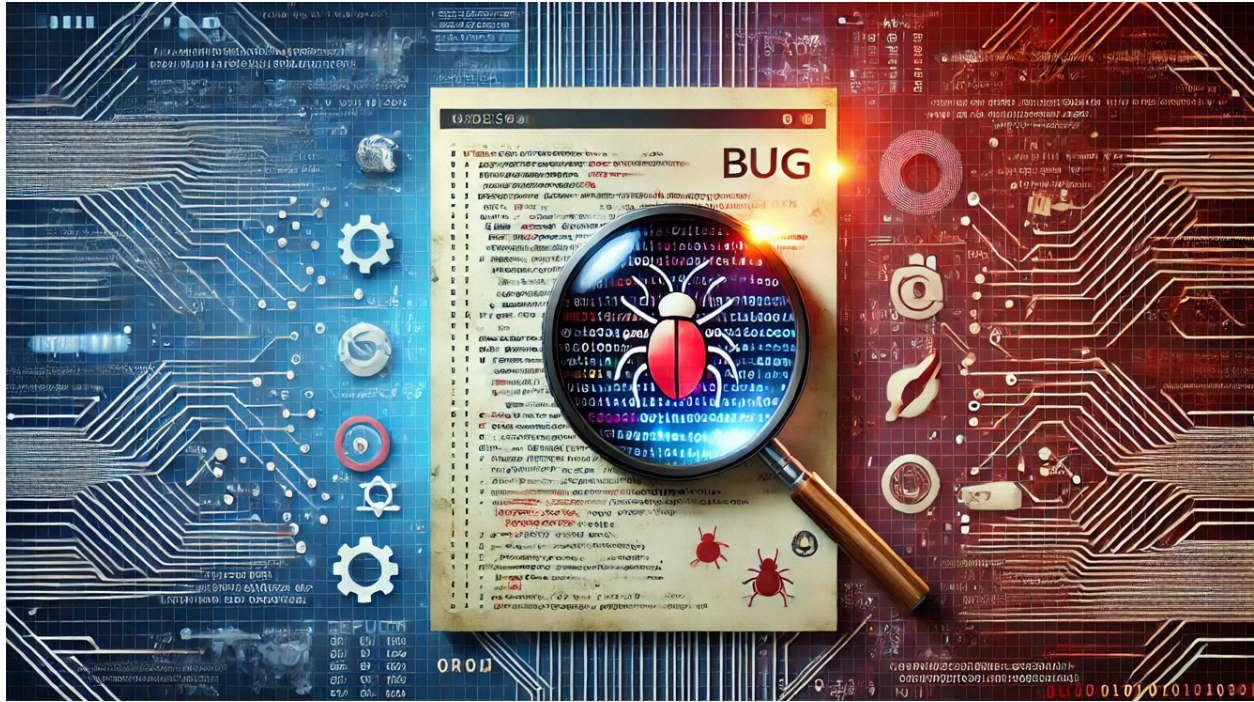


Exercici 4 python. Debugging

Jaume Albert Salas Pastor



Exercici 4 python. Debugging

1. Strings.py	1
Errores:	2
Punto de interrupción:	3
Solución:	3
2. Classes.py	4
Errores:	4
Punto de interrupción:	5
Solución:	5
3. Simple_calculator.py	6
Errores:	6
Punto de interrupción:	7
Solución:	7
4. Factorial.py	8
Errores:	8
Punto de interrupción:	9
Solución:	9

1. Strings.py

Errores:

Nos encontramos en un caso donde el ide ya nos detecta varios errores de sintaxis:

```
1  year == int.input('Greetings! What is your year of origin?')
2
3  if year <= 1900
4      print('Woah, that's the past!')
5  elif year > 1900 && year < 2020:
6      print("That's totally the present!")
7  elif:
8      print("Far out, that's the future!!")
```

