

# Manejo de excepciones

Pruebe la siguiente intrucción:

In [ ]:

```
5 / 0
```

Lo que se genera es un error de ejecución, que en Python se denominan "excepciones". Estas excepciones se indican en el mensaje de respuesta del interprete de Python (en este caso, ZeroDivisionError).

Para evitar que el interprete detenga la ejecución del programa y se pueda tomar alguna accion (como indicar con un mensaje de error, en este caso) se puede utilizar un bloque try...except que realiza una "prueba" en un bloque **try**: de alguna instrucción y evalua la respuesta de esta prueba.

Si la prueba no arroja excepciones, el interprete de Python continua con las siguientes lineas dentro del bloque y luego del bloque; sin embargo, si sucede un error se ejecutará lo que se encuentra dentro del bloque **except**:

In [ ]:

```
num, div = input("Ingrese una division: ").split('/')

try:
    result = int(num) / int(div)
    # Si se produce un error en la linea anterior, esta linea siguiente
    # no se ejecuta
    print("{} / {} = {:.2f}".format(num, div, result))
except ZeroDivisionError:
    print("ERROR: Division entre cero")
```

Hay varias excepciones estandar usadas comunmente:

- IOError: Error al tratar de acceder a un archivo
- ValueError: Error con el dato ingresado (no es un valor valido)
- EOFError: Error con el manejo de archivos (End Of File)
- ImportError: Error al momento de importar un modulo
- KeyboardInterrupt: Error por cancelar una operacion

El uso tipico de un bloque try...except es el manejo de los errores en un script sin tener que recurrir a operaciones condicionales que evaluen todos los posibles casos de error. Si sucede un error la ejecución del programa ira directamente al bloque **except**: y aqui se pueden evaluar directamente todos los errores posibles, lo que hace que el codigo se vea ordenado

Otra detalle que debe de observar es que los bloques de código en Python estan definidos por la sangría. Si alguna de las lineas del bloque no esta alineada a las demás, esta línea no esta en el bloque. Los editores de codigo manejan este orden de manera automática (luego de colocar los : para definir un bloque). Sin embargo, esta característica de Python hace que el código sea más legible.

## Ejercicio

Ejecute el programa líneas abajo, pero esta vez no escriba nada en el cuadro de entrada de datos o coloque solo un número. Observe el nombre de la excepción retornada por el intérprete de Python. Agregue un nuevo bloque de excepción que maneje este error con el mensaje "Debe de ingresar una división"

Sugerencia: Fíjese en las excepciones estándar indicadas líneas arriba y evalúe cuál usar para resolver este ejercicio

In [ ]:

```
num, div = input("Ingrese una division: ").split('/')

try:
    result = int(num) / int(div)
    # Si se produce un error en la línea anterior, esta línea siguiente
    # no se ejecuta
    print("{} / {} = {:.2f}".format(num, div, result))
except ZeroDivisionError:
    print("ERROR: Division entre cero")
```

In [ ]: