

# Programación I

## *Manejo avanzado de datos*

*Pablo Garaizar  
Borja Sanz Urquijo  
M.<sup>a</sup> Luz Guenaga  
Jenny Fajardo*

Facultad de Ingeniería

# Índice

- Listas multidimensionales.
- Diccionarios.
- Ficheros.



***Listas multidimensionales***

# Listas multidimensionales

- Python permite añadir **listas a listas** para crear listas multidimensionales.
- Ejemplo de lista **bidimensional**:

```
tabla = [ [1, 2, 3], [4, 5, 6], [7, 8, 9] ]
```

	tabla[0][0]	tabla[0][1]	tabla[0][2]
tabla[0][0]	1	2	3
tabla[1][0]	4	5	6
tabla[2][0]	7	8	9

# Listas multidimensionales

- Ejemplo de lista **tridimensional**:

```
cubo = [ [ [1, 2, 3], [4, 5, 6], [7, 8, 9] ],  
         [ [10, 11, 12], [13, 14, 15], [16, 17, 18] ],  
         [ [19, 20, 21], [22, 23, 24], [25, 26, 28] ] ]
```

	cubo[2][0][0]	19	20	21	
	cubo[1][0][0]	10	11	12	24
cubo[0][0][0]	1	2	3	15	27
cubo[0][1][0]	4	5	6	18	
cubo[0][2][0]	7	8	9		



# Listas multidimensionales

- Para ***recorrer*** las listas multidimensionales se suelen utilizar **bucles anidados**:

```
for fila in tabla:
```

```
    for elemento in fila:
```

```
        print(elemento)
```

```
for i in range(len(tabla)):
```

```
    for j in range(len(tabla[i])):
```

```
        print(tabla[i][j])
```

# Diccionarios

# Diccionarios

- Un **diccionario** es como una lista en la que los índices no son **numéricos** sino que pueden ser de **otros tipos**.
  - También se conocen como **mapas** (clave, valor).
  - **Importante:** las claves o índices son **únicos**.
- Ejemplo:

```
poblaciones = { 'Francia': 67407241, 'Alemania': 83149300,  
                'España': 47450795 }
```

```
> print(poblaciones['Francia'])
```

```
> 67407241
```



# Diccionarios

- Para **asignar** un valor a una clave o índice, se hace de manera similar a las listas:  

```
poblaciones['Francia'] = 67407241
```
- Si queremos saber si una clave está en un diccionario, podemos usar el operador **in**:  

```
> 'Francia' in poblaciones  
> True
```
- Para eliminar una clave del diccionario, podemos usar **del**:  

```
if 'Francia' in poblaciones:  
    del poblaciones['Francia']
```

# Diccionarios

- Para **recorrer** un diccionario, lo más habitual es recorrer sus **claves**:

```
for pais in poblaciones:  
    print(poblaciones[pais])
```

- También podemos **recorrer** solo sus **valores**:

```
for poblacion in poblaciones.values():  
    print(poblacion)
```

- O utilizar **iteritems** para tener las dos cosas:

```
for clave, valor in poblaciones.items():  
    print(clave, valor)
```



