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

$ pwd

/c/Users/ANKCHAVA

$ cd project\_dir

$ pwd

/c/Users/ANKCHAVA/project\_dir

1. Initialize git version database. (git init)

$ git init

Initialized empty Git repository in C:/Users/ANKCHAVA/project\_dir/.git/

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

On branch master

No commits yet

Untracked files:

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

index.html

nothing added to commit but untracked files present (use "git add" to track)

1. Stage the index.html file.

git add index.html

On branch master

No commits yet

Changes to be committed:

(use "git rm --cached <file>..." to unstage)

new file: index.html

1. Commit index.html

$ git commit -m"commited the file"

[master (root-commit) 3deb14c] commited the file

1 file changed, 16 insertions(+)

create mode 100644 index.html

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

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)

modified: index.html

Untracked files:

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

info.txt

no changes added to commit (use "git add" and/or "git commit -a")

1. Configure GIT to ignore all txt files.

\*.txt //inside .gitignore file

$ git add .gitignore

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

ankchava@LIN24008126 MINGW64 ~/project\_dir (master)

$ git status

branch master

Changes to be committed:

(use "git restore --staged <file>..." to unstage)

new file: .gitignore

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)

modified: index.html

1. State & commit index.html

git commit -am "commited file"

[master b9d0b88] commited file

1 file changed, 2 insertions(+), 1 deletion(-)

1. Log all your comments so far.

git log -p

OR

git log –pretty=oneline

b9d0b88bafafde6fed5e08d80ca5adc819d38583 (HEAD -> master) commited file

f76d5224ad125121ab1bbf2e4511e056ae03d293 Changes made commited

3deb14c685b7c438c25c29d1ae2eabd4aea36cd5 commited the file

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

$ git checkout index.html

Updated 1 path from the index

1. Again change index.html.
2. Stage index.html

$ git add index.html

ankchava@LIN24008126 MINGW64 ~/project\_dir (master)

$ git status

On branch master

Changes to be committed:

(use "git restore --staged <file>..." to unstage)

modified: index.html

1. Revert back the last stage.

git revert HEAD

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 -m”added index file”

1. Revert the last commit.

git revert 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.

git remote add origin github.com/ChavanAnkitaa

1. 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.
2. Commit README.txt file.

git init

git add README.txt

git commit -m”readme file ”

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 a.html

touch b.html

touch c.html

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

git add .

git status

git commit -m"html files a,b,c commit"

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

git add .

git status

git commit -m"changed"

1. Switch to master branch.

git checkout master

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

git add .

git status

git commit -m"readme"

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

git checkout html-assignments

Switched to branch 'html-assignments'

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

$ git status

On branch html-assignments

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)

modified: b.html

no changes added to commit (use "git add" and/or "git commit -a")

ankchava@LIN24008126 MINGW64 ~/Assignments (html-assignments)

$ git add b.html

ankchava@LIN24008126 MINGW64 ~/Assignments (html-assignments)

$ git status

On branch html-assignments

Changes to be committed:

(use "git restore --staged <file>..." to unstage)

modified: b.html

ankchava@LIN24008126 MINGW64 ~/Assignments (html-assignments)

$ git commit -m"b modify"

[html-assignments f61aeaf] b modify

1 file changed, 2 insertions(+), 2 deletions(-)

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

ls

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

gitbranch -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.

touch a.css

touch b.css

touch c.css

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

git add -f a.css b.css c.css

git commit -m”css file”

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

1. Commit those changed files.

git init

git add README.txt

git commit -m"readme"

1. Switch to master branch.

git checkout master

1. Make minor changes into README.txt file on line 3 & commit those changes into master.
2. Again switch to ‘css-assignments’ branch.

git checkout css-assignments

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

git init

git add c.css

git commit -m"ccss"

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.

git branch -D css-assignments

Deleted branch css-assignments (was 8a6bbe8).

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.

touch a.js

touch b.js

touch c.js

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

git add -f a.js b.js c.js

git commit -m”js committing”

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

git add .

git status

git commit -m”readme”

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-assignments

1. Make minor changes into few files belonging to ‘js-assignments’ branch.
2. Commit those changes.
3. git add .
4. git status
5. git commit -m”b.js”
6. 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.