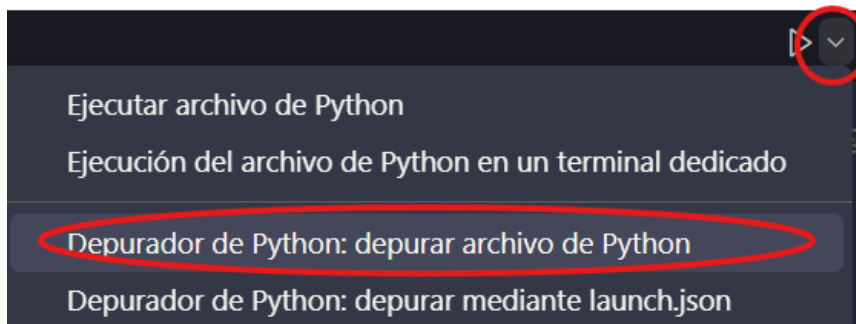
- El primer error que podemos detectar es “year” en la primera línea, no está bien definida ya que hay dos “=” y solamente tendría que haber uno.
- En segundo lugar, la cadena de texto de la línea 1 debe de estar entre comillas dobles y además, en “int.input” hay que sustituir el punto por un paréntesis de apertura“(”.
- En tercer lugar vemos que en la línea 3, el constructor if está mal creado ya que se espera encontrar dos puntos “:” al final de la línea.
- En la línea 4, hay que sustituir las comillas simples por dobles.
- En la línea 5, hay dos “&&” y hay que quitar uno (esto no es java).
- Y para terminar en la septima línea, hay que cambiar el “elif” por un “else”

Punto de interrupción:

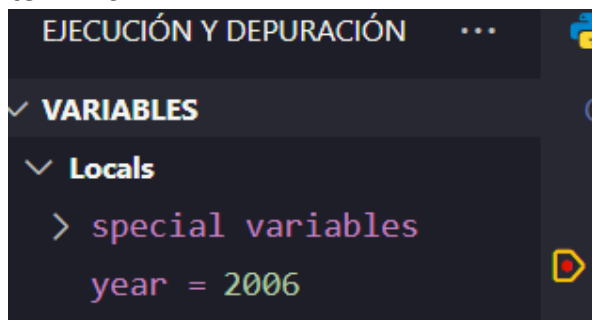
Ponemos un punto de interrupción en la línea 3 para saber el valor de "year":

```
1 year = int(input("Greetings! What is your year of origin? "))
2
3 if year <= 1900:
4     print("Woah, that's the past!")
5 elif year > 1900 & year < 2020:
6     print("That's totally the present!")
```

Ejecutamos como depurador:



Vemos que el valor es 2006 ya que es el año que le hemos insertado por terminal:



Solución:

Así se vería el código con las correcciones pertinentes previamente mencionadas:

```
string.py X
C: > Users > alumne-DAM > Desktop > string.py > ...
1 year = int(input("Greetings! What is your year of origin? "))
2
3 if year <= 1900:
4     print("Woah, that's the past!")
5 elif year > 1900 & year < 2020:
6     print("That's totally the present!")
7 else:
8     print("Far out, that's the future!!")
9
10
```

2. Classes.py

Errores:

Nos encontramos varios errores de sintaxis:

```
1 class Person:
2     def __initalize__(self, first_name, last_name):
3         self.first = first_name
4         self.last = lname
5     def speak(self):
6         print("My name is + " self.fname + " " + self.last)
7
8     me = Person("Brandon", "Walsh")
9     you = Person("Ethan", "Reed")
10
11     me.speak()
12     vou.self.speak
```

- En la línea 2, el método del constructor está mal escrito “__initalize__” debería de ser “__init__”.
- En la línea 4, “lname” no está definido, y por el contexto podríamos esperar que la opción correcta sería “last_name”.
- En la línea 6, en “My name is ” la doble comilla no está cerrando correctamente, así que debería de estar antes del “+”.
- En la línea 6, “self.fname” no existe, así que lo deberíamos sustituir por “self.first”.
- En la línea 12, la función no está escrita correctamente debería ser “you.speak()”.

