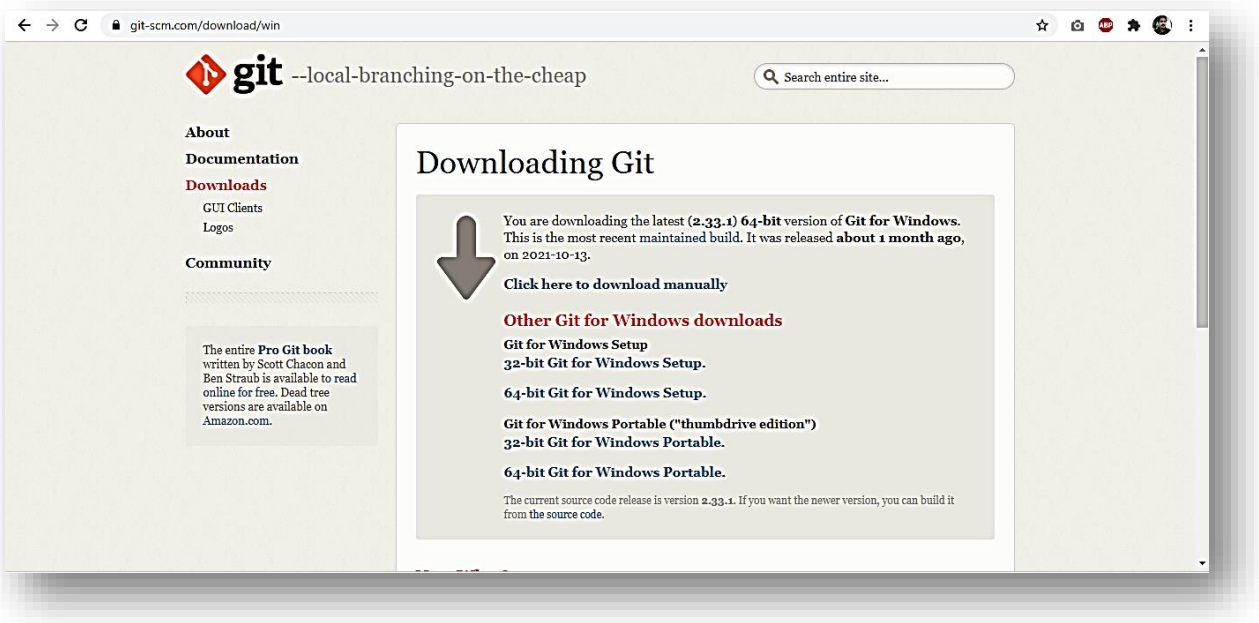
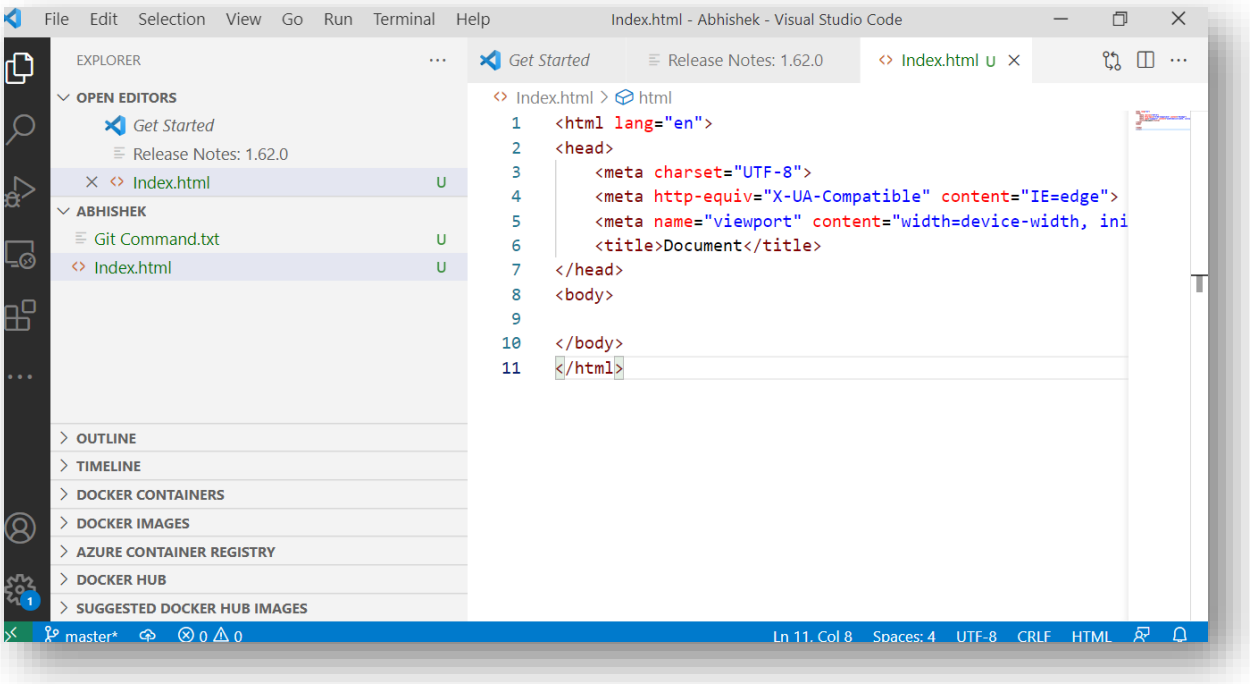


Git Command

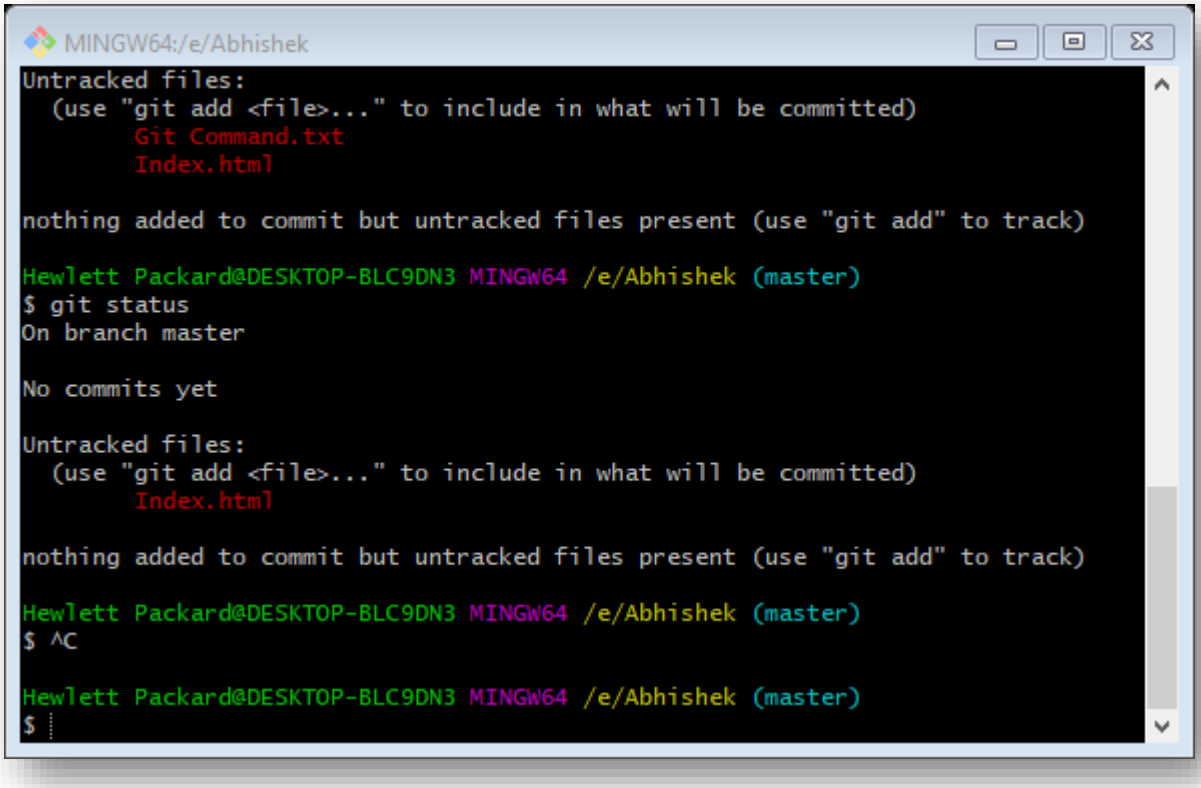
```
Hewlett Packard@DESKTOP-BLC9DN3 MINGW64 /e/Abhishek
$ git init

Hewlett Packard@DESKTOP-BLC9DN3 MINGW64 /e/Abhishek
$ git config --global user.name Abhishek

Hewlett Packard@DESKTOP-BLC9DN3 MINGW64 /e/Abhishek
$ git config --global user. email abhisharma1108@yahoo.co.in

Hewlett Packard@DESKTOP-BLC9DN3 MINGW64 /e/Abhishek

$ git config --global user.email
abhisharma1108@yahoo.co.in

Hewlett Packard@DESKTOP-BLC9DN3 MINGW64 /e/Abhishek
$ git config --global user.name
Abhishek

Hewlett Packard@DESKTOP-BLC9DN3 MINGW64 /e/Abhishek
$ code.

Hewlett Packard@DESKTOP-BLC9DN3 MINGW64 /e/Abhishek
$ git init
Initialized empty Git repository in E:/Abhishek/.git/

Hewlett Packard@DESKTOP-BLC9DN3 MINGW64 /e/Abhishek (master)
$ ls -lart
total 13
drwxr-xr-x 1 Hewlett Packard 197121   0 Nov 15 18:53  ../
-rw-r--r-- 1 Hewlett Packard 197121 498 Nov 15 19:05 'Git Command.txt'
drwxr-xr-x 1 Hewlett Packard 197121   0 Nov 15 19:07  ./
```

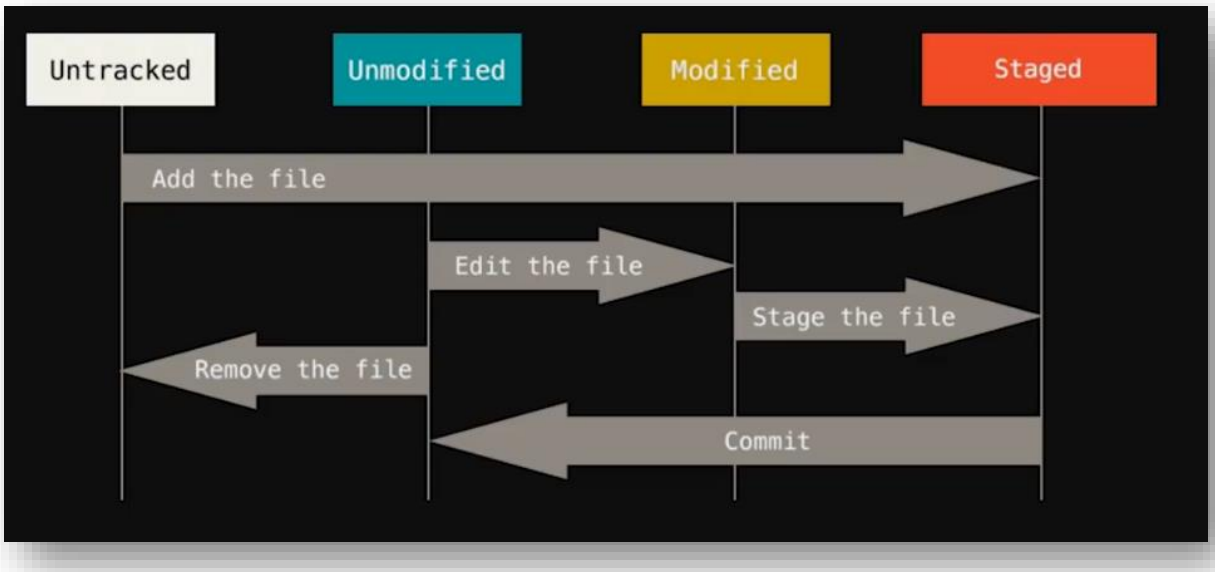
drwxr-xr-x 1 Hewlett Packard 197121 0 Nov 15 19:07. git/

```
Hewlett Packard@DESKTOP-BLC9DN3 MINGW64 /e/Abhishek (master)
```




```
$ git status
```





Untracked – files are **files that have been created within your repo's working directory but have not yet been added to the repository's tracking index** using the git add command.

Unmodified – These are **any files that haven't been modified since the last commit**. They will still be included in the next commit, but remain as is. modified – These are files that have been modified since the last commit (we probably modified these as part of bug fixes).

Modified – **As** you edit files, Git sees them as modified, because you've **changed** them since your last commit. As you work, you selectively stage these modified files and then commit all those staged changes, and the cycle repeats.

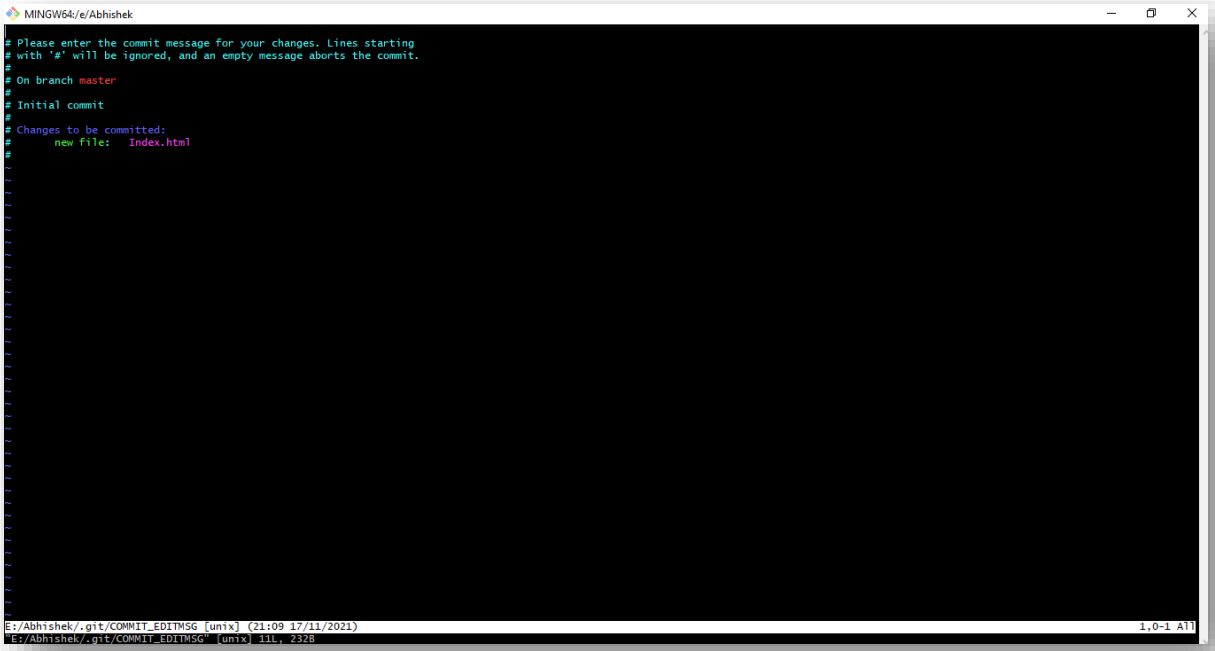
Stage – To stage a file is **simply to prepare it finely for a commit.** Git, with its index allows you to commit only certain parts of the changes you've done since the last commit. Say you're working on two features – one is finished, and one still needs some work done

```
No commits yet

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

NOW WE HAVE TO MOVE IN STAGING AREA

```
Hewlett Packard@DESKTOP-BLC9DN3 MINGW64 /e/Abhishek (master)
$ git add Index.html

