

Listas en Python

🕒 Fecha	@1 de octubre de 2023 19:40
📄 Asignatura	Programación Imperativa
☑ Revisado	<input type="checkbox"/>

Ejercicios

1. Crea una lista vacía llamada `numeros`.
2. Agrega los números del 1 al 5 a la lista utilizando el método `append()`.
3. Imprime el número en la posición 3 de la lista.
4. Modifica el número en la posición 2 de la lista y cámbialo por 10.
5. Elimina el último número de la lista utilizando el método `pop()`.
6. Ordena la lista de forma ascendente utilizando el método `sort()`.
7. Imprime la lista final.

```
numeros = []
numeros.append(1)
numeros.append(2)
numeros.append(3)
numeros.append(4)
numeros.append(5)

print(numeros[2]) # Imprime 3

numeros[1] = 10

numeros.pop()

numeros.sort()

print(numeros) # Imprime [1, 3, 4, 10]
```

¡Inténtalo por ti mismo y practica con las listas en Python!

Definición

En Python, las listas son una estructura de datos muy versátil y ampliamente utilizada. Una lista es una colección ordenada de elementos, donde cada elemento puede ser de cualquier tipo de dato, como enteros, cadenas de texto, booleanos e incluso otras listas.

Para crear una lista en Python, simplemente se encierran los elementos entre corchetes y se separan por comas. Por ejemplo:

```
mi_lista = [1, 2, 3, "Hola", True]
```

Una vez que se ha creado una lista, se puede acceder a sus elementos utilizando índices. El primer elemento de la lista tiene un índice de 0, el segundo tiene un índice de 1, y así sucesivamente. También se pueden utilizar índices negativos para acceder a los elementos desde el final de la lista. Por ejemplo:

```
mi_lista = [1, 2, 3, "Hola", True]
print(mi_lista[0]) # Imprime 1
print(mi_lista[-1]) # Imprime True
```

Las listas en Python son mutables, lo que significa que se pueden modificar una vez creadas. Se pueden agregar elementos a una lista utilizando el método `append()`. Por ejemplo:

```
mi_lista = [1, 2, 3]
mi_lista.append(4)
print(mi_lista) # Imprime [1, 2, 3, 4]
```

También se pueden modificar los elementos existentes de una lista asignando un nuevo valor a un índice específico. Por ejemplo:

```
mi_lista = [1, 2, 3]
mi_lista[1] = "Hola"
print(mi_lista) # Imprime [1, "Hola", 3]
```

Además de agregar y modificar elementos, las listas en Python también ofrecen una variedad de métodos útiles, como `insert()`, `remove()`, `pop()`, `sort()`, entre otros. Estos métodos permiten realizar operaciones como insertar un elemento en una

posición específica, eliminar un elemento por su valor, eliminar el último elemento de la lista, ordenar los elementos de la lista, entre otros.

En resumen, las listas en Python son una herramienta poderosa y flexible para almacenar y manipular colecciones de elementos. Su capacidad de ser modificadas y su amplia gama de métodos las convierten en una opción popular para trabajar con datos en Python.