

## GithubBasics

14 July 2023 11:27

SO Now I am going with GITHUB stuff so in this section I had downloaded the slides pdf what ever is telling in this video for that downloaded pdf is there in the folder 5 okay ...  
Here in onenote folder creation is important if u forgot to do that it will now be saved so where which folder you are saving that should not be deleted that is what I understood here so now Move on to the topic

Github is a hosting platform for git repositories. You can put your own Git repos on Github and access them from anywhere and share them with people around the world.Beyond hosting repos, Github also provides additional collaboration features that are not native to Git (but are super useful). Basically, Github helps people share and collaborate on repos.



I'll try to write as much as less notes so that you can refer video better mostly I'll try to do practical demo involving my type of example by comparing the video which I'm watching ok  
The home page link is <https://github.com/>

Some more links telling why we should use github okay ...this is 3rd video part of this section and  
<https://github.com/facebook/react>

This link shows lot of contributors for the project u can check it by opening above link okay

Git Has become an home of open source projects in Internet

Now let us go with the actual demo of cloning

So far we've created our own Git repositories from scratch, but often we want to get a local copy of an existing repository instead. To do this, we can clone a remote repository hosted on Github or similar websites. All we need is a URL that we can tell Git to clone for use.

To clone a repo, simply run `git clone <url>`. Git will retrieve all the files associated with the repository and will copy them to your local machine. In addition, Git initializes a new repository on your machine, giving you access to the full Git history of the cloned project.

<https://github.com/gabrielecirulli/2048>

Here in this url some game is there now I am Going to clone it  
So first create a folder github on desktop and it Should not be repository means you have not initialised init Method into it go to that folder github and do the following things of how to implement cloning okay

```
MINGW64:/c/Users/raghavendra/Desktop/github
raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~
$ cd desktop
raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/Desktop
$ cd github
raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/Desktop/github
$ git status
fatal: not a git repository (or any of the parent directories): .git
raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/Desktop/github
$
```

So till now checked all this

Next

From here I clone and will write the following command

```
raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop/github
$ git clone https://github.com/gabrielecirulli/2048.git
Cloning into '2048'...
remote: Enumerating objects: 1315, done.
remote: Total 1315 (delta 0), reused 0 (delta 0), pack-reused 1315
Receiving objects: 100% (1315/1315), 623.29 KiB | 248.00 KiB/s, done.
Resolving deltas: 100% (765/765), done.
```

```
raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop/github
$
```

Now u go to github folder and see there now I will check what is there in folder and if I go into the folder and check it what is there in that

```
raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop/github
$ ls
2048/
raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop/github
$ cd 2048
raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop/github/2048 (master)
$ ls
CONTRIBUTING.md LICENSE.txt README.md Rakefile favicon.ico index.html js/ meta/ style/
t|raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop/github/2048 (master)
$ |
```

So you can see here okay And it is a git repo and see all commits of it b typing git log okay

```
raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop/github/2048 (master)
$ git status
On branch master
Your branch is up to date with 'origin/master'.

nothing to commit, working tree clean

raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop/github/2048 (master)
$ git log
commit fc1ef4fe5a5fcce7590f3e4c187c75980b353f (HEAD -> master, origin/master, origin/HEAD)
Author: Gabriele cirulli <gabrielecirulli@users.noreply.github.com>
Date:   Sat Oct 27 18:25:12 2018 +0200

    Remove Adsense code

commit ffa8559d622e6c1b216e43de969f037204cdb102
Author: Gabriele cirulli <gabrielecirulli@users.noreply.github.com>
Date:   Sat Oct 27 13:21:45 2018 +0200

    Add Adsense code

commit ac03b1f01628038039b74b67f2e284b233bd143e
Merge: d39dcf7 8ae5cccd
Author: Anna Harren <iirelu@users.noreply.github.com>
Date:   Fri Oct 6 07:36:36 2017 +0200

    Merge pull request #463 from marvin08/master

    Update main.scss (Add missing semicolon)

commit 8ae5cccd1e0bdbe1298e5da4f023b14c0eaec25da
Author: Akshay Shrimali <akshaytryingtocode@gmail.com>
Date:   Sun Oct 1 00:31:17 2017 +0530
    ....skipping...
```

Type the start .

```
raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop/github/2048 (master)
$ start .
raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop/github/2048 (master)
$ |
```

You can see the folder opened and open index.html file okay ..

I've gone to main css file and change the background as olive And now I run the website of Index Html file some change I will find it but it is in my local repository .

```
MINGW64:/c/Users/raghavendra/Desktop/github/2048/2048
raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop/github/2048 (master)
$ git status
On branch master
Your branch is up to date with 'origin/master'.

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:   style/main.css

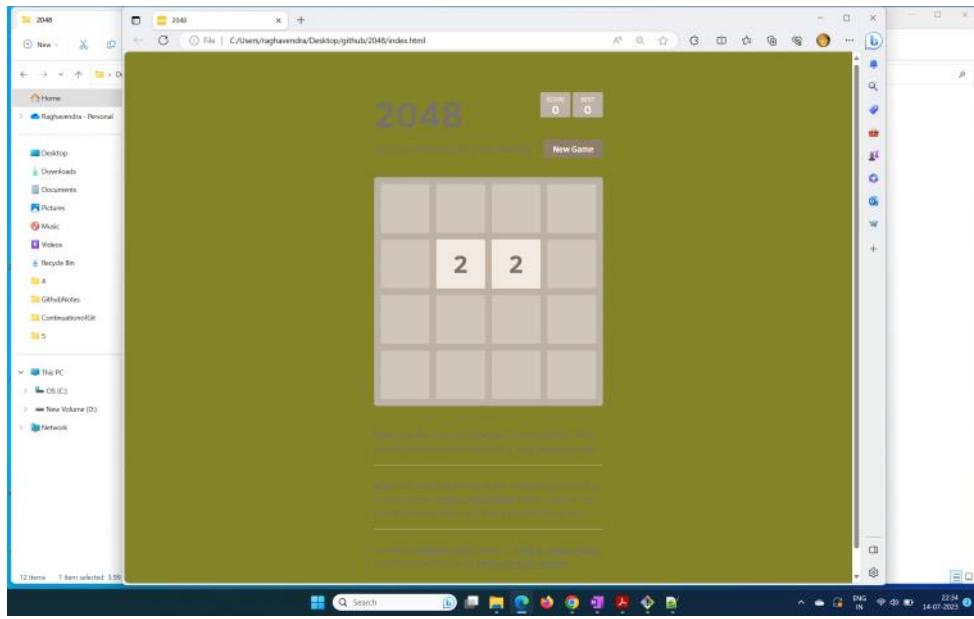
no changes added to commit (use "git add" and/or "git commit -a")

raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop/github/2048 (master)
$ git add .

raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop/github/2048 (master)
$ git commit -m "main .css file has been modified "
[master 0b3c8b9] main .css file has been modified
 1 file changed, 1 insertion(+), 1 deletion(-)

raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop/github/2048 (master)
$ |
```

So after doing this and check the change okay



So now it looks like this okay

This is a git command it is not a github command so this is all about git clone usage next is

# Permissions?

**Anyone can clone a repository from Github**, provided the repo is public. You do not need to be an owner or collaborator to clone the repo locally to your machine. You just need the URL from Github.



Pushing up your own changes to the Github repo...that's another story entirely! You need permission to do that!

If it is a private you don't even see the url in github. But even though it is public you cannot push your changes into some other's repository in github. If it is public only you can clone it, not push by taking clone doing changes and again low pushing into his repository is not possible. Make a note of it.

I want to clone not only github but also other online repositories like gitlab and bitbucket etc.

So I am now going to some url like this <https://gitlab.com/gitlab-tests/sample-project/>

Here it is not compulsory to clone by clicking the clone button just the url also you can take it. As of how I have taken here:

```
MINGW64:/c/Users/raghavendra/Desktop/github/sample-project
raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/Desktop/github
$ git clone https://gitlab.com/gitlab-tests/sample-project/
Cloning into 'sample-project'...
warning: redirecting to https://gitlab.com/gitlab-tests/sample-project.git/
remote: Enumerating objects: 12, done.
remote: Total 12 (delta 0), reused 0 (delta 0), pack-reused 12
Receiving objects: 100% (12/12), 30.73 KiB | 10.24 MiB/s, done.
Resolving deltas: 100% (1/1), done.

raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/Desktop/github
$ cd sample-project

raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/Desktop/github/sample-project (master)
$ git status
On branch master
Your branch is up to date with 'origin/master'.

nothing to commit, working tree clean

raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/Desktop/github/sample-project (master)
$ git log
commit 90c439634077a85bcf42d38c2c79cd94664a94ad (HEAD -> master, origin/master, origin/HEAD)
Author: Marcia Ramos <marcia@gitlab.com>
Date:   Wed Apr 29 11:32:58 2020 +0000

    Update README.md

commit 3ab915edfd7c6dc8d89c78cc13133ff1df41472
Author: Marcia Ramos <marcia@gitlab.com>
Date:   Wed Apr 29 11:27:23 2020 +0000

    Add screenshot

commit 1818c9036e0ad39cd8ff764c2a48643dac6e114e
Author: Marcia Ramos <marcia@gitlab.com>
Date:   Wed Apr 29 11:24:21 2020 +0000

    Add new file

commit 46081376f47e386db96af3a58aa704a2279c4375
Author: Marcia Ramos <marcia@gitlab.com>
Date:   Wed Apr 29 11:23:42 2020 +0000

    Initial commit

raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/Desktop/github/sample-project (master)
$
```

Now let us go with the github setup which you have already done. I'll just check it by seeing the commands from the menu whether I have done it earlier when I created github account. As I had already done it, it is showing me to it.

```
raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/Desktop/github/sample-project (master)
$ git config user.email
raghu070882@gmail.com
```

```
raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/Desktop/github/sample-project (master)
$ git config user.name
raghu
```

```
raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/Desktop/github/sample-project (master)
```

So whatever is there, it is showing it to me okay...so after doing this I need to register okay...here registering why I am doing because every time I don't want to enter user credentials of github so if I register with ssh key automatically it will push my code without asking for username and password of github okay..

Some url they have given I am just usng it for this video okay <https://docs.github.com/en/authentication/connecting-to-github-with-ssh>

So Now let us create a repository in git hub How I can host my folder or website anything into the github repository

# How Do I Get My Code On Github?



Option 1

## Option 1: Existing Repo

If you already have an existing repo locally that you want to get on Github...

- Create a new repo on Github
- Connect your local repo (add a remote)
- Push up your changes to Github



Option 2:

## Option 2: Start From Scratch

If you haven't begun work on your local repo, you can...

- Create a brand new repo on Github
- Clone it down to your machine
- Do some work locally
- Push up your changes to Github



Now I'm going to try option one as I am already having one folder `githubsample`

And in that folder do the following things which u have done to timetravel folder and this is just a demo folder for doing so me tasks

```
raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~  
$ cd desktop
```

```
raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop  
$ cd githubsample
```

```
raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop/githubsample  
$ git init  
Initialized empty Git repository in C:/Users/raghavendra/Desktop/githubsample/.git/
```

```
raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop/githubsample (master)  
$ git status  
On branch master
```

No commits yet

nothing to commit (create/copy files and use "git add" to track)

```
raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop/githubsample (master)  
$ touch file1.txt file2.txt file3.txt
```

```
raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop/githubsample (master)
$ git status
On branch master
```

No commits yet

Untracked files:  
(use "git add <file>..." to include in what will be committed)  
  file1.txt  
  file2.txt  
  file3.txt

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

```
raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop/githubsample (master)
$ git add .
```

```
raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop/githubsample (master)
$ git status
On branch master
```

No commits yet

Changes to be committed:  
(use "git rm --cached <file>..." to unstage)  
  new file: file1.txt  
  new file: file2.txt  
  new file: file3.txt

```
raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop/githubsample (master)
$ git commit -m "three files added"
[master (root-commit) f17ec79] three files added
3 files changed, 0 insertions(+), 0 deletions(-)
create mode 100644 file1.txt
create mode 100644 file2.txt
create mode 100644 file3.txt
```

```
raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop/githubsample (master)
$ git status
On branch master
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: file1.txt
```

no changes added to commit (use "git add" and/or "git commit -a")

```
raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop/githubsample (master)
$ git status
On branch master
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: file1.txt
    modified: file2.txt
    modified: file3.txt
```

no changes added to commit (use "git add" and/or "git commit -a")

```
raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop/githubsample (master)
$ git add .
raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop/githubsample (master)
$ git commit -m "data added in three files"
[master 89ddb03] data added in three files
3 files changed, 3 insertions(+)
```

```
raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop/githubsample (master)
$ git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    images/
nothing added to commit but untracked files present (use "git add" to track)
```

```
raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop/githubsample (master)
$ git add .
```

```
raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop/githubsample (master)
$ git commit -m "Images folder added and images added into it"
[master dc4349f] Images folder added and images added into it
2 files changed, 0 insertions(+), 0 deletions(-)
create mode 100644 images/Mylimage.jpeg
create mode 100644 images/merlin.gif
```

```
raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop/githubsample (master)
$ git log
commit 89ddb03ac3c5f7614302a007ec69a5b07bfb0b4
Author: raghu <raghu070882@gmail.com>
Date:  Sat Jul 15 12:09:13 2023 +0530
```

Images folder added and images added into it

```
commit 89ddb03ac3c5f7614302a007ec69a5b07bfb0b4
Author: raghu <raghu070882@gmail.com>
Date:  Sat Jul 15 12:07:56 2023 +0530
```

data added in three files

```
commit f17ec79acf570934059d781967d5a242205b7db
Author: raghu <raghu070882@gmail.com>
Date:  Sat Jul 15 12:06:17 2023 +0530
```

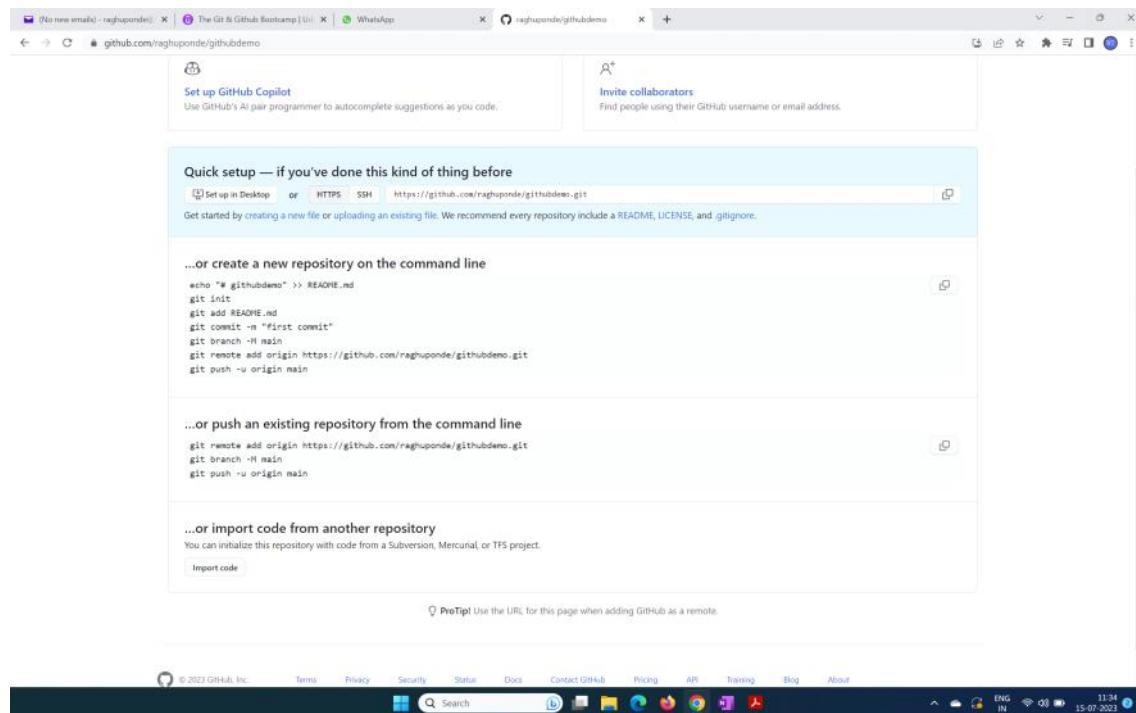
three files added

: So here you are facing problem to come out of the prompt just press q that is enough okay

with the following steps which have done earlier and now I want to put this folder in my repository of github

first create a repository It should be empty and you can give any name to the repository

As I am having a github account I am creating a new empty repository with the name `githubdemo` something like that  
Make it public and just create a repository



So it will look like this..

