Git - 0 to Pro Reference

By: supersimple.dev
Tutorial link: https://www.youtube.com/watch?v=hrTQipWp6co

Command Line (Terminal / PowerShell)

`ls`	List the files and folders in the current folder
`cd ~/Desktop/folder`	Change the folder that the command line is running in

Note: git commands must be run inside the folder that contains all the code.

Creating Commits

In Git, version = commit
Version history = commit history

`git init`	Git will start tracking all changes in the current folder
`git status`	Show all changes since the previous commit
`git add <file|folder>`	Pick changes to go into next commit
`  git add file`	Pick individual file
`  git add folder/`	Pick all files inside a folder (and subfolders)
`  git add .`	Pick all files (in folder command line is running in)
`git commit -m "message"`	Creates a commit with a message attached
`git commit -m "message" --amend`	Update previous commit instead of creating new one
`git log`	View the commit history
`  git log --all`	Show all commits (not just current branch)
`  git log --all --graph`	Show branching visually in the command line

Configure Name & Email for Commits

```
git config --global user.name "Your Name"
git config --global user.email "email@example.com"
```

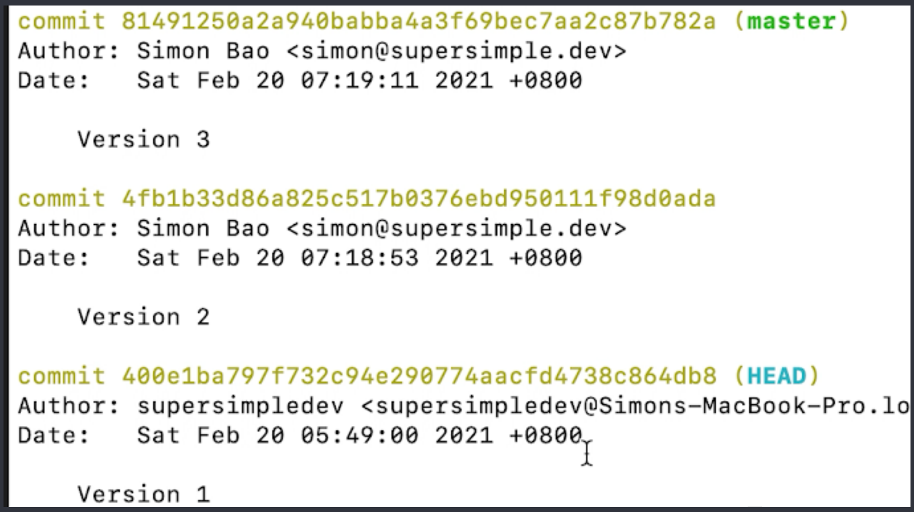
Working Area = contains changes start in the working area

Staging Area = contains changes that will go into the next commit

`git add .`	working => staging
`git commit -m "message"`	staging => commit history
`git reset <file|folder>`	staging => working
`  git reset file`	
`  git reset folder/`	
`  git reset .`	
`git checkout -- <file|folder>`	working => remove the changes
`  git checkout -- file`	
`  git checkout -- folder/`	
`  git checkout -- .`	

Viewing Previous Commits

`git checkout <commit_hash|branch_name>` View a previous commit



```
commit 81491250a2a940babba4a3f69bec7aa2c87b782a (master)
Author: Simon Bao <simon@supersimple.dev>
Date:   Sat Feb 20 07:19:11 2021 +0800

    Version 3

commit 4fb1b33d86a825c517b0376ebd950111f98d0ada
Author: Simon Bao <simon@supersimple.dev>
Date:   Sat Feb 20 07:18:53 2021 +0800

    Version 2

commit 400e1ba797f732c94e290774aacfd4738c864db8 (HEAD)
Author: supersimpledev <supersimpledev@Simons-MacBook-Pro.lo
Date:   Sat Feb 20 05:49:00 2021 +0800

    Version 1
```

master = branch name
1. You can `git checkout branch`
2. Always points to latest commit on the branch.

HEAD = indicates which commit you are currently viewing

Restoring to a Previous Commit

`git checkout <hash|branch> <file|folder>`	Restore the contents of files back to a previous commit
`  git checkout <hash|branch> file`	Restore a file
`  git checkout <hash|branch> folder/`	Restore all files in folder (& subfolders)
`  git checkout <hash|branch> .`	Restore all files in project

Other Features of Git

`git config --global alias.shortcut <command>`	Creates an alias (a shortcut)
`  git config --global alias.s "status"`	`git s = git status`

`.gitignore`	Tell git which files/folders it SHOULD NOT track
`rm -rf .git`	Remove git from project

GitHub

Repository = a folder containing code where any changes to the code are tracked by git.
(To create a repository, we create a new folder on our computer, and then run `git init`)

GitHub = a service that lets us save our git repositories online. It also helps us:

- backup our code in case we delete it on our computer
- see the history of our code changes more easily
- alternatives include Bitbucket and GitLab

Local repository = a git repository saved on our computer
Remote repository = a git repository saved online (for example on GitHub)

Uploading Code to GitHub

`git remote add <remote_name> <url>`	Link a local repository to a remote repository and give a name for this link

```
git remote add origin https://github.com/SuperSimpleDev/repository1
^
```

The above command links a local repository to a GitHub repository (located at the url `https://github.com/SuperSimpleDev/repository1`) and gives it a name "origin"

`git remote`	List all remote repositories that are linked
`git remote -v`	List all remote repositories (but with more detail)
`git remote remove <remote_name>`	Removes a link to a remote repository
`git remote remove origin`	Removes the link to the remote repository named "origin"
`git config --global credential.username <username>`	Configure your GitHub username so you can get access to your Github repository
`git push <remote_name> <branch>`	Upload a branch of your git version history to your remote repository
`git branch`	Shows a list of available branches
`git log --all --graph`	Shows the branches visually in the history