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’.

Prince@prince MINGW64 ~/Desktop (master)

$ mkdir project\_dir

Prince@prince MINGW64 ~/Desktop (master)

$ cd project\_dir

2. Initialize git version database. (git init)

Prince@prince MINGW64 ~/Desktop/project\_dir (master)

$ git init

Initialized empty Git repository in C:/Users/Prince/Desktop/project\_dir/.git/

3. Create a new file index.html.

Prince@prince MINGW64 ~/Desktop/project\_dir (master)

$ touch index.html

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

Prince@prince MINGW64 ~/Desktop/project\_dir (master)

$ git status

starting fsmonitor-daemon in 'C:/Users/Prince/Desktop/project\_dir'

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.

Prince@prince MINGW64 ~/Desktop/project\_dir (master)

$ git add .

Prince@prince MINGW64 ~/Desktop/project\_dir (master)

$ git status

On branch master

No commits yet

Changes to be committed:

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

new file: index.html

6. Commit index.html

Prince@prince MINGW64 ~/Desktop/project\_dir (master)

$ git commit -m"index.html commit"

[master (root-commit) 556738c] index.html 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.

Prince@prince MINGW64 ~/Desktop/project\_dir (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

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

Prince@prince MINGW64 ~/Desktop/project\_dir (master)

$ touch info.txt

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

Prince@prince MINGW64 ~/Desktop/project\_dir (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)

info.txt

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

9. Configure GIT to ignore all txt files.

Prince@prince MINGW64 ~/Desktop/project\_dir (master)

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

$ touch .gitignore

Prince@prince MINGW64 ~/Desktop/project\_dir (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

11. State & commit index.html

Prince@prince MINGW64 ~/Desktop/project\_dir (master)

$ git add index.html

Prince@prince MINGW64 ~/Desktop/project\_dir (master)

$ git commit -m"2nd commit for index"

[master 42868e8] 2nd commit for index

1 file changed, 1 insertion(+)

create mode 100644 .gitignore

12. Log all your comments so far.

Prince@prince MINGW64 ~/Desktop/project\_dir (master)

$ git log

commit 42868e8e04104e1e2eef57aca8eedb46dc91e058 (HEAD -> master)

Author: prince <princeke00@gmail.com>

Date: Tue Oct 26 19:56:21 2021 +0530

2nd commit for index

commit 556738c72a9a391016739aae19549785192c3653

Author: prince <princeke00@gmail.com>

Date: Tue Oct 26 19:23:42 2021 +0530

index.html commit

13. Make some changes in index.html.

Prince@prince MINGW64 ~/Desktop/project\_dir (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

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

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

Prince@prince MINGW64 ~/Desktop/project\_dir (master)

$ git restore index.html

15. Again change index.html.

Prince@prince MINGW64 ~/Desktop/project\_dir (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

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

16. Stage index.html

Prince@prince MINGW64 ~/Desktop/project\_dir (master)

$ git add index.html

17. Revert back the last stage.

Prince@prince MINGW64 ~/Desktop/project\_dir (master)

$ git restore --staged index.html

18. Rename ‘add’ command to ‘my-add’.

Prince@prince MINGW64 ~/Desktop/project\_dir (master)

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

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

Prince@prince MINGW64 ~/Desktop/project\_dir (master)

$ git my-add .

Prince@prince MINGW64 ~/Desktop/project\_dir (master)

$ git commit -m"3rd commit"

[master 0342430] 3rd commit

1 file changed, 1 insertion(+)

20. Revert the last commit.

Prince@prince MINGW64 ~/Desktop/project\_dir (master)

$ git log

commit 0342430b3823162d159e355e7a00131912cc3876 (HEAD -> master)

Author: prince <princeke00@gmail.com>

Date: Tue Oct 26 20:05:32 2021 +0530

3rd commit

commit 42868e8e04104e1e2eef57aca8eedb46dc91e058

Author: prince <princeke00@gmail.com>

Date: Tue Oct 26 19:56:21 2021 +0530

2nd commit for index

commit 556738c72a9a391016739aae19549785192c3653

Author: prince <princeke00@gmail.com>

Date: Tue Oct 26 19:23:42 2021 +0530

index.html commit

Prince@prince MINGW64 ~/Desktop/project\_dir (master)

$ git revert 0342430b3823162d159e355e7a00131912cc3876

hint: Waiting for your editor to close the file...

[master 88b6da1] Revert "3rd commit"

1 file changed, 1 deletion(-)

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.

