

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

Las listas y las tuplas son tipos de datos secuenciales que almacenan colecciones ordenadas de elementos. Mientras que las listas son mutables, las tuplas son inmutables. Se recomienda utilizar las listas para datos que necesitan cambiarse con frecuencia, mientras que las tuplas se utilizarían para datos que no necesitan modificaciones frecuentes. Las listas también permiten agregar o eliminar contenido. Las listas se definen con [] y las tuplas con ().

```
lista = ['a', 'b', 'c', 'd']
tupla = ('a', 'b', 'c', 'd')

# Ejemplo de que las listas son mutables y las tuplas no
lista[0] = 'aa'
print(lista)
tupla[0] = 'aa' # Da error porque las tuplas no son mutables
```

¿Cuál es el orden de las operaciones?

Se sigue el orden PEMDAS: Primero se harían los paréntesis, después los exponenciales, seguido de multiplicaciones y divisiones, luego las sumas y por último las restas.

¿Qué es un diccionario Python?

Es una estructura de datos que guarda variables y sus valores en un diccionario común (clave:valor). Se usan cuando necesitas buscar elementos rápidamente o almacenar datos relacionados, permite el acceso rápido a los valores a través de sus claves. Son mutables, por lo que se pueden modificar. No permite duplicar claves. Los diccionarios se definen con {}.

```
dict_example = {'name': 'Beñat', 'Last name': 'Iglesias', 'age': 32}
print(dict_example['age']) #32

# Ejemplo de modificación de diccionario
dict_example['city'] = 'Donostia'
print(dict_example) #{'name': 'Beñat', 'Last name': 'Iglesias', 'age': 32, 'city': 'Donostia'}
```

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

Las dos son maneras de ordenar las listas. Sin embargo, el método .sort() modifica la lista original y la función sorted(), en cambio, no. Utilizando .sort() la lista queda ordenada permanentemente y no se puede modificar, mientras que sorted() no modifica la lista original y devuelve una nueva lista ordenada. Por tanto, sorted() es adecuado para cuando se necesita mantener intacta la lista original.

```
lista = [2, 7, 8, 14, 3, 19, 9]
sort_lista = lista.sort()
print(sort_lista) # None

lista = [2, 7, 8, 14, 3, 19, 9]
sorted_lista = sorted(lista)
print(sorted_lista) # [2, 3, 7, 8, 9, 14, 19]
```

¿Qué es un operador de reasignación?

Es una forma abreviada de modificar el valor de una variable utilizando operadores matemáticos en combinación con el símbolo =

Permite modificar el valor de una variable de manera eficiente sin necesidad de escribir el nombre de la variable dos veces.

```
total = 100

total = total + 10 # 110
total += 10 # 120
total -= 10 # 110
total *= 2 # 220
total /= 10 # 22.0
total //= 10 # 2.0
total **= 2 # 4.0
total %= 2 # 0.0
```