CHECKPOINT 4

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

La mayor diferencia es que las tuplas son inmutables, no se pueden modificar una vez creadas, mientras que las listas se pueden modificar añadiendo o eliminando elementos.

Las tuplas se crean usando paréntesis y las listas corchetes.

Se debe trabajar con tuplas cuando no queremos que los elementos cambien, necesitamos que se queden inmutables. Las listas son para trabajar en los casos en los que necesitamos que los datos sean dinámicos y se puedan modificar.

tupla = (“manzana”, “pera”, “mandarina”)

lista = [“manzana”, “pera”, “mandarina”]

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

-Primero se opera lo que esté dentro de un paréntesis.

-Después tienen prioridad los exponentes.

-Luego está la multiplicación, seguida de la división. Realmente, no hay mucha diferencia en invertir el orden de estas dos, pero me gusta más multiplicar primero.

-Seguimos con la suma.

-Terminamos con la resta.

3.- ¿Qué es un diccionario Python?

Es una estructura de datos en las que cada elemento tiene una clave y un valor, por lo que, para la búsqueda de alguno de los elementos, en lugar de utilizar el índice, se utiliza la clave. Son estructuras que se pueden modificar. Un ejemplo sería, un diccionario de los de toda la vida, en la que hay palabras y cada una de ellas está asociada a una descripción.

En los diccionarios, los elementos pueden ser string, números…

```
alumnos = {
   “nombre”: “Sandra”,
   “edad”: 24,
   “carrera”: “psicología”
}
```

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

Si bien en ambos casos la lista se puede ordenador en sentido ascendente o descendente, en el caso del método (sort), se modifica la lista original, mientras que con la función (sorted) se crea una nueva lista y se mantiene intacta la original.

Como podemos ver en el ejemplo, en el caso del método, cogemos la lista y le añadimos el método sort. Esto hará que, a partir de ahora, cuando nombremos esa lista, aparecerá ordenada, ya no saldrá la original.

```
alumnos = ["Sandra", "Alberto",
"Jairo", "Nieves", "María"]
alumnos.sort()
print(alumnos)
```

```
['Alberto', 'Jairo', 'María',
'Nieves', 'Sandra']
```

En este segundo caso, debemos crear una nueva lista en la que tomamos la lista anterior ordenada. Pero, por eso, podemos seguir llamando también a la primera lista, que aparecerá sin modificar.

```
alumnos = ["Sandra", "Alberto",
"Jairo", "Nieves", "María"]
nuevos_alumnos = sorted(alumnos)

print(nuevos_alumnos)
print(alumnos)
```

```
['Alberto', 'Jairo', 'María',
'Nieves', 'Sandra']
['Sandra', 'Alberto', 'Jairo',
 'Nieves', 'María']
```

5.- ¿Qué es un operador de reasignación?

Un operador de reasignación nos ayuda a dar un valor a una variable. Cuando creamos una variable, ésta no tiene valor, debemos dárselo nosotros. Por ejemplo, creo una variable que se llama unidades. Debo decir a qué es igual esa variable.

unidades = 10

En este caso, he usado un operador de asignación muy básico. Lo que hace el símbolo = es coger el valor que tiene a la derecha y se lo asigna a la variable de la izquierda.

Los operadores también pueden ser matemáticos. Es decir, podemos coger la variable y, a partir de un valor inicial, modificarla haciendo operaciones matemáticas.

```
unidades = 10
print(unidades)
print()
unidades += 5
print(unidades)
print()
unidades *= 2
print(unidades)
```

```
10

15

30
```