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

Prince@prince MINGW64 ~/Desktop (master)

$ mkdir Assignment

Prince@prince MINGW64 ~/Desktop (master)

$ cd Assignment

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

Prince@prince MINGW64 ~/Desktop/Assignment (master)

$ touch README.txt

24. Commit README.txt file.

Prince@prince MINGW64 ~/Desktop/Assignment (master)

$ git add .

Prince@prince MINGW64 ~/Desktop/Assignment (master)

$ git commit -m"commiting assignment"

[master (root-commit) dd34f50] commiting assignment

1 file changed, 1 insertion(+)

create mode 100644 Desktop/Assignment/README.txt

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

Prince@prince MINGW64 ~/Desktop/Assignment (master)

$ git branch html-assignment

26. Switch to ‘html-assignments’ branch.

Prince@prince MINGW64 ~/Desktop/Assignment (master)

$ git checkout html-assignment

Switched to branch 'html-assignment'

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

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

Prince@prince MINGW64 ~/Desktop/Assignment (html-assignment)

$ git commit -m"commit in assignment"

On branch html-assignment

nothing to commit, working tree clean

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

30. Commit those changed files.

Prince@prince MINGW64 ~/Desktop/Assignment (html-assignment)

$ git commit -m"another comit for assignment"

On branch html-assignment

nothing to commit, working tree clean

31. Switch to master branch.

Prince@prince MINGW64 ~/Desktop/Assignment (html-assignment)

$ git checkout master

Switched to branch 'master'

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

Prince@prince MINGW64 ~/Desktop/Assignment (master)

$ git add .

Prince@prince MINGW64 ~/Desktop/Assignment (master)

$ git commit -m" another commit"

On branch master

nothing to commit, working tree clean

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

Prince@prince MINGW64 ~/Desktop/Assignment (master)

$ git checkout html-assignment

Switched to branch 'html-assignment'

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

35. Commit those changes.

Prince@prince MINGW64 ~/Desktop/Assignment (html-assignment)

$ git add .

Prince@prince MINGW64 ~/Desktop/Assignment (master)

$ git commit -m" another commit"

On branch master

nothing to commit, working tree clean

36. Switch to master.

Prince@prince MINGW64 ~/Desktop/Assignment (html-assignment)

$ git checkout master

Switched to branch 'master'

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

Prince@prince MINGW64 ~/Desktop/Assignment (master)

$ git merge html-assignment

Already up to date.

Prince@prince MINGW64 ~/Desktop/Assignment (master)

$ ls

README.txt index.html info.txt

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

Prince@prince MINGW64 ~/Desktop/Assignment (master)

$ git branch -d html-assignment

Deleted branch html-assignment (was 88b6da1).

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

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

Prince@prince MINGW64 ~/Desktop/Assignment (master)

$ git branch css-assignment

2. Switch to ‘css-assignments’ branch.

Prince@prince MINGW64 ~/Desktop/Assignment (master)

$ git checkout css-assignment

Switched to branch 'css-assignment'

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

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

Prince@prince MINGW64 ~/Desktop/Assignment (css-assignment)

$ git add .

Prince@prince MINGW64 ~/Desktop/Assignment (css-assignment)

$ git commit -m"st css commit"

On branch css-assignment

nothing to commit, working tree clean

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

6. Commit those changed files.

Prince@prince MINGW64 ~/Desktop/Assignment (css-assignment)

$ git add .

Prince@prince MINGW64 ~/Desktop/Assignment (css-assignment)

$ git status

On branch css-assignment

nothing to commit, working tree clean

Prince@prince MINGW64 ~/Desktop/Assignment (css-assignment)

$ git commit -m"nd commit"

On branch css-assignment

nothing to commit, working tree clean

7. Switch to master branch.

Prince@prince MINGW64 ~/Desktop/Assignment (css-assignment)

$ git checkout master

Switched to branch 'master'

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

Prince@prince MINGW64 ~/Desktop/Assignment (master)

$ vi README.txt

Prince@prince MINGW64 ~/Desktop/Assignment (master)

$ git add .

