**Git Command Line Reference**

Michael Wang, 27 January 2017

**# set up**

git --version

git config --global user.name "ozweatherman"

git config --global user.email "ozweatherman@yahoo.com.au"

git config --list

**# help**

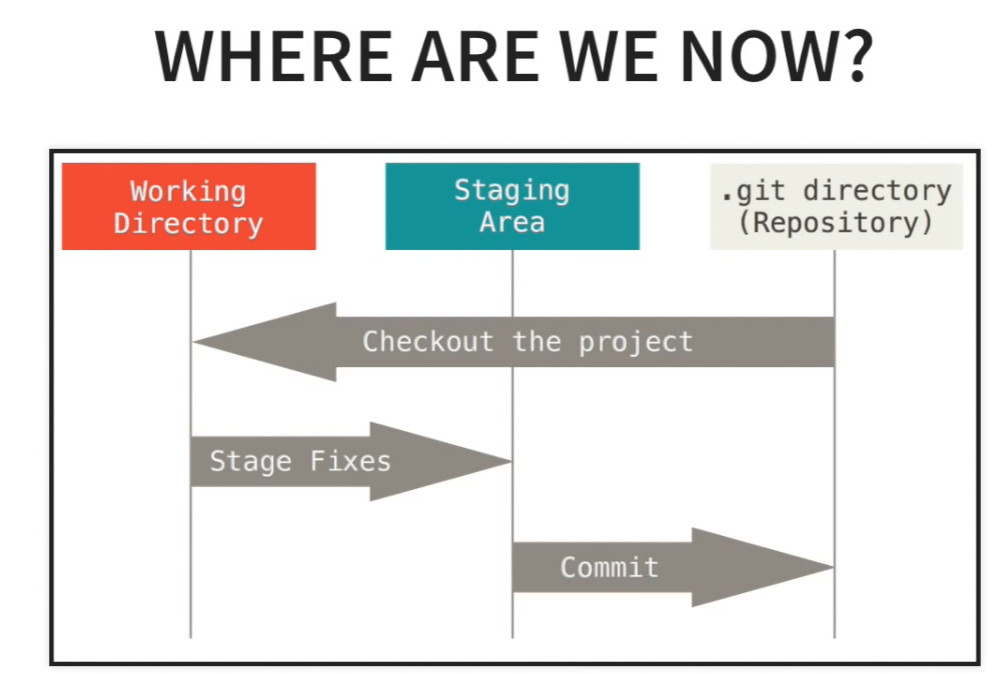
git help <verb>

**# For existing proj on local machine**

git init # track all files in the wdir, .git created

git status

touch .gitignore #create .gitignore file (a text file) to include files to be ignored



git add –A # add all files in staging area

git add <filename> # add <finename> to the staging area

git reset <filename> # remove <filename from staging area

git reset # remove everything from staging area

git commit –m “msg” # commit to local repo. “msg” should describe what’s in the changes

git log # see what’s been commited

**# Cloning a Remote Repo**

git clone <url> <dir>

eg. git clone <https://github.com/someone/repo.git> . # clone the remote repo to current dir.

**# View Information about the Remote Repo**

git remote –v # view info of the repo

git branch –a # view all branches remotely and locally

**# Pushing Changes**

git diff # see the change made to the code

git status

git add –A

git commit –m “made changes on price”

git pull origin master

git push origin master # push the changes to remote repo master branch

**# Common Workflow**

1. **Create a branch for desired feature**

git branch sub-branch # created a local branch called ‘sub-branch’

git branch # list all local branches, notice ‘master’ is the active branch

git checkout sub-branch # work on the sub-branch

git branch # list all local branches, notice the active branch becomes ‘sub-branch’ instead of ‘master’

1. **Make changes on files then commit to the sub-branch**

git add –A

git commit –m ‘changes on sub-branch’ # no effect local ‘master’, no effect on remote repo

1. **Push sub-branch to remote**

git push –u origin sub-branch # ‘-u’ establish link between local branch and remote branch

git branch –a # see all branches including local and remote

1. **Merge branch with master**

git checkout master # switch to branch master

git pull origin master

git merge sub-branch # merge the sub-branch into master

git branch –-merged # make sure it’s merged

git push origin master

1. **Removing a branch**

git branch --merged #make sure ‘sub-branch’ merged into master

git branch –d sub-branch # delete ‘sub-branch’ locally

git push origin -- delete sub-branch # delete ‘sub-branch’ remotely

git branch –a # check all branches at local and remote

Note on: ‘*git remote add origin https://github.com/someone/somerepo.git’*

if you get a msg ‘fatal: remote origin already exists’, this means you’ve already establish a remote connection called origin. To remove a remote repository you enter, use ‘*git remote rm origin’*

**Faster Example**

* *git branch substract*
* *git checkout substract*
* *git status*
* *git add –A*
* *git commit –m “made some change on substract function”*
* *git push –u origin substract*
* *git checkout master*
* *git pull origin master*
* *git merge substract*
* *git branch --merged*
* *git push origin master*
* *git branch –d substract*
* *git push origin -- delete substract*

*Reference:* [*https://www.youtube.com/watch?v=HVsySz-h9r4*](https://www.youtube.com/watch?v=HVsySz-h9r4)