

Javascript challenges

Valores, variables, tipos de datos, operadores básicos

1. Calculadora de BMI (Índice de Masa Corporal)

En este desafío, crearás un programa que calcula el Índice de Masa Corporal (BMI, por sus siglas en inglés) a partir del peso y la altura de una persona. Aquí están los requerimientos:

- Crear dos variables para almacenar el peso (en kilogramos) y la altura (en metros). Estos serán los datos de entrada para tu programa.
- Calcula el BMI. El BMI se calcula con la fórmula $\text{peso} / \text{altura}^2$.
- Imprime el BMI. Finalmente, imprime el BMI calculado.
- Interpreta el BMI. Según los estándares de la Organización Mundial de la Salud, un BMI menor a 18.5 se considera bajo peso, entre 18.5 y 24.9 se considera normal, entre 25 y 29.9 se considera sobrepeso, y 30 o más se considera obesidad. Imprime un mensaje correspondiente al rango en el que se encuentra el BMI calculado.

2. Conversor de temperaturas

En este desafío, crearás un programa que pueda convertir grados Celsius a Fahrenheit y viceversa. Aquí están los requerimientos:

- Crea dos variables para almacenar la temperatura en grados Celsius y Fahrenheit:
 - Una variable (por ejemplo, `temperaturaCelsius`) representará la temperatura en grados Celsius.
 - Otra variable (por ejemplo, `temperaturaFahrenheit`) representará la temperatura en grados Fahrenheit.
- Calcula y almacena la conversión de la temperatura:
 - Para convertir de Celsius a Fahrenheit, usa la fórmula $F = C * 9/5 + 32$ y almacena el resultado en `temperaturaFahrenheit`.
 - Para convertir de Fahrenheit a Celsius, usa la fórmula $C = (F - 32) * 5/9$ y almacena el resultado en `temperaturaCelsius`.
- Imprime las temperaturas convertidas:
 - Imprime la temperatura en grados Fahrenheit.
 - Imprime la temperatura en grados Celsius.

3. Generador de Nombres de Usuario

Para este desafío, crearás un programa que generará un nombre de usuario único a partir de un nombre y un apellido. Aquí están los requerimientos:

- Crea dos variables para almacenar el nombre y el apellido. Estos serán los datos de entrada para tu programa.
- Genera el nombre de usuario. Un enfoque común para la generación de nombres de usuario es combinar partes del nombre y el apellido y añadir un número al final. Por ejemplo,

podrías tomar las tres primeras letras del nombre, las tres primeras letras del apellido, y un número aleatorio para crear un nombre de usuario.

- Imprime el nombre de usuario. Finalmente, imprime el nombre de usuario generado para mostrar el resultado de tu programa.

4. Juego de Adivinanzas

En este desafío, vas a crear un simple juego de adivinanzas. Aquí están los requerimientos:

- Genera un número aleatorio entre 1 y 10: Este será el número que el jugador debe adivinar.
- Crea una variable para almacenar la suposición del jugador: Para este desafío, puedes asignar un valor fijo a esta variable. Sin embargo, en un programa real, probablemente obtendrías este valor de alguna entrada del usuario.
- Comprueba si la suposición del jugador es correcta:
 - Si el jugador adivina el número, imprime un mensaje de felicitación.
 - Si la suposición es demasiado alta, imprime un mensaje que indique que la suposición es demasiado alta.
 - Si la suposición es demasiado baja, imprime un mensaje que indique que la suposición es demasiado baja.
- Permite al jugador tener un máximo de 3 intentos para adivinar el número. Si no adivinan el número en 3 intentos, informa al jugador que ha perdido y revela el número.

5. Agenda Telefónica

En este desafío, crearás una simple agenda telefónica. Aquí están los requerimientos:

- Crear un objeto para almacenar los contactos de la agenda telefónica. Cada contacto estará representado por un par de propiedades: el nombre de la persona y su número de teléfono.
- Añadir algunos contactos a la agenda. Cada contacto debe ser añadido como una nueva propiedad del objeto de la agenda. El nombre de la persona será la clave, y el número de teléfono será el valor.
- Crea una variable que almacene el nombre de un contacto para buscar.
- Buscar un contacto en la agenda. Si el contacto existe, imprime su número de teléfono. Si no existe, imprime un mensaje indicando que el contacto no se encontró.

