**Install GIT & make sure it is added into PATH.**

**Use GIT as local VCS. Steps to follow:**

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

* Open git bash and make project\_dir using mkdir cmd

Text

Description automatically generated

2.Initialize git version database.

* Enter git init command creates a new Git repository

Text

Description automatically generated

3. Create a new file index.html.

* Using touch cmd I had created index.html

Text

Description automatically generated

Content added in index.html

Text

Description automatically generated

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

* Git statuts used for to check current status of directory

Text

Description automatically generated

5. Stage the index.html file.

* git add cmd used for to add staging area

Text

Description automatically generated

6. Commit index.html

* by using git commit cmd I had committed index.html

Text

Description automatically generated

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

* touch info.txt are used for creating file and edited some content indext.html

Text

Description automatically generated

Text

Description automatically generated

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

* Using git status you can see status of directory

Text

Description automatically generated

9. Configure GIT to ignore all txt files.

* Create .gitignore file and add files that you need to hide

Text

Description automatically generated

Graphical user interface, text

Description automatically generated with medium confidence

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

* Create .gitignore file and add files that you need to hide

Text

Description automatically generated

11. State & commit index.html

* Git commit -am cmd is used for add and commit in single cmd

Text

Description automatically generated

12. Log all your comments so far.

* git log used for to see commit history

Text

Description automatically generated

13. Make some changes in index.html.

* made some changes in index.html

Text

Description automatically generated

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

* Git check – index.html used for undoing recent change

Text

Description automatically generated

* Here you can see recently added line was removed

Text

Description automatically generated

15. Again change index.html.

* Again Capgemini training was added

Text

Description automatically generated

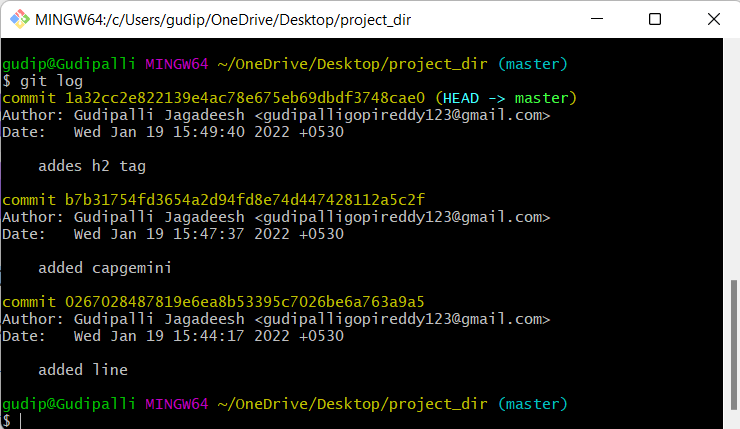
16. Stage index.html

* git add used to staging area

Text

Description automatically generated

17. Revert back the last stage.



18. Rename ‘add’ command to ‘my-add’.

* To rename any command we have use alias names

A screenshot of a computer

Description automatically generated with medium confidence

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

Text

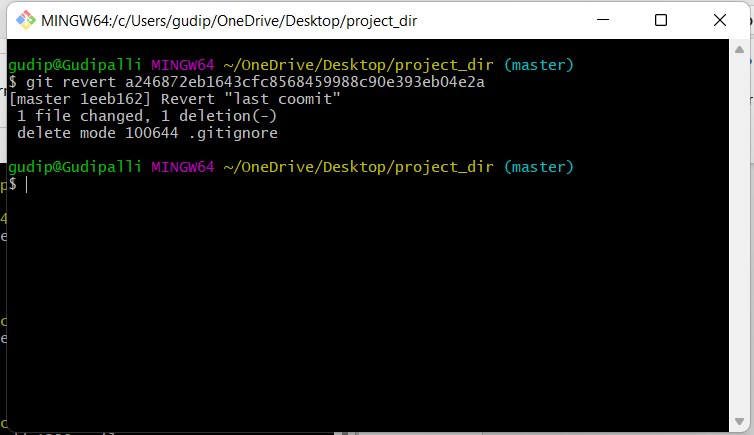
Description automatically generated

20. Revert the last commit.

* To revert last stage copy commit id

Text

Description automatically generated



Text

Description automatically generated

**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 had taken backup my assignment folder into google drive

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

A screenshot of a computer

Description automatically generated with medium confidence

23. Create a file README.txt inside ‘Assignments’ & write few lines about the

contents of ‘Assignments’ folder.

