### El bloque else

Podemos hacer que este programa sea más resistente a los errores envolviendo la línea que podría producir errores en un bloque try-except. El error se produce en la línea que realiza la división, así que ahí es donde pondremos el bloque try-except. Este ejemplo también incluye un bloque else. Cualquier código que dependa de que el bloque try se ejecute correctamente va en el bloque else:

--snip--  
while True:  
 --snip--  
 if second\_number == 'q':  
 break  
❶ try:  
 answer = int(first\_number) / int(second\_number)  
❷ except ZeroDivisionError:  
 print("You can't divide by 0!")  
❸ else:  
 print(answer)

Pedimos a Python que intente completar la operación de división en un bloque try ❶, que incluye sólo el código que podría causar un error. Cualquier código que dependa del éxito del bloque try se añade al bloque else. En este caso, si la operación de división tiene éxito, utilizamos el bloque else para imprimir el resultado ❸.

El bloque except indica a Python cómo responder cuando surge un ❷ ZeroDivisionError . Si el bloque try no tiene éxito debido a un error de división por cero, imprimimos un mensaje amigable indicando al usuario cómo evitar este tipo de error. El programa sigue ejecutándose y el usuario nunca ve un rastreo:

Give me two numbers, and I'll divide them.  
Enter 'q' to quit.  
  
First number: 5  
Second number: 0  
You can't divide by 0!  
  
First number: 5  
Second number: 2  
2.5  
  
First number: q

El único código que debe ir en un bloque try es el código que pueda provocar el lanzamiento de una excepción. A veces tendrás código adicional que sólo debe ejecutarse si el bloque try ha tenido éxito; este código va en el bloque else. El bloque except indica a Python qué hacer en caso de que se produzca una determinada excepción al intentar ejecutar el código del bloque try.

Al anticiparte a las posibles fuentes de error, puedes escribir programas robustos que sigan ejecutándose aunque se encuentren con datos no válidos y recursos ausentes. Tu código será resistente a los errores inocentes del usuario y a los ataques maliciosos.

[anterior](c10_23.html)[Subtema 24 de 37: (Ver todo)](c10.html)[siguiente](c10_25.html)