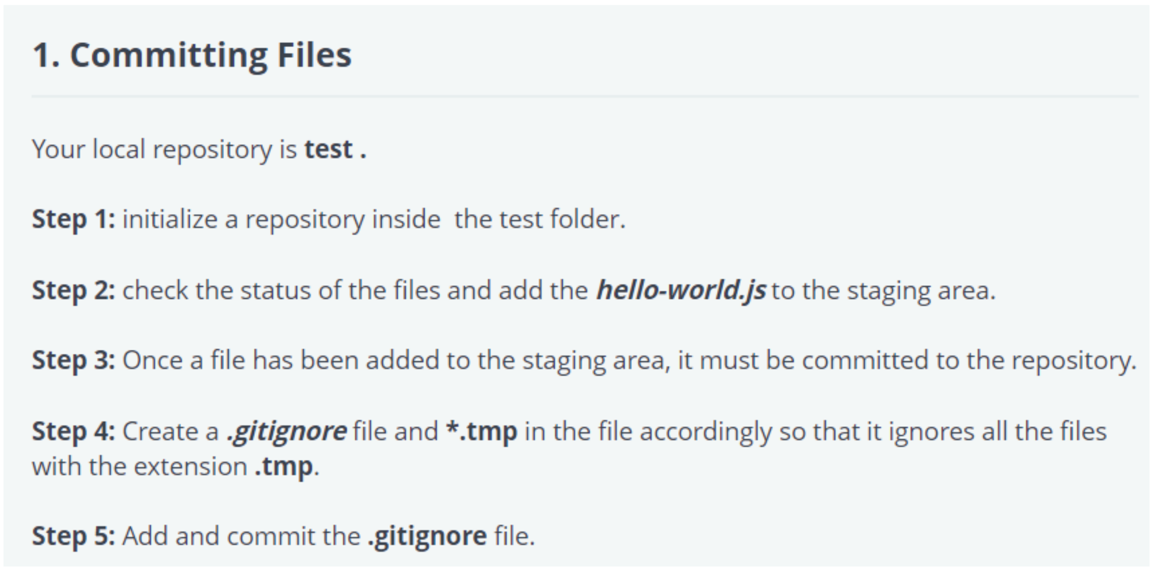


1. Committing Files

Your local repository is **test .**

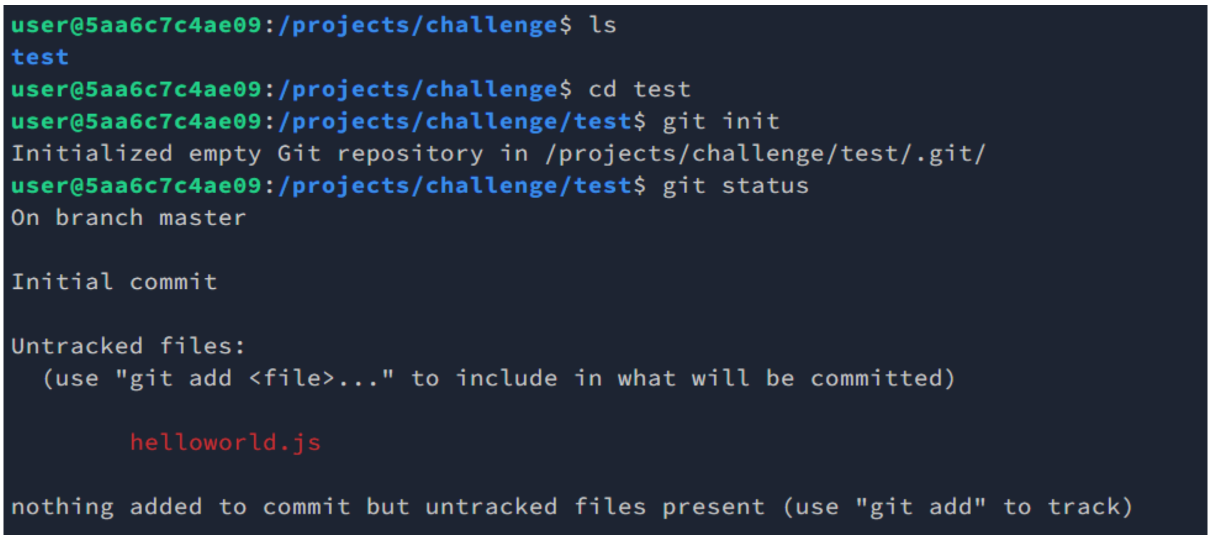
Step 1: initialize a repository inside the test folder.

Step 2: check the status of the files and add the ***hello-world.js*** to the staging area.

