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

Ans-The shutil.copy() and shutil.copytree() functions in the shutil module in Python are used for file and directory copying, respectively. The main differences between them are:

shutil.copy(src, dst): This function is used to copy a single file from the source path (src) to the destination path (dst). It preserves the filename and copies the file to the specified destination. If the destination path is a directory, it will preserve the filename and copy the file into that directory.

shutil.copytree(src, dst): This function is used to recursively copy an entire directory tree from the source path (src) to the destination path (dst). It copies all the files and directories within the source directory, including subdirectories and their contents, to the specified destination. If the destination directory already exists, it raises an error.

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

Ans-In Python, the os.rename() function is used to rename files. It takes two arguments: the current file name (path) and the new file name (path).

Example: import os

current\_name = "old\_file.txt"

new\_name = "new\_file.txt"

os.rename(current\_name, new\_name)

##This will rename the file from "old\_file.txt" to "new\_file.txt" in the same directory.

**3. 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 different functionalities:

send2trash.send2trash(path): This function from the send2trash module moves a file or directory to the operating system's trash or recycle bin instead of permanently deleting it. It provides a safer alternative to deleting files, as they can be later restored from the trash if needed.

shutil.rmtree(path): This function from the shutil module removes an entire directory tree permanently, including all files and subdirectories within it. It deletes the specified directory and all its contents without moving them to the trash.

The key difference is that send2trash.send2trash() sends files and directories to the trash, while shutil.rmtree() permanently deletes them.

**4.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 that is equivalent to File objects' open() method is also called ZipFile(). This method is used to create a new ZipFile object and open an existing ZIP file for reading, writing, or appending. It takes the path to the ZIP file as its first argument and an optional mode parameter that specifies the mode to open the file in. The mode parameter can take one of three values: "r" for reading (the default), "w" for writing, or "a" for appending.

For example:

import zipfile

with zipfile.ZipFile('example.zip', 'r') as zip\_file:

## do something with the ZipFile object

**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.**

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\_with\_extension(source\_dir, destination\_dir, extension):

for foldername, subfolders, filenames in os.walk(source\_dir):

for filename in filenames:

if filename.endswith(extension):

file\_path = os.path.join(foldername, filename)

shutil.copy(file\_path, destination\_dir)

# Example usage:

source\_dir = 'C:/Users/username/source\_folder'

destination\_dir = 'C:/Users/username/destination\_folder'

extension = '.pdf'

copy\_files\_with\_extension(source\_dir, destination\_dir, extension)