GIT & GITHUB

==========

Check out our notes, projects, and source code here: -

Linkedin :- https://www.linkedin.com/in/swapnill-singh-osab

Github :- https://github.com/theswapnillsinghosab

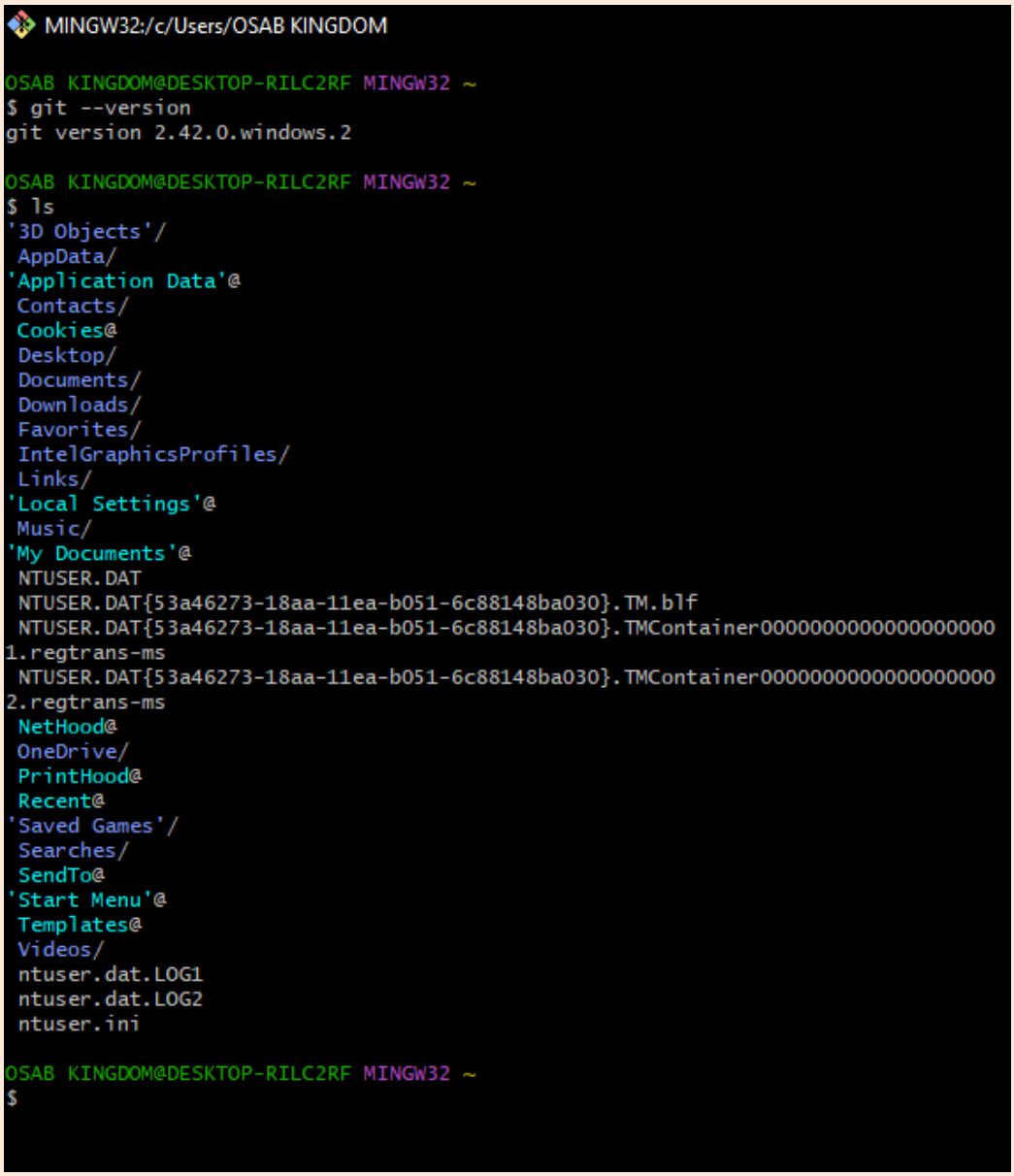
Telegram Channel :- https://t.me/cseswap

[Connect with us for more]

==========

➔Download git from

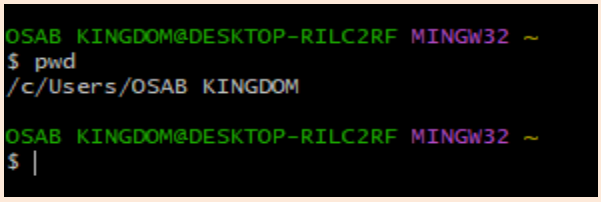
https://git-scm.com/downloads

- Click next next next…… for setup
- Try any command:- **git –version & ls**
- 

```
MINGW32:/c/Users/OSAB KINGDOM

OSAB KINGDOM@DESKTOP-RILC2RF MINGW32 ~
$ git --version
git version 2.42.0.windows.2

OSAB KINGDOM@DESKTOP-RILC2RF MINGW32 ~
$ ls
'3D Objects'/
 AppData/
'Application Data'@
 Contacts/
 Cookies@
 Desktop/
 Documents/
 Downloads/
 Favorites/
 IntelGraphicsProfiles/
 Links/
'Local Settings'@
 Music/
'My Documents'@
 NTUSER.DAT
 NTUSER.DAT{53a46273-18aa-11ea-b051-6c88148ba030}.TM.blf
 NTUSER.DAT{53a46273-18aa-11ea-b051-6c88148ba030}.TMContainer00000000000000000000
1.regtrans-ms
 NTUSER.DAT{53a46273-18aa-11ea-b051-6c88148ba030}.TMContainer00000000000000000000
2.regtrans-ms
 NetHood@
 OneDrive/
 PrintHood@
 Recent@
'Saved Games'/
 Searches/
 SendTo@
'Start Menu'@
 Templates@
 Videos/
 ntuser.dat.LOG1
 ntuser.dat.LOG2
 ntuser.ini

OSAB KINGDOM@DESKTOP-RILC2RF MINGW32 ~
$
```

