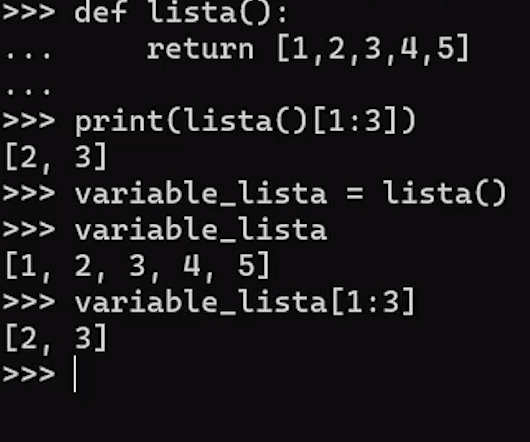
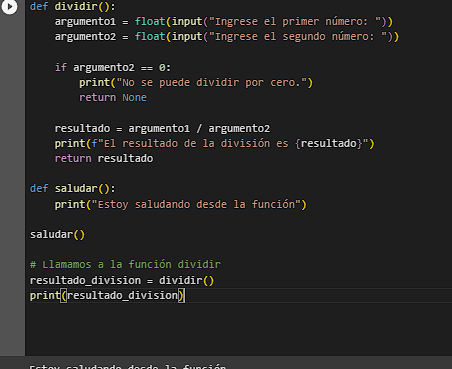


Es conveniente en algunos casos asignar una variable con el valor sea una función para invocarla simplemente con la variable:

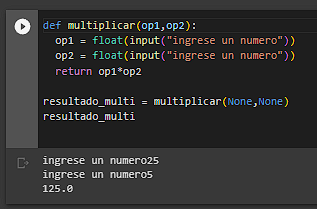


En el ejemplo de arriba de la calculadora que divide, seria algo asi:  
  


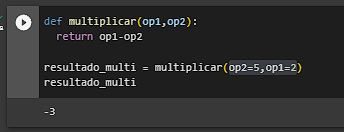
Esto si bien ocupa un lugar mas en memoria hace que el código se ejecuta mas rápido, ya que no se esta ejecutando de corrido la función, sino solo una vez cuando la variable es invocada. Se vuelve mas per formable

Otro ejemplo para multiplicar, importante poner none al invocar para indicar que será puesto por usuario

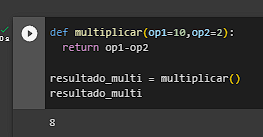
Hago la multiplicación mismo en el return para optimizar código



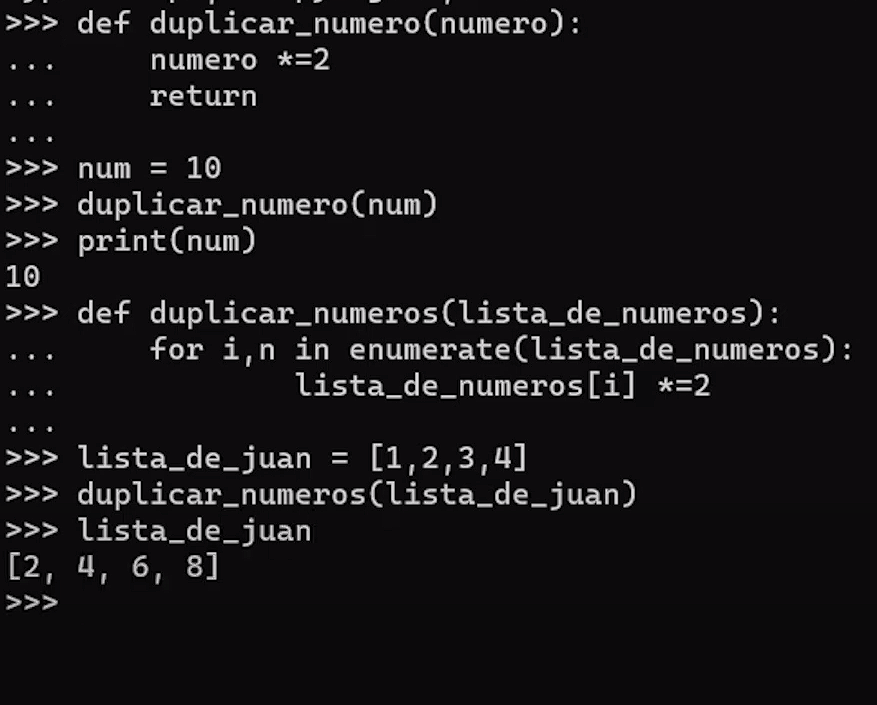
Puedo en los argumentos modificar el orden de parámetros sin alterar resultado, asignado la variable por ejemplo op2 y asignando valor nuevo



