

Git Introduction:

Initialized git local repository using git init , renamed the master branch into main

```
keerthi kanth@XtraPunch MINGW64 ~/Documents/python_lab
$ git init
Initialized empty Git repository in C:/Users/keert/Documents/python_lab/.git/

keerthi kanth@XtraPunch MINGW64 ~/Documents/python_lab (master)
$ git status
On branch master

No commits yet

nothing to commit (create/copy files and use "git add" to track)

keerthi kanth@XtraPunch MINGW64 ~/Documents/python_lab (master)
$ git branch -m main

keerthi kanth@XtraPunch MINGW64 ~/Documents/python_lab (main)
$ git status
On branch main

No commits yet

nothing to commit (create/copy files and use "git add" to track)
```

Added a file into main branch using git add and git commit commands

```
keerthi kanth@XtraPunch MINGW64 ~/Documents/python_lab (main)
$ touch file_for_main_branch.txt

keerthi kanth@XtraPunch MINGW64 ~/Documents/python_lab (main)
$ vi file_for_main_branch.txt

keerthi kanth@XtraPunch MINGW64 ~/Documents/python_lab (main)
$ cat file_for_main_branch.txt
This file belongs to main branch

keerthi kanth@XtraPunch MINGW64 ~/Documents/python_lab (main)
$ git add .
warning: in the working copy of 'file_for_main_branch.txt', LF will be replaced by CRLF the next time git touches it

keerthi kanth@XtraPunch MINGW64 ~/Documents/python_lab (main)
$ git commit -m "Initial commit"
[main (root-commit) c065e17] Initial commit
1 file changed, 1 insertion(+)
create mode 100644 file_for_main_branch.txt
```

Created remote git repository and referenced it using the git add remote command

```
keerthi kanth@XtraPunch MINGW64 ~/Documents/python_lab (main)
$ git remote add origin https://github.com/keertikanth/gitrepoforttesting.git

keerthi kanth@XtraPunch MINGW64 ~/Documents/python_lab (main)
$ git branch -a
* main

keerthi kanth@XtraPunch MINGW64 ~/Documents/python_lab (main)
$ git branch -r
```

Until this point, there is only one main branch and no remote branches are present

```

keerthi kanth@XtraPunch MINGW64 ~/Documents/python_lab (main)
$ git push origin main
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 20 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 262 bytes | 262.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/keertikanth/gitrepofortesting.git
 * [new branch]      main -> main

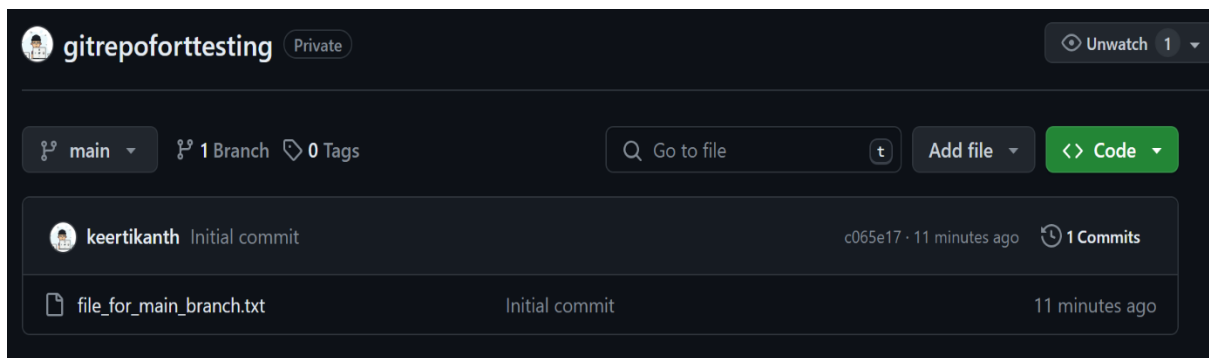
keerthi kanth@XtraPunch MINGW64 ~/Documents/python_lab (main)
$ git branch -r
origin/main

keerthi kanth@XtraPunch MINGW64 ~/Documents/python_lab (main)
$ git branch -a
* main
remotes/origin/main

