### Utilizar bloques try-except

Cuando creas que se puede producir un error, puedes escribir un bloque try-except para manejar la excepción que se pueda producir. Le dices a Python que intente ejecutar algún código, y le dices qué hacer si el código produce un tipo concreto de excepción.

Este es el aspecto de un bloque try-except para manejar la excepción ZeroDivisionError:

try:  
 print(5/0)  
except ZeroDivisionError:  
 print("You can't divide by zero!")

Colocamos print(5/0), la línea que causó el error, dentro de un bloque try. Si el código del bloque try funciona, Python se salta el bloque except. Si el código del bloque try provoca un error, Python busca un bloque except cuyo error coincida con el que se ha producido, y ejecuta el código de ese bloque.

En este ejemplo, el código del bloque try produce un ZeroDivisionError, por lo que Python busca un bloque except que le indique cómo responder. Python ejecuta entonces el código de ese bloque, y el usuario ve un mensaje de error amigable en lugar de un rastreo:

You can't divide by zero!

Si hubiera más código después del bloque try-except, el programa seguiría ejecutándose porque le hemos dicho a Python cómo manejar el error. Veamos un ejemplo en el que la captura de un error puede permitir que un programa siga ejecutándose.

[anterior](c10_21.html)[Subtema 22 de 37: (Ver todo)](c10.html)[siguiente](c10_23.html)