ASHRAF NOMANI

GIT AND GITHUB ASSIGNMENT

Install GIT & make sure it is added into PATH.

Section 0 –

Use GIT as local VCS. Steps to follow:

1. Create a directory ‘project\_dir’ & cd to ‘project\_dir’.

ASHRAFNOMANI@ASHRAF\_NOMANI MINGW64 ~/Desktop/git

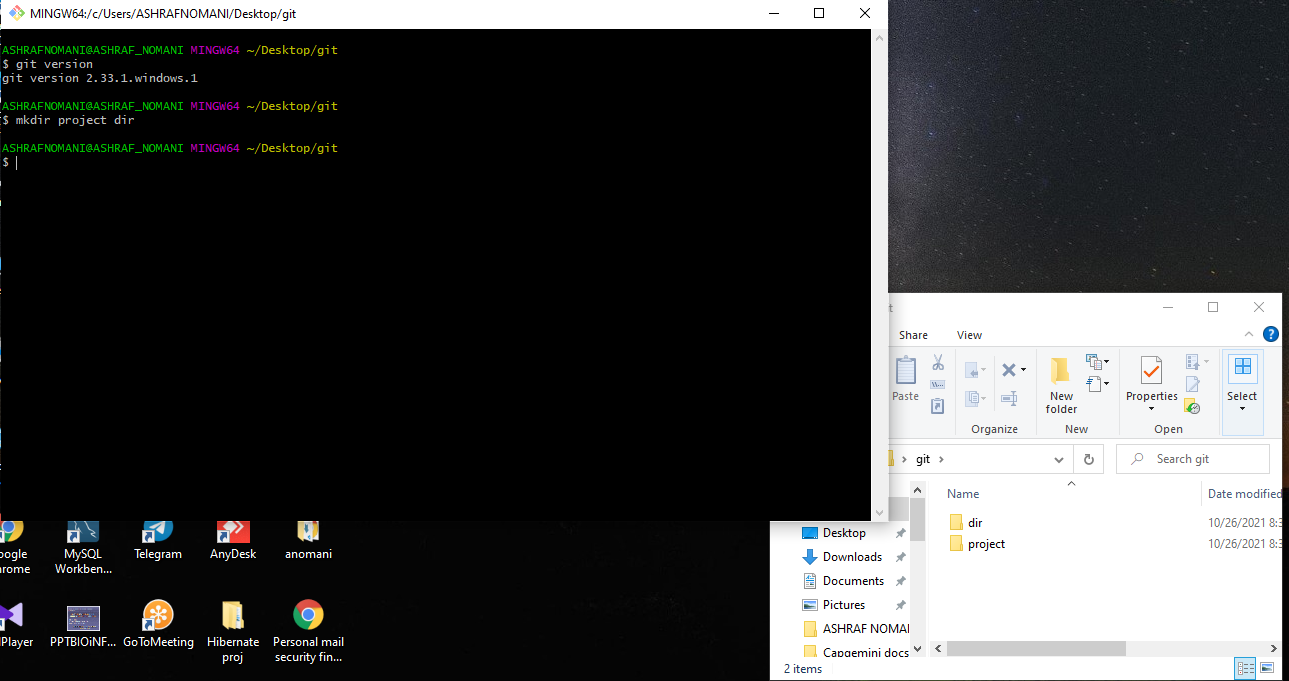
$ git version

git version 2.33.1.windows.1

ASHRAFNOMANI@ASHRAF\_NOMANI MINGW64 ~/Desktop/git

$ mkdir project dir

ASHRAFNOMANI@ASHRAF\_NOMANI MINGW64 ~/Desktop/git



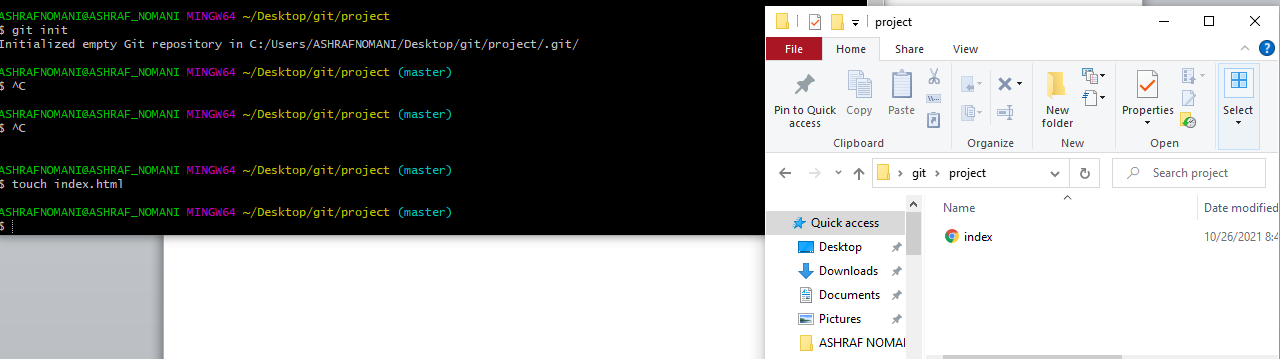
2.Initialize git version database. (git init)

ASHRAFNOMANI@ASHRAF\_NOMANI MINGW64 ~/Desktop/git/project

$ git init

Initialized empty Git repository in C:/Users/ASHRAFNOMANI/Desktop/git/project/.git/

3.Create a new file index.html.



4.Check the git status. You should find index.html as untracked file.

$ git status

On branch master

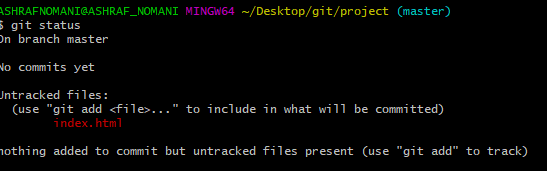
No commits yet

Untracked files:

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

index.html

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



5.Stage the index.html file.

