

¿Cómo saber si mi tarea está
buena?

Por Cristóbal Sepúlveda Cobo

The image shows a screenshot of the IDLE Python IDE. The window title is "ejemplo.py - /Users/user". The menu bar includes "IDLE", "File", "Edit", "Format", "Run", "Window", and "Help". The "Run" menu is open, showing options: "Python Shell", "Check Module ⌘X", and "Run Module F5". The code in the editor is as follows:

```
#suma: int int -> int
#suma dos numeros
#ej: suma(2,3) da 5
def suma(x,y):
    return x+y

#Test
assert suma(2,3)==5
assert suma(1,2)==3
```

The status bar at the bottom right shows "Ln: 9 Col: 19".

- Si hicieron los tests suficientes, y su función hace lo que se supone que debe, entonces deben correr el módulo (o en su defecto, oprimir F5)

The screenshot shows the IDLE Python Shell window. The menu bar includes 'IDLE', 'File', 'Edit', 'Shell', 'Debug', 'Window', and 'Help'. The window title is 'ejemplo.py - /Users/user/Docu... Python Shell'. The left pane contains the following Python code:

```
#suma: int int -> int
#suma dos numeros
#ej: suma(2,3) da 5
def suma(x,y):
    return x+y

#Test
assert suma(2,3)==5
assert suma(1,2)==3
```

The right pane shows the Python Shell output:

```
Python 2.7 (r27:82508, Jul 3 2010, 20:17:05)
[GCC 4.0.1 (Apple Inc. build 5493)] on darwin
Type "copyright", "credits" or "license()" for more information.
>>> ===== RESTART =====
>>>
>>> |
```

The status bar at the bottom right indicates 'Ln: 6 Col: 4'.

- Si al correrlo aparece esta nueva ventana y no se suma ningún problema, entonces no hay errores evidentes y pueden entregar la tarea si están conformes con su trabajo :D

The screenshot shows the IDLE Python Shell interface. The left pane contains the source code for a Python script named 'ejemplo.py'. The code defines a function 'suma(x,y)' that returns 'x-y'. It includes test assertions: 'assert suma(2,3)==5' and 'assert suma(1,2)==3'. The right pane shows the Python Shell output, which includes the Python version (2.7), GCC version (4.0.1), and a traceback error. The error message is 'AssertionError' at line 8, column 4, in the file '/Users/user/Documents/ejemplo.py'. The shell prompt is '>>>'.

```
#suma: int int -> int
#suma dos numeros
#ej: suma(2,3) da 5
def suma(x,y):
    return x-y

#Test
assert suma(2,3)==5
assert suma(1,2)==3
```

```
Python 2.7 (r27:82508, Jul 3 2010, 20:17:05)
[GCC 4.0.1 (Apple Inc. build 5493)] on darwin
Type "copyright", "credits" or "license()" for more information.
>>> ===== RESTART =====
>>>
>>> ===== RESTART =====
>>>

Traceback (most recent call last):
  File "/Users/user/Documents/ejemplo.py", line 8, in <module>
    assert suma(2,3)==5
AssertionError
>>>
```

Ln: 13 Col: 4

- Por otro lado puede que aparezca un error 😞
- En este caso el error es del assert.

The screenshot shows the Python IDLE interface. The left pane contains the source code for a file named 'ejemplo.py'. The code defines a function 'suma' that takes two integers and returns their difference. It also includes a test section with two assertions: 'assert suma(2,3)==5' and 'assert suma(1,2)==3'. The right pane shows the Python Shell output, which includes the Python version (2.7), GCC version (4.0.1), and a traceback for an 'AssertionError' raised at line 8 of the script. The error message is 'assert suma(2,3)==5'. The shell also shows the execution of 'suma(2,3)' returning -1.

```
Python 2.7 (r27:82508, Jul 3 2010, 20:17:05)
[GCC 4.0.1 (Apple Inc. build 5493)] on darwin
Type "copyright", "credits" or "license()" for more information.
>>> ===== RESTART =====
>>>

Traceback (most recent call last):
  File "/Users/user/Documents/ejemplo.py", line 8, in <module>
    assert suma(2,3)==5
AssertionError
>>> suma(2,3)
-1
>>> |
```

- Puede ser que el test esté mal hecho, o que su función esté fallando.
- Para ver qué es qué, intenten ejecutar su función en el IDLE y ver que entrega.
- Vemos que en este caso la función es la mala :o

The image shows a screenshot of the Python IDLE environment. The window title is "ejemplo.py - /Users/user/Documents/ejemplo.py". The left pane contains the following Python code:

```
#suma: int int -> int
#suma dos numeros
#ej: suma(2,3) da 5
def suma(x,y):
    return x-y
|
#Test
assert suma(2,3)==5
assert suma(1,2)==3
```

The right pane, titled "Python Shell", shows the execution output:

```
1 3 2010, 20:17:05)
ild 5493)] on darwin
s" or "license()" for more information.
===== RESTART =====

ll last):
ents/ejemplo.py", line 8, in <module>
```

The status bar at the bottom of the editor shows "Ln: 6 Col: 0" and the Python Shell status bar shows "Ln: 13 Col: 4".

