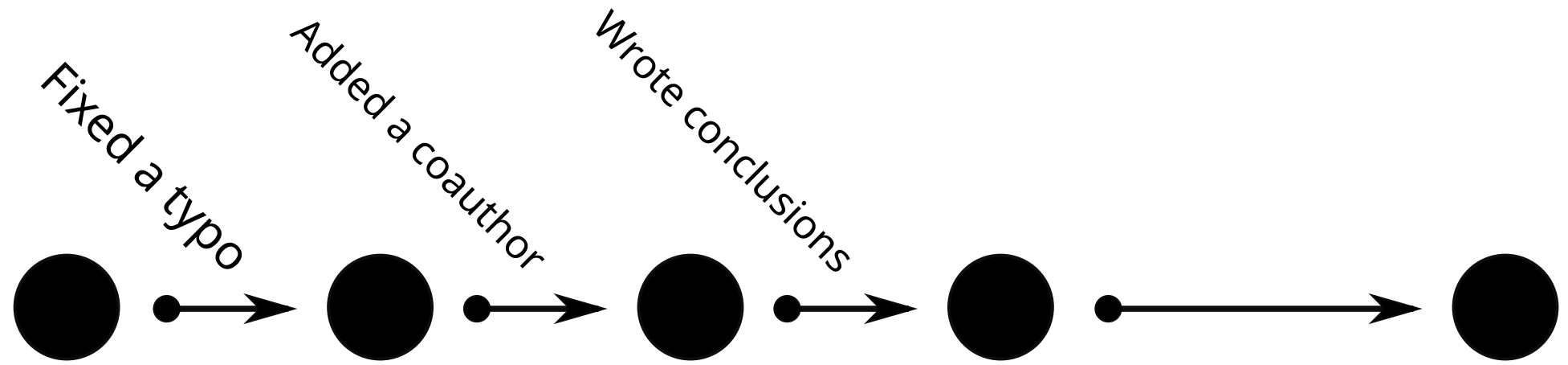
your turn !



