

¿Qué es una lista?

Una lista es una estructura de datos y un tipo de dato en python con características especiales. Lo especial de las listas en Python es que nos permiten almacenar cualquier tipo de valor como enteros, cadenas y hasta otras funciones; por ejemplo:

```
lista = [1, 2.5, 'DevCode', [5,6], 4]
```

Una lista es un arreglo de elementos donde podemos ingresar cualquier tipo de dato, para acceder a estos datos podemos hacer mediante un índice.

```
print lista[0] # 1
print lista[1] # 2.5
print lista[2] # DevCode
print lista[3] # [5,6]
print lista[3][0] # 5
print lista[3][1] # 6
print lista[1:3] # [2.5, 'DevCode']
print lista[1:6] # [2.5, 'DevCode', [5, 6], 4]
print lista[1:6:2] # [2.5, [5, 6]]
```

Como pueden darse cuenta podemos hasta insertar una lista dentro de otra lista.

Si no quieres estar imprimir uno por uno los elementos de una lista, puedes recorrerlo utilizando un for.

```
for element in lista:
    print element
```

Métodos de las Listas

Las listas en Python tienen muchos métodos que podemos utilizar, entre todos ellos vamos a nombrar los más importantes. Para esto utilizaremos esta lista de ejemplo.

```
my_list = [2, 5, 'DevCode', 1.2, 5]
```

Sobre esta vamos a realizar diferentes métodos que son propios de las listas.

Append()

Este método nos permite agregar nuevos elementos a una lista.

```
my_list.append(10) # [2, 5, 'DevCode', 1.2, 5, 10]
my_list.append([2,5]) # [2, 5, 'DevCode', 1.2, 5, [2, 5]]
```

Podemos agregar cualquier tipo de elemento a una lista, pero tengan en cuenta lo que pasa cuando agregamos una lista dentro de otra, esta lista se agrega como uno y solo un elemento.

Extend()

Extend también nos permite agregar elementos dentro de una lista, pero a diferencia de append al momento de agregar una lista, cada elemento de esta lista se agrega como un elemento más dentro de la otra lista.

```
my_list.extend([2,5]) # [2, 5, 'DevCode', 1.2, 5, 2, 5]
```

Remove()

El método remove va a remover un elemento que se le pase como parámetro de la lista a donde se le esté aplicando.

```
my_list.remove(2) # [5, 'DevCode', 1.2, 5]
```

En este ejemplo estamos removiendo el elemento 2, de la lista que tiene por nombre "my_list".

Index()

Index devuelve el número de índice del elemento que le pasemos por parámetro.

```
my_list.index('DevCode') # 2
```

Aquí estamos preguntando por el índice de la cadena 'DevCode' dentro de la lista "my_list", esto devuelve 2.

Count()

Para saber cuántas veces un elemento de una lista se repite podemos utilizar el método count().

```
my_list.count(5) # 2
```

Contamos cuantas veces se repite el número 5 dentro de la lista, y esto devuelve 2.

Reverse()

También podemos invertir los elementos de una lista.

```
my_list.reverse() # [5, 1.2, 'DevCode', 5, 2]
```

Estas son algunos de los métodos más útiles y más utilizados en las listas.