

touch .gitignore

git init # initialize git repository

rm -rf .git # remove git repository

git config --list # list configurations

git config --global user.name ' ' ,

git config --global user.email ' ' ,

local .git/config	--local
global ~/.gitconfig	--global
System C:\ProgramData\git\config	--system

git add . OR git add -A # both tracked & untracked  
--a

git add -u # only tracked

git add --ignore-removal # ignored removed files

git diff # compare staging area with working directory

git diff --staged # compare staging area with previous commit

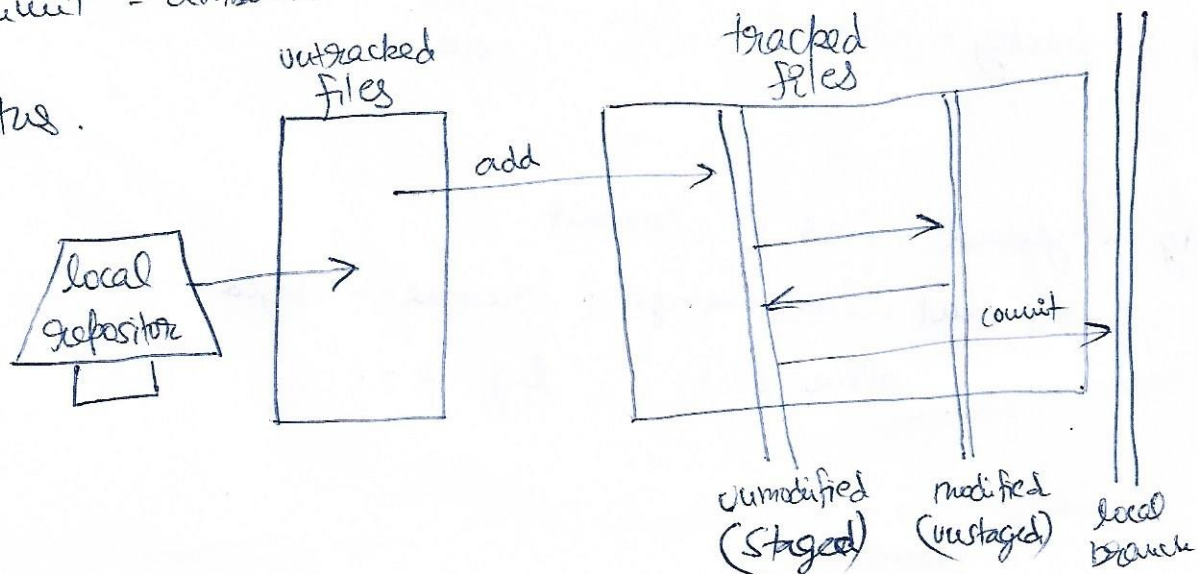
git commit -m 'message'

git commit -a -m 'direct commit' # skips staging area only for tracked files.

git commit --amend -m "an updated message"

git commit --amend --no-edit

git status.



git remote      git remote -v  
git remote add origin git@github.com: EashanKausik/github-for-dummies.git  
# should have permission to remote  
git push --set-upstream origin master

Editing  
git rm hello.py # removes file  
git restore --staged hello.py # to unstage  
git restore hello.py # restore working directory  
git rm --cached hello.py # untrack  
git mv hello.py hello-1.py # rename  
git checkout -- hello.txt # undo to prev commit only if unstaged  
git checkout -f # all prev. to commit

log  
git log  
git log -p -3  
git log --stat  
git --pretty = one  
                 short  
                 full

git log --since = 2 days  
                 2 weeks  
git log --pretty = format: "%h -- %au"  
                                 ae

alias  
git config --global alias.ci commit  
git config --global alias.unstage 'restore --staged --'  
                 alias.last 'log -p -1'



## Branching

(-gitignore is not affected)

git checkout -b develop # develop master branch  
git checkout master # switch to master  
git branch # show branches -v --merged --no-merged  
git merge develop # merge branch

## Merge conflicts

Conflicts arise when two people have changed same line of code  
or if one developer deleted a file while other was modifying it.

git branch -d develop # want del if not merged  
-D # forced del

"long run branches"  
"topic branches"

git push origin develop # push branch to remote repository  
should be in develop  
:development # will name development in remote

git push -d origin develop # del branch in remote repository

## Re-writing history

git checkout --hello.py # uncommitted changes # works on unstaged  
git reset id # committed changes # directly committed makes new commit  
git reset -h id # explicitly used to commit  
git reset --hard id # del commit upto that id