QUESTIONS CHECKPOINT 4

1. **¿Cuál es la diferencia entre una lista y una tupla en Python?**

   Las tuplas y las listas son dos de los 4 tipos de datos usados en Python, en ambas se pueden almacenar una colección almacenada de uno o más elementos.

- **Una de las diferencias es la sintaxis:**
  Las tuplas se crean mediante paréntesis ()

  mi_tupla = ()

  mi_tupla = (5, 55, 9, ¨Hi¨, True)

  Las listas se crean usando corchetes []

  mi_list = []

  mi_list = [10, 20, 20, ¨Python¨, False]

- Utilizo tuplas cuando necesito una colección de elementos **inmutables**.
- Utilizo listas cuando necesito una colección de elementos que puedan cambiar de tamaño es decir son **mutables**.

2. ¿ Cuál es el orden de las operaciones?

La jerarquía de los operadores determina el orden que se resuelven las expre siones de las operaciones matemáticas.

Es muy simple para poder recordar usando **PEMDAS**
P significa paréntesis,
E significa exponencial,
MD significa multiplicación y división y
AS significa suma y resta

3. ¿Qué es un diccionario en Python?

Un diccionario en Python es una colección de elementos donde cada uno tiene:
key:value
key son únicas e inmutables
value pueden ser modificados después de su creación

Los diccionarios se definen por corchetes {}

Ejemplo:

mi_diccionario = {¨nombre¨: ¨Sara¨, ¨altura¨: 160, ¨ documento¨:10050101}

4. ¿Cuál es la diferencia entre el método ordenado y la función de ordenación?

Hay dos funciones que se utiliza para ordenar en Python, **sort** y **sorted**
La función **sort** ordena la lista en su lugar intercambiando los valores en los ín dices

Ejemplo:

```
myList = [2, 19, 8, 32, 5, 10]
print(myList)
myList.sort()
print(myList)

[ 2, 19, 8, 32, 5, 10]
[ 2, 5, 8, 10, 19, 32]
```

Como podemos ver la lista se ordeno con la función **sort**

La función **sorted** no modifica la secuencia original, crea una nueva secuencia ordenada basados en la secuencia original. **sorted** puede pasarlo como un argumento.

```
numeros = [8, 2, 6, 10, 17]
    sorted_numeros = sorted(numeros)
    Print(sorted_numeros)

    [2, 6, 8, 10, 17]
```

sort y **sorted** tienen el mismo comportamiento
Se pueden obtener listas en orden ascendentes y descendentes.

5. ¿Qué es un operador de reasignación?

El termino «reasignar» se utiliza cuando se asigna un nuevo valor a una variable ya existente. Se utiliza para actualizar variables, cambiar el comportamiento de un programa, Hay que tener mucho cuidado al usar una reasigna-

ción ya que puede también producir errores si no lo utilizamos correctamente.

Ejemplo:

Si tenemos una variable X con un valor de 10 y deseamos cambiar su valor a 20
X = 10
X= 20

Tenemos que considera que al reasignar una variable se pierden los valores de referencia anteriores, entonces si tenemos una función o método que usa X como argumento y se reasigna , el nuevo valor de X no será accesible fuera de la función.