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

1. Initialize git version database. (git init)

git init

1. Create a new file index.html.

touch index.html

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

git status

1. Stage the index.html file.

git add index.html

1. Commit index.html

git commit

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

touch info.txt

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 rm –cached

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 ‘ index.html committed’

1. Log all your comments so far.

git log

1. Make some changes in index.html.
2. Revert the change made in the previous step using git command.
3. Again change index.html.
4. Stage index.html

git add index.html

1. Revert back the last stage.
2. Rename ‘add’ command to ‘my-add’.
3. Using my\_add command Stage index.html again & commit the changes.
4. Revert the last commit.

*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.

git clone (link)

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

mkdir Assignments

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

Touch README.txt

Git add .

1. Commit README.txt file.

git commit

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

git branch html-assignments

1. Switch to ‘html-assignments’ branch.

git checkout html-assignments

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

touch html-Assignments

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

git add .

git commit –m “html assignments commited”

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

git add .

git commit -m “changes in html file”

1. Switch to master branch.

git checkout master

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

git add .

git commit –m “changed README.txt”

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

Git checkout html-Assignments

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

Git add .

Git commit –m “changes in html-assignments”

1. Switch to master.

git checkout master

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

git merge master

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

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 csss-assignments

1. Copy all CSS assignments inside ‘Assignments’ folder.
2. Commit CSS assignments into ‘css-assignments’ branch.

git add .

git commit –m “Style of heading changed in css”

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

git add .

git commit –m “readme.txt 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 add .

git commit

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

git checkout css-assignments

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

1. Commit those changes.

git add .

git commit

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.

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

git add .

git commit –m “js- assignments commited”

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

git add .

git commit –m “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 add .

git commit

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

Git checkout js-assignments

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

git add .

1. Commit those changes.

git commit

1. Switch to master.

git checkout master

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

git merge master

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

*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.
2. Login on into github account.
3. Create new public repository ‘freshersbatch-oct16’.
4. Commit & push any sample file to this repository under ‘Assignments’ directory.

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.