

Ejercicios Propuestos 4

Ejercicios Practica 04-07 JS Métodos Arrays para Programación funcional con funciones callback. filter,flat,flatMap,forEach,map

Pagina 24 Apuntes

| | |
|--|----|
| Categoria: filtros..... | 22 |
| 25. Array.filter(funcioncallback(currentValue, index, array),Argumentos)..... | 22 |
| 26. Array.flat(arrayMultidim)..... | 23 |
| 27 .Array.flatMap(funcioncallback(currentValue, index, array))..... | 23 |
| 28.Array.forEach(funcioncallback(currentValue, index, array))..... | 23 |
| 29.Array.map(funcioncallback(currentValue, index, array))..... | 24 |

Estos ejercicios muestran cómo utilizar funciones como Array.filter(), Array.flat(), Array.flatMap(), Array.forEach(), y Array.map() para resolver problemas comunes en la gestión de datos, como filtrado, transformación y procesamiento de arrays. Estas funciones son esenciales para el manejo de colecciones de datos en JavaScript, y los ejercicios están contextualizados en ejemplos de la vida real para ayudarte a aplicarlas en situaciones cotidianas. ¡Espero que te resulten útiles!

Ejercicio 1: Filtrar productos con precios menores a un valor

Contexto: Tienes un catálogo de productos con sus precios y necesitas mostrar aquellos productos que estén en oferta (precio menor a 50).

1. **Descripción:** Tienes un array con los precios de los productos [100, 45, 30, 70, 25]. Luego:
 - Usa Array.filter() para filtrar los productos que tengan un precio menor a 50.
2. **Resultado esperado:**

```
const precios = [100, 45, 30, 70, 25];
const enOferta = precios.filter(precio => precio < 50);
console.log(enOferta); // [45, 30, 25], estos productos están en oferta
```

Ejercicio 2: Desanidar listas de comentarios en un foro

Contexto: Tienes una lista de comentarios en un foro, pero algunos comentarios son respuestas a otros (anidados) y deseas obtener una lista plana de todos los comentarios.

1. **Descripción:** Tienes un array anidado con los comentarios:

```
const comentarios = [  
  ["Buen post", "Gracias"],  
  ["Interesante"],  
  ["No estoy de acuerdo", ["Muy cierto", "Concuerdo"]]  
];
```

- Usa `Array.flat()` para crear un array plano con todos los comentarios, eliminando los niveles de anidación.

2. **Resultado esperado:**

```
const comentarios = [  
  ["Buen post", "Gracias"],  
  ["Interesante"],  
  ["No estoy de acuerdo", ["Muy cierto", "Concuerdo"]]  
];  
const todosComentarios = comentarios.flat(2);  
console.log(todosComentarios);  
// ["Buen post", "Gracias", "Interesante", "No estoy de acuerdo", "Muy cierto", "Concuerdo"]
```

Ejercicio 3: Procesamiento de datos de usuarios y asignación de etiquetas

Contexto: Tienes una lista de usuarios y quieres asignarles etiquetas dependiendo de su estatus y, al mismo tiempo, crear un array plano con los estatus.

1. **Descripción:** Tienes un array con objetos de usuarios con el formato `{ nombre: string, roles: string[] }`. Luego:

- Usa `Array.flatMap()` para descomponer los roles de cada usuario y obtener un array plano con todas las etiquetas.

2. **Resultado esperado:**

```
const usuarios = [  
  { nombre: "Juan", roles: ["admin", "editor"] },  
  { nombre: "María", roles: ["usuario"] },  
  { nombre: "Pedro", roles: ["moderador", "editor"] }  
];  
const rolesPlanos = usuarios.flatMap(usuario => usuario.roles);  
console.log(rolesPlanos); // ["admin", "editor", "usuario", "moderador", "editor"]
```

Ejercicio 4: Imprimir una lista de tareas pendientes

Contexto: Tienes una lista de tareas y quieres imprimirlas una por una en la consola.