Text

Description automatically generated

Graphical user interface, text, application

Description automatically generated

24. Commit README.txt file.

A screenshot of a computer

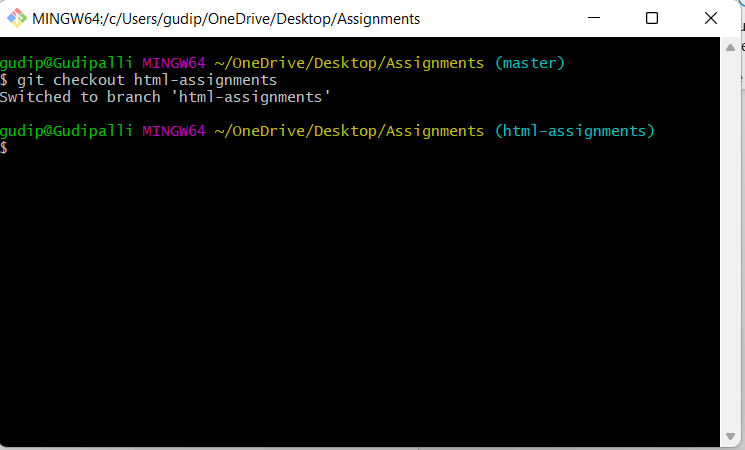
Description automatically generated with medium confidence

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

Text

Description automatically generated

26. Switch to ‘html-assignments’ branch.



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

Shape

Description automatically generated with medium confidence

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

Text

Description automatically generated

Text

Description automatically generated

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

Text

Description automatically generated

30. Commit those changed files.

Text

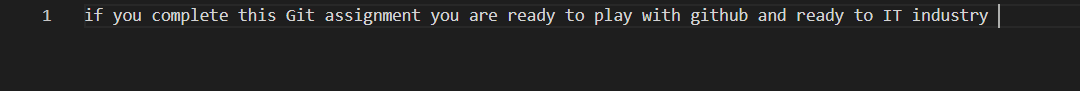
Description automatically generated

31. Switch to master branch.

Text

Description automatically generated

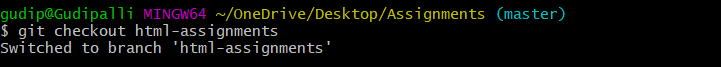
32. Make minor changes into README.txt file & commit those changes into master.



Text

Description automatically generated

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



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

Text

Description automatically generated

35. Commit those changes.

Text

Description automatically generated

36. Switch to master.

Text

Description automatically generated

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

Text

Description automatically generated

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

Text

Description automatically generated

**SECTION-2 - (CSS assignments) Steps to follow:**

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

Text

Description automatically generated

2. Switch to ‘css-assignments’ branch.

Text

Description automatically generated

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

Text

Description automatically generated

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

Text

Description automatically generated

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

A screenshot of a computer

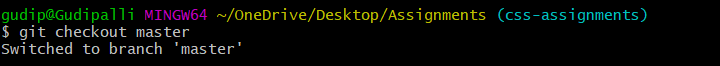
Description automatically generated with medium confidence

6. Commit those changed files.

Text

Description automatically generated

7. Switch to master branch.



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

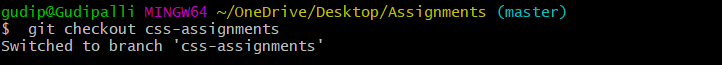
Text

Description automatically generated

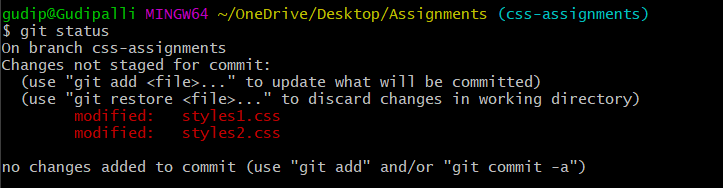
Text

Description automatically generated

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



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

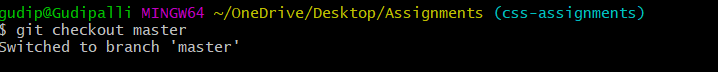


11. Commit those changes.

Text

Description automatically generated

12. Switch to master.



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

Text

Description automatically generated



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

Text

Description automatically generated

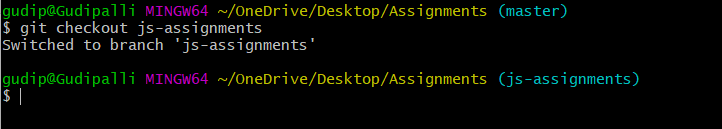
**SECTION-3 - (JavaScript assignments) Steps to follow:**

