

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

Una tupla es similar a una lista en el sentido de que también puede almacenar varios elementos en una misma variable. Sin embargo, las tuplas son inmutables, lo que significa que sus elementos no se pueden modificar después de su creación.

En general se utilizan tuplas para tipos de datos distintos, mientras que las listas se utilizan para el mismo tipo de datos. Recorrer todos los elementos de una tupla es más rápido que hacerlo sobre una lista.

```
my_list = ['Ibon', 'Markel', 'Irantzu', 'Aitor']  
my_tuple = ('PHP', 'Python', 'Java', 'CSS')
```

- ¿Cuál es el orden de las operaciones?

El orden en que se resuelven las operaciones en Python se puede identificar con las siglas PEMDAS = Parentesis, Exponentes, Multiplicación, División, Adición (suma), Sustracción (resta); el cual en realidad es un acrónimo del orden del proceso de resolución de las operaciones matemáticas.

```
4 / 2 + 5 - 2 * (3+2) ** 2  
4 / 2 + 5 - 2 * 5 ** 2  
4 / 2 + 5 - 2 * 25  
4 / 2 + 5 - 50  
2 + 5 - 50  
7 - 50  
-43
```

- ¿Qué es un diccionario Python?

Es una estructura de datos que permite almacenar cualquier tipo de información, desde cadenas de texto o caracteres hasta números enteros, con decimales, listas e incluso otros diccionarios.

Son dinámicos, pueden crecer o decrecer, se pueden añadir o eliminar elementos. Son indexados, los elementos del diccionario son accesibles a través del key. Y son anidados, un diccionario puede contener a otro diccionario en su campo value.

```
my_dictionary = {  
    'Name': 'Ibon',  
    'Age': 30,  
    'Country': 'Spain',  
    'City': 'Bilbo'  
}
```

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

Una diferencia clave entre `sort()` y `sorted()` es que `sorted()` retornará una nueva lista, mientras que `sort()` ordena la lista en su lugar. Además, `sorted()` es aplicable a cualquier iterable y no solamente a objetos de tipo lista, mientras que `sort()` se invoca dentro de una instancia de lista.

```
my_list = ['Ibon', 'Markel', 'Irantzu', 'Aitor']
my_list.sort()
print(my_list)

my_list = ['Ibon', 'Markel', 'Irantzu', 'Aitor']
sorted_list = sorted(my_list)
print(sorted_list)
print(my_list)
```

```
['Aitor', 'Ibon', 'Irantzu', 'Markel']
['Aitor', 'Ibon', 'Irantzu', 'Markel']
['Ibon', 'Markel', 'Irantzu', 'Aitor']
```

- ¿Qué es un operador de reasignación?

Son una herramienta importante en Python para modificar el valor de las variables. En lugar de tener que redefinir la variable cada vez que se modifica su valor, los operadores de reasignación permiten modificarla directamente; son una herramienta importante en Python para modificar el valor de las variables.

```
age = 30
age += 1
# age variable will stay as 31
```