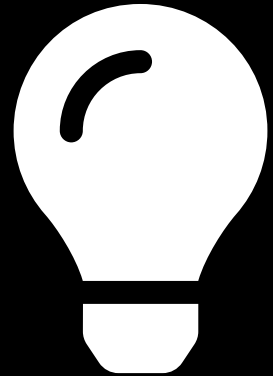


Programación con IA Generativa

Depuración de errores



Identificar y corregir **errores** o **mal funcionamiento** en el código solicitando ayuda a un asistente de código.









Índice

1. Depuración de errores
2. Depuración de un mensaje de error
3. Depuración de una salida incorrecta

1 | Depuración de errores

Depuración de errores

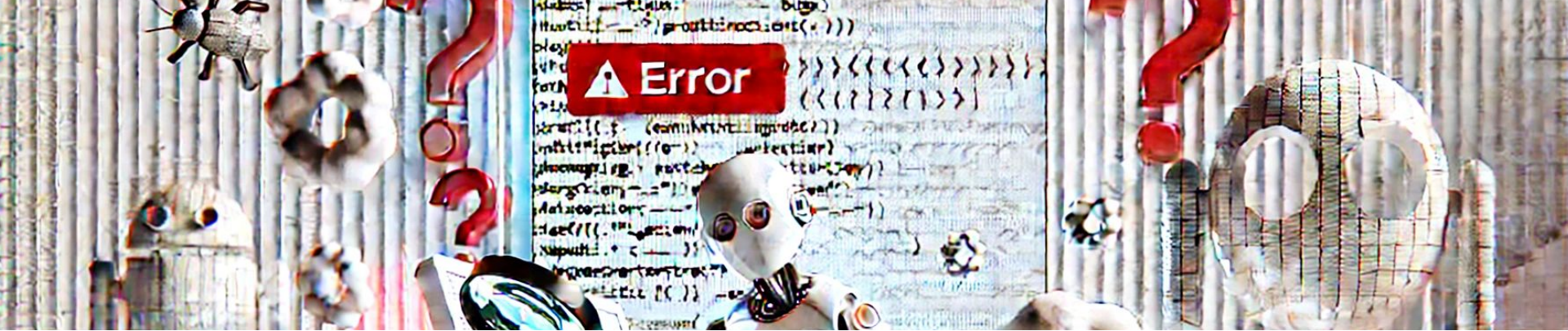
Hay dos situaciones principales al intentar ejecutar código con errores:

El código se cierra antes de que termine de ejecutarse y obtenemos un mensaje de error/excepción.

El código se ejecuta sin errores, pero la salida es diferente de la esperada.



2 | **Depuración de un mensaje de error o excepción**



Depuración de errores con un asistente de IA

Al ejecutar código se puede recibir un mensaje de error. El mensaje te indica el archivo y la línea donde se produjo el error y describe brevemente lo que salió mal. Sin embargo, los mensajes de error pueden resultar confusos. Para comprender y corregir completamente el error, podemos pedir ayuda a un asistente.

Depuración de errores



```
# Import the random module to easily generate pseudo-random numbers
import random
def dice_battle():
    # Generate random numbers between 1 and 6 for each player's die roll
    die_1 = random.randint(1, 6)
    die_2 = random.randint(1, 6)
    # Compare the die rolls and return the result as a string
    if die_1 > die_2:
        return "Player 1 rolled a " + die_1 + " and Player 2 rolled a " + die_2 + ". Player 1 wins!"
    elif die_1 < die_2:
        return "Player 1 rolled a " + die_1 + " and Player 2 rolled a " + die_2 + ". Player 2 wins!"
    else:
        return "Player 1 rolled a " + die_1 + " and Player 2 rolled a " + die_2 + ". It's a tie!"
print(dice_battle())
```

TypeError: can only concatenate str (not "int") to str

Depuración de errores

The screenshot shows the Visual Studio Code interface with a Python file named `ejemplo_bugs1.py` open. The code defines a `dice_battle` function that simulates a dice battle between two players. The function generates random numbers for each player's die roll and compares them to determine the winner. The code is as follows:

```
1 # Import the random module to easily generate pseudo-random numbers
2 import random
3
4 # Define a function that simulates a dice battle between two players
5 def dice_battle():
6
7     # Generate random numbers between 1 and 6 for each player's die roll
8     die_1 = random.randint(1, 6)
9     die_2 = random.randint(1, 6)
10
11     # Compare the die rolls and return the result as a string
12     if die_1 > die_2:
13         return f"Player 1 rolled a {die_1} and Player 2 rolled a {die_2}. Player 1 wins!"
14     elif die_1 < die_2:
15         return f"Player 1 rolled a {die_1} and Player 2 rolled a {die_2}. Player 2 wins!"
16     else:
17         return f"Player 1 rolled a {die_1} and Player 2 rolled a {die_2}. It's a tie!"
18
19 print(dice_battle())
```

