Git & GitHub

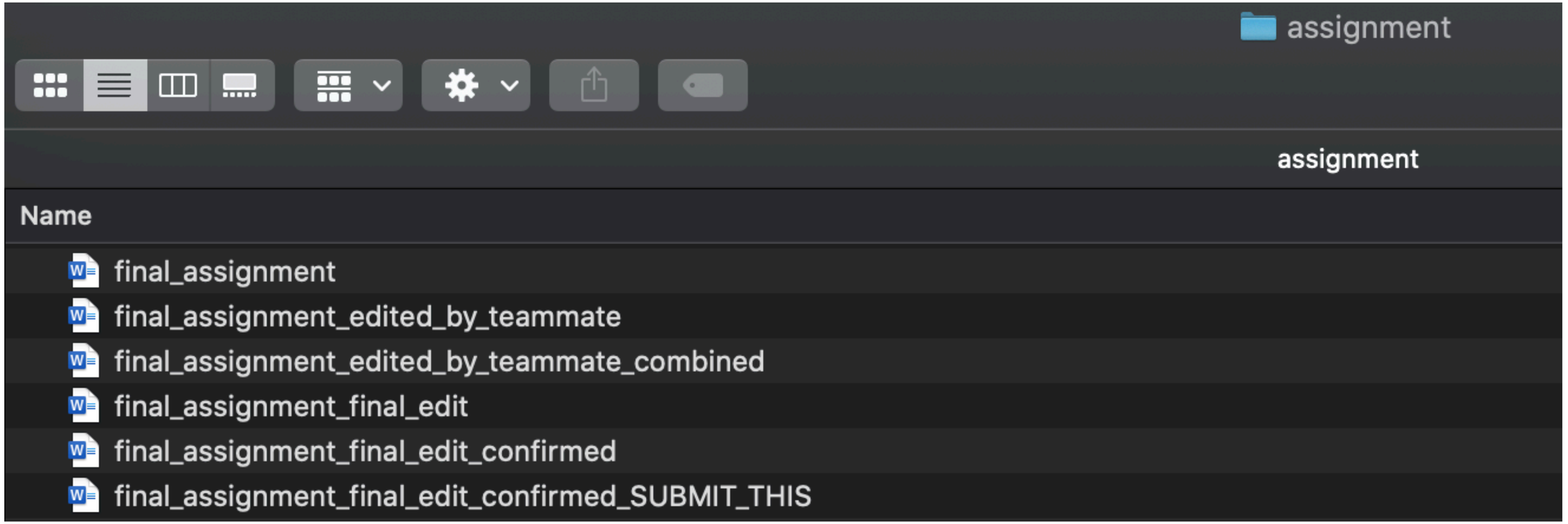
3DC - Early Mat Workshop Series

https://3dc.opensutd.org/Introduction_to_Programming/

What is Git and why is it important?

Standard version control system used by most developers and companies

What is Git and why is it important?



What is Git and why is it important?

Version control system

“Saves your game progress midway”

Host git repositories

“Google drive to store git repositories”

How does git work?

Legend

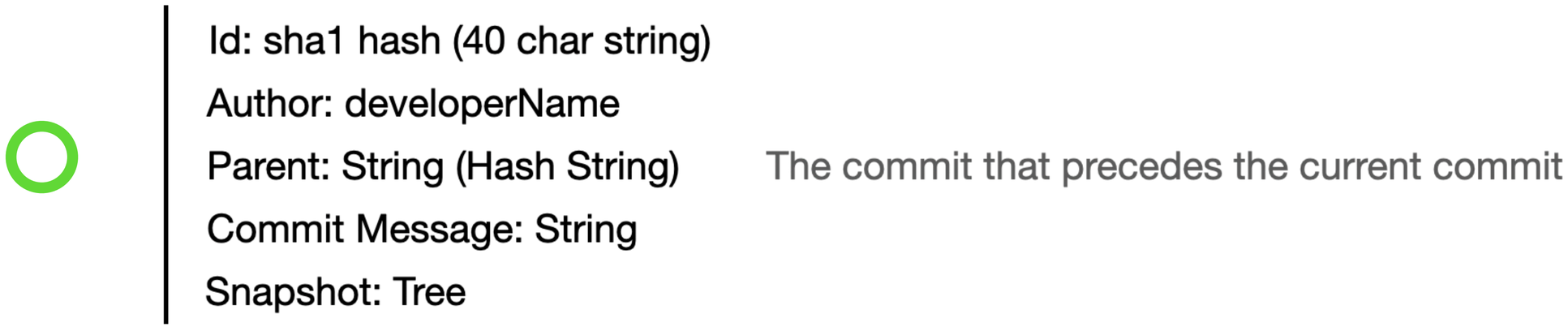
Snapshot

Pointer to parent snapshot

Git data model

Common terminology	Git terminology
Snapshot	Commit
Files	Blob
Folders	Tree

Git data model



Key takeaway

- Every commit has information of the current state of the files.
- It can be identified by a unique ID.
- It contains a commit message that describes the current state.
  - (eg. Added new feature, fixed a bug)

