

Programación 9163

Listas

Prof. Antonio Carrillo Ledesma

Ayudantes:

- Diego Alberto Magallanes Ramírez
- Adriana Esmeralda Corona Mundo
- Orlando Alain Apipilhuasco Rosas

Contenido

- Introducción
- Crear una lista
- Mostrar una lista
- Conocer la cantidad de elementos dentro de una lista
- Acceder a los elementos dentro de la lista
- Recorrer una lista
- Modificar los elementos de una lista
- Verificar si un elemento está dentro de la lista
- Agregar un elemento a una lista
- Insertar elementos en una lista
- Eliminar elementos de una lista
- Eliminar elementos de una lista por índice
- Limpiar una lista
- Realizar una copia
- Unir listas
- Contar elementos dentro de una lista
- Encontrar el índice de un elemento
- Invertir una lista
- Ordenar una lista

Introducción

Las **Listas** forman parte de lo que se conoce como estructuras de datos que nos ofrece Python (en otros lenguajes se conocen como arreglos o vectores). Y es una estructura de datos formada por una secuencia ordenada de objetos.

Las listas en Python son utilizadas para almacenar múltiples valores en una única variable. Por ejemplo, almacenar los nombres de las personas ingresadas en tu programa. En este caso, las listas nos permiten solucionar este tipo de problemas, pues en una misma lista podremos almacenar todos esos valores, sin importar cuántos sean.

Las listas en Python son:

- Heterogéneas:** Pueden estar conformadas por elementos de distintos tipo, incluidos otras listas.
- Mutables:** Sus elementos pueden modificarse.

Crear una lista

En Python podemos crear listas de dos formas:

- Haciendo uso de la función incorporada de lista:

```
mLista = list() # Esto es una lista vacía, no tiene elementos en la lista.
```
- Haciendo uso de corchetes []

```
mLista = [] # Esto es una lista vacía, no tiene elementos en la lista.
```

Trabajaremos con las siguientes listas:

```
In [1]: frutas = ['manzana', 'naranja', 'mango', 'pera', 'platano', 'mandarina'] # Lista de frutas
vegetales = ['Tomate', 'Papa', 'Col', 'Cebolla', 'Zanahoria'] # Lista de vegetales
productos = ['leche', 'carne', 'mantequilla', 'yogurt'] # Lista de productos
países = ['Finlandia', 'Estonia', 'Dinamarca', 'Suecia', 'Noruega'] # Lista de países
```

Mostrar una lista

Para mostrar una lista en pantalla basta con pasar la lista a la función **print()**.

```
In [2]: print('Frutas: ', frutas)
print('Vegetales: ', vegetales)
print('Productos: ', productos)
print('Países: ', países)

Frutas: ['manzana', 'naranja', 'mango', 'pera', 'platano', 'mandarina']
Vegetales: ['Tomate', 'Papa', 'Col', 'Cebolla', 'Zanahoria']
Productos: ['leche', 'carne', 'mantequilla', 'yogurt']
Países: ['Finlandia', 'Estonia', 'Dinamarca', 'Suecia', 'Noruega']
```

Conocer la cantidad de elementos dentro de una lista

Podemos conocer el número de elementos dentro de una lista haciendo uso de la función integrada **len()**.

```
In [3]: print('Total de frutas: ', len(frutas))
print('Total de vegetales: ', len(vegetales))
print('Total de productos: ', len(productos))
print('Total de países: ', len(países))

Total de frutas: 6
Total de vegetales: 5
Total de productos: 4
Total de países: 5
```

