

# Git | GitHub | Git Commands

~ Rahul Raj

## # Some important Git commands :-

Terminal Commands:

- ✓ - cd : Changes directory. Example cd innerDirectory
  - ✓ - pwd : Displays the location of the current working directory. Example pwd
  - ✓ - ls : Displays a list of content of a directory. Example ls
  - ✓ - mkdir : Creates a new directory. Example mkdir newDirectory
  - ✓ - rmdir : Deletes an empty directory Example rmdir oldEmptyDirectory
  - ✓ - rm -r <dir> : Deletes the directory and its contents. Example rm -r oldDirectory
  - rm : removes a file. Example rm file
  - ✓ touch : Creates a file without any content. Example touch newFile
  - ✓ cat : Reads data from the file and gives their content as output. Use CTRL+D to terminate
- Example i) cat existingFile  
ii) cat > someFile (Writes to a file)  
iii) cat >> someFile (Appends text to a file)
- history : Shows recently used commands
  - cp : cp source destination
  - mv : mv source destination

# to terminate we can use **ctrl + d**

→ These are some of the commands...

```
MINGW64:/c/assignment

Lenovo@LAPTOP-CSGP8BLS MINGW64 /c/assignment
$ cat hello.txt

Lenovo@LAPTOP-CSGP8BLS MINGW64 /c/assignment
$ cat > hello.txt
Rahul: My name is Rahul Raj.
Lenovo@LAPTOP-CSGP8BLS MINGW64 /c/assignment
$ cat hello.txt
Rahul: My name is Rahul Raj.
Lenovo@LAPTOP-CSGP8BLS MINGW64 /c/assignment
$ cat >> hello.txt
Prince: My name is Prince Raj.
Lenovo@LAPTOP-CSGP8BLS MINGW64 /c/assignment
$ cat hello.txt
Rahul: My name is Rahul Raj.Prince: My name is Prince Raj.
Lenovo@LAPTOP-CSGP8BLS MINGW64 /c/assignment
$
```

# Here, we see how a txt file present in the source folder is copied to the destination folder using **cp** command.

```
Lenovo@LAPTOP-CSGP8BLS MINGW64 ~  
$ cd /c/assignment/  
  
Lenovo@LAPTOP-CSGP8BLS MINGW64 /c/assignment  
$ cd source/  
  
Lenovo@LAPTOP-CSGP8BLS MINGW64 /c/assignment/source  
$ ls  
hello.txt  
  
Lenovo@LAPTOP-CSGP8BLS MINGW64 /c/assignment/source  
$ cd ..  
  
Lenovo@LAPTOP-CSGP8BLS MINGW64 /c/assignment  
$ cp source/hello.txt destination/  
  
Lenovo@LAPTOP-CSGP8BLS MINGW64 /c/assignment  
$
```

# Here, we see how a txt file can moved from **source folder** to **destination folder**:

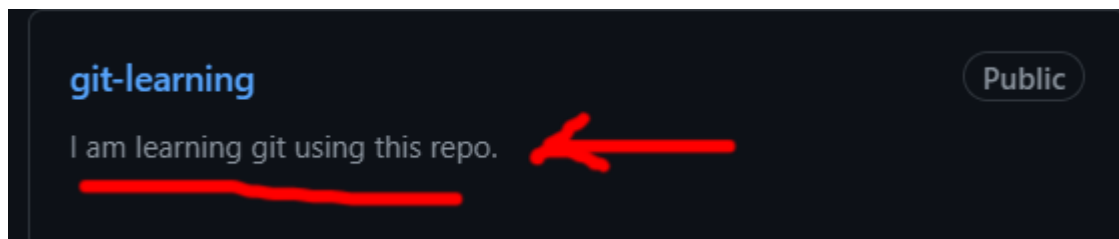
```
MINGW64:/c/assignment/destination  
  
Lenovo@LAPTOP-CSGP8BLS MINGW64 /c/assignment  
$ mv source/hello.txt destination/  
  
Lenovo@LAPTOP-CSGP8BLS MINGW64 /c/assignment  
$ cd destination/  
  
Lenovo@LAPTOP-CSGP8BLS MINGW64 /c/assignment/destination  
$ ls  
hello.txt  
  
Lenovo@LAPTOP-CSGP8BLS MINGW64 /c/assignment/destination  
$
```

→ to remove/delete the file: `rm <fileName>` (enter)

## # Git Commands Tutorial | Git Clone, Add, Commit, Push, Pull, Checkout, Branch, Status

[Source](#)

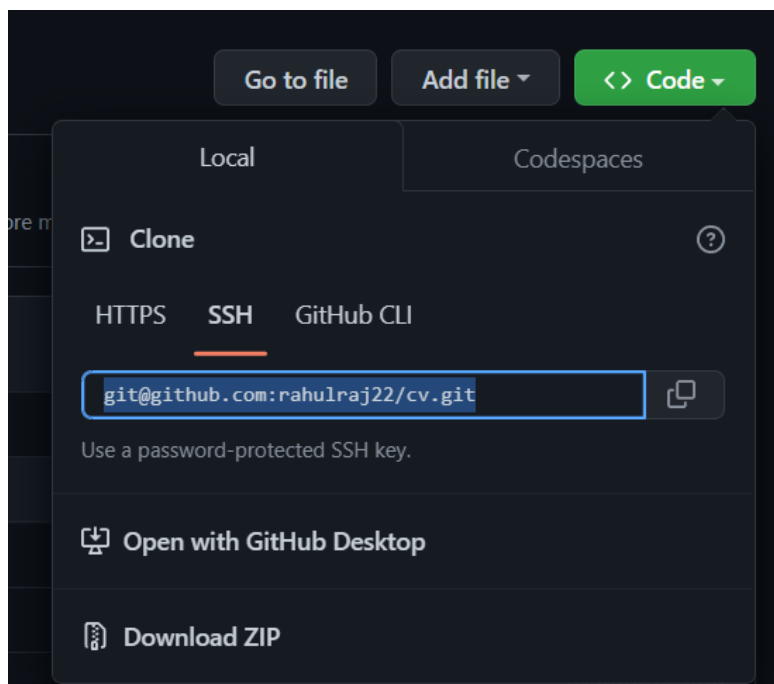
The description we write while making the git repo. Look like this see SS:-



1.

⇒ command: `git clone <repo path i.e ssh path>`

Repo path will be this as shown:

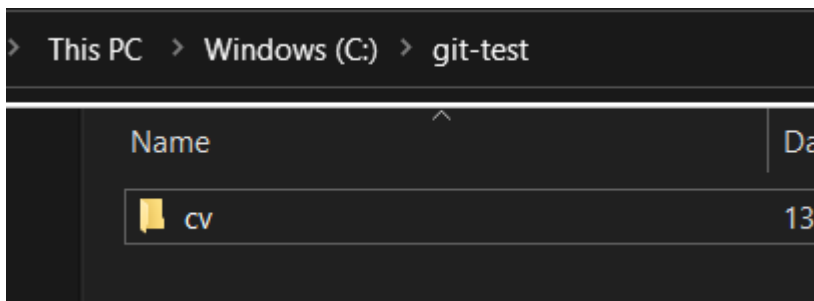


```
MINGW64:/c/git-test

Lenovo@LAPTOP-CSGP8BLS MINGW64 /c/git-test
$ git clone git@github.com:rahulraj22/cv.git
Cloning into 'cv'...
remote: Enumerating objects: 9, done.
remote: Counting objects: 100% (9/9), done.
remote: Compressing objects: 100% (7/7), done.
remote: Total 9 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 77
Receiving objects: 100% (9/9), 275.53 KiB | 503.00 KiB/s, done.

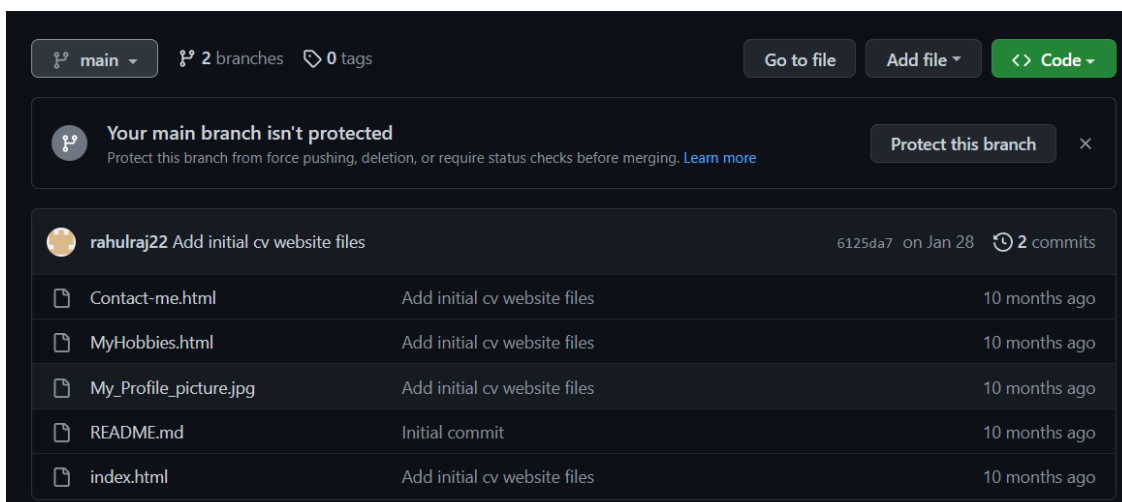
Lenovo@LAPTOP-CSGP8BLS MINGW64 /c/git-test
$
```

