Module 4: Git Workflows

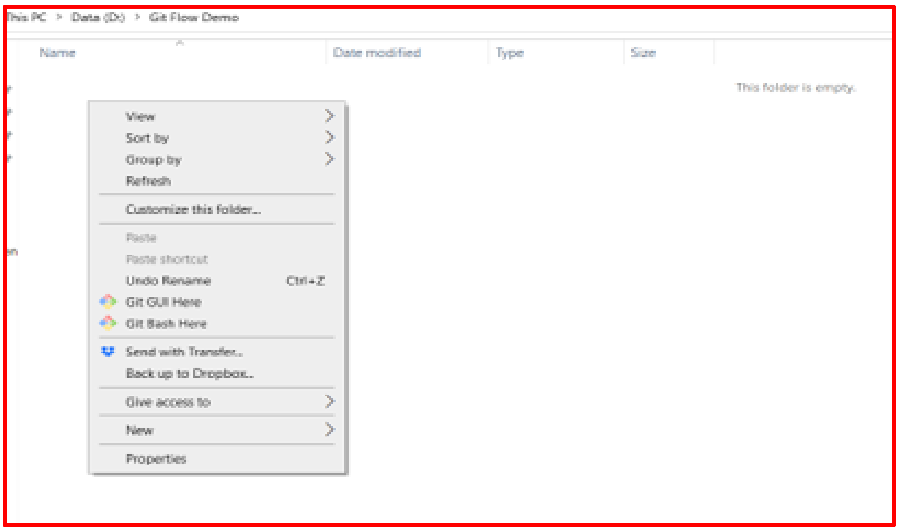
Demo2: Demo on execution on movement of code from feature branch to master/main branch using git flow.

Problem Statement:

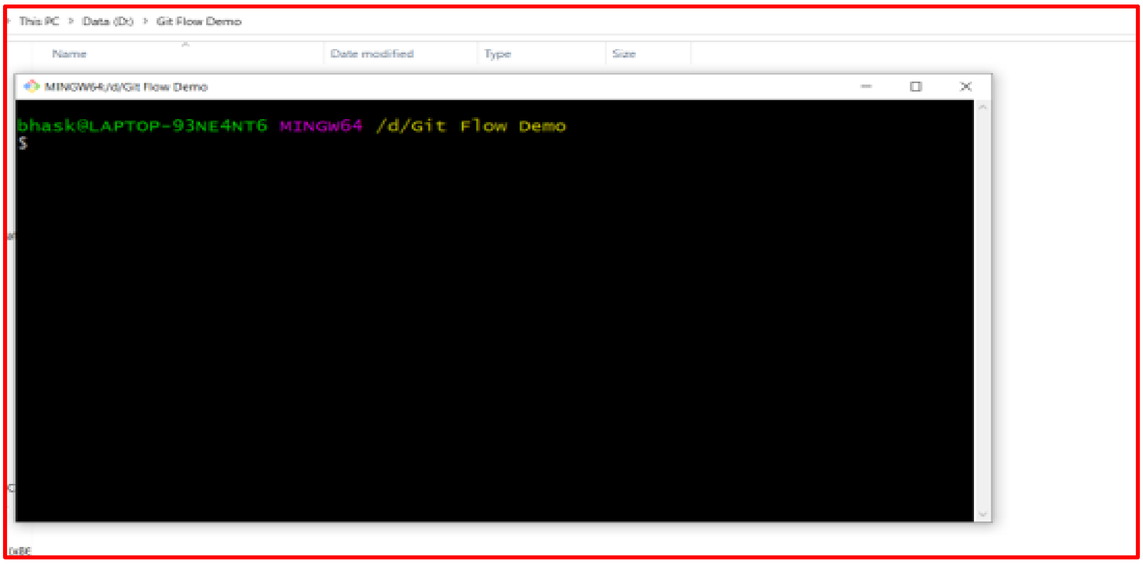
How to move code from feature branch to master/main branch using git flow.

Solution Steps:

1. Navigate to any directory path in you local system and create **NEW FOLDER** as **'Git Flow Demo'** and **ENTER** to the folder. Now right click your mousepad and select **'Git Bash Here'** from the list as shown in below screenshot.



