**Git Notes**

Step 1:Download Git Bash

Step 2: Set up username and email in the git bash terminal

git config -global user.name “June Li”

git config -global user.email [xxx@gmail.com](mailto:xxx@gmail.com)

Step 3: Initialize a local repo

* Go to local directory or make a directory: cd dir
* Point to the local directory and initialize git: git init
* Add all file in the local dir to git: git add .
* Show status: git status
* Check log: git log
* Commit: git commit -m “comments”
* Enter commands on notepad++
* Check log: git log
* git remote add origin: <https://github.com/miaojunlee/xxx>
* Update local repo to make sure local repo has up to date with the remote repo: git pull origin master (always do a pull before push)
* Push the remote repo (origin) under the master branch: git push -u origin master

**Clone**

* Git clone repo url

**Merge**

To merge local repo with an existing git repo

* git remote add origin: <https://github.com/miaojunlee/xxx>
* Run a pull first: git pull –allowed-unrelated-histories
* Set branch: git branch –set -upstream-to=origin/master master
* Push files: git push

To edit global config file

* Go to master directory: cd
* npp .gitconfig

To combine add and commit into one line: git commit -am “comments”

To track a file: git ls-files (ls list files in the local directory, git ls-files lists files in git staging)

To remove a stage change: git reset HEAD filename

**Recover File**

To recover a deleted file but still available in staging: git checkout filename

To backout changes: git checkout --filename

To recover a deleted file that has also been committed:

1. git log to find the key
2. git checkout 04f788928d8b2b9ea5c1cc6b945b4ce15aa1f033^ -- testv2.txt
3. to read the file from git: git show HEAD:filename

* git rm filename
* To un-staged the delete: git reset HEAD filename (spell it out) (this is considered a staging change)
* To recover the file from staging: Git checkout -- filename

**Rename**

* To rename the current file name: git mv oldfilename newfilename
* To rename the name at the system level: mv oldname newname
* Git add -A (recursively add changes but also update any file that has been renamed, moved or deleted)

**Delete File**

* Delete a file not being tracked by git: rm file
* Delete a file being tracked by git: git rm file, followed by git commit

Using bash command:

* rm filename
* git add -A
* git commit

To remove folder: rm -rf foldername (be very careful with this change)

**History**

* git log
* git log --abbrev-commit
* git log --all --oneline --graph –decorate
* git log e36c22c...55bd792
* git log --since=”3 days ago”
* git log --filename
* git show commitID

**Alias**

* git config --global alias.hist “git --all --oneline --graph --decorate”

**Ignore Unwanted File**

**…or create a new repository on the command line**

echo "# tensorflow\_bootcamp" >> README.md

git init

git add README.md

git commit -m "first commit"

git remote add origin https://github.com/miaojunlee/tensorflow\_bootcamp.git

git push -u origin master

**…or push an existing repository from the command line**

git remote add origin https://github.com/miaojunlee/tensorflow\_bootcamp.git

git push -u origin master

**…or import code from another repository**

You can initialize this repository with code from a Subversion, Mercurial, or TFS project.

You can also use git init project\_name

List all files including hidden files : ls – al

Rm -rf foldername

* To copy repo from others: find a repo and click on fork; go to Git Bash and type in git clone url
* To go to the git folder: cd .git/
* To remove a file from git staging: git rm –cached filename
* To ignore the files from git add ., type: npp .gitignore (list the files want to ignore in the notepad++)
* To manually add a file that listed in the .gitignore, use: git add -f filename
* To remove a repo on Github, go to Danger Zone and delete the repo; also, on Git Bash: git remove remote origin
* When encounter remote commit has more files than local, use: git pull