

Shree

Git is a version control system. A version control system is a tool that helps to track the changes made in the code.

Why is git popular:

- Free and open source
- Fast and reliable
- Scalable

Git helps us to:

1. Track the changes and we can save the data.
2. Helps us to collaborate with our peers.

Github is a website which allows us to store and manage code using git.

github.com

We upload our projects in folders.

In git terms a folder is called a repository.

Changes in git are called commits.

Initial commit means the first change in the project (usually it should be a readme file).

Adding a repository / project on git is a 2 step process:

1. Add
2. Commit

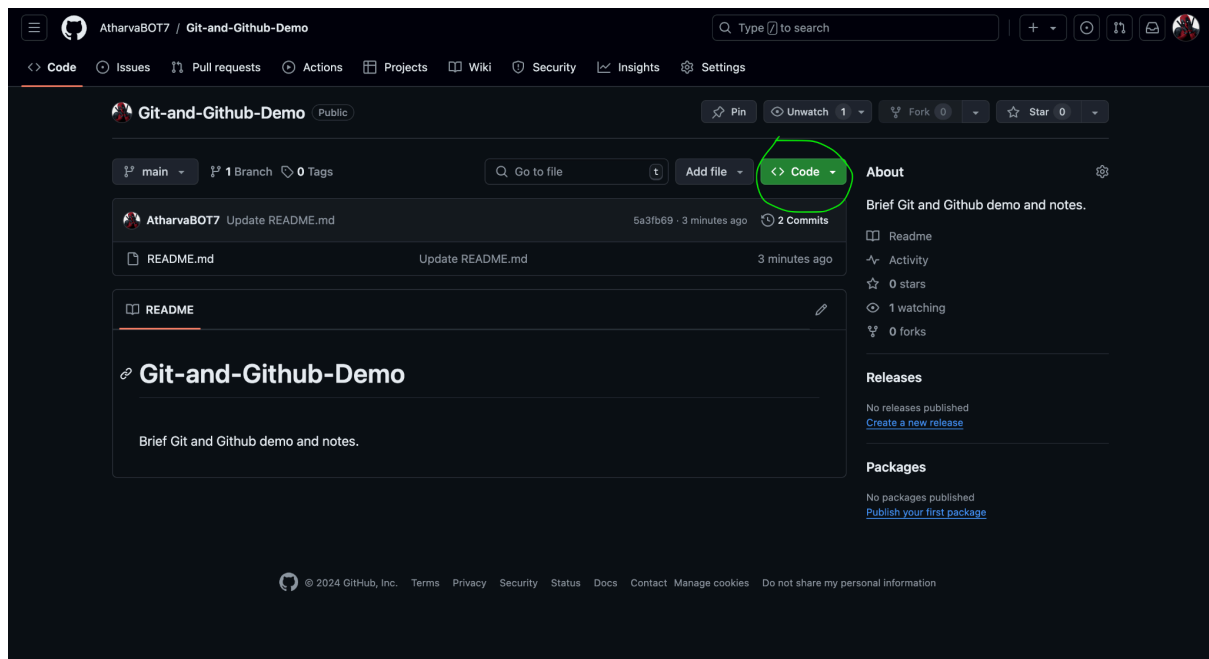
Clone Command

As there are many repositories on github, we can clone any of the repositories from the global level to our local machine.

Clone command helps us to clone a repository on git.

Syntax : `git clone <- link ->`

For the first time, clone your repository, you can find the link after clicking the repository which you created for the first time.



After cloning the repository, use the `cd` command in terminal or use visual studio code terminal.

Change the directory to the new file.

Syntax : `cd - file name ->`

Use the `ls` command to list down all the files present in the present directory.

Use the `ls -a` command to access all the files in the directory including the hidden files.

You will find a **.git** folder in each and every directory which the git is tracking.

Status Command

Status displays the status of the code.

