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

Ambas son colecciones de datos que se utilizan para almacenar distintos valores en una misma variable. La diferencia es que la tupla es inmutable, por lo que una vez creada no se puede modificar.

Las listas están mejor preparadas para cuando los datos necesiten ser modificados con frecuencia o el tamaño de la colección cambie, también cuando el orden de los elementos sea importante.

Por otro lado, las tuplas son más rápidas y eficientes en cuanto al uso de memoria, por lo que son más adecuadas para situaciones que requieran a acceder a los datos con frecuencia, pero no necesiten ser modificados.

La sintaxis es muy parecida, salvo el uso de paréntesis para la tupla y corchetes para la lista.

tupla = (1,2,3)

lista = [1,2,3]

¿Cuál es el orden de las operaciones?

El orden que se utiliza es el PEMDAS:

1. Paréntesis
2. Exponentes
3. Multiplicación
4. División
5. Suma
6. Resta

¿Qué es un diccionario Python?

Es una colección de datos similar a las listas o tuplas que permite almacenar gran cantidad de datos de forma ordenada. A diferencia de estas, los valores de los diccionarios constan de dos partes. La key, que es la clave que nos permite encontrar el dato en el diccionario y el valor o dato a la que esta representa.

Algunas de las ventajas de los diccionarios son su flexibilidad al poder almacenar distintos tipos de datos, incluyendo listas, tuplas o incluso otros diccionarios. Además por su naturaleza dinámica, pueden reducirse o agrandarse en cualquier momento, haciéndolos ideales para situaciones en las que no sepas de antemano el numero de elementos que vas a utilizar.

El uso de claves hace que sean valores de acceso rápido, haciendo que puedas acceder a los datos contenidos de manera eficiente.

Ej.

diccionario = {‘nombre’: ‘Mikel’, ‘edad’: 29}

Añadir y eliminar elementos al diccionario

- diccionario[“ciudad”] = “Bilbao”
- del diccionario[“ciudad”]

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

La función sort() ordena los datos directamente en la lista, mientras que el método sorted() crea una copia de la lista y ordena esta última. Por defecto se ordenan los datos de manera ascendente, aunque también pueden ordenarse de forma descendente usando reverse.

El metodo sort() solo funciona con listas, mientras que la funcion sorted() es compatible con cualquier elemento.

El metodo sort() es más eficiente en el uso de memoria, al no tener que crear una nueva lista, es ideal para grandes bases de datos, mientras que sorted() es más versátil al poder usarse con listas, tuplas, strings, etc...

Ej. 1

lista = [1,2,3,4,5]

list.sort(reverse=True)

Ej. 2

list = [1,2,3,4,5]

sorted_list = sorted(list, reverse=True)

¿Qué es un operador de reasignación?

Los operadores de asignación se usan para dar valor a una variable, el utilizado más a menudo es el operador igual (=), que se usa para asignar un valor a una variable. Por ejemplo, x = 3, el asignador estaría asignándole un valor de 3 a la variable x.

Otros operadores de asignación serian para acortar operaciones aritméticas, pudiendo almacenar el resultado en la primera variable sin necesidad de crear una nueva.

Ej.

total = 100

tip = 20

total += tip (total = total + tip)

Esto daría como resultado 120