Nombre: Amanda Velásquez

Fecha: 09 de marzo de 2025

Curso: PT Full Stack Development with JavaScript, Python, React

M2C4 Python Assignment

I. Diferencia entre una lista y una tupla en Python

La diferencia es que una se puede modificar y la otra no. Por ejemplo:

Una lista sería como hacer un castillo con bloques de LEGO, se puede armar el castillo como uno quiera. Si más tarde se decide que se quiere hacerlo más alto, más ancho o cambiarle una torre, se pueden quitar y poner piezas. Esto es como una lista en Python, porque permite agregar, quitar y cambiar elementos.

Una tupla, sería como construir el castillo con palitos y pegamento. Una vez que se pegan las piezas y el pegamento se seca, ya no se pueden mover sin romperlo. Es decir, esto es como una tupla en Python, porque después de crearla, no puedes modificarla.

Entonces, si se necesita algo que se pueda cambiar en cualquier momento, como un castillo de LEGO, se usa una lista. Pero si se quiere algo que se mantenga fijo como un castillo de palitos pegados, se usa una tupla.

II. **Orden de las operaciones**

Para asegurar el orden correcto de las operaciones, se sigue una regla conocida como PEMDAS, un acrónimo en inglés, que ayuda a recordar qué se debe resolver primero. Para iniciar, las operaciones dentro de paréntesis se realizan antes que cualquier otra cosa, sin importar lo que haya fuera.

Después, vienen los exponentes, representados con **. O sea que cualquier número elevado a una potencia se calculará antes de continuar con las demás operaciones.
A continuación, la multiplicación (*) y la división (/). Estas tienen la misma prioridad y se resuelven en el orden en el que aparecen, de izquierda a derecha.

Por último, la suma (+) y la resta (-), que también se resuelven de izquierda a derecha, una vez que todas las operaciones han sido completadas.
Por ejemplo:

```
operación= 8 + 2 * 5 - (9 + 2) ** 2
print(operación)
```

- Paréntesis: (9 + 2) = 11.
- Exponente: 11 ** 2 = 121.
- Multiplicación: 2 * 5 = 10.

- Suma y resta en orden: 8 + 10 = 18, y 18 - 121 = -103.

Es importante seguir esa lógica porque si realizamos los cálculos en otro orden, se puede obtener un resultado incorrecto.

III. Diccionario Python

Un diccionario en Python es como un almacenamiento donde cada cosa tiene una etiqueta para encontrarla fácilmente. En un diccionario cada elemento tiene un nombre único o clave, y cada clave está asociada a un valor.

Por ejemplo una agenda de contactos. En una lista, los números de teléfono estarían en orden, pero no se podría saber de quién es cada uno sin revisar toda la lista. En un diccionario, cada número de teléfono está guardado con el nombre de la persona. Por ejemplo:

```
agenda = {
  "Alan": "555-1234",
  "Pedro": "555-5678",
  "Carla": "555-9876"
}
```

Entonces, para saber el número de Ana, basta con preguntar por su clave:

```
print(agenda["Ana"])
```

Esto me dirá el teléfono de ese contacto. Este sería el ejemplo más básico, aunque también se puede guardar información más detallada. Además, los diccionarios permiten modificar, agregar y eliminar información.

IV. Diferencia entre el método ordenado y la función de ordenación

El método .sort() modifica la lista original y no devuelve ningún valor.
Por ejemplo:

```
números = [3, 1, 4, 2]
números.sort()
print(números)  Esto ordena la lista y la modifica permanentemente.
```

La función sorted() toma una lista y devuelve una nueva lista ordenada, sin modificar la original. Es útil cuando se quiere ordenar los datos, pero al mismo tiempo conservar la versión sin ordenar. Por ejemplo:

números = [3, 1, 4, 2]
nueva_números = sorted(números)
print(nueva_lista) Nos mostrará la lista ordenada.
print(numeros) Nos mostrará la lista original sin cambios.

V.Operador de reasignación

Un operador de reasignación en Python es una herramienta que permite actualizar el valor de una variable de forma más simple. En lugar de escribir la operación completa, se usa una versión abreviada que realiza la operación y guarda el nuevo valor en la misma variable.
Por ejemplo, si tenemos una variable que vale 5 y queremos sumarle 2, podríamos escribir:
x = x + 5 Pero si se usa un operador de reasignación, se puede escribir así:
x += 5 y representa lo mismo, de igual forma se puede hacer para el resto de operaciones matemáticas. Esto cambia el valor de la variable, ahora x vale 7.