

Estructuras de Datos (II) - list (Lista)

En Python, una **lista** es una estructura de datos que almacena una **colección ordenada** de elementos. **Cada elemento puede ser de cualquier tipo y se puede acceder a ellos mediante su índice.**

Para **crear una lista** vacía en Python puedes usar los corchetes `[]` o el constructor `list()`. Aquí tienes un ejemplo de ambas formas:

```
my_list = []  
my_other_list = list()
```

Ambas variables `my_list` y `my_other_list` son listas vacías.

Para **acceder a los elementos de una lista** en Python puedes **usar el índice del elemento entre corchetes**. Los índices **comienzan en 0** para el primer elemento **y aumentan en 1** para cada elemento siguiente. También puedes usar índices negativos para acceder a los elementos desde el final de la lista.

```
another_list = ['a', 'b', 'c', 'd']  
  
print(another_list[0]) # 'a'  
print(another_list[2]) # 'c'  
print(another_list[-1]) # 'd'
```

Puedes verificar si un elemento está presente (o no) en la lista utilizando el operador `in` y `not in`. Aquí tienes un ejemplo:

```
# Crear la lista  
mi_lista = ['manzana', 'naranja', 'plátano']  
# Verificar si un elemento está presente en la lista  
print('manzana' in mi_lista) # True  
print('pera' in mi_lista) # False  
# Verificar si un elemento NO está presente en la lista  
print('pera' not in mi_lista) # True  
print('manzana' not in mi_lista) # False
```

Algunos métodos comunes de las listas son:

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

```
my_list = [1, 2, 3]  
my_list.append(4)  
  
print(my_list) # [1, 2, 3, 4]
```

- **.extend(iterable)**: Agrega todos los elementos del iterable al final de la lista.

```
my_list.extend([5, 6])

print(my_list) # [1, 2, 3, 4, 5, 6]
```

- **.insert(i, x)**: Inserta un elemento en una posición dada.

```
my_list.insert(0, -1)

print(my_list) # [-1, 1, 2, 3, 4, 5, 6]
```

- **.remove(x)**: Elimina el primer elemento con el valor especificado.

```
my_list.remove(-1)

print(my_list) # [1, 2, 3, 4, 5, 6]
```

- **.pop([i])**: Elimina el elemento en la posición dada y lo devuelve. Si no se especifica una posición, elimina y devuelve el último elemento.

```
last_element = my_list.pop()

print(last_element) # 6
```

- **.index(x[, start[, end]])**: Devuelve el índice del primer elemento con el valor especificado. Puedes especificar dónde empezar y terminar la búsqueda.

```
second_list = [1, 2, 3, 2]
index_of_first_2 = second_list.index(2)

print(index_of_first_2) # 1
```

- **.count(x)**: Devuelve el número de veces que aparece el valor especificado.

```
number_of_2s = second_list.count(2)

print(number_of_2s) # 2
```

- **.sort(key=None, reverse=False)**: Ordena la lista. Puedes especificar una función de clave para personalizar el orden y si quieres ordenar en orden inverso.

```
second_list.sort(reverse=True)

print(second_list) # [3, 2, 2, 1]
```

- **.reverse()**: Invierte el orden de los elementos en la lista.

```
second_list.reverse()

print(second_list) # [1, 2, 2, 3]
```