➔to see the all file of folder :- ls

➔Find Directory :- pwd

- 

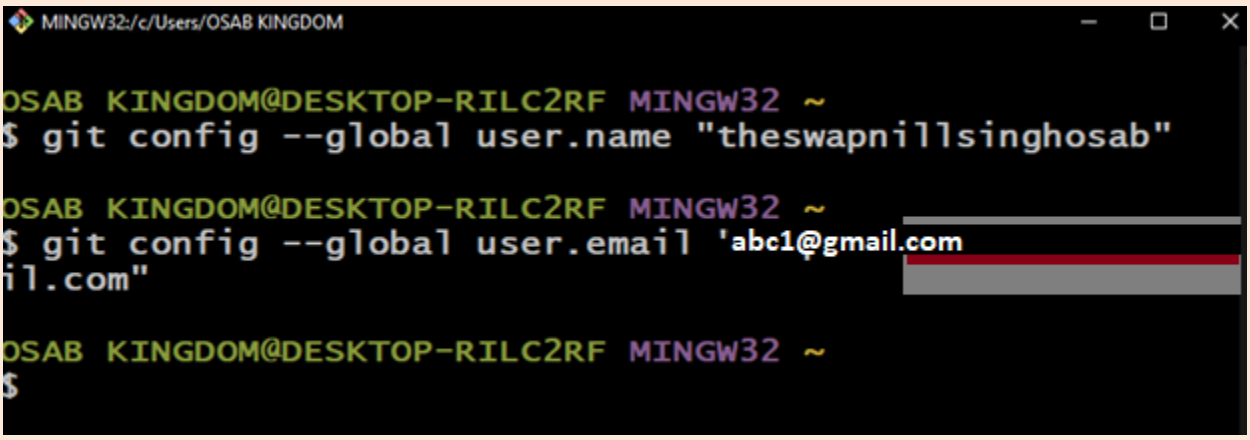
```
OSAB KINGDOM@DESKTOP-RILC2RF MINGW32 ~
$ pwd
/c/Users/OSAB KINGDOM

OSAB KINGDOM@DESKTOP-RILC2RF MINGW32 ~
$
```

➔CLEAR SCREEN :- **clear**

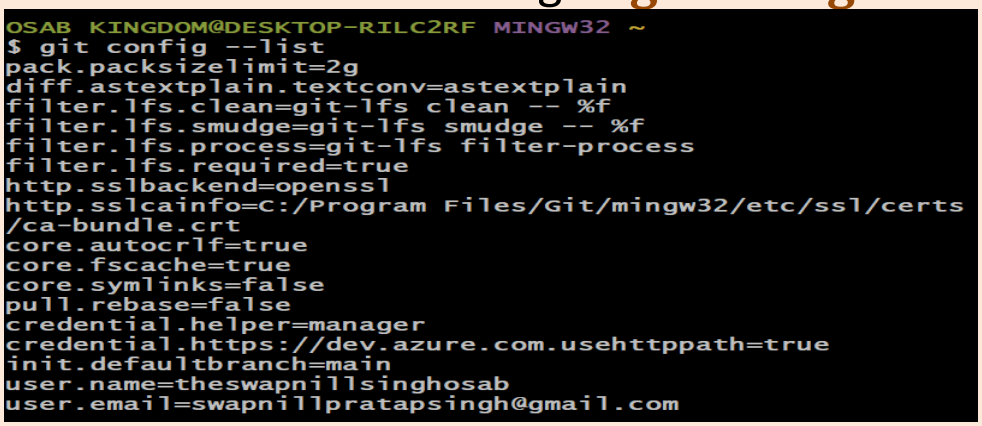
Configuring git :-

➔Meaning of ~ (tild) :- we are in root directoey of system

➔config user_name and email :- **git config**



➔how to see what I config :- **git config –list**



-:- USE GIT WITH VISUAL CODE STUDIO -:-

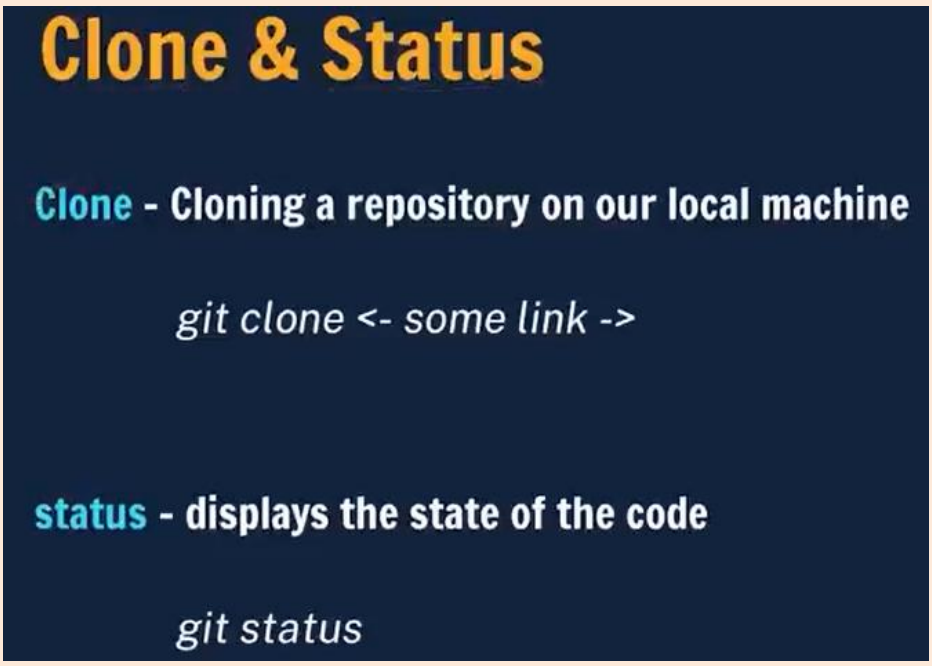
➔create a folder gitdemo and open in in visual studio code

