



Curso de Python

8 – Fichero

Ramón Invarato Menéndez

Ricardo Moya García



Ficheros y Directorios: Ruta (Path)



Código

```
ruta_fichero = "C:\\Usuarios\\Paco\\Siembra2020.txt"
```

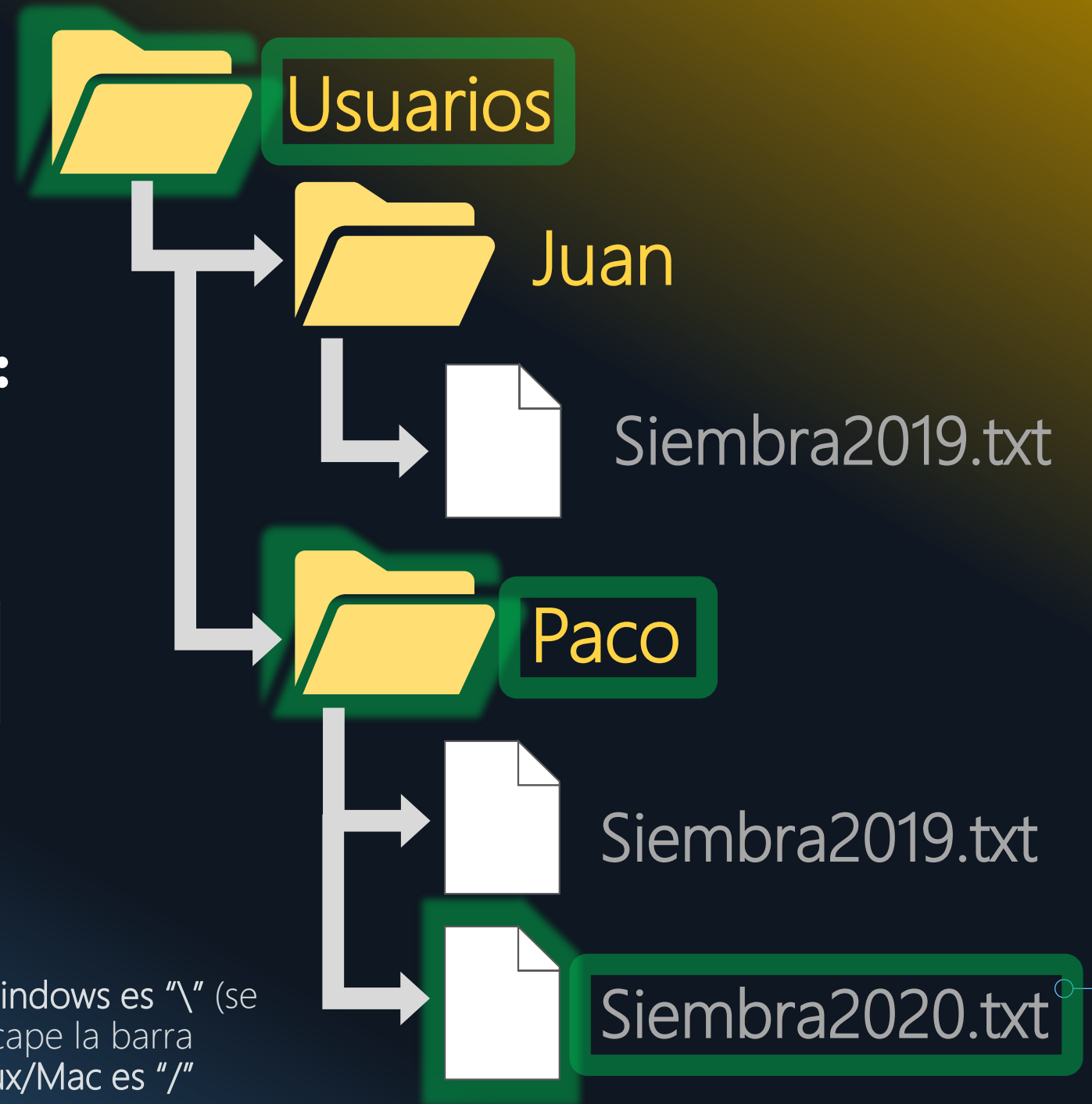


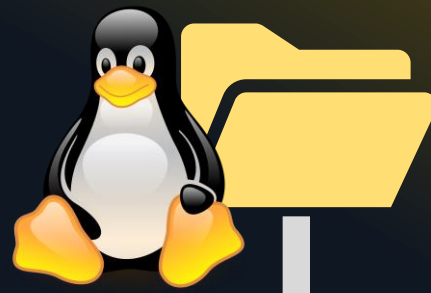
Código

```
ruta_fichero = "Usuarios/Paco/Siembra2020.txt"
```

