**Codificación y Programación.**

**Quiz Capítulos #3 Y #4.**

**Q. 03-01.** Diseñe un programa que reciba las coordenadas (x1, y1), (x2, y2) de dos puntos del usuario e imprima la distancia entre los dos puntos. Para hacer esto, implemente la función distancia (x1, y1, x2, y2). Pauta de código: consulte la ecuación para encontrar la distancia entre dos puntos.



Codigo:

import math

def distancia(*x1*, *y1*, *x2*, *y2*):

return math.sqrt((*x2*-*x1*)\*\*2 + (*y2*-*y1*)\*\*2)

x1 = float(input("Ingrese la coordenada x1: "))

y1 = float(input("Ingrese la coordenada y1: "))

x2 = float(input("Ingrese la coordenada x2: "))

y2 = float(input("Ingrese la coordenada y2: "))

print("La distancia entre los puntos es: ", distancia(x1, y1, x2, y2))

print(" ")

print("Los puntos escogidos son: ")

def recorre(*lista*):

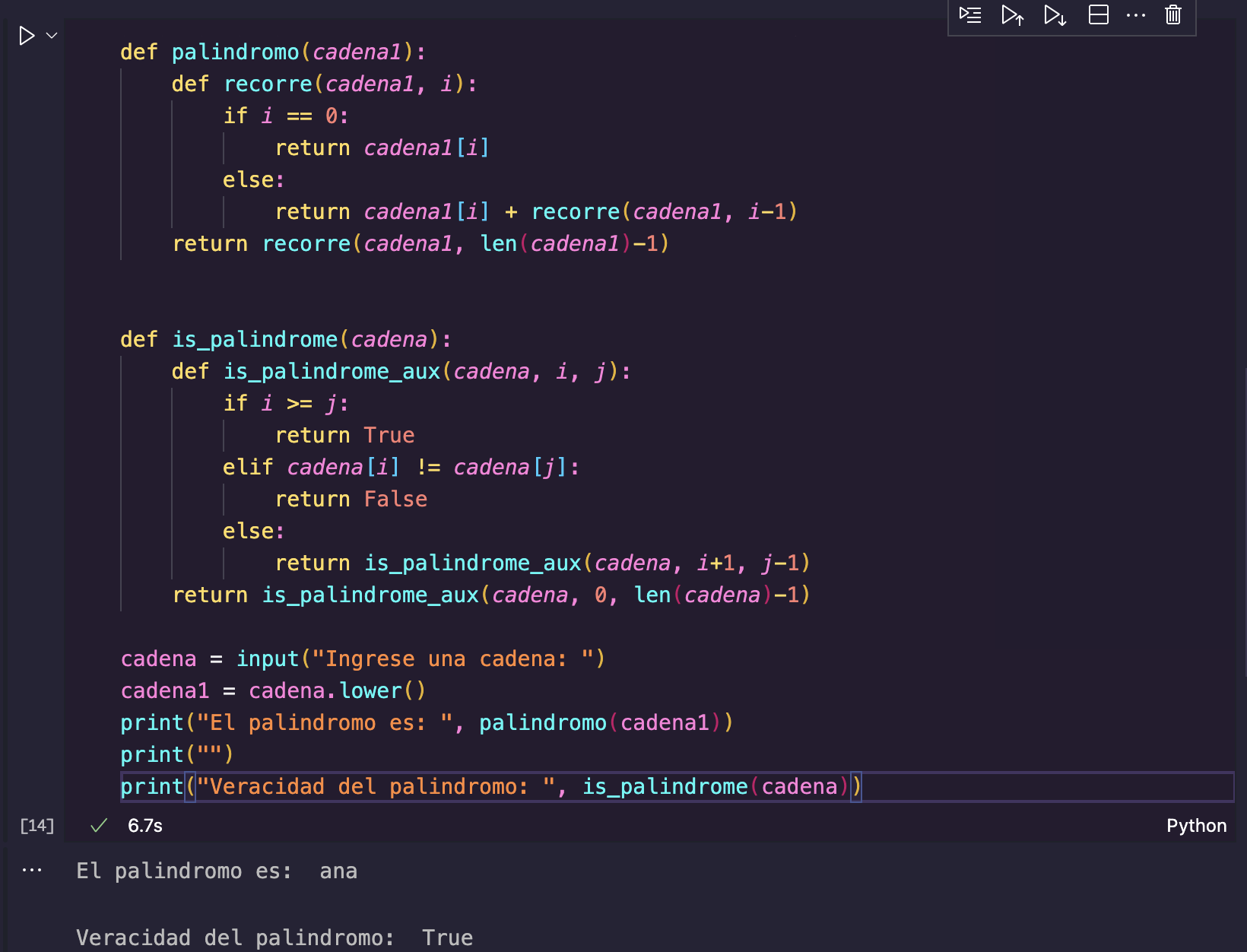
for i in range(len(*lista*)):

print(*lista*[i])

lista = [x1, y1, x2, y2]

recorre(lista)

**Q. 03-02.** Un palíndromo es una oración, palabra o cadena que se lee igual al derecho o al revés. Por ejemplo, reconocer, radar o la frase “anita lava la tina”. Usemos una llamada recursiva para determinar el palíndromo. Defina una función llamada is\_palindrome y escriba un programa que reciba una cadena del usuario e imprima si el palíndromo es correcto o no. Pauta de código: Llame a la función is\_palindrome dentro de la función is\_palindrome(función recursiva).



Codigo:

def palindromo(*cadena1*):

def recorre(*cadena1*, *i*):

if *i* == 0:

return *cadena1*[*i*]

else:

return *cadena1*[*i*] + recorre(*cadena1*, *i*-1)

return recorre(*cadena1*, len(*cadena1*)-1)

def is\_palindrome(*cadena*):

def is\_palindrome\_aux(*cadena*, *i*, *j*):

if *i* >= *j*:

return True

elif *cadena*[*i*] != *cadena*[*j*]:

return False

else:

return is\_palindrome\_aux(*cadena*, *i*+1, *j*-1)

return is\_palindrome\_aux(*cadena*, 0, len(*cadena*)-1)

cadena = input("Ingrese una cadena: ")

cadena1 = cadena.lower()

