

8. FICHEROS

Programación I
Grado en Inteligencia Artificial
Curso 2022/2023

Contenidos

- Protocolo de acceso a ficheros
- Ficheros de texto
 - Lectura
 - Escritura
 - Adición
- Paso de parámetros por línea de comandos
- Ficheros especiales
- Gestión de ficheros y directorios
- Serialización

Protocolo de acceso a ficheros

- **Apertura** indicando *ruta* y *modo* de acceso
 - **Lectura**: hace posible leer información pero no modificarla ni añadirla
 - **Escritura**: sólo permite escribir información, *truncando* el contenido del fichero
 - **Lectura/escritura**: hace posible leer y escribir
 - **Adición**: permite añadir información sin alterar la existente
- **Lectura o escritura** de información
- **Cierre** del fichero

Lectura de ficheros de texto

1. Apertura
del fichero

Paso 1: abrir el fichero
fichero = **open**('quevedo.txt', 'r')

2. Lectura
línea a línea

Paso 2: leer los datos
for linea **in** fichero:
 print(linea, end = " ")

3. Cierre del
fichero

Paso 3: cerrar el fichero
fichero.close()

Comprobación de apertura

```
from os import path
```

```
if path.exists('quevedo.txt'):
    fichero = open('quevedo.txt', 'r')
    for linea in fichero:
        print(linea, end = "")
    fichero.close()
else:
    print('El fichero no existe')
```

```
try:
    fichero = open('quevedo.txt', 'r')
    for linea in fichero:
        print(linea, end = "")
    fichero.close()
except IOError:
    print('El fichero no existe')
```

Se debe evitar intentar abrir un fichero
sin antes comprobar su existencia

Paso de parámetros desde el *exterior*

```
from sys import argv

if len(argv) != 3:
    print('Necesito dos parámetros')
else: #Tomo parámetros adicionales
    nombre = argv[1]
    numero = int(argv[2])
    fichero = open(nombre, 'r')
    contador = 0
    for linea in fichero:
        print(linea, end = '')
        contador += 1
        if contador == numero:
            break
    fichero.close()
```

Lista con los argumentos
suministrados al programa

El segundo elemento de la
lista contiene el nombre del
fichero a inspeccionar

El tercer elemento de la
lista indica el nº de líneas

Para ejecutar desde el
terminal:
**python programa.py
fichero.txt 10**

Lectura carácter a carácter

```
nombre = input('Nombre del fichero: ')
fichero = open(nombre, 'r')

contador = 0
caracter = fichero.read(1)
while caracter != "":
    contador += 1
    caracter = fichero.read(1)

fichero.close()
print(contador)
```

El parámetro limita el nº de caracteres a capturar

El método **read()** devuelve la cadena vacía cuando no queda texto que retirar

Lectura línea a línea

```
fichero = open('quevedo.txt', 'r')  
linea = fichero.readline()  
while linea != "":  
    print(linea, end = "  
    linea = fichero.readline()  
fichero.close()
```

El método **readline()** devuelve la cadena vacía cuando no queda texto que retirar

Escritura carácter a carácter

```
fichero_entrada = open('quevedo.txt', 'r')
fichero_salida = open('cifrado.txt', 'w')
caracter = fichero_entrada.read(1)
while caracter != "":
    if caracter >= 'a' and caracter <= 'y':
        codificado = chr(ord(caracter) + 1)
    elif caracter == 'z':
        codificado = 'a'
    else:
        codificado = caracter
    fichero_salida.write(codificado)
    caracter = fichero_entrada.read(1)
fichero_entrada.close()
fichero_salida.close()
```

El modo escritura crea el archivo en caso de que no exista

El método **write()** vuelca la cadena recibida como parámetro

Escritura línea a línea

```
fichero_entrada = open('quevedo.txt', 'r')
fichero_salida = open('cifrado.txt', 'w')
for linea in fichero_entrada:
    nueva_linea = ""
    for caracter in linea:
        if caracter >= 'a' and caracter <= 'y':
            codificado = chr(ord(caracter) + 1)
        elif caracter == 'z':
            codificado = 'a'
        else:
            codificado = caracter
    nueva_linea += codificado
    fichero_salida.write(nueva_linea)
fichero_entrada.close()
fichero_salida.close()
```

El modo escritura *trunca* el contenido del archivo en caso de que exista

En este caso, se vuelca una línea entera de cada vez

Escritura de números en ficheros

El método **write()** necesita cadenas de caracteres!!!

```
fichero = open('tabla.txt', 'w')  
for i in range(1, 5001):  
    fichero.write(str(i) + ' ' + str(i**2) + '\n')  
fichero.close()
```

```
fichero = open('tabla.txt', 'w')  
for i in range(1, 5001):  
    fichero.write('{0} {1}\n'.format(i, i**2))  
fichero.close()
```

Adición de texto a un fichero

```
nota = input('Nota a añadir: ')\n fichero = open('notas.txt', 'a')\n fichero.write(nota + '\n')\n fichero.close()
```

El modo adición abre el fichero y ubica el *cursor* al final para evitar sobrescribir su contenido

Limitaciones de los ficheros de texto:

1. No permiten el *acceso directo* para actuar en un punto específico del archivo (por ejemplo, para leer)
2. No se puede modificar el contenido
3. No se puede borrar selectivamente un fragmento

Ficheros especiales

- Entrada estándar (**stdin**): teclado
- Salida estándar (**stdout**): pantalla

```
from sys import stdin

print('Teclea un texto')
linea = stdin.readline()
print(linea)
```

```
from sys import stdin

for linea in stdin:
    print(linea)
```

Gestión de ficheros y directorios

- Funciones del módulo **os**:
 - Eliminación de archivos: **remove()**:
 - Renombrado de un archivo: **rename()**
 - Creación de directorios: **mkdir()**:
 - Eliminación de directorios: **rmdir()**
 - Cambio de directorio: **chdir()**
 - Listado de archivos en un directorio: **listdir()**
- Módulo **tempfile**: función **mktemp()**

Serialización: módulo pickle

```
from pickle import dump
```

```
#Creo una lista...
```

```
lista = [1, 2, 3, 4]
```

```
#...y la guardo en un fichero
```

```
fichero = open('datos', 'wb')
```

```
dump(lista, fichero)
```

```
fichero.close()
```

```
from pickle import load
```

```
#Cargo la lista del fichero...
```

```
fichero = open('datos', 'rb')
```

```
lista = load(fichero)
```

```
fichero.close()
```

```
#...y la muestro por pantalla
```

```
print(lista)
```