Hewlett Packard@DESKTOP-BLC9DN3 MINGW64 /e/Abhishek (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   Index.html
```

NOW FILE MOVE IN STAGING AREA & WE HAVE TO COMMIT THE CHANGES

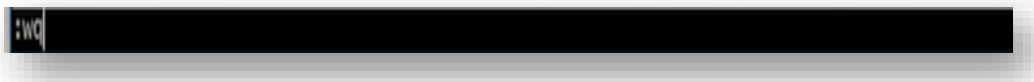


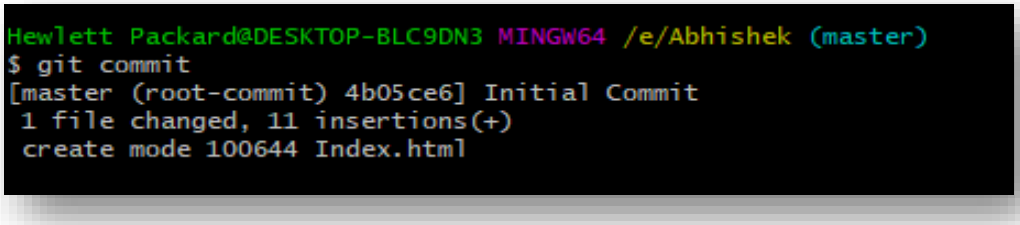
Press I and start typing

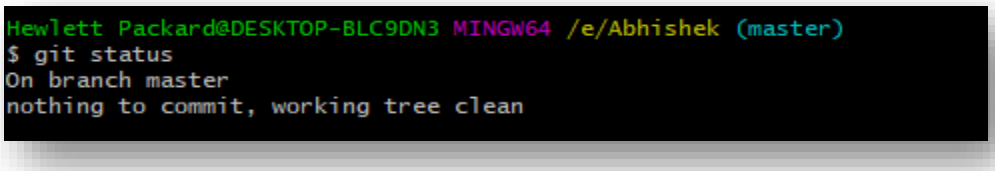
Initial Commit

Now for saving press ESC KEY -> W -> Q

```
Hewlett Packard@DESKTOP-BLC9DN3 MINGW64 /e/Abhishek (master)
$ git commit
[master (root-commit) 4b05ce6] Initial Commit
 1 file changed, 11 insertions (+)
 create mode 100644 Index.html
```







```
Hewlett Packard@DESKTOP-BLC9DN3 MINGW64 /e/Abhishek (master)
$ git status
On branch master
nothing to commit, working tree clean

Hewlett Packard@DESKTOP-BLC9DN3 MINGW64 /e/Abhishek (master)
$ touch about.html

Hewlett Packard@DESKTOP-BLC9DN3 MINGW64 /e/Abhishek (master)
$ touch contact.html

Hewlett Packard@DESKTOP-BLC9DN3 MINGW64 /e/Abhishek (master)
$ touch Home.html
```

```
Hewlett Packard@DESKTOP-BLC9DN3 MINGW64 /e/Abhishek (master)
$ touch about.html

Hewlett Packard@DESKTOP-BLC9DN3 MINGW64 /e/Abhishek (master)
$ touch contact.html

Hewlett Packard@DESKTOP-BLC9DN3 MINGW64 /e/Abhishek (master)
$ touch Home.html
```

```
Hewlett Packard@DESKTOP-BLC9DN3 MINGW64 /e/Abhishek (master)
$ git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        Home.html
        about.html
        contact.html

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

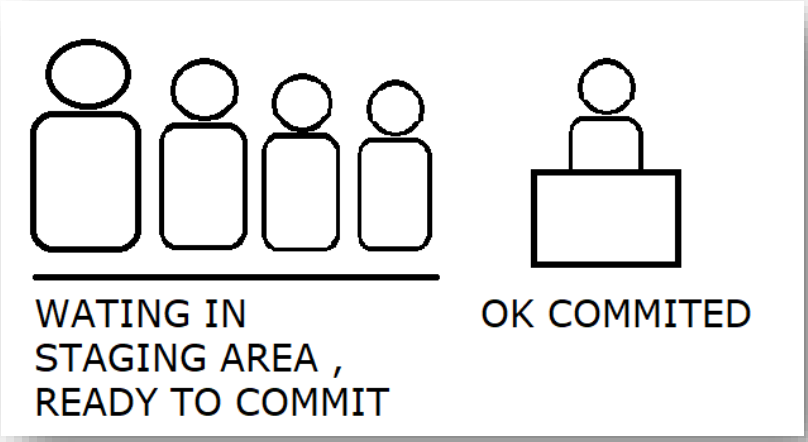
NOW MOVE ALL FILE IN STAGING AREA

```
Hewlett Packard@DESKTOP-BLC9DN3 MINGW64 /e/Abhishek (master)
$ git add -A
```

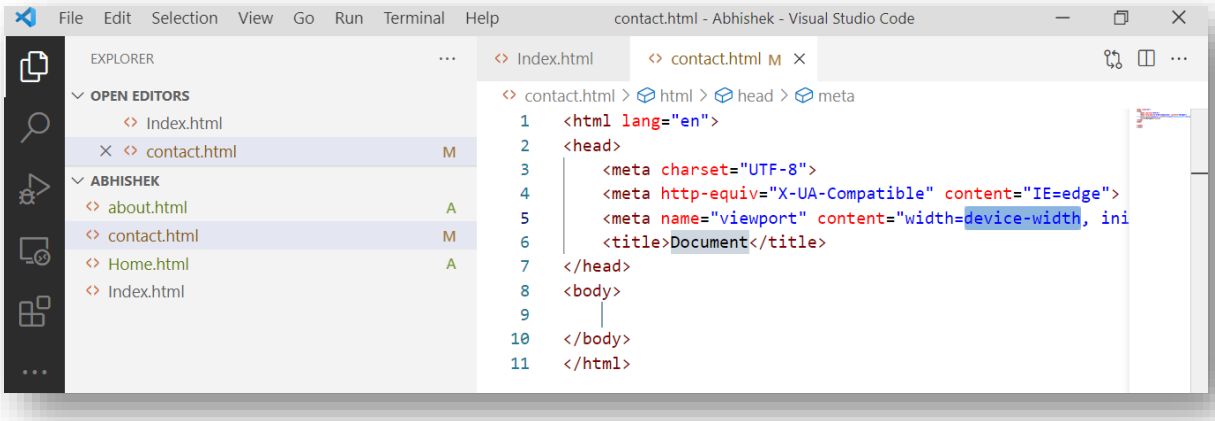
```
Hewlett Packard@DESKTOP-BLC9DN3 MINGW64 /e/Abhishek (master)
$ git add -A
```

NOW RUN GIT STATUS TO SEE ALL FILE IN STAGING AREA

```
Hewlett Packard@DESKTOP-BLC9DN3 MINGW64 /e/Abhishek (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   Home.html
        new file:   about.html
        new file:   contact.html
```



NOW I WILL MAKE CHANGES IN CONTACT .HTML AND SEE



```
Hewlett Packard@DESKTOP-BLC9DN3 MINGW64 /e/Abhishek (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   Home.html
        new file:   about.html
        new file:   contact.html

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:   contact.html


Hewlett Packard@DESKTOP-BLC9DN3 MINGW64 /e/Abhishek (master)
$
```

NOW IT SHOWING ITS IN MODIFIED CHANGES, NOW I HAVE TO COMMIT CHAGES ALL OR INDIVIDUAL FILE.

```
Hewlett Packard@DESKTOP-BLC9DN3 MINGW64 /e/Abhishek (master)
$ git add -A
```

```
Hewlett Packard@DESKTOP-BLC9DN3 MINGW64 /e/Abhishek (master)
$ git add -A

Hewlett Packard@DESKTOP-BLC9DN3 MINGW64 /e/Abhishek (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   Home.html
        new file:   about.html
        new file:   contact.html
```

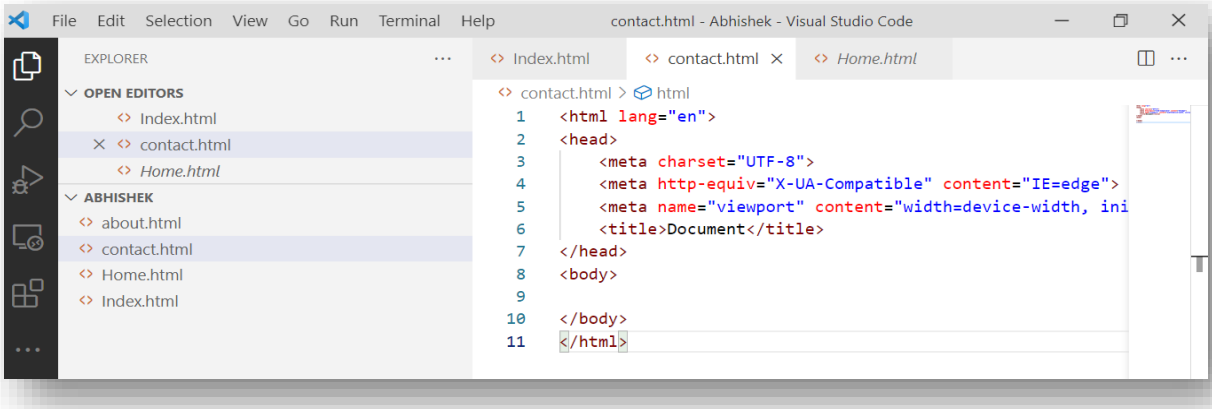
```
$ git commit -m "Added more htmls"
```

```
Hewlett Packard@DESKTOP-BLC9DN3 MINGW64 /e/Abhishek (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   Home.html
        new file:   about.html
        new file:   contact.html


Hewlett Packard@DESKTOP-BLC9DN3 MINGW64 /e/Abhishek (master)
$ git commit -m "Added more htmls"
[master 679edbe] Added more htmls
 3 files changed, 11 insertions(+)
 create mode 100644 Home.html
 create mode 100644 about.html
 create mode 100644 contact.html

Hewlett Packard@DESKTOP-BLC9DN3 MINGW64 /e/Abhishek (master)
$
```

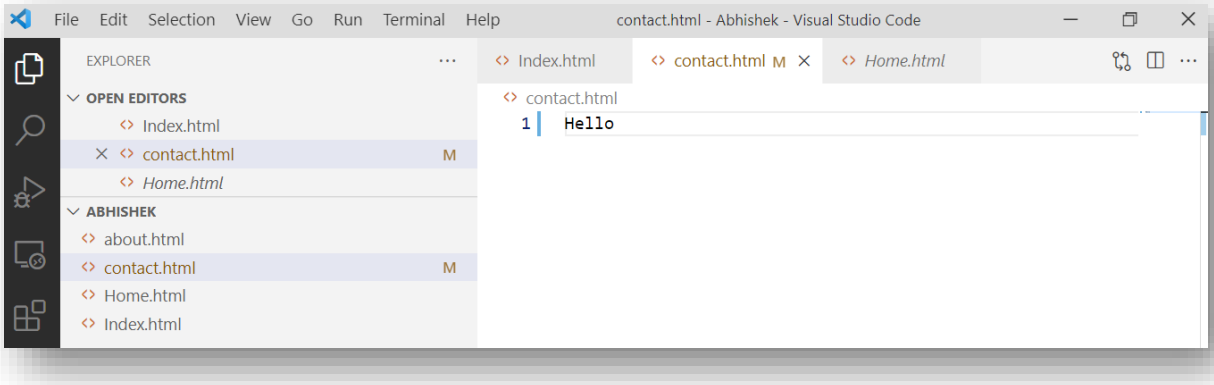
Hewlett Packard@DESKTOP-BLC9DN3 MINGW64 /e/Abhishek (master)
$ git status

```
Hewlett Packard@DESKTOP-BLC9DN3 MINGW64 /e/Abhishek (master)
$ git status
On branch master
nothing to commit, working tree clean
```

Hewlett Packard@DESKTOP-BLC9DN3 MINGW64 /e/Abhishek (master)
$ clear



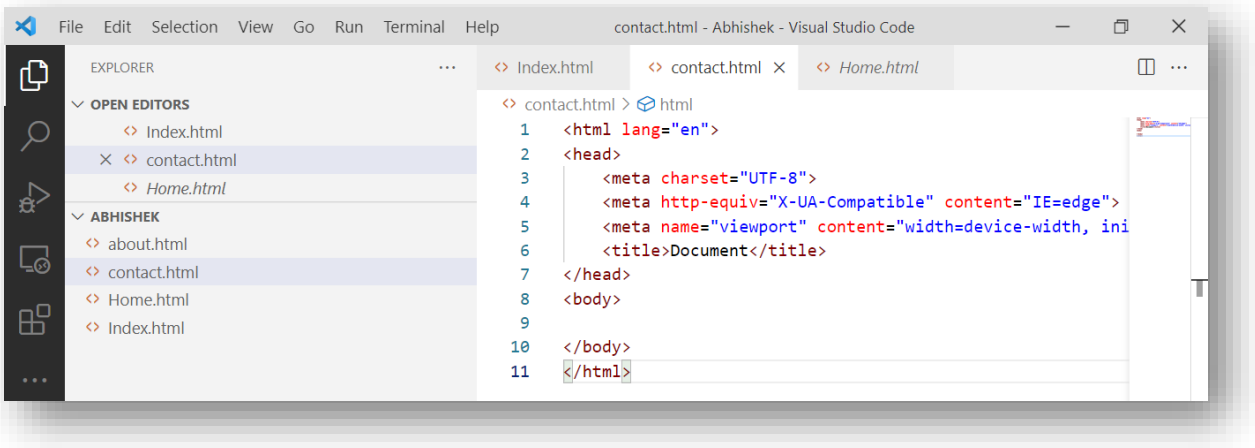
Now some one deleted your code



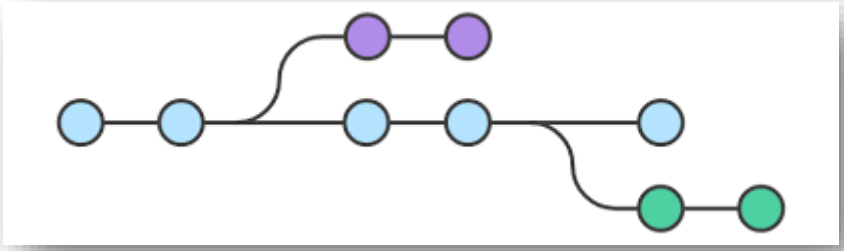

```
Hewlett Packard@DESKTOP-BLC9DN3 MINGW64 /e/Abhishek (master)
$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   contact.html

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

Hewlett Packard@DESKTOP-BLC9DN3 MINGW64 /e/Abhishek (master)
$
```

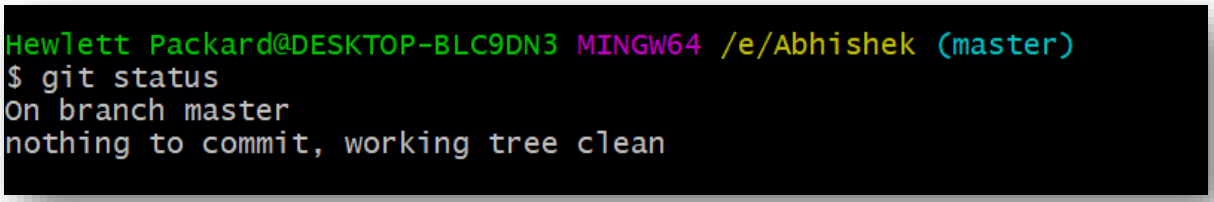
```
Hewlett Packard@DESKTOP-BLC9DN3 MINGW64
/e/Abhishek (master)
$ git checkout contact.html
```

```
Hewlett Packard@DESKTOP-BLC9DN3 MINGW64 /e/Abhishek (master)
$ git checkout contact.html
Updated 1 path from the index
```



The git checkout command lets **you navigate between the branches created by git branch**. Checking out a branch updates the files in the working directory to match the version stored in that branch, and it tells Git to record all new commits on that branch.






```
Hewlett Packard@DESKTOP-BLC9DN3 MINGW64
/e/Abhishek (master)
$ git checkout -f
```

Checking out a branch updates the all files in the working directory to match the version stored in that branch

Note: for all files.

