Git Cheatsheet

Note that you can find more information with the --help option. Ex: git init --help will give us more information about the git init command while generic git help will show us a list of command.

Starting a project

git init

Initialize a repository. Default is current directory. Other usage:

git init <dir>

- Initializes a repository in dir

git clone <url>

Clone a repository from <url>

Local commits

git add

Stage a file or files for commit. Common variants include:

git add <filename>

- stages a file for commit

git add .

- stage everything in the current folder and its sub-folders except those listed in the .gitignore
-

git commit

Commits currently staged files. Common flags are

git commit -m “Message”

- Commits the currently staged files with the specified commit message

git revert

Creates a new commit that undoes the changes from a previous commit. Example:

git revert <hash>

- Reverts the commit with the specified hash

git log

Show the history of commits on the current branch.

git reset

Reset to a specific hash. E.g.

git reset --hard HEAD

- Throw away staged and unstaged commits

git reset --hard <hash>

- Move the current branch HEAD to <hash>

git cherry-pick

Apply a specific commit to the current branch. Usage:

git cherry-pick <hash>

Working with remotes

git fetch

Get updates about remote changes but do not merge into the local repo. Commonly used with -a to get all remote info.

git push

Push local changes to remote. Common variants are

git push -u origin branchname

- Creates a remote branch with the same name and pushes to it

git push --force

- Overwrite the remote branch

git push --force-with-lease

- Overwrite the remote branch but don’t touch other people’s commits

git pull

Update local repository with remote changes

git remote

Managed the remote branches. Usage:

git remote add <name> <url>

- Add a new remote with the specified name and url

git remote -vv

- Show info about current remote branches

git remote remove <name>

- Remove a remote

Branches

git checkout

Checkout a specific branch or hash. Usage:

git checkout <branchname>

Switch to the specified branch

git checkout <hash>

Switch to the specified hash. This is known as a detached head state since we are not on any branch.

git checkout -b <branchname>

Create a new branch and switch to it

git branch

Managed branches. Example usage:

git branch

- Show branches

git branch <name>

- Create new branch with the specified name at the current HEAD

git merge

Merge another branch into the current branch. E.g.

git merge otherbranch

git rebase

Move commits from the current branch onto another branch. Ex:

git rebase main

- Replay the commits from the point the current branch diverged from master onto the latest master

git rebase --onto branchA branchB

- Replay the commits in the current branch from the point it diverged from branchA onto branchB