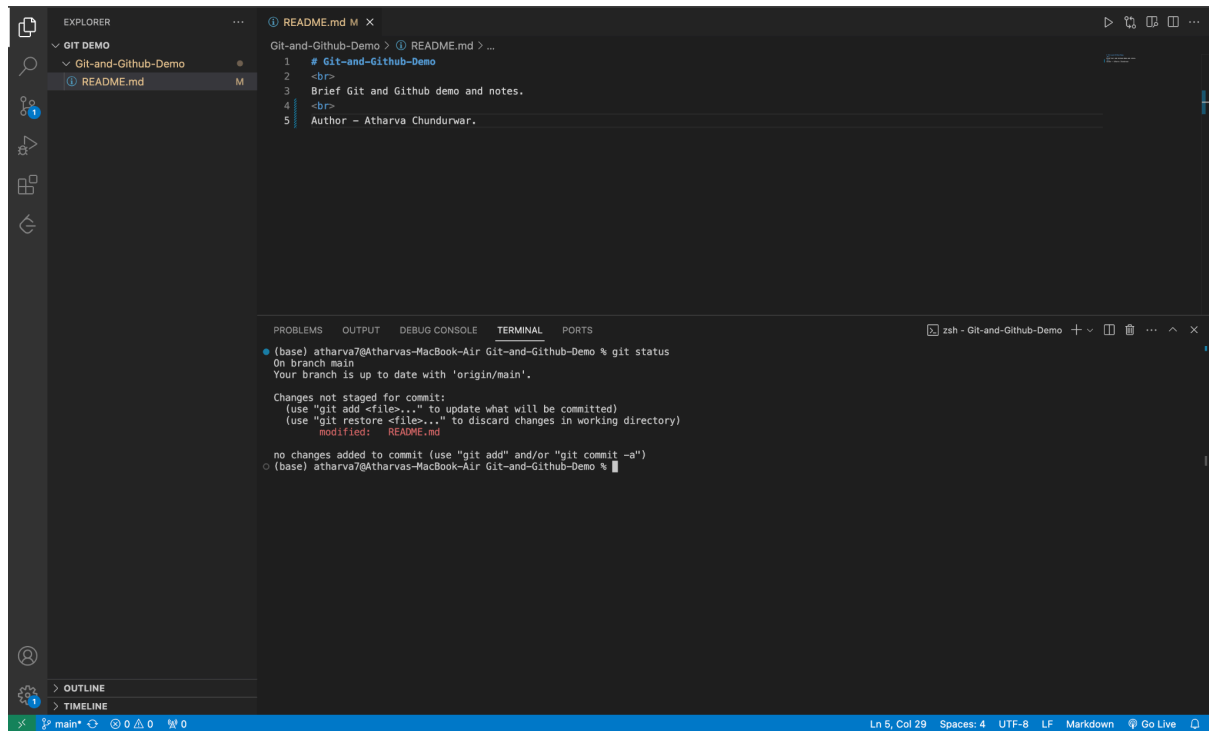
Syntax : `git status`

There are 4 types of status in git:

1. Untracked : new files which are not tracked by git yet.
2. Modified : updated / modified files with changes not yet committed.
3. Staged : File ready to be committed.
4. Unmodified : unchanged.

Open VS code and then open the git folder in vs code.

Modify the `readme.md` file which we had created in github while creating a new repository.



The screenshot shows the Visual Studio Code interface. The Explorer sidebar on the left shows a project named 'Git-and-Github-Demo' with a file 'README.md' marked as modified (M). The main editor area displays the content of 'README.md':

```
1 # Git-and-Github-Demo
2 <br>
3 Brief Git and Github demo and notes.
4 <br>
5 Author - Atharva Chundurwar.
```

Below the editor is a panel with tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL, and PORTS. The TERMINAL tab is active, showing the output of a 'git status' command:

```
(base) atharva7@Atharvas-MacBook-Air Git-and-Github-Demo % 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:   README.md

no changes added to commit (use "git add" and/or "git commit -a")
(base) atharva7@Atharvas-MacBook-Air Git-and-Github-Demo %
```

The status bar at the bottom indicates the current file is 'main', with 0 changes and 0 files. It also shows the cursor is at line 5, column 29, with 4 spaces, in UTF-8 encoding, using LF line endings, in a Markdown file, and that the Go Live extension is installed.

Add Command

You can add new or changed files in your working directory to the git staging area.

Syntax : `git add <- file name ->`

(`git add .`) is the command used to add all the command to the staging area.

Commit Command

You use the commit command to save the changes. It is the record of the changes made to the project.