Git key concepts

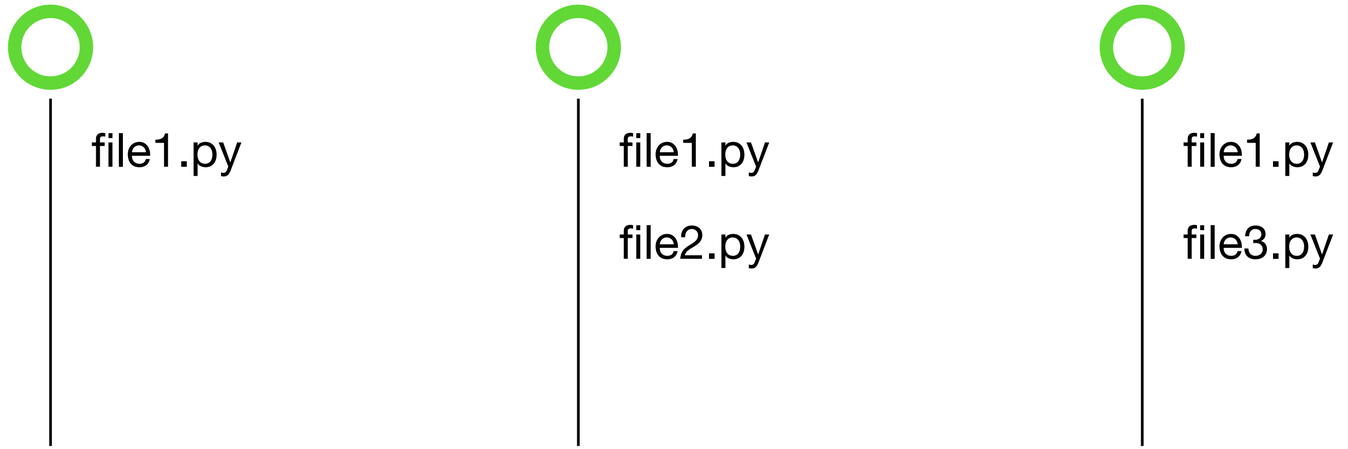
Staging Area

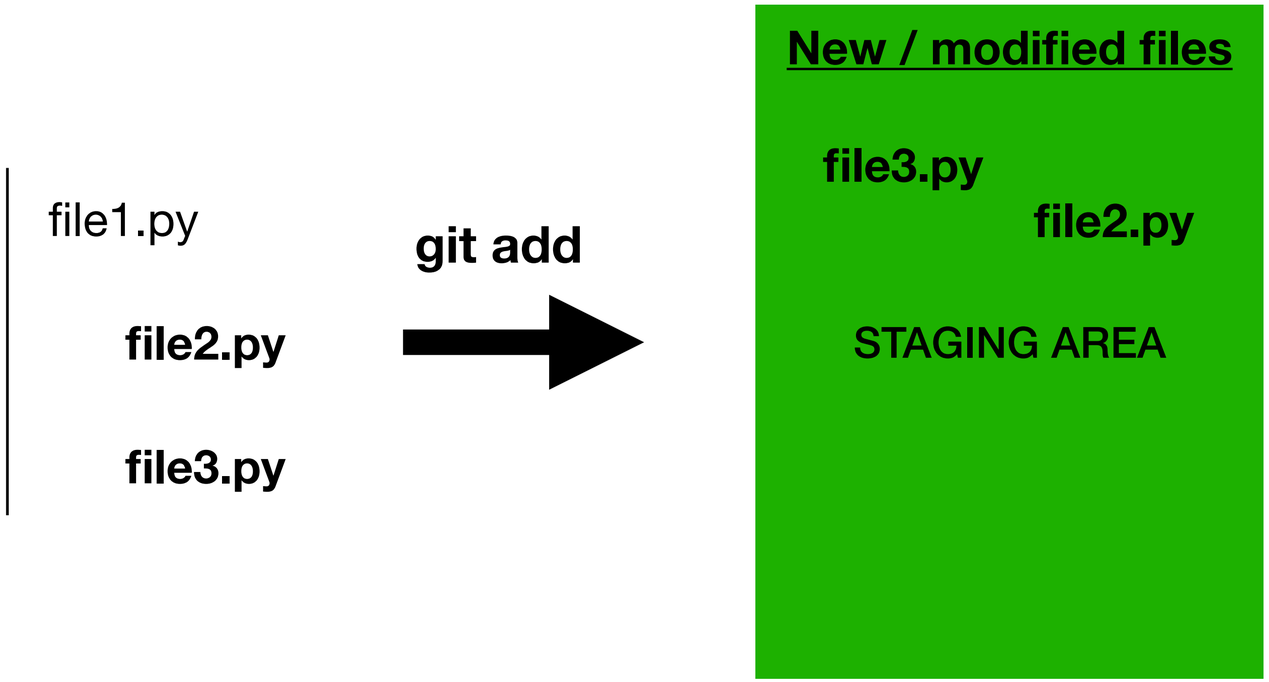
We add files into the staging area to tell git that those are the specific files that I want to create a snapshot / commit of.

Git key concepts

Staging Area

We want to create commits like this





By being able to choose what files goes into the commit, we can segment our work.

Git commands

Staging Area

git add .

git add <filename>

Git key concepts

Commit

Git commands

Staging Area

git add .

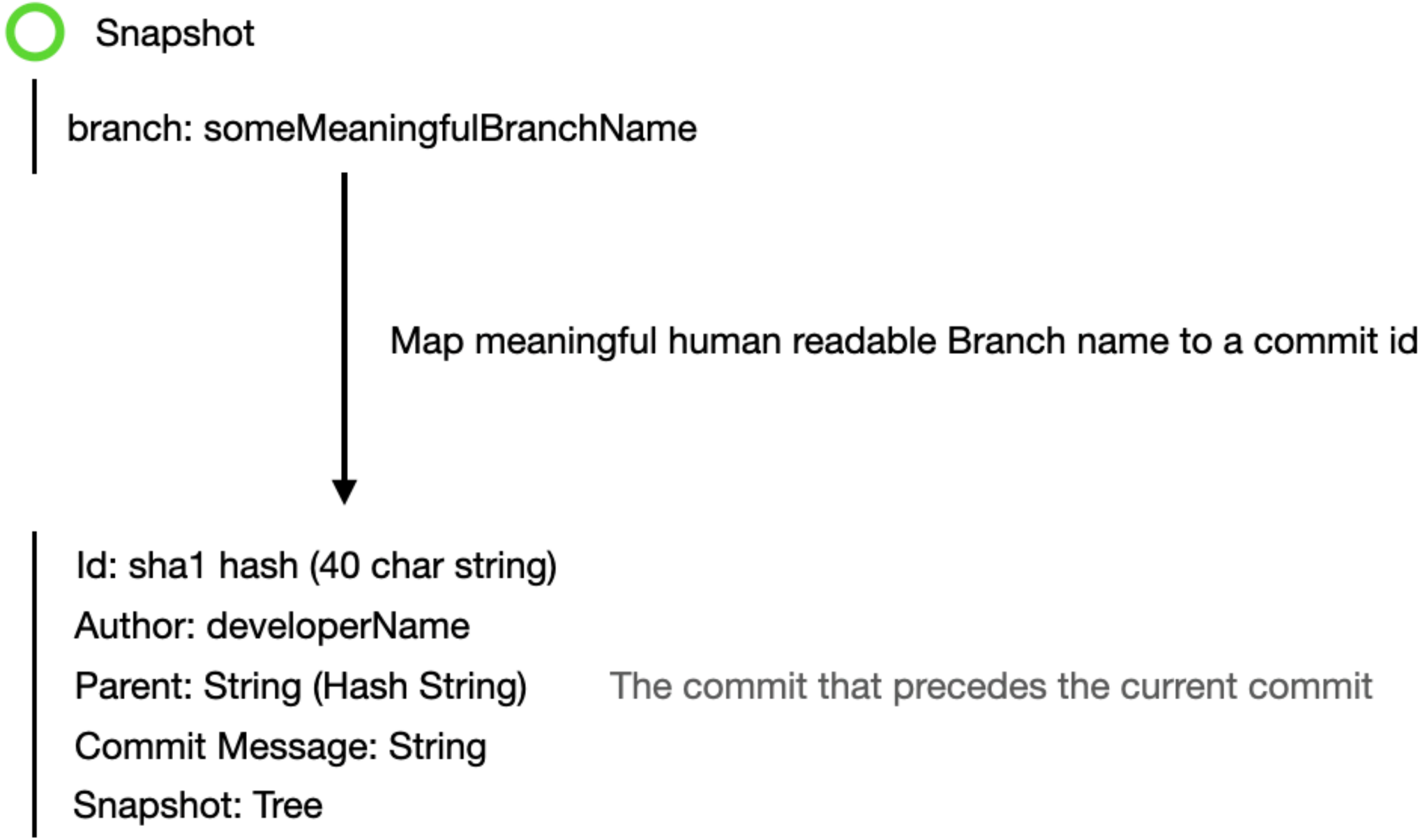
git add <filename>

Commit

git commit -m “message”

