**Git commands:**

1. git clone <https://name-of-the-repository-link> =

Git clone is a command for downloading existing source code from a remote repository (like Git hub, for example). In other words, Git clone basically makes an identical copy of the latest version of a project in a repository and saves it to your computer

1. git branch <branch-name> =

create branch

1. git push -u <remote> <branch-name>

This command will create a branch **locally**. To push the new branch into the remote repository, you need to use this.

1. git branch or git branch –list

listout branches

1. git branch -d <branch-name>

used to delete branch one more option is -D

1. git commit -m "commit message" =

commit changes with message

1. git add =

Moves changes from the working directory to the staging area. This gives you the opportunity to prepare a snapshot before committing it to the official history

1. git rebase=

Rebasing lets you move branches around, which helps you avoid unnecessary merge commits. The resulting linear history is often much easier to understand and explore.

1. git revert =

Undoes a committed snapshot. When you discover a faulty commit, reverting is a safe and easy way to completely remove it from the code base

1. git status =

Displays the state of the working directory and the staged snapshot. You’ll want to run this in conjunction with git add and git commit to see exactly what’s being included in the next snapshot.

1. HEAD=

Git’s way of referring to the current snapshot. Internally, the git checkout command simply updates the HEAD to point to either the specified branch or commit. When it points to a branch, Git doesn't complain, but when you check out a commit, it switches into a “detached HEAD” state

1. Git stash=

 temporarily shelves (or *stashes*) changes you've made to your working copy so you can work on something else, and then come back and re-apply them later on. Stashing is handy if you need to quickly switch context and work on something else, but you're mid-way through a code change and aren't quite ready to commit.

1. Git config

A convenient way to set configuration options for your Git installation. You’ll typically only need to use this immediately after installing Git on a new development machine.

1. Git fetch=

Fetching downloads a branch from another repository, along with all of its associated commits and files. But, it doesn't try to integrate anything into your local repository. This gives you a chance to inspect changes before merging them with your project.

1. Git init =

Initializes a new Git repository. If you want to place a project under revision control, this is the first command you need to learn.