Git Repository

In Git, the repository is like a data structure used by VCS to store metadata for a set of files and directories. It contains the collection of the files as well as the history of changes made to those files. Repository in Git is considered as your project folder. A repository has all the project-related data. Distinct projects have distinct repositories.

Getting a Git Repository

There are two ways to obtain a repository. They are as follows:

* Create a local repository and make it as Git repository.
* Clone a remote repository (already exists on a server).

In either case, you can start working on a Git repository.

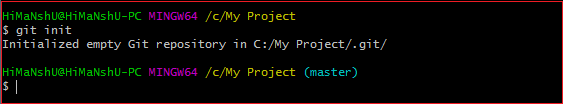
Initializing a Repository

If you want to share your project on a version control system and control it with Git. Then, browse your project's directory and start the git command line (Git Bash for Windows) here. To initialize a new repository, run the below command:

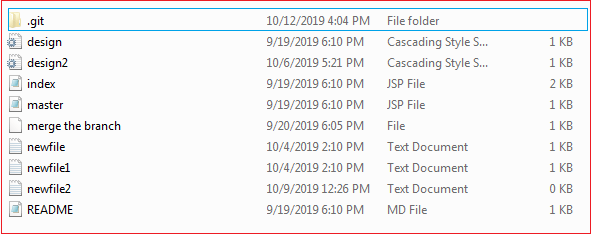
**Syntax:**

1. $ git init

**Output:**



The above command will create a new subdirectory named .git that holds all necessary repository files. The **.git** subdirectory can be understood as a Git repository skeleton. Consider the below image:

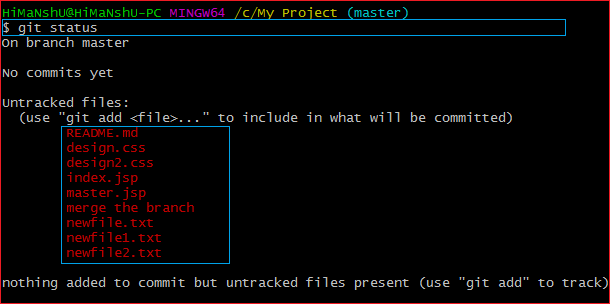


An empty repository .git is added to my existing project. If we want to start version-controlling for existing files, we should track these files with git add command, followed by a commit.

We can list all the untracked files by git status command.

1. $ git status

Consider the below output:



In the above output, the list of all untracked files is displayed by the git status command. To share these files on the version control system, we have to track it with git add command followed by a commit. To track the files, operate git add command as follows:

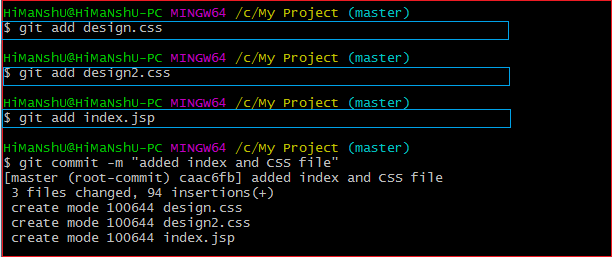
**Syntax:**

1. $ git add **<filename>**

To commit a file, perform the git commit command as follows:

1. $ git commit -m "Commit message."

**Output:**



In the above output, I have added three of my existing files by git add command and commit it for sharing.

We can also create new files. To share the new file, follow the same procedure as described above; add and commit it for sharing. Now, you have a repository to share.

Cloning an Existing Repository

We can clone an existing repository. Suppose we have a repository on a version control system like subversion, GitHub, or any other remote server, and we want to share it with someone to contribute. The git clone command will make a copy for any user to contribute.

We can get nearly all data from server with git clone command. It can be done as:

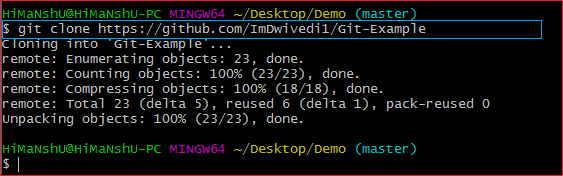
**Syntax:**

1. $ git clone **<Repository** URL**>**

Suppose one of my friends has a repository on my GitHub account, and I want to contribute to it. So the first thing I will do, make a copy of this project to my local system for a better work interface. The essential element needed for cloning the repository URL. I have a repository URL "<https://github.com/ImDwivedi1/Git-Example>". To clone this repository, operate the clone command as:

1. $ git clone https://github.com/ImDwivedi1/Git-Example

Consider the below output:



In the above output, the repository Git-Example has been cloned. Now this repository is available on your local storage. You can commit it and contribute to the project by pushing it on a remote server.

A single repository can be cloned any number of times. So we can clone a repository on various locations and various systems.