

Git commands

Note

In jupyter file aslo execute cmd by using prefix ! eg: !git add filename.txt

Execution of commmands done by

- Git Bash
- Windows Command Prompt

cat and **type** command are same

1.Initialize Git

```
In [3]: git init -b main
```

Initialized empty Git repository in E:/Git_material_datamites/Git_repo_project/.git/

2.Git Version

```
In [ ]:
```

```
In [4]: git --version
```

git version 2.49.0.windows.1

3.Git Config

```
In [5]: git config --global user.name "username_github"
```

```
In [6]: git config --global user.email "github_email"
```

4.Git Status

```
In [7]: git status
```

On branch main

No commits yet

Untracked files:

(use "git add <file>..." to include in what will be committed)

GIT_NOTES.ipynb

demo.txt

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

Lets create some files in current directory--demo.txt

Check No. of file is cwd

Read the file using cat command

```
In [ ]: # can use cat or type command in git bash
```

```
In [10]: type demo.txt
```

this is line 1

this is line 2

```
In [11]: pwd
```

```
Out[11]: 'E:\\Git_material_datamites\\Git_repo_project'
```

Check The status

```
In [12]: git status
```

On branch main

No commits yet

Untracked files:

(use "git add <file>..." to include in what will be committed)

.ipynb_checkpoints/

GIT_NOTES.ipynb

demo.txt

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

5.Add the files to local repo-Stagging the files

```
In [13]: git add demo.txt
```

```
In [14]: git status
```

On branch main

No commits yet

Changes to be committed:

(use "git rm --cached <file>..." to unstage)

new file: demo.txt

Untracked files:

(use "git add <file>..." to include in what will be committed)

.ipynb_checkpoints/

GIT_NOTES.ipynb

Check log Details

In [23]: `git log`

fatal: your current branch 'main' does not have any commits yet

6.Commit the file

In [15]: `git commit -m "first commit"`

[main (root-commit) fa41be8] first commit
1 file changed, 2 insertions(+)
create mode 100644 demo.txt

7.Check log Details

In [16]: `git log`

commit fa41be8fe752a75c618f0e394cfc444ab5b962ea
Author: jsakshu <jsakshuammi@gmail.com>
Date: Tue Jul 1 18:00:29 2025 +0530

first commit

Modify the demo.txt file

In [17]: `type demo.txt`

this is line 1
this is line 2

this is line 3
this is line 4

Check Git status for modified file

In [18]: `git status`

On branch main

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: demo.txt

Untracked files:

(use "git add <file>..." to include in what will be committed)

.ipynb_checkpoints/

GIT_NOTES.ipynb

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

commit for second time before that staging the file agin

```
In [19]: git add demo.txt
```

```
In [20]: git status
```

On branch main

Changes to be committed:

(use "git restore --staged <file>..." to unstage)

modified: demo.txt

Untracked files:

(use "git add <file>..." to include in what will be committed)

.ipynb_checkpoints/

GIT_NOTES.ipynb

```
In [21]: git commit -m "second commit"
```

[main b270235] second commit

1 file changed, 4 insertions(+), 1 deletion(-)

```
In [22]: git log
```

commit b2702350818585bfc35e3d05e23bb9f8b7813379

Author: jsakshu <jsakshuammi@gmail.com>

Date: Tue Jul 1 18:02:42 2025 +0530

second commit

commit fa41be8fe752a75c618f0e394cfc444ab5b962ea

Author: jsakshu <jsakshuammi@gmail.com>

Date: Tue Jul 1 18:00:29 2025 +0530

first commit

```
In [23]: git status
```

On branch main

Untracked files:

(use "git add <file>..." to include in what will be committed)

.ipynb_checkpoints/

GIT_NOTES.ipynb

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

8.Create a remote repo

```
In [24]: git remote add origin "https://github.com/jsakshu/first_repo.git"
```

```
In [25]: git push origin main
```

```
To https://github.com/jsakshu/first_repo.git
* [new branch]      main -> main
```

**** Note **** The below content is not an output of above, but when we run push, it asks to authenticate git in system-login to github info: please complete authentication in your browser... Enumerating objects: 6, done. Counting objects: 100% (6/6), done. Delta compression using up to 8 threads Compressing objects: 100% (3/3), done. Writing objects: 100% (6/6), 502 bytes | 167.00 KiB/s, done. Total 6 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)

- Check GITHUB and can see files are pushed
- create a file in github (pull_check.txt), let's try to pull the file
- before that check the status whether pull_check.txt can see in git

```
In [26]: git status
```