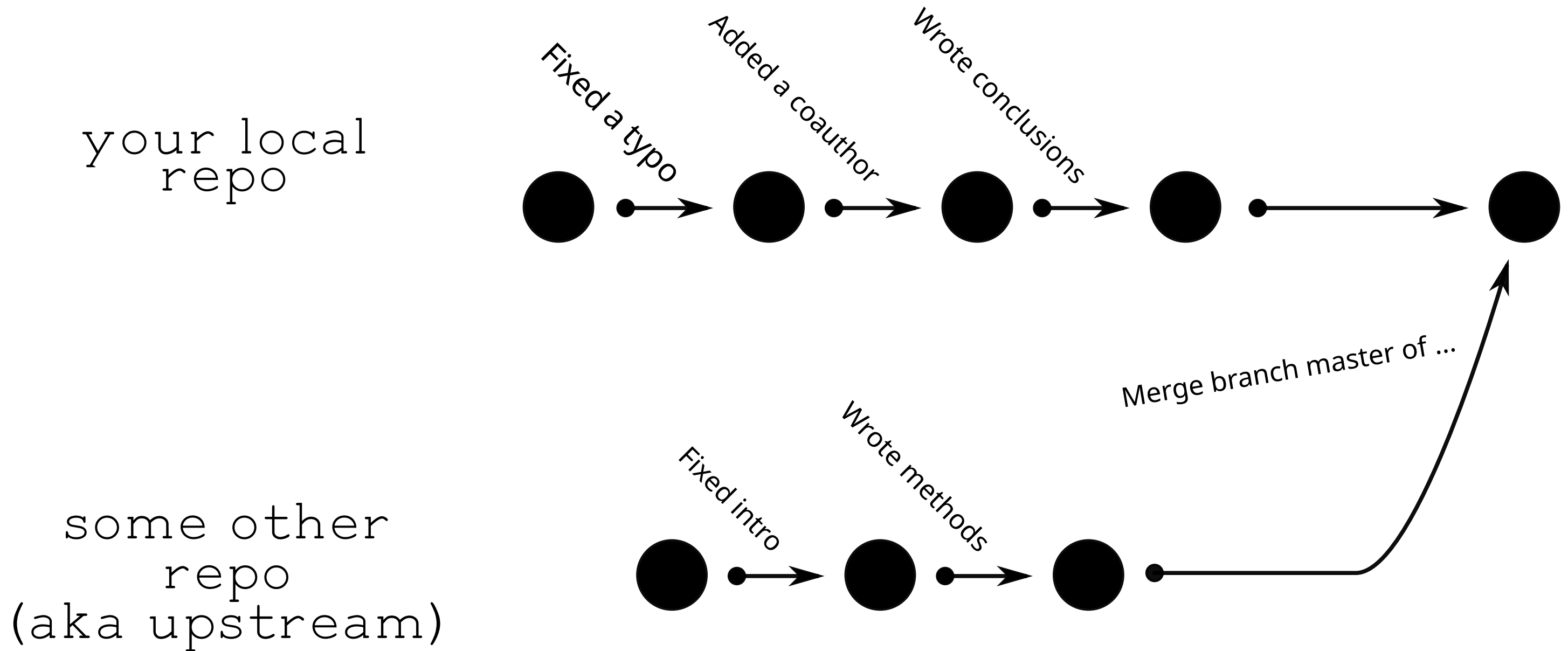


# tracking changes across repos

your local  
repo



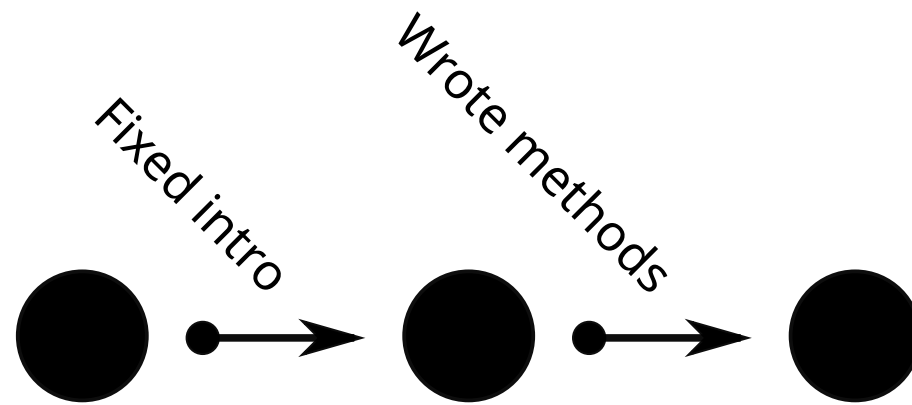
# tracking changes across repos



# tracking changes across repos

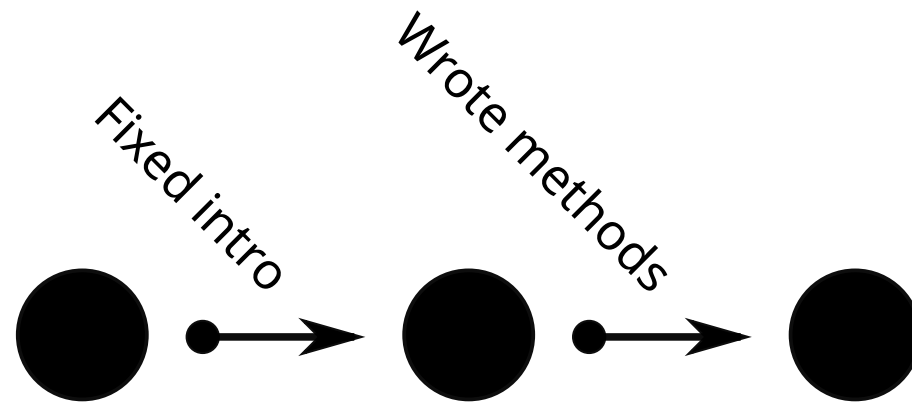
your local  
repo

some other  
repo  
(aka upstream)