6. Simulador de Lanzamiento de Dados

Para este desafío, vas a crear un programa que simula el lanzamiento de un par de dados. Aquí están los requerimientos:

- Crear una variable para cada dado. Cada dado puede tener un valor entre 1 y 6, que puedes generar utilizando una función para obtener números aleatorios.
- Lanzar los dados. Para simular el lanzamiento de los dados, asigna a cada variable un número aleatorio entre 1 y 6.
- Calcula la suma de los valores de los dados.

- Imprime los valores de los dados y su suma. Tu programa debería imprimir un mensaje que muestre los valores de cada dado y la suma de ambos.

7. Validador de Dirección de Correo Electrónico

En este desafío, crearás un programa que valide una dirección de correo electrónico. Aquí están los requerimientos:

- Crear una variable para almacenar la dirección de correo electrónico. Esta será la dirección de correo electrónico que tu programa comprobará.
- Comprobar si la dirección de correo electrónico es válida. Para ser válida, una dirección de correo electrónico debe tener exactamente un símbolo de arroba (@), al menos un punto después del símbolo de arroba, y al menos un carácter antes del símbolo de arroba y después del punto. Este es un chequeo básico y no cubre todas las posibilidades de formatos de correo electrónico válidos, pero servirá para este desafío.
- Imprime un mensaje indicando si la dirección de correo electrónico es válida o no.

8. Calculadora de Propinas

En este desafío, vas a crear un programa que calcula cuánto deberías dejar de propina en un restaurante. Aquí están los requerimientos:

- Crear una variable para almacenar el total de la factura.
- Crear una variable para almacenar el porcentaje de propina que quieres dejar. Esto podría ser un número fijo (por ejemplo, siempre podrías dejar el 15% de propina), o podrías ajustarlo dependiendo de la calidad del servicio.
- Calcula la propina. La propina se calcula como $\text{factura} * \text{porcentaje} / 100$.
- Imprime la propina. Finalmente, imprime la propina calculada.

9. Cifrado César

En este desafío, crearás un programa que implemente el cifrado César, una técnica simple de cifrado de sustitución en la que cada letra en el texto original es reemplazada por una letra cierto número de posiciones más adelante en el alfabeto. Aquí están los requerimientos:

- Crear una variable para almacenar el mensaje a cifrar.
- Crear una variable para almacenar el desplazamiento. Este es el número de posiciones que cada letra del mensaje será desplazada en el alfabeto.
- Cifra el mensaje. Para cada letra del mensaje, encuentra la letra que está desplazada en el alfabeto y reemplaza la original por ella. Ten en cuenta que debes tratar de manera especial los espacios y otros caracteres no alfabéticos.
- Imprime el mensaje cifrado.

10. Validador de tarjetas de crédito.

En este desafío, vas a crear un validador de tarjetas de crédito utilizando el algoritmo de Luhn, también conocido como "fórmula de módulo 10". Esta es una fórmula sencilla utilizada para validar varios números de identificación; las tarjetas de crédito y débito son un ejemplo común. Aquí están los requerimientos:

- Crear una función que acepte un número de tarjeta de crédito: Esta función deberá convertir el número a un array de dígitos.
- Implementar el algoritmo de Luhn:
 - Comenzando por el segundo dígito desde la derecha, duplica cada segundo dígito.
 - Si el doble de un dígito resulta ser un número de dos dígitos, suma esos dos dígitos para obtener un número de un solo dígito.
 - Suma todos los dígitos.
 - Si la suma total termina en 0 (es decir, es divisible por 10), entonces el número es válido.
- Probar la función: Prueba la función con varios números de tarjetas de crédito para asegurarte de que funciona correctamente. Recuerda usar solo números de tarjetas de crédito de ejemplo, nunca uses números de tarjetas de crédito reales para estas pruebas.