

5.2.9. Pertenencia de un elemento a una lista

Diseñemos un programa que, dados un elemento y una lista, nos diga si el elemento pertenece o no a la lista mostrando en pantalla el mensaje «Pertenece» o «No pertenece» en función del resultado.

```
pertenencia.5.py      pertenencia.py
1 elemento = 5
2 lista = [1, 4, 5, 1, 3, 8]
3
4 pertenece = False
5 for i in lista:
6     if elemento == i:
7         pertenece = True
8         break
9
10 if pertenece:
11     print 'Pertenece'
12 else:
13     print 'No pertenece'
```

EJERCICIOS

► 236 ¿Por qué este otro programa es erróneo?

```
pertenencia.6.py      ⚡ pertenencia.py ⚡
1 elemento = 5
2 lista = [1, 4, 5, 1, 3, 8]
3
4 for i in lista:
5     if elemento == i:
6         pertenece = True
7     else:
8         pertenece = False
9     break
10
11 if pertenece:
12     print 'Pertenece'
13 else:
14     print 'No pertenece'
```

La pregunta de si un elemento pertenece o no a una lista es tan frecuente que Python nos proporciona un operador predefinido que hace eso mismo. El operador es binario y se denota con la palabra **in** (que en inglés significa «en» o «pertenece a»). El operador **in** recibe un elemento por su parte izquierda y una lista por su parte derecha y devuelve cierto o falso. No necesitamos, pues, definir la función *pertenece*. Un programa que necesita determinar si un elemento pertenece o no a una lista y actuar en consecuencia puede hacerlo así:

```
pertenencia.7.py      pertenencia.py
1 conjunto = [1, 2, 3]
2 elemento = int(raw_input('Dame un número:'))
3 if not elemento in conjunto:
4     conjunto.append(elemento)
```

O, equivalentemente:

```
pertenencia.8.py      pertenencia.py
1 conjunto = [1, 2, 3]
2 elemento = int(raw_input('Dame un número:'))
3 if elemento not in conjunto:
4     conjunto.append(elemento)
```

El operador «**not in**» es el operador **in** negado.

EJERCICIOS

► 237 ¿Qué hace este programa?

```
1 letra = raw_input('Dame una letra:')
2 if (len(letra) == 1 and 'a' <= letra <= 'z') or letra in ['á', 'é', 'í', 'ó', 'ú', 'ü', 'ñ']:
3     print letra, 'es una letra minúscula'
```

► 238 ¿Qué hace este programa?

```
1 letra = raw_input('Dame una letra:')
2 if len(letra) == 1 and ('a' <= letra <= 'z' or letra in 'áéíóúüñ'):
3     print letra, 'es una letra minúscula'
```

Ya te hemos dicho que Python ofrece funcionalidades similares entre tipos de datos similares. Si el operador **in** funciona con listas, ¿funcionará con cadenas, que también son secuencias? Sí. El operador **in** comprueba si una cadena forma parte o no de otra⁴:

```
>>> 'a' in 'cadena'
True
>>> 'ade' in 'cadena'
True
>>> 'ada' in 'cadena'
False
```

⁴Este comportamiento sólo se da desde la versión 2.3 de Python. Versiones anteriores sólo aceptaban que, si ambos operandos eran cadenas, el operador izquierdo fuera de longitud 1.