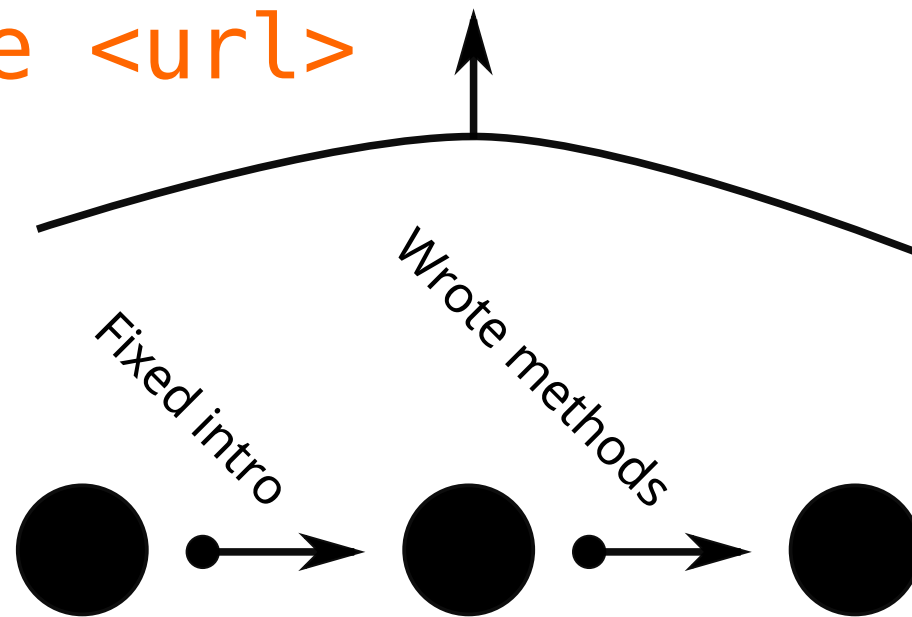
# tracking changes across repos

your local  
repo

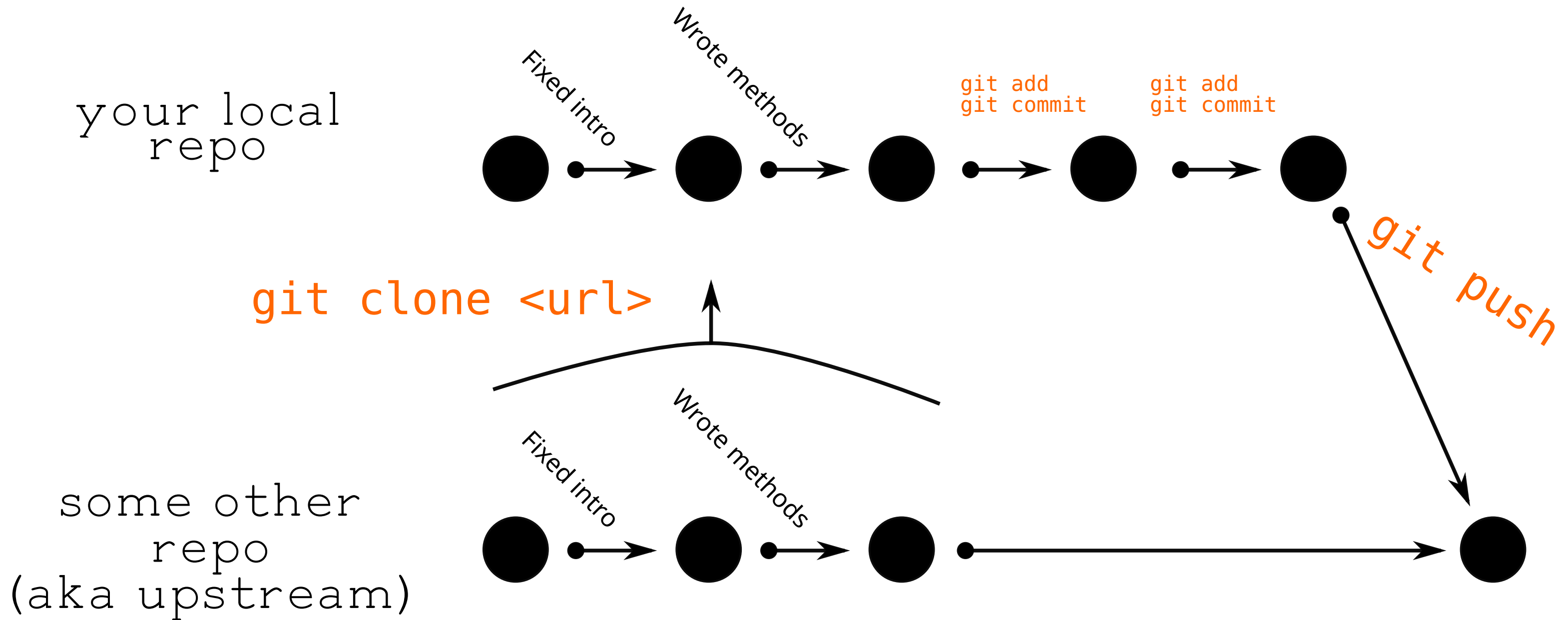


git clone <url>

some other  
repo  
(aka upstream)

