
COLECCIONES O MÉTODOS

❖ LISTAS

Hay cuatro tipos de colecciones expandibles dinámicamente: **Listas, Tuplas, Diccionarios y Conjuntos.**

- **Listas: Una lista es una secuencia**

Así como una cadena, una *lista* es una secuencia de valores. En una cadena, los valores son caracteres; en una lista, pueden ser cualquier tipo.

Los valores en una lista son llamados *elementos* o a veces *ítems*

- Pueden ser de cualquier tipo
- Están indexadas... es decir tienen orden (como los vectores)

Indexación negativa

La indexación negativa significa comenzar desde el final

-1 se refiere al último elemento, -2 se refiere al penúltimo elemento, etc.

- Son modificables
- Pueden estar duplicados
- Definición:
Nombre_Lista = [elem1, elem2, elem3, ...]
Arrancan la posición 0

Hay varias formas de crear una nueva lista; la más simple es encerrar los elementos en corchetes ("[" y "]"):

[10, 20, 30, 40] (ejemplo es una lista de 4 enteros)

['rana crujiente', 'pulpo a la española, 'matambre de cerdo'] (una lista de tres cadenas.)

Los elementos de una lista no tienen que ser del mismo tipo. La siguiente lista contiene una cadena, un flotante, un entero, y (¡mira!) otra lista:

['spam', 2.0, 5, [10, 20]]

- Una lista dentro de otra lista está *anidada*.
- Una lista que no contiene elementos es llamada una lista vacía; se puede crear una con corchetes vacíos, [].

A diferencia de las cadenas, las listas son mutables porque pueden cambiar el orden de los elementos o reasignar un elemento en una lista.

Cuando el operador corchete aparece en el lado izquierdo de una asignación, éste identifica el elemento de la lista que será asignado

RANGO DE ÍNDICES

Se puede especificar un rango de índices indicando dónde comenzar y dónde terminar el mismo.

Al especificar un rango, el valor de retorno será una nueva lista con los elementos especificados.

```
mi_lista = ["manzana", " banana ", "cereza", "naranja", "kiwi", "melón", "mango"]  
print (mi_lista [2: 5])
```

Esto devolverá los artículos de la posición 2 a la 5.

['cereza', 'naranja', 'kiwi']

Recuerde que el primer elemento es la posición 0, y tenga en cuenta que el artículo en la posición 5 **NO** está incluido

```
mi_lista = ["manzana", " banana ", "cereza", "naranja", "kiwi", "melón", "mango"]  
print (mi_lista [: 4])
```

Esto devolverá los elementos del índice 0 al índice 4.

['manzana', 'plátano', 'cereza', 'naranja']

Recuerde que el índice 0 es el primer elemento y el índice 4 es el quinto elemento
Recuerde que el ítem en el índice 4 **NO** está incluido

```
mi_lista = ["manzana", " banana ", "cereza", "naranja", "kiwi", "melón", "mango"]  
print (mi_lista [2: ])
```

Esto devolverá los elementos del índice 2 al final.

["cereza", "naranja", "kiwi", "melón", "mango"]

Recuerde que el índice 0 es el primer elemento y el índice 2 es el tercero

```
mi_lista = ["manzana", " banana ", "cereza", "naranja", "kiwi", "melón", "mango"]  
print (mi_lista [-4:-1 ])
```

La indexación negativa significa comenzar desde el final de la lista.

["naranja", "kiwi", "melón"]

Este ejemplo devuelve los elementos del índice -4 (incluido) al índice -1 (excluido)
Recuerda que el último ítem tiene el índice -1,

CAMBIAR ELEMENTOS DE LA LISTA

Para cambiar el valor de un artículo específico, consulte el número de índice:

```
mi_lista2 = ["manzana", "banana", "cereza"]
mi_lista2[2]= "pera"
print (mi_lista2)
```

Devolverá:

```
["manzana", "banana", "pera"]
```

```
mi_lista2 = ["manzana", "banana", "cereza"]
mi_lista2[1:2]= ["pera", "ananá"]
print (mi_lista2)
```

Devolverá:

```
["manzana", "pera", "ananá", "cereza"]
```

Si inserta *más* elementos de los que reemplaza, los elementos nuevos se insertarán donde haya especificado y los elementos restantes se moverán en consecuencia

La longitud de la lista cambiará cuando el número de elementos insertados no coincida con el número de elementos reemplazados.

Si inserta *menos* elementos de los que reemplaza, los elementos nuevos se insertarán donde haya especificado y los elementos restantes se moverán en consecuencia

```
mi_lista2 = ["manzana", "banana", "cereza"]
mi_lista2[1:3]= ["ananá"]
print (mi_lista2)
```

Devolverá:

```
["manzana", "ananá"]
```

INSERTAR ELEMENTOS

Para insertar un nuevo elemento de lista, sin reemplazar ninguno de los valores existentes, podemos usar el método **insert()**.