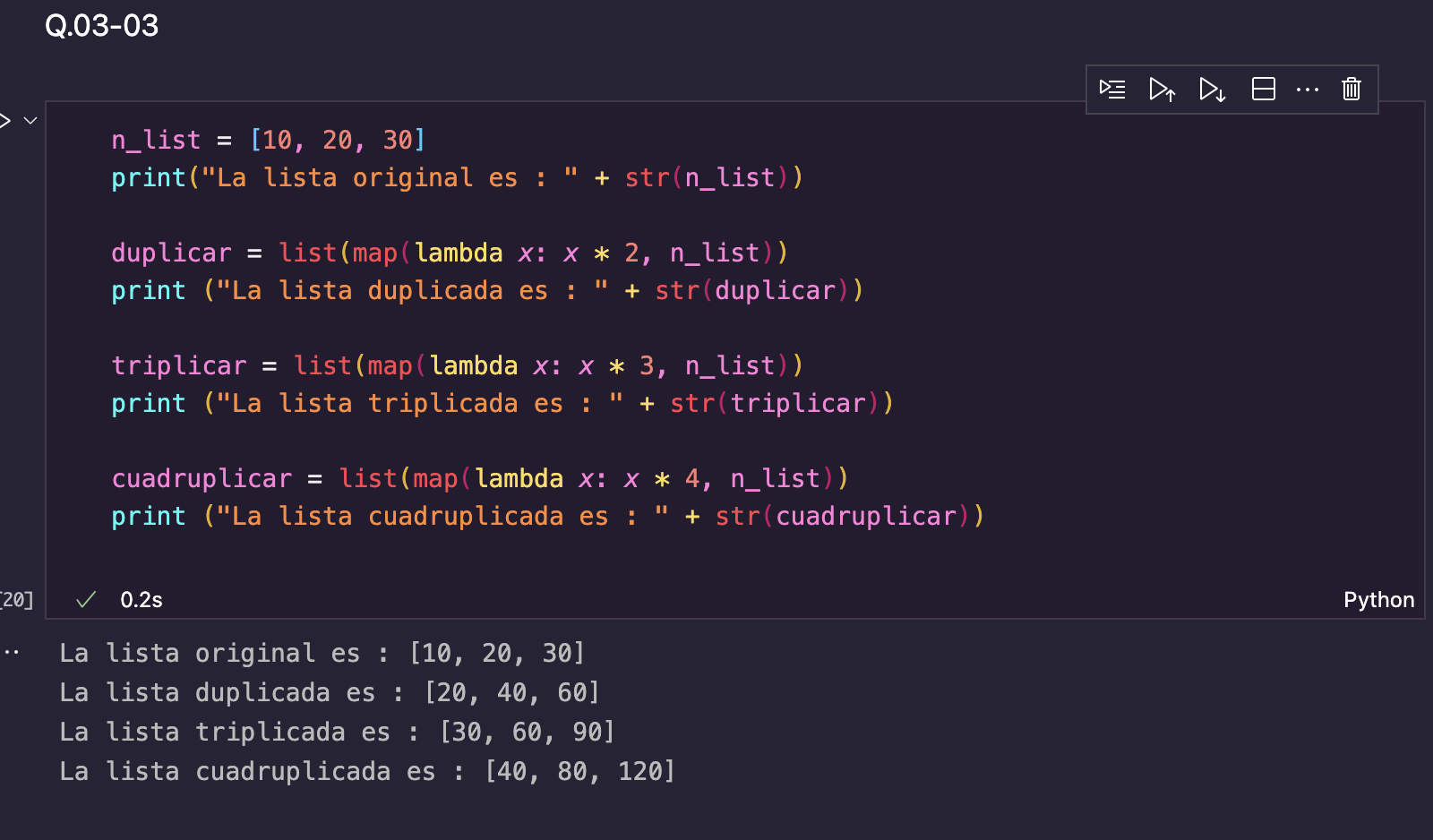
print("El palindromo es: ", palindromo(cadena1))

print("")

print("Veracidad del palindromo: ", is\_palindrome(cadena))

**Q. 03-03.** Defina una lista llamada n\_list con valores de [10, 20, 30]. Utilice la función map y lambda para imprimir los resultados de duplicar, triplicar y cuadruplicar los objetos de la lista

ejemplo:mapped\_numbers = list(map(lambda x: x \* 2 + 3, numbers))



Codigo:

n\_list = [10, 20, 30]

print("La lista original es : " + str(n\_list))

duplicar = list(map(lambda *x*: *x* \* 2, n\_list))

print ("La lista duplicada es : " + str(duplicar))

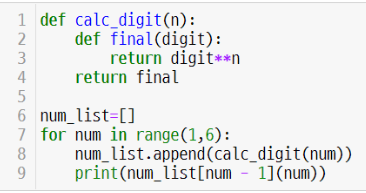
triplicar = list(map(lambda *x*: *x* \* 3, n\_list))

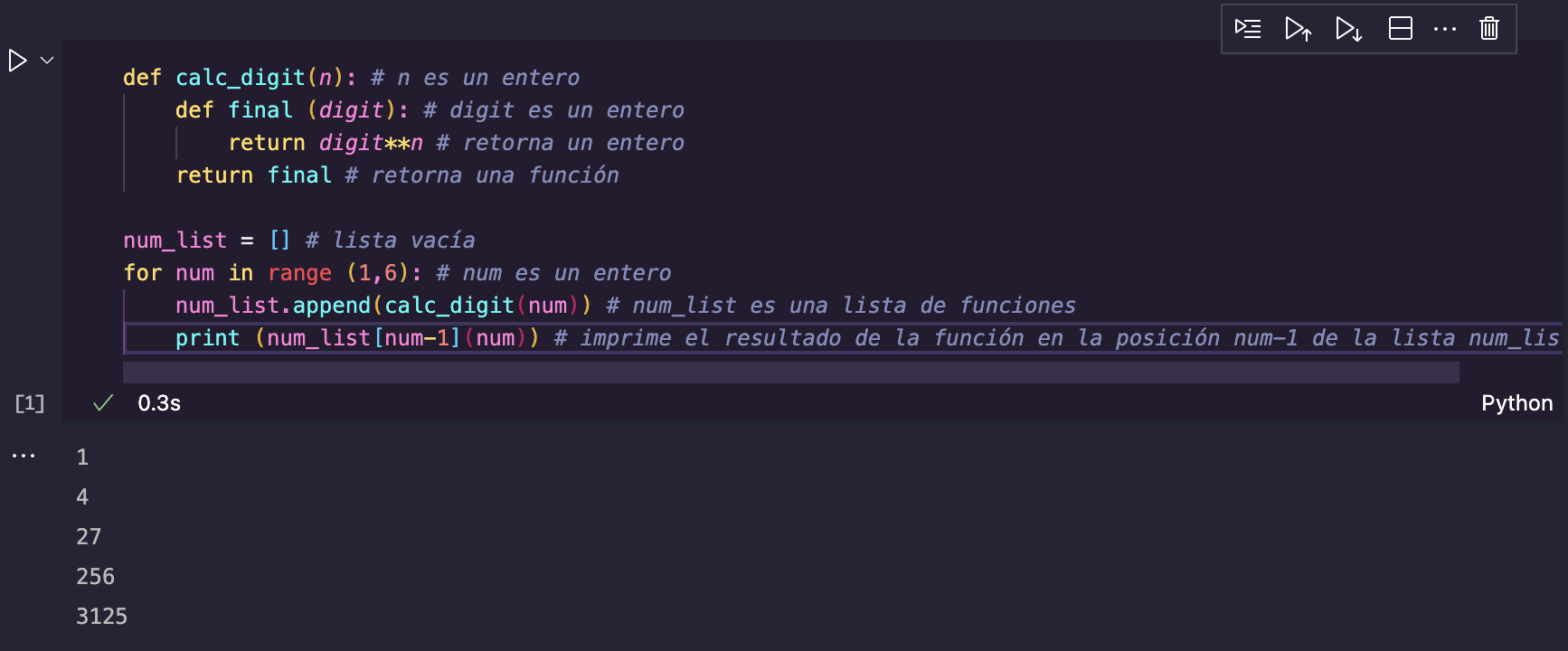
print ("La lista triplicada es : " + str(triplicar))

cuadruplicar = list(map(lambda *x*: *x* \* 4, n\_list))

print ("La lista cuadruplicada es : " + str(cuadruplicar))