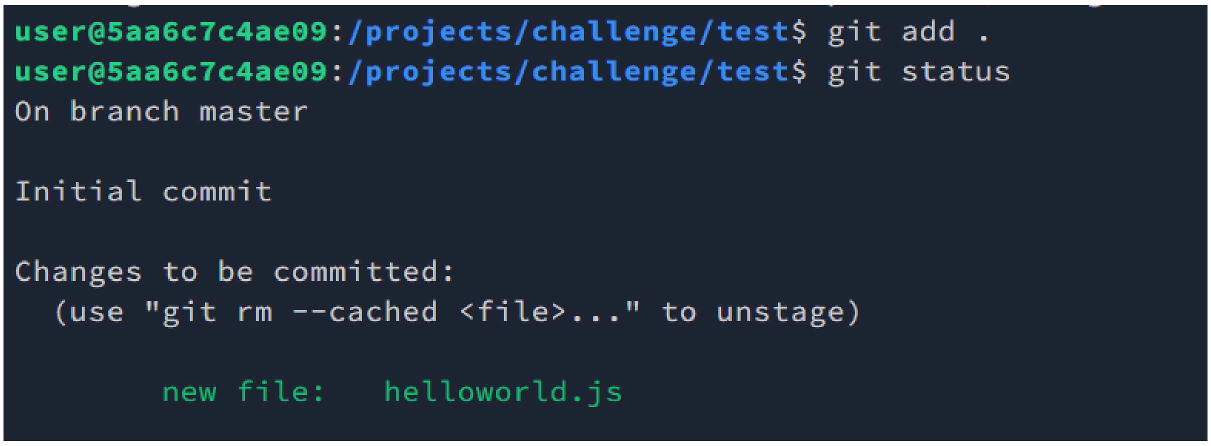
Step 3: Once a file has been added to the staging area, it must be committed to the repository.

Step 4: Create a ***.gitignore*** file and ***.tmp** in the file accordingly so that it ignores all the files with the extension **.tmp**.

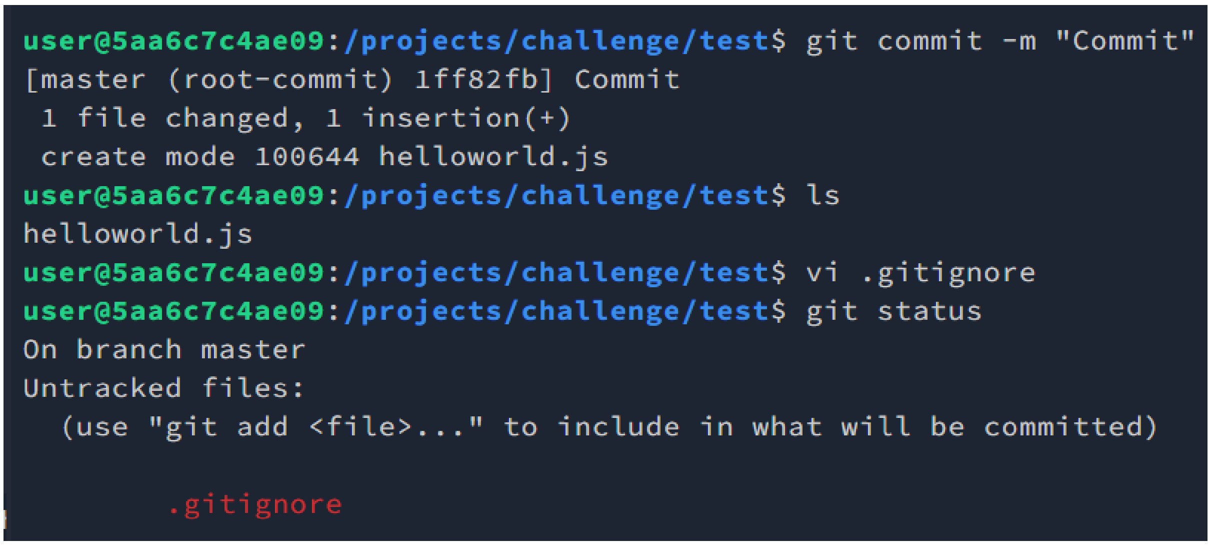
Step 5: Add and commit the **.gitignore** file.



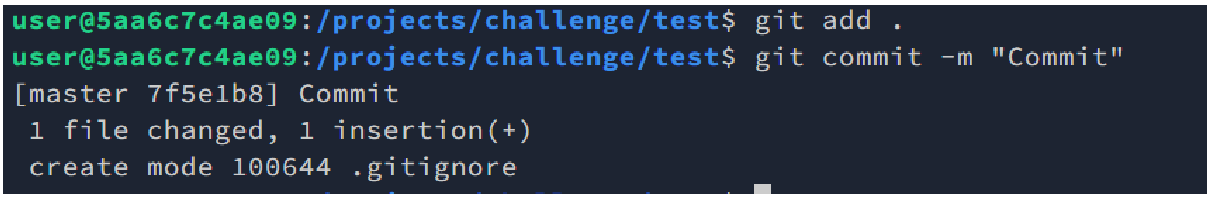
```
user@5aa6c7c4ae09:/projects/challenge$ ls
test
user@5aa6c7c4ae09:/projects/challenge$ cd test
user@5aa6c7c4ae09:/projects/challenge/test$ git init
Initialized empty Git repository in /projects/challenge/test/.git/
user@5aa6c7c4ae09:/projects/challenge/test$ git status
On branch master

Initial commit

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

        helloworld.js

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



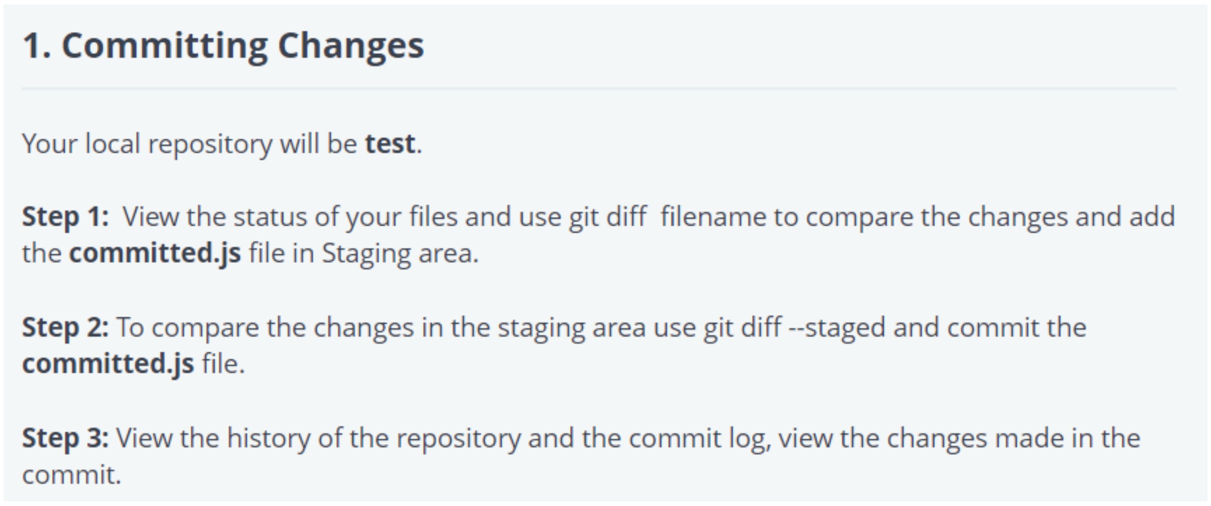
```
user@5aa6c7c4ae09:/projects/challenge/test$ git add .
user@5aa6c7c4ae09:/projects/challenge/test$ git status
On branch master

