**ANSWERS**

**Q1. How do you distinguish between shutil.copy() and shutil.copytree()?**

**Ans.** The **shutil.copy()** function is used to copy a single file from one location to another. It takes two arguments: the source file path and the destination file path. It creates a new file at the destination path and copies the contents of the source file into it.

On the other hand, the **shutil.copytree()** function is used to recursively copy an entire directory tree from the source directory to the destination directory. It takes two arguments: the source directory path and the destination directory path. It creates a new directory at the destination path and copies all files and subdirectories from the source directory to the destination directory.

**Q2. What function is used to rename files??**

**Ans.** The **os.rename()** function is used to rename files in Python. It takes two arguments: the current name (or path) of the file and the new name (or path) that you want to assign to the file. The function renames the file accordingly.

**Q3. What is the difference between the delete functions in the send2trash and shutil modules?**

**Ans.** The delete functions in the **send2trash** and **shutil** modules have the following differences:

* **send2trash**: The **send2trash** module provides a **send2trash()** function that moves a file or directory to the recycle bin or trash instead of permanently deleting it. This allows for a safer deletion process, as the file can be restored from the recycle bin if needed.
* **shutil**: The **shutil** module provides various functions for file and directory operations, including file deletion. The **shutil.rmtree()** function is commonly used to delete a directory and all its contents recursively. It permanently deletes the files and directories, and they cannot be easily restored.

**Q4.ZipFile objects have a close() method just like File objects’ close() method. What ZipFile method is equivalent to File objects’ open() method?**

**Ans.** The **ZipFile** method equivalent to File objects' **open()** method is the **ZipFile()** constructor. When working with **ZipFile** objects, you can create a new instance by using the **ZipFile()** constructor and providing the path to the ZIP file. This is similar to how you use the **open()** function to create a new **File** object by providing the path to the file.

**Q5. 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.**

**Ans.** Here's an example program that searches a folder tree for files with a specific file extension and copies them to a new folder:

import os

import shutil

def copy\_files\_by\_extension(source\_folder, destination\_folder, extension):

for root, dirs, files in os.walk(source\_folder):

for file in files:

if file.endswith(extension):

source\_path = os.path.join(root, file)

destination\_path = os.path.join(destination\_folder, file)

shutil.copy2(source\_path, destination\_path)

print(f"File copied: {file}")

# Example usage

source\_folder = "path/to/source/folder"

destination\_folder = "path/to/destination/folder"

extension = ".pdf" # Specify the desired file extension

copy\_files\_by\_extension(source\_folder, destination\_folder, extension)

In the above program, **copy\_files\_by\_extension()** function takes three parameters: the source folder, the destination folder, and the desired file extension. It recursively searches for files with the specified extension in the source folder and its subdirectories. For each matching file found, it copies it to the destination folder using **shutil.copy2()**. The function prints the name of each file that is successfully copied.