

Errores en Python

En este punto del viaje, es posible que haya encontrado uno o dos errores al ejecutar el código. Aquí hay algunos errores comunes de Python:

- `SyntaxError`: esto ocurre cuando hay código Python no válido.
- `NameError`: esto ocurre cuando intentas utilizar una variable sin declararla primero.
- `TypeError`: esto ocurre cuando el tipo de datos que estás usando no se adapta a lo que estás intentando hacer.
- Durante tu aventura de codificación, estás destinado a encontrar innumerables pequeños y molestos errores rojos. Pero lo que hace grande a un programador no es evitar los errores, sino saber afrontarlos y resolverlos. 🐍

¡Exploremos estos a continuación!

- **Error de sintaxis** Uno de los errores más comunes es el `SyntaxError`, que ocurre cuando intentas ejecutar código que no es Python válido, como una palabra clave mal escrita, dos puntos faltantes :o un paréntesis de cierre faltante.

Por ejemplo:

```
print(Hello, World!)

# SyntaxError: invalid syntax
```

El fragmento anterior generará un mensaje `SyntaxError` cuando se ejecute porque la `print()` función requiere un paréntesis de cierre. Además, '¡Hola mundo!' debe estar entre comillas.

El mensaje de error completo podría verse así:

```
File "main.py", line 1
    print(Hello, World!
            ^
SyntaxError: invalid syntax
```

Describe File "main.py", line 1 el nombre del archivo y el número de línea. La pequeña flecha ^ señala donde se detectó el error.

Nota: La flecha ^ puede ser engañosa a veces porque es ahí donde el programa "piensa" que está el error; ¡Hay ocasiones en las que los errores ocurren antes de donde apunta!

Error de nombre

Otro error con el que te encontrarás a menudo es el `NameError`, que ocurre cuando intentas hacer referencia a una variable que aún no se ha creado; podría deberse a que escribiste mal el nombre de una variable o te olvidaste de definirla.

Por ejemplo:

```
print(greetings)

# NameError: name 'greetings' is not defined
```

El fragmento anterior arroja un `NameError` porque no habíamos definido una `greetings` variable. Podemos solucionar esto definiendo la variable de antemano: solución:

```
greetings = 'Howdy 🐍 '  
print(greetings)
```

Output: Howdy 🐍

Error de teclado

Otro error común que veremos es el `TypeError`. Cuando trabaje con variables de varios tipos de datos (por ejemplo, números, cadenas y valores booleanos), probablemente encontrará este error.

Por ejemplo:

```
message = 'The air quality is '  
print(message + 28)  
  
# TypeError: can only concatenate str (not "int") to str
```

La `message` variable es un tipo de datos de cadena. Si intentamos agregarle un número entero 28, `TypeError` se arrojará a. Esto se puede solucionar con algo como la función incorporada `str()` o encerrando el número entre comillas: solución

```
message = 'The air quality is '  
print(message + str(28))  
  
# Output: The air quality is 28
```

¡Eso es todo por ahora! Si te encuentras con un error que no está en la lista, asegúrate de visitar la documentación oficial de Python para ver el resto.

Y no olvide pegar también su mensaje de error en una búsqueda de Google.