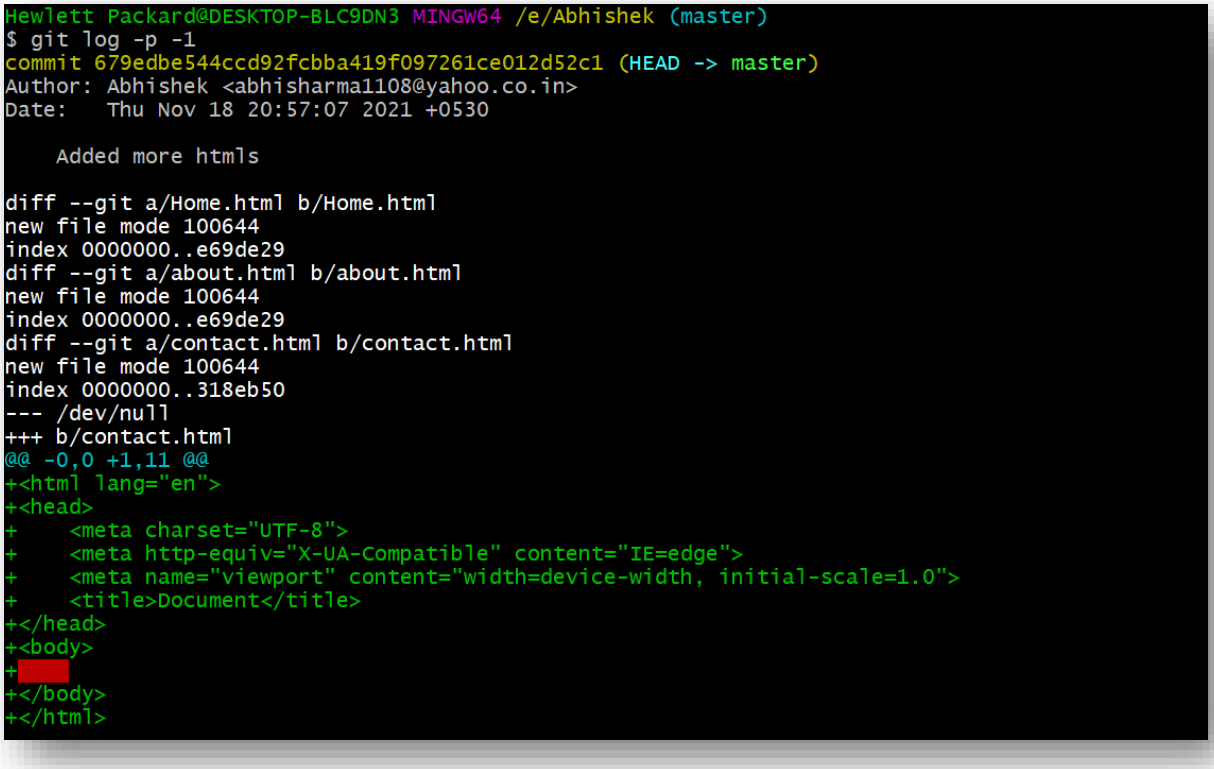
```
Hewlett Packard@DESKTOP-BLC9DN3 MINGW64 /e/Abhishek (master)
$ git status
On branch master
nothing to commit, working tree clean
```

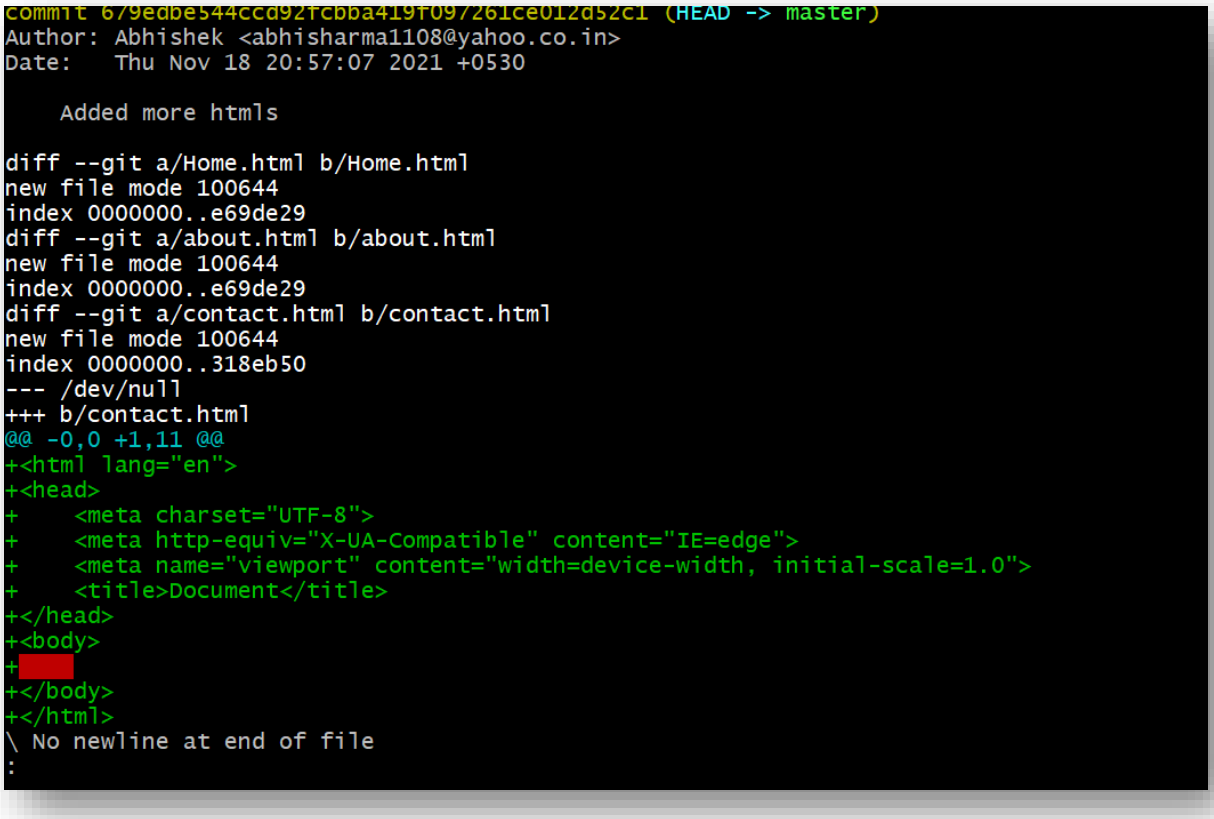
The git log command **shows a list of all the commits made to a repository**. You can see the hash of each Git commit, the message associated with each commit, and more metadata. This command is useful for displaying the history of a repository

```
Hewlett Packard@DESKTOP-BLC9DN3 MINGW64 /e/Abhishek (master)
$ git log
commit 679edbe544ccd92fcbba419f097261ce012d52c1 (HEAD -> master)
Author: Abhishek <abhisharma1108@yahoo.co.in>
Date:   Thu Nov 18 20:57:07 2021 +0530

    Added more htmls

commit 4b05ce6a5518be2887e6590546d11dc0ecc13691
Author: Abhishek <abhisharma1108@yahoo.co.in>
Date:   Wed Nov 17 21:09:08 2021 +0530

    Initial Commit

Hewlett Packard@DESKTOP-BLC9DN3 MINGW64 /e/Abhishek (master)
$
```

```
Hewlett Packard@DESKTOP-BLC9DN3 MINGW64
/e/Abhishek (master)
$ git log -p -1
```

Note: you can also specify the number of changes e.g. `git log -p -5.`



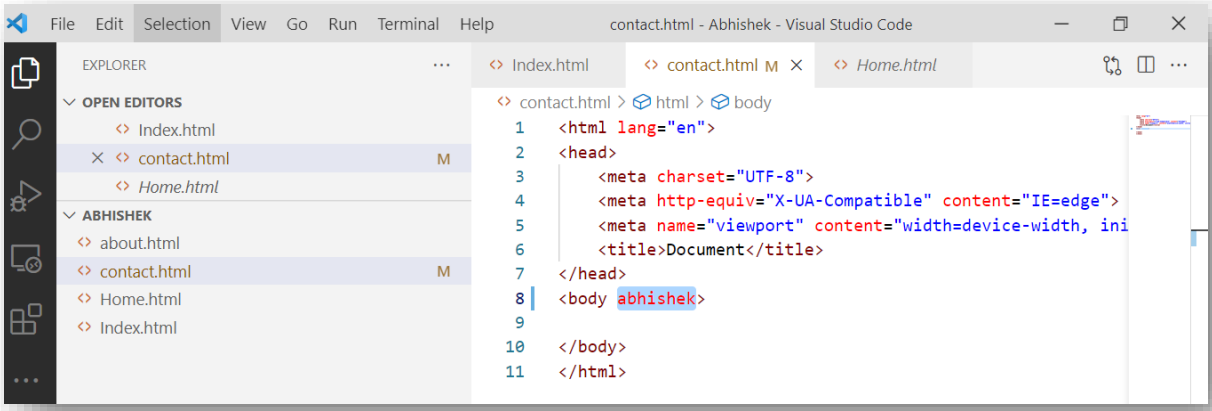


Note: Press q to quit

```
+    <meta name="viewport" content="width=device-width, initial-s
+    <title>Document</title>
+</head>
+<body>
+
+</body>
+</html>
\ No newline at end of file

Hewlett Packard@DESKTOP-BLC9DN3 MINGW64 /e/Abhishek (master)
$
```

What is git diff command?

The git diff command **displays the differences between files in two commits or between a commit and your current repository**. You can see what text has been added to, removed from, and changed in a file. ... Often, Git diff is used for comparing branches in a Git repository



Now suppose I have made changes in the contact.html and I have to see the changes

```
Hewlett Packard@DESKTOP-BLC9DN3 MINGW64 /e/Abhishek
(master)
$ git status
```

```
Hewlett Packard@DESKTOP-BLC9DN3 MINGW64 /e/Abhishek (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:   contact.html

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

Hewlett Packard@DESKTOP-BLC9DN3 MINGW64 /e/Abhishek (master)
$
```

```
Hewlett Packard@DESKTOP-BLC9DN3 MINGW64 /e/Abhishek (master)
$ git diff
```

Note: It will be comparing the file between working tree with staging area.

```
Hewlett Packard@DESKTOP-BLC9DN3 MINGW64 /e/Abhishek (master)
$ git diff
diff --git a/contact.html b/contact.html
index 318eb50..57df5c2 100644
--- a/contact.html
+++ b/contact.html
@@ -5,7 +5,7 @@
     <meta name="viewport" content="width=device-width, initial-scale=1.0">
     <title>Document</title>
 </head>
-<body>
+<body abhishek>

 </body>
 </html>
\ No newline at end of file

Hewlett Packard@DESKTOP-BLC9DN3 MINGW64 /e/Abhishek (master)
$
```

Note: Now move working tree in the staging area.

```
Hewlett Packard@DESKTOP-BLC9DN3 MINGW64 /e/Abhishek (master)
$ git add -A

Hewlett Packard@DESKTOP-BLC9DN3 MINGW64 /e/Abhishek (master)
$ git diff
```

Note: Now it won't show anything, because our file has been moved to staging area

```
Hewlett Packard@DESKTOP-BLC9DN3 MINGW64 /e/Abhishek (master)
$ git add -A

Hewlett Packard@DESKTOP-BLC9DN3 MINGW64 /e/Abhishek (master)
$ git diff
```

Note: Now let's check the status, because the file moved into staging area.

```
Hewlett Packard@DESKTOP-BLC9DN3 MINGW64 /e/Abhishek (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   contact.html
```

Note: We have to compare the last changes in staging area

```
Hewlett Packard@DESKTOP-BLC9DN3 MINGW64
/e/Abhishek (master)
$ git diff --staged
```

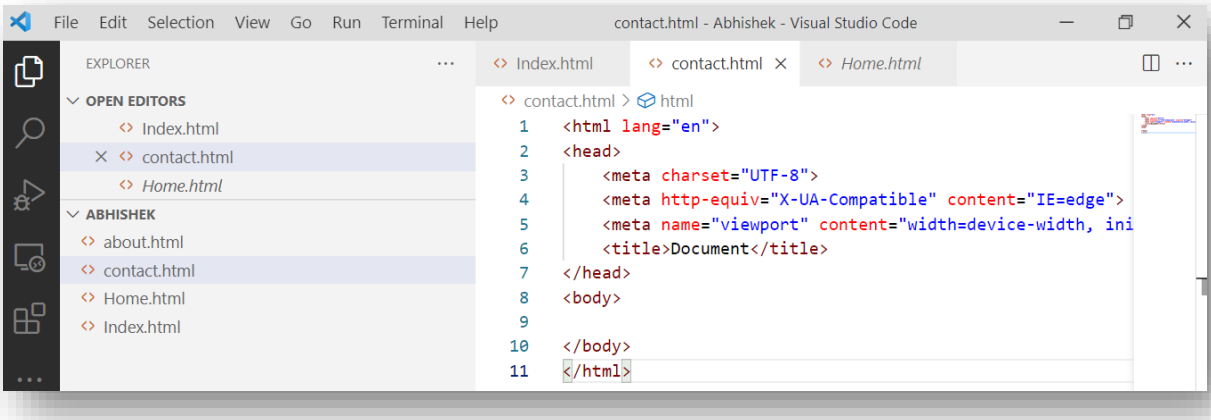
```
Hewlett Packard@DESKTOP-BLC9DN3 MINGW64 /e/Abhishek (master)
$ git diff --staged
diff --git a/contact.html b/contact.html
index 318eb50..57df5c2 100644
--- a/contact.html
+++ b/contact.html
@@ -5,7 +5,7 @@
     <meta name="viewport" content="width=devi
ce-width, initial-scale=1.0">
     <title>Document</title>
 </head>
-<body>
+<body abhishek>

 </body>
 </html>
:...skipping...
diff --git a/contact.html b/contact.html
index 318eb50..57df5c2 100644
--- a/contact.html
+++ b/contact.html
@@ -5,7 +5,7 @@
     <meta name="viewport" content="width=device-width, initial-scale=1.0">
     <title>Document</title>
 </head>
-<body>
+<body abhishek>
```

Note: Press q to come out of editor

```
Hewlett Packard@DESKTOP-BLC9DN3 MINGW64 /e/Abhishek (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   contact.html


Hewlett Packard@DESKTOP-BLC9DN3 MINGW64 /e/Abhishek (master)
$
```

```
Hewlett Packard@DESKTOP-BLC9DN3 MINGW64 /e/Abhishek (master)
$ git checkout -f
```

```
Hewlett Packard@DESKTOP-BLC9DN3 MINGW64 /e/Abhishek (master)
$ git status
On branch master
nothing to commit, working tree clean
```

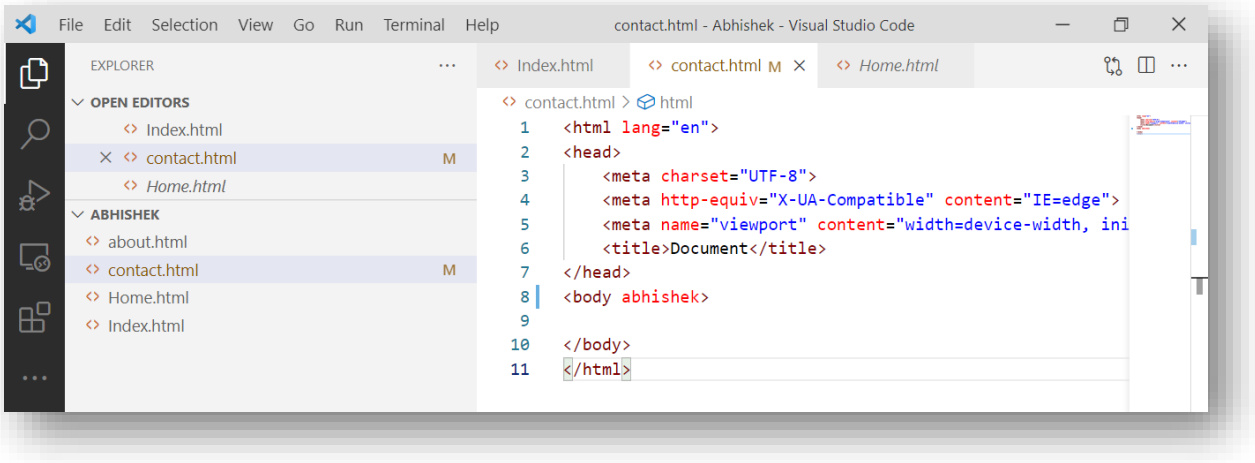


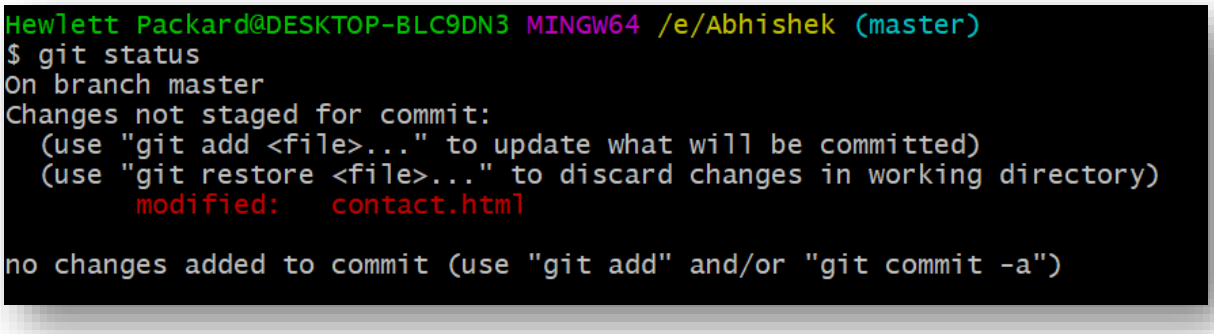
Note: Changes revert

Now if don't want every time first file to move from working tree to move staging area and then commit. Follow the following command.

