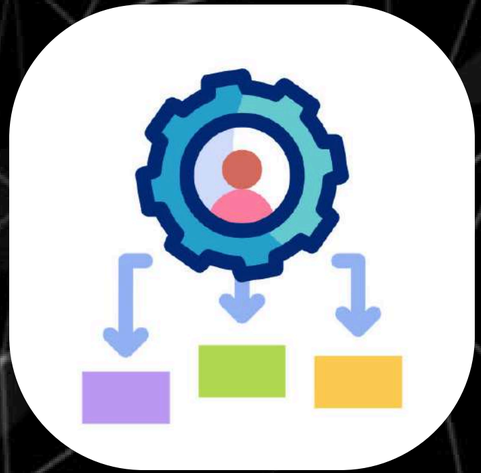
CURSO DE PYTHON DESDE CERO

¿COMO ES EL MANEJO DE ERRORES EN PYTHON?

¿QUÉ ES EL MANEJO DE ERRORES?

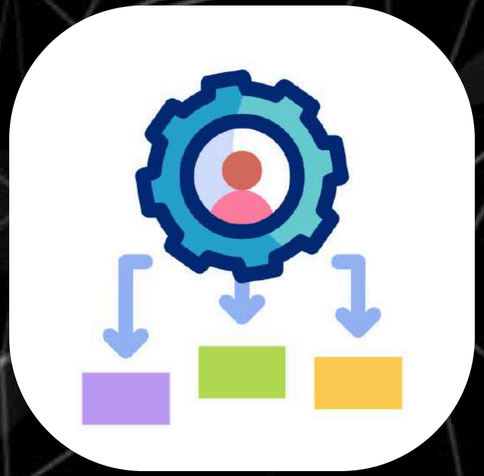
Es la forma de evitar que un programa se detenga cuando ocurre un error inesperado. (por ejemplo: dividir entre cero, escribir letras cuando se espera un número, etc).



🟦✔️ ¿QUÉ ES EL MANEJO DE ERRORES?

Python usa para esto:

👉 try, except, finally



¿POR QUÉ OCURREN ERRORES?

Algunos errores comunes:

- Dividir entre 0
- Convertir texto a número incorrectamente
- Acceder a un archivo que no existe
- Índices fuera de rango

✅ ESTRUCTURA BÁSICA

```
try:
    # Código que puede fallar
except:
    # Qué hacer si ocurre un error
finally:
    # Código que se ejecuta siempre
```

TRY → INTENTAR

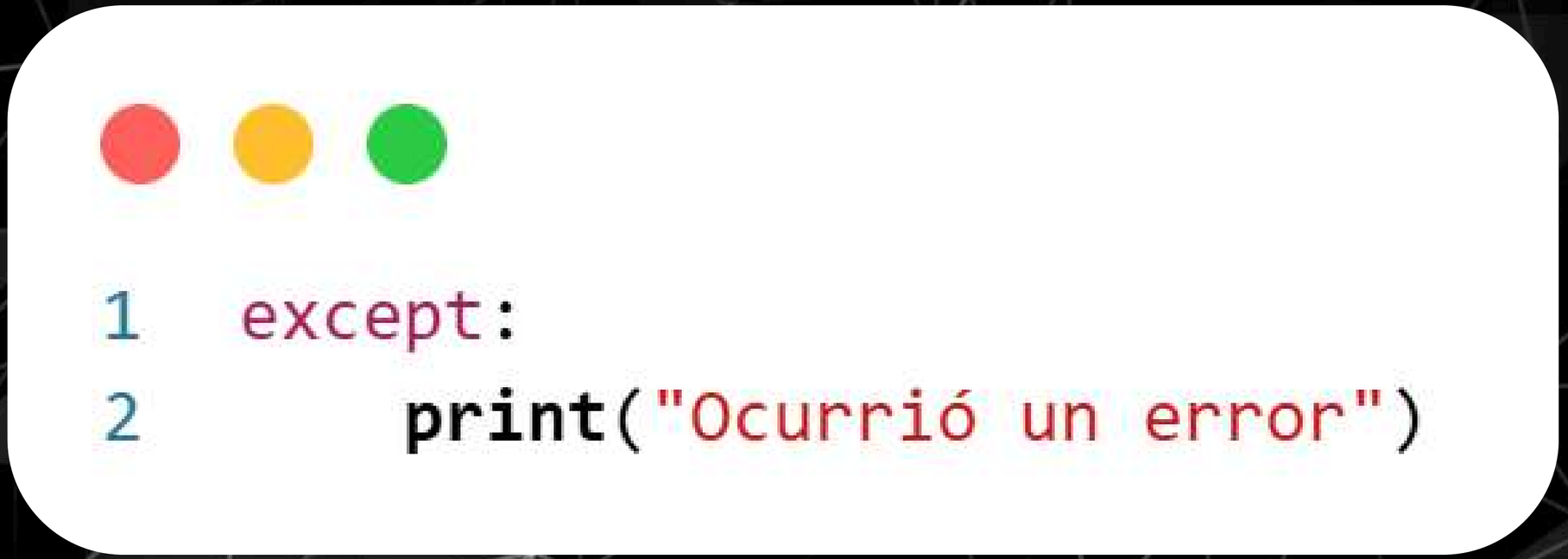
En try colocas el código que puede provocar un error.

