

Git Commands

1. Git init

To initialise the new working directory

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  GITLENS  JUPYTER  COMMENTS

chandra@LAPTOP-SNAKEM79 MINGW64 ~/Documents/Github Repositories/Testproject
$ git init
Initialized empty Git repository in C:/Users/chandra/Documents/Github Repositories/Testproject
```

2. Git status

Shows the working tree status

```
chandra@LAPTOP-SNAKEM79 MINGW64 ~/Documents/Github Repositories/Testproject (main)
$ git status
On branch main

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
      README.md

nothing added to commit but untracked files present (use "git add" to track)
```

3. Git add .

To stage all files in the entire repository

```
chandra@LAPTOP-SNAKEM79 MINGW64 ~/Documents/Github Repositories/Testproject (main)
$ git add .
```

4. Git commit -m "commit message"

Creates a commit, which is like a snapshot of the repository. These commits are snapshots of the entire repository at specific times.

```
chandra@LAPTOP-SNAKEM79 MINGW64 ~/Documents/Github Repositories/Testproject (main)
$ git commit -m "First commit"
[main (root-commit) 8f5bba8] First commit
 2 files changed, 3 insertions(+)
 create mode 100644 README.md
 create mode 100644 Requirement.txt
```

5. Git push

Used to upload local repository content to a remote repository

```
$ git push -u origin main
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (4/4), 292 bytes | 292.00 KiB/s, done.
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/Ramachandra87/Testproject.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
```

6. Git log

To show the commit logs

```
$ git log
commit 907e3d560e8cd75708f46f76d140c2553d157e24 (HEAD -> main, origin/main, dev2)
Author: Ramachandra87 <chandraananth87@gmail.com>
Date:   Sat Oct 1 20:01:47 2022 +0530

    commit in the dev2 branch

commit 9674c12482ed7ffc729d21e8f531043177ee3fa6 (dev1)
Author: Ramachandra87 <chandraananth87@gmail.com>
Date:   Sat Oct 1 19:45:20 2022 +0530
```

7. Git log -p

To see complete diffs at each step

```
$ git log -p
commit 907e3d560e8cd75708f46f76d140c2553d157e24 (HEAD -> main, origin/main, dev2)
Author: Ramachandra87 <chandraananth87@gmail.com>
Date:   Sat Oct 1 20:01:47 2022 +0530

    commit in the dev2 branch

diff --git a/README.md b/README.md
index f92469d..61964a9 100644
--- a/README.md
+++ b/README.md
@@ -1,2 +1,3 @@
```

8. Git branch

List all of the branches in a repository

```
$ git branch
* main
```

9. Git branch <branch>

To create a new branch

```
$ git branch dev1

chandra@LAPTOP-SNAKEM79 MINGW64 ~/Documents/Github Repositories/Testproject (main)
$ git branch
  dev1
* main
```

10. Git checkout

Switch branches and check content out into working directory

```
$ git checkout dev1
Switched to branch 'dev1'
```

11. Git merge

Used to merge one or more branches into the checked out branch. It will advance the current branch to the result of the merge

```
$ git merge dev1
Updating 8be5805..9674c12
Fast-forward
 README.md | 2 +-
 1 file changed, 1 insertion(+), 1 deletion(-)
```

12. Git delete <branch>

To delete the branch

```
$ git branch -d dev1
Deleted branch dev1 (was 9674c12).
```

13. Git remote -v

List the remote connections to the repositories including the URL of each connection

```
$ git remote -v
origin https://github.com/Ramachandra87/Testproject.git (fetch)
origin https://github.com/Ramachandra87/Testproject.git (push)
```

14. Git remote add <name> <url>

Creates a new connection to a remote repository. The provided name can later be used instead of the URL when referencing the remote. A typical default shortname is "origin": this is used for the remote which the local repository was cloned from.

```
$ git remote add origin https://github.com/Ramachandra87/Testproject.git  
chandra@LAPTOP-SNAKEM79 MINGW64 ~/Documents/Github Repositories/Testproject (main)  
$ git status  
On branch main  
nothing to commit, working tree clean
```

15. Git stash

Takes the uncommitted changes (both staged and unstaged), saves them away for later use, and then reverts them from the working copy

```
$ git stash  
Saved working directory and index state WIP on main: 31ac549 modified
```