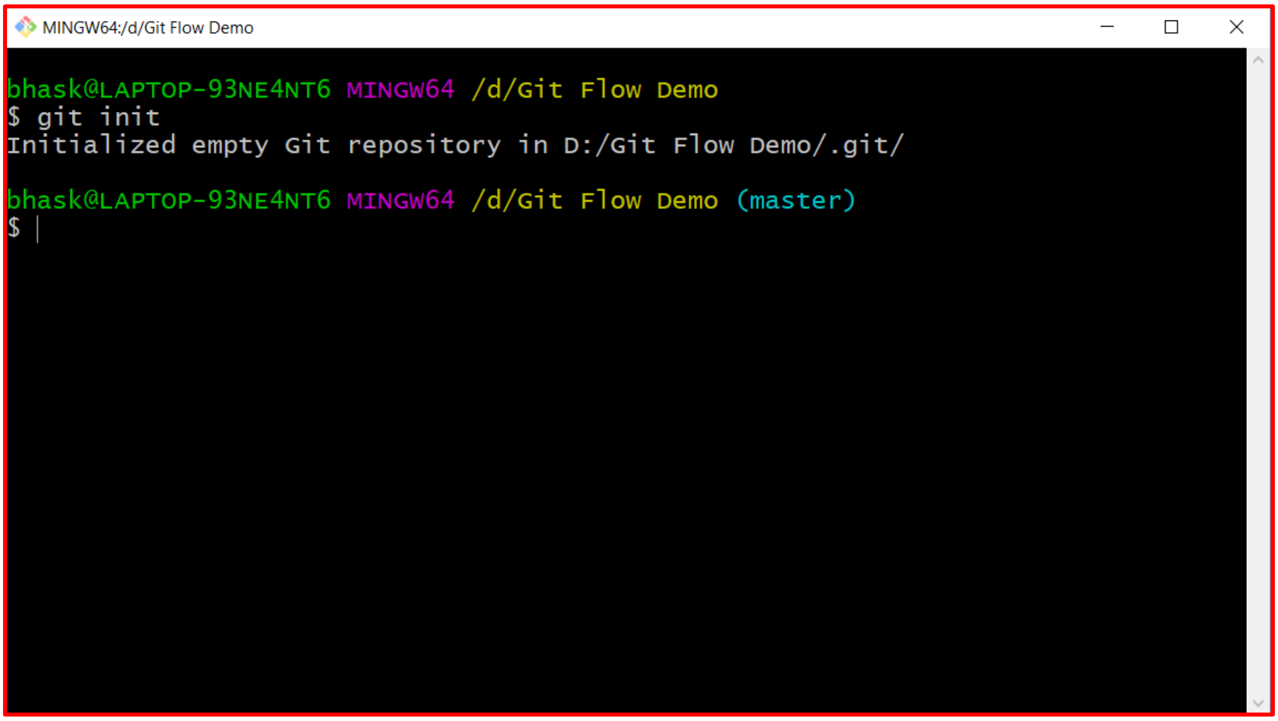
2. **'Git Bash'** launched successfully.



```
bhask@LAPTOP-93NE4NT6 MINGW64 /d/Git Flow Demo
$
```

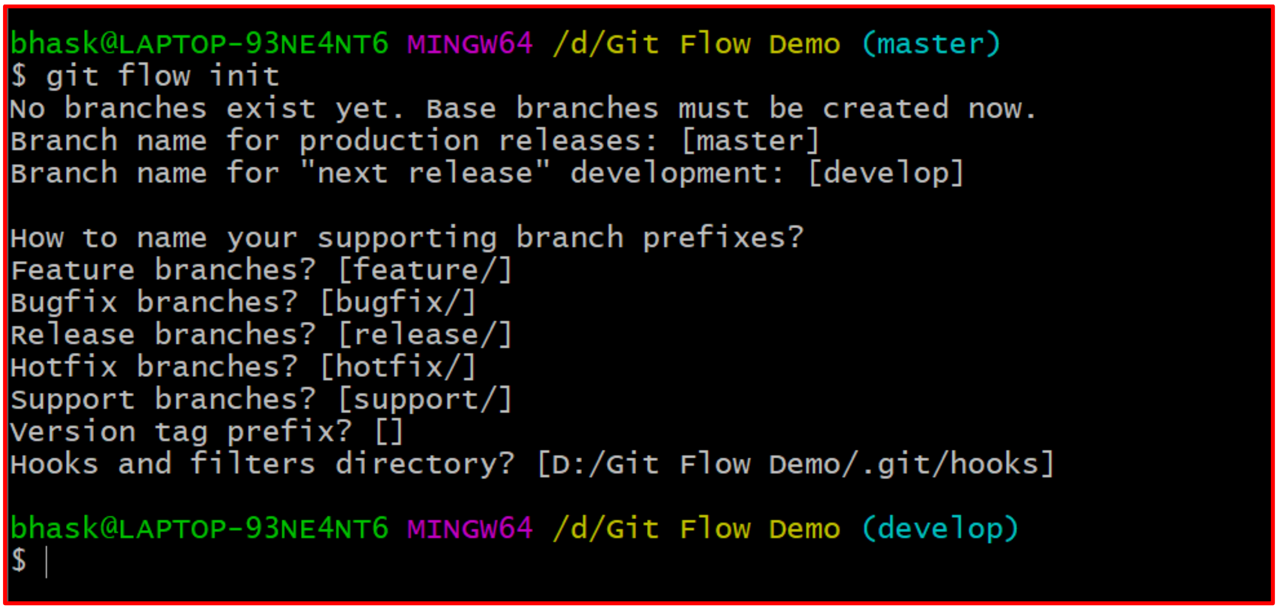
3. Now initialize your directory with **GIT** using command:

