1.1 ¿Qué es una Tupla? ¿Cómo funciona? Dar ejemplos.

Una tupla en Python es una estructura de datos inmutable que puede contener múltiples elementos. Se define utilizando paréntesis () y los elementos se separan por comas. A diferencia de las listas, las tuplas no se pueden modificar después de su creación. Pueden contener elementos de diferentes tipos de datos. Aquí tienes un ejemplo:

tupla = (1, 2, "tres", 4.5)

En este ejemplo, hemos creado una tupla llamada tupla que contiene cuatro elementos: el número entero 1, el número entero 2, la cadena de texto "tres" y el número decimal 4.5.

Las tuplas son útiles cuando necesitas almacenar un conjunto de valores que no deben cambiar, como las coordenadas de un punto en un plano.

1.2 ¿Qué es una Lista? ¿Cómo funciona? Dar ejemplos.

Una lista en Python es una estructura de datos mutable que permite almacenar múltiples elementos en un orden específico. Se define utilizando corchetes [] y los elementos se separan por comas. A diferencia de las tuplas, las listas se pueden modificar después de su creación. Pueden contener elementos de diferentes tipos de datos. Aquí tienes un ejemplo:

lista = [1, 2, "tres", 4.5]

En este ejemplo, hemos creado una lista llamada lista que contiene cuatro elementos: el número entero 1, el número entero 2, la cadena de texto "tres" y el número decimal 4.5.

Las listas son muy versátiles y se utilizan ampliamente en Python. Puedes agregar, eliminar o modificar elementos de una lista según sea necesario.

1.3 ¿Qué es un Diccionario? ¿Cómo funciona? Dar ejemplos.

Un diccionario en Python es una estructura de datos que permite almacenar pares clave-valor. Se define utilizando llaves {} y los pares clave-valor se separan por comas. Cada clave debe ser única dentro de un diccionario, y se utiliza para acceder a su valor correspondiente. Aquí tienes un ejemplo:

diccionario = {"nombre": "Juan", "edad": 25, "ciudad": "Madrid"}

En este ejemplo, hemos creado un diccionario llamado diccionario que contiene tres pares clave-valor: "nombre" con el valor "Juan", "edad" con el valor 25, y "ciudad" con el valor "Madrid".

Los diccionarios son muy útiles cuando necesitas almacenar y recuperar valores asociados a claves específicas. Puedes acceder a los valores del diccionario utilizando las claves y también puedes agregar, modificar o eliminar pares clave-valor según sea necesario.

4.1 Describir las sentencias While y For

La sentencia while se utiliza para crear un bucle que se repetirá mientras se cumpla una condición. El bloque de código dentro del while se ejecuta repetidamente hasta que la condición se evalúe como falsa. Aquí tienes un ejemplo:

contador = 0

while contador < 5:

print(contador)

contador += 1

En este ejemplo, el bucle while se ejecutará mientras el contador sea menor que 5. En cada iteración, se imprime el valor del contador y se incrementa en 1. El bucle continuará hasta que el contador alcance el valor 5.

La sentencia for se utiliza para iterar sobre una secuencia de elementos, como una lista, una tupla o una cadena de texto. En cada iteración, la variable de iteración toma el valor de uno de los elementos de la secuencia. Aquí tienes un ejemplo:

frutas = ["manzana", "plátano", "naranja"]

for fruta in frutas:

print(fruta)

En este ejemplo, el bucle for itera sobre cada elemento de la lista frutas. En cada iteración, la variable fruta toma el valor de uno de los elementos de la lista y se imprime. El bucle se repetirá hasta que se hayan recorrido todos los elementos de la lista.

4.2 Describir y dar ejemplos de Range y Switch

range es una función incorporada de Python que se utiliza para generar una secuencia de números. Puede ser utilizada en combinación con un bucle for para iterar un número específico de veces. Aquí tienes un ejemplo:

for numero in range(1, 6):

print(numero)

En este ejemplo, utilizamos range(1, 6) para generar una secuencia de números del 1 al 5. El bucle for itera sobre cada número de la secuencia y lo imprime.

En Python, no hay una sentencia switch como en otros lenguajes de programación. Sin embargo, se puede simular un comportamiento similar utilizando una serie de declaraciones if-elif-else. Aquí tienes un ejemplo:

opcion = 2

if opcion == 1:

print("Opción 1 seleccionada")

elif opcion == 2:

print("Opción 2 seleccionada")

elif opcion == 3:

print("Opción 3 seleccionada")

else:

print("Opción no válida")

En este ejemplo, dependiendo del valor de la variable opcion, se imprimirá un mensaje diferente. Si opcion es igual a 2, se imprimirá "Opción 2 seleccionada". Si ninguna de las condiciones se cumple, se ejecutará el bloque else y se imprimirá "Opción no válida".

4.3 ¿Qué función tiene la sentencia break y continue?

La sentencia break se utiliza para interrumpir la ejecución de un bucle, ya sea un bucle for o un bucle while, de forma prematura. Cuando se encuentra la sentencia break, el bucle se detiene y el control del programa se transfiere a la siguiente línea de código después del bucle. Aquí tienes un ejemplo:

for numero in range(1, 6):

if numero == 4:

break

print(numero)

En este ejemplo, el bucle for itera sobre los números del 1 al 5. Cuando numero es igual a 4, se encuentra la sentencia break y el bucle se detiene. Por lo tanto, solo se imprimirán los números 1, 2 y 3.

La sentencia continue se utiliza para saltar la ejecución de las instrucciones restantes dentro de un bucle en una iteración específica y pasar a la siguiente