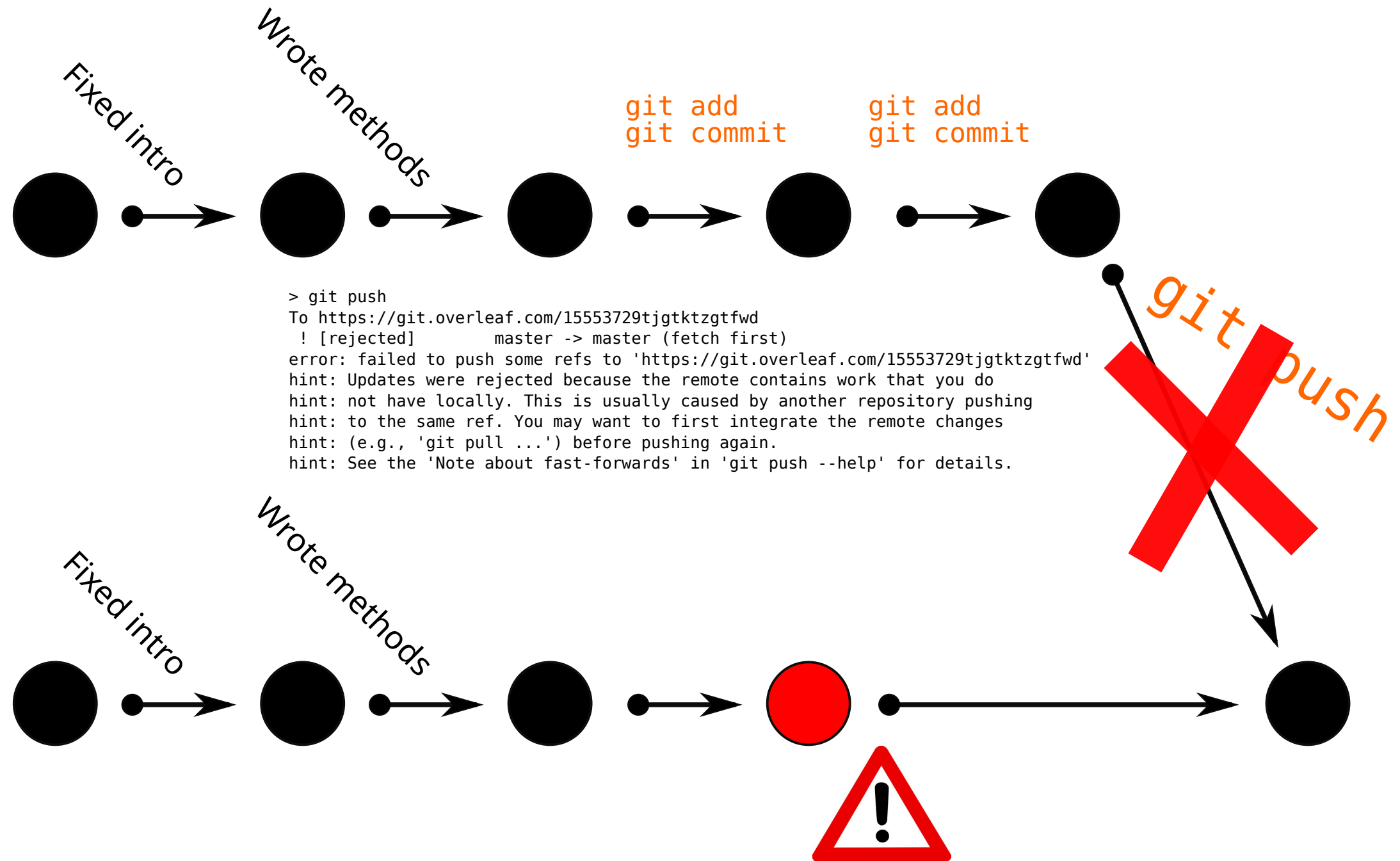


# tracking changes across repos



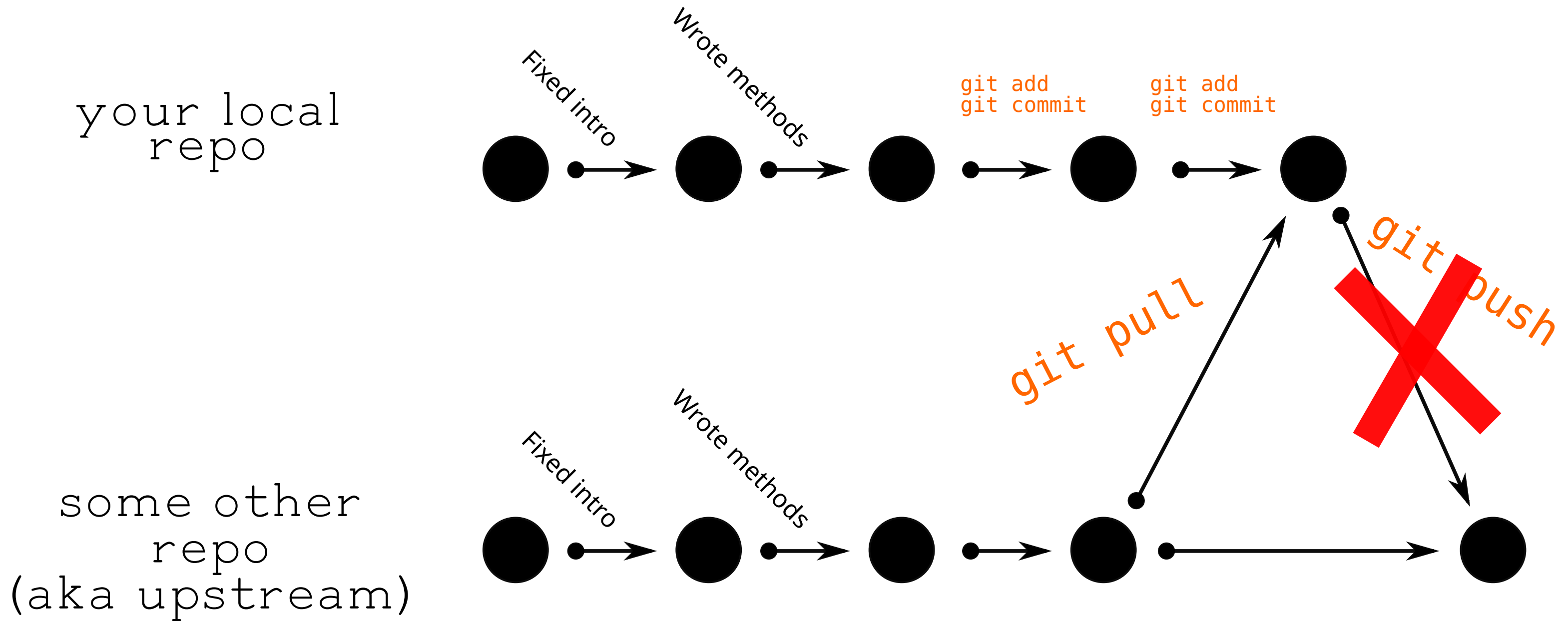
