

6.6 Clase en vivo - Profesores



Status

Done

Unidad 6: Funciones del orden superior

Actividad: Ordenar un Arreglo con el Método `sort()`

Duración: 25 minutos

Instrucciones para el Profesor:

Explique cómo el método `sort()` puede ordenar elementos de un array según un criterio definido. Realice un ejemplo en vivo y permita a los estudiantes resolver el problema en sus propias computadoras.

Posible Solución:

```
const numeros = [40, 1, 5, 200];
numeros.sort((a, b) => a - b);
console.log(numeros); // [1, 5, 40, 200]
```

Actividad: Encuentra el Error - `forEach()`

Duración: 20 minutos

Instrucciones para el Profesor:

Explique cómo identificar errores comunes en el uso de `forEach()` y guíe a los estudiantes para que detecten y corrijan el error en el código proporcionado.

Posible Solución:

```
const numeros = [1, 2, 3, 4, 5];
let suma = 0;

numeros.forEach((num) => suma += num);
```

```
console.log(suma); // 15
```

Actividad: Filtrado de un array con `filter`

Instrucciones para el profesor:

Muestra cómo el método `filter` retorna un nuevo array con todos los elementos que cumplen la condición definida en la función de callback. Realiza el ejemplo en vivo filtrando los cursos con precio superior a 20000.

Posible solución:

```
const cursosCaros = cursos.filter((curso) => curso.precio > 20000);
console.log(cursosCaros);
// Salida esperada: [{ nombre: 'ReactJS', precio: 22000 }, { nombre: 'AngularJS', precio: 22000 }]
```

Find the Bug:

Presenta un código donde `filter` es usado incorrectamente, como olvidando retornar el valor en la función de callback.

```
const cursosCaros = cursos.filter((curso) => { curso.precio > 20000 }); // Error
```

Pide a los estudiantes que identifiquen y corrijan el error.

Actividad: Transformación de un array con `map`

Instrucciones para el profesor:

Explica cómo el método `map` crea un nuevo array transformando cada elemento del array original según la función de callback. Muestra cómo convertir las temperaturas de Fahrenheit a Celsius usando `map`.

Posible solución:

```
const celsiusTemps = fahrenheitTemps.map((fahrenheit) => ((fahrenheit - 32) * 5) / 9);
console.log(celsiusTemps);
// Salida esperada: [0, 20, 30, 40, 100]
```

Find the Bug:

Muestra un código con un error común, como olvidar retornar el valor dentro de la función de callback de `map`.

```
const celsiusTemps = fahrenheitTemps.map((fahrenheit) => {
  (fahrenheit - 32) * 5 / 9; // Error
});
```

Pide a los estudiantes que identifiquen y corrijan el error.

Actividad: Reducción de un array con `reduce`

Instrucciones para el profesor:

Muestra cómo el método `reduce` permite acumular un resultado al iterar sobre el array, devolviendo un único valor. Realiza el ejemplo en vivo calculando la suma total de los números en el array.

Posible solución:

```
const sumaTotal = numeros.reduce((acumulador, numero) => acumulador + numero, 0);
console.log(sumaTotal);
// Salida esperada: 21
```

Find the Bug:

Muestra un error típico, como olvidar definir el valor inicial del acumulador.

```
const sumaTotal = numeros.reduce((acumulador, numero) => acumulador + numero); // Error
```

Pide a los estudiantes que identifiquen y corrijan el error.

Preclase - 30 minutos previos

Repaso de Conceptos Clave (10 minutos)

- **Funciones de Orden Superior:** Revisión rápida sobre qué son las funciones de orden superior y su importancia en la programación funcional.
- **Recibir Funciones por Parámetro y Retornar Funciones:** Explicación sobre cómo se pueden recibir funciones como argumentos y devolver funciones como resultados.
- **Métodos de Búsqueda y Transformación:** Repaso de métodos clave como `forEach`, `map`, `filter`, `find`, `some`, `reduce`, y `sort`, enfocándose en sus usos y diferencias.

Espacio para Preguntas y Respuestas (15 minutos)

- Abrir el espacio para que los estudiantes hagan preguntas específicas sobre los métodos vistos y el concepto de funciones de orden superior.
- Resolver dudas con ejemplos en vivo si es necesario, como cómo aplicar `map` para transformar arrays o `reduce` para acumular valores.
- Si no hay preguntas, sugerir temas comunes, como diferencias entre `filter` y `find`, o el impacto de `sort` en arrays.

Demostraciones Prácticas (Opcional, dependiendo de las preguntas)

- Realizar demostraciones en vivo si las preguntas lo requieren, por ejemplo, mostrar cómo combinar varios métodos de orden superior en un solo flujo de trabajo.
- Ejemplos posibles: Uso de `reduce` para calcular la suma de precios de productos o `map` para modificar los valores de un array.