Full Stack Programme_M2C4: Checkpoint 4

Iñigo Uribarri

24/04/2024

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

La diferencia principal entre una lista y una tupla es que la primera es mutable, es decir pudiendo manipular los valores existentes y añadir nuevos, y la segunda no lo es. Por lo que podemos decir que las listas son mutables y las tuplas inmutables.

Además, según la web geekforgeeks existen ventajas y desventajas en el uso de listas o tuplas:

- Por un lado, la iteración sobre las listas necesita de más tiempo en comparación con las tuplas que es más rápida en términos temporales, las listas consumen más memoria en comparación con las tuplas. Por último, debido a su mutabilidad, las listas son más propensas a cambios y errores inesperados en comparación con las tuplas que debido a su inmutabilidad son menos dadas a causar errores.
- Por otro lado, las listas permiten llevar a cabo operaciones y métodos 'built-in' lo que hace más fácil operar con ellas en comparación con las tuplas que permiten menos métodos 'built-in'.

2. ¿Cuál es el orden de las operaciones?

Es útil utilizar la regla nemotécnica PEMDAS para: en primer lugar, P de paréntesis seguido de, E de exponentes, M de multiplicación, D de división, A de adición o suma, y S de sustracción o resta.

En la operación: 4*2+(3*2) /3-3^2. Se desarrolla de la siguiente manera en Python.

Paréntesis: 4*2+6/3-3^2

Exponentes: 4*2+6/3-9

Multiplicación: 8+6/3-9

División: 8+2-9

Adición: 10-9

Substracción: 1

3. ¿Qué es un diccionario Python?

Un diccionario es una colección de pares compuesto por claves y sus respectivos valores. Estas son estructuras mutables (como las listas) y no ordenadas. Además, los diccionarios pueden anidar claves dentro de otras de forma que se generen objetos complejos. Un ejemplo de un diccionario sería el siguiente:

```
matemáticos famosos= {
        "Leonhard Euler": {“año nacimiento": 1707, "origen": "Suiza"},

        "Carl Friedrich Gauss": {“año nacimiento": 1777, "origen": "Alemania”},

          "Ada Lovelace": {“año nacimiento": 1815,"origen": "Reino Unido"    }}
```

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

A continuación, se realiza un resumen del propósito de la función sorted () y el método. sort usado en Python:

Por un lado, la función sorted () es una función integrada que toma un conjunto de datos desordenados, como una lista, tupla o conjunto, y devuelve una nueva lista ordenada que contiene los mismos elementos que el conjunto original. Es importante destacar que sorted () no altera el conjunto original, sino que crea una nueva lista ordenada basada en los elementos del conjunto original.

Por otro lado, El método. sort () es una función específica de las listas que organiza los elementos de la lista en su lugar, lo que significa que modifica la lista original y no devuelve un nuevo objeto. Este método reorganiza la lista original en orden ascendente por defecto, sin crear una nueva lista. Se utiliza llamando al método. sort () en una lista existente.

5. ¿Qué es un operador de reasignación?

Un operador de reasignación, también llamado operador de asignación compuesta es una herramienta en programación que combina una operación aritmética con la asignación de valores en una sola expresión. En Python, un ejemplo común es el operador +=, que suma un valor al valor actual de una variable y luego asigna el resultado de vuelta a la misma variable. Este, se utiliza para actualizar el valor de una variable, sin necesidad de escribir una línea adicional.