



UF3.7 Objeto File de Python



Centro Profesional
Universidad Europea de Madrid
LAUREATE INTERNATIONAL UNIVERSITIES





Contenidos

- Trabajar con archivos
- Objeto File
- Modos de aperture
- Métodos del objeto File
- Propiedades del objeto File



Trabajar con archivos

Python nos permite trabajar en dos niveles diferentes con respecto al sistema de archivos y directorios.

Uno de ellos, es a través del **módulo os**, que como su nombre indica, nos facilita el trabajo con todo el sistema de archivos y directorios, a nivel del Sistema Operativo.

El segundo nivel -más simple-, es el que nos permite trabajar con archivos, manipulando su lectura y escritura a nivel de la aplicación y tratando a cada archivo como un objeto.



Objeto File

En Python podemos manipular una variable como un objeto File asignándole como valor un archivo.

Para asignar a una variable un valor de tipo file, solo es necesario recurrir a la función integrada **open()**, la cuál está destinada a la apertura de un archivo.

La función integrada `open()`, recibe dos parámetros:

- El primero es la ruta hacia el archivo que se desea abrir
- Y el segundo, el modo en el cual abrirlo



Modos de apertura

Cada vez que abrimos un archivo estamos creando un puntero, el cuál se posicionará dentro del archivo en un lugar determinado (al comienzo o al final) y este puntero podrá moverse dentro de ese archivo. Este puntero, se creará dependiendo del modo de apertura indicado.

Modos de apertura posibles y posición del puntero.

Cod.	Modo de apertura	Ubicación del puntero
r	Solo lectura	Al inicio del archivo
rb	Solo lectura en modo binario	Al inicio del archivo
r+	Lectura y escritura	Al inicio del archivo
rb+	Lectura y escritura en modo binario	Al inicio del archivo
w	Solo escritura. Sobreescibe el archivo si existe. Crea el archivo si no existe	Al inicio del archivo
wb	Solo escritura en modo binario. Sobreescibe el archivo si existe. Crea el archivo si no existe	Al inicio del archivo
w+	Escritura y lectura. Sobreescibe el archivo si existe. Crea el archivo si no existe	Al inicio del archivo
wb+	Escritura y lectura en modo binario. Sobreescibe el archivo si existe. Crea el archivo si no existe	Al inicio del archivo
a	Añadido (agregar contenido). Crea el archivo si éste no existe	Si el archivo existe, al final de éste. Si el archivo no existe, al comienzo
ab	Añadido en modo binario (agregar contenido). Crea el archivo si éste no existe	Si el archivo existe, al final de éste. Si el archivo no existe, al comienzo
a+	Añadido (agregar contenido) y lectura. Crea el archivo si éste no existe.	Si el archivo existe, al final de éste. Si el archivo no existe, al comienzo
ab+	Añadido (agregar contenido) y lectura en modo binario. Crea el archivo si éste no existe	Si el archivo existe, al final de éste. Si el archivo no existe, al comienzo



Métodos del Objeto File

Método	Retorno	Ejemplo
read([bytes])	Lee todo el contenido de un archivo. Si se le pasa la longitud de bytes, leerá solo el contenido hasta la longitud indicada.	<pre>archivo = open("archivo.txt", "r") contenido = archivo.read() print contenido</pre>
readline([bytes])	Lee una línea del archivo.	<pre>archivo = open("archivo.txt", "r") linea1 = archivo.readline() print linea1</pre>
readlines()	Lee todas las líneas de un archivo	<pre>archivo = open("archivo.txt", "r") for linea in archivo.readlines(): print línea</pre>



Métodos del Objeto File

Método	Retorno	Ejemplo
seek(byte)	Mueve el puntero hacia el byte indicado	<pre>archivo = open("archivo.txt", "r") contenido = archivo.read() archivo.seek(0)</pre>
tell()	Retorna la posición actual del puntero	<pre>archivo = open("archivo.txt", "r") linea1 = archivo.readline() print archivo.tell()</pre>
close()	Cierra un archivo	<pre>archivo = open("remeras.txt", "r") contenido = archivo.read() archivo.close() print contenido</pre>



Métodos del Objeto File

Método	Retorno	Ejemplo
write(cadena)	Escribe cadena dentro del archivo	<pre>archivo = open("archivo.txt", "r+") contenido = archivo.read() final_de_archivo = archivo.tell() archivo.write('Nueva linea') archivo.seek(final_de_archivo) nuevo_contenido = archivo.read() print nuevo_contenido # Nueva linea</pre>
writelines(secuencia)	Secuencia será cualquier iterable cuyos elementos serán escritos uno por línea	<pre>archivo = open("archivo.txt", "r+") contenido = archivo.read() final_de_archivo = archivo.tell() archivo.seek(final_de_archivo) lista = ['Línea 1\n', 'Línea 2'] archivo.writelines(lista) archivo.seek(final_de_archivo) print archivo.readline() # Línea 1 print archivo.readline() # Línea 2</pre>



Propiedades del Objeto File

Se pueden acceder a las siguientes **propiedades** del objeto file:

- **closed**: retorna verdadero si el archivo se ha cerrado. De lo contrario, retornará falso.
- **mode**: retorna el modo de apertura.
- **name**: retorna el nombre del archivo
- **encoding**: retorna la codificación de caracteres de un archivo de texto



Propiedades del Objeto File

```
archivo = open("archivo.txt", "r+")
contenido = archivo.read()
nombre = archivo.name
modo = archivo.mode
archivo.close()
if archivo.closed:
    print "El archivo se ha cerrado correctamente"
else:
    print "El archivo permanece abierto"
```



Bibliografía y Webgrafía

<http://librosweb.es/libro/python/>