

## Diferencia entre una lista y una tupla en Python:

Una lista es una colección ordenada y mutable de elementos en Python, lo que significa que puedes cambiar, agregar o eliminar elementos después de crear la lista.

Una tupla, por otro lado, es una colección ordenada e inmutable de elementos en Python. Una vez que se crea una tupla, no se pueden modificar sus elementos.

Ejemplo :

```
--- Tupla (immutable)---
```

```
mi_tupla = (1, 2, 3)
```

```
mi_tupla.append(4) Esto causaría un error, ya que las tuplas son  
inmutables print(mi_tupla) tendría la Salida: (1, 2, 3)
```

### Orden de las operaciones:

En Python, las operaciones se realizan siguiendo las reglas estándar de precedencia de operadores, donde las operaciones entre paréntesis tienen la mayor prioridad, seguidas de la exponenciación, la multiplicación y la división, y luego la suma y la resta.

```
resultado = 5 + 3 * 2
```

La multiplicación se realiza primero, luego la suma print(resultado) la Salida: 11

### Diccionario en Python:

Un diccionario en Python es una colección no ordenada y mutable de pares clave-valor. Cada clave debe ser única y está asociada a un valor. Los diccionarios son útiles para almacenar y recuperar datos rápidamente utilizando claves en lugar de índices numéricos como en las listas

ejemplo :

```
# Diccionario (clave-valor)
```

```
mi_diccionario = {'nombre': 'Abdellah', 'edad': 40, 'ciudad': 'Bilbao'}
```

```
print(mi_diccionario['nombre'])
```

```
# Acceso al valor utilizando la clave
```

```
la salida sera : Abdellah
```

### Diferencia entre el método sorted y la función sorted:

El método sorted se aplica a objetos iterables como listas, tuplas, etc., y devuelve una nueva lista ordenada sin modificar la lista original.

La función sorted se puede aplicar a cualquier objeto iterable y devuelve

una nueva lista ordenada. Es más flexible que el método sorted ya que no está vinculada a ningún tipo de objeto específico.

# Método sorted

```
lista_original = [3, 1, 4, 1, 5]
lista_ordenada = sorted(lista_original)
print(lista_ordenada) # Salida: [1, 1, 3, 4, 5]
print(lista_original) # Salida : [3, 1, 4, 1, 5]
La lista original no se modifica
```

# Función sorted

```
tupla_original = (3, 1, 4, 1, 5)
tupla_ordenada = sorted(tupla_original)
print(tupla_ordenada) # Salida: [1, 1, 3, 4, 5]
print(tupla_original) # Salida : (3, 1, 4, 1, 5) no se modifica
```

### Operador de reasignación:

En Python, el operador de reasignación = se utiliza para asignar un valor a una variable. Por ejemplo, en `x = 5`, el valor 5 se asigna a la variable x. También se puede usar junto con otros operadores, como +=, -= para realizar una operación y luego asignar el resultado de nuevo a la misma variable.

```
X = 5
x += 3 # Equivalente a x = x + 3
print(x) #Salida: 8
x -= 3 # Equivalente a x = x - 3
print(x) #Salida: 2
x *= 3 # Equivalente a x = x * 3
print(x) #Salida: 15
```