```

Remote tracking branches are created once the git command "git push origin main" is used

Screenshot of git repo after the above step:



Created another branch (new-branch) using git checkout command using -b option; also committed a file into git local repository

```

keerthi kanth@XtraPunch MINGW64 ~/Documents/python_lab (main)
$ git checkout -b new-branch
Switched to a new branch 'new-branch'

keerthi kanth@XtraPunch MINGW64 ~/Documents/python_lab (new-branch)
$ touch file_for_new_branch.txt

keerthi kanth@XtraPunch MINGW64 ~/Documents/python_lab (new-branch)
$ vi file_for_new_branch.txt

keerthi kanth@XtraPunch MINGW64 ~/Documents/python_lab (new-branch)
$ git status
On branch new-branch
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    file_for_new_branch.txt

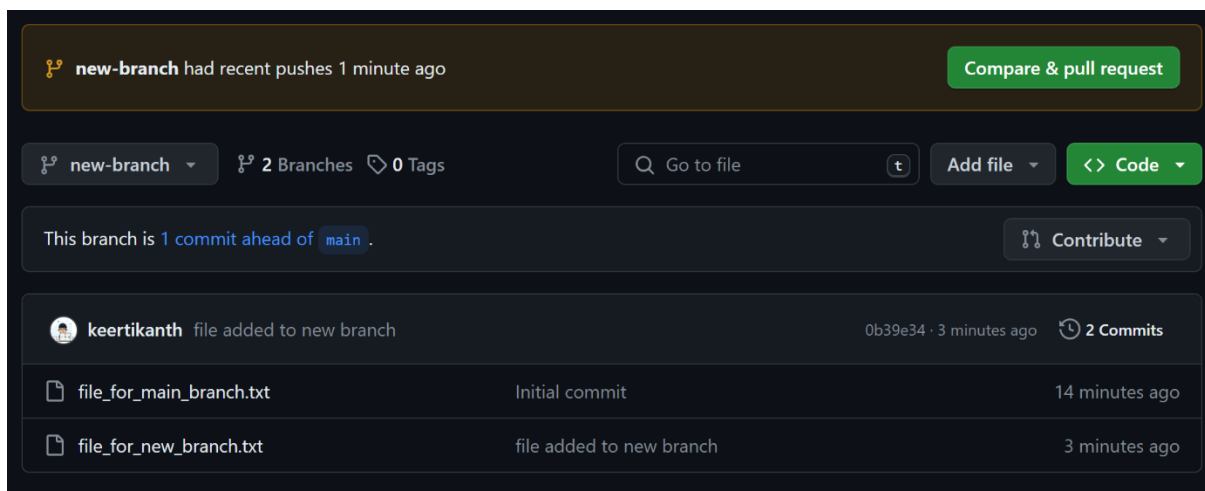
nothing added to commit but untracked files present (use "git add" to track)

keerthi kanth@XtraPunch MINGW64 ~/Documents/python_lab (new-branch)
$ git add .
warning: in the working copy of 'file_for_new_branch.txt', LF will be replaced by CRLF the next time Git touches it

keerthi kanth@XtraPunch MINGW64 ~/Documents/python_lab (new-branch)
$ git commit -m 'file added to new branch'
[new-branch 0b39e34] file added to new branch
1 file changed, 1 insertion(+)
create mode 100644 file_for_new_branch.txt

```

Pushed the commit into github repository



To un-stage a staged file:

created a sample file and staged it using git add command

```
keerthi kanth@XtraPunch MINGW64 ~/Documents/python_lab (main)
$ touch file_to_unstage.txt

keerthi kanth@XtraPunch MINGW64 ~/Documents/python_lab (main)
$ vi file_to_unstage.txt

keerthi kanth@XtraPunch MINGW64 ~/Documents/python_lab (main)
$ cat file_to_unstage.txt
This file is explain how to unstage a file once it is stage (after git add command)

keerthi kanth@XtraPunch MINGW64 ~/Documents/python_lab (main)
$ git status
On branch main
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        file_to_unstage.txt

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

keerthi kanth@XtraPunch MINGW64 ~/Documents/python_lab (main)
$ git add .
warning: in the working copy of 'file_to_unstage.txt', LF will be replaced by CRLF the next time Git touches it

keerthi kanth@XtraPunch MINGW64 ~/Documents/python_lab (main)
$ git status
On branch main
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   file_to_unstage.txt
```

