

Git Assignment

Assignment 1: Initialize a new Git repository in a directory of your choice. Add a simple text file to the repository and make the first commit.

Initialize a new Git Repository:

Step 1: Firstly we need to create an account in [GitHub](#). After that we need to create a new Repository with a name in the GitHub where all the files be stored.

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Required fields are marked with an asterisk ().*

Owner * harishrongala123 / **Repository name *** Wipro Assignment

✓ Your new repository will be created as Wipro-Assignment.
The repository name can only contain ASCII letters, digits, and the characters -, ., and _.

Great repository names are short and memorable. Need inspiration? How about [potential-fortnight](#) ?

Description (optional)

☒ **Public**
Anyone on the internet can see this repository. You choose who can commit.

☐ **Private**
You choose who can see and commit to this repository.

Initialize this repository with:

☒ **Add a README file**
This is where you can write a long description for your project. [Learn more about READMEs.](#)

Add .gitignore
.gitignore template: **None**

Choose which files not to track from a list of templates. [Learn more about ignoring files.](#)

Choose a license
License: **None**

A license tells others what they can and can't do with your code. [Learn more about licenses.](#)

ⓘ You are creating a public repository in your personal account.

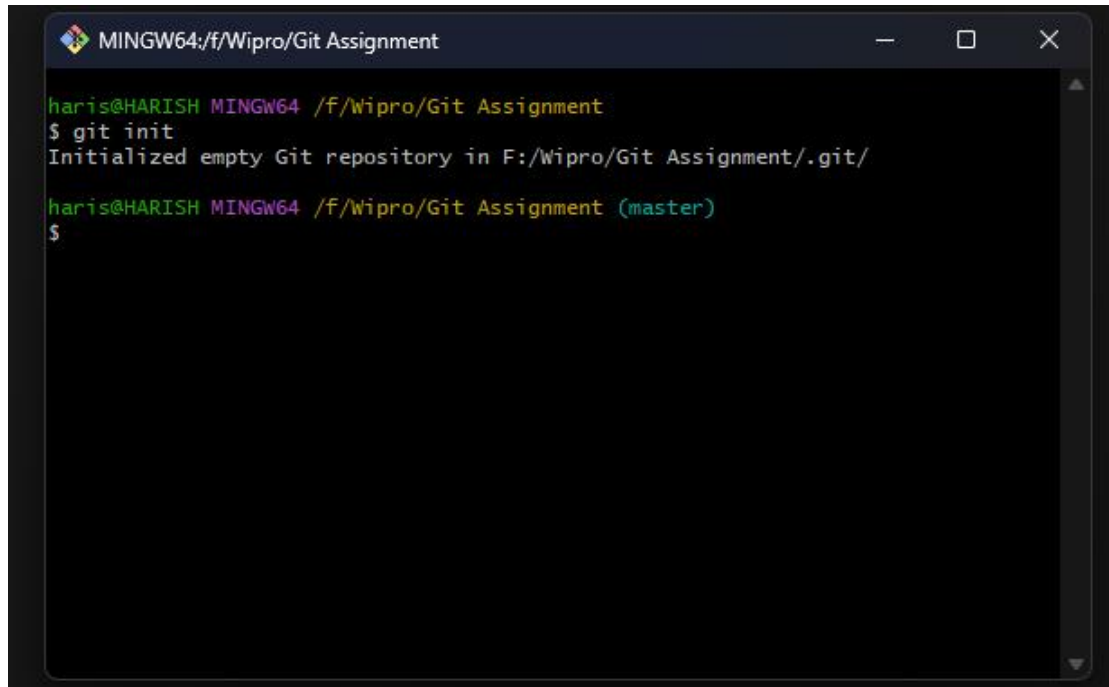
Create repository

Step 2:

Make Sure we have Git installed in the PC. Go to the file which we need to add in the repository and open the git Bash or command prompt with the same address.

Step 3:

In the Command prompt use "**git init**" command to initialize the Local Repository

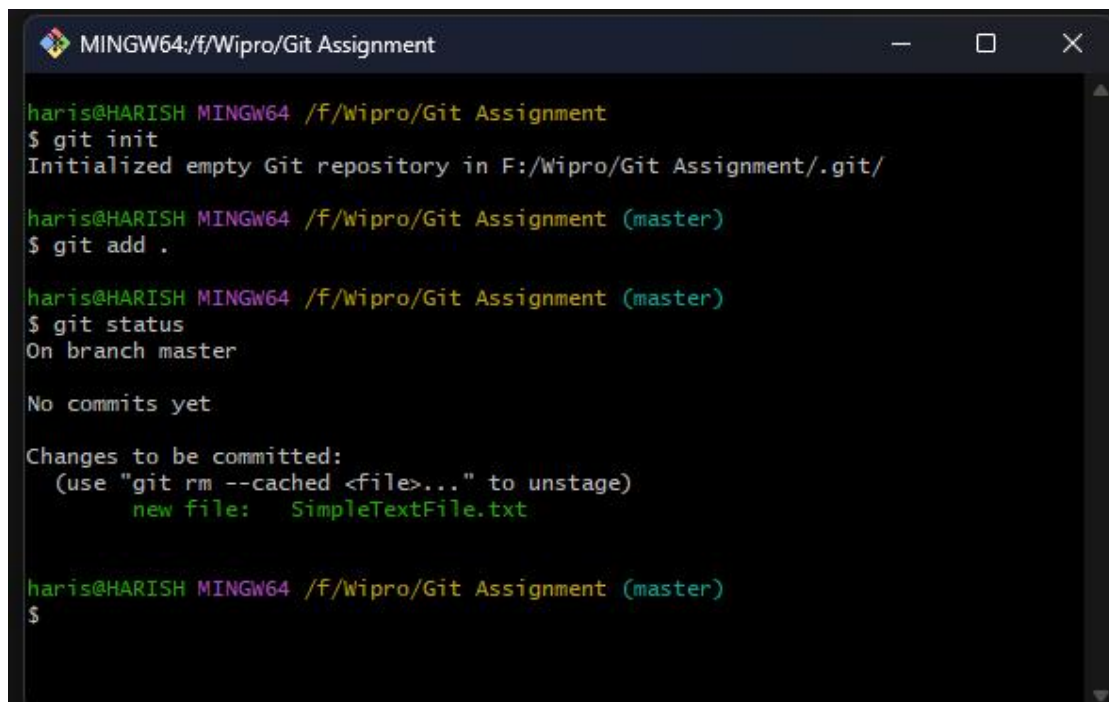
A terminal window titled 'MINGW64:/f/Wipro/Git Assignment' with standard window controls. The prompt is 'haris@HARISH MINGW64 /f/Wipro/Git Assignment'. The command '\$ git init' is entered, and the output is 'Initialized empty Git repository in F:/Wipro/Git Assignment/.git/'. The prompt then changes to 'haris@HARISH MINGW64 /f/Wipro/Git Assignment (master)' and '\$' is entered on the next line.

```
haris@HARISH MINGW64 /f/Wipro/Git Assignment
$ git init
Initialized empty Git repository in F:/Wipro/Git Assignment/.git/

haris@HARISH MINGW64 /f/Wipro/Git Assignment (master)
$
```

Step 4:

To add all the files in the local repository we use "**git add .**" command. Here dot represents all the files. "git status" command is used to show the status of the local repository.

