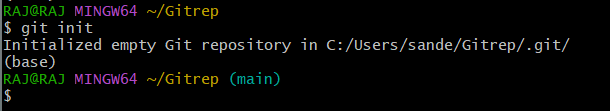
**Git commands**

**1. git init**

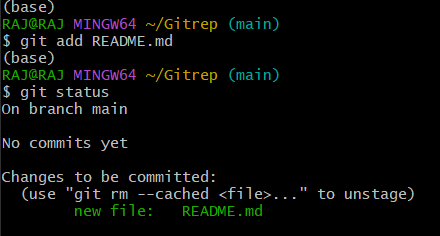
**Usage: git init [repository name]**

We have to navigate to our project directory and type the command git init to initialize a Git repository for our local project folder. Its files are organized in subdirectories through its hidden .git directory. 

**2. git add**

**Usage: git add [file(s) name] (or) git add .**

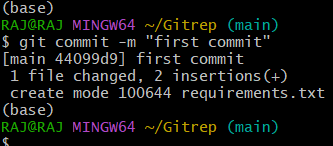
This will add the specified file(s) to the Git repository, the staging area, where they are already being tracked by Git and are ready to be committed.



**3. git commit**

**Usage: git commit -m “message”**

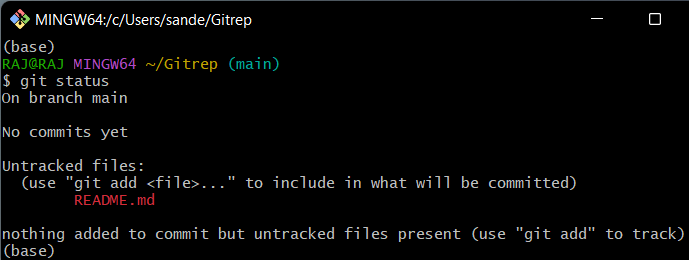
Using this command, all files that are in directory right now are being recorded or snapshotted permanently in the Git file system.



**4. git status**

**Usage: git status**

The command shows the modified status of an existing file and the added status of a new file, if any, that has to be committed.

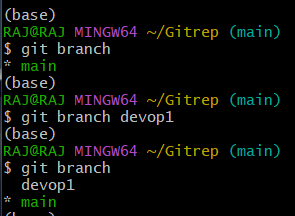
****

**5. git branch**

**Usage: git branch (for checking)**

**git branch [name -of-the-branch] (for creating)**

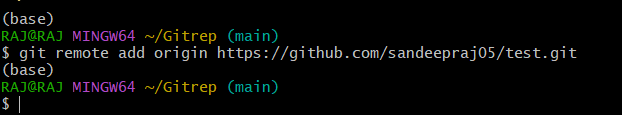
This command is for checking branch and creating branches in current git repository.



**6. git remote**

**Usage: git remote add origin “[URL]”**

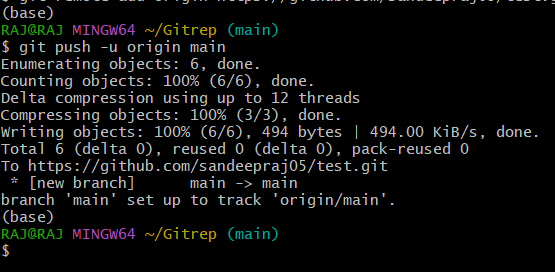
**This command helps you to manage connections to remote repositories.** It allows you to show which remotes are currently connected, but also to add new connections or remove existing ones.

****

**7. git push**

**Usage: git push -u origin [branch name]**

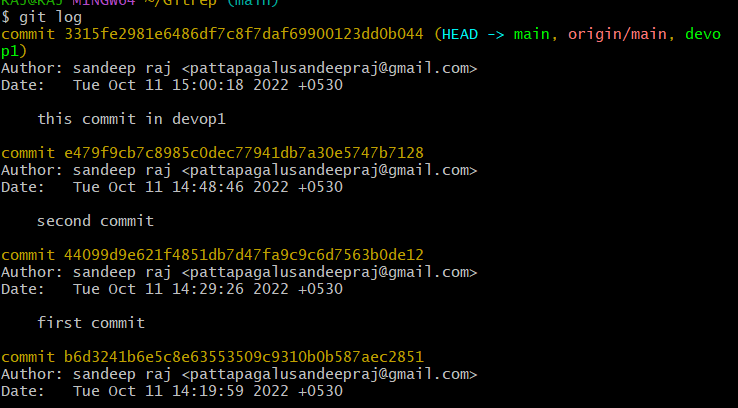
**This command is used to push all directory files from branch to main Github repository.**

****

**8. git log**

**Usage: git log**

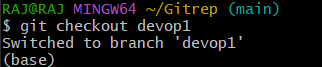
This command is used when we want to check the log for every commit in detail in our repository.

****

**9. git checkout**

**Usage: git checkout [name-of-the-new-branch]**

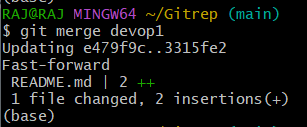
We use this command to navigate to an existing branch, add new files, and commit the files.



**10. git merge**

**Usage: git merge [another branch name]**

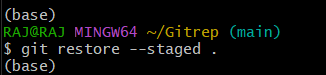
This git merge command is used to combine two branches.



**11. git restore**

**Usage: git restore –staged .**

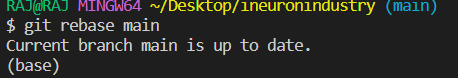
This git restore command used to unstage or even discard uncommitted local changes.



**12. git rebase**

**Usage: git rebase [branch]**

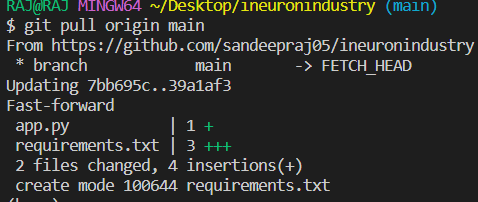
The git rebase command performs an automatic git checkout branch before doing anything else. Otherwise, it remains on the current branch.



**13. git pull**

**Usage: git pull origin [branch name]**

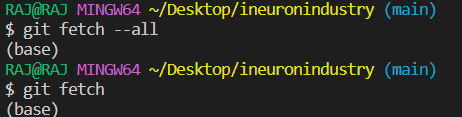
This command which downloads the content from the specified remote repository and then immediately updates the local repo to match the content.



**14. git fetch**

**Usage: git fetch or git fetch –all**

When we use the command git fetch, Git gathers any commit from the target branch that does not exist in our current branch and stores it in our local repository. However, **it does not merge it with our current branch.**



**15. git stash**

**Usage: git stash -u & git stash pop**

git stash -u command is used when we want to stash the untracked files.

git stash pop command is used when we are back on our branch and want to retrieve the code.

