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

- `shutil.copy(src, dst)`: Copies a single file from the source path (`src`) to the destination path (`dst`). It can also accept an optional `follow\_symlinks` argument to control how symbolic links are handled.

- `shutil.copytree(src, dst)`: Recursively copies an entire directory tree rooted at `src` to a new directory at `dst`. All the files and subdirectories in the source directory are copied to the destination directory.

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

The `os.rename()` function is used to rename files or directories.

#### Example:

```python

import os

os.rename('old\_filename.txt', 'new\_filename.txt')

```

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

- `send2trash.send2trash(path)`: Sends the file or directory at the specified path to the trash or recycle bin. It is safer because the file can be recovered from the trash.

- `shutil.rmtree(path)`: Permanently deletes the directory and all its contents. It cannot be recovered once deleted.

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

The `ZipFile` method equivalent to File objects’ `open()` method is `ZipFile(filename, mode)`. It is used to open a ZIP file.

#### Example:

```python

from zipfile import ZipFile

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

# Perform operations on the ZIP file

zip\_file.extractall('extracted\_files')

```

### 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.

Here is a Python program that accomplishes this task:

```python

import os

import shutil

def search\_and\_copy(src\_folder, dest\_folder, file\_extension):

# Ensure the destination folder exists

if not os.path.exists(dest\_folder):

os.makedirs(dest\_folder)

# Walk through the folder tree

for foldername, subfolders, filenames in os.walk(src\_folder):

for filename in filenames:

if filename.lower().endswith(file\_extension):

# Construct the full file path

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

# Copy the file to the destination folder

shutil.copy(file\_path, dest\_folder)

print(f'Copied {file\_path} to {dest\_folder}')

# Usage example

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

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

file\_ext = '.pdf' # Change this to the desired file extension

search\_and\_copy(source\_folder, destination\_folder, file\_ext)

```

### Explanation

1. \*\*Imports\*\*: Import the necessary modules, `os` for directory traversal and `shutil` for copying files.

2. \*\*Function Definition\*\*: Define a function `search\_and\_copy` that takes the source folder, destination folder, and file extension as arguments.

3. \*\*Destination Folder Check\*\*: Ensure the destination folder exists. If not, create it using `os.makedirs()`.

4. \*\*Walking the Folder Tree\*\*: Use `os.walk()` to traverse the folder tree starting from the source folder.

5. \*\*File Matching and Copying\*\*: Check if each file's name ends with the desired file extension. If so, copy it to the destination folder using `shutil.copy()` and print a message indicating the copy operation.