**What Is Git?**

* Git is a version control system for tracking changes in computer files. It helps in coordinating work amongst several people in a project and tracks progress over time. Unlike the centralized version control system(which will have only main branch), Git branches can be easily merged. A new branch is created every time a developer wants to start working on something. This ensures that the master branch always has a production-quality code.
* Git is a distributed version control system, so here, every developer gets their local repository with full commit history. The commit history makes Git fast, as now a network connection is not needed to create commits or perform diffs between commits.

**What Is GitHub?**

* GitHub is a Git repository hosting service that provides a web-based graphical interface (GUI). It helps every team member work together on a project from anywhere, making it easy to collaborate.
* GitHub is one place where project managers and developers coordinate, track, and update their work, so projects stay transparent and on schedule. The packages can be published privately, within the team, or publicly for the open-source community. Downloading packages from GitHub enables them to be used and reused. GitHub helps all team members stay on the same page and stay organized. Moderation tools, like issue and pull request locking, helps the team focus on the code.

**Note : Bitbucket** is similar to **GitHub**

**Git Installation:**

* Download the **[latest version of Git](https://git-scm.com/downloads)** and choose the 64/32 bit version. After the file is downloaded, install it in the system. Once installed, select Launch the Git Bash, then click on finish. The Git Bash is now launched

1 . Repo access

2 . clone the repo (default master)

**Commands:**

**1.git clone:**

* The git clone command is used to create a local working copy of an existing remote repository.
* The command downloads the remote repository to the computer. It is equivalent to the Git init command when working with a remote repository.
* Command : **git clone remote\_URL**

**2. git status :**

* The git status command tells the current state of the repository.
* The command provides the current working branch. If the files are in the staging area, but not committed, it will be shown by the git status. Also, if there are no changes, it will show the message no changes to commit, working directory clean.
* Command : **git status**

**3. git checkout:**

* The git checkout command is used to switch branches, whenever the work is to be started on a different branch.
* # Checkout an existing branch

Command : **git checkout <branch\_name>**

* # Checkout and create a new branch with that name

Command : **git checkout -b <new\_branch>**

4. git diff (exit Q)

**5. git pull:**

* The git pull command is used to fetch and merge changes from the remote repository to the local repository.
* The command "git pull origin master" copies all the files from the master branch of the remote repository to the local repository.
* Command : **git pull origin branch\_name**

**6.git merge:**

* The git merge command is used to integrate the branches together. The command combines the changes from one branch to another branch.
* It is used to merge the changes in the staging branch to the stable branch.
* Command : **git merge branch\_name**

**7. git add:**

* Add command is used after checking the status of the files, to add those files to the staging area.
* Before running the commit command, "git add" is used to add any new or modified files.
* #To add specific file

Command : **git add path\_of\_file**

* #To add all files

Command : **git add .**

**8. git commit:**

* The commit command makes sure that the changes are saved to the local repository.
* The command "git commit –m <message>" allows you to describe everyone and help them understand what has happened.
* Command : **git commit -m “commit message”**

**9. git push:**

* The command git push is used to transfer the commits or pushing the content from the local repository to the remote repository.
* The command is used after a local repository has been modified, and the modifications are to be shared with the remote team members.
* Command : **git push origin branch\_name**

**10.git stash:**

* The git stash command takes your modified tracked files and saves it on a pile of incomplete changes that you can reapply at any time. To go back to work, you can use the stash pop.
* The git stash command will help a developer switch branches to work on something else without committing to incomplete work.
* # Store current work with untracked files

Command : **git stash**

* # Bring stashed work back to the working directory

Command : **git stash pop**