LAB MST-1

1. Basic local workflow: Part – 1

Related – Git basic local workflow

Main Task

1. Create a new directory and change into it.

2. Use the init command to create a Git repository in that directory.

3. Observe that there is now a .git directory.

4. Create a README file.

5. Look at the output of the status command; the README you created should appear as an untracked file.

6. Use the add command to add the new file to the staging area. Again, look at the output of the status command.

7. Now use the commit command to commit the contents of the staging area.

8. Create a src directory and add a couple of files to it.

9. Use the add command, but name the directory, not the individual files. Use the status command. See how both files have been staged. Commit them.

10. Make a change to one of the files. Use the diff command to view the details of the change.

11. Next, add the changed file, and notice how it moves to the staging area in the status output. Also observe that the diff command you did before using add now gives no output. Why not? What do you have to do to see a diff of the things in the staging area? (Hint: review the slides if you can't remember.)

12. Now without committing, make another change to the same file you changed in step 10. Look at the status output, and the diff output. Notice how you can have both staged and unstaged changes, even when you are talking about a single file. Observe the difference when you use the add command to stage the latest round of changes. Finally, commit them. You should now have started to get a feel for the staging area.

13. Use the log command in order to see all of the commits you made so far.

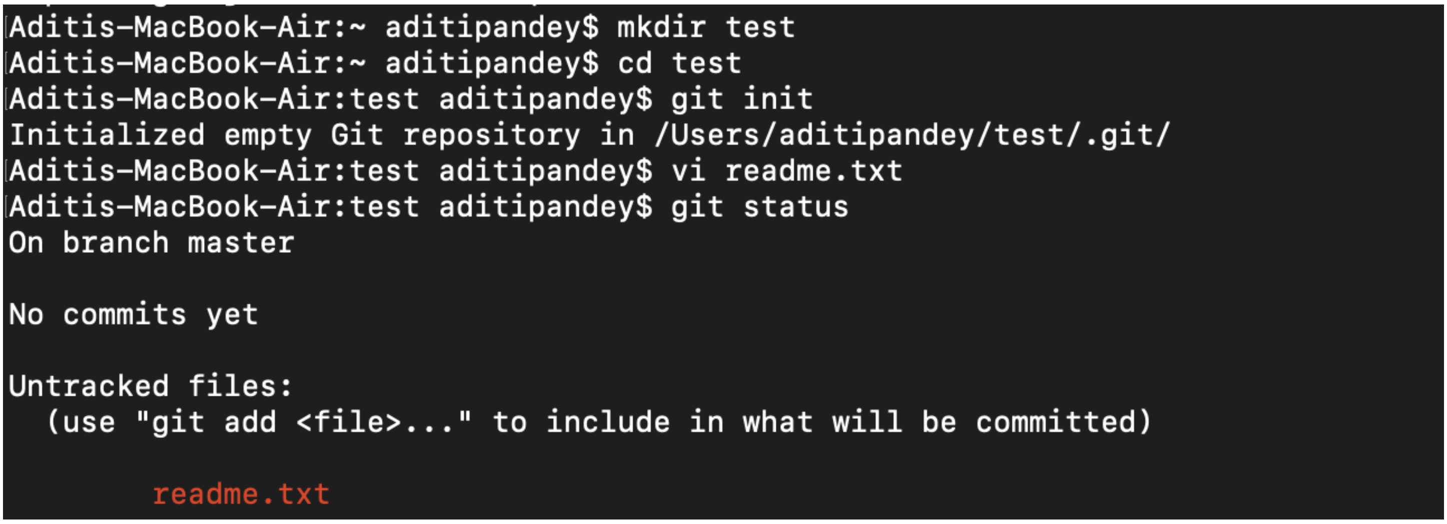
14. Use the show command to look at an individual commit. How many characters of the

commit identifier can you get away with typing at a minimum?