**Q. 03-04.** Explique el funcionamiento del siguiente código, para ello aplique una prueba de escritorio hecha “a mano”





Codigo (prueba de escritorio):

def calc\_digit(*n*): *# n es un entero*

def final (*digit*): *# digit es un entero*

return *digit*\*\**n* *# retorna un entero*

return final *# retorna una función*

num\_list = [] *# lista vacía*

for num in range (1,6): *# num es un entero*

num\_list.append(calc\_digit(num)) *# num\_list es una lista de funciones*

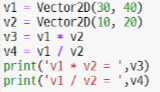
print (num\_list[num-1](num)) *# imprime el resultado de la función en la posición num-1 de la lista num\_list*

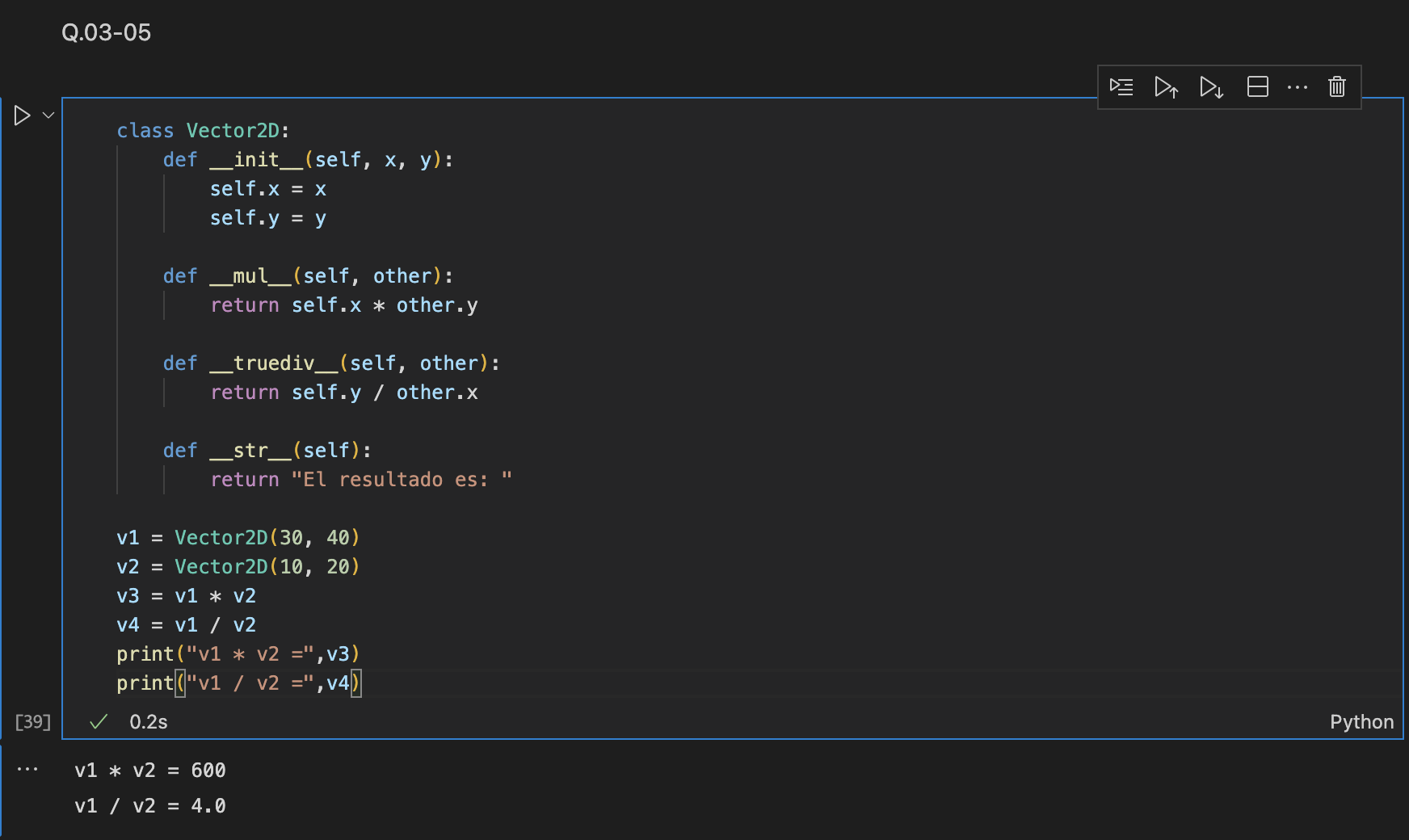
**Q. 03-05.** Implemente las funciones de multiplicación (\*) y división (/) de dos vectores usando los métodos especiales \_\_mul\_\_ y \_\_truediv\_\_. Suponiendo que v1 es (30, 40) y v2 es (10, 20), codifique para devolver el siguiente resultado como resultado de la multiplicación y división de dos vectores.



Pauta de codificación: escriba un código implementando clases que reciba 2 objetos de la clase vector2D y los opere como se ha solicitado

Hint:





Codigo:

class Vector2D:

def \_\_init\_\_(self, x, y):

self.x = x

self.y = y

def \_\_mul\_\_(self, other):

return self.x \* other.y

def \_\_truediv\_\_(self, other):

return self.y / other.x

def \_\_str\_\_(self):

return "El resultado es: "

v1 = Vector2D(30, 40)

v2 = Vector2D(10, 20)

