



**IEEE
COMPUTER
SOCIETY**

*Universidad de San Carlos de Guatemala
Student Branch Chapter*

Introducción a Python

CLASE 5

Manejo de excepciones

Que es una excepcion?

Son errores que ocurren durante la ejecución del programa

Si una excepcion no es controlada se detiene la ejecución del programa

Si existe error en esta línea de código, se detendrá el programa



Solucion

Capturar o controlar la excepcion

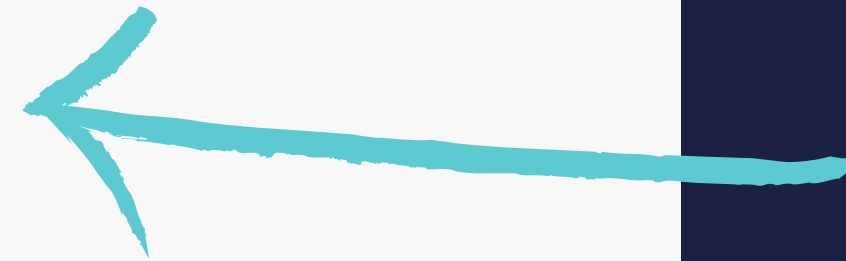
1. Buscar en la pila de llamadas la linea de error y el nombre/descripcion del error

Pila de llamadas



```
>>> print(1 / 0)  # Error al intentar dividir por 0.  
Traceback (most recent call last):  
  File "<stdin>", line 1, in <module>  
ZeroDivisionError: division by zero
```

Linea de error



Tipo de error



2. Ingresar el siguiente codigo

```
try:
```

bloque código 1

```
except excepción:
```

bloque código 2

Codigo que puede
contener el error

Tipo de
error

Codigo a ejecutar si
encuentra un error

Tipos de errores

TypeError

Ocorre cuando se aplica una operación o función a un dato del tipo inapropiado

ZeroDivisionError

Ocorre cuando se intenta dividir por cero

OverflowError

Ocorre cuando un calculo excede el límite para un tipo de dato numérico

IndexError

Ocorre cuando se intenta acceder a una secuencia con un indice que no existe

Tipos de errores

KeyError

Ocurre cuando se intenta acceder a un diccionario con una clave que no existe

FileNotFoundError

Ocurre cuando se intenta acceder a un fichero que no existe en la ruta indicada

ImportError

Ocurre cuando falla la importación de un módulo

Capturar varias excepciones

```
try:
    raise ValueError()
except ValueError:
    print("Value Error")
except KeyError:
    print("Key Error")
```

Nombre del
error

Nombre del
error

Podemos agregar
varios except para
capturar errores de
forma simultanea

Finally

Este código siempre se ejecutará
aunque no venga error

```
finally:  
    print("The try...except block is finished")
```

raise

Sirve para generar una excepcion.

Se define el tipo de error y el mensaje a mostrar al usuario

```
x = "hello"

if not type(x) is int:
    raise TypeError("Only integers are allowed")
```



Tipo de
error



Mensaje

Manejo de Archivos



Modos de Acceso

SOLO LECTURA

'r' --> abre el archivo únicamente para leerlo.

SOLO ESCRITURA

'w' --> abre el archivo únicamente para escribir (si no existe lo crea, si existe lo destruye).

ADJUNTAR

'a' --> abre el archivo para agregar información (si no existe lo crea).

Modos de Acceso

LECTURA Y ESCRITURA

'r+' --> abre el archivo para lectura/escritura (comienza al principio del archivo).

ESCRITURA Y LECTURA

'w+' --> abre el archivo para lectura/escritura, sobre-escribe el archivo si este ya existe o lo crea si no).

LECTURA Y ESCRITURA

'a+' --> abre el archivo para lectura/escritura (se sitúa al final del archivo).

Abrir y Cerrar un Archivo

OPEN()

Esta instrucción indica la apertura de un archivo.

SINTAXIS

```
archivo = open('nombreArchivo.extension', 'modoAcceso')
```

CLOSE()

Esta instrucción indica el cierre de un archivo.

SINTAXIS

```
archivo.close()
```

Manejar el Puntero

`SEEK()`

Ubica el puntero en la posición indicada.

`READ()`

Lee los caracteres hasta la posición indicada.

¡Muchas gracias!