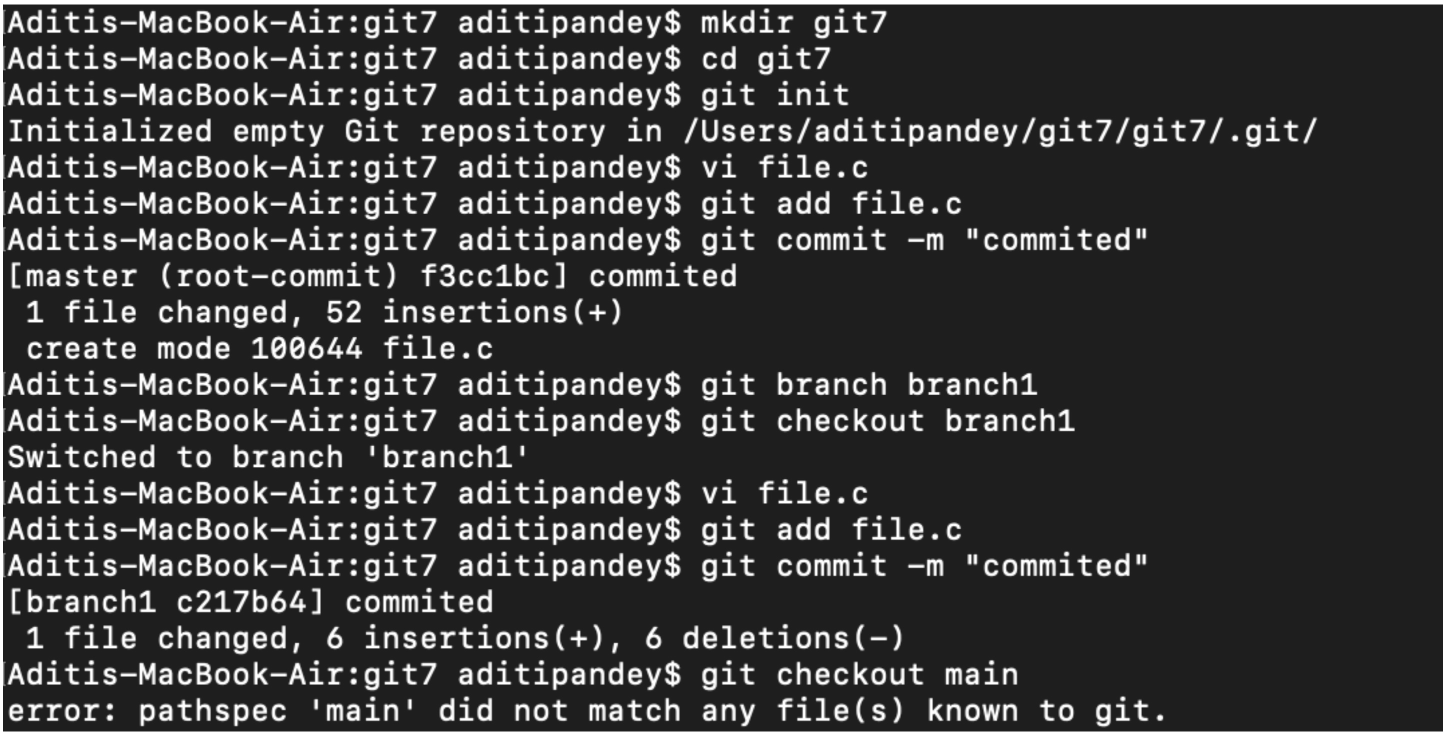
15. Make a couple more commits, at least one of which should add an extra file.

STEPS-

Here we make the file but we donot staged which showes the file untracked

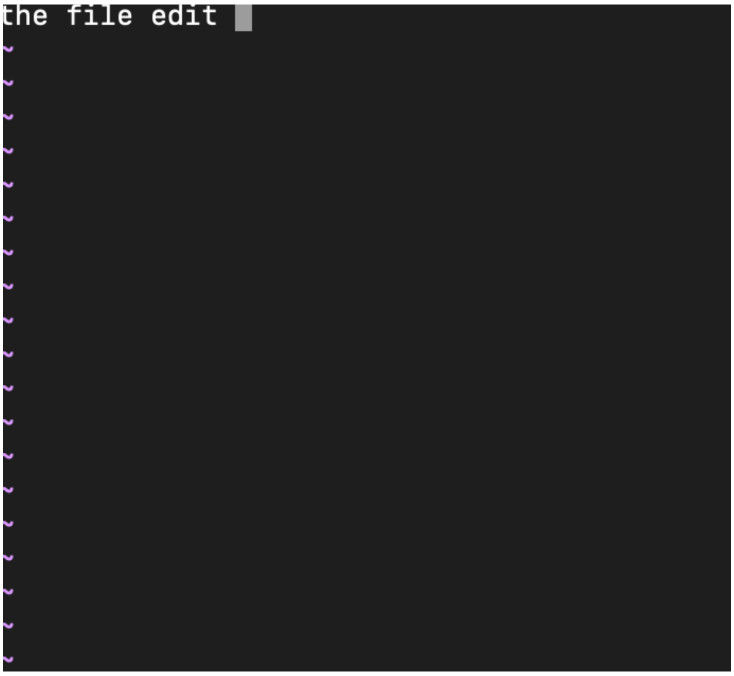


```
Aditis-MacBook-Air:~ aditipandey$ mkdir test
Aditis-MacBook-Air:~ aditipandey$ cd test
Aditis-MacBook-Air:test aditipandey$ git init
Initialized empty Git repository in /Users/aditipandey/test/.git/
Aditis-MacBook-Air:test aditipandey$ vi readme.txt
Aditis-MacBook-Air:test aditipandey$ git status
On branch master

No commits yet

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

        readme.txt
```