On branch main

Untracked files:

```
(use "git add <file>..." to include in what will be committed)
.ipynb_checkpoints/
GIT_NOTES.ipynb
```

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

9.pull request

```
In [27]: git pull origin main
```

Updating b270235..145edce

Fast-forward

```
requirement.txt | 1 +
1 file changed, 1 insertion(+)
create mode 100644 requirement.txt
```

```
From https://github.com/jsakshu/first_repo
* branch          main      -> FETCH_HEAD
b270235..145edce  main      -> origin/main
```

```
In [28]: git status
```

On branch main

Untracked files:

```
(use "git add <file>..." to include in what will be committed)
.ipynb_checkpoints/
GIT_NOTES.ipynb
```

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

```
In [29]: ls
```

Volume in drive E is Technical
Volume Serial Number is 0C14-7CEB

Directory of E:\Git_material_datamites\Git_repo_project

```
01-07-2025  18:06    <DIR>          .
01-07-2025  17:53    <DIR>          ..
01-07-2025  17:59    <DIR>          .ipynb_checkpoints
01-07-2025  18:02             64 demo.txt
01-07-2025  18:05        17,677 GIT_NOTES.ipynb
01-07-2025  18:06         52 requirement.txt
               3 File(s)        17,793 bytes
               3 Dir(s)  96,316,895,232 bytes free
```

10. create new branches

In [30]: `git branch`

```
* main
```

In [31]: `git branch b1`

In [32]: `git branch b2`

In [33]: `git branch`

```
b1
b2
* main
```

In [34]: `git branch --all`

```
b1
b2
* main
remotes/origin/main
```

In [35]: `git branch -d b2`

Deleted branch b2 (was 145edce).

In [36]: `git branch`

```
b1
* main
```

In [37]: `git branch --all`

```
b1
* main
remotes/origin/main
```

Create files in branch1

1. Switch from main to branch b1

```
In [39]: git checkout b1
```

Switched to branch 'b1'

```
In [40]: git status
```

On branch b1

Untracked files:

(use "git add <file>..." to include in what will be committed)

- .ipynb_checkpoints/
- GIT_NOTES.ipynb

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

2.lets work on demo.txt

a.check files in b1 and all files in the main branch will reflected in b1

```
In [43]: git ls-tree --name-only -r b1
```

demo.txt
requirement.txt

b.modify the file demo.txt

```
In [46]: type demo.txt
```

this is line 1
this is line 2

this is line 3
this is line 4

this is line 5-->modified by b1
this is line 6-->modified by b1

```
In [50]: git status
```

On branch b1

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: demo.txt

Untracked files:

(use "git add <file>..." to include in what will be committed)

- .ipynb_checkpoints/
- GIT_NOTES.ipynb

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

```
In [52]: git add demo.txt
```

```
In [53]: git status
```

On branch b1

Changes to be committed:

(use "git restore --staged <file>..." to unstage)
modified: demo.txt

Untracked files:

(use "git add <file>..." to include in what will be committed)
.ipynb_checkpoints/
GIT_NOTES.ipynb

```
In [54]: git commit -m "first commit by b1"
```

```
[b1 0555602] first commit by b1  
1 file changed, 5 insertions(+), 1 deletion(-)
```

```
In [56]: git log
```

```
commit 05556021ff1780895d82924560af5e8a0e8fc774  
Author: jsakshu <jsakshuammi@gmail.com>  
Date: Tue Jul 1 18:21:53 2025 +0530  
  
    first commit by b1  
  
commit 145edcee3b47d93c701eccbf7f3f8cb751ffa03a  
Author: jsakshu <96901179+jsakshu@users.noreply.github.com>  
Date: Tue Jul 1 18:06:12 2025 +0530  
  
    Create requirement.txt  
  
commit b2702350818585bfc35e3d05e23bb9f8b7813379  
Author: jsakshu <jsakshuammi@gmail.com>  
Date: Tue Jul 1 18:02:42 2025 +0530  
  
    second commit  
  
commit fa41be8fe752a75c618f0e394cfc444ab5b962ea  
Author: jsakshu <jsakshuammi@gmail.com>  
Date: Tue Jul 1 18:00:29 2025 +0530  
  
    first commit
```

2.switch to main branch

```
In [58]: git checkout main
```

```
Switched to branch 'main'
```

```
In [59]: git status
```

On branch main

Untracked files:

(use "git add <file>..." to include in what will be committed)
.ipynb_checkpoints/
GIT_NOTES.ipynb

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

```
In [60]: git ls-tree --name-only -r b1
```



```
demo.txt  
requirement.txt
```

```
In [61]: type demo.txt
```

```
this is line 1  
this is line 2
```

```
this is line 3  
this is line 4
```

