

1. How do you distinguish between `shutil.copy()` and `shutil.copytree()`?

Ans: `shutil.copy()` method is used to copy the contents of a file from one file to another file/folder, it primary takes two arguments `src,dest`, `src` represents the file to be copied where as destination refers to the file/folder to where the `src` data should be copied, if `dest` is a folder name the `src` with exact name will be copied to the `dest` folder, if its a file then the contents of `src` will be copied to `dest` where `dest` retains it name.

`shutil.copytree()` function is used to copy the entire contents of a folder to other folder. it also takes two arguments `src & dest`, it copies all the content recursively and stores it in `dest`. the important catch here is `dest` must not exist prior to this and it will be created during the copy operation. Permissions and times of directories are copied with `shutil.copystat()` and individual files are copied using `shutil.copy2()` by default which can be modified using `copy_function` attribute.

2. What function is used to rename files??

Ans: `os.rename()` function is used to rename files or directories using a python program, this function takes two arguments `src` and `dest`, `src` represents the name file/directory which we want to rename, whereas `dest` represents the new name of the file/directory.

3. What is the difference between the delete functions in the `send2trash` and `shutil` modules?

Ans: `Shutil` module provides a function called as `shutil.rmtree()` which deletes a directory and all its contents. The other functions with similar functionality are `os.remove()` -> removes a file, `os.rmdir()` removes a empty directory. The problem with these functions is once a file is deleted. it will be lost permanently, if a file is deleted accidentally using these methods there is no way we can recover the deleted file

Where as `send2trash` module provides a function called `send2trash.send2trash()` to delete a file/directory. these methods moves the files/directories to trash folder instead of permanently deleting them. hence if a file/folder is deleted accidentally it can be still recovered from trash folder, if is deleted using the `send2trash.send2trash()` function. `send2trash` is not included with python standard library like `os & shutil` modules. it needs to be installed explicitly using the command `!pip install send2trash`

4. `ZipFile` objects have a `close()` method just like `File` objects' `close()` method. What `ZipFile` method is equivalent to `File` objects' `open()` method?

Ans: `ZipFile` Module provides a method called as `zipfile.ZipFile()` to read and write to zipFiles. it takes arguments like filename and mode etc `zipfile.ZipFile('filename', mode = 'r')`

5. Create a programme that searches a folder tree for files with a certain file extension (such as `.pdf` or `.jpg`). Copy these files from whatever location they are in to a new folder

```
import os
```

```
import shutil
```

```
def search_and_copy(source,destination,extensions):
    source = os.path.abspath(source)
    destination = os.path.abspath(destination)
    for foldername, subfolder, filenames in os.walk(source):
        print(f'Folder Name → {foldername}',end='\n\n')
        print(f'Sub Folders → {subfolder}',end='\n\n')
        print(f'Files → {filenames}',end='\n\n')
        for filename in filenames:
            fileName,extension = os.path.splitext(filename)
            if extension in extensions:
```

```
targetFile = foldername+os.path.sep+fileName+extension
shutil.copy(targetFile, destination)
print(f'Files copied successfully from {source} to {destination}')
```

```
extensions = ['.pdf', '.jpg', '.ipynb']
source = 'Dummy Source'
destination = 'Dummy Destination'
search_and_copy(source, destination, extensions)
```