

Manejo de Archivos (módulo OS)

El módulo **OS** en Python proporciona y expone los detalles y la funcionalidad del sistema operativo.

Proporciona funcionalidad independiente del sistema operativo.

Este módulo es parte de la biblioteca estándar de Python por lo que no es necesario instalar algo adicional para utilizarlo.

Para poder acceder a las funciones disponibles en el módulo os es necesario importarlo en nuestro código o en el interprete:

```
import os
```

Las funciones que veremos en esta entrada te serán útiles cuando quieras trabajar con archivos o carpetas.

Función chdir()

chdir() nos permite cambiar la ruta de trabajo. El parámetro que debemos colocar es la ruta a la que debemos cambiarnos.

```
os.chdir("C:\\Ejemplos-Python\\Seccion-1")
```

En caso de que la ruta no exista obtendremos un error de tipo FileNotFoundError.

Función getcwd()

getcwd() nos regresa una cadena indicando la ruta en la que nos encontramos trabajando.

```
os.getcwd()
```

```
>>> os.getcwd()
'C:\\Ejemplos-Python\\Seccion-1'
```

Función listdir()

listdir() regresa una lista con el nombre de los archivos y carpetas que se encuentran en la ruta especificada. El parámetro que debemos colocar es la ruta que queremos utilizar. Si no especificamos una ruta entonces lista el contenido de la ubicación actual de trabajo.

```
os.listdir()
os.listdir("C:\\Ejemplos-Python")
```

```
>>> os.listdir()
['borrarCarpeta.py', 'carpetaPrueba', 'crearCarpeta.py']
```

Función mkdir()

mkdir() crea una carpeta en una ruta que especifiquemos. El parámetro que colocamos es la ruta junto con el nombre de la carpeta, en caso de que no coloquemos una ruta, la carpeta se creará en la ubicación de trabajo actual.

```
os.mkdir("nuevaCarpeta")
```

En caso de que la carpeta ya exista obtendremos un error de tipo FileExistsError.

Función remove()

remove() elimina un archivo. El parámetro que debemos colocar es el nombre del archivo que deseamos eliminar y se puede especificar junto con su ruta, de lo contrario buscará el archivo en la ruta de trabajo actual.

```
os.remove("archivoPrueba.py")
os.remove("C:\\Ejemplos-Python\\Seccion-1\\archivoPrueba.py")
```

En caso de que no se encuentre el archivo se generará un error tipo FileNotFoundError.

Función removedirs()

removedirs() borrará carpetas recursivamente. El parametro que debemos pasarle es la ruta que involucra las carpetas que se deben borrar. Borrará todas las carpetas de esa ruta **mientras estén vacías**.

```
os.removedirs("carpetaPrueba\\Nivel 1\\Nivel 2")
```

Siguiendo la ruta de la línea anterior, primero borraría la carpeta Nivel 2, luego la carpeta Nivel 1 y finalmente la carpeta carpetaPrueba. Este proceso se detendría sin marcar algún error si la carpeta que esta por borrar tiene algún otro archivo o carpeta. Solo regresaría un error si la ruta especificada tiene algún nombre incorrecto, en ese caso el error seria de tipo FileNotFoundError.

Función rename()

rename() renombra archivos o carpetas. Los parámetros que se deben colocar son el nombre actual separado por una coma del nombre que queremos. Se puede incluir la ruta del archivo o carpeta, pero debemos colocarla en ambos parámetro, de lo contrario el archivo o carpeta será reubicado.

```
os.rename("carpetaPrueba", "cambioNombre")
os.rename("archivoPrueba.py", "cambioNombre.py")
```

Si el nombre que queremos utilizar ya existe entonces recibiremos un error de tipo FileExistsError.

Función replace()

replace() renombra archivos. Los parámetros que se deben pasarle también son el nombre actual separado por una coma del nombre que queremos. La diferencia con respecto a rename() es que si el archivo ya existe, entonces lo reemplaza sin generar ningún error. Ten cuidado con el contenido del archivo que existía previamente porque será reemplazado.

```
os.replace("archivoPrueba.py", "cambioNombre.py")
```

Función rmdir()

rmdir() elimina una carpeta. El parámetro que debemos colocarle es el nombre de la carpeta y podemos incluir la ruta, en caso de que no se coloque la ruta se buscará la carpeta indicada en la ruta de trabajo actual.

```
os.rmdir("carpetaPrueba")
```

Esta función solo **elimina carpetas vacías**, si la carpeta no está vacía recibiremos un error tipo OSError. Por otro lado si la carpeta no existe recibiremos un error tipo FileNotFoundError.

Función scandir()

scandir() regresa un iterador con el contenido de la carpeta ya sean archivos o carpetas. A diferencia de listdir() que regresa una lista con los nombres, scandir() nos proporciona más información. Es posible utilizar el atributo name o los métodos is_file o is_dir que nos permiten conocer más detalles del contenido de la carpeta.

```
for elem in os.scandir():  
    print(elem.name)
```

Función walk()

walk() nos permite recorrer un árbol de carpetas. El parámetro que debemos colocar es la ruta en la que queremos empezar el recorrido. Si queremos que tome la ruta actual de trabajo podemos colocar «.». Esta función regresa una tupla de 3 tuplas con la ruta, el nombre de la carpeta y el nombre de archivo.

```
for elem in os.walk("."):   
    print(elem)
```