📌 Python intenta ejecutar este bloque.

```python
try:
    numero = int(input("Ingresa un número: "))
    print(10 / numero)
```

EXCEPT → CAPTURAR EL ERRO

Si ocurre un error dentro de try, Python salta al except.
Evita que el programa se caiga y Permite mostrar un mensaje



```
except:
    print("Ocurrió un error")
```

EJEMPLO COMPLETO

```python
try:
    numero = int(input("Ingresa un número: "))
    print(10 / numero)
except:
    print("Entrada inválida o división por cero")
```

EXCEPT ESPECÍFICO

Puedes capturar errores específicos.

```python
try:
    numero = int(input("Ingresa un número: "))
    print(10 / numero)
except ValueError:
    print("Debes ingresar un número")
except ZeroDivisionError:
    print("No se puede dividir entre cero")
```

ELSE (OPCIONAL)

Se ejecuta solo si NO ocurrió ningún error.

```
1  try:
2      numero = int(input("Ingresa un número: "))
3      resultado = 10 / numero
4  except ZeroDivisionError:
5      print("Error: división por cero")
6  else:
7      print("Resultado:", resultado)
```

FINALLY → SIEMPRE SE EJECUTA

Se ejecuta haya error o no.

```
try:
    archivo = open("datos.txt", "r")
except FileNotFoundError:
    print("Archivo no encontrado")
finally:
    print("Fin del programa")
```

EJEMPLO REAL Y COMPLETO

```python
try:
    edad = int(input("Ingresa tu edad: "))
    print("Tu edad es:", edad)
except ValueError:
    print("Error: debes ingresar un número")
finally:
    print("Gracias por usar el programa")
```

✓ FLUJO VISUAL

- Python intenta ejecutar try
- Si hay error → entra al except
- Si no hay error → entra al else
- finally se ejecuta siempre

ERRORES COMUNES (EXCEPT)

- ValueError: Valor incorrecto para una operación (ej: int("abc"))
- TypeError: Tipo de dato incorrecto (ej: "5" + 2)
- IndexError: Índice fuera de rango
- NameError: Variable no definida

ERRORES COMUNES (EXCEPT)

- AttributeError: Atributo inexistente
- FileNotFoundError: Archivo no encontrado
- SyntaxError: Error de sintaxis
- IndentationError: Error de indentación

✅ BUENAS PRÁCTICAS

- ✓ Usa except específicos
- ✓ No abuses de except: sin tipo
- ✓ No ocultes errores importantes
- ✓ Usa finally para limpieza

🧠 RESUMEN SENCILLO

- try: Intenta ejecutar código
- except: Maneja el error
- else: Se ejecuta si no hay error
- finally: Se ejecuta siempre

NOS VEMOS EN UN PRÓXIMO VIDEO DE ESTE CURSO, SALUDOS 🚀