➔open terminal and check git is accessible from vs code or not

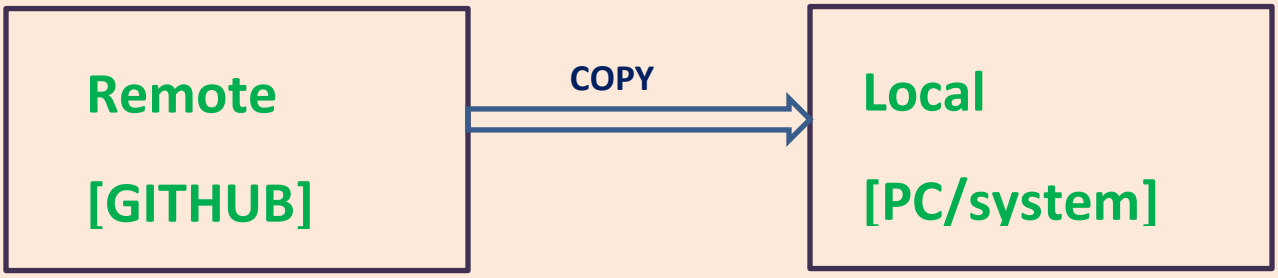
```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS

PS C:\Users\OSAB KINGDOM\OneDrive\Desktop\GITDEMO> git --version
git version 2.42.0.windows.2
PS C:\Users\OSAB KINGDOM\OneDrive\Desktop\GITDEMO>
```

✓ **Yes accessible**

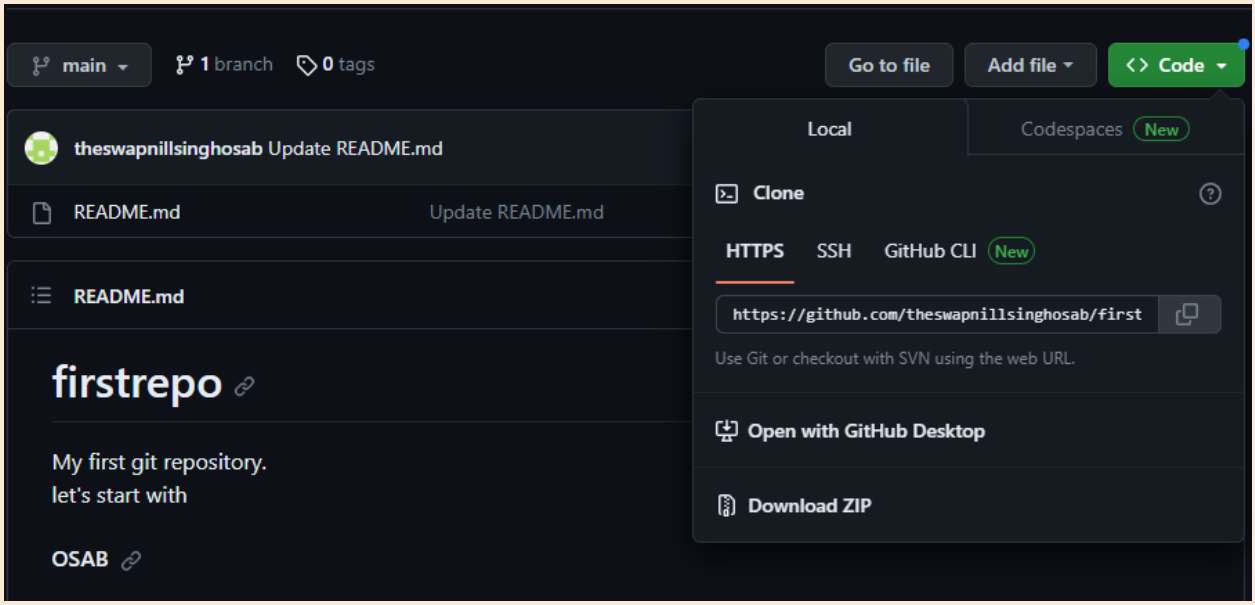
clone & status ---------



➢ **Clone :-** if we want to copy a repository from github to our system we use Clone command.

