Superpython talks - **git quick reference** - superpythontalks.github.io

Git is a system for version control – think of it as a series of snapshots for your code, that lets you go back in time.

The basic principle is to create a repository (using git init on a new empty directory), add files (using git add). Each time you want to make a snapshot, stage the files to the commit (git add) and then use git commit (with a descriptive commit message of what you changed). There are a variety of git commands that let you revert back to previous commits, compare code between commits, and see previous commits.

git help <command>	Show help for a command
git init <directory>	Create a new repository in directory - if no directory given, uses current dir
git add <directory or file>	Add directory or files to the next commit
git commit -m "<message>"	Commits changed files to the repository – providing snapshot
git clone <path>	Make a copy of an existing repository (can be from a remote location)
git config user.name	Set author name for the current repository
git status	List the status of all files
git log –number	Display commit history (if number is given, only show that many commits)
git diff	Show differences between current and committed version
git reset <file>	Reset a file (or the whole repository) to the previously committed version
git revert <commit>	Revert an already committed commit
git branch	List, create, or delete branches
git checkout –b <branch>	Switch branches
git merge <branch>	Merge a branch into the current branch
git remote add <name> <url>	Add a remote location
git fetch <remote> <branch>	Download from a remote repository
git pull <remote>	Fetch and merge from a remote
git push <remote> <branch>	Update to a remote location

Useful links:

git-scm.com

try.github.io

gitref.org