

## ▼ UNIDAD 2 - FUNDAMENTOS DE PYTHON

### Condicionales, Funciones Iterativas y Funciones

Docente: MSc. Ruth Chirinos

Fecha Entrega: 08/Sept/2023 23:59

Referencia: <https://www.kaggle.com/learn/python>

1. Realizar un programa que almacene las asignaturas de la carrera de Sistemas (por ejemplo Algebra Lineal, Probabilidad Estadística, Programación, Algorítmica y Probabilidad) en una lista y la muestre por pantalla.

```
1

1 # Escribe aqui tu respuesta
2 #Bucle para inputs:
3 def input_1():
4     asignaturas = []
5
6     while True:
7         materia = input("Ingresa materias para la carrera de Ing. en Sistemas o escribe 'show' para mostrar inputs")
8         if materia.lower() == 'show':
9             break
10        asignaturas.append(materia)
11    return asignaturas
12
13 #Display la lista de materias:
14 def show_asignaturas(asignaturas):
15     print("\nAsignaturas de la carrera de Ingenieria en Sistemas:")
16     for i, materia in enumerate(asignaturas, start=1):
17         print(f"{i}. {materia}")
18
19 #Programa principal:
20 if __name__ == "__main__":
21     print("Programa para almacenar y mostrar asignaturas de la carrera de Ingenieria en Sistemas")
22
23     asignaturas = input_1()
24     show_asignaturas(asignaturas)
```

```
Programa para almacenar y mostrar asignaturas de la carrera de Ingenieria en Sistemas
Ingresa materias para la carrera de Ing. en Sistemas o escribe 'show' para mostrar inputsCalculo I
Ingresa materias para la carrera de Ing. en Sistemas o escribe 'show' para mostrar inputsCalculo II
Ingresa materias para la carrera de Ing. en Sistemas o escribe 'show' para mostrar inputsAlgoritmos
Ingresa materias para la carrera de Ing. en Sistemas o escribe 'show' para mostrar inputsEstadística I
Ingresa materias para la carrera de Ing. en Sistemas o escribe 'show' para mostrar inputsshow
```

```
Asignaturas de la carrera de Ingenieria en Sistemas:
1. Calculo I
2. Calculo II
3. Algoritmos
4. Estadística I
```

2. Realizar un programa que almacene las asignaturas de la carrera de Sistemas (por ejemplo Algebra Lineal, Probabilidad Estadística, Programación, Algorítmica y Probabilidad) en una lista, pregunte al usuario la nota que ha sacado en cada asignatura, y después las muestre por pantalla con el mensaje: 1) En **asignatura** has sacado **nota** ", donde **asignatura** es cada una de las asignaturas de la lista y **nota** cada una de las notas introducidas por el usuario.

```
1 # Escribe aqui tu respuesta
2 # Lista de asignaturas:
3 def asignaturas_1():
4     asignaturas = ["Calculo I", "Calculo II", "Ecuaciones Diferenciales", "Algebra Lineal", "Probabilidad y Estadística I","Valores"]
5     return asignaturas
6
7 # Input de notas
8 def input_notas(asignaturas):
```

```

9     notas = {}
10
11     for i in asignaturas:
12         nota = input(f"Ingresar la nota que has sacado en {i}: ")
13         notas[i] = nota
14
15     return notas
16
17 # Mostrar notas:
18 def display_notas(notas):
19     print("="*50)
20     print("Notas obtenidas en las asignaturas: ")
21     print("="*50)
22     for asignatura, nota in notas.items():
23         espacios = "_" * (30 - len(asignatura))
24         print(f"En {asignatura} has sacado{espacios}{nota}")
25
26 # Programa principal
27 if __name__ == "__main__":
28     print("Programa para almacenar y mostrar notas de asignaturas de la carrera de Sistemas")
29
30     asignaturas = asignaturas_1()
31     notas = input_notas(asignaturas)
32     display_notas(notas)
33

```

```

Programa para almacenar y mostrar notas de asignaturas de la carrera de Sistemas
Ingresar la nota que has sacado en Calculo I: 98
Ingresar la nota que has sacado en Calculo II: 99
Ingresar la nota que has sacado en Ecuaciones Diferenciales: 90
Ingresar la nota que has sacado en Algebra Lineal: 98
Ingresar la nota que has sacado en Probabilidad y Estadística I: 78
Ingresar la nota que has sacado en Valores: 89
=====
Notas obtenidas en las asignaturas:
=====
En Calculo I has sacado_____98
En Calculo II has sacado_____99
En Ecuaciones Diferenciales has sacado_____90
En Algebra Lineal has sacado_____98
En Probabilidad y Estadística I has sacado____78
En Valores has sacado_____89

