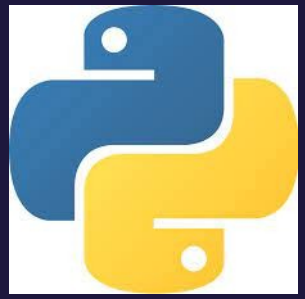
LENGUAJES y HERRAMIENTA PARA CIENCIAS DE DATOS I

Excepciones en Python



UCO ONLINE

Estructura excepción

- Última línea indica lo que sucedió
  - Tipo de error
  - Qué la causó
- Líneas anteriores
  - Contexto donde la excepción sucedió

```
>>> 8/0
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
ZeroDivisionError: division by zero
>>> 2 + '2'
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: unsupported operand type(s) for +: 'int' and 'str'
>>>
```

Manejo excepciones

- Capturar la excepción

```
try:
   bloque código propenso errores
except:
   bloque a ejecutar cuando se produce una excepción
```

```
1 try:
2     cociente = dividend/divisor
3 except:
4     print('Error: División por cero')
```

Manejo excepciones

- Capturar la excepción

```
try:
    dividendo = int(input('Dividendo: '))
    divisor = int(input('Divisor: '))
    cociente = dividendo/divisor
except ZeroDivisionError:
    print('Error: Division por cero')
except ValueError:
    print('Error: Tipo incorrecto')
```

Manejo excepciones

- Capturar la excepción

```python
try:
    dividendo = int(input('Dividendo: '))
    divisor = int(input('divisor: '))
    cociente = dividendo/divisor
except ZeroDivisionError:
    print('Error: División por cero')
except ValueError:
    print('Tipo incorrecto')
except:
    print('Error desconocido')
else:
    print(f'{dividendo}/{divisor}={cociente}')
```

Lanzar una excepción

- Sentencia **`raise`**
  - Forzar que ocurra una excepción
  - Usos
    - Propagar misma excepción hacia arriba
      - `raise`
    - Lanzar una excepción más significativa

```
raise <nombreExcepcion>
```

Lanzar una excepción

- Sentencia **`raise`**

```
def dividir(dividendo, divisor):
    try:
        cociente = dividendo/divisor
    except ZeroDivisionError:
        raise

dividendo = int(input('Dividendo: '))
divisor = int(input('Divisor: '))
try:
    dividir(dividendo, divisor)
except ZeroDivisionError:
    print('Error')
```

Acceso información contexto

- Función *exc_info*
  - Módulo sys

```python
import sys

def dividir(dividendo, divisor):
    try:
        cociente = dividendo/divisor
    except ZeroDivisionError:
        raise ZeroDivisionError('Division por cero')

dividendo = int(input('Dividendo: '))
divisor = int(input('Divisor: '))
try:
    dividir(dividendo, divisor)
except ZeroDivisionError:
    print(sys.exc_info())
```

```
(<class 'ZeroDivisionError'>, ZeroDivisionError('Division por cero',), <tracebac
k object at 0x7fe4486fcc08>)
```

Acceso información contexto

- Definir un identificador para la excepción
  - *except NombreExcepcion as identificador*
  - Utilizar el atributo `args`

```
try:
    dividir(dividendo, divisor)
except ZeroDivisionError as exc:
    print(exc.args)
```

```
mluque@hydrogen:~/ejemplos$ python3 errorSemantico.py
Dividendo: 2
Divisor: 0
('Division por cero',)
```

Tipos excepciones predefinidas

- TypeError
  - Tipo inapropidado
- Zero Division
  - División por cero
- OverflowError
  - Calculo excede límite tipo dato
- IndexError
  - Índice que no existe
- KeyError
  - Clave que no existe
- FileNotFoundError
  - Acceso fichero no existe
- ImportError
  - Al importar un módulo