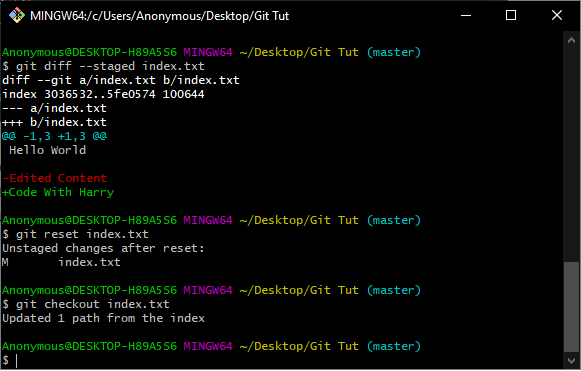
Git Commands

* “git add -A” stages all the files present in the directory.
* “git add .” stages new and modified, without deleted.

It means this command will put all the new and modified files in the staging area, excluding the deleted files.

* “git add -u” stages modified and deleted, without new.

It means this command will put all the modified and deleted files in the staging area, excluding the new files.

* git clone ‘GitHub repository link’
* We can create a file named “.gitignore” and this file contains the list of all the files or folders that we don’t want our Git to track.
* You can enter the data in the following ways,
  + index.txt - You can enter the file name with extension.
  + \*.py - You can put an asterisk symbol and then extension. Now the Git will not track any file with the .py extension.
  + folderName/ - It will ignore the files and folders in this particular directory.
* touch .gitignore --> It creates the .gitignore folder
* Now you have to open the .gitignore file in Notepad, and for that execute this command,
* notepad.gitignore
* We all know that we can make changes in the local repository, and Git keeps track of all those changes. Suppose we’ve made some changes and then we want to see what all changes are done, then we can use the "git diff" command.
* Example of modifying index.txt 
* The command “rm -rf .git” 🡪 Our git repository will no longer a git repository.
* Commits History🡪 To see the commits made on the git repo, we need to type “git log”. After typing this you can see the commits that have been made on the repo.
* Diff in Commits🡪

To see the Diff in a commit we need to use “git log -p”. It will show what has been changed on a commit. To see specific no commits with changes we need to use **“git log -p -2”**(for seeing the last 2 changes).

* Brief Summary🡪

Also we can get a brief summary of commits by typing “git log --stat”.

* If you want to see the commits on one line then type

**“git log --pretty=oneline”.**

* Customized Commit Output🡪
* If you just want to see the commits and The author then use “git log --pretty=short”. And if you want a little bit more info like who is the committer, commit message, then use **“git log --pretty=full”**.
* If you want to filter commits by time, then use this command, **“git log --since=2.weeks”**. To see the last 2 months of data type **“git log --since=2.months”**.
* We can format the output by using the formatting codes provided in Git’s website. For an example we can use this format to print just the author name and the hash : **“git log --pretty=format:“%h --an”** .
* Changing A Commit Message🡪
  + Now to change a commit. So, to change the most recent commit we need to type **“git commit --amend”** and press enter. Then an editor will open where you can change the commit message. All you need to do is change the message and then you need to close it by pressing i then esc key then type “:wq” to exit the editor. Now you have successfully edited the commit.
* Unstage and Unmodify🡪 To unstage a file use **“git restore --staged file.ext”**.
  + It will unstage the file and you can verify it by using “git status”.
  + Now let’s say we have modified the unstaged file and made some changes which were not necessary. Now the program is not working. So we have to restore that file to its previous state where it was working. To do it use **“git checkout -- file.ext”**. Now git will restore that file to its last commit state.
  + To restore your entire working directory to the previous commit use **“git checkout -f”**. It will restore your entire directory to the last commit.