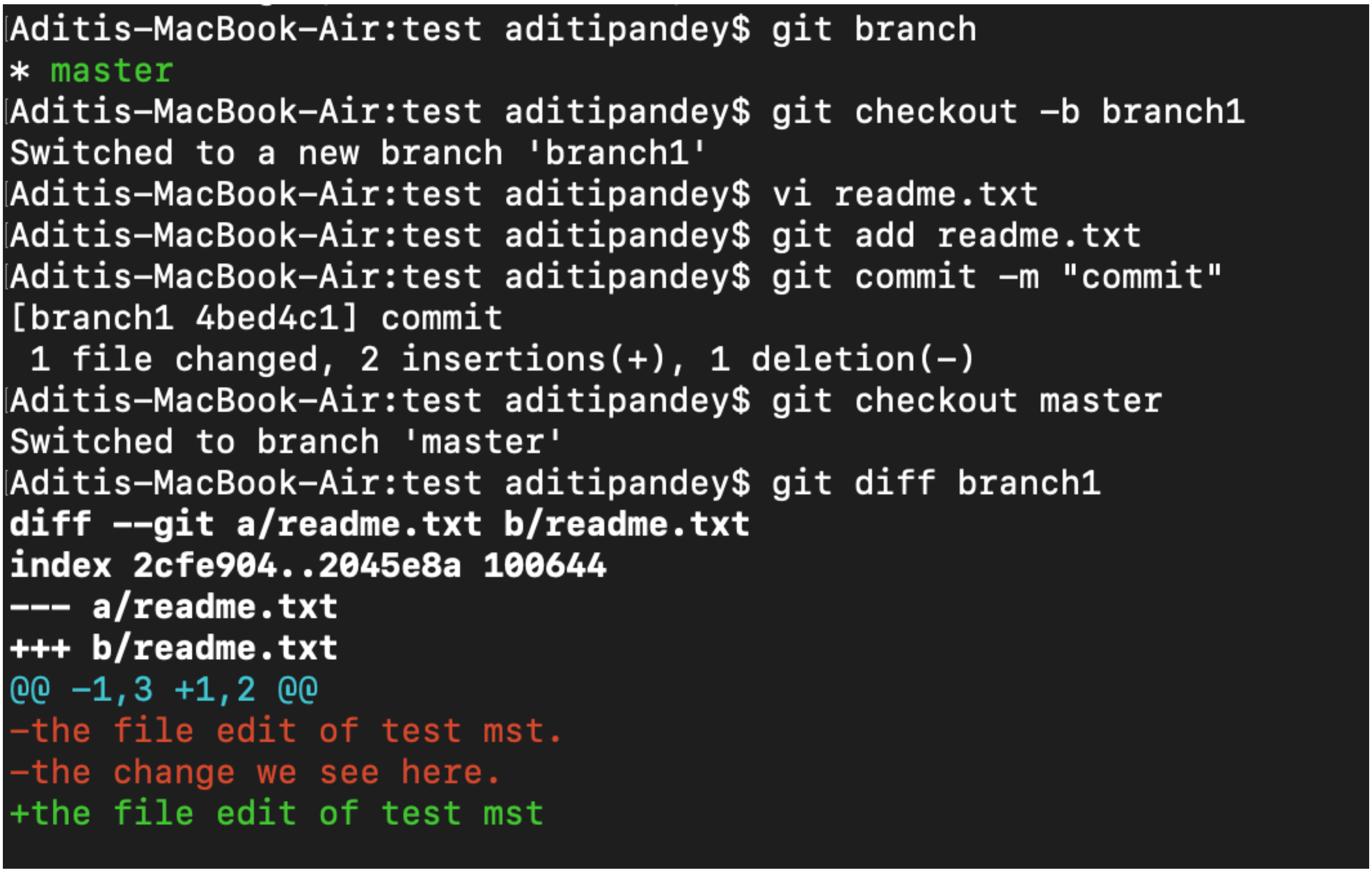
```
Aditis-MacBook-Air:git7 aditipandey$ mkdir git7
Aditis-MacBook-Air:git7 aditipandey$ cd git7
Aditis-MacBook-Air:git7 aditipandey$ git init
Initialized empty Git repository in /Users/aditipandey/git7/git7/.git/
Aditis-MacBook-Air:git7 aditipandey$ vi file.c
Aditis-MacBook-Air:git7 aditipandey$ git add file.c
Aditis-MacBook-Air:git7 aditipandey$ git commit -m "commited"
[master (root-commit) f3cc1bc] commited
 1 file changed, 52 insertions(+)
 create mode 100644 file.c
Aditis-MacBook-Air:git7 aditipandey$ git branch branch1
Aditis-MacBook-Air:git7 aditipandey$ git checkout branch1
Switched to branch 'branch1'
Aditis-MacBook-Air:git7 aditipandey$ vi file.c
Aditis-MacBook-Air:git7 aditipandey$ git add file.c
Aditis-MacBook-Air:git7 aditipandey$ git commit -m "commited"
[branch1 c217b64] commited
 1 file changed, 6 insertions(+), 6 deletions(-)
Aditis-MacBook-Air:git7 aditipandey$ git checkout main
error: pathspec 'main' did not match any file(s) known to git.
```

