
python
try except

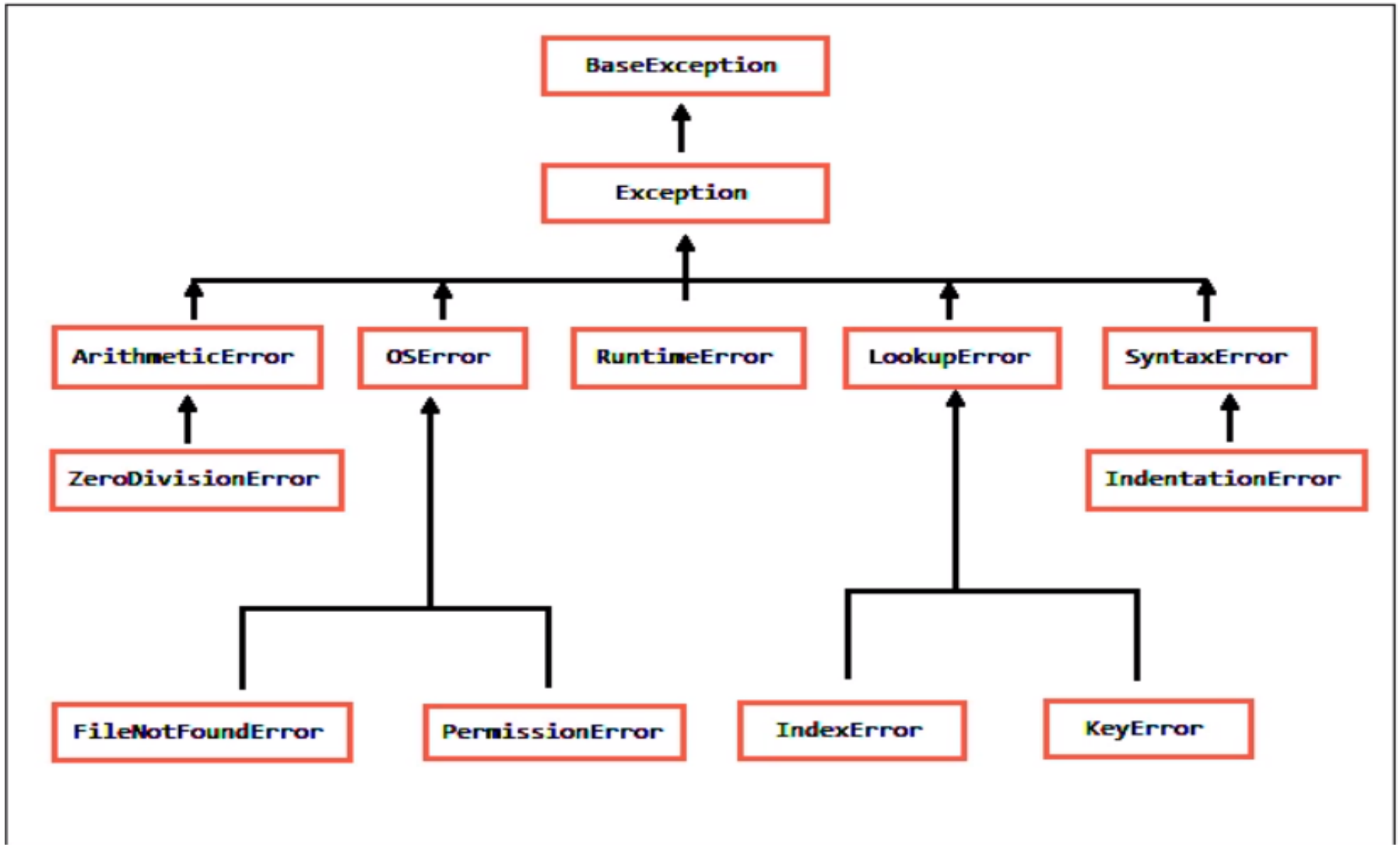
Manejo de excepciones

Las excepciones son errores que se disparan durante la ejecución de un programa. En Python podemos dejar que dichas excepciones detengan el programa o en caso contrario escribir un algoritmo para reaccionar a dicha situación.

