

Git and Github

> Global config

```
git config --global user.name "Hugo"  
git config --global user.email "hugofernandezmuelas03@gmail.com"  
git config --global init.defaultBranch "main"  
git config --global core.editor "notepad"  
git config --global alias.tree "log --all --graph --decorate --oneline"  
git config pull.rebase false
```

Local repository

> Basic use

git init	(starts a git repository in the working directory)
git add .	(add all files to staging)
git status	(repository status)
git log	(log of commits)
git diff	(difference since last commit)
git commit -m <commit message>	(commit all changes with a given message)
git reset	(list of uncommitted changes)
git checkout <filename>	(reset a file to last commit)
git checkout <hash>	(go to the given hash)

> more git log (git tree):

--graph	(branch graph)
--decorate	(short hash)
--oneline	(one commit per line)

> Advanced use

git reflog	(log of ALL commits even reseted ones)
git reset --hard <hash>	(move head to given hash (works to go back and forth))
git tag	(list all tags)
git tag <name>	(tag a commit as a release, lowercase no spaced)
git diff <branch>	(git diff for branch)
git rebase	(move a branch to the current head (altering commits))
git cherry-pick <hash>	(bring a single commit to current head)

Collaborating/Online repository

git branch	(list all branches)
git branch <name>	(create a branch from the current head)
git branch -d <name>	(delete given branch)
git switch <name>	(change to the given branch name)
git merge <hash/tag/branch>	(merge hash to current head)
git stash	(save changes on a branch locally without commit)
git stash list	(list all stashed changes)
git stash pop	(restore last stash for current branch)
git stash drop	(delete last stash for current branch)

Github

git remote add origin git@github.com:<username>/<repository>.git	(add a remote origin)
git fetch	(download change tree (not changes) from origin)
git pull origin <branch>	(download changes to branch from origin)
git push origin <branch>	(push local commit to branch on origin)
git clone <.git>	(clone git repository locally)

[github ssh](#)

Git flow

[about git flow](#)

git flow init
git flow feature start <name>
git flow feature finish <name>
git flow release start <version>
git flow release finish <version>
git flow hotfix start <version>
git flow hotfix finish <version>

[download git flow](#)

(start a git flow project)
(create a new feature branch)
(merges and deletes branch to develop branch)
(merges develop to release and creates a tag)
(merges release to main and creates a tag)
(create a new hotfix branch)
(merges hotfix to develop and main)