Git init



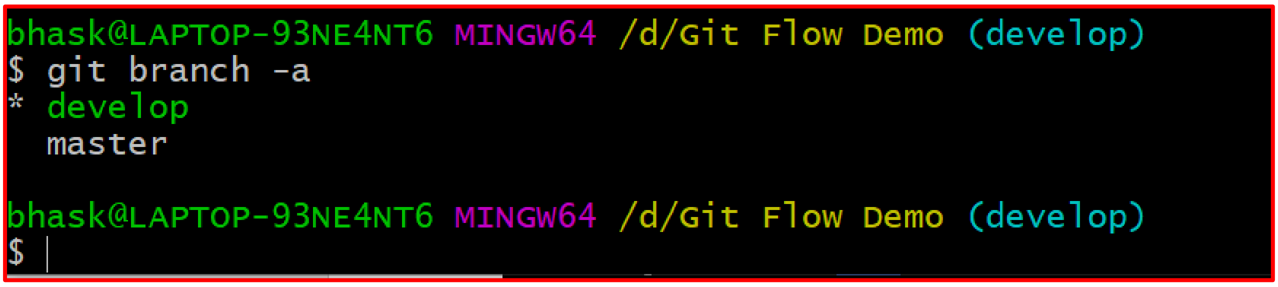
```
bhask@LAPTOP-93NE4NT6 MINGW64 /d/Git Flow Demo
$ git init
Initialized empty Git repository in D:/Git Flow Demo/.git/

bhask@LAPTOP-93NE4NT6 MINGW64 /d/Git Flow Demo (master)
$ |
```

4. Now initialise **'GIT FLOW'** using command:
   git init flow
   You will get option to provide name to different type of branches. Keep everything as **DEFAULT** and click on ENTER.



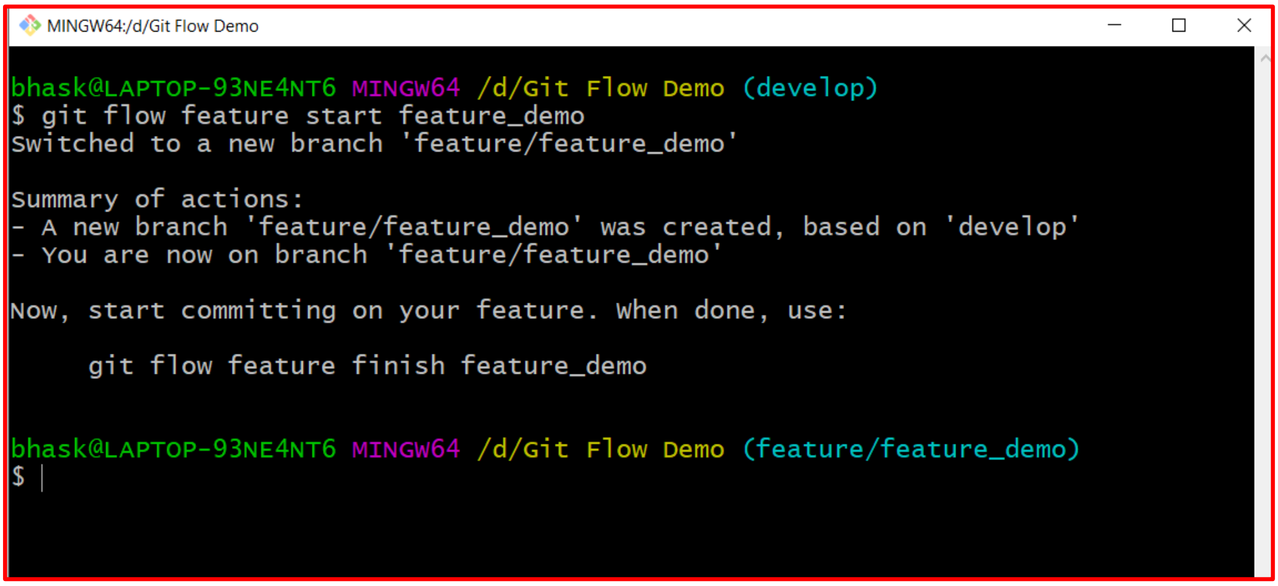
```
bhask@LAPTOP-93NE4NT6 MINGW64 /d/Git Flow Demo (master)
$ git flow init
No branches exist yet. Base branches must be created now.
Branch name for production releases: [master]
Branch name for "next release" development: [develop]

How to name your supporting branch prefixes?
Feature branches? [feature/]
Bugfix branches? [bugfix/]
Release branches? [release/]
Hotfix branches? [hotfix/]
Support branches? [support/]
Version tag prefix? []
Hooks and filters directory? [D:/Git Flow Demo/.git/hooks]

bhask@LAPTOP-93NE4NT6 MINGW64 /d/Git Flow Demo (develop)
$ |
```

5. Now you will be switched from **'Master'** branch to **'Develop'** branch. Now confirm the current branch using command:
   git branch -a



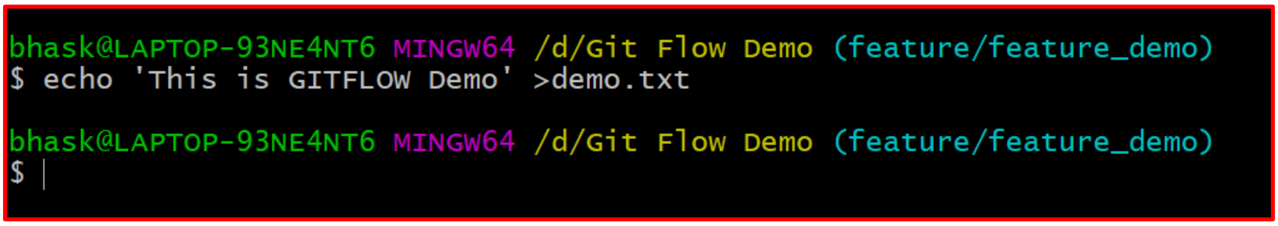
```
bhask@LAPTOP-93NE4NT6 MINGW64 /d/Git Flow Demo (develop)
$ git branch -a
* develop
  master

bhask@LAPTOP-93NE4NT6 MINGW64 /d/Git Flow Demo (develop)
$ |
```

