Git Week 3

Internals

Where is my data being stored ?

- Check the .git folder
- View hidden files for OSX users …

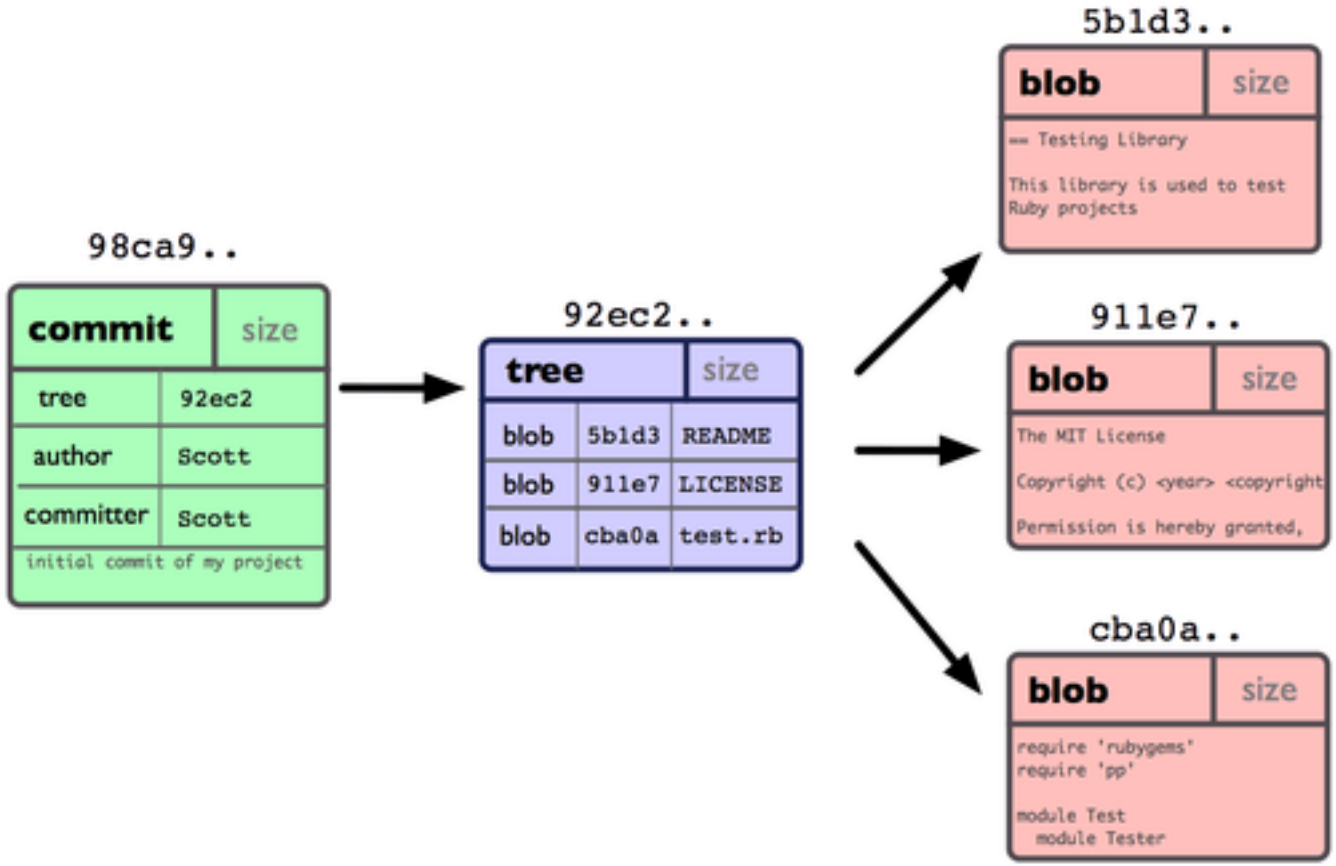
  “defaults write com.apple.Finder AppleShowAllFiles TRUE”

