

# Manejo de errores

El manejo de errores en Python se refiere al uso de estructuras y técnicas para detectar, gestionar y responder a errores (excepciones) que pueden ocurrir durante la ejecución de un programa. Esto se hace principalmente utilizando bloques *try*, *except*, *else* y *finally*.

## Try-Except

- try:
  - Contiene el código que podría causar una excepción.
  - Si ocurre una excepción, el flujo del programa se transfiere al bloque except.
- except:
  - Contiene el código que se ejecuta si se produce una excepción en el bloque try.
  - Puede especificar tipos de excepción para manejar diferentes tipos de errores de manera distinta.

El siguiente código muestra un tipo de error básico que se puede generar:

```
# Este código dará error, ya que se está
# realizando una división entre cero
numerador = 10
denominador = 0
numerador / denominador
```

```
ZeroDivisionError: division by zero
```

```
[1;31m-----[0m
```

```
[1;31mZeroDivisionError[0m                                Traceback (most recent call last)
```

```
Cell [1;32mIn[4], line 5[0m
```

```
[0;32m      3[0m numerador [38;5;241m=[39m [38;5;241m10[39m
```

```
[0;32m      4[0m denominador [38;5;241m=[39m [38;5;241m0[39m
```

```
[1;32m----> 5[0m numerador [38;5;241m/[39m denominador
```

```
[1;31mZeroDivisionError[0m: division by zero
```

El cual es error `ZeroDivisionError: division by zero`. Para evitar el error anterior, se realiza el manejo de la excepción de división por cero utilizando una estructura `try-except`.

#### 1. Bloque `try`:

- Se intenta realizar una división entre `numerador` y `denominador`:
- La división por cero genera una excepción en Python. En este caso, el código dentro del bloque `try` no se ejecutará correctamente debido a esta excepción.

#### 2. Bloque `except`:

- Se captura la excepción utilizando un bloque `except` que maneja cualquier tipo de error (en este caso, una excepción de división por cero):
- Cuando se produce una excepción, el programa no termina abruptamente, sino que se ejecuta el código dentro del bloque `except`, que imprime el mensaje `"Error"`.

```
numerador = 10
denominador = 0

try:
    division = numerador / denominador
except:
    print("Error")
```

Error

También podemos definir una variable `e` que captura la excepción generada:

#### 1. Bloque `try`:

- Se intenta realizar una división entre `numerador` y `denominador`:
- La operación `numerador / denominador` generará una excepción de división por cero porque `denominador` es igual a 0.

#### 2. Bloque `except`:

- Se captura la excepción utilizando un bloque `except` que maneja cualquier tipo de error y proporciona un mensaje detallado sobre el error:
- El `as e` captura la excepción y almacena el mensaje de error en la variable `e`. Luego, se imprime el mensaje de error con el formato `"Error: {e}"`, donde `{e}` es el mensaje de la excepción.

```
numerador = 10
denominador = 0

try:
    division = numerador / denominador
except Exception as e:
    print(f"Error: {e}")
```

Error: division by zero