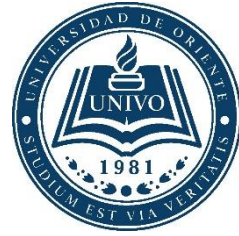


Universidad de Oriente

Facultad de ingeniería y Arquitectura

Instructor: Cristian Alexander Rivera Romero




Guía Practica 2 || Listas

Miercoles 22 de agosto de 2024

Introducción

Las listas en Python son una estructura de datos fundamental que te permite almacenar colecciones ordenadas de elementos en una sola variable. A diferencia de otros lenguajes de programación, las listas en Python pueden contener elementos de diferentes tipos, como números, cadenas de texto e incluso otras listas. Además, las listas son mutables, lo que significa que puedes modificar sus elementos después de haberlas creado.

¿Como se crea una lista?




```
1 mi_lista = [1, "Hola", 3.14, True]
2
```

Primero se le asigna un nombre puede ser cualquiera en este caso se le puso `mi_lista`, luego la parte importante es ponerle los corchetes estos:

[]


Como ver el contenido de la lista

Los elementos de una lista se acceden usando índices, comenzando desde 0 para el primer elemento.



```
1 mi_lista = [10, 20, 30, 40, 50]
2 print(mi_lista[0]) # Accede al primer elemento: 10
3 print(mi_lista[2]) # Accede al tercer elemento: 30
4 print(mi_lista[-1]) # Accede al último elemento: 50
5
```

Cuando quieran mostrar un elemento de la lista solo ponen entre corchetes el índice del ítem digamos si quisiera mostrar el **20** escribiría el siguiente comando

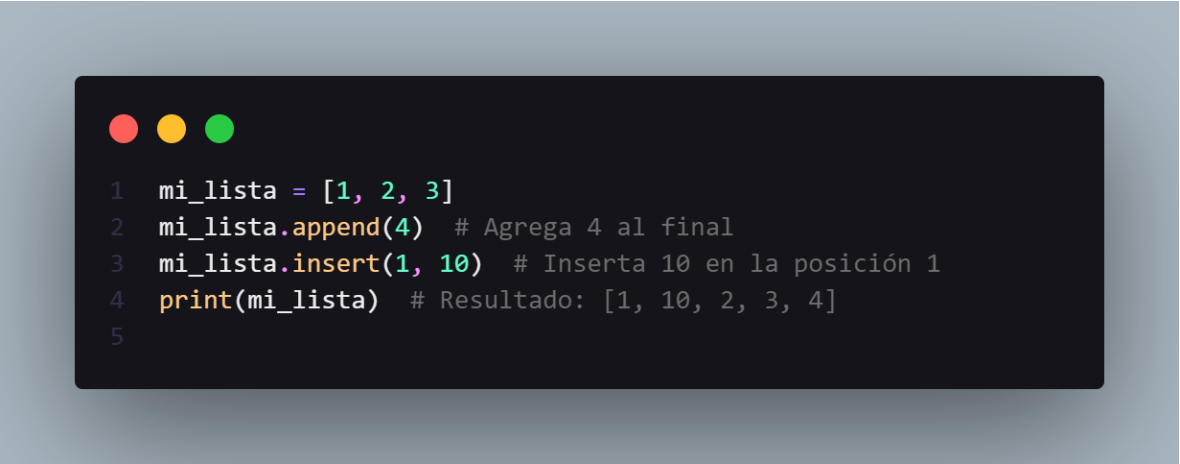


```
1 print(mi_lista[1])
```

¿Ahora como agregamos elementos?

`append()`: Agrega un elemento al final de la lista.

`insert()`: Inserta un elemento en una posición específica.

A terminal window with a dark background and three colored window control buttons (red, yellow, green) in the top left corner. It contains five lines of Python code with line numbers 1 through 5 on the left. The code demonstrates the use of `append()` and `insert()` methods on a list.

```
1 mi_lista = [1, 2, 3]
2 mi_lista.append(4) # Agrega 4 al final
3 mi_lista.insert(1, 10) # Inserta 10 en la posición 1
4 print(mi_lista) # Resultado: [1, 10, 2, 3, 4]
5
```

