

Ejercicios Python

1. **Introducción a Variables y Tipos de Datos:** Solicite al usuario que ingrese su nombre y edad. Posteriormente, genere un mensaje que diga: "Estimado [nombre], el año próximo usted tendrá [edad+1] años".
2. **Estructuras de Control - Condicionales:** Desarrolle un programa que requiera del usuario la introducción de un número. Si el número es positivo, debe mostrar el mensaje "El número es positivo". Si es negativo, "El número es negativo". Y si es cero, "El número es cero".
3. **Estructuras de Control - Bucles con For:** Implemente un programa que imprima los números del 1 al 10 empleando un bucle for.
4. **Estructuras de Control - Bucles con While:** Diseñe un programa que solicite la introducción de números de manera iterativa. Este programa deberá sumar los números ingresados y finalizar cuando la suma exceda 100, mostrando el total acumulado.
5. **Funciones:** Defina una función llamada **maximo** que reciba dos números como argumentos y retorne el mayor de ellos. No se debe utilizar la función **max()** de Python para este ejercicio.
6. **Listas y Bucles:** Elabore un programa que solicite la entrada de 5 nombres, almacénelos en una lista y luego los imprima en orden inverso.
7. **Diccionarios:** Cree un diccionario que contenga información sobre un libro, incluyendo: título, autor, año de publicación y género. Desarrolle un programa que imprima cada detalle en líneas separadas.
8. **Manipulación de Cadenas:** Solicite al usuario que ingrese una frase. El programa debe contar cuántas veces aparece cada vocal en la frase e imprimir los resultados.
9. **Listas por comprensión:** Dada una lista de números, utilice una comprensión de lista para generar una nueva lista que contenga únicamente los números pares de la lista original.
10. **Trabajo con Archivos:** Elabore un programa que cree un archivo denominado "numeros.txt", escriba los números del 1 al 20 (uno por línea) y luego lea el archivo e imprima su contenido.
11. **Introducción a las Clases:** Desarrolle una clase llamada **Auto**, que contenga atributos como **marca**, **modelo** y **kilometraje**. Incluya un método **mostrar_descripcion()** que imprima la información del auto y otro método **viajar()** que acepte una cantidad de kilómetros y los añada al **kilometraje** del auto.
12. **Herencia:** Extienda la clase **Auto** para crear una nueva clase **Camioneta**, que herede de **Auto** pero añada un atributo adicional llamado **capacidad_carga** (en kilogramos). Incluya un método **verificar_carga()** que determine si un peso dado puede ser transportado.
13. **Polimorfismo:** Implemente una clase base **Animal** con un método **hacer_sonido()** que imprima "Este animal hace un sonido". Luego, desarrolle clases derivadas **Perro** y **Gato**,

sobrescribiendo el método **hacer_sonido()** para que imprima "Guau" y "Miau", respectivamente.

14. **Composición:** Construya una clase **Libreria** que contenga una lista de instancias de otra clase, **Libro**, la cual debe tener atributos como **titulo**, **autor** y **año_publicacion**. La clase **Libreria** debe incluir métodos para agregar y remover libros, además de un método para mostrar todos los libros disponibles.
15. **Manejo de Excepciones:** Desarrolle una función **dividir** que tome dos números y devuelva el resultado de su división. Implemente un manejo de excepciones para capturar la división por cero, mostrando un mensaje apropiado en tal caso.