1. **Descripción:** Tienes un array con las tareas pendientes ["Lavar platos", "Comprar comida", "Hacer ejercicio"]. Luego:
 - Usa `Array.forEach()` para recorrer el array y mostrar cada tarea con su índice.
2. **Resultado esperado:**

```
const tareas = ["Lavar platos", "Comprar comida", "Hacer ejercicio"];
tareas.forEach((tarea, index) => {
  console.log(`${index + 1}. ${tarea}`);
});
// 1. Lavar platos
// 2. Comprar comida
// 3. Hacer ejercicio
```

Ejercicio 5: Generar una lista de precios con IVA aplicado

Contexto: Tienes una lista de precios de productos y quieres generar una nueva lista en la que apliques el 21% de IVA (Impuesto sobre el Valor Añadido).

1. **Descripción:** Tienes un array con los precios originales [100, 200, 300]. Luego:
 - Usa `Array.map()` para crear un nuevo array en el que cada precio tenga el IVA incluido.
2. **Resultado esperado:**

```
const precios = [100, 200, 300];
const preciosConIVA = precios.map(precio => precio * 1.21);
console.log(preciosConIVA); // [121, 242, 363], precios con IVA aplicado
```

Ejercicio 6: Filtrar usuarios activos en un sistema

Contexto: Tienes una lista de usuarios y quieres filtrar aquellos que están activos.

1. **Descripción:** Tienes un array de objetos que representan usuarios con el formato `{ nombre: string, activo: boolean }`. Luego:

- Usa `Array.filter()` para obtener una lista con solo los usuarios activos.

2. Resultado esperado:

```
const usuarios = [  
  { nombre: "Juan", activo: true },  
  { nombre: "María", activo: false },  
  { nombre: "Pedro", activo: true }  
];  
const usuariosActivos = usuarios.filter(usuario => usuario.activo);  
console.log(usuariosActivos);  
// [{ nombre: "Juan", activo: true }, { nombre: "Pedro", activo: true }]
```

Ejercicio 7: Simplificación de listas de categorías de productos

Contexto: Tienes una lista con categorías de productos, algunas de las cuales están anidadas, y necesitas obtener una lista plana con todas las categorías.

1. **Descripción:** Tienes un array anidado con las categorías ["Electrónica", ["Teléfonos", "Computadoras"], "Ropa"]. Luego:
 - Usa `Array.flat()` para descomponer las categorías anidadas en una lista plana.
2. **Resultado esperado:**

```
const categorias = ["Electrónica", ["Teléfonos", "Computadoras"], "Ropa"];  
const categoriasPlanas = categorias.flat();  
console.log(categoriasPlanas);  
// ["Electrónica", "Teléfonos", "Computadoras", "Ropa"]
```

Ejercicio 8: Crear una lista de nombres en mayúsculas

Contexto: Tienes una lista de nombres y quieres convertirlos todos a mayúsculas para mostrarlos en un informe.

1. **Descripción:** Tienes un array con los nombres ["juan", "maría", "pedro"]. Luego:
 - Usa `Array.map()` para crear un nuevo array con todos los nombres en mayúsculas.
2. **Resultado esperado:**

```
const nombres = ["juan", "maría", "pedro"];
```

```
const nombresMayus = nombres.map(nombre => nombre.toUpperCase());
console.log(nombresMayus); // ["JUAN", "MARÍA", "PEDRO"]
```

Ejercicio 9: Imprimir las cantidades de productos vendidos

Contexto: Tienes una lista con las cantidades de productos vendidos y necesitas imprimirlas en la consola para llevar un control.

1. **Descripción:** Tienes un array con las cantidades [10, 20, 30, 40]. Luego:
 - Usa `Array.forEach()` para recorrer el array y mostrar cada cantidad junto con un mensaje.
2. **Resultado esperado:**

```
const cantidades = [10, 20, 30, 40];
cantidades.forEach(cantidad => {
  console.log(`Se vendieron ${cantidad} productos.`);
});
// Se vendieron 10 productos.
// Se vendieron 20 productos.
// Se vendieron 30 productos.
// Se vendieron 40 productos.
```

Ejercicio 10: Calcular el precio total de productos con descuento

Contexto: Tienes una lista de precios de productos y necesitas calcular el precio con un descuento aplicado del 10%.

1. **Descripción:** Tienes un array con los precios [50, 100, 150]. Luego:
 - Usa `Array.map()` para crear un nuevo array donde cada precio tenga aplicado el descuento del 10%.
2. **Resultado esperado:**

```
const precios = [50, 100, 150];
const preciosConDescuento = precios.map(precio => precio * 0.9);
console.log(preciosConDescuento); // [45, 90, 135], precios con descuento aplicado
```