Note: Staging area feature is good feature. Please be in practice.

Again, made the changes and do the following





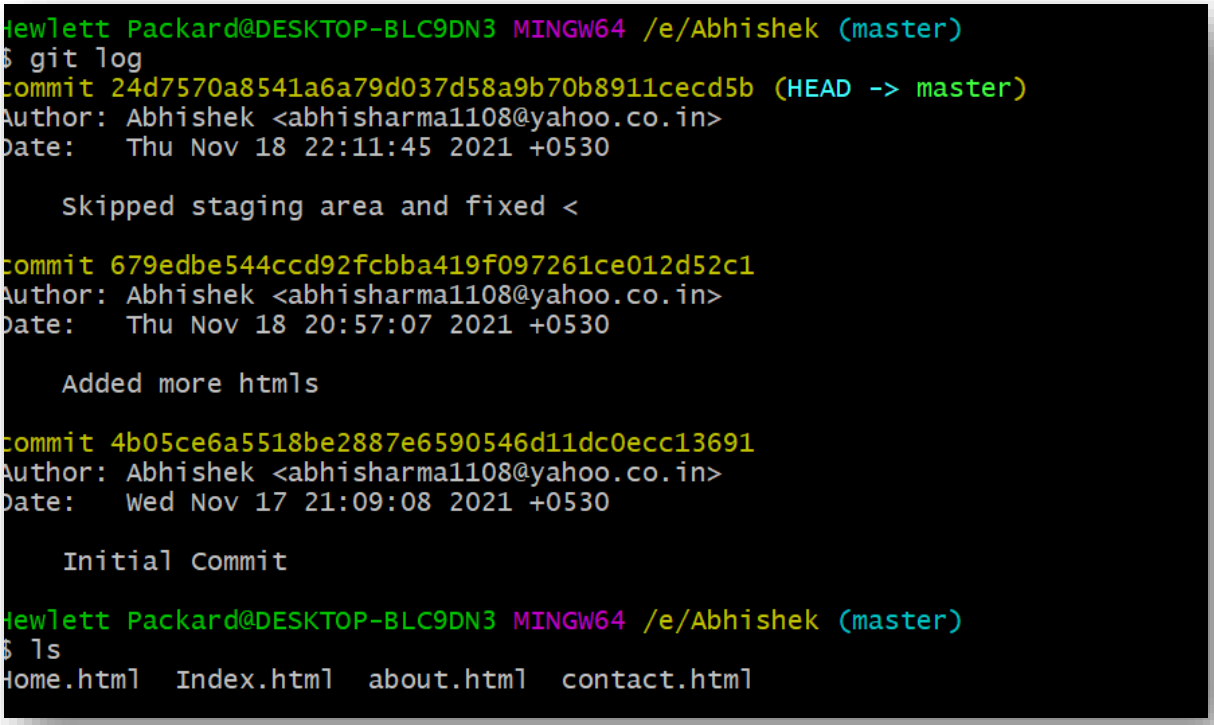

```
Hewlett Packard@DESKTOP-BLC9DN3 MINGW64 /e/Abhishek (master)
$ git commit -a -m "Skipped staging area and fixed <"
```

```
Hewlett Packard@DESKTOP-BLC9DN3 MINGW64 /e/Abhishek (master)
$ git commit -a -m "Skipped staging area and fixed <"
[master 24d7570] Skipped staging area and fixed <
 1 file changed, 1 insertion(+), 1 deletion(-)
```

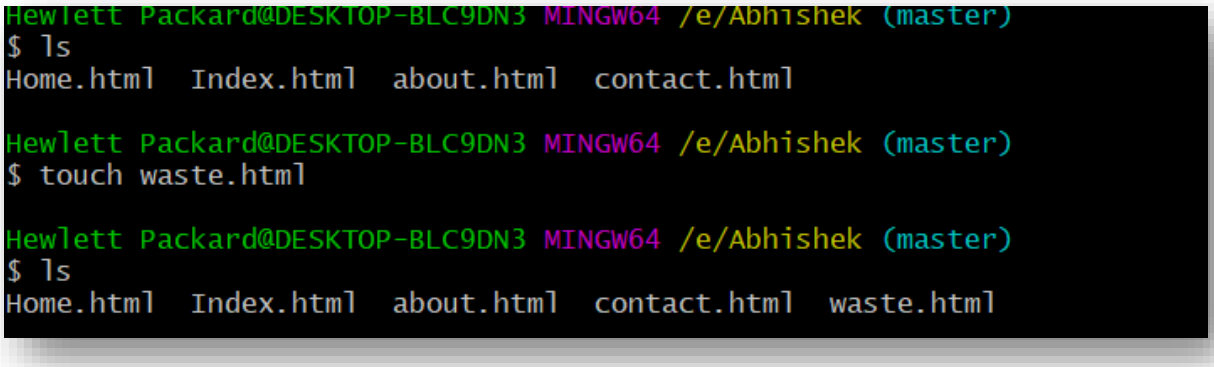
Now check git log

```
Hewlett Packard@DESKTOP-BLC9DN3 MINGW64 /e/Abhishek (master)
$ git log
```

```
Hewlett Packard@DESKTOP-BLC9DN3 MINGW64 /e/Abhishek (master)
$ git log
commit 24d7570a8541a6a79d037d58a9b70b8911cecd5b (HEAD -> master)
Author: Abhishek <abhisharma1108@yahoo.co.in>
Date:   Thu Nov 18 22:11:45 2021 +0530

    Skipped staging area and fixed <

commit 679edbe544ccd92fcbba419f097261ce012d52c1
Author: Abhishek <abhisharma1108@yahoo.co.in>
Date:   Thu Nov 18 20:57:07 2021 +0530

    Added more htmls

commit 4b05ce6a5518be2887e6590546d11dc0ecc13691
Author: Abhishek <abhisharma1108@yahoo.co.in>
Date:   Wed Nov 17 21:09:08 2021 +0530

    Initial Commit
```




```
Hewlett Packard@DESKTOP-BLC9DN3 MINGW64 /e/Abhishek (master)
$ ls
Home.html Index.html about.html contact.html

Hewlett Packard@DESKTOP-BLC9DN3 MINGW64 /e/Abhishek (master)
$ touch waste.html

Hewlett Packard@DESKTOP-BLC9DN3 MINGW64 /e/Abhishek (master)
$ ls
Home.html Index.html about.html contact.html waste.html
```



Now add waste.html to staging area.

```
Hewlett Packard@DESKTOP-BLC9DN3 MINGW64
/e/Abhishek (master)
$ git add -A
```

```
Hewlett Packard@DESKTOP-BLC9DN3 MINGW64 /e/Abhishek (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   waste.html
```

Now commit the changes

```
Hewlett Packard@DESKTOP-BLC9DN3 MINGW64 /e/Abhishek
(master)
$ git commit -a -m "Adding waste"
```

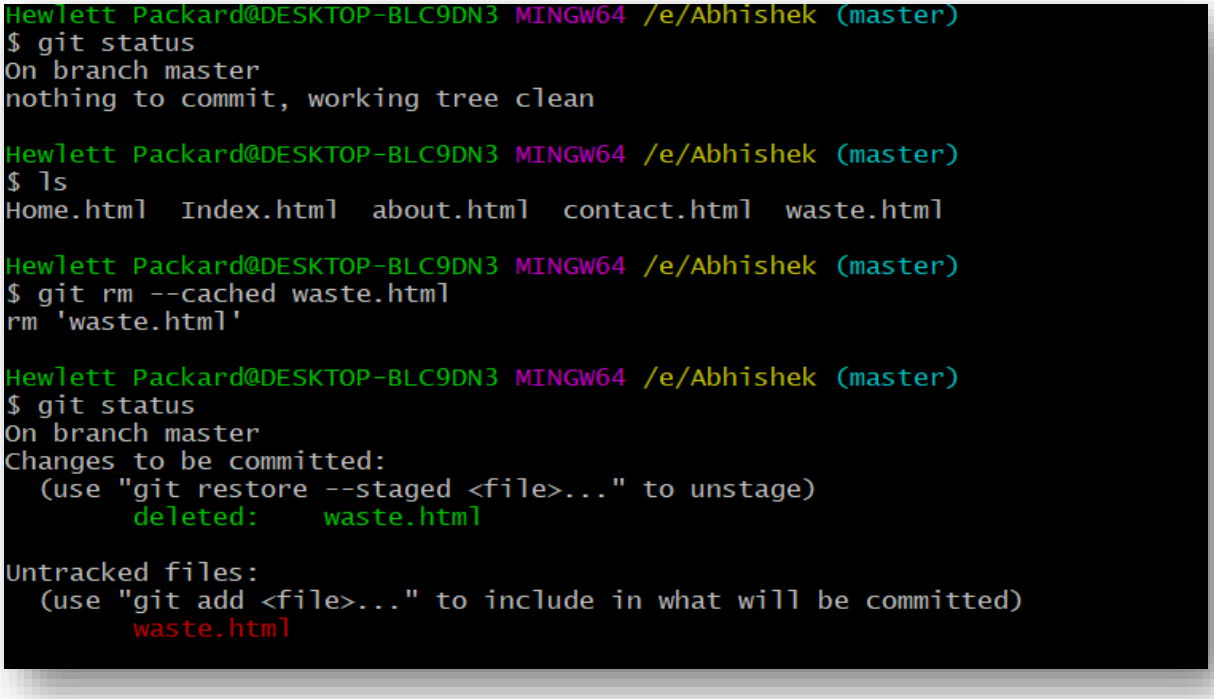
```
Hewlett Packard@DESKTOP-BLC9DN3 MINGW64 /e/Abhishek (master)
$ git commit -a -m "Adding waste"
[master f9aa0f1] Adding waste
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 waste.html
```

```
Hewlett Packard@DESKTOP-BLC9DN3 MINGW64 /e/Abhishek (mas
ter)
$ ls
Home.html  Index.html  about.html  contact.html  waste.html
```

```
Hewlett Packard@DESKTOP-BLC9DN3 MINGW64 /e/Abhishek
(master)
$ git status
On branch master
nothing to commit, working tree clean

Hewlett Packard@DESKTOP-BLC9DN3 MINGW64 /e/Abhishek
(master)
$ ls
Home.html  Index.html  about.html  contact.html
waste.html
```

```
Hewlett Packard@DESKTOP-BLC9DN3 MINGW64 /e/Abhishek
(master)
$ git rm --cached waste.html
rm 'waste.html'

Hewlett Packard@DESKTOP-BLC9DN3 MINGW64 /e/Abhishek
(master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        deleted:    waste.html

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

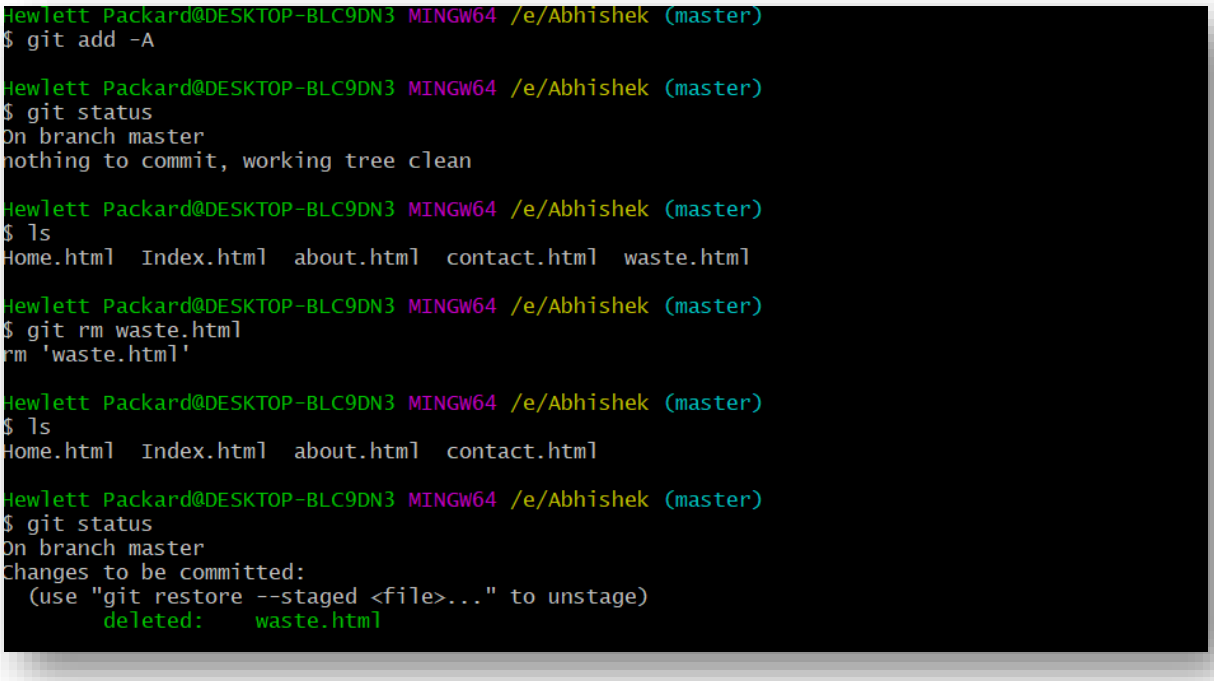
Note: Git rm –cached will remove the file from staging are.



Now if you want to delete file permanently from git then we need use git rm

```
Hewlett Packard@DESKTOP-BLC9DN3 MINGW64 /e/Abhishek (master)
$ git add -A

Hewlett Packard@DESKTOP-BLC9DN3 MINGW64 /e/Abhishek (master)
$ git status
```

```
On branch master
nothing to commit, working tree clean

Hewlett Packard@DESKTOP-BLC9DN3 MINGW64 /e/Abhishek (master)
$ ls
Home.html  Index.html  about.html  contact.html  waste.html

Hewlett Packard@DESKTOP-BLC9DN3 MINGW64 /e/Abhishek (master)
$ git rm waste.html
rm 'waste.html'

Hewlett Packard@DESKTOP-BLC9DN3 MINGW64 /e/Abhishek (master)
$ ls
Home.html  Index.html  about.html  contact.html

Hewlett Packard@DESKTOP-BLC9DN3 MINGW64 /e/Abhishek (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        deleted:    waste.html
```




```
Hewlett Packard@DESKTOP-BLC9DN3 MINGW64 /e/Abhishek (master)
$ git commit -a -m "Removing waste"
[master f2caa12] Removing waste
 1 file changed, 0 insertions(+), 0 deletions(-)
 delete mode 100644 waste.html

Hewlett Packard@DESKTOP-BLC9DN3 MINGW64 /e/Abhishek (master)
$ git status
On branch master
nothing to commit, working tree clean
```

```
Hewlett Packard@DESKTOP-BLC9DN3 MINGW64 /e/Abhishek (master)
$ git commit -a -m "Removing waste"
[master f2caa12] Removing waste
 1 file changed, 0 insertions(+), 0 deletions(-)
 delete mode 100644 waste.html

Hewlett Packard@DESKTOP-BLC9DN3 MINGW64 /e/Abhishek (mas
ter)
$ git status
On branch master
nothing to commit, working tree clean
```

```
Hewlett Packard@DESKTOP-BLC9DN3 MINGW64 /e/Abhishek
(master)
$ git log -p -2
```

```
Hewlett Packard@DESKTOP-BLC9DN3 MINGW64 /e/Abhishek (mas
ter)
$ git log -p -2
commit f2caa127cf375fa50e58660001b251851ee09e20 (HEAD -> master)
Author: Abhishek <abhisharma1108@yahoo.co.in>
Date:   Fri Nov 19 19:58:34 2021 +0530

    Removing waste

diff --git a/waste.html b/waste.html
deleted file mode 100644
index e69de29..0000000

commit f9aa0f15cd23d855de41f98cdae7b5dacdb98392
Author: Abhishek <abhisharma1108@yahoo.co.in>
Date:   Fri Nov 19 19:46:02 2021 +0530

    Adding waste

diff --git a/waste.html b/waste.html
new file mode 100644
index 0000000..e69de29
```

```
Hewlett Packard@DESKTOP-BLC9DN3 MINGW64 /e/Abhishek (master)
$ git log -p -1
commit f2caa127cf375fa50e58660001b251851ee09e20 (HEAD ->
master)
Author: Abhishek <abhisharma1108@yahoo.co.in>
Date:   Fri Nov 19 19:58:34 2021 +0530

    Removing waste

diff --git a/waste.html b/waste.html
deleted file mode 100644
```

```
index e69de29..0000000
```

```
Hewlett Packard@DESKTOP-BLC9DN3 MINGW64 /e/Abhishek (master)
$ git log -p -1
commit f2caa127cf375fa50e58660001b251851ee09e20 (HEAD -> master)
Author: Abhishek <abhisharma1108@yahoo.co.in>
Date:   Fri Nov 19 19:58:34 2021 +0530

    Removing waste

diff --git a/waste.html b/waste.html
deleted file mode 100644
index e69de29..0000000
```

Now I have made the change in my all three files, contact, home and about .If you want to see the summary of all the changes , what file got changes.

