

## Useful git commands

### git init

Initialize a repository at the current directory

### git clone <remote URL>

Clone an existing remote repository into a newly created local repository. The local repository is created in a new folder with the same name as the remote repository.

### git clone <--depth N> <remote URL> <directory>

As above, but limit depth of cloned commits to N and put the cloned repository in specified directory.

### git status

Display lists of files that are untracked, tracked-but-not-staged, or staged-but-not-committed.

### git add .

Add all untracked files AND all tracked, changed or new but unstaged files, to the staging area.

### git commit -m "Commit message goes here"

Commit all staged files on the current branch.

### git commit -am "Commit message goes here"

Same as doing "git add a ." followed by "git commit -m."

### git branch -v

List branches.

### git branch -a

List branches, including remote branches.

### git branch -d <local branch name>

Delete local branch.

### git branch <new branch name>

Create a new branch in the local repository.

### git checkout <branch name>

Make the indicated branch the current branch.

### git checkout -b <new branch name>

Create and checkout a new branch.

### git merge <branch name>

Merge named branch into current branch.

### git remote -v

List remote repositories.

`git remote add <remote> <remote URL>`

Add a new remote repository, to be referred to by the shorthand name represented by <remote>.

`git remote remove <remote>`

Remove the remote repository.

`git fetch <remote>`

Fetch all material from the remote repository that the local repository doesn't already have.

`git remote set-url - -push <remote branch> DISABLE`

Make this remote branch read-only by providing a non-functioning url for push.

`git merge <remote>/<remote branch>`

Used after fetch, merges remote branch onto the current branch.

`Git pull <remote>`

Combines the effects of fetch and merge.

`git remote show <remote>`

Show detailed information about a remote repository.

`git push <remote> <branch>`

Pushes local branch onto a branch of the same name on the remote repository (creating such a branch on the remote repository if need be). This will fail if the remote has information that the local repository doesn't already have. If that happens, it is necessary to first do a pull or fetch/merge.

`git push -u <remote> <branch>`

Like the above push command, but the '-u' option causes <branch> to track the remote branch.

`git push <remote> --delete <branch>`

Delete branch from the remote repository.

`git checkout -b <localbranch> <remote>/<remotebranch>`

Create new local branch and set it to track remote branch.

`git branch -u <remote>/<remotebranch>`

Set current branch to track the remote branch.

`git branch -d -r <remote>/<remotebranch>`

Delete remote tracking branch (does NOT delete the branch on the remote repository).

`touch <newFileName>`

Create a new file. Example: good way to get a .gitignore file after initializing repository.

`git rm --cached <filename>`

Remove a file from the index (i.e., stop tracking it), but do not remove it from the working directory.