Git key concepts

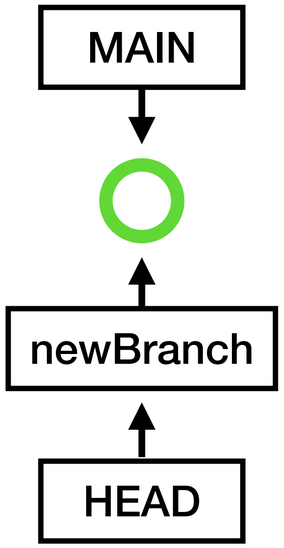
Branches



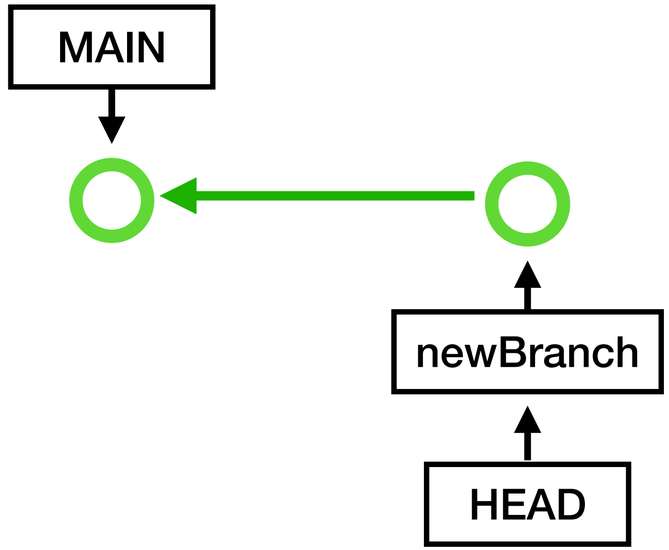
Branches

HEAD branch points to the current branch you are on.

Main / master branch is the default initial branch name. Can be seen as the main line of development where production code lives.




```
Make some changes
Git add .
Git commit -m “1st change”
```



Git commands

Staging Area

git add .

git add <filename>

Commit

git commit -m “message”

