

MANEJO DE EXCEPCIONES EN PYTHON



Manejo de Excepciones en Python

Introducción

En esta lección aprenderás cómo manejar errores de ejecución en Python utilizando bloques `try` y `except`. Exploraremos el uso de clases de excepciones ya definidas en el lenguaje, la jerarquía entre ellas y cómo capturar errores para evitar que nuestro programa termine abruptamente. ✨

📌 Paso 1: Crear el proyecto de excepciones

Comenzamos creando una nueva carpeta para trabajar el tema de excepciones:

📁 Ruta sugerida:

Cursos/Python/Excepciones/Leccion01/

Una vez creado el proyecto, se generará el entorno virtual automáticamente.

Paso 2: Crear el archivo principal

Ahora creamos un nuevo archivo dentro de la carpeta. Este archivo contendrá los ejemplos para manejar errores.

Ruta y nombre del archivo:

```
manejo-excepciones.py
```

Paso 3: Simular una excepción

Descripción:

Vamos a provocar una excepción con una división entre cero para observar el mensaje de error que produce Python.

Archivo: `manejo-excepciones.py`

```
resultado = 10 / 0
```

Explicación:

Al ejecutar este código se genera una excepción del tipo `ZeroDivisionError`, ya que no es posible dividir entre cero.

Paso 4: Manejar la excepción con try-except

Descripción:

Envolvemos el código anterior en un bloque `try - except` para manejar el error sin que el programa termine abruptamente.

Archivo: `manejop-excepciones.py`

```
try:
    resultado = 10 / 0
except Exception as e:
    print(f'Ocurrió un error: {e}')
```

📖 Explicación:

Usamos la clase `Exception`, que es la clase padre para muchas otras excepciones. Capturamos el error con la variable `e` y mostramos el mensaje al usuario.

🎯 Paso 5: Capturar una excepción específica

🔍 Descripción:

Ahora hacemos más específico el bloque `except` para capturar solamente errores de tipo `ZeroDivisionError`.

📄 Archivo: `manejo-excepciones.py`

```
try:
    resultado = 10 / 0
except ZeroDivisionError as e:
    print(f'Ocurrió un error específico: {e}')
```

📖 Explicación:

Este código captura únicamente errores de tipo `ZeroDivisionError`, lo que es útil cuando queremos actuar solo ante ese tipo de error específico.

✅ Conclusión

En esta lección aprendiste cómo manejar excepciones en Python usando bloques `try` y `except`. Descubriste cómo atrapar errores en Python y continuaremos profundizando en este tema en las siguientes lecciones.

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

¡Saludos! 🙌

Ing. Marcela Gamiño e Ing. Ubaldo Acosta

Fundadores de [GlobalMentoring.com.mx](https://www.globalmentoring.com.mx)