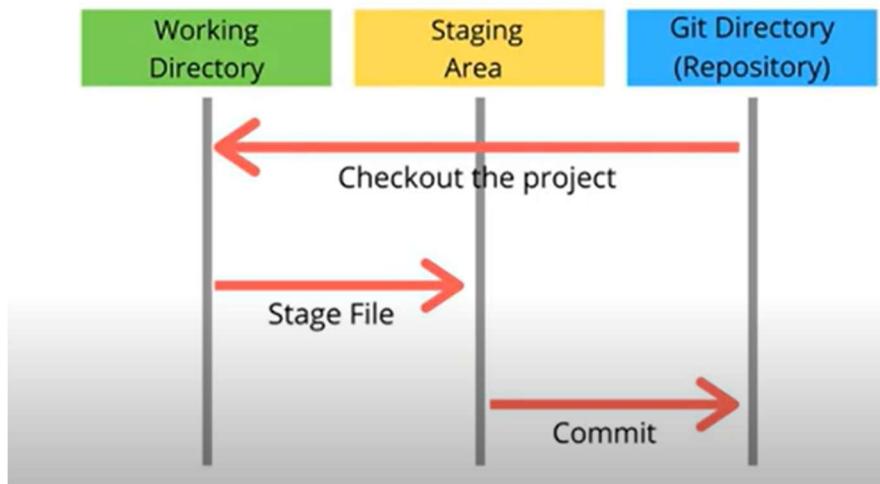


Architecture of Git- Three-Stage Architecture



Working Directory- Local PC directory (Example- C/D/E Drive)

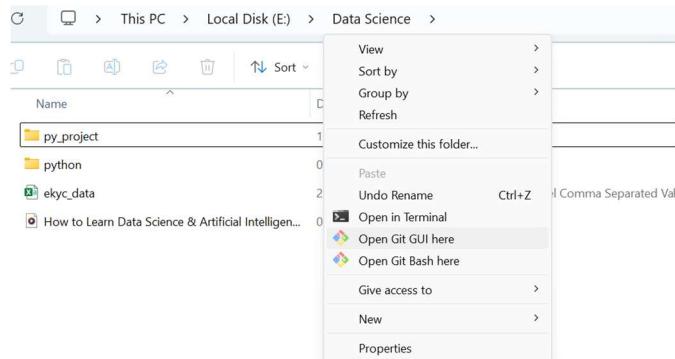
Staging Area-

Git Directory (Repository)-

- i) Local Repository (Local PC .git folder where all commit history is reserved)
- ii) Remote Repository (Online Platform like GitHub/Git Lab where we can preserve all changes or create new ones)

Upload Folder in Git/GitHub repository:

- i) Right click on blank space inside the directory where folder exist- for example we want to upload “py_project” folder from E:\Data Science\



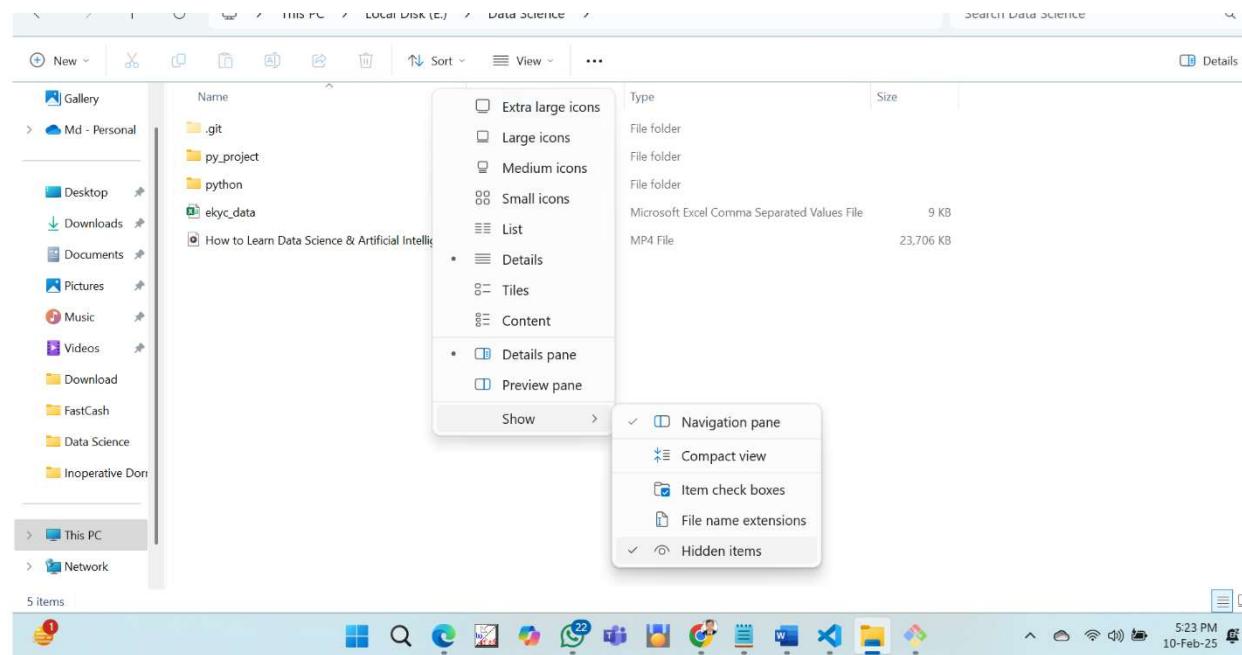
- ii) Click on “Open Git Bash Here”

- iii) Enter command “git status” if it shows Error message “fatal: not a git repository (or any of the parent directories): .git” that means we haven’t told Git about this project yet.
- iv) To resolve this error, we have to initialize the project
- v) Write command “git init”
This command initiate .git folder as below

Name	Date modified	Type
.git	10-Feb-25 5:14 PM	File folder
py_project	10-Feb-25 2:41 PM	File folder

If this folder is not shows, enable “Hidden items” from “View” option

go to view> Show>Hidden items



Sending file to “Staging Area” for Commit:

- i) Enter command “git add .”

```
User@DESKTOP-VI2MRGR MINGW64 /e/Data Science (main)
$ git add .
warning: in the working copy of 'python/.ipynb_checkpoints/day1-checkpoint.ipynb',
', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'python/day1.ipynb', LF will be replaced by CRLF
the next time Git touches it

User@DESKTOP-VI2MRGR MINGW64 /e/Data Science (main)
```

