4.2.6. for-in como forma compacta de ciertos while

Ciertos bucles se ejecutan un número de veces fijo y conocido *a priori*. Por ejemplo, al desarrollar el programa que calcula el sumatorio de los 1000 primeros números utilizamos un bucle que iteraba exactamente 1000 veces:

sumatorio_6.py — sumatorio.py

```
1 sumatorio = 0
2 i = 1
3 while i <= 1000:
4     sumatorio += i
5     i += 1
6 print sumatorio
```

El bucle se ha construido de acuerdo con un patrón, una especie de «frase hecha» del lenguaje de programación:

```
i = valor inicial
while i <= valor final:
    acciones
    i += 1
```

En este patrón la variable *i* suele denominarse *índice* del bucle.

Podemos expresar de forma compacta este tipo de bucles con un **for–in** siguiendo este otro patrón:

```
for i in range(valor inicial, valor final + 1):
    acciones
```

Fíjate en que las cuatro líneas del fragmento con **while** pasan a expresarse con sólo dos gracias al **for–in** con *range*.

El programa de cálculo del sumatorio de los 1000 primeros números se puede expresar ahora de este modo:

sumatorio.py — sumatorio.py

```
1 sumatorio = 0
2 for i in range(1, 1001):
3     sumatorio += i
4
5 print sumatorio
```

¡Bastante más fácil de leer que usando un **while**!

EJERCICIOS

▶ 123 Haz un programa que pida el valor de dos enteros n y m y que muestre por pantalla el valor de

$$\sum_{i=n}^{m} i.$$

Debes usar un bucle **for–in** para el cálculo del sumatorio.

▶ 124 Haz un programa que pida el valor de dos enteros n y m y que muestre por pantalla el valor de

$$\sum_{i=n}^{m} i^2.$$

▶ 125 Haz un programa que pida el valor de dos enteros n y m y calcule el sumatorio de todos los números pares comprendidos entre ellos (incluyéndolos en el caso de que sean pares).