NAME: Dhruv Bhavesh Bhanushali
DIVISION: IT A
ROLL NO: 08
DevOps

EXPERIMENT NUMBER 2

AIM: To study Version Control using GIT.

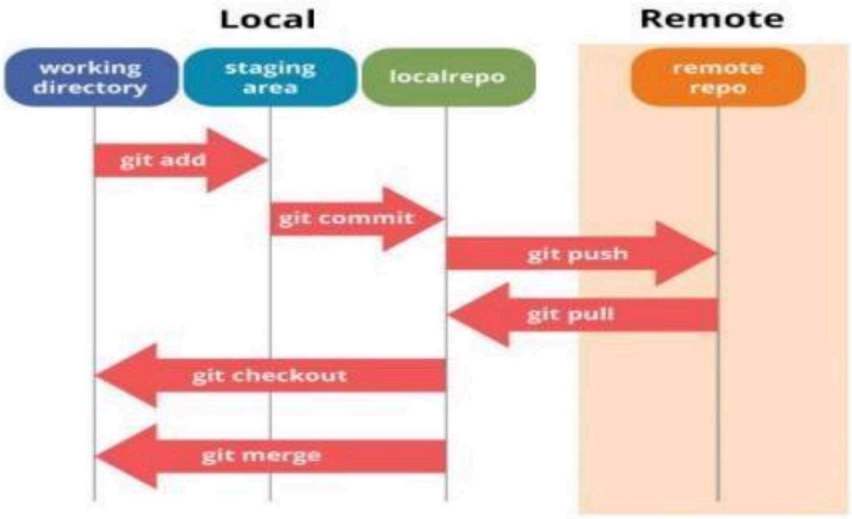
THEORY:

Git is a free and open-source distributed version control system designed to handle everything from small to very large projects with speed and efficiency. Git is easy to learn and has a tiny footprint with lightning-fast performance. It outclasses SCM tools like Subversion, CVS, Perforce, and ClearCase with features like cheap local branching, convenient staging areas, and multiple workflows.

Some of the basic operations in Git are:
1. Initialize
2. Add
3. Commit
4. Pull
5. Push

Some advanced Git operations are:
1. Branching
2. Merging
3. Rebasing

The following diagram depict the all supported operations in GIT

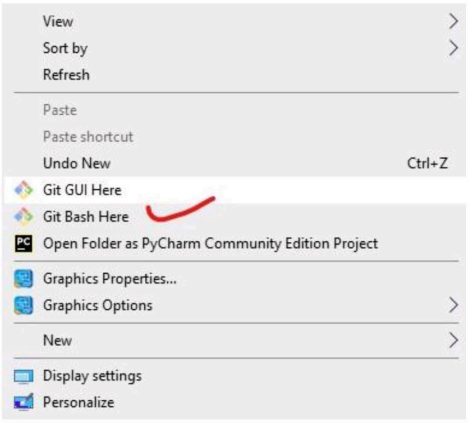


Installation of GIT

1) In windows, download GIT from https://git-scm.com/ and perform the straightforward installation.

2) In Ubuntu, install GIT using $sudo apt install git, Confirm the version after installation $git --version

Once installation is done, open the terminal in Ubuntu and perform the following steps or in windows Right click and select Git bash here.



EXECUTION:

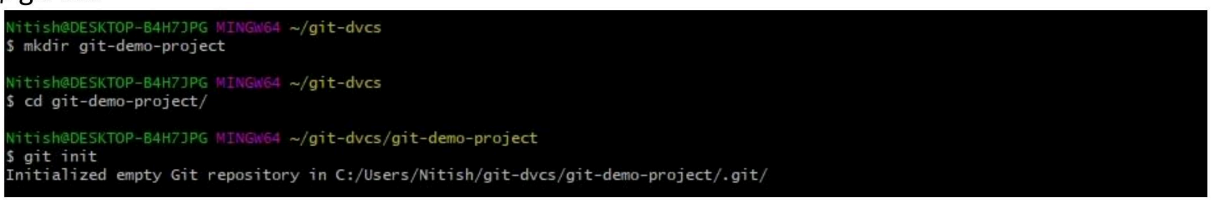
To perform version control, let us create a directory dvcs (Distributed version control system) and change directory to dvcs.
$ mkdir git-dvcs
$ cd git-dvcs/