$ git add index.html

ASHRAFNOMANI@ASHRAF\_NOMANI MINGW64 ~/Desktop/git/project (master)

$ git status

On branch master

No commits yet

Changes to be committed:

(use "git rm --cached <file>..." to unstage)

new file: index.html // green color shows file in staging area

6.Commit index.html

ASHRAFNOMANI@ASHRAF\_NOMANI MINGW64 ~/Desktop/git/project (master)

$ git commit -m 'My first commit'

[master (root-commit) 406b59f] My first commit

1 file changed, 0 insertions(+), 0 deletions(-)

create mode 100644 index.html

7.Make few changes in index.html & create a new file info.txt file

$ git status

On branch master

Changes not staged for commit:

(use "git add <file>..." to update what will be committed)

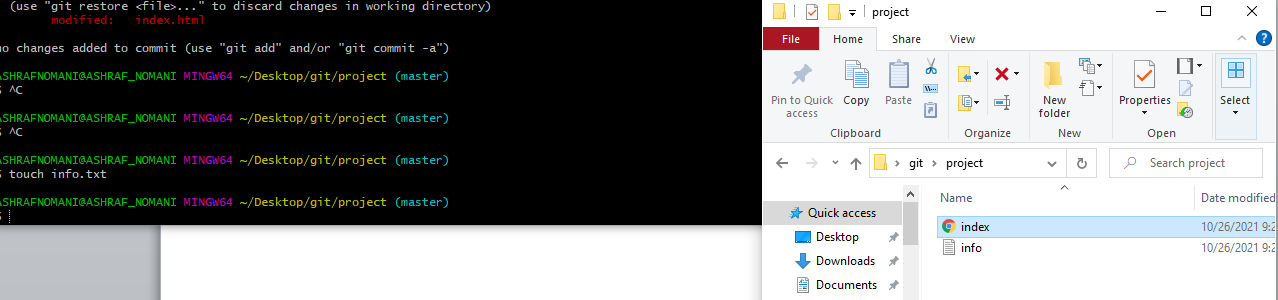
(use "git restore <file>..." to discard changes in working directory)

modified: index.html

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

ASHRAFNOMANI@ASHRAF\_NOMANI MINGW64 ~/Desktop/git/project (master)

$ touch info.txt



8.Check git status. You should find index.html & info.txt as untracked files.

