



# Programación Python

José Javier Galán Hernández:



# Programación Python

## 8. LISTAS

# 8. LISTAS

Secuencia ordenada de valores a los que se puede acceder por índice o posición.

Declaración:

Separados por comas y entre corchetes:

```
In [1]: lista = ["Lunes", "Martes", "Miercoles", "Jueves", "Viernes", "Sabado", "Domingo"]  
print(lista)  
  
['Lunes', 'Martes', 'Miercoles', 'Jueves', 'Viernes', 'Sabado', 'Domingo']
```

Usando la función list(), ojo al doble paréntesis

```
In [3]: lista = list(("Lunes", "Martes", "Miercoles", "Jueves", "Viernes", "Sabado", "Domingo"))  
print(lista)  
  
['Lunes', 'Martes', 'Miercoles', 'Jueves', 'Viernes', 'Sabado', 'Domingo']
```

Declaración lista vacía

```
In [6]: 1 listaVacia = []  
2 print(listaVacia)  
3  
  
[]
```



# 8. LISTAS

**Accedemos** a un elemento de la lista mediante su índice, la posición del primer elemento es 0.

```
In [10]: lista = ["Lunes", "Martes", "Miercoles", "Jueves", "Viernes", "Sabado", "Domingo"]  
         print(lista[5])
```

Sabado

```
In [11]: lista[5] = "Sábado"  
         print("He modificado ", lista[5], " de la lista ", lista)
```

He modificado Sábado de la lista ['Lunes', 'Martes', 'Miercoles', 'Jueves', 'Viernes', 'Sábado', 'Domingo']



# 8. LISTAS

Podemos **añadir** elementos a la lista:

```
In [12]: lista.append("Dia extra")#Indicamos que añada un valor mas en la lista creada
print(lista)
```

```
['Lunes', 'Martes', 'Miercoles', 'Jueves', 'Viernes', 'Sábado', 'Domingo', 'Dia extra']
```

```
In [13]: lista[len(lista):] = ["Otro dia"]#Asignamos un valor en la posicion longitud, al empezar en 0, sera como añadir
print(lista)
```

```
['Lunes', 'Martes', 'Miercoles', 'Jueves', 'Viernes', 'Sábado', 'Domingo', 'Dia extra', 'Otro dia']
```

```
In [14]: lista.extend(["Tercer valor añadido"])
print(lista)
```

```
['Lunes', 'Martes', 'Miercoles', 'Jueves', 'Viernes', 'Sábado', 'Domingo', 'Dia extra', 'Otro dia', 'Tercer valor añadido']
```

**Eliminar** un elemento:

```
In [17]: del lista[9]
print(lista)
```

```
['Lunes', 'Martes', 'Miercoles', 'Jueves', 'Viernes', 'Sábado', 'Domingo', 'Dia extra', 'Otro dia']
```

```
In [22]: del lista[len(lista)-1]
print(lista)
```

```
['Lunes', 'Martes', 'Miercoles', 'Jueves', 'Viernes', 'Sábado', 'Domingo', 'Dia extra']
```



# 8. LISTAS

Errores fuera de rango:

```
In [15]: lista[15] = ["15"]#Error  
print(lista)
```

```
-----  
IndexError                                Traceback (most recent call last)  
<ipython-input-15-c4158152d14a> in <module>  
----> 1 lista[15] = ["15"]#Error  
      2 print(lista)  
  
IndexError: list assignment index out of range
```

```
In [16]: print(lista[15])
```

```
-----  
IndexError                                Traceback (most recent call last)  
<ipython-input-16-111230de3482> in <module>  
----> 1 print(lista[15])  
  
IndexError: list index out of range
```

# 8. LISTAS

Las listas pueden incluir elementos de **todo tipo**, incluso **listas dentro de listas**.

```
In [24]: superLista = [["1","hola",1],[["as"],[2],[5],["df"],[324]]]#Lista con 2 listas de valores de distinto tipo  
print(superLista[1][2])
```

[5]



# 8. LISTAS

Se puede hacer referencia a la lista en **sentido inverso** haciendo uso de números negativos.

Se puede hacer referencia a una sección (slicing, rebanado) de la lista indicando:

la posición del valor inicial : numero anterior al segundo indicado

```
In [25]: listaNumeros = [1,2,3,4,5,6,7,8,9,10]  
print(listaNumeros)
```

```
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
```

```
In [26]: print(listaNumeros[-1])
```

```
10
```

```
In [27]: print(listaNumeros[-5])
```

```
6
```

```
In [28]: print(listaNumeros[1:5])
```

```
[2, 3, 4, 5]
```

```
In [31]: nuevaLista = (listaNumeros[1:5])  
print(nuevaLista)
```

```
[2, 3, 4, 5]
```