git restore command is used to un-stage a staged file, below screenshot is result of restore command

```
keerthi kanth@XtraPunch MINGW64 ~/Documents/python_lab (main)
$ git restore --staged file_to_unstage.txt

keerthi kanth@XtraPunch MINGW64 ~/Documents/python_lab (main)
$ git status
On branch main
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        file_to_unstage.txt

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

To update a file to its last commit:

Pushed a file into github and later added one more line using vim

```
keerthi kanth@XtraPunch MINGW64 ~/Documents/python_lab (main)
$ git commit -m 'adding file to have a initial state'
[main 442e746] adding file to have a initial state
1 file changed, 3 insertions(+)
create mode 100644 file_to_rm_updates.txt

keerthi kanth@XtraPunch MINGW64 ~/Documents/python_lab (main)
$ git push origin main
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 20 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 469 bytes | 469.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/keertikanth/gitrepoforttesting.git
 c065e17..442e746  main -> main

keerthi kanth@XtraPunch MINGW64 ~/Documents/python_lab (main)
$ vi file_to_rm_updates.txt

keerthi kanth@XtraPunch MINGW64 ~/Documents/python_lab (main)
$ cat file_to_rm_updates.txt
This is my initial code and added this file to the repository;
if this a config file; i dont want to change it;
if i accidentally updated it; my local repo tracks the file ; Hence it can update it to the initial state (i.e file contains only three lines)

I started adding new line
```

In order to retrieve its original/ last-commit state, git checkout command is used on the file itself

```
keerthi kanth@XtraPunch MINGW64 ~/Documents/python_lab (main)
$ git checkout file_to_rm_updates.txt
Updated 1 path from the index

keerthi kanth@XtraPunch MINGW64 ~/Documents/python_lab (main)
$ cat file_to_rm_updates.txt
This is my initial code and added this file to the repository;
if this a config file; i dont want to change it;
if i accidentally updated it; my local repo tracks the file ; Hence it can update it to the initial state (i.e file contains only three lines)
```

Delete a file from the repository.

```
keerthi kanth@XtraPunch MINGW64 ~/Documents/python_lab (main)
$ git ls-files
file_for_main_branch.txt
file_to_rm_updates.txt

keerthi kanth@XtraPunch MINGW64 ~/Documents/python_lab (main)
$ git status
On branch main
nothing to commit, working tree clean

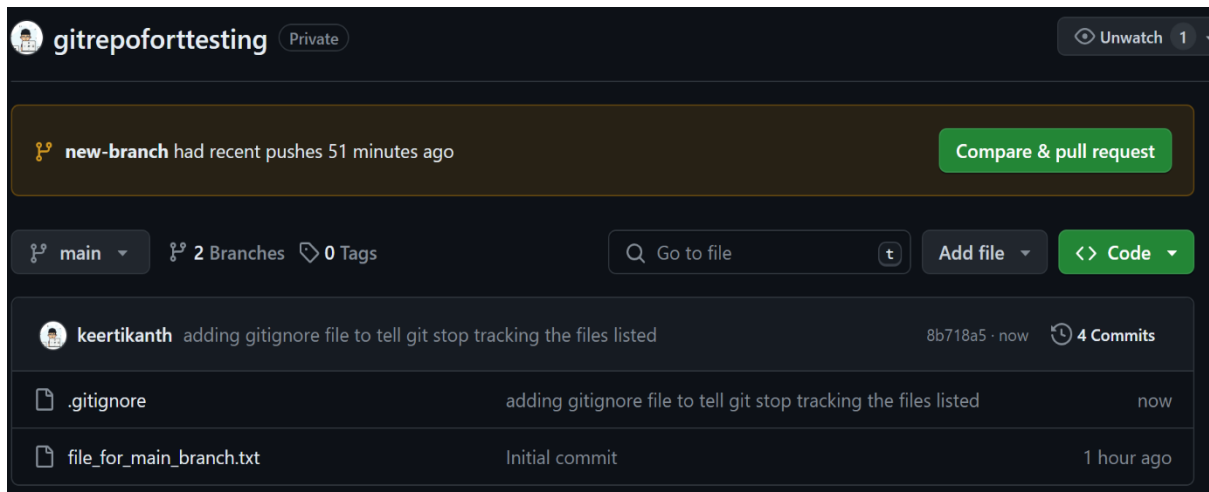
keerthi kanth@XtraPunch MINGW64 ~/Documents/python_lab (main)
$ git rm file_to_rm_updates.txt
rm 'file_to_rm_updates.txt'