→ after doing this a folder/repo got downloaded or cloned to my current/local directory i.e **/c/git-test**



2.

⇒ command: **git add .**



→ we can see here, that this repo has the branch name as **'main'**

```
MINGW64:/c/git-test/cv

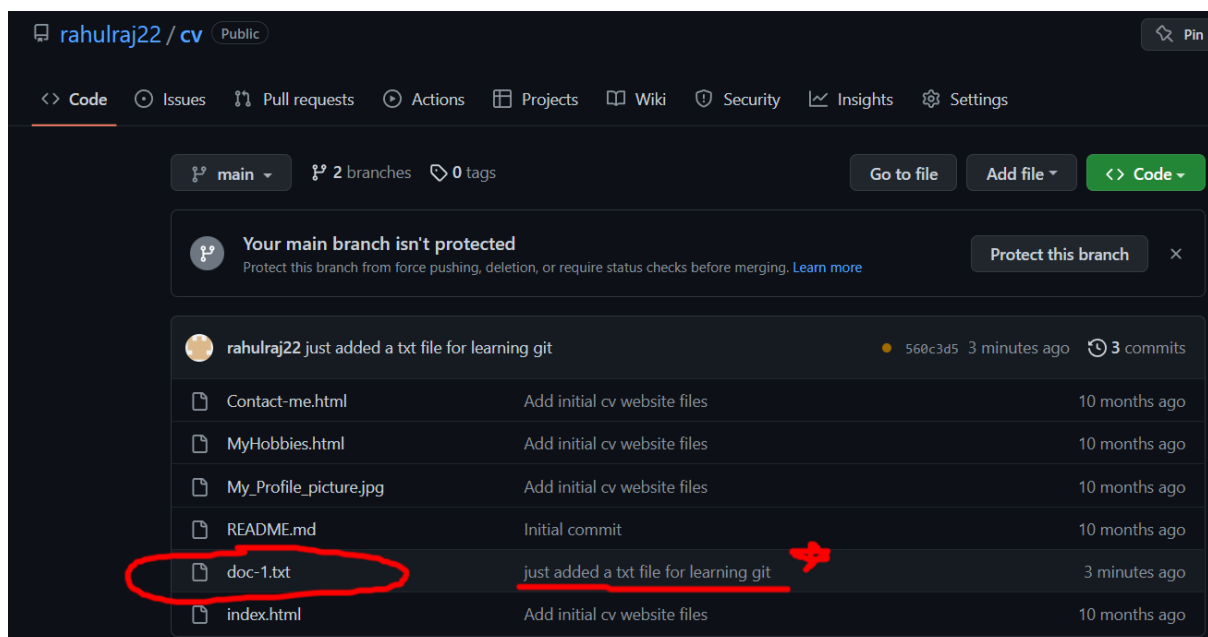
Lenovo@LAPTOP-CSGP8BLS MINGW64 /c/git-test/cv (main)
$ git add .

Lenovo@LAPTOP-CSGP8BLS MINGW64 /c/git-test/cv (main)
$ git commit -m "just added a txt file for learning git"
[main 560c3d5] just added a txt file for learning git
1 file changed, 1 insertion(+)
create mode 100644 doc-1.txt

Lenovo@LAPTOP-CSGP8BLS MINGW64 /c/git-test/cv (main)
$ git push origin main
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 301 bytes | 301.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To github.com:rahu1raj22/cv.git
6125da7..560c3d5 main -> main

Lenovo@LAPTOP-CSGP8BLS MINGW64 /c/git-test/cv (main)
$
```

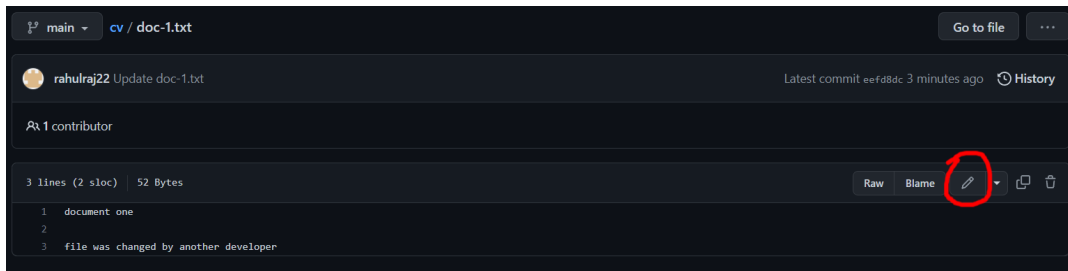
→ After this, file gets reflected in our github, see below:



### 3. Similar to Pushing, we can also Pull :-

⇒ command: **git pull origin <branch-name>**

→ after editing the file in github, we can also pull requests: that means whatever we have made changes in github(**remote**) can also be copied/reflected in our **local** repo this process is called “**Pulling request**”



```
MINGW64:/c/git-test/cv

Lenovo@LAPTOP-CSGP8BLS MINGW64 /c/git-test/cv (main)
$ git pull origin main
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 680 bytes | 3.00 KiB/s, done.
From github.com:rahu1raj22/cv
 * branch          main      -> FETCH_HEAD
   560c3d5..eefd8dc  main      -> origin/main
Updating 560c3d5..eefd8dc
Fast-forward
 doc-1.txt | 4 +++-
 1 file changed, 3 insertions(+), 1 deletion(-)

Lenovo@LAPTOP-CSGP8BLS MINGW64 /c/git-test/cv (main)
$
```

### 3.

⇒ command: **git branch**

⇒ This command will tell which branch we are working in (current working branch marked with \* star, text color will be green). It will also list all other branches we have created. It will tell name of that branch :-

```
Lenovo@LAPTOP-CSGP8BLS MINGW64 /c/git-test/cv (main)
$ git branch
* main

Lenovo@LAPTOP-CSGP8BLS MINGW64 /c/git-test/cv (main)
$
```

### 4. Lets create a new branch:-

⇒ command: **git branch <branch-name>**

→ **main** is the current branch we are working on.

