

# git cheatsheet

---

`git init`: initialise git repository

`git config --global user.email "<email>":` set default author email address (will appear in all commits authored by you)

`git config --global user.name "Your Name":` set default author name (will appear in all commits authored by you)

`git status`: get current repository status

`git add <filenames>`: track a file and move it to the staging area (stage), e.g. `git add myfile.py`, also `git add .` to add all files and folders under the current directory

`git commit`: add staged changes to a commit and apply the commit to the local repo. Add a commit message from your default text editor (`$EDITOR` env var)

`git commit -m "<message>":` same as `git commit`, but inline commit message (commit summary)

`git remote add <remote> <url>`: add a remote, pointing your local repo to a remote git repository, e.g. `git remote add origin git@github.com:ghorg/ghproject`

`git remote remove <remote>`: remove a remote, deleting a link between your local repo and the referenced remote repo, e.g. `git remote remove origin`

`git remote -v`: list out all of your current remotes

`git push`: upload any changes to your local repository to your remote repository

`git push -f`: upload any changes to your local repository to your remote repository, rewriting the git history (index). **USE WITH CAUTION.**

`git diff`: see current un-staged changes in the working directory

`git diff --staged`: see current changes in the staging area

`git diff <commit_hash>^!`: see changes introduced in a specific commit

`git show`: see changes introduced in a specific commit

`git log`: see list of all commits created in the repository, along with author and messages

`git blame <commit_hash>`: get the author of a commit

`git clone`: download a copy of a remote git repository into your development environment

`git pull`: download any changes in a remote repository not currently present in your local repository

`git branch <branch_name>`: create a new branch in your local repository (won't switch automatically)

`git branch`: list out all branches in the local repository

`git branch -a`: list out all branches in both the local and remote repositories

`git checkout <branch_name>`: switch to the specified branch

`git checkout -b <branch_name>`: switch to the specified branch. Creates the branch in your local repository if it doesn't already exist.

`git merge <remote> <branch>`: merge specified branch into the current branch, e.g. `git merge a_branch`

`git reset <commit>`: set the HEAD ref to point at the specified commit

`git reset --hard <commit>`: rewrite git history so the specified commit becomes the new HEAD, removing any commits that came after it forever. **USE WITH CAUTION.**

`git stash` stow away the current changes in the working directory

`git stash pop` recover the last set of changes that were stashed and apply them to the working directory

`git log --graph --abbrev-commit --decorate --format=format:'%C(bold blue)%h%C(reset) - %C(bold cyan)%aD%C(reset) %C(bold green)(%ar)%C(reset)%C(bold yellow)%d%C(reset)%n' %C(white)%s%C(reset) %C(dim white)- %an%C(reset)' --all:`  
pretty-print git log