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’.

Ans: addeshmu@LIN24009066 MINGW64 ~

$ mkdir project\_dir

1. Initialize git version database. (git init)

Ans:$git init

1. Create a new file index.html.

Ans:created new .txt file in repository .

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

Ans: $ git status.

1. Stage the index.html file.

Ans:for staging we used,

$git add .

1. Commit index.html

Ans: $ git commit -m'first commit'.

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

Ans: made few changes in above files in repository.

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

Ans: $ git status

On branch master

Changes not staged for commit:

(use "git add <file>..." to update what will be committed)

(use "git restore <file>..." to discard changes in working directory)

Untracked files:

(use "git add <file>..." to include in what will be committed)

info.txt

html.txt

1. Configure GIT to ignore all txt files.

Ans: addeshmu@LIN24009066 MINGW64 ~/project\_dir (master)

$ git config --global user.username addeshmukh-github

addeshmu@LIN24009066 MINGW64 ~/project\_dir (master)

$ touch .gitignore

And after this added the .txt files to .gitignore file in repository .

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

Ans: $ git status

1. State & commit index.html

Ans: $ git commit -m'adding index.html'

[master f91efe6] adding index.html

Committer: Deshmukh <adarsh.deshmukh@capgemini.com>

Your name and email address were configured automatically based

on your username and hostname. Please check that they are accurate.

You can suppress this message by setting them explicitly:

git config --global user.name "Your Name"

git config --global user.email you@example.com

After doing this, you may fix the identity used for this commit with:

git commit --amend --reset-author

3 files changed, 2 insertions(+)

create mode 100644 .gitignore

create mode 100644 info.txt

however I was working on .txt file I forgot used .html here .

1. Log all your comments so far.

Ans: $ git log

commit f91efe60817a76bc10b6fed8bd9f23fc1961ebc4 (HEAD -> master)

Author: Deshmukh <adarsh.deshmukh@capgemini.com>

Date: Thu Jan 6 15:52:10 2022 +0530

adding index.html

commit 618294d4da96050b17d86944433b814ddb6e1f9f

Author: Deshmukh <adarsh.deshmukh@capgemini.com>

Date: Thu Jan 6 14:54:51 2022 +0530

first commit

1. Make some changes in index.html.

Ans: made changes in file in repository.

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

Ans:used git stash for reverting

$ git stash

Saved working directory and index state WIP on master: f91efe6 adding index.html

1. Again change index.html.

Ans: made changes in repository.

1. Stage index.html

Ans: $ git add html.txt

As I am working on .txt file.

1. Revert back the last stage.

Ans: $ git stash.

Or git revert <id >

We can use any one command here.

1. Rename ‘add’ command to ‘my-add’.

Ans: $ git config --global alias.myadd add

Here we have to focus on spaces for not getting error.

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

Ans: $ git myadd html.txt

Cause I was using .txt file .

$ git commit -m’ committing after renaming’

1. Revert the last commit.

Ans: $ git revert <id >

*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’.

Ans: $ mkdir assignment

For creating thr directory.

$ cd assignment

For coming in to assignment repository.

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

Ans: created README.txt file in assignments repository and write some thing in that file.

1. Commit README.txt file.

Ans $ git commit -m’first commit’

1. Now create a new branch ‘html-assignments’.
2. Ans: $ git checkout -b html-assignment

Switched to a new branch 'html-assignment'

$ git branch html-assignment

fatal: Not a valid object name: 'master'.

addeshmu@LIN24009066 MINGW64 ~/assignment (master)

1. Switch to ‘html-assignments’ branch.

Ans: $ git checkout -b html-assignment

Switched to a new branch 'html-assignment'

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

Ans: created html file in repository.

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

Ans:$ git commit -m’branches’

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

Ans: did few changes in file I repository.

1. Commit those changed files.

Ans: $ git commit -m’changed file’

1. Switch to master branch.

Ans: $ git checkout -b master

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

Ans: did some changes in file and for committing

$ git committing -m’committing’

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

Ans: : $ git checkout -b html-assignment

Switched to a new branch 'html-assignment'

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

Ans: did minor changes in files.

1. Commit those changes.

$ git commit -m’message’

1. Switch to master.

Ans: : $ git checkout -b master

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

Ans: $ git merge html-assignment

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

Ans: $ git branch -d html-assignment

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

1. Create a new branch ‘css-assignments’.
2. Switch to ‘css-assignments’ branch.
3. Copy all CSS assignments inside ‘Assignments’ folder.
4. Commit CSS assignments into ‘css-assignments’ branch.
5. Make minor changes into README.txt file on line 1 belonging to ‘css-assignments’ branch.
6. Commit those changed files.
7. Switch to master branch.
8. Make minor changes into README.txt file on line 3 & commit those changes into master.
9. Again switch to ‘css-assignments’ branch.
10. Make minor changes into few files belonging to ‘css-assignments’ branch.
11. Commit those changes.
12. Switch to master.
13. Merge ‘css-assignments’ branch into master. Confirm all css assignments are shown in master.
14. Finally delete the ‘css-assignments’ branch.

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

1. Create a new branch ‘js-assignments’.
2. Switch to ‘js-assignments’ branch.
3. Copy all JavaScript assignments inside ‘Assignments’ folder.
4. Commit JavaScript assignments into ‘js-assignments’ branch.
5. Make minor changes into README.txt file on line 1 belonging to ‘js-assignments’ branch.
6. Commit those changed files.
7. Switch to master branch.
8. Make minor changes into README.txt file on line 1 & commit those changes into master.
9. Again switch to ‘js-assignments’ branch.
10. Make minor changes into few files belonging to ‘js-assignments’ branch.
11. Commit those changes.
12. Switch to master.
13. Merge ‘js-assignments’ branch into master. Confirm all JavaScript assignments are shown in master.
14. 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.