  – *Then relaunch Finder (context click icon + alt)*

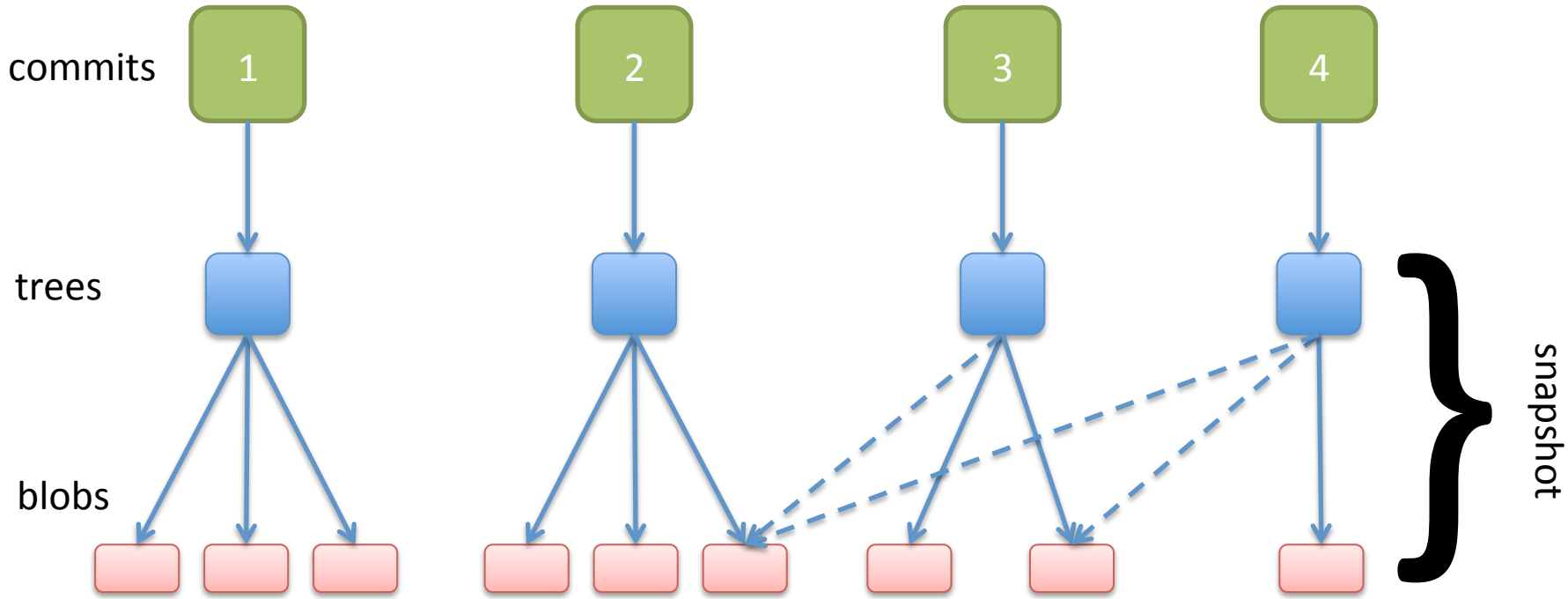
Immutable objects

- Blobs
  - Where the contents of files are stored
  - NB Not file names / permissions
  - Only new blobs for changed files/ fragments of files
  - SHA name for each
- Tree
  - A list of all the blobs for the directory
  - Also has a list of connected trees (i.e. sub directories)
- Commit
  - A pointer to the root tree
  - Has all the other information