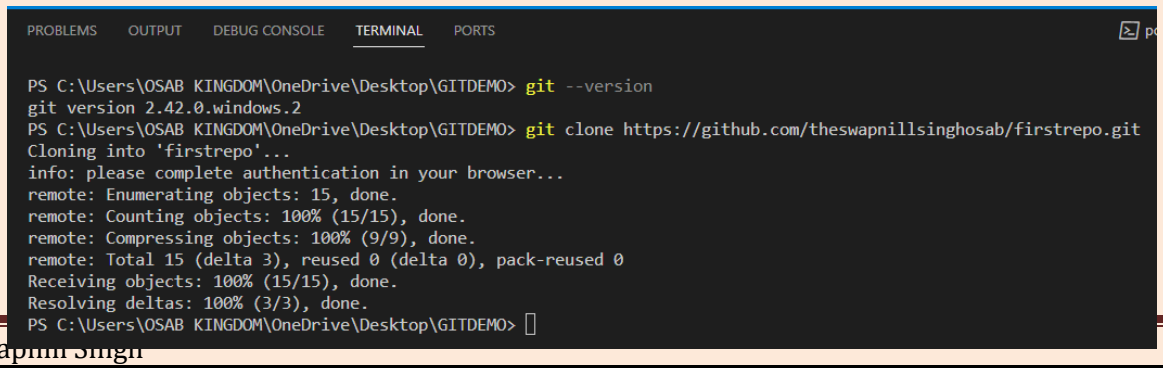


- **FIRST WE COPY THE REPOSITORY ADDRESS FROM GITHUB**

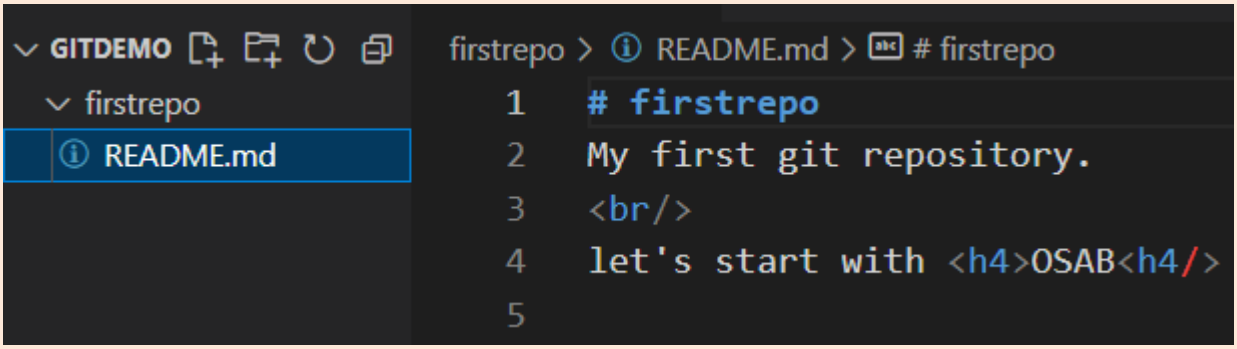


➢ **Use command :-**

Git clone <--repo_link-->



Here you can see that folder is copied:-



- **Status :- check the status**

```
PS C:\Users\OSAB KINGDOM\OneDrive\Desktop\GITDEMO\firstrepo> git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
PS C:\Users\OSAB KINGDOM\OneDrive\Desktop\GITDEMO\firstrepo>
```

➔**there are four type of status**

a) **Unrtracked :- new file that git doesn't yet track it's mean your file exist but not commited .**

b) **Modified:- When you do any changes in file.**

c) **Staged :-File is ready to committed.**

d) **Unmodified:- When you don't have any changes in file.**

Add & Commit :-

- **Add :- Add new or changed files in your working directory to the git staging area.**

Git add <-file name->

```
PS C:\Users\OSAB KINGDOM\OneDrive\Desktop\GITDEMO\firstrepo> git add index.html
PS C:\Users\OSAB KINGDOM\OneDrive\Desktop\GITDEMO\firstrepo>  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)
        new file:   index.html

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

PS C:\Users\OSAB KINGDOM\OneDrive\Desktop\GITDEMO\firstrepo>
```

How to add all changes :- **git add .**

```
PS C:\Users\OSAB KINGDOM\OneDrive\Desktop\GITDEMO\firstrepo> git add .
PS C:\Users\OSAB KINGDOM\OneDrive\Desktop\GITDEMO\firstrepo> 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

PS C:\Users\OSAB KINGDOM\OneDrive\Desktop\GITDEMO\firstrepo>
```

Commit :- It is record the changes

➔ **git commit –m “some message”**

```
PS C:\Users\OSAB KINGDOM\OneDrive\Desktop\GITDEMO\firstrepo>  git commit -m "add new paragraph"
[main 583b32f] add new paragraph
 2 files changed, 2 insertions(+), 1 deletion(-)
 create mode 100644 index.html
PS C:\Users\OSAB KINGDOM\OneDrive\Desktop\GITDEMO\firstrepo> 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
```

➢ Push Command :- Uplpoad local repo content to remote repo

➔ git push origin main

```
PS C:\Users\OSAB KINGDOM\OneDrive\Desktop\GITDEMO\firstrepo> git push origin main
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 4 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (4/4), 342 bytes | 171.00 KiB/s, done.
Total 4 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/theswapnillsinghosab/firstrepo.git
   1e00135..583b32f  main -> main
```

Now you can see the changes:-

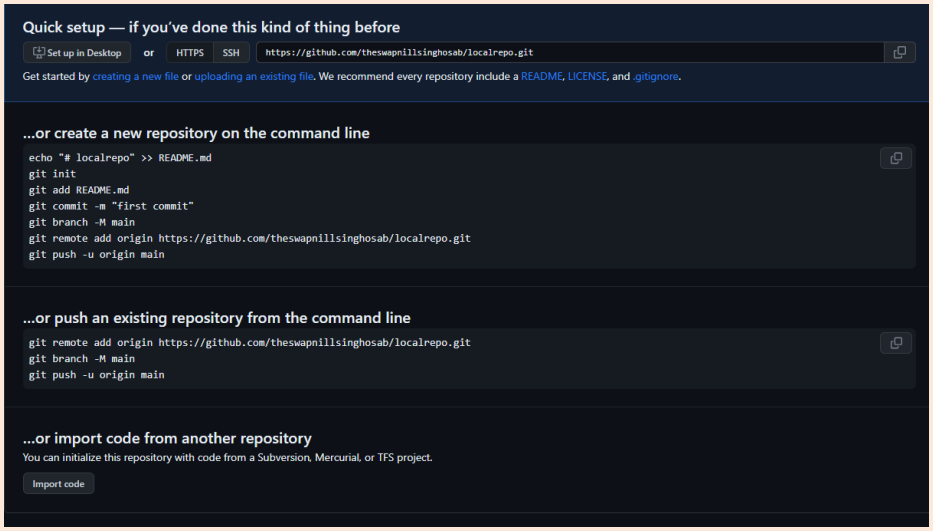
➢ Init Command :- Used to create a new git repo