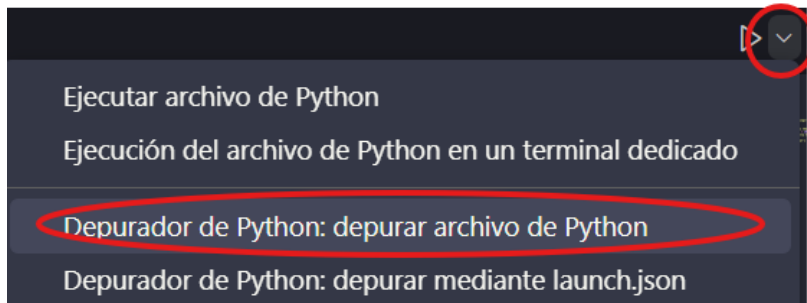
Punto de interrupción:

Ponemos un punto de interrupción

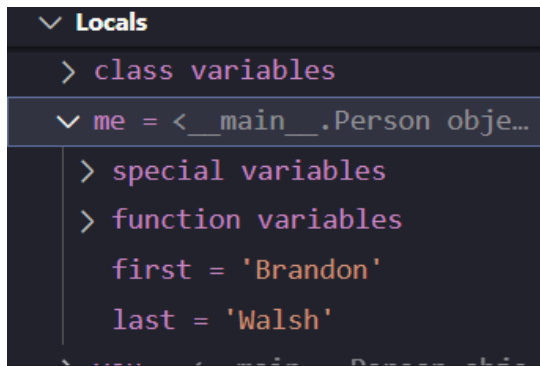
```
8     me = Person("Brandon", "Walsh")
9     you = Person("Ethan", "Reed")
10
11     me.speak()
12     you.speak()
13
```

en la línea 11 para saber su valor
en ese estado de la memoria:

Ejecutamos como depurador:



Podemos observar que están el primer y el segundo nombre de la persona “me”:



Solución:

Así se vería el código con las correcciones pertinentes previamente mencionadas:

```
C:\Users\jaume> OneDrive\Escritorio > Untitled-1.py > ...
1  class Person:
2      def __init__(self, first_name, last_name):
3          self.first = first_name
4          self.last = last_name
5      def speak(self):
6          print("My name is " + self.first + " " + self.last)
7
8  me = Person("Brandon", "Walsh")
9  you = Person("Ethan", "Reed")
10
11 me.speak()
12 you.speak()
```

3. Simple_calculator.py

Errores:

Nos encontramos varios errores que hay que corregir para que funcione correctamente la calculadora:

- En la línea 8, está multiplicando el primer número por el primer número de nuevo y debería de ser por el segundo número.

```
6
7  def multiplication(number1, number2):
8      return number1 * number1
9
```

- En las líneas 23, 24 y 26. hay que pasar los valores a valores numéricos metiendo todo en un “int(...)”

```
23  number1 = input("Enter a number to calculate: ")
24  number2 = input("Enter a number to calculate: ")
25
26  option = ("Enter a option: ")
```

- En la línea 26, además option no recoge ningún valor, así que además del int() hay que meter un “input()”.
- En la línea 6, para hacer el calculador un poco más funcional habría que manejar la división por 0 así que pondremos una condición de que si el primer número es 0 devolverá un error por pantalla.