Initial commit

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

        new file:   helloworld.js
```



```
user@5aa6c7c4ae09:/projects/challenge/test$ git commit -m "Commit"
[master (root-commit) 1ff82fb] Commit
 1 file changed, 1 insertion(+)
 create mode 100644 helloworld.js
user@5aa6c7c4ae09:/projects/challenge/test$ ls
helloworld.js
user@5aa6c7c4ae09:/projects/challenge/test$ vi .gitignore
user@5aa6c7c4ae09:/projects/challenge/test$ git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)

        .gitignore
```



```
user@5aa6c7c4ae09:/projects/challenge/test$ git add .
user@5aa6c7c4ae09:/projects/challenge/test$ git commit -m "Commit"
[master 7f5e1b8] Commit
 1 file changed, 1 insertion(+)
 create mode 100644 .gitignore
```



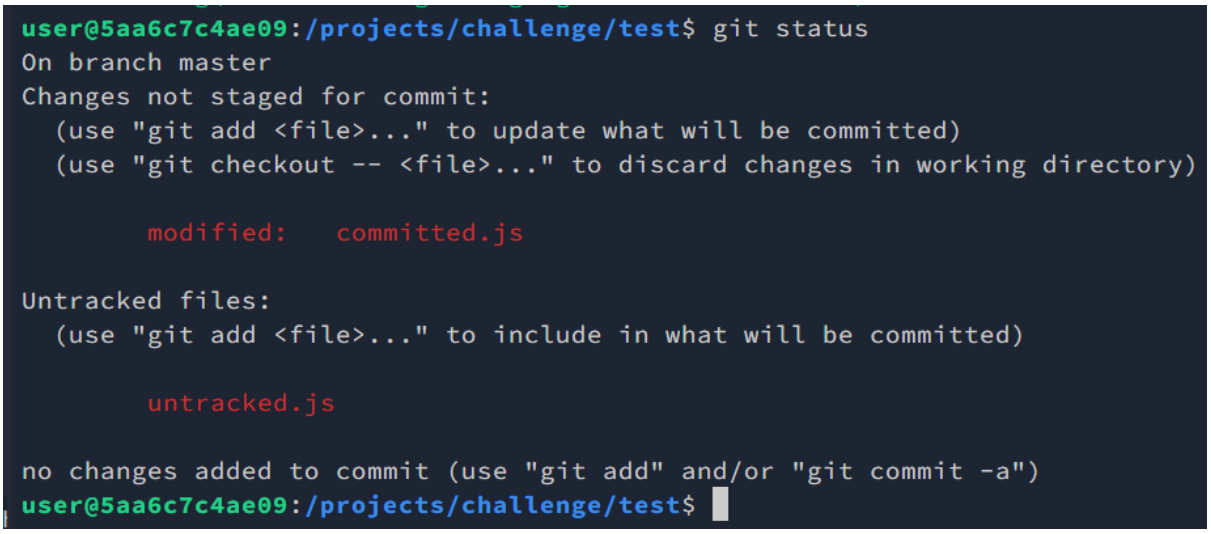
1. Committing Changes

Your local repository will be **test**.

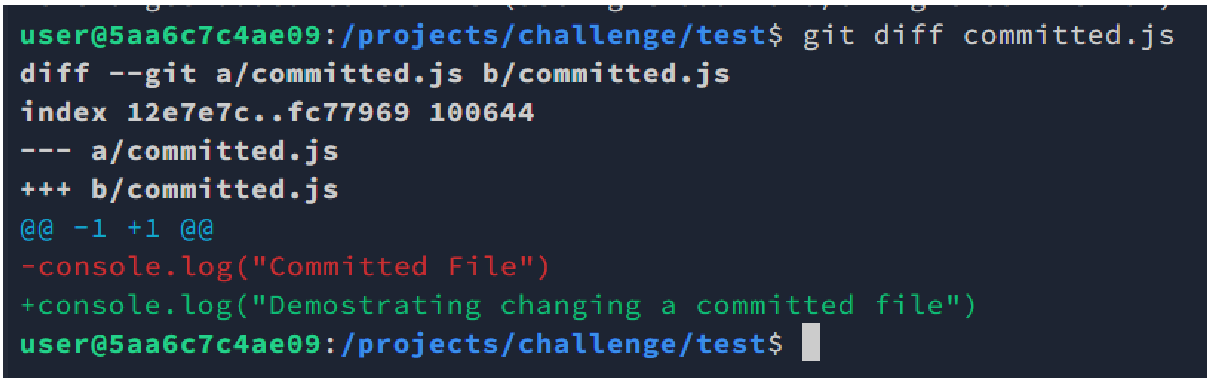
Step 1: View the status of your files and use git diff filename to compare the changes and add the **committed.js** file in Staging area.

Step 2: To compare the changes in the staging area use git diff --staged and commit the **committed.js** file.

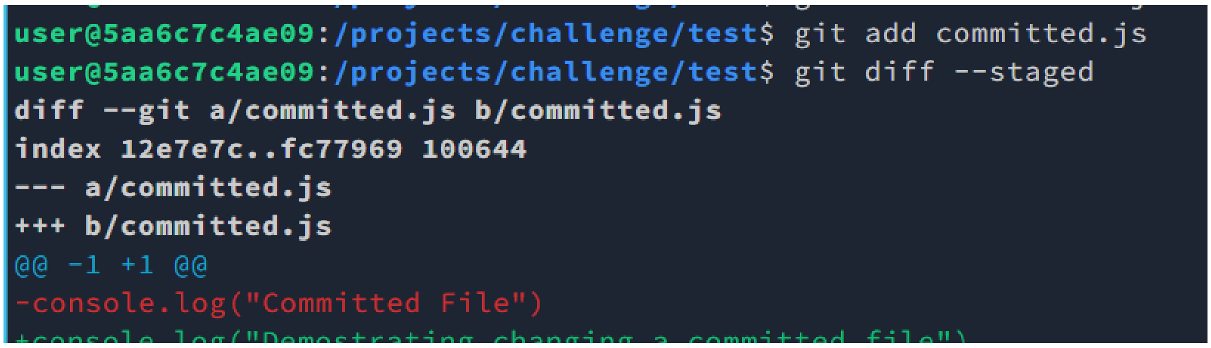
Step 3: View the history of the repository and the commit log, view the changes made in the commit.



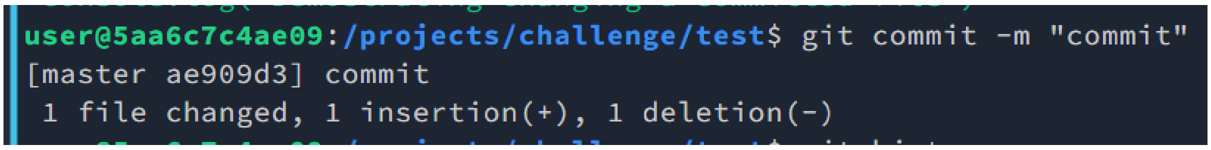
```
user@5aa6c7c4ae09:/projects/challenge/test$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   committed.js