→ we have created new branch called **dev2** using command:

Command: **git branch dev2**

```

Lenovo@LAPTOP-CSGP8BLS MINGW64 /c/git-test/cv (main)
$ git branch
* main

Lenovo@LAPTOP-CSGP8BLS MINGW64 /c/git-test/cv (main)
$ git branch dev2

Lenovo@LAPTOP-CSGP8BLS MINGW64 /c/git-test/cv (main)
$

```

→ we can also check no. and name of all the branches like this:

```

Lenovo@LAPTOP-CSGP8BLS MINGW64 /c/git-test/cv (main)
$ git branch
dev2
* main

Lenovo@LAPTOP-CSGP8BLS MINGW64 /c/git-test/cv (main)
$

```

So, now we have 2 branches one is '**main**' and other one which we made just now i.e '**dev2**'

```

Lenovo@LAPTOP-CSGP8BLS MINGW64 /c/git-test/cv (main)
$ git branch dev3

Lenovo@LAPTOP-CSGP8BLS MINGW64 /c/git-test/cv (main)
$ git branch
dev2
dev3
* main

Lenovo@LAPTOP-CSGP8BLS MINGW64 /c/git-test/cv (main)
$

```

→ After doing this, it will not get reflected in our github repo, so we need to commit those changes. How ? Let's see...

```

Lenovo@LAPTOP-CSGP8BLS MINGW64 /c/git-test/cv (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean

Lenovo@LAPTOP-CSGP8BLS MINGW64 /c/git-test/cv (main)
$ git add .

Lenovo@LAPTOP-CSGP8BLS MINGW64 /c/git-test/cv (main)
$ git commit -m "created new branches like dev dev2 dev3"
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean

Lenovo@LAPTOP-CSGP8BLS MINGW64 /c/git-test/cv (main)
$ git push origin main
Everything up-to-date

Lenovo@LAPTOP-CSGP8BLS MINGW64 /c/git-test/cv (main)

```

## 5. Lets see about 'Checkout' :-

→ checkout is to switch branch(if we want to switch branch for eg. from **\*main** branch to any other branch then we do below command)

⇒ command: **git checkout <branch-name>**

# This is how we can switch the branch :-



```
Lenovo@LAPTOP-CSGP8BLS MINGW64 /c/git-test/cv (main)
$ git branch
  dev
  dev2
  dev3
* main

Lenovo@LAPTOP-CSGP8BLS MINGW64 /c/git-test/cv (main)
$ git checkout dev
Switched to branch 'dev'

Lenovo@LAPTOP-CSGP8BLS MINGW64 /c/git-test/cv (dev)
$ git branch
* dev
  dev2
  dev3
  main

Lenovo@LAPTOP-CSGP8BLS MINGW64 /c/git-test/cv (dev)
$
```

```
Lenovo@LAPTOP-CSGP8BLS MINGW64 /c/git-test/cv (dev)
$ git add .

Lenovo@LAPTOP-CSGP8BLS MINGW64 /c/git-test/cv (dev)
$ git commit -m "add branch dev"
On branch dev
nothing to commit, working tree clean

Lenovo@LAPTOP-CSGP8BLS MINGW64 /c/git-test/cv (dev)
$ git push origin dev
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'dev' on GitHub by visiting:
remote:   https://github.com/rahulraj22/cv/pull/new/dev
remote:
To github.com:rahulraj22/cv.git
 * [new branch]      dev -> dev

Lenovo@LAPTOP-CSGP8BLS MINGW64 /c/git-test/cv (dev)
$ git status
On branch dev
Untracked files:
  (use "git add <file>..." to include in what will be committed)
   dev.txt

nothing added to commit but untracked files present (use "git add" to track)

Lenovo@LAPTOP-CSGP8BLS MINGW64 /c/git-test/cv (dev)
$ git add .

Lenovo@LAPTOP-CSGP8BLS MINGW64 /c/git-test/cv (dev)
$ git commit -m "added a file named dev.txt"
[dev d40bcdd] added a file named dev.txt
1 file changed, 1 insertion(+)
create mode 100644 dev.txt
```

```

Lenovo@LAPTOP-CSGP8BLS MINGW64 /c/git-test/cv (dev)
$ git add .

Lenovo@LAPTOP-CSGP8BLS MINGW64 /c/git-test/cv (dev)
$ git commit -m "added a file named dev.txt"
[dev d40bcdd] added a file named dev.txt
1 file changed, 1 insertion(+)
create mode 100644 dev.txt

Lenovo@LAPTOP-CSGP8BLS MINGW64 /c/git-test/cv (dev)
$ git push origin dev
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 296 bytes | 148.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To github.com:rahulraj22/cv.git
   eef8dc..d40bcdd dev -> dev

Lenovo@LAPTOP-CSGP8BLS MINGW64 /c/git-test/cv (dev)
$ git status
On branch dev
nothing to commit, working tree clean

Lenovo@LAPTOP-CSGP8BLS MINGW64 /c/git-test/cv (dev)
$ git branch
* dev
  dev2
  dev3
  main

Lenovo@LAPTOP-CSGP8BLS MINGW64 /c/git-test/cv (dev)
$ |

```

# Here, we can see that dev.txt is not found in **main** branch:-

dev had recent pushes 4 minutes ago [Compare & pull request](#)

main 3 branches 0 tags [Go to file](#) [Add file](#) [Code](#)

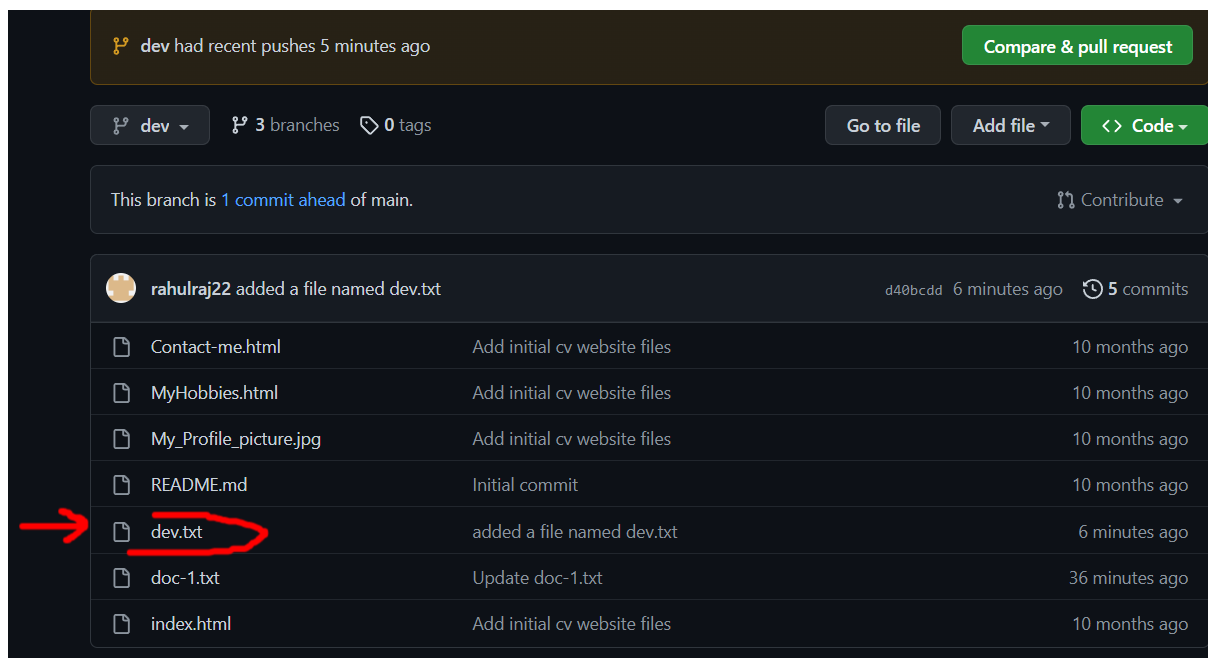
**Your main branch isn't protected**  
Protect this branch from force pushing, deletion, or require status checks before merging. [Learn more](#) [Protect this branch](#)

**rahulraj22** Update doc-1.txt ✓ eef8dc 35 minutes ago 4 commits

File	Commit Message	Time
Contact-me.html	Add initial cv website files	10 months ago
MyHobbies.html	Add initial cv website files	10 months ago
My_Profile_picture.jpg	Add initial cv website files	10 months ago
README.md	Initial commit	10 months ago
doc-1.txt	Update doc-1.txt	35 minutes ago
index.html	Add initial cv website files	10 months ago

Because we have created **dev.txt** file in the **dev branch only**. So, it reflects in the **dev branch only**.