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

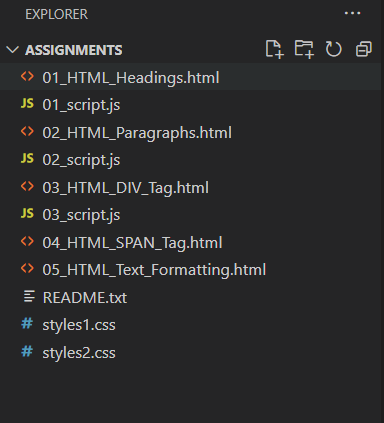
Text

Description automatically generated

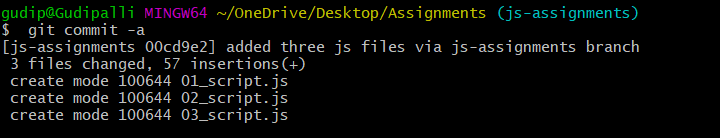
2. Switch to ‘js-assignments’ branch.



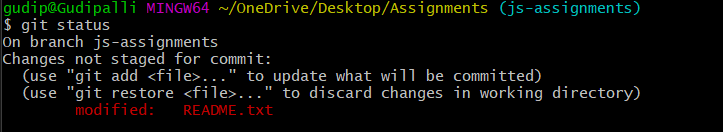
3. Copy all JavaScript assignments inside ‘Assignments’ folder.



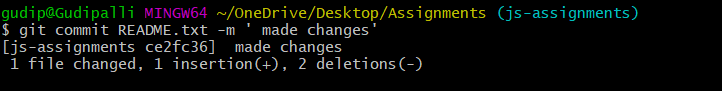
4. Commit JavaScript assignments into ‘js-assignments’ branch.



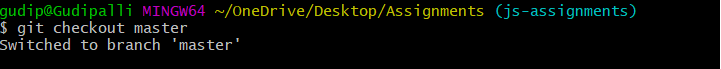
5. Make minor changes into README.txt file on line 1 belonging to ‘js-assignments’ branch.



6. Commit those changed files.

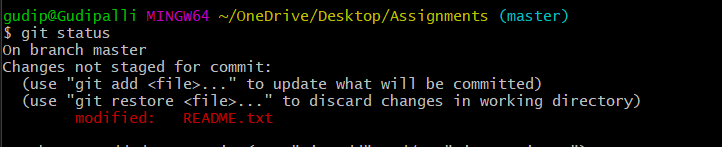


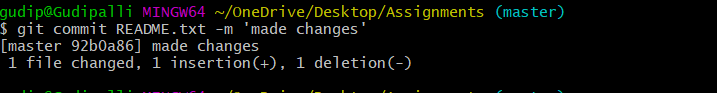
7. Switch to master branch.



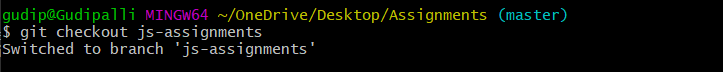
8. Make minor changes into README.txt file on line 1 & commit those

changes into master.

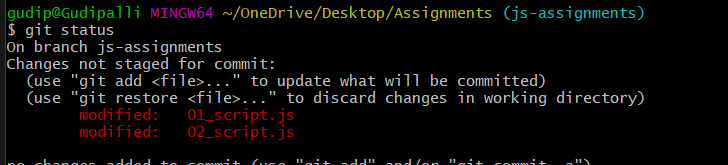




9. Again switch to ‘js-assignments’ branch.



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

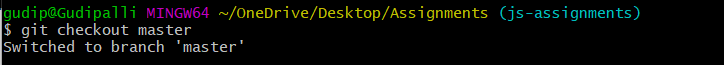


11. Commit those changes.

Text

Description automatically generated

12. Switch to master.



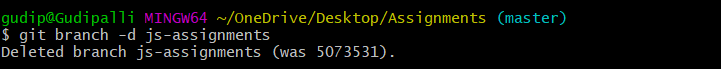
13. Merge ‘js-assignments’ branch into master. Confirm all JavaScript

assignments are shown in master.

Text

Description automatically generated

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



**GIT Remoting**

Objective: Pushing source code into GITHUB & collaborate team members.

**SECTION-1 (Pushing assignments to remote repository) - Steps to follow:**

39. Create a github account if you do not have already.

Graphical user interface, application, Word

Description automatically generated

40. Login on into github account.

Graphical user interface, application, website

Description automatically generated

41. Create new public repository ‘freshersbatch-oct16’.

Graphical user interface, text, application, email

Description automatically generated

42. Commit & push any sample file to this repository under ‘Assignments’ directory.

Text

Description automatically generated

Text

Description automatically generated

Graphical user interface, application, email

Description automatically generated

------------------------------------THE END------------------------------------------