- ii) For checking status enter “git status”

```
User@DESKTOP-VI2MRGR MINGW64 /e/Data Science (main)
$ git status
On branch main

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:  How to Learn Data Science & Artificial Intelligence The Ultimate Guide on ML & AI (Bangla).mp4
    new file:  ekyc_data.csv
    new file:  py_project/print_range.py
    new file:  python/.day1.ipynb.layout
    new file:  python/.ipynb_checkpoints/day1-checkpoint.ipynb
    new file:  python/.virtual_documents/day1.ipynb
    new file:  python/anaconda_projects/db/project_filebrowser.db
    new file:  python/day1.ipynb
```

```
User@DESKTOP-VI2MRGR MINGW64 /e/Data Science (main)
$ |
```

- iii) Enter git commit command and provide a remarks
- \$ git commit -m "python project"

```
User@DESKTOP-VI2MRGR MINGW64 /e/Data Science (main)
$ git commit -m "python project"
[main (root-commit) d87aada] python project
 8 files changed, 2390 insertions(+)
 create mode 100644 How to Learn Data Science & Artificial Intelligence The Ultimate Guide on ML & AI (Bangla).mp4
 create mode 100644 ekyc_data.csv
 create mode 100644 py_project/print_range.py
 create mode 100644 python/.day1.ipynb.layout
 create mode 100644 python/.ipynb_checkpoints/day1-checkpoint.ipynb
 create mode 100644 python/.virtual_documents/day1.ipynb
 create mode 100644 python/anaconda_projects/db/project_filebrowser.db
 create mode 100644 python/day1.ipynb

User@DESKTOP-VI2MRGR MINGW64 /e/Data Science (main)
$ git status
On branch main
nothing to commit, working tree clean

User@DESKTOP-VI2MRGR MINGW64 /e/Data Science (main)
$ git log
commit d87aada4eed195dd16e5a67d26a96efa3341cca3 (HEAD -> main)
Author: anisjkb <anisjkb@gmail.com>
Date:   Mon Feb 10 18:54:33 2025 +0600

  python project

User@DESKTOP-VI2MRGR MINGW64 /e/Data Science (main)
$ |
```

- iv) Create a new repository at GitHub

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere?
[Import a repository.](#)

Required fields are marked with an asterisk (*).

Owner * Repository name *

anisjkb / python

python is available.

Great repository names are short and memorable. Need inspiration? How about [congenial-succotash](#) ?

Description (optional)

Public

Anyone on the internet can see this repository. You choose who can commit.

Private

You choose who can see and commit to this repository.

Initialize this repository with:

Add a README file

v) git remote add origin

copy these two line “git remote add origin https://github.com/anisjkb/python.git” and

“git push -u origin main” after creating a new repository

Start coding with Codespaces
Add a README file and start coding in a secure, configurable, and dedicated development environment.
[Create a codespace](#)

Add collaborators to this repository
Search for people using their GitHub username or email address.
[Invite collaborators](#)

Quick setup — if you've done this kind of thing before
Set up in Desktop or HTTPS SSH https://github.com/anisjkb/pyhon_project.git
Get started by [creating a new file](#) or [uploading an existing file](#). We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

...or create a new repository on the command line

```
echo "# pyhon_project" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/anisjkb/pyhon_project.git
git push -u origin main
```

```
MINGW64:/e/Data Science
User@DESKTOP-VI2MRGR MINGW64 /e/Data Science (main)
$ git status
On branch main
nothing to commit, working tree clean

User@DESKTOP-VI2MRGR MINGW64 /e/Data Science (main)
$ git remote add origin https://github.com/anisjkb/python_project.git

User@DESKTOP-VI2MRGR MINGW64 /e/Data Science (main)
$ git push -u origin main
info: please complete authentication in your browser...
Enumerating objects: 15, done.
Counting objects: 100% (15/15), done.
Delta compression using up to 8 threads
Compressing objects: 100% (10/10), done.
Writing objects: 100% (15/15), 22.29 MiB | 2.71 MiB/s, done.
Total 15 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/anisjkb/python_project.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.

User@DESKTOP-VI2MRGR MINGW64 /e/Data Science (main)
$
```

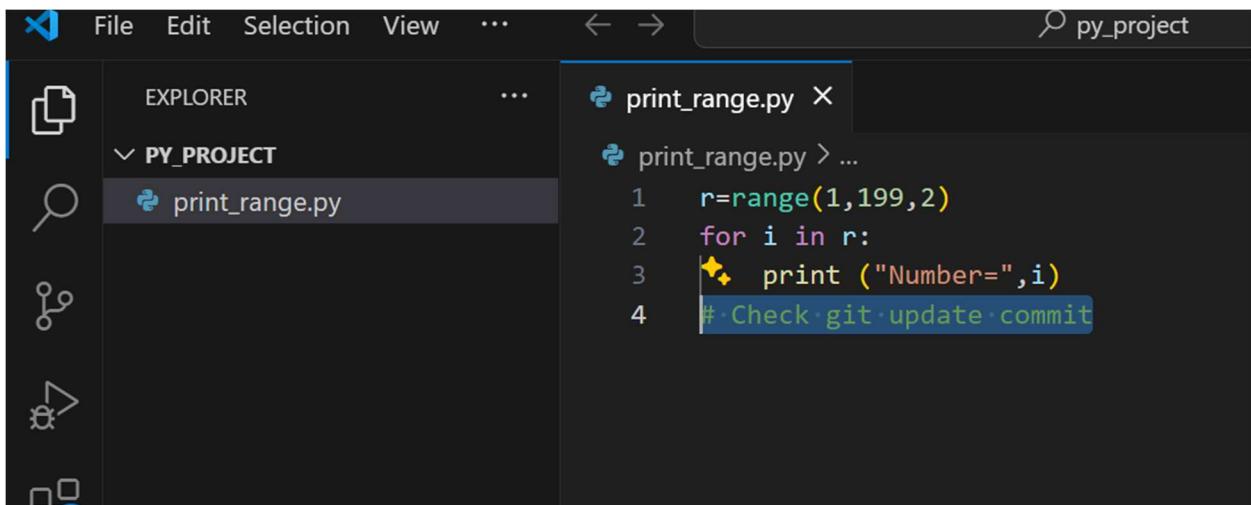
The screenshot shows a GitHub repository page for 'anisjkb/python_project'. The URL in the address bar is 'github.com/anisjkb/python_project/blob/main/py_project/print_range.py'. The repository name is 'anisjkb / python_project'. The navigation bar includes links for Code, Issues, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings. The 'Code' tab is selected. On the left, there's a sidebar with a 'Files' section showing a tree view of files: 'main' (selected), 'py_project' (expanded), 'print_range.py' (selected), 'python' (closed), 'How to Learn Data Science & Ar...', and 'ekyc_data.csv'. The main content area displays the code for 'print_range.py':

```
1 r=range(1,199,2)
2 for i in r:
3     print ("Number=",i)
```

Update repository after modify file:

Suppose we add a new line "#Check git update commit" to the file "print_range.py"

Now you want to update the Git/Git Hub similerly



Step-1: After updating print_range.py file you can check the git by "git status" command. It will shows "modified" message

```
User@DESKTOP-VI2MRGR MINGW64 /e/Data Science (main)
$ git status
On branch main
Your branch is up to date with 'origin/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:   py_project/print_range.py

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

User@DESKTOP-VI2MRGR MINGW64 /e/Data Science (main)
```

2. git add .

