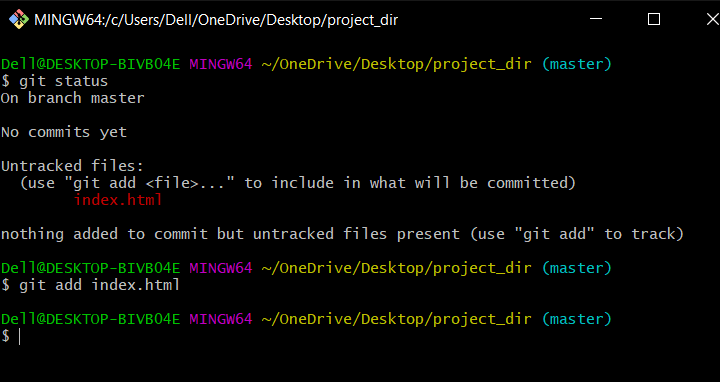
Install GIT& make sure it is added into PATH.

Section 0 -Use GIT as local VCS. Steps to follow:

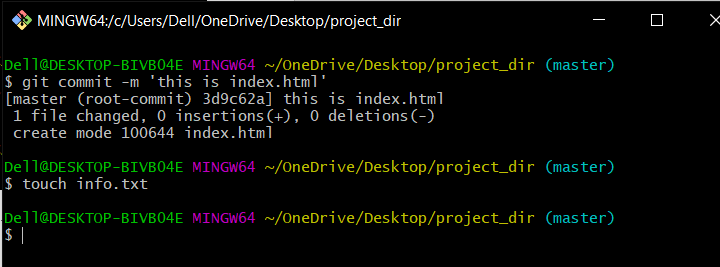
1. Create a directory ‘project\_dir’ & cd to ‘project\_dir’.
2. Initialize git version database. (git init)
3. Create a new file index.html.



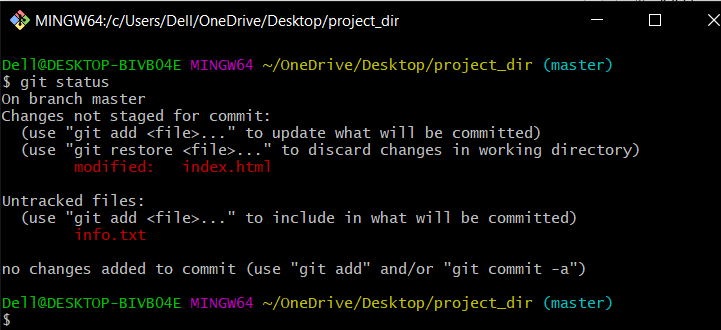
1. Check the git status. You should find index.html as untracked file.
2. Stage the index.html file.



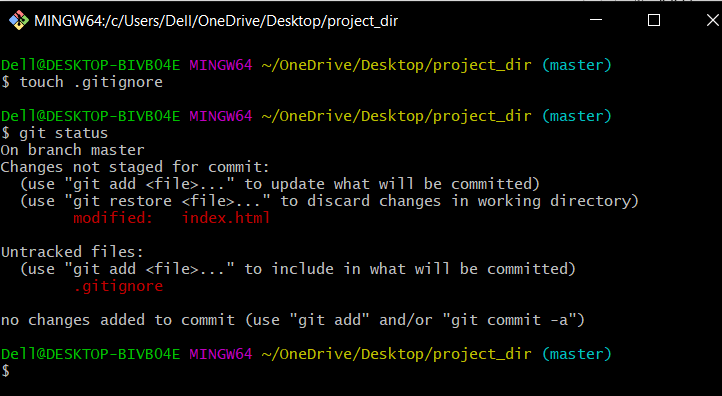
1. Commit index.html
2. Make few changes in index.html & create a new file info.txt file.



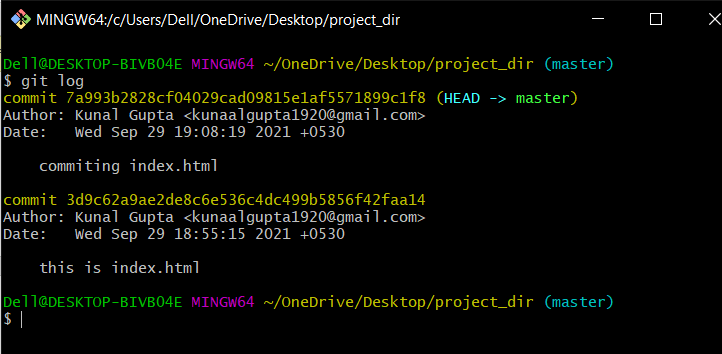
1. Check git status. You should find index.html & info.txt as untracked files.



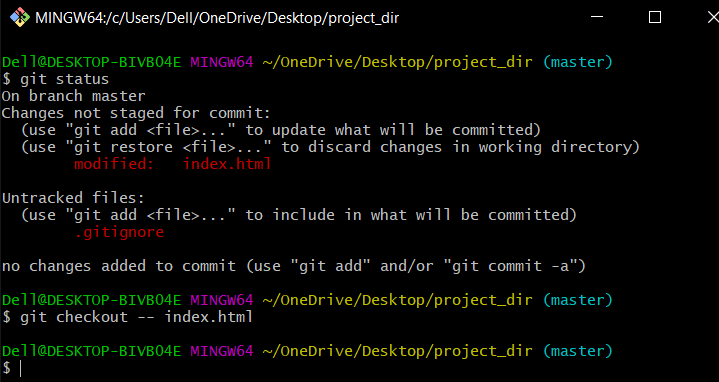
1. Configure GIT to ignore all txt files.
2. Again check the git status. You should find only index.html as untracked file.