Now check the user information using
$ git config –global

As there are no users defined, let us define it using following two commands
$ git config --global user.name "Nitish"
$ git config --global user.email dhruvchandra01.dc@gmail.com

Now, check the list of users
$ git config --global --list user.name=Nitish
user.email=dhruvchandra01.dc@gmail.com

Let us create a repository for version control named ”git-demo-project”
$ mkdir git-demo-project
$ cd git-demo-project/
Now, initialize the repository using following command
$ git init



If you have existing repository, then simply delete .git file and reinitialize it. $
rm -rf .git/

$ git init
Initialized empty Git repository in C:/Users/ADMIN/Desktop/git-dvcs/git-demo- project/.git/
Now, let us add some files inside our repository “git-demo-project”
To add files in index and staging area, add command is used along with dot (. Dot means current directory) $ git add .

Index and staging area
To check the status of repository, use
$ git status
Which will show you some untrack files, so untracks files can be tracked using commit command.
Now, let us commit the changes
$ git commit -m "First Commit" (#here -m for message)

Add index.html in our directory $
git add .

$ git commit -am "express Commit" (#Here -a used for express commit) $ nano index.html

```
Nitish@DESKTOP-B4H7JPG MINGW64 ~/git-dvcs/git-demo-project (master)
$ git add .

Nitish@DESKTOP-B4H7JPG MINGW64 ~/git-dvcs/git-demo-project (master)
$ git status
On branch master

No commits yet

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

```
Nitish@DESKTOP-B4H7JPG MINGW64 ~/git-dvcs/git-demo-project (master)
$ git commit -m "First Commit"
On branch master

Initial commit

nothing to commit
```

```
Nitish@DESKTOP-B4H7JPG MINGW64 ~/git-dvcs/git-demo-project (master)
$ git status
On branch master

No commits yet

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

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

```
Nitish@DESKTOP-B4H7JPG MINGW64 ~/git-dvcs/git-demo-project (master)
$ git add .
warning: LF will be replaced by CRLF in index.html.
The file will have its original line endings in your working directory

Nitish@DESKTOP-B4H7JPG MINGW64 ~/git-dvcs/git-demo-project (master)
$ git commit -am "express Commit"
[master (root-commit) 9c140b8] express Commit
 1 file changed, 1 insertion(+)
 create mode 100644 index.html
```

```
Nitish@DESKTOP-B4H7JPG MINGW64 ~/git-dvcs/git-demo-project (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:   index.html

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

$ touch teststatus

```
Nitish@DESKTOP-B4H7JPG MINGW64 ~/git-dvcs/git-demo-project (master)
$ touch teststatus

Nitish@DESKTOP-B4H7JPG MINGW64 ~/git-dvcs/git-demo-project (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:   index.html

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

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

```
Nitish@DESKTOP-B4H7JPG MINGW64 ~/git-dvcs/git-demo-project (master)
$ git checkout -- teststatus
error: pathspec 'teststatus' did not match any file(s) known to git

Nitish@DESKTOP-B4H7JPG MINGW64 ~/git-dvcs/git-demo-project (master)
$ git checkout -- index.html
```

Changes are Discarded by checkout

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

(use "git restore <file>..." to discard changes in working directory)

```
Nitish@DESKTOP-B4H7JPG MINGW64 ~/git-dvcs/git-demo-project (master)
$ git add index.html

Nitish@DESKTOP-B4H7JPG MINGW64 ~/git-dvcs/git-demo-project (master)
$ git add teststatus
```

$ git add index.html

$ git add teststatus

```
Nitish@DESKTOP-B4H7JPG MINGW64 ~/git-dvcs/git-demo-project (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   teststatus

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:   index.html
```

```
Nitish@DESKTOP-B4H7JPG MINGW64 ~/git-dvcs/git-demo-project (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   index.html
        new file:   teststatus
```

$ git commit -am "Express commit"

```
Nitish@DESKTOP-B4H7JPG MINGW64 ~/git-dvcs/git-demo-project (master)
$ git commit -am "Express Commit"
[master 0eb61c6] Express Commit
 2 files changed, 1 insertion(+), 1 deletion(-)
 create mode 100644 teststatus

Nitish@DESKTOP-B4H7JPG MINGW64 ~/git-dvcs/git-demo-project (master)
$ git status
On branch master
nothing to commit, working tree clean
```