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

        untracked.js

no changes added to commit (use "git add" and/or "git commit -a")
user@5aa6c7c4ae09:/projects/challenge/test$
```



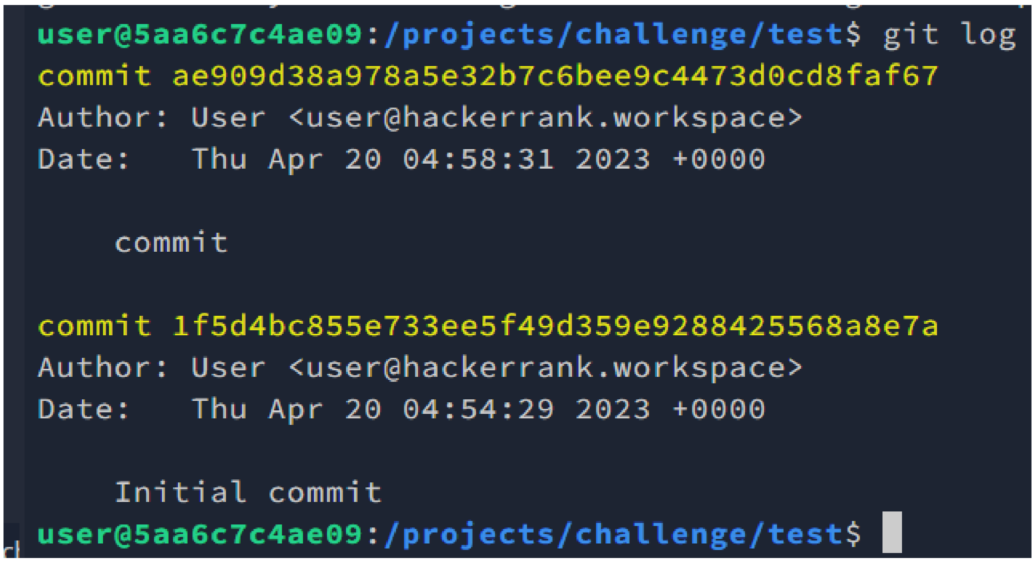
```
user@5aa6c7c4ae09:/projects/challenge/test$ git diff committed.js
diff --git a/committed.js b/committed.js
index 12e7e7c..fc77969 100644
--- a/committed.js
+++ b/committed.js
@@ -1 +1 @@
-console.log("Committed File")
+console.log("Demostrating changing a committed file")
user@5aa6c7c4ae09:/projects/challenge/test$
```



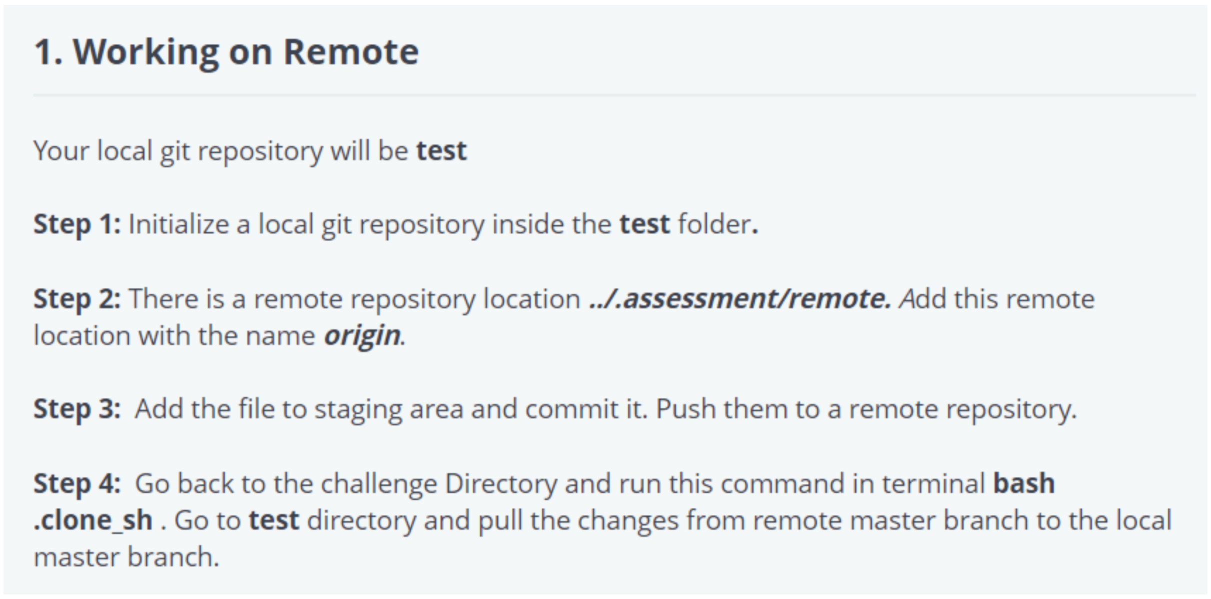
```
user@5aa6c7c4ae09:/projects/challenge/test$ git add committed.js
user@5aa6c7c4ae09:/projects/challenge/test$ git diff --staged
diff --git a/committed.js b/committed.js
index 12e7e7c..fc77969 100644
--- a/committed.js
+++ b/committed.js
@@ -1 +1 @@
-console.log("Committed File")
+console.log("Demostrating changing a committed file")
```



```
user@5aa6c7c4ae09:/projects/challenge/test$ git commit -m "commit"
[master ae909d3] commit
 1 file changed, 1 insertion(+), 1 deletion(-)
```



```
user@5aa6c7c4ae09:/projects/challenge/test$ git log
commit ae909d38a978a5e32b7c6bee9c4473d0cd8faf67
Author: User <user@hackerrank.workspace>
Date:   Thu Apr 20 04:58:31 2023 +0000

    commit

commit 1f5d4bc855e733ee5f49d359e9288425568a8e7a
Author: User <user@hackerrank.workspace>
Date:   Thu Apr 20 04:54:29 2023 +0000