# tracking changes across repos

your local  
repo

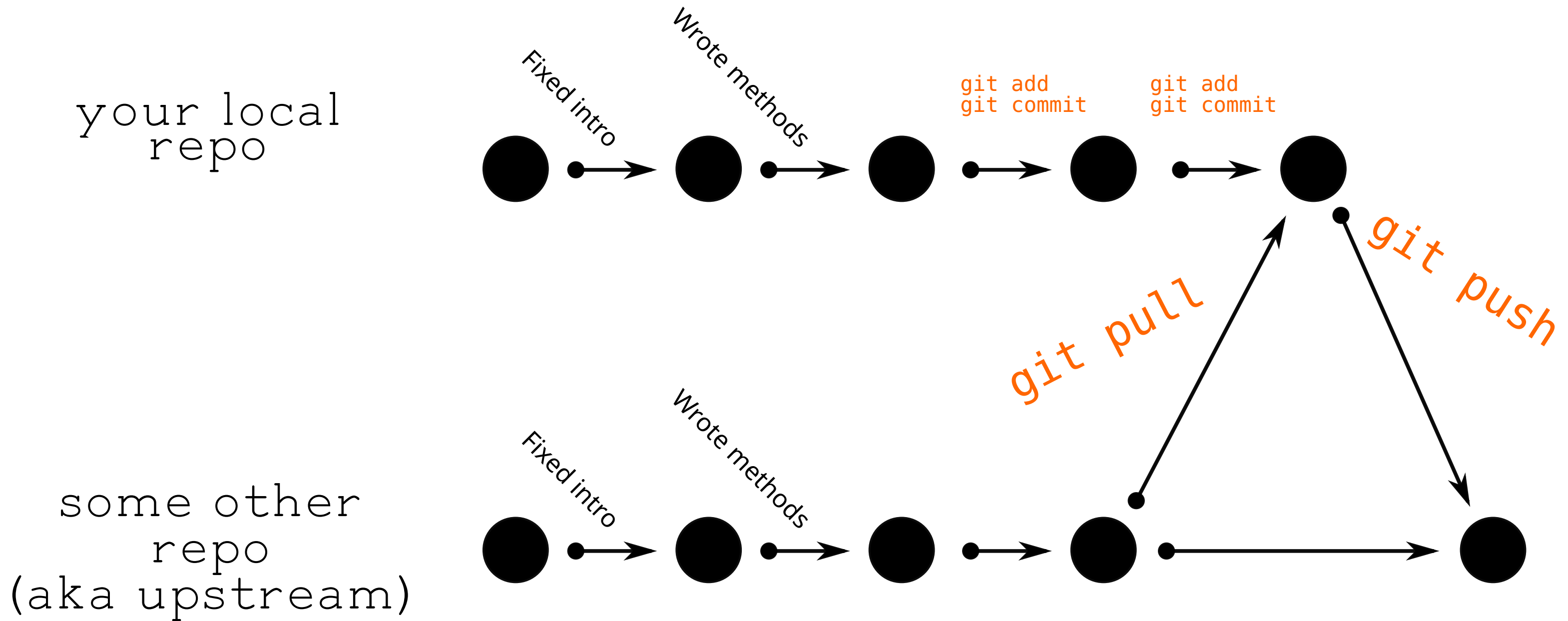


some other  
repo  
(aka upstream)

