

CHECKPOINT 4 – PREGUNTAS TEÓRICAS

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

Mutabilidad: Las listas son mutables, las tuplas son inmutables.

Sintaxis: Las listas usan [], las tuplas usan ().

Uso: Las listas son útiles cuando necesitas una colección de elementos que puede cambiar. Las tuplas son útiles para colecciones de elementos que no deben cambiar.

2. ¿Cuál es el orden de las operaciones?

En Python, el orden de las operaciones sigue las reglas matemáticas estándar, conocidas como PEMDAS:

1. **Paréntesis ():** Las operaciones dentro de paréntesis se realizan primero.
2. **Exponentes **:** Los exponentes se calculan a continuación.
3. **Multiplicación *, División /, División entera //, Módulo %:** Estas operaciones se realizan de izquierda a derecha.
4. **Suma + y Resta -:** Estas operaciones también se realizan de izquierda a derecha.

Ejemplo:

```
resultado = (2 + 3) * 4 ** 2 / (1 + 1)
print(resultado)
```

En este caso, el orden de las operaciones sería:

1. Paréntesis: (2 + 3) y (1 + 1)
2. Exponentes: 4 ** 2
3. Multiplicación y División: (5) * 16 / (2)
4. Resultado final: 40.0

3. ¿Qué es un diccionario de Python?

Un diccionario en Python es una **estructura de datos que permite almacenar su contenido en forma de llave y valor**. Un diccionario en Python es una colección de elementos, donde cada uno tiene una llave (key) y un valor (value).

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

En Python, tanto el método `sort()` como la función `sorted()` se utilizan para ordenar listas, pero tienen algunas diferencias clave:

Método `sort()`

- **Modifica la lista original:** El método `sort()` ordena la lista en su lugar, es decir, modifica la lista original.
- **No devuelve una nueva lista:** No devuelve una nueva lista, simplemente retorna `None`.
- **Solo para listas:** Este método solo se puede usar con listas.

```
mi_lista = [3, 1, 4, 1, 5, 9]
mi_lista.sort()
print(mi_lista) # Salida: [1, 1, 3, 4, 5, 9]
```

Función `sorted()`

- **No modifica la lista original:** La función `sorted()` devuelve una nueva lista ordenada y no modifica la lista original.
- **Funciona con cualquier iterable:** Puede usarse con cualquier iterable, no solo con listas (por ejemplo, tuplas, cadenas, etc.).
- **Más flexible:** Permite ordenar cualquier iterable sin necesidad de convertirlo a una lista primero.

```
mi_lista = [3, 1, 4, 1, 5, 9]
```

```
nueva_lista = sorted(mi_lista)
print(mi_lista)    # salida: [3, 1, 4, 1, 5, 9]
print(nueva_lista) # salida: [1, 1, 3, 4, 5, 9]
```

5. ¿Qué es un operador de reasignación?

Un operador de reasignación es un operador que se utiliza para actualizar el valor de una variable existente. Estos operadores combinan una operación aritmética la asignación.

Ejemplos:

Suma y asignación (+=):

```
x = 5
x += 3  # Equivalente a x = x + 3
```

Resta y asignación (-=):

```
y = 10
y -= 4  # Equivalente a y = y - 4
```