El método **insert()** inserta un elemento en el índice especificado:

```
mi_lista2 = ["manzana", "banana", "cereza"]
mi_lista2.insert(2,"ananá")
print (mi_lista2)
```

Devolverá:

```
["manzana", "banana", "ananá", "cereza"]
```

AGREGAR ELEMENTOS

Para agregar un elemento al final de la lista, use el método **append ()**:

```
mi_lista2 = ["manzana", "banana", "cereza"]
mi_lista.append("ananá")
print (mi_lista2)
```

Devolverá:

```
["manzana", "banana", "cereza", "ananá"]
```

AMPLIAR LISTA

Para agregar elementos de *otra lista* a la lista actual, use el método **extend()**.

```
mi_lista_diasLaborales = ["lunes", "martes", "miércoles", "jueves", "viernes"]
Finde = ["sábado", "domingo"]
mi_lista_diasLaborales.extend(Finde)
print(mi_lista_diasLaborales)
```

Devolverá:

```
["lunes", "martes", "miércoles", "jueves", "viernes", "sábado", "domingo"]
```

ELIMINAR ARTÍCULO ESPECIFICADO

El método **remove()** elimina el elemento especificado

```
lista_colores = ["azul", "blanco", "amarillo", "celeste", "verde"]
lista_colores.remove("verde")
print(lista_colores)
```

Devolverá:

```
["azul", "blanco", "amarillo", "celeste"]
```

ELIMINAR ÍNDICE ESPECIFICADO

El método **pop()** elimina el índice especificado

```
ClubDeFutbol = ["Gimnasia", "Boca", "Racing", "Tigre", "Newells"]
ClubDeFutbol.pop (4)
print(ClubDeFutbol)
```

Devolverá:

```
["Gimnasia", "Boca", "Racing", "Tigre"]
```

Si no especifica el índice, el pop() método elimina el último elemento.
La palabra del clave también elimina el índice especificado:

```
L1 = ['a', 'b', 'c', 'd']  
del L1[2]  
print(L1)  
Devolverá:  
['a', 'b', 'd']
```

La palabra **del** clave también puede eliminar la lista por completo.

```
L2 = ["Leo", "Escorpio", "Tauro"]  
del L2  
print(L2)
```

Esto provocará un error porque se eliminó **L2** con éxito. Me va a decir que **L2**, no está definida

OTROS MÉTODOS DE LISTAS

Python tiene un conjunto de métodos integrados que puede usar en listas.

Método	Descripción
<code>append()</code>	Agrega un elemento al final de la lista.
<code>clear()</code>	Elimina todos los elementos de la lista.
<code>copy()</code>	Devuelve una copia de la lista.
<code>count()</code>	Devuelve el número de elementos con el valor especificado.
<code>extend()</code>	Agregue los elementos de una lista (o cualquier iterable), al final de la lista actual
<code>index()</code>	Devuelve el índice del primer elemento con el valor especificado.
<code>insert()</code>	Agrega un elemento en la posición especificada
<code>pop()</code>	Elimina el elemento en la posición especificada
<code>remove()</code>	Elimina el artículo con el valor especificado
<code>reverse()</code>	Invierte el orden de la lista
<code>sort()</code>	Ordena la lista

RECORRER UNA LISTA

```
números=[3,6,9,12,15]
print(números)
for i in range(len(números)):
    números[i] = números[i] * 2
    print(números)
```

Devolverá

```
[3, 6, 9, 12, 15]
[6, 12, 18, 24, 30]
```

Un bucle **for** a través de una lista vacía nunca ejecuta el código contenido en el cuerpo:

LISTAS Y FUNCIONES

Hay otras funciones que se pueden utilizar en las listas que permiten mirar rápidamente a través de una lista sin escribir los bucles:

```
números = [3, 41, 12, 9, 74, 15]

print(len(números))
6
print(max(números))
74
print(min(números))
3
print(sum(números))
154
print(sum(números)/len(números))
25
```

La función **sum()** **solamente funciona cuando los elementos de la lista son números**. Las funciones (`max()`, `len()`, etc.) funcionan con listas de cadenas y otros tipos que pueden ser comparados entre sí.

Ejemplo: Escribimos un programa que calcule el promedio de una lista de números ingresados por el usuario, utilizando una lista.

- Primero, el programa para calcular un promedio sin una lista:
-

```
total = 0
contador = 0
while (True):
    aux = input('Ingresa un número: ')
    if aux == 'x': break
    valor = float(aux)
    total = total + valor
    contador = contador + 1

promedio = total / contador
print('Promedio:', promedio)
```

- Cálculo del mismo promedio utilizando una lista

```
numlista = list()
while (True):
    aux = input('Ingresa un número: ')
    if aux == 'x': break
    valor = float(aux)
    numlista.append(valor)

promedio = sum(numlista) / len(numlista)
print('Promedio:', promedio)
```
