

Lista

Es una colección ordenada y modificable. En Python, éstas se declaran entre corchetes.

```
1 lista_frutas = ["fresa", "uva", "cereza"]
2 print(lista_frutas)
3
```

Si deseamos obtener acceso a algún elemento en la lista, usaremos la siguiente indicación, colocando el número de index (recordemos que en programación éstos comienzan en 0) que queremos imprimir entre los corchetes. También podemos pedirle al programa que lea el número de index de forma inversa añadiendo un signo «-» antes del número. Así, -1 se refiere al último elemento, -2 al penúltimo, etc.

La **función len()** cuenta los elementos de una lista, ejemplo:

```
>>>estudiantes = ["Jose", "Raul", "Marcelo"]
>>>len(estudiantes)
3
```

Añadir elementos a una lista con el método insert(indice, objeto)

El método insert espera un índice y el valor a agregar, ejemplo:

```
>>>estudiantes = ["Jose", "Raul", "Marcelo"]
>>>estudiantes.insert(0, "Maria")
>>>print estudiantes
["Maria", "Jose", "Raul", "Marcelo"]
```

Añadir elementos a una lista con el método append(objeto)

```
>>>estudiantes = ["Jose", "Raul", "Marcelo"]
>>>estudiantes.append("Mariela")
>>>print estudiantes
["Jose", "Raul", "Marcelo", "Mariela"]
```

Añadir varios elementos a una lista con el método extend(segmento)

```
>>>estudiantes = ["Jose", "Raul", "Marcelo"]
>>>estudiantes.extend(["Dario", "Natalia"])
>>>print estudiantes
["Jose", "Raul", "Marcelo", "Dario", "Natalia"]
```

Buscar elementos de una lista con el método index(objeto)

```
>>>estudiantes = ["Jose", "Raul", "Marcelo"]
>>>estudiantes.index("Raul")
1 #Retorna el índice del elemento "Raul"
```

Eliminar un elemento de la lista con el método remove(objeto)

```
>>>estudiantes = ["Jose", "Raul", "Marcelo"]
>>>estudiantes.remove("Raul")
>>>print estudiantes
["Jose", "Marcelo"]
```

Convertir cadenas en una lista con el método split()

Con el **método split()** podemos convertir una cadena de caracteres en una lista delimitada por los índices que indiquemos. Si no se indica nada se utilizan los espacios:

```
>>>"Hola mi nombre es Diego".split()
```

```
["Hola","mi","nombre","es","Diego"]
```

```
>>>"1-2-3-4-5-6".split("-")
```

```
["1","2","3","4","5","6"]
```

Buscar y saber si hay elemento duplicados con el método set()

```
>>>estudiantes = ["Martin", "Jose", "Raul", "Jose"]
```

```
>>>estudiantes_unicos = list(set(estudiantes))
```

```
>>>print estudiantes_unicos
```

```
["Raul", "Jose", "Martin"]
```

Veamos un ejemplo con una función:

```
def duplicado(lista):
```

```
    new_list = list(set(lista))
```

```
    if len(new_list) != len(lista):
```

```
        return True #Retorna True si hay duplicados
```

```
    else:
```

```
        return False #Retorna False si no hay duplicados
```

Con el **método pop()**, podremos eliminar y mostrar el ultimo elemento de la lista, ejemplo:

```
>>>estudiantes = ["Jose", "Raul", "Marcelo"]
```

```
>>>estudiantes.pop()
```

```
"Marcelo"
```

```
>>>estudiantes
```

```
["Jose", "Raul"]
```