**How to GitLab**

Instructions via Gitlab: <https://docs.gitlab.com/ee/gitlab-basics/start-using-git.html#open-a-shell>

Instructions via Github: <http://kbroman.org/github_tutorial/pages/init.html>

Download git bash: <https://git-scm.com/downloads>

* This comes with a shell client to run all the commands
* For non Windows you need to install git separately and can just use terminal

Create an account on Gitlab

Find/create an SSH key pair:

* Create: <https://docs.gitlab.com/ee/ssh/>
* Paste into gitlab: <https://gitlab.com/profile/keys>

Navigate to a folder where you want to clone the repo

Open the client (which is called Git Bash in Windows):

Run (this initializes git in the folder)

git init

git clone **https://gitlab.com/kmjoshi/co-acting.git**

Navigate to ./co-acting/

Add the url as remote

git remote add origin https://gitlab.com/kmjoshi/co-acting.git

Download all the updates

git pull origin master

branches are used to manage different parallel versions. master is as the namesake the main version.

Use branches when pushing to master, then merge. Always work on branches first (or more so when actually writing production code).

Branches: <https://git-scm.com/book/en/v2/Git-Branching-Branches-in-a-Nutshell>

Merges: <https://git-scm.com/book/en/v2/Git-Branching-Branches-in-a-Nutshell>

Other tips:

* Pull commits from origin and merge conflicts locally before pushing
* Work on branches first!

Cheatsheet: <https://www.atlassian.com/git/tutorials/atlassian-git-cheatsheet>

A lot of git is confusing and I never got around to learning it, because you hardly need it.

Now you have the entire repo! Tread carefully!

**Other common commands**

add all edited files

git add .

add a specific file

git add filename.ext

commit files

git commit -m ‘nice message’

push to online repo (cannot push until changes are committed with a message)

git push origin

**more commands:**

git status - inspect staging area and working directory

git diff - show the difference between working file and staging file

git log - history of previous commits

git checkout HEAD filename - discard changes in working

git reset HEAD filename - unstage filename

git reset SHA - reset to a previous commit (use git log to look up SHA)

git branch - list all branches

git branch branch\_name - create a new branch

git checkout branch\_name - switch to a branch

git checkout -b branch\_name - create and switch to branch

git merge branch\_name - merge branch onto master

git branch -d branch\_name - delete branch

git remote -v - list the remote locations of a git repo

git fetch - update repo from origin

git merge origin/master - update local master with fetched repo

git pull - fetch and merge together

list all files in git repo

git ls-tree --full-tree -r --name-only HEAD

remove all files from staging

git rm -r --cached .

dry run | all do the same thing, the former in an editor

git push --dry-run

git diff --stat --cached origin/master

git ls-files