Git

Git is an open source distributed version control system. It is design to handle minor and major projects with high speed and efficiency.
It was created by Linus Torvalds in 2005 to develop linux Kernel.

Characteristics

1. Strong support for non-linear development.

- It supports rapid branching and merging, and includes specific Tools for visualizing and navigating a non linear development history.

2. Distributed Development

It provides each developer a local copy of the entire development history and changes are copied from one such repository to another.

3. Efficient handling of large projects

4. Data Assurance

The GIT history is stored in such a way that The ID of a particular version depends upon the complete development history leading up to that commit.

5. Automatic Garbage collection.

It automatically performs garbage collection when enough loose objects have been created in the repository.

Purpose of Git

- Manage projects with Repositories
- Clone a project to work on a local copy
- Control and track changes with staging & 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.

Install git

```
Sudo apt-get install git
```

Check version

```
git --version
```

Configure git

```
git config --global user.name "your name"
git config --global user.email "your email"
```

upload files on Github using command line.

1. Create repository in Github
2. Then in terminal follow these commands :-
3. git init
4. git add filename
5. git commit -m "~~fixs~~commit message"
6. git branch -M main
7. git remote add origin <repository_URL>
8. git push -u origin main

Basic GIT commands

1 Git config

This command configures the user. It is the first & necessary command used on the git command line. It sets the author name and email address to be used with your commits.

Syntax

```
git config --global user.name "Name"
git config --global user.email "emailid"
```

2. Git init command

This command used to create a local repository.

Syntax :- `git init`

3. Git clone command

This command is used to make a copy of a repository from an existing URL.

Syntax

```
git clone URL
```

4. Git add command

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

Syntax :-

```
git add filename
```

5. Git commit command

- Git commit -m :- This command changes the head. It records or snapshots the file permanently in the version history with a message.

Syntax

```
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

```
git commit -a
```

6. Git status :- This command used to display the state of the working directory and the staging area. It allows us to see which changes have been staged, which haven't and which files aren't being tracked by Git

Syntax

```
git status
```

7. git push :- 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.

git push origin master

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

Syntax

```
git push [variablename] master.
```

git push -all

This command pushes all the branches to the server repo.

Syntax

```
$ git push --all
```

8. git pull command

pull command is used to receive data from github. It fetches & merges changes on the remote server to your working directory.

Syntax

```
$ git pull URL
```

9. Git Branch Command

This command lists all the branches available in the repository.

Syntax

```
$ git branch
```

10 Git merge command

This command is used to merge the specified branch's history into the current branch.

```
$ git merge BranchName
```