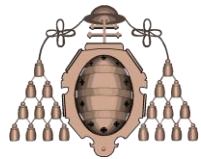


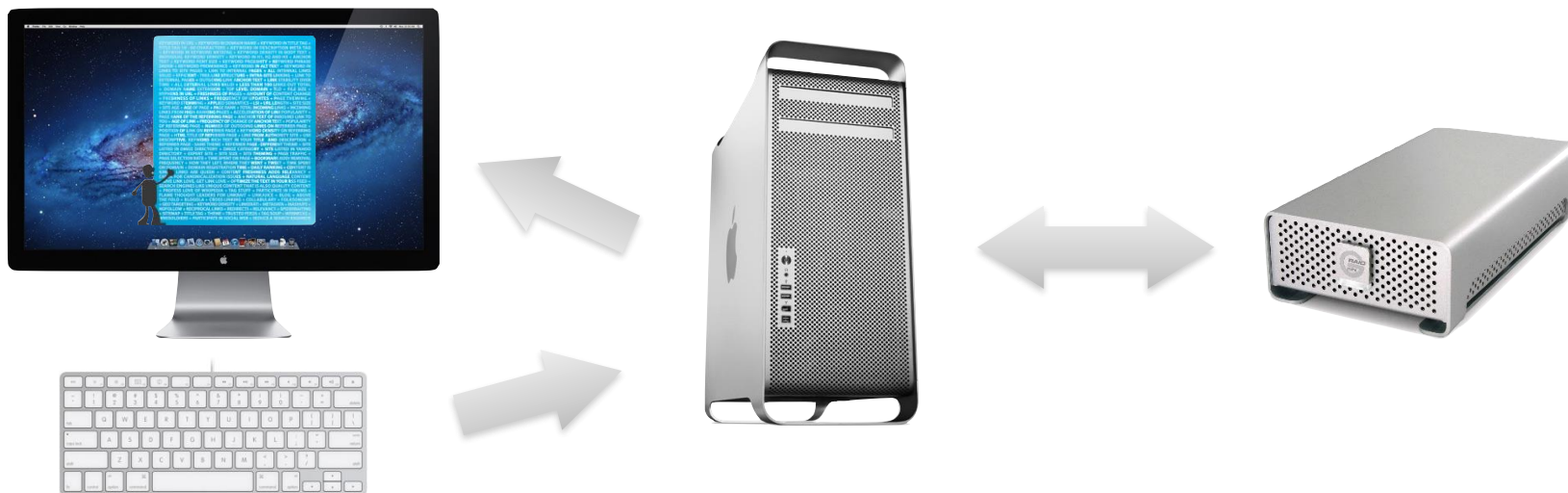
2.6 Entrada/salida a ficheros

- 2.1 Abstracción de problemas para su programación. Conceptos fundamentales
- 2.2 Variables, expresiones, asignación
- 2.3 Uso de entrada/salida por consola
- 2.4 Manejo de estructuras básicas de control de flujo: secuencial, alternativa y repetitiva
- 2.5 Definición y uso de subprogramas y funciones. Ámbito de variables
- **2.6 Entrada/salida a ficheros**
- 2.7 Tipos y estructuras de datos básicas: arrays

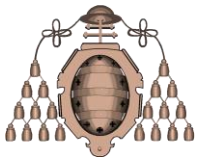


Introducción

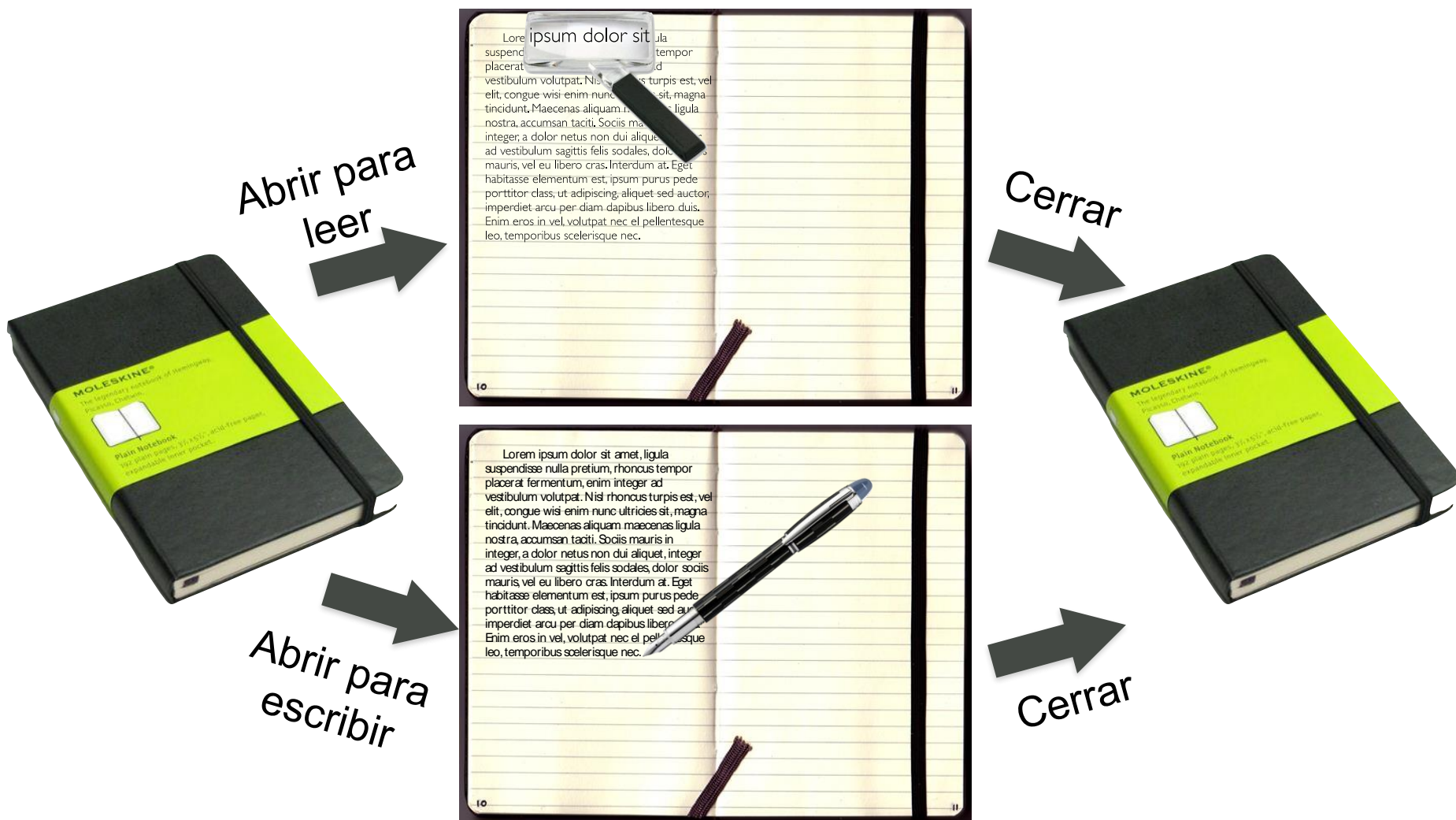
- Hasta ahora nuestros programas almacenan sus datos en memoria
 - Necesitamos un mecanismo de almacenamiento para guardar permanentemente la información y poder usarla posteriormente



