



Persistencia de datos

Acceso a ficheros

Índice

- Manipulación de ficheros en Python
- La función `open()`
- El método `close()`
- Escritura en un fichero
- Lectura de un fichero
- Otros métodos para trabajar con ficheros
- El módulo `os`

Manipulación de ficheros en Python

- Dispone de instrucciones para manipulación de ficheros de texto.
- Es posible escribir texto en un fichero en modo sobrescritura o añadiendo al contenido actual
- Permite recorrer el fichero línea a línea o recuperar el contenido completo en una lista

La funcion open()

➤ Devuelve un objeto para poder operar contra un fichero.

➤ Formato:

```
open(dir_fichero, modo)
```

➤ El segundo parámetro indica el modo de apertura del fichero. Entre sus valores:

- "r". Abre el fichero para operaciones de lectura
- "w". Abre el fichero para operaciones de escritura, sobrescribiendo el contenido del mismo
- "a". Abre el fichero para escritura, añadiendo nuevo contenido al existente

El método close()

- Después de operar con un fichero, se debe llamar al método close() sobre el objeto para cerrar el mismo

```
objFichero=open(dir_fichero, modo)
#operaciones de lectura o escritura
objFichero.close()
```


Escritura en un fichero

- Una vez abierto el fichero como escritura, se puede utilizar el método `write()` para escribir sobre el mismo. En función del modo de apertura, se sobrescribirá el contenido del fichero o se añade al existente.
- Si el fichero no existe se creará

```
f=open("datos.txt","a")  
f.write("nuevo texto\n")  
f.write("otro texto más\n")  
f.close()
```


Lectura de un fichero

- Una vez abierto el fichero como lectura, se puede recuperar el contenido del mismo utilizando los métodos:
 - `readLine()`. Devuelve la línea actual, situando el puntero en la siguiente línea. Si no hay ninguna línea devolverá cadena vacía
 - `readLines()`. Devuelve una lista con todas las líneas de texto del fichero
- Si el fichero no existe se produce una excepción

```
f=open("datos.txt","r")
cad=f.readline()
while cad!="":
    print(cad)
    cad=f.readline()
```


Otros métodos para trabajar con ficheros

- Una vez abierto el fichero, además de los métodos de lectura y escritura podemos aplicar los siguientes métodos sobre el objeto:
 - `tell()`. Devuelve la posición actual del puntero. Inicialmente será 0, tras una llamada a `readLine()`, será la posición del carácter siguiente de lectura.
 - `seek (desp, pos)`. Sirve para desplazar el puntero un número de bytes indicado en primer parámetro, a partir de la posición indicada en el segundo. El segundo parámetro puede valer 0 (posición inicial), 1 (posición actual) o 2 (posición último carácter)

```
f=open("datos.txt","r")  
f.readLine()  
f.seek(0,0) #se posiciona de nuevo al principio del fichero
```


El módulo os

- Proporciona una serie de métodos para operar con ficheros y directorios:
 - `rename(nombre_actual, nuevo_nombre)`. Renombra un fichero
 - `remove(fichero)`. Elimina el fichero cuyo nombre se indica
 - `getcwd()`. Muestra el directorio actual

```
import os
print(os.getcwd())
os.rename("ejemplo.txt", "pruebas.txt")
os.remove("pruebas.txt")
```