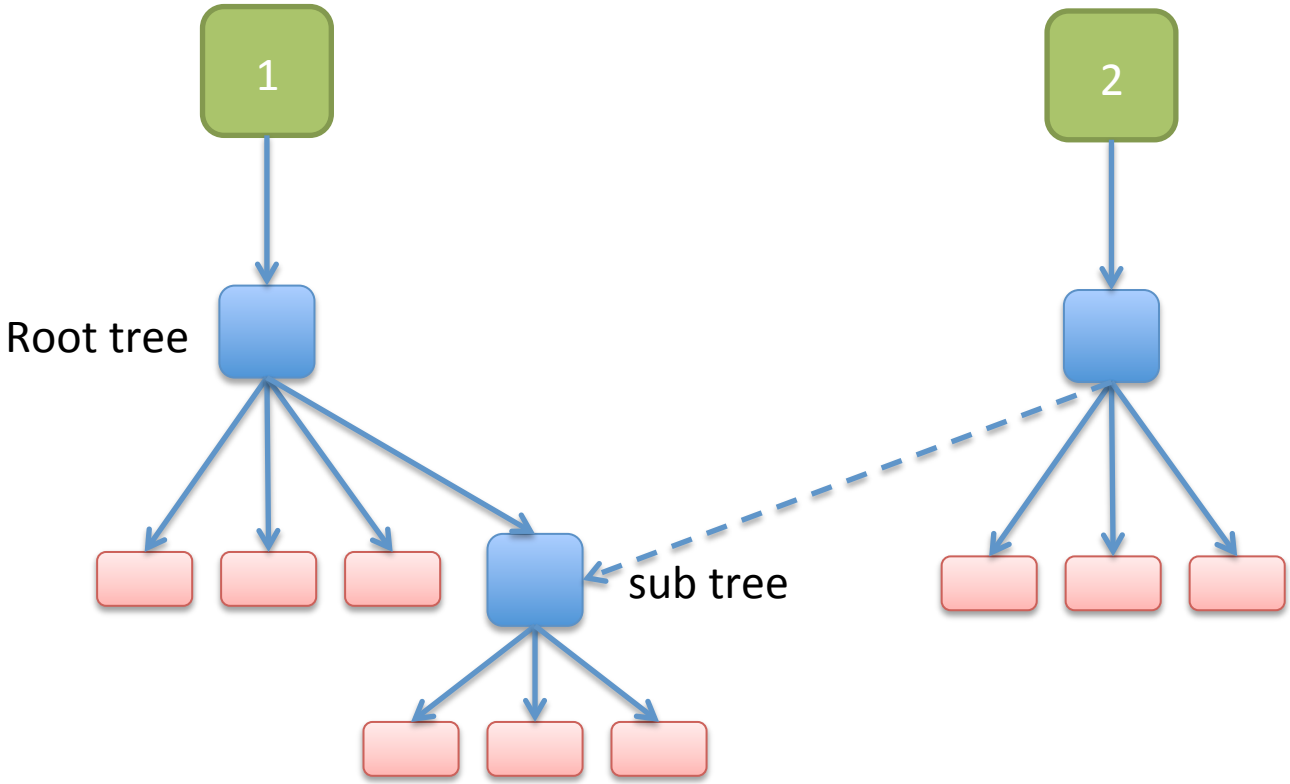
What it looks like …



With multiple commits



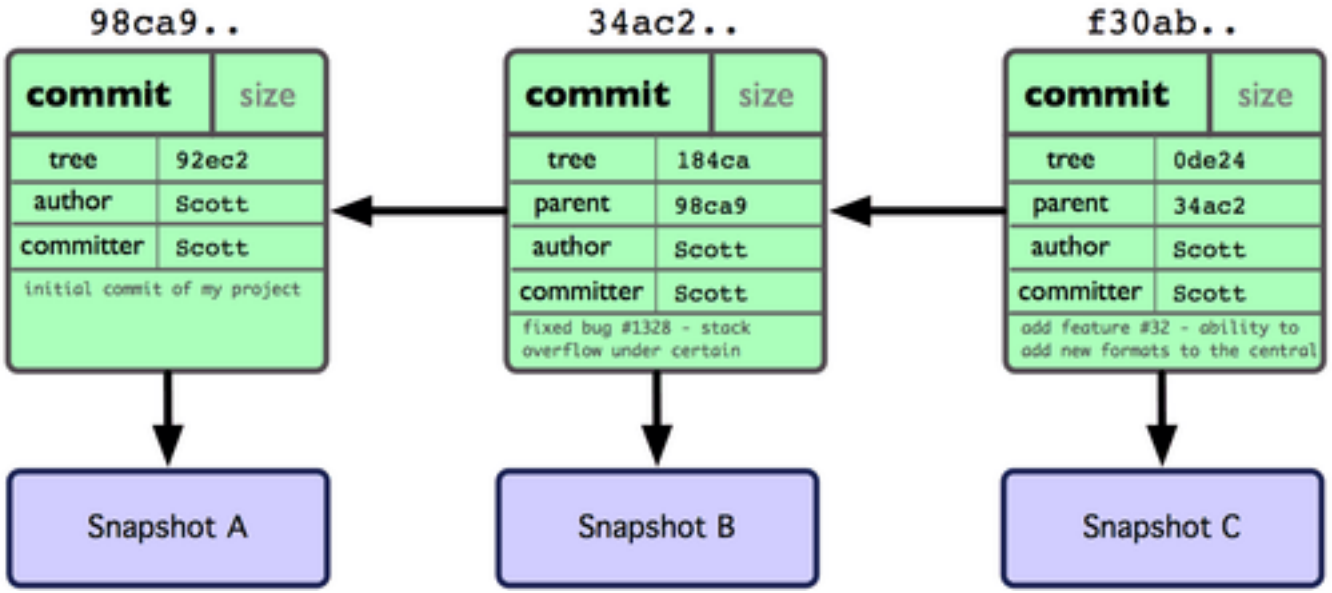
More complex tree arrangements



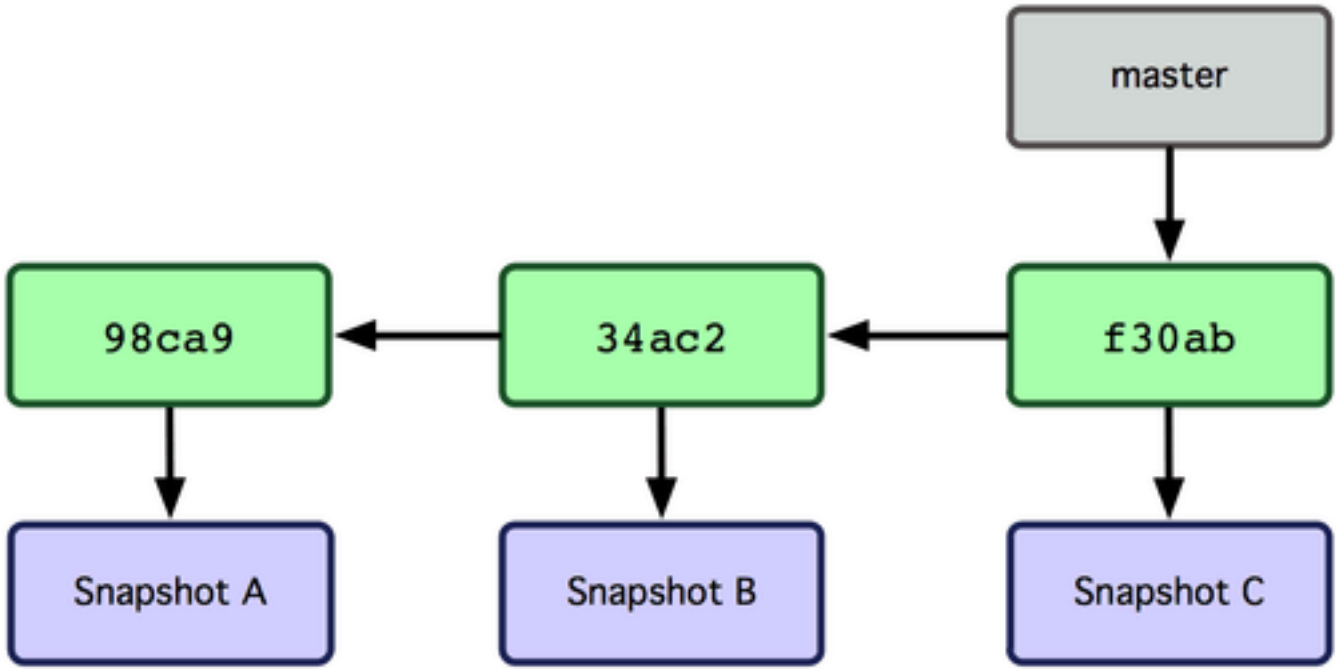
Immutable & mutable objects

- Commits, Blobs & Trees are immutable
- In other words, once created they can’t be modified

