

TP3 LISTAS Y EL CICLO FOR

- 1) Implementar un algoritmo que permita crear una lista con tres datos de entrada, el valor inicial, valor final y el paso. Por ejemplo **valorInicial** = 10, **valorFinal** = 30 y **paso** = 5, se debe crear una lista que contenga [10, 15, 20, 25, 30].
- 2) En cada uno de los siguientes incisos, deducir cuál es el criterio para generar la lista de valores indicados. Luego implementar un programa en Python que permita al usuario ingresar valores para las variables **primerValor** y **cantidadValores**, y genere las listas que se muestran.

- a) 6,8,10,12,14,...
- b) 8, -8, 8, -8, 8,...
- c) 9,8,7,6,5,...

- 3) Dada la siguiente lista:

miLista = [4,8,10,-1,0,-8,5,4]

Realizar un algoritmo que haga lo siguiente:

- Encuentre el mayor y el menor en la lista.
- Que me diga la cantidad de elementos que hay en la lista.
- Dado un valor ingresado por el usuario, decirme si está o no en la lista, como por ejemplo el cero.
- Dado una posición ingresado por el usuario, decirme que elemento pertenece en la lista, esta posición tiene que ser ingresada por el usuario.
- Eliminar un valor ingresado por el usuario y me diga la cantidad de elementos que actualmente hay en la lista.
- Encontrar valores duplicados.

Datazo: en este ejercicio se piden muchas cosas, la mayoría de ellos se pudo ver teoría, la idea es ver como resuelven cada inciso y se organizan para que todo funcione de manera fluida, recuerden que cada vez que vamos a usar un **input** deben haberle mostrado un mensaje explicativo al usuario de que le están pidiendo.

- 4) Dada la siguiente lista:

palabra = ["O","D","N","U","M"," ","A","L","O","H"]

Realizar un programa que me invierta la lista, letra a letra.

Datazo: este ejercicio se puede encarar por índices o por las mismas letras, este dato es medio confuso y es así para que piensen en que maneras hay de recorrer una lista usando los ciclos **for**.

- 5) Implementar un programa en Python que le permita al usuario ingresar HASTA un máximo de 10 números, por ejemplo si el usuario piensa ingresar más de 10 números debe mostrar un error, si el usuario piensa en ingresar 5 debe mostrarme el promedio de los datos ingresados y los números que se ingresaron.

Datazo: este es un ejemplo: imaginen que ingreso al programa 7, es decir voy a ingresar 7 números, como por ejemplo [1, 5, 3 ,10, -1, 0, 3], como ingrese solo 7 números debo calcular el promedio de todos los valores ingresados, el cual me da $(1+5+3+10+(-1)+0+3) / 7 = 3$, tres es el promedio, debo mostrarle eso al usuario y además todos los números que ingrese [1, 5, 3 ,10, -1, 0, 3].

- 6) Implementar un programa que solicite al usuario un numero entre 5 y 20, en base al valor ingresado, se deberá generar la siguiente “escalera” de números como se muestra en los ejemplos a continuación:

Si el valor ingresado es 5:	Si el valor ingresado es 7:	Si el valor ingresado es 10:
1	1	1
21	21	21
321	321	321
4321	4321	4321
54321	54321	54321
	654321	654321
	7654321	7654321
		87654321
		987654321
		10987654321

Datazo: para hacer esta escalera lo más practico es ir concatenando los números, recuerden que para concatenar si o si deben ser de tipo STRING.

- 7) En un comercio se generan tres listas por cada carrito de compra, la primera lista nos muestra el producto que se compró, la segunda lista muestra la cantidad del producto que hay en la primera lista, la tercera lista muestra el precio por unidad de cada producto. Para el ejemplo planteado, calcular, cual es el costo total para las siguientes compras:

listaProducto=["Sal", "Jabón", "Fideo", "Arroz", "Pasta de dientes"]

listaCantidad =[1 , 4, 2, 9, 1]

listaPrecioPorUnidad =[45.50 , 34.5 , 98.0 , 68.5, 201.0]

Finalmente mostrar un tiket, de la siguiente forma:

PRODUCTO	CANTIDAD	COSTO
"Sal"	1	45.50
"Jabón"	4	138.0
"Fideo"	2	196.0
"Arroz"	9	616.5
"Pasta de dientes"	1	201.0
=====		
TOTAL		1197.0

Datazo: en este ejercicio está pensado para que vean la manera de cómo generar estructuras que sean visualmente más entendibles para el usuario.

- 8) Implementar un algoritmo que haga lo siguiente:
 - a) Dadas dos listas con distintos elementos crear una lista nueva que estén los elementos que se encuentran en ambas. Por ejemplo:
lista1 = [0, 5, 8, 'perro', 'gato', 'jirafa', 5.7, True]
lista2 = [6, 7, 8, 'miau', 'gato', False, True]
respuesta = [8, 'gato', True]
 - b) hacer otra lista con los elementos que NO comparten
 - c) hacer otra lista con los elementos de lista1 más los de la lista2 que no tiene lista1. Por ejemplo:
 se espera **respuesta = [0, 5, 6, 7, 8, 'perro', 'gato', 'miau', 'jirafa', 5.7, True, False]**. El orden de la lista puede ser cualquiera.

- 9) Se tiene la siguiente lista:
lista = [62, 78, 37, 51, 49, 62, 46, 76, 35, 48, 81, 1, 92, 82, 86]
 - a) generar una lista que contenga los números pares de la lista original.
 - b) generar una lista que contenga los números impares de la lista original

- 10) Se tiene una lista con números ascendentes. Hacer un código que haga una lista con los números faltantes entre el primer número de la lista y el último.
 Ejemplo:
lista1 = [5, 7, 8, 10, 11, 12, 13, 15, 16, 17, 20]
respuesta = [6, 9, 14, 18, 19]
 - a) Repetir el ejercicio pero que solo muestre los números pares

- 11) Crear una lista de productos, otra con sus precios hace un mes y otra con los precios ahora (pueden ser reales o inventados). Hacer un código que calcule la inflación promedio de esta lista de productos. Esto se hace calculando el porcentaje de aumento de cada producto.

- 12) a) Crear un programa que genere dos listas a partir de inputs. Una lista de productos y la otra de precios.
- b) Crear un programa que introduciendo el producto nos devuelva el precio.
- c) A partir de estas listas de precios y productos hacer un código que devuelva el producto más caro, más barato y el más cercano al promedio de precios.
- d) A este programa súmenle la posibilidad de eliminar productos por medio de su nombre. El programa tiene que eliminar el producto y el precio. Van a tener que modificar las dos listas.
- e) Hagan un código aparte que ordene las dos listas según su precio.