A terminal window titled 'MINGW64:/f/Wipro/Git Assignment' with standard window controls. The prompt is 'haris@HARISH MINGW64 /f/Wipro/Git Assignment'. The command '\$ git init' is entered, and the output is 'Initialized empty Git repository in F:/Wipro/Git Assignment/.git/'. The prompt then changes to 'haris@HARISH MINGW64 /f/Wipro/Git Assignment (master)'. The command '\$ git add .' is entered. The prompt changes to 'haris@HARISH MINGW64 /f/Wipro/Git Assignment (master)'. The command '\$ git status' is entered, and the output is 'On branch master', 'No commits yet', and 'Changes to be committed: (use "git rm --cached <file>..." to unstage) new file: SimpleTextFile.txt'. The prompt then changes to 'haris@HARISH MINGW64 /f/Wipro/Git Assignment (master)' and '\$' is entered on the next line.

```
haris@HARISH MINGW64 /f/Wipro/Git Assignment
$ git init
Initialized empty Git repository in F:/Wipro/Git Assignment/.git/

haris@HARISH MINGW64 /f/Wipro/Git Assignment (master)
$ git add .

haris@HARISH MINGW64 /f/Wipro/Git Assignment (master)
$ git status
On branch master

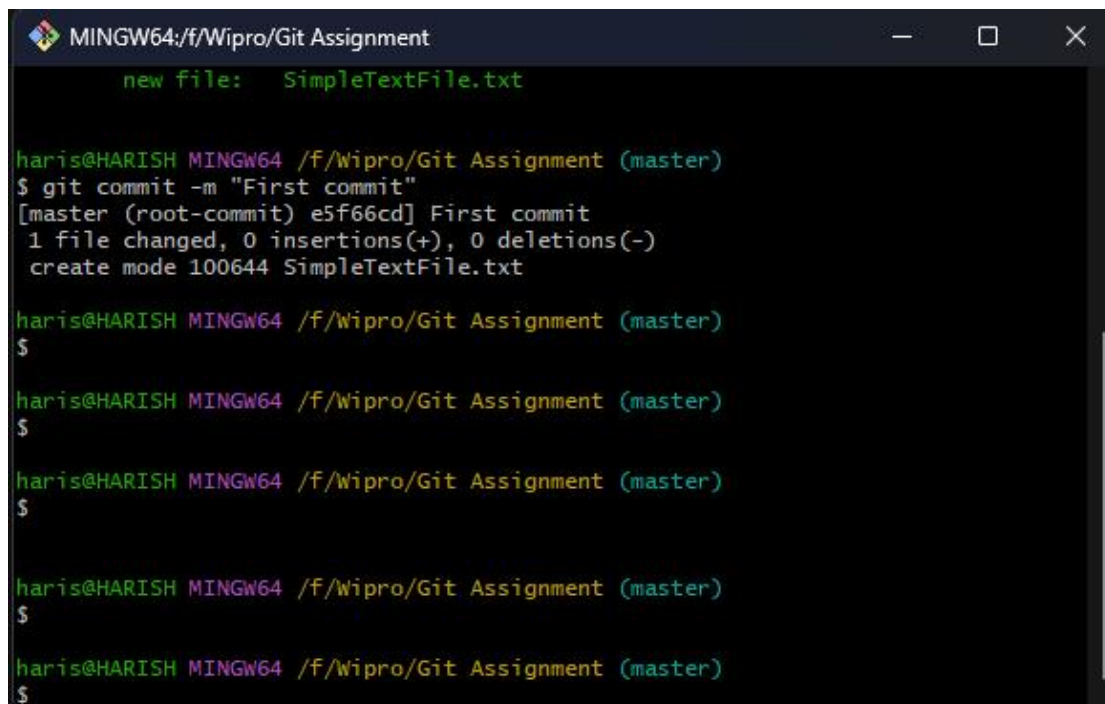
No commits yet

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

haris@HARISH MINGW64 /f/Wipro/Git Assignment (master)
$
```

Step 5:

Commit the added file using "git commit -m"

A terminal window titled 'MINGW64:/f/Wipro/Git Assignment' with standard window controls. It shows the execution of 'git commit -m "First commit"'. The output indicates a successful commit on the master branch with hash e5f66cd, showing 1 file changed and 0 insertions or deletions. The file 'SimpleTextFile.txt' is listed as created. Subsequent prompts for the commit message are shown without input.

```
MINGW64:/f/Wipro/Git Assignment

new file:   SimpleTextFile.txt

haris@HARISH MINGW64 /f/Wipro/Git Assignment (master)
$ git commit -m "First commit"
[master (root-commit) e5f66cd] First commit
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 SimpleTextFile.txt

haris@HARISH MINGW64 /f/Wipro/Git Assignment (master)
$

haris@HARISH MINGW64 /f/Wipro/Git Assignment (master)
$

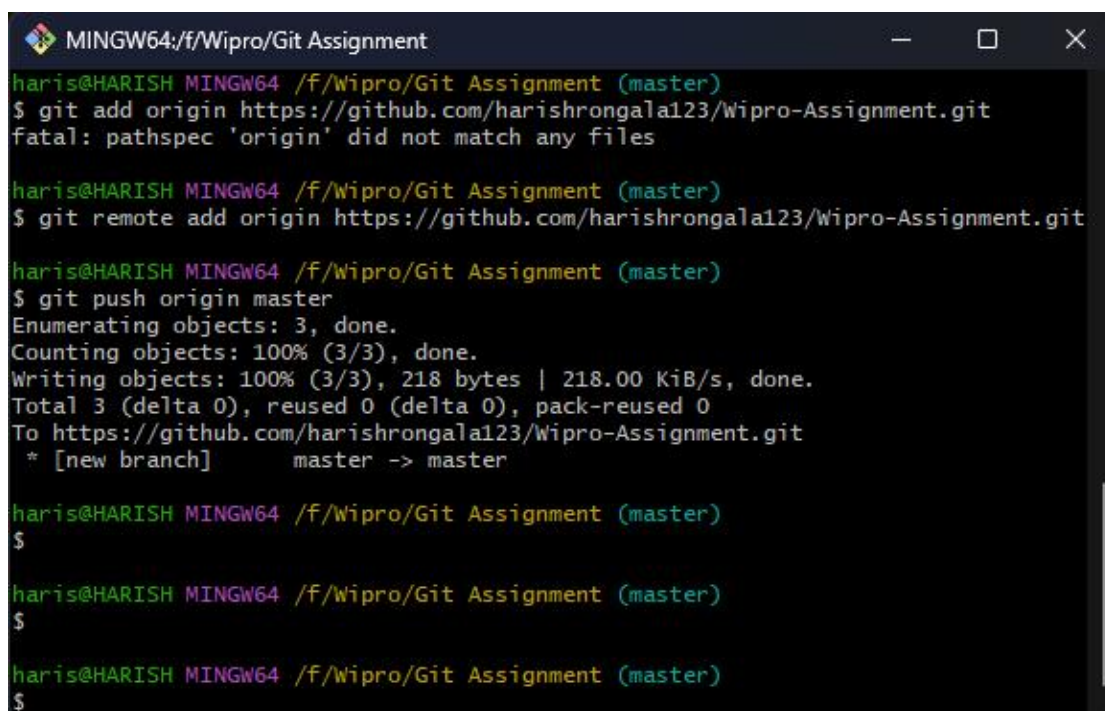
haris@HARISH MINGW64 /f/Wipro/Git Assignment (master)
$

haris@HARISH MINGW64 /f/Wipro/Git Assignment (master)
$

haris@HARISH MINGW64 /f/Wipro/Git Assignment (master)
$
```