3: git commit -m "first change"

```
User@DESKTOP-VI2MRGR MINGW64 /e/Data Science (main)
$ git add .

User@DESKTOP-VI2MRGR MINGW64 /e/Data Science (main)
$ git commit -m "first change"
[main 918262a] first change
 1 file changed, 2 insertions(+), 1 deletion(-)

User@DESKTOP-VI2MRGR MINGW64 /e/Data Science (main)
$ |
```

4. Input this command to push (upload the change to github)

"git push -u origin main"

```

User@DESKTOP-VI2MRGR MINGW64 /e/Data Science (main)
$ git add .

User@DESKTOP-VI2MRGR MINGW64 /e/Data Science (main)
$ git commit -m "second commit"
[main 9f10263] second commit
 1 file changed, 2 insertions(+), 2 deletions(-)

User@DESKTOP-VI2MRGR MINGW64 /e/Data Science (main)
$ git push -u origin main
Enumerating objects: 15, done.
Counting objects: 100% (15/15), done.
Delta compression using up to 8 threads
Compressing objects: 100% (9/9), done.
Writing objects: 100% (12/12), 951 bytes | 951.00 KiB/s, done.
Total 12 (delta 5), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (5/5), completed with 1 local object.
remote: This repository moved. Please use the new location:
remote: https://github.com/anisjkb/python_project.git
To https://github.com/anisjkb/python_project.git
 d87aada..9f10263 main -> main
branch 'main' set up to track 'origin/main'.

User@DESKTOP-VI2MRGR MINGW64 /e/Data Science (main)
$ |

```

The screenshot shows a GitHub repository page for 'anisjkb / python_project'. The repository has one branch, 'main', which was last updated via a commit from 'anisjkb' titled 'second commit'. The 'Code' tab is selected, displaying the contents of the 'print_range.py' file. The code prints numbers from 1 to 199 and then from 1 to 9.

```

python_project / py_project / print_range.py □

anisjkb second commit

Code Blame 8 lines (7 loc) · 131 Bytes Code 55% faster with GitHub Copilot

1   r=range(1,199,2)
2   for i in r:
3       print ("Number=",i)
4   # Check git update commit
5
6   p=range(1,10,9)
7   for i in p:
8       print ("Number=",i)

```

Get started by [creating a new file](#) or [uploading an existing file](#). We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

...or create a new repository on the command line

```
echo "# python_project" >> README.md
```

```
git init
```

```
git add README.md
```

```
git commit -m "first commit"
```

```
git branch -M main
```

```
git remote add origin https://github.com/anisjkb/python\_project.git
```

```
git push -u origin main
```

...or push an existing repository from the command line

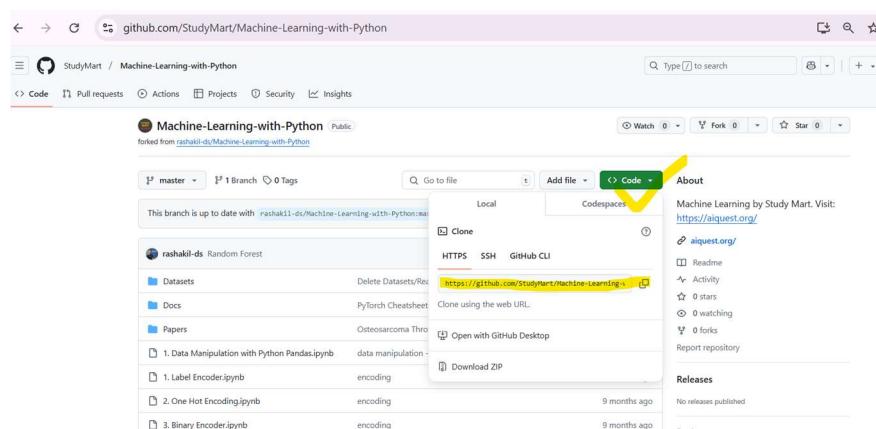
```
git remote add origin https://github.com/anisjkb/python\_project.git
```

```
git branch -M main
```

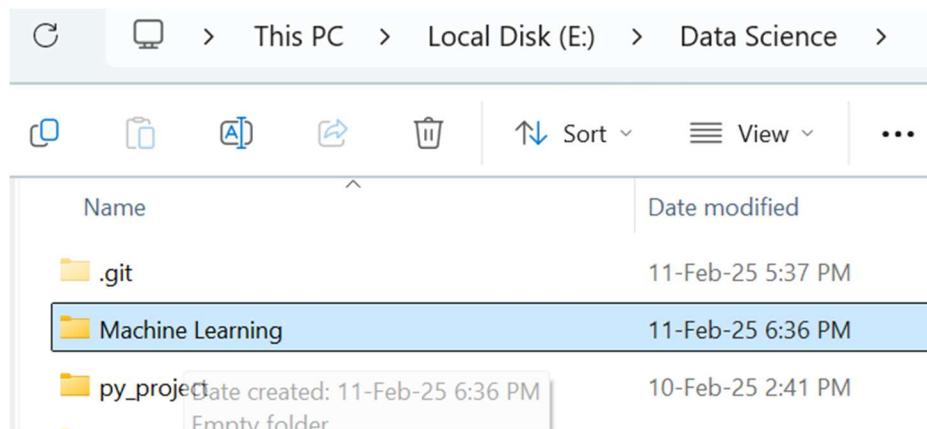
```
git push -u origin main
```

Download project from remote repository to local PC (Git Clone):

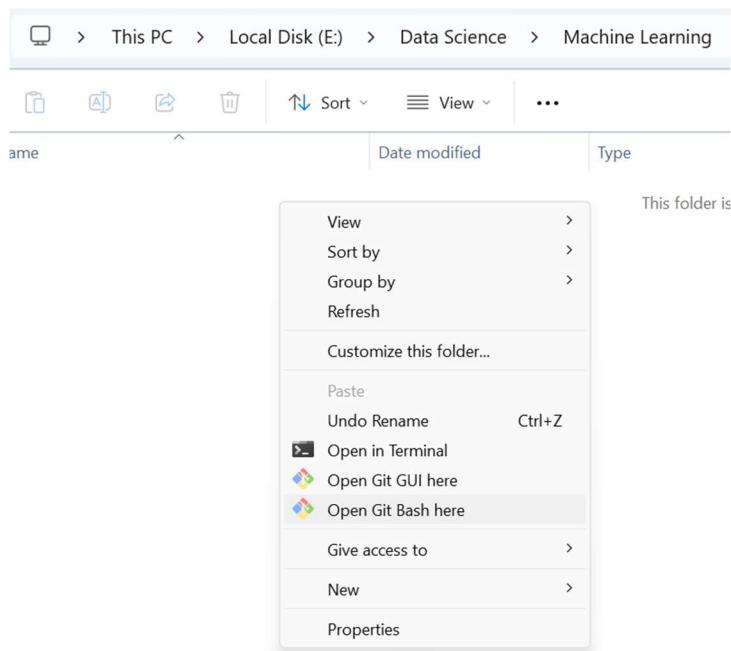
Step 1: Go to the desired repository as the following image. Copy the HTTPS URL

