

Git Basic Commands:

1.Set Your Global Identity (only once per system)

```
git config --global user.name "Your Name"  
git config --global user.email "your.email@example.com"
```

```
rit@rit:~$ git config --global user.name "Your Name"  
git config --global user.email "your.email@example.com"
```

2. Initialize a Repository in Your Project Folder

If your project is in ~/ping, then go inside it:

```
cd ~/ping  
git init
```

This creates a .git folder inside your project.

```
rit@rit:~$cd git
```

```
rit@rit:~$git:-git init
```

hint: Using 'master' as the name for the initial branch. This default branch name

hint: is subject to change. To configure the initial branch name to use in all

hint: of your new repositories, which will suppress this warning, call:

hint:

3.Check Files

You can verify what files exist:

```
ls
```

Example output:

```
ping.py  Dockerfile
```

4.Add Files to the Staging Area

You tried:

```
git add filename
```

```
rit@rit:~$ git add filename
```

but Git said:

```
fatal: pathspec 'filename' did not match any files
```

That’s because there’s **no file literally named “filename”** — you must specify real filenames.

So run:

```
git add ping.py Dockerfile
```

Or simply:

```
git add .
```

(. adds all files in the current directory)

5. Commit Your Changes

Now save the current state with a message:

```
git commit -m "Added ping.py and Dockerfile"
```

6. (Optional) Rename Default Branch

Git by default uses “master,” but many projects now use “main.”

You can rename:

```
git branch -M main
```

7. Create a New Branch (for example, a feature)

```
git checkout -b new-feature
```

Now you’re working on the “new-feature” branch.

Make some changes, then:

```
git add .  
git commit -m "Added feature X"
```

8. Merge Back into Main

Switch to main branch:

```
git checkout main
```

Then merge your feature branch:

```
git merge new-feature
```

9 .Connect to GitHub (Remote Repository)

If you have a GitHub repo, link it:

```
git remote add origin https://github.com/your-username/your-repo.git
```

```
git remote add origin https://github.com/username/repo.git
```

10. Push Code to GitHub

Push your main branch:

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

If you renamed your branch, push accordingly:

```
git push -u origin new-feature
```

11. Pull New Updates (from GitHub to Local)

If someone else updated the repository or you want to sync:

```
git pull
```

12.TO check Git Status

```
rit@rit:~$ git status
```

On branch master

No commits yet

Untracked files:

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

.bash