    Initial commit
user@5aa6c7c4ae09:/projects/challenge/test$
```



1. Working on Remote

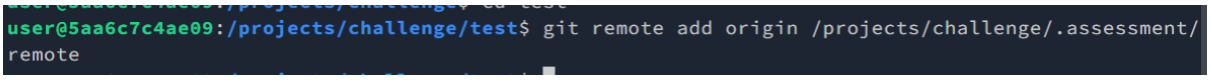
Your local git repository will be **test**

Step 1: Initialize a local git repository inside the **test** folder.

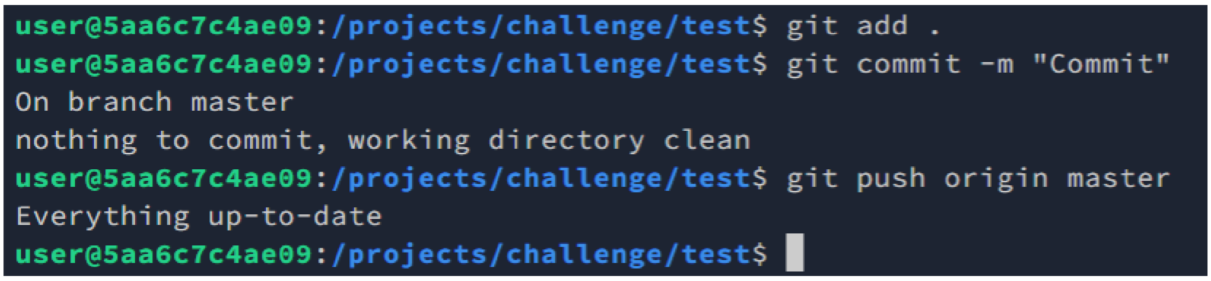
Step 2: There is a remote repository location ***../.assessment/remote.*** *Add this remote location with the name* ***origin****.*

Step 3: Add the file to staging area and commit it. Push them to a remote repository.

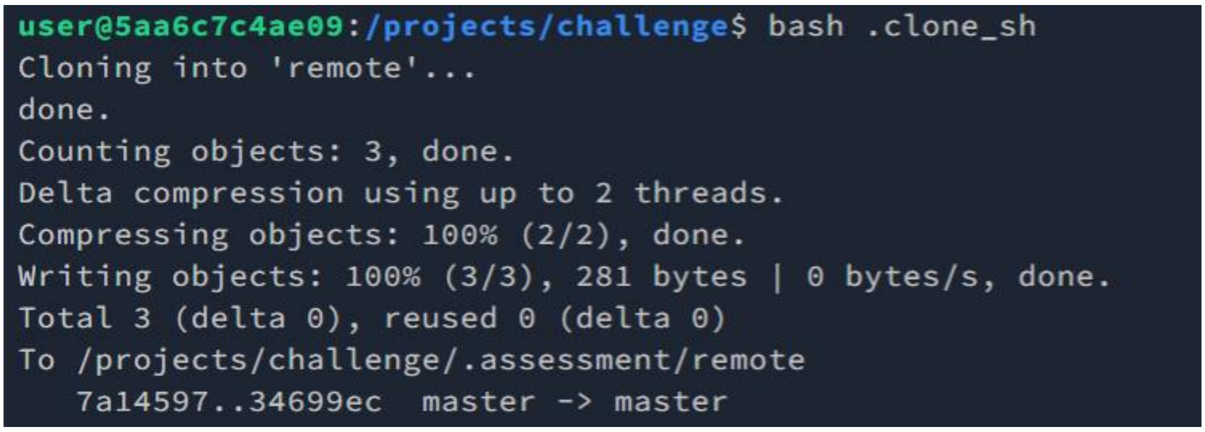
Step 4: Go back to the challenge Directory and run this command in terminal **bash .clone_sh** . Go to **test** directory and pull the changes from remote master branch to the local master branch.



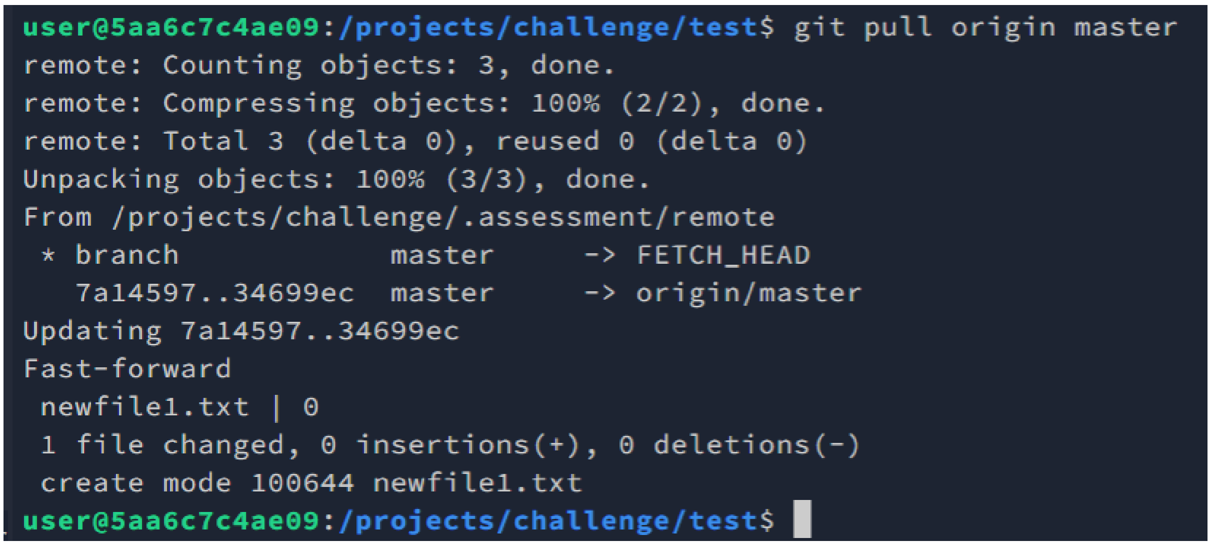
```
user@5aa6c7c4ae09:/projects/challenge/test$ git remote add origin /projects/challenge/.assessment/
remote
```



```
user@5aa6c7c4ae09:/projects/challenge/test$ git add .
user@5aa6c7c4ae09:/projects/challenge/test$ git commit -m "Commit"
On branch master
nothing to commit, working directory clean
user@5aa6c7c4ae09:/projects/challenge/test$ git push origin master
Everything up-to-date
user@5aa6c7c4ae09:/projects/challenge/test$
```



```
user@5aa6c7c4ae09:/projects/challenge$ bash .clone_sh
Cloning into 'remote'...
done.
Counting objects: 3, done.
Delta compression using up to 2 threads.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 281 bytes | 0 bytes/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To /projects/challenge/.assessment/remote
   7a14597..34699ec  master -> master
```