Acceder a los elementos dentro de la lista

Accedemos a cada elemento de una lista utilizando su índice. El índice de una lista comienza desde 0.



```
In [4]: primera_fruta = frutas[0] # Obtenemos el primer elemento de la lista
print(primera_fruta)

segunda_fruta = frutas[1] # Obtenemos el segundo elemento de la lista
print(segunda_fruta)

ultima_fruta = frutas[5] # Obtenemos el último elemento de la lista
print(ultima_fruta)

ultimo_indice = len(frutas) - 1 # Último índice
ultima_fruta = frutas[ultimo_indice]
```

manzana
naranja
mandarina

Python permite acceder a los elementos de una lista usando indexación negativa, es decir:



```
In [5]: primera_fruta = frutas[-4]
ultima_fruta = frutas[-1]
segunda_fruta = frutas[-2]

print(primera_fruta)
print(ultima_fruta)
print(segunda_fruta)
```

mango
mandarina
platano

Recorrer una lista

Con la ayuda de un ciclo for es posible recorrer toda la lista o sólo una parte de ella.

```
In [6]: # Recorre toda la lista de países.
for pais in países:
    print(pais)

Finlandia
Estonia
Dinamarca
Suecia
Noruega
```

```
In [7]: # Recorre desde el índice 1 hasta un índice antes del 4.
for pais in países[1:4]:
    print(pais)

Estonia
Dinamarca
Suecia
```

```
In [8]: # Recorre la lista de 2 en 2.
for pais in países[:2]:
    print(pais)

Finlandia
Dinamarca
Noruega
```

Modificar los elementos de una lista

Podemos modificar los elementos de una lista accediendo a su posición y asignarle un nuevo valor.

```
In [9]: frutas[0] = 'Durazno'
frutas[4] = 'Uva'
print(frutas)

['Durazno', 'naranja', 'mango', 'pera', 'Uva', 'mandarina']
```

Verificar si un elemento está dentro de la lista

Podemos utilizar el operador **in** para saber si un elemento ya está dentro de la lista.

```
In [10]: print('Hay Cebolla: ', 'Cebolla' in vegetales)
print('Hay apio: ', 'apio' in vegetales)

Hay Cebolla: True
Hay apio: False
```

Agregar un elemento a una lista

Para agregar un elemento al final de una lista, usamos el método **append()**.

```
In [11]: print('Países: ', países)
países.append('México')
print('Países: ', países)

Países: ['Finlandia', 'Estonia', 'Dinamarca', 'Suecia', 'Noruega']
Países: ['Finlandia', 'Estonia', 'Dinamarca', 'Suecia', 'Noruega', 'México']
```

Insertar elementos en una lista

Podemos usar el método **insert()** para insertar un solo elemento en un índice específico en una lista. Tenga en cuenta que otros elementos se desplazan a la derecha. El método insert() toma dos argumentos: índice y un elemento para insertar.

```
In [12]: países.insert(4, 'Japón')
países.insert(0, 'Australia')

print(países)

['Australia', 'Finlandia', 'Estonia', 'Dinamarca', 'Suecia', 'Japón', 'Noruega', 'México']
```

Eliminar elementos de una lista

El método **remove()** elimina un elemento específico de una lista.

```
In [13]: países.remove('Dinamarca')
print(países)

['Australia', 'Finlandia', 'Estonia', 'Suecia', 'Japón', 'Noruega', 'México']
```

Eliminar elementos de una lista por índice

El método **pop()** elimina el índice especificado (o el último elemento si no se especifica el índice).

```
In [14]: países.pop(0)
print(países)

países.pop()
print(países)

['Finlandia', 'Estonia', 'Suecia', 'Japón', 'Noruega', 'México']
['Finlandia', 'Estonia', 'Suecia', 'Japón', 'Noruega']
```

Limpiar una lista

El método **clear()** vacía la lista, es decir, elimina todos los elementos de una lista.

```
In [15]: países.clear()
print(países)

[]
```

Realizar una copia

Es posible copiar una lista reasignándola a una nueva variable de la siguiente forma: `lista_2 = lista_1`. Ahora, `lista_2` es una referencia de `lista_1`, cualquier cambio que hagamos en `lista_2` también modificará el original. Pero hay muchos casos en los que no nos gusta modificar el original sino que nos gusta tener una copia diferente. Una forma de evitar el problema anterior es usar el método **copy()**.

```
In [16]: copia_vegetales = vegetales.copy()

print(vegetales)
print(copia_vegetales)

['Tomate', 'Papa', 'Col', 'Cebolla', 'Zanahoria']
['Tomate', 'Papa', 'Col', 'Cebolla', 'Zanahoria']
```

Unir listas

Hay varias formas de unir o concatenar dos o más listas en Python.