v3 = v1 \* v2

v4 = v1 / v2

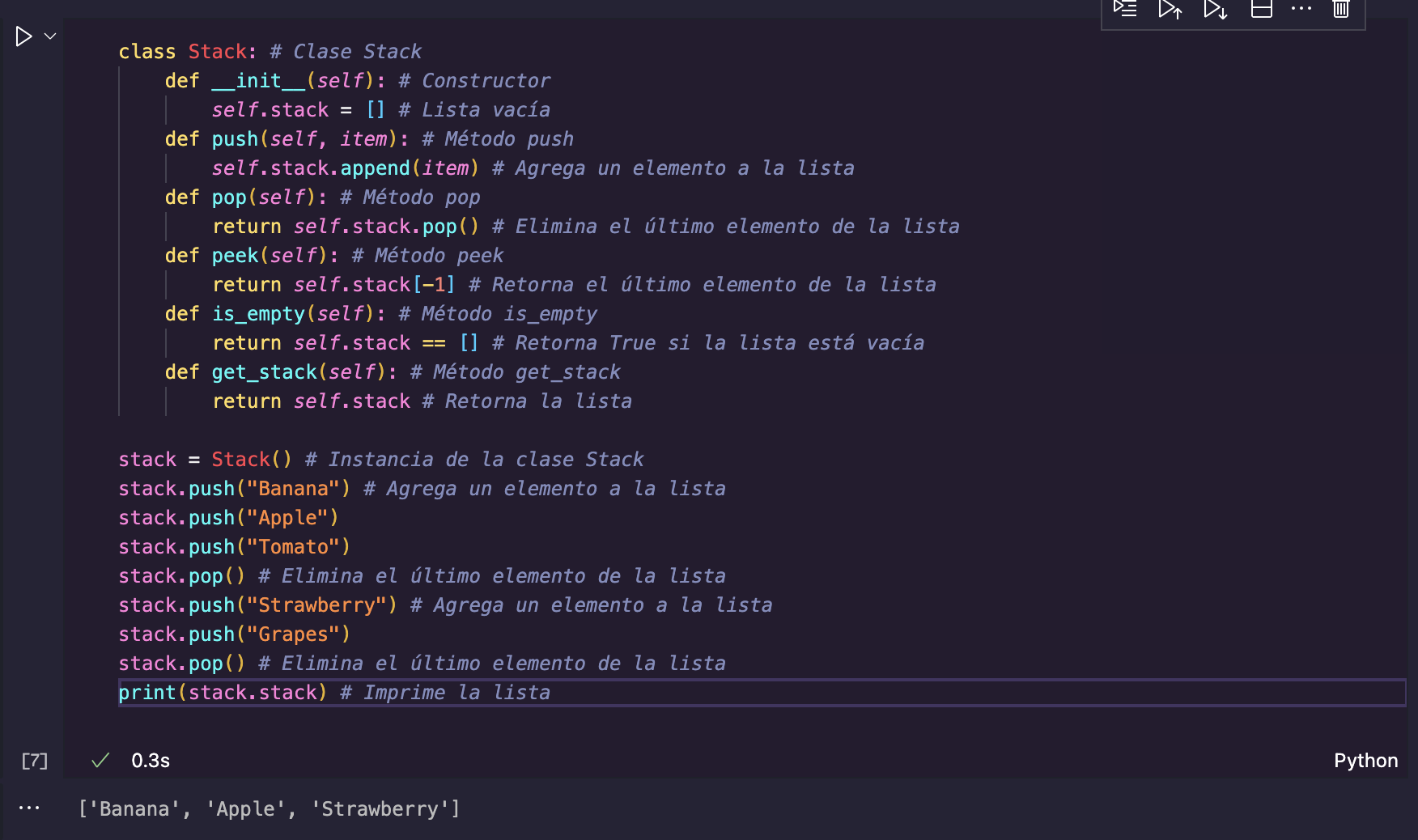
print("v1 \* v2 =",v3)

print("v1 / v2 =",v4)

**Q. 04-01.** La siguiente es la implementación de una pila en python. ¿Cuál será el resultado del siguiente código?



Pauta de codificación: escriba cual es el resultado esperado de ejecutar cada línea (prueba de escritorio)



Codigo:

class Stack: *# Clase Stack*

def \_\_init\_\_(*self*): *# Constructor*

*self*.stack = [] *# Lista vacía*

def push(*self*, *item*): *# Método push*

*self*.stack.append(*item*) *# Agrega un elemento a la lista*

def pop(*self*): *# Método pop*

return *self*.stack.pop() *# Elimina el último elemento de la lista*

def peek(*self*): *# Método peek*

return *self*.stack[-1] *# Retorna el último elemento de la lista*

def is\_empty(*self*): *# Método is\_empty*

return *self*.stack == [] *# Retorna True si la lista está vacía*

def get\_stack(*self*): *# Método get\_stack*

return *self*.stack *# Retorna la lista*

stack = Stack() *# Instancia de la clase Stack*

stack.push("Banana") *# Agrega un elemento a la lista*

stack.push("Apple")

stack.push("Tomato")

stack.pop() *# Elimina el último elemento de la lista*

stack.push("Strawberry") *# Agrega un elemento a la lista*

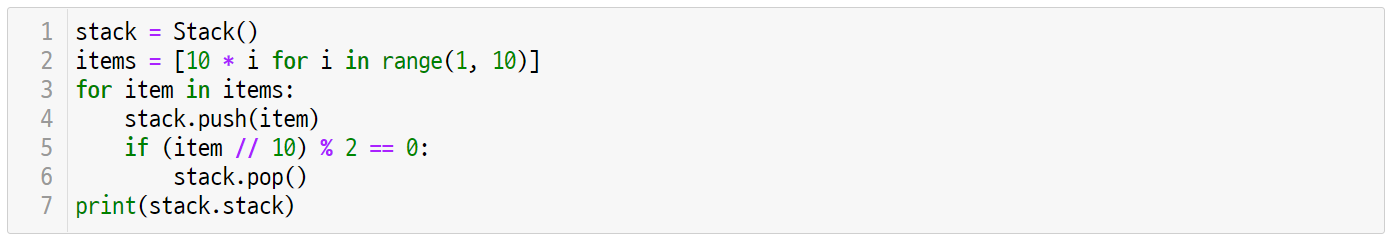
stack.push("Grapes")

stack.pop() *# Elimina el último elemento de la lista*

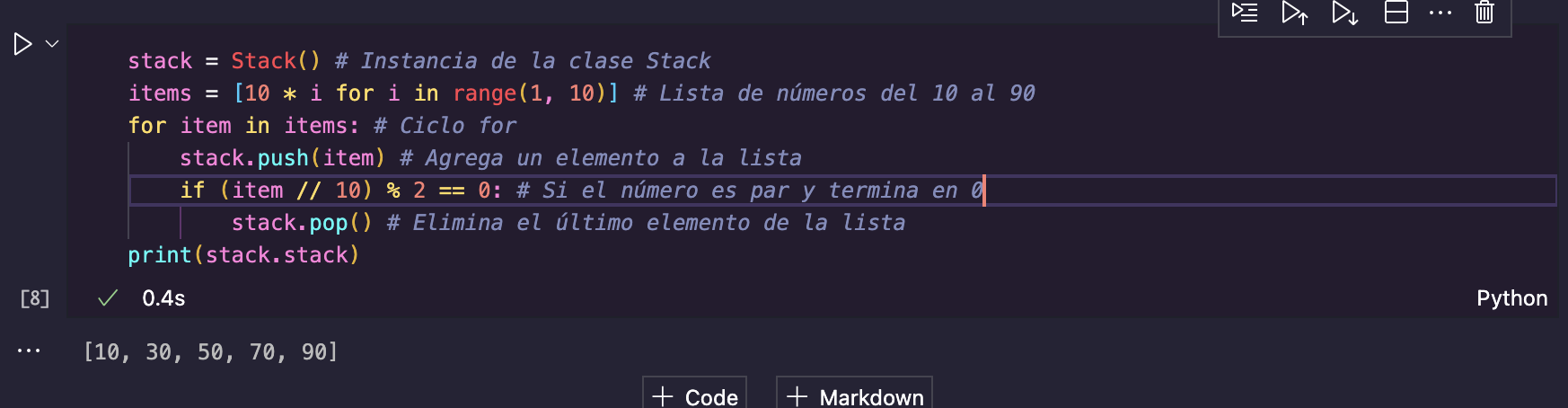
print(stack.stack) *# Imprime la lista*

El resultado esperado era que imprima la lista pero que el “pop” elimine el ultimo elemento de cada lista.

