

Ficheros

Python puede leer y escribir ficheros de texto o binarios.

Trabajar con ficheros: **with ... as ...**

Las palabras reservadas **with** y **as** se introdujeron en Python 2.6 (publicado en septiembre de 2006, aunque estaban disponibles en Python 2.5 en el módulo `__future__`) para facilitar, entre otros, la forma de trabajar con ficheros.

Su sintaxis general es la siguiente:

```
with EXPRESION as VARIABLE:  
    BLOQUE DE INSTRUCCIONES
```

En el caso de los ficheros, la expresión es una llamada a [la función `open\(\)`](#) y la variable es la conexión con el fichero:

```
with open("FICHERO") as fichero:  
    BLOQUE DE INSTRUCCIONES
```

La función **open** puede tener varios argumentos. Los más importantes son

```
with open("FICHERO", mode="MODO", encoding="CODIFICACION") as fichero:  
    BLOQUE DE INSTRUCCIONES
```

- **"FICHERO"** es la ruta absoluta o relativa hasta el fichero.
- **"MODO"** indica si el fichero se abre para leer, escribir o ambas cosas y si se trata de un fichero de texto o binario. El modo predeterminado es lectura en modo texto.
- **"CODIFICACION"** indica el juego de caracteres del fichero: **"utf-8"** (UTF-8), **"cp1252"** (ASCII para europa occidental), [etc.](#)

En estos apuntes se trabaja únicamente con ficheros de texto y se utiliza siempre el juego de caracteres UTF-8.

Si el sistema operativo utiliza de forma predeterminada el juego de caracteres UTF-8, no haría falta indicar el juego de caracteres al abrir el fichero. Para saber cuál es el juego de caracteres predeterminado se puede ejecutar en IDLE las instrucciones siguientes (en el ejemplo, el juego de caracteres es [CP-1252](#), que utiliza Windows):

```
>>> import locale  
>>> print(locale.getpreferredencoding(False))  
cp1512
```

De todas formas, aunque el juego de caracteres sea UTF-8, se recomienda indicar siempre explícitamente el juego de caracteres al abrir un fichero, ya que los programas escritos en un sistema operativo se pueden acabar ejecutando en otro.

Modos de escritura

Los modos de escritura son:

- "x": únicamente crear el fichero (da error si ya existe el fichero)
- "w": escribir (crea el fichero si no existe y borra el contenido anterior del fichero)
- "a": añadir (crea el fichero si no existe, no borra el contenido existente y escribe al final del fichero)

Se puede escribir en el fichero

- con la función `print()` añadiendo el argumento `file=fichero`, donde `fichero` es la variable utilizada en la expresión `with ... as ...`
- con el método `write` sobre el objeto `fichero`, donde `fichero` es la variable utilizada en la expresión `with ... as ...`

La función `print()` añade un salto de línea al final de la cadena añadida al fichero, pero el método `write` no lo hace, por lo que habrá que añadirlo explícitamente.

Los ejemplos siguientes muestran la diferencia entre cada uno de los modos de escritura (usando la función `print()` o el método `write`):

- [w] Cada vez que se ejecuta este programa, se crea nuevamente el fichero, por lo que este sólo contiene la palabra `Hola`.

```
ruta = "prueba.txt"
with open(ruta, mode="w", encoding="utf-8") as fichero:
    print("Hola", file=fichero)
```

```
ruta = "prueba.txt"
with open(ruta, mode="w", encoding="utf-8") as fichero:
    fichero.write("Hola\n")
```

- [x] Al ejecutar por segunda vez este programa (o a la primera si el fichero ya existe), se genera un error.

```
ruta = "prueba.txt"
with open(ruta, mode="x", encoding="utf-8") as fichero:
    print("Hola", file=fichero)
```

```
ruta = "prueba.txt"
with open(ruta, mode="x", encoding="utf-8") as fichero:
    fichero.write("Hola\n")
```

- [a] Cada vez que se ejecuta este programa, se añade al fichero una línea con la palabra `Hola` (si el fichero no existe, se crea).

```
ruta = "prueba.txt"
with open(ruta, mode="a", encoding="utf-8") as fichero:
    print("Hola", file=fichero)
```

```
ruta = "prueba.txt"
with open(ruta, mode="a", encoding="utf-8") as fichero:
    fichero.write("Hola\n")
```

Modos de lectura

Los modos de lectura son:

- "r": únicamente leer el fichero (da error si no existe el fichero, lee desde el principio y es posible desplazarse)
- "r+": leer y escribir (da error si no existe el fichero, lee desde el principio y es posible desplazarse, pero sólo escribe al final del fichero)



Faltan ejemplos de lectura y desplazamiento.

Abrir automáticamente en el navegador

Si el fichero que se ha creado con Python es un página web, puede ser de utilidad que el fichero se abra automáticamente en el navegador para ver el resultado inmediatamente.

Para ello se puede utilizar la función `open()` del [módulo webbrowser](#) de la biblioteca estándar, que abre el fichero indicado con la aplicación asociada en el sistema operativo.

```
1 import webbrowser
2
3 ruta = "prueba.html"
4
5 with open(ruta, mode="w", encoding="utf-8") as fichero:
6     print("<!DOCTYPE html>", file=fichero)
7     print("<html lang='es'>", file=fichero)
8     print("<head>", file=fichero)
9     print("  <meta charset='utf-8' />", file=fichero)
10    print("  <title>HTML 5</title>", file=fichero)
11    print("  <meta name='viewport' content='width=device-width, initial-
12 scale=1.0' />", file=fichero)
13    print("</head>", file=fichero)
14    print("", file=fichero)
15    print("<body>", file=fichero)
16    print("  <p>Esta página es HTML 5 válido.</p>", file=fichero)
17    print("</body>", file=fichero)
18    print("</html>", file=fichero)
19
20 webbrowser.open(ruta)
```

Es fundamental que el juego de caracteres en el que se guarda el fichero (línea 5 en el ejemplo anterior) sea el mismo que el que se declara en la etiqueta `<meta />` (línea 9 en el ejemplo anterior). Si no es así, el navegador tendrá problemas al mostrar caracteres "no ingleses" (acentos, eñes, etc.). En el ejemplo anterior se utiliza UTF-8, el juego de caracteres recomendado para páginas web.

Para saber más

- [PEP 343: The "with" Statement](#)
- Python docs: [La función open\(\)](#)

Última modificación de esta página: 11 de abril de 2018



Esta página forma parte del curso [Introducción a la programación con Python](#) por [Bartolomé Sintes Marco](#)

que se distribuye bajo una [Licencia Creative Commons Reconocimiento-CompartirIgual 4.0 Internacional \(CC BY-SA 4.0\)](#).