# tracking changes across repos



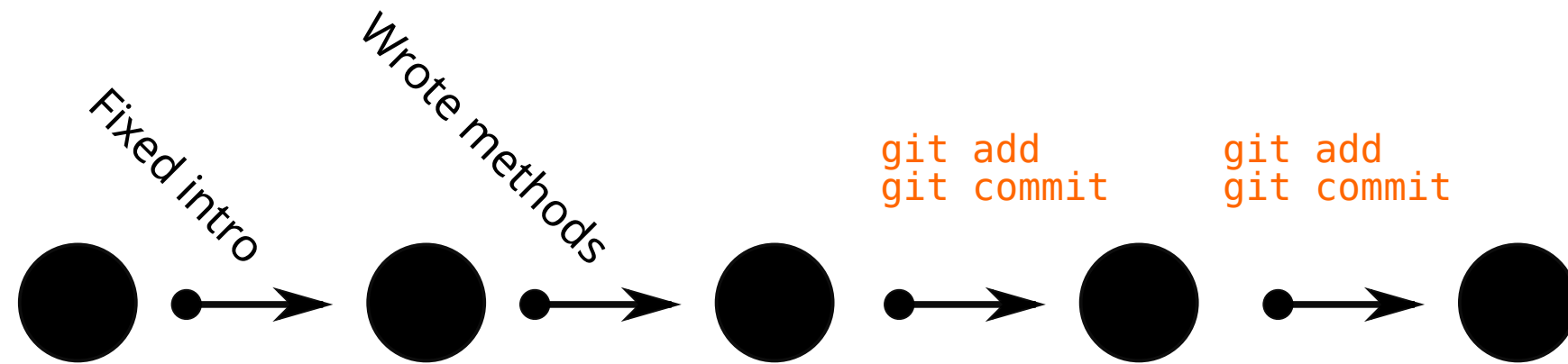
# tracking changes across repos





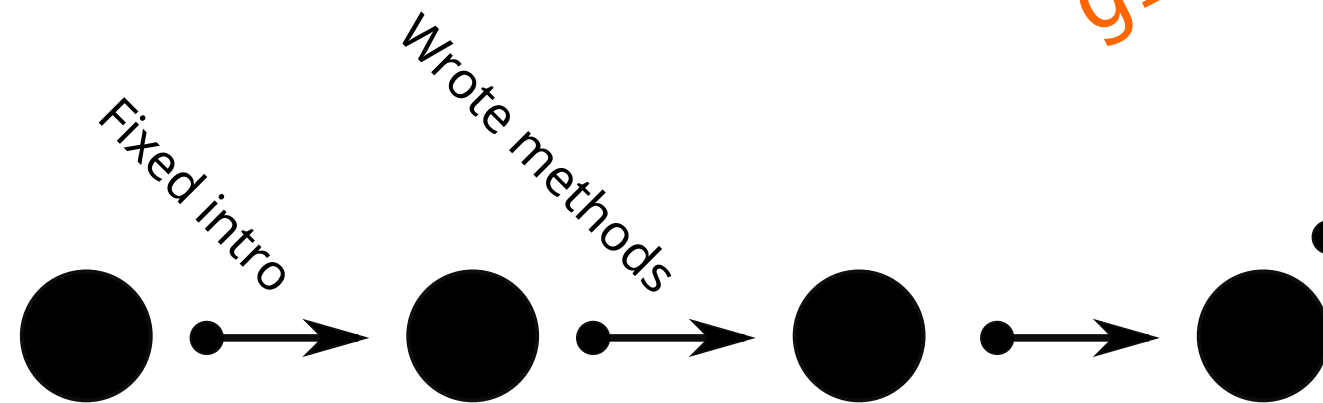
# tracking changes across repos

your local  
repo



```
From https://git.overleaf.com/15553729tjgkztzgtfwd
 caefe81..599757b  master    -> origin/master
Auto-merging main.tex
CONFLICT (content): Merge conflict in main.tex
Automatic merge failed; fix conflicts and then commit the result.
```

some other  
repo  
(aka upstream)



git pull

