

errores

January 28, 2020

0.1 Cometamos algunos errores

Todos cometemos errores, especialmente cuando estamos programando.

Los errores en un programa se llaman **errores** y existen dos tipos principales de errores: 1. Syntax errors (errores de sintaxis) 2. Runtime errors (errores de ejecucion)

En esta seccion veremos algunos ejemplos de errores de codigo y aprenderos como utilizar los datos de salida de Python para entender que error ocurrio y que pedazo de codigo lo ocasiono.

0.2 Errores de Sintaxis (Syntax Errors)

En sentido abierto, un error de sintaxis ocurre cuando escribe codigo que no es permitido en Python.

```
[1]: # Todos cometemos errores
      print('Hola Mundo)
```

```
File "<ipython-input-1-e8af2fd168ec>", line 2
print('Hola Mundo)
      ^
```

```
SyntaxError: EOL while scanning string literal
```

EOL se traduce End Of Line (Final de la Linea). Este mensaje nos dice que Python ha leído hasta el final de la línea sin encontrar el final de algo llamado **string literal**, que es texto contenido entre dos citaciones. Por ejemplo, el texto `Hola Mundo` es un ejemplo de un **string literal**. String literals son conocidos como **strings**.

0.3 Errores de Ejecucion (Runtime Errors)

IDLE atrapa errores de sintaxis antes que el programa se ejecute, pero algunos errores no se pueden atrapar sino hasta que el programa se ejecute. Estos errores se llaman **errores de ejecucion**, porque solo ocurren en el momento que un programa se ejecute.

Vamos a generar un error de ejecucion.

```
[3]: # Vamos a cambiar el codigo anterior a:
      print(Hola)
```

```

      □
↳ -----

NameError                                Traceback (most recent call↳
↳ last)

<ipython-input-3-580b8859920c> in <module>
      1 # Vamos a cambiar el codigo anterior a:
----> 2 print(Hola)

NameError: name 'Hola' is not defined

```

Que sucedio? Cuando el programa se ejecuta, Python **alza** o **raises** un error. Cuando un error ocurre, Python detiene la ejecucion del programa y produce el error en la ventana.

El texto del error se llama **traceback** o **rastro** toda vez que uno puede rastrear el error con ayuda del mensaje. Este rastro nos da informacion sobre el error: 1. El error ocurrio en la linea 2 2. La linea que genera el error es `print(Hola)` 3. Un `NameError` ocurre 4. El error especifico es que `Hola` no esta definido

Ya que `Hola` no tiene citas, Python no entiende que es un string de texto y asume que `Hola` es el nombre de alguna otra cosa en el codigo, y como `Hola` no ha sido definido en ninguna parte de nuestro codigo, el programa ocasiona el error.

0.4 Practica

1. Escribe codigo que produce un error de sintaxis (syntax error)
2. Escribe codigo que produce un error de ejecucion (runtime error)