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’.

**mkdir project\_dir**

**cd project\_dir**

1. Initialize git version database.(git init)

**git init**

1. Create a new file index.html.

**created**

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

**git status**

1. Stage the index.html file.

**git add.**

1. Commit index.html

**git commit –m “html file “**

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

**Created**

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

**git status**

1. Configure GIT to ignore all txt files.

**git config --global core.excludesfile ~/.gitignore\_global**

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

**git status**

1. State & commit index.html

**git commit –m “html file”**

1. Log all your comments so far.

**git log**

1. Make some changes in index.html.

**changed**

1. Revert the change made in the previous step using git command.-

**$ git revert “index.html abc”**

1. Again change index.html.

**Changed**

1. Stage index.html –

**git add .**

Revert back the last stage.

**git revert “index.html ”**

1. Rename ‘add’ command to ‘my-add’.

**git my-add .**

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

**git commit –m “html changes”**

1. Revert the last commit.

**git revert “html changes”**

*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’.

**$ git mkdir Assignment**

**$ git cd Assignment**

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

**README.txt**

24.Commit README.txt file.

**git commit –m “committed**”

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

**git branch html-assignments**

26. Switch to ‘html-assignments’ branch.

**git checkout html-assignment**

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

**copied**

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

**git commit –m “html-assgnt”**

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

**changed**

30.Commit those changed files.

**git commit –m “committed”**

31.Switch to master branch.

**git checkout master**

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

**git commit –m “readme.txt updated”**

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

**git checkout html-assignments**

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

**changed**

35.Commit those changes.

**git commit –m “added new paragraph”**

36.Switch to master.

**git checkout master**

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

**git merge html-assignments**

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

**git branch –d html-assignments**

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

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

**git branch css-assignments**

1. Switch to ‘css-assignments’ branch.

**git checkout css-assignments**

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

**Copied**

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

**git commit –m “committed css assignments**”

1. Make minor changes into README.txt file on line 1 belonging to ‘css-assignments’ branch.
2. Commit those changed files.

**git commit –m “readme txt has changed”**

1. Switch to master branch.

**git checkout master**

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

**git commit –m “readme txt has changed again”**

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

**git checkout css-assignments**

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

**Changed**

1. Commit those changes.

**git commit –m “css-assignments has changed”**

1. Switch to master.

**git checkout master**

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

**git merge css-assignments**

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

**git branch –D css-assignments**

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

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

**git branch js-assignments**

1. Switch to ‘js-assignments’ branch.

**git checkout js-assignments**

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

**Copied**

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

**git commit –m “committed javascript assignments”**

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

**Changed**

1. Commit those changed files.

**git commit –m “changes done in readme.txt”**

1. Switch to master branch.

**git checkout master**

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

**git commit –m “changes done in readme into master”**

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

**git checkout js-assignments**

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

**Changed**

1. Commit those changes.

**git commit –m “changes done in javascript assignments”**

1. Switch to master.

**git checkout master**

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

**git merge js-assignments**

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

**git branch –D js-assignments**

*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.

* Open github page and sign in to github account
* Enter username, email and password
* Click create account
* Account created

40. Login on into github account.

* Open github page and click on login
* Enter email id and password
* Click login

41. Create new public repository ‘freshersbatch-jan22’.

* Click on New repository
* Enter repository name ‘freshersbatch-jan22’
* Click on public and click on repository

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

* $ git remote add origin <https://github.com/AhamedUnnisa/fresher-jan22.git>
* $ git push –u origin master

Enumerating objects: 32, done.

Counting objects: 100% (32/32), done.

Delta compression using up to 8 threads

Compressing objects: 100% (29/29), done.

Writing objects: 100% (32/32), 3.42 KiB | 583.00 KiB/s, done.

Total 32 (delta 9), reused 0 (delta 0), pack-reused 0

remote: Resolving deltas: 100% (9/9), done.

To https://github.com/AhamedUnnisa/fresher-jan22.git

\* [new branch] master -> master

Branch 'master' set up to track remote branch ‘master’ from ‘origin’

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.

* Open eclipse IDE then shift to git repository
* Select add on existing local repository
* Browse the repository folder and add to it
* Select to clone a repository and enter our github url. And finish.

1. Collaborate other team members with your github account so that they can also modify the committed files.
2. 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.
3. Commit & push the ‘SampleProj’ project.

* Right click on project and select commit.
* Select commit
* We can git staging view then select commit
* Then right click on the project and select team
* In that team menu select push branch master.