

## Sesión 5. Más repeticiones y archivos

### Ejemplos

1. Para contar el número de veces que se cumple alguna condición en un ciclo for se usa un entero que va incrementando su valor cada vez que dicha condición se satisface. Por ejemplo, el siguiente programa guarda en la variable `num_divisores` el número de divisores que tiene el entero `n`.

```
1  num_divisores = 0
2  # Se revisan los enteros 1, ..., n
3  for i in range(1,n+1):
4      if n % i == 0:
5          num_divisores = num_divisores + 1
6  print(f'{n} tiene {num_divisores} divisores')
```

En la línea 1 se inicia la variable `num_divisores` a cero; esta variable funge como *contador*, es decir, en esta variable se van contando los divisores de `n`. Para conseguir esto, en la línea 5 se incrementa en uno el valor de `num_divisores`, pero únicamente cuando la variable `i` divide a `n` (comparación en línea 4). Con mayor detalle, lo que ocurre en la línea 5 es que el nuevo valor de `num_divisores` es igual a su valor anterior más uno.<sup>1</sup>

### Ejercicios

1. Crea un programa que reciba el nombre de un archivo de texto y que imprima en pantalla el número de líneas que tiene el texto. Por ejemplo, si el archivo `quijote.txt` contiene las líneas

Nunca fuera caballero  
de damas tan bien servido  
como fuera don Quijote  
cuando de su aldea vino:  
doncellas curaban dél;  
princesas, del su rocino,

o Rocinante, que éste es el nombre, señoras mías,

el programa debe ejecutarse de forma similar a lo siguiente:

Introduce el nombre del archivo: `quijote.txt`  
El archivo `quijote.txt` tiene 8 líneas.

---

<sup>1</sup>Para matemáticos: esto no es una autoreferencia ni una igualdad sin sentido, simplemente primero se lee el valor de `num_divisores`, se le suma 1 y este nuevo valor se guarda en `num_divisores`.

Recuerda que se puede usar «`for` línea in archivo:» para recorrer el contenido de un archivo línea por línea.

2. Las calificaciones de un grupo de estudiantes se encuentran en un archivo de texto cuyo contenido tiene la siguiente forma:

```
-Juan Pérez
7
8
10
5
6
5
*****
-Alicia Gutiérrez López
5
8
10
9
*****
-Alberto Rosas Camacho
10
7
7
*****
```

Haz un programa que lea el archivo de calificaciones y que imprima en pantalla una lista de los nombres de los alumnos junto con su calificación promedio. Para el ejemplo anterior se debería imprimir algo como

```
Juan Pérez: 6.8333
Alicia Gutiérrez López: 8.0
Alberto Rosas Camacho: 8.0
```

*Pista: Los guiones y asteriscos sirven como guía. Puedes remover el guión inicial de una cadena usando `cadena.strip('-')`.*

3. El método `split()` de las variables de tipo cadena sirve para partir una cadena en palabras, usando cualquier espacio en blanco (incluyendo saltos de línea y tabuladores) como separador de palabras.

Usando el método anterior haz un programa que cuente las palabras de un archivo de texto. La salida debe ser similar a la siguiente:

Nombre del archivo: ejemplo.txt

El archivo ejemplo.txt tiene 2344 palabras.

4. Muchos programas que se ejecutan en una terminal no piden los datos al usuario como hemos hecho hasta ahora (usando la función `input()`), en vez de eso se introducen los datos al mismo tiempo que se ejecuta el programa. Usando el Ejercicio 1 como ejemplo, podría ejecutarse así

```
$ python num_lineas.py quijote.txt
```

El archivo quijote.txt tiene 8 líneas.

Nota que el nombre quijote.txt se introduce en la misma línea que el nombre del programa, y cuando se ejecuta el programa simplemente imprime el resultado. Para tener esta funcionalidad, en Python se usa la lista `sys.argv` del módulo `sys`. Esta lista contiene las palabras que se escriben en la terminal cuando se ejecuta el programa de Python (excepto el comando python). En el ejemplo anterior, al ejecutar

```
$ python num_lineas.py quijote.txt
```

la lista `sys.argv` sería

```
['num_lineas.py', 'quijote.txt']
```

Haz los cambios necesarios en los ejercicios 1, 2 y 3 para que el nombre del archivo se introduzca como parámetro del programa, así ya no debe usarse `input()` para leer el dato de entrada. Recuerda agregar «`import sys`» para poder usar `sys.argv`.