GIT COMMANDS

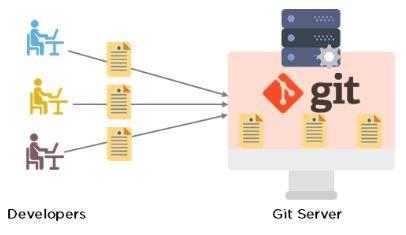
**What is Git?**

Git is a distributed version control system used for tracking changes in computer files. It is generally used for source code management in software development.

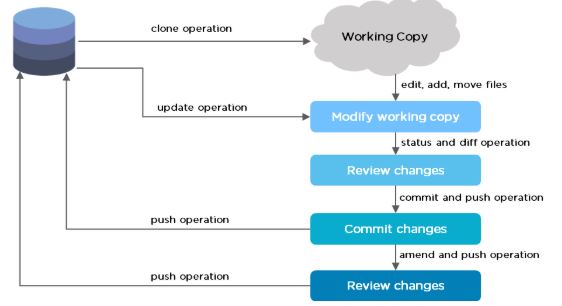
* Git is used to tracking changes in the source code
* It allows multiple developers to work together
* It supports non-linear development through its thousands of parallel branches

**Features of Git?**

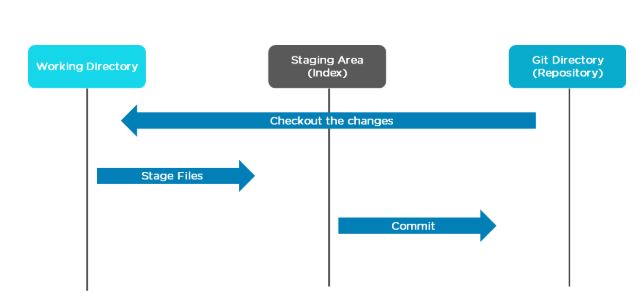
* Tracks history
* Free and open source
* Supports non-linear development
* Creates backups
* Scalable
* Supports collaboration
* Branching is easier
* Distributed development



Git Workflow

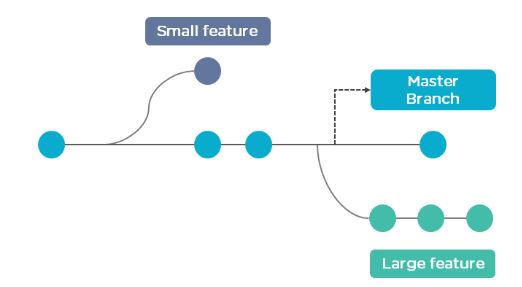


The Git workflow is divided into three states:

* Working directory - Modify files in your working directory
* Staging area (Index) - Stage the files and add snapshots of them to your staging area
* Git directory (Repository) - Perform a commit that stores the snapshots permanently to your Git directory. Checkout any existing version, make changes, stage them and commit.

**Branch in Git**

* Branch in Git is used to keep your changes until they are ready. You can do your work on a branch while the main branch (master) remains stable. After you are done with your work, you can merge it with the main office.



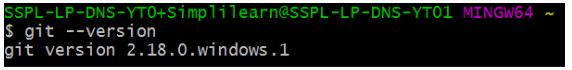
* The above diagram shows there is a master branch. There are two separate branches called “small feature” and “large feature.” Once you are finished working with the two separate branches, you can merge them and create a master branch.

**Commands in Git**

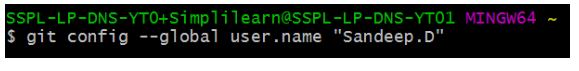
* Create Repositories  
  git init
* Make Changes  
  add  
  commit  
  status
* Parallel Development  
  branch  
  merge  
  rebase
* Sync Repositories  
  push  
  pull  
  add origin

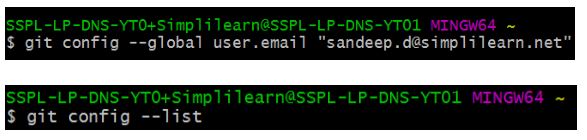
**Command**

* Check the version of Git.

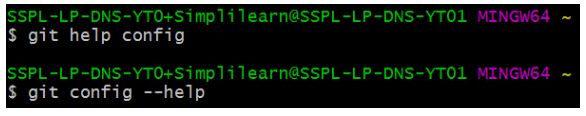


* Set up global config variables - If you are working with other developers, you need to know who is checking the code in and out, and to make the changes.