Para evitar el error de invocar una función y al no dar argumento me tire error, lo que puedo hacer es dejar unos parámetros por defecto, cosa que si no los cambio agarra de ahí el valor por ejemplo:



Al llamar una lista por valor podemos modificar los valores a partir de ese punto de la lista original por ejemplo:

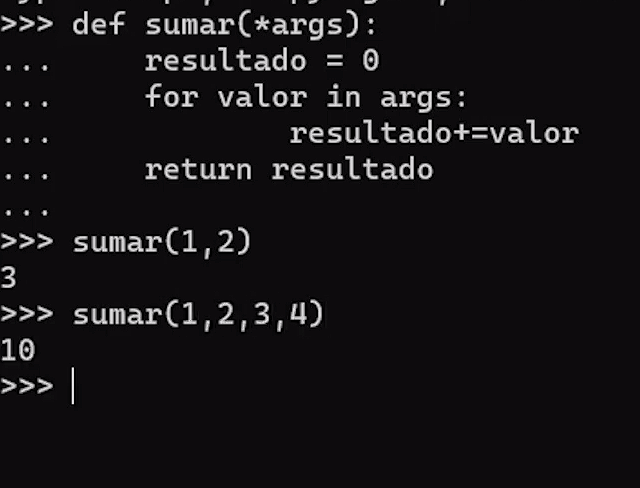


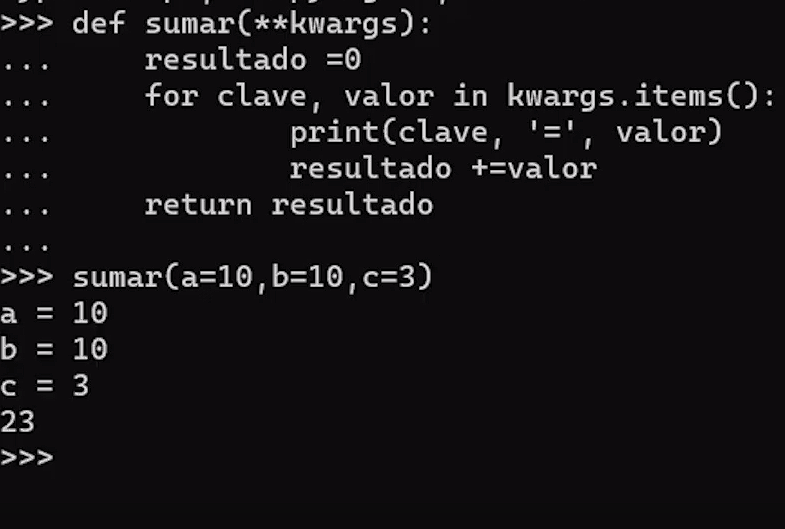
Hay formas de subsanar para que quede como paso por referencia pero Python no tiene forma de que le indiquemos cuando queremos pasar por valor o cuando por referencia

