Métodos de lista

Breve Explicacion

Además de las funciones integradas, Python tiene una serie de **métodos de lista integrados** que son muy útiles.

Éstos son algunos de ellos:

- **`.append()`** : El método agrega un elemento al final de la lista.
- **`.insert()`** : El método agrega un elemento a un índice específico.
- **`.remove()`** : El método elimina un elemento de una lista según el valor.
- **`.pop()`** : El método elimina el elemento en un índice particular.

¡Usemos secuencias de ADN como ejemplo para esto!

```
dna = ['AUG', 'AUC', 'UCG']

dna.append('UAA')     # ['AUG', 'AUC', 'UCG', 'UAA']
dna.insert(2, 'GAU')  # ['AUG', 'AUC', 'GAU', 'UCG', 'UAA']
dna.remove('AUC')     # ['AUG', 'GAU', 'UCG', 'UAA']
dna.pop(0)            # ['GAU', 'UCG', 'UAA']
```

La diferencia entre **funciones integradas** y **métodos** en una lista es que los métodos usan la sintaxis de notación de puntos en la variable de lista que creamos. Las funciones integradas se pueden llamar por sí mismas, pero los métodos siempre están adjuntos a una variable de lista desde la cual se llaman.

Otra diferencia notable es que, si bien no todas las funciones integradas están definidas para funcionar con listas, los métodos de lista solo funcionan con listas.

Aquí están los 11 métodos de lista para guardar en sus notas:

Método de lista	Descripción
`.append()`	Agregar un elemento al final de la lista
`.clear()`	Eliminar todos los elementos de la lista
`.copy()`	Devolver una copia superficial de la lista
`.count()`	Devuelve el número de veces que aparece el valor en la lista
`.extend()`	Agrega otra lista a la lista actual extendiéndola
`.index()`	Devuelve el índice de un valor dentro de la lista
`.insert()`	Insertar un elemento en una posición especificada en la lista
`.pop()`	Eliminar un elemento de una posición específica en la lista
`.remove()`	Eliminar un artículo de la lista según el valor del artículo
`.reverse()`	Invierte la lista en su lugar
`.sort()`	Ordena la lista en su lugar

Aca Estan Los ejemplos de cada uno de ellos y hay mas terminos

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

```
numbers = [1, 2, 3]
numbers.append(4)
print(numbers)  # Salida: [1, 2, 3, 4]
```

2. **`.clear()`**: Elimina todos los elementos de la lista.

```
numbers = [1, 2, 3]
numbers.clear()
print(numbers)  # Salida: []
```

3. **`.copy()`**: Devuelve una copia superficial de la lista.

```
numbers = [1, 2, 3]
numbers_copy = numbers.copy()
print(numbers_copy)  # Salida: [1, 2, 3]
```

4. **`.count()`**: Devuelve el número de veces que aparece un valor en la lista.

```
numbers = [1, 2, 2, 3, 3, 3]
count = numbers.count(3)
print(count)  # Salida: 3
```

5. **`.extend()`**: Agrega los elementos de una lista (o cualquier iterable) al final de la lista actual.

```
numbers = [1, 2, 3]
more_numbers = [4, 5, 6]
numbers.extend(more_numbers)
print(numbers)  # Salida: [1, 2, 3, 4, 5, 6]
```

6. **`.index()`**: Devuelve el índice del primer elemento con el valor especificado.

```
numbers = [10, 20, 30, 40, 50]
index = numbers.index(30)
print(index)  # Salida: 2
```

7. **`.insert()`**: Inserta un elemento en la posición especificada.

```
numbers = [1, 2, 3, 4, 5]
numbers.insert(2, 10)
print(numbers)  # Salida: [1, 2, 10, 3, 4, 5]
```

8. **`.pop()`**: Elimina y devuelve el elemento en la posición dada.

```
numbers = [1, 2, 3, 4, 5]
popped_element = numbers.pop(2)
print(numbers)        # Salida: [1, 2, 4, 5]
print(popped_element) # Salida: 3
```

9. **`.remove()`**: Elimina el primer elemento con el valor especificado.

```
numbers = [1, 2, 3, 4, 5]
numbers.remove(3)
print(numbers)  # Salida: [1, 2, 4, 5]
```

10. `.reverse()` : Invierte los elementos de la lista.

```
numbers = [1, 2, 3, 4, 5]
numbers.reverse()
print(numbers)  # Salida: [5, 4, 3, 2, 1]
```

11. `.sort()` : Ordena los elementos de la lista.

```
numbers = [3, 2, 1, 5, 4]
numbers.sort()
print(numbers)  # Salida: [1, 2, 3, 4, 5]
```

12. `sorted()` : Devuelve una nueva lista ordenada sin modificar la original.

```
numbers = [3, 2, 1, 5, 4]
sorted_numbers = sorted(numbers)
print(sorted_numbers)  # Salida: [1, 2, 3, 4, 5]
```

13. `reversed()` : Devuelve un iterador que accede a los elementos de la lista en orden inverso.

```
numbers = [1, 2, 3, 4, 5]
reversed_numbers = list(reversed(numbers))
print(reversed_numbers)  # Salida: [5, 4, 3, 2, 1]
```

14. `max()` : Devuelve el elemento máximo de la lista.

```
numbers = [3, 7, 2, 8, 5]
max_number = max(numbers)
print(max_number)  # Salida: 8
```

15. `min()` : Devuelve el elemento mínimo de la lista.

```
numbers = [3, 7, 2, 8, 5]
min_number = min(numbers)
print(min_number)  # Salida: 2
```

16. `sum()` : Devuelve la suma de todos los elementos de la lista.

```
numbers = [1, 2, 3, 4, 5]
total = sum(numbers)
print(total)  # Salida: 15
```

17. `len()` : Devuelve la longitud de la lista.

```
numbers = [1, 2, 3, 4, 5]
length = len(numbers)
print(length)  # Salida: 5
```