**Q. 04-02.** La siguiente es la implementación de una pila en python. ¿Cuál será el resultado del siguiente código?



Pauta de codificación: escriba cual es el resultado esperado de ejecutar cada línea (prueba de escritorio)



Codigo:

stack = Stack() *# Instancia de la clase Stack*

items = [10 \* i for i in range(1, 10)] *# Lista de números del 10 al 90*

for item in items: *# Ciclo for*

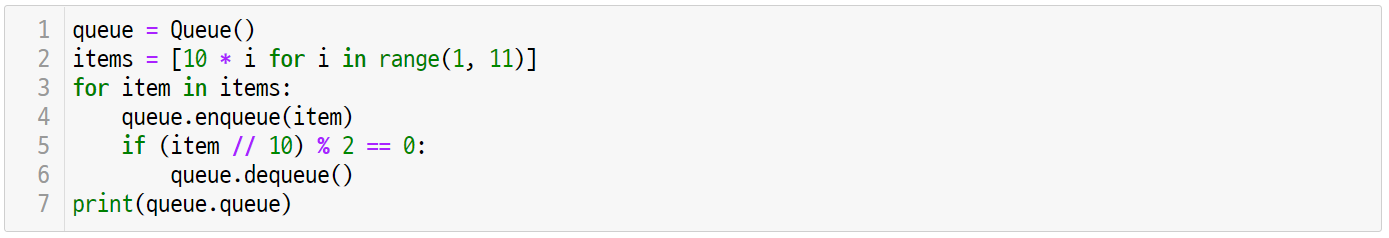
stack.push(item) *# Agrega un elemento a la lista*

if (item // 10) % 2 == 0: *# Si el número es par y termina en 0*

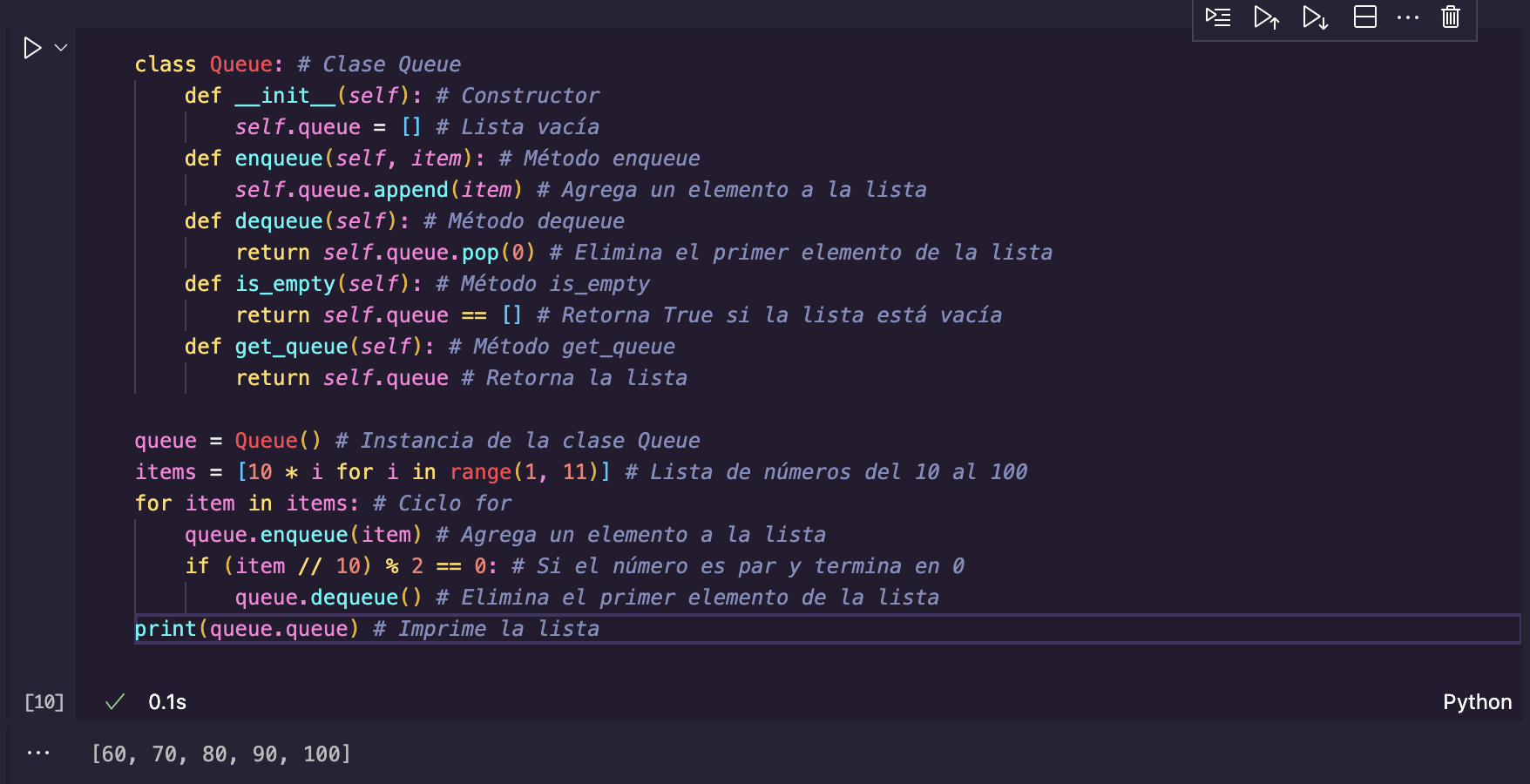
stack.pop() *# Elimina el último elemento de la lista*

print(stack.stack) *# Imprime la lista*

**Q. 04-03.** A continuación se muestra la implementación de una cola en python(queue). ¿Cuál será el resultado del siguiente código?



Pauta de codificación: escriba cual es el resultado esperado de ejecutar cada línea (prueba de escritorio)

****

Codigo:

class Queue: *# Clase Queue*

def \_\_init\_\_(*self*): *# Constructor*

*self*.queue = [] *# Lista vacía*

def enqueue(*self*, *item*): *# Método enqueue*

*self*.queue.append(*item*) *# Agrega un elemento a la lista*

def dequeue(*self*): *# Método dequeue*

return *self*.queue.pop(0) *# Elimina el primer elemento de la lista*

def is\_empty(*self*): *# Método is\_empty*

return *self*.queue == [] *# Retorna True si la lista está vacía*

def get\_queue(*self*): *# Método get\_queue*

return *self*.queue *# Retorna la lista*

queue = Queue() *# Instancia de la clase Queue*

items = [10 \* i for i in range(1, 11)] *# Lista de números del 10 al 100*

for item in items: *# Ciclo for*

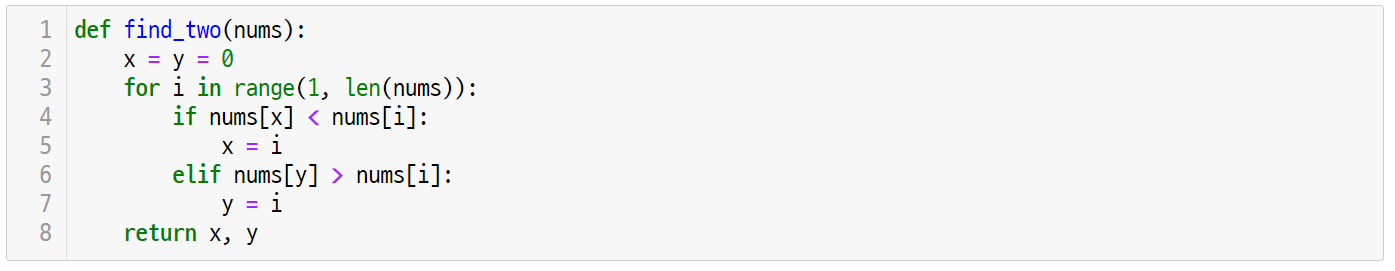
queue.enqueue(item) *# Agrega un elemento a la lista*

