

Programación con python

List

List

En Python, una lista es un tipo de dato que almacena una colección ordenada de elementos. Puedes acceder a elementos individuales de una lista usando sus índices y añadir elementos a la lista utilizando el método `.append(item)`.

ejemplo:

```
lst = [1, 10, 4, True, "str"]
lst.append(2)      # lst ahora es igual a [1, 10, 4, True, "str", 2]
print(lst[1])     # esto imprime 10
print(lst[-1])    # esto imprime 2
```

Palabra clave in

La palabra clave `in` en Python te permite comprobar si un valor está contenido en una colección (como una lista, set, diccionario, etc).

```
print("hello" in ["hello", "world"]) # True
```

La palabra clave `in` también se puede usar para iterar sobre los elementos de una colección al usar un ciclo `for`.

Lista vacía

```
lista_vacia = []
print("Lista vacía:", lista_vacia) # []
```

Lista de números

```
numeros = [10, 20, 30, 40, 50]  
print("Lista de números:", numeros)
```

Lista de strings

```
nombres = ["Ana", "Luis", "María", "Carlos"]  
print("Lista de strings:", nombres)
```

Lista combinada

```
combinada = [25, "Hola", True, 3.14]  
print("Lista combinada:", combinada)
```

Lista dentro de listas

```
listas = [[1, 2, 3], ["a", "b", "c"], [True, False]]  
print("Lista dentro de listas:", listas)
```

Acceso a elementos por índice

```
colores = ["rojo", "verde", "azul", "amarillo"]  
print("Primer color:", colores[0]) # rojo  
print("Tercer color:", colores[2]) # azul
```

Longitud de la lista con len()

```
frutas = ["manzana", "plátano", "naranja"]  
print("Número de frutas:", len(frutas)) # 3
```

len() en strings

```
palabra = "Python"  
print("Número de caracteres en 'Python':", len(palabra)) # 6
```

Cambiar elemento en la lista

```
animales = ["perro", "gato", "conejo"]
animales[1] = "tigre"
print("Lista modificada:", animales)
```

Error al acceder a un índice inexistente

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

```
print(numeros[5]) # IndexError: list index out of range
```

Agregar elementos con append()

```
alumnos = ["Ana", "Luis"]
alumnos.append("María")
print("Lista con append:", alumnos)
```

pop(): eliminar y devolver un elemento

```
numeros = [10, 20, 30, 40, 50]
print("Lista original:", numeros)
```

```
# Elimina el último elemento
ultimo = numeros.pop()
print("Elemento eliminado:", ultimo)      # 50
print("Lista después de pop():", numeros) # [10, 20, 30, 40]
```

count(): contar cuántas veces aparece un valor

```
letras = ["a", "b", "c", "a", "b", "a"]
print("Lista:", letras)
print("La 'a' aparece:", letras.count("a"), "veces") # 3
print("La 'b' aparece:", letras.count("b"), "veces") # 2
print("La 'z' aparece:", letras.count("z"), "veces") # 0
```

index(): obtener la posición de un elemento

```
colores = ["rojo", "verde", "azul", "amarillo", "verde"]
print("Lista de colores:", colores)
```

```
pos_verde = colores.index("verde")
print("Primera posición de 'verde':", pos_verde) # 1
```

```
# Si el elemento no existe, da error:
# print(colores.index("negro")) # ValueError: 'negro' is not in list
```

remove(): eliminar la primera aparición de un valor

```
animales = ["perro", "gato", "conejo", "gato"]
print("Lista original:", animales)
```

```
animales.remove("gato")
print("Después de remove('gato'):", animales) # ['perro', 'conejo',
'gato']
```

```
# Si el valor no existe, da error:
# animales.remove("tigre") # ValueError: list.remove(x): x not in list
```

Uso de índices negativos en listas

```
frutas = ["manzana", "plátano", "naranja", "pera", "uva"]
print("Lista de frutas:", frutas)
```

```
# Último elemento con índice -1
print("Último elemento (índice -1):", frutas[-1]) # uva
```

```
# Penúltimo elemento con índice -2
print("Penúltimo elemento (índice -2):", frutas[-2]) # pera
```

```
# Antepenúltimo elemento con índice -3
print("Antepenúltimo elemento (índice -3):", frutas[-3]) # naranja
```

Combinar listas con el operador +

```
# Dos listas diferentes
```

```
lista1 = [1, 2, 3]
```

```
lista2 = [4, 5, 6]
```

```
# Combinar usando +
```

```
lista_combinada = lista1 + lista2
```

```
print("Lista 1:", lista1)      # [1, 2, 3]
```

```
print("Lista 2:", lista2)      # [4, 5, 6]
```

```
print("Lista combinada:", lista_combinada) # [1, 2, 3, 4, 5, 6]
```