6. Now create **‘Feature Branch’** using command:
   git flow feature start feature_demo



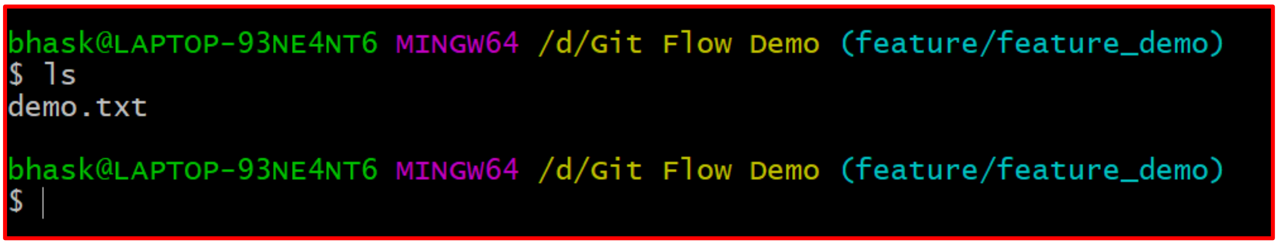
```
bhask@LAPTOP-93NE4NT6 MINGW64 /d/Git Flow Demo (develop)
$ git flow feature start feature_demo
Switched to a new branch 'feature/feature_demo'

Summary of actions:
- A new branch 'feature/feature_demo' was created, based on 'develop'
- You are now on branch 'feature/feature_demo'

Now, start committing on your feature. When done, use:

     git flow feature finish feature_demo


bhask@LAPTOP-93NE4NT6 MINGW64 /d/Git Flow Demo (feature/feature_demo)
$ |
```

7. Now create one new **text file** inside **Feature Branch** using command:
   echo ‘This is GITFLOW Demo’ >demo.txt



```
bhask@LAPTOP-93NE4NT6 MINGW64 /d/Git Flow Demo (feature/feature_demo)
$ echo 'This is GITFLOW Demo' >demo.txt

bhask@LAPTOP-93NE4NT6 MINGW64 /d/Git Flow Demo (feature/feature_demo)
$ |
```

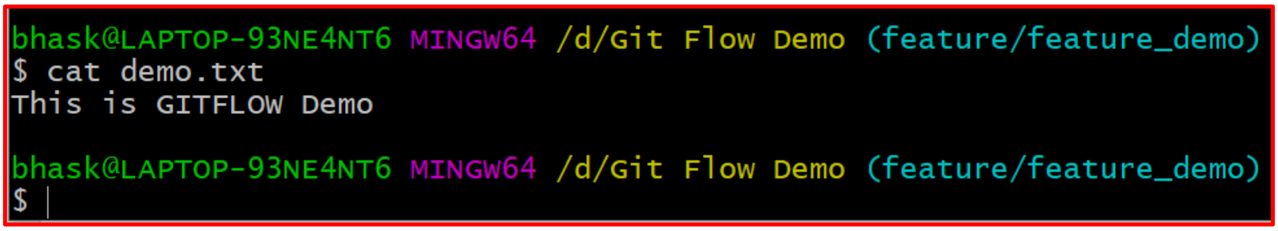
8. Now list your current directory to confirm the creation of **demo.txt file** using command:
   ls



```
bhask@LAPTOP-93NE4NT6 MINGW64 /d/Git Flow Demo (feature/feature_demo)
$ ls
demo.txt

bhask@LAPTOP-93NE4NT6 MINGW64 /d/Git Flow Demo (feature/feature_demo)
$ |
```

9. Use below command to validate the **demo.txt file** content.

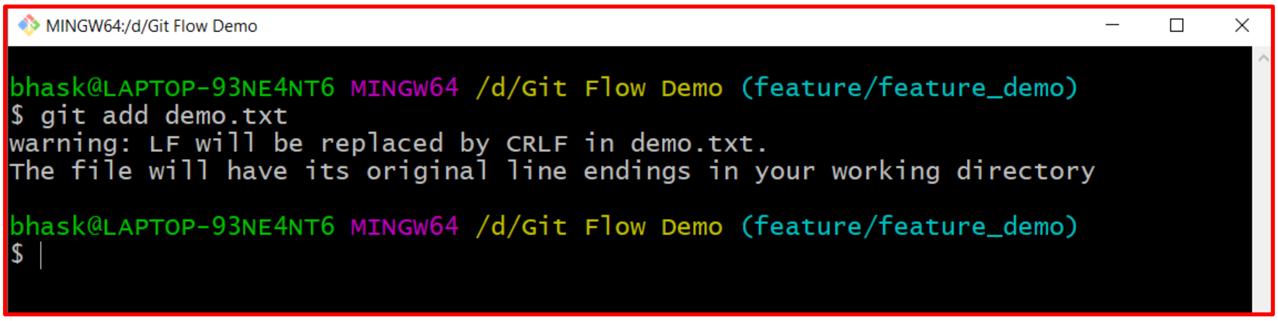
cat demo.txt



```
bhask@LAPTOP-93NE4NT6 MINGW64 /d/Git Flow Demo (feature/feature_demo)
$ cat demo.txt
This is GITFLOW Demo

bhask@LAPTOP-93NE4NT6 MINGW64 /d/Git Flow Demo (feature/feature_demo)
$ |
```