Branches

git branch

git checkout -b <new branch name>

Git key concepts

Merging

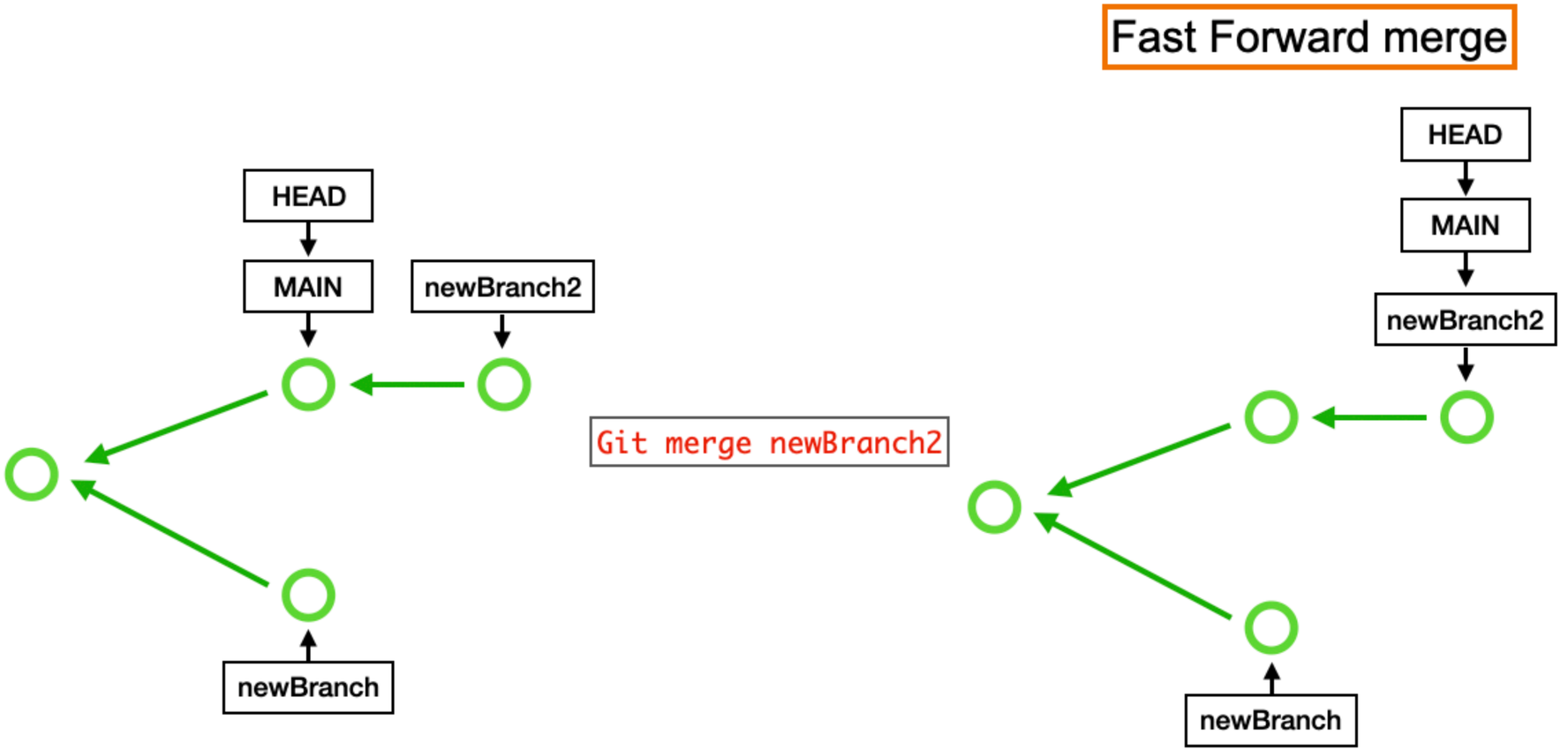
fast-forward merge

3 way merge

Git key concepts

Merging

fast-forward merge

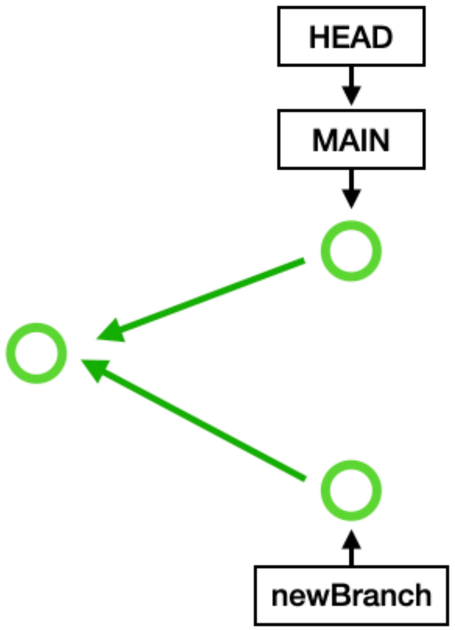


Git key concepts

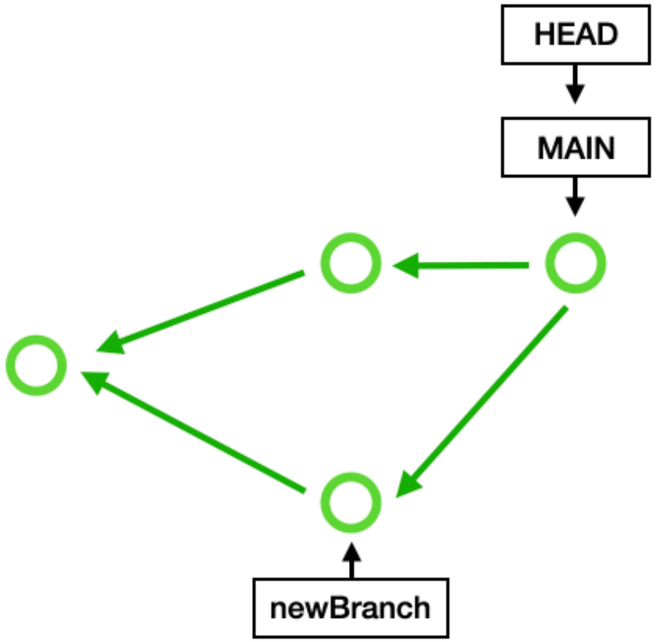
Merging

3 way merge

Three way merge

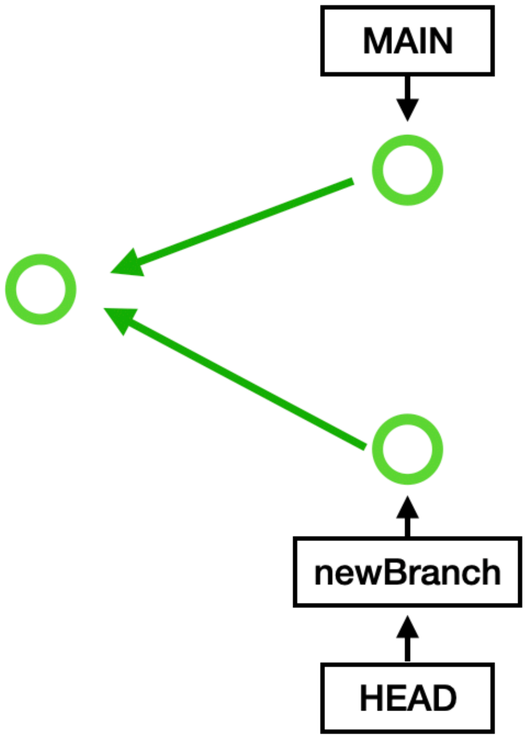



```
Git merge newBranch
```



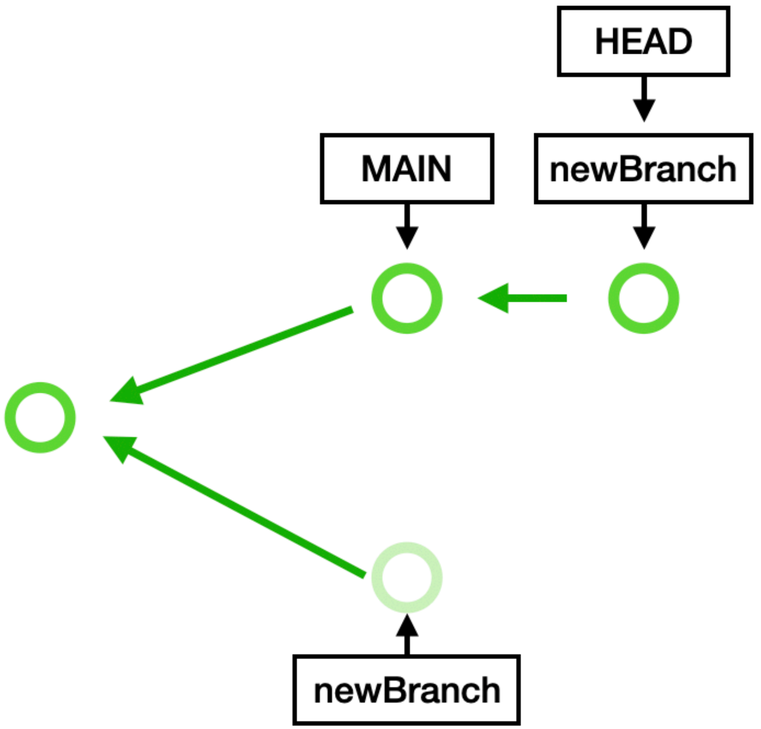
Git key concepts

Merging



Git rebase main

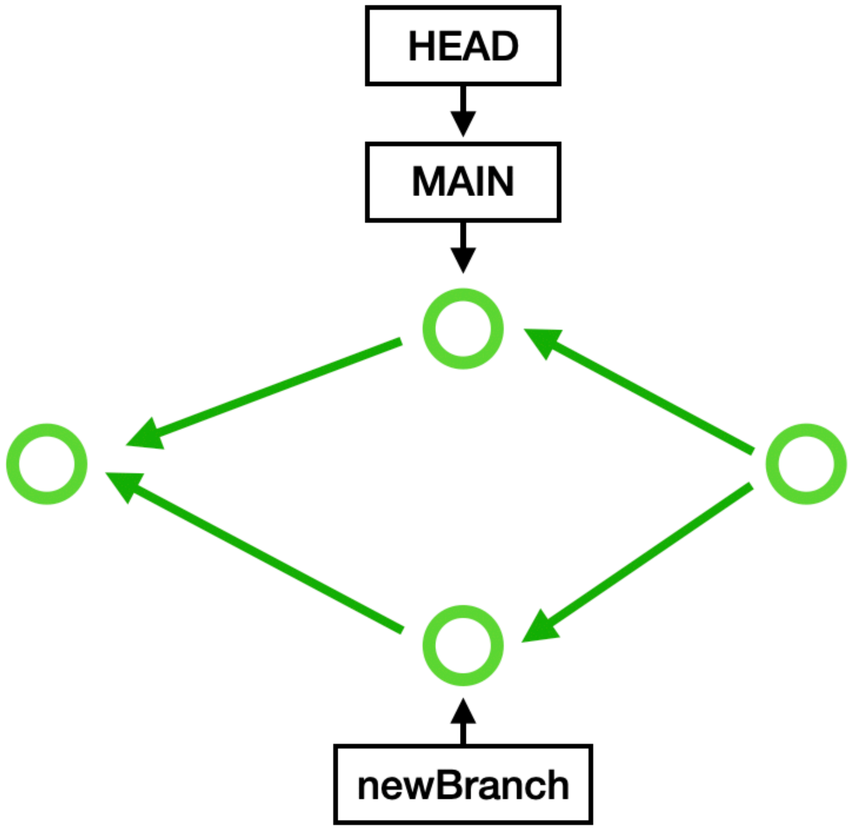
Git rebase



Git key concepts

Code.txt file contents:

