

PROGRAMACIÓN DE INTELIGENCIA ARTIFICIAL

PYTHON

Una lista es una colección **ORDENADA** y **MUTABLE** de elementos separados por comas y entre corchetes. Los elementos pueden ser de diferente tipo.

```
>>> type(my_list)
<class 'list'>
```

```
>>> len(my_list)
5
```

La función **len()** permite verificar la longitud de la lista

```
>>> lista_1 = [1]
>>> lista_2 = lista_1 #lista_1 y lista_2 apuntan a la misma memoria
>>> lista_1[0] = 3    #lista_2[0] también es el valor 3
```

El nombre de una lista es el nombre de una ubicación de memoria donde se almacenan los elementos de la lista

```
my_list = [1.0, None, True, "Soy una cadena", 256]
```

```
my_list = []
```

CREA UNA LISTA **VACIA**

También se puede crear una lista con la función constructor **list()** pasándole de parámetro un *objeto iterable*

```
>>> another_list = list(range(5))
>>> print(another_list)
[0, 1, 2, 3, 4]
```

```
>>> other_list = list("Cadena")
>>> print(other_list)
['C', 'a', 'd', 'e', 'n', 'a']
```

CONCATENAR LISTAS

```
>>> a=[1,2,3]
>>> b = a + [True, 2.5] + ["b", False]
```

```
>>> print (b)
[1, 2, 3, True, 2.5, 'b', False]
```

PROGRAMACIÓN DE INTELIGENCIA ARTIFICIAL

PYTHON

ACCESO A ELEMENTOS DE LA LISTA: INDICES

```
>>> for i in range(len(my_list)):
...     print(my_list[i])
```

Recorrido de **izquierda a derecha** →
índices: 0, 1, ..., len() - 1

```
>>> for i in range(-1, -len(my_list), -1):
...     print(my_list[i])
```

Recorrido de **derecha a izquierda** ←
índices: -1, -2, -3, ..., -len

```
a = [90, "Python", 3.87]
print(a[0])    #90
print(a[2])    #3.87

print(a[-1])   # 3.87
print(a[-3])   #90
```

PROCESAR ELEMENTOS DE LA LISTA

```
>>> a = [1, 2, 3, 4, 5]
>>> for i in range(len(a)):
...     # i toma valores entre 0 y (5-1)
...     a[i] += 1
...
>>> print(a)
[2, 3, 4, 5, 6]
```

Procedimiento para
modificar o consultar
elementos de una lista

```
>>> a = [1, 2, 3, 4, 5]
>>> for elemento in a:
...     print(elemento)
...
1
2
3
4
5
>>>
```

Procedimiento para
consultar elementos de
una lista

ELIMINAR ELEMENTOS DE LA LISTA

```
>>> lista = [1,2,3,4,5,6,7]
>>> del lista[1]      [1, 3, 4, 5, 6, 7]
>>> del lista[1:3]    [1, 5, 6, 7]
>>> del lista[:]      []
```

INSERTAR ELEMENTOS EN LA LISTA

La clase **list** incluye dos métodos para añadir elementos a una lista

`my_list.append(valor)`

Añade **valor** como último elemento en la lista **my_list**

`my_list.insert(pos,valor)`

Añade **valor** ANTES de la posición **pos** en la lista **my_list**

OTROS MÉTODOS DE LA CLASE list

`my_list.pop()` Devuelve y elimina de la lista **my_list** el último elemento

`my_list.extend("Ana")` Concatena a la lista **my_list** los elementos del iterable "Ana"

`my_list.index("a")` Devuelve el primer índice del valor "a" en la lista **my_list** (Error si no existe)

`my_list.remove("a")` Elimina la primera ocurrencia del valor "a" en la lista **my_list** (Error si no existe)

`my_list.sort()` Ordena los elementos de la lista **my_list** (Error si los elementos no son del mismo tipo)

`my_list.reverse()` Invierte la lista **my_list**

**SEGMENTACIÓN DE LISTAS:
REBANADAS, PORCIONES
O SLICES:**

```
lista = [0, 1, 2, 3, 4, 5, 6, 7, 8]
```

```
>>> lista[3:]  
[3, 4, 5, 6, 7, 8]
```

```
>>> lista[:3]  
[0, 1, 2]
```

```
>>> lista[1:3]  
[1, 2]
```

```
>>> lista[3:-2]  
[3, 4, 5, 6]
```

```
>>> lista[::2]  
[0, 2, 4, 6, 8]
```

```
>>> lista[1::2]  
[1, 3, 5, 7]
```

```
>>> lista[-3:4]  
[]
```

```
>>> lista[-3:4:-1]  
[6, 5]
```

PROGRAMACIÓN DE INTELIGENCIA ARTIFICIAL

PYTHON

La **comprensión de listas** permite crear listas con una sola línea de código.

lista = [expresión for elemento in iterable]

lo que será añadido a la lista en cada iteración

powers = [2 ** i for i in range(16)]

powers = []
for i in range(16):
 powers.append(2 ** i)

temps = [[0.0 for h in range(24)] for d in range(31)]

temps = []
for d in range(31):
 for h in range(24):
 temps[d][h] = 0.0

lista = [expresión for elemento in iterable if condicion]

lo que será añadido a la lista en cada iteración si se cumple la condición

filtro aplicable a la iteración

lista = [i for i in range(30) if i % 3 == 0]

lista = [0, 3, 6, 9, 12, 15, 18, 21, 24, 27]

La expresión puede contener condicionales como forma de manipular el resultado

a = [2, -1, -4, 1, 5, -6]

b = [n if n > 0 else 0 for n in a] b = [2, 0, 0, 1, 5, 0]

nombre = ["Eva", "Adán", "Ruth", "Noé"]

edad = [21, 23, 17, 30]

a = [(n, e) for n in nombre for e in edad if n.startswith("E") if e > 22]

Producto de dos listas según condiciones

a = [('Eva', 23), ('Eva', 30)]