Batería ejercicios listas

Ejercicio 1: Creación y Acceso a Listas

- Crea una lista llamada nombres que contenga los nombres de tres compañeros de clase.
- Imprime el segundo nombre de la lista.
- Agrega tu nombre a la lista y muestra la lista actualizada.

Ejercicio 2: Modificación de Listas

- Dada la siguiente lista de números: numeros= [1, 2, 3, 4, 5], cambia el tercer elemento (3) por 10.
- Elimina el último número de la lista y muestra la lista resultante.

Ejercicio 3: Bucles y Listas

- Crea una lista llamada números que contenga los números del 1 al 10.
- Utiliza un bucle for para imprimir cada número de la lista en una línea separada.

Ejercicio 4: Filtrado de Listas

- Dada la lista de números: numeros= [1, 2, 3, 4, 5, 6, 7, 8, 9, 10], crea una nueva lista llamada pares que contenga solo los números pares de la lista original.
- Imprime la lista pares.

Ejercicio 5: Suma de Listas

- Dada una lista de números: numeros = [1, 2, 3, 4, 5], calcula la suma de todos los números en la lista y muestra el resultado.

Ejercicio 6: Matrices y Listas Anidadas

- Crea una lista que represente una matriz 3x3 (una lista de listas) y llámala matriz.
- Imprime la matriz en forma de tabla.
- Calcula la suma de todos los elementos en la matriz.

Ejercicio 7: Listas y Funciones

- Define una función llamada promedio_lista que tome una lista de números como argumento y devuelva el promedio de los números en la lista.
- Utiliza la función para calcular el promedio de una lista de números de tu elección.

Ejercicio 8: Listas y Condicionales

- Crea una lista de calificaciones de alumnos.
- Escribe un programa que determine cuántos alumnos aprobaron (calificación >= 5) y cuántos reprobaron (calificación < 5).
- Imprime el número de aprobados y reprobados.

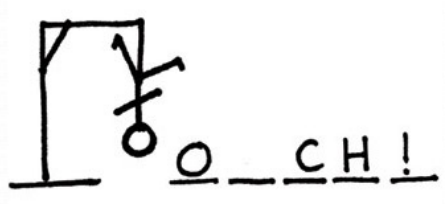
Ejercicio 9: Listas y Ordenamiento

- Crea una lista de nombres desordenados.
- Utiliza el método sort() para ordenar la lista en orden alfabético.
- Imprime la lista ordenada.

Ejercicio 10: Listas y Búsqueda

- Crea una lista de palabras.
- Escribe un programa que permita al usuario ingresar una palabra y determine si esa palabra está en la lista.
- Muestra un mensaje indicando si la palabra fue encontrada o no.

Ejercicio Extra : Juego del Ahorcado en Python



El objetivo de este ejercicio es crear un juego del ahorcado en el que un jugador intente adivinar una palabra oculta. La palabra oculta se almacenará en una lista de caracteres y se mostrará como una serie de guiones bajos al principio. El jugador tendrá un número limitado de intentos para adivinar la palabra completa antes de ser "ahorcado".

Instrucciones:

- Define una lista llamada **palabra_oculta** que contenga la palabra que el jugador debe adivinar. Puedes elegir cualquier palabra secreta que desees, pero asegúrate de convertirla en minúsculas para evitar problemas de mayúsculas/minúsculas durante la comparación.

- Crea una lista llamada adivinanza que inicialmente contenga guiones bajos (_) en lugar de letras para representar la palabra oculta. Por ejemplo, si la palabra oculta es "python", la lista adivinanza sería ['_', '_', '_', '_', '_', '_'].

- Establece un número máximo de intentos (por ejemplo, 6).

- Inicia un bucle que solicite al jugador que ingrese una letra.

- Verifica si la letra ingresada está en la palabra_oculta. Si es así, actualiza la lista adivinanza para mostrar la letra en la posición correcta. Si la letra no está en la palabra oculta, reduce el número de intentos disponibles.

- Muestra la lista adivinanza actualizada y el número de intentos restantes después de cada intento del jugador.

- Continúa el bucle hasta que el jugador adivine la palabra o se quede sin intentos.

- Si el jugador adivina la palabra correctamente, muestra un mensaje de felicitaciones. Si se queda sin intentos, muestra un mensaje de que el jugador ha sido "ahorcado" y revela la palabra oculta.

- Pregunta al jugador si desea jugar de nuevo y, si lo desea, reinicia el juego con una nueva palabra oculta.

Última prueba
Retoca el juego para que el jugador tenga **5 vidas** y cada vez que diga una letra equivida se le reste una vida. Si sus vidas llegan a 0 , el programa acabará indicando al jugador que ha perdido.