if (item // 10) % 2 == 0: *# Si el número es par y termina en 0*

queue.dequeue() *# Elimina el primer elemento de la lista*

print(queue.queue) *# Imprime la lista*

**Q. 04-04.** ¿Cuál es el algoritmo de la siguiente función find\_two()? Analice el código y escriba el resultado de la ejecución.



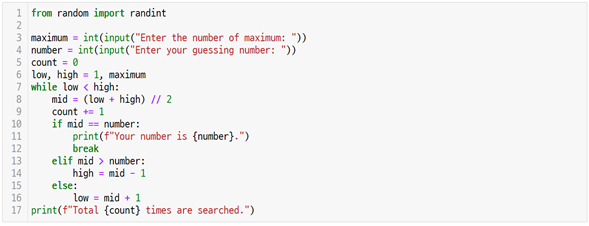
R: El resultado de la ejecucion nos arroja los 2 numeros mas pequeños (11, 11)



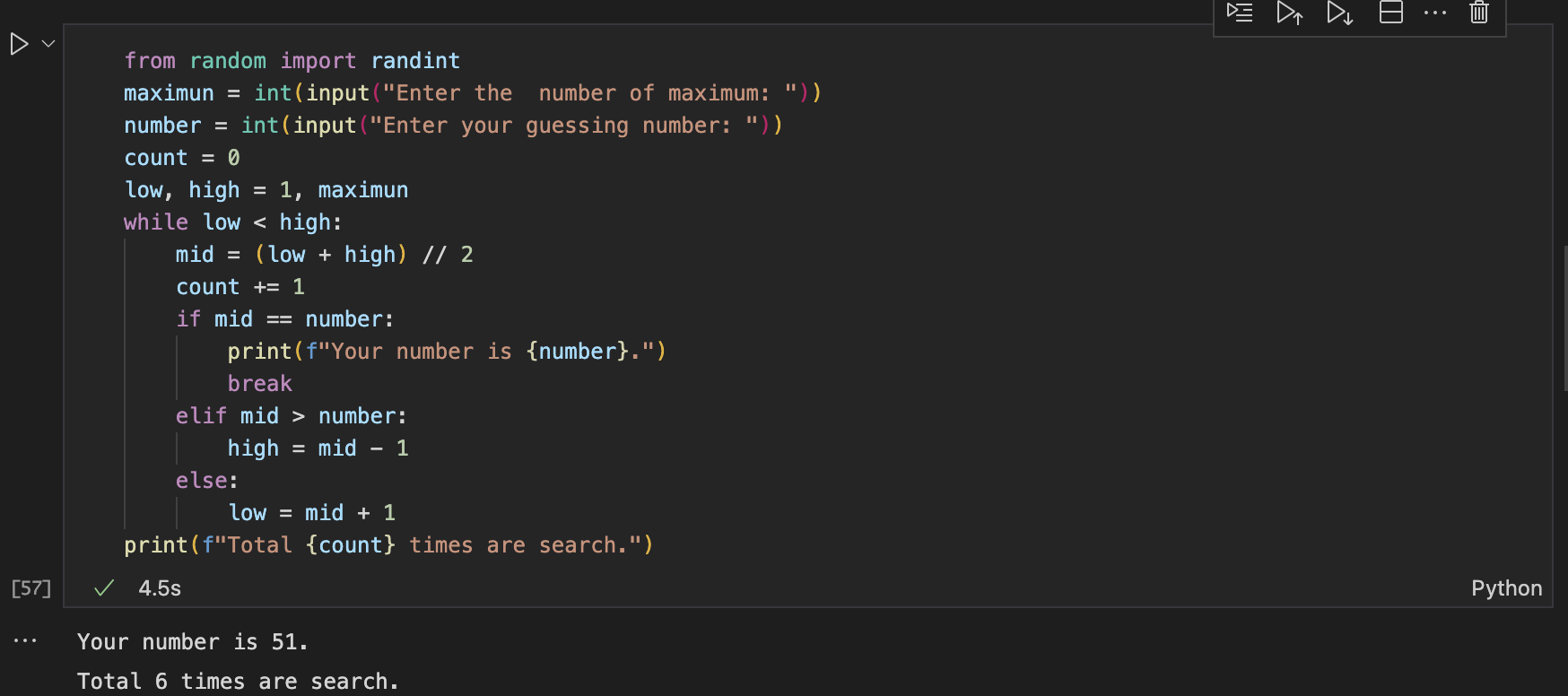
**Q. 04-05.** ¿Cuántas comparaciones debe realizar la función find\_two() implementada en la pregunta anterior (Q.04-04)?

R: Debe realizar 4 comparaciones.

**Q. 04-06.** El siguiente es el código para el juego de combinación de números. Si el máximo es 100 y el número es 51, ¿cuál es la salida de count?