Lets ignore blobs & trees

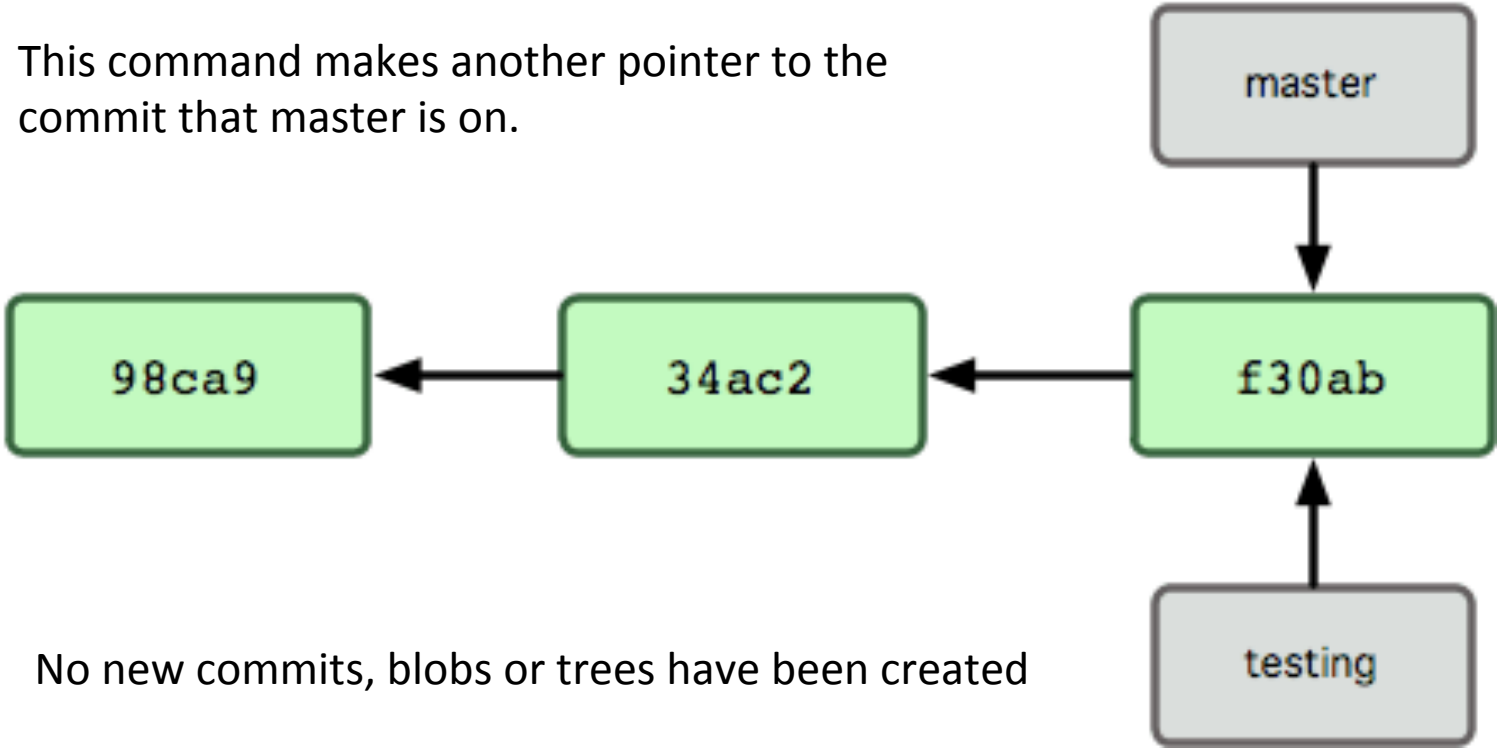


A branch is just a pointer



$ git branch testing

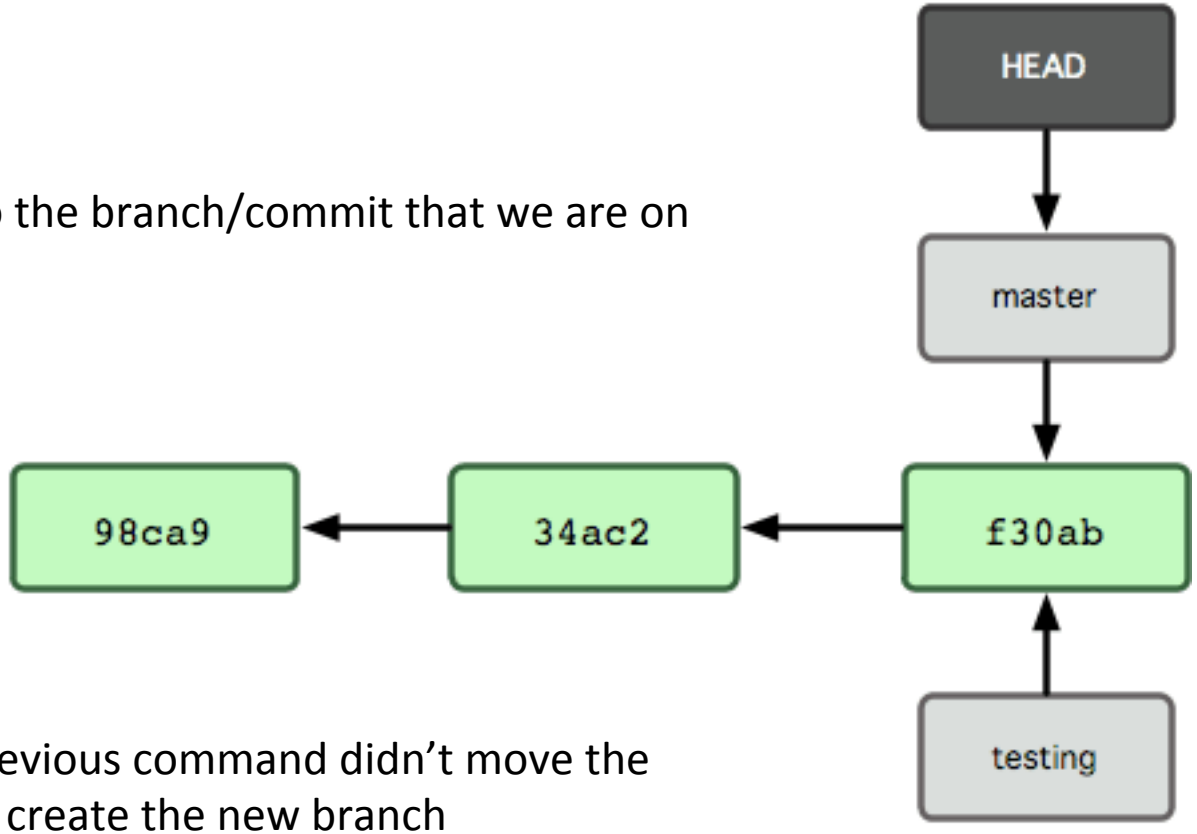
This command makes another pointer to the commit that master is on.



