

Guía de ejercicios 7 - listas y cadenas

Ejercicio 1:

1. A partir de una cadena de texto iniciada en el código, mostrar la cadena en color azul.
2. Crear un nuevo_texto con los caracteres que sean solo letras y espacios, mostrar nuevo_texto en color blanco (aplicar **join** combinado con sintaxis por comprensión para filtrar o **filter**).
3. Convertir el texto en una **lista** de palabras y mostrar todos los elementos y su posición en color blanco (**enumerate**).
4. Encontrar y mostrar el elemento menor, mayor y su posición en la lista de palabras (**min**, **max** e **index**, min y max deben proporcionar el argumento key = len para comparar cadenas).
5. Mostrar la lista con formato de texto en color blanco, una palabra al lado de la otra separada por un espacio (**join**).
6. Mostrar la palabra menor en rojo en cada aparición, contar cuantas apariciones tiene.
7. Mostrar la palabra mayor en verde en cada aparición, contar cuantas apariciones tiene.
8. Mostrar cuántas palabras menores y cuántas mayores se encontraron.

Ejercicio 2:

1. Generar una lista de 30 elementos enteros al azar entre 50 y 250 inclusivos y guardarlos en una lista v1 (**random.sample**).
2. Hacer una copia en una segunda lista v2 con los valores del primero que se encuentren entre 75 y 225 (aplicar sintaxis por comprensión).
3. Crear una función para mostrar todos los elementos de una lista cualquiera y su posición, llamarla desde el main para mostrar v1 y v2.
4. Crear una función para encontrar el mayor, menor y el promedio de una lista cualquiera, llamar desde el main para mostrar los resultados de v1 y v2.

Ejercicio 3:

1. A partir de una cadena de texto inicializada en el código, convertir a una lista p de palabras y mostrar.

2. Mostrar el elemento menor (en cantidad de caracteres), mayor (en cantidad de caracteres) y su posición en la lista p (aplicar min y max con el parámetro key = len) ejemplo: menor = min(palabras, key = len).
3. Concatenar los elementos de la lista en una nueva cadena nuevo_texto pero en sentido inverso, es decir donde el primer elemento sea el último elemento de p y mostrar al finalizar (aplicar **join** con for **reversed**).
4. Crear una función para mostrar la lista p, llamar desde main.

Ejercicio 4:

1. Crear un código para ingresar n valores decimales entre 1 y 100, salir con cero.
2. Guardar los valores ingresados en una lista v1 y hacer una copia en una segunda lista v2 con los valores del primero que sean mayor a 50 (sintaxis por comprensión).
3. Crear una función para mostrar todos los elementos de una lista cualquiera y su posición, llamarla desde el main para mostrar v1 y v2.
4. Crear una función para encontrar el mayor menor y el promedio de una lista cualquiera, llamar desde el main para mostrar los resultados de v1 y v2.

Ejercicio 5:

1. Permitir el ingreso de una serie de temperaturas entre -60 y 60 (no debe permitir valores fuera del rango).
2. Salir con 100.
3. Obtener la cantidad y el promedio de las lecturas menores a cero grados.
4. Obtener la cantidad y el promedio de las lecturas mayores o igual a cero grados.
5. Obtener la lectura menor y mayor de la serie.

Ejercicio 6:

1. A partir de una cadena de texto (no pueden ser todas mayúsculas) realizar cada punto y mostrar el resultado en pantalla.
2. Colocar la palabra en una lista (**Split**).
3. Desarrollar un bucle **while** para el ingreso de una palabra a buscar en la lista.
4. Salir cuando no se ingrese ningún valor (Enter).
5. Buscar la palabra en la lista y mostrar si se encontró o no, y su posición en la lista.
6. Unir la lista en una nueva cadena con el primer carácter en mayúscula y el resto como está originalmente (**Join, capitalize**).

Ejercicio 7:

1. Generar una lista de 30 números enteros al azar de 2 dígitos.
2. Mostrar cada elemento y su posición (**enumerate**).
3. Encontrar el elemento menor y su posición por arriba del valor 50 (aplicar sintaxis por comprensión o **filter**).
4. Encontrar el elemento mayor y su posición por debajo del valor 50 (aplicar sintaxis por comprensión o **filter**).