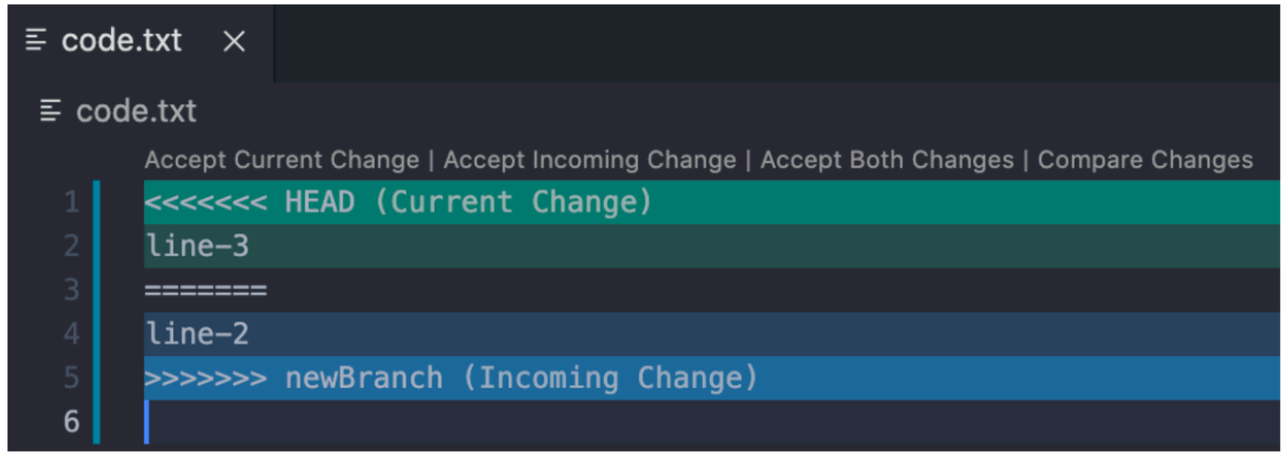
Line3



Code.txt file contents:

Line2

Merge conflicts



Git commands

Staging Area

git add .

git add <filename>

Commit

git commit -m “message”

Branches

git branch

git checkout -b <new branch name>

Merging

git merge <branch to merge>

git rebase <branch to rebase to>

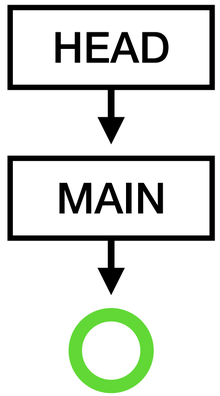
Utility

git status

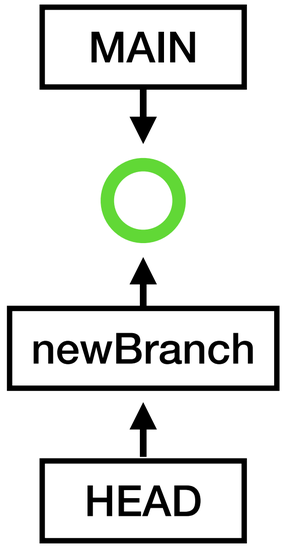
git log --all --graph --oneline

git diff <filename>

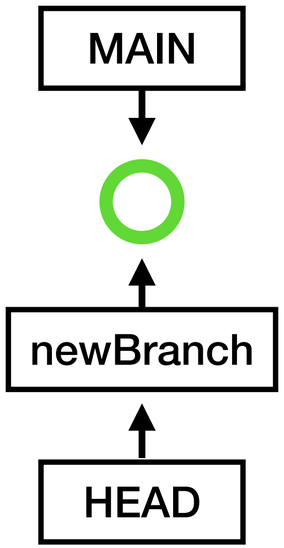
Visualise git



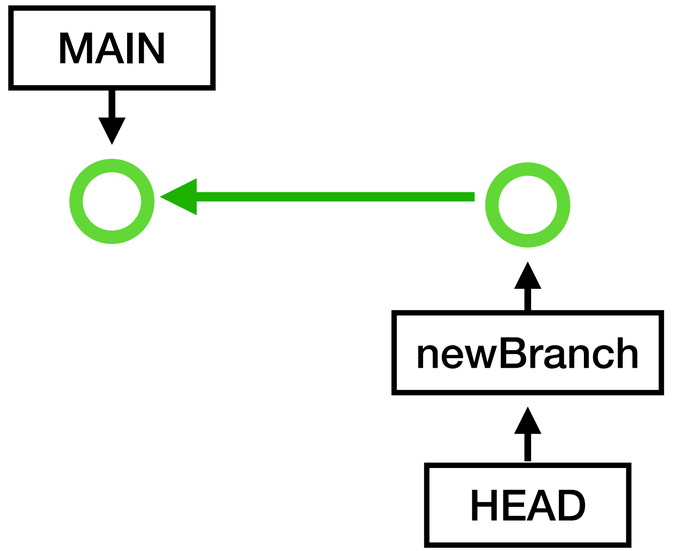

```
Git checkout -b “newBranch”
```



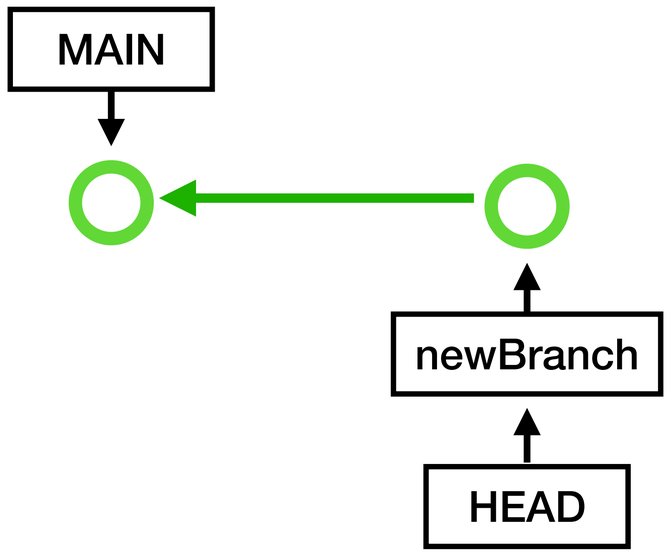
Visualise git



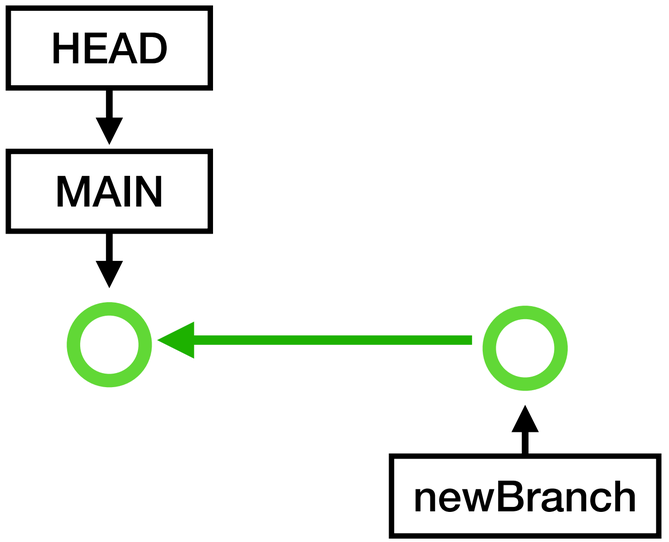

```
Make some changes
Git add .
Git commit -m “1st change”
```



