Chapter-3 ORGANIZING FILES

3.3.1 The shutil Module

The shutil (or shell utilities) module has functions to let you copy, move, rename, and delete files in your Python programs. To use the shutil functions, you will first need to use import shutil.

3.3.1.1 Copying Files and Folders

shutil.copy(source, destination) : copies the file at path source to path to the folder at the path destination. If destination is a filename, it will be used as the new name of the copied file. This function returns a string or Path object of the copied file.

In [ ]:

```
import shutil, os
from pathlib import Path
p = Path.home()
shutil.copy(p / 'hello.txt', p /'spam') # copies hello.txt to folder Desktop with name hell
```

In [ ]:

```
shutil.copy(p / 'hello.txt', p / 'spam/hello2.txt')# copies hello.txt to folder Desktop wit
```

In [ ]:

```
shutil.copytree(p / 'spam', p / 'spam_backup')
```

3.3.1.2 Moving and Renaming Files and Folders

shutil.move(source, destination) will move the file or folder at the path source to the path destination and will return a string of the absolute path of the new location.

In [ ]:

```
p = Path.home()
shutil.move(p /'spam//hello2.txt',p/ 'eggs')
```

In [ ]:

```
# The destination path can also specify a filename.
shutil.move(p/'spam//hello.txt', p/'spam_backup//helo.txt')
```

3.3.1.3 Permanently Deleting Files and Folders

To delete a folder and all of its contents,use the shutil module.

- Calling os.unlink(path) will delete the file at path.

- Calling os.rmdir(path) will delete the folder at path. This folder must be empty of any files or folders.
- Calling shutil.rmtree(path) will remove the folder at path, and all files and folders it contains will also be deleted.

In [ ]:

```
import os
from pathlib import Path
for filename in Path.home().glob('*.txt'):
 #os.unlink(filename)
 print(filename)
```

3.3.1.4 Safe Deletes with the send2trash Module

A much better way to delete files and folders is with the third-party send2trash module,which can be installed by running
pip install --user send2trash from a Terminal window.Using send2trash is much safer than Python’s regular delete functions, because it will send folders and files to your computer’s trash or recycle bin instead of permanently deleting them.

In [ ]:

```
import send2trash
```

In [ ]:

```
baconFile = open('bacon.txt', 'a')
```

In [ ]:

```
baconFile.write('Bacon is not a vegetable.')
baconFile.close()
```

In [ ]:

```
send2trash.send2trash('bacon.txt')
```

```
### 3.3.1.5 Walking a Directory Tree
Say you want to rename every file in some folder and also every file in every subfolder
of that folder. That is, you want to walk through the directory tree, touching each file
as you go. \
Let’s look at the C:\delicious folder with its contents, shown in Figure below
![image.png](attachment:image.png)
```

In [ ]:

```
import os

for folderName, subfolders, filenames in os.walk('C:\\Users\\91984\\Documents'):
    print('The current folder is ' + folderName)

    for subfolder in subfolders:
        print('SUBFOLDER OF ' + folderName + ': ' + subfolder)

    for filename in filenames:
        print('FILE INSIDE ' + folderName + ': '+ filename)

    print('')
```

Unlike range(), the os.walk() function will return three values on each iteration through the loop:\
- A string of the current folder’s name
- A list of strings of the folders in the current folder
- A list of strings of the files in the current folder

3.3.1.6 Compressing Files with the zipfile Module

Python programs can create and open (or extract) ZIP files using functions in the zipfile module.\
Say you have a ZIP file named example.zip that has the contents shown in Figure below
![image.png](attachment:image.png)

3.3.1.6.1 Reading ZIP Files

To read the contents of a ZIP file, first create a ZipFile objectTo create a ZipFile object, call the zipfile.ZipFile() function, passing it a string of the .ZIP file’s filename. \
For example:

In [ ]:

```
import zipfile, os

from pathlib import Path
p = Path.home()
exampleZip = zipfile.ZipFile(p / 'example.zip')
exampleZip.namelist()A ZipFile object # has a namelist() method that returns a list of stri
 ['spam.txt', 'cats/', 'cats/catnames.txt', 'cats/zophie.jpg']
spamInfo = exampleZip.getinfo('spam.txt')
spamInfo.file_size
   13908
spamInfo.compress_size
   3828
f'Compressed file is {round(spamInfo.file_size / spamInfo   .compress_size, 2)}x smaller!')
exampleZip.close()
```

3.3.1.6.2 Extracting from ZIP Files

The extractall() method for ZipFile objects extracts all the files and folders from a ZIP file into the current working directory.\
import zipfile, os \
  from pathlib import Path\

```
  p = Path.home() \
  exampleZip = zipfile.ZipFile(p / 'example.zip')\
❶ exampleZip.extractall()\
  exampleZip.close()
```

The extract() method for ZipFile objects will extract a single file from the ZIP file.
exampleZip.extract('spam.txt')
'C:\spam.txt'
exampleZip.extract('spam.txt', 'C:\some\new\folders')
'C:\some\new\folders\spam.txt'
exampleZip.close()

3.3.1.6.3 Creating and Adding to ZIP Files

To create your own compressed ZIP files, you must open the ZipFile object in write mode by passing 'w' as the second argument.
When you pass a path to the write() method of a ZipFile object, Python will compress the file at that path and add it into the ZIP file. The write() method’s first argument is a string of the filename to add. The second argument is the compression type parameter, which tells the computer what algorithm it should use to compress the files; you can always just set this value to zipfile.ZIP_DEFLATED.

In [ ]:

```
import zipfile
newZip = zipfile.ZipFile('delicious.zip', 'w')
newZip.write('C:\\Users\\91984\\Documents\\python\\hello.txt', compress_type=zipfile.ZIP_DE
newZip.close()
```

3.3.2 Project: Renaming Files with American-Style Dates to European-Style Dates

Here’s what the program does:\

- It searches all the filenames in the current working directory for American-style dates.
- When one is found, it renames the file with the month and day swapped to make it European-style.

  This means the code will need to do the following:\
- Create a regex that can identify the text pattern of American-style dates.
- Call os.listdir() to find all the files in the working directory.
- Loop over each filename, using the regex to check whether it has a date.
- If it has a date, rename the file with shutil.move().