**Intro to GIT #1**

Have a look at spider- one ide

Jetbeam – pycharm

Stack overflow

Two types- centralized distributed version control system

Repository – sthg on your machine and something on your cloud- these two need to be in sync

Git’s centralized distributed system so the machine version and cloud version are always the same

Git vs github

Git is the software we are using

Github is the facebook for developers- its where all your code is public

Git features

* Distributed
* Compatible
* Non-Linear
* Branching
* Lightweight and speed

Git was written in C- makes it super fast

New repo: <https://github.com/prichakrabarti/Pri_GA201807.git>

Git

Go to empty folder

Open git bash

Type git init

You can also access a folder using the command line typing cd and the folder path that you are interested in initializing

Hidden file -> .git

Then go to github, clone or download and get the url for the repo you just created

Go back to git bash

Git remote add origin + the url in quotes

Hit enter

This connects git and github

Git pull origin master -> pulling files from master branch of the github page we have! So now our local folder has the readme.md file including the edited changes you just made.

[[Create a branch first, view changes then merge these changes onto the master- this is the best practice]]

This has pulled files from Github to your local folder

You need to send to the index first

This is the staging area

1. Add to staging
2. Then commit then -> once you commit you have to add a message
3. Only then you can push your file to the centralized repo

So first do git add “filename” -> you don’t need to manually add just type the first three letters and hit the tab key and it will autocomplete

Hit git status -> now the file you have added comes to green – this validates that we have just indexed this file. The file is now indexed- its in the staging area.

Now you are ready to commit these changes. Commit is where you put the reason for your commit- you need to know who has done what and why this change was necessary.

Side note: You can change the origin or the url from github

Git remote set-url origin “yourlink”

To manually set your email and username type the following two commands

Git config –global user.email “chakrabartipri@gmail.com”

Git config –global user.name “Pri Chakrabarti”

Then you commit!

Git commit -m “Our first commit”

Then create multiple files

Create 4 copies in local folder

Git add -A 🡪 this adds everything in your working directory to your staging area

Anytime you want to check whats happening to git status -> so now the Add has added everything to the staging area- they have been indexed

Commit multiple files -> git commit -a “My message!”/ git commit -m “My message!”

Run git log – this shows all the changes you have done until then for the day

To push multiple files with one message its git commit -m “Message for all files!”

To push multiple files with one message (not separate one for each file) its git commit -a “Message!”

Dir – windows command to see what files you have in your directory

STEPS

1. Create a folder where you will store all your files locally
2. Right click inside the folder and open git bash
3. In git bash type git init- this initializes your folder
4. Now you need to connect your local repo to the centralized repo on github -> you can do this through the command git remote add origin + url from github inside quotation marks. Eg: $ git remote add origin “<https://github.com/prichakrabarti/Pri_GA201807.git>”
5. Then pull in all the files from the centralized repo to your local repo – git pull origin master. To check you should now see the readme.md inside your local folder
6. Now you want to do the opposite- take a file locally created and push it to the repo. This is a multi step process. First you need to stage the file, then commit, then git push origin master command.
7. Staging files- git add “filename.extension”
8. Commiting files git commit -m “Message!”
9. Git push origin master

You shouldn’t directly push to the master branch. The solution is to create a clone of the master branch and create a new branch- this is called branching. Do whatever changes in your branch- then when the changes have been signed off on you merge this change with your master

Merge conflict is when 2 people have edited the same branch and try to push to master. When you are not working on the latest copy of the branch it causes merge conflict.

This is where we use something called rebasing.

Creating a branch

On gitbash

git branch “first\_branch” -> or whatever you want the branch name to be

then:

git checkout “branchname” -> in this case git checkout firstbranch

now hit enter and you are no longer in your master branch

git add -A

git commit -m “Message!”

then its on the first branch

git checkout master to go back to the master branch

now do dir again to see whats in the master branch

(this is all locally)