- Como el error es lógico, solo nos quedará analizar la función hasta encontrar el error.

```
Apple IDLE File Edit Shell Debug Window Help
ejemplo.py - /Users/user/Documents Python Shell

#suma: int int -> int
#suma dos numeros
#ej: suma(2,3) da 5
def suma(x,y):
    print x+y

#Test
assert suma(2,3)==5
assert suma(1,2)==3

Python 2.7 (r27:82508, Jul 3 2010, 20:17:05)
[GCC 4.0.1 (Apple Inc. build 5493)] on darwin
Type "copyright", "credits" or "license()" for more information.
>>> ===== RESTART =====
>>>
5

Traceback (most recent call last):
  File "/Users/user/Documents/ejemplo.py", line 8, in <module>
    assert suma(2,3)==5
AssertionError
>>>
```

Ln: 12 | Col: 4

- Puede ser que el `assert` falle también porque `estamos imprimiendo el resultado en lugar de entregarlo.`
- Esto se arregla cambiando el `print` por un `return`.

```
Apple IDLE File Edit Shell Debug Window Help
ejemplo.py - /Users/ Python Shell

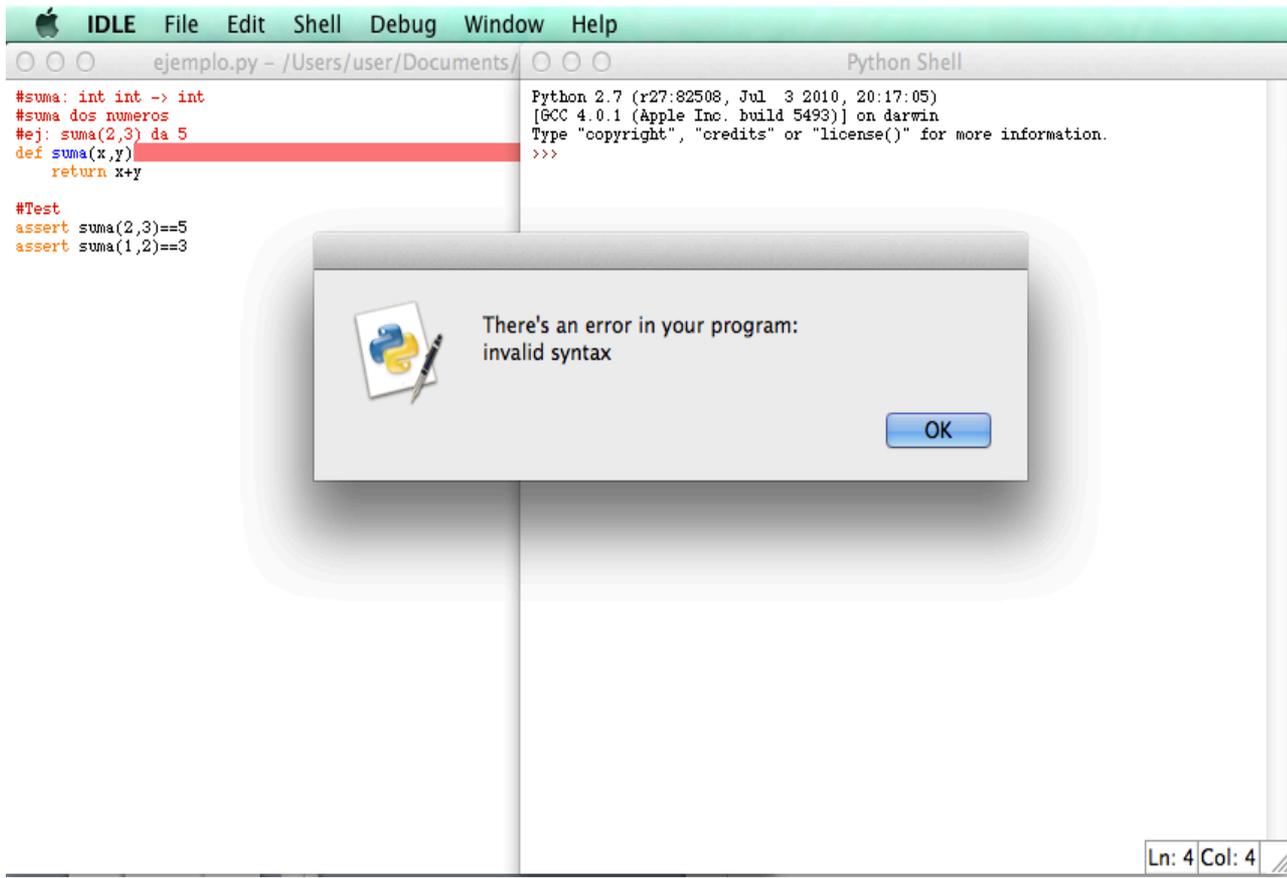
#suma: int int -> int
#suma dos numeros
#ej: suma(2,3) da 5
def suma(x,y):
    return x+y

#Test
assert suma(2,3)==5
assert suma(1,2)==3

Python 2.7 (r27:82508, Jul 3 2010, 20:17:05)
[GCC 4.0.1 (Apple Inc. build 5493)] on darwin
Type "copyright", "credits" or "license()" for more information.
>>> ===== RESTART =====
>>>
Traceback (most recent call last):
  File "/Users/user/Documents/ejemplo.py", line 8, in <module>
    assert suma(2,3)==5
AssertionError
>>> |
```

