**Python Assignment 10**

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

shutil.copy() and shutil.copytree() are both functions from the shutil module in Python, used for copying files and directories respectively. However, they have different purposes and usage:

shutil.copy(src, dst):

The shutil.copy() function is used to copy a single file from the source (src) to the destination (dst).

It copies the contents of the source file to the specified destination file or directory.

If the destination is a directory, the copied file will have the same name as the source file.

If the destination is a file, the function will copy the source file and create or overwrite the destination file.

Example:

import shutil

# Copy a single file to another location

shutil.copy('source\_file.txt', 'destination\_file.txt')

shutil.copytree(src, dst):

The shutil.copytree() function is used to recursively copy an entire directory and its contents from the source (src) to the destination (dst).

It creates a new directory at the destination and replicates the entire directory structure and contents from the source directory.

If the destination directory already exists, the function will raise an error unless the ignore parameter is used to specify how to handle existing files and directories.

Example:

import shutil

# Recursively copy a directory and its contents

shutil.copytree('source\_directory', 'destination\_directory')

2. What function is used to rename files?

The function used to rename files in Python is os.rename() from the os module. This function allows you to rename a file by providing the current file name (path) and the new desired file name (path).

Here's the basic syntax of the os.rename() function:

import os

# Rename a file

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

Where:

current\_file\_name is the path of the file you want to rename.

new\_file\_name is the new desired name or path for the file.

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

Both the send2trash and shutil modules in Python provide functions to delete files and directories, but they have different behaviors and purposes:

send2trash Module:

The send2trash module provides a safe way to move files and directories to the system's trash or recycle bin, rather than immediately deleting them.

The send2trash function sends the specified file or directory to the trash, allowing users to recover the deleted items if needed.

This module is often used when you want to provide users with a "soft delete" option, where files are moved to the trash instead of being permanently deleted.

Example:

from send2trash import send2trash

# Move a file to the trash

send2trash('myfile.txt')

shutil Module:

The shutil module provides more direct file and directory operations, including functions like shutil.rmtree() to remove directories and os.remove() to remove files.

The functions in the shutil module delete files and directories permanently, without moving them to the trash or recycle bin.

These functions are suitable when you want to permanently remove files and directories from the file system.

Example:

import os

import shutil

# Permanently delete a file using shutil

os.remove('myfile.txt')

# Permanently delete a directory and its contents using shutil

shutil.rmtree('mydirectory')

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 ZipFile method to File objects' open() method is the ZipFile() constructor itself. Just like you use the open() function to create a file object and open a file, you use the ZipFile() constructor to create a ZipFile object and open a ZIP archive.

Here's a comparison:

File Objects:

open() method is used to create a file object and open a file.

Example:

file\_obj = open('myfile.txt', 'r')

ZipFile Objects:

ZipFile() constructor is used to create a ZipFile object and open a ZIP archive.

Example:

from zipfile import ZipFile

zip\_obj = ZipFile('myarchive.zip', 'r')

In both cases, you create an object that provides methods to interact with the file or archive. For File objects, you use methods like read(), write(), etc., to work with the file's content. For ZipFile objects, you use methods like extract(), extractall(), etc., to work with the archive's content.

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 copy\_files\_by\_extension(source\_folder, target\_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)

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

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

print(f"Copied: {source\_path} -> {target\_path}")

source\_folder = '/path/to/source/folder' # Replace with your source folder path

target\_folder = '/path/to/target/folder' # Replace with your target folder path

extension = '.pdf' # Replace with the desired file extension

# Create the target folder if it doesn't exist

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

os.makedirs(target\_folder)

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