

Universidad Americana

Facultad de Ingeniería y Arquitectura



Algoritmos y Estructuras de Datos

Grupo 4

Manejo de Funciones

Realizado por:

Joaquín Alberto Pérez Zúñiga

Docente:

MSc. César Marín López

Ejercicio #1- Código:

```
1  """
2  Ejercicio de funciones con Python: suma, menor y mayor de los números.
3  """
4
5
6  def suma(lista: list[int]) -> int:
7      """
8      Retorna la suma de todos los números en una lista de enteros.
9      """
10
11     total = 0
12     for num in lista:
13         total += num
14
15     return total
16
17
18 def mayor(lista: list[int]) -> int:
19     """
20     Retorna el número más grande en una lista de enteros.
21     """
22
23     maximo = 0
24     for num in lista:
25         # La funcion max() tambien acepta argumentos de
26         # lista e itera sobre ella de esta misma manera
27         maximo = max(maximo, num)
28
29     return maximo
30
31
```

```

32 def menor(lista: list[int]) -> int:
33     """
34     ...Retorna el número más pequeño en una lista de enteros.
35     ..."""
36
37     ...minimo = lista[0]
38     ...for num in lista:
39         ...# la funcion min() tambien acepta argumentos de
40         ...# lista e itera sobre ella de esta misma manera
41         ...minimo = min(minimo, num)
42
43     ...return minimo
44
45
46 def main() -> None:
47     """
48     ...Ejecución del programa.
49     ..."""
50
51     ...lista = []
52     ...n = int(input("\nIngrese la cantidad de elementos a procesar: "))
53
54     ...print()
55
56     ...for i in range(n):
57         ...lista.append(int(input(f"Ingrese el elemento #{i + 1}: ")))
58
59     ...print(f"\nElementos de la lista: {lista}")
60     ...print(f"Suma de los elementos de la lista: {suma(lista)}")
61     ...print(f"Número más grande de la lista: {mayor(lista)}")
62     ...print(f"Número más pequeño de la lista: {menor(lista)}")
63
64
65 if __name__ == "__main__":
66     ...main()
67

```

Ejercicio #1 – Ouput:

```
C:\dev\algo-structs> python 31-3-25\funciones.py
```

```
Ingresa la cantidad de elementos a procesar: 4
```

```
Ingresa el elemento #1: 1
```

```
Ingresa el elemento #2: 2
```

```
Ingresa el elemento #3: 3
```

```
Ingresa el elemento #4: 4
```

○

```
Elementos de la lista: [1, 2, 3, 4]
```

```
Suma de los elementos de la lista: 10
```

```
Número más grande de la lista: 4
```

```
Número más pequeño de la lista: 1
```

```
C:\dev\algo-structs> █
```

Ejercicio #2 – Código:

```
1  """
2  Desarrollar un programa que cargue los datos de un triángulo.
3  Implementar un método/función para determinar el tipo
4  de triángulo (equilátero, isósceles o escaleno).
5  """
6
7
8  def main() -> None:
9      """
10     Ejecución del programa.
11     """
12
13     tipo = ""
14     triangulo: list[float] = []
15
16     for i in range(3):
17         triangulo.append(float(input(f"Ingrese la longitud del lado #{i + 1}: ")))
18
19     if triangulo[0] == triangulo[1] == triangulo[2]:
20         tipo = "equilátero"
21     elif (
22         triangulo[0] == triangulo[1] or
23         triangulo[1] == triangulo[2] or
24         triangulo[0] == triangulo[2]
25     ):
26         tipo = "isósceles"
27     else:
28         tipo = "escaleno"
29
30     print(f"\nEl triángulo ingresado es {tipo}.")
31
32
33 if __name__ == "__main__":
34     main()
35
```

Ejercicio #2 - Output:

● C:\dev\algo-structs> python 31-3-25\ej2.py

Ingrese la longitud del lado #1: 1.2
Ingrese la longitud del lado #2: 1.2
Ingrese la longitud del lado #3: 3

El triángulo ingresado es isósceles.

● C:\dev\algo-structs> python 31-3-25\ej2.py

Ingrese la longitud del lado #1: 1
Ingrese la longitud del lado #2: 2
Ingrese la longitud del lado #3: 3

El triángulo ingresado es escaleno.

● C:\dev\algo-structs> python 31-3-25\ej2.py

Ingrese la longitud del lado #1: 3
Ingrese la longitud del lado #2: 3
Ingrese la longitud del lado #3: 3

El triángulo ingresado es equilátero.

○ C:\dev\algo-structs> █