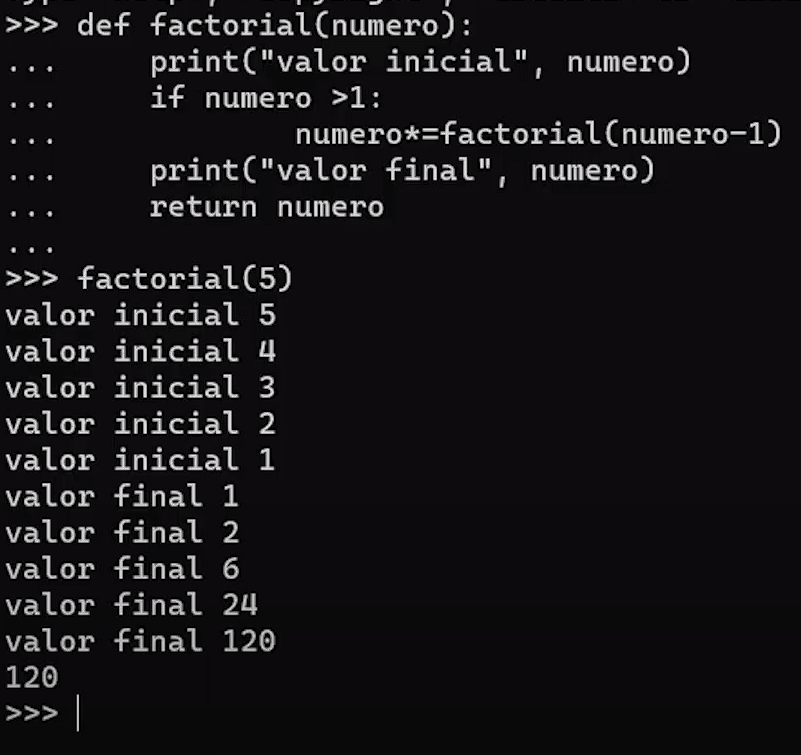
Otros lenguajes si

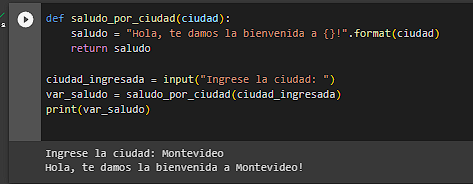
Uso de args \* y kwargs \*\* - los kwargs son un diccionario podemos agregarle un nombre a cada valor

* Cabe aclarar que args y kwargs son por convención podrían llamarse cualquier cosa, como pepinillos, o valores, o lo que sea

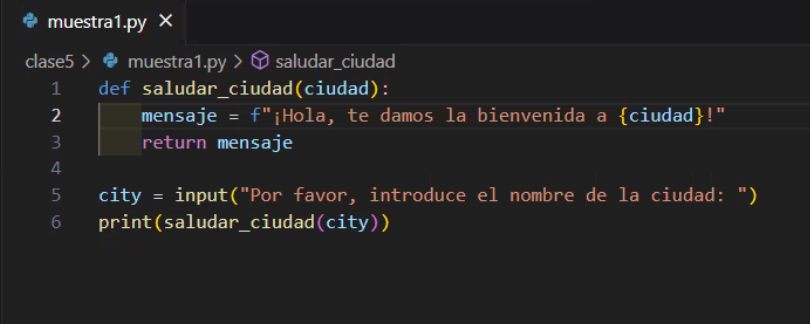




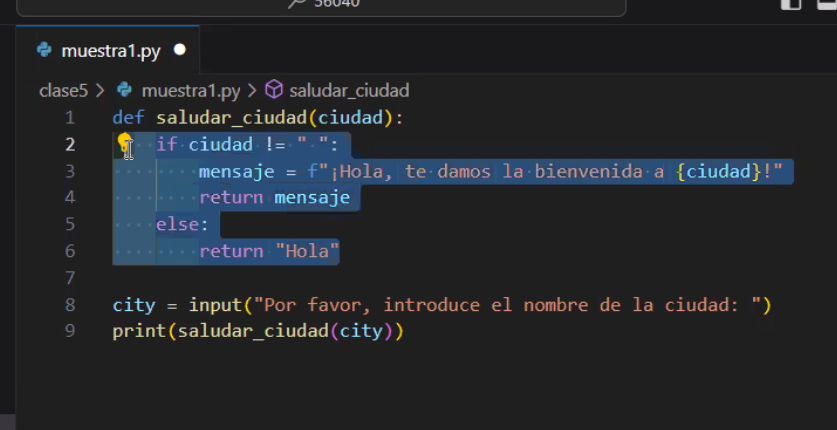




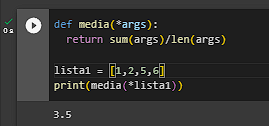
Solucion profe:

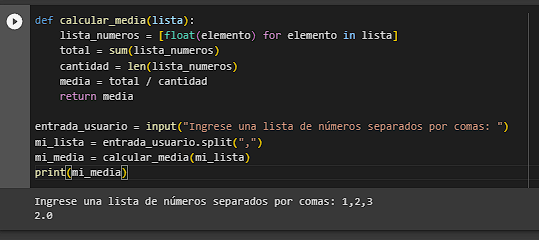


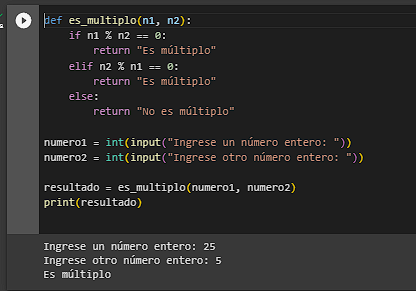
Puedo usar varios return si no es en el mismo bloque de código, aca como el if no se cumple, pasa al else al ser un “ ”, en ese caso retorna “hola”



