



# GIT

## Commands

# 1. git init

This command let us **create a new repository**.  
A hidden **.git** directory is added to the folder.

Most of the git command do not work outside initialized project , so this is the first command you will run in a project

Go to project folder > run git init



```
$ git init
```

## 2. git clone

This command **creates a local copy** of a **remote repository** .

When you clone a repo the source code gets automatically downloaded to local machine. This local repo will point to remote repo and can **PUSH** and **PULL** changes to it



```
$ git clone < git-repo-url>
```

### 3. git add

This command **add your changes** to **staging area** where you can compare you local version with remote repo code.

It is **mandatory to stage the code** before commit(push to remote) using git add command.

To stage all files use (.) - **git add .** in the same repo



```
$ git add <file-1> <file-2>
```

## 4. git commit

This command **saves your changes** to your local repository.

Everytime you commit you have to add a **small message** about the changes you made. This will help to keep track of the changes later.



```
$ git commit -m "commit-message"
```



## 5. git push

This command **push your changes** from local repository to your remote repository. One can only push the committed changes.

It also **creates the repository** with the branch name you enter if repository does not exist on remote location.

If branch is already connected to remote then run - **git push**



```
$ git push <remote> <branch-name>
```

## 6. git pull

This command **fetches latest changes** from remote repository to your local. This is helpful when multiple people are working on same repository. It will help to keep your local repo **updated with latest code.**

If branch is already connected to remote then run - **git pull**



```
$ git pull <remote> <branch-name>
```

## 7. git checkout

This command helps to **switch to an existing branch or create a new branch**.

Before checking out make sure the **branch exist in your local machine** and the changes in the current branch is already **staged or committed**.



```
$ git checkout -b <branch>
```



# Reference Document

There are many other command as well. One needs to have knowledge of Git irrespective of domain.

Below mentioned are some documents which can be helpfull.

**Blog** - <https://www.hostinger.com/tutorials/basic-git-commands>

**Youtube** - <https://www.youtube.com/watch?v=RGOj5yH7evk>

**Youtube (Hindi)** - <https://youtu.be/Ez8F0nW6S-w?feature=shared>