**Fig:-**



**# Now, lets checkout to “main” branch :-**

→ since, we are currently in **dev** branch, so we need to switch it to main branch. Lets see how ?

```
Lenovo@LAPTOP-CSGP8BLS MINGW64 /c/git-test/cv (dev)
$ git branch
* dev
  dev2
  dev3
  main

Lenovo@LAPTOP-CSGP8BLS MINGW64 /c/git-test/cv (dev)
$ git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.

Lenovo@LAPTOP-CSGP8BLS MINGW64 /c/git-test/cv (main)
$ git branch
  dev
  dev2
  dev3
* main

Lenovo@LAPTOP-CSGP8BLS MINGW64 /c/git-test/cv (main)
```

# changes happening in local folder :-

→ while being in main branch, we find this: here, we can't see "dev.txt" file

```
Lenovo@LAPTOP-CSGP8BLS MINGW64 /c/git-test/cv (dev)
$ git branch
* dev
  dev2
  dev3
  main

Lenovo@LAPTOP-CSGP8BLS MINGW64 /c/git-test/cv (dev)
$ git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.

Lenovo@LAPTOP-CSGP8BLS MINGW64 /c/git-test/cv (main)
$ git branch
  dev
  dev2
  dev3
* main
```

This PC > Windows (C:) > git-test > cv

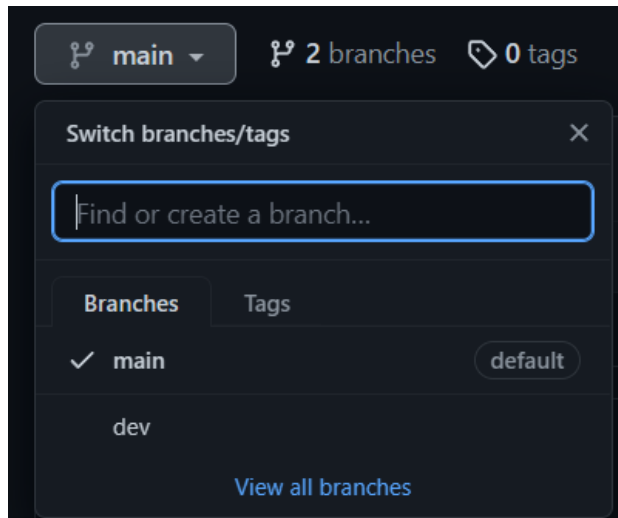
Name	Date modified	Type	Size
.git	04-Dec-22 4:36 AM	File folder	
Contact-me.html	04-Dec-22 3:33 AM	Microsoft Edge HT...	2 KB
doc-1.txt	04-Dec-22 3:55 AM	Text Document	1 KB
index.html	04-Dec-22 3:33 AM	Microsoft Edge HT...	5 KB
My_Profile_picture.jpg	04-Dec-22 3:33 AM	JPG File	289 KB
MyHobbies.html	04-Dec-22 3:33 AM	Microsoft Edge HT...	1 KB
README.md	04-Dec-22 3:33 AM	MD File	1 KB

→ while being in dev branch, we find this: here, we see "dev.txt" file

This PC > Windows (C:) > git-test > cv

Name	Date modified	Type	Size
.git	04-Dec-22 4:34 AM	File folder	
Contact-me.html	04-Dec-22 3:33 AM	Microsoft Edge HT...	2 KB
dev.txt	04-Dec-22 4:34 AM	Text Document	1 KB
doc-1.txt	04-Dec-22 3:55 AM	Text Document	1 KB
index.html	04-Dec-22 3:33 AM	Microsoft Edge HT...	5 KB
My_Profile_picture.jpg	04-Dec-22 3:33 AM	JPG File	289 KB
MyHobbies.html	04-Dec-22 3:33 AM	Microsoft Edge HT...	1 KB
README.md	04-Dec-22 3:33 AM	MD File	1 KB

# We can't see **dev2** in our remote location i.e github repo branches:-



→ So, what to do then ?

Can we push **dev2** branch directly using command:

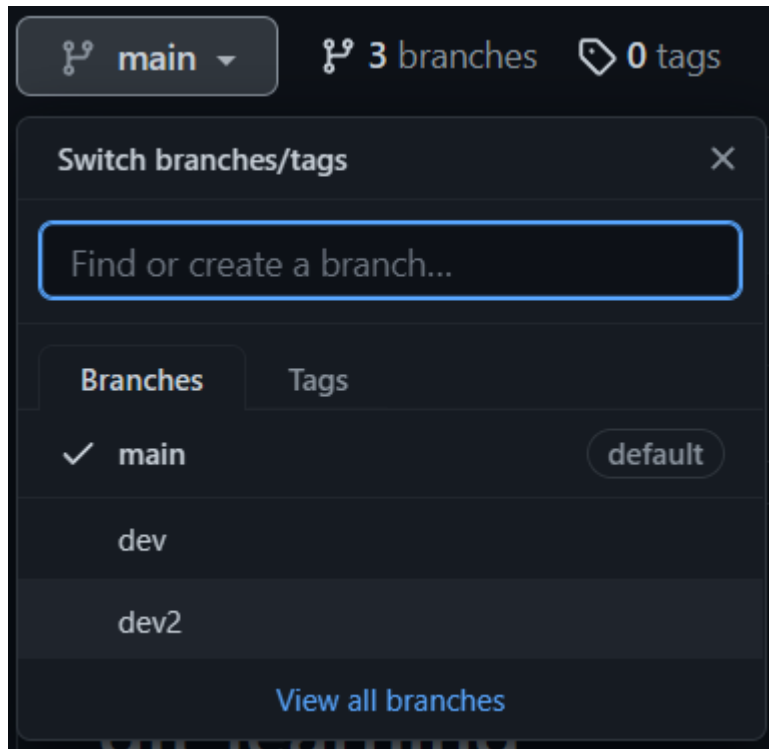
**Command: git push origin dev2**

```
Lenovo@LAPTOP-CSGP8BLS MINGW64 /c/git-test/git-learning (dev2)
$ git branch
dev
* dev2
dev3
main

Lenovo@LAPTOP-CSGP8BLS MINGW64 /c/git-test/git-learning (dev2)
$ git push origin dev2
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'dev2' on GitHub by visiting:
remote:   https://github.com/rahulraj22/git-learning/pull/new/dev2
remote:
To github.com:rahulraj22/git-learning.git
 * [new branch]      dev2 -> dev2

Lenovo@LAPTOP-CSGP8BLS MINGW64 /c/git-test/git-learning (dev2)
$ |
```

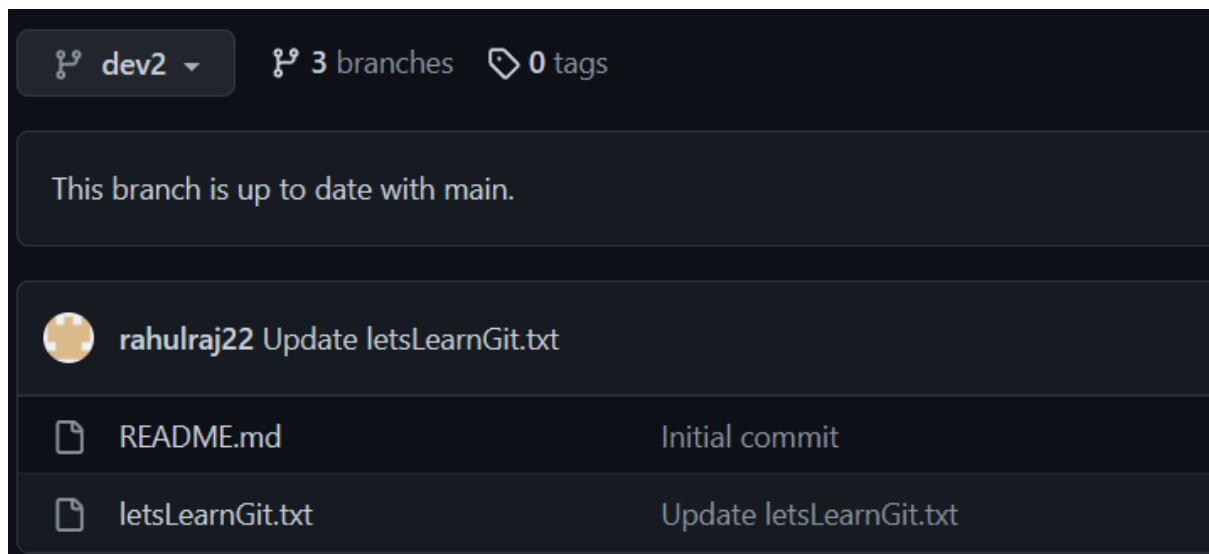
→ we successfully pushed this(**dev2**) branch in our remote repo.  
Fig:



→ After creating a new file named **dev2.txt** and pushing this using:

```
Lenovo@LAPTOP-CSGP8BLS MINGW64 /c/git-test/git-learning (dev2)
$ git push origin dev2
Everything up-to-date
```

And looking in the remote repo. We find that its not updated i.e dev2.txt file is not showing there.