- Operador más (+)

```
In [17]: nuevos_vegetales = ['Jitomate', 'Lechuga', 'Rabano']
vegetales = vegetales + nuevos_vegetales

print(vegetales)

['Tomate', 'Papa', 'Col', 'Cebolla', 'Zanahoria', 'Jitomate', 'Lechuga', 'Rabano']
```

- Unir usando el método **extend()**. El método extend() permite agregar los elementos de una lista a otra.

```
In [18]: mas_vegetales = ['Espinaca', 'Apio']
vegetales.extend(mas_vegetales)

print(vegetales)

['Tomate', 'Papa', 'Col', 'Cebolla', 'Zanahoria', 'Jitomate', 'Lechuga', 'Rabano', 'Espinaca', 'Apio']
```

Contar elementos dentro de una lista

El método **count()** devuelve el número de veces que aparece un elemento en una lista:

```
In [19]: numeros = [1, 2, 2, 2, 3, 3, 4, 4, 4, 4, 5, 5, 5, 5, 5]

cuenta = numeros.count(3)

print('¿Cuántos 3 hay dentro de la lista? ', cuenta)

¿Cuántos 3 hay dentro de la lista? 3
```

Encontrar el índice de un elemento

El método **index()** devuelve el índice de un elemento (primera aparición) dentro de la lista.

```
In [20]: índice = numeros.index(2)

print('El numero 2 se encuentra en la posición: ', índice)

El numero 2 se encuentra en la posición: 1
```

Invertir una lista

El método **reverse()** invierte el orden de una lista.

```
In [21]: numeros.reverse()

print(numeros)

[5, 5, 5, 5, 4, 4, 4, 4, 4, 3, 3, 3, 2, 2, 1]
```

Ordenar una lista

El método **sorted()** reordena los elementos de la lista en orden ascendente y modifica la lista original. El método sorted() se encuentra dentro de las funciones integradas de Python.

```
In [22]: print('Listas desordenadas:\n')
print('Frutas: ', frutas)
print('Vegetales: ', vegetales)
print('Productos: ', productos)
print('Países: ', países)
print('Números:', numeros)

print('\nListas ordenadas:\n')
print('Frutas: ', sorted(frutas))
print('Vegetales: ', sorted(vegetales))
print('Productos: ', sorted(productos))
print('Países: ', sorted(países))
print('Números:', sorted(numeros))

Listas desordenadas:

Frutas: ['Durazno', 'naranja', 'mango', 'pera', 'Uva', 'mandarina']
Vegetales: ['Tomate', 'Papa', 'Col', 'Cebolla', 'Zanahoria', 'Jitomate', 'Lechuga', 'Rabano', 'Espinaca', 'Apio']
Productos: ['leche', 'carne', 'mantequilla', 'yogurt']
Países: []
Números: [5, 5, 5, 5, 4, 4, 4, 4, 3, 3, 3, 2, 2, 1]

Listas ordenadas:

Frutas: ['Durazno', 'Uva', 'mandarina', 'mango', 'naranja', 'pera']
Vegetales: ['Apio', 'Cebolla', 'Col', 'Espinaca', 'Jitomate', 'Lechuga', 'Rabano', 'Rabano', 'Tomate', 'Zanahoria']
Productos: ['carne', 'leche', 'mantequilla', 'yogurt']
Países: []
Números: [1, 2, 2, 3, 3, 3, 4, 4, 4, 4, 5, 5, 5, 5, 5]
```