

Excepciones en Python

Los errores de ejecución son llamados comúnmente *excepciones*. Durante la ejecución de un programa, si dentro de una función surge una excepción y la función no la maneja, la excepción se propaga hacia la función que la invocó, si esta otra tampoco la maneja, la excepción continúa propagándose hasta llegar a la función inicial del programa y si esta tampoco la maneja se interrumpe la ejecución del programa. Veamos entonces como manejar excepciones.

- Manejo de excepciones
Para el manejo de excepciones los lenguajes proveen ciertas palabras reservadas, que nos permiten manejar las excepciones que puedan surgir y tomar acciones de recuperación para evitar la interrupción del programa

En el caso de Python, el manejo de excepciones se hace mediante los bloques que utilizan las sentencias try, except y finally.

A continuación, se ubica el bloque except:

```
>>> dividendo = 5
>>> divisor = 0
>>> dividendo / divisor
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
ZeroDivisionError: integer division or modulo by zero
```

En este caso, se levantó la excepción ZeroDivisionError cuando se quiso hacer la división. Para evitar que se levante la excepción y se detenga la ejecución del programa, se utiliza el bloque try-except.

```
>>> try:
...     cociente = dividendo / divisor
... except:
...     print "No se permite la división por cero"
...
```

Es importante destacar que, si bien luego de un bloque try puede haber varios bloques except, se ejecutará, a lo sumo, uno de ellos.

```
try:
    # aquí ponemos el código que puede lanzar excepciones
except IOError:
    # entrará aquí en caso que se haya producido
    # una excepción IOError
except ZeroDivisionError:
    # entrará aquí en caso que se haya producido
    # una excepción ZeroDivisionError
except:
    # entrará aquí en caso que se haya producido
    # una excepción que no corresponda a ninguno
    # de los tipos especificados en los except previos
```