1. Git :

* Made by Linus Torwalds
* Does Version Control
* Creates a .git hidden folder in pwd(present working directory) to store snapshots of only the changes commited by programmer along with their name and email id and time of commit.
* Saves storage , avoids giving bad names to files and folders.

Like project\_final1,project\_final\_final,…

* Almost every operation is local (except pushing/pulling files from online forums)
* Has integrity :

Creates SHA1(64bit) encryption for files while sending and receiving and matches both. But SHA1 encryption is now cracked and hence not suggested.(checksum)

1. Local VCS :

* We are storing all our copies on our laptop(in a database). So our laptop damaged => Data lost

Centralized VCS :

* We are storing all our copies on a centralized server. So our laptop damaged => data is safe , but server damaged => our data lost.

Distributed VCS :

* Individual computers and centralized server both have copy of data. So if any PC or server gets damaged => no problem.

1. GitHub : A online forum which is free to use , fast and widely used to store data , files and codes. Created by Microsoft.

But the code is visible to Microsoft and hence it can use it’s code(like in open AI – Chat gpt ), so private companies don’t use this.

1. BitBucket : Same as GitHub but they make sure that GitHub server is given to the company and so no other company can see/use their software.
2. Commands :

* pwd : present working directory
* cd <folder name> : If that folder is there in pwd , then changes current directory to that folder.(Setting pwd relative to current pwd)
* cd <file name> : Gives an error.

Say you are in desktop folder. cd HTML or cd HTML/, cd ‘SEM 3’ will take you in respective folders

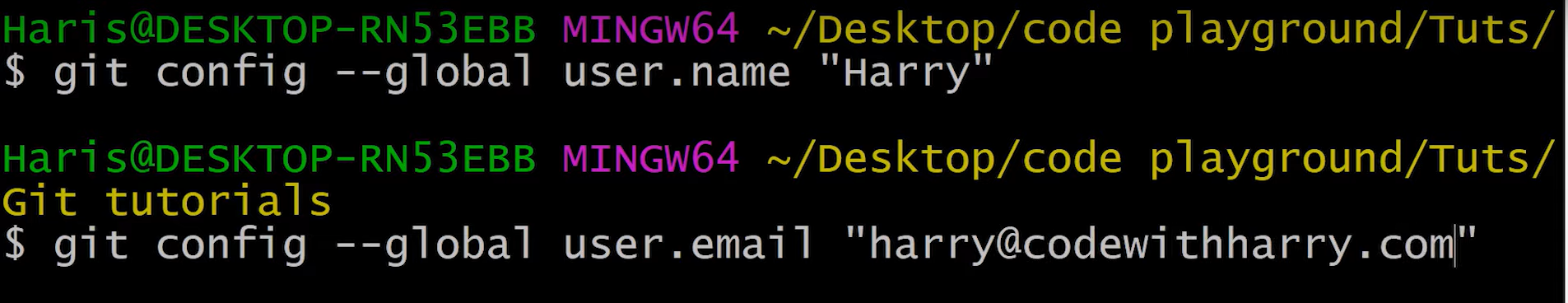
cd HTML/'Chapter 0' : takes you in chapter 0 folder in HTML folder.

use TAB to auto-complete.

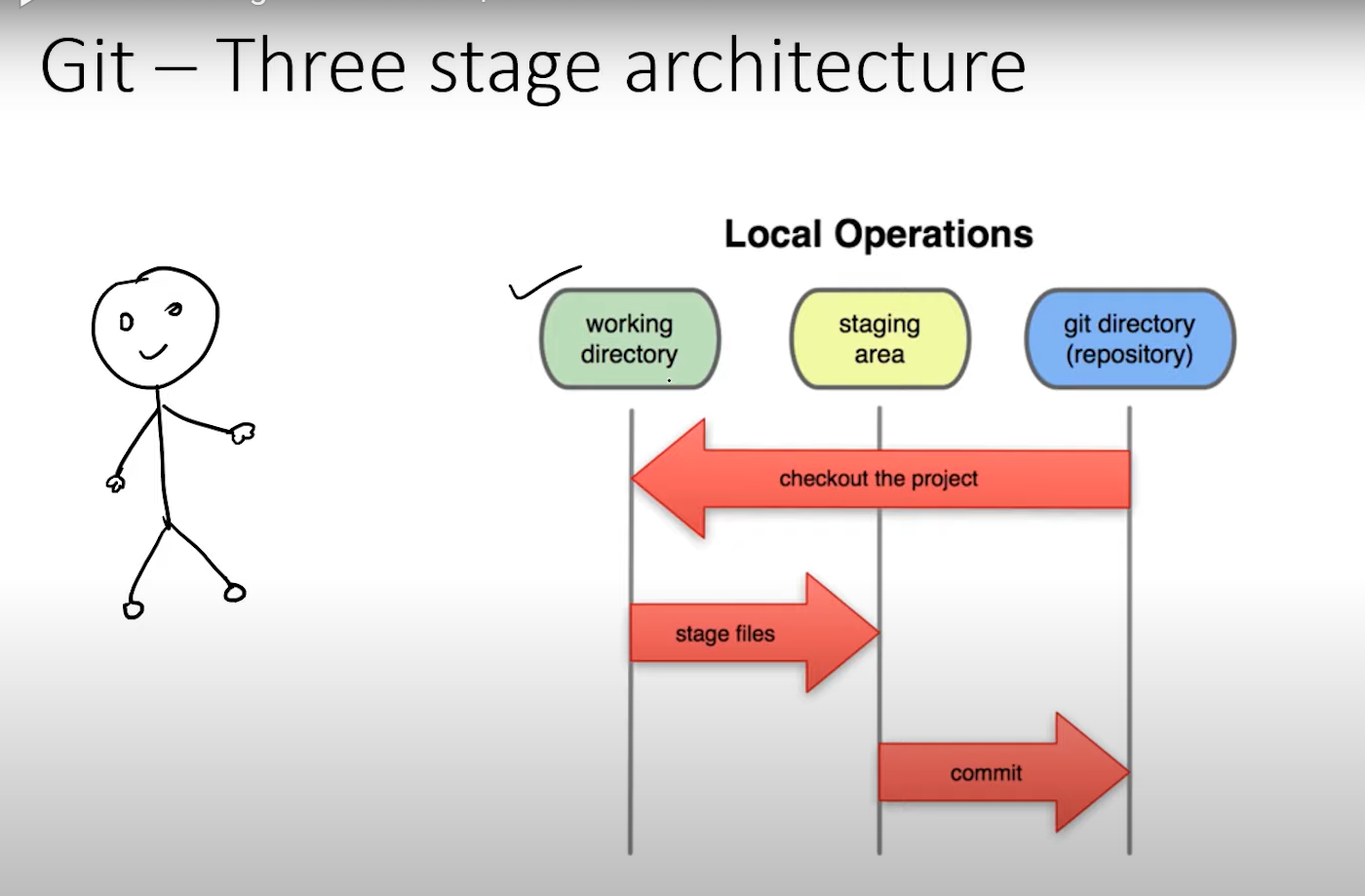
* cd C:\\Users\\.... : Sets pwd to that address
* ls : Lists all the files and folders in that directory
* cd .. : Goes one folder out of pwd