Use the add command, but name the directory, not the individual files. Use the status command. See how both files have been staged. Commit them.

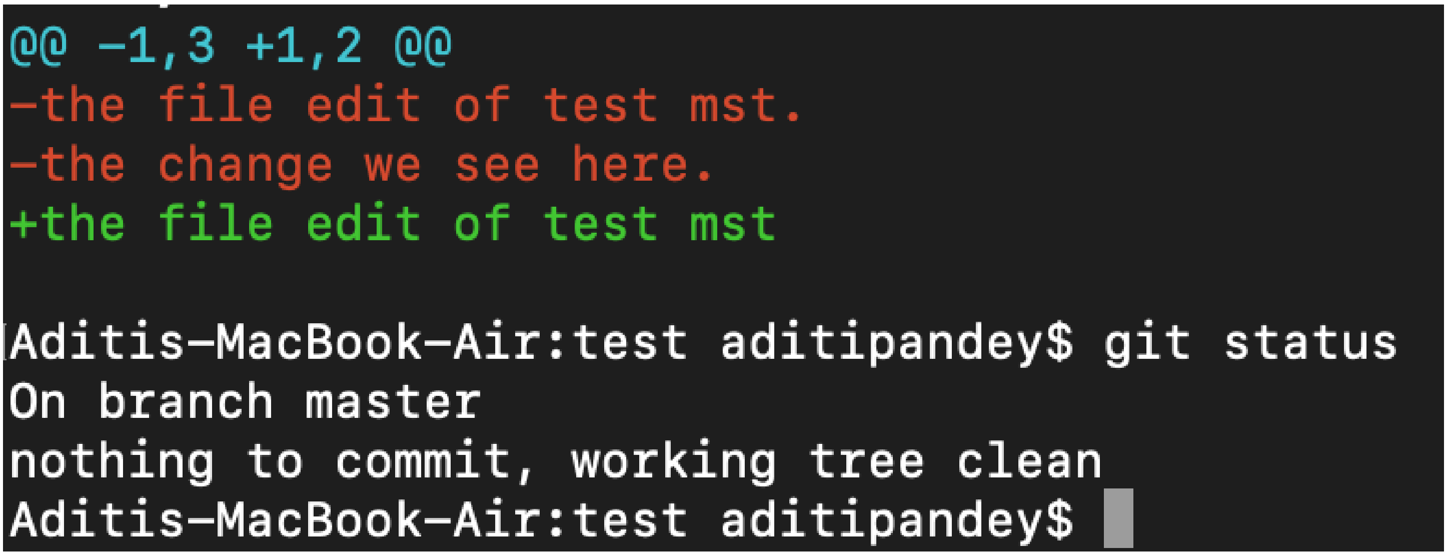


```
the file edit 
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
```

git diff to see the difference between both branches in master branch and branch1