No new commits, blobs or trees have been created

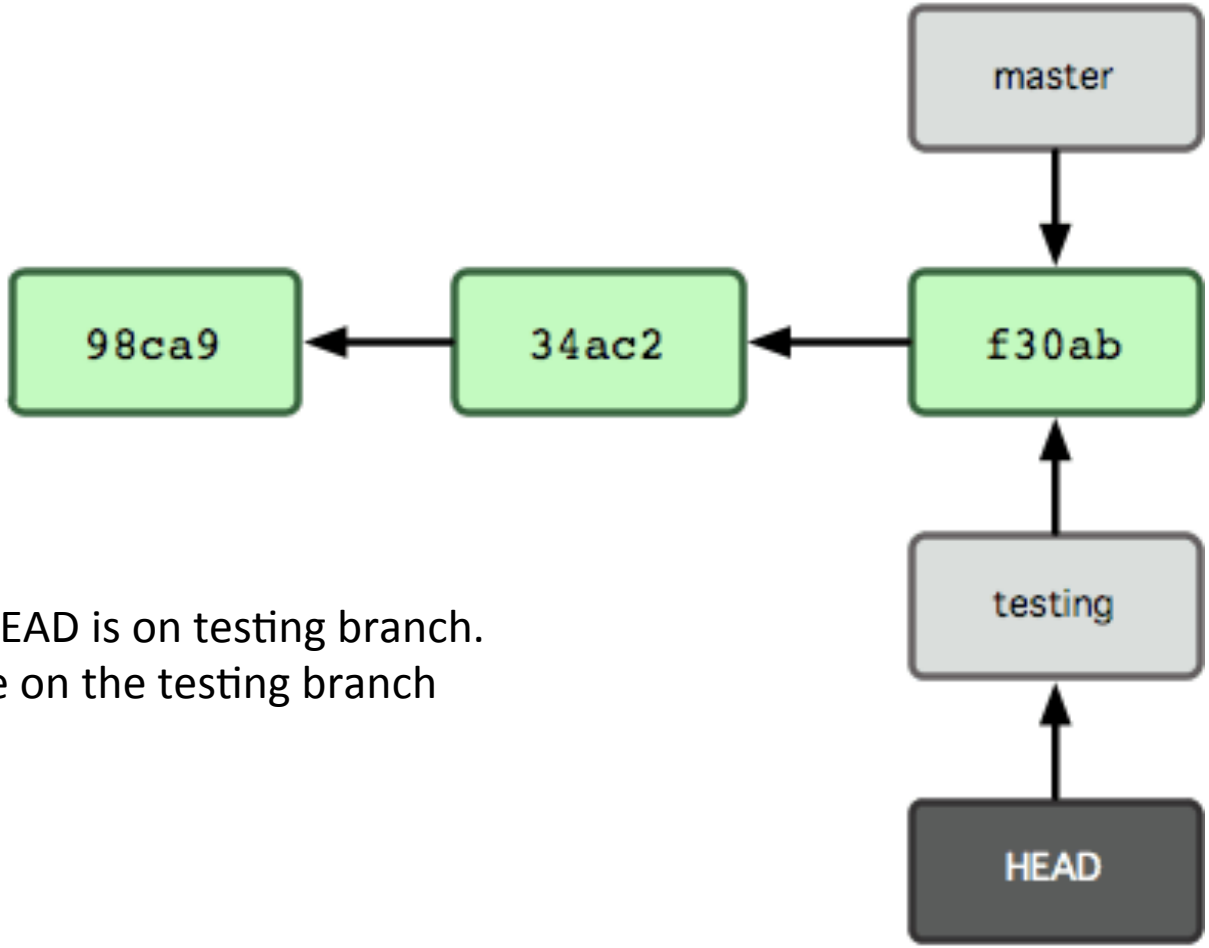
HEAD is a special pointer

It points to the branch/commit that we are on



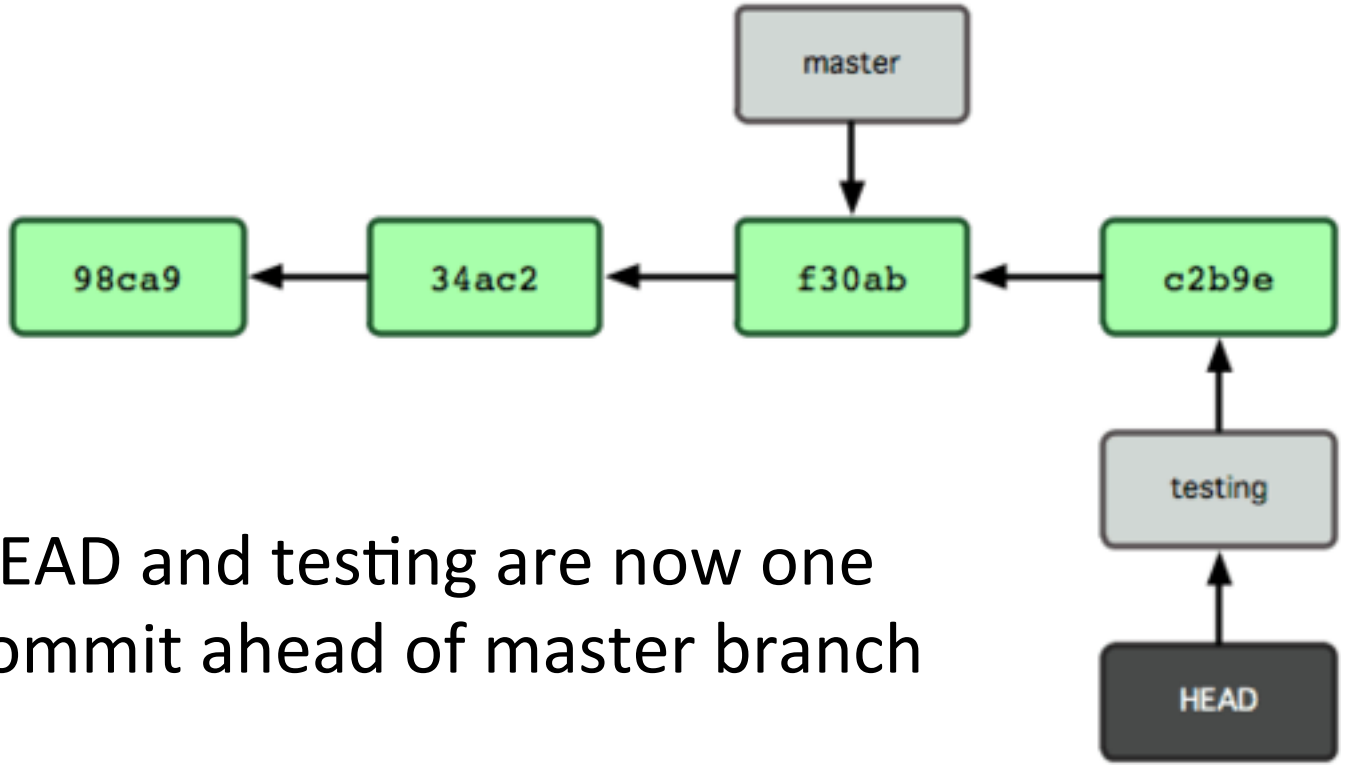
NB. The previous command didn't move the HEAD only create the new branch