History of Commits

$ git log

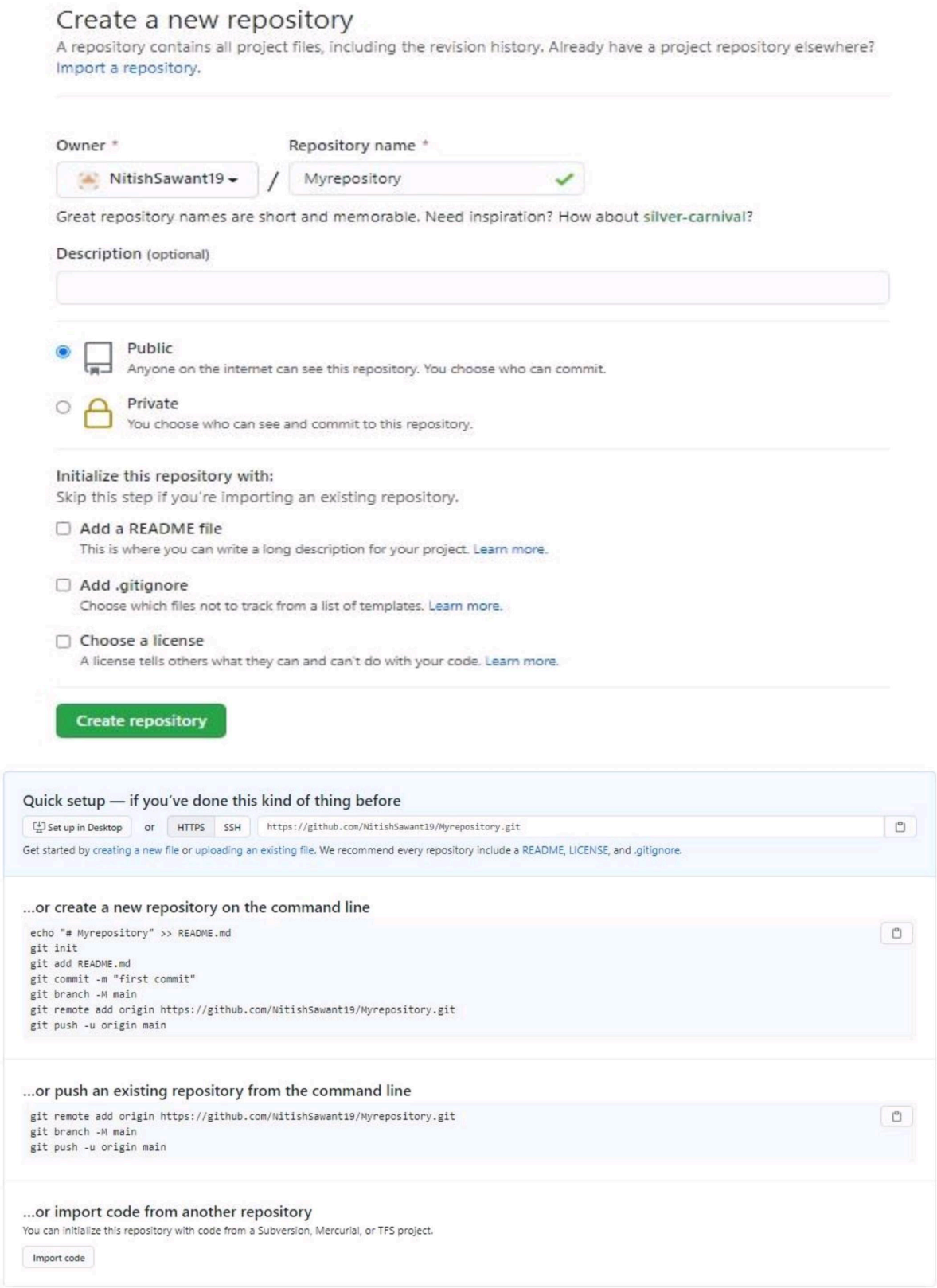
```
Nitish@DESKTOP-B4H7JPG MINGW64 ~/git-dvcs/git-demo-project (master)
$ git log
commit 0eb61c643916742446dbcb284c474aad50fd6fc6 (HEAD -> master)
Author: Nitish <nitish19sawant@gmail.com>
Date:   Tue Aug 17 19:57:23 2021 +0530

    Express Commit

commit 9c140b823e9c2f15d2dd058dd07b90cd3350c8b1
Author: Nitish <nitish19sawant@gmail.com>
Date:   Tue Aug 17 19:43:38 2021 +0530

    express Commit
```

```
Nitish@DESKTOP-B4H7JPG MINGW64 ~/git-dvcs/git-demo-project (master)
$ git log --oneline
0eb61c6 (HEAD -> master) Express Commit
9c140b8 express Commit

Nitish@DESKTOP-B4H7JPG MINGW64 ~/git-dvcs/git-demo-project (master)
$ git log --oneline teststatus
0eb61c6 (HEAD -> master) Express Commit

Nitish@DESKTOP-B4H7JPG MINGW64 ~/git-dvcs/git-demo-project (master)
$ git log --oneline
0eb61c6 (HEAD -> master) Express Commit
9c140b8 express Commit
```

```
Nitish@DESKTOP-B4H7JPG MINGW64 ~/git-dvcs/git-demo-project (master)
$ git log --oneline 9c140b8..0eb61c6
0eb61c6 (HEAD -> master) Express Commit

Nitish@DESKTOP-B4H7JPG MINGW64 ~/git-dvcs/git-demo-project (master)
$ git log --oneline -n 2
0eb61c6 (HEAD -> master) Express Commit
9c140b8 express Commit
```