The Chat window on the left shows the "Edit with Copilot" section. A prompt is entered in the chat input field:

Explique detalladamente por qué mi código produce el siguiente error y cómo puedo solucionarlo:
TypeError: can only concatenate str (not "int") to str

The Terminal window at the bottom shows the output of the script, which is a traceback of the error:

```
Traceback (most recent call last):
  File "c:\Users\jrabuch\up\ejemplo_bugs1.py", line 19, in <module>
    print(dice_battle())
    ~~~~~^
  File "c:\Users\jrabuch\up\ejemplo_bugs1.py", line 13, in dice_battle
    return "Player 1 rolled a " + die_1 + " and Player 2 rolled a " + die_2 + ". Playe
    ~~~~~^
TypeError: can only concatenate str (not "int") to str
PS C:\Users\jrabuch\up>
```

The status bar at the bottom indicates the current file is `main*`, the cursor is at line 12, column 22, and the file encoding is UTF-8. The Python version is 3.11.7 (anaconda3: conda).

Depuración de errores

CHAT

GitHub Copilot

Te explicaré paso a paso el problema y la solución:

1. El error `TypeError: can only concatenate str (not "int")` ocurre cuando intentas concatenar (unir) una cadena de texto (string) con un número entero (int) directamente.
2. En tu código, aunque estás usando f-strings (lo cual es correcto), el error probablemente ocurre porque estás ejecutando el código en una versión anterior de Python que no soporta f-strings (fueron introducidas en Python 3.6+).
3. Para solucionar esto, hay dos opciones:
 - Asegurarte de usar Python 3.6 o superior (recomendado)
 - Modificar el código para usar el método `.format()` o la concatenación tradicional con conversión explícita a string

Aquí está la solución usando el método `.format()` que funcionará en versiones anteriores de Python:

[ejemplo_bugs1.py](#)

Modificar los strings de retorno para usar `.format()` en lugar de f-strings:

[ejemplo_bugs1.py](#) +3 -3

1 file changed

[Keep](#) [Undo](#) [📄](#)

[Add Context...](#) [ejemplo_bugs1.py](#) Current file [👁](#)

[🔊](#) Edit Claude 3.5 Sonnet [👇](#) [👆](#)

ejemplo_bugs1.py

```
1 # Import the random module to easily generate pseudo-random numbers
2 import random
3
4 # Define a function that simulates a dice battle between two players
5 def diceBattle():
6
7     # Generate random numbers between 1 and 6 for each player's die roll
8     die_1 = random.randint(1, 6)
9     die_2 = random.randint(1, 6)
10
11     # Compare the die rolls and return the result as a string
12     if die_1 > die_2:
13         return f"Player 1 rolled a {die_1} and Player 2 rolled a {die_2}. Player 1 wins!"
14     elif die_1 < die_2:
15         return f"Player 1 rolled a {die_1} and Player 2 rolled a {die_2}. Player 2 wins!"
16     else:
17         return f"Player 1 rolled a {die_1} and Player 2 rolled a {die_2}. It's a tie!"
18     return "Player 1 rolled a {} and Player 2 rolled a {}. It's a tie!".format(die_1, die_2)
19 print(diceBattle())
```

[Keep](#) [Undo](#) [📄](#) | 1 of 3 [↑](#) [↓](#) [🔍](#)

Ln 19, Col 21 Spaces: 4 UTF-8 {} Python 3.11.7 (anaconda3: conda)

3 | Depuración de una salida incorrecta



Depuración de una salida/funcionalidad incorrecta

A veces, se ejecuta código con errores sin producir ningún error, pero la salida es claramente incorrecta. En este caso, la depuración puede ser más difícil porque un compilador no puede indicarle la ubicación ni la descripción del error.

Para estos errores "invisibles", un asistente de IA es particularmente útil.