2

- La separación de los directorios en Windows es "\\" (se pone doble barra "\\" para que se escape la barra dentro del string de Python) y en Linux/Mac es "/"





Usuarios

Paco



Siembra2020.txt

Ficheros: Abrir y cerrar

- Un fichero se abre con una ruta hasta el fichero que se pasa a la función `open()`
- Un fichero se puede abrir de varios modos:
 - `r`: solo lectura
 - `w`: escritura; borra todo y sobrescribe el contenido
 - `a`: añadir (append); como el de escritura pero añade al final del fichero
- Un fichero abierto se tiene que cerrar cuando se termina de utilizar
- Es recomendable controlar las excepciones `IOException` y `FileNotFoundError`

Modo	De palabra	Significado
r	Read	Solo lectura
w	Write	Sobrescribir contenido
a	Append	Añadir al final

Código

```
ruta_fichero = "Usuarios/Paco/Siembra2020.txt"

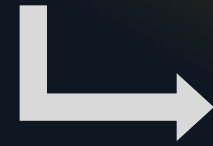
fichero = open(ruta_fichero, "r")

# Hacer algo con el fichero abierto

fichero.close()
```



Usuarios



Paco



Siembra2020.txt

Ficheros: Abrir y cerrar (with)

with open(ruta, modo) as fichero:

Hacer algo con el fichero abierto

- Para simplificar la tarea de abrir un fichero y cerrarlo se puede utilizar la palabra reservada "with"

Código

```
ruta_fichero = "Usuarios/Paco/Siembra2020.txt"

fichero = open(ruta_fichero, "r")

# Hacer algo con el fichero abierto

fichero.close()
```

Código

```
ruta_fichero = "Usuarios/Paco/Siembra2020.txt"

with open(ruta_fichero, "r") as fichero:
    # Hacer algo con el fichero abierto
```



Usuarios



Paco

Siembra2020.txt

Leer desde fichero todas las líneas

`listado_lineas = fichero.readlines()`

- `readlines()`: obtiene un listado de todas las líneas. Es poco útil si el fichero es muy grande

Código

```
ruta_fichero = "Usuarios/Paco/Siembra2020.txt"

with open(ruta_fichero, "r") as fichero:

    for linea in fichero.readlines():
        print(linea)
```

print

Consola

```
Manzanas, 8
Uvas, 10
Lechugas, 15
Cerezas, 3
```

```
Manzanas, 8
Uvas, 10
Lechugas, 15
Cerezas, 3
```



Usuarios



Paco

Siembra2020.txt

Leer desde fichero línea a línea

`linea = fichero.readline()`

- `readline()`: cada vez que se llama devuelve una línea. Se recomienda su uso pues lee una línea, se puede procesar y descartar

Manzanas, 8
Uvas, 10
Lechugas, 15
Cerezas, 3

Código

```
ruta_fichero = "Usuarios/Paco/Siembra2020.txt"

with open(ruta_fichero, "r") as fichero:

    linea = fichero.readline()
    while linea:
        print(linea.strip())
        linea = fichero.readline()
```

print

Consola

Manzanas, 8
Uvas, 10
Lechugas, 15
Cerezas, 3



Usuarios



Paco

Siembra2020.txt

Escribir a fichero todas las líneas

`fichero.writelines(iterable)`

Código

```
mi_iterable = ["Kiwis, 7\n", "Melocotones 12\n"]  
  
ruta_fichero = "Usuarios/Paco/Siembra2020.txt"  
  
with open(ruta_fichero , "a") as fichero:  
  
    fichero.writelines(mi_iterable)
```

Manzanas, 8
Uvas, 10
Lechugas, 15
Cerezas, 3
Kiwis, 7
Melocotones, 12

- `writelines(iterable)`: Guarda un listado en fichero. Se suele utilizar `write` para añadir el salto de línea den el fichero



