

MANEJO DE EXCEPCIONES EN PYTHON



💡 **Uso del bloque `else` y `finally` en el manejo de excepciones en Python**

Introducción

En esta lección aprenderás a utilizar los bloques opcionales `else` y `finally` junto con `try` y `except`. Estos bloques te permiten controlar aún mejor el flujo de tu programa, sabiendo exactamente cuándo se ejecutan y para qué sirven. 🛡️

📌 Paso 1: Crear el archivo del proyecto

📄 **Ruta y nombre del archivo:** `manejo-excepciones.py`

💡 Paso 2: Agregar el bloque `else`

🔍 Descripción:

El bloque `else` se ejecuta solo si **no ocurre ninguna excepción** en el bloque `try`.

📄 Archivo: `manejo-excepciones.py`

```
resultado = None

try:
    a = int(input('Proporciona el primer número: '))
    b = int(input('Proporciona el segundo número: '))
    resultado = a / b
except ZeroDivisionError as e:
    print(f'Ocurrió un error (ZeroDivisionError): {e}, tipo: {type(e)}')
except TypeError as e:
    print(f'Ocurrió un error (TypeError): {e}, tipo: {type(e)}')
except Exception as e:
    print(f'Ocurrió un error (Exception): {e}, tipo: {type(e)}')
else:
    print('No se arrojó ninguna excepción.')
```

📘 Explicación:

El bloque `else` permite ejecutar instrucciones si **todo salió bien** en el bloque `try`. No se ejecuta si ocurre un error.

🔄 Paso 3: Agregar el bloque `finally`

🔍 Descripción:

El bloque `finally` se ejecuta **siempre**, ocurra o no una excepción.

📄 Archivo: `manejo-excepciones.py`

```
finally:
    print('Ejecución del bloque finally')
```

📘 Explicación:

Este bloque es útil para cerrar archivos, liberar recursos o simplemente dejar un mensaje final. Siempre se ejecuta, incluso si ocurrió un error o si el programa terminó normalmente.

Paso 4: Realizar distintas pruebas

Descripción:

Probaremos el código con distintos valores para observar cuándo se ejecutan los bloques `else` y `finally`.

Casos sugeridos:

- Ingresar 10 y luego 2
→ Se ejecutan `else` y `finally`.
- Ingresar 10 y luego 0
→ Se ejecutan `except` y `finally` (no `else`).
- Ingresar a como primer número
→ Se ejecutan `except` y `finally` (no `else`).

Explicación:

Con esto confirmas el comportamiento de los bloques:

- `else`: solo si **no hay error**
- `finally`: **siempre se ejecuta**

Sección final: Código completo del archivo trabajado

Archivo trabajado: `manejo-excepciones.py`

```
resultado = None

try:
    a = int(input('Primer número: '))
    b = int(input('Segundo número: '))
    resultado = a/b
except ZeroDivisionError as e:
    print(f'ZeroDivisionError - Ocurrió un error: {e} , {type(e)}')
except TypeError as e:
    print(f'TypeError - Ocurrió un error: {e} , {type(e)}')
except Exception as e:
    print(f'Exception - Ocurrió un error: {e} , {type(e)}')
else:
    print('No se arrojó ninguna excepción')
finally:
    print('Ejecución del bloque finally')
```

```
print(f'Resultado: {resultado}')
print('Continuamos...')
```

Conclusión

En esta lección aprendiste a utilizar los bloques `else` y `finally` dentro del manejo de excepciones.

- `else` se ejecuta **solo si no hay errores**
- `finally` se ejecuta **siempre**, ocurra o no una excepción

Estos bloques te permiten escribir programas más completos, seguros y con una mejor estructura. 

¡Sigue practicando y dominarás el manejo de errores en Python como un profesional!   

Sigue adelante con tu aprendizaje  , ¡el esfuerzo vale la pena!

¡Saludos! 

Ing. Marcela Gamiño e Ing. Ubaldo Acosta

Fundadores de [GlobalMentoring.com.mx](#)