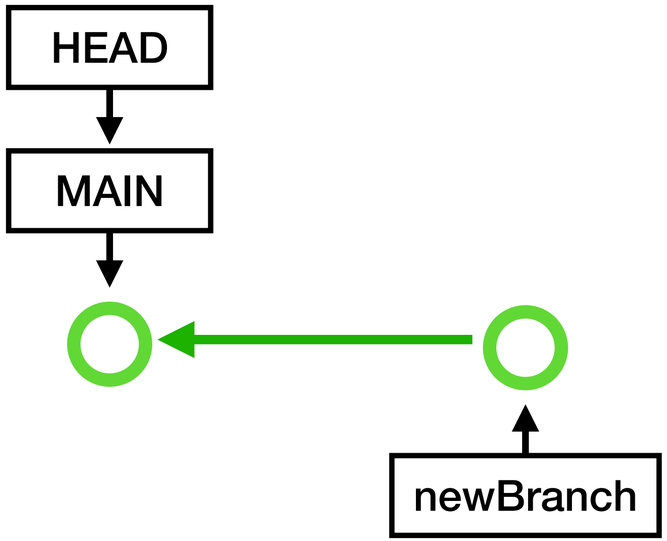
Visualise git



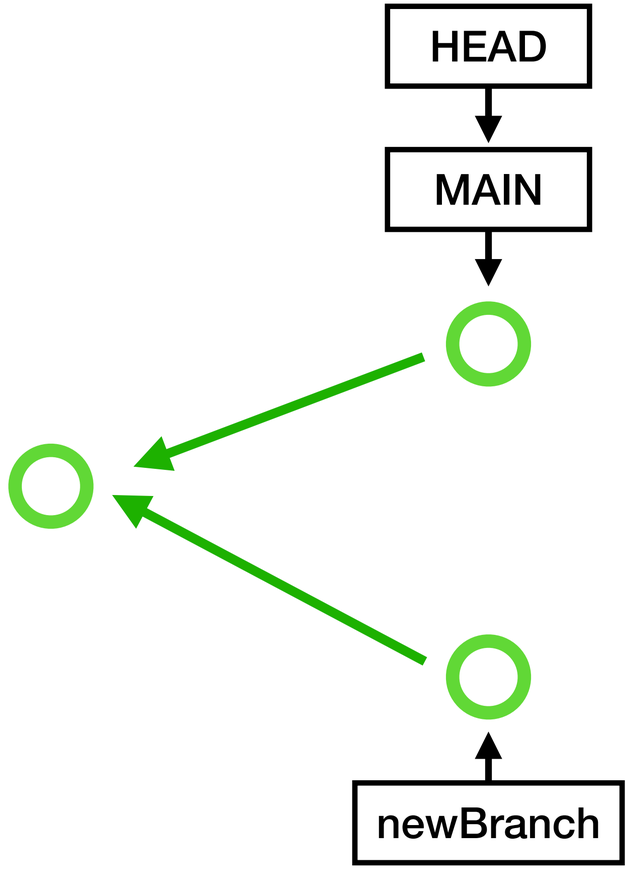

```
Git checkout main
```



Visualise git

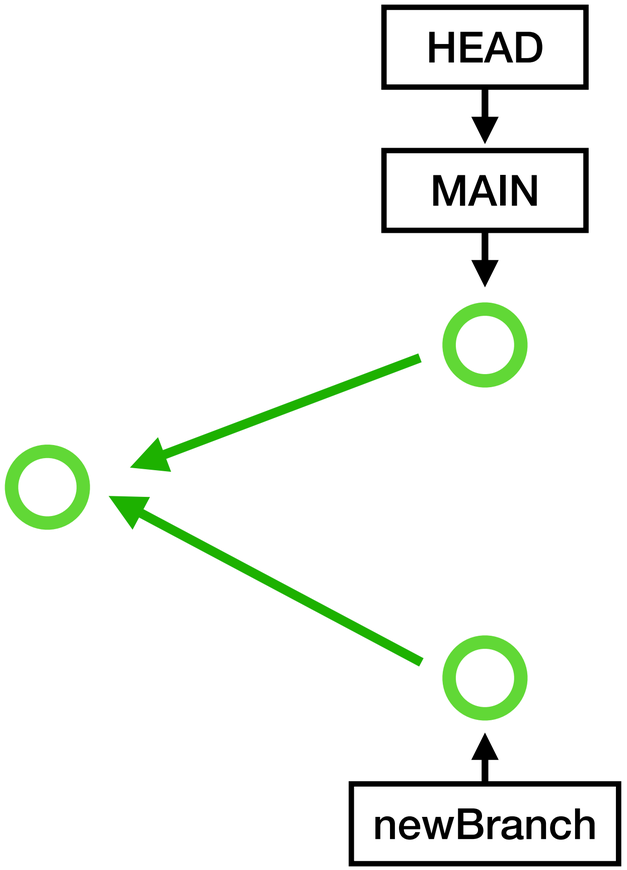



```
Make some edits
Git add .
Git commit -m “second commit”
```

