

Python Programming

Módulo 3

Sistema de archivos

Sistema de archivos, OS

La librería estándar de Python provee varias funciones para trabajar con el sistema de archivos: listar, copiar, mover, eliminar, renombrar, etc. Están desperdigadas en dos módulos: `os` y `shutil`.

Dentro del módulo `os` nos interesan estas funciones:

- `os.listdir()`
- `os.getcwd()`
- `os.mkdir()`
- `os.path.exists()`
- `os.rename()`
- `os.remove()`
- `os.rmdir()`
- `os.system()`
- `os.name`

La función `os.listdir()` retorna una lista con los archivos y carpetas en una ruta en particular que se pasa como argumento.

```
>>> import os
>>> os.listdir("C:\\Users\\Anonimo\\AppData\\Local\\Programs\\Python\\Python39")
['DLLs', 'include', 'Lib', 'libs', 'LICENSE.txt', 'NEWS.txt', 'python.exe',
'python3.dll', 'python37.dll', 'pythonw.exe', 'Scripts', 'tcl', 'Tools',
'vcruntime140.dll']
```

Nota. Cuando indicamos una ruta como una cadena, por ejemplo, "C:/ruta/carpeta", en Windows se pueden usar tanto barras convencionales / como barras invertidas \. No obstante, puesto que la barra invertida se emplea para generar algunos caracteres especiales, como el de salto de línea `\n`, cuando queramos utilizarla como separador en una ruta deberemos indicar doble barra invertida `\\`. Así, lo correcto sería "C:\\ruta\\carpeta" y no "C:\ruta\carpeta".

Si omitimos el argumento, se toma por defecto la ruta desde donde se esté ejecutando el programa. Esta ruta lleva el nombre de *current working directory* (directorio actual de trabajo) y es retornada por la función `os.getcwd()`. Por ejemplo:

```
>>> import os
>>> os.getcwd()
'C:\Users\Usuario\Desktop'
```

De modo que las siguientes dos llamadas son equivalentes:

```
os.listdir(os.getcwd())
os.listdir()
```

La función `os.path.exists()` toma como argumento una ruta (sea una carpeta o un archivo) y retorna un booleano indicando si la misma existe (`True`) o no (`False`).

```
>>>os.path.exists("C:\\Users\\Anonimo\\AppData\\Local\\Programs\\Python\\Python39")
True

>>> os.path.exists("C:\\ruta\\inexistente.txt")
False
```

Mientras que `os.mkdir()` (***make directory***) toma como argumento la ruta hacia una carpeta y la crea.

¡Atención! las experiencias de borrar directorios o de borrar archivos hágase sobre archivos sin importancia y creados para estas experiencias. Ya que una vez borrados no terminan en la papelera. Son borrados definitivamente.

Por otro lado, `os.rmdir()` (***remove directory***) elimina una carpeta **vacía**, únicamente carpetas vacías. Para eliminar carpetas con contenido, más adelante veremos la función `shutil.rmtree()`.

```
>>> os.rmdir("c:\\ruta\\directorio")
```

Para remover archivos se utiliza `os.remove()`.

```
>>> os.remove("c:\\ruta\\archivo.ext")
```

La función `os.rename()` toma como primer argumento la ruta a un archivo cuyo nombre será cambiado por el segundo argumento.

```
>>> os.rename("c:\\ruta\\archivo.ext", "c:\\ruta\\nuevo_nombre.ext")
```

Si deseamos ejecutar alguna instrucción o comando del sistema operativo podemos usar `os.system()`

```
>>> os.system("cls") #cls comando borrar pantalla en SO Windows
```

Y por último `os`, también nos provee una manera de saber sobre que sistema operativo estamos ejecutando nuestro código.

```
>>> os.name # "nt" es Windows , "posix" es linux o iOS
```


Sistema de archivos, SHUTIL

Por otro lado, dentro del módulo `shutil` nos interesan las siguientes:

- `shutil.copy()`
- `shutil.move()`
- `shutil.rmtree()`

Vía `shutil.copy()` copiamos un archivo de una ubicación a otra, manteniendo el original.

```
>>> import shutil
```

```
>>> shutil.copy("C:\\ruta\\archivo.txt", "C:\\otra_ruta\\archivo.txt")
```

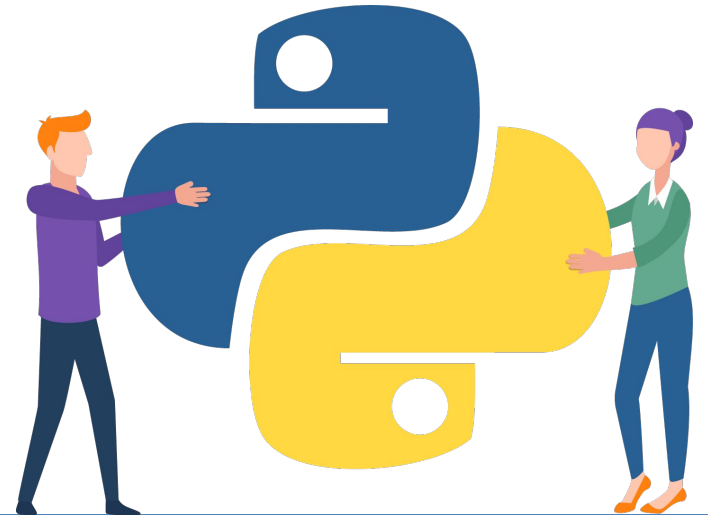
Del mismo modo opera `shutil.move()`, que mueve un archivo de una ubicación a otra, quitándolo de su ruta original.

```
>>> shutil.move("C:\\ruta\\archivo.txt",  
"C:\\otra_ruta\\archivo.txt")
```

Por último, para eliminar una carpeta que no esté vacía empleamos `shutil.rmtree()`:

¡Cuidado! Borra todos los archivos en la ruta especificada para siempre.

```
>>> shutil.rmtree("C:\\ruta\\carpeta_no_vacia")
```



¡Muchas gracias!

¡Sigamos trabajando!