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’.
2. Initialize git version database.(git init)
3. Create a new file index.html.
4. Check the git status. You should find index.html as untracked file.

$ git rm –cached index.html

1. Stage the index.html file.
2. Commit index.html
3. Make few changes in index.html & create a new file info.txt file.
4. Check git status. You should find index.html &info.txt as untracked files.
5. Configure GIT to ignore all txt files.
6. Again check the git status. You should find only index.html as untracked file.
7. State & commit index.html
8. Log all your comments so far.
9. Make some changes in index.html.
10. Revert the change made in the previous step using git command.
11. Again change index.html.
12. Stage index.html
13. Revert back the last stage.
14. Rename ‘add’ command to ‘my-add’.
15. Using my\_add command Stage index.html again & commit the changes.
16. 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.
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.

**$ touch Readme.txt**

1. Commit README.txt file.

**$ git commit -m "readme file is about to 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.
2. Commit HTML assignments into ‘html-assignments’ branch.

**$ git commit –m “going to commit assignment”**

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

**$ git commit –m “changes has been done”**

commit ed76ca64e4892695d71d8b6b45191f018007800b (HEAD -> master)

| Author: sumit kumar joshi <sumitjoshi236@gmail.com>

| Date: Thu Sep 30 09:20:46 2021 +0530

|

| again index is commiting step 2

|

\* commit 044c992a77fa93489f80ea578f857681033e586d

Author: sumit kumar joshi <sumitjoshi236@gmail.com>

Date: Thu Sep 30 09:13:18 2021 +0530

index is going to commit

1. Switch to master branch.

**$ git checkout master**

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

**Made changes and $ git commit –m "changed in 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 commit –m "some minor changes in html files"**

1. Switch to master.

**$ git checkout master**

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-assign**

1. Switch to ‘css-assignments’ branch.

**$ git checkout css-assign**

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

**$ git commit –m "this time is css commiting"**

**$ git touch readme.txt**

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

**$ git commit –m** **"changes in readme files are done"**

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 "commit is done again"**

1. Again switch to ‘css-assignments’ branch

**$ git checkout css-assign**

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

**$ git touch new1.css**

**$ git touch new2.css**

**Changes are done in both the files**

1. Commit those changes.

**$ git commit –m** **"committing css files"**

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-assign**

commit 59418a6f0470f2bc0114921dcea7859796136e9b (HEAD -> css-assign)

| Author: sumit kumar joshi <sumitjoshi236@gmail.com>

| Date: Thu Sep 30 12:58:57 2021 +0530

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

**$ git branch –d css-assign**

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

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

**$ git branch javascriptassignments**

1. Switch to ‘js-assignments’ branch.

**$ git checkout javascriptassignments**

1. Copy all JavaScript assignments inside ‘Assignments’ folder.
2. Commit JavaScript assignments into ‘js-assignments’ branch.

**$ git commit –m "copy has been done in JS assignment"**

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

**$ git commit –m "changes done in JS file"**

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 javascriptassignments**

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

**$ git commit –m "committing in javascript branch"**

commit 563bccac2eb8d3b109f378444cefd62a80480dce (HEAD -> javascriptassignment)

| Author: sumit kumar joshi <sumitjoshi236@gmail.com>

| Date: Thu Sep 30 13:10:37 2021 +0530

|

| committing in javascript branch

1. Switch to master.

**$ git checkout master**

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

**$ git merge javascriptassignments**

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

**$ git branch –d javascriptassignments**

*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.