$ git checkout testing



Now HEAD is on testing branch.
We are on the testing branch

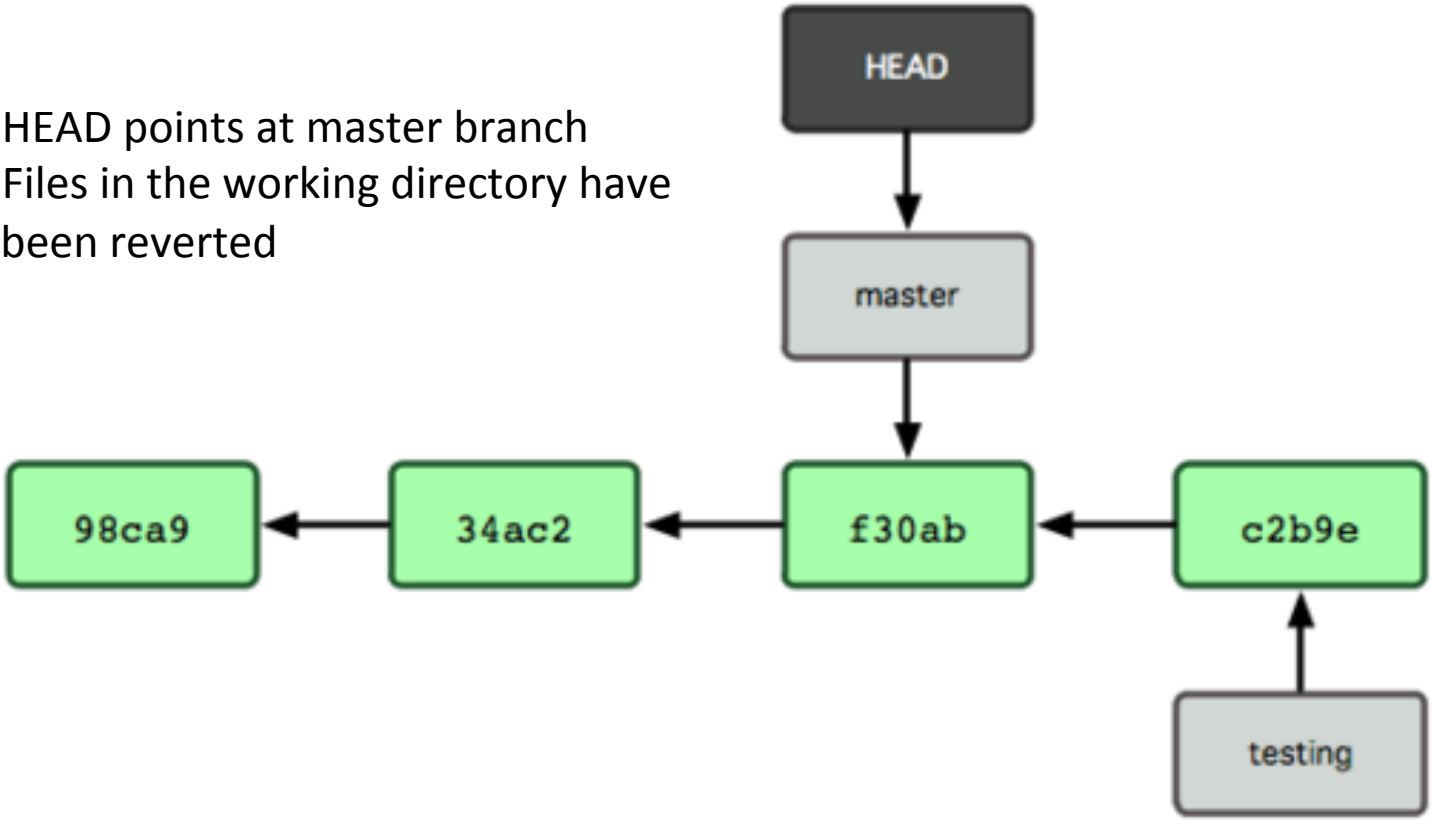
```
$ vim index.html
$ git commit –a –m ‘made a change’
```



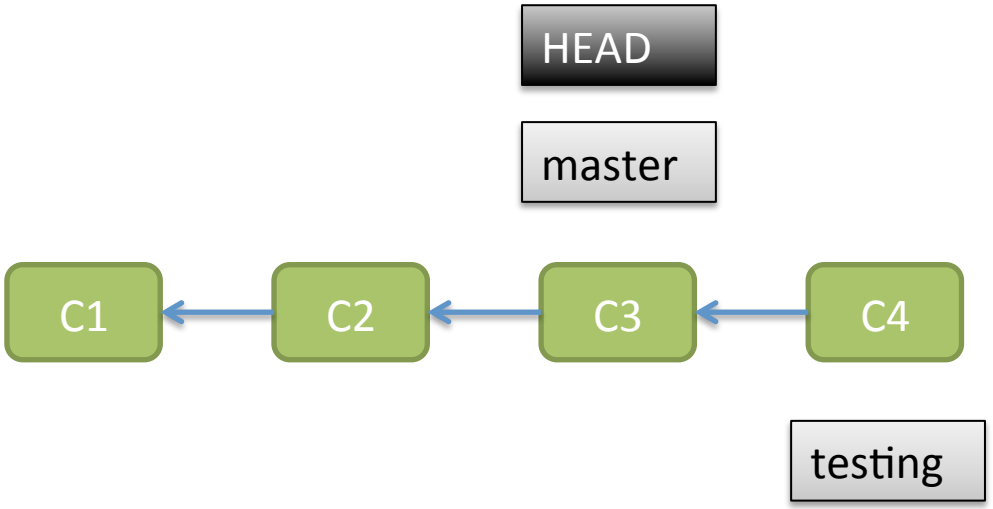
HEAD and testing are now one commit ahead of master branch

$ git checkout master

- HEAD points at master branch
- Files in the working directory have been reverted



