T07_Excepciones

January 29, 2021

1 Python de cero a experto

Autor: Luis Miguel de la Cruz Salas

1.1 Pythonico es más bonito

1.1.1 Excepciones: *try, except, finally*

Tenemos dos tipos principales de errores: - De sintaxis: ocurren cuando no se escriben correctamente las expresiones y declaraciones, siguiendo la especificación de la interfaz de Python:

[1]:
```
prit('Hola mundo!')
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-1-131468bc977a> in <module>
----> 1 prit('Hola mundo!')

NameError: name 'prit' is not defined
```

Observe que el tipo de error se imprime cuando éste ocurre. En el caso anterior el error fue de tipo `NameError`, por lo que hay que revisar que todo esté correctamente escrito.

- Excepciones: Son errores lógicos, que detienen la ejecución de un programa aún cuando la sintaxis sea la correcta:

[2]:
```
def raizCuadrada(numero):
    numero = float(numero)
    print("La raíz cuadrada del número %f es %f" % (numero, numero ** 0.5))
```

[3]:
```
raizCuadrada(1)
```

```
La raíz cuadrada del número 1.000000 es 1.000000
```

[4]:
```
raizCuadrada(-1)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-4-e2d7ee78ed43> in <module>
----> 1 raizCuadrada(-1)

<ipython-input-2-11b51ac7fa7c> in raizCuadrada(numero)
      1 def raizCuadrada(numero):
      2     numero = float(numero)
----> 3     print("La raíz cuadrada del número %f es %f" % (numero, numero ** 0
 ↪5))

TypeError: can't convert complex to float
```

[5]:
```
raizCuadrada(1+1j)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-5-d8114a290366> in <module>
----> 1 raizCuadrada(1+1j)

<ipython-input-2-11b51ac7fa7c> in raizCuadrada(numero)
      1 def raizCuadrada(numero):
----> 2     numero = float(numero)
      3     print("La raíz cuadrada del número %f es %f" % (numero, numero ** 0
 ↪5))

TypeError: can't convert complex to float
```

[6]:
```
raizCuadrada("hola")
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-6-30a738df0326> in <module>
----> 1 raizCuadrada("hola")

<ipython-input-2-11b51ac7fa7c> in raizCuadrada(numero)
      1 def raizCuadrada(numero):
----> 2     numero = float(numero)
      3     print("La raíz cuadrada del número %f es %f" % (numero, numero ** 0
 ↪5))

ValueError: could not convert string to float: 'hola'
```

En los ejemplos anteriores hay errores es de tipo `TypeError`, es decir ocurrió un error con los tipos de datos que se están manipulando; y errores de tipo `ValueError`, es decir hay un problema con el

contenido del objeto.

1.1.2 Tipos de excepciones.

Todas las excepciones en Python son ejemplos concretos (objetos) de una clase (*instance*) que se derivan de la clase principal BaseExcepcion. Más detalles se pueden consultar aquí.

Las excepciones se pueden capturar y manejar adecuadamente. Para ello se tienen las siguientes herramientas:

- `try`
- `except`
- `else`
- `finally`

Cuando se identifica una sección de código susceptible de errores, ésta puede ser delimitada con la expresión `try`. Cualquier excepción que ocurra dentro de esta sección de código podrá ser capturada y gestionada.

La expresión `except` es la encargada de gestionar las excepciones que se capturan. Si se utiliza sin mayor información, ésta ejecutará el código que contiene para todas las excepciones que ocurran.

[7]:
```
def raizCuadrada(numero):
    try:
        numero = float(numero)
        print("La raíz cuadrada del número {numero} es {numero**0.5}")
    except:
        pass

    print('Gracias por usar Python!.')
```

[8]:
```
raizCuadrada(1)
```

```
La raíz cuadrada del número {numero} es {numero**0.5}
Gracias por usar Python!.
```

[9]:
```
raizCuadrada(-1)
```

```
La raíz cuadrada del número {numero} es {numero**0.5}
Gracias por usar Python!.
```

[10]:
```
raizCuadrada("hola")
```

```
Gracias por usar Python!.
```

1.1.3 Tratamiento de las excepciones

[11]:
```
def raizCuadrada(numero):
    ocurre_error = False
    try:
```

```
        numero = float(numero)
        print("La raíz cuadrada del número %f es %f" % (numero, numero ** 0.5))
    except:
        ocurre_error = True

    if ocurre_error:
        print("Hubo una falla en el programa, no se pudo realizar el cálculo")
    else:
        print('Gracias por usar Python!.')
```

[12]:
```
raizCuadrada(1)
```

```
La raíz cuadrada del número 1.000000 es 1.000000
Gracias por usar Python!.
```

[13]:
```
raizCuadrada(-1)
```

```
Hubo una falla en el programa, no se pudo realizar el cálculo
```

[14]:
```
raizCuadrada("hola")
```

```
Hubo una falla en el programa, no se pudo realizar el cálculo
```

Gestión de excepciones por su tipo. La expresión `except` puede ser utilizada de forma tal que ejecute código dependiendo del tipo de error que ocurra. Para más información de los tipos de error que existen en Python, consulte Concrete exceptions .