→ So, let's see the status of this branch:-

```
Lenovo@LAPTOP-CSGP8BLS MINGW64 /c/git-test/git-learning (dev2)
$ git status
On branch dev2
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        dev2.txt

nothing added to commit but untracked files present (use "git add" to track)
Lenovo@LAPTOP-CSGP8BLS MINGW64 /c/git-test/git-learning (dev2)
$
```

Here, we can see that **dev2.txt** file is not committed or it's untracked yet.

So, now we have to do all the same steps we used to do in order to push files or any changed file.

i.e

**git add . ,**

**git commit -m "message related to pushing file" ,**

**git push origin < branchName>**

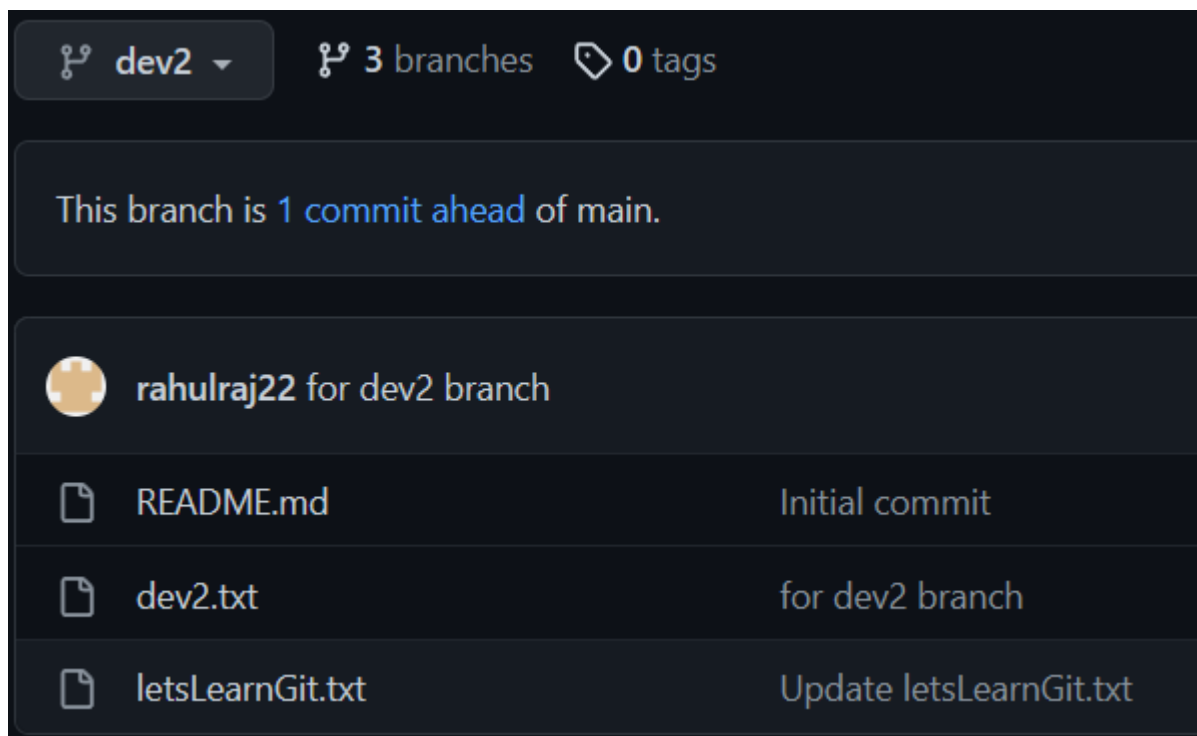
```
Lenovo@LAPTOP-CSGP8BLS MINGW64 /c/git-test/git-learning (dev2)
$ git add .

Lenovo@LAPTOP-CSGP8BLS MINGW64 /c/git-test/git-learning (dev2)
$ git commit -m "for dev2 branch"
[dev2 053a34e] for dev2 branch
1 file changed, 1 insertion(+)
create mode 100644 dev2.txt

Lenovo@LAPTOP-CSGP8BLS MINGW64 /c/git-test/git-learning (dev2)
$ git push origin dev2
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 341 bytes | 170.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:rahulraj22/git-learning.git
   bdb4e96..053a34e  dev2 -> dev2

Lenovo@LAPTOP-CSGP8BLS MINGW64 /c/git-test/git-learning (dev2)
$
```

# finally, dev2 txt file appearing in dev2 branch :-



# After switching from **dev2** branch to **main** branch, **dev2.txt** disappeared :-

```
Lenovo@LAPTOP-CSGP8BLS MINGW64 /c/git-test/git-learning (dev2)
$ git branch
  checkout
  dev
* dev2
  dev3
  main

Lenovo@LAPTOP-CSGP8BLS MINGW64 /c/git-test/git-learning (dev2)
$ git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.

Lenovo@LAPTOP-CSGP8BLS MINGW64 /c/git-test/git-learning (main)
$ git branch
  checkout
  dev
  dev2
  dev3
* main

Lenovo@LAPTOP-CSGP8BLS MINGW64 /c/git-test/git-learning (main)
$ |
```



# after switching from dev2 branch to main branch:-

Name	Date modified	Type	Size
.git	29-Dec-22 5:05 PM	File folder	
letsLearnGit.txt	29-Dec-22 4:23 PM	Text Document	1 KB
README.md	29-Dec-22 3:58 PM	MD File	1 KB

# Now, we will switch to **dev3** branch:-

```
Lenovo@LAPTOP-CSGP8BLS MINGW64 /c/git-test/git-learning (main)
$ git checkout dev3
Switched to branch 'dev3'

Lenovo@LAPTOP-CSGP8BLS MINGW64 /c/git-test/git-learning (dev3)
$ git branch
  dev
  dev2
* dev3
  main

Lenovo@LAPTOP-CSGP8BLS MINGW64 /c/git-test/git-learning (dev3)
$
```

→ being in dev3 branch, lets change or make new file in git-learning folder(local)/repo(remote) :-

