

1 Elementos iterables

Listas: Se cuenta desde la posición 0 hasta la posición (largo - 1). Son mutables, por lo que se puede modificar un elemento directamente. Además, son iterables, por lo que se puede recorrer o acceder a cada elemento usando un ciclo While o un ciclo For-In.

```
[2]: lista = [3,4,1,2,6,7,3,5]

contador = 0
    # 0 < 8?
    # 1 < 8?
    # 2 < 8?
while contador < len(lista):
    numero = lista[contador] # 0 = 3
    print(numero) # 3
    contador += 1
```

3
4
1
2
6
7
3
5

Strings: Se cuenta desde la posición 0 hasta la posición (largo - 1). Son inmutables, por lo que NO se puede modificar un elemento directamente. Pero al igual que las listas, son iterables, por lo que se puede recorrer o acceder a cada elemento usando un ciclo While o un ciclo For-In.

```
[3]: nombre = "John Serrano"

contador = 0
    # 0 < 12?
    # 1 < 12?
    # 2 < 12?
while contador < len(nombre):
    letra = nombre[contador] # 0 = J
    print(letra) # J
    contador += 1
```

J
o
h
n

S

e
r
r
a
n
o

2 Strip()

El método `.strip()` puede ser aplicado a un String para eliminar los espacios al inicio y al final de este. Además, elimina el caracter de salto de línea (`"\n"`).

```
[5]: string = "    hola    a    todos    "  
print(string)  
print(string.strip())
```

```
    hola    a    todos  
hola    a    todos
```

3 Split()

El método `.split()` puede ser aplicado a un String y funciona como una especie de reemplazo a `eval()`, ya que permite separar elementos por algún elemento específico y un resultado es una lista que contiene los elementos separados.

Por ejemplo, mediante un input, podría tener lo siguiente: e1,e2,e3,e4 Notemos que e1, e2, e3 y e4 están separados por comas (,). Al aplicar `split(",")`, el resultado sería el siguiente:

```
[e1,e2,e3,e4]
```

Se debe tener en consideración que no siempre el elemento separador es una coma (","), puede ser cualquier string.

```
[6]: lista = input("Ingrese una lista: ")  
lista_elementos = lista.split(",")  
print(lista_elementos)  
print(lista_elementos[0])  
print(type(lista_elementos))
```

```
Ingrese una lista: 3,4,5,6,2,1,4,7,8,3,2
```

```
['3', '4', '5', '6', '2', '1', '4', '7', '8', '3', '2']
```

```
3
```

```
<class 'list'>
```

```
[7]: lista = input("Ingrese una lista: ")  
lista_elementos = lista.split("/")  
print(lista_elementos)  
print(lista_elementos[0])  
print(type(lista_elementos))
```

Ingrese una lista: 29/10/2024

```
['29', '10', '2024']
```

```
29
```

```
<class 'list'>
```

```
[9]: lista = input("Ingrese una lista: ")
      lista_elementos = lista.split("hol")
      print(lista_elementos)
      print(lista_elementos[0])
      print(type(lista_elementos))
```

Ingrese una lista: perrito, hola, mundo, como estan

```
['perrito, ', 'a, mundo, como estan']
```

```
perrito,
```

```
<class 'list'>
```

Nótese que en el ejemplo anterior “a, mundo, como están”, es un solo string.

4 Listas de Listas

Una lista de listas es simplemente una lista grande donde sus elementos son otras listas. Dado lo anterior, aplican todas las reglas, métodos y operaciones que cuando se trabaja con listas, tanto a la hora de trabajar con la lista principal (la lista que contiene las demás listas) y las sub-listas (los elementos de la lista principal que contienen a otros elementos).

```
[10]: listas_de_listas = [[3,4,2],["hola", "mundo"],[True,False]]
      print(listas_de_listas[1][1])
      print("-----")
      listas = []
      numeros = [3,4]
      palabras = ["perrito", "arbol"]
      primer_numero = numeros[0]
      segundo_numero = numeros[1]
      primera_palabra = palabras[0]
      segunda_palabra = palabras[1]
      listas.append([primer_numero, segundo_numero])
      listas.append([primera_palabra, segunda_palabra])

      print(listas)
```

```
mundo
```

```
-----
```

```
[[3, 4], ['perrito', 'arbol']]
```

```
[ ]:
```