```
Hewlett Packard@DESKTOP-BLC9DN3 MINGW64 /e/Abhishek (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:   Home.html
        modified:   contact.html

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

Hewlett Packard@DESKTOP-BLC9DN3 MINGW64 /e/Abhishek (master)
$ git status -s
 M Home.html
 M contact.html

Hewlett Packard@DESKTOP-BLC9DN3 MINGW64 /e/Abhishek (master)
$
```

Now add contact.html in the staging area

```
Hewlett Packard@DESKTOP-BLC9DN3 MINGW64
/e/Abhishek (master)
$ git add contact.html
```

```
Hewlett Packard@DESKTOP-BLC9DN3 MINGW64 /e/Abhishek (master)
$ git status -s
 M Home.html
M  contact.html
```

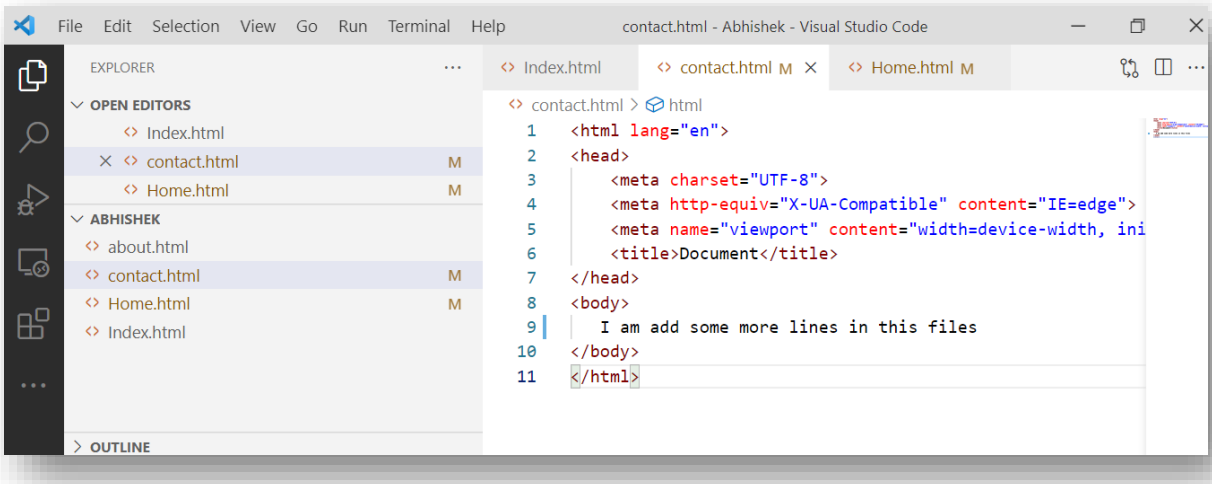
Sort status command

```
Hewlett Packard@DESKTOP-BLC9DN3 MINGW64 /e/Abhishek (master)
$ git status -s
 M Home.html
M contact.html
```

M Home.html this is untracked | modified in working tree

M contact.html this is in staging area.

Now edit contact.html in staging area.



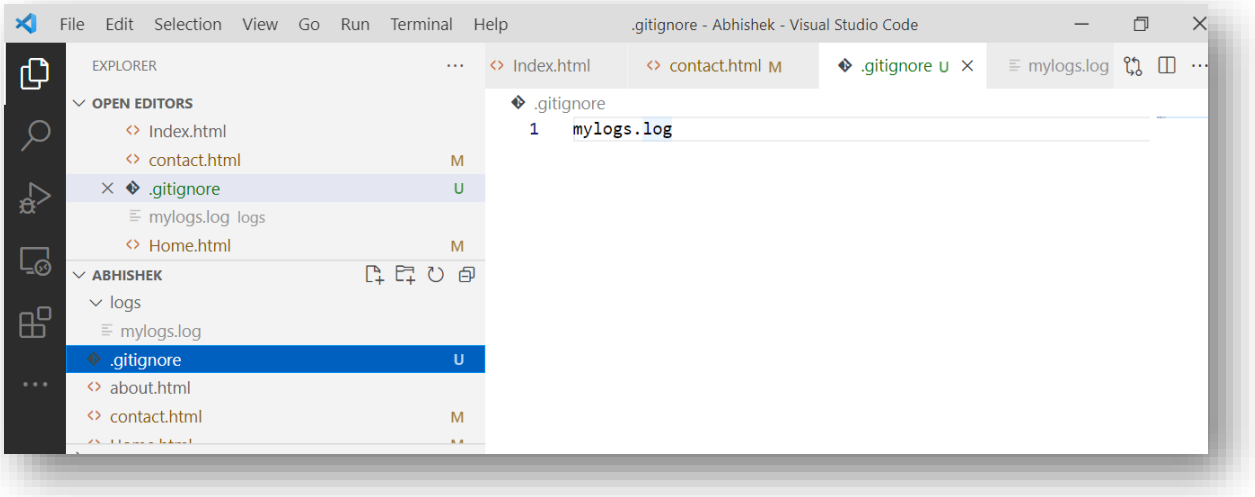

```
Hewlett Packard@DESKTOP-BLC9DN3 MINGW64 /e/Abhishek (master)
$ git status -s
 M Home.html
MM contact.html
```

MM contact.html – Now this is showing edit in staging area also.

```
Hewlett Packard@DESKTOP-BLC9DN3 MINGW64
/e/Abhishek (master)
$ git add contact.html

Hewlett Packard@DESKTOP-BLC9DN3 MINGW64
/e/Abhishek (master)
$ git status -s
 M Home.html
M contact.html
```

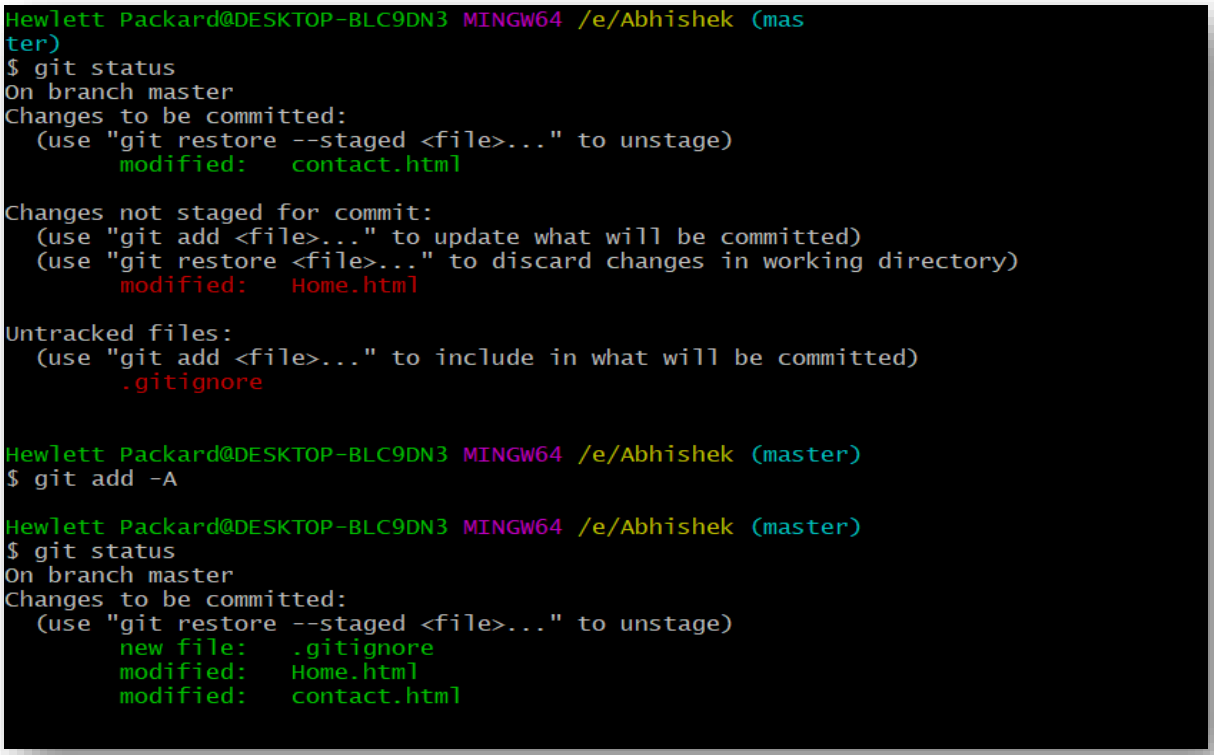
Now you can notice this is ignore mylog.log fie

```
Hewlett Packard@DESKTOP-BLC9DN3 MINGW64
/e/Abhishek (master)
$ touch .gitignore

Hewlett Packard@DESKTOP-BLC9DN3 MINGW64 /e/Abhishek
(master)
$ touch mylogs.log
```

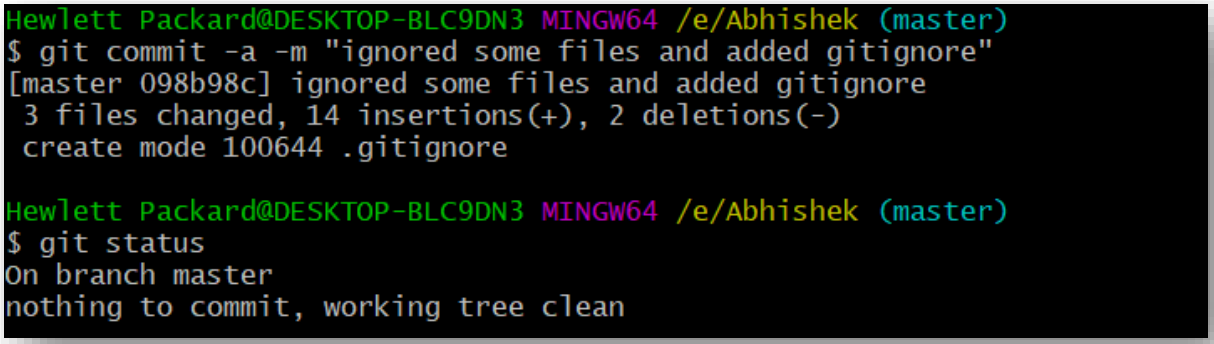



```
Hewlett Packard@DESKTOP-BLC9DN3 MINGW64 /e/Abhishek
(master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   contact.html

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:   Home.html

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


Hewlett Packard@DESKTOP-BLC9DN3 MINGW64 /e/Abhishek
(master)
$ git add -A

Hewlett Packard@DESKTOP-BLC9DN3 MINGW64 /e/Abhishek
(master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   .gitignore
        modified:   Home.html
        modified:   contact.html
```

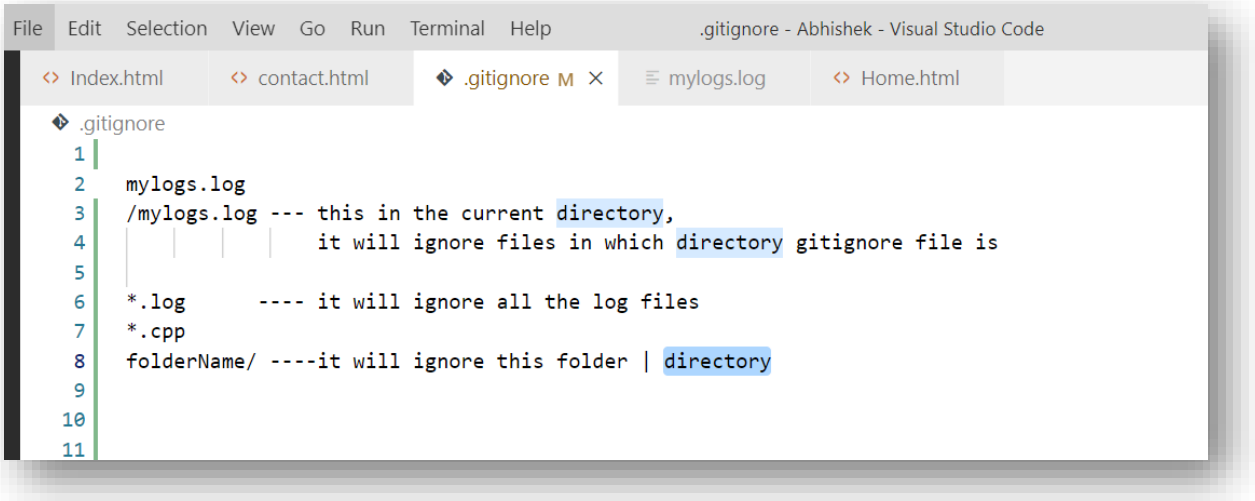


```
Hewlett Packard@DESKTOP-BLC9DN3 MINGW64 /e/Abhishek
(master)
$ git commit -a -m "ignored some files and added
gitignore"
[master 098b98c] ignored some files and added
gitignore
 3 files changed, 14 insertions (+), 2 deletions (-)
 create mode 100644 .gitignore

Hewlett Packard@DESKTOP-BLC9DN3 MINGW64 /e/Abhishek
(master)
$ git status
On branch master
nothing to commit, working tree clean

Hewlett Packard@DESKTOP-BLC9DN3 MINGW64 /e/Abhishek
(master)
```



Point to ponder

```
mylogs.log
/mylogs.log --- this in the current directory,
                it will ignore files in which directory gitignore file is

*.log      ---- it will ignore all the log files
*.cpp
folderName/ ----it will ignore this folder | directory
```




```
Hewlett Packard@DESKTOP-BLC9DN3 MINGW64 /e/Abhishek
(master)
$ git commit -a -m "ignored some files and added
gitignore"
[master 098b98c] ignored some files and added
gitignore
 3 files changed, 14 insertions(+), 2 deletions(-)
 create mode 100644 .gitignore

Hewlett Packard@DESKTOP-BLC9DN3 MINGW64 /e/Abhishek
(master)
$ git status
On branch master
nothing to commit, working tree clean


Hewlett Packard@DESKTOP-BLC9DN3 MINGW64 /e/Abhishek
(master)
$ git commit -a -m "commit changes"
[master 5492045] commit changes
 1 file changed, 8 insertions(+), 1 deletion(-)

Hewlett Packard@DESKTOP-BLC9DN3 MINGW64 /e/Abhishek
(master)
$ git status
On branch master
nothing to commit, working tree clean