```

3. Realizar un programa que almacene los siguientes vectores (2,-4,6) y (-1,8,0) en dos listas y muestre por pantalla su producto escalar.

```

1 # Escribe aquí tu respuesta
2 M = 'ERROR!: los vectores no tienen la misma notación'
3 def logica_pe(vector1, vector2):
4     if len(vector1) != len(vector2):
5         return M
6
7     resultado = 0
8     for i in range(len(vector1)):
9         resultado += vector1[i] * vector2[i]
10
11     return resultado
12
13 vector1 = [2, -4, 6]
14 vector2 = [-1, 8, 0]
15
16 producto_escalar = logica_pe(vector1, vector2)
17 print(producto_escalar)
18

```

-34

4. Realizar un programa que pregunte al usuario su nombre, edad, dirección y equipo de fútbol favorito y lo guarde en un diccionario.

Después debe mostrar por pantalla el mensaje <nombre> tiene <edad> años, vive en <dirección> y su equipo de fútbol favorito es <equipoFutbol>.

```

1 # Escribe aquí tu respuesta
2 nombre = input("Type nombre")
3 edad = input("Type edad")

```

```

4 direccion = input("Type direccion")
5 equipo_futbol = input("Type nombre equipo futbol favorito")
6
7 # Crear un diccionario con la información
8 usuario_info = {
9     'nombre': nombre,
10    'edad': edad,
11    'direccion': direccion,
12    'equipo_futbol': equipo_futbol
13 }
14
15 # Mostrar la información por pantalla
16 print(f"{usuario_info['nombre']} tiene {usuario_info['edad']} años, vive en {usuario_info['direccion']} y su equipo de fútbol favorito es
17

```

```

Type nombreNicolas
Type edad26
Type direccionSanta Cruz
Type nombre equipo futbol favoritoReal Madrid
Nicolas tiene 26 años, vive en Santa Cruz y su equipo de fútbol favorito es Real Madrid.

```

5. Realizar un programa que simule una cesta de compra en un supermercado creando un diccionario. El programa debe preguntar el artículo y su precio y añadir el par al diccionario, hasta que el usuario decida terminar. Después se debe mostrar por pantalla la lista de la compra y el coste total, con el siguiente formato

#### Lista de la compra

Artículo 1	Precio
Artículo 2	Precio
Artículo 3	Precio
...	...
Total	Coste

```

1 # Escribe aqui tu respuesta
2
3 def lista_compras(Compras):
4     print("| LISTA DE COMPRAS |")
5     print("="*81)
6     space= " "*15
7     print(f' |{space}ARTICULO{space}|{space} PRECIO {space}|')
8     total = 0
9
10    for i, (articulo, precio) in enumerate(lista_compra.items(), start=1):
11        space_articulo= "_"*(39-len(articulo))
12        space_price= "_"*(39-len(str(precio)))
13        space_total= "_"*(39-len(str(total)))
14        print(f"{articulo}{space_articulo}|{space_price}{precio}|")
15        total += precio
16
17    print("-"*81)
18    print(f' |{space} TOTAL {space}|{space_total}{total}|')
19
20 # Programa principal
21 if __name__ == "__main__":
22     lista_compra = {}
23
24     while True:
25         articulo = input("Ingresa el nombre del articulo o x para terminar")
26         if articulo.lower() == 'x':
27             break
28
29         try:
30             precio = float(input(f"Ingresa el precio de {articulo}: "))
31             lista_compra[articulo] = precio
32         except ValueError:
33             print("Ingresa un numero valido para el precio")
34
35     lista_compras(lista_compra)
36

```

```

Ingresa el nombre del articulo o x para terminarHuevos
Ingresa el precio de Huevos: 20
Ingresa el nombre del articulo o x para terminarCafe
Ingresa el precio de Cafe: 10
Ingresa el nombre del articulo o x para terminarTocino

```

```

Ingresa el precio de Tocino: 14
Ingresa el nombre del articulo o x para terminarPan
Ingresa el precio de Pan: 3
Ingresa el nombre del articulo o x para terminarMantequilla
Ingresa el precio de Mantequilla: 18
Ingresa el nombre del articulo o x para terminarx
| LISTA DE COMPRAS |
=====
| ARTICULO | PRECIO |
|-----|-----|
| Huevos | 20.0 |
| Cafe | 10.0 |
| Tocino | 14.0 |
| Pan | 3.0 |
| Mantequilla | 18.0 |
|-----|-----|
| TOTAL | 65.0 |