1. Git Commands :

* For the 1st time do this :



* git config user.name -> Gives name
* git config user.email -> Gives email
* git config --list -> Gives the entire list.
* git status :
* If we get fatal error -> not initialized as git repository (Do git init)
* If we get other message : nothing to commit , modified , untracked ,… -> already initialized as a git repository.
* If we re-initialized as git repo , our prvious ones are lost and it is considered as a new get repository.
* 3 Stage Architecture of Git :



Add files(stage them) : working 🡪 staging area

Staging area actually has 2 parts : modified and unmodified. Once we commit , only the modified area files go in git repo and then all the files are returned to unmodified area.

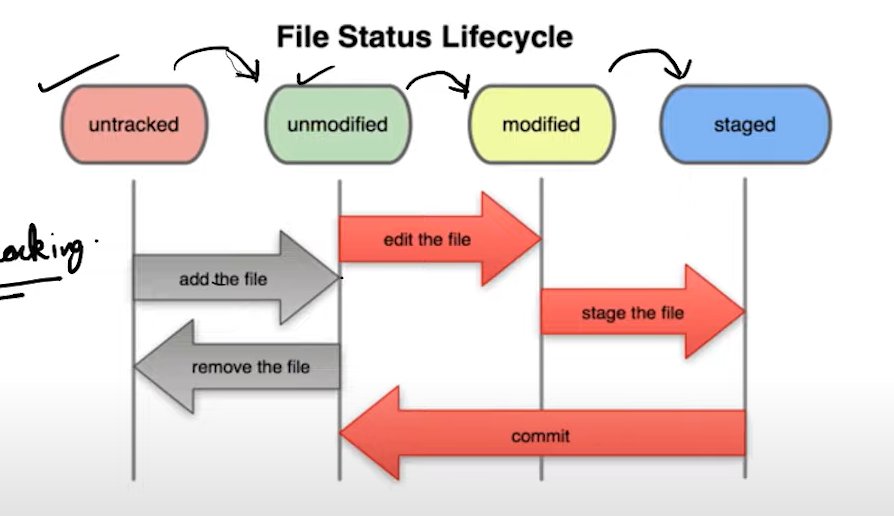
* rm – rf .git : remove recursively files and folders in .git folder(basically delete the .git folder)
* git clone <paste the link to clone> : Will download this folder in pwd

e.g. git clone <https://github.com/sannidhiteredesai/BDD-in-Python.git>

* cd - : Goes to previous working directory(may not be 1 up of pwd)

Say our 1st pwd was Users and 2nd pwd was Chapter 0 in HTML folder. Then cd – navigates among these 2 .

* File Status Lifecycle:



* git init : all files are initially untracked.
* git add <file name> : Add those files to unmodified state.
* When me modify the added files , we take them to modified state.
* When we commit the files, the snapshot of those files and folders in modifed section only is stored and all the files and folders return to unmodified section.
* Unmodified = staging area = green colour text in git status
* Even if we edit files , the unmodified copy/staging area copy doesn’t change.It will only change when we do git add.
* .gitignore : Create this in our pwd and type the file names. Then all the file names in this will not be tracked at any point of time and hence will not be committed.So there are some files changes in which we want to ignore.
* Error.log : Ignore this particular file
* \*.log : ignores all .log files
* First/ : Ignores all folders having name First anywhere in the pwd
* /First/ : Ignores only the outermost First folder in pwd, not inner ones.
* git diff : Compares the staging area(unmodified) with working area(currently how the file looks,modified area).
* git rm file.txt : Deletes file.txt and even add this change in staging area.
* git mv file.txt renamed\_file.txt : Renames file.txt🡪renamed\_file.txt and adds to staging area.
* git rm --cached file.txt : If file.txt is already tracked then even adding it to .gitignore doesn’t help. Then we need to remove it from staging area first so as to stop its tracking. For this, use this command.
* Git log : Shows all the commit history and even what we changed at what time,email id, message,etc.
* git clone <https://github.com/pandas-dev/pandas.git> mypanda :

This will clone the repo in mypanda folder at the given location.

The folder in which we are is not git repo , but mypanda folder is a git repo.