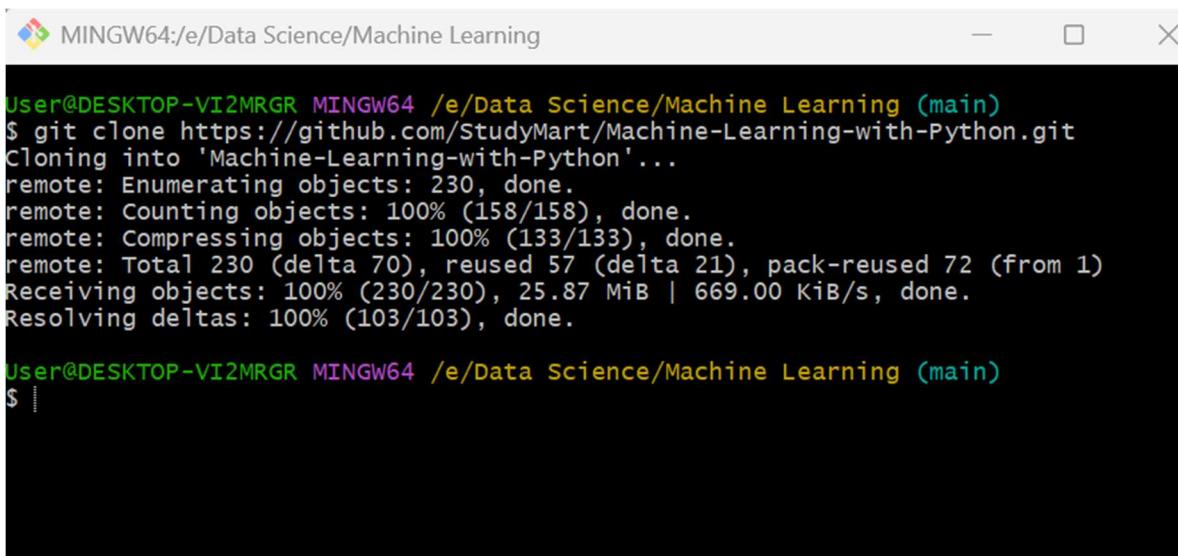


Create a folder in local drive



Enter the folder > right click and open "" Git Bash here"





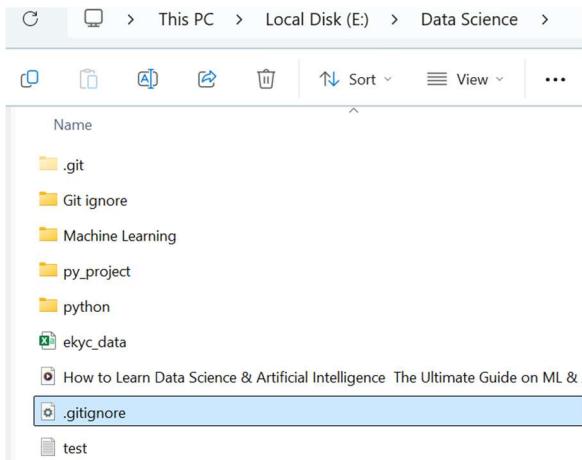
```
User@DESKTOP-VI2MRGR MINGW64 /e/Data Science/Machine Learning (main)
$ git clone https://github.com/StudyMart/Machine-Learning-with-Python.git
Cloning into 'Machine-Learning-with-Python'...
remote: Enumerating objects: 230, done.
remote: Counting objects: 100% (158/158), done.
remote: Compressing objects: 100% (133/133), done.
remote: Total 230 (delta 70), reused 57 (delta 21), pack-reused 72 (from 1)
Receiving objects: 100% (230/230), 25.87 MiB | 669.00 KiB/s, done.
Resolving deltas: 100% (103/103), done.

User@DESKTOP-VI2MRGR MINGW64 /e/Data Science/Machine Learning (main)
$ ..
```

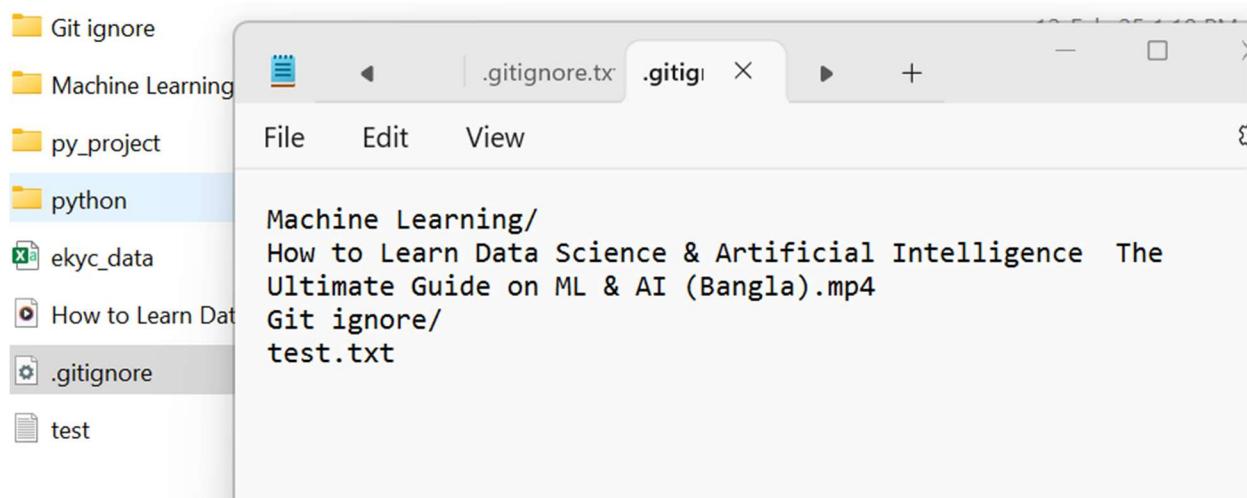
.gitignore

gitignore file is used in a git repository to ignore the files and directories that are unnecessary to the project this will be ignored by the git once the changes have been committed to the Remote repository.

