

# Listas en Python

Téllez Gerardo Rubén

9/4/2021

## Definición

Son listas que permiten almacenar una colección de objetos en un orden determinado. Se pueden acceder, añadir o eliminar objetos de forma sencilla.

```
#Lista vacía
L1 = []

#Lista con valores iniciales
L2 = ["a", "e", "i", "o", "u"]

#Lista con valores enteros
L3 = [2, 3, 5, 7, 11]

#Lista heterogénea con varios tipos de datos diferentes
L4 = ["a", 5, True, "Rubén Téllez"]

#Repetir listas
L5 = [0]*10

print("Lista vacía:", L1)

## Lista vacía: []

print("Lista con valores iniciales:", L2)

## Lista con valores iniciales: ['a', 'e', 'i', 'o', 'u']

print("Lista con valores enteros:", L3)

## Lista con valores enteros: [2, 3, 5, 7, 11]

print("Lista heterogénea con varios tipos de datos diferentes:", L4)

## Lista heterogénea con varios tipos de datos diferentes: ['a', 5, True, 'Rubén Téllez']
```

```
print("Lista con 10 ceros, similar a rep(vec, n):", L5)
```

```
## Lista con 10 ceros, similar a rep(vec, n): [0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
```

## Indices

Las listas en Python empiezan en el número 0, igual que Python o Java. Es decir, considera el 0 como el primer número natural.

Se accede mediante la sintaxis de corchete, permite modificarlo.

Intentar acceder a un índice fuera de rango arroja un error.

Se puede acceder a una sublista, es decir accesos de posiciones múltiples, tal que + **Lista[0:4]**: Empieza en 0 y acaba en 3, pero incluirá 4 valores. + **Lista[:5]**: Recupera los índices antes del indicado. + **Lista[1:]**: Recupera los índices del índice indicado hasta el último. + **Lista[1::2]**: De uno en adelante, de dos en dos, el primer elemento es el primer índice, el segundo elemento los saltos

```
L3 = [2, 3, 5, 7, 11]
```

```
print(L3)
```

```
## [2, 3, 5, 7, 11]
```

```
print(L3[0::2])
```

```
## [2, 5, 11]
```

```
L3[0] = 10
```

```
print(L3)
```

```
## [10, 3, 5, 7, 11]
```

```
print(L3[-1]) #Obtener el último elemento
```

```
## 11
```

```
L3 = [2, 3, 5, 7, 11]
```

```
print(L3[0:4])
```

```
#Para modificar valores múltiples de golpe
```

```
## [2, 3, 5, 7]
```

```
L3[2:5] = (10, 20, 30)
```

```
print(L3)
```

```
## [2, 3, 10, 20, 30]
```

## Metodos de las listas

- **lista.append(x)**: Agrega  $x$  al final de la lista
- **lista.insert(indice, x)**: Agrega  $x$  exactamente antes del índice
- **lista.remove(x)**: Elimina  $x$  de la lista, sólo el primer objeto
- **lista.clear()**: Elimina todos los elementos de una lista
- **lista.pop()**: Elimina el último elemento de una lista
- **del lista[indice]**: Remover el índice indicado de una lista
- **lista.reverse()**: Invierte el orden de la lista
- **lista.count(x)**: Para contar las ocurrencias de un elemento  $x$
- **lista.index(x)**: Para recuperar el índice donde se encuentra  $x$
- **len(lista)**: Para contar los elementos de una lista
- **x in lista**: Devuelve el bool de si está  $x$  en la lista

```
L3 = [2, 3, 5, 7, 11]
print(L3)
```

```
## [2, 3, 5, 7, 11]
```

```
L3.append(100)
print(L3)
```

```
## [2, 3, 5, 7, 11, 100]
```

```
L3.remove(2)
print(L3)
```

```
## [3, 5, 7, 11, 100]
```

```
print(L3.count(100))
```

```
## 1
```

```
print(L3.index(100))
```

```
## 4
```

```
print(100 in L3)
```

```
## True
```

```
print(len(L3))
```

```
## 5
```

```
print(L3.reverse())
```

```
## None
```

```
print(L3.clear())
```

```
## None
```