**Remote**

Before we can push anything up to GitHub, we need to tell Git about our remote repository on GitHub. We need to setup a "destination" to push up to.

In Git, we refer to these "destinations" as remotes. Each remote is simply a URL where a hosted repository lives.

## Viewing Remotes

To view any existing remotes for your repository, we can run `git remote` or `git remote -v` (verbose, for more info)

```
git remote -v
```

This just displays a list of remotes. If you haven't added any remotes yet, you won't see anything!

Now I had created a `githubdemo` repository in cloud and now in `githubsample` I have some works done earlier so type above command and see the status once Means you are checking remote over here okay

```
MINGW64:/c/Users/raghavendra/Desktop/githubsample
$ git remote
raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/Desktop/githubsample (master)
$
```

So nothing it is giving so my remote for this local repository is not yet set okay ..

But when I do to repos 2048 git remote it gives me origin as result okay ...and type same command with git remote and with -v it gives the result like this okay

```
MINGW64:/c/Users/raghavendra/Desktop/github/2048
$ git remote
raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/Desktop/githubsample (master)
$ cd ..
raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/Desktop/githubsample (master)
$ cd github
raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/Desktop/github
$ cd 2048
raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/Desktop/github/2048 (master)
$ git remote
origin https://github.com/gabrielecirulli/2048.git (fetch)
origin https://github.com/gabrielecirulli/2048.git (push)
raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/Desktop/github/2048 (master)
$ |
```

I Have not configured the remote because it is somebodys repository

And to my folder githubsample remote also nothinng it shows as its remote is not yet configured okay

```
MINGW64:/c/Users/raghavendra/Desktop/githubsample
$ cd ..
raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/Desktop/github (master)
$ cd ..
raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/Desktop/github
$ cd githubsample
raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/Desktop/githubsample (master)
$ git remote
raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/Desktop/githubsample (master)
$ git remote -v
raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/Desktop/githubsample (master)
$ |
```

Same thing if I am doing to gitlab repositories it shows me like this

```
MINGW64:/c/Users/raghavendra/Desktop/github/sample-project
$ cd ..
raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/Desktop/githubsample (master)
$ cd github
raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/Desktop/github
$ cd sam*
raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/Desktop/github/sample-project (master)
$ git remote -v
origin https://gitlab.com/gitlab-tests/sample-project/ (fetch)
origin https://gitlab.com/gitlab-tests/sample-project/ (push)
raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/Desktop/github/sample-project (master)
$ |
```

So for them it is configured okay

# Adding A New Remote

A remote is really two things: a URL and a label.  
To add a new remote, we need to provide both to Git.

```
❯ git remote add <name> <url>
```

# Adding A New Remote

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

Okay Git, anytime I use the name "origin", I'm referring to this particular Github repo URL.

## Origin?

Origin is a conventional Git remote name, but it is not at all special. It's just a name for a URL.

When we clone a Github repo, the default remote name setup for us is called origin. You can change it. Most people leave it.



In the website where I have created `gitubdemo` empty repository there are some instructions have already been given second in structions I am talking about we can follow that one

The screenshot shows a browser window with three tabs open: 'No new emails' - raghuponde@gmail.com, 'The Git & GitHub Bootcamp | Udemy', and 'raghuponde/githubdemo'. The main content area displays a 'Quick setup' guide for creating a new repository. It includes three code snippets:

- ...or create a new repository on the command line**

```
echo "# githubdemo" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/raghuponde/githubdemo.git
git push -u origin main
```
- ...or push an existing repository from the command line**

```
git remote add origin https://github.com/raghuponde/githubdemo.git
git branch -M main
git push -u origin main
```
- ...or import code from another repository**

You can initialize this repository with code from a Subversion, Mercurial, or TFS project.

A red arrow points to the second code snippet under '...or push an existing repository from the command line'.



You can actually directly copy the whole command from the line there

The terminal window shows the following command being run:

```
raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop
$ cd gitbooksample
raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop/gitbooksample (master)
$ git remote add origin https://github.com/raghuponde/githubdemo.git
```

Now refresh the website .repo and see whether things have come or not okay ..as I had not pushed it we will not see the things uploaded okay

## Pushing

Now that we have a remote set up, let's push some work up to Github! To do this, we need to use the `git push` command.

We need to specify the remote we want to push up to AND the specific local branch we want to push up to that remote.

git push <remote> <branch>

## An Example

`git push origin master` tells git to push up the master branch to our origin remote.

git push origin master

```

raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop/githubsample (master)
$ git push origin master
Enumerating objects: 13, done.
Counting objects: 100% (13/13), done.
Delta compression using up to 20 threads
Compressing objects: 100% (9/9), done.
Writing objects: 100% (13/13), 42.70 KiB | 14.23 MiB/s, done.
Total 13 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), done.
To https://github.com/raghuponde/githubdemo.git
 * [new branch]      master -> master

raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop/githubsample (master)
$ |

```

Now we can see like this

The screenshot shows a GitHub repository named 'githubdemo'. The repository details are as follows:

- Branches:** master (1 branch)
- Tags:** 0 tags
- Commits:** 3 commits by 'raghuponde' (41 minutes ago)
- Activity:** 0 stars, 1 watching, 0 forks
- Releases:** No releases published. Create a new release.
- Packages:** No packages published. Publish your first package.



Here we can push any branch its not compulsory that we have to push master branch only

Now create a new branch like this and in that delete Images folder and again push that thing into gitgub again and to push u can be in any branch it is not compulsory that u should be in that branchonly which u are pushing let us see the demo now on this

```

MINGW64:/c/Users/raghavendra/Desktop/githubsample
* [new branch]      master -> master

raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/Desktop/githubsample (master)
$ git switch -c empty
Switched to a new branch 'empty'

raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/Desktop/githubsample (empty)
$ ls
file1.txt file2.txt file3.txt images/

raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/Desktop/githubsample (empty)
$ rm -rf images

raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/Desktop/githubsample (empty)
$ ls
file1.txt file2.txt file3.txt

raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/Desktop/githubsample (empty)
$ git status
On branch empty
Changes not staged for commit:
  (use "git add" and/or "git commit -a")
    (use "git add/rm <file>..." to update what will be committed)
    (use "git restore <file>..." to discard changes in working directory)
      deleted:   images/MyImage.jpeg
      deleted:   images/merlin.gif

no changes added to commit (use "git add" and/or "git commit -a")

raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/Desktop/githubsample (empty)
$ git add .

raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/Desktop/githubsample (empty)
$ git commit -m "removed the folder with images"
[empty e7a296f] removed the folder with images
 2 files changed, 0 insertions(+), 0 deletions(-)
 delete mode 100644 images/MyImage.jpeg
 delete mode 100644 images/merlin.gif

raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/Desktop/githubsample (empty)
$ |

```

```

MINGW64:/c/Users/raghavendra/Desktop/githubsample
raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/Desktop/githubsample (empty)
$ git log
commit e7a296fbe5f3a416342e868c977f131950a64d54 (HEAD -> empty)
Author: raghu <raghu070882@gmail.com>
Date:   Sat Jul 15 13:06:08 2023 +0530

    removed the folder with images

commit dc4349f5d366600735085323dd37feb35555989 (origin/master, master)
Author: raghu <raghu070882@gmail.com>
Date:   Sat Jul 15 12:09:13 2023 +0530

    Images folder added and images added into it

commit 89dbb03ac3c5f7614302a007ec69a5b07bfb0b4
Author: raghu <raghu070882@gmail.com>
Date:   Sat Jul 15 12:07:56 2023 +0530

    data added in three files

commit f17ec79acf570934059d781967d5a242205b7db
Author: raghu <raghu070882@gmail.com>
Date:   Sat Jul 15 12:06:17 2023 +0530

    three files added

raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/Desktop/githubsample (empty)
$ git push origin empty
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 20 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (2/2), 227 bytes | 227.00 KiB/s, done.
Total 2 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
remote:
remote: Create a pull request for 'empty' on GitHub by visiting:
remote:     https://github.com/raghuponde/githubdemo/pull/new/empty
remote:
To https://github.com/raghuponde/githubdemo.git
 * [new branch]      empty -> empty

raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/Desktop/githubsample (empty)
$
```

Now again refresh the github repository

The screenshot shows a GitHub repository page for 'raghuponde/githubdemo'. The repository has 2 branches and 0 tags. The master branch is selected. A commit from 'raghuponde' titled 'Images folder added and images added into it' is shown, with 3 commits. The commit message includes 'data added in three files'. The repository has 0 stars, 1 watching, and 0 forks. The 'About' section notes 'No description, website, or topics provided.'



Now you can see two branches and if you go in that branch its code at that time I can see and its commits and all we can see..

Check once pdf slides also for better understanding of what you are doing how you are creating a new branch as a new feature and adding it okay to local and then pushing it to GitHub okay

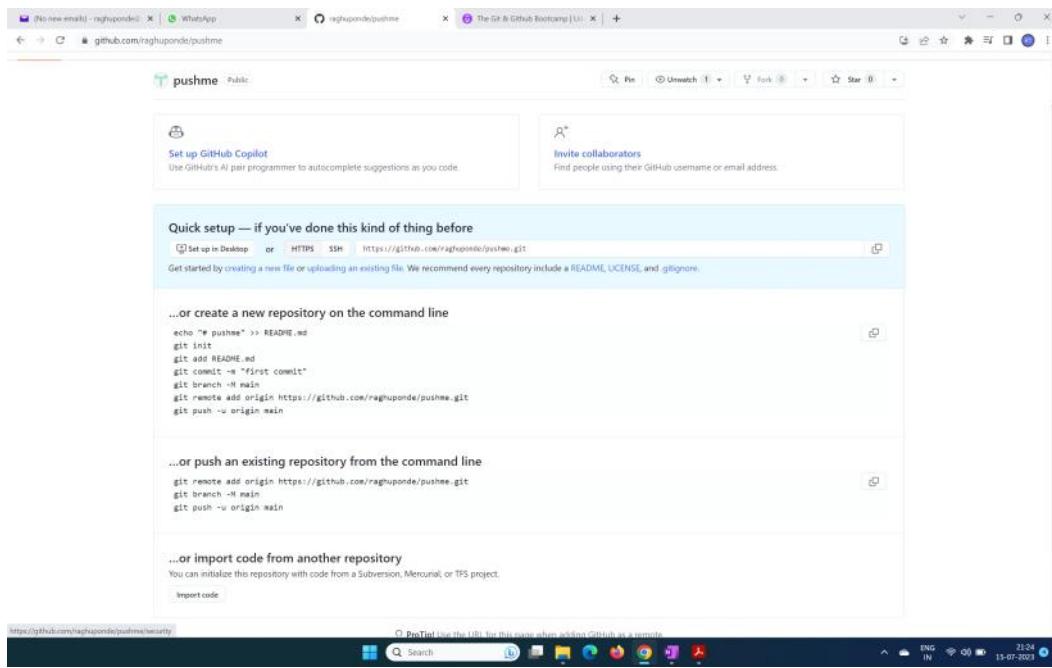
You can do another thing is that again go to the Master Branch create one new file check the status do the commit and finally push again but with master branch so master to master child to child you have to be connected from local to GitHub server

Check once 101 video for more practice here what is done is that simultaneously I had done 2 changes one in master branch and then one in empty branch means suppose I had done is

Removed images from master and then added one text file to it so after doing all changes also later on from any branch you can push just specific to which branch is getting pushed and So just refer 101 video for more idea into it

Before doing this above implementation let us now basically some new stuff ready

Create a new repository with the name Pushme



In desktop create a new folder pushing and make it init and add one file over there candy.txt and git commit and all do it so set of commands in gitbash as provided below .

```
MINGW64/c/Users/raghavendra/Desktop/pushing
raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~
$ cd desktop
raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop
$ cd pushing
raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop/pushing
$ git init
Initialized empty Git repository in c:/users/raghavendra/Desktop/pushing/.git/
raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop/pushing (master)
$ git status
On branch master

No commits yet

nothing to commit (create/copy files and use "git add" to track)

raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop/pushing (master)
$ touch candy.txt
raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop/pushing (master)
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    candy.txt

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

raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop/pushing (master)
$ git add .

raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop/pushing (master)
$ git commit -m "candy file of text created"
[master (root-commit) aa8fb4a] candy file of text created
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 candy.txt
raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop/pushing (master)
```

```
raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop/pushing (master)
$ git remote -v
```

```
raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop/pushing (master)
$ git remote add origin https://github.com/raghuponde/pushme.git
```

```
raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop/pushing (master)
$ git push origin master
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 218 bytes | 218.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/raghuponde/pushme.git
 * [new branch]  master -> master
```

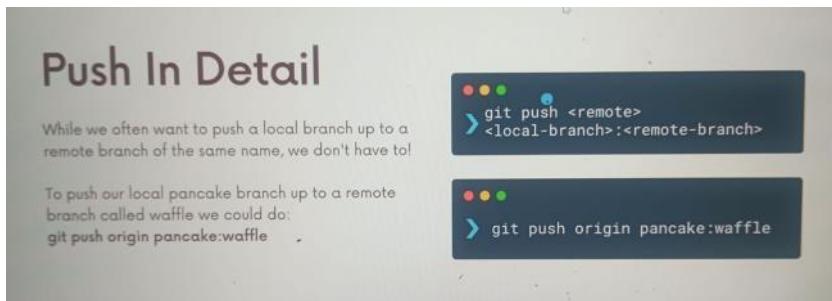
Now refresh the page and you can see the output okay

The screenshot shows a GitHub repository page for 'pushme'. The repository has 1 branch and 0 tags. It contains one file, 'candy.txt', which was created by 'raghuponde' 6 minutes ago. The repository has 0 stars, 1 watch, and 0 forks. There are sections for About, Releases, and Packages.



