Sí, antes o después cualquier programador se tiene que enfrentar a la tarea de escribir en un fichero. Y en Python, como en cualquier otro lenguaje, esto es posible. Por eso, este tutorial sobre cómo escribir en un fichero no podía faltar en la sección «Desayuno con Python».

**El objeto File y sus tipos**

El objeto File en Python representa un fichero del sistema operativo. Los ficheros, sirven para guardar datos en disco que podrán ser leídos posteriormente (recuerda que ya vimos [cómo leer ficheros en Python](https://j2logo.com/leer-fichero-linea-a-linea-python/)).

Normalmente, existen dos tipos de ficheros: **ficheros de texto** y **ficheros binarios**. Un fichero de texto contiene caracteres que son legibles por el ser humano y están guardados con una codificación (ASCII, UTF-8, …). Por el contrario, un fichero binario está compuesto por un flujo de bytes y solo tienen sentido para los programas o aplicaciones para los que son creados. Un ejemplo de este tipo de archivos son las imágenes o la música.

Cuando se trabaja con un fichero de texto, hay que tener en cuenta que este se estructura como una secuencia de líneas. Cada una de estas líneas acaba con un carácter especial conocido como **EOL** (fin de línea). En función del sistema operativo, este carácter puede variar. Puede ser \n (Unix) o \r\n (Windows). No obstante, en Python, cuando escribimos o leemos el carácter \n en un fichero de texto, el propio lenguaje se encarga de convertir dicho carácter al correspondiente por el sistema operativo, por lo que es algo a prever si nuestro código se va a ejecutar en diferentes sistemas.

**La función open() y sus modos**

En Python, para escribir en un fichero o simplemente leer su contenido utilizaremos la función predefinida open(). Al invocar a esta, se crea un objeto de tipo File.

Lo más común es llamar a la función open() con dos parámetros:

* El primero de ellos es la ruta del fichero (en la que está o donde se va a crear).
* El segundo es el modo en el que se abre el fichero: lectura, escritura, …

Por defecto, cuando se invoca a la función open(path, modo), el fichero **se abre en modo texto**. Si quisiéramos abrir un fichero en forma binaria, habría que añadir el carácter b al parámetro modo.

Los diferentes modos en los que se puede abrir un fichero son:

|  |  |
| --- | --- |
| r | Solo lectura. El fichero solo se puede leer. Es el modo por defecto si no se indica. |
| w | Solo escritura. En el fichero solo se puede escribir. Si ya existe el fichero, machaca su contenido. |
| a | Adición. En el fichero solo se puede escribir. Si ya existe el fichero, todo lo que se escriba se añadirá al final del mismo. |
| x | Como ‘w’ pero si existe el fichero lanza una excepción. |
| r+ | Lectura y escritura. El fichero se puede leer y escribir. |

Como te he indicado, todos estos modos abren el fichero en modo texto. Su versión correspondiente para abrir el fichero en modo binario sería rb, wb, ab, xb, rb+.

**Chuleta para escribir en un fichero**

Bueno, ahora que ya conocemos los principales conceptos del objeto File y cómo trata Python a los ficheros, es hora de ver cómo escribir en un fichero en Python.

La estructura para abrir un fichero en modo escritura es la siguiente:

f = open('mi\_fichero', 'w')

try:

# Procesamiento para escribir en el fichero

finally:

f.close()

Aunque personalmente prefiero usar la sentencia with, ya que ella se encarga de cerrar el fichero y liberar sus recursos (incluso si ocurre cualquier error):

with open('mi\_fichero', 'w') as f:

# Procesamiento del fichero

El objeto f de  los ejemplos anteriores representa un fichero y este pone a nuestra disposición el método write() para escribir cualquier contenido sobre el fichero.

**Escribir un fichero de texto**

Con todo lo anterior, si quisiéramos escribir un fichero de texto haríamos lo siguiente:

with open('mi\_fichero', 'w') as f:

f.write('Hola mundo\n')

**Escribir un fichero binario**

Para escribir un fichero binario, simplemente añadimos el carácter b al parámetro modo y escribimos bytes:

with open('mi\_fichero', 'wb') as f:

f.write(b'0x28')

## Conclusión

Bueno, este post es muy importante tenerlo en cuenta para cualquier pythonista. En él hemos repasado el objeto File de Python, los distintos tipos de ficheros que existen y cómo escribir en un fichero. Espero que te haya gustado.