Errores y excepciones

En Python, los errores y excepciones son eventos que ocurren durante la ejecución de un programa y que interrumpen su flujo normal. Estos errores pueden deberse a diversas razones, como errores de sintaxis, problemas de tiempo de ejecución o situaciones excepcionales. Python proporciona un mecanismo de manejo de excepciones que permite detectar, controlar y manejar estos errores de manera elegante.

- **Errores de sintaxis:** Son errores que ocurren cuando el intérprete de Python encuentra un código que viola las reglas de la gramática del lenguaje. Por ejemplo, olvidar colocar dos puntos al final de una declaración if o usar una variable no definida. Estos errores se detectan durante la fase de análisis del código y generalmente impiden que el programa se ejecute.

INFORMATORIO
Hacia una mejor industria informática

- **Excepciones de tiempo de ejecución:** También conocidas como excepciones, son errores que ocurren durante la ejecución de un programa. Estas excepciones pueden ser causadas por diferentes situaciones, como dividir un número entre cero, intentar acceder a un índice fuera de rango en una lista o llamar a un método inexistente en un objeto. Cuando se produce una excepción, el flujo normal del programa se interrumpe y se busca un mecanismo de manejo adecuado para evitar la finalización abrupta.
- **Bloques try-except:** En Python, las excepciones se manejan mediante bloques try y except. El código sospechoso de generar una excepción se coloca dentro del bloque try, y si se produce una excepción, se busca un bloque except que coincida con el tipo de excepción lanzada. El bloque except especifica el código a ejecutar para manejar la excepción. Si no se encuentra un bloque except adecuado, la excepción se propaga hacia arriba en la pila de llamadas hasta que se encuentre un bloque except o el programa se termine.

```
try:
    # Código propenso a excepciones
except TipoDeExcepcion:
    # Código para manejar la excepción
```

La sentencia try funciona de la siguiente manera.

- Primero, se ejecuta la cláusula try (la(s) linea(s) entre las palabras reservadas try y la except).
- Si no ocurre ninguna excepción, la cláusula except se omite y la ejecución de la cláusula try finaliza.
- Si ocurre una excepción durante la ejecución de la cláusula try, se omite el resto de la cláusula. Luego, si su tipo coincide con la excepción nombrada después de la palabra clave except, se ejecuta la cláusula except, y luego la ejecución continúa después del bloque try/except.
- Si ocurre una excepción que no coincide con la indicada en la cláusula except se pasa a los try más externos; si no se encuentra un gestor, se genera una unhandled exception (excepción no gestionada) y la ejecución se interrumpe con un mensaje.

INFORMATORIO
Hacia una mejor industria informática

- **Bloque finally:** Opcionalmente, se puede agregar un bloque finally después de los bloques try y except. El código dentro del bloque finally siempre se ejecuta, ya sea que se produzca o no una excepción. Esto se utiliza para realizar tareas de limpieza o liberación de recursos que deben ocurrir independientemente del resultado de la ejecución.

```
try:
    # Código propenso a excepciones
except TipoDeExcepcion:
    # Código para manejar la excepción
finally:
    # Código que se ejecuta siempre
```

- **Excepciones personalizadas:** Además de las excepciones predefinidas en Python, puedes crear tus propias excepciones personalizadas para manejar situaciones específicas en tu programa. Para ello, puedes definir una clase que herede de la clase base **Exception** o cualquier otra excepción existente. Esto te permite lanzar y capturar tus propias excepciones según sea necesario.

INFORMATORIO

```
class MiExcepcion(Exception):
    pass

try:
    raise MiExcepcion("Mensaje de error")
except MiExcepcion as e:
    # Código para manejar la excepción personalizada
```

La declaración raise permite al programador forzar a que ocurra una excepción específica. El único argumento de raise indica la excepción a lanzar. Debe ser una instancia de excepción o una clase de excepción (una clase que derive de BaseException, como Exception o una de sus subclases). Si se pasa una clase de excepción, se instanciará implícitamente llamando a su constructor sin argumentos

INFORMATORIO
Hacia una mejor industria informática

Tipos de excepciones: # https://docs.python.org/3/library/exceptions.html

Los principales excepciones definidas en Python son:

- **TypeError** : Ocurre cuando se aplica una operación o función a un dato del tipo inapropiado.
- **ZeroDivisionError** : Ocurre cuando se itenta dividir por cero.
- **OverflowError** : Ocurre cuando un cálculo excede el límite para un tipo de dato numérico.
- **IndexError** : Ocurre cuando se intenta acceder a una secuencia con un índice que no existe.
- **KeyError** : Ocurre cuando se intenta acceder a un diccionario con una clave que no existe.
- **FileNotFoundError** : Ocurre cuando se intenta acceder a un fichero que no existe en la ruta indicada.
- **ImportError** : Ocurre cuando falla la importación de un módulo.

INFORMATORIO
Hacia una mejor industria informática

try - except - else

Para evitar la interrución de la ejecución del programa cuando se produce un error, es posible controlar la exepción que se genera con la siguiente instrucción:

```
try:
    bloque código 1
except excepción:
    bloque código 2
else:
    bloque código 3
```

Esta instrucción ejecuta el primer bloque de código y si se produce un error que genera una excepción del tipo excepción entonces ejecuta el segundo bloque de código, mientras que si no se produce ningún error, se ejecuta el tercer bloque de código.

INFORMATORIO
Hacia una mejor industria informática

Ejemplo:

```
def division(a, b):
    try:
        result = a / b
    except ZeroDivisionError:
        print('¡No se puede dividir por cero!')
    else:
        print(result)
>>> division(1, 0)
¡No se puede dividir por cero!
>>> division(1, 2)
0.5
```

Te invitamos a investigar más a fondo y probar todos estos tipos de errores y excepciones!

INFORMATORIO
Hacia una mejor industria informática