So u can see master to master from local to cloud okay

Now I will be having a closer look at the git push right here I want to push from local branch data into remote some other branch okay



```

MINGW64:/c/Users/raghavendra/Desktop/pushing
raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/Desktop/pushing (master)
$ touch flowers.txt
raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/Desktop/pushing (master)
$ git add .
raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/Desktop/pushing (master)
$ git commit -m "create flowers file"
[master 4ded9fe] create flowers file
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 flowers.txt

raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/Desktop/pushing (master)
$ git push origin master

Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 20 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (2/2), 245 bytes | 245.00 KiB/s, done.
Total 2 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/raghuponde/pushme.git
 aa8fba4..4ded9fe master -> master

raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/Desktop/pushing (master)
$
```

Now again it will not create separate branch okay ...

The screenshot shows a GitHub repository page for 'pushme'. The repository has 1 branch (master) and 0 tags. It contains two files: 'candy.txt' and 'flowers.txt'. The 'About' section notes 'No description, website, or topics provided.' and shows 0 stars, 1 watching, and 0 forks. The 'Releases' section indicates 'No releases published'.



So I want to push to some other branch so let us have a demo now

```

raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop/pushing (master)
$ git switch -c cats
Switched to a new branch 'cats'

raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop/pushing (cats)
$ touch cats.txt

raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop/pushing (cats)
$ git add cats.txt

raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop/pushing (cats)
$ git commit -m "Create cats file"
[cats 5522547] Create cats file
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 cats.txt

raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop/pushing (cats)
$ |

```

So above fresh cat branch is created

Now if I type this command

```

raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop/pushing (cats)
$ git push origin cats
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 20 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (2/2), 249 bytes | 249.00 KiB/s, done.
Total 2 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'cats' on GitHub by visiting:
remote:   https://github.com/raghuponde/pushme/pull/new/cats
remote:
To https://github.com/raghuponde/pushme.git
 * [new branch]      cats -> cats

raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop/pushing (cats)
$ |

```

So a branch is created and there I can see cats file okay

The screenshot shows a GitHub repository page for 'pushme' by 'raghuponde'. The repository has 2 branches and 0 tags. A recent commit from 'raghuponde' titled 'Create cats file' was made 3 minutes ago. The commit details show three files: candy.txt, cats.txt, and flowers.txt. The repository has 0 stars, 1 watching, and 0 forks. There are sections for About, Releases, and Packages.

[View code](#) [Compare & pull request](#) [About](#) [Activity](#) [Contribute](#) [Releases](#) [Packages](#)



So some how other files also have come but cats is there in cats branch only okay

```
raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop/pushing (cats)
$ git push origin cats
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 20 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (2/2), 249 bytes | 249.00 KiB/s, done.
Total 2 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: create a pull request for 'cats' on GitHub by visiting:
remote:   https://github.com/raghuponde/pushme/pull/new/cats
remote:
To https://github.com/raghuponde/pushme.git
 * [new branch]      cats -> cats

raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop/pushing (cats)
$ git push origin cats:master
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/raghuponde/pushme.git
 4ded9fe..5522547  cats -> master

raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop/pushing (cats)
$ |
```

So u can see second command is from cats to master



So in master branch also I can see now cats file okay

Now just go through 103 video which is on upstream u option which means it will remember and next time onwards can say just git push only okay

Now I am moving to 104 video which is second option okay

## Option 2: Start From Scratch

If you haven't begun work on your local repo, you can...

- Create a brand new repo on Github
- Clone it down to your machine
- Do some work locally
- Push up your changes to Github

Now create a new repository chickensdemo in github now

And then on desktop create a folder cloning and by going there into folder from gitbash do cloning like this

```
MINGW64:/c/Users/raghvendra/Desktop/cloning/chickensdemo
raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/Desktop/pushing (cats)
$ cd ..
raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/Desktop
$ mkdir cloning
raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/Desktop
$ cd cloning
raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/Desktop/cloning
$ git clone https://github.com/raghuponde/chickensdemo.git
Cloning into 'chickensdemo'...
warning: You appear to have cloned an empty repository.

raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/Desktop/cloning
$ ls
chickensdemo/
raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/Desktop/cloning
$ cd chickensdemo
bash: cd: chickensdemo: No such file or directory
raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/Desktop/cloning
$ cd chickensdemo
raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/Desktop/cloning/chickensdemo (main)
```

so I am going into that folder which is created from clone and write the following commands to see what and all is needed okay

```

raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop/cloning
$ cd chickensdemo
raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop/cloning/chickensdemo (main)
$ ls
raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop/cloning/chickensdemo (main)
$ git status
on branch main
No commits yet
nothing to commit (create/copy files and use "git add" to track)
raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop/cloning/chickensdemo (main)
$ git remote -v
origin https://github.com/raghuponde/chickensdemo.git (fetch)
origin https://github.com/raghuponde/chickensdemo.git (push)
raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop/cloning/chickensdemo (main)
$
```

```

raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop/cloning/chickensdemo (main)
$ touch chickens.txt
```

```

raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop/cloning/chickensdemo (main)
$ start .
```

```

raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop/cloning/chickensdemo (main)
$ open .
bash: open: command not found
```

```

raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop/cloning/chickensdemo (main)
$ code .
```

```

raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop/cloning/chickensdemo (main)
$ git status
On branch main
```

```
No commits yet
```

```

Untracked files:
(use "git add <file>..." to include in what will be committed)
  chickens.txt
```

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

```

raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop/cloning/chickensdemo (main)
$ git add .
```

```

raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop/cloning/chickensdemo (main)
$ git commit -m "added chicken file and data also"
[main (root-commit) 023ecd7] added chicken file and data also
 1 file changed, 1 insertion(+)
 create mode 100644 chickens.txt
```

```

raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop/cloning/chickensdemo (main)
$ git push origin master
error: src refspec master does not match any
error: failed to push some refs to 'https://github.com/raghuponde/chickensdemo.git'
```

```

raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop/cloning/chickensdemo (main)
$ git push origin main
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 240 bytes | 240.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/raghuponde/chickensdemo.git
 * [new branch]  main -> main
```

```

raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop/cloning/chickensdemo (main)
$
```

Now refresh and u can check it okay

Now just refer 105 video for information and next video is of excersie okay for which link is below and also refere 106 video like this this section is done now

<https://plum-poppy-0ea.notion.site/Github-Basics-Exercise-1c12200326db47d7890702017602d698>

And I am putting the screenshot also here okay

**Github Basics Exercise**

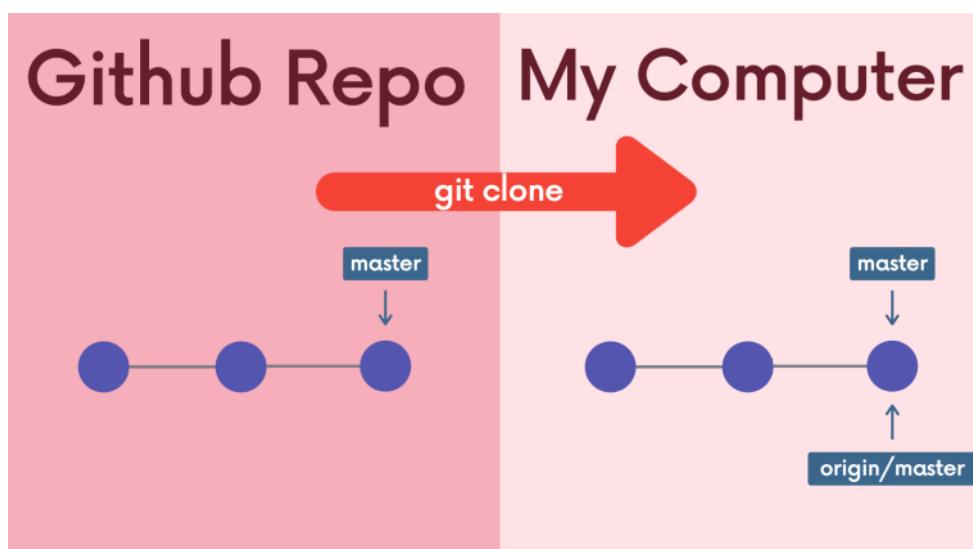
1. Create a new repository locally on your machine.
2. Create a new Github repository. Name it whatever you want.
3. Connect your local repo to the Github repo.
4. Optional: rename the default branch from master to main.
5. Make a new file called `Favorites.txt`. Leave it empty. Make your first commit on the `main` branch.
6. Push up your `main` branch to Github! Make sure you see your empty `Favorites.txt` file on Github.
7. Next, create two branches: `Food` and `Movies`.
8. Switch to the `Food` branch. Add three (or more) of your favorite foods to the `Favorites.txt` file. Add and commit your changes on the `Food` branch.
9. Switch to the `Movies` branch and add three or more of your favorite movies to the `Favorites.txt` file. Add and commit your changes on the `Movies` branch.
10. Push up your `Food` branch to Github. Make sure you see it on Github!
11. Push up your `Movies` branch to Github. Make sure you see it on Github!
12. Merge the `Food` branch into the `main` branch. Then merge the `Movies` branch into the `main` branch. If necessary, resolve conflicts so that you end up with your favorite foods and favorite movies in the same `Favorites.txt` file.
13. Push up the latest work on your `main` branch to Github.



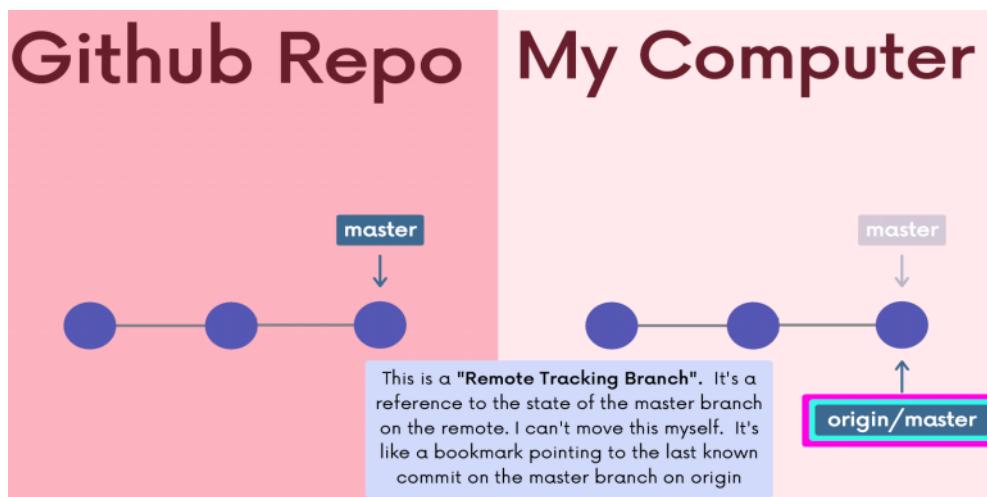
## Fetch and pull Notes

15 July 2023 22:58

So for this section I had already downloaded pdf slides now let us move on to the notes now so So here fetching and pulling is important when I'm working in a collaborative environment where different people will be working on the same project what they have modified I can fetch it or I can pull it so mostly this will be github code not git local code



So when ever we do clone what happens is that all will be downloaded into local and point also will come and I will get another pointer which will point to last commit which I had done in github repository so that I cannot move by myself for that u need to do some thing and this is chapter is all about that okay



So let do some demo and then we will come to know what to do and all okay

Create a repository with name animals In github okay

And then add one file in that with the name pets.txt one link will be there create new file and save it as .txt and in that add some pets name okay and say control + s and write some commit message and save it okay

The screenshot shows a GitHub repository page for 'raghuponde/animals'. The repository has 1 branch (main) and 0 tags. A single commit named 'Create pets.txt' was made by 'raghuponde' 1 minute ago. The commit message is 'Create pets.txt'. There is a note to 'Add a README'.



Next create one folder on desktop fetchandpull and in that folder clone above github repository okay ..

```
MINGW64:/c/Users/raghavendra/Desktop/fetchandpull
raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~
$ cd desktop
raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop
$ cd fetchandpull
raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop/fetchandpull
$ git status
fatal: not a git repository (or any of the parent directories): .git
raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop/fetchandpull
$ git clone https://github.com/raghuponde/animals.git
cloning into 'animals'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (3/3), done.

raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop/fetchandpull
$
```

```
raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop/fetchandpull
$ cd animals
```

```
raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop/fetchandpull/animals (main)
$ ls
pets.txt
```

```
raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop/fetchandpull/animals (main)
$ git branch
* main
```

```
raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop/fetchandpull/animals (main)
$
```

Further commands which I had written here is above

And we know that whenever I clone some thing I automatically gets this remote

```

raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop/fetchandpull
$ cd animals

raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop/fetchandpull/animals (main)
$ ls
pets.txt

raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop/fetchandpull/animals (main)
$ git branch
* main

raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop/fetchandpull/animals (main)
$ git remote -v
origin https://github.com/raghuponde/animals.git (fetch)
origin https://github.com/raghuponde/animals.git (push)

raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop/fetchandpull/animals (main)
$
```

Which is called as origin and one branch which we automatically have is main okay ..