Step 6:

Use "git push origin master" command to push all the files which are committed in the GitHub repository

A terminal window titled 'MINGW64:/f/Wipro/Git Assignment' with standard window controls. It shows the execution of 'git push origin master' after adding the remote origin. The output shows the push was successful, enumerating 3 objects and writing them to the repository. The final status is '[new branch] master -> master'. Subsequent prompts are shown without input.

```
MINGW64:/f/Wipro/Git Assignment

haris@HARISH MINGW64 /f/Wipro/Git Assignment (master)
$ git add origin https://github.com/harishrongala123/Wipro-Assignment.git
fatal: pathspec 'origin' did not match any files

haris@HARISH MINGW64 /f/Wipro/Git Assignment (master)
$ git remote add origin https://github.com/harishrongala123/Wipro-Assignment.git

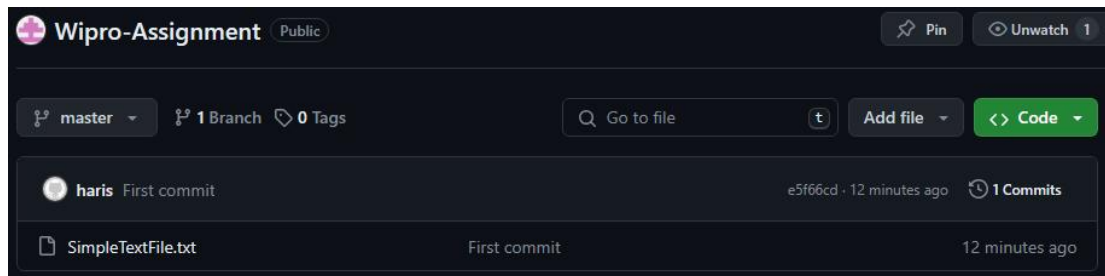
haris@HARISH MINGW64 /f/Wipro/Git Assignment (master)
$ git push origin master
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 218 bytes | 218.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/harishrongala123/Wipro-Assignment.git
 * [new branch]      master -> master

haris@HARISH MINGW64 /f/Wipro/Git Assignment (master)
$

haris@HARISH MINGW64 /f/Wipro/Git Assignment (master)
$

haris@HARISH MINGW64 /f/Wipro/Git Assignment (master)
$
```

We can see the file added in the GitHub



Assignment 2: Branch Creation and Switching

Create a new branch named 'feature' and switch to it. Make changes in the 'feature' branch and commit them

Step 1:

Create a new branch named **"feature"** in the local repository using **"git checkout -b feature"**.