Ln: 11 Col: 4

- También puede darse que estemos comparando distintos tipos de datos.
- Como un string con un int.



- Otros errores pueden ser que la sintaxis esté mal.
- En este caso python nos señala donde está el error

```
#suma: int int -> int
#suma dos numeros
#ej: suma(2,3) da 5
def suma(x,y):
return x+y

#Test
assert suma(2,3)==5
assert suma(1,2)==3
```

Python 2.7 (r27:82508, Jul 3 2010, 20:17:05)
[GCC 4.0.1 (Apple Inc. build 5493)] on darwin
Type "copyright", "credits" or "license()" for more information.
>>>

There's an error in your program:
expected an indented block

OK

Ln: 4 Col: 4

- Igualmente puede ser un error de indentación.
- Como Python es buena onda nos dice donde está el error :D

Otros errores...

```
Python 2.7 (r27:82508, Jul 3 2010, 20:17:05)
[GCC 4.0.1 (Apple Inc. build 5493)] on darwin
Type "copyright", "credits" or "license()" for more information.
>>> ===== RESTART =====
>>>

Traceback (most recent call last):
  File "/Users/user/Documents/ejemplo.py", line 8, in <module>
    assert suma("2",3)==5
  File "/Users/user/Documents/ejemplo.py", line 5, in suma
    return x+y
TypeError: cannot concatenate 'str' and 'int' objects
>>>
```

- Error de tipo.

```
Python 2.7 (r27:82508, Jul 3 2010, 20:17:05)
[GCC 4.0.1 (Apple Inc. build 5493)] on darwin
Type "copyright", "credits" or "license()" for more information.
>>> ===== RESTART =====
>>>

Traceback (most recent call last):
  File "/Users/user/Documents/ejemplo.py", line 8, in <module>
    assert suma(2,3)==5
  File "/Users/user/Documents/ejemplo.py", line 5, in suma
    return x+z
NameError: global name 'z' is not defined
>>> |
```

- Error de nombre.

```
Python 2.7 (r27:82508, Jul 3 2010, 20:17:05)
[GCC 4.0.1 (Apple Inc. build 5493)] on darwin
Type "copyright", "credits" or "license()" for more information.
>>> ===== RESTART =====
>>>

Traceback (most recent call last):
  File "/Users/user/Documents/ejemplo.py", line 8, in <module>
    assert suma(2,"tres")==5
  File "/Users/user/Documents/ejemplo.py", line 5, in suma
    return x+int(y)
ValueError: invalid literal for int() with base 10: 'tres'
>>> |
```

- Error de valor.

- ¡Si todo sale bien podemos subir la tarea a U-Cursos! :D
- Es aconsejable que estén subiendo periódicamente su trabajo por si acaso saliera cualquier percance a último minuto que les evitara subir su tarea u.u (como que se les corté la luz, el internet, su computador muera o comience el apocalipsis)
- Por si acaso siempre es recomendable descargar la tarea que subieron para verificar que es el archivo correcto :o