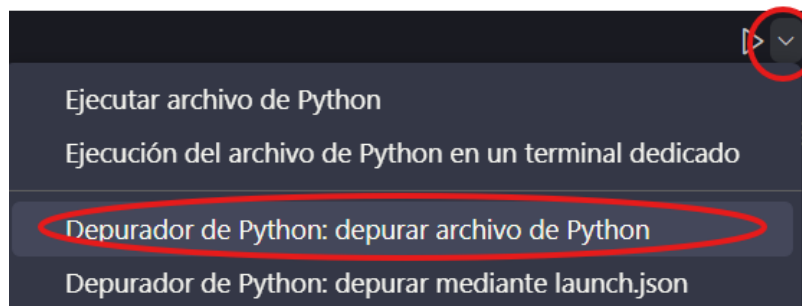
```
9
10  def division(number1, number2):
11
12
13      return number2 / number1
14
```

Punto de interrupción:

Ponemos un punto de interrupción en la línea 31 para saber su valor en ese estado de la memoria:

```
30     if option == 1:
31         print(add(number1, number2))
32     return False
```

Ejecutamos como depurador:



Podemos observar de los valores de cada elemento de la línea según lo que le hemos puesto por terminal:

```
▼ Locals
    number1 = 5
    number2 = 8
    option = 1
```

Solución:

Así se verían las correcciones pertinentes previamente mencionadas:

```
7     def multiplication(number1, number2):
8         return number1 * number2
9
10    def division(number1, number2):
11        if number1 == 0:
12            return "Error: division by 0"
13        return number2 / number1
25    number1 = int(input("Enter a number to calculate: "))
26    number2 = int(input("Enter a number to calculate: "))
27
28    option = int(input(("Enter a opction: ")))
```


4. Factorial.py

Errores:

Nos encontramos varios errores de sintaxis y de lógica:

```
1  def factioral(x):  
2      """  
3      Calculating the factorial of number using recursion  
4  
5      Parameters:  
6      |   x: integer to calculate  
7      Returns:  
8      |   result: Factorial of number  
9  
10     """  
11     if x == 1 or x == 0:  
12         return 1  
13     result = x - factioral(x - 1)  
14     return result  
15  
16     num = input("Enter a number to calculate it factorial: ")  
17  
18     fact = factioral(num)  
19  
20     print(f"The factorial of {num} is {fact}")
```

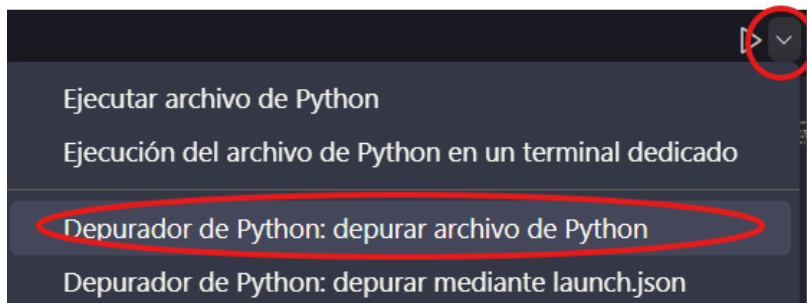
- En la línea 1 , 13 y 18, el nombre del método “factioral” está mal escrito y esto puede causar dolor en los ojos así que lo cambiaremos por “factorial”.
- En la línea 13, la operación está mal escrita, debería de ser “x * factorial(x - 1)”.
- En la línea 16, “num” no se convierte en valor numérico así que tendremos que añadir un “int()” .

Punto de interrupción:

Ponemos un punto de interrupción en la línea 20 para saber su valor en ese estado de la memoria:

```
19
● 20  print(f"The factorial of {num} is {fact}")
21
```

Ejecutamos como depurador:



Podemos observar de los valores de cada elemento de la línea según lo que le hemos puesto por terminal:

```
▼ Locals
> special variables
> function variables
fact = 120
num = 5
```

Solución:

Así se vería el código con las correcciones pertinentes previamente mencionadas:

```
1  def factorial(x):
2
3      if x == 1 or x == 0:
4          return 1
5      result = x * factorial(x - 1)
6      return result
7
8  num = int(input("Enter a number to calculate it factorial: "))
9
10 fact = factorial(num)
11
12 print(f"The factorial of {num} is {fact}")
```