A screenshot of a Windows command prompt window titled "MINGW64:/f/Wipro/Git Assignment". The window has standard Windows window controls (minimize, maximize, close) in the top right corner. The terminal shows the following sequence of commands and output:

```
bash: checkout: command not found  
  
haris@HARISH MINGW64 /f/Wipro/Git Assignment (master)  
$ git checkout -b feature  
Switched to a new branch 'feature'  
  
haris@HARISH MINGW64 /f/Wipro/Git Assignment (feature)  
$  
  
haris@HARISH MINGW64 /f/Wipro/Git Assignment (feature)  
$  
  
haris@HARISH MINGW64 /f/Wipro/Git Assignment (feature)  
$  
  
haris@HARISH MINGW64 /f/Wipro/Git Assignment (feature)  
$  
  
haris@HARISH MINGW64 /f/Wipro/Git Assignment (feature)  
$
```

The prompt changes from "haris@HARISH" to "haris@HARISH MINGW64" after the first successful checkout. Each subsequent "\$" prompt indicates a new command input.

Step 2:

Add a file in the feature branch

```
MINGW64:/f/Wipro/Git Assignment
haris@HARISH MINGW64 /f/Wipro/Git Assignment (feature)
$ git commit -m "feature"
[feature cc275b8] feature
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 Feature.txt

haris@HARISH MINGW64 /f/Wipro/Git Assignment (feature)
$ git push origin feature
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 4 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (2/2), 247 bytes | 123.00 KiB/s, done.
Total 2 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'feature' on GitHub by visiting:
remote:   https://github.com/harishrongala123/Wipro-Assignment/pull/new/feature
remote:
To https://github.com/harishrongala123/Wipro-Assignment.git
 * [new branch]      feature -> feature

haris@HARISH MINGW64 /f/Wipro/Git Assignment (feature)
$
```

Step 3:

In the feature branch we can see the feature file in it

Name	Date modified	Type	Size
.git	15-05-2024 09:25 PM	File folder	
SimpleTextFile	15-05-2024 08:38 PM	Text Document	0 KB
Feature	15-05-2024 09:24 PM	Text Document	0 KB

Step 4:

Once if we change the branch from feature to master, in the same address the master branch has no feature file, use git checkout master to switch from feature to master.

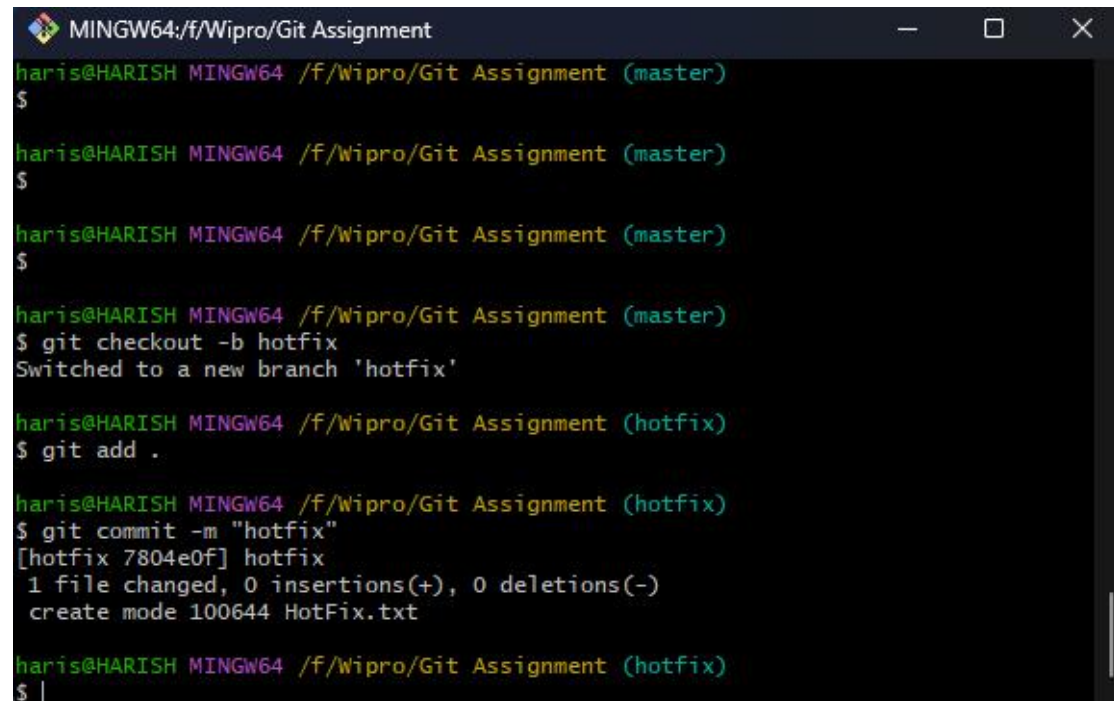
Name	Date modified	Type	Size
.git	15-05-2024 09:35 PM	File folder	
SimpleTextFile	15-05-2024 08:38 PM	Text Document	0 KB

Assignment 3: Feature Branches and Hotfixes

Create a 'hotfix' branch to fix an issue in the main code. Merge the 'hotfix' branch into 'main' ensuring that the issue is resolved.

Step 1:

Create a new branch "hotfix" using "git checkout -b hotfix" and add file.

A screenshot of a Windows command prompt window titled "MINGW64:/f/Wipro/Git Assignment". The window shows a series of terminal commands and their outputs. The user is in the "master" branch. They run "git checkout -b hotfix", which switches to a new branch "hotfix". Then they run "git add .", which adds the current directory. Finally, they run "git commit -m 'hotfix'", which creates a new commit on the "hotfix" branch. The output shows the commit hash "7804e0f" and the file "HotFix.txt" being created.

```
haris@HARISH MINGW64 /f/Wipro/Git Assignment (master)
$

haris@HARISH MINGW64 /f/Wipro/Git Assignment (master)
$

haris@HARISH MINGW64 /f/Wipro/Git Assignment (master)
$

haris@HARISH MINGW64 /f/Wipro/Git Assignment (master)
$ git checkout -b hotfix
Switched to a new branch 'hotfix'

haris@HARISH MINGW64 /f/Wipro/Git Assignment (hotfix)
$ git add .

haris@HARISH MINGW64 /f/Wipro/Git Assignment (hotfix)
$ git commit -m "hotfix"
[hotfix 7804e0f] hotfix
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 HotFix.txt

haris@HARISH MINGW64 /f/Wipro/Git Assignment (hotfix)
$ |
```

Step 2:

Switch from hotfix branch to master branch using "git checkout master" command.

```
MINGW64:/f/Wipro/Git Assignment
haris@HARISH MINGW64 /f/Wipro/Git Assignment (master)
$ git checkout -b hotfix
Switched to a new branch 'hotfix'

haris@HARISH MINGW64 /f/Wipro/Git Assignment (hotfix)
$ git add .

haris@HARISH MINGW64 /f/Wipro/Git Assignment (hotfix)
$ git commit -m "hotfix"
[hotfix 7804e0f] hotfix
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 HotFix.txt

haris@HARISH MINGW64 /f/Wipro/Git Assignment (hotfix)
$ checkout master
bash: checkout: command not found

haris@HARISH MINGW64 /f/Wipro/Git Assignment (hotfix)
$ git checkout master
Switched to branch 'master'

haris@HARISH MINGW64 /f/Wipro/Git Assignment (master)
$ |
```

Step 3:

Merge the master and all the branches using **"git merge master feature"** after that **"git merge master hotfix"**.

```
MINGW64:/f/Wipro/Git Assignment
$ checkout master
bash: checkout: command not found

haris@HARISH MINGW64 /f/Wipro/Git Assignment (hotfix)
$ git checkout master
Switched to branch 'master'

haris@HARISH MINGW64 /f/Wipro/Git Assignment (master)
$ git merge master feature
Updating e5f66cd..cc275b8
Fast-forward
 Feature.txt | 0
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 Feature.txt

haris@HARISH MINGW64 /f/Wipro/Git Assignment (master)
$ git merge master hotfix
Merge made by the 'ort' strategy.
 HotFix.txt | 0
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 HotFix.txt

haris@HARISH MINGW64 /f/Wipro/Git Assignment (master)
$
```

Step 4:

Check the files add from hotfix and feature to master

Name	Date modified	Type	Size
.git	15-05-2024 09:58 PM	File folder	
SimpleTextFile	15-05-2024 08:38 PM	Text Document	0 KB
Feature	15-05-2024 09:58 PM	Text Document	0 KB
HotFix	15-05-2024 09:58 PM	Text Document	0 KB

Step 5:

But in the github the files are not added to do so we need to pull request.

Step 6:

After pull request and approved by the github owner the files get added.

Wipro-Assignment

Public

Pin

Unwatch

1

master

3 Branches

0 Tags

Go to file

Add file

Code

harishrongala123

Merge pull request #2 from harishrongala123/hotfix

1244c3e · now

5 Commits

Feature.txt

feature

45 minutes ago

HotFix.txt

hotfix

20 minutes ago

SimpleTextFile.txt

First commit

1 hour ago