**MANEJO DE EXCEPCIONES EN PYTHON**

En Python las excepciones se manejan por “jerarquía” eso significa que dependiendo del tipo de excepción que se “escriba” y el tipo de excepción que se “resulte”, los resultados serán distintos.

Por ejemplo:

try:

    resultado = a/input('introduce el divisor ')

    print(resultado)

except Exception as e:

    print(f'Error: "{e}"')

print('Next...')

En este caso, al utilizar la clase Exception, se ejecuta el código y se continúa la ejecución del programa.

En este otro caso, la situación es distinta. Al procesar una clase de excepción más específica, se interrumpe la ejecución del programa y la línea print(next…) no se llega a ejecutar.

try:

    resultado = a/input('introduce el divisor ')

    print(resultado)

except ZeroDivisionError as e:

    print(f'Error: "{e}"')

print('Next...')

**UN BLOQUE TRY EXCEPT PUEDE CONTENER VARIOS EXCEPT PARA PROCESAR MÁS DE UNA CLASE DE EXCEPCIONES**

**EL ORDEN DE EXCEPTIONS DEBE SER DE MAYOR ESPECIFICIDAD A MENOR.**

**DESCUBRIR TIPO DE EXCEPTION**

Una forma de “descubrir” que tipo de excepción es la que se está “atrapando” es imprimiendo el type() de excepción lanzado por la excepción de mayor jerarquía, o sea, menor especificidad

except Exception as e:

    print(f'Error: "{e}", {type(e)}')

de este modo obtenemos la <class> de la excepción lanzada

**BLOQUES ELSE Y FINALLY**

Los bloques else-finally son opcionales.

El bloque else se ejecuta si y solo si, no se arroja execpcion alguna.

El bloque finally se ejecuta al final del bloque try-except, independientemente del resultado de la misma.

Estos bloques pueden servir para arrojar mensajes al programador como “señalización”

**ARROJANDO EXCEPCIONES PERSONALIZADAS**

Para arrojar excepciones se utiliza la palabra reservada **raise**, esta se usa tanto para excepciones personalizadas como excepciones ya establecidas.

Para utilizar excepciones personalizadas es necesario crear una clase, como por ejemplo:

class ExcepcionPersonalizada(Exception):

    def \_\_init\_\_(self, mensaje) -> None:

        self.message = mensaje

Esta clase extiende de la clase Exception y sobrescribe el atributo **message**  de la misma.

Para utilizarla en el ejemplo anterior:

    if (a == b):

        raise ExcepcionPersonalizada('ambos numeros son identicos')

en caso de que los números introducidos sean iguales,se lanza esa excepción con el mensaje ‘ambos números son identicos’.

Esta excepción es manejada por este bloque:

except Exception as e:

    print(f'Error: "{e}", {type(e)}')