

S.E.

Programar con Python

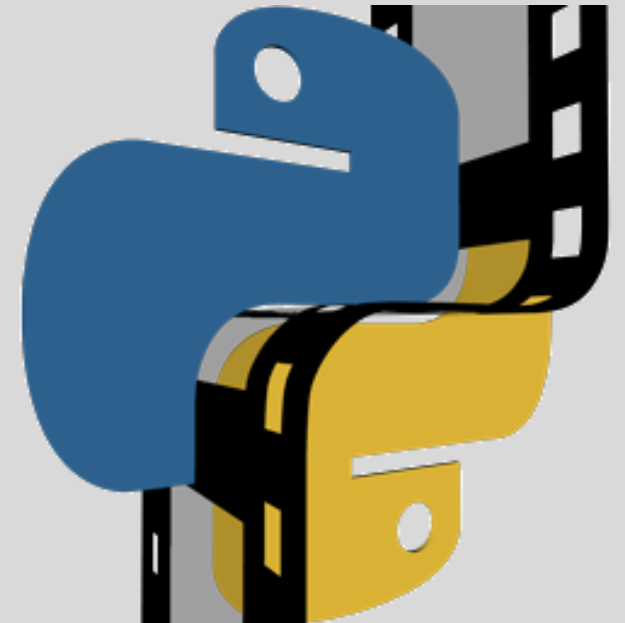
Nivel inicial

Clase 8

```
edit Selection View Go Run ... entrega final
PFI_Sylvina_Enriquez.py funciones.py X
funciones.py > registrar_productos
30 def crearTabla(nombreBdD):
42     # Confirmar la creación de la tabla y cerrar la conexión
43     conexion.commit()
44     print(Fore.GREEN+"\nLa tabla Productos fue creada con éxito.\n"+Style.RESET_ALL)
45     conexion.close()
46
47     #-----
48     # acá comienzan las definiciones de las funciones del menú principal
49     #-----
50     # esta función limpia la pantalla y muestra un nuevo título por la opción elegida
51     def titulo(numero,descripcion):
52         system("cls")
53         print(Fore.BLUE + Back.BLACK+f" OPCIÓN ELEGIDA: {numero} - {descripcion}\n"+Style.RESET_ALL)
54
55     #-----
56     def registrar_productos():
57         titulo(1,"REGISTRAR PRODUCTO/S")
58         cant=validar_positivo(int(input("¿cuántos productos registrará?: ")))
59         (conexion,cursor)=conectar()
60         for i in range(0,cant):
61             nombre=input(f"Ingresar el nombre del producto {i+1}: ")
62             while producto_existente(cursor,nombre):
63                 print(Fore.RED+"El producto ya se encuentra registrado. Se debe cambiar el nombre")
64                 nombre=input(f"Ingresar el nombre del producto {i+1}: ")
65             descripcion=input("Ingresar una descripción del producto: ")
66             cantidad=validar_positivo(int(input(f"Ingresar la cantidad en stock del producto {nombre}: ")))
67             precio=validar_positivo(float(input(f"Ingresar el precio de cada {nombre}: ")))
68             categoria=input(f"Ingresar la categoría de {nombre}: ")
69             print("")
70             #se conecta a la base de datos 'local'
71             cursor.execute(f"INSERT INTO productos (nombre,descripcion,cantidad,precio,categoria)
72             #se suben los datos a la base de datos 'externa'
73             conexion.commit()
```

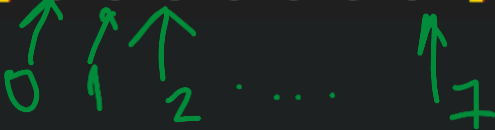
Contenido

- **Operaciones con LISTAS**
- **Vamos a armar una agenda con un menú interactivo**



ESTRUCTURAS DE DATOS: LISTA

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



Cada elemento está en una posición:

```
primero = lista[0]
```

```
ultimo = lista[-1]
```

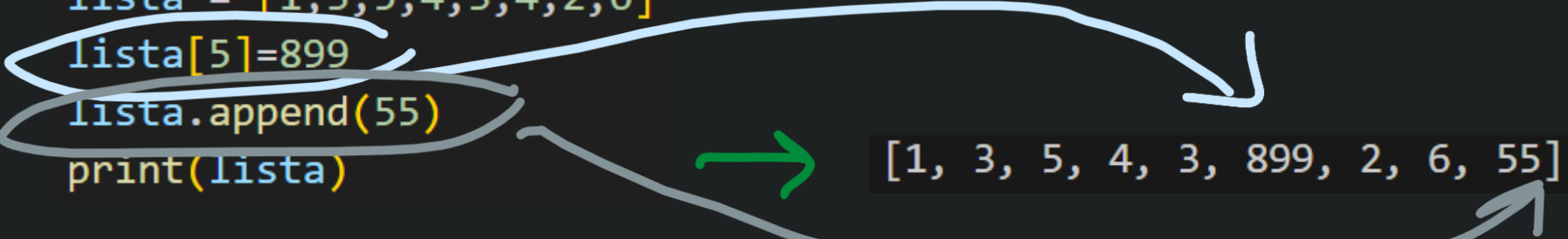
Cambiar un elemento en la lista. Agregar un elemento al final de la lista

