**Github Commands**

🡪Github is a widely used Free-to-use cloud Storage platform with version control and many other essential features that specifically helps to developers to manage and deploy their projects into Github.

**Benefits of using Github:**

* History tracking
* Collaboration
* Branching and Merging
* Offline work

**Git installation commands:**

🡪For windows we can download the git from the chrome.

🡪**git –version** -- Shows the current version of Git.

**Git Configuration and Setup:**

**git config –global user.name ‘Your name’**

🡪Set username globally.

**git config –global user.email ‘email’**

🡪Set email globally.

**git help**

🡪Display the main help documentation, showing a list of commonly used Git commands.

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

**Initializing a repository**

**git init**

🡪Initializes a new git repository in the current directory.

**git init <directory>**

🡪Creates a new git repository in the specified directory.

**git clone <repository\_url>**

🡪This clones a repository from a remote server to your local machine.

**git clone –branch <branch name> <repository url>**

🡪It clones a specific branch from a repository.

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

**Basic Git Commands.**

**git add <file>**

🡪Add a specific file to the staging area.

**git add . or git add –all**

🡪Adds the all modified and new files to the staging area.

**git status**

🡪Shows the current state of the repository, including tracked and untracked files, modified files and branch information.

**git status –ignored**

🡪It displays the ignored files.

**git remote add origin ‘url’**

🡪It is used to add the origin.

**git push -u origin main**

🡪To push the files into the github repository.

**git diff**

🡪It shows the changes between the working directory and staging area.

**git diff <commit 1> <commit 2>**

🡪It displays the difference between the two commits.

**git diff HEAD**

🡪It displays the difference between the current directory and the last commit.

**git commit**

🡪It creates a new commit with the changes in the staging area and opens the default text editor for adding a commit message.

**git commit -m ‘Message’**

🡪It creates a new commit with the changes in the staging area and specifies the commit message inline.

**git commit -a or git commit –all**

🡪It commits all modified and deleted files in the repository without explicitly using git add to stage the changes.

**git restore <file>**

🡪It restores the file in the working directory to its state in the last commit.

**git reset <commit>**

🡪Moves the branch pointer to a specific commit, resetting the staging area and the working directory to match the specified commit.

**git rm <file>**

🡪It removes a file from the both the working directory and the repository ,staging the deletion.

**git mv**

🡪Moves or rename a file or directory in your git repository.

**Branching and Merging**

**git branch**

🡪It lists all the branches in the repository

**git branch <branch-name>**

🡪It creates a new branch with the specified name.

**git branch -d <branch-name>**

🡪Deletes the specified branch.

**git checkout <branch-name>**

🡪Switches to the specified branch.

**git checkout -b <new-branch-name>**

🡪Creates a new branch and switches to it.

**Git checkout --- <file>**

🡪Discards changes made to the specified file and revert it to the version in the last commit.

**git merge <branch>**

🡪It merges the specified branch into the current branch.

**git log**

🡪Displays the commit history of the current branch.

**git stash**

🡪 Stashes the changes in the working directory, allowing you to switch to a different branch or commit without committing the changes.

**git stash list**

🡪Lists all stashes in the repository.

**git stash pop**

🡪Applies and removes the most recent stash form the stash list.

**git stash drop**

🡪Removes the most recent stash from the stash list.

**git tag**

🡪Lists all tags in the repository.

**git fetch**

🡪 Retrieves change from a remote repository, including new branches and commit.

**git fetch <remote>**

🡪 Retrieves change from the specified remote repository.

**git pull**

🡪 Fetches changes from the remote repository and merges them into the current branch.

**git pull –rebase**

🡪 Fetches changes from the remote repository and rebases the current branch onto the updated branch.

**git push**

🡪 Pushes local commits to the remote repository.

**git push <remote>**

**🡪** Pushes local commits to the specified remote repository.

**git push –all**

🡪 Pushes all branches to the remote repository.

**git remote**

🡪 Lists all remote repositories.

**git show**

🡪 Shows the details of a specific commit, including its changes.

**git show <commit>**

🡪 Shows the details of the specified commit, including its changes.

**git revert <commit>**

🡪Creates a new commit that undoes the changes introduced by the specified commit.

**git revert –no-commit <commit>**

🡪 Undoes the changes introduced by the specified commit, but does not create a new commit.

**git rebase <branch>**

**🡪** Reapplies commits on the current branch onto the tip of the specified branch.