M2C4

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

Tienen dos diferencias fundamentales, una es la forma de construirse, es decir, los tuples se definen con "()" y las listas con "[]" y la característica de que los tuples son inmutables y las listas no. Esto quiere decir que no se puede modificar un tuple una vez creado, solo se puede volver a generar otro distinto con los cambios.

Aparte, las tuplas tienen un mayor rendimiento y usan menos memoria debido a dicha inmutabilidad. También, cabe destacar que las tuples se pueden usar como llaves para diccionarios.

Por último, hay muchos métodos que solo están pensado para listas, lo cual implica que hay que hacer código específico para estos últimos.

2. ¿Cuál es el orden de las operaciones?

Depende del medio, en Python, siguen el método que se suele usar en matemáticas general de PEMDAS y lo que define es el orden en el que se ejecutan las diferentes operaciones en ecuaciones combinadas o operaciones compuestas de múltiples partes.

-(P)aréntesis

-(E)xponentes, también entran aquí las raíces ya que en definitiva son exponentes fraccionados.

-(M)ultiplicación y (D)ivisión

-(A)dición y (S)ustracción

De las letras en paréntesis sale el acrónimo.

Hay que entender que el orden de las operaciones varía el resultado de la misma, hay muchas calculadoras antiguas que no utilizan PEMDAS y esperan que sea uno mismo el que de el orden deseado de operación. Tristemente aun siguiendo este modelo se pueden dar ambigüedades y se deben resolver con los apropiados paréntesis.

Un ejemplo de esto podría ser "a = 2 + 3 / 3 + 6". En Python que sigue PEMDAS, resultaría en 9 pero puede ser un valor no deseado. Si lo que queríamos era $\frac{2+3}{3+6}$, para que sea correcto en programación debería ser escrito así: "a = (2 + 3) / (3 + 6)" dando prioridad a lo que está entre paréntesis.

3. ¿Qué es un diccionario Python?

Es un tipo de dato que permite organizar información permitiendo coger datos dispersos y organizarlos en algo coherente. Enlaza llaves con valores. Las llaves son inmutables y únicas, siempre pueden ser números y textos(strings). Los tuples pueden ser llaves si solo contienen contenido inmutable. El orden se preserva dentro del diccionario y se considera eficiente en cuanto a recursos para almacenar información.

Se definen con "{}" y cada par está enlazado con el símbolo ":" donde a la izquierda está la llave y a la derecha el valor. Ejemplo: {2 : 3} la llave es 2 y el valor es 3.

Se le pueden añadir valores extra haciendo asignaciones y se pueden modificar los valores de cada llave. Para borrar se tiene que borrar el conjunto llave/valor (función "del"

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

La diferencia entre sorted() (función) y .sort() (método) es que el método modifica la lista directamente y no da un valor de retorno, en cambio la función no modifica la lista original pero da en el retorno la función ordenada. Sorted() siempre requiere un argumento que es el que se va a ordenar sorted(lista). Sort() requiere que esté se aplique el método a un elemento lista.sort().

Ejemplo de uso sorted():

Sorted(lista)

#No haría nada, se tiene que asignar a una variable o imprimirlo

a = sorted(lista)

print(a) -> Obtendríamos la lista ordenada.

Ejemplo de uso sort():

a = lista.sort()

print(a)

#Obtenemos None, la función sort() no devuelve nada.

lista.sort()

print(lista) -> Obtendríamos la lista ordenada, indicando que se ha modificado la lista original al ejecutar el método.

5. ¿Qué es un operador de reasignación?

Son operadores de asignación que aplican a la propia variable de salida. Sirven para simplificar código y mejorar rendimiento ya que la variable solo se evalúa una vez al ejecutar la línea. Como ejemplo:

Valor = Valor + 3 -> Valor += 3

Dependiendo al tipo de variable a la que se le aplique puede funcionar diferente o dar error si se utilizan datos incompatibles. Como ejemplo, en un tuple se puede usar para añadir valores “tupla += (3,2)” añadirá esos valores al final de la secuencia, sin embargo, si le decimos “tupla += 3” nos dará un fallo por no ser compatibles los datos.

Existen para la mayoría de operaciones aritméticas básicas:

- “+=” suma
- “-=” resta
- “*=” multiplicación
- “/=” división
- “//=” división a entero
- “%=” modulo o resto de la división
- “**=” exponente