Hewlett Packard@DESKTOP-BLC9DN3 MINGW64 /e/Abhishek
(master)
$ git log -p -3
```

Note: press Q to quit log file

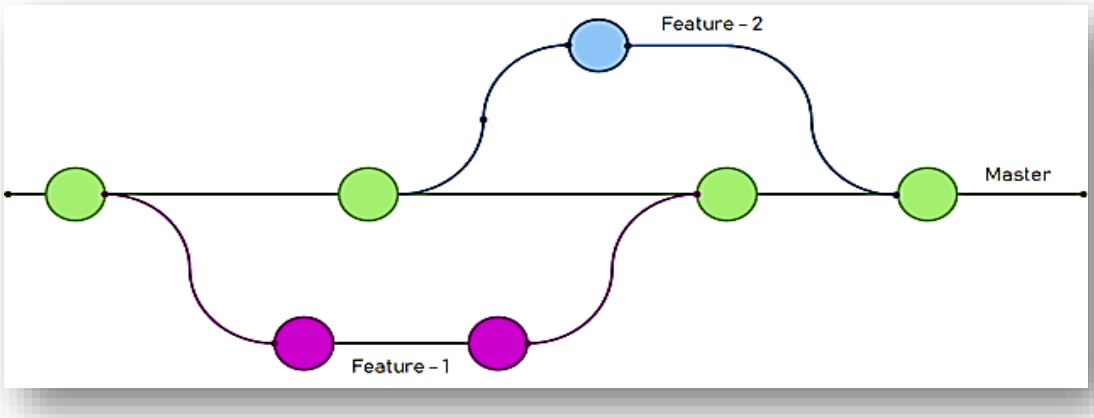
Branch

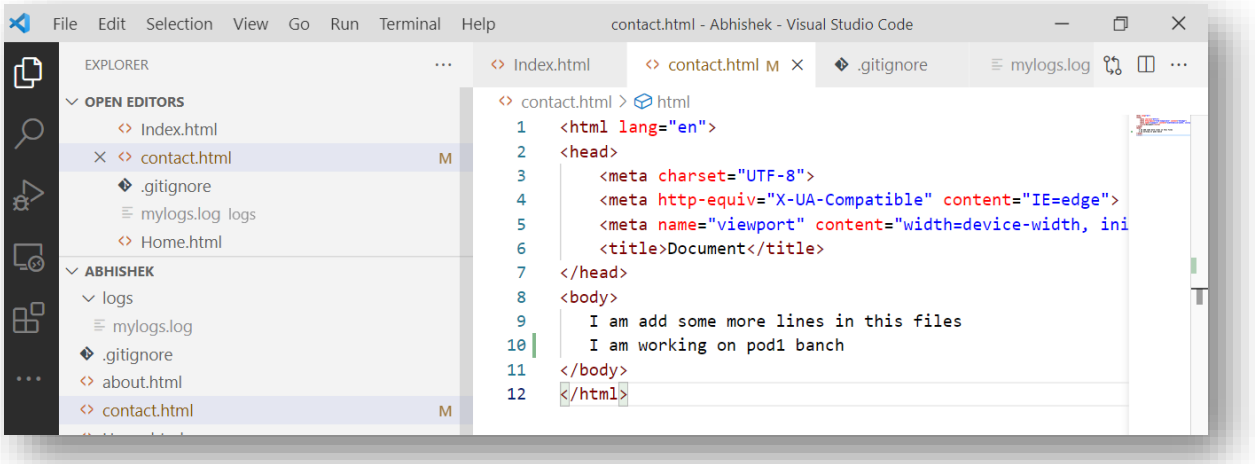
```
Hewlett Packard@DESKTOP-BLC9DN3 MINGW64 /e/Abhishek
(master)
$ git status
On branch master
nothing to commit, working tree clean
```

Git branches are effectively **a pointer to a snapshot of your changes**. ... Instead of copying files from directory to directory, Git stores a branch as a reference to a commit. In this sense, a branch represents the tip of a series of commits—it's not a container for commits.

GIT is the most used and powerful among the Version Control Systems. GIT branch is an important feature of the modern Version Control System. We can say branches in GIT is like a logical pointer of all the set of changes that you have made in the code base. In easy words, GIT branch is the local copy of the main codebase which will have the changes of the new feature or bug fix, which will later be merged with the main codebase once after verified. So, let's deep-dive into the GIT Branch its Operations.

Note: Branch creates a new copy, which doesn't impact on master.



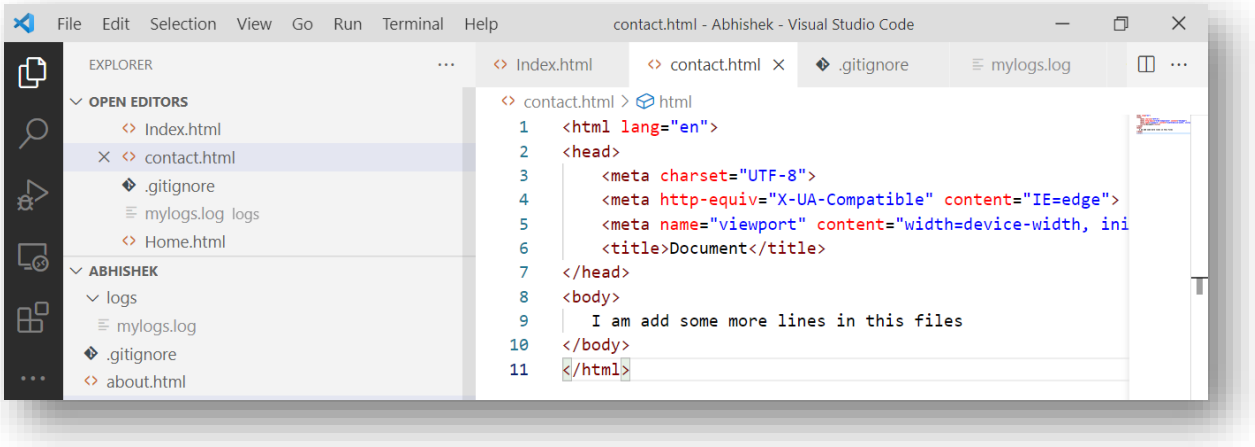



```
Hewlett Packard@DESKTOP-BLC9DN3 MINGW64 /e/Abhishek (master)
$ git status
On branch master
nothing to commit, working tree clean

Hewlett Packard@DESKTOP-BLC9DN3 MINGW64 /e/Abhishek (master)
$ code .

Hewlett Packard@DESKTOP-BLC9DN3 MINGW64 /e/Abhishek (master)
$ git branch
* master

Hewlett Packard@DESKTOP-BLC9DN3 MINGW64 /e/Abhishek (master)
$ git branch pod1

Hewlett Packard@DESKTOP-BLC9DN3 MINGW64 /e/Abhishek (master)
$ git branch
* master
  pod1

Hewlett Packard@DESKTOP-BLC9DN3 MINGW64 /e/Abhishek (master)
$ git checkout pod1
Switched to branch 'pod1'

Hewlett Packard@DESKTOP-BLC9DN3 MINGW64 /e/Abhishek (pod1)
$ git status
On branch pod1
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:   contact.html

no changes added to commit (use "git add" and/or "git commit -a")
Hewlett Packard@DESKTOP-BLC9DN3 MINGW64 /e/Abhishek (pod1)
$ git status
On branch pod1
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:   contact.html

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

Hewlett Packard@DESKTOP-BLC9DN3 MINGW64 /e/Abhishek (pod1)
$ git add -A

Hewlett Packard@DESKTOP-BLC9DN3 MINGW64 /e/Abhishek (pod1)
$ git commit -m "Adding extra line in contact.html file"
[pod1 d46fd86] Adding extra line in contact.html file
 1 file changed, 1 insertion(+)

Hewlett Packard@DESKTOP-BLC9DN3 MINGW64 /e/Abhishek (pod1)
$ git status
On branch pod1
```

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

Now back to main branch and you notice contact.html file back to previous state.

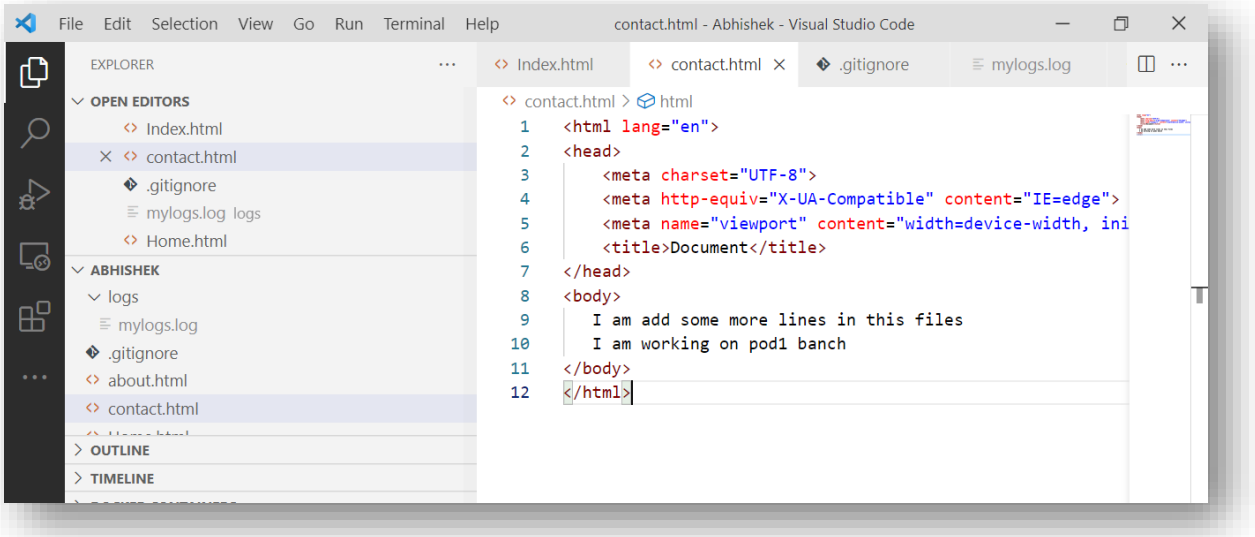
```
Hewlett Packard@DESKTOP-BLC9DN3 MINGW64 /e/Abhishek (pod1)
$ git checkout master
Switched to branch 'master'
```



Now we have to merge pod1 to master

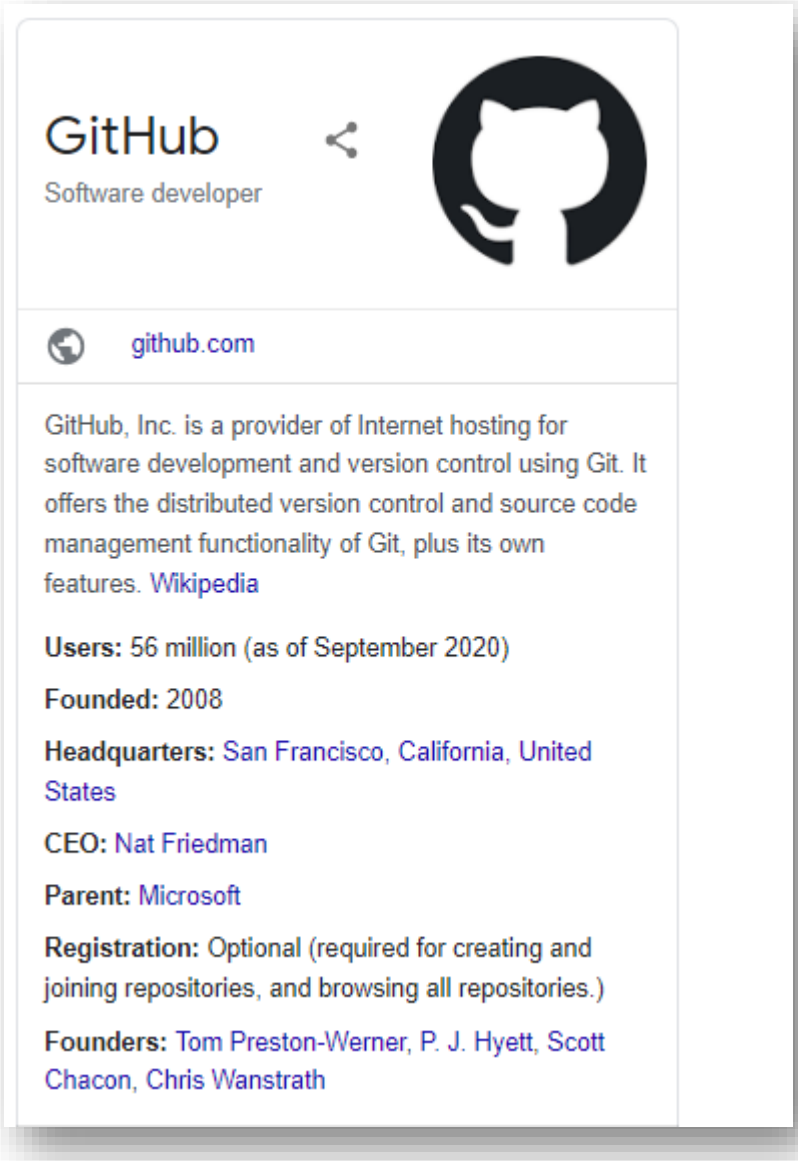
```
Hewlett Packard@DESKTOP-BLC9DN3 MINGW64 /e/Abhishek (master)
$ git status
On branch master
nothing to commit, working tree clean

Hewlett Packard@DESKTOP-BLC9DN3 MINGW64 /e/Abhishek (master)
$ git branch
* master
  pod1

Hewlett Packard@DESKTOP-BLC9DN3 MINGW64 /e/Abhishek (master)
$ git merge pod1
Updating 5492045..d46fd86
Fast-forward
 contact.html | 1 +
 1 file changed, 1 insertion(+)
```

Now you can also notice my branch pod1 merge with master.

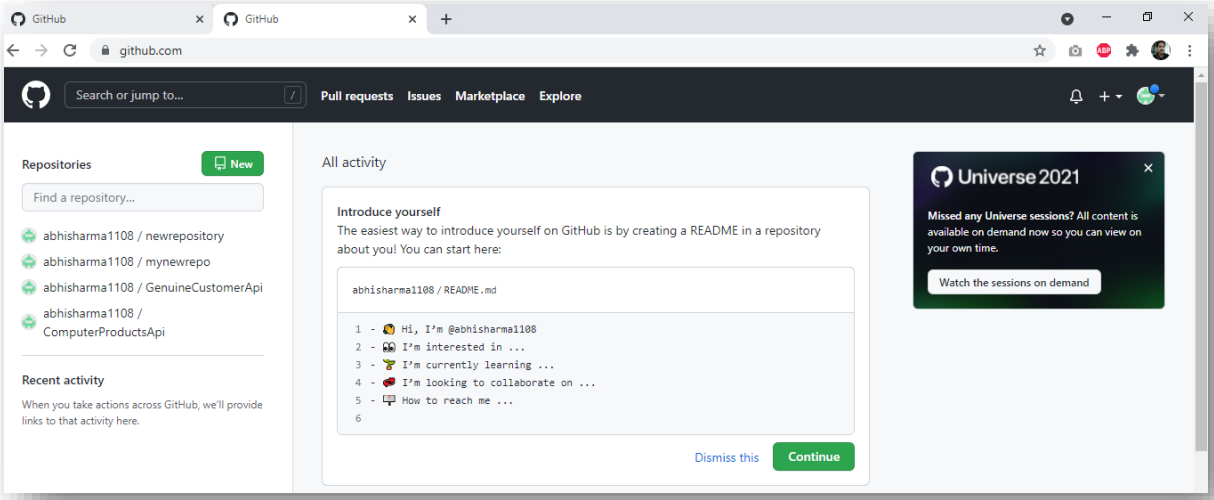


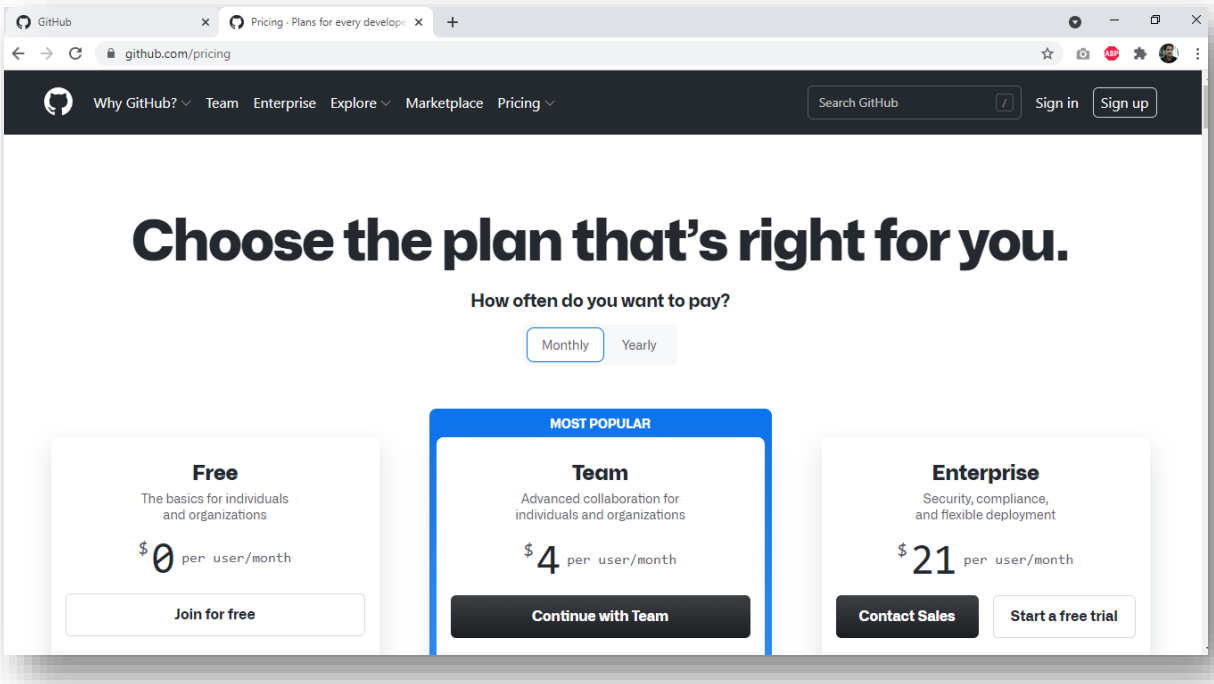