Syntax : `git commit -m "some message"`

```
Git-and-Github-Demo > README.md > ...
1 # Git-and-Github-Demo
2 <br>
3 Brief Git and Github demo and notes.
4 <br>
5 Author - Atharva Chundurwar.

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
zsh - Git-and-Github-Demo + - - - - - X

• (base) atharva7@Atharvas-MacBook-Air Git Demo % git status
fatal: not a git repository (or any of the parent directories): .git
• (base) atharva7@Atharvas-MacBook-Air Git Demo % cd Git-and-Github-Demo
• (base) atharva7@Atharvas-MacBook-Air Git-and-Github-Demo % git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   README.md

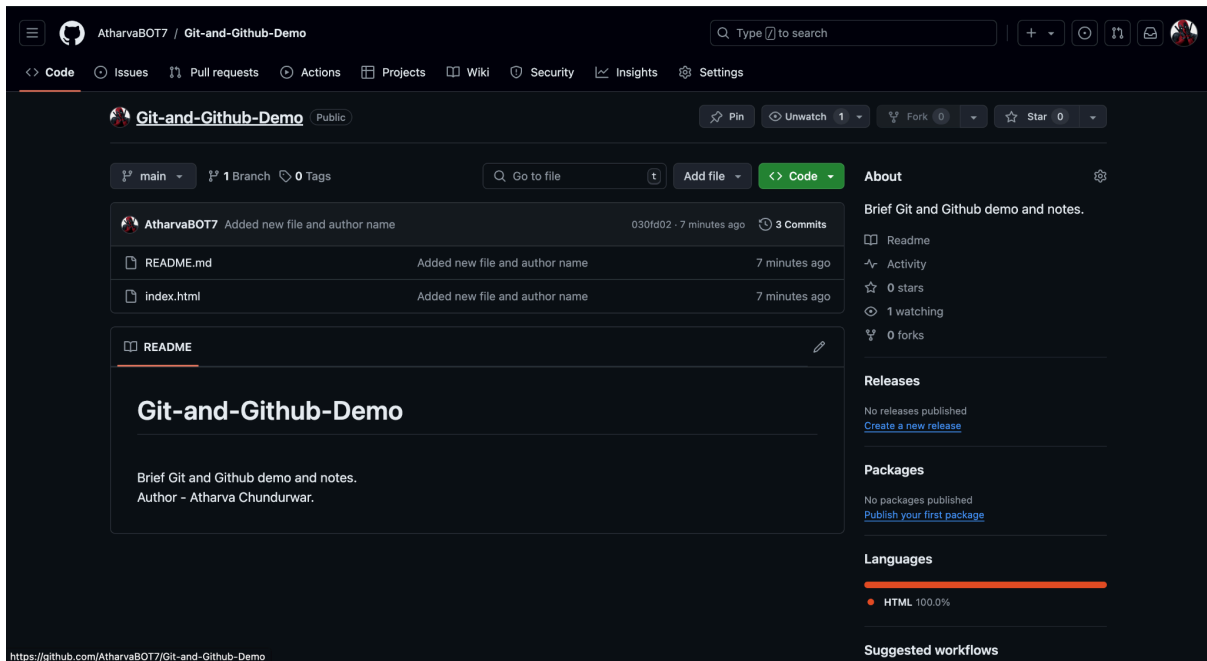
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        index.html
• (base) atharva7@Atharvas-MacBook-Air Git-and-Github-Demo % git add .
• (base) atharva7@Atharvas-MacBook-Air Git-and-Github-Demo % git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   README.md
        new file:   index.html
• (base) atharva7@Atharvas-MacBook-Air Git-and-Github-Demo % git commit -m "Added new file and author name"
```

Even after the git commit command the changes have not been pushed to our github account, this is because we have committed the changes to our local machine.

Push Command

The git push command is used to upload a local repo content to the remote repo. In simple terms, it is used to upload our changes to the github website on our account. Syntax : git push origin main



Init Commands

The init command is used to create a new repository on github.

git init command

git remote add origin <- link -> : adds a remote branch named origin.

git remote -v : used to verify the remote

git branch : used to check the branch

git branch -M main : used to rename the branch

git push origin main

General Git workflow:

- 1) Create a new github repository.
- 2) Clone the repository
- 3) Make changes to the repository
- 4) Add the changes to the staging area
- 5) Commit the changes for final push
- 6) Push the changes on the remote area.

Branch commands:

- 1) **git branch** : to check the current branch on which you are working
- 2) **git branch -M main** : change the branch name to main.
- 3) **git checkout <- branch name ->** : used to navigate.
- 4) **git checkout -b <- new branch name ->** : used to create a new branch
- 5) **git branch -d <- branch name ->** : to delete a branch (you can't delete the branch in which you are currently working in)

Merging Branches:

Way 1:

- 1) **git diff <- branch name ->** : used to compare commits, branches, files and more.
- 2) **git merge <- branch name ->** : used to merge 2 branches

Way 2:

Create a PR (Pull request), a pull request lets you tell others about the changes that you have pushed on the branch in a repository on github.

Using the 2nd method, you can directly merge the PR with the main branch on github.

The changes in code will not reflect in VS code here, for that you will have to use **git pull origin main** command. This command is used to fetch and download content from a remote repository and immediately update the local repository to match the content.

Resolving merge conflicts:

A merge conflict is an event that takes place when github is not able to automatically merge two commits due to differences in their codes.

