



Funciones avanzadas

Listas

Índice

- ¿Qué es una lista?
- Acceso a una lista
- Recorrido de una lista
- Funciones para trabajar con listas
- Métodos de listas
- Tuplas
- Comprensión de listas

¿Qué es una lista?

- Conjunto de elementos, que pueden ser del mismo o de diferente tipo , separados por comas y encerrados entre corchetes.
- Las listas son dinámicas, lo que significa que se pueden añadir y eliminar elementos de la misma en tiempo de ejecución
- Ejemplo:

```
lista1=["lunes","martes","miercoles","jueves","viernes"]  
lista2=[100,"palabra",True].
```


Acceso a una lista

- Se accede al contenido de una lista igual que en una cadena de caracteres, utilizando los corchetes y un índice o rango de índices:

```
print(lista1[0:3])    #muestra ['lunes', 'martes', 'miercoles']  
print(lista2[2])     #muestra True
```

- Se puede modificar cualquier elemento de la lista accediendo mediante el índice:

```
lista2[2]="nuevo"
```

- Si el **índice está fuera de rango se produce un error.**

Recorrido de una lista

- Se puede recorrer una lista utilizando la instrucción for:

```
for n in lista1:  
    print("El días es el "+n)
```

- Si lo que queremos es comprobar si un elemento está o no en una lista, no necesitamos recorrerla:

```
if "lunes" in lista1:  
    print("Ese día está incluido "+n)
```


Funciones para trabajar con listas

➤ Python soporta las siguientes funciones para manejar listas de datos:

- **len(lista)**. Devuelve el número total de elementos de la lista
- **max(lista)**. Devuelve el mayor de los elementos de la lista. Deben ser del mismo tipo
- **min(lista)**. Devuelve el menor de los elementos de la lista

```
lista1=["lunes","martes","miercoles","jueves","viernes"]  
print(min(lista1)) #jueves  
print(max(lista1)) #viernes
```


Métodos de listas

- **append(obj)**. Añade un nuevo elemento a la lista
- **count(obj)**. Indica el número de veces que el objeto aparece en la lista
- **remove(obj)**. Elimina el elemento de la lista
- **sort()**. Ordena la lista de menor a mayor
- **insert(pos,obj)**. Inserta el elemento en la posición indicada en el primer parámetro
- **reverse()**. Invierte el orden de la lista
- **index(obj)**. Devuelve la posición del elemento en la lista. Si no existe genera una excepción

Tuplas

- Una tupla es un conjunto de valores separados por comas y encerrados entre paréntesis
- Similar a la lista, pero a diferencia de esta, la tupla es inmutable (no se puede modificar el contenido)
- Ejemplo:

```
tupla=(30, 25, 67, 11)
```
- La manipulación se hace de la misma manera que con las listas
- La función *tuple()* recibe como parámetro una lista y la convierte en una tupla

Comprensión de listas

- Permite generar una lista a partir de una expresión
- Ejemplo: Generar una lista con los 10 primeros múltiplos de 3:

```
lista=[n*3 for n in range(0,11)]
```

- Ejemplo: Generar una lista con los números pares comprendidos entre 1 y 10:

```
lista=[n for n in range(0,11) if n%2==0]
```