```

6. Realizar un programa que administre las facturas pendientes de cobro de una empresa. Las facturas se almacenarán en un diccionario donde la clave de cada factura será el número de factura y el valor el coste de la factura. El programa debe preguntar al usuario si quiere añadir una nueva factura, pagar una existente o terminar. Si desea añadir una nueva factura se preguntará por el número de factura y su coste y se añadirá al diccionario. Si se desea pagar una factura se preguntará por el número de factura y se eliminará del diccionario. Después de cada operación el programa debe mostrar por pantalla la cantidad cobrada hasta el momento y la cantidad pendiente de cobro.

```

1 # Escribe aqui tu respuesta
2 def estado_facturas(facturas):
3     cobrado = sum(facturas.values())
4     pendiente = sumar_total_facturas - cobrado
5
6     print("Estado de las facturas:")
7     print(f"Total cobrado: {cobrado}")
8     print(f"Total pendiente de cobro: {pendiente}")
9
10 if __name__ == "__main__":
11     facturas = {}
12
13     while True:
14
15
16         action = input("Pulsa el numero de una accion: 1. Añadir una nueva factura, 2. Pagar una factura existente,3. Mostrar estado de la
17
18         if action == '1':
19             numero_factura = input("Ingresa el numero de factura: ")
20             costo_factura = float(input("Ingresa el costo de la factura: "))
21             facturas[numero_factura] = costo_factura
22         elif action == '2':
23             numero_factura = input("Ingresa el numero de factura a pagar: ")
24             if numero_factura in facturas:
25                 costo_factura = facturas.pop(numero_factura)
26                 print(f"Factura {numero_factura} pagada por un total de {costo_factura}")
27             else:
28                 print(f"La factura {numero_factura} no existe.")
29         elif action == '3':
30             sumar_total_facturas = sum(facturas.values())
31             estado_facturas(facturas)
32         elif action == '4':
33             break
34         elif action == 'x':
35             break
36         else:
37             print("ERROR: Por favor, selecciona una accion valida.")
38
39
40
41
42

```

```

Pulsa el numero de una accion: 1. Añadir una nueva factura, 2. Pagar una factura existente,3. Mostrar estado de las facturas o 4 o x par
Ingresa el numero de factura: 200
Ingresa el costo de la factura: 2
Pulsa el numero de una accion: 1. Añadir una nueva factura, 2. Pagar una factura existente,3. Mostrar estado de las facturas o 4 o x par
ERROR: Por favor, selecciona una accion valida.
Pulsa el numero de una accion: 1. Añadir una nueva factura, 2. Pagar una factura existente,3. Mostrar estado de las facturas o 4 o x par
Estado de las facturas:
Total cobrado: 2.0

```

Total pendiente de cobro: 0.0

Pulsa el numero de una accion: 1. Añadir una nueva factura, 2. Pagar una factura existente,3. Mostrar estado de las facturas o 4 o x par

7. Escribir una función a la que se le pase una cadena <nombre> y muestre por pantalla el saludo ¡hola <nombre>!.

```
1 # Escribe aqui tu respuesta
2 def hi(nombre):
3     print(f'Hola {nombre}!:')
4
5 nombre = input('Ingresa Tu NOMBRE')
6 hi(nombre)

Ingresa Tu NOMBRENicolas Acha
Hola Nicolas Acha!:
```

8. Realizar una función que reciba una muestra de números en una lista y devuelva otra lista con sus cuadrados.

```
1 #Logica
2 def sqr(numbers):
3     rsqrt = [i ** 2 for i in numbers ]
4     return rsqrt
5
6 def input_numbers():
7     numbers = []
8
9     while True:
10        number = input("Ingresa un numero o 'x' para calcular los cuadrados: ")
11        if number.lower() == 'x':
12            break
13        try:
14            number = int(number)
15            numbers.append(number)
16        except ValueError:
17            print("ERROR: Utiliza numeros enteros unicamente")
18
19    return numbers
20
21 #Programa
22
23 if __name__ == "__main__":
24     muestra_numeros = input_numbers()
25     cuadrados = sqr(muestra_numeros)
26
27     print("Muestra de numeros:", muestra_numeros)
28     print("Cuadrados de la muestra:", cuadrados)
29
30
```

```
Ingresa un numero o 'x' para calcular los cuadrados: 5
Ingresa un numero o 'x' para calcular los cuadrados: 8
Ingresa un numero o 'x' para calcular los cuadrados: 12
Ingresa un numero o 'x' para calcular los cuadrados: 16
Ingresa un numero o 'x' para calcular los cuadrados: 55
Ingresa un numero o 'x' para calcular los cuadrados: 69
Ingresa un numero o 'x' para calcular los cuadrados: 75
Ingresa un numero o 'x' para calcular los cuadrados: 99
Ingresa un numero o 'x' para calcular los cuadrados: 114
Ingresa un numero o 'x' para calcular los cuadrados: x
Muestra de numeros: [5, 8, 12, 16, 55, 69, 75, 99, 114]
Cuadrados de la muestra: [25, 64, 144, 256, 3025, 4761, 5625, 9801, 12996]
```

✓ 16 s se ejecutó 23:26