$ 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: index.html

Untracked files:

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

info.txt

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

9.Configure GIT to ignore all txt files.

$ touch .gitignore

ASHRAFNOMANI@ASHRAF\_NOMANI MINGW64 ~/Desktop/git/project (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: index.html

Untracked files:

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

.gitignore

info.txt

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

10.Again check the git status. You should find only index.html as untracked file.

$ 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: index.html

Untracked files:

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

.gitignore

info.txt

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

11 .State & commit index.html

// before commit we have to add file in staging area otherwise we are unable to commit

$ git add index.html

ASHRAFNOMANI@ASHRAF\_NOMANI MINGW64 ~/Desktop/git/project (master)

$ git add info.txt

ASHRAFNOMANI@ASHRAF\_NOMANI MINGW64 ~/Desktop/git/project (master)

$ git status

On branch master

Changes to be committed:

(use "git restore --staged <file>..." to unstage)

modified: index.html

new file: info.txt

Untracked files:

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

.gitignore

ASHRAFNOMANI@ASHRAF\_NOMANI MINGW64 ~/Desktop/git/project (master)

$ git commit -m 'second commit in html file for edit html'

[master ce65b4f] second commit in html file for edit html

2 files changed, 12 insertions(+)

create mode 100644 info.txt

12. Log all your comments so far.

$ git log

commit ce65b4f5350ed4c4a79cadc58bf1c9181bd99101 (HEAD -> master)

Author: Ashraf Nomani <57210441+anomani3@users.noreply.github.com>

Date: Tue Oct 26 09:42:05 2021 +0530

second commit in html file for edit html

commit 406b59f8eaec8b0a13465de4ac3c2359f03b286b

Author: Ashraf Nomani <57210441+anomani3@users.noreply.github.com>

Date: Tue Oct 26 08:59:15 2021 +0530

My first commit

13. Make some changes in index.html.

git add index.html

ASHRAFNOMANI@ASHRAF\_NOMANI MINGW64 ~/Desktop/git/project (master)

$ git status

On branch master

Changes to be committed:

(use "git restore --staged <file>..." to unstage)

modified: index.html

Untracked files:

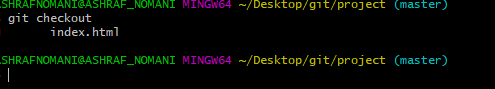
(use "git add <file>..." to include in what will be committed)

.gitignore

14.Revert the change made in the previous step using git command.

$ git checkout

M index.html



15. Again change index.html.

$ git status

On branch master

Changes to be committed:

(use "git restore --staged <file>..." to unstage)

modified: index.html

Changes not staged for commit:

(use "git add <file>..." to update what will be committed)

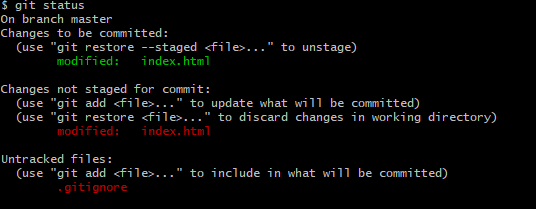
(use "git restore <file>..." to discard changes in working directory)

modified: index.html

Untracked files:

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

.gitignore



16. Stage index.html

$ git add .

ASHRAFNOMANI@ASHRAF\_NOMANI MINGW64 ~/Desktop/git/project (master)

$ git status

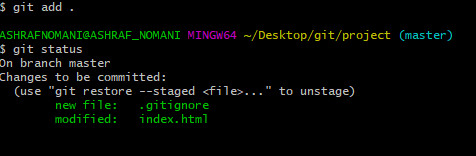
On branch master

Changes to be committed:

(use "git restore --staged <file>..." to unstage)

new file: .gitignore

modified: index.html



17. Revert back the last stage.

ASHRAFNOMANI@ASHRAF\_NOMANI MINGW64 ~/Desktop/git/project (master)

$ git checkout index.html

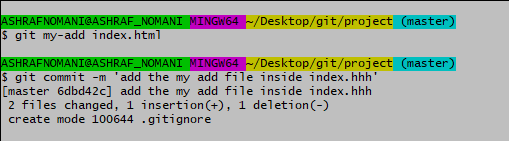
Updated 0 paths from the index



18.Rename ‘add’ command to ‘my-add’.

