

Preguntas teóricas

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

Si bien tanto una lista como una tupla son colecciones, se diferencian no sólo en su sintaxis, sino en el hecho de que la lista es mutable y la tupla inmutable.

Esto significa que una lista puede ser modificada, puedo agregar, remover y cambiar elementos por otros. Además, al aplicar la función `sort`, esta modificará la lista de forma permanente (mientras que esta no puede aplicarse a la tupla).

Ejemplo: `mi_lista = [2,3,4]`

La tupla no puede modificarse. En caso de que sea necesario hacerlo, lo que se hace es "sobrescribir" el nombre de la variable donde se encuentra almacenada esa tupla y agregar el elemento que se desee. La tupla original no será cambiada, habremos creado una nueva con el mismo nombre pero un id diferente. Para modificarla se puede convertir en una lista, remover el valor que se quiera y volver a convertirlo en una tupla. Sin embargo, estas no son buenas prácticas. Si estamos intentando modificar una tupla, posiblemente hemos elegido el tipo de dato equivocado y debimos elegir una lista desde el principio, además de que estas modificaciones pueden provocar bugs.

Ejemplo: `mi_tupla = (2,3,4)`

¿Cuál es el orden de las operaciones?

El orden de las operaciones corresponde al acrónimo PEMDAS: Paréntesis, exponente, multiplicación, división, adición(suma) y sustracción(resta). La división y multiplicación pueden ser intercambiables.

¿Qué es un diccionario Python?

Un diccionario Python sería similar a un objeto en JavaScript. Es un tipo de colección mutable que se conforma por pares llave(o clave)/valor. En lugar de acceder a los valores por índice, como sucede con las tuplas y listas, en el diccionario accedo al valor a través de la clave. Esa clave debe ser inmutable (no se puede modificar pero sí se puede eliminar el par clave/valor).

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

Método `sort`

`mi_lista.sort()` - Ordena los elementos de una lista en orden alfabético (o creciente si se trata de números).

`mi_lista.sort(reverse=True)` - Ordena los elementos de una lista en orden alfabético inverso (o decreciente si se trata de números).

Esta función además modifica la lista, por lo que no permite almacenar esos datos en una nueva variable (la lista ya ha sido modificada), ni puede aplicarse a una tupla porque esta no se puede modificar.

Función `sorted`

nueva_lista = sorted(mi_lista) - Permite ordenar la lista o tupla y almacenar los datos en una nueva variable.

¿Qué es un operador de asignación?

Un operador de asignación se utiliza para asignar valor a una variable. El operador de asignación más común es =, pero existen varios.

my_variable = 1

Otros ejemplos:

my_variable -= 1 --> Equivale a: my_variable = my_variable - 1

my_variable /= 1 --> Equivale a: my_variable = my_variable / 1

my_variable *= 1 --> Equivale a: my_variable = my_variable * 1

my_variable %= 1 --> Equivale a: my_variable = my_variable % 1

my_variable **= 1 --> Equivale a: my_variable = my_variable ** 1

my_variable += 1 --> Equivale a: my_variable = my_variable + 1