Quiz#1

Time: 5 mins

Q#1 Given the following command sequence:

- mkdir -p project/docs/images
- cd project/docs
- touch file.txt

Now, you want to remove the entire **project** directory, including all subdirectories and files. Why would **rmdir -p project/docs/images** fail, and how can you modify the command to successfully remove the entire structure?

Solution:

The *rmdir -p project/docs/images* command would fail because **rmdir** only removes empty directories. In this case, the directory *project/docs* contains a file (*file.txt*), which makes it non-empty. Therefore, *rmdir* cannot remove *project/docs* or its parent directories.

To successfully remove the entire *project* directory, including all non-empty subdirectories and their files, you would need to use the *rm* command with the recursive option (*-r*). The correct command is:

rm -r project

This command will recursively delete the *project* directory, its subdirectories (*docs* and *images*), and all files within (*file.txt*).

Q#2 You are inside a directory that contains three subdirectories: **`dir1`**, **`dir2`**, and **`dir3`**. You need to list all files inside **`dir1`** whose names start with the letters "log". Write the command that achieves this and explain how the wildcard character is used in your command.

Solution:

The correct command to list all files inside dir1 whose names start with "log" is:

*ls dir1/log**

Explanation:

ls is used to list the contents of a directory.
dir1/ specifies that we want to look inside the **dir1** directory.

log* uses the wildcard ***** (asterisk) to match any characters following "log". This means that it will list all files that begin with "log" (e.g., **logfile.txt**, **logdata.csv**, **log1.log**).
The wildcard ***** stands for "zero or more characters," so it can match any file that starts with "log" and has any extension or additional characters following it.