

Listas

Semestre 02, 2025

Definición

Una lista es una estructura que puede almacenar una colección de elementos. Una lista es ordenada y cada elemento dentro de sí tiene un índice asociado.

A diferencia de las cadenas (sub tipo de listas) las listas si son mutables. Es posible modificar un elemento específico de la lista.

Las listas usan la notación de corchetes para definirse `[]`.

```
1  lista = [1, 2, 3]
2
3
4  lista_vacia = []
5
6
7
8  print(lista)
9
10 print(lista_vacia)
11
```

Una lista puede ser `homogénea` si todos los elementos que contiene tienen el mismo tipo de datos. Una lista es `heterogénea` si sus elementos contienen varios tipos de datos.

```
1
2  homogenea1 = [1, 2, 3]
3
4  homogenea2 = ["a", "e", "i", "o", "u"]
5
6  homogenea3 = [3.14, 3.15, 3.16]
7
8  homogenea4 = [True, False]
9
```

```
10  
11  
12 heterogena = [1, "a", 3.14, True]  
13
```

Funciones disponibles

<https://docs.python.org/3/tutorial/datastructures.html#more-on-lists>

Las listas también tienen varias funciones que se pueden aprovechar:

len()

Retorna la cantidad de elementos de una lista.

```
1  
2 lista = [2, 2, 3, 4, 5, 6]  
3  
4  
5  
6 largo = len(lista)  
7  
8  
9  
10 print(largo) # 6  
11
```

Suma de listas (+)

Concatena dos listas en una sola.

```
1  
2 lista = [2, 2, 3, 4, 5, 6]  
3  
4 lista2 = [9, 8, 7]  
5  
6  
7
```

```
8  print(lista + lista2)
9
10 # [2, 2, 3, 4, 5, 6, 9, 8, 7]
11
```

Multiplicación de listas (*)

Repite los elementos de una lista las veces indicadas.

```
1
2  lista = [1, 2]
3
4
5
6  print(lista * 3)
7
8  # [1, 2, 1, 2, 1, 2]
9
```

Modificar un elemento por índice

Permite cambiar el valor de un elemento en una posición específica.

```
1
2  lista = [2, 2, 3, 4]
3
4
5
6  lista[0] = 1
7
8
9
10 print(lista) # [1, 2, 3, 4]
11
```

Slice de lista (lista[inicio:fin])

Crea una sublista desde la posición `inicio` hasta `fin - 1`.

```
1
2     lista = [2, 2, 3, 4, 5, 6]
3
4
5
6     sub_lista = lista[1:3]
7
8
9
10    print(sub_lista) # [2, 3]
11
```

Asignar con slice

Permite reemplazar varios elementos de una lista a la vez.

```
1
2     lista = [1, 2, 3, 4, 5]
3
4
5
6     lista[1:4] = [8, 9, 10]
7
8
9
10    print(lista) # [1, 8, 9, 10, 5]
11
```

.append()

Agrega un elemento al final de la lista.

```
1
2     lista = [1, 2, 3]
3
4
5
6     lista.append(4)
7
8
```

```
9  
10 print(lista) # [1, 2, 3, 4]  
11
```

.insert()

Inserta un elemento en una posición específica.

```
1  
2 lista = [1, 2, 3]  
3  
4  
5  
6 lista.insert(1, 99)  
7  
8  
9  
10 print(lista) # [1, 99, 2, 3]  
11
```

.remove()

Elimina la **primera aparición** de un valor en la lista.

```
1  
2 lista = [1, 2, 3, 2, 4]  
3  
4  
5  
6 lista.remove(2)  
7  
8  
9  
10 print(lista) # [1, 3, 2, 4]  
11
```

del

Elimina un elemento por su índice.

```
1
2  lista = [10, 20, 30]
3
4
5
6  del lista[0]
7
8
9
10 print(lista) # [20, 30]
11
```

.pop()

Elimina y retorna el último elemento de la lista (o el índice especificado).

```
1
2  lista = [1, 2, 3]
3
4
5
6  valor = lista.pop()
7
8
9
10 print(valor) # 3
11
12 print(lista) # [1, 2]
13
```

.clear()

Elimina todos los elementos de la lista.

```
1
2  lista = [1, 2, 3]
3
```

```
4
5
6 lista.clear()
7
8
9
10 print(lista) # []
11
```

.sort()

Ordena los elementos de la lista en orden ascendente (numérico o alfabético).

```
1
2 lista = [5, 1, 4, 3]
3
4
5
6 lista.sort()
7
8
9
10 print(lista) # [1, 3, 4, 5]
11
```

.sort(reverse=True)

Ordena los elementos en orden descendente.

```
1
2 lista = [5, 1, 4, 3]
3
4
5
6 lista.sort(reverse=True)
7
8
9
10 print(lista) # [5, 4, 3, 1]
```

.extend()

Agrega todos los elementos de otra lista al final (diferente de `append`, que agrega la lista como un único elemento).

```
1
2  lista = [1, 2, 3]
3
4  lista2 = [4, 5]
5
6
7
8  lista.extend(lista2)
9
10
11
12  print(lista) # [1, 2, 3, 4, 5]
13
```

list() a partir de cadena

Convierte una cadena en lista de caracteres.

```
1
2  cadena = "Hello"
3
4
5
6  nueva_lista = list(cadena)
7
8
9
10 print(nueva_lista) # ['H', 'e', 'l', 'l', 'o']
11
```

.index()

Devuelve la posición (índice) de la **primera aparición** de un elemento.

```
1
2 lista = [10, 20, 30, 20]
3
4
5
6 print(lista.index(20)) # 1
7
```

.split()

Divide una cadena en lista, usando un separador.

```
1
2 cadena = "Hola Mundo Python"
3
4
5
6 nueva_lista = cadena.split(" ")
7
8
9
10 print(nueva_lista) # ['Hola', 'Mundo', 'Python']
11
```

.join()

Une los elementos de una lista en un string, separados por un delimitador.

```
1
2 lista = ["Hola", "Mundo"]
3
4
5
6 nuevo_string = " ".join(lista)
7
8
```

```
9  
10 print(nuevo_string) # "Hola Mundo"  
11
```

Recorrer listas

Es posible recorrer todos los elementos de una lista usando cualquier de ambos ciclos:

for:

```
1  
2 lista = ["uno", "dos", "tres"]  
3  
4  
5  
6 for elemento in lista:  
7  
8 print(elemento)  
9
```

while:

```
1  
2 lista = ["uno", "dos", "tres"]  
3  
4 indice = 0  
5  
6  
7  
8 while indice < len(lista):  
9  
10 print(lista[indice])  
11  
12 indice += 1  
13
```