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

shutil.copy() and shutil.copytree() are both functions in the shutil module used for file and directory operations in Python. Here's how they differ:

* shutil.copy(src, dst): This function is used to copy a single file from the source (src) to the destination (dst). It takes two arguments: the path of the source file and the path of the destination file or directory. If dst specifies a directory, the source file will be copied into that directory with the same filename. If dst specifies a file, the source file will be copied and renamed to the specified filename. This function preserves the file's permissions and other metadata.
* shutil.copytree(src, dst): This function is used to recursively copy an entire directory tree from the source (src) to the destination (dst). It takes two arguments: the path of the source directory and the path of the destination directory. The entire directory structure and all files and subdirectories within the source directory will be copied to the destination directory. If the destination directory already exists, an error will be raised. This function preserves the directory structure, file permissions, and other metadata.

In summary, shutil.copy() is used to copy individual files, while shutil.copytree() is used to copy entire directory trees.

Eg:

import shutil

# Copy a single file

shutil.copy('source/file.txt', 'destination/file\_copy.txt')

# Copy a directory tree

shutil.copytree('source/directory', 'destination/directory\_copy')

2. What function is used to rename files??

The function used to rename files is os.rename(). It is part of the os module in Python's standard library. Here's an example that demonstrates how to use os.rename() to rename a file:

import os

# Rename a file

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

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

The send2trash and shutil modules in Python provide different ways to delete files and directories.

The send2trash module provides a safer way to delete files or directories by sending them to the operating system's trash or recycle bin instead of permanently deleting them. This allows for the possibility of recovering the deleted items if needed. The send2trash module provides a single function called send2trash() which takes the path to the file or directory as an argument.

On the other hand, the shutil module provides various functions for file and directory operations, including deleting files and directories. The shutil module's rmtree() function is commonly used to delete directories and their contents recursively. It permanently deletes the specified files and directories, without the option of recovering them.

Eg:

import os

import send2trash

import shutil

# Create a temporary file

filename = 'temp\_file.txt'

with open(filename, 'w') as file:

file.write('This is a temporary file.')

# Delete the file using send2trash

send2trash.send2trash(filename)

# Check if the file still exists

if os.path.exists(filename):

print(f"{filename} still exists.")

else:

print(f"{filename} has been moved to the trash.")

# Create another temporary file

filename = 'temp\_file.txt'

with open(filename, 'w') as file:

file.write('This is another temporary file.')

# Delete the file using shutil

shutil.rmtree(filename)

# Check if the file still exists

if os.path.exists(filename):

print(f"{filename} still exists.")

else:

print(f"{filename} has been permanently 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 equivalent method in ZipFile objects to File objects' open() method is ZipFile's constructor, which is also called open().

To open a ZIP file for reading or writing, you can use the ZipFile constructor with the file path and mode as arguments.

Eg:

import zipfile

# Open a ZIP file for reading

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

# Perform operations on the ZIP file

print(zip\_file.namelist())

# Open a ZIP file for writing

with zipfile.ZipFile('new.zip', 'w') as zip\_file:

# Perform operations on the ZIP file

zip\_file.write('file.txt')

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\_files(source\_folder, target\_folder, file\_extension):

# Create the target folder if it doesn't exist

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

os.makedirs(target\_folder)

# Walk through the source folder and its subdirectories

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

for file in files:

if file.endswith(file\_extension):

source\_path = os.path.join(root, file) # Full path of the source file

target\_path = os.path.join(target\_folder, file) # Full path of the target file

shutil.copy2(source\_path, target\_path) # Copy the file to the target folder

# Example usage

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

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

file\_extension = '.pdf'

search\_and\_copy\_files(source\_folder, target\_folder, file\_extension)