10. Now add your **demo.txt** file to **stage** area using command.

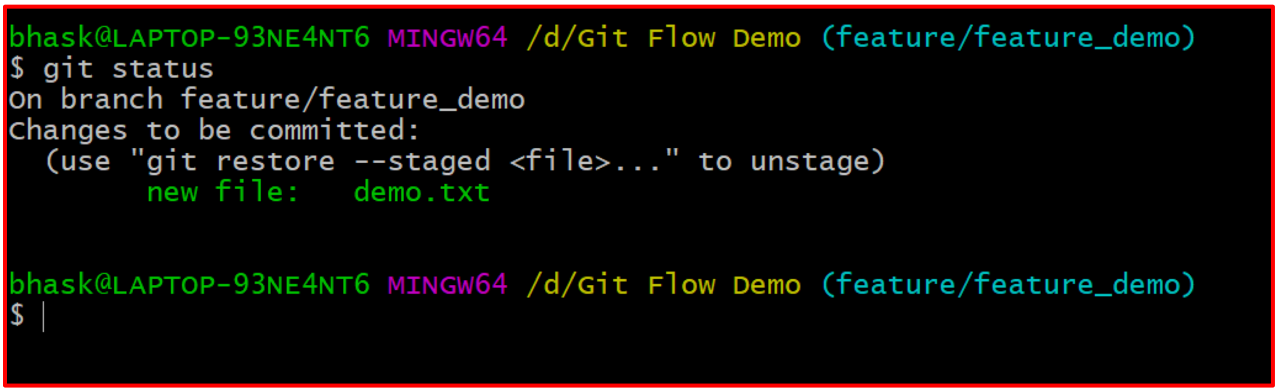
git add demo.txt



```
bhask@LAPTOP-93NE4NT6 MINGW64 /d/Git Flow Demo (feature/feature_demo)
$ git add demo.txt
warning: LF will be replaced by CRLF in demo.txt.
The file will have its original line endings in your working directory

bhask@LAPTOP-93NE4NT6 MINGW64 /d/Git Flow Demo (feature/feature_demo)
$ |
```

11. To check the current status of your working directory use below command:

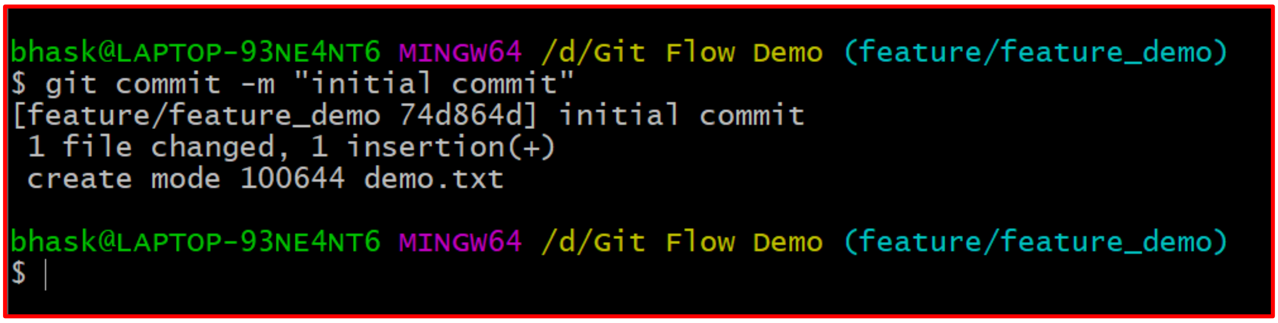
git status



```
bhask@LAPTOP-93NE4NT6 MINGW64 /d/Git Flow Demo (feature/feature_demo)
$ git status
On branch feature/feature_demo
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   demo.txt


bhask@LAPTOP-93NE4NT6 MINGW64 /d/Git Flow Demo (feature/feature_demo)
$ |
```