Undoing changes:

Case 1: Staged changes

git reset <- file name ->
git reset

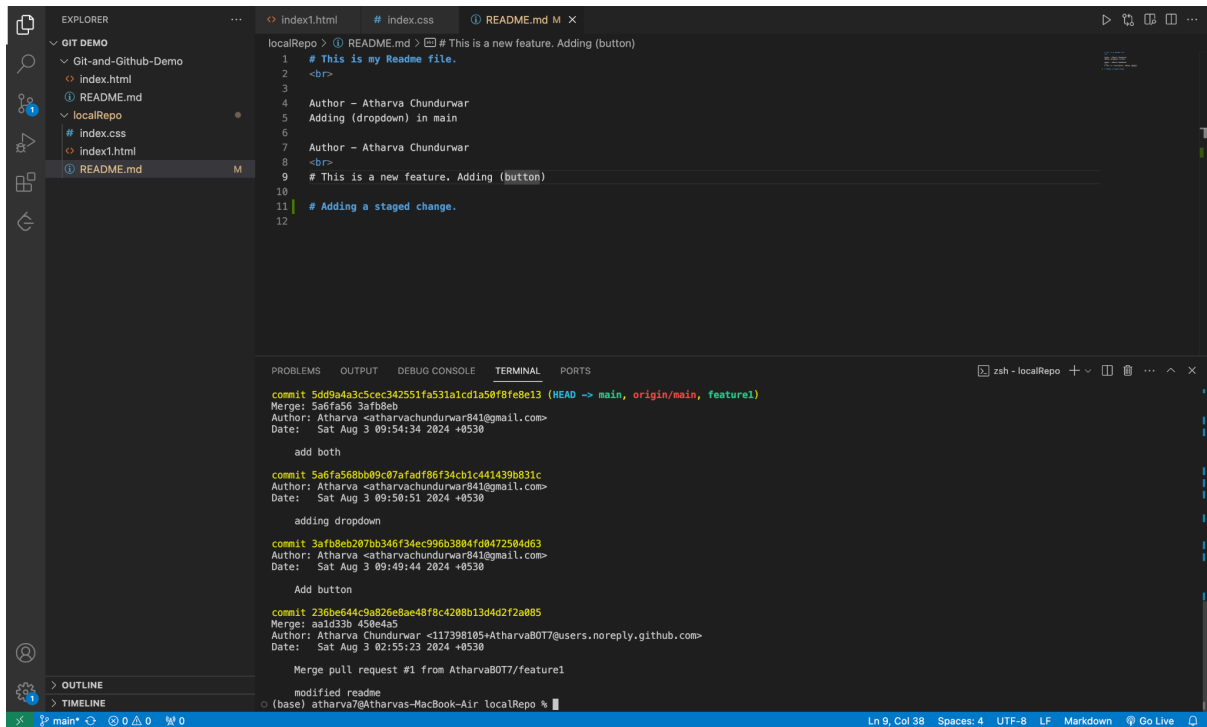
Case 2: Committed changes (for one commit)

git reset head ~1 (by default, git stores commit in form of stack and the last commit is named HEAD)

Case 3: Committed changes (for many changes)

git reset <- commit # ->
git reset --hard <- commit # ->

Use command (**git log**) to display all the commits. (press q button to quit)



```
localRepo > @ README.md > # This is a new feature. Adding (button)
1  # This is my Readme file.
2  <br>
3
4  Author - Atharva Chundurwar
5  Adding (dropdown) in main
6
7  Author - Atharva Chundurwar
8  <br>
9  # This is a new feature. Adding (button)
10
11 # Adding a staged change.
12

commit 5dd9a43c5cec342551fa531a1cd1a50f8fe8e13 (HEAD -> main, origin/main, feature1)
Merge: 5a6fa56 3afb8eb
Author: Atharva <atharvachundurwar841@gmail.com>
Date: Sat Aug 3 09:54:34 2024 +0530

add both

commit 5a6fa568bb09c07afadf86f34cb1c441439b831c
Author: Atharva <atharvachundurwar841@gmail.com>
Date: Sat Aug 3 09:50:51 2024 +0530

adding dropdown

commit 3afb8eb207bb345f34ec996b3804f40472504d63
Author: Atharva <atharvachundurwar841@gmail.com>
Date: Sat Aug 3 09:49:44 2024 +0530

Add button

commit 236be644c9a826e8ae48f8c4208b13d4d2f2a085
Merge: a01d33b 480e4a5
Author: Atharva Chundurwar <117398185+Atharva807@users.noreply.github.com>
Date: Sat Aug 3 02:55:23 2024 +0530

Merge pull request #1 from Atharva807/feature1

modified readme
(base) atharva7@Atharvas-MacBook-Air localRepo %
```

Each commit has its own commit hash, we have to enter git reset commit hash to go to that commit.

Fork

Fork in git is a new repository that shares code and visibility settings with the original “upstream” repository. **Upstream repo = main branch (main repo)**

Fork is a rough copy used to check bugs / changes.