

Las preguntas teóricas son:

1. ¿Cuál es la diferencia entre una lista y una tupla en Python?
2. ¿Cuál es el orden de las operaciones?
3. ¿Qué es un diccionario Python?
4. ¿Cuál es la diferencia entre el método ordenado y la función de ordenación?
5. ¿Qué es un operador de reasignación?

Respuestas a las preguntas teóricas:

- 1) En Python, las listas y las tuplas son dos tipos de estructuras de datos similares pero con algunas diferencias clave:

Mutabilidad:

- a. **Listas:** Son mutables, lo que significa que puedes cambiar, agregar o eliminar elementos después de que la lista ha sido creada.

Ejemplo:

`numeros = [1, 2, 3, 4, 5]` esta es una lista de numeros

`numeros[1] = 10` aqui cambiamos un elemento de la lista por otro

`numeros.append(6)` aqui añadimos otro elemento a la lista

`del numeros[2]` aqui eliminamos un elemento de la lista

`print(numeros)` salida: `[1, 10, 4, 5, 6]`

Como puedes ver si puede manipular los elementos de la lista con facilidad sin la necesidad de crear otra lista.

- b. **Tuplas:** Son inmutables, lo que significa que una vez que se crea una tupla, no se pueden cambiar sus elementos. Si necesitas modificar una tupla, debes crear una nueva.

Ejemplo:

`colores = ('rojo', 'verde', 'azul')`. Hemos creado una tupla de colores

`colores_lista = list(colores)` Convertir la tupla en una lista para modificarla

`colores_lista[1] = 'amarillo'` .Modificar el segundo elemento de la lista

`colores_modificados = tuple(colores_lista)` Convertir la lista modificada de nuevo en una tupla

Sintaxis:

- c. **Listas:** Se definen mediante corchetes `[]` y los elementos están separados por comas.
- d. **Tuplas:** Se definen mediante paréntesis `()` y los elementos están separados por comas.

Uso:

- e. **Listas:** Se utilizan cuando necesitas una colección ordenada de elementos que puedan ser modificados. Por ejemplo, una lista de nombres de estudiantes.
- f. **Tuplas:** Se utilizan cuando necesitas una colección ordenada de elementos que no cambiarán. Por ejemplo, coordenadas geográficas o una fecha y hora.

2) En Python, el orden de las operaciones sigue las reglas estándar de las matemáticas, lo que se conoce como PEMDAS :

1. **Paréntesis** ()
2. **Exponentes** **
3. **Multiplicación** *
4. **División** /
5. **Adición** +
6. **Sustracción** -

Ejemplo:

Parentesis:

`resultado = (5 + 2) * (4 - 1)` Primero se resuelve la expresión dentro de los paréntesis, luego se multiplican los resultados

Exponentes:

`resultado = 2 ** 3` 2 elevado a la potencia de 3

Multiplicación y División:

`resultado = 10 * 2 / 5` Primero se realiza la multiplicación y luego la división

Suma y Resta:

`resultado = 10 - 3 + 2` Primero se realiza la sustracción y luego la adición

3) Un diccionario en python es un metodo que sirve para crear mas de un elemento en el y funciona de la siguiente manera:

```
nombre_del_diccionario = { clave: 'Valor'}
```

donde se puede acceder al tipo de dato de la siguiente manera :

```
print(nombre_del_diccionario[clave])
```

y en el se puede crear diferentes tipos de datos como cadena, numeros, lista y la principal carateristica del diccionario son las claves con la cuales se puede acceder al valor de cada modificarlo y hacer otras operaciones..

Ejemplo:

```
mi_diccionario = {"clave1": valor1, "clave2": valor2, "clave3": valor3}
print(mi_diccionario["clave1"])
```

- 4) El metodo ordenado actua en la lista original, mientras que la funcion de ordenacion mantiene la lista original y devuelve una nueva lista de ordenacion.

Ejemplo:

```
lista = [3, 1, 4, 1, 5]
```

```
lista.sort() , metodo de ordenacion actua en la lista original
```

```
print(lista)
```

```
nueva_lista = sorted(lista) . funcion de ordenacion devuelve una nueva lista
```

```
print(nueva_lista)
```

- 5) Un operador de reasignacion es el que se usa en python para la suma, resta ,multiplicacion y division elevacion al cuadrado y sacar el modulo de un elemento numerico en python (num +=, num -=, num* =, *num/=*, *num //=*, *num%=*, *num**=*)

ejemplo:

```
total = 100
```

```
total += 10 resultado 110
```

```
total -= 10 resultado 90
```

```
total *= 2 resultado 200
```

```
total /= 10 resultado 10
```

```
total //= 10 resultado 10
```

```
total **= 2 resultado 100
```

```
total %= 2 resultado 0
```

Como puede observar con estos operadores hemos hecho varias operaciones matematicas.