# I am creating a dev3.txt file :-

Name	Date modified	Type	Size
.git	29-Dec-22 5:12 PM	File folder	
dev3.txt	29-Dec-22 5:11 PM	Text Document	1 KB
letsLearnGit.txt	29-Dec-22 4:23 PM	Text Document	1 KB
README.md	29-Dec-22 3:58 PM	MD File	1 KB

```

Lenovo@LAPTOP-CSGP8BLS MINGW64 /c/git-test/git-learning (dev3)
$ git branch
  checkout
  dev
  dev2
* dev3
  main

Lenovo@LAPTOP-CSGP8BLS MINGW64 /c/git-test/git-learning (dev3)
$ git add .

Lenovo@LAPTOP-CSGP8BLS MINGW64 /c/git-test/git-learning (dev3)
$ git status
On branch dev3
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   dev3.txt

Lenovo@LAPTOP-CSGP8BLS MINGW64 /c/git-test/git-learning (dev3)
$ git commit -m "added new file for dev3 branch"
[dev3 8859730] added new file for dev3 branch
 1 file changed, 1 insertion(+)
 create mode 100644 dev3.txt

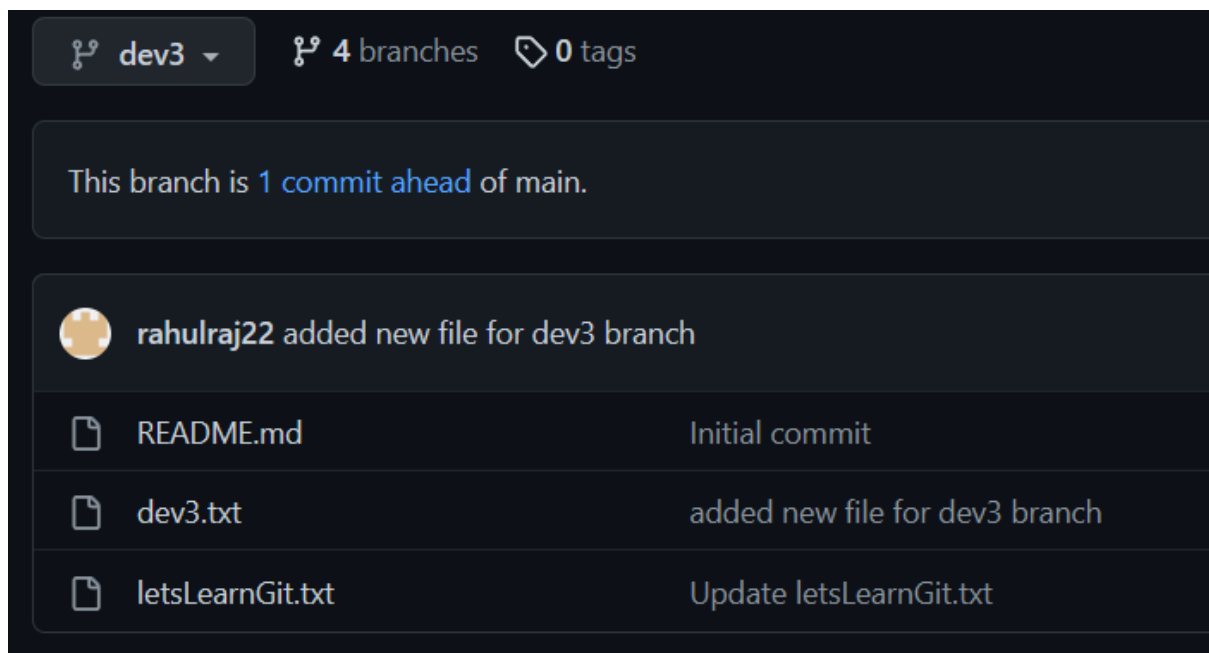
Lenovo@LAPTOP-CSGP8BLS MINGW64 /c/git-test/git-learning (dev3)
$ git push origin dev3
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 354 bytes | 118.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'dev3' on GitHub by visiting:
remote:   https://github.com/rahulraj22/git-learning/pull/new/dev3
remote:
To github.com:rahulraj22/git-learning.git
 * [new branch]      dev3 -> dev3

Lenovo@LAPTOP-CSGP8BLS MINGW64 /c/git-test/git-learning (dev3)
$

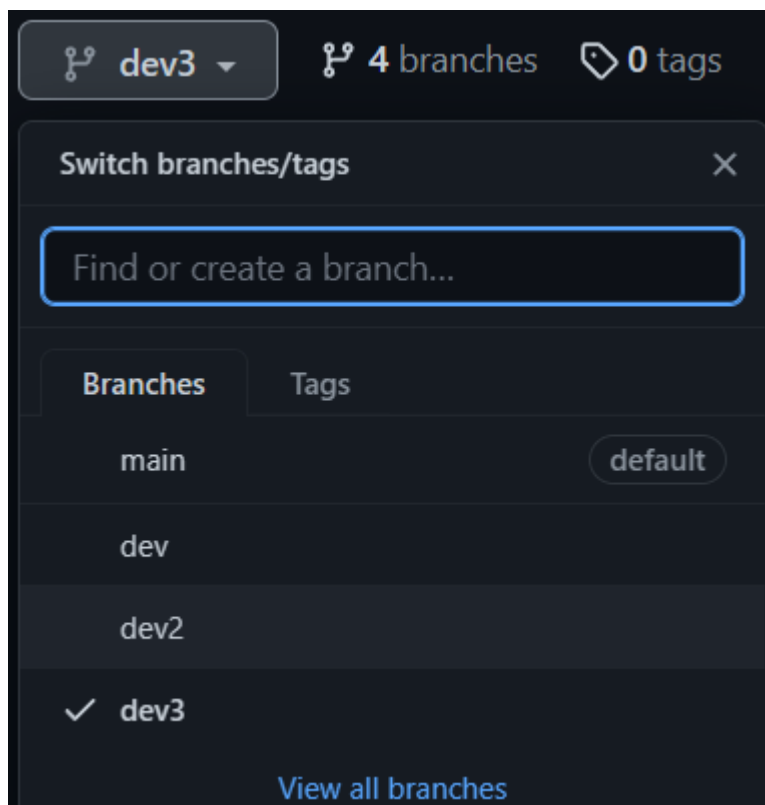
```

# Last command i.e **git push origin dev3** is used to push this branch into the remote repo.

# Fig:-



# Fig. showing all the branches:-



→ Now, different developers can work in all these branches, and final changes are merged with this main branch.

# If we want to create a new branch and switch immediately to that branch :-

**Command : git checkout -b <new branch name>**

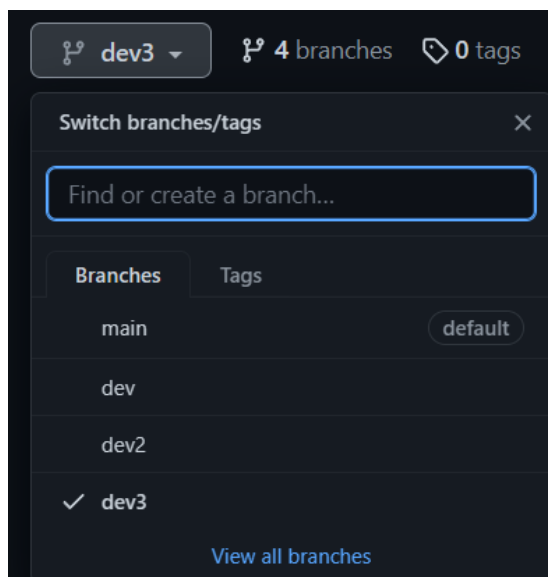
```
Lenovo@LAPTOP-CSGP8BLS MINGW64 /c/git-test/git-learning (main)
$ git checkout -b dev4
Switched to a new branch 'dev4'

Lenovo@LAPTOP-CSGP8BLS MINGW64 /c/git-test/git-learning (dev4)
$
```

```
Lenovo@LAPTOP-CSGP8BLS MINGW64 /c/git-test/git-learning (dev4)
$ git branch
  checkout
  dev
  dev2
  dev3
* dev4
  main

Lenovo@LAPTOP-CSGP8BLS MINGW64 /c/git-test/git-learning (dev4)
$
```

# But the dev4 branch is not present in the remote repository.



So, we need to push it.

Ans: we need to repeat those processes.

```

Lenovo@LAPTOP-CSGP8BLS MINGW64 /c/git-test/git-learning (dev4)
$ git add .

Lenovo@LAPTOP-CSGP8BLS MINGW64 /c/git-test/git-learning (dev4)
$ git status
On branch dev4
nothing to commit, working tree clean

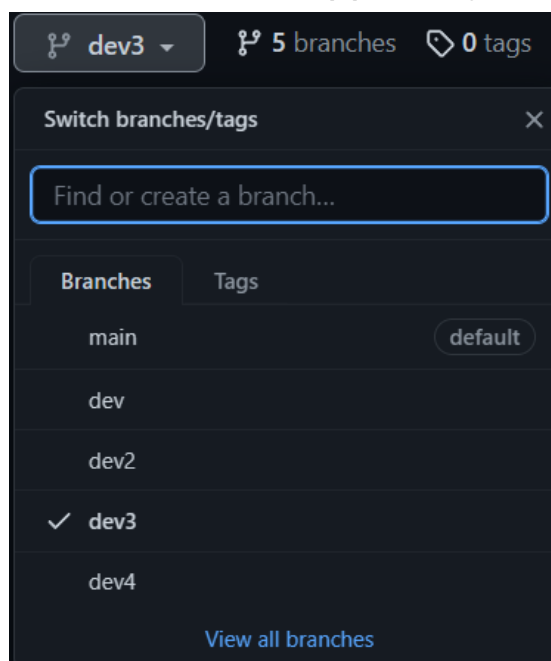
Lenovo@LAPTOP-CSGP8BLS MINGW64 /c/git-test/git-learning (dev4)
$ git commit -m "new branch dev4"
On branch dev4
nothing to commit, working tree clean

Lenovo@LAPTOP-CSGP8BLS MINGW64 /c/git-test/git-learning (dev4)
$ git push origin dev4
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'dev4' on GitHub by visiting:
remote:   https://github.com/rahulraj22/git-learning/pull/new/dev4
remote:
To github.com:rahulraj22/git-learning.git
 * [new branch]      dev4 -> dev4

Lenovo@LAPTOP-CSGP8BLS MINGW64 /c/git-test/git-learning (dev4)
$ |

```

# **dev4** also now appearing in our remote repository finally :-



-----X-----X-----

## # About git-github from angela course :-

→ **git init** : this will create a git local repository inside the current working directory.

After this we will see the **.git file** also.

→ **git status**: it will tell which r those file/folders which are untracked(in red color).

→ **git add <fileName>** : to bring the file to a staging area

→ after checking **git status** we see that files has turned green, since its been successfully staged/or taken photos.

→ now we are ready for commit: **git commit -m <message in present tense>**

→ **git log** : to get details with hash code , or what commit we have made and by whom, using this command.

