Git useful commands:

**git diff** : show the differences between whats new changed

but has not been committed

**git log**  : show all commits

**git log --oneline**  : show all commits in one line

**git show <commit-id>**  : show one perticular commit details

**git blame <file-name>** : show who changed what in this file in detail

**git revert <commit-id>**  : remove all codes/files which has been added to

this commit and creates a new commit

**git reset --hard <commit-id>** : reset to a specific commit (deletes all in between

commits)

Concepts:

**Staging area:** Codes/files which are added and ready to commit are in the staging area. If no codes/files are added, then the staging area will be empty.

**Head:** Points to the latest commit (like a linked list). Changing the head (make any previous code to head) can revert back to the previous code, but loses all changes in between commits.

Github useful commands:

**git remote -v**  : check the remote server/origin that has been added

**git remote add origin** [**https://github.com/mahdi-islam-pranto/git-practice.git**](https://github.com/mahdi-islam-pranto/git-practice.git): add a new

remote server

**git push origin main** : push local code to the newly created remote server

**git push -f**  : forcefully push the local to the remote server

Branching

git branch : check all local branches

git branch <branch-name> : create new local branch

git checkout <branch-name> : switch to a specific branch

git push --set-upstream origin john-doe : push the code from a local branch to a

remote branch (not main)

git checkout main : switch to main branch

git merge origin/<branch-name> : merge the specified branch code to the main branch

git push : put the latest main code to the remote server

git pull : get the code to the local branch from the remote branch

**Git Branching shorter ways:**   
-git checkout -b “**feature-login**” : goes to feature-login branch

: Make changes

-git status

-git commit -m “make change in login”

-git push origin **feature-login:feature-login :**creates feature-login branch in github and push the changes to feature-login branch

-git checkout main : goes to main branch

-git pull origin main : pull previous main code

- git merge **feature-login** : merged main with **feature-login** (**feature-login** and main code is same now)

If git conflicts happens, simply accept incoming changes

If I want to go back to previous commits,   
-git log – oneline

-git checkout gyad657

-git pull origin ***branch-name*** :Get the specific Branch code to another Branch