[15]:
```
def raizCuadrada(numero):
    ocurre_error = False
    try:
        numero = float(numero)
        print("La raíz cuadrada del número %f es %f" % (numero, numero ** 0.5))
    except TypeError:
        ocurre_error = True
        print("Ocurrió un error de tipo: TypeError, verifique que los tipos␣
 ↪sean compatibles.")
    except:
        ocurre_error = True
        print("Ocurrió algo misterioso")

    if ocurre_error:
        print("Hubo una falla en el programa, no se pudo realizar el cálculo")
    else:
        print('Gracias por usar Python!.')
```

[16]:
```
raizCuadrada(1)
```

```
La raíz cuadrada del número 1.000000 es 1.000000
Gracias por usar Python!.
```

[17]:
```
raizCuadrada(-1)
```

```
Ocurrió un error de tipo: TypeError, verifique que los tipos sean compatibles.
Hubo una falla en el programa, no se pudo realizar el cálculo
```

[18]:
```
raizCuadrada("hola")
```

```
Ocurrió algo misterioso
Hubo una falla en el programa, no se pudo realizar el cálculo
```

[19]:
```
raizCuadrada(numero)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-19-95dda1286cf1> in <module>
----> 1 raizCuadrada(numero)

NameError: name 'numero' is not defined
```

Información del error

[20]:
```
def raizCuadrada(numero):
    ocurre_error = False
    try:
        numero = float(numero)
        print("La raíz cuadrada del número %f es %f" % (numero, numero ** 0.5))
    except TypeError as detalles:
        ocurre_error = True
        print("Ocurrió un error (TypeError):", detalles)
    except ValueError as detalles:
        ocurre_error = True
        print("Ocurrió un error (ValueError):", detalles)
    except:
        ocurre_error = True
        print("Ocurrió algo misterioso")

    if ocurre_error:
        print("Hubo una falla en el programa, no se pudo realizar el cálculo")
    else:
        print('Gracias por usar Python!.')
```

[21]:
```
raizCuadrada(1)
```

```
La raíz cuadrada del número 1.000000 es 1.000000
Gracias por usar Python!.
```

[22]:
```
raizCuadrada(-1)
```

```
Ocurrió un error (TypeError): can't convert complex to float
Hubo una falla en el programa, no se pudo realizar el cálculo
```

[23]:
```
raizCuadrada('dd')
```

```
Ocurrió un error (ValueError): could not convert string to float: 'dd'
Hubo una falla en el programa, no se pudo realizar el cálculo
```

Sección `finally` Esta sección se ejecuta siempre, sin importar si hubo una excepción o no.

[24]:
```
def raizCuadrada(numero):
    ocurre_error = False
    try:
        numero = float(numero)
        print("La raíz cuadrada del número %f es %f" % (numero, numero ** 0.5))
    except TypeError as detalles:
        ocurre_error = True
        print("Ocurrió un error (TypeError):", detalles)
    except ValueError as detalles:
        ocurre_error = True
        print("Ocurrió un error (ValueError):", detalles)
    except:
        ocurre_error = True
        print("Ocurrió algo misterioso")
    finally:
        if ocurre_error:
            print("Hubo una falla en el programa, no se pudo realizar el␣
 ↪cálculo")
        else:
            print('Gracias por usar Python!.')
```

[25]:
```
raizCuadrada(1)
```

```
La raíz cuadrada del número 1.000000 es 1.000000
Gracias por usar Python!.
```

[26]:
```
raizCuadrada(-1)
```

```
Ocurrió un error (TypeError): can't convert complex to float
Hubo una falla en el programa, no se pudo realizar el cálculo
```

[27]:
```
raizCuadrada(1j)
```

```
Ocurrió un error (TypeError): can't convert complex to float
Hubo una falla en el programa, no se pudo realizar el cálculo
```

[28]:
```
raizCuadrada("hola")
```

```
Ocurrió un error (ValueError): could not convert string to float: 'hola'
Hubo una falla en el programa, no se pudo realizar el cálculo
```

[ ]:
```

```