GIT

GIT is a free and open source distributed version Control system designed to hande everything from small to very large projects with speed and efficiency.

- GIT is easy to learn
- It has a tiny footprint with lightning fast performance
- It outclasses SCM tools like subversion, CVS, perforce with features like cheap local branching, Convenient staging areas, and multiple workflows.

GIT stands for Global Information Tracker.

1 git init

```
git init [repository name]
```

This Command is used to start a new repository

2 git clone

```
git clone [url]
```

This Command is used to obtain a repository from existing url

3 git add

```
git add [file]
```

This Command adds a file to the staging area.

```
git add *
```

This Command adds one or more to the staging area.

4 git commit

```
git commit -m "[Type in the commit message]"
```

This command records or snapshots the file permanently in the version history.

```
git commit -a
```

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

5 git diff

```
git diff
```

This command shows the file difference which are not yet staged

6 git push

```
git push [variable name] master.
```

This command sends the committed changes of master branch to your remote repository

```
git push [variable name] [branch]
```

This command sends the branch commits to your remote repository

```
git push -all [variable name]
```

This command pushes all branches to your remote repository

```
git push [variable name] :[branch name]
```

This command deletes a branch on your remote repository

7 git pull

```
git pull [Repository Link]
```

This Command fetchs and merges changes on the remote server to your working directory.

8 git log

```
git log.
```

This Command is used to list the version history for the Current branch.

9 git rm

```
git rm [file]
```

This command deletes the files from your working directory and stages the deletion

10 git status

```
git status
```

This Command lists all the files that have to be committe

The 3 states of git (workflow) are:

i) Committed: It takes the file from the staging area and stores it to snapshots permanently in the git.

ii) modified: It takes files from working directory

iii) staged: Adds snapshots to the staging area.

The 3 main parts of git project

i) Git directory: It stores the object database where cloning takes place in the repository to another remote Computer

ii) working tree: Files are pulled out from the database and placed on the disk for modifications

iii) staging area: This acts as an index part that directs to go the next Commit.

Function

One often-overlooked feature of Bourne shell script programming is that you can easily write functions for use within your script. This is generally done in one of two ways: i) with a simple script ii) the function is simply declared in the same file as it is called.

Creating Functions

Syntax

```
function_name () {
        list of commands
     }
```

Pass parameters to a function

The parameters can be represented by $1, $2 and so on.

example: `echo "Hello $1 $2"`

Nested Functions

A function that call itself is called recursive function

example:

```
#!/bin/sh
number_one () {
        echo "This is 1st function"
        number_two
     }
number_two () {
        echo "This is 2nd function"
     }
number_one
```

Returning values from Functions

It is used to just terminate the execution of function and not entire shell program.

syntax

return code

Defination

Functions are a way to group Command for later execution using a single name for the group. when the name of a shell function is used as a simple Command name, the list of Commands associated with that function name is Executed.