- **git init**

```
PS C:\Users\OSAB KINGDOM\OneDrive\Desktop\gitdemo\localrepo> git init
Initialized empty Git repository in C:/Users/OSAB KINGDOM/OneDrive/Desktop/GITDEMO/Localrepo/.git/
PS C:\Users\OSAB KINGDOM\OneDrive\Desktop\gitdemo\localrepo>
```

now it is a git repository

➔create a new repo in github (localrepo)



- **git remote add origin <--link >** **(to set origin)**
- **git remote –v** **(to verify remote)**

→

```
PS C:\Users\OSAB KINGDOM\OneDrive\Desktop\gitdemo\localrepo> git remote add origin https://github.com/theswapnillsinghosab/localrepo.git
PS C:\Users\OSAB KINGDOM\OneDrive\Desktop\gitdemo\localrepo> git remote -v
origin  https://github.com/theswapnillsinghosab/localrepo.git (fetch)
origin  https://github.com/theswapnillsinghosab/localrepo.git (push)
PS C:\Users\OSAB KINGDOM\OneDrive\Desktop\gitdemo\localrepo>
```

- **git branch** **(to check the branch)**
-

```
PS C:\Users\OSAB KINGDOM\Desktop\GITDEMO\localrepo> git branch
* main
```

- **git branch –M Mian** **(to change the branch name)**
- **git push origin main**

  ➔for shortcut use **git push –u origin main** →after you push the code by git push

```
PS C:\Users\OSAB KINGDOM\Desktop\GITDEMO\localrepo> git push -u origin main
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 4 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (4/4), 324 bytes | 162.00 KiB/s, done.
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/theswapnillsinghosab/localrepo.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
PS C:\Users\OSAB KINGDOM\Desktop\GITDEMO\localrepo>
```

init - used to create a new git repo

git init

git remote add origin <- link ->

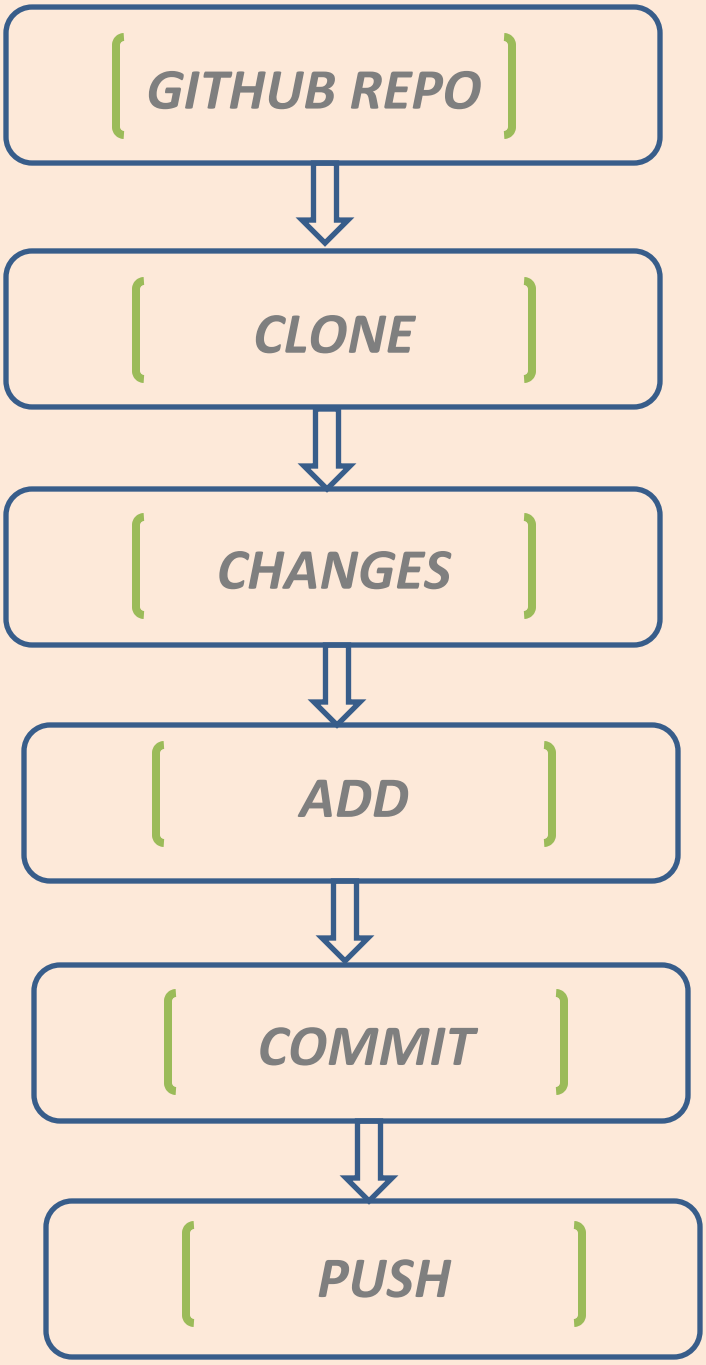
git remote -v (to verify remote)

git branch (to check branch)

