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\_dr

$ cd project\_dr

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 -m ‘Adding index.htlm’

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

$ git add index.html

$ git status

$ mkdir test

$ cd test

$ git init

$ git add info.txt

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

$ git add info.txt

$ git status

$ git log

1. Configure GIT to ignore all txt files.

$ touch .gitignore(add lnfo.txt in gitignore)

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

$ git status

1. State & commit index.html

$ git add index.html

$ git commit -m “index.html added”

1. Log all your comments so far.

$ git log

1. Make some changes in index.html.

$ git add index.html

$ git status

$ git commit -m “Changes Made”

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

$ git checkout –index.html

1. Again change index.html.

$ git add index.html

$ git status

$ git commit

1. Stage index.html

$ git add index.html

1. Revert back the last stage.

$ git checkout –index.html

1. Rename ‘add’ command to ‘my-add’.

$ git config –global alias.my\_add add

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

$ git my\_add index.html

$ git commit

1. Revert the last commit.

$ git checkout –index.html

*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.
2. Create an empty directory ‘Assignments’ & cd to ‘Assignments’.

$ mkdir assignments

$ cd assignments

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

$ git init

$ git add readme.txt

$ git status

1. Commit README.txt file.

$ git commit -m ‘readme.txt is added’

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.

$ git checkout html assignments –assignments

$git add assignments

$ git commit -m ‘adding assignment’

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

$ git commit -m ‘html assignments added’

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

$ git add html-assignments

$ git status

1. Commit those changed files.

$ git commit

1. Switch to master branch.

$ git checkout html-assignments

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

$ git add readme.txt

$ git status

$ git commit -m ‘Changes made’

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

$ git checkout html-assignments

$ git commit

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

$ git stash

$ git checkout html assignments

$ git add.

1. Commit those changes.

$ git commit -m html-assignments

1. Switch to master.

$ git checkout html-assignments

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

$ git merge html-assignments

1. 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.

$ git checkout css assignment --asignments

$ git add assignments

$ git commit -m ‘adding assignment’

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

$ git init

$ touch assignment.txt

Git commit -m ‘initial commit’

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

$ git stash

$ git checkout css assignments

$ git add readme.txt

1. Commit those changed files.

$ git commit -m ‘made changes’

1. Switch to master branch.

$ git checkout css-assignments

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

$ git stash

$ git checkout css assignments

$ git add readme.txt

$ git commit -m ‘changes made’

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

$ git checkout css-assignmnet

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

$ git stash

$ git checkout css-assignments

$ git add.

1. Commit those changes.

$ git commit -m ‘Changes made for few files’

1. Switch to master.

$ git checkout master

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

$ git checkout css-assignmnet

$ git pull origin master

$ git checkout master

$ git merge css assignment

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

$ git branch -d css-assignment

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

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

$ git branch js-assignments

1. Switch to ‘js-assignments’ branch.

$ giit checkout js-assignments

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

$ git checkout js assignment --asignments

$ git add assignments

$ git commit -m ‘adding assignment’

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

$ git commit -m ‘changes in js files’

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

$ git checkout js-assignments

$ git add.

1. Commit those changed files.

$ git commit -m ‘changes made’

1. Switch to master branch.

$ git checkout master

1. Make minor changes into README.txt file on line 1 & commit those changes into master.
2. Again switch to ‘js-assignments’ branch.

$ git checkout js-assignmnet

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

$ git stash

$ git checkout js-assignments

$ git add

1. Commit those changes.

$ git commit -m “changes in files”

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:

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.