

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

La lista es modificable (mutability) mientras que la tupla una vez creada no se puede modificar (unmutability).

Como la tupla es inmutable hay que hacer reasignación de variables para realizar modificaciones, mientras que en la lista se pueden añadir y quitar elementos, ordenarla alfabéticamente, etc

La lista se encapsula entre [] y la tupla entre ().

Ejemplo de lista:

```
dias_semana = ["lunes", "martes", "miercoles", "jueves", "viernes", "sabado", "domingo"]
```

```
tercer_dia = dias_semana[2] # miercoles
```

Ejemplo de tupla:

```
estaciones = ("verano", "otoño", "invierno", "primavera")
```

```
segundo_elemento = estaciones[1] # otoño
```

2.- ¿Cuál es el orden de las operaciones?

Paréntesis

Exponente

Multiplicación

División

Suma

Resta

La multiplicación y división tienen la misma prioridad.

La suma y la resta también tienen la misma prioridad, aunque la convención suele ser que el orden va de izquierda a derecha,

3.- ¿Qué es un diccionario Python?

Un diccionario es una colección de elementos formados por pares de clave-valor.

A través de la clave, podemos obtener el valor asociado a ella.

La clave tiene que ser única, es decir, no se puede repetir.

Se puede anidar un diccionario dentro de otro, se pueden incluir listas, etc como se ven en el siguiente ejemplo.

```

arbol_familiar = {
    "abuelos": {
        "paternos" : ["Manuel", "Gracia"],
        "maternos": ["Ricardo", "Marina"]
    }
    "padre" : "Angel",
    "madre": "Yolanda",
    "hijos": ["Enrique", "Isabel", "Teresa"]
}

print(arbol_familiar["padre"])           # Angel
print(arbol_familiar["hijos"][1] )      # Isabel
print(arbol_familiar["abuelos"]["maternos"][0])  #Ricardo

```

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

El método ordenado **sort()** modifica el orden de los elementos de una lista, mientras que la función de ordenación **sorted** ordena los elementos de una lista, el resultado se asigna a una nueva variable y la lista original queda como estaba.

En los ejemplos siguientes como dentro de la lista tenemos cadenas de caracteres, los elementos de la misma se ordenan alfabéticamente.

Ejemplo 1: Método ordenado

```

nombres = ["Pablo", "Elena", "Isabel", "Teresa", "Ana" ]
nombres.sort()
print(nombres)    # ["Ana", "Elena", "Isabel", "Pablo", "Teresa"]

```

Ejemplo 2: Función de ordenación

```

nombres = ["Pablo", "Elena", "Isabel", "Teresa", "Ana" ]
nombres_ordenados = sorted(nombres)
print(nombres)           # ["Pablo", "Elena", "Isabel", "Teresa", "Ana" ]
print(nombres_ordenados) # ["Ana", "Elena", "Isabel", "Pablo", "Teresa"]

```

La función de ordenación la utilizamos cuando queremos trabajar con los elementos de una lista y que estos elementos estén ordenados, pero sin modificar la lista original.

5.- ¿Qué es un operador de asignación?

El operador de asignación es aquel que se utiliza para asignar un valor a una variable. El operador de asignación básico es el símbolo “=”.

```
nombre = "Teresa"
```

A la variable nombre se le asigna la cadena alfanumérica “Teresa”

También existen los operadores de asignación abreviados, que permiten realizar una operación aritmética y asignar el resultado a una variable en un solo paso.

Los operadores de asignación abreviados son:

`+=`

`-=`

`*=`

`/=`

`//=`

`%=`

`**=`

Por ejemplo, “total += 20” es equivalente a “total = total + 20”