Usuarios



Paco

Siembra2020.txt

Escribir a fichero línea a línea

`fichero.writeline("línea \n")`

Código

```
mi_iterable = ["Kiwis, 7", "Melocotones 12"]  
  
ruta_fichero = "Usuarios/Paco/Siembra2020.txt"  
  
with open(ruta_fichero, "a") as fichero:  
    for linea in mi_iterable:  
        fichero.write("{}\n".format(linea))
```

Manzanas, 8
Uvas, 10
Lechugas, 15
Cerezas, 3
Kiwis, 7
Melocotones, 12

- `writeline("línea\n")`: cada vez que se llama escribe una línea en al final del fichero. Si se quiere ver el fichero con saltos de línea, hay que poner "\n" al final del fichero

Pseudo-Código

```
obj_ruta = Path("/ruta/hasta/directorio/o/fichero")
```

Importar

```
from pathlib import Path
```

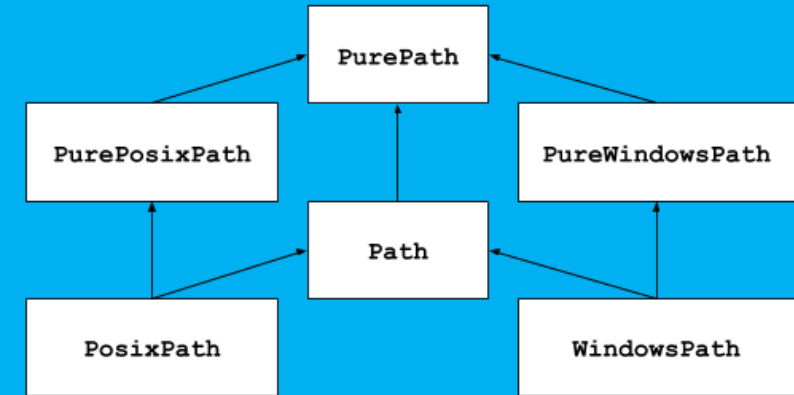
Path (Ruta)

- Para crear rutas independientemente del sistema y transformarlas de Windows a Linux y viceversa
- Se pueden concatenar rutas sumando dos objetos Path con "+" y dará otro Path resultante
- Se puede ver si la ruta existe con "exists()"
- Se pueden seleccionar varios ficheros con "glob(patron)"
- Se puede crear un fichero con "touch()" o un directorio con "mkdir()"
- Se puede eliminar un fichero con "unlink()" o un directorio con "rmdir()"
- Se puede obtener las partes del fichero con las variables: name, suffix, parent, etc

9

Código

```
obj_ruta_resultante.unlink()
```



Código

```
obj_ruta_dir = Path("C:\\ruta\\directorio\\")  
obj_ruta_fichero = Path("directorio2", "fichero2.txt")
```

Código

```
existe = obj_ruta_dir.exists()
```

print

Consola

True

Código

```
obj_ruta_resultante = obj_ruta_dir + obj_ruta_fichero
```

Consola

C:\ruta\directorio\directorio2\fichero2.txt



Usuarios



Paco

Siembra2020.txt

Path (Ruta)

```
ruta = Path("ruta/entera")
```

```
ruta = Path("ruta", "por", "partes")
```

- Para crear rutas independientemente del sistema y transformarlas de Windows a Linux y viceversa
- Se pueden concatenar rutas sumando dos objetos Path con "+" y dará otro Path resultante

Importar

```
from pathlib import Path
```

```
# O bien la ruta entera:
```

```
ruta_siembra_paco = Path("Usuarios/Paco/Siembra2020.txt")
```

```
# O la ruta por partes:
```

```
ruta_siembra_paco = Path("Usuarios", "Paco", "Siembra2020.txt")
```

```
# Se puede usar directamente el objeto Path en open:
```

```
fichero = open(ruta_siembra_paco, "r")
```

```
# Se puede obtener la ruta en formato string:
```

```
ruta_str = str(ruta_siembra_paco)
```



Usuarios



Paco



Siembra2020.txt



Usuarios



Juan



Siembra2019.txt

Path (Ruta)

```
ruta = Path("ruta/entera")
```

```
ruta = Path("ruta", "por", "partes")
```

- Se pueden **concatenar rutas** sumando dos objetos Path con "+" y dará otro Path resultante

Importar

```
from pathlib import Path  
  
ruta_general = Path("Usuarios")  
  
ruta_siembra_paco = ruta_general + Path("Paco", "Siembra2020.txt")  
  
ruta_siembra_juan = ruta_general + Path("Juan", "Siembra2019.txt")
```



 Usuarios /  Paco / Siembra2020.txt

Path (Ruta): ¿Existe?

