Github is a code hosting platform for version control and collaboration. It lets us to work together on projects from anywhere. Github Flow is a lightweight, branch-based workflow that supports teams and projects where deployment is made regularly.

Git Commands

* Cloning Remote repository-  
  command- git clone <remote\_repository\_url>

To know which remote repository we cloned- git remote  
  
To create repository in remote  
git remote add <name\_of\_repository><URL\_ofRemote>

* Fetching and Pulling  
  git fetch <remote>- this command will fetch the remote file into local without merging the files

git pull <remote>- this command will fetch file into local by merging the files

* Adding a repository into git  
  git remote add name<url of repository>- adding the repository in git to perform operations on remote repository.  
    
  git add . – add all the files created in the local to the remote repository (for singular files git add <file name>)  
    
  git commit -m “add comment”– commits the files to be pushed in the remote repository
* Pushing to remote  
  git push <remote><branch>- eg- git push -u origin –all 🡪 -u for updating remote repository.  
   🡪 origin means local by default.  
   🡪 --all means to push all the updates  
    
  git remote- gives the keyword to which the remote repository is mapped  
    
  git remote rename <from><to>- To rename the repository  
    
  git remote remove<name>- To rename the repository
* Tagging  
  Git has the ability to tag specific points in history as being important  
  git tag- to know the list of tags available  
    
  git tag -l “version”- to view tags matching a certain pattern
* Creating Tags  
  Types of tags-  
  1. A lightweight tag is very much like a branch that doesn’t change- it’s just a pointer to a specific commit  
    
  2. Annotate tags, however, are stored as full objects in the git database. They’re checksummed; contain the tagger name, email and date; have a tagging message and can be signed and verified with GNU Privacy Guard(GPG)
* Creating Annotated Tags  
  git tag -a v1.1 -m “comment”  
    
  to show the tag- git show v1.1
* Creating lightweight Tags  
  This is used to store commit checksum in the file no other information is kept. To create do not supply any of the -a, -s or -m option. Just proving tag name.  
    
  git tag v1.3
* Tagging after some time  
  We can also tag even after some time we have committed. Following are the steps  
  git log -pretty =oneline  
  know the commit which you want to tag  
  Add below line to tag  
  git tag -a v1.5 <commitstartingpart>
* Pushing tags to remote repository  
  to push specific tag  
  git push origin <tagname>--eg- git push origin v1.5  
    
  to push all the tags  
  git push origin –tags
* Fork Repository

To access the repository of some other person in github  
search for the repository 🡪 open the repository 🡪 click on the fork button 🡪 repository will be copied into our github account.