12. Now **commit** your changes using command:

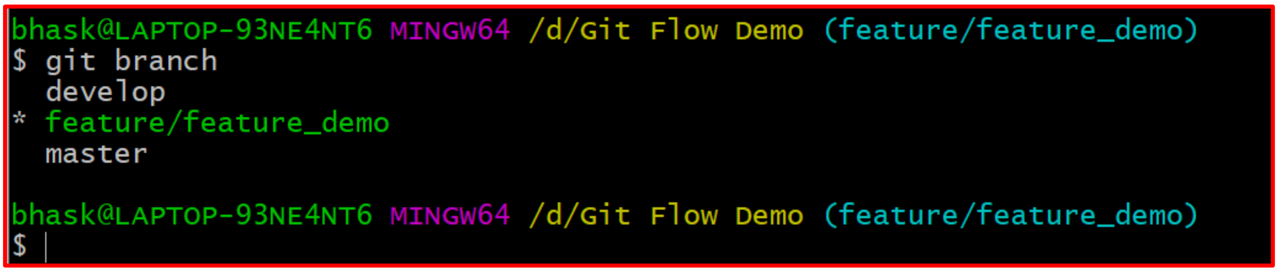
git command -m “initial command”



```
bhask@LAPTOP-93NE4NT6 MINGW64 /d/Git Flow Demo (feature/feature_demo)
$ git commit -m "initial commit"
[feature/feature_demo 74d864d] initial commit
 1 file changed, 1 insertion(+)
 create mode 100644 demo.txt

bhask@LAPTOP-93NE4NT6 MINGW64 /d/Git Flow Demo (feature/feature_demo)
$ |
```

13. Now to confirm the current branch use command:

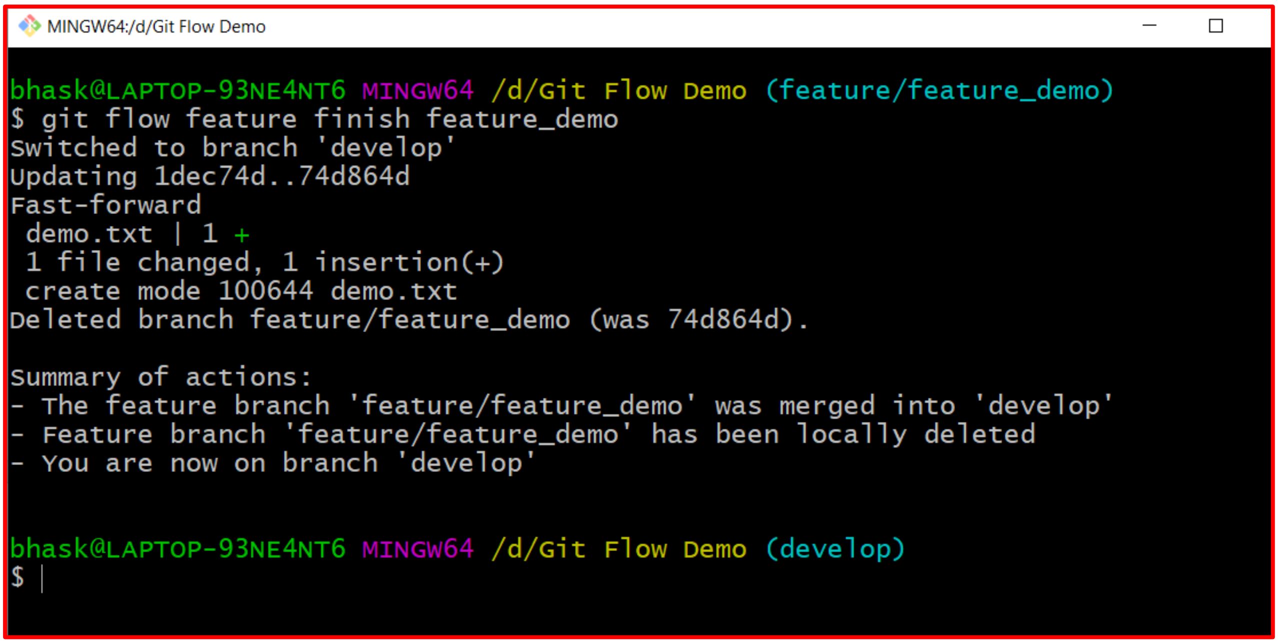
git branch



```
bhask@LAPTOP-93NE4NT6 MINGW64 /d/Git Flow Demo (feature/feature_demo)
$ git branch
  develop
* feature/feature_demo
  master

bhask@LAPTOP-93NE4NT6 MINGW64 /d/Git Flow Demo (feature/feature_demo)
$ |
```

14. Now **finish** the **FEATURE** branch to move our changes to **'DEVELOP'** branch using command:

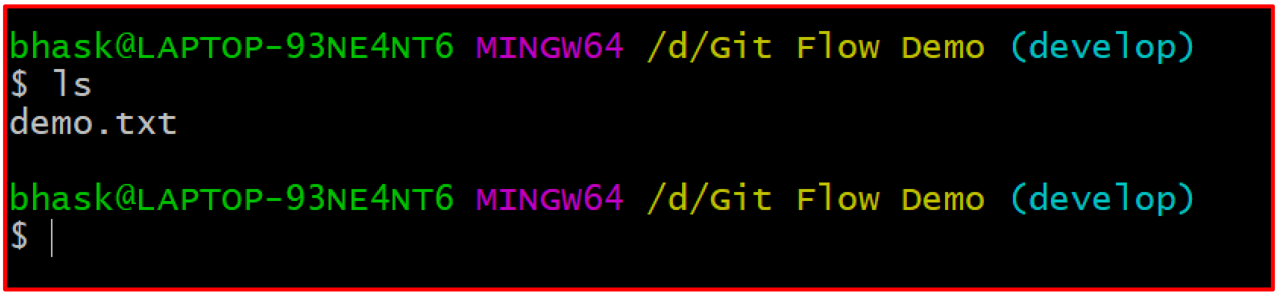
git flow feature finish feature_demo



```
bhask@LAPTOP-93NE4NT6 MINGW64 /d/Git Flow Demo (feature/feature_demo)
$ git flow feature finish feature_demo
Switched to branch 'develop'
Updating 1dec74d..74d864d
Fast-forward
 demo.txt | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 demo.txt
Deleted branch feature/feature_demo (was 74d864d).

Summary of actions:
- The feature branch 'feature/feature_demo' was merged into 'develop'
- Feature branch 'feature/feature_demo' has been locally deleted
- You are now on branch 'develop'


bhask@LAPTOP-93NE4NT6 MINGW64 /d/Git Flow Demo (develop)
$ |
```

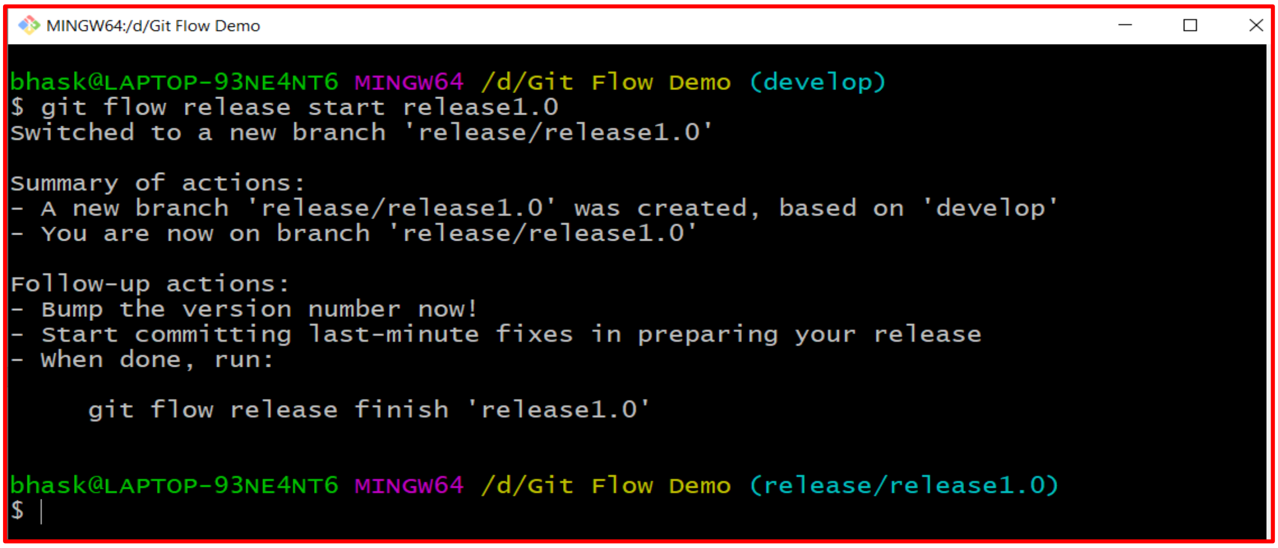
15. You will be navigated back to **'DEVELOP'** branch from **'Feature'** branch. Now list the directory to confirm the push of **'demo.txt'** file from **'Feature'** branch to **'DEVELOP'** branch using command:

ls



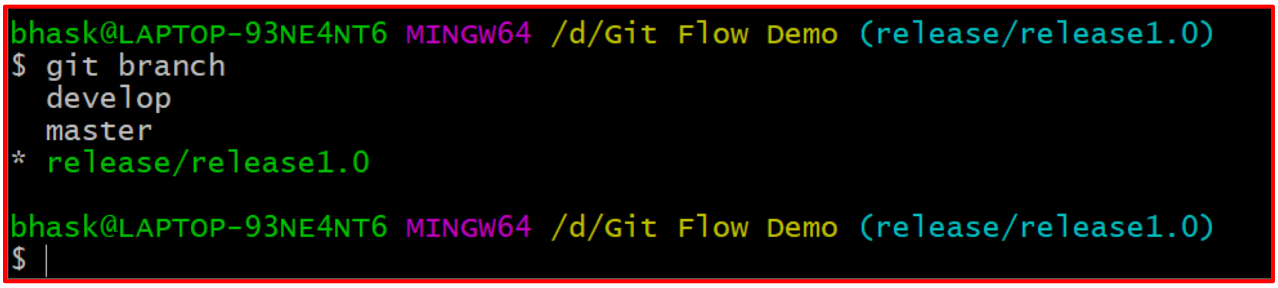
```
bhask@LAPTOP-93NE4NT6 MINGW64 /d/Git Flow Demo (develop)
$ ls
demo.txt

bhask@LAPTOP-93NE4NT6 MINGW64 /d/Git Flow Demo (develop)
$ |
```

16. Now start the **release process** based on your **'DEVELOP'** branch using command:
git flow release start release1.0