$ git config --global alias.my-add add

19.Using my\_add command Stage index.html again & commit the changes.



20. Revert the last commit

// I will do later

*GIT Branching*

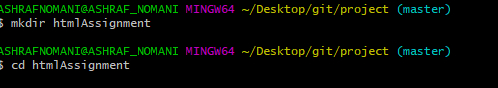
Objective: Commit HTML, CSS & JavaScript assignments into GIT.

SECTION-1 (HTML assignments) - Steps to follow:

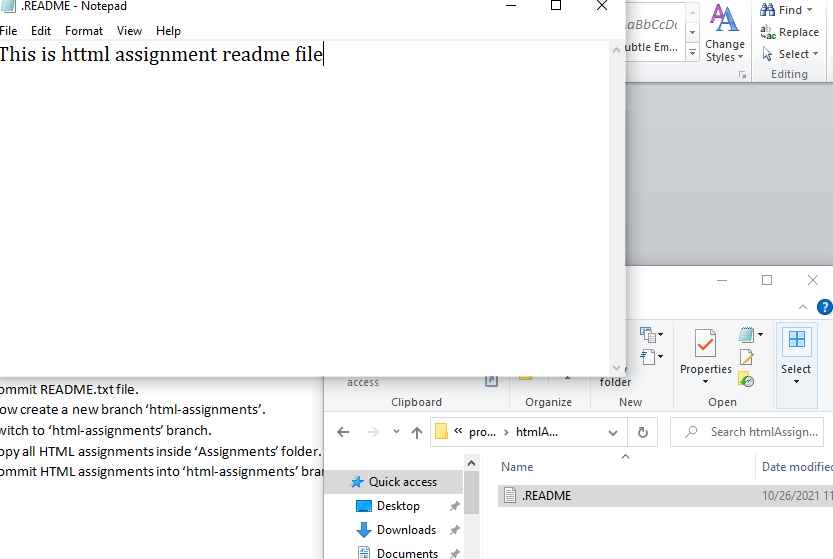
21.First take a backup of your assignments & projects. This is required because due to incorrect GIT operation you may lose your files.

I already took back up

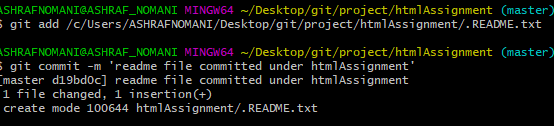
22. Create an empty directory ‘Assignments’ & cd to ‘Assignments’.



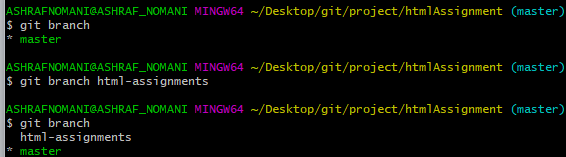
1. Create a file README.txt inside ‘Assignments’ & write few lines about the contents of ‘Assignments’ folder.



1. Commit README.txt file.



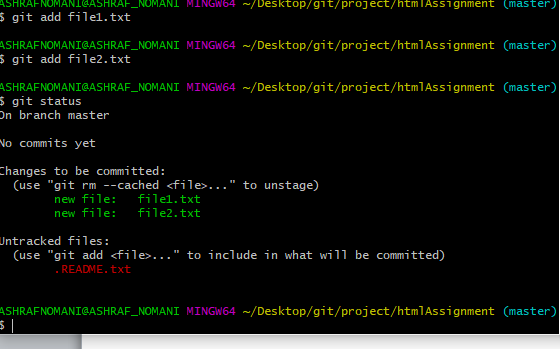
25.Now create a new branch ‘html-assignments’.



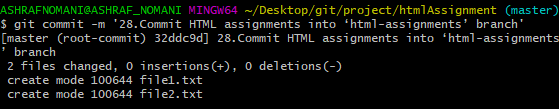
1. Switch to ‘html-assignments’ branch.