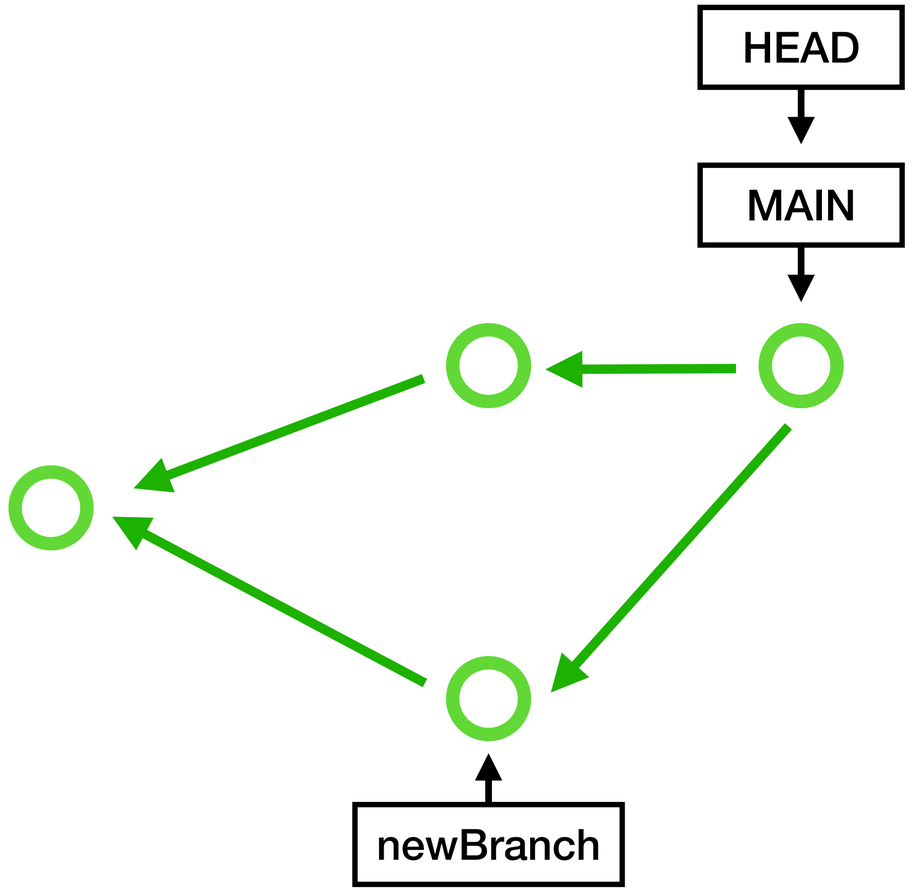


Visualise git

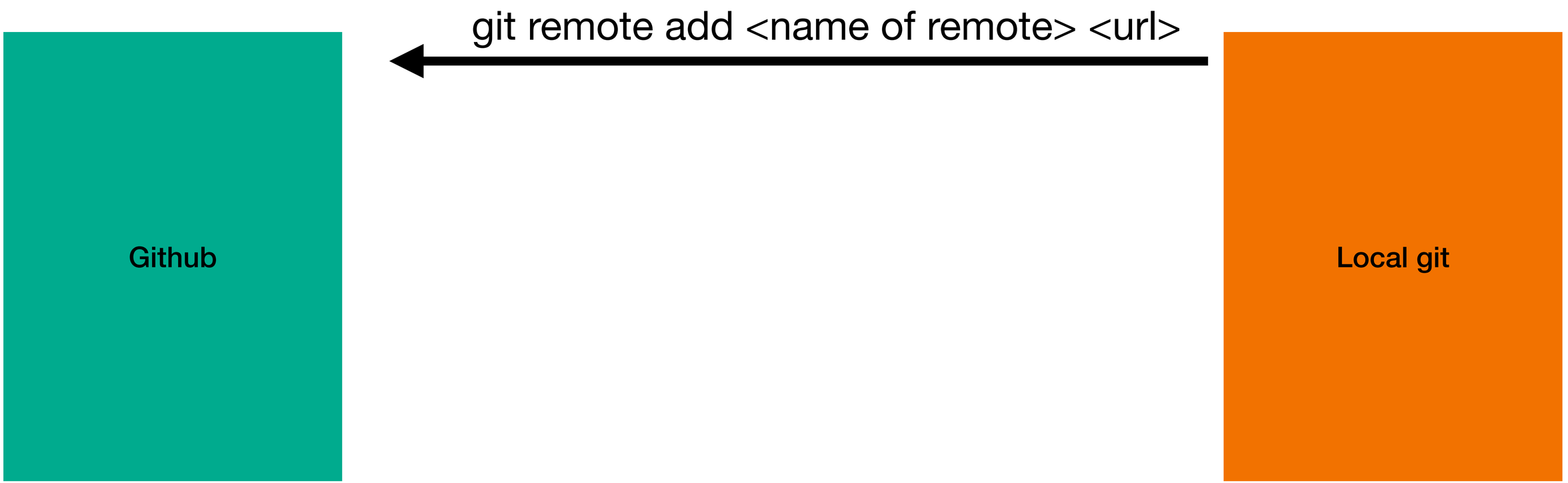
Three way merge



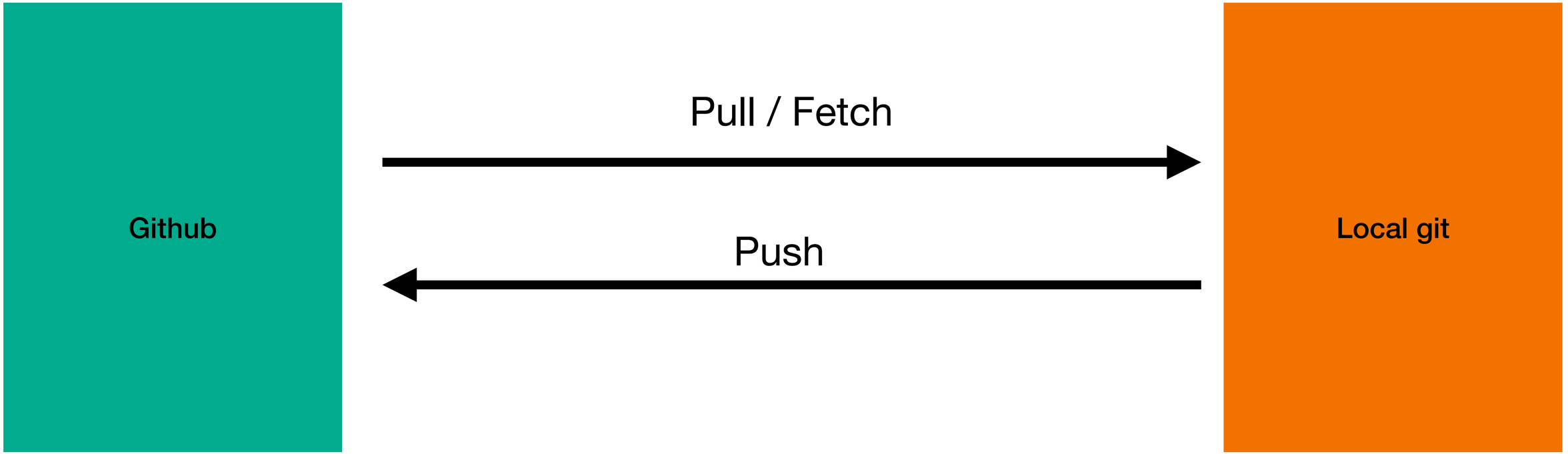

```
Git merge newBranch
```



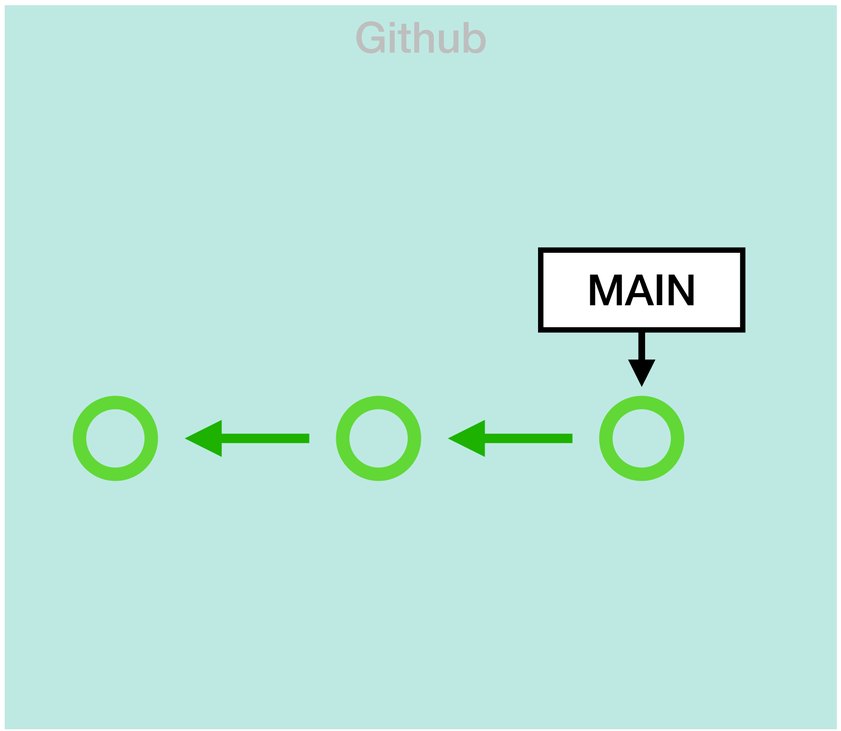
Working with GitHub

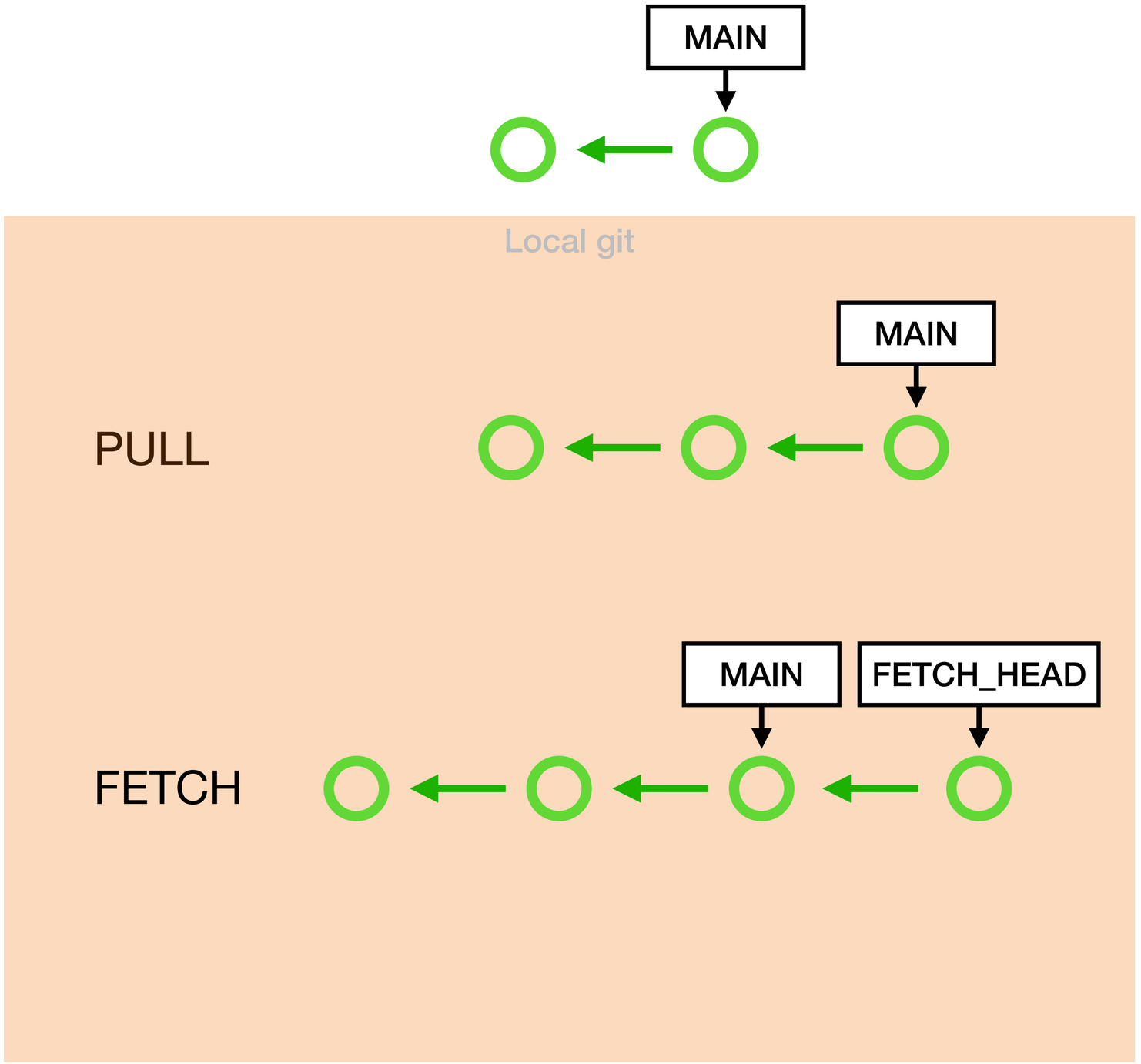


Working with GitHub



Working with GitHub





Working with GitHub

.gitignore

Tells git to ignore specific files when pushing to GitHub.

Secret files such as API keys, config files

Git commands

Staging Area

git add .