Step-1: run the command “touch .gitignore” to create .gitignore file



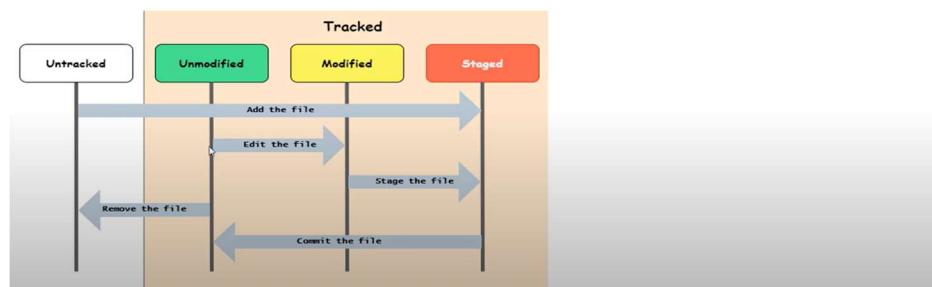
2. Open the file and add which are need to ignore as following way (Folder name must ended by /)



(You can create a file from (<https://www.toptal.com/developers/gitignore/>)

Git file status lifecycle:

The file status lifecycle in Git describes the different states a file can have (untracked, tracked, modified, staged, or committed) as it moves through the version control process.



i) Untracked:

No .git folder/file here; Initialized empty Git repository (by command git init)

MINGW64:/e/Data Science

```
User@DESKTOP-VI2MRGR MINGW64 /e/Data Science
$ git status
fatal: not a git repository (or any of the parent directories): .git
```

Initialized empty Git repository in E:/Data Science/.git/

MINGW64:/e/Data Science

```
User@DESKTOP-VI2MRGR MINGW64 /e/Data Science
$ git status
fatal: not a git repository (or any of the parent directories): .git

User@DESKTOP-VI2MRGR MINGW64 /e/Data Science
$ git init
Initialized empty Git repository in E:/Data Science/.git/

User@DESKTOP-VI2MRGR MINGW64 /e/Data Science (main)
$ |
```

```
MINGW64:/e/Data Science
$ git status
fatal: not a git repository (or any of the parent directories): .git

User@DESKTOP-VI2MRGR MINGW64 /e/Data Science
$ git init
Initialized empty Git repository in E:/Data Science/.git/

User@DESKTOP-VI2MRGR MINGW64 /e/Data Science (main)
$ git status
On branch main

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    .gitignore
    ekyc_data.csv
    py_project/
    python/

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

User@DESKTOP-VI2MRGR MINGW64 /e/Data Science (main)
$
```

ii) **Tracked:** execute command (git add .) to track

```
MINGW64:/e/Data Science
User@DESKTOP-VI2MRGR MINGW64 /e/Data Science (main)
$ git status
On branch main

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    .gitignore
    ekyc_data.csv
    py_project/
    python/

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

User@DESKTOP-VI2MRGR MINGW64 /e/Data Science (main)
$ git add .
warning: in the working copy of 'python/.ipynb_checkpoints/day1-checkpoint.ipynb',
  LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'python/day1.ipynb', LF will be replaced by CRLF
  the next time Git touches it

User@DESKTOP-VI2MRGR MINGW64 /e/Data Science (main)
$
```

```
MINGW64:/e/Data Science
', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'python/day1.ipynb', LF will be replaced by CRLF
the next time Git touches it

User@DESKTOP-VI2MRGR MINGW64 /e/Data Science (main)
$ git status
On branch main

No commits yet

Changes to be committed:
(use "git rm --cached <file>..." to unstage)
  new file: .gitignore
  new file: ekyc_data.csv
  new file: py_project/print_range.py
  new file: python/.day1.ipynb.layout
  new file: python/.ipynb_checkpoints/day1-checkpoint.ipynb
  new file: python/.virtual_documents/day1.ipynb
  new file: python/anaconda_projects/db/project_filebrowser.db
  new file: python/day1.ipynb

User@DESKTOP-VI2MRGR MINGW64 /e/Data Science (main)
$
```

Committed:

```
User@DESKTOP-VI2MRGR MINGW64 /e/Data Science (main)
$ git commit -m 'check git lifecycle'
[main (root-commit) f919142] check git lifecycle
8 files changed, 2399 insertions(+)
create mode 100644 .gitignore
create mode 100644 ekyc_data.csv
create mode 100644 py_project/print_range.py
create mode 100644 python/.day1.ipynb.layout
create mode 100644 python/.ipynb_checkpoints/day1-checkpoint.ipynb
create mode 100644 python/.virtual_documents/day1.ipynb
create mode 100644 python/anaconda_projects/db/project_filebrowser.db
create mode 100644 python/day1.ipynb

User@DESKTOP-VI2MRGR MINGW64 /e/Data Science (main)
$
```

Unmodified stage: When working tree shows clean

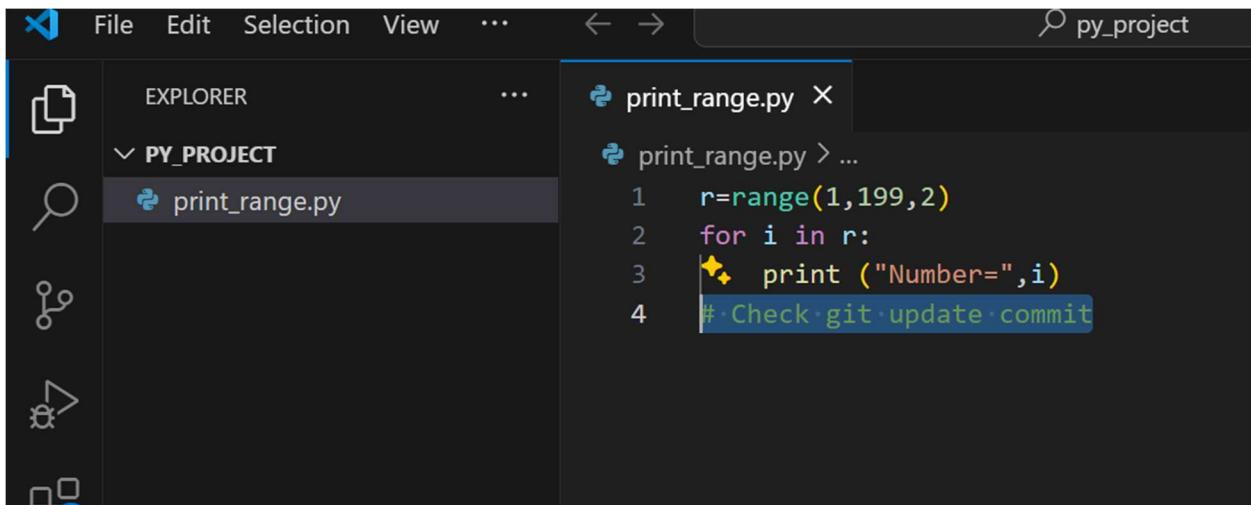
```
User@DESKTOP-VI2MRGR MINGW64 /e/Data Science (main)
$ git status
On branch main
nothing to commit, working tree clean

User@DESKTOP-VI2MRGR MINGW64 /e/Data Science (main)
$
```

Modified stage:

Suppose we add a new line "#Check git update commit" to the file "print_range.py"

Now you want to update the Git/Git Hub similerly



Step-1: After updating print_range.py file you can check the git by “git status” command. It will shows “modified” message

```
User@DESKTOP-VI2MRGR MINGW64 /e/Data Science (main)
$ git status
On branch main
Your branch is up to date with 'origin/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:   py_project/print_range.py

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

User@DESKTOP-VI2MRGR MINGW64 /e/Data Science (main)
```

2. git add .