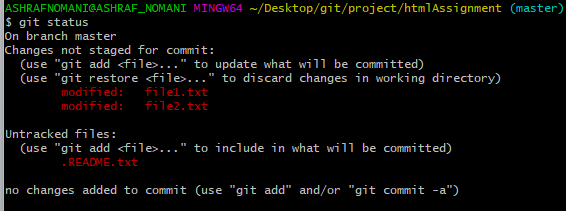
1. Copy all HTML assignments inside ‘Assignments’ folder.



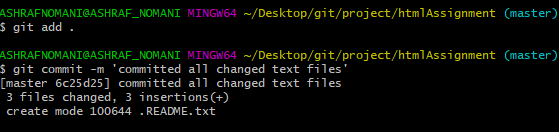
1. Commit HTML assignments into ‘html-assignments’ branch.



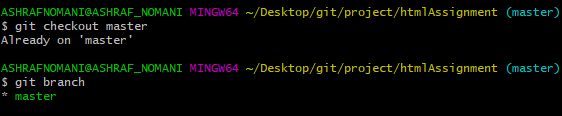
29.Make minor changes into few files belonging to ‘html-assignments’ branch.



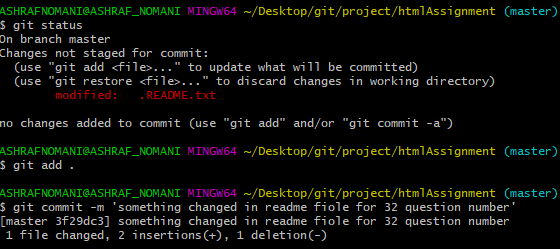
1. Commit those changed files.



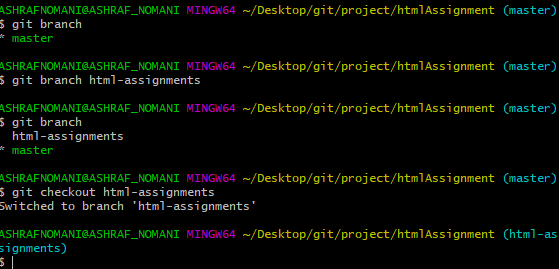
1. Switch to master branch.



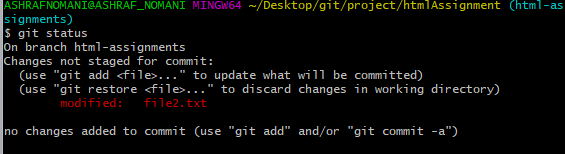
1. Make minor changes into README.txt file & commit those changes into master.



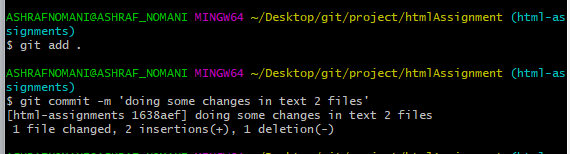
1. Again switch to ‘html-assignments’ branch.



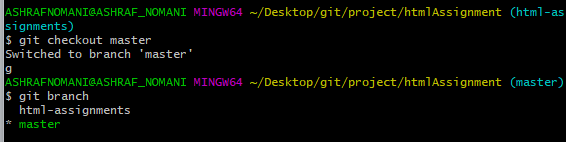
34.Make minor changes into few files belonging to ‘html-assignments’ branch.



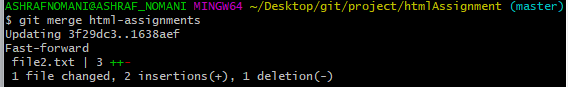
35.Commit those changes.



36. Switch to master.



37. Merge ‘html-assignments’ branch into master. Confirm all html assignments are shown in master.

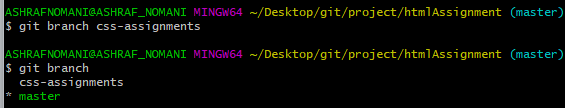


38. Finally delete the ‘html-assignments’ branch.

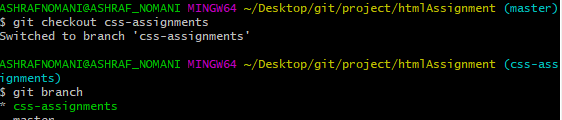


SECTION-2 - (CSS assignments)

1. Create a new branch ‘css-assignments’.



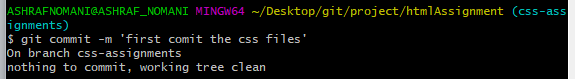
1. Switch to ‘css-assignments’ branch.