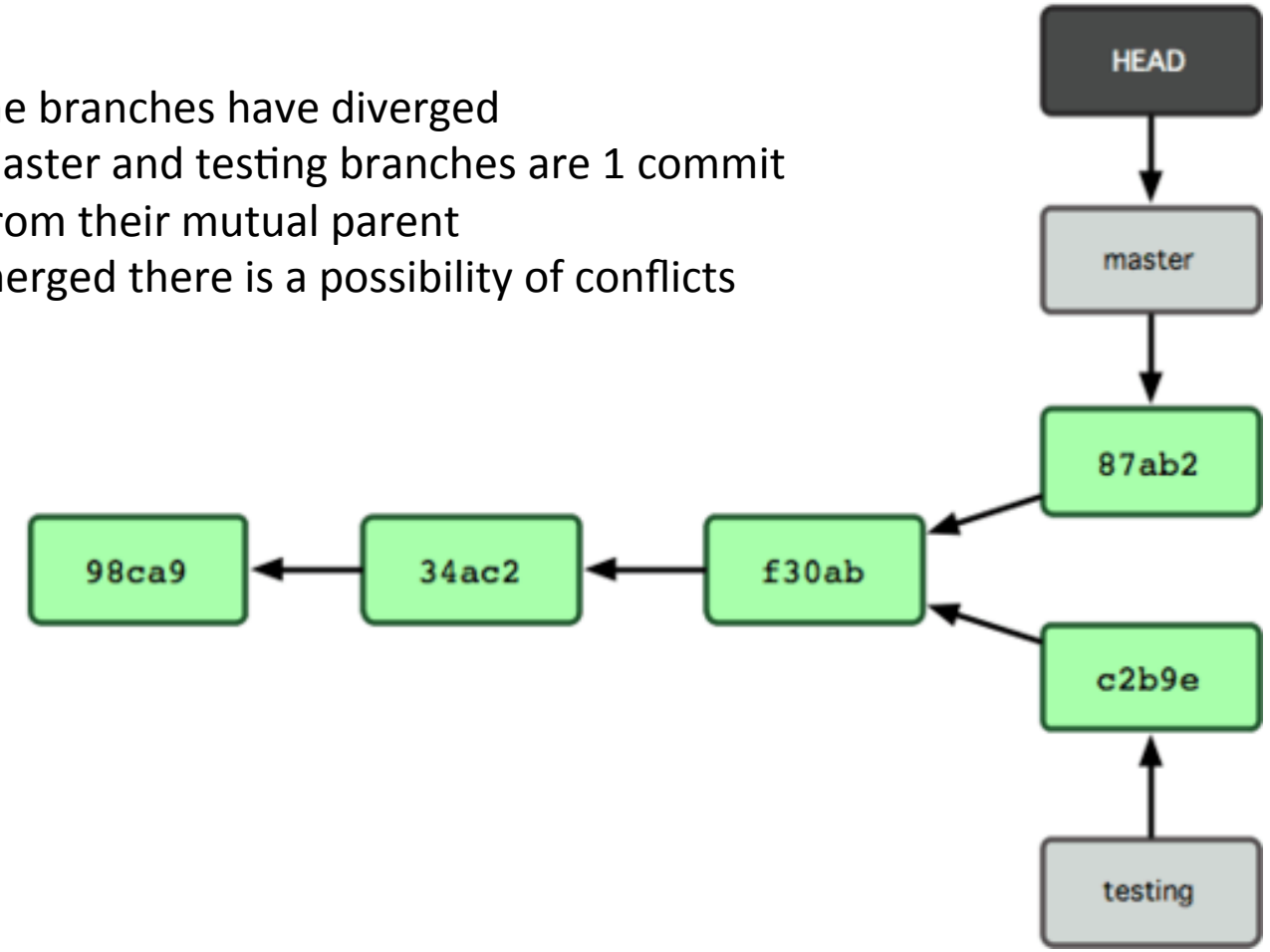
a simple ffwd merge




```
$ git merge testing
```

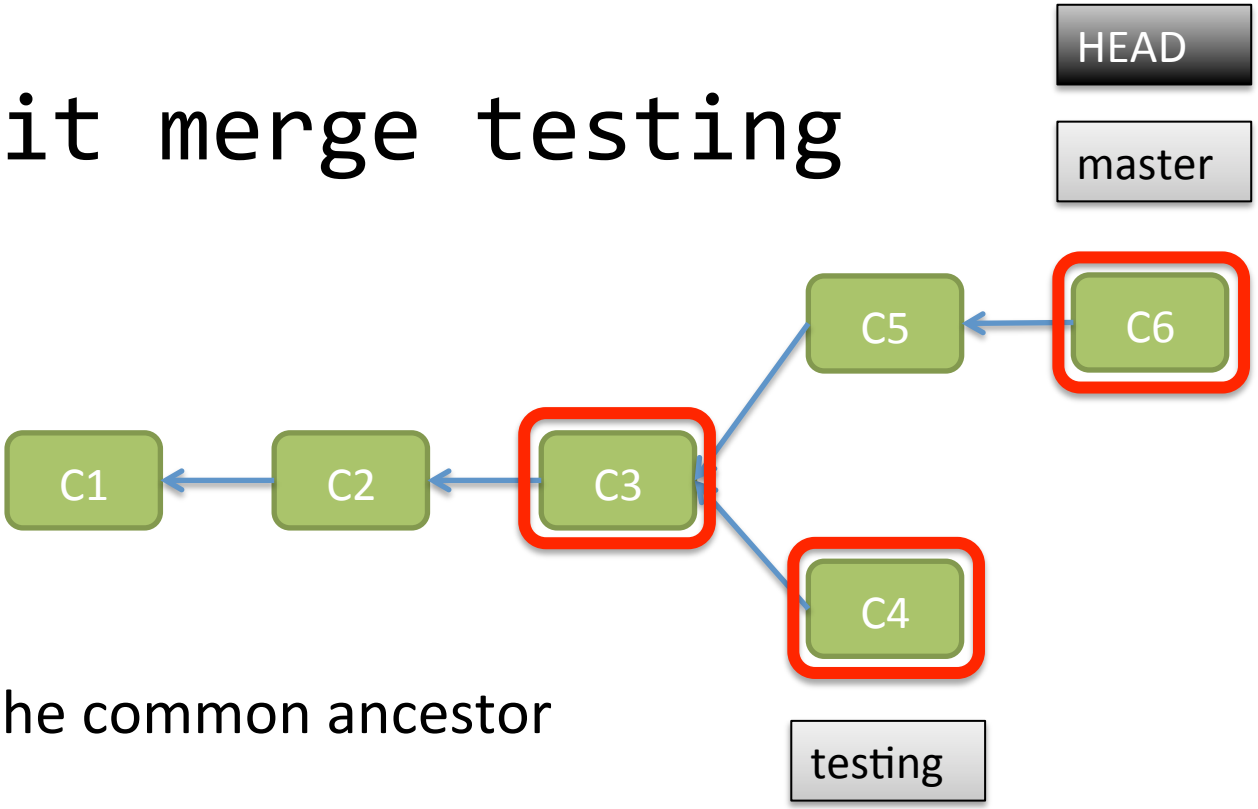
```
$ vim index.html
$ git commit –a –m  “some more changes”
```