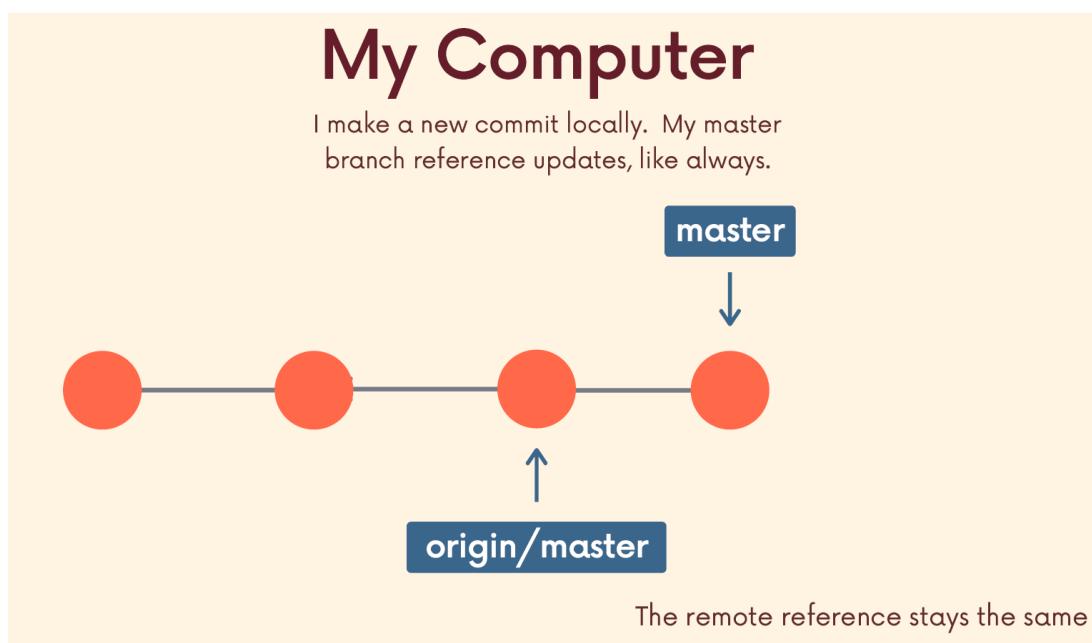
```

raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop/fetchandpull/animals (main)
$ git branch -r
  origin/HEAD -> origin/main
  origin/main

raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop/fetchandpull/animals (main)
$
```

So whenever I type -r it will show me on server origin it is also having a pointer from local git okay

As I just now cloned it so both will be pointing to one latest commit only as per the diagram in slides and now I will move on to further doing changes in local downloaded code and do commit local only but the pointer will not change for github lastest change so we will check this live okay



Now add two cats also into the pets file and do as usual stuff

```

MINGW64:/c/Users/raghavendra/Desktop/fetchandpull/animals
raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/Desktop/fetchandpull/animals (main)
$ code .

raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/Desktop/fetchandpull/animals (main)
$ 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:   pets.txt

no changes added to commit (use "git add" and/or "git commit -a")

raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/Desktop/fetchandpull/animals (main)
$ git add .

raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/Desktop/fetchandpull/animals (main)
$ git commit -m "two cats added"
[main b76afdf] two cats added
 1 file changed, 2 insertions(+)

raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/Desktop/fetchandpull/animals (main)
$ |
```

So I have two commits one just done locally okay

```
MINGW64:/c/Users/raghavendra/Desktop/fetchandpull/animals
raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/Desktop/fetchandpull/animals (main)
$ git log
commit b76af83ba3743f02684f4f100be503208687e41 (HEAD -> main)
Author: raghu <raghu070882@gmail.com>
Date:   Sun Jul 16 22:15:27 2023 +0530

    two cats added

commit 3d43081040efda9c60a84293ecdbee655c4a1f48 (origin/main, origin/HEAD)
Author: raghuponde <33595609+raghuponde@users.noreply.github.com>
Date:   Sun Jul 16 21:46:26 2023 +0530

    Create pets.txt

raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/Desktop/fetchandpull/animals (main)
$ |
```

Now type git status

```
raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/Desktop/fetchandpull/animals (main)
$ 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

raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/Desktop/fetchandpull/animals (main)
$
```

It is telling something and it is understandable of what I am saying okay

Now locally again add frogs in pets file and same stuff do it and check what it says

```
MINGW64:/c/Users/raghavendra/Desktop/fetchandpull/animals
raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/Desktop/fetchandpull/animals (main)
$ code .

raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/Desktop/fetchandpull/animals (main)
$ git status
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)

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:   pets.txt

no changes added to commit (use "git add" and/or "git commit -a")

raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/Desktop/fetchandpull/animals (main)
$ git add .

raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/Desktop/fetchandpull/animals (main)
$ git commit -m "two frogs added"
[main c38c033] two frogs added
 1 file changed, 3 insertions(+), 1 deletion(-)

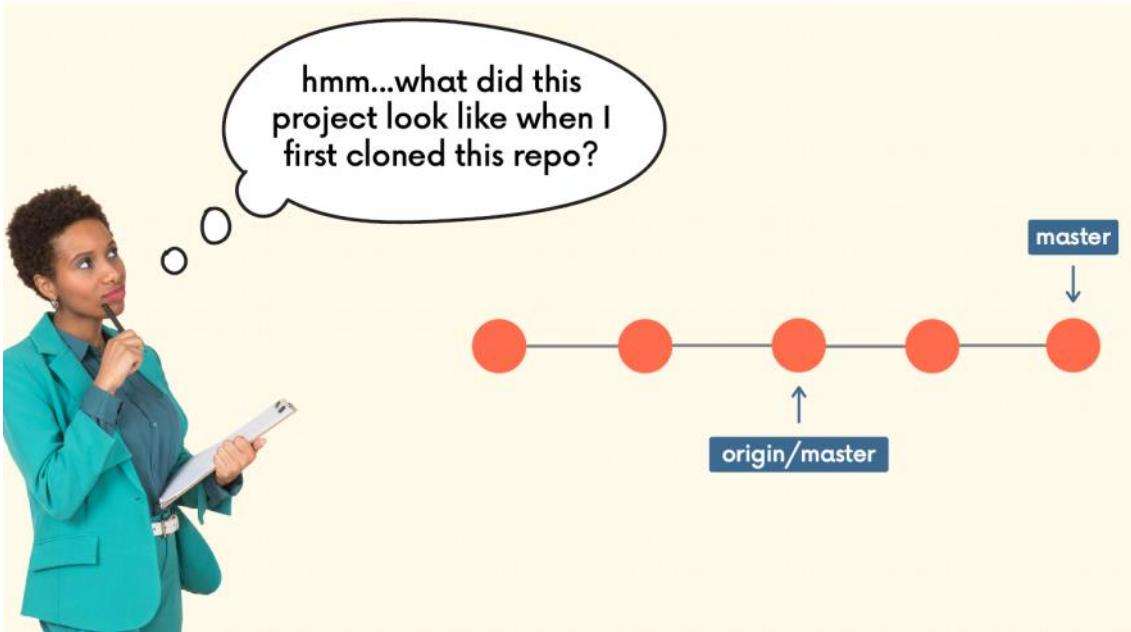
raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/Desktop/fetchandpull/animals (main)
$ git status
On branch main
Your branch is ahead of 'origin/main' by 2 commits.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean

raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/Desktop/fetchandpull/animals (main)
$ |
```

So this is thing I have done u can check it above from code okay

I want to know how this project look like when I first clone it



So run this command u will get it

## You can checkout these remote branch pointers

```
❯ git checkout origin/master
Note: switching to 'origin/master'.
You are in 'detached HEAD' state. You can look
around, make experimental changes and commit
them, and you can discard any commits you make
in this blah blah blah
```

Detached HEAD! Don't panic. It's fine.

So this is what I got after the command

```
MINGW64:/c/Users/raghavendra/Desktop/fetchandpull/animals
raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/Desktop/fetchandpull/animals (main)
$ git branch -r
  origin/HEAD -> origin/main
  origin/main

raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/Desktop/fetchandpull/animals (main)
$ git checkout origin/main
Note: switching to 'origin/main'.

You are in 'detached HEAD' state. You can look around, make experimental
changes and commit them, and you can discard any commits you make in this
state without impacting any branches by switching back to a branch.

If you want to create a new branch to retain commits you create, you may
do so (now or later) by using -c with the switch command. Example:

  git switch -c <new-branch-name>

or undo this operation with:

  git switch -

Turn off this advice by setting config variable advice.detachedHead to false

HEAD is now at 3d43081 Create pets.txt

raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/Desktop/fetchandpull/animals ((3d43081...))
$ |
```

Now go and see there it will show the content before two local commits okay

```
puppy
rusty
musty
kusum
bunty
```

This much data is there in file next

We know here that detached head the problem came As we have done time travel like earlier code it will definitely show detached head so come back to main and then just push the code to make it synchronize with the server of github okay

```
raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/Desktop/fetchandpull/animals ((3d43081...))
$ git switch main
Previous HEAD position was 3d43081 Create pets.txt
Switched to branch 'main'
Your branch is ahead of 'origin/main' by 2 commits.
  (use "git push" to publish your local commits)

raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/Desktop/fetchandpull/animals (main)
$ git push origin main
Enumerating objects: 8, done.
Counting objects: 100% (8/8), done.
Delta compression using up to 20 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (6/6), 504 bytes | 504.00 KiB/s, done.
Total 6 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), done.
To https://github.com/raghuponde/animals.git
  3d43081..c38c033  main -> main

raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/Desktop/fetchandpull/animals (main)
$ |
```

And now see the code in pets.txt file

```
puppy
rusty
musty
kusum
bunty
catname1
cat name 2
frog 1
frog 2
```

Okay so now

```
MINGW64:/c/Users/raghavendra/Desktop/fetchandpull/animals
raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/Desktop/fetchandpull/animals (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.
nothing to commit, working tree clean
raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/Desktop/fetchandpull/animals (main)
$ |
```

So now it is up to date next

Still I have not gone through Fetch and Pull Still working on some important concepts now I will be going with working with remote branches how I can make those remote batches to be synchronised with local machine repositories

<https://github.com/Colt/remote-branches-demo>

Check above link for doing this demo okay

Now I will clone the above url into my local machine fetchandpull folder okay

```
MINGW64:/c/Users/raghavendra/Desktop/fetchandpull
raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~
$ cd desktop
raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/Desktop
$ cd fetchandpull
raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/Desktop/fetchandpull
$ git status
fatal: not a git repository (or any of the parent directories): .git
raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/Desktop/fetchandpull
$ git clone https://github.com/Colt/remote-branches-demo.git
Cloning into 'remote-branches-demo'...
remote: Enumerating objects: 57, done.
remote: Counting objects: 100% (21/21), done.
remote: Compressing objects: 100% (14/14), done.
remote: Total 57 (delta 10), reused 7 (delta 7), pack-reused 36
Receiving objects: 100% (57/57), 14.89 KiB | 1.06 MiB/s, done.
Resolving deltas: 100% (11/11), done.
raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/Desktop/fetchandpull
$ |
```

Next

```
MINGW64:/c/Users/raghavendra/Desktop/fetchandpull/remote-branches-demo
raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/Desktop/fetchandpull
$ cd remote*
raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/Desktop/fetchandpull/remote-branches-demo (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.
nothing to commit, working tree clean
raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/Desktop/fetchandpull/remote-branches-demo (main)
$ git branch
* main
raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/Desktop/fetchandpull/remote-branches-demo (main)
$ |
```

You can see when I cloned I don't have other branches only master content I can see okay

Now the question is how do I work in food branch etc some other branch other than main branch ok

So here in github we have 4 branches I guess

The screenshot shows a browser window with several tabs open. The active tab is 'Colt/remote-branches-demo' on GitHub. The page displays a repository with the following details:

- Branches:** main (selected), fantasy, food, more-fantasy, morefood, movies
- Tags:** 0
- Commits:** 2 (7fde881 on Feb 2, 2021)
- Files:** Create cactus.txt, Create rose.txt
- Activity:** 2 commits
- Statistics:** 15 stars, 3 watching, 379 forks
- Repository Options:** Watch, Fork, Star, Report repository
- Releases:** No releases published
- Packages:** No packages published

At the bottom of the GitHub page, there's a footer with links to GitHub's Terms, Privacy, Security, Status, Docs, Contact GitHub, Pricing, API, Training, Blog, and About.