```
user@5aa6c7c4ae09:/projects/challenge/test$ git pull origin master
remote: Counting objects: 3, done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0)
Unpacking objects: 100% (3/3), done.
From /projects/challenge/.assessment/remote
 * branch            master     -> FETCH_HEAD
   7a14597..34699ec  master     -> origin/master
Updating 7a14597..34699ec
Fast-forward
 newfile1.txt | 0
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 newfile1.txt
user@5aa6c7c4ae09:/projects/challenge/test$
```

1. Try it Out - Branch and Merge Conflicts

Your local repository is **test** directory.

Step1: A local Git repository with the file **hello.py** is created. Add two branches to the repository with name of **feature1** and **feature2.** Go to feature1 branch and make changes in the hello.py file. Remove the existing content and update the content as **print('hello people').** And commit the file.

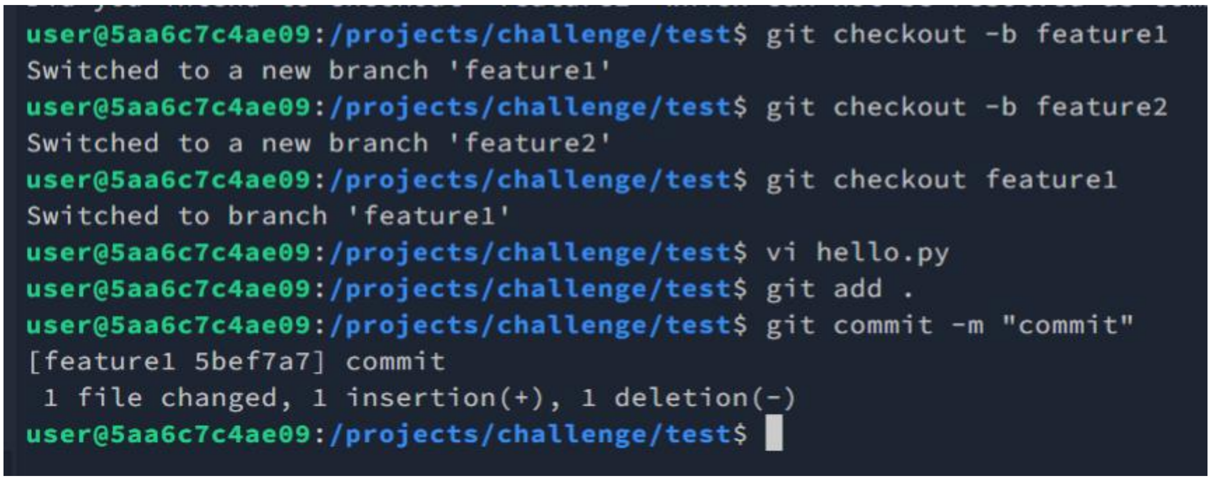
Step 2: Go to master and merge the changes made in the feature1 branch to the master branch.
Result Sample:

```
Updating f133940..ad6bdf3
Fast-forward
 hello.py | 2 +-
 1 file changed, 1 insertion(+), 1 deletion(-)
```

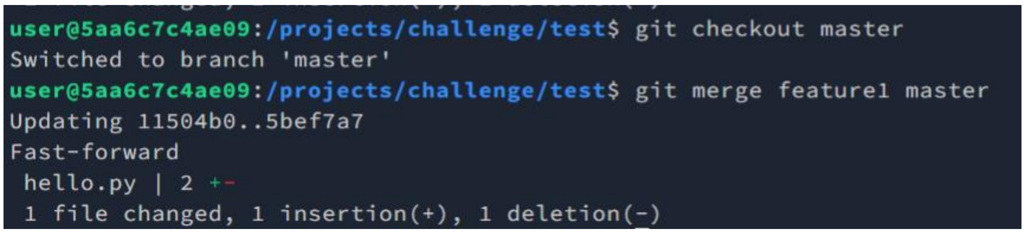
Step 3: Go to feature2 branch and make changes in the hello.py file. Remove the existing content and update the content as **print('keep playing').** And commit the file. Go to master and merge the changes made in the feature2 branch to the master branch.
Auto-merging hello.py CONFLICT (content): Merge conflict in hello.py Automatic merge failed; fix conflicts and then commit the result.

Step 4: View the **hello.py** file, remove all the unwanted content and only keep **print('keep playing').** After that stage and commit the changes.

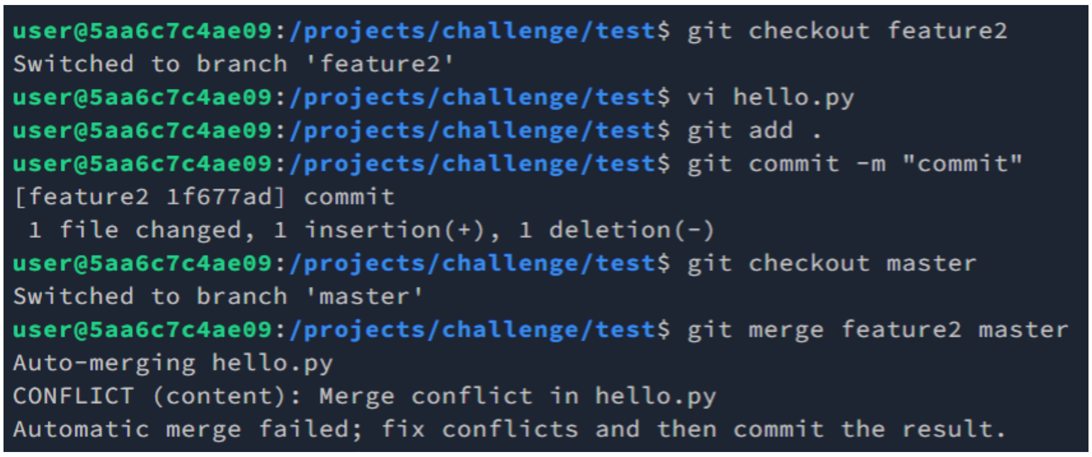
Step 5: The branch is merged in the master. Delete all the feature branches to keep your repository clean and understandable.



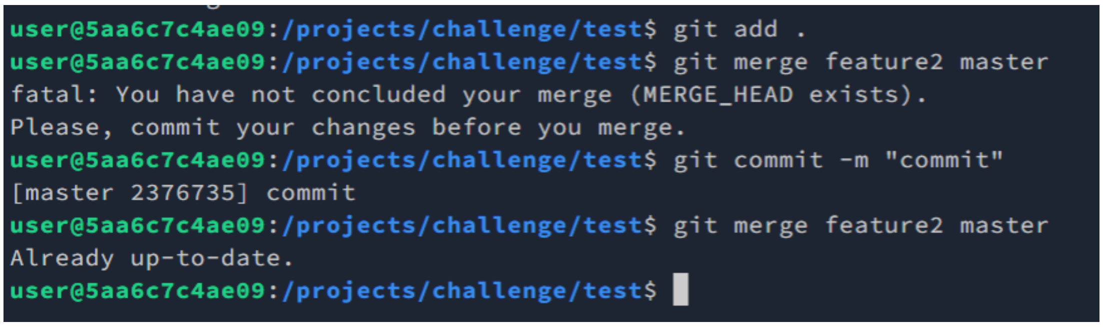
```
user@5aa6c7c4ae09:/projects/challenge/test$ git checkout -b feature1
Switched to a new branch 'feature1'
user@5aa6c7c4ae09:/projects/challenge/test$ git checkout -b feature2
Switched to a new branch 'feature2'
user@5aa6c7c4ae09:/projects/challenge/test$ git checkout feature1
Switched to branch 'feature1'
user@5aa6c7c4ae09:/projects/challenge/test$ vi hello.py
user@5aa6c7c4ae09:/projects/challenge/test$ git add .
user@5aa6c7c4ae09:/projects/challenge/test$ git commit -m "commit"
[feature1 5bef7a7] commit
 1 file changed, 1 insertion(+), 1 deletion(-)
user@5aa6c7c4ae09:/projects/challenge/test$
```



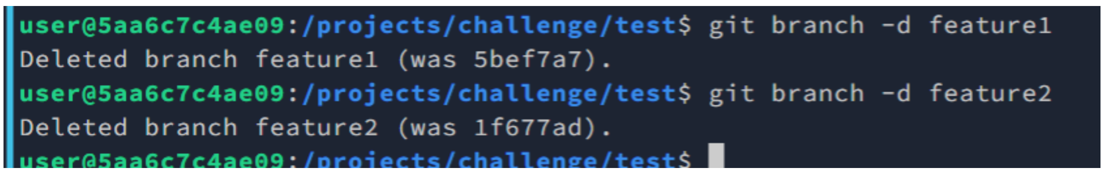
```
user@5aa6c7c4ae09:/projects/challenge/test$ git checkout master