```
Hewlett Packard@DESKTOP-BLC9DN3 MINGW64 /e/Abhishek (master)
$ git checkout -b pod2
Switched to a new branch 'pod2'

Hewlett Packard@DESKTOP-BLC9DN3 MINGW64 /e/Abhishek (pod2)
$ git branch
  master
  pod1
* pod2

Hewlett Packard@DESKTOP-BLC9DN3 MINGW64 /e/Abhishek (pod2)
$ git status
On branch pod2
nothing to commit, working tree clean

Hewlett Packard@DESKTOP-BLC9DN3 MINGW64 /e/Abhishek (pod2)
```

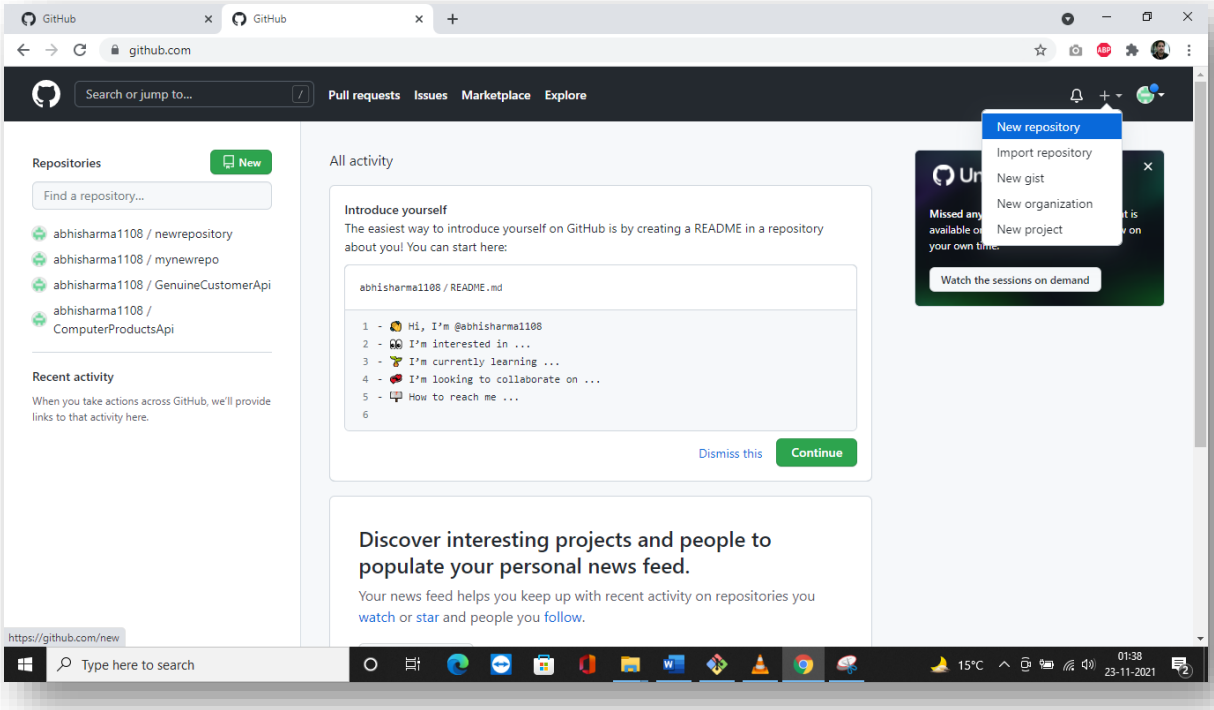
```
Hewlett Packard@DESKTOP-BLC9DN3 MINGW64 /e/Abhishek (master)
$ git checkout -b pod2
Switched to a new branch 'pod2'

Hewlett Packard@DESKTOP-BLC9DN3 MINGW64 /e/Abhishek (pod2)
$ git branch
  master
  pod1
* pod2

Hewlett Packard@DESKTOP-BLC9DN3 MINGW64 /e/Abhishek (pod2)
$ git status
On branch pod2
nothing to commit, working tree clean

Hewlett Packard@DESKTOP-BLC9DN3 MINGW64 /e/Abhishek (pod2)
$
```

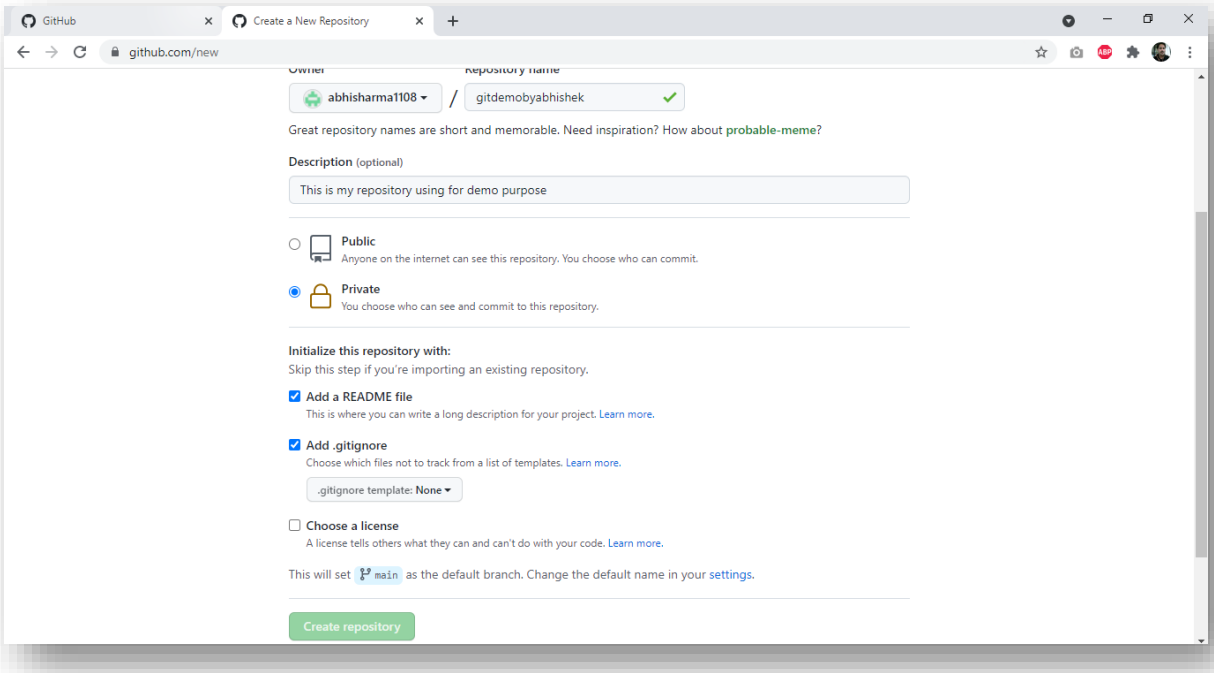


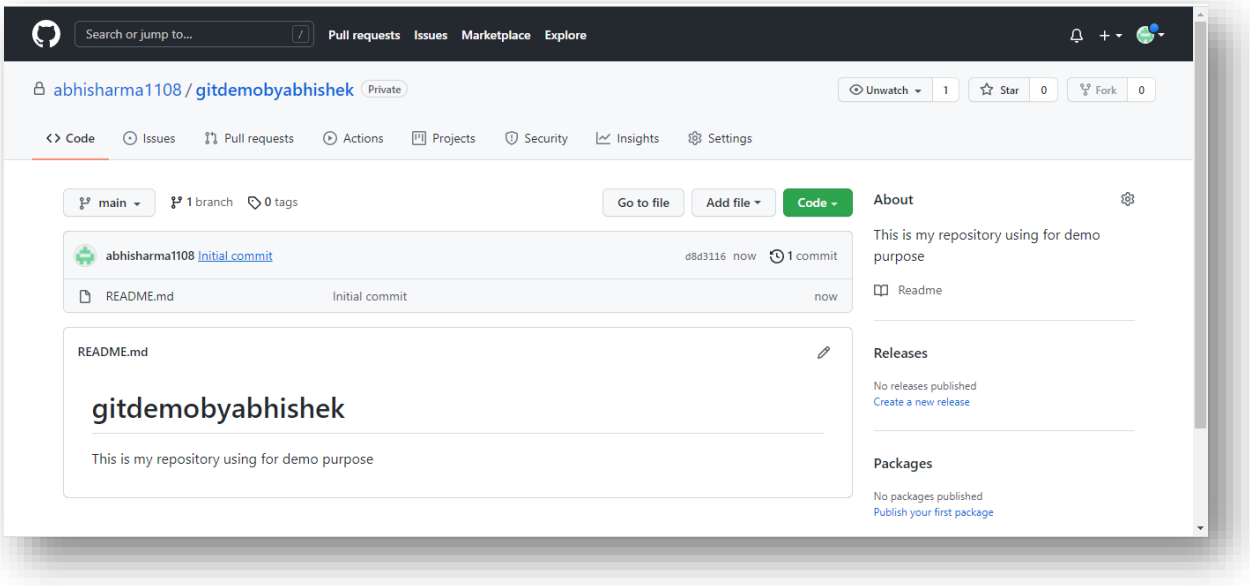
GitHub, Inc. is a provider of Internet hosting for software development and version control using Git. It offers the distributed version control and source code management functionality of Git, plus its own features. Wikipedia


GitHub
GitHub
github.com
Search or jump to...
Pull requests
Issues
Marketplace
Explore
Repositories
New
Find a repository...
abhisharma1108 / newrepository
abhisharma1108 / mynewrepo
abhisharma1108 / GenuineCustomerApi
abhisharma1108 / ComputerProductsApi
Recent activity
When you take actions across GitHub, we'll provide links to that activity here.
All activity
Introduce yourself
The easiest way to introduce yourself on GitHub is by creating a README in a repository about you! You can start here:
abhisharma1108 / README.md
1 - Hi, I'm @abhisharma1108
2 - I'm interested in ...
3 - I'm currently learning ...
4 - I'm looking to collaborate on ...
5 - How to reach me ...
6
Dismiss this
Continue
Universe 2021
Missed any Universe sessions? All content is available on demand now so you can view on your own time.
Watch the sessions on demand

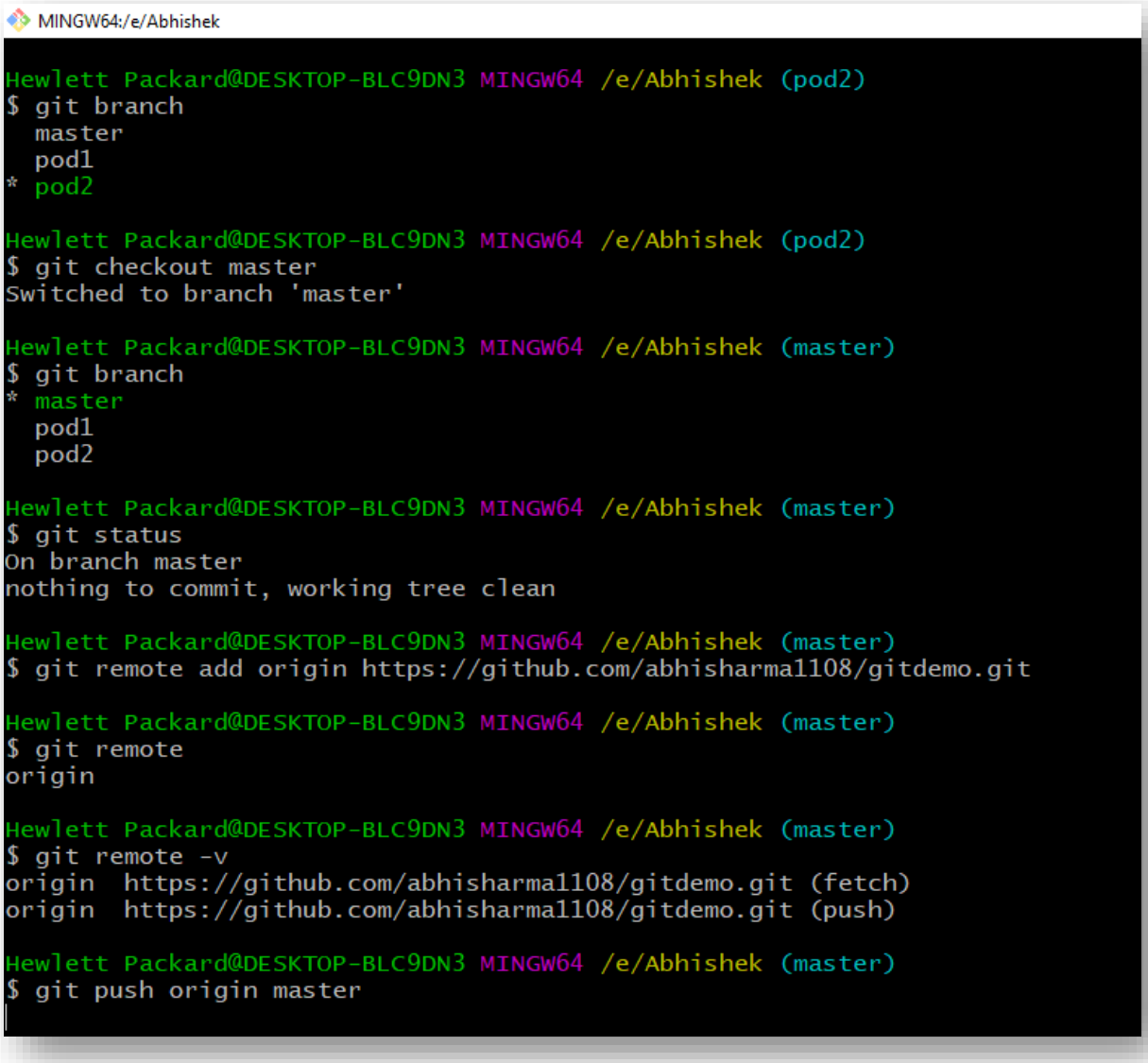

GitHub
Pricing · Plans for every develope
github.com/pricing
Why GitHub?
Team
Enterprise
Explore
Marketplace
Pricing
Search GitHub
Sign in
Sign up
Choose the plan that's right for you.
How often do you want to pay?
Monthly
Yearly
MOST POPULAR
Free
The basics for individuals and organizations
$0 per user/month
Join for free
Team
Advanced collaboration for individuals and organizations
$4 per user/month
Continue with Team
Enterprise
Security, compliance, and flexible deployment
$21 per user/month
Contact Sales
Start a free trial


GitHub
GitHub
github.com
Search or jump to...
Pull requests
Issues
Marketplace
Explore
Repositories
New
Find a repository...
abhisharma1108 / newrepository
abhisharma1108 / mynewrepo
abhisharma1108 / GenuineCustomerApi
abhisharma1108 / ComputerProductsApi
Recent activity
When you take actions across GitHub, we'll provide links to that activity here.
All activity
Introduce yourself
The easiest way to introduce yourself on GitHub is by creating a README in a repository about you! You can start here:
abhisharma1108 / README.md
1 - Hi, I'm @abhisharma1108
2 - I'm interested in ...
3 - I'm currently learning ...
4 - I'm looking to collaborate on ...
5 - How to reach me ...
6
Dismiss this
Continue
Discover interesting projects and people to populate your personal news feed.
Your news feed helps you keep up with recent activity on repositories you watch or star and people you follow.
New repository
Import repository
New gist
New organization
New project
Watch the sessions on demand
https://github.com/new
Type here to search
15°C
01:38
23-11-2021


GitHub
Create a New Repository
github.com/new
abhisharma1108
gitdemobyabhishek
Great repository names are short and memorable. Need inspiration? How about probable-meme?
Description (optional)
This is my repository using for demo purpose
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:
Skip this step if you're importing an existing repository.
Add a README file
This is where you can write a long description for your project. Learn more.
Add .gitignore
Choose which files not to track from a list of templates. Learn more.
.gitignore template: None
Choose a license
A license tells others what they can and can't do with your code. Learn more.
This will set main as the default branch. Change the default name in your settings.
Create repository