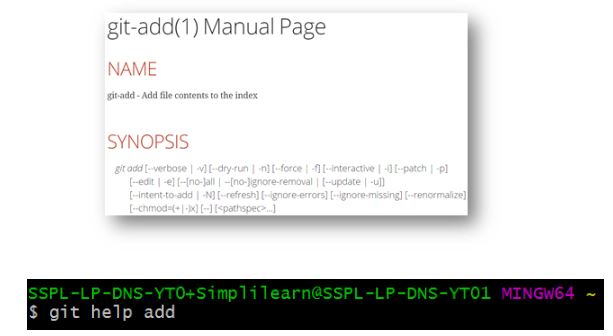




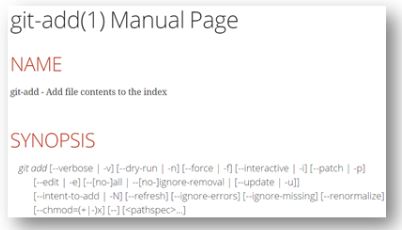
* If in case you need help, use the following [commands](https://www.simplilearn.com/tutorials/git-tutorial/git-commands):



This will lead you to the Git help page on the browser, which will display the following:



This will lead you to the Git help page on the browser, which will display the following:



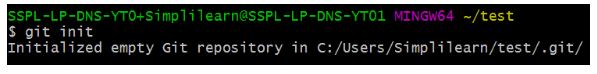
* Create a “test” repository in the local system.



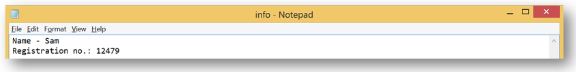
* Move to the test repository.



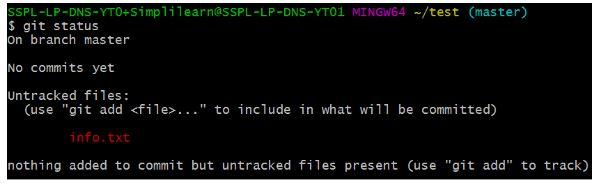
* Create a new git instance for a project.



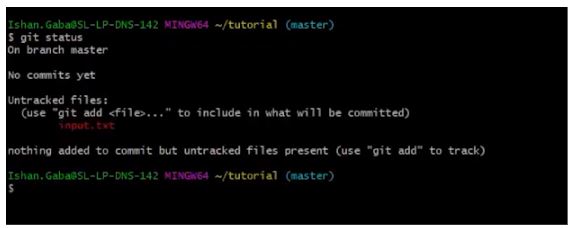
* Create a text file called info.txt in the test folder; write something and save it.



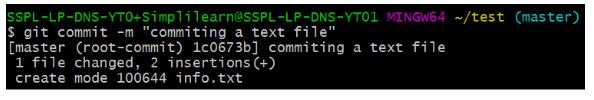
* Check the status of the repository.



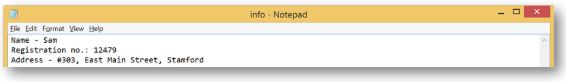
* Add the file you created to make a commit.



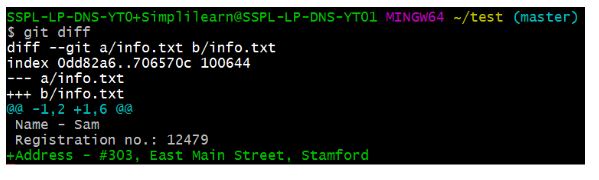
* Commit those changes to the repository’s history with a short message.



* Make any necessary changes to the file and save.



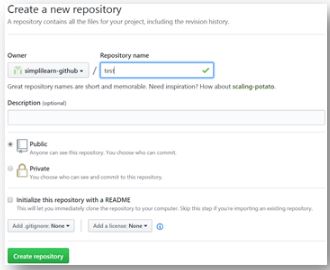
* Now that you’ve made changes to the file, you can compare the differences since your last commit.



* Add [GitHub](https://www.simplilearn.com/tutorials/git-tutorial/what-is-github) username to Git Configuration.

username

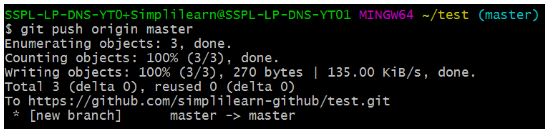
* Create a remote repository.



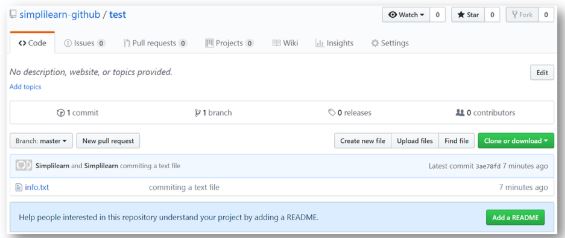
* Connect the local repository to your remote repository.

local-repo

* Push the file to the remote repository.

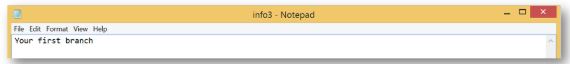


* Refresh your repository page on GitHub. You will get your local file on your remote [GitHub repository](https://www.simplilearn.com/tutorials/git-tutorial/git-vs-github).



* Create three more text files in the local repository - “info1.txt”, “info2.txt”, “info3.txt”.





* Create a branch “first\_branch” and merge it to the main (master) branch.

first-branch.

The above command creates a branch.



The above command switches to the new branch from the master branch.



The above command creates and adds “info3.txt” to the first\_branch.

info3text