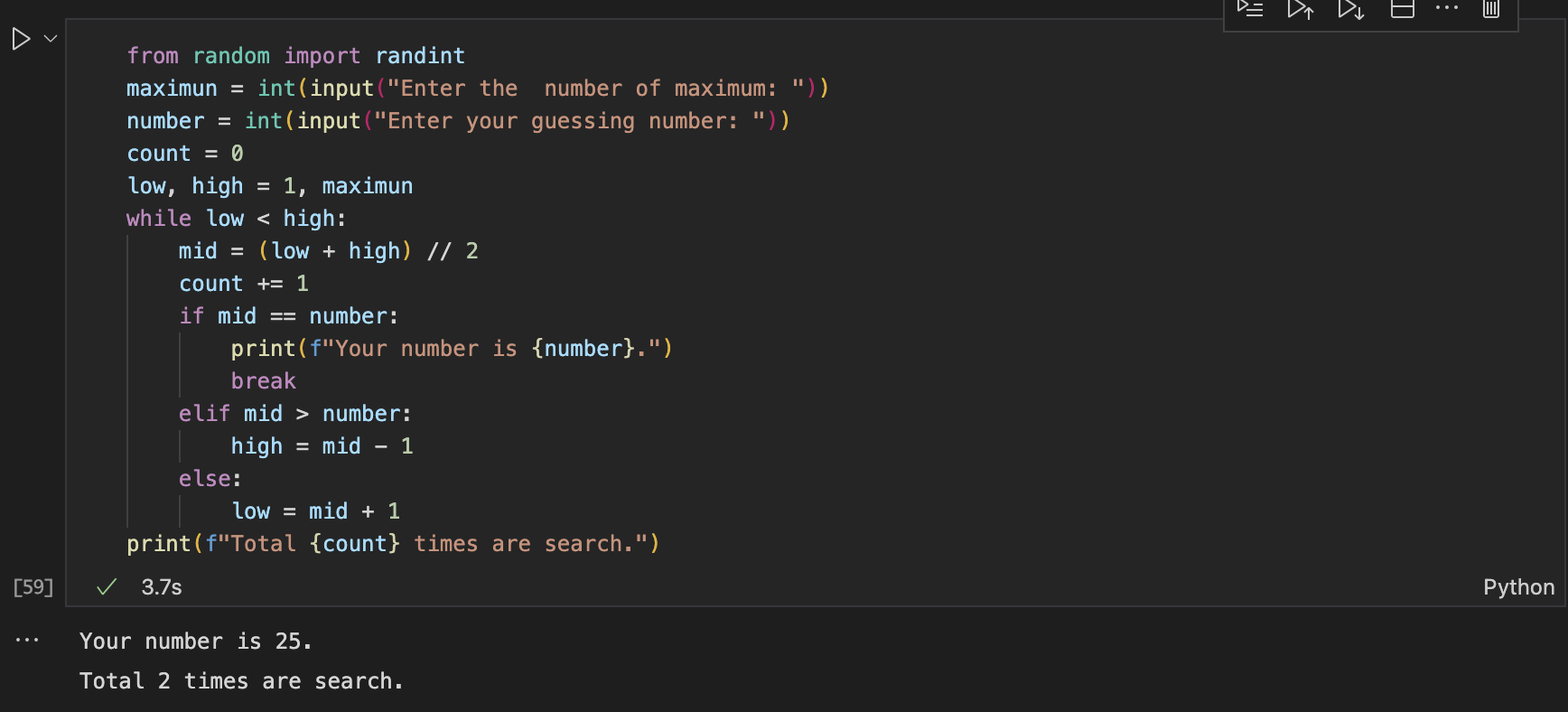


R: La salida de “count” es el total de veces que se buscó, en este caso 6 veces.



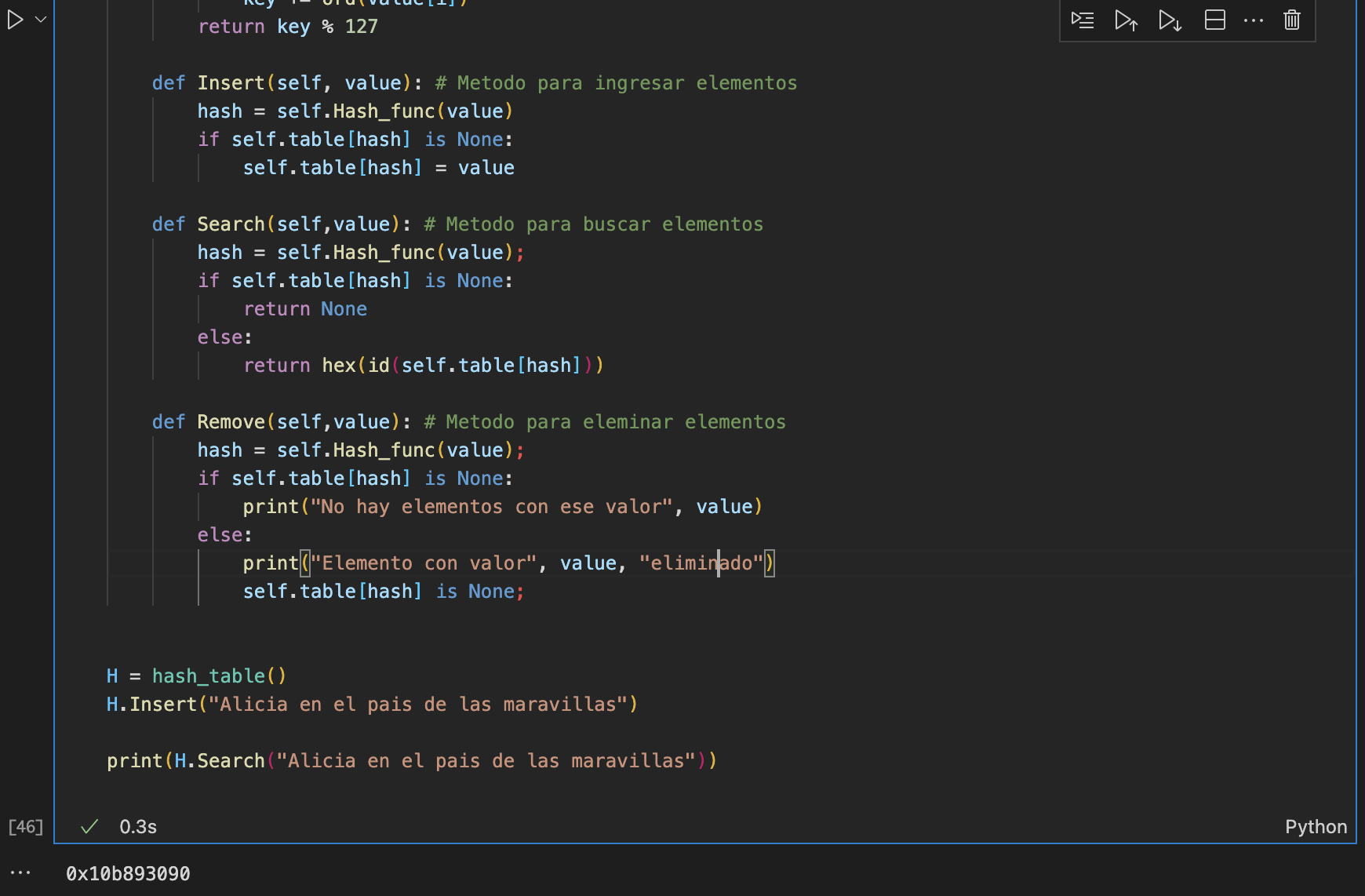
**Q. 04-07.** En el código del juego de combianación de números, si el máximo es 100 y el número es 25, ¿cuál es el resultado del conteo?

R: La busqueda arroja un total de 2.



