**Git and Git hub**

**Git**

It is a version control system.

**Version Control system**

----------------------------------

Maintain the multiple version of a file /project

Or

Maintain the history of the changes in file.

**How to work in Git?**

Download git

**Lifecycle of git:**

**---------------------**

**Working directory-🡪staging area🡪commit**

**Working directory:**

* It is a folder where we can keep all the source files of project.
* Once we added the git init inside working directory .git folder(Local repo) will generated. It will monitor our working directory and notify the changes in the file between our working directory and local repo while checking using **git status** command.

**Staging area**

* Once we add the file by using **git add** then the file will be moved to **staging area.(ready to move to local repo)**
* Important point is even though the file has been commited .the staging area stores the cache of the file to track the changes in file compare with working directory

**Commit**

* Git commit to commit(save) the file in the Local repository.
* And the file which was added can be tracked by the **git**(maintains the changes of file).

**Push :**

* Push the code from local repo to remote repository.(Git hub)

**Git commands:**

|  |  |
| --- | --- |
| Commands | Usage of commands |
| Git init | For making a normal folder in to local repo |
| Git status | Tracking the changes between local repo and working directory. |
| Git add <filename > | Add the file in staging area and make the file ready to push the |
| Git rm - -cached <filename > | Remove the file from staging area and moved to unstaging |
| Git commit -m “commit message” | To save the file inside local repo |
| Git log | To check the logs(history of commit and who did the commit) |
| Git restore < staged filename> | To discard the changes added in file in working directory can be deleted and restore what exactly present in git for the file |
| Git checkout <commit id> | This is for moving to the particular commit. |
| Git add . | To add all the changes files in working directory to the staging area. instead of adding one by one. |
| Git add .\*txt | Adding all the .txt files in the staging area |
| Git restore - -staged . \*txt | Remove all the .txt files from staging area.which files are not commited. |
| Git rm –cached -r \*.txt | Which remove all the file cached by the staging area even though the .txt file has been commited already .so that files are un tracked |
| Git branch | To list all the branches in our local repo |
| Git branch <branch name> | Add branch in our local repo |
| Git checkout <branch name> | Enter into particular branch |
| Git push <remote repo > <branchname> | This is for pushing our code from local repo master branch to the particular branch in remote repo |
| Git pull <remote repo > <branchname> | This is for pull the code from particular branch in remote repo to the local repoand merge with our working directory |
| git remote add <alias name for remote repo> <remote repo link> | This is for adding our remote repo to push or pull purpose |
| Git clone | This is for taking the code clone from remote repo to local repo |
| Git merge <branch name> | Merge the one branch in to another |
| Git checkout . | Moved to the previous commit |
| Git log –oneline | Showing the logs in one line |
| Git remote | Shows the name of remote repo alias |
| Git remote rm <upstream name> | Remove the remote repo |
| Git fetch upstream | Just fetch from local repo and put it in local repo |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |