

Assignment 1:

Initialize a new Git repository in a directory of your choice. Add a simple text file to the repository and make the first commit.

1. Open Command Prompt : Open command prompt on your computer.

2. Navigate to the Directory:

Use the ``cd`` command to navigate to the directory where you want to initialize the Git repository.

For example:

```
cd path/to/your/directory
```

3. Initialize Git:

Use the ``git init`` command to initialize a new Git repository in the chosen directory

```
git init
```

4. Create a Simple Text File:

You can create a new text file using a text editor like Notepad (Windows), TextEdit (Mac), or any code editor of your choice.

For example,

create a file named ``sample.txt`` and add some content to it.

5. Add the Text File to the Repository:

Use the ``git add`` command to add the text file to the staging area:

```
git add sample.txt
```

6. Make the First Commit:

Use the `git commit` command to commit the changes to the repository:

```
git commit -m "Initial commit: Added sample.txt"
```

Replace `"Initial commit: Added sample.txt"` with your own commit message.

```

PS D:\Wipro Tranning\Phase 2 Traning\Day 5\demo> git init
Initialized empty Git repository in D:/Wipro Tranning/Phase 2 Traning/Day 5/demo/.git/
PS D:\Wipro Tranning\Phase 2 Traning\Day 5\demo> ls

Directory: D:\Wipro Tranning\Phase 2 Traning\Day 5\demo

Mode                LastWriteTime         Length Name
----                -
-a-----         08-05-2024    20:50             0 sample.txt

PS D:\Wipro Tranning\Phase 2 Traning\Day 5\demo> git add ..
fatal: ../: '..' is outside repository at 'D:/Wipro Tranning/Phase 2 Traning/Day 5/demo'
PS D:\Wipro Tranning\Phase 2 Traning\Day 5\demo> git add .
PS D:\Wipro Tranning\Phase 2 Traning\Day 5\demo> git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   sample.txt

PS D:\Wipro Tranning\Phase 2 Traning\Day 5\demo> git commit -m "first commit"
[master (root-commit) cb16ae3] first commit
1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 sample.txt
PS D:\Wipro Tranning\Phase 2 Traning\Day 5\demo>

```

Assignment 2:

Branch Creation and Switching

Create a new branch named 'feature' and switch to it. Make changes in the 'feature' branch and commit them.

1. Check Current Branch(Optional):

Before creating a new branch, you can check which branch you are currently on using the command:

```
git branch
```

2. Create a New Branch:

Use the `git branch` command followed by the branch name (`feature` in this case) to create a new branch:

```
git branch feature
```

3. Switch to the New Branch:

Use the `git checkout` command to switch to the newly created `feature` branch:

```
git checkout feature
```

Alternatively, you can combine branch creation and switching in one step using:

```
git checkout -b feature
```

4. Make Changes in the 'feature' Branch:

Edit the text file (`sample.txt`) or make any changes you want in the `feature` branch.

5. Add and Commit Changes:

Add the modified file to the staging area and commit the changes:

```
git add sample.txt
```

```
git commit -m "Made changes in the feature branch"
```

Replace ``"Made changes in the feature branch"`` with your own commit message.

```

PS D:\Wipro Tranning\Phase 2 Traning\Day 5\demo> git branch
* main
PS D:\Wipro Tranning\Phase 2 Traning\Day 5\demo> git branch feature
PS D:\Wipro Tranning\Phase 2 Traning\Day 5\demo> git checkout feature
Switched to branch 'feature'
PS D:\Wipro Tranning\Phase 2 Traning\Day 5\demo> ls

4. Make Changes in the 'feature' Branch:
Edit the text file (`sample.txt`) or make any changes you want in the `feature`
branch.

5. Add and Commit Changes:
Add the modified file to the staging area and commit the changes:

git add sample.txt
git commit -m "Made changes in the feature branch"

PS D:\Wipro Tranning\Phase 2 Traning\Day 5\demo> git add sample.txt
PS D:\Wipro Tranning\Phase 2 Traning\Day 5\demo> git commit -m "Made changes in the feature branch"
On branch feature
nothing to commit, working tree clean
PS D:\Wipro Tranning\Phase 2 Traning\Day 5\demo> |
replace "Made changes in the feature branch" with your own commit
message.

```

Assignment 3:

Feature Branches and Hotfixes

Create a 'hotfix' branch to fix an issue in the main code. Merge the 'hotfix' branch into 'main' ensuring that the issue is resolved.

1. Create a Hotfix Branch:

Use the `git checkout -b` command to create and switch to a new branch named `hotfix`:

```
git checkout -b hotfix
```

2. Fix the Issue:

Make the necessary changes to fix the issue in your codebase. For example, edit the `sample.txt` file to resolve the issue.

3. Add and Commit the Changes:

Add the modified file to the staging area and commit the changes in the `hotfix` branch:

```
git add sample.txt
```

```
git commit -m "Fixed the issue in the hotfix branch"
```

Replace `"Fixed the issue in the hotfix branch"` with your own commit message.

4. Switch to the Main Branch:

Switch back to the `main` branch using the `git checkout` command:

```
git checkout main
```

5. Merge the Hotfix Branch into Main:

Merge the changes from the `hotfix` branch into the `main` branch using the `git merge` command:

```
git merge hotfix
```

6. Resolve Merge Conflicts (if any):

If there are any merge conflicts, Git will prompt you to resolve them. Open the conflicted files in a text editor, resolve the conflicts, save the changes, and then add and commit the resolved files.

7. Push the Changes (if needed):

If you're working with a remote repository, use the `git push` command to push the merged changes to the remote repository:

```
git push origin main
```

Assignment of Day 6

```
PS D:\Wipro Tranning\Phase 2 Traning\Day 5\demo> git status
On branch feature
nothing to commit, working tree clean
PS D:\Wipro Tranning\Phase 2 Traning\Day 5\demo> git checkout -b hotfixes
Switched to a new branch 'hotfixes'
PS D:\Wipro Tranning\Phase 2 Traning\Day 5\demo> git branch
* hotfixes
  main
  master
PS D:\Wipro Tranning\Phase 2 Traning\Day 5\demo> git add sample.txt
PS D:\Wipro Tranning\Phase 2 Traning\Day 5\demo> git commit -m "Fixed the issue in the hotfix branch"
On branch hotfixes
nothing to commit, working tree clean
PS D:\Wipro Tranning\Phase 2 Traning\Day 5\demo> git checkout master
Switched to branch 'master'
PS D:\Wipro Tranning\Phase 2 Traning\Day 5\demo> git merge hotfixes
Already up to date.
```