CHECKPOINT 4

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

Una lista es un conjunto de elementos que pueden ser de distintos tipos de datos y las listas son mutables, en otras palabras, la lista se puede modificar agrando, reemplazando o eliminando elementos.

- **Ejemplo lista:**

planetas = ['neptuno', ´tierra´, 'marte']

- **Ejemplo como agregar un elemento a la lista:**

```
planetas.append('jupiter')
```

print(planetas)

- **ejemplo de como eliminar un elemento de la lista:**

planetas.remove('tierra')

print(planetas)

- **ejemplo para reemplazar un elemento de una lista:**

planetas [2] = 'saturno'

print(planetas)

Una tupla al igual que una lista puede contener una serie de elementos, pero a diferencia de la lista la tupla es inmutable. En otras palabras, no se pueden eliminar elementos ni editarlos y ene l caso de que se requiera agregar elementos se debe hacer mediante una *reasignación*.

- **Ejemplo tupla:**

numeros_pares = (2,4,6,8,10)

- **Ejemplo de cómo agregar un elemento a una tupla:**

numeros_pares = numeros_pares + (16,)

print(numeros_pares)

2. **¿Cuál es el orden de las operaciones?**

Se operan de la siguiente forma:

1. Paréntesis ()
2. Exponentes **

3. Multiplicación *
4. División /
5. Adición +
6. Resta -

Ejemplo:

20 + 10 * 2 – (1+3)**2

20 + 10 * 2 – (4)**2

20 + 10 * 2 – 16

20 + 20 – 16

40 – 16

24

3. ¿Qué es un diccionario Python?

Es un almacén de datos *clave-valor*, esto significa que podemos almacenarlo en una variable y crear no solo elementos, sino también una clave con su valor correspondiente.

Ejemplo:

cantidad_festivos_meses = {

‘enero’ : 2,

‘abril’ : 3,

‘mayo’ : 1,

}

Una buena práctica es la identación de los valores dentro de la definición del diccionario de datos.

- Como obtener el primer elemento:

```
print(cantidad_festivos_meses ['enero'])
```

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

Las dos permiten trabajar con listas, pero la función de ordenación *sort()* ordena todos los elementos de la lista pero alterando su posición original por el contrario el método *sorted* permite almacenar el valor ordenado dentro de una variable diferente y también se puede obtener la ordenación de forma inversa cuando se requiera.

Ejemplos:

temperaturas = [

100,

30.

20,

10,

80

]

Función sort()

Temperaturas.sort()

Print(temperaturas)

[10, 20, 30, 80, 100]

Método sorted

temperaturas_ordenadas = sorted(temperaturas)

Print(temperaturas_ordenadas)

[10, 20, 30, 80, 100]

5. ¿Qué es un operador de reasignación?

Es aquel que permite realizar un tipo de operación sobre una variable y al mismo tiempo asignar el resultado de dicha operación a la variable con la que se esta trabajando.

Ejemplo:

Precio = 20.99

Precio = Precio + 10

Print(precio)

30.99

Pero una buena práctica seria:

Precio = 20.99

Precio += 10

Print(precio)

30.99

Esta operación se puede realizar para diferentes tipos de datos, se pueden hacer operaciones de reasignación por ejemplo a strings, enteros, tuplas.