- **Un fichero es una secuencia de bytes que se guarda en almacenamiento permanente**



¿Cómo se utiliza un fichero?





Abrir un fichero

```
f = open('mifichero.txt', 'r')
```



Variable a la
que se asigna
el fichero

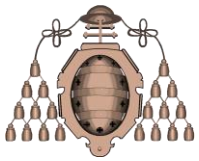


Nombre del fichero



Modo de apertura

- Modos de apertura:
 - **'r'** : abrir para lectura; si el fichero no existe se produce un error
 - **'w'** : abrir para escritura; si el fichero no existe, se crea, y si existe se sobrescribe su contenido
 - **'a'** : abrir para añadir; si el fichero no existe se crea, y si existe, se crea con permiso de escritura, y se añaden los datos al final

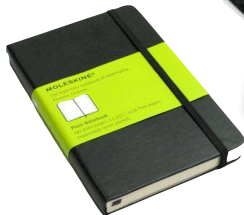


Un ejemplo de lectura



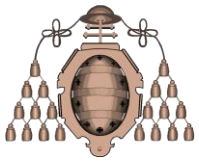
Primera línea
Segunda línea

Primera línea

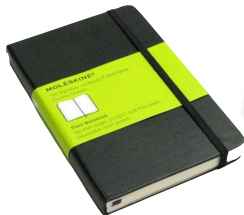


```
>>> f = open('mifichero.txt', 'r')
>>> lin = f.readlines()
>>> f.close()
>>> print(lin)
['Primera línea\n', 'Segunda línea\n']
```

OJO: El salto de línea también se lee y se almacena en la variable `lin`, que luego es impresa por `print`

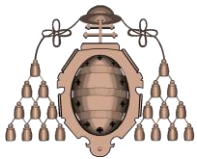


Un ejemplo de escritura (añadiendo)



```
>>> f = open('mifichero.txt', 'a')  
>>> f.write('Tercera linea\n')  
>>> f.close()
```

Debemos indicar explícitamente el separador de líneas para que la próxima escritura sea en la línea siguiente



Lectura completa sobre variable o lista

- Python puede leer con una sola instrucción todo un fichero

```
>>> f = open('mifichero.txt', 'r')
>>> todo = f.read()
>>> f.close()
>>> print(todo)
```

Primera línea
Segunda línea

→

P	r	i	m	e	r	a		l	i	n	e	a	\n	S	e	g	u	n	d	a		l	i	n	e	a	\n
---	---	---	---	---	---	---	--	---	---	---	---	---	----	---	---	---	---	---	---	---	--	---	---	---	---	---	----

- En vez de leer una cadena de texto con el contenido completo, podemos cargar una lista con las líneas:

```
>>> f = open('mifichero.txt', 'r')
>>> lista = f.readlines()
>>> f.close()
>>> print(lista[0])
```

Primera línea

→

P	r	i	m	e	r	a		l	i	n	e	a	\n
---	---	---	---	---	---	---	--	---	---	---	---	---	----

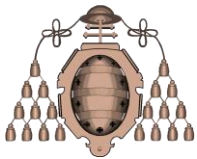
```
>>> print(lista[1])
```

Segunda línea

→

S	e	g	u	n	d	a		l	i	n	e	a	\n
---	---	---	---	---	---	---	--	---	---	---	---	---	----

```
>>>
```



Lectura por líneas

- Un bucle `for` puede servir para leer línea a línea un fichero

```
f = open('mifichero.txt')
contador = 0
for l in f:
    contador = contador + 1
    print('Línea:', contador, 'Texto:', l.strip())
f.close()
```

Usamos `l.strip()` para que no salga un salto de línea adicional en la consola