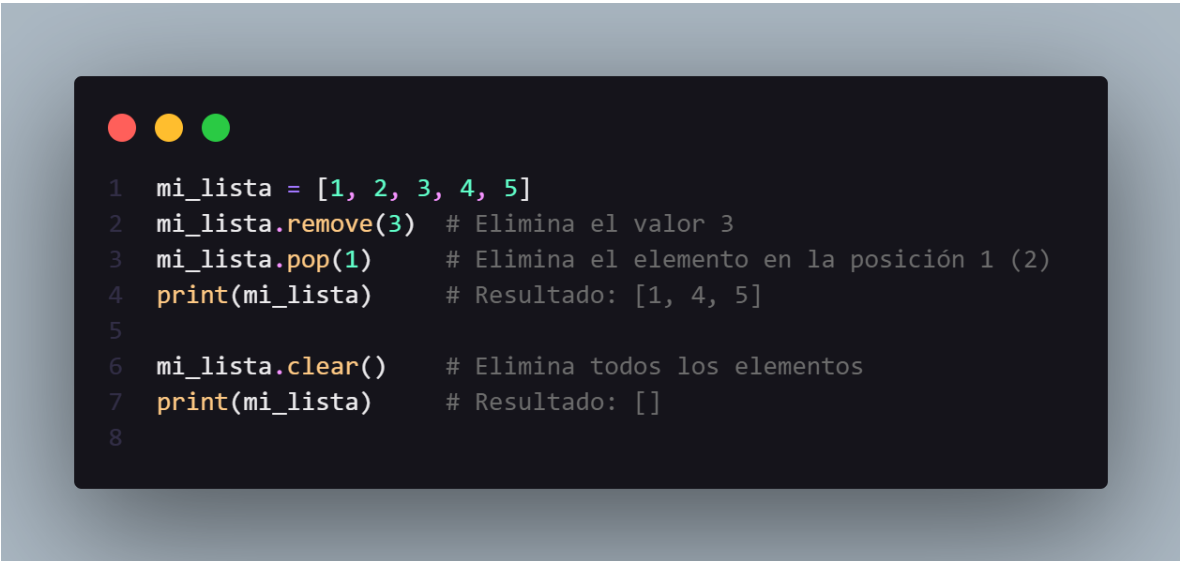
En palabras mas sencillas, lo que hace `append` agrega hasta el final como se ve que el **4** se agrega hasta el final de toda la lista y en cambio **`insert`** como dice inserta en una parte que se le diga en este caso sería en la posición 1 recordemos que el uno, esta en la segunda posición, así que por eso el 10 esta en la segunda posición al final

¿Y ahora como eliminamos?

`remove()`: Elimina la primera aparición de un valor específico.

`pop()`: Elimina un elemento por su índice y lo devuelve.

`clear()`: Elimina todos los elementos de la lista

A terminal window with a dark background and three colored window control buttons (red, yellow, green) at the top left. It contains Python code for list manipulation.

```
1 mi_lista = [1, 2, 3, 4, 5]
2 mi_lista.remove(3) # Elimina el valor 3
3 mi_lista.pop(1)    # Elimina el elemento en la posición 1 (2)
4 print(mi_lista)    # Resultado: [1, 4, 5]
5
6 mi_lista.clear()    # Elimina todos los elementos
7 print(mi_lista)    # Resultado: []
8
```

Remove elimina la primera vez que encuentra 3 o sea si hay mas 3 no los eliminaría,

Y **pop** se le da el índice del elemento y lo elimina y pues **clear** borra todo lo que este en la lista

Operaciones Comunes

Algunas operaciones comunes que se pueden realizar en listas incluyen:

- Obtener la longitud de una lista: **len(lista)**
- Contar ocurrencias de un elemento: **lista.count(elemento)**
- Encontrar el índice de un elemento: **lista.index(elemento)**

```
1 mi_lista = [1, 2, 3, 2, 5]
2 print(len(mi_lista))      # Resultado: 5
3 print(mi_lista.count(2))  # Resultado: 2 (dos ocurrencias de 2)
4 print(mi_lista.index(3))  # Resultado: 2 (el índice del valor 3)
5
```

Len, nos sirve para saber cuantos elementos tiene la lista en este caso muestra que son 5 elementos, y el **count** nos sirve para contar cuantas veces se repite un elemento en este caso cuenta el 2, que lo cuenta 2 veces también, el **index** sirve para saber el índice de elemento que se le pide

Y ahora si queremos saber que hay en las filas

```
1 mi_lista = ["a", "b", "c", "d"]
2 for elemento in mi_lista:
3     print(elemento)
4
```

Solo usamos un for que recorra cada elemento en la lista y fácil tenemos como mostrar lo que tengamos en la lista

Ejercicios

Crear y Acceder a Elementos

- Crea una lista llamada colores que contenga tres colores de tu elección. Luego, imprime el primer y el último color de la lista.

Modificar Elementos

- Crea una lista llamada numeros con los valores [10, 20, 30, 40, 50]. Cambia el tercer valor de la lista por 35 y luego imprime la lista modificada.

Agregar Elementos

- Crea una lista vacía llamada animales. Luego, agrega tres animales a la lista usando el método append() y uno más en la segunda posición usando insert(). Finalmente, imprime la lista completa.

Eliminar Elementos

- Dada la lista frutas = ["manzana", "banana", "naranja", "mango", "pera"], elimina "naranja" de la lista usando remove(), y luego elimina el último elemento usando pop(). Imprime la lista después de cada eliminación.