Depuración de una salida/funcionalidad incorrecta



```
# Initialize the factorial result to 1
factorial = 1

# Initialize the input number to 6
number = 6

# Loop from 1 to number (inclusive) and multiply factorial by each number
for i in range(1, number + 1):
    factorial *= factorial * i

print(f"The factorial of {number} is {factorial}")
```

Desafortunadamente, **el código no funciona según lo esperado**. Queremos que devuelva 720, el valor correcto de 6 factoriales, pero la salida es mucho mayor que esa.

Depuración de una salida/funcionalidad incorrecta

The screenshot shows the Visual Studio Code interface with a Python file named `ejercicio_bugs_2.py` open. The code calculates the factorial of 6, but the output in the terminal is incorrect.

```
C: > Users > jrabuch > Downloads > ejercicio_bugs_2.py > ...
1 # Initialize the factorial result to 1
2 factorial = 1
3
4 # Initialize the input number to 6
5 number = 6
6
7 # Loop from 1 to number (inclusive) and multiply factorial by each number
8 for i in range(1, number + 1):
9     factorial *= factorial * i
10
11 print(f"The factorial of {number} is {factorial}")
```

The terminal output shows:

```
PS C:\Users\jrabuch\up> & C:/Users/jrabuch/AppData/Local/anaconda3/python.exe c:/Users/jrabuch/Downl
oads/ejercicio_bugs_2.py
The factorial of 6 is 16511297126400
PS C:\Users\jrabuch\up>
```

The chat window on the left shows the following message:

¿Por qué el resultado de este código es mucho mayor de lo esperado? Explicámelos detalladamente y sugiera una solución.

The status bar at the bottom indicates the file is at line 11, column 51, with 4 spaces, UTF-8 encoding, Python language, and 3.11.7 (anaconda3: conda) version.

Depuración de una salida/funcionalidad incorrecta

The screenshot shows the Visual Studio Code interface with a Python file named `ejercicio_bugs_2.py` open. The code is a factorial function that contains a bug. The left sidebar shows the 'CHAT' panel with a message explaining the bug and a list of iterations. The bottom panel shows the 'TERMINAL' panel with the output of the program.

CHAT

Esta línea está multiplicando el factorial por sí mismo y luego por `i` en cada iteración, lo que causa que el número crezca exponencialmente más de lo necesario.

Por ejemplo, para número 6, las iteraciones serían:

1. factorial = $1 * 1 * 1 = 1$
2. factorial = $1 * 1 * 2 = 2$
3. factorial = $2 * 2 * 3 = 12$
4. factorial = $12 * 12 * 4 = 576$
5. factorial = $576 * 576 * 5 = 1,658,880$
6. factorial = $1,658,880 * 1,658,880 * 6 = 16,511,297,126,400$

Cuando el factorial de 6 debería ser simplemente: $6 * 5 * 4 * 3 * 2 * 1 = 720$

La solución es simplemente multiplicar el factorial por `i` en cada iteración:

`ejercicio_bugs_2.py`

`ejercicio_bugs_2.py` +1 -1

Con este cambio, el programa calculará correctamente el factorial multiplicando cada número una sola vez.

1 file changed

`ejercicio_bugs_2.py` C:\Users\jrabuch\Downloads

Add Context... `ejercicio_bugs_2.py` Current file

Edit files in your workspace

Edit Claude 3.5 Sonnet

ejercicio_bugs_2.py

```
C: > Users > jrabuch > Downloads > ejercicio_bugs_2.py > ...
1 # Initialize the factorial result to 1
2 factorial = 1
3
4 # Initialize the input number to 6
5 number = 6
6
7 # Loop from 1 to number (inclusive) and multiply factorial by i
8 for i in range(1, number + 1):
9     factorial *= factorial * i
10
11 print(f"The factorial of {number} is {factorial}")
```

Keep Undo 1 of 1

PROBLEMS 52 OUTPUT DEBUG CONSOLE TERMINAL PORTS

Python powershell Deactivate
Python
Python

Ln 11, Col 51 Spaces: 4 UTF-8 Python 3.11.7 (anaconda3: conda)

Actividad en grupo

Armar un ejemplo con el asistente de código utilizando uno de los dos tipos de resolución de depuración de errores:

- Al mostrar mensaje de error
- Al obtener una salida/funcionalidad incorrecta

20:05 a 20: 25 - Actividad en equipos
20:25 volvemos a la sala principal