Now Create a Repository on github.com. Open github.com→ create an account→After login Select New repository from the menu.

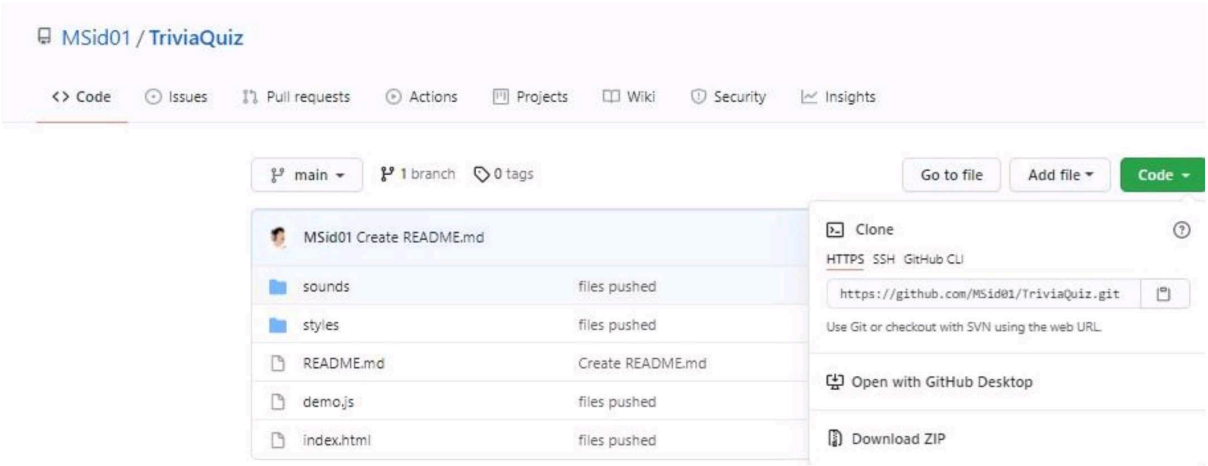
Now Specify a Name to repository and select public option followed by create repository



Now fork the repository (Sharing with other users who wants to contribute).

Login with another account→Copy and Paste URL of repository→then just click on fork to clone to others account.

```
Nitish@DESKTOP-B4H7JPG MINGW64 ~/git-dvcs/git-demo-project (master)
$ git clone https://github.com/MSid01/TriviaQuiz.git
Cloning into 'TriviaQuiz'...
remote: Enumerating objects: 12, done.
remote: Counting objects: 100% (12/12), done.
remote: Compressing objects: 100% (11/11), done.
remote: Total 12 (delta 1), reused 8 (delta 0), pack-reused 0
Unpacking objects: 100% (12/12), done.

Nitish@DESKTOP-B4H7JPG MINGW64 ~/git-dvcs/git-demo-project (master)
$ ls
index.html  Myrepository/  teststatus  TriviaQuiz/
```



