

GUÍA DE EJERCICIOS N°1

TEMA: "Introducción a Python"

Resolver los siguientes enunciados. Generar un archivo ".py" por cada ejercicio. De manera que se puedan ejecutar desde el intérprete de comandos de Linux o de Windows. Por ejemplo: la resolución del ejercicio 3 de Strings en Python debería llamarse: eje13.py
Para resolver alguno de los ejercicios vamos a hacer uso de la función "input()" de python que permite la entrada de datos por teclado (entrada standard). Ejemplo:

```
nombre = input("Por favor ingrese su nombre: ")
```

Lo que el usuario ingrese por teclado quedará almacenado en la variable nombre como un string.

1. Strings en Python

- 1.1. Realizar la carga del nombre de una persona y luego mostrar el primer caracter del nombre y la cantidad de letras que lo componen.
- 1.2. Solicitar la carga del nombre de una persona en minúsculas. Mostrar un mensaje si comienza con vocal dicho nombre.
- 1.3. Ingresar un mail por teclado. Verificar si el string ingresado contiene solo un caracter "@". Mostrar un mensaje informando dicha condición.
- 1.4. Inicializar un string con la cadena "mAriA" luego llamar a sus métodos upper(), lower() y capitalize(), guardar los datos retornados en otros string y mostrarlos por pantalla.
- 1.5. Cargar una oración por teclado. Mostrar luego cuantos espacios en blanco se ingresaron. Tener en cuenta que un espacio en blanco es igual a " ", en cambio una cadena vacía es "".
- 1.6. Ingresar una oración que pueden tener letras tanto en mayúsculas como minúsculas. Contar la cantidad de vocales. Crear un segundo string con toda la oración en minúsculas para que sea más fácil disponer la condición que verifica que es una vocal.

- 1.7. Solicitar el ingreso de una clave por teclado y almacenarla en una cadena de caracteres. Controlar que el string ingresado tenga entre 10 y 20 caracteres para que sea válido, en caso contrario mostrar un mensaje de error.

2. Listas en Python

- 2.1. Definir una lista que almacene 5 enteros. Sumar todos sus elementos y mostrar dicha suma.
- 2.2. Definir una lista por asignación que almacene los nombres de los primeros cuatro meses de año. Mostrar el primer y último elemento de la lista solamente.
- 2.3. Definir una lista por asignación que almacene en la primer componente el nombre de un alumno y en las dos siguientes sus notas. Imprimir luego el nombre y el promedio de las dos notas.
- 2.4. Definir por asignación una lista con 8 elementos enteros. Contar cuantos de dichos valores almacenan un valor superior a 100.
- 2.5. Definir una lista que almacene por asignación los nombres de 5 personas. Contar cuantos de esos nombres tienen 5 o más caracteres.
- 2.6. Definir una lista vacía y luego solicitar la carga de 5 enteros por teclado y añadirlos a la lista. Imprimir la lista generada.
- 2.7. Realizar la carga de valores enteros por teclado, almacenarlos en una lista. Finalizar la carga de enteros al ingresar el cero. Mostrar finalmente el tamaño de la lista.
- 2.8. Cargar por teclado y almacenar en una lista las alturas de 5 personas (valores float)
- 2.9. Obtener el promedio de las mismas. Contar cuántas personas son más altas que el promedio y cuántas más bajas.
- 2.10. Una empresa tiene dos turnos (mañana y tarde) en los que trabajan 8 empleados (4 por la mañana y 4 por la tarde) Confeccionar un programa que permita almacenar los sueldos de los empleados agrupados en dos listas. Imprimir las dos listas de sueldos.
- 2.11. Crear y cargar una lista con 5 enteros. Implementar un algoritmo que identifique el mayor valor de la lista.
- 2.12. Crear y cargar una lista con 5 enteros por teclado. Implementar un algoritmo que identifique el menor valor de la lista y la posición donde se encuentra.
- 2.13. Ingresar por teclado los nombres de 5 personas y almacenarlos en una lista. Mostrar el nombre de persona menor en orden alfabético.

- 2.14. Cargar una lista con 5 elementos enteros. Imprimir el mayor y un mensaje si se repite dentro de la lista (es decir si dicho valor se encuentra en 2 o más posiciones en la lista)
- 2.15. Se debe crear y cargar una lista donde almacenar 5 sueldos. Desplazar el valor mayor de la lista a la última posición.
- 2.16. Se debe crear y cargar una lista donde almacenar 5 sueldos. Ordenar de menor a mayor la lista.
- 2.17. Crear una lista por asignación. La lista tiene que tener 2 elementos. Cada elemento debe ser una lista de 5 enteros. Calcular y mostrar la suma de cada lista contenida en la lista principal.
- 2.18. Crear una lista por asignación. La lista tiene que tener 5 elementos. Cada elemento debe ser una lista, la primera lista tiene que tener un elemento, la segunda dos elementos, la tercera tres elementos y así sucesivamente. Sumar todos los valores de las listas.
- 2.19. Se tiene la siguiente lista:

lista=[[4,12,5,66], [14,6,25], [3,4,5,67,89,23,1], [78,56]]

Imprimir la lista. Luego fijar con el valor cero todos los elementos mayores a 10 contenidos en todos los elementos de la variable "lista". Volver a imprimir la lista.