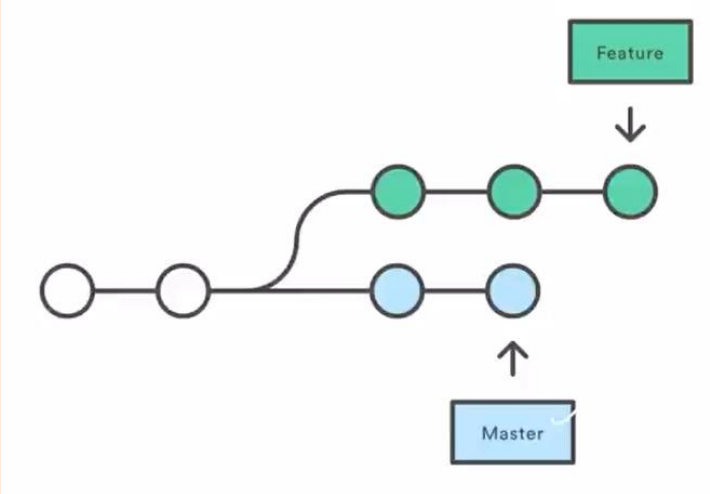
git branch -M main (to rename branch)

git push origin main

-: WORKFLOW :-



▪ GIT HUB BRANCHES :-



➢ git branch :- (to check the branch)

```
PS C:\Users\OSAB KINGDOM\Desktop\GITDEMO\localrepo>  git branch
* feature1
  main
```

we are in feature1 branch

➢ git branch –M main :- (to rename branch)

➢ git checkout <-branch name-> :- (to navigate)

if we want to switch fature1 to main

```
PS C:\Users\OSAB KINGDOM\Desktop\GITDEMO\localrepo> git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
PS C:\Users\OSAB KINGDOM\Desktop\GITDEMO\localrepo> git branch
  feature1
* main
PS C:\Users\OSAB KINGDOM\Desktop\GITDEMO\localrepo>
```

➢ git checkout –b <-new branch name->:- (to create new branch)

```
PS C:\Users\OSAB KINGDOM\Desktop\GITDEMO\localrepo> git checkout -b feature1
Switched to a new branch 'feature1'
```

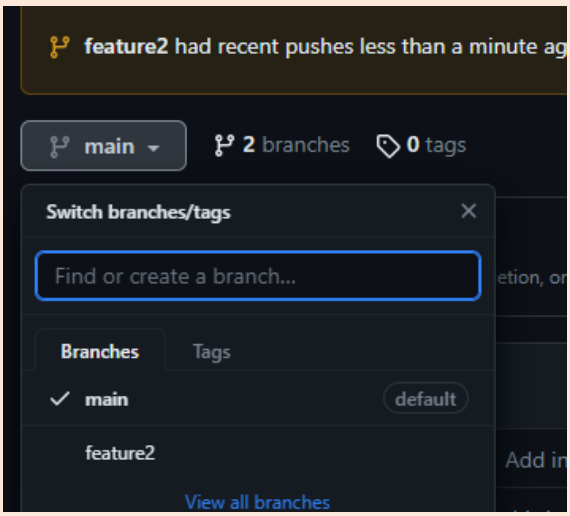
➢ git branch –d <-branch name->:- (to delete branch)

```
PS C:\Users\OSAB KINGDOM\Desktop\GITDEMO\localrepo> git branch -d  feature1
Deleted branch feature1 (was f4b2fb7).
```

Note:-if we try to delete branch where we are exist.That branch can't be deleted.

Push the new branch :-

```
PS C:\Users\OSAB KINGDOM\Desktop\GITDEMO\localrepo> git push origin feature2
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 4 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 510 bytes | 510.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'feature2' on GitHub by visiting:
remote:      https://github.com/theswapnillsinghosab/localrepo/pull/new/feature2
remote:
To https://github.com/theswapnillsinghosab/localrepo.git
 * [new branch]      feature2 -> feature2
```



❖ Merging Code :-

Way-1

➢ **git diff <- branch name->** (to compare commits ,branches, files & more)

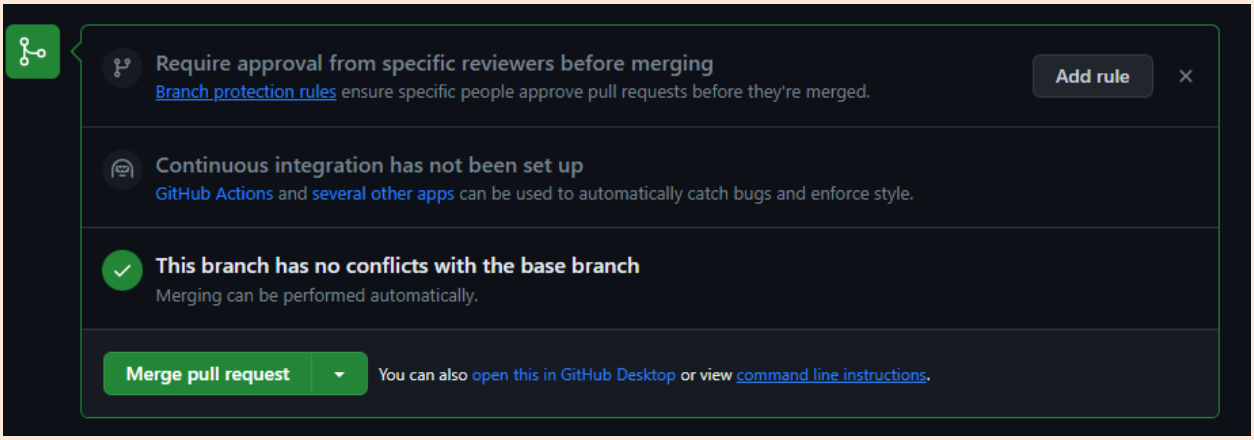
```
PS C:\Users\OSAB KINGDOM\Desktop\GITDEMO\localrepo> git diff main
diff --git a/index.html b/index.html
index e51aee8..2074e81 100644
--- a/index.html
+++ b/index.html
@@ -1 +1,12 @@
 <p>HELLO BROTHER</p>
+<!DOCTYPE html>
+<html lang="en">
+<head>
+    <meta charset="UTF-8">
+    <meta name="viewport" content="width=device-width, initial-scale=1.0">
+    <title>Document</title>
+</head>
+<body>
+<h1>THIS IS A FEATURE</h1>
\ No newline at end of file
```

