**Git:**

Git is a popular version control system. It is used for:

* Tracking code changes
* Tracking who made changes
* Coding collaboration

Git can

* Manage projects with Repositories
* Clone a project to work on a local copy
* Control and track changes with Staging and Committing
* Branch and Merge to allow for work on different parts and versions of a project
* Pull the latest version of the project to a local copy
* Push local updates to the main project

Some of the prominent reasons for using Git are:

* Developers can work together from anywhere.
* Developers can see the full history and can compare the previous and new changes of the project.
* Developers can retreat to earlier versions of a project.

**Repositories:**

Repositories in [GIT](https://www.geeksforgeeks.org/git-lets-get-into-it/) contain a collection of files of various different versions of a Project. These files are imported from the repository into the local server of the user for further updations and modifications in the content of the file. A VCS or the [Version Control System](https://www.geeksforgeeks.org/version-control-systems/) is used to create these versions and store them in a specific place termed a repository

**Cloning:**

The process of copying the content from an existing Git Repository with the help of various Git Tools is termed **cloning**. Once the cloning process is done, the user gets the complete repository on his local machine. Git by default assumes the work to be done on the repository is as a user, once the cloning is done.

**Git Hub:**

GitHub is a hosting service for Git repositories and if you have a project hosted on GitHub, you can access and download that project with commands on any computer you have access and make your changes and push the latest version back to GitHub. GitHub allows you to store your repo on their platform. It is also comes with GitHub, ability to collaborate with other developers from any location.

**Git Commands:**

Some of the basic commands,

**1) Git config command**

This command configures the user. The Git config command is the first and necessary command used on the Git command line. This command sets the author name and email address to be used with your commits. Git config is also used in other scenarios.

**Syntax**

1. $ git config --global user.name "ImDwivedi1"
2. $ git config --global user.email "Himanshudubey481@gmail.com"

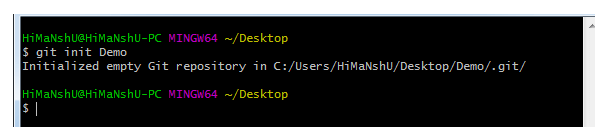
**2) Git Init command**

This command is used to create a local repository.

**Syntax**

1. $ git init Demo

The init command will initialize an empty repository. See the below screenshot.

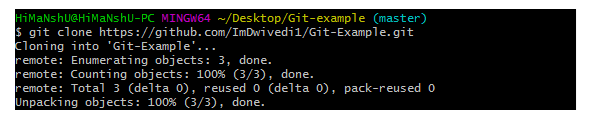


**3) Git clone command**

This command is used to make a copy of a repository from an existing URL. If I want a local copy of my repository from GitHub, this command allows creating a local copy of that repository on your local directory from the repository URL.

**Syntax**

1. $ git clone URL



**4) Git add command**

This command is used to add one or more files to staging (Index) area.

**Syntax**

To add one file

1. $ git add Filename

To add more than one file

1. $ git add\*

Git Commands

**5) Git commit command**

Commit command is used in two scenarios. They are as follows.

**Git commit -m**

This command changes the head. It records or snapshots the file permanently in the version history with a message.

**Syntax**

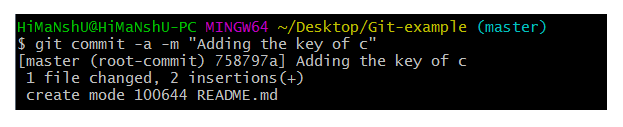
1. $ git commit -m " Commit Message"

**Git commit -a**

This command commits any files added in the repository with git add and also commits any files you've changed since then.