```
Hewlett Packard@DESKTOP-BLC9DN3 MINGW64 /e/Abhishek (pod2)
$ git branch
  master
  pod1
* pod2

Hewlett Packard@DESKTOP-BLC9DN3 MINGW64 /e/Abhishek (pod2)
$ git checkout master
Switched to branch 'master'

Hewlett Packard@DESKTOP-BLC9DN3 MINGW64 /e/Abhishek (master)
$ git branch
* master
  pod1
  pod2

Hewlett Packard@DESKTOP-BLC9DN3 MINGW64 /e/Abhishek (master)
$ git status
On branch master
nothing to commit, working tree clean

Hewlett Packard@DESKTOP-BLC9DN3 MINGW64 /e/Abhishek (master)
$ git remote add origin https://github.com/abhisharma1108/gitdemo.git

Hewlett Packard@DESKTOP-BLC9DN3 MINGW64 /e/Abhishek (master)
$ git remote
origin

Hewlett Packard@DESKTOP-BLC9DN3 MINGW64 /e/Abhishek (master)
$ git remote -v
origin  https://github.com/abhisharma1108/gitdemo.git (fetch)
origin  https://github.com/abhisharma1108/gitdemo.git (push)

Hewlett Packard@DESKTOP-BLC9DN3 MINGW64 /e/Abhishek (master)
$ git push origin master
```



POINT TO PONDER:

```
$ git push -u origin master
```

With the -u tag

Next time it will push to the origin master

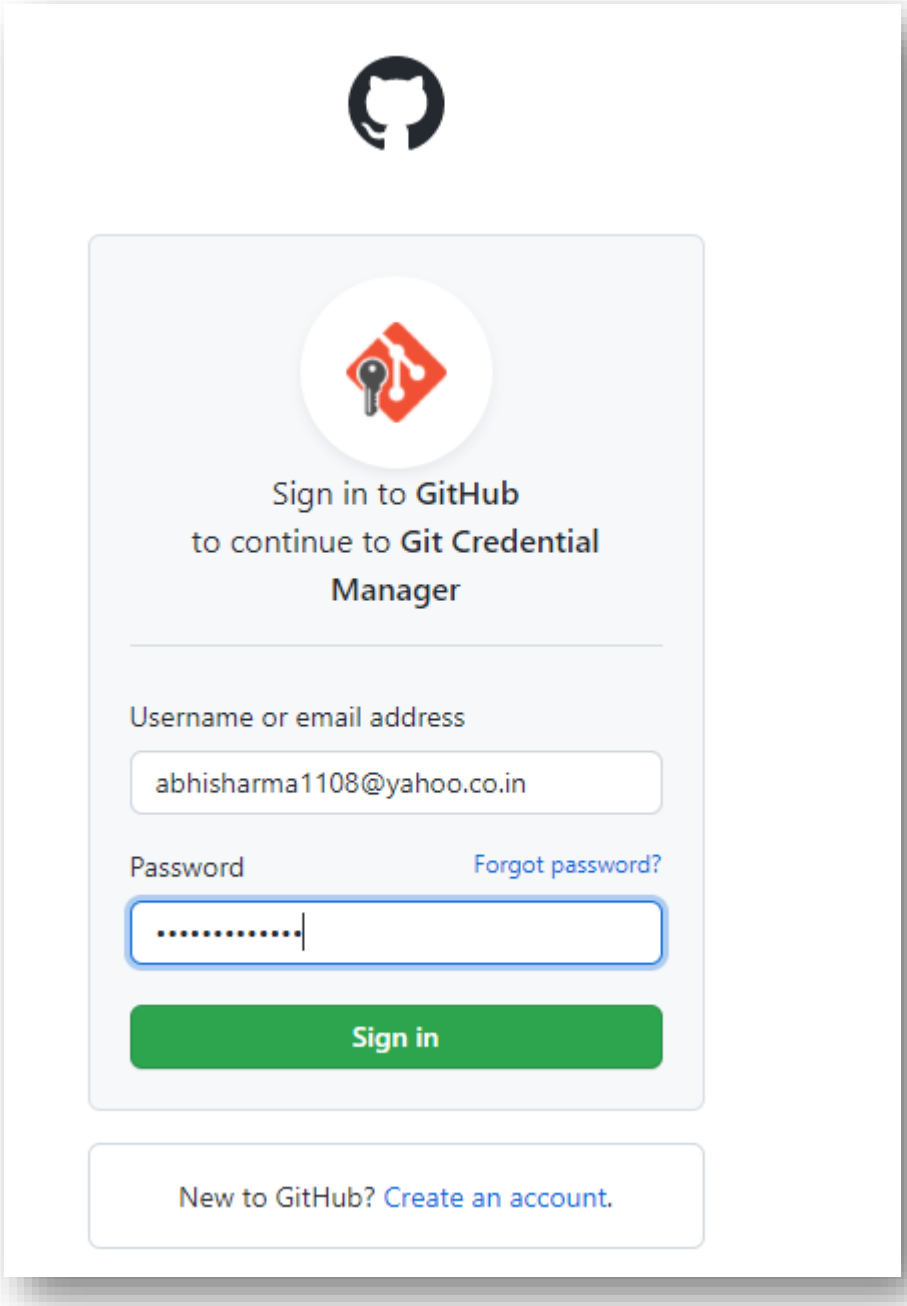
```
$ git push
```

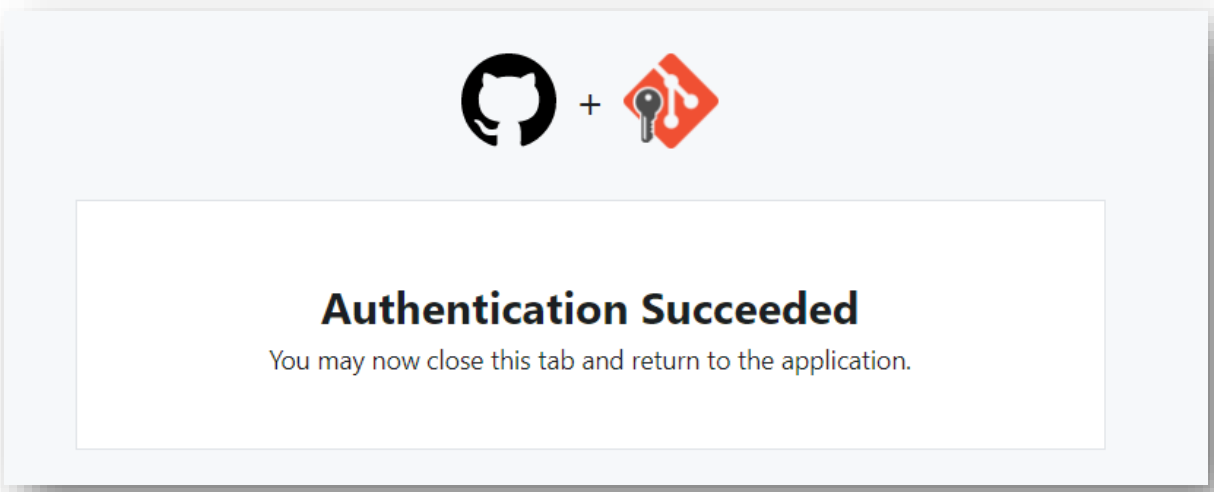
```
$ git push -f origin master
```

With the -f tag you will override the *remote branch code* with your local repo code.

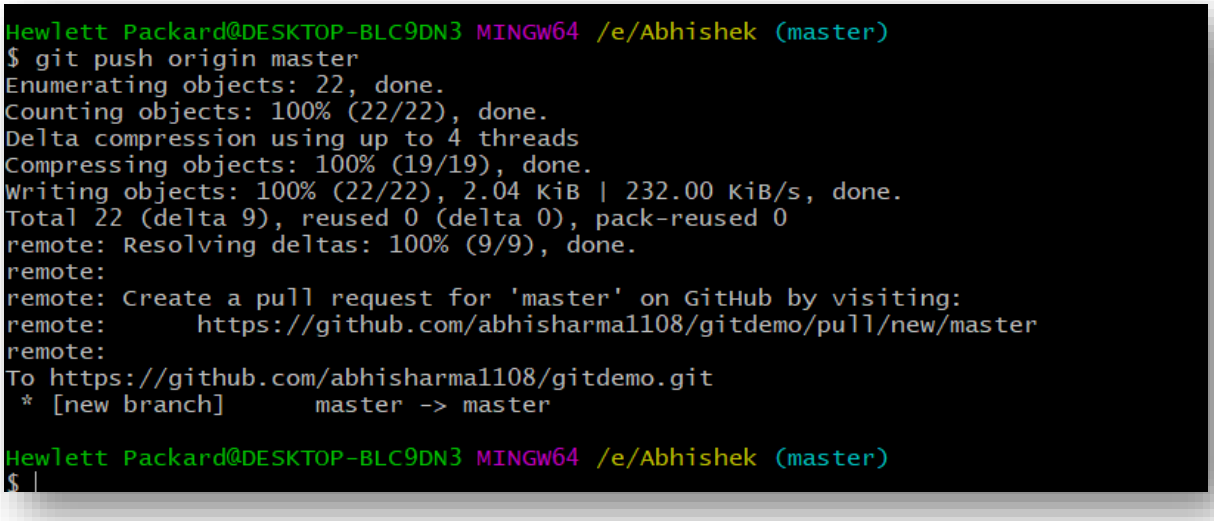
Git Push Origin vs Git Push Origin Master

Git Push Origin	Git Push Origin Master
Git Push Origin pushes all the branches to the main branch.	Git Push Origin Master pushes your master branch to the origin.
Command for this: *git push origin*	Command for this: *git push origin master*
Behavior could be changed via git config.	Behaviour is by default.


Sign in to GitHub
to continue to Git Credential Manager
Username or email address
abhisharma1108@yahoo.co.in
Password
Forgot password?
Sign in
New to GitHub? Create an account.

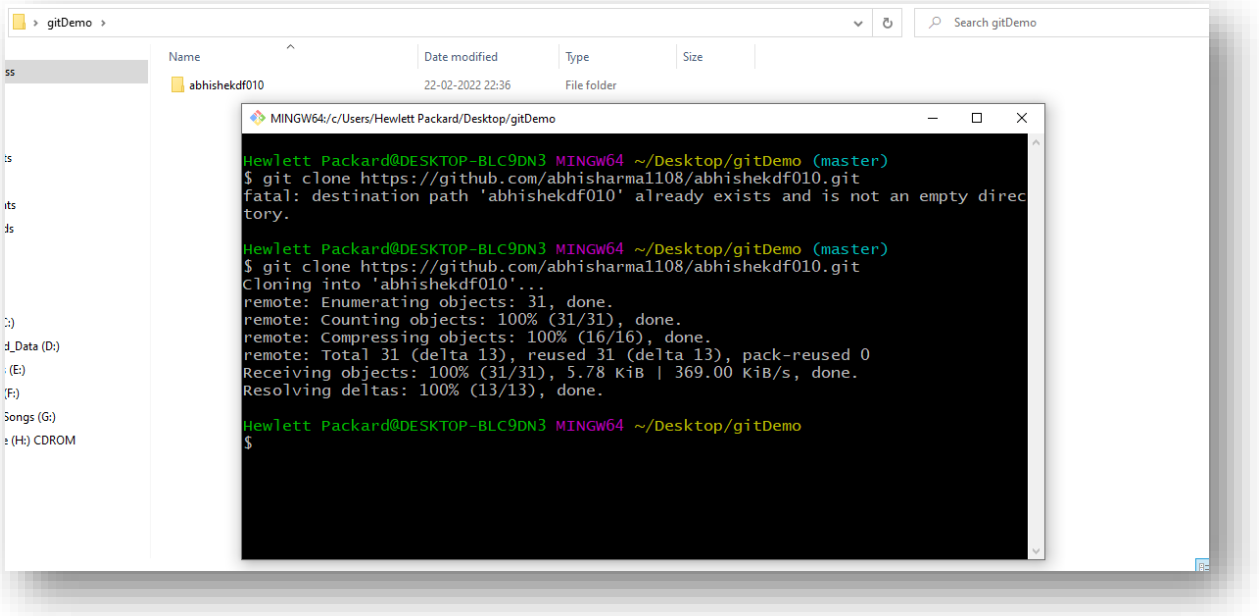



```
Hewlett Packard@DESKTOP-BLC9DN3 MINGW64 /e/Abhishek (master)
$ git push origin master
Enumerating objects: 22, done.
Counting objects: 100% (22/22), done.
Delta compression using up to 4 threads
Compressing objects: 100% (19/19), done.
Writing objects: 100% (22/22), 2.04 KiB | 232.00 KiB/s, done.
Total 22 (delta 9), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (9/9), done.
remote:
remote: Create a pull request for 'master' on GitHub by visiting:
remote:      https://github.com/abhisharma1108/gitdemo/pull/new/master
remote:
To https://github.com/abhisharma1108/gitdemo.git
 * [new branch]      master -> master
```



Note: Creating clone of existing repository

git clone https://github.com/abhisharma1108/abhishekdf010.git



AZURE : DEPLOYMENT

```
Hewlett Packard@DESKTOP-BLC9DN3 MINGW64
~/source/repos/ComputerProductWebAppGitHub (master)
$ git push -f origin master
Enumerating objects: 49, done.
Counting objects: 100% (49/49), done.
Delta compression using up to 4 threads
Compressing objects: 100% (22/22), done.
Writing objects: 100% (28/28), 31.40 KiB | 1004.00 KiB/s,
done.
Total 28 (delta 14), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (14/14), completed with 12
local objects.
To
https://github.com/abhisharma1108/ComputerProductWebAppGitHub.
git
 + 633e85a...7856b8b master -> master (forced update)
```

```
Hewlett Packard@DESKTOP-BLC9DN3 MINGW64
/e/INTADM22DF043/GITDEMO/temp/INTADM22DF043 (main)
$ git remote
origin

Hewlett Packard@DESKTOP-BLC9DN3 MINGW64
/e/INTADM22DF043/GITDEMO/temp/INTADM22DF043 (main)
$ git remote -v
origin  https://github.com/abhisharma1108/INTADM22DF043.git
(fetch)
origin  https://github.com/abhisharma1108/INTADM22DF043.git
(push)

Hewlett Packard@DESKTOP-BLC9DN3 MINGW64
/e/INTADM22DF043/GITDEMO/temp/INTADM22DF043 (main)
$ git fetch --all
Fetching origin

Hewlett Packard@DESKTOP-BLC9DN3 MINGW64
/e/INTADM22DF043/GITDEMO/temp/INTADM22DF043 (main)
$ ls -l
total 1
-rw-r--r-- 1 Hewlett Packard 197121 76 Jun  1 18:17 README.md

Hewlett Packard@DESKTOP-BLC9DN3 MINGW64
/e/INTADM22DF043/GITDEMO/temp/INTADM22DF043 (main)
$ git pull --all
Fetching origin
Already up to date.

Hewlett Packard@DESKTOP-BLC9DN3 MINGW64
/e/INTADM22DF043/GITDEMO/temp/INTADM22DF043 (main)
$ git checkout master
Switched to a new branch 'master'
Branch 'master' set up to track remote branch 'master' from
'origin'.

Hewlett Packard@DESKTOP-BLC9DN3 MINGW64
/e/INTADM22DF043/GITDEMO/temp/INTADM22DF043 (master)
$ git pull --all
Fetching origin
Already up to date.

Hewlett Packard@DESKTOP-BLC9DN3 MINGW64
/e/INTADM22DF043/GITDEMO/temp/INTADM22DF043 (master)
$ git fetch --all
Fetching origin

Hewlett Packard@DESKTOP-BLC9DN3 MINGW64
/e/INTADM22DF043/GITDEMO/temp/INTADM22DF043 (master)
$ git branch
  main
* master

Hewlett Packard@DESKTOP-BLC9DN3 MINGW64
/e/INTADM22DF043/GITDEMO/temp/INTADM22DF043 (master)
$
```