**TL;DR version**

// delete branch locally

git branch -D localBranchName

// delete branch remotely

git push origin --delete remoteBranchName

// add new local repo to github

git remote add origin [git@github.com:username/new\_repo\_in\_github](mailto:git@github.com:username/new_repo_in_github)

git commit -m “xxxx”

git push -u origin master

// Show the origin of a project

Git remote show origin

Git remote get-url origin

// how to add a .gitignore file in a local repo

Touch .gitignore

Write into the .gitignore the relative path of all file and repo that should not be tracked

git rm -r --cached .

git add .

git commit -m ".gitignore is now working"

|  |
| --- |
| **How to stash changes in git**  **https://opensource.com/article/21/4/git-stash#:~:text=Retrieving%20stashed%20changes,them%20to%20the%20working%20copy.** |
| * [**https://opensource.com/article/21/4/git-stash**](https://opensource.com/article/21/4/git-stash)   **git stash => all changes in tracked files**  **git stash -u or git stash --include-untracked => stash changes on untracked files**  **git stash -a or git stash –all => stash all untracked and ignored files**  **git stash -p or git stash –patch => stash specific files**  **git stash show -p stash@{0} => show all diff into a specific stash**  **git stash show stash@{0} –-stat => list all filenames into stash**  **git restore --source=stash@{0} -- <filename> => restore stash only for that specific file**  **git stash delete 0 => delete stash in index 0 (stash@{0})**  **git stash drop =>** delete the stashes one by one (starts from top)  **git stash clear** => remove all of the them |

|  |  |
| --- | --- |
| **How to add modification to last unpushed commit** | |
| **Unpushed commit** | 1. **git add** 2. **git commit --amend --no-edit (without msg edition)** 3. **git commit –-amend (with message edition)** |
| **Pushed commit** | 1. **git add** 2. **git commit –-amend –no-edit** 3. **git push -f** |

|  |
| --- |
| **How to list all files changed in a commit** |
| **git diff-tree --no-commit-id --name-only -r bd61ad98** |

|  |
| --- |
| **How to get all commit that have involved a specific file** |
| **git rev-list HEAD --oneline FileName** |

|  |
| --- |
| **How to remove specific file from a commit** |
| 1. git reset --soft HEAD~1 (Undo the last commit but keep all the changes) 2. **git rm path/of/file or git reset HEAD path/of/file (unstagged or rm file from git)** 3. **git add .** 4. **git commit (commit change to a new commit) or** 5. **git commit –-amend (commit change to last commit)** |

ref (https://www.nobledesktop.com/learn/git/git-branches)

|  |  |
| --- | --- |
| **Push commit** | |
| Git push -u origin brancheNameOnRemote | If the local branch does not have remote branch. BranchNameOnRemote is the new corresponding branch on remote server |
| Git push -u origin HEAD | If local branch does not have remote branch. In this case, remote branch name will have the same name as the local one |
| Git push | If the remote branch already exist |

|  |
| --- |
| Git rebase |
| https://www.internalpointers.com/post/squash-commits-into-one-git |