

4.2.3. Otro programa de ejemplo: requisitos en la entrada

Vamos con otro programa sencillo pero ilustrativo. Estudia este programa:

```

raiz.4.py
1 from math import sqrt
2
3 x = float(raw_input('Introduce un número positivo: '))
4
5 print 'La raíz cuadrada de %f es %f' % (x, sqrt(x))

```

Como puedes ver, es muy sencillo: pide un número (flotante) y muestra por pantalla su raíz cuadrada. Como *sqrt* no puede trabajar con números negativos, *pedimos* al usuario que introduzca un número positivo. Pero nada *obliga* al usuario a introducir un número positivo.

En lugar de adoptar una solución como las estudiadas anteriormente, esto es, evitando ejecutar el cálculo de la raíz cuadrada cuando el número es negativo con la ayuda de una sentencia condicional, vamos a obligar a que el usuario introduzca un número positivo *repetiendo* la sentencia de la línea 3 cuantas veces sea preciso. Dado que vamos a repetir un fragmento de programa, utilizaremos una sentencia **while**. En principio, nuestro programa presentará este aspecto:

```

raiz.py
1 from math import sqrt
2
3 while condición:
4     x = float(raw_input('Introduce un número positivo: '))
5
6 print 'La raíz cuadrada de %f es %f' % (x, sqrt(x))

```

¿Qué condición poner? Está claro: el bucle debería leerse así «mientras *x* sea un valor inválido, hacer...», es decir, «mientras *x* sea menor que cero, hacer...»; y esa última frase se traduce a Python así:

```

raiz.5.py
1 from math import sqrt
2
3 while x < 0:
4     x = float(raw_input('Introduce un número positivo: '))
5
6 print 'La raíz cuadrada de %f es %f' % (x, sqrt(x))

```

Pero el programa no funciona correctamente. Mira qué obtenemos al ejecutarlo:

```

Traceback (innermost last):
  File 'raiz.py', line 3, in ?
    while x < 0:
NameError: x

```

Python nos indica que la variable *x* no está definida (no existe) en la línea 3. ¿Qué ocurre? Vayamos paso a paso: Python empieza ejecutando la línea 1, con lo que importa la función *sqrt* del módulo *math*; la línea 2 está en blanco, así que, a continuación, Python ejecuta la línea 3, lo cual pasa por saber si la condición del **while** es cierta o falsa. Y ahí se produce el error, pues se intenta conocer el valor de *x* cuando *x* no está inicializada. Es necesario, pues, inicializar antes la variable; pero, ¿con qué valor? Desde luego, no con un valor positivo. Si *x* empieza tomando un valor positivo, la línea 4 no se ejecutará. Probemos, por ejemplo, con el valor -1 .

```

raiz.py
1 from math import sqrt
2
3 x = -1
4 while x < 0:

```

```

5  x = float(raw_input('Introduce un número positivo:'))
6
7  print 'La raíz cuadrada de %f es %f' % (x, sqrt(x))

```

Ahora sí. Hagamos una traza.

1. Empezamos ejecutando la línea 1, con lo que importa la función *sqrt*.
2. La línea 2 se ignora.
3. Ahora ejecutamos la línea 3, con lo que *x* vale -1 .
4. En la línea 4 nos preguntamos: ¿es *x* menor que cero? La respuesta es sí, de modo que debemos ejecutar la línea 5.
5. La línea 5 hace que se solicite al usuario un valor para *x*. Supongamos que el usuario introduce un número negativo, por ejemplo, -3 .
6. Como hemos llegado al final de un bucle **while**, volvemos a la línea 4 y nos volvemos a preguntar ¿es *x* menor que cero? De nuevo, la respuesta es sí, así que pasamos a la línea 4.
7. Supongamos que ahora el usuario introduce un número positivo, pongamos que el 16.
8. Por llegar al final de un bucle, toca volver a la línea 4 y plantearse la condición: ¿es *x* menor que cero? En este caso la respuesta es no, así que salimos del bucle y pasamos a ejecutar la línea 7, pues la línea 6 está vacía.
9. La línea 7 muestra por pantalla «La raíz cuadrada de 16.000000 es 4.000000». Y ya hemos acabado.

Fíjate en que las líneas 4–5 se pueden repetir cuantas veces haga falta: sólo es posible salir del bucle introduciendo un valor positivo en *x*. Ciertamente hemos conseguido obligar al usuario a que los datos que introduce satisfagan una cierta restricción.

..... EJERCICIOS

- **111** ¿Qué te parece esta otra versión del mismo programa?

```

raiz.6.py
raiz.py
1  from math import sqrt
2
3  x = float(raw_input('Introduce un número positivo:'))
4  while x < 0:
5      x = float(raw_input('Introduce un número positivo:'))
6
7  print 'La raíz cuadrada de %f es %f' % (x, sqrt(x))

```

- **112** Diseña un programa que solicite la lectura de un número entre 0 y 10 (ambos inclusive). Si el usuario teclea un número fuera del rango válido, el programa solicitará nuevamente la introducción del valor cuantas veces sea menester.
- **113** Diseña un programa que solicite la lectura de un texto que no contenga letras mayúsculas. Si el usuario teclea una letra mayúscula, el programa solicitará nuevamente la introducción del texto cuantas veces sea preciso.
-