Pull and Push Processes

Push → Push changes to Web repository

Pull → Pull changes to Local repository

1) Push command to remote reference origin master

$ git remote add origin https://github.com/MSid01/TriviaQuiz.git

$ git remote show origin

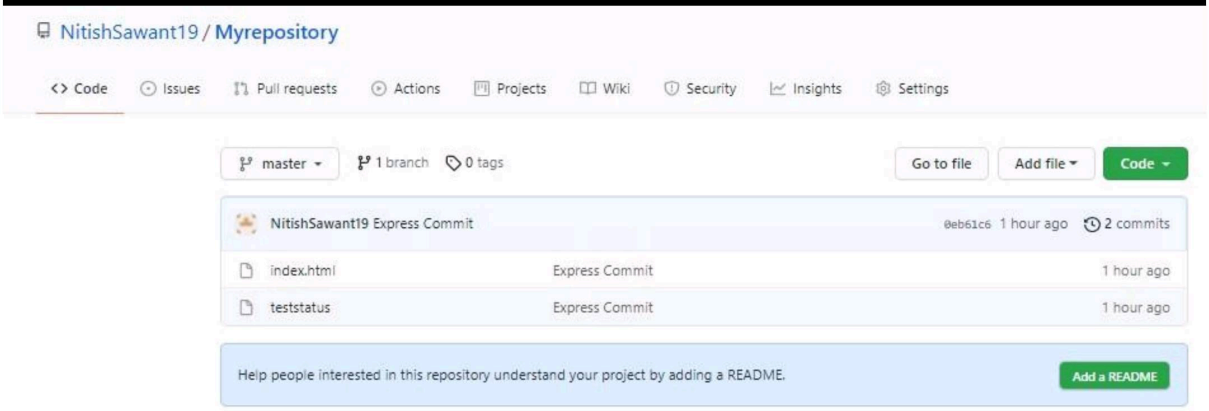
$ git remote add origin https://github.com/bhushanjadhav1/Myrepository.git fatal: remote origin already exists.

$ git remote rm origin

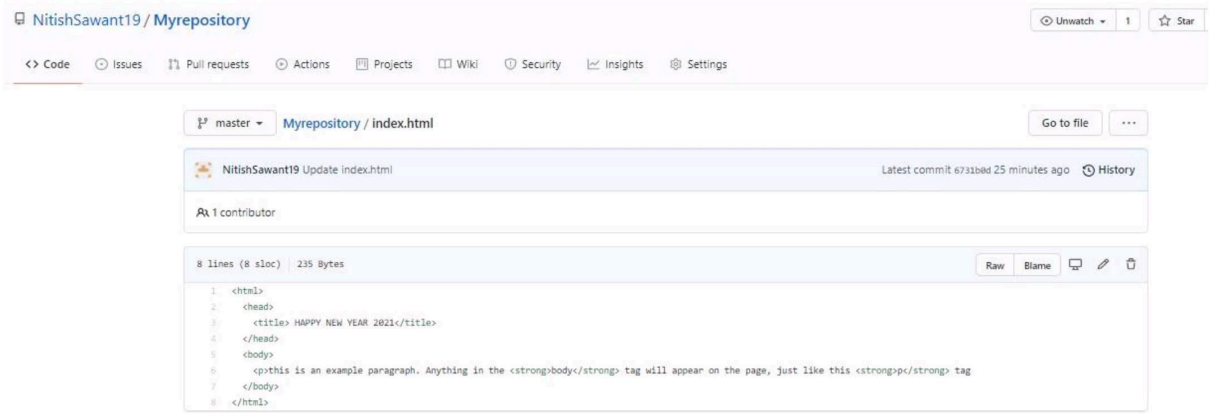
$ git push -u origin master

```
Nitish@DESKTOP-B4H7JPG MINGW64 ~/git-dvcs/git-demo-project (master)
$ git remote add origin https://github.com/MSid01/TriviaQuiz.git

Nitish@DESKTOP-B4H7JPG MINGW64 ~/git-dvcs/git-demo-project (master)
$ git remote show origin
* remote origin
  Fetch URL: https://github.com/MSid01/TriviaQuiz.git
  Push  URL: https://github.com/MSid01/TriviaQuiz.git
  HEAD branch: main
  Remote branch:
    main new (next fetch will store in remotes/origin)

Nitish@DESKTOP-B4H7JPG MINGW64 ~/git-dvcs/git-demo-project (master)
$ git remote add origin https://github.com/NitishSawant19/Myrepository.git
fatal: remote origin already exists.

Nitish@DESKTOP-B4H7JPG MINGW64 ~/git-dvcs/git-demo-project (master)
$ git remote rm origin
```

```
Nitish@DESKTOP-B4H7JPG MINGW64 ~/git-dvcs/git-demo-project (master)
$ git push --set-upstream https://github.com/NitishSawant19/Myrepository.git master
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 2 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (7/7), 502 bytes | 251.00 KiB/s, done.
Total 7 (delta 0), reused 0 (delta 0)
To https://github.com/NitishSawant19/Myrepository.git
 * [new branch]      master -> master
Branch 'master' set up to track remote branch 'master' from 'https://github.com/NitishSawant19/Myrepository.git'.
```