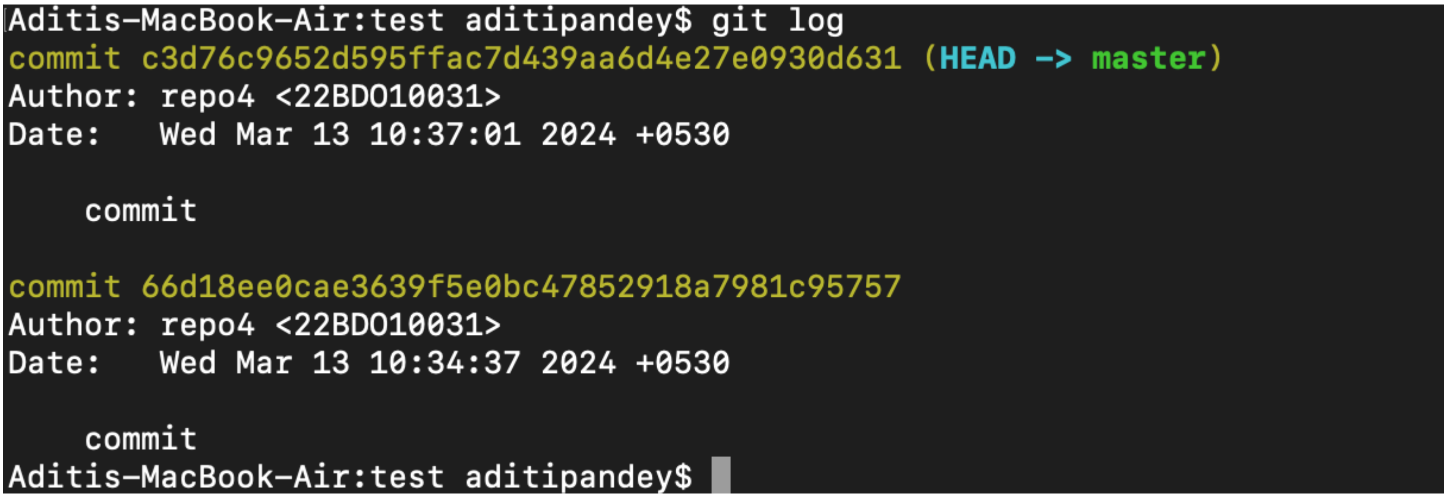


```
[Aditis-MacBook-Air:test aditipandey$ git branch
* master
[Aditis-MacBook-Air:test aditipandey$ git checkout -b branch1
Switched to a new branch 'branch1'
[Aditis-MacBook-Air:test aditipandey$ vi readme.txt
[Aditis-MacBook-Air:test aditipandey$ git add readme.txt
[Aditis-MacBook-Air:test aditipandey$ git commit -m "commit"
[branch1 4bed4c1] commit
 1 file changed, 2 insertions(+), 1 deletion(-)
[Aditis-MacBook-Air:test aditipandey$ git checkout master
Switched to branch 'master'
[Aditis-MacBook-Air:test aditipandey$ git diff branch1
diff --git a/readme.txt b/readme.txt
index 2cfe904..2045e8a 100644
--- a/readme.txt
+++ b/readme.txt
@@ -1,3 +1,2 @@
-the file edit of test mst.
-the change we see here.
+the file edit of test mst
```



```
@@ -1,3 +1,2 @@
-the file edit of test mst.
-the change we see here.
+the file edit of test mst

Aditis-MacBook-Air:test aditipandey$ git status
On branch master
nothing to commit, working tree clean
Aditis-MacBook-Air:test aditipandey$ 
```

git status command to see the status .



```
Aditis-MacBook-Air:test aditipandey$ git log
commit c3d76c9652d595ffac7d439aa6d4e27e0930d631 (HEAD -> master)
Author: repo4 <22BD010031>
Date:   Wed Mar 13 10:37:01 2024 +0530

    commit

commit 66d18ee0cae3639f5e0bc47852918a7981c95757
Author: repo4 <22BD010031>
Date:   Wed Mar 13 10:34:37 2024 +0530

    commit
Aditis-MacBook-Air:test aditipandey$ 
```

git log command is used show all the changes we have done .it showes all the commit history.