**INSTALLATION OF GIT:**

**Step 1:** Type “Git” in google and open the first link that appears on the screen i.e, open the website <https://git-scm.com/>

**Step 2:** Click on “Downloads” and click on the installer suitable for you depending on your operating system.For example, Windows users click on Windows button.The exe file starts downloading.

**Step 3:** Now double click on the exe file. Accept the license agreement and click on “Next”.

**Step 4:** Browse and select the destination folder for git.

**Step 5:** Then, click on “Finish” to finish installation of git.

**COMMANDS OF GIT:**

**git init**

This command turns a directory into an empty Git repository. This is the first step in creating a repository. After running git init, adding and committing files/directories is possible.

Usage: $ git init

### git add

Adds files in the to the staging area for Git. Before a file is available to commit to a repository, the file needs to be added to the Git index (staging area).

Usage: $ git add <file name or directory name>

### git commit

Record the changes made to the files to a local repository.We can also add a message for each commit.

Usage: $ git commit -m “comment for the commit”

### git status

This command returns the current state of the repository.It shows the files that are modified and the files that are staged.

Usage: $ git status

### git config

This is to assign these settings. Two important settings are user user.name and user.email. These values set what email address and name commits will be from on a local computer.

Usage: $ git config <setting> <command>

### git branch

To determine what branch the local repository is on, add a new branch, or delete a branch.

Usage: $ git branch <branch-name> : to create a branch.

$ git branch -a : list all the branches.

$ git branch -d <branch-name> : to delete a branch.

### git checkout

To start working in a different branch, use git checkout to switch branches.

Usage: $ git checkout <branch-name> : switch to an existing branch.

$ git checkout -b <new-branch>: create a branch and switch to it.

### git merge

Integrate branches together. git merge combines the changes from one branch to another branch.

Usage: $ git merge <branch-name> : merges the branch into current branch.

### git remote

To connect a local repository with a remote repository.

Usage: $ git remote -v : lists all the remote repositories.

### git clone

To create a local working copy of an existing remote repository, use git clone to copy and download the repository to a computer.

Usage: $ git clone <remote url>

### git pull

To get the latest version of a repository run git pull. This pulls the changes from the remote repository to the local computer.

Usage: $ git pull <branch-name> <remote url>

### git push

Sends local commits to the remote repository. git push requires two parameters: the remote repository and the branch that the push is for.

Usage: $ git push <remote url/remote name> <branch>

### git stash

To save changes made when they’re not in a state to commit them to a repository. This will store the work and give a clean working directory. For instance, when working on a new feature that’s not complete, but an urgent bug needs attention.

Usage: $ git stash -u : store current work with untracked files.

$ git stash pop : bring stashed work back to the working directory.

### git log

To show the chronological commit history for a repository. This helps give context and history for a repository.

Usage: $ git log

### git rm

Remove files or directories from the working index (staging area). With git rm, there are two options to keep in mind: force and cached. Running the command with force deletes the file. The cached command removes the file from the working index. When removing an entire directory, a recursive command is necessary.

Usage: $ git rm -<option> <filename>

**STARTING WITH GITHUB:**

Step 1: Open the website <https://github.com/>

Step 2: Sign up in github by giving personal details.

Step 3: Verify your email address by following the instructions in the mail sent by github.

**PROCESS FOR INITIALIZING GIT:**

1. Configuration of username and email address.

Commands: $ git config --global user.name "<user name>"

$ git config --global user.email <email address>

1. Initializing git repository.

Commands: $ git init

**CREATING A NEW REPOSITORY IN GIT:**

1. Create a new directory
2. Type “git init” in git bash
3. Add some files to that folder.
4. Type “git add “ to add files to staging area.
5. Type “git commit” to commit the files to github.

**BRANCHING IN GIT:**

* Branching is a feature through which two paths of development can happen at a time.
* It can happen through a command “git branch”
* It allows us to list the branches,create and delete the branches.

Command: $ git branch --<option>

**CHECKOUT:**

* Checkout command is used to switch between branches.
* It can also be used to create a branch and switch to that branch.

Command: $ git checkout -<option> <branch>