Pull Changes

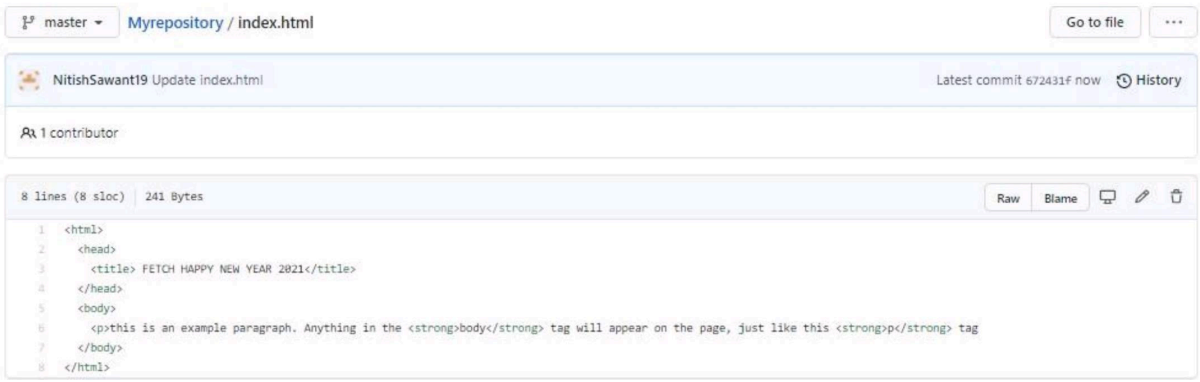


$ git pull

```
Nitish@DESKTOP-B4H7JPG MINGW64 ~/git-dvcs/git-demo-project (master)
$ git pull
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), 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), done.
From https://github.com/NitishSawant19/Myrepository
 * branch            master     -> FETCH_HEAD
Updating c8d99b3..6731b0d
Fast-forward
 index.html | 9 ++++++++-
 1 file changed, 8 insertions(+), 1 deletion(-)
```

Fetch

```
$ git log --oneline origin/master
9f72b33 (origin/master) changed
d3a6a76 Express commit
97d0a76 express Commit
```



$ git fetch

Here fetch will not show you like updated changes file as like push. So use merge command to merge the changes.

```
$ git fetch
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), done.
From https://github.com/NitishSawant19/Myrepository
 * branch            master     -> FETCH_HEAD
```

```
Nitish@DESKTOP-B4H7JPG MINGW64 ~/git-dvcs/git-demo-project (master)
$ git log --oneline origin/master
21e9ada (origin/master) Fetch
9f72b33 changed
d3a6a76 Express commit
97d0a76 express Commit
```

$ git merge origin/master

```
Nitish@DESKTOP-B4H7JPG MINGW64 ~/git-dvcs/git-demo-project (master)
$ cat index.html
<html>
  <head>
    <title>FETCH..HAPPY NEW YEAR 2021</title>
  </head>
  <body>
    <p>this is an example paragraph. Anything in the <strong>body</strong> tag will appear on the page, just like this <strong>p</strong> tag
  </body>
</html>
```

```
$ git merge
Merge made by the 'recursive' strategy.
 index.html | 2 +-
 1 file changed, 1 insertion(+), 1 deletion(-)
```

```
$ cat index.html
<html>
  <head>
    <title>HAPPY NEW YEAR 2021</title>
  </head>
  <body>
    <p>this is an example paragraph. Anything in the <strong>body</strong> tag will appear on the page, just like this <strong>p</strong> tag
  </body>
</html>

Nitish@DESKTOP-B4H7JPG MINGW64 ~/git-dvcs/git-demo-project (master)
$
```