keerthi kanth@XtraPunch MINGW64 ~/Documents/python_lab (main)
$ git status
On branch main
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        deleted:    file_to_rm_updates.txt

keerthi kanth@XtraPunch MINGW64 ~/Documents/python_lab (main)
$ git commit -m 'Deleted a config file'
[main 1f22db0] Deleted a config file
1 file changed, 3 deletions(-)
delete mode 100644 file_to_rm_updates.txt

keerthi kanth@XtraPunch MINGW64 ~/Documents/python_lab (main)
$ git push origin main
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 20 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (2/2), 226 bytes | 226.00 KiB/s, done.
Total 2 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/keertikanth/gitrepoforttesting.git
 442e746..1f22db0  main -> main
```

Restricted a config.txt file using .gitignore file:



Rollback a commit:

Created a commit to rollback from the commit stage (commit comment : "To undo a commit")

```
keerthi kanth@XtraPunch MINGW64 ~/Documents/python_lab (main)
$ git log
commit 231ffe78d0d3ed37491ce371fba67b829c43372f (HEAD -> main)
Author: keertikanth <keerthikanth51@gmail.com>
Date: Sun May 5 19:53:53 2024 +0530

    To undo a commit
```

git reset using --soft option

```
keerthi kanth@XtraPunch MINGW64 ~/Documents/python_lab (main)
$ git reset --soft HEAD~1

keerthi kanth@XtraPunch MINGW64 ~/Documents/python_lab (main)
$ git log
commit 8b718a575c2b57cd268128d2891970b019cebe45 (HEAD -> main, origin/main)
Author: keertikanth <keerthikanth51@gmail.com>
Date: Sun May 5 19:41:57 2024 +0530

    adding gitignore file to tell git stop tracking the files listed

commit 1f22db0ef25c436bb95539f6908e0eb800155671
Author: keertikanth <keerthikanth51@gmail.com>
Date: Sun May 5 19:29:49 2024 +0530

    Deleted a config file

commit 442e746905514ee9c00a1f894444bae3b75e1f27
Author: keertikanth <keerthikanth51@gmail.com>
Date: Sun May 5 19:17:47 2024 +0530

    adding file to have a initial state

commit c065e17899a7aa637c606bb9bf119e28c05bf4c5
Author: keertikanth <keerthikanth51@gmail.com>
Date: Sun May 5 18:38:35 2024 +0530

    Initial commit

keerthi kanth@XtraPunch MINGW64 ~/Documents/python_lab (main)
$ git status
On branch main
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   file_to_rm_after_commit.txt
```

git reset using --hard option; File is completely removed from the working directory using the git reset --hard option

```
keerthi kanth@XtraPunch MINGW64 ~/Documents/python_lab (main)
$ git reset --hard HEAD~1
HEAD is now at 8b718a5 adding gitignore file to tell git stop tracking the files listed

keerthi kanth@XtraPunch MINGW64 ~/Documents/python_lab (main)
$ git status
On branch main
nothing to commit, working tree clean

keerthi kanth@XtraPunch MINGW64 ~/Documents/python_lab (main)
$ ls
config.txt  file_for_main_branch.txt

keerthi kanth@XtraPunch MINGW64 ~/Documents/python_lab (main)
$ git log
commit 8b718a575c2b57cd268128d2891970b019cebe45 (HEAD -> main, origin/main)
Author: keertikanth <keertikanth51@gmail.com>
Date: Sun May 5 19:41:57 2024 +0530

    adding gitignore file to tell git stop tracking the files listed
```

Learning about file modifications using git diff command:

Pushed a file to check git diff command