3: git commit -m “first change”

```
User@DESKTOP-VI2MRGR MINGW64 /e/Data Science (main)
$ git add .

User@DESKTOP-VI2MRGR MINGW64 /e/Data Science (main)
$ git commit -m "first change"
[main 918262a] first change
 1 file changed, 2 insertions(+), 1 deletion(-)

User@DESKTOP-VI2MRGR MINGW64 /e/Data Science (main)
$
```

4. Input this command to push (upload the change to github)

“git push -u origin main”

How Git Diff Works

The **git diff** command shows the differences between two stages of a repository. It compares the changes in your working directory (local PC), staging area, or between commits.

Command- **git diff**

Git diff --staged

```
MINGW64:/e/Data Science
  </footer>
</body>
</html>
\ No newline at end of file

User@DESKTOP-VI2MRGR MINGW64 /e/Data Science (main)
$ git add .

User@DESKTOP-VI2MRGR MINGW64 /e/Data Science (main)
$ git diff

User@DESKTOP-VI2MRGR MINGW64 /e/Data Science (main)
$ git status
On branch main
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    modified:   py_project/home.html

User@DESKTOP-VI2MRGR MINGW64 /e/Data Science (main)
$ git diff --staged
diff --git a/py_project/home.html b/py_project/home.html
index ab70b3f..fa6e8f0 100644
--- a/py_project/home.html
+++ b/py_project/home.html
@@ -24,15 +24,15 @@
      </section>
      <section>
        <h2>Middle Section</h2>
-       <p>Check regularly.</p>
+       <p>Check regularly modification check</p>
      </section>
      <section>
        <h2>Latest News check modification</h2>
-       <p>Stay tuned for the latest updates and news.</p>
+       <p>Latest updates and news-- added line</p>
      </section>
    </main>
    <footer>
-      <p>&copy; 2023 My Website</p>
+      <p>&copy; 2025 My Website</p>
    </footer>
  </body>
</html>
\ No newline at end of file

User@DESKTOP-VI2MRGR MINGW64 /e/Data Science (main)
$ |
```

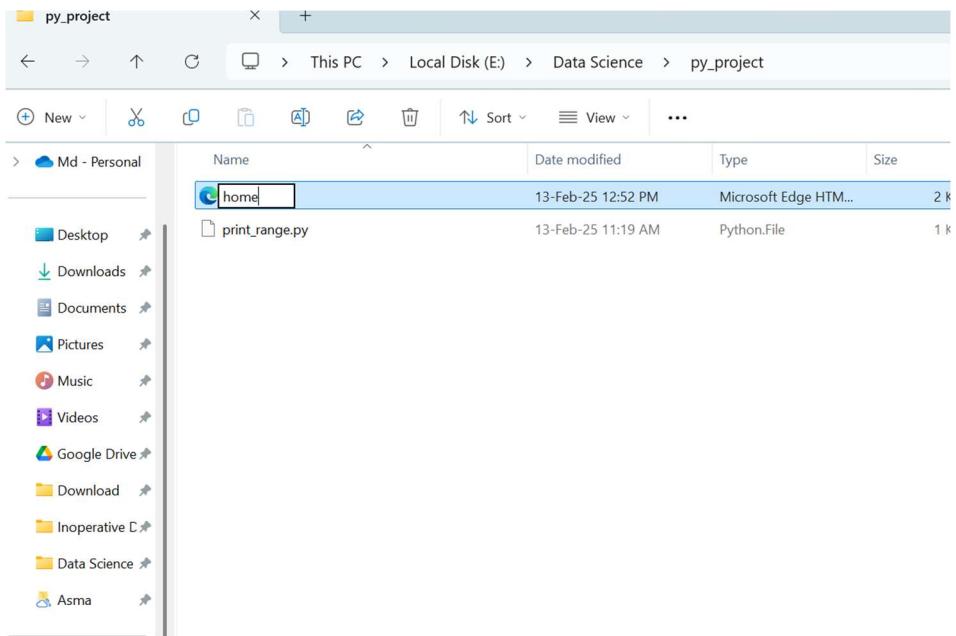
Git - Rename & Move File

Rename a file:

Way-1:

In Git, rename means changing a file's name.

- i) Manually rename a file (Example- from 'home' to 'index')



ii. It shows home.html file is deleted

```
MINGW64:/e/Data Science
On branch main
nothing to commit, working tree clean

User@DESKTOP-VI2MRGR MINGW64 /e/Data Science (main)
$ git status
On branch main
Changes not staged for commit:
  (use "git add/rm <file>..." to update what will be committed)
    (use "git restore <file>..." to discard changes in working directory)
      deleted:   py_project/home.html

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    py_project/index.html

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

User@DESKTOP-VI2MRGR MINGW64 /e/Data Science (main)
$ |
```

A screenshot of a terminal window titled 'MINGW64:/e/Data Science'. The window displays the output of the 'git status' command. The output shows that the file 'py_project/home.html' has been deleted ('deleted'). There are also untracked files listed, specifically 'py_project/index.html'. The terminal window is overlaid on a File Explorer interface, which shows a folder structure with 'index' and 'print_range.py' files.

iii. To resolve this send the file at staging area by command “git add .”

```
User@DESKTOP-VI2MRGR MINGW64 /e/Data Science (main)
$ git add .

User@DESKTOP-VI2MRGR MINGW64 /e/Data Science (main)
$ git status
On branch main
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    renamed:  py_project/home.html -> py_project/index.html

User@DESKTOP-VI2MRGR MINGW64 /e/Data Science (main)
$
```

It shows renamed complete (renamed: py_project/home.html -> py_project/index.html)

iv. Commit the changes by command (git commit -m 'rename done')