git add <filename>

Commit

git commit -m “message”

Branches

git branch

git checkout -b <new branch name>

Merging

git merge <branch to merge>

git rebase <branch to rebase to>

Utility

git status

git log --all --graph --oneline

git diff <filename>

.gitignore

**/foldername
— every folder that has the specified folder name

*.png
— every file with .png extension

Licenses

Tells other developers on GitHub what they can do with your work

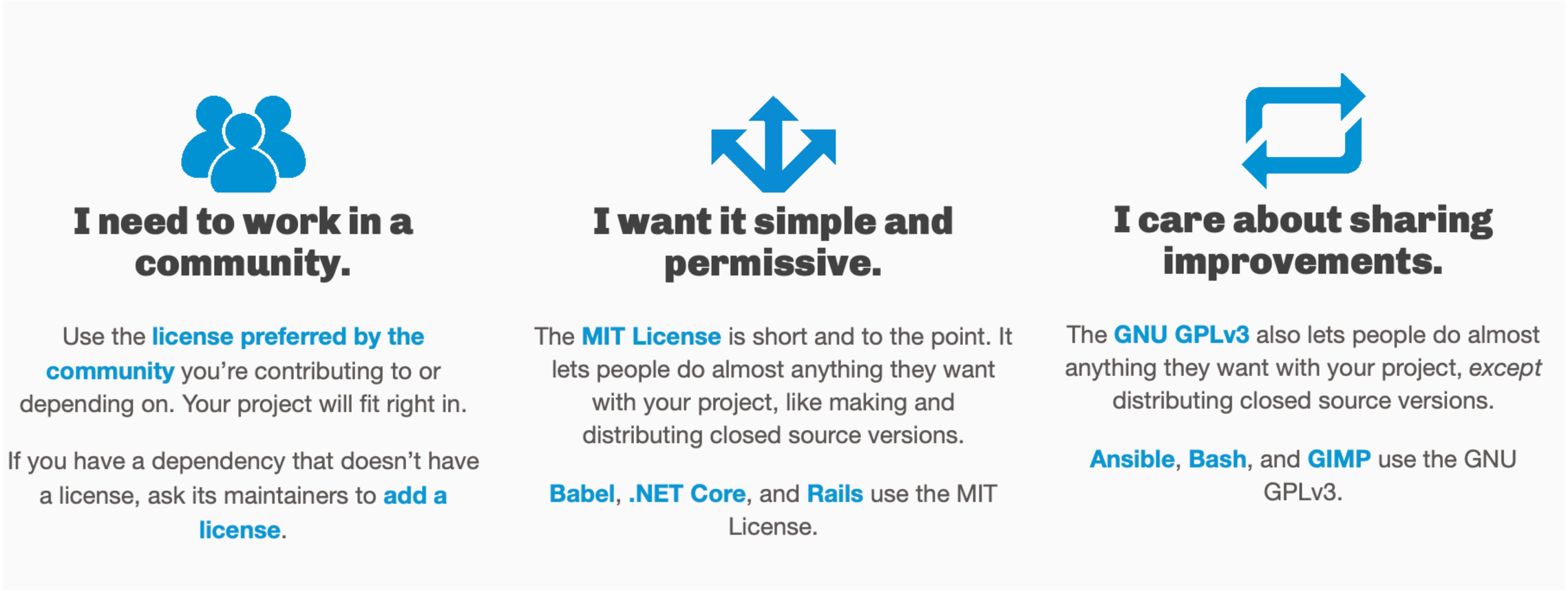


Image from: https://choosealicense.com