EJERCICIOS DE PROGRAMACIÓN EN JAVASCRIPT

1. VALIDACIÓN DE USUARIO Y CONTRASEÑA

Crea una función que reciba un nombre de usuario y una contraseña. Si el usuario es "admin" y la contraseña es "1234", la función debe retornar "Acceso concedido", de lo contrario, "Acceso denegado".

Objetivo:
Comprobar si el usuario y la contraseña ingresados coinciden con los valores establecidos en la aplicación.

Razonamiento:

- Necesitamos comparar dos valores de entrada (usuario y contraseña) con credenciales predefinidas.
- Si coinciden, devolvemos "Acceso concedido", de lo contrario, "Acceso denegado".
- Se usa una simple estructura condicional (if).

2. GENERADOR DE SLUG PARA URLS

Escribe una función que reciba un título de una publicación (cadena de texto) y lo convierta en un "slug" para URL. Ejemplo:

```
Entrada: "Aprendiendo JavaScript desde cero"
Salida: "aprendiendo-javascript-desde-cero"
```

Objetivo:
Transformar un título de texto en una URL amigable ("slug").

Razonamiento:

- Para mejorar el SEO en las URLs, convertimos el texto en minúsculas.
- Reemplazamos los espacios con guiones (-) para mejorar la legibilidad en la URL.
- Eliminamos los acentos (normalize + regex) para evitar caracteres especiales en la URL.

3. ALMACENAR PREFERENCIAS DEL USUARIO

Crea un map donde se guarden las preferencias de idioma y color de un usuario para una futura personalización de la web. Luego, permite cambiar los valores y mostrar el resultado final.

Objetivo:
Guardar las preferencias del usuario (idioma, tema, etc.) en un Map.

Razonamiento:

- Los mapas (Map) permiten asociar claves (idioma, tema) con valores (español, oscuro).
- Se pueden actualizar fácilmente, permitiendo que el usuario cambie de preferencia.

4. FILTRO DE PRODUCTOS

Dado un array de productos con precios, crea una función que filtre y devuelva solo los productos con un precio inferior a 50€.

```
const productos = [
  { nombre: "Teclado", precio: 30 },
  { nombre: "Ratón", precio: 25 },
  { nombre: "Monitor", precio: 200 },
  { nombre: "USB", precio: 15 }
];
```

Objetivo:
Filtrar un array de productos y devolver solo los que tienen un precio inferior a 50€.

Razonamiento:

- Usamos filter() para recorrer el array y devolver solo los productos cuyo precio sea menor a 50.

5. CONVERSOR DE FORMATO DE TEXTO

Crea una función que reciba una cadena y una opción para convertir el texto:

- "MAYUSCULAS" → Todo en mayúsculas
- "minusculas" → Todo en minúsculas
- "Capitalizado" → Primera letra de cada palabra en mayúscula

Objetivo:
Permitir que un usuario cambie el formato de una cadena de texto (mayúsculas, minúsculas, capitalizado).

Razonamiento:

- toUpperCase() para convertir todo en mayúsculas.
- toLowerCase() para convertir todo en minúsculas.
- Para capitalizar cada palabra, dividimos el texto en palabras (split), convertimos la primera letra en mayúscula (charAt(0).toUpperCase()) y luego unimos todo (join(' ')).

6. COMPROBADOR DE DISPONIBILIDAD DE USUARIO

Crea un set con nombres de usuario registrados en una web. Luego, haz una función que reciba un nuevo nombre de usuario y verifique si está disponible.

Objetivo:
Verificar si un nombre de usuario ya está registrado en un Set.

Razonamiento:

- Un Set es ideal porque almacena valores únicos.
- Usamos .has() para comprobar si el usuario ya existe.

7. CONTADOR DE CARACTERES PARA META DESCRIPTION

Para mejorar el SEO de una página web, las meta descriptions no deben superar los 160 caracteres. Crea una función que reciba una cadena y devuelva si la descripción es válida o si excede el límite.

Objetivo:
Determinar si una descripción es válida para SEO (máximo 160 caracteres).

Razonamiento:

- Usamos length para contar los caracteres.
- Un operador ternario decide si la descripción es válida o demasiado larga.

8. SIMULACIÓN DE UN CARRITO DE COMPRAS

Crea una función que reciba un array de productos agregados al carrito y calcule el total. Luego, permite eliminar un producto por nombre y actualiza el total.

Objetivo:
Calcular el total de un carrito de compras y permitir eliminar productos.

Razonamiento:

- reduce() suma los precios de los productos en el carrito.
- filter() elimina un producto del carrito al excluirlo por su nombre.

9. GENERADOR DE RESUMEN DE COMENTARIOS

Dado un array con comentarios de usuarios en una web, crea una función que devuelva un resumen con el número total de comentarios y cuántos son positivos (contienen la palabra "bueno") y negativos (contienen la palabra "malo").

```
const comentarios = [
  "Este producto es muy bueno",
  "La calidad es mala",
  "Me encantó, es un buen producto",
  "No me gustó, es malo"
];
```

Objetivo:
Contar el total de comentarios y cuántos son positivos o negativos.

Razonamiento:

- filter() cuenta los comentarios que incluyen "bueno" (positivos) o "malo" (negativos).

10. CONTROL DE ACCESOS A UNA PÁGINA WEB

Crea un sistema de roles con un map, donde cada usuario tenga un rol asignado ("admin", "editor", "visitante"). Luego, haz una función que, dado un nombre de usuario, devuelva qué permisos tiene en la web.

Objetivo:
Determinar qué permisos tiene un usuario según su rol.

Razonamiento:

- Map almacena roles con sus permisos.
- Usamos .get() para obtener los permisos del usuario.