Notes from thenewboston git tutorial

The Git bash shell is like any other linux/unix shell, so

$pwd prints the working directory,

$cd ~ changes to the home directory,

$ls -la lists all files including hidden ones, etc. $clear clears the screen

$[up arrow] shows previous command

Setting up

$git config --global user.email "you@example.com"

$git config --global user.name "Your Name"

Initialize an empty repo in the current directory

$git init

Add all directory changes to the project staging area

(working file>>staging area>>repo)

$git add .

Add a single file to the project staging area

$git add aSingleFile

Commit the changes from staging area to repo

$git commit -m "message here"

Add changed files to staging area and commit them

$git commit -am "message here" (don't use this if changes include things like new, deleted, or moved files)

Compare repo and working directory

$git status

View commit history

$git log

$git log --author="A Name" (commits by A. Name only)

See differences between unstaged files in the working directory and repo

$git diff

See differences between staged files and repo

$git diff --staged

Delete a file

$git rm aFileName

$git commit

(rm removes file in working directory and puts the change in staging area ... must still commit)

Rename or move a file (within working directory)

$git mv oldFileName newFileNameOrPath

$git commit

(mv moves/renames file in working directory and puts the change in staging area ... must still commit)

Revert a file changed in working directory to version in from last commit

$git checkout – aChangedFileName

Unstage a file

(file is still changed, still in working directory, just not staged for commit)

$git reset HEAD aFileName

Revert file to previous version

$git checkout abcd -- aFileName

where abcd is first few chars of desired version's unique commit ID, and aFileName is file being reverted. Note: working directory will change; directory needs to be staged and committed.

After creating a GitHub repository, to push an existing repository

$git remote add origin https://github.com/acolburn/notes.git

$git push -u origin master

(first line makes "origin" short version for URL we're pushing to, a repo called "notes")

(second line pushes everything)

Clone existing GitHub repository

[cd to desired location]

$git clone https://github.com/acolburn/notes.git Notes

where Notes is a folder to hold files

Note: new folder is NOT a repo