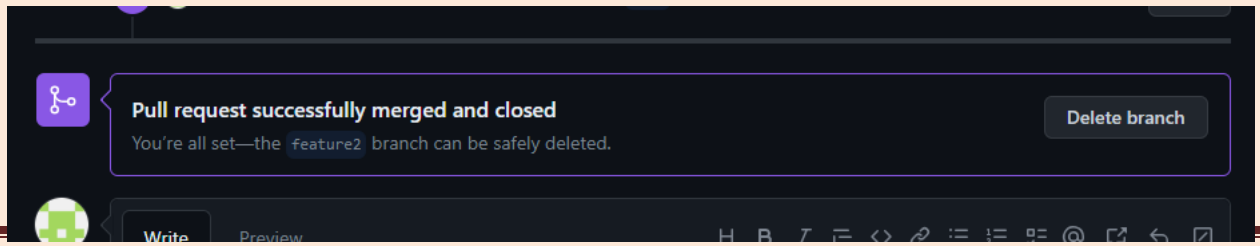
➢ **git merge <- branch name->** (to merge 2 branches)

Way-2

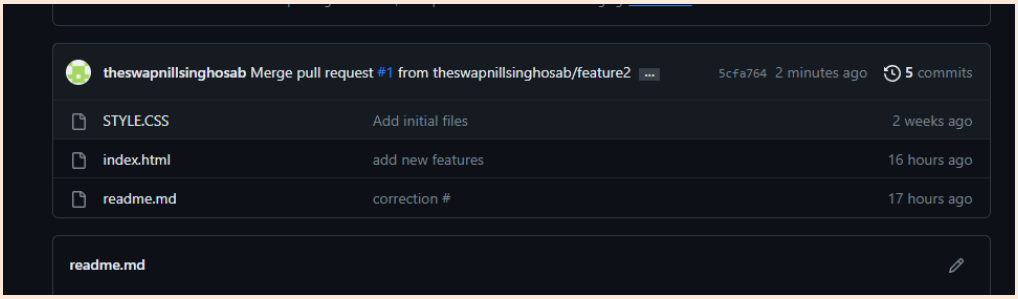
Create PR (Pull Request):-

It let's you tell others about changes you've pushed to arepository on Github.





Here both branch are merged :-



If we want to see these changes in our local system:-

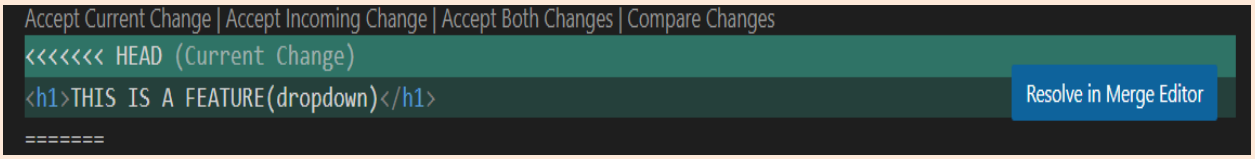
- **Pull command :-** used to fetch and dowmload content from a remote repo and immediately update the local repo to matchthat content.

- **git pull origin main**

```
PS C:\Users\OSAB KINGDOM\Desktop\GITDEMO\localrepo> git pull origin main
remote: Enumerating objects: 1, done.
remote: Counting objects: 100% (1/1), done.
remote: Total 1 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (1/1), 640 bytes | 80.00 KiB/s, done.
From https://github.com/theswapnillsinghosab/localrepo
 * branch            main       -> FETCH_HEAD
   f4b2fb7..5cfa764  main       -> origin/main
Updating f4b2fb7..5cfa764
Fast-forward
 index.html | 11 ++++++++++
 1 file changed, 11 insertions(+)
PS C:\Users\OSAB KINGDOM\Desktop\GITDEMO\localrepo>
```

Resolving Merge Conflicts:- An event that take place when it is unable to automatically resolve difference in code between two commits.

Here vs code suggest resolving methods :-




```
PS C:\Users\OSAB KINGDOM\Desktop\GITDEMO\localrepo> git diff main
diff --git a/index.html b/index.html
index b9403d7..765d7f9 100644
--- a/index.html
+++ b/index.html
@@ -7,7 +7,6 @@
     <title>Document</title>
 </head>
 <body>
-<h1>THIS IS A FEATURE (button)</h1>
-
+<h1>THIS IS A FEATURE(dropdown)</h1>
 </body>
 </html>
\ No newline at end of file
PS C:\Users\OSAB KINGDOM\Desktop\GITDEMO\localrepo> git merge main
Auto-merging index.html
CONFLICT (content): Merge conflict in index.html
Automatic merge failed; fix conflicts and then commit the result.
```

- **git log :-** to check all commits

```
PS C:\Users\OSAB KINGDOM\Desktop\GITDEMO\localrepo> git log
commit aa47a86521bfcf7de64614b8ae156ce12e01cf3f (HEAD -> main, origin/main, feature2)
Author: theswapnillsinghosab <swapnillpratapsingh@gmail.com>
Date:   Sat Oct 21 16:35:08 2023 +0530

    merge both features

commit 9755988c684c831f1700878c0b6e218595f42d7b
Merge: 0a86cca fe1f673
Author: theswapnillsinghosab <swapnillpratapsingh@gmail.com>
```

Note:- **q** after it

Undoing Changes :-

Case1: staged changes

- **git reset <--file name>**
- **git reset** (for all files)

Case2: committed changes (for one commit)

- **git reset HEAD~1**

Case3: commited changes (for one commit)

- **git reset <-commit hash->**
- **git reset --hard <-commit hash->** (for vs code)

Fork :- A fork is a new repository thatshares codeand visibility settings with the original “upstream” repository. Fork is rough copy.