- Now the branches have diverged
- Both master and testing branches are 1 commit away from their mutual parent
- If we merged there is a possibility of conflicts



recursive merge

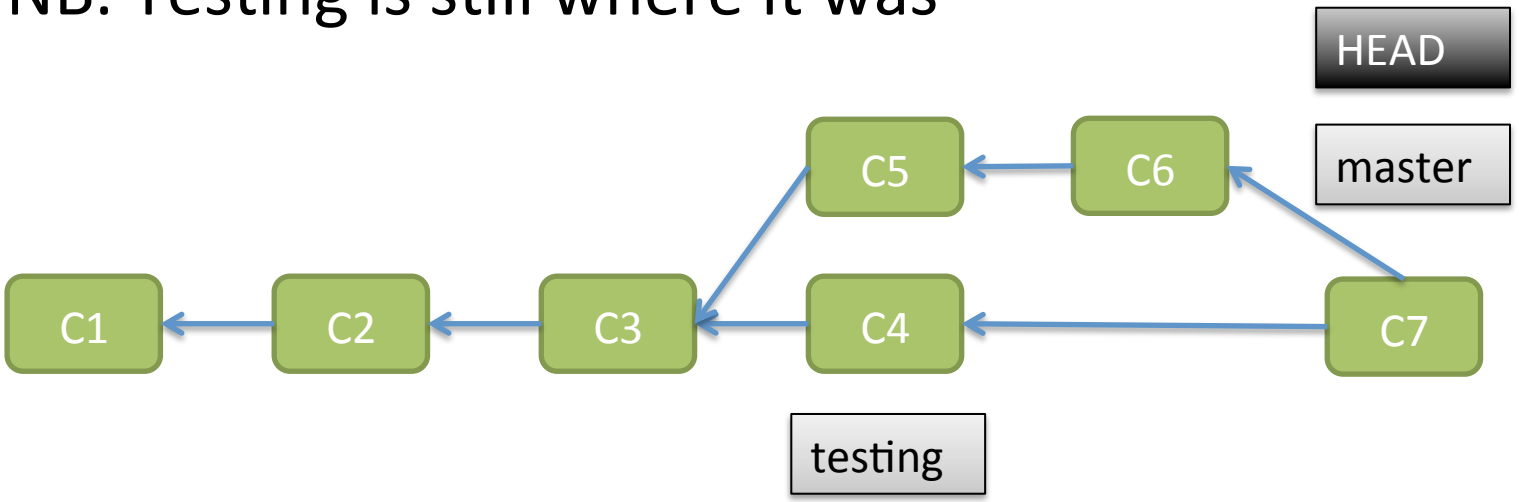
```
$ git merge testing
```



C3 is the common ancestor

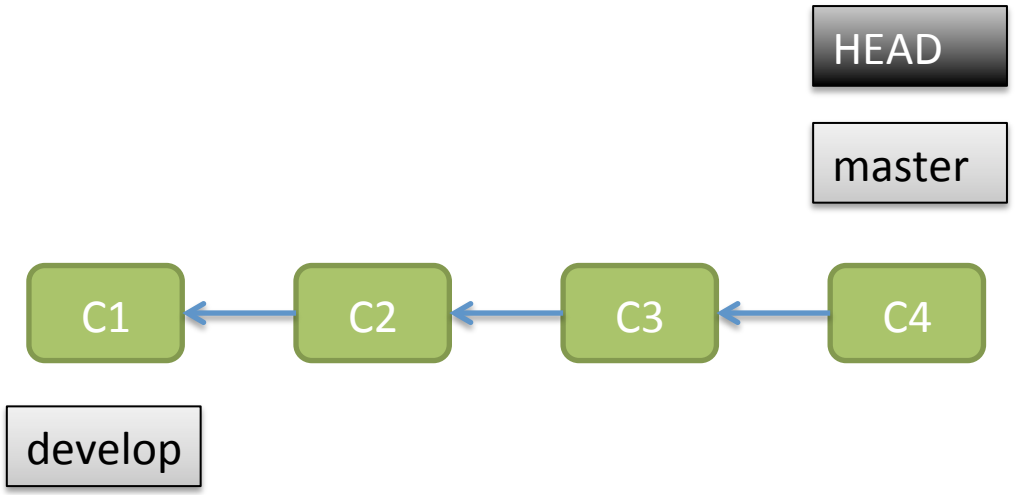
the result

NB. Testing is still where it was



$ git checkout HEAD~3

- This detaches the HEAD from the branch




```
$ git branch develop
$ git checkout develop
```

Your turn

A GUI to help us

http://www.sourcetreeapp.com/

Use command line and check whats happened in the GUI

Make sure you really understand what is happening with branching, checkouts and merging

Research projects

- Use sourcetree to work out what reset does to the repo
- What's the difference between --hard and –soft
- What does --amend do to the repo
- Read pro-git 3.3 and 3.4 for understanding management and work flows.
- Try a branching workflow on your own project

Command test 1

- I want to initialise a repository for a project in a directory called “myProject”.

  What commands do I use ?

- I need to configure a local username and email ?

Command test 2

- If I want to check the status of my repository ?
- If I want to see the history of my repository ?
- the history of just the last 3 commits
- each commit on one line each

Command test 3

- Start tracking a file
- Start tracking all files
- Stage an individual file
- Stage and commit all files that have changed
- Commit staged files only

Command test 4

- Create a new branch
- Switch to a different branch
- Create a new branch and switch to it at the same time