→ **git diff hello2.txt**: it tells the difference between current hello2.txt file and past(first time committed) hello2.txt file. This happens if we tempered the code or the data present in hello2.txt.

→ **git checkout hello2.txt**: to revert back hello2.txt file to its initial commit looks(pahle kaise dikhta tha waisa hi agar chahiye to ye command likhna hoga).

MINGW64:/c/git-test

Lenovo@LAPTOP-CSGP8BLS MINGW64 ~

\$ cd /c/git-test/

Lenovo@LAPTOP-CSGP8BLS MINGW64 /c/git-test

\$ cat hello.txt

hello there how are you??

Lenovo@LAPTOP-CSGP8BLS MINGW64 /c/git-test

\$ git init

Initialized empty Git repository in C:/git-test/.git/

Lenovo@LAPTOP-CSGP8BLS MINGW64 /c/git-test (master)

\$ ls -a

./ ../ .git/ hello.txt

Lenovo@LAPTOP-CSGP8BLS MINGW64 /c/git-test (master)

\$ git status

On branch master

No commits yet

Untracked files:

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

hello.txt

nothing added to commit but untracked files present (use "git add" to track)

Lenovo@LAPTOP-CSGP8BLS MINGW64 /c/git-test (master)

\$ git add hello.txt

warning: LF will be replaced by CRLF in hello.txt.

The file will have its original line endings in your working directory

Lenovo@LAPTOP-CSGP8BLS MINGW64 /c/git-test (master)

\$ git status

On branch master

No commits yet

Changes to be committed:

(use "git rm --cached <file>..." to unstage)

new file: hello.txt

Lenovo@LAPTOP-CSGP8BLS MINGW64 /c/git-test (master)

\$ git commit -m "Intial Commit"

[master (root-commit) 66228bd] Intial Commit

1 file changed, 1 insertion(+)

create mode 100644 hello.txt

```

MINGW64:/c/git-test
Lenovo@LAPTOP-CSGP8BLS MINGW64 /c/git-test (master)
$ git log
commit 66228bd965e1b7e6516007866767e9e008449281 (HEAD -> master)
Author: Rahul Raj <raj.rahul@iitgn.ac.in>
Date: Sat Oct 15 01:08:11 2022 +0530

    Intial Commit

Lenovo@LAPTOP-CSGP8BLS MINGW64 /c/git-test (master)
$ touch hello1.txt

Lenovo@LAPTOP-CSGP8BLS MINGW64 /c/git-test (master)
$ touch hello2.txt

Lenovo@LAPTOP-CSGP8BLS MINGW64 /c/git-test (master)
$ cat > hello1.txt
ye capter 1 hai humara bhai

Lenovo@LAPTOP-CSGP8BLS MINGW64 /c/git-test (master)
$ cat hello1.txt
ye capter 1 hai humara bhai

Lenovo@LAPTOP-CSGP8BLS MINGW64 /c/git-test (master)
$ cat > hello2.txt
ye chapter 2 hai humara bhai

Lenovo@LAPTOP-CSGP8BLS MINGW64 /c/git-test (master)
$ cat hello2
cat: hello2: No such file or directory

Lenovo@LAPTOP-CSGP8BLS MINGW64 /c/git-test (master)
$ cat hello2.txt
ye chapter 2 hai humara bhai

Lenovo@LAPTOP-CSGP8BLS MINGW64 /c/git-test (master)
$ git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        hello1.txt
        hello2.txt

nothing added to commit but untracked files present (use "git add" to track)

Lenovo@LAPTOP-CSGP8BLS MINGW64 /c/git-test (master)
$ git add .
warning: LF will be replaced by CRLF in hello1.txt.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in hello2.txt.
The file will have its original line endings in your working directory

```



```
Lenovo@LAPTOP-CSGP8BLS MINGW64 /c/git-test (master)
```

```
$ git status
```

```
On branch master
```

```
Changes to be committed:
```

```
(use "git restore --staged <file>..." to unstage)
```

```
new file:   hello1.txt
```

```
new file:   hello2.txt
```

```
Lenovo@LAPTOP-CSGP8BLS MINGW64 /c/git-test (master)
```

```
$ git commit -m "Second Commit"
```

```
[master 9b3900e] Second Commit
```

```
2 files changed, 2 insertions(+)
```

```
create mode 100644 hello1.txt
```

```
create mode 100644 hello2.txt
```

```
Lenovo@LAPTOP-CSGP8BLS MINGW64 /c/git-test (master)
```

```
$ git log
```

```
commit 9b3900ee50c4ade0eacced51a372afdb23e4069f (HEAD -> master)
```

```
Author: Rahul Raj <raj.rahul@iitgn.ac.in>
```

```
Date: Sat Oct 15 01:14:33 2022 +0530
```

Second Commit

```
commit 66228bd965e1b7e6516007866767e9e008449281
```

```
Author: Rahul Raj <raj.rahul@iitgn.ac.in>
```

```
Date: Sat Oct 15 01:08:11 2022 +0530
```

Intial Commit

```
Lenovo@LAPTOP-CSGP8BLS MINGW64 /c/git-test (master)
```

```
$ git diff hello2.txt
```

```
warning: LF will be replaced by CRLF in hello2.txt.
```

```
The file will have its original line endings in your working directory
```

```
diff --git a/hello2.txt b/hello2.txt
```

```
index 4d83ccb..81f0d50 100644
```

```
--- a/hello2.txt
```

```
+++ b/hello2.txt
```

```
@@ -1,1 @@
```

```
-ye chapter 2 hai humara bhai
```

```
+aur maine gadbadi kardi re baba....:)
```

```
Lenovo@LAPTOP-CSGP8BLS MINGW64 /c/git-test (master)
```

```
$ git checkout hello2.txt
```

```
Updated 1 path from the index
```

```
Lenovo@LAPTOP-CSGP8BLS MINGW64 /c/git-test (master)
```

```
$
```

```

Lenovo@LAPTOP-CSGP8BLS MINGW64 ~
$ cd /c/Story/

Lenovo@LAPTOP-CSGP8BLS MINGW64 /c/Story (master)
$ git log
commit 51503761182fee8a84374be523faa040c7187168 (HEAD -> master)
Author: Rahul Raj <raj.rahul@iitgn.ac.in>
Date: Sat Oct 15 01:28:27 2022 +0530

    Initial Commit

Lenovo@LAPTOP-CSGP8BLS MINGW64 /c/Story (master)
$ git remote add origin https://github.com/rahulraj22/Story.git

Lenovo@LAPTOP-CSGP8BLS MINGW64 /c/Story (master)
$ git push -u origin master
info: please complete authentication in your browser...
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (5/5), 384 bytes | 76.00 KiB/s, done.
Total 5 (delta 0), reused 0 (delta 0), pack-reused 0
remote: error: GH007: Your push would publish a private email address.
remote: You can make your email public or disable this protection by visiting:
remote: http://github.com/settings/emails
To https://github.com/rahulraj22/Story.git
! [remote rejected] master -> master (push declined due to email privacy restrictions)
error: failed to push some refs to 'https://github.com/rahulraj22/Story.git'

```

```

Lenovo@LAPTOP-CSGP8BLS MINGW64 /c/Story (master)
$ git push -u origin master
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (5/5), 384 bytes | 76.00 KiB/s, done.
Total 5 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/rahulraj22/Story.git
* [new branch]      master -> master
Branch 'master' set up to track remote branch 'master' from 'origin'.

Lenovo@LAPTOP-CSGP8BLS MINGW64 /c/Story (master)
$ |

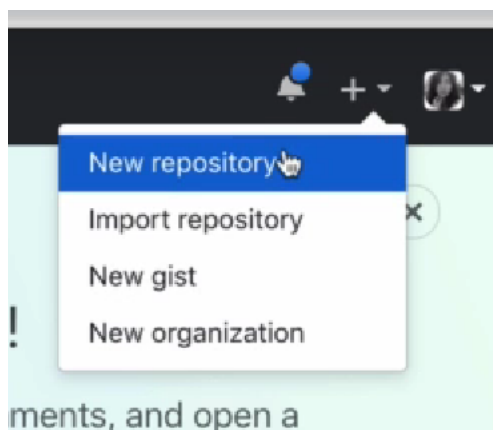
```

→ after removing the errors....