Switched to branch 'master'
user@5aa6c7c4ae09:/projects/challenge/test$ git merge feature1 master
Updating 11504b0..5bef7a7
Fast-forward
 hello.py | 2 +-
 1 file changed, 1 insertion(+), 1 deletion(-)
```



```
user@5aa6c7c4ae09:/projects/challenge/test$ git checkout feature2
Switched to branch 'feature2'
user@5aa6c7c4ae09:/projects/challenge/test$ vi hello.py
user@5aa6c7c4ae09:/projects/challenge/test$ git add .
user@5aa6c7c4ae09:/projects/challenge/test$ git commit -m "commit"
[feature2 1f677ad] commit
 1 file changed, 1 insertion(+), 1 deletion(-)
user@5aa6c7c4ae09:/projects/challenge/test$ git checkout master
Switched to branch 'master'
user@5aa6c7c4ae09:/projects/challenge/test$ git merge feature2 master
Auto-merging hello.py
CONFLICT (content): Merge conflict in hello.py
Automatic merge failed; fix conflicts and then commit the result.
```



```
user@5aa6c7c4ae09:/projects/challenge/test$ git add .
user@5aa6c7c4ae09:/projects/challenge/test$ git merge feature2 master
fatal: You have not concluded your merge (MERGE_HEAD exists).
Please, commit your changes before you merge.
user@5aa6c7c4ae09:/projects/challenge/test$ git commit -m "commit"
[master 2376735] commit
user@5aa6c7c4ae09:/projects/challenge/test$ git merge feature2 master
Already up-to-date.
user@5aa6c7c4ae09:/projects/challenge/test$
```



```
user@5aa6c7c4ae09:/projects/challenge/test$ git branch -d feature1
Deleted branch feature1 (was 5bef7a7).
user@5aa6c7c4ae09:/projects/challenge/test$ git branch -d feature2
Deleted branch feature2 (was 1f677ad).
user@5aa6c7c4ae09:/projects/challenge/test$
```