So you can see actually we have 5 more branches with us along with master branch okay

```
raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop/fetchandpull/remote-branches-demo (main)
$ git branch
* main

raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop/fetchandpull/remote-branches-demo (main)
$ git branch -r
origin/HEAD -> origin/main
origin/fantasy
origin/food
origin/main
origin/more-fantasy
origin/morefood
origin/movies

raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop/fetchandpull/remote-branches-demo (main)
$ |
```

So here if type -r command it is showing to me next  
My local branch knows everything and it is having reference of all the branches present in github okay

```

MINGW64:/c/Users/raghavendra/Desktop/fetchandpull/remote-branches-demo
raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/Desktop/fetchandpull/remote-branches-demo (main)
$ git branch -r
origin/HEAD -> origin/main
origin/fantasy
origin/food
origin/main
origin/more-fantasy
origin/morefood
origin/movies

raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/Desktop/fetchandpull/remote-branches-demo (main)
$ git checkout origin/food
Note: switching to 'origin/food'.

You are in 'detached HEAD' state. You can look around, make experimental
changes and commit them, and you can discard any commits you make in this
state without impacting any branches by switching back to a branch.

If you want to create a new branch to retain commits you create, you may
do so (now or later) by using -c with the switch command. Example:

  git switch -c <new-branch-name>

or undo this operation with:

  git switch -

Turn off this advice by setting config variable advice.detachedHead to false

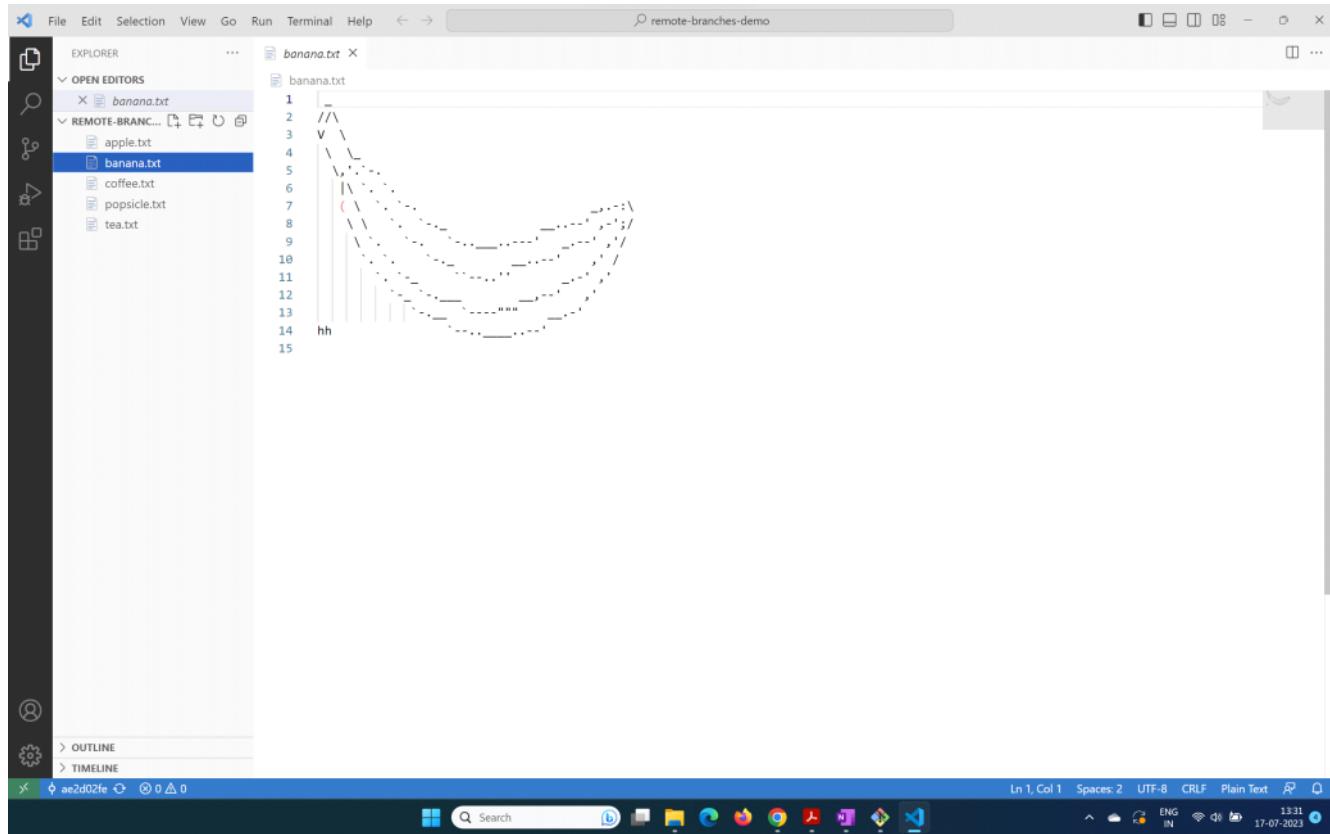
HEAD is now at ae2d02f Create tea.txt

raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/Desktop/fetchandpull/remote-branches-demo ((ae2d02f...))
$ ls
apple.txt banana.txt coffee.txt popsicle.txt tea.txt

raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/Desktop/fetchandpull/remote-branches-demo ((ae2d02f...))
$
```

So here I can see some what list them how many files are there and all okay in food branch

So I am able to check out those I am in detached head



I can just check out it is not included in my working directory so I need a branch where I can work out for that food branch

So even though branch is not there I can write a command like this

```

raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop/fetchandpull/remote-branches-demo ((ae2d02f...))
$ git switch main
Previous HEAD position was ae2d02f Create tea.txt
Switched to branch 'main'
Your branch is up to date with 'origin/main'.

raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop/fetchandpull/remote-branches-demo (main)
$ git switch food
Switched to a new branch 'food'
branch 'food' set up to track 'origin/food'.

raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop/fetchandpull/remote-branches-demo (food)
$
```

Now u can see I am on food branch and now if I type git branch

It will show some physical connection okay

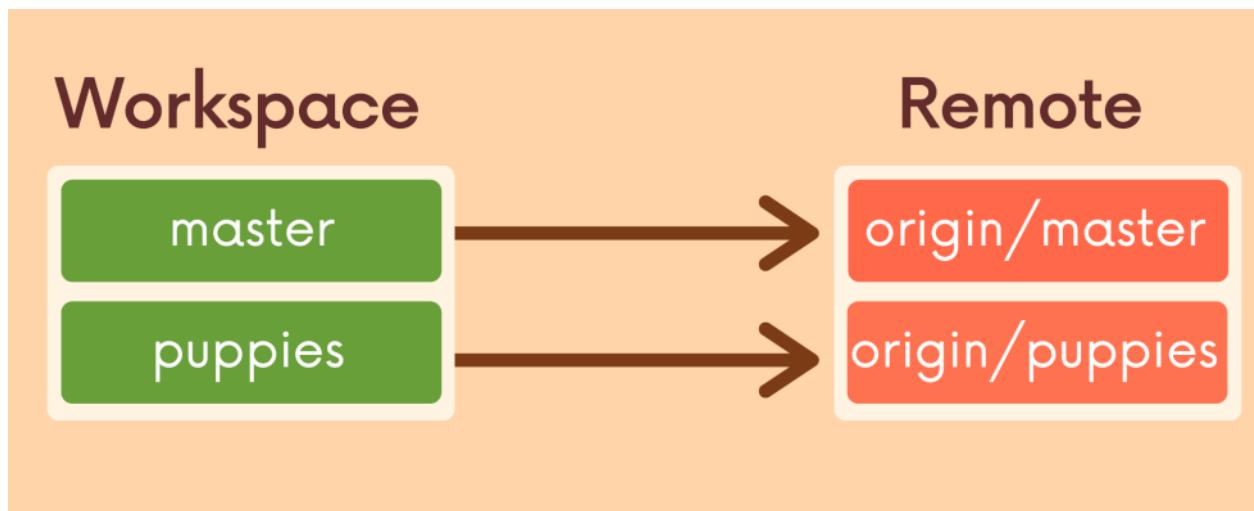
```

raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop/fetchandpull/remote-branches-demo (food)
$ git branch
* food
  main

raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop/fetchandpull/remote-branches-demo (food)
$ git status
On branch food
Your branch is up to date with 'origin/food'.
nothing to commit, working tree clean

raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop/fetchandpull/remote-branches-demo (food)
$ |
```

Something like this connection will happen



Now in the same manner go to fantasy branch and add some file in that

```

raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop/fetchandpull/remote-branches-demo (food)
$ git branch -r
origin/HEAD -> origin/main
origin/fantasy
origin/food
origin/main
origin/more-fantasy
origin/morefood
origin/movies
```

```

raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop/fetchandpull/remote-branches-demo (food)
$ git switch fantasy
Switched to a new branch 'fantasy'
branch 'fantasy' set up to track 'origin/fantasy'.
```

```

raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop/fetchandpull/remote-branches-demo (fantasy)
$ git status
On branch fantasy
Your branch is up to date with 'origin/fantasy'.
```

nothing to commit, working tree clean

```

raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop/fetchandpull/remote-branches-demo (fantasy)
$ touch phonix.txt
```

```

raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop/fetchandpull/remote-branches-demo (fantasy)
```

```
$ code .
raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop/fetchandpull/remote-branches-demo (fantasy)
$ git status
On branch fantasy
Your branch is up to date with 'origin/fantasy'.
```

```
Untracked files:
(use "git add <file>..." to include in what will be committed)
phonix2.txt
```

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

```
raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop/fetchandpull/remote-branches-demo (fantasy)
$ git add .
```

```
raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop/fetchandpull/remote-branches-demo (fantasy)
$ git commit -m "phonex2 file added"
[fantasy a876b25] phonex2 file added
1 file changed, 24 insertions(+)
create mode 100644 phonex2.txt
```

```
raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop/fetchandpull/remote-branches-demo (fantasy)
$ git status
On branch fantasy
Your branch is ahead of 'origin/fantasy' by 1 commit.
(use "git push" to publish your local commits)
```

```
nothing to commit, working tree clean
```

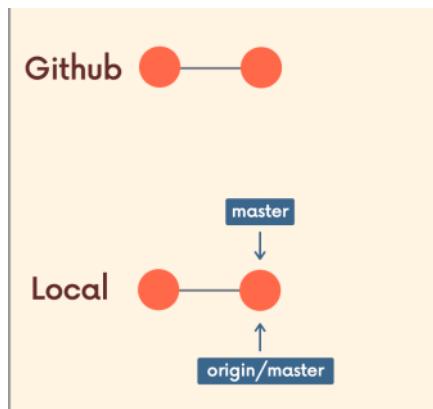
```
raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop/fetchandpull/remote-branches-demo (fantasy)
$
```

Here in touch I had written phonex but later on I chnaged name to phinex2 because when cloned phonix was already there next command just check It as per video  
okay

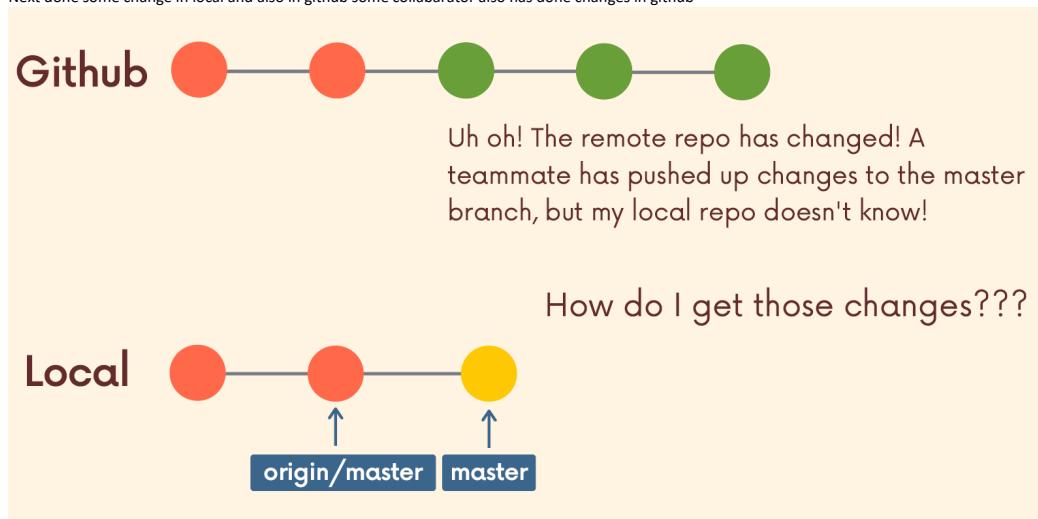
```
raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop/fetchandpull/remote-branches-demo (fantasy)
$ git push origin fantasy
remote: Permission to Colt/remote-branches-demo.git denied to raghuponde.
fatal: unable to access 'https://github.com/Colt/remote-branches-demo.git/': The requested URL returned error: 403
```

So it is not allwoing me as it is some body url so but it will create a change with new phonix2 file and one option for this is to fork the branch into yur login credentials  
remember branch is not present and you are switching it will allow but same name should be there for the branch which is there in github and any changes from here u  
can push It over there ..easily okay

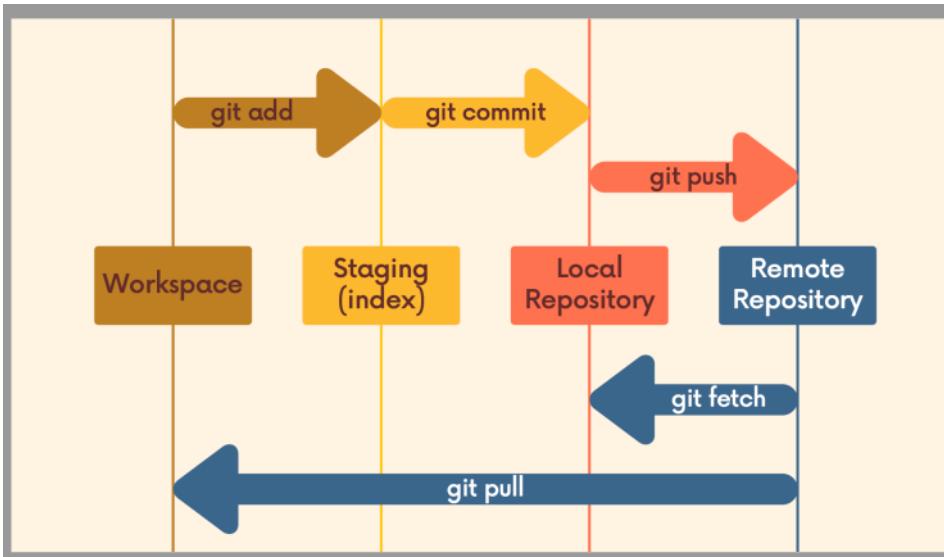
Now let us talk about fetch new commmand okay



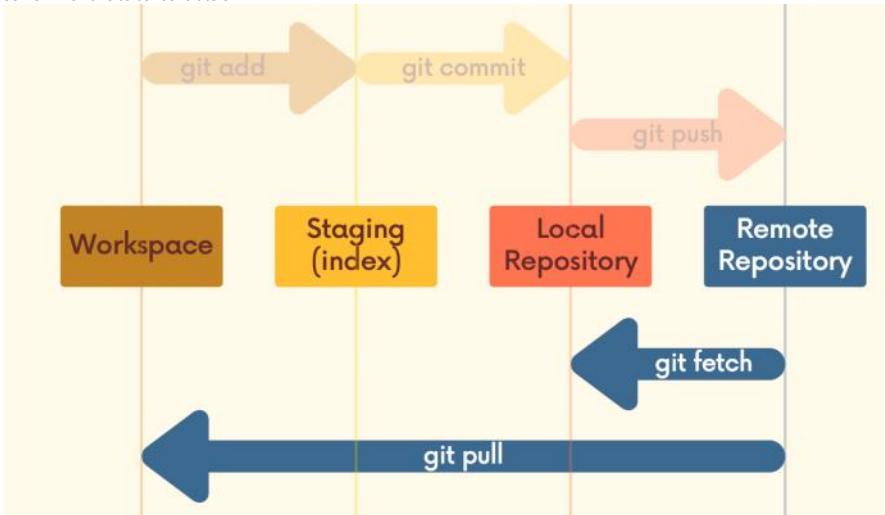
So this is the scenario here from github I cloned and whole tree is up to date  
Next done some change in local and also in github some collaborator also has done changes in github



NO Body knows anybody changes so this is the diagram ...so below diagram shows all stuff okay



So now we have to concentrate on



So the diagrams from slides tell all

## Fetching

Fetching allows us to download changes from a remote repository, BUT those changes will not be automatically integrated into our working files.