**Syntax**

1. $ git commit -a

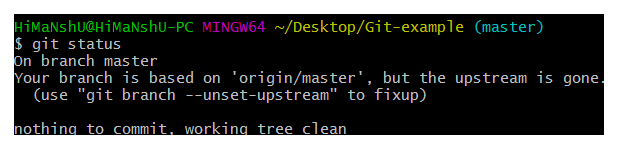


**6) Git status command**

The status command is used to display the state of the working directory and the staging area. It allows you to see which changes have been staged, which haven't, and which files aren?t being tracked by Git. It does not show you any information about the committed project history. For this, you need to use the git log. It also lists the files that you've changed and those you still need to add or commit.

**Syntax**

1. $ git status



**7) Git push Command**

It is used to upload local repository content to a remote repository. Pushing is an act of transfer commits from your local repository to a remote repo. It's the complement to git fetch, but whereas fetching imports commits to local branches on comparatively pushing exports commits to remote branches. Remote branches are configured by using the git remote command. Pushing is capable of overwriting changes, and caution should be taken when pushing.

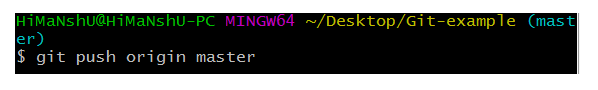
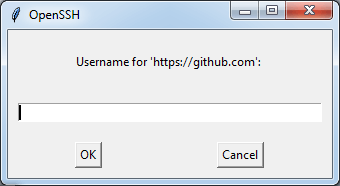
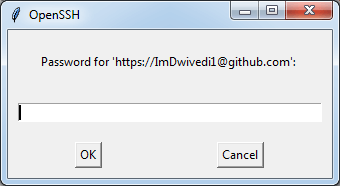
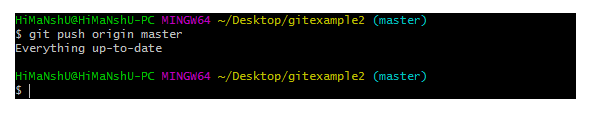
Git push command can be used as follows.

**Git push origin master**

This command sends the changes made on the master branch, to your remote repository.

**Syntax**

1. $ git push [variable name] master

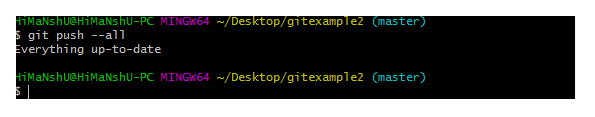
  
  
  


**Git push -all**

This command pushes all the branches to the server repository.

**Syntax**

1. $ git push --all

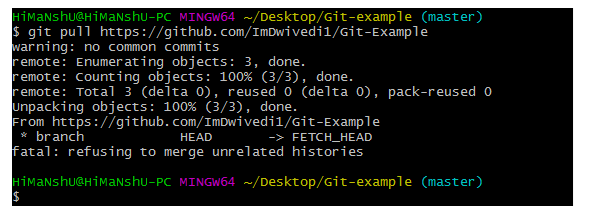


**8) Git pull command**

Pull command is used to receive data from GitHub. It fetches and merges changes on the remote server to your working directory.

**Syntax**

1. $ git pull URL



**Reference:**

* Tamil video : <https://www.youtube.com/watch?v=VIBWdLLq9kQ>
* To create repository and open GitHub desktop: <https://www.youtube.com/watch?v=iv8rSLsi1xo>
* Crash course for beginners: <https://www.youtube.com/watch?v=RGOj5yH7evk>

**Tutorial Documentations:**

* <https://www.youtube.com/watch?v=RGOj5yH7evk>
* <https://www.atlassian.com/git/tutorials/learn-git-with-bitbucket-cloud>
* <https://www.javatpoint.com/git>
* <https://www.geeksforgeeks.org/what-is-a-git-repository/>
* <https://www.simplilearn.com/tutorials/git-tutorial/git-tutorial-for-beginner>

**Downloads:**

* GIT: <https://git-scm.com/download>
* GIT HUB Desktop : <https://desktop.github.com/>

**Website:**

* GIT HUB: <https://github.com>