**Git & GitHub Crash Course for Beginners** Feb 5, 2017

<https://www.youtube.com/watch?v=SWYqp7iY_Tc>

Most frequent used GIT command

|  |  |
| --- | --- |
| **$git –version** |  |
|  |  |
| **$git flow init -d** |  |
|  |  |
| **$git flow feature start <feature\_branch>** |  |
|  |  |
| **$git checkout develop** |  |
|  |  |
| **$git branch** |  |
| **$git checkout feature/<feature\_branch>** |  |
|  |  |
| **$git add .** |  |
| **$git commit -m “commit message”** |  |
| **$git push** |  |
|  | |
| **$git push -u origin feature/<feature\_branch>** | |
|  | |
| **$git branch -d <local\_branch>** | |
|  | |
| **$git branch -d -r orgin/<feature\_branch\_name>** | |
| **git flow feature finish my\_feature\_brach** |  |
| $git --version |  |
| git$git flow init -d | Initialize Local Git Repository with default settings |
| $git status | Check Status of working Tree |
| $git commit -m “commit message” | Commit Changes with commit message |
| $git push | Push From local To Remote Repository |
| $git pull | Pull Latest from Remote Repository |
| $git clone | Clone Repository Into a New Directory |
| $git add . | Add all in current directory |
| $git rm <filename> |  |
| $git checkout <my-branch> | Switch to branch <my-branch> |
| $git push -u origin develop  $git push -u origin feature/<f\_branch\_name> |  |
| $ git branch -d <local-branch> | To delete a local branch |
| $git push remote\_name -d remote\_branch\_name | To delete a remote branch  git push origin -d test |
| Git remote prune origin, as suggested in the other answer, will remove all such branches. That’s probably what you would want in most cases, but if you want to just remove the particular remote-traching branch, you should do: | |
| $git branch -d -r origin/<branch\_name> | |
| $ git fetch -p | Which fetches and prunes all origins |
| $git push |  |
| $git clone <https://brustoloniyj@bitbucket.org/brustoloniyj/my-gitdemo.git> |  |

Distributed Version Control System (Decentralized), require internet connection for remote

|  |  |
| --- | --- |
| $git --version |  |
| $git init | Initialize Local Git Repository |
| $git add <file> | Add file(s) To Index (Staging area) |
| $git status | Check Status of working Tree |
| $git commit | Commit Changes in Index |
| $git push | Push From local To Remote Repository |
| $git pull | Pull Latest from Remote Repository |
| $git clone | Clone Repository Into a New Directory |
| $git add \*.html |  |
| $git add . | Add all in current directory |
| $git branch <my-branch> | Create a branch from current |
| $git checkout <my-branch> | Switch to branch <my-branch> |
| $git merge <my-branch> | Merge <my-branch> to current |
| $git remote add origin <https://brustoloniyj@bitbucket.org/brustoloniyj/my-gitdemo.git>  $git push -u origin develop  $git push -u origin feature/butler\_labs |  |
| $git checkout feature  $git merge main  Or --- a one-liner  $git merge feature main | Merge the main branch into feature |
| $git checkout feature  $git rebase main | Rebase the feature branch onto main branch |
| **Interactive Rebasing**  **This is used to clean up a messy history before merging a feature into main** | |
| $git checkout feature  $git rebase -i main | To begin an interactive rebasing session, pass the I option to the git rebase command: |
| pick 33d5b7a Message for commit #1 pick 9480b3d Message for commit #2 pick 5c67e61 Message for commit #3 | This will open a text editor listing all of the commits that are about to be moved: |
| $ git branch -d <local-branch> | To delete a local branch |
| $git push remote\_name -d remote\_branch\_name | To delete a remote branch  git push origin -d test |
| $git push |  |
| $git clone <https://brustoloniyj@bitbucket.org/brustoloniyj/my-gitdemo.git> |  |
| $git rm --cached index.html |  |
| git config --global user.name 'brustoloniyj' | |
| $ git config --global user.email 'brustoloniyj@askesis.com' | |

* + Download from git-scm.com
  + Use “Git Bash” from file explore by right click mouth
  + $git commit (will prompt you to an editor)
    - Type i for insert
    - Type Esc and type “:wq” to quite the editor
  + $git commit -m ‘changed app.js’ (will commit with your message in ‘’)
  + $touch .gitignore (to create ignore file)

Add files, type of file, or directories you do not want to be included in repository in this file

Features for GitLib

GitLab Beginner Tutorial 2 | Getting started with Git Commands

279,789 views

Sep 23, 2019

<https://www.youtube.com/watch?v=OWaZXtgq28c>

Install GIT

[lesxxxxx@yahoo.com](mailto:lesxxxxx@yahoo.com)

$git config –global user.name “lesxxxxx”

$git config –global user.name

$git config –global user.email [lesxxxxxxx@yahoo.com](mailto:lesxxxxxxx@yahoo.com)

My Gitlab Test, create folder MyGitlibTest and go to folder root type ‘cmd’

To open command window or open bash window

$git init //This will create a .git folder, which initial git

Or

$git clon https://gitlab.com/MyGLTest/first-gitlab-project.git

$git colon

$git status

Add a foldergit

$git add ‘template config’/git push

$git add .

$git commit -m “my first commit”

$git push –u https://gitlab.com/MyGLTest/first-gitlab-project.git master

Great branch and create merge request

>git checkout -b update-readme // make change to readme.txt

>git add.

>git commit -m "commit branch update-readme"

**Full Gitflow init workflow example tutorial**

<https://www.theserverside.com/blog/Coffee-Talk-Java-News-Stories-and-Opinions/init-Gitflow-example-workflow-tutorial>

**>Change the master and develop branches to main and development. We leave the rest at their defaults.**

**$git flow init -d**

brusyj@BKSQ457J5S2 MINGW64 /c/codebaseroot/python-allscripts-ccd-extract (develop)

$ git flow init -f

Which branch should be used for bringing forth production releases?

- develop

- master

Branch name for production releases: [master]

Which branch should be used for integration of the "next release"?

- develop

Branch name for "next release" development: [develop]

How to name your supporting branch prefixes?

Feature branches? [feature/]

Bugfix branches? [bugfix/]

Release branches? [release/]

Hotfix branches? [hotfix/]

Support branches? [support/]

Version tag prefix? []

Hooks and filters directory? [git branch -a] C:/codebaseroot/python-allscripts-ccd-extract/.git/hooks

brusyj@BKSQ457J5S2 MINGW64 /c/codebaseroot/python-allscripts-ccd-extract (develop)

Notice there are only two branches after a “git flow init” command.

$ git branch -a

\* develop

master

remotes/origin/HEAD -> origin/master

remotes/origin/master

brusyj@BKSQ457J5S2 MINGW64 /c/codebaseroot/python-allscripts-ccd-extract (develop)

A picture containing chart

Description automatically generated

The Gitflow release branch is made from the develop branch and gets merged into both master and develop when finished.

**>This creates a new feature branch named “feature\_branch”**

**(Read the article using link above, for the complete Gitflow example)**

* **Git flow feature start my\_feature\_branch**

**…Add file , update files**

**…Git commit your code on feature branch…**

**>git commit -m “Add …”**

**>** git push -u origin feature/< **my\_feature\_branch** >

**Crete merge request**

**To Finish Feature Branch, This merges and deletes it**

**>git flow feature finish my\_feature\_brach**