También funciona con listas de strings

```
nombres1 = ["Ana", "Luis"]
nombres2 = ["María", "Carlos"]
todos = nombres1 + nombres2
```

```
print("Nombres combinados:", todos) # ['Ana', 'Luis', 'María', 'Carlos']
```

Usar .extend() para combinar listas

```
# Dos listas
```

```
lista1 = [1, 2, 3]
```

```
lista2 = [4, 5, 6]
```

```
print("Lista 1 antes:", lista1) # [1, 2, 3]
```

```
print("Lista 2:", lista2)      # [4, 5, 6]
```

```
# Agregar los elementos de lista2 a lista1 con extend()
```

```
lista1.extend(lista2)
```

```
print("Lista 1 después de extend:", lista1) # [1, 2, 3, 4, 5, 6]
```

```
print("Lista 2 no cambia:", lista2)      # [4, 5, 6]
```

Diferencia entre + y extend()

```
# Con +
```

```
lista_a = [10, 20]
```

```
lista_b = [30, 40]
```

```
resultado = lista_a + lista_b
```

```
print("Con +:", resultado) # [10, 20, 30, 40]
```

```
print("Lista A sigue igual:", lista_a) # [10, 20]
```

```
# Con extend()
```

```
lista_a.extend(lista_b)
```

```
print("Con extend:", lista_a) # [10, 20, 30, 40]
```

Ejemplo de listas anidadas e índices negativos

```
lst = [[5, 6, []], [2, 3], [1, 2, 3]]
```

```
# Acceder al último elemento con índice -1 → [1, 2, 3]
# Luego acceder al índice 1 de esa lista → 2
print(lst[-1][1]) # Resultado: 2
```

Ejemplo de listas anidadas e índices negativos

```
lst = [[5, 6, []], [2, 3], [1, 2, 3]]
```

```
# Acceder al último elemento con índice -1 → [1, 2, 3]
# Luego acceder al índice 1 de esa lista → 2
print(lst[-1][1]) # Resultado: 2
```

Actividad

Escribe un programa que pida al usuario ingresar cinco cadenas de texto y las almacene en una lista.

Luego, el programa debe pedir al usuario ingresar tres números en el rango de 0 a 4 (incluidos). Estos números representarán índices de la lista.

Finalmente, el programa debe usar esos tres números para acceder a las tres cadenas en los índices correspondientes de la lista previamente creada, concatenarlas y mostrar la cadena resultante en pantalla.

NO USE INTELIGENCIA ARTIFICIAL PARA SU TRABAJO
ENTREGUELA EN LA SECCION DE TAREAS

Actividad

```
Enter a string: Professor
Enter a string: Luis
Enter a string: is
Enter a string: the
Enter a string: Best!
Enter a number (0-4): 1
Enter a number (0-4): 2
Enter a number (0-4): 3
Resultado final: Luisisthe
```

NO USE INTELIGENCIA ARTIFICIAL PARA SU TRABAJO
ENTREGUELA EN LA SECCION DE TAREAS

Ejercicio 2

Pide al usuario 5 nombres y guárdalos en una lista llamada nombres.

Muestra en pantalla:

El primer y el último nombre usando índices.

La longitud de la lista con len().

Pide un nombre a eliminar y quítalo de la lista con .remove().

Pide un nombre para agregar y añádelo al final con .append().

Pide un nombre a contar y muestra cuántas veces aparece en la lista con .count().

Pide dos índices (en el rango válido) y concatena los nombres en esas posiciones.
Imprime el resultado final.

NO USE INTELIGENCIA ARTIFICIAL PARA SU TRABAJO
ENTREGUELA EN LA SECCION DE TAREAS

Ejercicio 2

```
Ingresa el nombre 1: luis
Ingresa el nombre 2: francisco
Ingresa el nombre 3: fer
Ingresa el nombre 4: alex
Ingresa el nombre 5: pedro
Primer nombre: luis
Último nombre: pedro
Total de nombres: 5
Nombre a eliminar: pedro
Nombre a agregar: barbara
Lista actual: ['luis', 'francisco', 'fer', 'alex', 'barbara']
Nombre a contar: luis
El nombre aparece 1 veces.
Índice 1 (0 a 4): 0
Índice 2 (0 a 4): 1
Concatenación: luisfrancisco
```

NO USE INTELIGENCIA ARTIFICIAL PARA SU TRABAJO
ENTREGUELA EN LA SECCION DE TAREAS