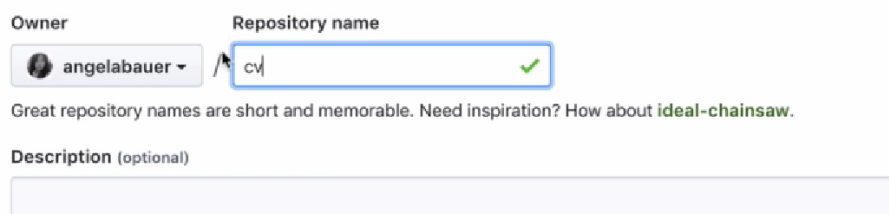
→ This will push the project using CLI git bash:-

## # How to publish our website ?

- first setup the github account.
- first click here to create a new **repository** .



- Now give name to the project: for eg. in my Cv website, i gave name of the project as **cv**

A screenshot of the 'Create new repository' form on GitHub. The 'Owner' field shows 'angelabauer' with a dropdown arrow. The 'Repository name' field contains 'cv' and has a green checkmark icon to its right. Below this, a message reads: 'Great repository names are short and memorable. Need inspiration? How about ideal-chainsaw.' The 'Description (optional)' field is an empty text box.

- Also give **readme** checked as shown:

## Create a new repository

A repository contains all the files for your project, including the revision history.

Owner

angelabauer

Repository name

cv

Great repository names are short and memorable. Need inspiration? How about **ideal-chainsaw**.

Description (optional)



Public

Anyone can see this repository. You choose who can commit.



Private

You choose who can see and commit to this repository.

☒ Initialize this repository with a README

This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

Add .gitignore: None

Add a license: None



Creating repository...

→ after clicking to the create repository button below: we will see this:

No description, website, or topics provided.

Edit

Manage topics

1 commit

1 branch

0 releases

1 contributor

Branch: master

New pull request

Create new file

Upload files

Find file

Clone or download

angelabauer Initial commit

Latest commit 450637e 25 seconds from now

README.md

Initial commit

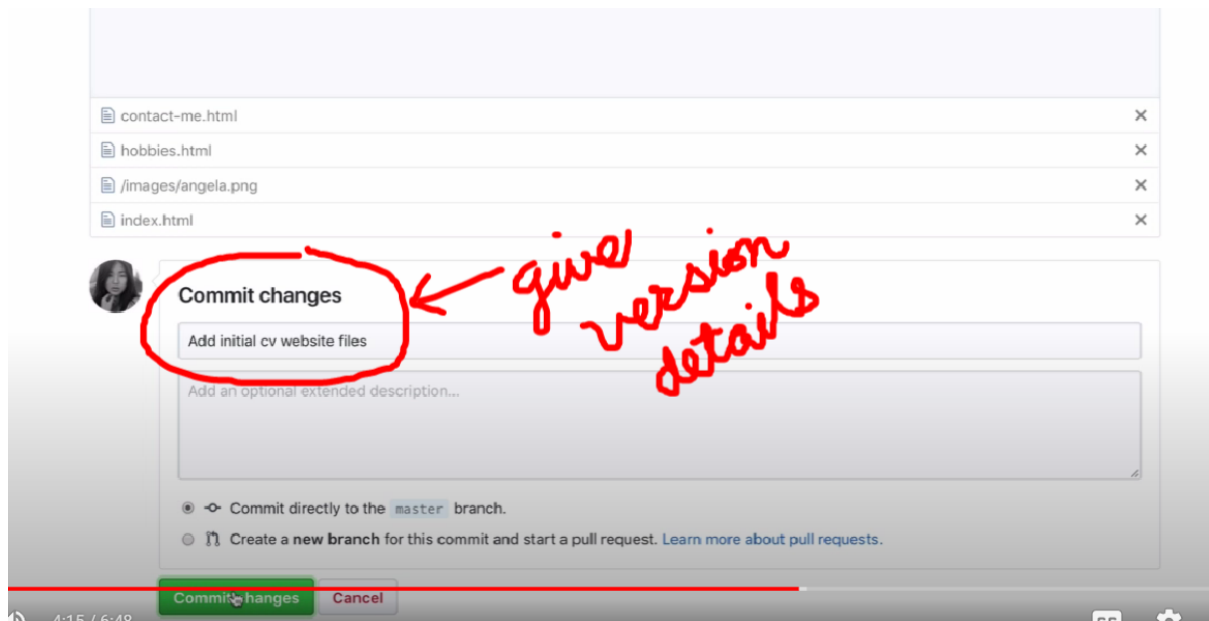
just now

README.md

CV

→ now click on the **Upload files** button and then either drag and drop that project folder or another way to upload is browse.

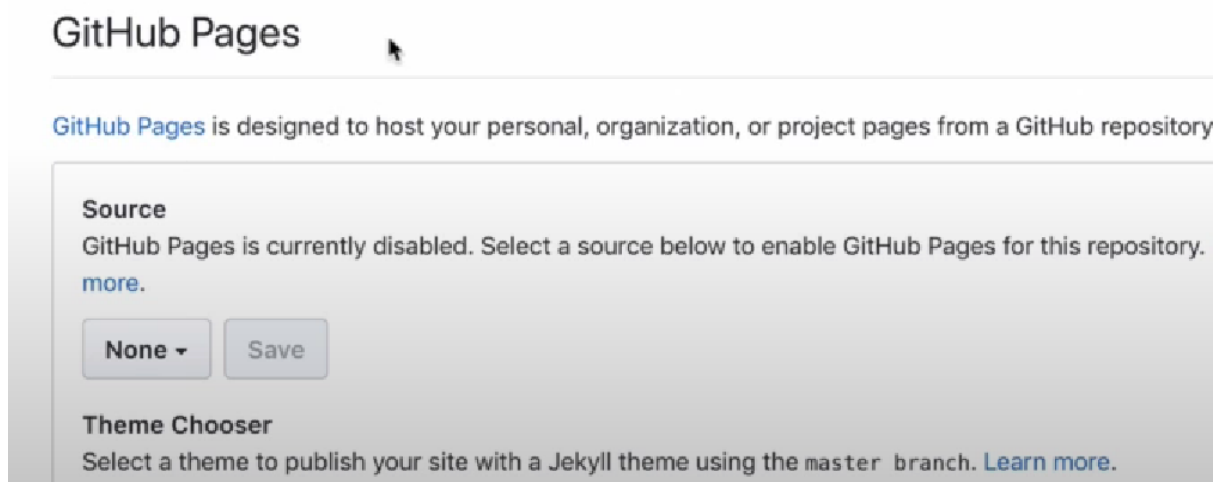
→ after uploading the files, do this:-



# In **Commit changes** we write about the version details of the project like after making changes to it. If it's an initial push or for a new project we write such as **Add initial cv website files**. And finally, click to commit this changes.

→ Now, after uploading the projects to github, we are ready to setup our github page, just click to **settings** and scroll down to see **GitHub Pages**(designed to host the pages online)

→ select **On branch master** after removing the **None** option.



# GitHub Pages

GitHub Pages is designed to host your personal, organization, or project

## Source

GitHub Pages is currently disabled. Select a source below to enable more.

None ▾

Save

### Select source

**master branch**

Use the master branch for GitHub Pages.

**master branch /docs folder**

Use only the /docs folder for GitHub Pages.

✓ **None**

Disable GitHub Pages.

The screenshot shows the GitHub repository settings page for a user named 'angelabauer'. The 'Source' tab is selected, and the 'master branch' is chosen as the source for GitHub Pages. Below the settings, there is a list of files in the repository, including 'images', 'README.md', 'contact-me.html', 'hobbies.html', and 'index.html'. The 'images' folder is highlighted, and its contents are listed below it.

Navigation: <> Code | 0 Issues | 0 Pull requests | 0 Projects | Wiki | Insights | Settings

No description, website, or topics provided. [Edit](#)

[Manage topics](#)

2 commits | 1 branch | 0 releases | 1 contributor

Branch: master ▾ [New pull request](#) [Create new file](#) [Upload files](#) [Find file](#) [Clone or download](#)

**angelabauer** Add initial cv website files Latest commit 4d42099 24 seconds from now

File	Commit	Time
images	Add initial cv website files	just now
README.md	Initial commit	a minute ago
contact-me.html	Add initial cv website files	just now
hobbies.html	Add initial cv website files	just now
index.html	Add initial cv website files	just now

→ after this click to save.