It lets you see what others have been working on, without having to merge those changes into your local repo.

Think of it as "please go and get the latest information from Github, but don't screw up my working directory."



We will go with direct demo if needed okay

# Git Fetch

The `git fetch <remote>` command fetches branches and history from a specific remote repository. It only updates remote tracking branches.

`git fetch origin` would fetch all changes from the origin remote repository.

```
git fetch <remote>
```

If not specified, `<remote>` defaults to `origin`

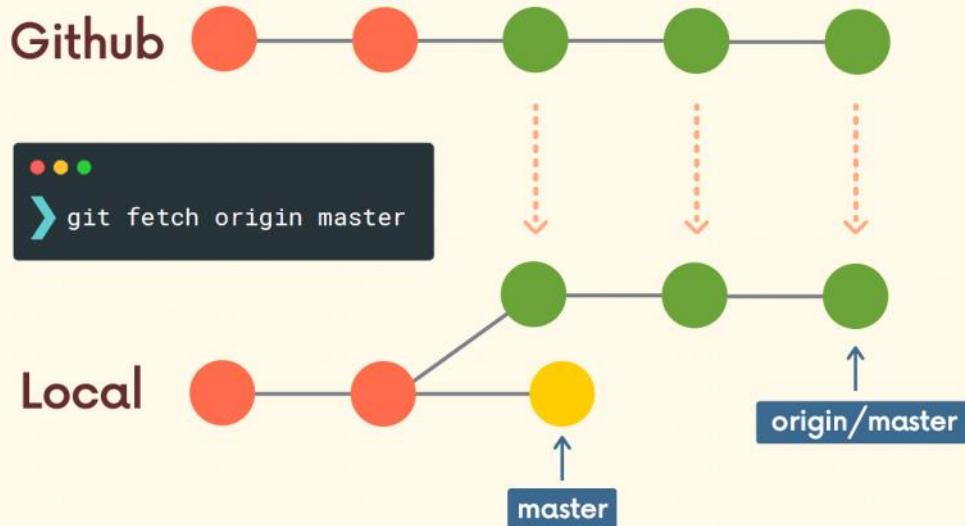
# Git Fetch

We can also fetch a specific branch from a remote using `git fetch <remote> <branch>`

For example, `git fetch origin master` would retrieve the latest information from the master branch on the origin remote repository.

```
git fetch <remote> <branch>
```

Without disturbing my folder in local I want to access those changes in github then

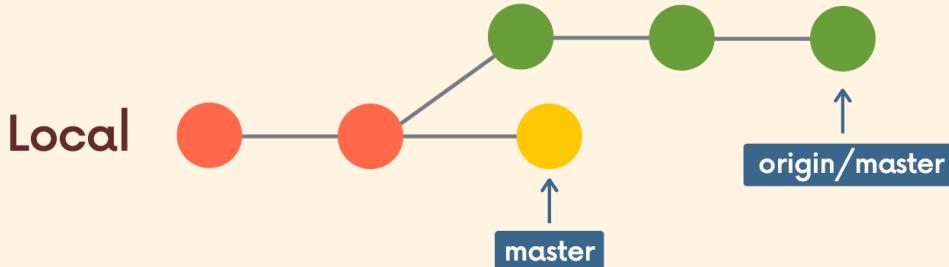


U need to write this command ..  
And after it has been changed means got not disturbing my local one just type check out command okay

# Github



I now have those changes on my machine, but if I want to see them I have to **checkout origin/master**. My master branch is untouched!



Now so actually go with some demo to understand what we have learned okay ..

Now this url first I had forked it with my credentials okay <https://github.com/Colt/remote-branches-demo/>, and after forking it I got like this in my credentials The url

<https://github.com/raghuponde/remote-branches-demo>

And then I clicked the button branches and

The screenshot shows a GitHub repository interface. At the top, there's a navigation bar with tabs for 'Code', 'Pull requests', 'Actions', 'Projects', 'Wiki', 'Security', 'Insights', and 'Settings'. Below the navigation bar, there's a search bar and a 'New branch' button. The main content area is titled 'Default branch' and shows the 'main' branch, which is marked as 'Default'. There's a note saying 'Your main branch isn't protected' with a 'Protect this branch' button. Below this, there's a section for 'Your branches' containing 'fantasy', 'food', and 'movies'. Each of these branches has a small icon, a timestamp ('Updated 3 years ago by Colt Steele'), and a 'New pull request' button. At the bottom, there's a section for 'Stale branches' with 'fantasy', 'food', and 'movies' listed again. The footer contains links for 'Terms', 'Privacy', 'Security', 'Status', 'Docs', 'Contact GitHub', 'Pricing', 'API', 'Training', 'Blog', and 'About'.

And from here created new dummy branches from main okay named them fantasy,food and movies okay

The screenshot shows the GitHub fork creation interface. The URL in the address bar is [github.com/Colt/remote-branches-demo/fork](https://github.com/Colt/remote-branches-demo/fork). The form fields include:

- Owner:** raghuponde
- Repository name:** remote-branches-demo
- Description (optional):** (empty)
- Copy the main branch only:** (unchecked)

At the bottom, there is a note: "You are creating a fork in your personal account." and a green "Create fork" button.

© 2023 GitHub, Inc. Terms Privacy Security Status Docs Contact GitHub Pricing API Training Blog About



So here I had unchecked the option copy the main branch only so all stuff will be forked along with branches

Now everything is copied into this url okay

<https://github.com/raghuponde/remote-branches-demo>

The screenshot shows the GitHub repository page for "remote-branches-demo". The URL is <https://github.com/raghuponde/remote-branches-demo>. The repository details are:

- Forked from:** Colt/remote-branches-demo
- Branches:** main, 6 branches
- Tags:** 0 tags
- Protected Branches:** Protect this branch
- Last Commit:** 7f6e88 on Feb 2, 2021
- Contributors:** 2 commits
- Activity:** 0 stars, 0 watching
- Forks:** 380 forks
- Releases:** No releases published. Create a new release
- Packages:** No packages published. Publish your first package

© 2023 GitHub, Inc. Terms Privacy Security Status Docs Contact GitHub Pricing API Training Blog About



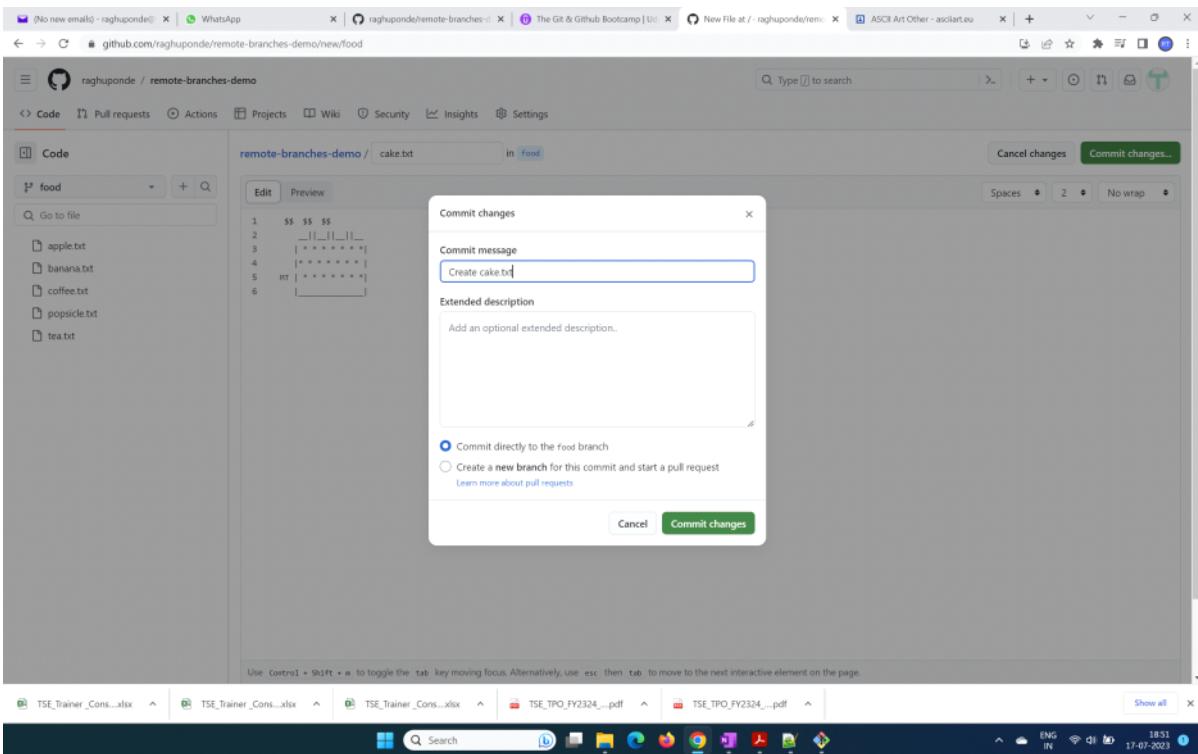
So now let us go with demo

Once check this website where I can copy pictures okay <https://www.asciiart.eu/>

Now u have your url with you..

<https://github.com/raghuponde/remote-branches-demo>

Go to food branch and add one file with name cake.txt and add one image from <https://www.asciiart.eu/>



When you do control + s this will come and just do commit okay

Now go to movies branch now there create a file name jamesband and add some image of 007 there and commit it okay in movies branch only

Now I have a folder fetch and pull and in that delete earleir folder and do cloning of code now what I want is all main with branches I want it in local folder okay

```
raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~
$ cd desktop
```

```
raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop
$ cd fetc*
```

```
raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop/fetchandpull
$ git status
fatal: not a git repository (or any of the parent directories): .git
```

```
raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop/fetchandpull
$ git clone https://github.com/raghuponde/remote-branches-demo.git
Cloning into 'remote-branches-demo'...
remote: Enumerating objects: 63, done.
remote: Counting objects: 100% (27/27), done.
remote: Compressing objects: 100% (19/19), done.
remote: Total 63 (delta 12), reused 8 (delta 8), pack-reused 36
Receiving objects: 100% (63/63), 16.24 KiB | 3.25 MiB/s, done.
Resolving deltas: 100% (13/13), done.
```

```
raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop/fetchandpull
$ cd remo*
```

After doing cloning I am jumping to that folder and from there I am going to movies which is not there locally

But I can do it like that

```
raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop/fetchandpull/remote-branches-demo (main)
```

```
$ git switch movies
```

Switched to a new branch 'movies'

branch 'movies' set up to track 'origin/movies'.

And doing clone here same command so all movies files will come here u can check it

```
raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop/fetchandpull/remote-branches-demo (movies)
$ git clone https://github.com/raghuponde/remote-branches-demo.git
Cloning into 'remote-branches-demo'...
remote: Enumerating objects: 63, done.
remote: Counting objects: 100% (27/27), done.
remote: Compressing objects: 100% (19/19), done.
remote: Total 63 (delta 12), reused 8 (delta 8), pack-reused 36
Receiving objects: 100% (63/63), 16.24 KiB | 1.48 MiB/s, done.
Resolving deltas: 100% (13/13), done.
```

Again go to food which not there but okay again

```
raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop/fetchandpull/remote-branches-demo (movies)
$ git switch food
Switched to a new branch 'food'
branch 'food' set up to track 'origin/food'.
```

Do clone don't worry if not showing eeffect but it will be there if u go and see being in that food branch

```
raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop/fetchandpull/remote-branches-demo (food)
$ git clone https://github.com/raghuponde/remote-branches-demo.git
fatal: destination path 'remote-branches-demo' already exists and is not an empty directory.
```

```

raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop/fetchandpull/remote-branches-demo (food)
$ cd food
bash: cd: food: No such file or directory

raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop/fetchandpull/remote-branches-demo (food)
$ git clone https://github.com/raghuponde/remote-branches-demo.git
fatal: destination path 'remote-branches-demo' already exists and is not an empty directory.

raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop/fetchandpull/remote-branches-demo (food)
$ git switch movies
Switched to branch 'movies'
Your branch is up to date with 'origin/movies'.

raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop/fetchandpull/remote-branches-demo (movies)
$ git switch main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.

raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop/fetchandpull/remote-branches-demo (main)
$
```

So by writing some commands I had checked it so locally food and movies created and files cloned properly

Now I had earlier added files in movies and food and they are reflecting completely and not used fetch yet now what I will do that I will do change in master and some other child and try to fetch earlier I should have not done those two changes I done because I thought here that my remote branch is locally arranged but not so again now I will do that let us see how some extra work done but okay ...

Added one file mushroom.txt and added one image in main branch of github okay and also done commit also

Now going into command of bash and typed and then git status

```

raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop/fetchandpull/remote-branches-demo (main)
$ git fetch origin
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 717 bytes | 179.00 KiB/s, done.
From https://github.com/raghuponde/remote-branches-demo
 7fdeb88..954d627 main      -> origin/main
```

```

raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop/fetchandpull/remote-branches-demo (main)
$ git status
On branch main
Your branch is behind 'origin/main' by 1 commit, and can be fast-forwarded.
(use "git pull" to update your local branch)
```

Untracked files:  
 (use "git add <file>..." to include in what will be committed)  
 remote-branches-demo/

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

So it is saying back side you are so and if we type commands nothing is seen yet means no mushroom file ...

```

raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop/fetchandpull/remote-branches-demo (main)
$ ls
cactus.txt  remote-branches-demo/  rose.txt
```

```

raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop/fetchandpull/remote-branches-demo (main)
$ git log
commit 7fdeb8873db735efaa8d8d09c43b245c5697052c (HEAD -> main)
Author: Colt Steele <5498438+Colt@users.noreply.github.com>
Date: Mon Feb 1 14:03:28 2021 -0800
```

Create rose.txt

```

commit a58e6ac648c90b84bc9875205a3229ebaecf5f9e
Author: Colt Steele <5498438+Colt@users.noreply.github.com>
Date: Mon Feb 1 14:02:11 2021 -0800
```

Create cactus.txt

Also no commits for mushroom I can see here okay

```

raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop/fetchandpull/remote-branches-demo (main)
$ git fetch origin main
From https://github.com/raghuponde/remote-branches-demo
 * branch      main      -> FETCH_HEAD
```

```

raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop/fetchandpull/remote-branches-demo (main)
$ ls
cactus.txt  remote-branches-demo/  rose.txt
```

```

raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop/fetchandpull/remote-branches-demo (main)
$ git status
On branch main
Your branch is behind 'origin/main' by 1 commit, and can be fast-forwarded.
(use "git pull" to update your local branch)
```

Untracked files:  
 (use "git add <file>..." to include in what will be committed)  
 remote-branches-demo/

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

So after writing /main also I cannot see the effect so when I write checkout the things will work ...

```
raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop/fetchandpull/remote-branches-demo (main)
$ git checkout origin/main
Note: switching to 'origin/main'.
```

You are in 'detached HEAD' state. You can look around, make experimental changes and commit them, and you can discard any commits you make in this state without impacting any branches by switching back to a branch.

