

MÓDULOS: ESP - ECP

LISTAS EN PYTHON

José L. Berenguel

4. Listas

- ▶ Las listas almacenan **datos de cualquier tipo**. Son **mutables** y **dinámicas**.
 - Se crea con los elementos entre corchetes [] y separados por comas o con la función list() pasando un objeto iterable.
 - Mantienen el orden y son accesibles a través de un índice entre corchetes. Los índices funcionan de igual forma que los slices vistos para las cadenas.

```
# Creamos una lista y visualizamos sus valores
a = [90, "Python", 3.87]
print(a[0]) #90
print(a[1]) #Python

# Modificamos el último elemento
a[-1] = [4,5,"Hola"]
print(a) # [90, 'Python', [4, 5, 'Hola']]

# Eliminamos el segundo elemento
del(a[1])
print(a) # [90, [4, 5, 'Hola']]
```

Listas

- ▶ **Listas anidadas.** Una lista puede contener como elemento otra lista, y esta a su vez otra lista. Para acceder a sus elementos tenemos que usar los corchetes tantas veces como sea necesario.

```
a = # [90, [4, 5, 'Hola']]  
print(a[1][2]) # Hola  
  
b = [1, 2, 3, ['p', 'q', [5, 6, 7]]]  
print(b[3][0]) #p  
print(b[3][2][0]) #5
```

Listas

► Acceso y modificación múltiple.

- Se pueden modificar múltiples elementos de la lista usando un slice (:).
- También es posible asignar múltiples valores de la lista a múltiples variables.

```
a = [1, 2, 3, 4, 5, 6]
a[1:3] = [7,7]
print(a) # [1, 7, 7, 4, 5, 6]

x, y, z = a[3:]
print(x) # 4
print(y) # 5
print(z) # 6
```

Listas

► Iterar listas.

- Se realiza de forma sencilla mediante **for** y el operador **in** (se recorren todos los elementos de la lista).
- Si necesitamos los índices se puede emplear la función **enumerate()**.

```
lista = [5, 9, 10]
# Recorremos todos los elementos
for l in lista:
    print(l)

# Obtenemos el índice y el elemento
for index, l in enumerate(lista):
    print(index, l)
```

Listas

► Iterar listas.

- También se puede iterar a través de los índices y la posición de cada elemento en la lista.

```
lista = [1, 2, 3, 4, 5, 6]
# Recorremos las posiciones pares
for i in range(0, len(lista), 2):
    print(lista[i])
```

Listas

► Iterar listas.

- Se pueden iterar dos listas a la vez con la función **zip()**.

```
lista1 = [5, 9, 10]
lista2 = ["Jazz", "Rock", "Djent"]
for l1, l2 in zip(lista1, lista2):
    print(l1, l2)
```

Manejo avanzado de datos

FIN