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’.

A: Mkdir project dir

1. Initialize git version database. (git init)

A : Git init

1. Create a new file index.html.

A: Touch index.html

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

A: Git status

1. Stage the index.html file.

A: git add index.html

1. Commit index.html

A: git commit-m “some changes has been made”

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

A: commit –m’ some changes in index.html & touch info.txt

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

A: git status

1. Configure GIT to ignore all txt files.

A: touch . git ignore

Vi .git ignore

\*.txt (added this in the .giti ignore file)

:wg

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

A: git status

1. State & commit index.html

A: Git commit-m’ commiting index.html file’

1. Log all your comments so far.

A: Git log

1. Make some changes in index.html.

A: vi index.html

(changed content in h1 element)

:wg

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

A: git reset – hard

1. Again change index.html.

A: vi index.html

(changed content in h1 element)

:wg

1. Stage index.html

A: git add index.html

1. Revert back the last stage.

A: git restore –staged index.html

1. Rename ‘add’ command to ‘my-add’.

A: Git mv add my-add

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

A: git add

Git commit-m “changes by adding”

Git commit

1. Revert the last commit.

A: git reset HEAD~

*GIT Branching*

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

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

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

A:

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

A: cd…

Mkdir assignments

Cd assignments

Git init

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

A: touch README.txt

1. Commit README.txt file.

A: Git commit-m “created readme file”

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

A: $ git checked -b html-assignments

1. Switch to ‘html-assignments’ branch.

A: switched to a new branch “html-assignments”

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

A: Cp a1.html assignments cp a2.html assignments

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

A: Git commit-m “copy html files”

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

A:

1. Commit those changed files.

A: git commt-m “changes made in html assignments branch”

1. Switch to master branch.

A: git checkout master

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

A: git add

Git commit-m “changes in readme files”

Git commit

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

A: git checkouts html-assignments

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

A:

1. Commit those changes.

A: Git commit-m “Again I made changes in html assignment branch ”

Git commit

1. Switch to master.

A: git checkout master

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

A: git merge html-assignments

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

A: git branch-d html-assignments

Deleted branch html- assignments’ (was b24ce2e)

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

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

A: git branch css-assignments

1. Switch to ‘css-assignments’ branch.

A: git checkout css-assignments

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

A: create css file

Cp\*css assignments

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

A: git add

Git commit-m “changes in css files”

Git commit

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

A:

1. Commit those changed files.

A: git add

Git commit-m “changes in readme file”

Git commit

1. Switch to master branch.

A: git checkout master

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

A:

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

A: git checkout css-assignments

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

A:

1. Commit those changes.

A: git add

Git commit-m “changes in readme”

Git commit

1. Switch to master.

A: git checkout master

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

A: git merger css- assignments

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

A: git branch-d html-assignments

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

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

A: git branch js - assignments

1. Switch to ‘js-assignments’ branch.

A: git checkout js-assignments

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

A: create js file

Cp\*js assignments

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

A: git add

Git commit-m “changes in js files”

Git commit

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

A:

1. Commit those changed files.

A: git add

Git commit-m “changes in js files”

Git commit

1. Switch to master branch.

A: git checkout master

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

master.

A:

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

A: git checkout js-assignments

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

A:

1. Commit those changes.

A: git add

Git commit-m “changes in js files”

Git commit

1. Switch to master.

A: git checkout master

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

A: git checkout js-assignments

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

A: 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:

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

A: search github.com and register using mail

1. Login on into github account.

A: provide username and password

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

A: select NEW-> repository Name->select public option-> Click on create repository

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

A:

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 git hub 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