If you want to create a new branch to retain commits you create, you may do so (now or later) by using -c with the switch command. Example:

```
git switch -c <new-branch-name>
```

Or undo this operation with:

```
git switch -
```

Turn off this advice by setting config variable advice.detachedHead to false

```
HEAD is now at 954d627 Create Mushroom
```

```
raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop/fetchandpull/remote-branches-demo ((954d627...))
$ git log
commit 954d627155e2e1dd55f64da84779908c3a0df1fb (HEAD, origin/main, origin/HEAD)
Author: raghuponde <33595609+raghuponde@users.noreply.github.com>
Date: Mon Jul 17 21:06:03 2023 +0530
```

Create Mushroom

```
commit 7fdb8873db735efaa8d6d09c43b245c5697052c (main)
Author: Colt Steele <5498438+Colt@users.noreply.github.com>
Date: Mon Feb 1 14:03:28 2021 -0800
```

Create rose.txt

```
commit a58e6ac648c90b84bc9875205a3229ebaecf5f9e
Author: Colt Steele <5498438+Colt@users.noreply.github.com>
Date: Mon Feb 1 14:02:11 2021 -0800
```

Create cactus.txt

```
raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop/fetchandpull/remote-branches-demo ((954d627...))
$
```

Now I am switching to movies

```
raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop/fetchandpull/remote-branches-demo (food)
$ git switch movies
Switched to branch 'movies'
Your branch is up to date with 'origin/movies'.
```

```
raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop/fetchandpull/remote-branches-demo (movies)
$ git status
On branch movies
Your branch is up to date with 'origin/movies'.
```

Untracked files:  
(use "git add <file>..." to include in what will be committed)  
remote-branches-demo/

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

And now in github will do changes in movies added one Lion .txt file and added image of ascii okay

```
raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop/fetchandpull/remote-branches-demo (movies)
$ git fetch origin
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (2/2), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 1
Unpacking objects: 100% (3/3), 1.55 KiB | 158.00 KiB/s, done.
From https://github.com/raehuponde/remote-branches-demo
 419d60e..2b2abfa movies -> origin/movies
```

```
raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop/fetchandpull/remote-branches-demo (movies)
$ git status
On branch movies
Your branch is behind 'origin/movies' by 1 commit, and can be fast-forwarded.
(use "git pull" to update your local branch)
```

Untracked files:  
(use "git add <file>..." to include in what will be committed)  
remote-branches-demo/

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

```
raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop/fetchandpull/remote-branches-demo (movies)
$ git checkout origin/movies
Note: switching to 'origin/movies'.
```

You are in 'detached HEAD' state. You can look around, make experimental changes and commit them, and you can discard any commits you make in this state without impacting any branches by switching back to a branch.

If you want to create a new branch to retain commits you create, you may do so (now or later) by using -c with the switch command. Example:

```
git switch -c <new-branch-name>
```

Or undo this operation with:

```
git switch -
```

Turn off this advice by setting config variable advice.detachedHead to false

HEAD is now at 2b2abfa Create Lion.txt

```
raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop/fetchandpull/remote-branches-demo ((2b2abfa...))
$ git status
HEAD detached at origin/movies
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    remote-branches-demo/
```

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

```
raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop/fetchandpull/remote-branches-demo ((2b2abfa...))
$ git log
commit 2b2abfa09bdc83698278d7e2bde137af4aa9ae8 (HEAD, origin/movies)
Author: raghuponde <33595609+raghuponde@users.noreply.github.com>
Date: Mon Jul 17 21:36:32 2023 +0530
```

Create Lion.txt

```
commit 419d60e6f5c423564de664e312767540e758e633 (movies)
Author: raghuponde <33595609+raghuponde@users.noreply.github.com>
Date: Mon Jul 17 18:54:50 2023 +0530
```

Create jamesbond

```
commit d94a89ed4af37475ecb89634c0aa31c44f4b2836
Author: Colt Steele <5498438+Colt@users.noreply.github.com>
Date: Mon Feb 1 16:42:21 2021 -0800
```

Create tinkerbell.txt

```
commit 6c32e319078957ebbf9a0df486b1afbd9a3952e7
Author: Colt Steele <5498438+Colt@users.noreply.github.com>
Date: Mon Feb 1 14:05:17 2021 -0800
```

Create bambi.txt

```
commit 335cbf5f7edb25a79cb666a93ee464a461089f4e
Author: Colt Steele <5498438+Colt@users.noreply.github.com>
Date: Mon Feb 1 14:04:48 2021 -0800
```

Create ghostbusters.txt

```
commit 2cfeb8a1982ad1f684a93b945ca95eacb901c8f0
Author: Colt Steele <5498438+Colt@users.noreply.github.com>
```

```
raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop/fetchandpull/remote-branches-demo ((2b2abfa...))
$
```

When I say git ls then same files it shows so my working directory has not changed okay or effected okay

Another time pasting the command for better understanding and also once go through video of demo u will understand well

```
raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop/fetchandpull/remote-branches-demo (movies)
$ git status
On branch movies
Your branch is behind 'origin/movies' by 1 commit, and can be fast-forwarded.
  (use "git pull" to update your local branch)
```

Untracked files:

```
  (use "git add <file>..." to include in what will be committed)
    remote-branches-demo/
```

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

```
raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop/fetchandpull/remote-branches-demo (movies)
$ git checkout origin/movies
Note: switching to 'origin/movies'.
```

You are in 'detached HEAD' state. You can look around, make experimental changes and commit them, and you can discard any commits you make in this state without impacting any branches by switching back to a branch.

If you want to create a new branch to retain commits you create, you may do so (now or later) by using -c with the switch command. Example:

```
git switch -c <new-branch-name>
```

Or undo this operation with:

```
git switch -
```

Turn off this advice by setting config variable advice.detachedHead to false

HEAD is now at 2b2abfa Create Lion.txt

```
raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop/fetchandpull/remote-branches-demo ((2b2abfa...))
$ ls
Lion.txt bambi.txt ghostbusters.txt jamesbond remote-branches-demo/ tinkerbell.txt
```

```
raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop/fetchandpull/remote-branches-demo ((2b2abfa...))
$ git switch movies
Previous HEAD position was 2b2abfa Create Lion.txt
Switched to branch 'movies'
```

```
Your branch is behind 'origin/movies' by 1 commit, and can be fast-forwarded.  
(use "git pull" to update your local branch)
```

```
raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop/fetchandpull/remote-branches-demo (movies)  
$ ls  
bambi.txt ghostbusters.txt jamesbond remote-branches-demo/ tinkerbell.txt
```

```
raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop/fetchandpull/remote-branches-demo (movies)  
$ git status  
On branch movies  
Your branch is behind 'origin/movies' by 1 commit, and can be fast-forwarded.  
(use "git pull" to update your local branch)
```

```
Untracked files:  
(use "git add <file>..." to include in what will be committed)  
remote-branches-demo/
```

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

```
raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop/fetchandpull/remote-branches-demo (movies)  
$
```

```
.
```

```
Now I will give it a last try with food
```

```
From movies I am going to food now
```

```
MINGW64:/c/Users/raghavendra/desktop/fetchandpull/remote-branches-demo  
raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop/fetchandpull/remote-branches-demo (movies)  
$ git switch food  
Switched to branch 'food'  
Your branch is up to date with 'origin/food'.  
raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop/fetchandpull/remote-branches-demo (food)  
$ git status  
On branch food  
Your branch is up to date with 'origin/food'.  
Untracked files:  
(use "git add <file>..." to include in what will be committed)  
remote-branches-demo/  
nothing added to commit but untracked files present (use "git add" to track)  
raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop/fetchandpull/remote-branches-demo (food)  
$ |
```

```
so food is up to date now
```

```
Now do change in food go to branch and add food etc stuff in github okay
```

```
So I had added icecream.txt
```

```
And also image of ascii into it okay
```

```
Now if I am typing git status again it is still saying up to date only means It is not reflecting the change done in github into local one okay
```

```
raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop/fetchandpull/remote-branches-demo (food)  
$ git fetch origin food  
remote: Enumerating objects: 3, done.  
remote: Counting objects: 100% (3/3), done.  
remote: Compressing objects: 100% (3/3), done.  
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0  
Unpacking objects: 100% (3/3), 872 bytes | 72.00 KiB/s, done.  
From https://github.com/raghuponde/remote-branches-demo  
* branch food      -> FETCH_HEAD  
  99917bd..bd4239d food      -> origin/food  
raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop/fetchandpull/remote-branches-demo (food)  
$ git status  
On branch food  
Your branch is behind 'origin/food' by 1 commit, and can be fast-forwarded.  
(use "git pull" to update your local branch)  
Untracked files:  
(use "git add <file>..." to include in what will be committed)  
remote-branches-demo/  
nothing added to commit but untracked files present (use "git add" to track)  
raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop/fetchandpull/remote-branches-demo (food)  
$
```

```
So now it shows this thing after firing fetch command
```

```
If you go and see the folder we cannot see ice cream yet as I had told you it will not effect my working directory and but still I want to see means checkout I need to do I  
guess
```

```

raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop/fetchandpull/remote-branches-demo (food)
$ git checkout origin/food
Note: switching to 'origin/food'.

You are in 'detached HEAD' state. You can look around, make experimental
changes and commit them, and you can discard any commits you make in this
state without impacting any branches by switching back to a branch.

If you want to create a new branch to retain commits you create, you may
do so (now or later) by using -c with the switch command. Example:

  git switch -c <new-branch-name>

Or undo this operation with:

  git switch -

Turn off this advice by setting config variable advice.detachedHead to false

HEAD is now at bd4239d create icecream.txt

raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop/fetchandpull/remote-branches-demo ((bd4239d...))
$ 

```

Now I can see the icecream.txt now also I had just moved there okay ...

Now let us learn about pull command now

## Pulling

**git pull** is another command we can use to retrieve changes from a remote repository. Unlike fetch, pull actually updates our HEAD branch with whatever changes are retrieved from the remote.

"go and download data from Github AND immediately update my local repo with those changes"



## git pull = git fetch + git merge

update the remote tracking branch  
with the latest changes from the  
remote repository

update my current branch with  
whatever changes are on the remote  
tracking branch

## git pull

To pull, we specify the particular remote and branch we want to pull using **git pull <remote> <branch>**. Just like with git merge, it matters WHERE we run this command from. Whatever branch we run it from is where the changes will be merged into.

**git pull origin master** would fetch the latest information from the origin's master branch and merge those changes into our current branch.

```

● ● ●
> git pull <remote> <branch>

```

What matters here is where means from which branch in local I will fire this command when we will see the example we will get to know okay

# pulls can result in merge conflicts!!

Just like how in git jumping from local from child to master there will be conflicts here also conflicts will be there in pull we will see all those now okay

So let us go with demo where no conflicts is there then with conflicts okay ..

Now I am going to movies branch there I had added lion.txt file in github but here down it is not affected so

You can see from set of commands I had fired okay

```
MINGW64:/c/Users/raghavendra/Desktop/fetchandpull/remote-branches-demo
raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~
$ cd desktop
raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/Desktop
$ cd fet*
raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/Desktop/fetchandpull
$ cd remo*
raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/Desktop/fetchandpull/remote-branches-demo ((bd4239d...))
$ git branch -r
origin/HEAD -> origin/main
origin/fantasy
origin/food
origin/main
origin/more-fantasy
origin/morefood
origin/movies

raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/Desktop/fetchandpull/remote-branches-demo ((bd4239d...))
$ git switch movies
Previous HEAD position was bd4239d Create icecream.txt
Switched to branch 'movies'.
Your branch is behind 'origin/movies' by 1 commit, and can be fast-forwarded.
  (use "git pull" to update your local branch)

raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/Desktop/fetchandpull/remote-branches-demo (movies)
$ ls
bambi.txt ghostbusters.txt jamesbond remote-branches-demo/ tinkerbell.txt

raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/Desktop/fetchandpull/remote-branches-demo (movies)
$ git status
On branch movies
Your branch is behind 'origin/movies' by 1 commit, and can be fast-forwarded.
  (use "git pull" to update your local branch)

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    remote-branches-demo/

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

raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/Desktop/fetchandpull/remote-branches-demo (movies)
$
```

Now I want to see that lion in local also

```
raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/Desktop/fetchandpull/remote-branches-demo (movies)
$ git pull origin movies
From https://github.com/raghuponde/remote-branches-demo
 * branch      movies  -> FETCH_HEAD
Updating 419d60e..2b2abfa
Fast-forward
Lion.txt | 51 ++++++
1 file changed, 51 insertions(+)
create mode 100644 Lion.txt
```

```
raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/Desktop/fetchandpull/remote-branches-demo (movies)
$ git log
commit 2b2abfa09bddd83698278d7e2bde137af4aa9ae8 (HEAD -> movies, origin/movies)
Author: raghuponde <33595609+raghuponde@users.noreply.github.com>
Date: Mon Jul 17 21:36:32 2023 +0530
```

Create Lion.txt

```
commit 419d60e6f5c423564de664e312767540e758e633
Author: raghuponde <33595609+raghuponde@users.noreply.github.com>
Date: Mon Jul 1 18:54:50 2023 +0530
```

Create jamesbond

```
commit d94a89ed4af37475ecb89634c0aa31c44f4b2836
Author: Colt Steele <5498438+Colt@users.noreply.github.com>
Date: Mon Feb 1 16:42:21 2021 -0800
```

Create tinkerbell.txt

```
commit 6c32e319078957ebbf9a0df486b1afbd9a3952e7
Author: Colt Steele <5498438+Colt@users.noreply.github.com>
Date: Mon Feb 1 14:05:17 2021 -0800
```

Create bambi.txt

```
commit 335cbf5f7edb25a79cb666a93ee464a461089f4e
Author: Colt Steele <5498438+Colt@users.noreply.github.com>
Date: Mon Feb 1 14:04:48 2021 -0800
```