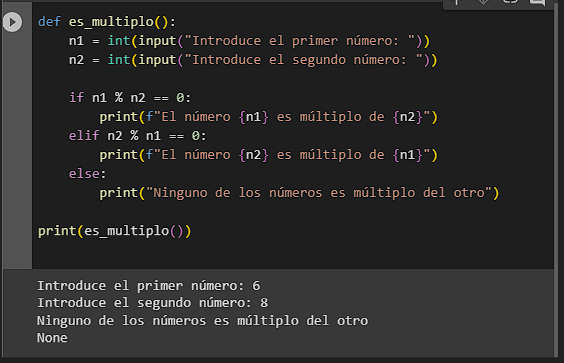


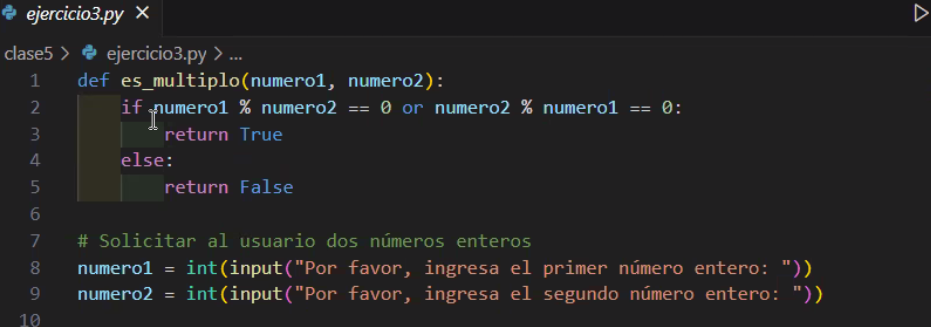




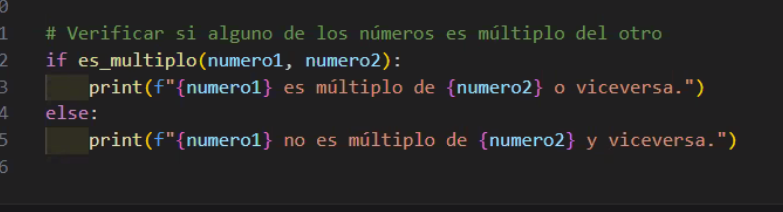


Solución de un compa algo mas prolija a nivel lo que muestra

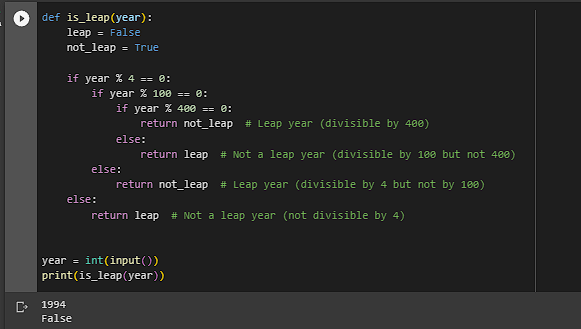


Alternativa profe con menos código   


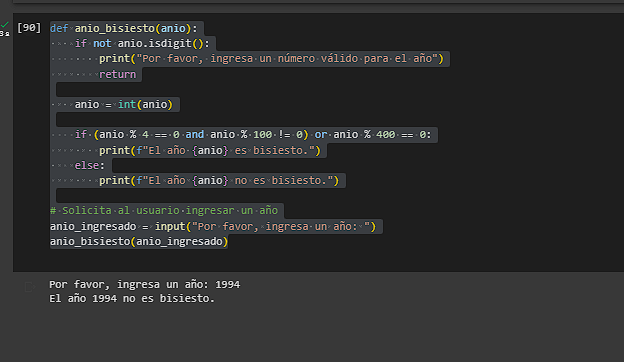
Aunque luego lo alarga con otro if para retornar la cadena de texto de respuesta



Mi solución



La del profe:



Ejercicio listas pares e impares:

