# Git

[Git](#header-n0)  
 [Undo / Amend commit](#header-n15)  
 [Unstage file](#header-n17)  
 [remove local untracked files from the current Git branch](#header-n23)  
 [Issue: The following untracked working tree files would be overwritten by merge](#header-n24)  
[Git Branch](#header-n29)  
 [show remote branches](#header-n35)  
 [Merge branches](#header-n37)  
 [Delete branch](#header-n39)  
 [Git get remote branch to local](#header-n42)  
 [Remove files](#header-n55)  
 [Git Reset](#header-n67)  
 [Update Local Branch](#header-n71)

**This is a simplified usage of git**

1. git commit --amend -m "amended comment"
2. git checkout -d deprachted\_branch
3. git checkout -b new\_branch
4. git log --graph: view the merge graph from different branches
5. git stash / git stash pop / git stash list

## Undo / Amend commit

git commit --amend

### Unstage file

git reset HEAD <file>

Unmodify file

git checkout -- <file>

## remove local untracked files from the current Git branch

## Issue: The following untracked working tree files would be overwritten by merge

common command:

git clean -fdx

git clean  
-f: force to remove  
-d: remove directories  
-x: remove all untracked files(ignored files and files unknown to Git)  
-X: remove only files ignored by Git

# Git Branch

***Branch can be seen as a pointer to one of those commits.***

**HEAD:** points to the current working branch.

**Creating a new branch and switching to it at the same time**

git checkout -b <newbranchname>

The branch and the HEAD branch point to the same series of commits.

### show remote branches

git remote show <remote>, i.e. git remote show origin

## Merge branches

git merge <branche-to-be-merged>

## Delete branch

git branch -d <branch>

## Git get remote branch to local

Common command:

git checkout --track <remote>/<branch>

which is equivalently to git checkout -b <branch> <remote>/<branch>

if the branch doesn’t exist and exactly matches a name on only one remote,

git checkout <branch>

If want to rename the branch from remote:

git checkout -b <rename> <remote>/<branch>

git checkout -- file

git branch -vv  
git fetch --all: fetch all the changes on the server that local doesn't have yet.

force switch branch

git checkout -f <branch>

## Remove files

1. remove files from projects

git rm <filename>

1. remove it from stages area and not tracking anymore

git rm --cached <filename>

Git: fatal: Pathspec is in submodule  
https://stackoverflow.com/questions/24472596/git-fatal-pathspec-is-in-submodule

git rm --cached directory  
git add directory

## Git Reset

--soft  
--mixed: (the default option)  
--hard

## Update Local Branch

Force update: git reset --hard <remote>/<branch>  
Keep commit: git push <remote>  
Keep changes: git merge --abort, git reset --merge, git pull  
git stash && git pull && git stash pop