Create ghostbusters.txt

```
commit 2cfeb8a1982ad1f684a93b945ca95eacb901c8f0
Author: Colt Steele <5498438+Colt@users.noreply.github.com>
Date: Mon Feb 1 14:04:32 2021 -0800
```

Delete rose.txt

```
commit ed1175a5111acb5d7461b0aa80df0d311295a08b
Author: Colt Steele <5498438+Colt@users.noreply.github.com>
Date: Mon Feb 1 14:04:24 2021 -0800
```

Delete cactus.txt

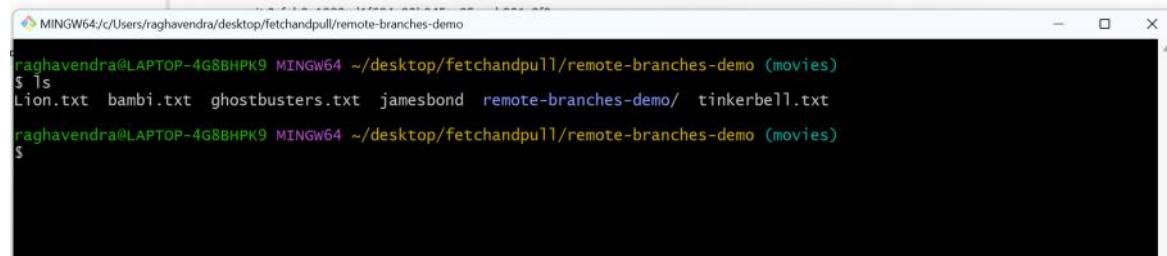
```
commit 7fdeb8873db735efaa8d6d09c43b245c5697052c (main)
Author: Colt Steele <5498438+Colt@users.noreply.github.com>
Date: Mon Feb 1 14:03:28 2021 -0800
```

Create rose.txt

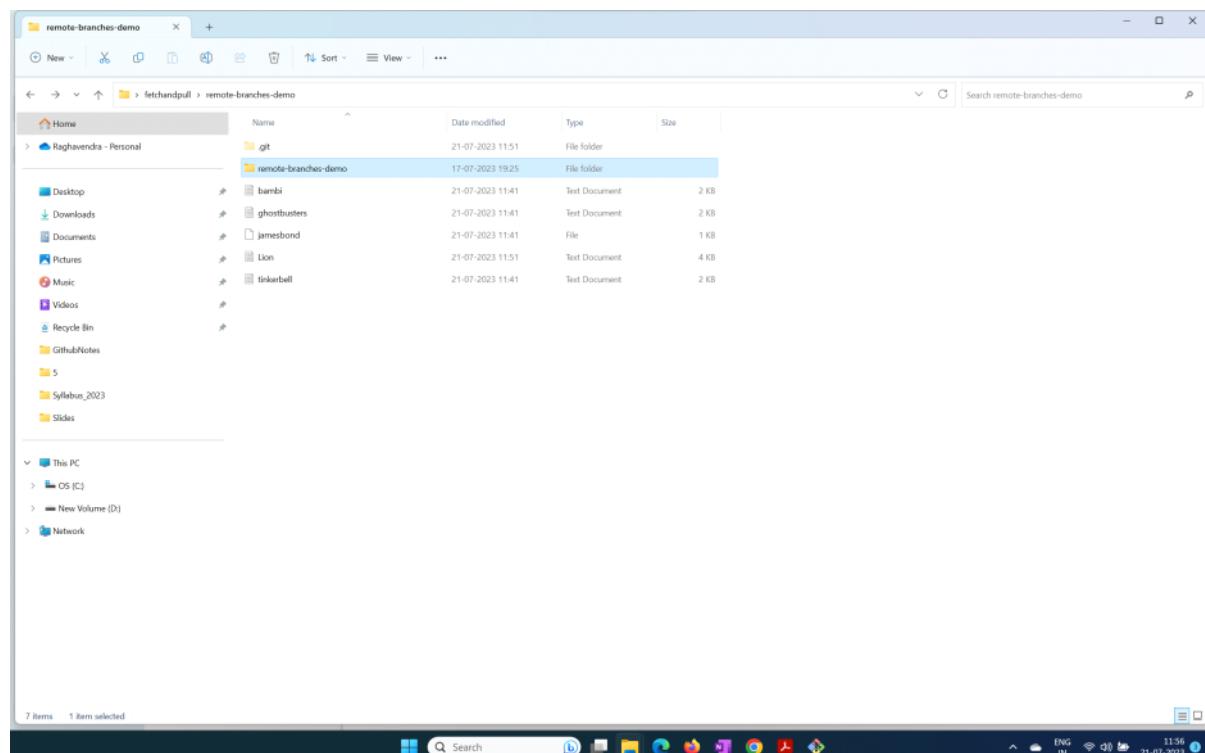
:

So now my movies branch and origin movies branch both are pointing to exact same commit okay ...

Now if I type ls we can see the lion file also go to vs code and in particular to that folder and check it



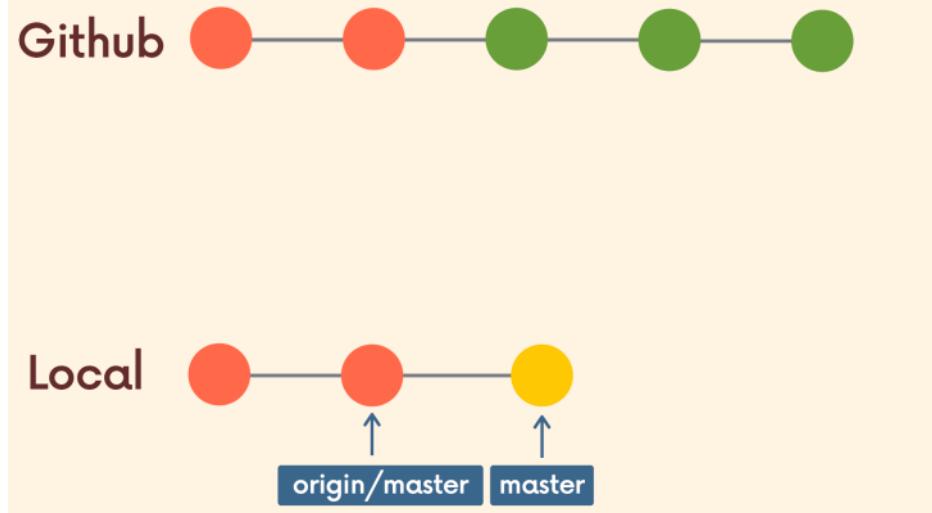
```
MINGW64/c/Users/raghavendra/Desktop/fetchandpull/remote-branches-demo
raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/Desktop/fetchandpull/remote-branches-demo (movies)
$ ls
Lion.txt bambi.txt ghostbusters.txt jamesbond remote-branches-demo/ tinkerbell.txt
raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/Desktop/fetchandpull/remote-branches-demo (movies)
$
```



So you can see the lion It is affecting my directory okay ..

Here pull has not resulted in merge conflicts As it was fast forward merge now let us see an example of merge conflicts now and main reason of this merge conflict is that when you are working on same file means many users are working on same file okay

Now u have done some changes in github or any collaborator has done It



Now before pulling those changes u have done some changes in local and then u are applying pull command then conflicts will come let us see one demo then we will understand it well

And then u have pulled it down like this

Now I am going to food

```
raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop/fetchandpull/remote-branches-demo (movies)
$ git switch food
Switched to branch 'food'
Your branch is behind 'origin/food' by 1 commit, and can be fast-forwarded.
(use "git pull" to update your local branch)
```

```
raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop/fetchandpull/remote-branches-demo (food)
$ ls
apple.txt banana.txt cake.txt coffee.txt popsicle.txt remote-branches-demo/ tea.txt
```

```
raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop/fetchandpull/remote-branches-demo (food)
$ git status
On branch food
Your branch is behind 'origin/food' by 1 commit, and can be fast-forwarded.
(use "git pull" to update your local branch)
```

```
Untracked files:
(use "git add <file>..." to include in what will be committed)
 remote-branches-demo/
```

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

```
raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop/fetchandpull/remote-branches-demo (food)
$
```

So if I observe I am missing icecream in local okay now next what I can do I can pull but instead of pulling I will make some changes in food branch locally means say I am creating locally a file with name icecream.txt okay

And there don't put the icecream diagram which is there in github okay  
Put your own diagram from acii

And then say git status  
And then git add and git commit etc stuff which is shown in below pic okay

```
[MINGW64:/c/Users/raghavendra/desktop/fetchandpull/remote-branches-demo]
raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop/fetchandpull/remote-branches-demo (food)
$ ls
apple.txt banana.txt cake.txt coffee.txt popsicle.txt remote-branches-demo/ tea.txt
raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop/fetchandpull/remote-branches-demo (food)
$ touch icecream.txt

raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop/fetchandpull/remote-branches-demo (food)
$ git status
On branch food
Your branch is behind 'origin/food' by 1 commit, and can be fast-forwarded.
  (use "git pull" to update your local branch)

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    icecream.txt
    remote-branches-demo/

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

raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop/fetchandpull/remote-branches-demo (food)
$ git add .
warning: adding embedded git repository: remote-branches-demo
hint: you've added another git repository inside your current repository.
hint: Clones of the outer repository will not contain the contents of
hint: the embedded repository and will not know how to obtain it.
hint: If you meant to add a submodule, use:
hint:
hint:   git submodule add <url> remote-branches-demo
hint:
hint: If you added this path by mistake, you can remove it from the
hint: index with:
hint:
hint:   git rm --cached remote-branches-demo
hint:
hint: see "git help submodule" for more information.

raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop/fetchandpull/remote-branches-demo (food)
$ git commit -m "icecream my own art added"
[food b0ea610] icecream my own art added
 2 files changed, 25 insertions(+)
 create mode 100644 icecream.txt
 create mode 160000 remote-branches-demo

raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop/fetchandpull/remote-branches-demo (food)
$ |
```

So after this I can do two things I can push the new icecream to github but good practise is to first pull the changes and then u can push it okay

Also notice here now I will fire a pull command so git has one commit which is not existing in github and also github also has one commit which is not existin in local and both are in food branch only

So u can see now conflict has come

```
[MINGW64:/c/Users/raghavendra/desktop/fetchandpull/remote-branches-demo]
raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop/fetchandpull/remote-branches-demo (food)
$ git pull origin food
From https://github.com/raghuponde/remote-branches-demo
 * branch      food      -> FETCH_HEAD
Auto-merging icecream.txt
CONFLICT (add/add): Merge conflict in icecream.txt
Automatic merge failed; fix conflicts and then commit the result.

raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop/fetchandpull/remote-branches-demo (food|MERGING)
$ |
```

So this is the conflict I am getting when I typed now code . And went to icecream file locally

```

1 <<<<< HEAD (Current Change)
2   .
3   oooooooo
4   6666666666
5   .
6   WWWWWWWWWWWGEEEEE
7   WWWWWWWWWWWZICKEEEEEE
8   WWWWWWWWWWWHGEEEEEE
9   WWWWWWWWWWWHOEEEEE
10  WWWWWWWWWWWOOOOOOOEI/a
11  MWW1111111111111111
12  MWW1111111111111111
13  |MWW1111111111111111
14  |MWW1111111111111111
15  MWW1111111111111111
16  1 1111111111111111
17  . 1111111111111111
18  "1111111111111111"
19  "1111111111111111"
20  "1111111111111111"
21  "1111111111111111"
22  "1111111111111111"
23  "1111111111111111"
24  "1111111111111111"
25  "1111111111111111"
26  ****
27
28
29  |"7 /h
30  // /h /
31  // /h /h
32  // /h /h /
33  // /h /h /
34  C: // /h /
35  "F: // /h
36  // /F: /h
37  // / " "
38  // / "
39  / / hh
40  >>>> bd4239d01ce90da8cc98b32f799fc1fd495493b (Incoming Change)
41

```

I am accepting both the changes now from the file okay ..

```

MINGW64:/c/Users/raghavendra/Desktop/fetchandpull/remote-branches-demo (food)
$ git pull origin food
From https://github.com/raghuponde/remote-branches-demo
 * branch      food      -> FETCH_HEAD
Auto-merging icecream.txt
CONFLICT (add/add): Merge conflict in icecream.txt
Automatic merge failed; fix conflicts and then commit the result.

raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/Desktop/fetchandpull/remote-branches-demo (Food|MERGING)
$ code .

raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/Desktop/fetchandpull/remote-branches-demo (Food|MERGING)
$ git status
On branch food
Your branch and 'origin/food' have diverged,
and have 1 and 1 different commits each, respectively.
 (use "git pull" to merge the remote branch into yours)

You have unmerged paths.
 (fix conflicts and run "git commit")
 (use "git merge --abort" to abort the merge)

Unmerged paths:
  (use "git add <file>..." to mark resolution)
    both added:    icecream.txt

no changes added to commit (use "git add" and/or "git commit -a")

raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/Desktop/fetchandpull/remote-branches-demo (Food|MERGING)
$ git add .

raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/Desktop/fetchandpull/remote-branches-demo (Food|MERGING)
$ git commit -m "fixed merge conflicts and added both icecreams"
[food 4b1a589] fixed merge conflicts and added both icecreams

raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/Desktop/fetchandpull/remote-branches-demo (Food)
$ git status
On branch food
Your branch is ahead of 'origin/food' by 2 commits.
 (use "git push" to publish your local commits)

nothing to commit, working tree clean

raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/Desktop/fetchandpull/remote-branches-demo (Food)
$
```

So after doing all stuff if I finally type git status it is saying it is still ahead of github means I need to push I guess let us see what to do so for better understanding u can see that log also by pressing or writing git log

So here below after pushing it is synchronized

```

raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop/fetchandpull/remote-branches-demo (food)
$ git push origin food
Enumerating objects: 9, done.
Counting objects: 100% (9/9), done.
Delta compression using up to 20 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (6/6), 820 bytes | 820.00 KiB/s, done.
Total 6 (delta 3), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (3/3), completed with 1 local object.
To https://github.com/raghuponde/remote-branches-demo.git
  bd4239d..4b1a589 food -> food

raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop/fetchandpull/remote-branches-demo (food)
$ git status
On branch food
Your branch is up to date with 'origin/food'.

nothing to commit, working tree clean

raghavendra@LAPTOP-4G8BHPK9 MINGW64 ~/desktop/fetchandpull/remote-branches-demo (food)
$ 

```

So now u go and see that another icecream of cone has come in github or not by refresing the github website okay ..