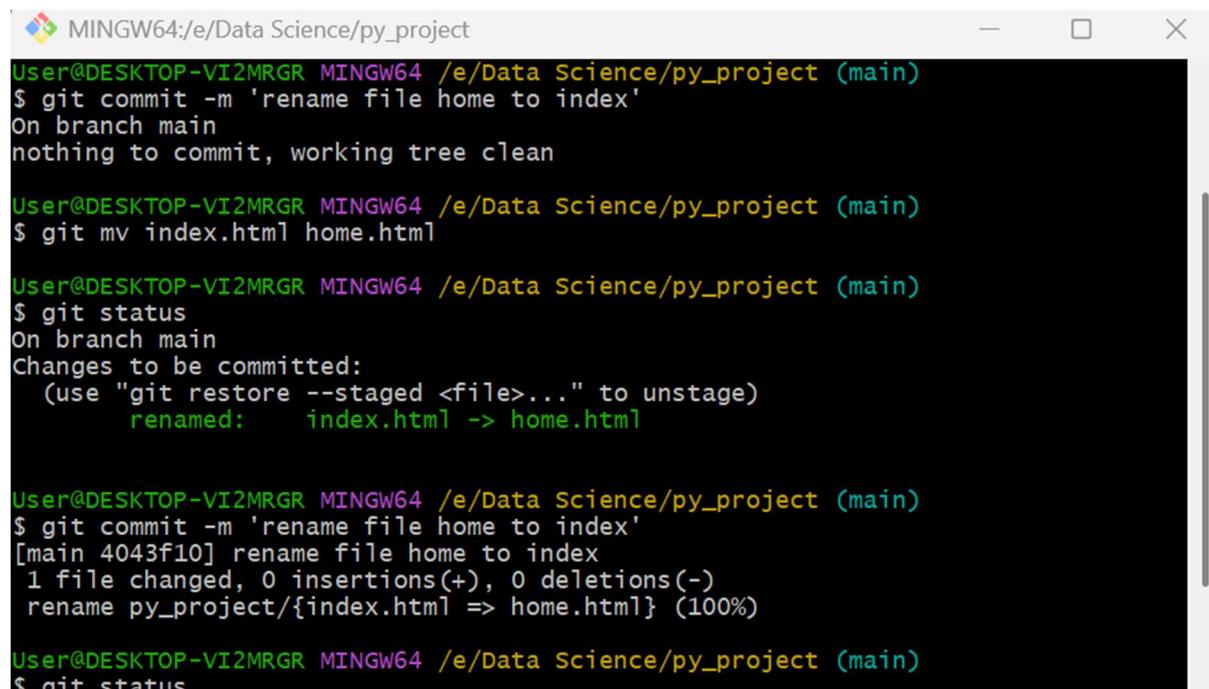
Way-2:

It will be done by git command “\$ git mv index.html home.html” here “index.html “ this file name change to “home.html”

i) \$ git mv index.html home.html or (\$ git mv py_project/home.html py_project/index.html)—here the file have in “py_project” folder.

ii commit the change by command

“git commit -m 'rename file home to index'”



```
MINGW64:/e/Data Science/py_project
User@DESKTOP-VI2MRGR MINGW64 /e/Data Science/py_project (main)
$ git commit -m 'rename file home to index'
On branch main
nothing to commit, working tree clean

User@DESKTOP-VI2MRGR MINGW64 /e/Data Science/py_project (main)
$ git mv index.html home.html

User@DESKTOP-VI2MRGR MINGW64 /e/Data Science/py_project (main)
$ git status
On branch main
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    renamed:  index.html -> home.html

User@DESKTOP-VI2MRGR MINGW64 /e/Data Science/py_project (main)
$ git commit -m 'rename file home to index'
[main 4043f10] rename file home to index
 1 file changed, 0 insertions(+), 0 deletions(-)
 rename py_project/{index.html => home.html} (100%)

User@DESKTOP-VI2MRGR MINGW64 /e/Data Science/py_project (main)
$ git status
```

Delete a file:

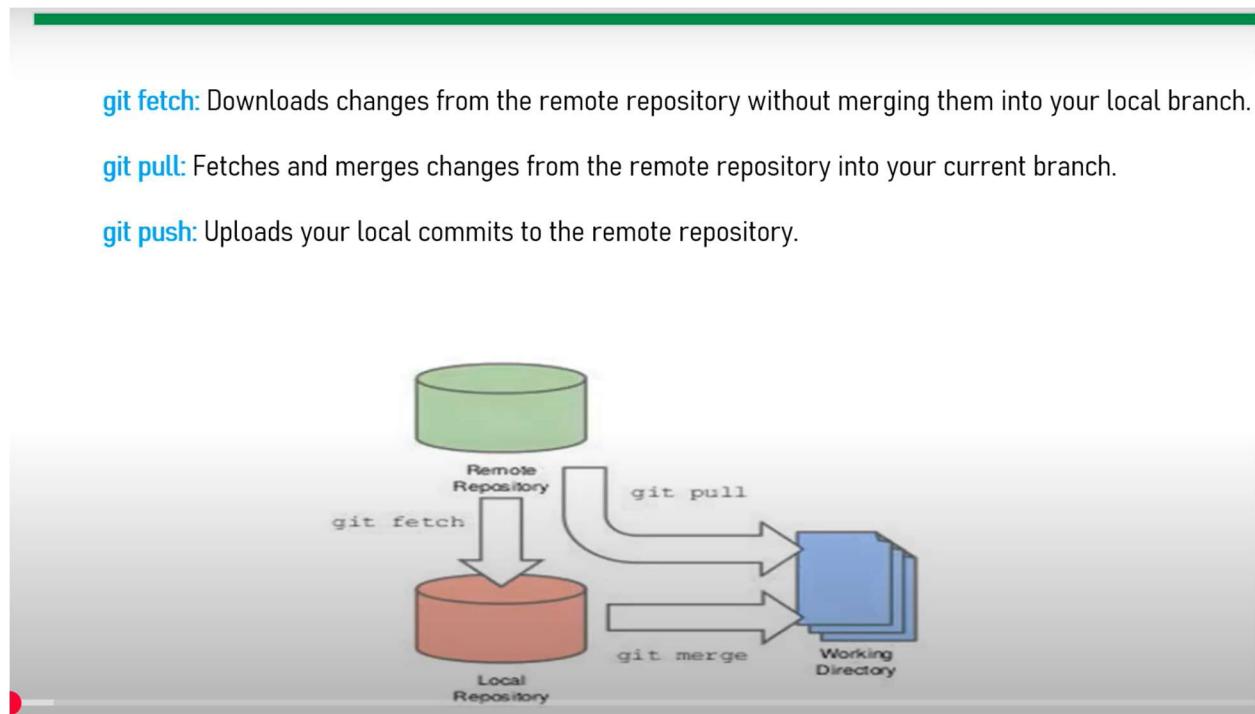
We can manually delete a file or by using git command we can do this

Command:

- (i) `git rm test.thml` (delete from pc/repository)
- (ii) `git rm -cached test.thml` (delete from cached)

Press: enter

Git fetch, pull, push



fetch (changes/updates any content/line)

If anyone changes/updates any content/line of a file at **the remote repository**, then to update the change at the **local PC/repository**, run the two commands serially.

