

Ejercicio 1: Filtrar productos por disponibilidad

Enunciado:

Tienes un objeto donde las claves son nombres de productos y los valores son booleanos que indican si el producto está en stock o no. Usa `Object.entries` para filtrar el objeto y obtener solo los productos disponibles. Crear un nuevo objeto `availableProducts` con los productos disponibles

```
const products = { "Laptop": true, "Smartphone": false, "Tablet": true, "Monitor": false };

// Resultado esperado: { "Laptop": true, "Tablet": true }
```

Ejercicio 2: Convertir datos en formato clave-valor a un array de strings

Enunciado:

Tienes un objeto que representa los resultados de una encuesta, con las claves como preguntas y los valores como respuestas. Usa `Object.entries` para convertir este objeto en un array de strings, donde cada string tiene el formato "Pregunta: Respuesta".

```
const surveyResults = { "favoriteColor": "Blue", "likesPizza": "Yes", "ageGroup": "25-34" };

// Resultado esperado: ["favoriteColor: Blue", "likesPizza: Yes", "ageGroup: 25-34"]
```

Ejercicio 3: Contar la cantidad de valores únicos

Enunciado:

Tienes un objeto que representa las calificaciones de un grupo de estudiantes en un examen, con sus nombres como claves y las calificaciones como valores. Usa `Object.values` para contar cuántas calificaciones únicas hay.

```
const grades = { "Alice": 85, "Bob": 75, "Carol": 85, "David": 90, "Eve": 75 };

// Resultado esperado: 3 (porque hay tres calificaciones únicas: 85, 75, 90)
```

Ejercicio 4: Calcular el total de precios

Enunciado:

Tienes un objeto que representa un carrito de compras, donde las claves son los nombres de los productos y los valores son los precios de esos productos. Usa `Object.entries` para calcular el precio total de todos los productos en el carrito.

```
const cart = { "Laptop": 1000, "Smartphone": 700, "Tablet": 300, "Monitor": 200 };

// Resultado esperado: 2200
```

Ejercicio 5: Listar nombres de empleados con salario alto

Enunciado:

Tienes un objeto donde las claves son los nombres de los empleados y los valores son sus salarios. Usa `Object.keys` para listar los nombres de los empleados que tienen un salario superior a 5000.

```
const employees = { "John": 4500, "Jane": 7000, "Mike": 3000, "Alice": 5500, "Bob": 4800 };
```

```
// Resultado esperado: ["Jane", "Alice"]
```

Ejercicio 6: Crear un Resumen de Producto con Contador de Caracteres

Enunciado:

Tienes un objeto donde las claves son nombres de productos y los valores son descripciones. Usa `Object.keys` para crear un nuevo objeto que contenga el nombre del producto y el número de caracteres en la descripción.

```
const products = { "Laptop": "A high performance laptop.", "Smartphone": "A latest model smartphone.", "Tablet": "A lightweight tablet." };
```

```
// Resultado esperado: // { "Laptop": 24, "Smartphone": 25, "Tablet": 21 }
```

Ejercicio 7: Arrays de colores

```
Let colores = [ { nombre: "rojo", rgb: "rgb(255, 0, 0)" }, { nombre: "verde", rgb: "rgb(0, 255, 0)" }, { nombre: "azul", rgb: "rgb(0, 0, 255)" }, { nombre: "amarillo", rgb: "rgb(255, 255, 0)" }, { nombre: "cian", rgb: "rgb(0, 255, 255)" }, { nombre: "magenta", rgb: "rgb(255, 0, 255)" }, { nombre: "blanco", rgb: "rgb(255, 255, 255)" }, { nombre: "negro", rgb: "rgb(0, 0, 0)" }, { nombre: "gris", rgb: "rgb(128, 128, 128)" }, { nombre: "naranja", rgb: "rgb(255, 165, 0)" }, { nombre: "morado", rgb: "rgb(128, 0, 128)" }, { nombre: "rosado", rgb: "rgb(255, 192, 203)" }, { nombre: "marrón", rgb: "rgb(165, 42, 42)" }, { nombre: "beige", rgb: "rgb(245, 245, 220)" }, { nombre: "turquesa", rgb: "rgb(64, 224, 208)" }, { nombre: "lila", rgb: "rgb(200, 162, 200)" }, { nombre: "oliva", rgb: "rgb(128, 128, 0)" }, { nombre: "salmon", rgb: "rgb(250, 128, 114)" }, { nombre: "púrpura", rgb: "rgb(128, 0, 128)" }, { nombre: "teal", rgb: "rgb(0, 128, 128)" }, { nombre: "azul claro", rgb: "rgb(173, 216, 230)" } ];
```

```
// Resultado esperado: console.log(colorExiste("rojo", "nombre")); // true
```

Crear una función que:

- Reciba por parámetro una key y un valor
- Debe asegurarse que la key exista
- Retorne si existe o no el color

Ejercicio 8: Equipos de magos

// Listas de magos para cada equipo

```
let equipo1 = ["Harry Potter", "Hermione Granger", "Ron Weasley", "Draco Malfoy", "Neville Longbottom", "Luna Lovegood", "Ginny Weasley", "Severus Snape", "Sirius Black", "Remus Lupin"];
let equipo2 = ["Luna Lovegood", "Draco Malfoy", "Harry Potter", "Gilderoy Lockhart", "Sirius Black", "Albus Dumbledore", "Minerva McGonagall", "Hermione Granger", "Ron Weasley", "Rubeus Hagrid"];
```

- Crear el objeto: magos contiene las dos listas bajo las propiedades equipo1 y equipo2.
- Crea una función que a partir de **dos** equipos:
 - Reciba por parámetros el objeto magos
 - Usa Object.values(magos) para obtener las listas de magos.
 - Devuelve un objeto con las tres listas requeridas.

// Resultado esperado:

```
magos { equipo1SinRepetidos: [ 'Neville Longbottom', 'Ginny Weasley', 'Severus Snape', 'Remus Lupin' ], equipo2SinRepetidos: [ 'Gilderoy Lockhart', 'Albus Dumbledore', 'Minerva McGonagall', 'Rubeus Hagrid' ], repetidos: [ 'Harry Potter', 'Hermione Granger', 'Ron Weasley', 'Draco Malfoy', 'Luna Lovegood', 'Sirius Black' ] }
```

Ejercicio 9: Filtrado por Key

```
let studentGrades = { John: 85, Jane: 92, Jim: 78, Jill: 88 }
```

// Resultado esperado { Jane: 92 }

- Crear una función filterByKey que reciba por parámetros el objeto studentGrades y una key
- Debe retornar un nuevo objeto que contenga solamente la key,valor encontrada, si no existe retornar "No se ha encontrado la key"
- La función debe ser dinámica, es decir, debo poder pasarle cualquier valor en key