***Ficheros***

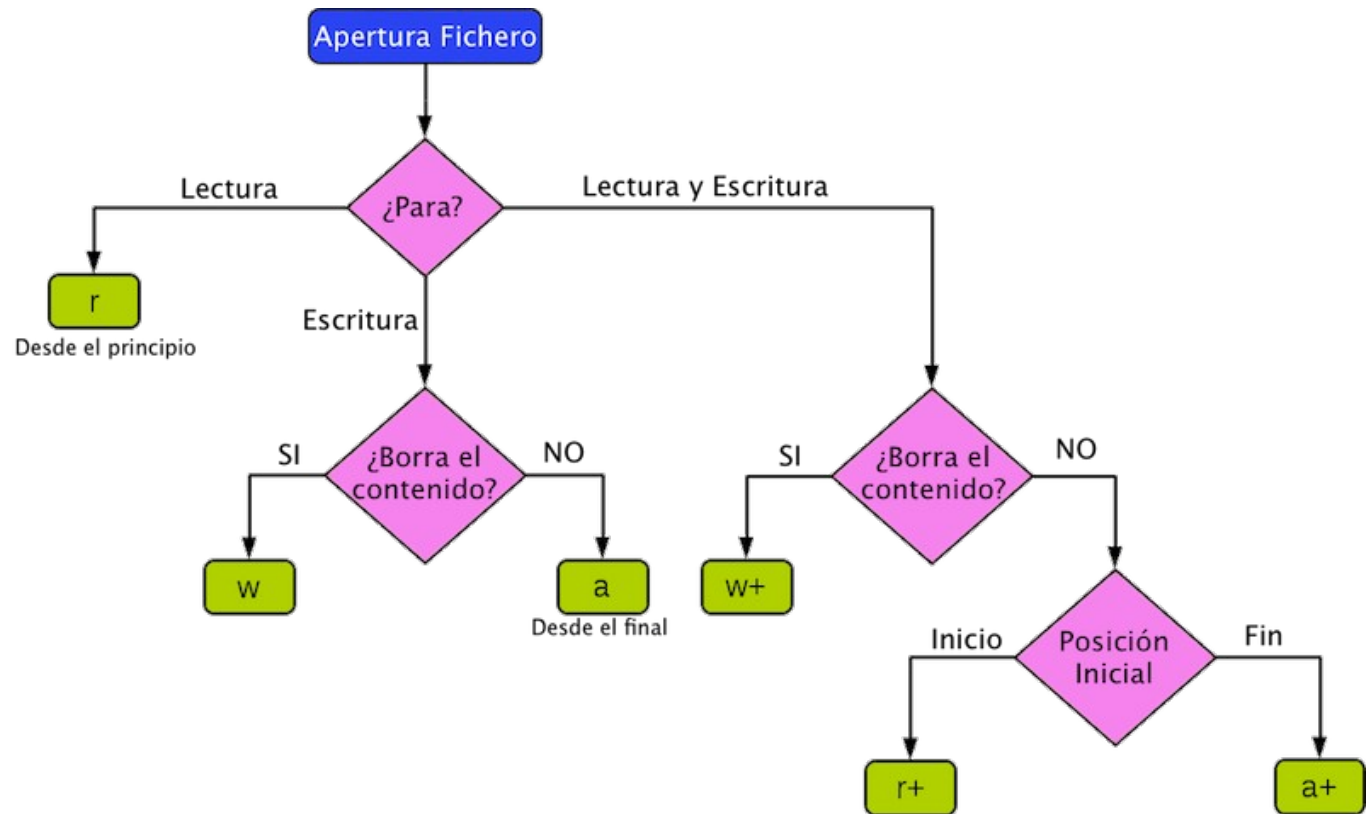
# Ficheros

- Los ficheros sirven para dotar de **persistencia** a nuestros programas (**guardar** datos para **leerlos** más adelante).
- Operaciones habituales:
  - **Crear / abrir** fichero.
  - **Leer** datos desde fichero.
  - **Escribir** datos en fichero.
  - **Cerrar** fichero.



# Abrir ficheros

- Usamos **open** para abrir un fichero, indicando:
  - La **ruta** al fichero: “fichero.txt”.
  - El **modo** de acceso:



# Abrir ficheros

- **open()** devuelve una referencia al fichero, que usaremos después para leer y escribir en él:

```
fichero = open('fichero.txt', 'r')  
for linea in fichero:  
    print(linea)
```

- Habitualmente, se usa **with** para englobar todas las sentencias que usan el fichero:

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

# Leer ficheros

- **read** lee el fichero entero como una cadena de texto:

```
fichero = open('fichero.txt', 'r')  
datos = fichero.read()  
print(datos)
```

- **readlines** devuelve las líneas del fichero en una lista:

```
fichero = open('fichero.txt', 'r')  
datos = fichero.readlines()  
print(datos)
```

# Leer ficheros

- Es muy habitual utilizar un **for** para procesar cada línea del fichero:

```
fichero = open('fichero.txt', 'r')  
for linea in fichero:  
    print(linea)
```



# Escribir ficheros

- **write** escribe datos en un fichero:  
`fichero = open('fichero.txt', 'w')`  
`datos = 'Quiero guardar estos datos'`  
`fichero.write(datos)`  
`fichero.write('Otros datos')`

# Escribir ficheros

- ¿**Dónde** se escribe? Donde esté el puntero de lectura/escritura.
  - Se define al **abrir** el fichero:
    - r/w: al principio del fichero.
    - a: append, al final del fichero.
  - Se puede modificar con **seek(salto, comienzo)**, donde 0 es al principio y 2 es al final:

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

```
datos = 'Guardo estos datos al final'
```

```
fichero.write(datos)
```

```
fichero.seek(0, 0)
```

```
fichero.write('Y esto otro al principio, sobrescribiendo')
```

```
fichero.seek(0, 2)
```

```
fichero.write('Volvemos al final')
```

# Cerrar ficheros

- **close** cierra el fichero (¡es importante hacerlo!):  
`fichero = open('fichero.txt', 'w')`  
`datos = 'Quiero guardar estos datos'`  
`fichero.write(datos)`  
`fichero.close()`

# Otras operaciones

- El sistema operativo es quien se encarga de otras operaciones como borrar o renombrar ficheros.
- Hay que importar el **módulo os**:

```
import os
```

```
if os.path.isfile('fichero.txt'):
    os.rename('fichero.txt', 'saludo.txt')
else:
    print('fichero.txt no existe')
```

```
if os.path.isfile('saludo.txt'):
    os.remove('saludo.txt')
else:
    print('saludo.txt no existe')
```



# Internet

- Para leer ficheros desde Internet hay que importar el **módulo urllib**:

```
import urllib
```

```
fichero = urllib.request.urlopen('https://...')
```

```
datos = fichero.read()
```

```
print(datos.decode('utf-8'))
```

```
# la mayoría de páginas web están codificadas en utf-8
```

```
# por eso el decode('utf-8')
```

# Referencias

- [https://www.snakify.org/es/lessons/dictionaries\\_dicts/](https://www.snakify.org/es/lessons/dictionaries_dicts/)
- <https://jarroba.com/lectura-y-escritura-de-ficheros-en-python-con-ejemplos/>
- <https://aprendeconalf.es/docencia/python/manual/ficheros/>
- John Sturtz, Real Python.
- John M. Zelle. Python Programming: An Introduction to Computer Science.
- Learnbyexample.org.
- Wikipedia.

# Referencias

- Imágenes:
  - <https://www.flickr.com/photos/stockcatalog/43194359892/>
  - <https://www.flickr.com/photos/takashi/18862634/>
  - <https://www.flickr.com/photos/takashi/18862634/>