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

```
lista[5]=899
```

```
lista.append(55)
```

```
print(lista)
```



```
[1, 3, 5, 4, 3, 899, 2, 6, 55]
```

ESTRUCTURAS DE DATOS: LISTA

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



Cantidad de elementos en la lista:

```
longitud = len(lista)
```



```
longitud = 8
```

Eliminar un elemento ubicado en una determinada posición:

```
del lista[2]
```



```
lista = [1, 3, 4, 3, 4, 2, 6]
```

Insertar un elemento en una posición dada:

```
lista.insert(2, 5)
```



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

En LISTA se inserta, en la posición **2** el elemento **5**

ESTRUCTURAS DE DATOS: LISTA

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

Unir dos listas:

```
nueva_lista = lista + [98,45] → nueva_lista= [1, 3, 5, 4, 3, 4, 2, 6, 98, 45]
```

Eliminar un elemento de la lista:

```
lista.remove(5) → lista= [1, 3, 4, 3, 4, 2, 6]
```

Extraer una serie de elementos de una lista:

```
lista_corta = lista[1:4] → lista_corta= [3, 5, 4]
```

De la LISTA se extraen los elementos desde la posición **1** hasta el anterior al de la posición **4**

ESTRUCTURAS DE DATOS: LISTA

```
nombres = ["José de San Martín", "Pedro", "Marcela"]
```

Imprimir la lista:

```
print(nombres)
```



```
['José de San Martín', 'Pedro', 'Marcela']
```

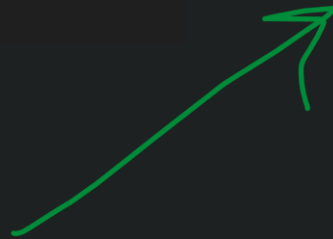
```
for i in range(0, len(nombres)):  
    print(nombres[i])
```



```
José de San Martín  
Pedro  
Marcela
```



```
for nombre in nombres:  
    print(nombre)
```



```
José de San Martín  
Pedro  
Marcela
```

LISTA DE TUPLAS:

```
lista_de_compras=[("rollo de cocina",2,"unidades"),("detergente",1,"unidad"),("bananas",2,"kilogramos")]
```

Imprimir la lista: `print(lista_de_compras)`

```
[('rollo de cocina', 2, 'unidades'), ('detergente', 1, 'unidad'), ('bananas', 2, 'kilogramos')]
```

```
for compra in lista_de_compras:  
    print(compra)
```



```
('rollo de cocina', 2, 'unidades')  
( 'detergente', 1, 'unidad')  
( 'bananas', 2, 'kilogramos')
```

```
print("cantidad\tproducto")  
for compra in lista_de_compras:  
    print(compra[1],compra[2],"\t",compra[0])
```



cantidad	producto
2 unidades	rollo de cocina
1 unidad	detergente
2 kilogramos	bananas

AGENDA E DICIONARIOS

8.1.: Desarrollar un programa que simule ser una agenda.

Debe tener un menú interactivo en el que se presenten las siguientes opciones:

- Ingresar nuevos datos de una persona
- Mostrar toda la agenda
- Borrar un contacto, según su ubicación en la agenda
- Salir



- EXPLORER
- ▼ EJ PYTHON
 - buscar.py
 - conjuntos.py
 - ej 5.1 - 1.1.py
 - ej 5.1 - 2.1.py
 - ej 5.1 - 3.1.py
 - ej 5.2.py
 - ej 5.3.py
 - ej 6.1 - 3.2.py
 - ej 6.1 - 3.3.py
 - ej 6.2 - 4.1.py
 - ej 6.2 - 4.2.py
 - ej 7.1.py
 - ej 7.2.py
 - ej 7.3.py



- Show All Commands `Ctrl + Shift + P`
- Go to File `Ctrl + P`
- Toggle Terminal `Ctrl + ñ`
- Open Settings `Ctrl + ,`
- Start Debugging `F5`

- > OUTLINE
- > TIMELINE



Si te quedó alguna duda: ¡ mandame un mail a cursos.se.programar@gmail.com indicando la clase y la pregunta!