Git Guide by Morvan

Link: <https://www.youtube.com/user/MorvanZhou/playlists>

|  |  |  |
| --- | --- | --- |
|  | Git Command | Usage |
| Git Initialize | git config –global use.name “Greg” |  |
| git config –global use.email “yuangreg@gmail.com” |  |
| Git log | git log --oneline |  |
| git reflog | Show all the logs, include reset |
| See difference of current vs committed | git diff | Unstaged vs Last Commit |
| git diff --cached | Staged vs Last Commit |
| git diff HEAD | Unstaged+Staged vs Last Commit |
| Commit Management | git commit --amend --no-edit | Commit changes to the last commit |
| git reset 1.py | From stage to unstaged |
| git reset --hard HEAD | Reset to last commit |
| git reset --hard 2a17846 | Reset to commit 2a17846  HEAD also set to 2a17846 |
| git checkout 2a17846 -- 1.py | Reset 1.py back to commit 2a17846 |
| Branch Management | git log --oneline --graph | Show branch graph |
| git branch | View all branches |
| git branch a123 | Create “a123” branch |
| git checkout a123 | Switch to “a123” branch |
| git branch -d a123 | Delete “a123” branch |
| git checkout -b a123 | Create “a123” branch and Switch to “a123” branch |
| Git Merge | git merge a123  git commit -m “solve conflict” | Merge “a123” into current branch.  Solve conflict by hand and then commit |
| git merge --no-ff -m “keep info” a123 | Keep merge info when merge |
| Git Rebase | git rebase master  *(currently on branch a123)*  git add 1.py  git rebase --continue | Rebase branch a123 to current version in master branch  Solve conflict by hand and then add the files  Continue with rebase |
| Git Stash | git stash | Stash changes and current branch reset to HEAD |
| git stash pop | Pop stash |
| Remote Origin | git remote add origin https://github.com/user/repo.git | Set remote origin |
| git remote set-url origin https://github.com/USERNAME/REPOSITORY.git | Change remote origin |
| git push -u origin a123  git push -u origin master | Push code to origin |
| git pull  (= git fetch + git merge) | downloads new data, and it also directly integrates it into your current working copy files |
| git fetch | downloads new data, but it doesn't integrate it into your working files |

Example:

*1. Working in dev branch, make some changes, stash it.*

>> git checkout dev

>> git stash

*2. Create a new branch “boss”, make changes*

>> git checkout -b boss

>> git commit -am “job from boss”

*3. Merge changes into master*

>> git checkout master

>> git merge --no-ff -m “merge boss” boss

>> git commit -am “solve conflict”

>> git branch -d boss

*4. Return to dev branch and restore stash*

>> git checkout dev

>> git stash pop