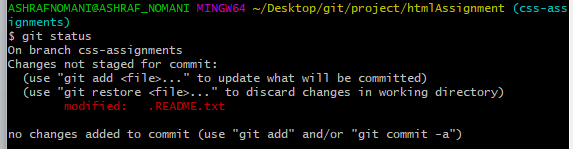
1. Copy all CSS assignments inside ‘Assignments’ folder.



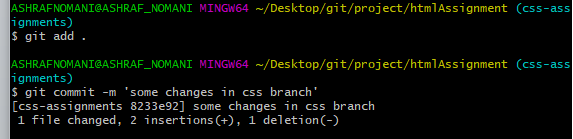
1. Commit CSS assignments into ‘css-assignments’ branch.



1. Make minor changes into README.txt file on line 1 belonging to ‘css-assignments’ branch.



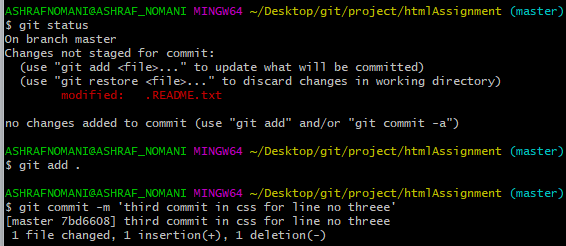
1. Commit those changed files.



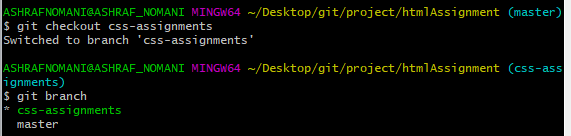
1. Switch to master branch.



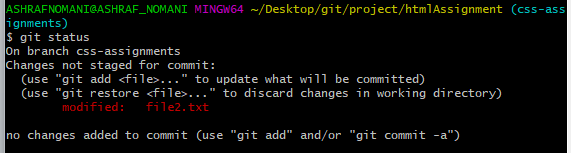
1. Make minor changes into README.txt file on line 3 & commit those changes into master.



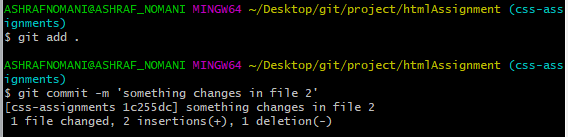
1. Again switch to ‘css-assignments’ branch.



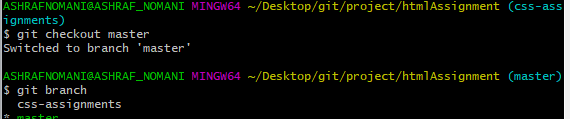
1. Make minor changes into few files belonging to ‘css-assignments’ branch.



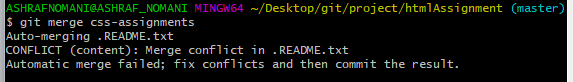
1. Commit those changes.

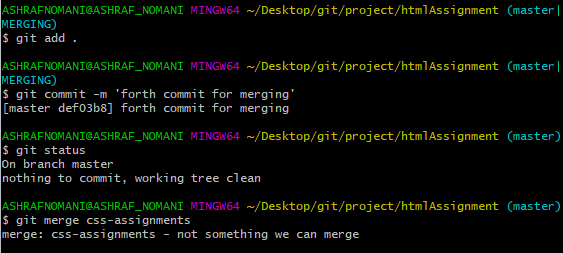


1. Switch to master.



1. Merge ‘css-assignments’ branch into master. Confirm all css assignments are shown in master.





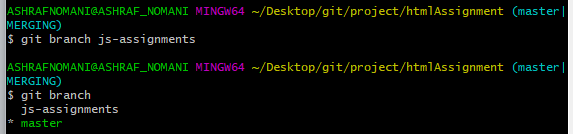
1. Finally delete the ‘css-assignments’ branch.