The main branch has no idea on the modification done by branch1

11.Track the version using commit id

create new file track.txt and commit it

```
In [63]: git checkout b1
```

```
Switched to branch 'b1'
```

```
In [64]: git add track.txt
```

```
In [65]: git commit -m "commit 1"
```

```
[b1 855dbdf] commit 1  
1 file changed, 12 insertions(+)  
create mode 100644 track.txt
```

```
In [66]: type track.txt
```

```
version 1  
Git is a version control system that helps you:
```

Track changes to your code

Collaborate with others

Revert to earlier versions

Manage branches for different features or fixes

Itâ€™s widely used in software development, especially with platforms like GitHub, GitLab, and Bitbucket.

modify in track.txt

```
In [69]: git status
```

On branch b1

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: track.txt

Untracked files:

(use "git add <file>..." to include in what will be committed)

.ipynb_checkpoints/

GIT_NOTES.ipynb

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

```
In [75]: git add track.txt
```

```
In [71]: git commit -m "commit 1"
```

```
[b1 511297e] commit 1
```

```
1 file changed, 8 insertions(+), 1 deletion(-)
```

```
In [72]: type track.txt
```

```
version 1
```

Git is a version control system that helps you:

Track changes to your code

Collaborate with others

Revert to earlier versions

Manage branches for different features or fixes

It's widely used in software development, especially with platforms like GitHub, GitLab, and Bitbucket.

```
version -2
```

```
git init
```

```
git status
```

```
git add .
```

```
git commit -m "Message"
```

```
git push origin branch_name
```

roll back to previous version

```
In [74]: git log
```

```
commit 511297e7580c83da59294e292e2be53396b8ecf2
Author: jsakshu <jsakshuammi@gmail.com>
Date: Tue Jul 1 18:31:00 2025 +0530
```

commit 1

```
commit 855dbdf6c5a851ea217e312bbda156d2fb4a980e
Author: jsakshu <jsakshuammi@gmail.com>
Date: Tue Jul 1 18:28:15 2025 +0530
```

commit 1

```
commit 05556021ff1780895d82924560af5e8a0e8fc774
Author: jsakshu <jsakshuammi@gmail.com>
Date: Tue Jul 1 18:21:53 2025 +0530
```

first commit by b1

```
commit 145edcee3b47d93c701eccbf7f3f8cb751ffa03a
Author: jsakshu <96901179+jsakshu@users.noreply.github.com>
Date: Tue Jul 1 18:06:12 2025 +0530
```

Create requirement.txt

```
commit b2702350818585bfc35e3d05e23bb9f8b7813379
Author: jsakshu <jsakshuammi@gmail.com>
Date: Tue Jul 1 18:02:42 2025 +0530
```

second commit

```
commit fa41be8fe752a75c618f0e394cfc444ab5b962ea
Author: jsakshu <jsakshuammi@gmail.com>
Date: Tue Jul 1 18:00:29 2025 +0530
```

first commit

Noteabove log ,given same commit name.so i modified dots at the end and committed with msg "commit 3"

```
In [77]: git add track.txt
```

```
In [78]: git commit -m "commit 3"
```

```
[b1 4b5b7db] commit 3
1 file changed, 1 insertion(+), 1 deletion(-)
```

```
In [79]: git log
```

```
commit 4b5b7db7f0cbfc14a0d25ea647a857ac91e6fa9a
Author: jsakshu <jsakshuammi@gmail.com>
Date: Tue Jul 1 18:35:20 2025 +0530
```

commit 3

```
commit 511297e7580c83da59294e292e2be53396b8ecf2
Author: jsakshu <jsakshuammi@gmail.com>
Date: Tue Jul 1 18:31:00 2025 +0530
```

commit 1

```
commit 855dbdf6c5a851ea217e312bbda156d2fb4a980e
Author: jsakshu <jsakshuammi@gmail.com>
Date: Tue Jul 1 18:28:15 2025 +0530
```

commit 1

```
commit 05556021ff1780895d82924560af5e8a0e8fc774
Author: jsakshu <jsakshuammi@gmail.com>
Date: Tue Jul 1 18:21:53 2025 +0530
```

first commit by b1

```
commit 145edcee3b47d93c701eccbf7f3f8cb751ffa03a
Author: jsakshu <96901179+jsakshu@users.noreply.github.com>
Date: Tue Jul 1 18:06:12 2025 +0530
```

Create requirement.txt

```
commit b2702350818585bfc35e3d05e23bb9f8b7813379
Author: jsakshu <jsakshuammi@gmail.com>
Date: Tue Jul 1 18:02:42 2025 +0530
```

second commit

```
commit fa41be8fe752a75c618f0e394cfc444ab5b962ea
Author: jsakshu <jsakshuammi@gmail.com>
Date: Tue Jul 1 18:00:29 2025 +0530
```

first commit

switch to commit 1

```
In [80]: git checkout 855dbdf6c5a851ea217e312bbda156d2fb4a980e
```

Note: switching to '855dbdf6c5a851ea217e312bbda156d2fb4a980e'.