**Q. 04-08.** Usando la función insert de la clase hash\_table, ingrese la clave "Alicia en el país de las maravillas", a continuación obtenga la clave hash de buscar dicha clave dentro de la hash\_table

Pauta de codificación: Ayúdese de la definición de la clase hash\_table del siguiente ejemplo: <https://pythondiario.com/2018/06/tabla-hash-en-python.html>

****

**Codigo:**

class hash\_table:

def \_\_init\_\_(self):

self.table = [None] \* 127

# Función hash

def Hash\_func(self, value):

key = 0

for i in range(0,len(value)):

key += ord(value[i])

return key % 127

def Insert(self, value): # Metodo para ingresar elementos

hash = self.Hash\_func(value)

if self.table[hash] is None:

self.table[hash] = value

def Search(self,value): # Metodo para buscar elementos

hash = self.Hash\_func(value);

if self.table[hash] is None:

return None

else:

return hex(id(self.table[hash]))

def Remove(self,value): # Metodo para eleminar elementos

hash = self.Hash\_func(value);

if self.table[hash] is None:

print("No hay elementos con ese valor", value)

else:

print("Elemento con valor", value, "eliminado")

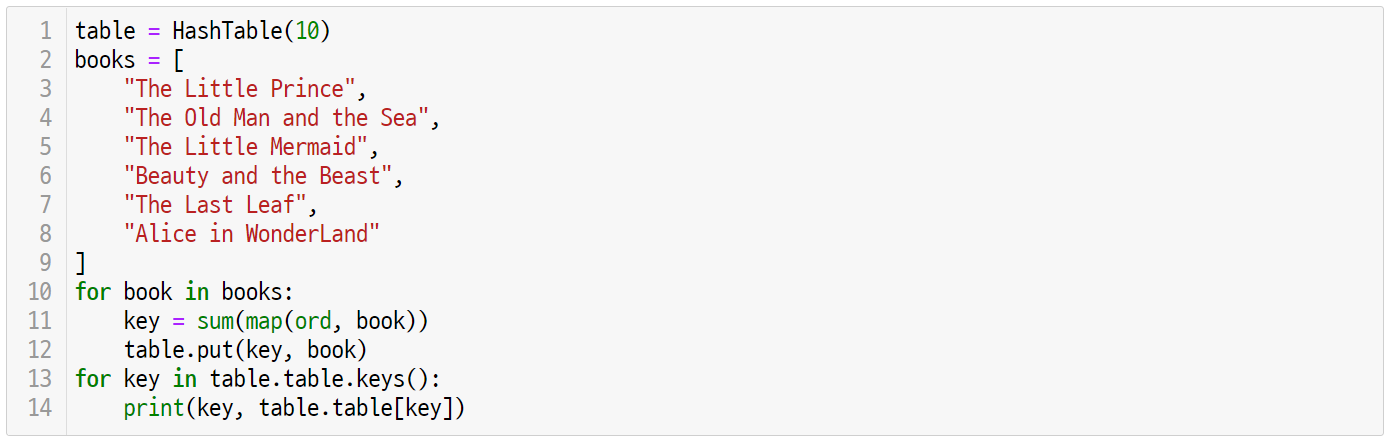
self.table[hash] is None;

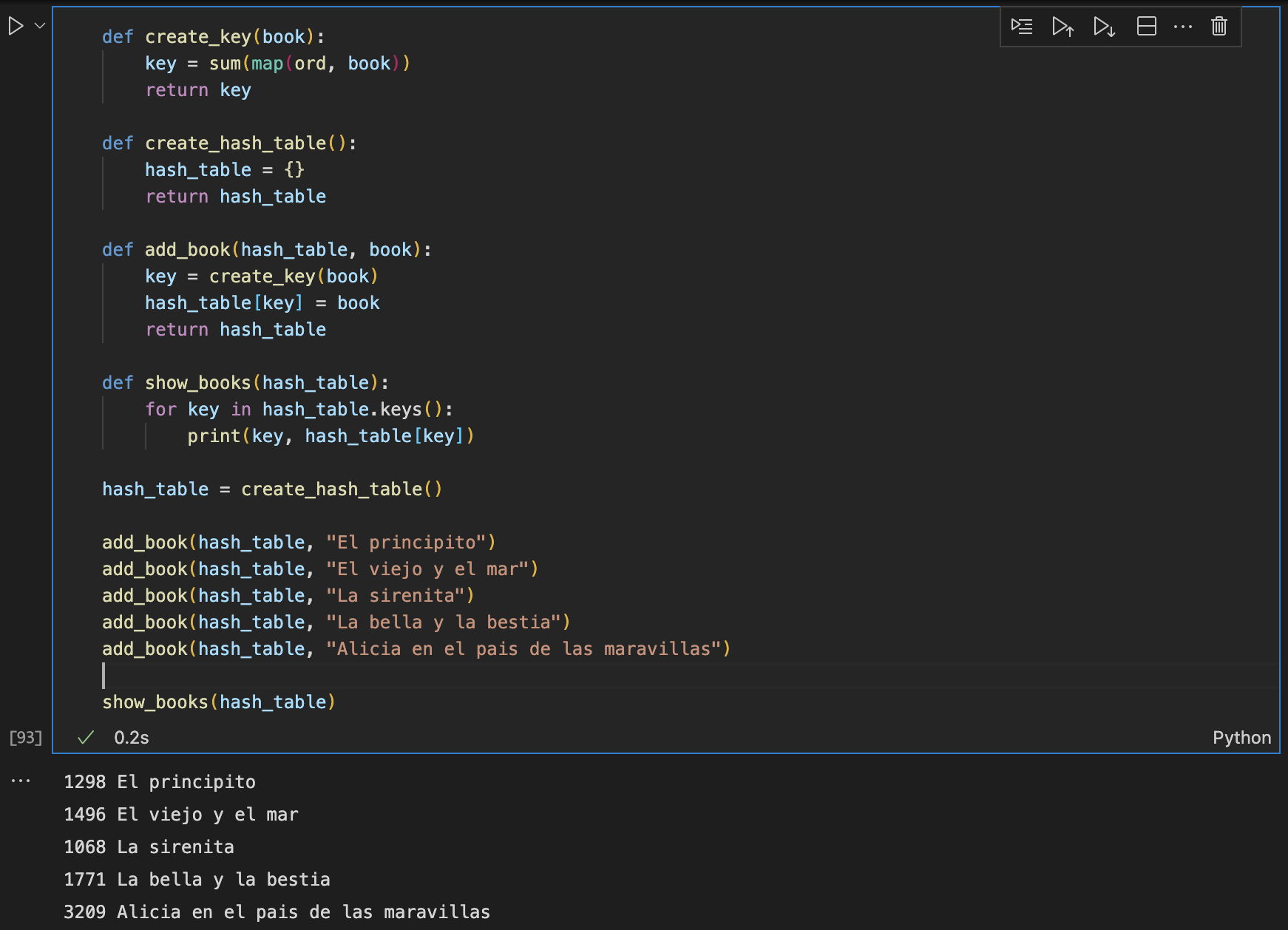
H = hash\_table()

H.Insert("Alicia en el pais de las maravillas")

print(H.Search("Alicia en el pais de las maravillas"))

**Q. 04-9.** Si la nueva estantería tiene 10 compartimentos, usa el siguiente código para averiguar qué libro hay en cada compartimento.





Codigo:

def create\_key(book):

key = sum(map(ord, book))

return key

def create\_hash\_table():

hash\_table = {}

return hash\_table

def add\_book(hash\_table, book):

key = create\_key(book)

hash\_table[key] = book

return hash\_table

def show\_books(hash\_table):

for key in hash\_table.keys():

print(key, hash\_table[key])

hash\_table = create\_hash\_table()

add\_book(hash\_table, "El principito")

add\_book(hash\_table, "El viejo y el mar")

add\_book(hash\_table, "La sirenita")

add\_book(hash\_table, "La bella y la bestia")

add\_book(hash\_table, "Alicia en el pais de las maravillas")

show\_books(hash\_table)