- i) `git fetch`
- ii) `git merge`

It will update local file.

Pull

If anyone **deletes any content** or line from a file at **the remote repository**, then to update the change, to the local PC/repository run the command.

`git pull`

The screenshot shows a terminal window with a code editor and a command-line interface. On the left, a file tree for 'DJANGO_PROJECT' is visible, showing 'Code', 'aquest', 'course', 'templates', and 'ml.html'. The 'ml.html' file is open in the editor, displaying the following content:

```
<h1>Welcome to Machine Learning</h1>
<h2>Welcome to GitHub</h2>
<h3>Welcome to Django</h3>
<h4>Data Analysis</h4>
```

To the right of the editor is a terminal window with the following command history:

```
nothing to commit, working tree clean
$ git fetch
$ git fetch
remote: Enumerating objects: 11, done.
remote: Counting objects: 100% (11/11), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 6 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
Unpacking objects: 100% (6/6), 1.08 KiB | 39.00 KiB/s, done.
From https://github.com/nomanbasar/django_project
    784114e..3afbec3  main      -> origin/main
$ git merge
Updating 784114e..3afbec3
Fast-forward
  Code/aquest/templates/ml.html | 3 +--
  1 file changed, 2 insertions(+), 1 deletion(-)
$
```

push

If anyone **changes/updates any content/line** of a file at **the local repository**, then to update the change to the **remote repository**, follow the following steps:

- i) `git add .`
- ii) `git commit -m 'push check'`
- iii) `git push`

It will update local file.

```
MINGW64:/d/Study Mart/django_project
(use "git restore <file>..." to discard changes in working directory)
  modified:   Code/aquest/templates/m1.html

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

HP@DESKTOP-8H25T2K MINGW64 /d/Study Mart/django_project (main)
$ git add .

HP@DESKTOP-8H25T2K MINGW64 /d/Study Mart/django_project (main)
$ git commit -m 'push'
[main 60f61c7] push
  1 file changed, 1 insertion(+)

HP@DESKTOP-8H25T2K MINGW64 /d/Study Mart/django_project (main)
$ git status
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean

HP@DESKTOP-8H25T2K MINGW64 /d/Study Mart/django_project (main)
$ git push
```

Git - Branch & Merge

Branch: A separate line of development in Git.

Merge: Combining changes from one branch into another.

Create Branch and Merge

Create and switch to a new branch: git checkout -b branch_name

List all local branches: git branch

Switch to an existing branch: git checkout branch_name

How to reset your Git Settings

```
git config --global --unset user.name  
git config --global --unset user.email  
git config --global --unset credential.helper  
cmdkey /delete:LegacyGeneric:target=git:https://github.com  
git config --global user.name "username"  
git config --global user.email emailaddr  
git config credential.helper 'store'  
git remote add origin <repolink>  
git remote -v  
git init  
git add . (Stage)  
git commit -m "message" (Commit)  
git push origin master (Push)
```

1, How to Delete Repository in Git

3 -- path of repository

git rev-parse -- show-toplevel

6 Git root directory name

basename 'git rev-parse -- show-toplevel'

[Create Local Repository and push to github/Push an existing repository to github](#)



- i) git init
- ii) \$ git add .
- iii) \$ git commit -m 'first commit'
- iv) \$ git remote add origin https://github.com/anisjkb/learn_python.git
- v) \$ git push origin main

After executing “\$ git push origin main” command if faced following error

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

Then run the command **to solve the error:** \$ git pull --rebase origin main

```
User@DESKTOP-VI2MRGR MINGW64 /e/Data Science/Programing Language/learn_python
$ git init
Initialized empty Git repository in E:/Data Science/Programing Language/learn_python/.git/

User@DESKTOP-VI2MRGR MINGW64 /e/Data Science/Programing Language/learn_python
(main)
$ git add .

User@DESKTOP-VI2MRGR MINGW64 /e/Data Science/Programing Language/learn_python
(main)
$ git commit -m 'first commit'
[main (root-commit) 95ddaad] first commit
6 files changed, 144 insertions(+)
create mode 100644 Student details.py
create mode 100644 abiatNushafa.py
create mode 100644 home.html
create mode 100644 index.html
create mode 100644 print_range.py
create mode 100644 test.py

User@DESKTOP-VI2MRGR MINGW64 /e/Data Science/Programing Language/learn_python
(main)
$ git remote add origin https://github.com/anisjkb/learn_python.git

User@DESKTOP-VI2MRGR MINGW64 /e/Data Science/Programing Language/learn_python
(main)
$ git push
fatal: The current branch main has no upstream branch.
To push the current branch and set the remote as upstream, use
```

```
User@DESKTOP-VI2MRGR MINGW64 /e/Data Science/Programing Language/learn_python  
(main)  
$ git push origin main  
To https://github.com/anisjkb/learn_python.git  
! [rejected]      main -> main (fetch first)  
error: failed to push some refs to 'https://github.com/anisjkb/learn_python.git'  
hint: Updates were rejected because the remote contains work that you do not  
hint: have locally. This is usually caused by another repository pushing to  
hint: the same ref. If you want to integrate the remote changes, use  
hint: 'git pull' before pushing again.  
hint: See the 'Note about fast-forwards' in 'git push --help' for details.
```

```
User@DESKTOP-VI2MRGR MINGW64 /e/Data Science/Programing Language/learn_python  
(main)  
$ git pull --rebase origin main  
remote: Enumerating objects: 3, done.  
remote: Counting objects: 100% (3/3), done.  
remote: Compressing objects: 100% (2/2), done.  
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)  
Unpacking objects: 100% (3/3), 913 bytes | 57.00 KiB/s, done.  
From https://github.com/anisjkb/learn_python  
* branch      main    -> FETCH_HEAD  
* [new branch] main    -> origin/main  
Successfully rebased and updated refs/heads/main.
```

```
User@DESKTOP-VI2MRGR MINGW64 /e/Data Science/Programing Language/learn_python  
(main)  
$ git push -u origin main  
Enumerating objects: 8, done.  
Counting objects: 100% (8/8), done.  
Delta compression using up to 8 threads  
Compressing objects: 100% (6/6), done.  
Writing objects: 100% (7/7), 1.59 KiB | 1.59 MiB/s, done.  
Total 7 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)  
To https://github.com/anisjkb/learn_python.git  
bc74788..1a44bca main -> main  
branch 'main' set up to track 'origin/main'.
```

```
User@DESKTOP-VI2MRGR MINGW64 /e/Data Science/Programing Language/learn_python  
(main)  
$
```