```
keerthi kanth@XtraPunch MINGW64 ~/Documents/python_lab (main)
$ touch file_for_diff_changes.txt

keerthi kanth@XtraPunch MINGW64 ~/Documents/python_lab (main)
$ echo "This file is for print the changed recognized by git" > file_for_diff_changes.txt

keerthi kanth@XtraPunch MINGW64 ~/Documents/python_lab (main)
$ cat file_for_diff_changes.txt
This file is for print the changed recognized by git

keerthi kanth@XtraPunch MINGW64 ~/Documents/python_lab (main)
$ git status
On branch main
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        file_for_diff_changes.txt

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

keerthi kanth@XtraPunch MINGW64 ~/Documents/python_lab (main)
$ git add .
warning: in the working copy of 'file_for_diff_changes.txt', LF will be replaced by CRLF the next time git touches it

keerthi kanth@XtraPunch MINGW64 ~/Documents/python_lab (main)
$ git commit -m "Commit to check diff command"
[main 4841f41] Commit to check diff command
1 file changed, 1 insertion(+)
create mode 100644 file_for_diff_changes.txt

keerthi kanth@XtraPunch MINGW64 ~/Documents/python_lab (main)
$ git push origin main
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 20 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 388 bytes | 388.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/keertikanth/gitrepofortesting.git
   8b718a5..4841f41  main -> main
```

git diff tracked the modifications; below is its result

```
keerthi kanth@XtraPunch MINGW64 ~/Documents/python_lab (main)
$ echo "Few more lines added to the same file to check the diff command" > file_for_diff_changes.txt

keerthi kanth@XtraPunch MINGW64 ~/Documents/python_lab (main)
$ git diff
warning: in the working copy of 'file_for_diff_changes.txt', LF will be replaced by CRLF the next time git touches it
diff --git a/file_for_diff_changes.txt b/file_for_diff_changes.txt
index 678214d..2f1e580 100644
--- a/file_for_diff_changes.txt
+++ b/file_for_diff_changes.txt
@@ -1,1 @@
-This file is for print the changed recognized by git
+Few more lines added to the same file to check the diff command
```

Merging a new-branch to the main branch:

git merge will merge the new-branch into main branch, so the files created inside the new-branch is also available inside the main branch. Below screenshot tells the details about git merge command

```
keerthi kanth@XtraPunch MINGW64 ~/Documents/python_lab (remote-branch)
$ git checkout main
Switched to branch 'main'

keerthi kanth@XtraPunch MINGW64 ~/Documents/python_lab (main)
$ ls
config.txt  file_for_diff_changes.txt  file_for_main_branch.txt

keerthi kanth@XtraPunch MINGW64 ~/Documents/python_lab (main)
$ git checkout new-branch
Switched to branch 'new-branch'

keerthi kanth@XtraPunch MINGW64 ~/Documents/python_lab (new-branch)
$ ls
config.txt  file_for_main_branch.txt  file_for_new_branch.txt

keerthi kanth@XtraPunch MINGW64 ~/Documents/python_lab (new-branch)
$ git checkout main
Switched to branch 'main'

keerthi kanth@XtraPunch MINGW64 ~/Documents/python_lab (main)
$ git merge new-branch
Merge made by the 'ort' strategy.
 file_for_new_branch.txt | 1 +
1 file changed, 1 insertion(+)
create mode 100644 file_for_new_branch.txt

keerthi kanth@XtraPunch MINGW64 ~/Documents/python_lab (main)
$ ls
config.txt  file_for_diff_changes.txt  file_for_main_branch.txt  file_for_new_branch.txt
```