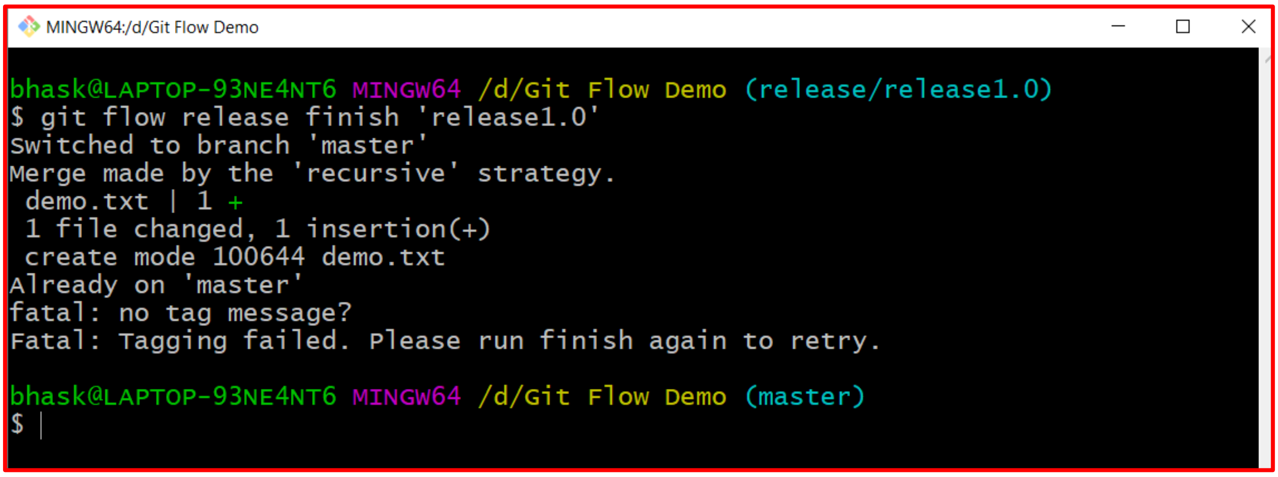
```
bhask@LAPTOP-93NE4NT6 MINGW64 /d/Git Flow Demo (develop)
$ git flow release start release1.0
Switched to a new branch 'release/release1.0'

Summary of actions:
- A new branch 'release/release1.0' was created, based on 'develop'
- You are now on branch 'release/release1.0'

Follow-up actions:
- Bump the version number now!
- Start committing last-minute fixes in preparing your release
- When done, run:

     git flow release finish 'release1.0'


bhask@LAPTOP-93NE4NT6 MINGW64 /d/Git Flow Demo (release/release1.0)
$ |
```

17. You will see a new **'RELEASE1.0'** branch is created successfully. To confirm the current branch use command:
git branch



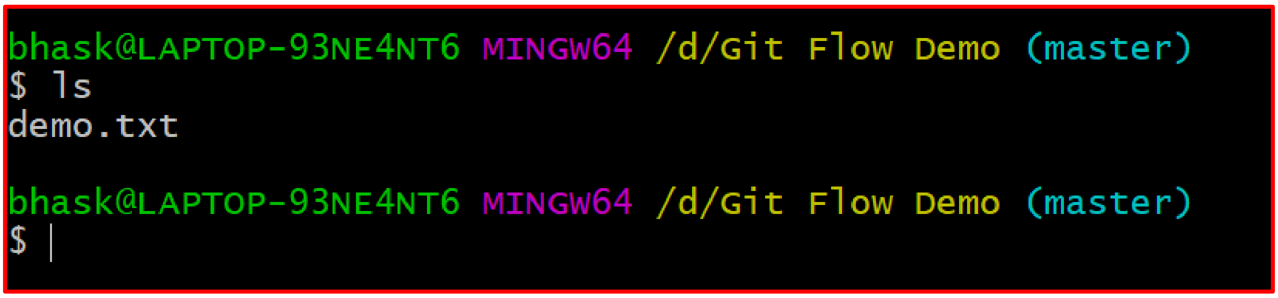
```
bhask@LAPTOP-93NE4NT6 MINGW64 /d/Git Flow Demo (release/release1.0)
$ git branch
  develop
  master
* release/release1.0

bhask@LAPTOP-93NE4NT6 MINGW64 /d/Git Flow Demo (release/release1.0)
$ |
```

18. Now finish the **'RELEASE1.0'** branch to merge your files to **'MASTER'** branch using command:
git flow release finish 'release1.0'



```
bhask@LAPTOP-93NE4NT6 MINGW64 /d/Git Flow Demo (release/release1.0)
$ git flow release finish 'release1.0'
Switched to branch 'master'
Merge made by the 'recursive' strategy.
 demo.txt | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 demo.txt
Already on 'master'
fatal: no tag message?
Fatal: Tagging failed. Please run finish again to retry.

bhask@LAPTOP-93NE4NT6 MINGW64 /d/Git Flow Demo (master)
$ |
```

19. You will be navigated back to **'Master'** branch from **'Develop'** branch. Now list the directory to confirm the **push** of **'demo.txt'** file from **'Develop'** branch to **'Master'** branch using command: **ls**



```
bhask@LAPTOP-93NE4NT6 MINGW64 /d/Git Flow Demo (master)
$ ls
demo.txt

bhask@LAPTOP-93NE4NT6 MINGW64 /d/Git Flow Demo (master)
$ |
```