## 1. git init

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    GITLENS    JUPYTER    COMMENTS


silri@Shonku MINGW64 ~/Applied_AI/Git_Assignment/Git_1 (master)
$ git init
Initialized empty Git repository in C:/Users/silri/Applied_AI/Git_Assignment/Git_1/.git/

silri@Shonku MINGW64 ~/Applied_AI/Git_Assignment/Git_1 (master)
$ []
```

## 2. git status

```
silri@Shonku MINGW64 ~/Applied_AI/Git_Assignment/Git_1 (master)
$ git status
On branch master

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)

silri@Shonku MINGW64 ~/Applied_AI/Git_Assignment/Git_1 (master)
$ []
```

## 3. git add .

```
silri@Shonku MINGW64 ~/Applied_AI/Git_Assignment/Git_1 (master)
$ git add .

silri@Shonku MINGW64 ~/Applied_AI/Git_Assignment/Git_1 (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   README.md
```

**4.  git commit -m "<commit message>"**

```
silri@Shonku MINGW64 ~/Applied_AI/Git_Assignment/Git_1 (master)
$ git commit -m "This is the first commit"
[master (root-commit) 60e9e2f] This is the first commit
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 README.md

silri@Shonku MINGW64 ~/Applied_AI/Git_Assignment/Git_1 (master)
$ ▯
```

**5.  git log**

```
silri@Shonku MINGW64 ~/Applied_AI/Git_Assignment/Git_1 (master)
$ git log
commit 60e9e2f61ffe5a8c8b556bef1204520e0bbbe135 (HEAD -> master)
Author: Ritesh Sil <sil.ritzzz@gmail.com>
Date:   Tue Oct 11 09:59:53 2022 +0530

    This is the first commit

silri@Shonku MINGW64 ~/Applied_AI/Git_Assignment/Git_1 (master)
$ ▯
```

**6.  git branch <branch name>**

```
silri@Shonku MINGW64 ~/Applied_AI/Git_Assignment/Git_1 (master)
$ git branch
  develop
* master

silri@Shonku MINGW64 ~/Applied_AI/Git_Assignment/Git_1 (master)
$ ▯
```

**7.  git branch –all**

```
silri@Shonku MINGW64 ~/Applied_AI/Git_Assignment/Git_1 (master)
$ git branch --all
  develop
* master

silri@Shonku MINGW64 ~/Applied_AI/Git_Assignment/Git_1 (master)
$ ▯
```

**8. git checkout develop**

```
silri@Shonku MINGW64 ~/Applied_AI/Git_Assignment/Git_1 (master)
$ git checkout develop
Switched to branch 'develop'

silri@Shonku MINGW64 ~/Applied_AI/Git_Assignment/Git_1 (develop)
$ 
```

**9. git merge <branch name>**

```
silri@Shonku MINGW64 ~/Applied_AI/Git_Assignment/Git_1 (master)
$ git merge develop
Updating 60e9e2f..ab1c252
Fast-forward
 README.md | 7 +++++++
 1 file changed, 7 insertions(+)

silri@Shonku MINGW64 ~/Applied_AI/Git_Assignment/Git_1 (master)
$ 
```

**10. git branch -d <branch name>**

```
silri@Shonku MINGW64 ~/Applied_AI/Git_Assignment/Git_1 (master)
$ git branch -d develop
Deleted branch develop (was ab1c252).

silri@Shonku MINGW64 ~/Applied_AI/Git_Assignment/Git_1 (master)
$ git branch --all
* master

silri@Shonku MINGW64 ~/Applied_AI/Git_Assignment/Git_1 (master)
$ 
```

**11. git restore –staged .**

```
silri@Shonku MINGW64 ~/Applied_AI/Git_Assignment/Git_1 (master)
$ git restore --staged .

silri@Shonku MINGW64 ~/Applied_AI/Git_Assignment/Git_1 (master)
$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   README.md

no changes added to commit (use "git add" and/or "git commit -a")

silri@Shonku MINGW64 ~/Applied_AI/Git_Assignment/Git_1 (master)
$ 
```

**12. git stash**

```
silri@Shonku MINGW64 ~/Applied_AI/Git_Assignment/Git_1 (master)
$ git stash
Saved working directory and index state WIP on master: ab1c252 2nd change
```

**13. git stash list**

```
silri@Shonku MINGW64 ~/Applied_AI/Git_Assignment/Git_1 (master)
$ git stash list
stash@{0}: WIP on master: ab1c252 2nd change
stash@{1}: WIP on master: ab1c252 2nd change

silri@Shonku MINGW64 ~/Applied_AI/Git_Assignment/Git_1 (master)
$ 
```

**14. git stash apply**

```
silri@Shonku MINGW64 ~/Applied_AI/Git_Assignment/Git_1 (master)
$ git stash apply
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   README.md

no changes added to commit (use "git add" and/or "git commit -a")
```

## 15. git clone <repo url>

```
silri@Shonku MINGW64 ~/Applied_AI/Git_Assignment (master)
$ git clone https://github.com/Ritesh-Sil/git-demo1.git
Cloning into 'git-demo1'...
remote: Enumerating objects: 13, done.
remote: Counting objects: 100% (13/13), done.
remote: Compressing objects: 100% (10/10), done.
remote: Total 13 (delta 2), reused 12 (delta 1), pack-reused 0
Receiving objects: 100% (13/13), done.
Resolving deltas: 100% (2/2), done.
```

## 16. git push origin main

```
silri@Shonku MINGW64 ~/Applied_AI/Git_Assignment/git-demo1 (main)
$ git push origin main
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 342 bytes | 342.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/Ritesh-Sil/git-demo1.git
   299eea9..a936c5c  main -> main

silri@Shonku MINGW64 ~/Applied_AI/Git_Assignment/git-demo1 (main)
$ 
```

## 17. git pull origin main

```
 ! [rejected]        main -> main (fetch first)
error: failed to push some refs to 'https://github.com/Ritesh-Sil/git-demo1.git'
hint: Updates were rejected because the remote contains work that you do
hint: not have locally. This is usually caused by another repository pushing
hint: to the same ref. You may want to first integrate the remote changes
hint: (e.g., 'git pull ...') before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.

silri@Shonku MINGW64 ~/Applied_AI/Git_Assignment/git-demo1 (main)
$ git pull origin main
From https://github.com/Ritesh-Sil/git-demo1
 * branch            main       -> FETCH_HEAD
Auto-merging requirements.txt
CONFLICT (content): Merge conflict in requirements.txt
Automatic merge failed; fix conflicts and then commit the result.

silri@Shonku MINGW64 ~/Applied_AI/Git_Assignment/git-demo1 (main|MERGING)
$ 
```

README.md ● ≡ requirements.txt ! ✕

≡ requirements.txt

You, 1 second ago | 2 authors (You and others)

```
1  pandas
2  numpy
3  matplotlib
   Accept Current Change | Accept Incoming Change | Accept Both Changes | Compare Changes
4  <<<<<<< HEAD (Current Change)
5  sklearn
6  =======
7  seaborn
8  >>>>>>> 88bcc67b43a9fc0c48360edcc34fc05cab22e1c7 (Incoming Change)
9
```