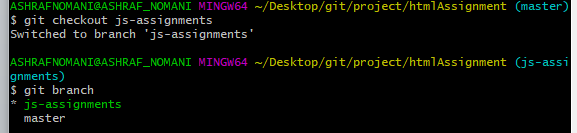
SECTION-3 –

(JavaScript assignments):

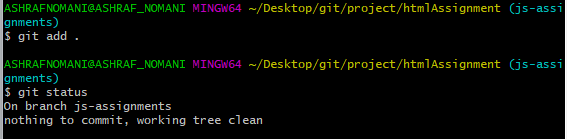
1. Create a new branch ‘js-assignments’.



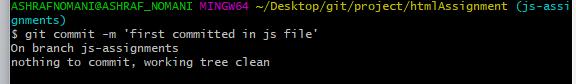
1. Switch to ‘js-assignments’ branch.



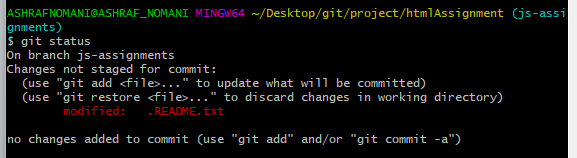
1. Copy all JavaScript assignments inside ‘Assignments’ folder.



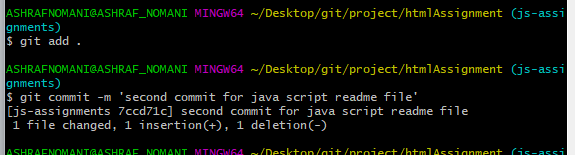
1. Commit JavaScript assignments into ‘js-assignments’ branch.



1. Make minor changes into README.txt file on line 1 belonging to ‘js-assignments’ branch.



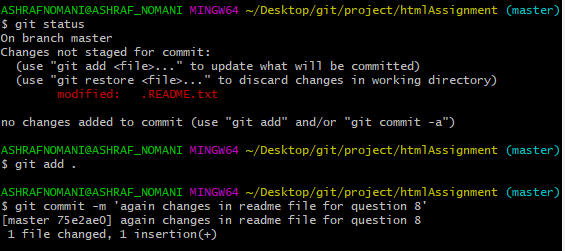
1. Commit those changed files.



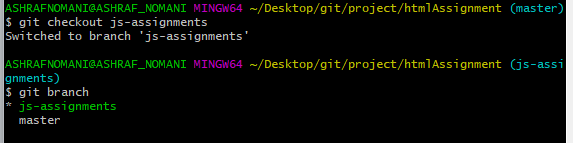
1. Switch to master branch.



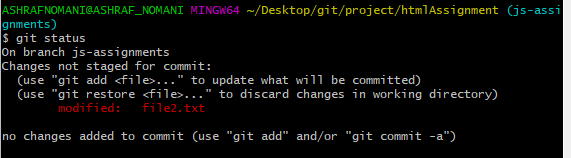
1. Make minor changes into README.txt file on line 1 & commit those changes into master.



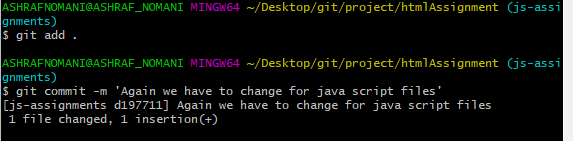
1. Again switch to ‘js-assignments’ branch.



1. Make minor changes into few files belonging to ‘js-assignments’ branch.



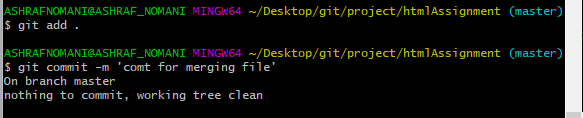
1. Commit those changes.



1. Switch to master.



1. Merge ‘js-assignments’ branch into master. Confirm all JavaScript assignments are shown in master.



1. Finally delete the ‘js-assignments’ branch.

