

## Ficheros

En Python, para abrir un fichero usaremos la función **open**, que recibe el nombre del archivo a abrir. Por defecto, si no indicamos nada, el fichero se abre en modo lectura.

### Open en modo lectura

```
fichero = open("./datos/cuna.txt")
```

La función **open** abrirá el fichero con el nombre indicado, en este caso el fichero **cuna.txt**. Si no tiene éxito, se lanzará una excepción. Si se ha podido abrir el fichero correctamente, la variable **fichero** nos permitirá manipularlo.

Recordar que existen funciones de comunicación con el sistema operativo:

```
%pwd # función mágica que devuelve el directorio actual
```

```
'C:\\Users\\Yoli\\Documents\\YOLANDA\\YOLANDA\\Cunef_1718\\Material curso 2017-2018\\Tema_4'
```

Si la función **open** te da error y no se abre el fichero, es posible que dicho fichero no se encuentre en la ruta que le has indicado. Otras función mágica que te permite explorar tu directorio de trabajo es **%ls**.

```
%ls
```

```
El volumen de la unidad C es Windows7_OS  
El número de serie del volumen es: 5CD9-3939
```

```
Directorio de C:\\Users\\Yoli\\Documents\\YOLANDA\\YOLANDA\\Cunef_1718\\Material curso 2017-2018\\Tema_4
```

```
21/09/2017 11:23 <DIR> .  
21/09/2017 11:23 <DIR> ..  
20/09/2017 10:23 <DIR> .ipynb_checkpoints  
24/02/2015 17:59      2.244 custom.css  
20/09/2017 10:23 <DIR> datos  
20/09/2017 10:23 <DIR> ejercicios  
20/09/2017 10:23 <DIR> images  
14/11/2016 13:50    286.397 s2_cica_funciones_02.html  
21/09/2017 11:05    276.109 t4_files.html  
21/09/2017 11:05     9.147 t4_files.ipynb  
21/09/2017 11:23    278.859 t4_funciones_02.html  
21/09/2017 11:23     8.784 t4_funciones_02.ipynb  
        6 archivos      861.540 bytes  
        6 dirs  15.325.134.848 bytes libres
```

La operación más sencilla a realizar sobre un archivo es leer su contenido. Para procesarlo línea por línea, es posible hacerlo de la siguiente forma:

```
fichero = open("./datos/cuna.txt")
for linea in fichero:
    print(linea)
```

Ya te vemos dormida.

Tu barca es de madera por la orilla.

Blanca princesa de nunca.

Duerme por la noche oscura.

Cuerpo y tierra de nieve.

Duerme por el alba, duerme.

Ya te alejas dormida.

Es posible, además, obtener todas las líneas del archivo utilizando una sola llamada a la función **readlines**. Esta función devuelve una lista de líneas.

```
fichero = open("./datos/cuna.txt")
lineas = fichero.readlines()
lineas
```

```
['Ya te vemos dormida. \n',
 'Tu barca es de madera por la orilla. \n',
 'Blanca princesa de nunca. \n',
 'Duerme por la noche oscura.\n',
 'Cuerpo y tierra de nieve. \n',
 'Duerme por el alba, duerme.\n',
 'Ya te alejas dormida. ']
```

En este caso, la variable `lineas` tendrá una lista de cadenas con todas las líneas del fichero.

Es importante tener en cuenta que cuando se utilizan funciones como **archivo.readlines()**, se está cargando en memoria el fichero completo. Siempre que una instrucción cargue un fichero completo debe tenerse cuidado de utilizarla sólo con ficheros pequeños, ya que de otro modo podría agotarse la memoria.

En el siguiente ejemplo, se accede a la primera línea del fichero y posteriormente se eliminan los saltos de línea mediante la función de cadenas `rstrip`.

```
lineas[0].rstrip()      # rstrip para eliminar saltos de línea
'Ya te vemos dormida.'
```

## Open en modo escritura (w)

Si queremos abrir un fichero en modo escritura, hay que indicarlo como segundo parámetro de la función **open**.

```
arc_write = open('./datos/nuevo.txt', 'w')
```

```
# Copia en el nuevo fichero solo las líneas pares del fichero arc
for i, line in enumerate(lineas):
    if i%2 == 0:
        arc_write.write(str(i) + ' ' + line)
    else:
        pass
```

Al terminar de trabajar con un fichero, es recomendable cerrarlo ya que lo que se haya escrito no se guardará realmente hasta no cerrar el fichero.

```
arc_write.close()      # cerramos el fichero
```

```
open('./datos/nuevo.txt').readlines()
```

```
['0 Ya te vemos dormida. \n',  
'2 Blanca princesa de nunca. \n',  
'4 Cuerpo y tierra de nieve. \n',  
'6 Ya te alejas dormida. ']
```

## Open en modo escritura posicionándose al final del archivo (a)

En este caso se crea el fichero, si no existe, pero en caso de que exista se posiciona al final, manteniendo el contenido original.

```
open('./datos/nuevo.txt', 'a').write('\nEste es el final')
```

```
17
```

```
open('./datos/nuevo.txt').readlines()
```

```
['0 Ya te vemos dormida. \n',  
'2 Blanca princesa de nunca. \n',  
'4 Cuerpo y tierra de nieve. \n',  
'6 Ya te alejas dormida. \n',  
'Este es el final']
```

Hemos visto una forma de abrir ficheros para recuperar la información. Más adelante veremos cómo se trabaja en Python con otro tipo de ficheros como son excel, json, csv, xml, etc.

---

Content on this site is licensed under a Creative Commons Attribution 4.0 International license.