You are in 'detached HEAD' state. You can look around, make experimental changes and commit them, and you can discard any commits you make in this state without impacting any branches by switching back to a branch.

If you want to create a new branch to retain commits you create, you may do so (now or later) by using -c with the switch command. Example:

```
git switch -c <new-branch-name>
```

Or undo this operation with:

```
git switch -
```

Turn off this advice by setting config variable advice.detachedHead to false

HEAD is now at 855dbdf commit 1

In [81]: `type track.txt`

version 1

Git is a version control system that helps you:

Track changes to your code

Collaborate with others

Revert to earlier versions

Manage branches for different features or fixes

It's widely used in software development, especially with platforms like GitHub, GitLab, and Bitbucket.

In [82]: `git checkout fa41be8fe752a75c618f0e394cfc444ab5b962ea`

Previous HEAD position was 855dbdf commit 1

HEAD is now at fa41be8 first commit

In [83]: `type demo.txt`

this is line 1

this is line 2

In [85]: `git status`

HEAD detached at fa41be8

Untracked files:

```
(use "git add <file>..." to include in what will be committed)
.ipynb_checkpoints/
GIT_NOTES.ipynb
```

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

In [86]: `git branch`

* (HEAD detached at fa41be8)

b1

main

```
In [87]: git checkout b1
```

```
Previous HEAD position was fa41be8 first commit  
Switched to branch 'b1'
```

```
In [88]: git branch
```

```
* b1  
main
```

12. push to Github

```
In [89]: git push origin b1
```

```
remote:  
remote: Create a pull request for 'b1' on GitHub by visiting:  
remote:      https://github.com/jsakshu/first_repo/pull/new/b1  
remote:  
To https://github.com/jsakshu/first_repo.git  
* [new branch]      b1 -> b1
```

```
In [1]: git branch
```

```
* b1  
main
```

```
In [2]: type demo.txt
```

```
this is line 1  
this is line 2
```

```
this is line 3  
this is line 4
```

```
this is line 5-->modified by b1  
this is line 6-->modified by b1
```

```
In [3]: git branch
```

```
* b1  
main
```

```
In [4]: git checkout main
```

```
Switched to branch 'main'
```

```
In [6]: type demo.txt
```

```
this is line 1  
this is line 2
```

```
this is line 3  
this is line 4
```

NOTE in main branch demo.txt is not affected

13. Tagging--->create,list of tags,push to remote,delete and switching tags

```
In [9]: git tag -a v.10 -m "release version 1.0"
```

```
In [11]: git tag
```

v.10

```
In [13]: git push origin --tags
```

```
To https://github.com/jsakshu/first_repo.git
* [new tag]          v.10 -> v.10
```

```
In [ ]: git tag -d v1.0
git push origin --delete v1.0
```

git tag -d v1.0 ---->delete locally
git push origin --delete v1.0----> rmotely

```
In [ ]: git checkout v.10
```

14. Merging Files with Main

```
In [17]: git checkout main
```

Already on 'main'

```
In [18]: git merge b1
```

```
Updating 145edce..4b5b7db
Fast-forward
 demo.txt | 6 +++++-
 track.txt | 19 +++++++++++++++++++++
 2 files changed, 24 insertions(+), 1 deletion(-)
 create mode 100644 track.txt
```

```
In [19]: type demo.txt
```

```
this is line 1
this is line 2
```

```
this is line 3
this is line 4
```

```
this is line 5-->modified by b1
this is line 6-->modified by b1
```

Note Now the main branch is affected

15 git clone

git clone is used to copy (download) an existing Git repository from GitHub (or any remote) to your local machine.

```
In [25]: git clone "https://github.com/jsakshu7/sample_repo.git"
```

Cloning into 'sample_repo'...

above repo consist of only one readme.md file and its downloaded in current directory

Delete Repo

```
In [ ]: git remote remove origin
```

16.GITHUB

Topics

- create account
- create repo
- CREATE,DELETE files
- ADD COLLABRATE,
- create and delete BRANCH
- COMMIT CHANGES
- tagging
- permission

Created By :Akshatha S
