Manejo de Excepciones en Python

Tiempo estimado necesario: 10 minutos

Objetivos

1. Comprender las Excepciones
2. Distinguir Errores de Excepciones
3. Familiaridad con Excepciones Comunes de Python
4. Manejar Excepciones de Manera Efectiva

¿Qué son las excepciones?

Las excepciones son alertas cuando algo inesperado sucede al ejecutar un programa. Podría ser un error en el código o una situación que no se había previsto. Python puede generar estas alertas automáticamente, pero también podemos activarlas intencionadamente utilizando el comando raise. La parte interesante es que podemos evitar que nuestro programa se bloquee al manejar excepciones.

Errores vs. Excepciones

Espera, ¿cuál es la diferencia entre errores y excepciones? Bueno, `errors` son generalmente problemas grandes que provienen de la computadora o el sistema. A menudo hacen que el programa deje de funcionar por completo. Por otro lado, `exceptions` son más como problemas que podemos controlar. Ocurren debido a algo que hicimos en nuestro código y generalmente se pueden solucionar, por lo que el programa sigue funcionando.

Aquí está la diferencia entre `Errors and exceptions`:-

Aspecto	Errores	Excepciones
Origen	Los errores son típicamente causados por el entorno, hardware o sistema operativo.	Las excepciones son generalmente el resultado de la ejecución problemática del código dentro del programa.
Naturaleza	Los errores son a menudo graves y pueden llevar a bloqueos del programa o a una terminación anormal.	Las excepciones son generalmente menos graves y pueden ser capturadas y manejadas para prevenir la terminación del programa.
Manejo	Los errores generalmente no son capturados ni manejados por el programa mismo.	Las excepciones pueden ser capturadas utilizando bloques try-except y manejadas de manera elegante, permitiendo que el programa continúe su ejecución.
Ejemplos	Ejemplos incluyen “SyntaxError” debido a una sintaxis incorrecta o “NameError” cuando una variable no está definida.	Ejemplos incluyen “ZeroDivisionError” al dividir por cero, o “FileNotFoundError” al intentar abrir un archivo inexistente.
Clasificación	Los errores no se clasifican en categorías.	Las excepciones se clasifican en varias clases, como “ArithmeticError,” “IOError,” “ValueError,” etc., según su naturaleza.

Excepciones Comunes en Python

Aquí hay algunos ejemplos de excepciones con las que a menudo nos encontramos y que podemos manejar usando esta herramienta:

- **ZeroDivisionError:** Este error surge cuando se intenta dividir un número entre cero. La división por cero no está definida en matemáticas, causando un error aritmético. Por ejemplo:

```
result = 10 / 0
print(result)
# Raises ZeroDivisionError
```

- **ValueError:** Este error ocurre cuando se utiliza un valor inapropiado dentro del código. Un ejemplo de esto es cuando se intenta convertir una cadena no numérica a un entero:
  Por ejemplo:

```
num = int("abc")
# Raises ValueError
```

- **FileNotFoundError:** Esta excepción se encuentra cuando se intenta acceder a un archivo que no existe.
  Por ejemplo:

```
with open("nonexistent_file.txt", "r") as file:
        content = file.read()   # Raises FileNotFoundError
```

- **IndexError:** Un IndexError ocurre cuando se utiliza un índice para acceder a un elemento en una lista que está fuera del rango de índice válido. Por ejemplo:

```
my_list = [1, 2, 3]
```

```
value = my_list[1]  # No IndexError, within range
missing = my_list[5]  # Raises IndexError
```

- **KeyError:** El KeyError surge cuando se intenta acceder a una clave que no existe en un diccionario.
  Por ejemplo:

```
my_dict = {"name": "Alice", "age": 30}
value = my_dict.get("city")  # No KeyError, usando el método .get()
missing = my_dict["city"]  # Lanza KeyError
```

- **TypeError:** El TypeError ocurre cuando un objeto se utiliza de manera incompatible. Un ejemplo incluye intentar concatenar una cadena y un entero:
  Por ejemplo:

```
result = "hello" + 5
# Raises TypeError
```

- **AttributeError:** Un AttributeError ocurre cuando se accede a un atributo o método en un objeto que no posee ese atributo o método específico. Por ejemplo:

```
text = "example"
length = len(text)  # No AttributeError, correct method usage
missing = text.some_method()  # Raises AttributeError
```

- **ImportError:** Este error se encuentra cuando se intenta importar un módulo que no está disponible. Por ejemplo: `import non_existent_module`

Nota: **Recuerda, las excepciones que encontrarás no se limitan solo a estas. Hay muchas más en Python. Sin embargo, no hay necesidad de preocuparse. Al utilizar la técnica proporcionada a continuación y siguiendo la sintaxis correcta, podrás manejar cualquier excepción que se presente.**

Manejo de Excepciones:

Python tiene una herramienta útil llamada `try and except` que nos ayuda a gestionar excepciones.

Try and Except : Puedes usar los bloques try y except para evitar que tu programa se bloquee debido a excepciones.

Así es como funcionan:

1. El código que puede resultar en una excepción se encuentra en el bloque try.
2. Si ocurre una excepción, el código salta directamente al bloque except.
3. En el bloque except, puedes definir cómo manejar la excepción de manera elegante, como mostrar un mensaje de error o tomar acciones alternativas.
4. Después del bloque except, el programa continúa ejecutando el código restante.

Ejemplo: Intentando dividir por cero

```
# usando Try- except
try:
    # Intentando dividir 10 entre 0
    result = 10 / 0
except ZeroDivisionError:
    # Manejo del ZeroDivisionError e impresión de un mensaje de error
    print("Error: No se puede dividir entre cero")
# Esta línea se ejecutará independientemente de si ocurrió una excepción
print("fuera del bloque try y except")
```

Next Step

As we finish up this reading, you are ready to move on to the next part where you will practice handling errors. For better learning, try out different types of data in the lab. This way, you will encounter various errors and learn how to deal with them effectively. This knowledge will help you write stronger and more reliable code in the future.

Author(s)

Akansha Yadav