**GIT Commands**

**SETUP**

Configuring user information used across all local repositories

$ git config --global user.name “firstname lastname”

set a name that is identifiable for credit when review version history

$ git config --global user.email “[valid-email]”

set an email address that will be associated with each history marker

**—----------------------------------------------------------------------------------------------------------------------------**

**Useful Commands**

$ git init

initialize an existing directory as a Git repository

$ git status

show modified files in working directory, staged for your next commit

$ git add fileName

add a file as it looks now to your next commit (stage)

$ git commit -m “descriptive message”

commit your staged content as a new commit snapshot

$ git branch -m master main

Renaming local master to main

$ git reset

Removing file from staged area

$ git log

History of all commits

$ git stash

Placing the changes in staged area to some other place(not committed ones)

$ git stash pop

Bring back the files from stash area to staging area

$ git stash clear

Remove files from stash area (permanently deleted)

—--------------------------------------------------------------------

→fork copying the project into your github account.

→from where you have forked this url is known as upstream url by convention

$ git remote add upstream pasteUrl

So that you can now fetch the changes

$git remote add origin pasteUrl

$ git remote -v

Shows URLs of remote repositories

—---------------------------------------------------------------------

→Always create a **new branch** while resolving a bug or working on a new feature.

→All the new commits will be added to the head.

$ git branch branchName

Create a new branch

$ git checkout branchName

Assigning head to this branchName

$ git show **HEAD**

Show head of the branch

**→one branch open only one pull request (recommended) –for adding new feature or fixing bugs**

**→never commit on main branch**

$ git push origin branchName -f

If online commits does not matches your commit(force push)

$ git pull (switch to main branch)

To make local repo in sync with remote repo

→make sure your main branch is up to date before creating new branch

**Issues**

[git@github.com](mailto:git@github.com) permission denied(publickey).

Support for password authentication was removed.please use a personal access token instead

## Create Personal Access Token on GitHub

From your GitHub account, go to **Settings** => **Developer Settings** => **Personal Access Token** => **Generate New Token** (Give your password) => **Fillup the form** => click **Generate token** => **Copy the generated Token**