

# Ficheros de texto

Hay varias formas de abrir un fichero, la más común es utilizando la función *open* del módulo *io*.

## Creación y escritura

```
from io import open

texto = "Una línea con texto\nOtra línea con texto"

# Ruta donde crearemos el fichero, w indica escritura (puntero al principio)
fichero = open('fichero.txt', 'w')

# Escribimos el texto
fichero.write(texto)

# Cerramos el fichero
fichero.close()
```

## Lectura

```
from io import open

# Ruta donde leeremos el fichero, r indica lectura (por defecto ya es r)
fichero = open('fichero.txt', 'r')

# Lectura completa
texto = fichero.read()

# Cerramos el fichero
fichero.close()

print(texto)
```

```
Una línea con texto
Otra línea con texto
```

Podemos usar el método `readlines()` del fichero para generar una lista con las líneas:

```
from io import open
fichero = open('fichero.txt', 'r')

# Leemos creando una lista de líneas
texto = fichero.readlines()

fichero.close()
print(texto)
```

```
['Una línea con texto\n', 'Otra línea con texto\n']
```

También se puede leer un fichero utilizando la instrucción estándar *with* de la siguiente forma:

```
with open("fichero.txt", "r") as fichero:
    for linea in fichero:
        print(linea)
```

Una línea con texto

Otra línea con texto

## Extensión

Este modo nos permite añadir datos al final de un fichero:

```
from io import open

# Ruta donde leeremos el fichero, a indica extensión (puntero al final)
fichero = open('fichero.txt', 'a')

fichero.write('\nOtra línea más abajo del todo')

fichero.close()
```

La variante 'a+' permite crear el fichero si no existe:

```
fichero = open('fichero_inventado.txt', 'a+')
```

## Manejando el puntero

Es posible posicioar el puntero en el fichero manualmente usando el método *seek* e indicando un número de caracteres para luego leer una cantidad de caracteres con el método *read*:

```
fichero = open('fichero.txt', 'r')
fichero.seek(0)    # Puntero al principio
fichero.read(10)   # Leemos 10 caracteres
```

'Una línea '

Para posicionar el puntero justo al inicio de la segunda línea, podríamos ponerlo justo en la longitud de la primera:

```
fichero = open('fichero.txt', 'r')
fichero.seek(0)

# Leemos la primera línea y situamos el puntero al principio de la segunda
fichero.seek( len(fichero.readline()) )
```

```
# Leemos todo lo que queda del puntero hasta el final
fichero.read()
```

```
'\n0tra línea con texto\n0tra línea más abajo del todo'
```

## Lectura con escritura

Se puede abrir un fichero en modo lectura con escritura, pero éste debe existir previamente. Además por defecto el puntero estará al principio y si escribimos algo sobreescribiremos el contenido actual, así que prestad atención a los saltos de línea y caracteres especiales:

```
# Creamos un fichero de prueba con 4 líneas
fichero = open('fichero2.txt', 'w')
texto = "Línea 1\nLínea 2\nLínea 3\nLínea 4"
fichero.write(texto)
fichero.close()

# Lo abrimos en lectura con escritura y escribimos algo
fichero = open('fichero2.txt', 'r+')
fichero.write("0123456")

# Volvemos a poner el puntero al inicio y leemos hasta el final
fichero.seek(0)
fichero.read()
fichero.close()
```

```
'0123456\nLínea 2\nLínea 3\nLínea 4'
```

## Modificar una línea

Para lograr este fin lo mejor es leer todas las líneas en una lista, modificar la línea en la lista, posicionar el puntero al principio y reescribir de nuevo todas las líneas:

```
fichero = open('fichero2.txt', 'r+')
texto = fichero.readlines()

# Modificamos la línea que queramos a partir del índice
texto[2] = "Esta es la línea 3 modificada\n"

# Volvemos a poner el puntero al inicio y reescribimos
fichero.seek(0)
fichero.writelines(texto)
fichero.close()

# Leemos el fichero de nuevo
with open("fichero2.txt", "r") as fichero:
    print(fichero.read())
```

```
0123456
Línea 2
Esta es la línea 3 modificada
Línea 4
```

