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

Ans-: **copy()** will **copy** a single file, **shutil**. **copytree()** will **copy** an entire folder and every folder and file contained in it. Calling **shutil**. **copytree**( source, destination ) will **copy the** folder at **the** path source , along with all of its files and subfolders, to **the** folder at **the** path destination .

Ex-:shutil.copy()

>>> **import shutil, os**

>>> **os.chdir('C:\\')**

❶ >>> **shutil.copy('C:\\spam.txt', 'C:\\delicious')**

'C:\\delicious\\spam.txt'

❷ >>> **shutil.copy('eggs.txt', 'C:\\delicious\\eggs2.txt')**

'C:\\delicious\\eggs2.txt'

Ex-:shutil.copytree()

>>> **import shutil, os**

>>> **os.chdir('C:\\')**

>>> **shutil.copytree('C:\\bacon', 'C:\\bacon\_backup')**

'C:\\bacon\_backup'

2. What is the name of the feature used to rename files?

Ans-: Open **File** Explorer by going to My Computer, or by pressing Windows Key + E on your keyboard. Find the **file** you want to **rename**, select it and select **Rename** on the ribbon (or press F2 on your keyboard). Type the new **name** you want the **file** to have and press Enter.

shutil.move(*source, destination*)

If *destination* points to a folder, the *source* file gets moved into *destination* and keeps its current filename.

1.>>> **import shutil**

>>> **shutil.move('C:\\bacon.txt', 'C:\\eggs')**

'C:\\eggs\\bacon.txt'

The *destination* path can also specify a filename. In the following example, the *source* file is moved and renamed.

2. >>> **shutil.move('C:\\bacon.txt', 'C:\\eggs\\new\_bacon.txt')**

'C:\\eggs\\new\_bacon.txt'

Both of the previous examples worked under the assumption that there was a folder eggs in the C:\ directory. But if there is no eggs folder, then move() will rename bacon.txt to a file named eggs.

>>> **shutil.move('C:\\bacon.txt', 'C:\\eggs')**

'C:\\eggs'

3. How are the delete functions in the send2trash and shutil modules different?

Ans-:Since Python’s built-in shutil.rmtree() function irreversibly deletes files and folders, it can be dangerous to use. A much better way to delete files and folders is with the third-party send2trash module. You can install this module by running pip install send2trash from a Terminal window.

>>> **import send2trash**

>>> **baconFile = open('bacon.txt', 'a') # creates the file**

>>> **baconFile.write('Bacon is not a vegetable.')**

25

>>> **baconFile.close()**

>>> **send2trash.send2trash('bacon.txt')**

4. ZipFile objects have a close() function that is similar to that of File objects. What ZipFile method is similar to the open() method of File objects?

Ans-: To create your own compressed ZIP **files**, you must **open** the **ZipFile object** in write mode by passing 'w' as the second argument. (This is similar to **opening** a text **file** in write mode by passing 'w' to the **open() function**.) This code will create a new **ZIP file** named new. zip that has the compressed contents of spam.

The zipfile.ZipFile() function is equivalent to the open() function; the first argument is the filename, and the second argument is the mode to open the ZIP file in (read, write, or append).

>>> **import zipfile**

>>> **newZip = zipfile.ZipFile('new.zip', 'w')**

>>> **newZip.write('spam.txt', compress\_type=zipfile.ZIP\_DEFLATED)**

>>> **newZip.close()**

5. Create a programme that searches a folder tree for files with a certain file extension (such as .pdf or .jpg). Copy these files to a new folder from wherever they are now.

Ans-:

# Write a program that walks through a folder tree

# and searches 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, shutil

def selectiveCopy(folder, extensions, destFolder):

folder = os.path.abspath(folder)

destFolder = os.path.abspath(destFolder)

print('Looking in', folder, 'for files with extensions of', ', '.join(extensions))

for foldername, subfolders, filenames in os.walk(folder):

for filename in filenames:

name, extension = os.path.splitext(filename)

if extension in extensions:

fileAbsPath = foldername + os.path.sep + filename

print('Coping', fileAbsPath, 'to', destFolder)

shutil.copy(fileAbsPath, destFolder)

extensions = ['.php', '.py']

folder = 'randomFolder'

destFolder = 'selectiveFolder'

selectiveCopy(folder, extensions, destFolder)

|  |
| --- |
|  |
|  |  |