Los ejemplos más comunes que podemos nombrar de excepciones :

- **Tratar de convertir a entero un string que no contiene valores numéricos**
- **Tratar de dividir por cero**
- **Abrir un archivo de texto inexistente o que se encuentre bloqueado por otra aplicación.**
- **Conectar con un servidor de Base de Datos que no se encuentra activo**
- **Acceder a subíndices de listas inexistentes**

Manejo de excepciones



Manejo de excepciones

```
try:
    10/0
except Exception as e:
    print(f'Ocurrió un error: {e}')
```

```
try:
    10/0
except ZeroDivisionError as e:
    print(f'Ocurrio un error: {e}')
```

```
resultado = None
a = '10'
b = 0
try:
    resultado = a/b
except Exception as e:
    print(f'Ocurrió un error: {e}')

print(f'Resultado: {resultado}')
print('Continuamos...')
```

Manejo de excepciones

```
resultado = None
try:
    a = '10’
    b = 0
    resultado = a/b
except ZeroDivisionError as e:
    print(f'ZeroDivisionError - Ocurrió un error: {e} , {type(e)}')
except TypeError as e:
    print(f'TypeError - Ocurrió un error: {e} , {type(e)}')
except Exception as e:
    print(f'Exception - Ocurrió un error: {e} , {type(e)}')

print(f'Resultado: {resultado}')
print('Continuamos...')
```

Bloques else y finally al manejar Excepciones

```
resultado = None
try:
    a = '10’
    b = 0
    resultado = a/b
except ZeroDivisionError as e:
    print(f'ZeroDivisionError - Ocurrió un error: {e} , {type(e)}')
except TypeError as e:
    print(f'TypeError - Ocurrió un error: {e} , {type(e)}')
except Exception as e:
    print(f'Exception - Ocurrió un error: {e} , {type(e)}’)
else:
    print(‘No se arrojó ninguna Excepción’)
finally:
    print(‘Ejecución del bloque finally’)

print(f'Resultado: {resultado}')
print('Continuamos...')
```

Creación de clases de Excepción personalizadas

```
class NumerosIdenticosException(Exception):

    def __init__(self, mensaje):
        self.message = mensaje
```

```
from NumerosIdenticosException import
NumerosIdenticosException

resultado = None
try:
    a = int(input('Primer número: '))
    b = int(input('Segundo número: '))
    if a == b:
        raise NumerosIdenticosException('números indénticos')
    resultado = a/b
except ZeroDivisionError as e:
    print(f'ZeroDivisionError - Ocurrió un error: {e} , {type(e)}')
except TypeError as e:
    print(f'TypeError - Ocurrió un error: {e} , {type(e)}')
except Exception as e:
  print(f'Exception - Ocurrió un error: {e} , {type(e)}')
else:
    print('No se arrojó ninguna excepción')
finally:
    print('Ejecución del bloque finally')

print(f'Resultado: {resultado}')
print('Continuamos...')
```

Manejo de excepciones

Problema 1

Realizar la carga de dos números enteros por teclado e imprimir su suma. Luego preguntar si quiere seguir sumando valores.
Codificar dos programas uno que capture la excepción de ingreso de datos numéricos y el otro que no tenga en cuenta el tipo de entrada de datos.

```
while True:
    try:
        valor1=int(input('Ingrese primer valor: '))
        valor2=int(input('Ingrese segundo valor: '))
        suma=valor1+valor2
        print('La suma es: ', suma)
    except ValueError:
        print('Debe ingresar numeros')

    respuesta=input(¿'Desea realizar la carga de otros valores? [s/n]')
    if respuesta=='n':
        break
```

Manejo de excepciones

Problema Propuestos

1. Realizar la carga de dos numeros por teclado e imprimir la división del primero respecto al segundo, capturar la excepción ZeroDivisionError

2. Almacenar en una lista los nombres de los meses del año. Solicitar el ingreso del número de mes y mostrar seguidamente el nombre de dicho mes. Capturar la excepción IndexError

3. Realizar la carga de dos numeros por teclado e imprimir la división del primero respecto al segundo , capturar las excepciones ZeroDivisionError y ValueError