# tracking changes across repos

[...]

<<<<<<< HEAD

% Add the actual text below


=====

% Add the actual text below (use your imagination)


>>>>>>> 599757bbe3dd97df3ee210c8da56322cc531f119

[...]

What you have



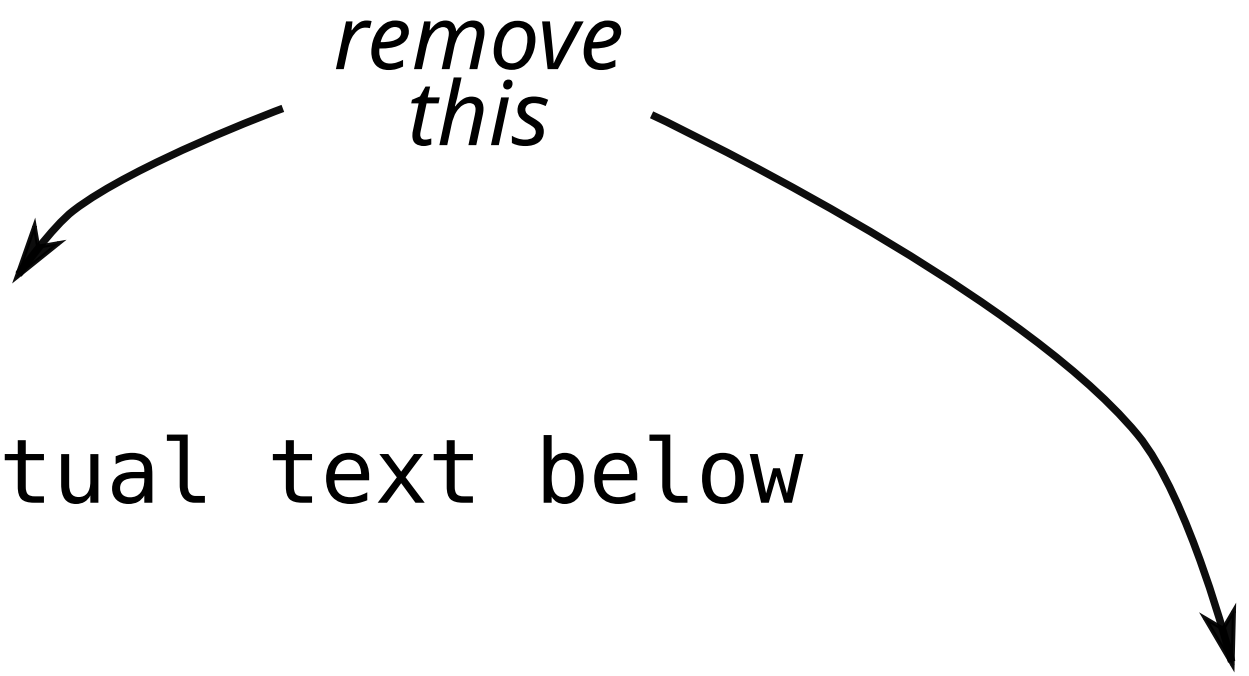
What they have



# tracking changes across repos

*remove  
this*

```
[...]  
<<<<<<< HEAD  
% Add the actual text below  
=====  
% Add the actual text below (use your imagination)  
>>>>>>> 599757bbe3dd97df3ee210c8da56322cc531f119  
[...]
```



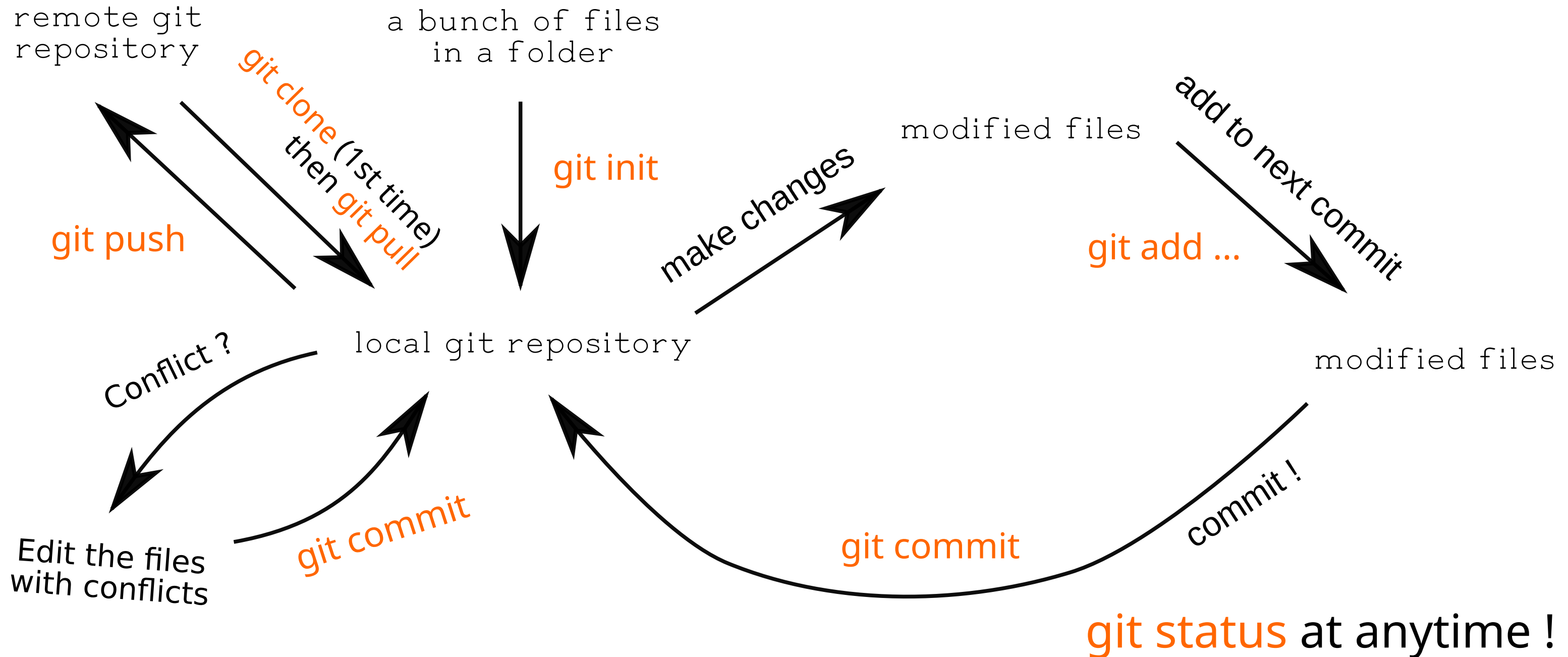
then **git commit** !

git cannot solve conflicts  
with non-text files!

- word docs
- most image files
- some data files (e.g. .Rdata)
- ...

=> You can still use version tracking,  
but conflicts will have to be resolved  
by hand

# tracking changes everywhere



# Graphical interfaces

git instaweb (view only, all)

Rstudio (all)

gitg (linux)

github client (all ?)

...