Prince@prince MINGW64 ~/Desktop/Assignment (master)

$ git commit -m"nd commit"

On branch master

nothing to commit, working tree clean

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

Prince@prince MINGW64 ~/Desktop/Assignment (master)

$ git checkout css-assignment

Switched to branch 'css-assignment'

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

Prince@prince MINGW64 ~/Desktop/Assignment (css-assignment)

$ vi cc.txt

11. Commit those changes.

Prince@prince MINGW64 ~/Desktop/Assignment (css-assignment)

$ git commit -m"3rd commit"

[css-assignment 50d433c] 3rd commit

1 file changed, 3 insertions(+)

create mode 100644 cc.txt

12. Switch to master.

Prince@prince MINGW64 ~/Desktop/Assignment (css-assignment)

$ git checkout master

Switched to branch 'master'

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

Prince@prince MINGW64 ~/Desktop/Assignment (master)

$ git merge css-assignment

Updating 88b6da1..50d433c

Fast-forward

cc.txt | 3 +++

1 file changed, 3 insertions(+)

create mode 100644 cc.txt

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

Prince@prince MINGW64 ~/Desktop/Assignment (master)

$ git branch -d css-assignment

Deleted branch css-assignment (was 50d433c).

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

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

Prince@prince MINGW64 ~/Desktop/Assignment (master)

$ git branch js-assignment

2. Switch to ‘js-assignments’ branch.

Prince@prince MINGW64 ~/Desktop/Assignment (master)

$ git checkout js-assignment

Switched to branch 'js-assignment'

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

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

Prince@prince MINGW64 ~/Desktop/Assignment (js-assignment)

$ git add .

Prince@prince MINGW64 ~/Desktop/Assignment (js-assignment)

$ git commit -m"1st js commit"

On branch js-assignment

nothing to commit, working tree clean

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

Prince@prince MINGW64 ~/Desktop/Assignment (js-assignment)

$ vi README.txt

6. Commit those changed files.

Prince@prince MINGW64 ~/Desktop/Assignment (js-assignment)

$ git commit -m"2nd change in"

On branch js-assignment

nothing to commit, working tree clean

7. Switch to master branch.

Prince@prince MINGW64 ~/Desktop/Assignment (js-assignment)

$ git checkout master

Switched to branch 'master'

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

Prince@prince MINGW64 ~/Desktop/Assignment (master)

$ vi README.txt

Prince@prince MINGW64 ~/Desktop/Assignment (master)

$ git add -f README.txt

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

Prince@prince MINGW64 ~/Desktop/Assignment (master)

$ git checkout js-assignment

Switched to branch 'js-assignment'

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

11. Commit those changes.

Prince@prince MINGW64 ~/Desktop/Assignment (js-assignment)

$ git add .

Prince@prince MINGW64 ~/Desktop/Assignment (js-assignment)

$ git commit -m"3rd commit to js"

[js-assignment 92d0b81] 3rd commit to js

1 file changed, 6 insertions(+)

create mode 100644 README.txt

12. Switch to master.

Prince@prince MINGW64 ~/Desktop/Assignment (js-assignment)

$ git checkout master

Switched to branch 'master'

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

Prince@prince MINGW64 ~/Desktop/Assignment (master)

$ git merge js-assignment

Updating 50d433c..92d0b81

Fast-forward

README.txt | 6 ++++++

1 file changed, 6 insertions(+)

create mode 100644 README.txt

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

Prince@prince MINGW64 ~/Desktop/Assignment (master)

$ git branch -d js-assignment

Deleted branch js-assignment (was 92d0b81).

GIT Remoting

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

SECTION-3 (Pushing assignments to remote repository) - Steps to follow:

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

40. Login on into github account.

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

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

Prince@prince MINGW64 ~/Desktop/Assignment (master)

$ git push origin master

Enumerating objects: 7, done.

Counting objects: 100% (7/7), done.

Delta compression using up to 4 threads

Compressing objects: 100% (6/6), done.

Writing objects: 100% (6/6), 645 bytes | 322.00 KiB/s, done.

Total 6 (delta 1), reused 0 (delta 0), pack-reused 0

remote: Resolving deltas: 100% (1/1), done.

To https://github.com/princeke00/project\_dir.git

88b6da1..92d0b81 master -> master

SECTION-4 (Pushing source code to remote repository using Eclipse GIT plugin) - Steps to follow:

1. One developer from project team will create eclipse projects ‘SampleProj’ & add sample source code files. Then commit all files through eclipse GIT plugin.

2. Collaborate other team members with your github account so that they can also modify the committed files.

3. Other developers from same team will checkout all files from remote repository. This might get conflicts since certain files fail to merge. In such case, merge it manually.

4. Commit & push the ‘SampleProj’ project.