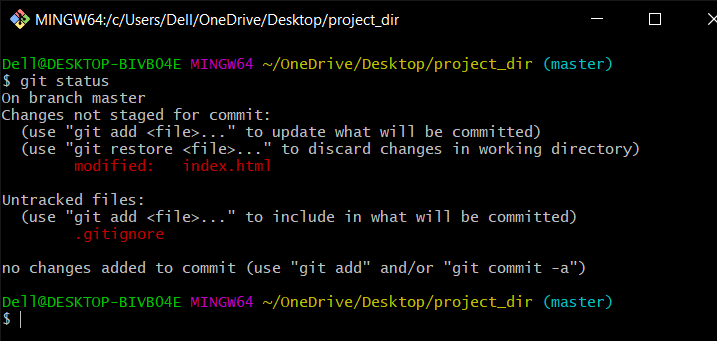
1. State & commit index.html
2. Log all your comments so far.



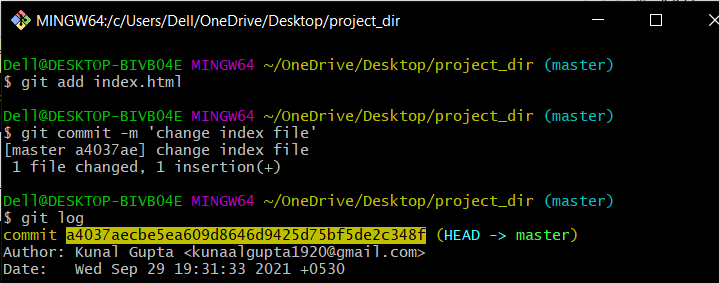
1. Make some changes in index.html.
2. Revert the change made in the previous step using git command.



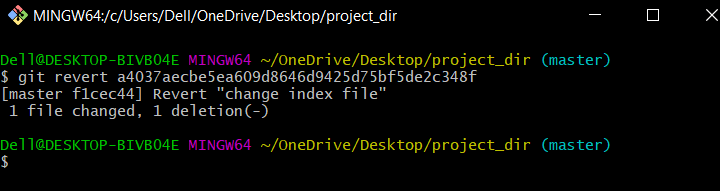
1. Again change index.html.



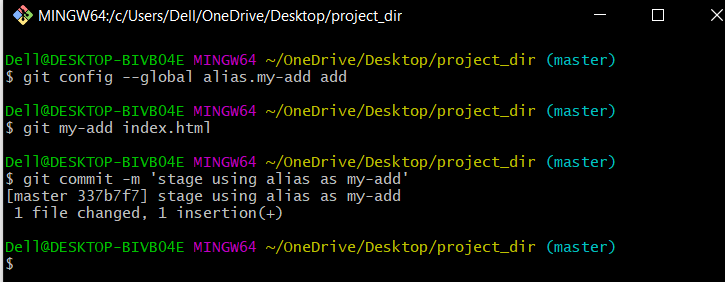
1. Stage index.html



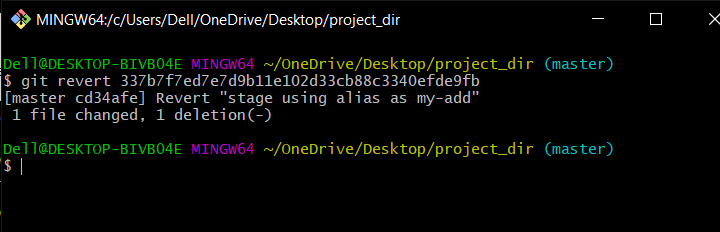
1. Revert back the last stage.



1. Rename ‘add’ command to ‘my-add’.
2. Using my-add command Stage index.html again & commit the changes.



1. 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 assignment

$ cd assignment

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

$ echo 'This is a git Assignment' > README.txt

1. Commit README.txt file.

$ git init

$ git add README.txt

$ git commit -m 'This is 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.
2. Commit HTML assignments into ‘html-assignments’ branch.

$ git add index.html

$ git commit -m 'add index.html in html-assignment branch'

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

$ git add index.html

$ git commit -m 'adding a paragraph'

1. Switch to master branch.

$ git checkout master

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

$ git add README.txt

$ git commit -m 'More description'

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 index.html

$ git commit -m 'Adding style'

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-assignments

1. Switch to ‘css-assignments’ branch

$ git checkout css-assignments

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

$ git add \*.css

$ git commit -m 'Adding css file'

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

$ git add README.txt

$ git commit -m 'changed README.txt in css-assignment branch'

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 README.txt

$ git commit -m 'changed README.txt in master branch'

1. 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 add styles.css

$ git commit -m 'add style to list'

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

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 \*.js

$ git commit -m 'Add js files'

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

$ git add README.txt

$ git commit -m 'update README in js-assignments’

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 README.txt

$ git commit -m 'update README in master'

1. 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.

$ git add \*.js

$ git commit -m 'update js 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.

