

## Open the terminal in VS Code

Run the following command:

```
git clone https://github.com/mrinmoycyber/Learning.git
```

Navigate to the cloned directory:

```
cd Learning
```

Open the repository in VS Code:

```
Code .
```

---

### Prerequisites

- Ensure Git is installed on your system.
- To check, run `git --version` in the terminal.
- Make sure you have set up Git in VS Code:

Configure your username:

```
bash
```

```
git config --global user.name "Your Name"
```

Configure your email:

```
bash
```

```
git config --global user.email "youremail@example.com"
```

## Git commands -

### git init

- Initializes a new Git repository in your project directory. It creates a `.git` folder where Git keeps its configuration and data.
- Usage:  
`git init`

### git status

- Displays the current state of the working directory and staging area. It tells you which files are modified, staged, or untracked.  
`git status`

## Create a New Branch

- Open the terminal in VS Code.
- Use the following command to create a new branch (replace `new-branch-name` with your preferred branch name):  
`git checkout -b new-branch-name`

## Make Changes in the New Branch

- Make the changes you want in the code (e.g., in the `InitialSet.ipynb` file).
- After making the changes, stage them by clicking the "+" icon next to each file in the **Source Control** panel or by running:  
`git add .`  
`git add <file-name> # Adds a single file`

## Commit the Changes

- Write a commit message for your changes in the **Source Control** panel or use the terminal:  
`git commit -m "Describe your changes here"`

## Switch to the Main Branch

- After committing, switch back to the main branch:  
`git checkout main`

## Merge the New Branch into Main

- Merge your changes from the new branch into the main branch:  
`git merge new-branch-name`

## Push the Changes to the Remote Repository

- Finally, push your changes to the remote main branch:  
`git push origin main`

## git pull

- Fetches changes from the remote repository and merges them into your local branch.  
`git pull origin main # Pulls changes from the main branch`

## git stash

- Temporarily stash changes that you don't want to commit yet. It's useful when you need to switch branches but have uncommitted changes.  
`git stash # Stashes changes`  
`git stash pop # Restores the most recent stashed changes`