`ruta.exists()`

- Se puede ver si la ruta existe con "exists()"

Importar

```
from pathlib import Path  
  
ruta_siembra_paco = Path("Usuarios/Paco/Siembra2020.txt")  
  
existe = ruta_siembra_paco.exists()  
  
existe
```

print

Consola

True



Usuarios



Paco

Siembra2018.txt
Siembra2019.txt
Siembra2020.txt
Memoria.docx

Path (Ruta): seleccionar varios

`ruta.glob(patron)`

- Se pueden seleccionar varios ficheros con "glob(patron)"

Importar

```
from pathlib import Path  
  
ruta_paco = Path("Usuarios/Paco")  
  
lista_rutas = ruta_paco.glob("*.txt")  
  
for obj_ruta in lista_rutas:  
    print(obj_ruta)
```

print

Consola

```
Usuarios/Paco/Siembra2018.txt  
Usuarios/Paco/Siembra2019.txt  
Usuarios/Paco/Siembra2020.txt
```



Usuarios



Paco

Siembra2020.txt

Path (Ruta): Padre

`ruta.parent`

- Se puede obtener las partes del fichero con las variables: name, suffix, parent, etc

Importar

```
from pathlib import Path  
ruta_siembra_paco = Path("Usuarios/Paco/Siembra2020.txt")  
ruta_paco = ruta_siembra_paco.parent  
ruta_paco
```

print

Consola

Usuarios/Paco



Usuarios



Paco

Siembra2020.txt

Path (Ruta): Nombre

`ruta.name`

Importar

```
from pathlib import Path  
ruta_siembra_paco = Path("Usuarios/Paco/Siembra2020.txt")  
nombre_fichero = ruta_siembra_paco.name  
nombre_fichero
```

print

Consola

Siembra2020.txt



Usuarios



Paco

Siembra2020.txt



María

MiFichero.txt

Path (Ruta): Crear

`ruta.touch()` #Crea un fichero

`ruta.mkdir()` #Crea un directorio

- Se puede crear un fichero con "touch()" o un directorio con "mkdir()"

Importar

```
from pathlib import Path

ruta_maria = Path("Usuarios/Maria")

ruta_maria.mkdir()

ruta_fichero_maria = ruta_maria + Path("MiFichero.txt")

ruta_fichero_maria.touch()
```




Usuarios



Paco

Siembra2020.txt



María

MiFichero.txt

Path (Ruta): eliminar

`ruta.unlink()` #Elimina un fichero

`ruta.rmdir()` #Elimina un directorio

- Se puede eliminar un fichero con “unlink()” o un directorio con “rmdir()”

Importar

```
from pathlib import Path

ruta_fichero_maria = Path("Usuarios/Maria/MiFichero.txt")

ruta_fichero_maria.unlink()

ruta_maría = ruta_fichero_maria .parent

ruta_maría.mkdir()
```

Resumen de métodos de Path

Comprueba si existe un fichero o un directorio

exists()

Pregunta si es un fichero

is_file()

Pregunta si es un directorio

is_dir()

Crea un nuevo directorio

mkdir()

Elimina un directorio

rmdir()

Crea un fichero

touch()

Elimina un fichero

unlink()

Renombra el fichero o directorio

rename("nuevo_nombre")

Selecciona varios ficheros en un directorio
(devuelve un listado de objetos Path)

*glob("**/*.patron")*

Obtiene la extensión del fichero

suffix

Obtiene el nombre del fichero

name

Obtiene una tupla con las partes de la ruta

parts

Obtienen el Path padre

parent

Obtiene un listado con todos los Paths padres

parents

¡GRACIAS!



Web: <https://jarroba.com/>

Ramón Invarato Menéndez

Linked-in

<https://www.linkedin.com/in/rinvarato/>

Github

<https://github.com/Invarato>

Ricardo Moya García

Linked-in

<https://www.linkedin.com/in/phdricardomoya>

Github

<https://github.com/RicardoMoya>