In short :-To upload a project to GitHub using Git, follow these steps:

1. Initialize a Git Repository (if not already initialized)

If your project is not already a Git repository, navigate to your project folder in the terminal and run:

```
git init
```

This will create a new Git repository in your project directory.

2. Add Your Project Files to the Git Repository

Add the files you want to track with Git:

```
git add .
```

The . will add all files in the directory. You can specify specific files instead if you only want to add a few.

3. Commit Your Changes

Once you've added the files, commit them to the repository with a message:

```
git commit -m "Initial commit"
```

Replace "Initial commit" with a descriptive message of your changes.

4. Create a Remote Repository on GitHub

Go to GitHub, create a new repository, and don't initialize it with a README or other files.

5. Add the Remote Repository

After creating the repository on GitHub, copy the repository URL. Now, link your local repository to the remote one by running the following command (replace the URL with the one from your GitHub repo):

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

6. Push Your Code to GitHub

Push the committed code to the GitHub repository:

```
git push -u origin master
```

This will upload your local commits to the master branch of the remote repository.

7. (Optional) Set Upstream Branch for Future Pushes

If you're working with a new branch and want to set it as the upstream branch for future pushes:

```
git push --set-upstream origin your-branch-name
```

This ensures that future pushes and pulls are done to the correct branch.

That's it! Your project is now uploaded to GitHub.

---- Errors in GitHub ----

1. Authentication Failed

Error Message:

```
fatal: Authentication failed for 'https://github.com/username/repository.git'
```

Cause: This happens when GitHub doesn't accept your username/password or token.

- **Solution:** Use a **Personal Access Token (PAT)** instead of your password. You can generate it from GitHub's settings (Settings > Developer settings > Personal access tokens). Then, replace your password with the token when prompted.

If you're using SSH, ensure that you've added your SSH key to GitHub (Settings > SSH and GPG keys).

2. Repository Not Found

Error Message:

```
fatal: repository 'https://github.com/username/repository.git/' not found
```

Cause: The repository URL is incorrect, or it doesn't exist.

- **Solution:** Double-check the URL you are using. Ensure that the repository exists and that you have the correct access rights.

3. Permission Denied (publickey)

Error Message:

```
git@github.com: Permission denied (publickey).
fatal: Could not read from remote repository.
```

Cause: SSH keys are not set up correctly or are missing.

- **Solution:**

1. Ensure that you've generated an SSH key (ssh-keygen -t rsa -b 4096 -C "your_email@example.com").
2. Add the SSH key to GitHub (under Settings > SSH and GPG keys).
3. Make sure that your remote URL uses SSH (git@github.com:username/repository.git).

4. Non-Fast-Forward Updates Were Rejected

Error Message:

```
To https://github.com/username/repository.git
 ! [rejected]        master -> master (non-fast-forward)
error: failed to push some refs to 'https://github.com/username/repository.git'
```

Cause: This error occurs when there are commits on the remote repository that aren't in your local repository (your local history is behind the remote).

- **Solution:** Pull the latest changes from the remote before pushing:
- git pull origin master

  Resolve any merge conflicts and then push your changes.

5. Merge Conflicts

Error Message:

```
Auto-merging <file_name>
CONFLICT (content): Merge conflict in <file_name>
```

Cause: Git is unable to automatically merge changes between branches.

- **Solution:**
  1. Open the conflicted file(s).
  2. Look for the <<<<<<<, =======, and >>>>>>> markers and manually resolve the conflict.
  3. After resolving, add and commit the changes:
  4. git add <file_name>
  5. git commit -m "Resolved merge conflict"

6. Push Rejected Due to Branch Protection Rules

Error Message:

remote: error: GH001: Your push was rejected because the branch is protected.

Cause: The branch you're trying to push to is protected (e.g., main or master).

- **Solution:** If you don’t have permission to push directly to the protected branch, you’ll need to create a new branch:
- git checkout -b new-branch
- git push origin new-branch

  Then, create a pull request from the new branch.

7. "Detached HEAD" State

Error Message:

You are in 'detached HEAD' state. You can look around, make experimental changes, and commit them, but you cannot commit them to any branch.

Cause: You're not on a branch; you've checked out a specific commit instead.

- **Solution:** Checkout the correct branch you want to work on:
- git checkout main

8. File Already Exists in the Repository (Conflict with Local Changes)

Error Message:

fatal: pathspec 'filename' did not match any files

Cause: The file you’re trying to add has already been added or exists in the repository.

- **Solution:** Check if the file is already being tracked by Git. You can use:
- git status

9. File Size Too Large

Error Message:

```
remote: error: File <file_name> is too large for Git LFS (exceeds 100.0 MB).
```

Cause: The file exceeds GitHub's file size limit for regular Git repositories.

- **Solution:** Use **Git Large File Storage (Git LFS)** to store large files. You can install it and track large files with:
- git lfs install
- git lfs track "<large_file>"
- git add .gitattributes

10. Unable to Clone a Repository

Error Message:

```
fatal: repository 'https://github.com/username/repository.git/' not found
```

Cause: The repository URL is incorrect or doesn't exist.

- **Solution:** Ensure the repository exists and you have access. Check the URL and ensure it's spelled correctly.

General Troubleshooting Tips:

- **Check your Git version:** Sometimes, an outdated version of Git can cause problems. Ensure you have the latest version.
- git --version
- **Network issues:** If you are unable to access GitHub, check your network connection or firewall settings.
- **GitHub Status:** Sometimes, GitHub might be down or facing issues. Check GitHub's status page for any ongoing outages.

==========

Check out our notes, projects, and source code here: -

Linkedin :- https://www.linkedin.com/in/swapnill-singh-osab

Github :- https://github.com/theswapnillsinghosab

Telegram Channel :- https://t.me/cseswap

[Connect with us for more]

==========