# Python

Listas



1

## Listas

- **Son una Colección o agrupación de datos (objetos), encerrada entre corchetes**

  **[2, 23, 8, 48, 5, 0]**

- **Pueden ser de distinto tipo**

  **[235, “azul”, 28.3, “rojo”]**

- **Cada uno ocupa una posición comenzando en cero**

  **[235, “azul”, 28.3, “rojo”]**

  **0 1 2 3**



2

## Listas

- Se accede a los elementos a través de la posición que ocupan

```
>>> lista = [2, 23, 8, 48, 5, 0]
>>> lista[3]
48
```

- También puedo acceder a una posición, desde el final de la lista

```
>>> lista[ - 1]      # accedo a la última posición de la lista
0
>>> lista[ -4 ]      # accedo al 4to elemento desde el final de la lista
8
```



## Listas

- Se pueden concatenar listas

```
>>> lista = lista + [-1, -10, 23]
```

- Se puede asignar una lista vacía

```
>>> lista = [ ]
```

- Podemos conocer la cantidad de elementos usando la función len()

```
>>> len( lista )
```

## Listas

- **Son mutables, por lo tanto podemos modificar, agregar y eliminar elementos**

```
>>> lista[3] = 53          # reemplaza el valor de la posición 3
>>> lista.append( -15 )    # agrega un valor al final de la lista
>>> lista += [4]           # esto también agrega al final
>>> lista.extend( range(5) )  # agrega los elementos de un iterable
```



5

## Listas

```
>>> lista.insert( 1, -20)  # inserta un valor en la posición 1
>>> del(lista[0])          # elimina el valor en la posición 0
>>> lista.pop( -2 )        # elimina el anteultimo elemento y retorna su valor
>>> lista.remove( -15 )    # elimina la primer ocurrencia del valor -15
>>> lista.clear( )         # elimina todos los elementos en la lista
```



6

## Listas

- **Podemos pedir la cantidad de ocurrencias de un valor**

**>>> lista.count( -15 )**

- **Podemos revertir los elementos en la lista**

**>>> lista.reverse( )**

- **Podemos ordenar la lista sobre sí misma**

**>>> lista.sort( )**

**ó**

- **Podemos obtener una lista ordenada sin afectar a la original**

**>>> sorded( lista )**



7

## Listas

**Podemos acceder a "rebanadas" de una lista**

```
>>> lista = [10, 1, -3, 8, 23, 18]
```

*Posición Incluida*

```
>>> lista[0:4]
[10, 1, -3, 8]
```

*Posición Excluida*

```
>>> lista[:4]
[10, 1, -3, 8]

>>> lista[4:]
[23, 18]
```

*La ausencia de posición, implica desde la posición inicial ó hasta la posición final (incluídas)*



8

## Listas

```
>>> lista = [10, 1, -3, 8, 23, 18]
```

```
>>> lista[-4:-1]
[-3, 8, 23]
```

*Podemos acceder a una rebanada pero desde la última posición*

```
>>> lista[2:4] = [ ]
        ó
>>> del(lista[2:4])
```

*Podemos eliminar una parte de la lista*



9

## Listas

*Mostrar el contenido de una lista, imprimiendo un valor por línea.*

```
for elemento in lista:
    print(elemento)
```

*Generar una lista que contenga los cuadrados de los números del 1 al 100.*

```
lista = [ ]
for x in range(1, 101):
    lista.append( x**2 )
```



10

# Listas

## Generación de listas por comprensión

- *Es un método conciso y rápido para la creación de listas.*
- *Tiene correspondencia con la definición de conjuntos por comprensión en matemáticas.*
- *Consiste de **corchetes** rodeando una **expresión seguida de una declaración for** y luego cero o más declaraciones for if.*

```
lista = [ x**2 for x in range(1, 101) ]
```

*Idem anterior pero usando Listas por Comprensión*



11

# Listas Anidadas

## Listas que contienen listas

```
l_super = [[“jabón”, 3], [“agua mineral”, 2], [“vinos”, 10]]

l_alum_notas = [ [“Juan Perez”,[4, 2, 7]], [“Julieta Garcia”, [8, 6]], [“Gabriel Diaz”, [ ]] ]
```

*Generar una lista que contenga ternas formadas por x, $x^2$, $x^3$ para x entre 1 y 5.*

```
l_terna = [ ]
for x in range(1, 6):
    l_terna.append( [x, x**2, x**3] )
```

```
l_terna = [ [x, x*2, x*3] for x in range(1, 6) ]
```

*Usando listas por comprensión*



12

**Ejercicio**:

Escribir un programa modular que permita:

1. El ingreso de una secuencia de valores, que termina con el valor 0.
2. Muestre los valores ingresados, anteponiendo al valor, el orden en el que fue ingresado.
3. Listar los valores hasta encontrar el 3er. valor impar ingresado.
4. Listar los elementos que se encuentren en posiciones pares.
5. Listar los elementos ordenados de menos a mayor, sin repetirlos.

Agregar diapositivas sobre:

- Tuplas

13