

Teoría JavaScript

Location

El objeto location representa la ubicación (URL) del objeto de documento actual y proporciona propiedades y métodos para redirigir la página a una nueva URL, actualizar la página y trabajar con la cadena de consulta (query string) de la URL.

Propiedades de location:

- **href**: Contiene la URL completa.
- **protocol**: El protocolo de la URL (por ejemplo, 'http:' o 'https:').
- **host**: El host de la URL (incluye el nombre de dominio y el puerto si es específico).
- **hostname**: El nombre de dominio de la URL.
- **port**: El número de puerto de la URL.
- **pathname**: El path de la URL.
- **search**: La cadena de consulta de la URL, incluyendo el signo de interrogación inicial.
- **hash**: El fragmento de la URL, incluyendo el signo de almohadilla inicial (#).

Métodos de location:

- **assign(url)**: Carga una nueva página web usando la URL proporcionada.
- **reload()**: Recarga la página actual.
- **replace(url)**: Reemplaza la URL actual por la nueva y no deja rastro de la URL anterior en el historial de navegación.

¿Qué es URLSearchParams?

URLSearchParams define métodos útiles para trabajar con la cadena de consulta de una URL. Permite manipular los parámetros de consulta (queries en inglés) de una URL de manera sencilla:

- **new URLSearchParams(queries)**: es utilizado para crear y manipular los parámetros de consulta de una URL de manera fácil y eficiente.
- **get(nombre)**: Retorna el primer valor asociado al parámetro de búsqueda dado.
- **getAll(nombre)**: Retorna todos los valores asociados con un parámetro de búsqueda específico.
- **set(nombre, valor)**: Establece o actualiza el valor para un parámetro de búsqueda.

- **append**(nombre, valor): Agrega un nuevo valor para el parámetro de búsqueda especificado.
- **delete**(nombre): Elimina el parámetro de búsqueda especificado.
- **has**(nombre): Retorna true si el parámetro de búsqueda existe, de lo contrario false.

Repaso II - Funciones de Orden Superior

¿Qué es find()?

El método find en JavaScript es una herramienta poderosa para buscar dentro de un array el primer elemento que cumple con una condición especificada.

A diferencia de otros métodos de búsqueda y filtrado, find detiene su ejecución tan pronto como encuentra un elemento que satisface la condición dada, lo que lo hace ideal para localizar un único elemento dentro de una colección sin necesidad de procesar todos los elementos.

A modo de ejemplo:

```
Unset
const numeros = [1, 12, 23, 8, 5, 7, 31];
const mayorADiez = numeros.find(cadaElemento => cadaElemento > 10);
console.log(mayorADiez); // 12
```

¿Para qué se utiliza find()?

El método find devuelve el primer elemento del array que cumple con la condición especificada en la función de prueba. Si no se encuentra ningún elemento que cumpla con la condición, find retorna undefined. La sintaxis básica del método find es la siguiente:

```
Unset
array.find((each, index, arr) => { })
```

- **array**: Es el array a iterar y evaluar con el método find.
- **function**(currentValue, index, arr): Es la función de prueba para cada elemento del array y debe retornar una expresión que se resuelva a un valor booleano (true o false).
 - **each**: Es el valor del elemento actual que está siendo procesado en el array.
 - **index** (opcional): Es el índice del elemento actual en el array.
 - **arr** (opcional): Es el array al que pertenece el elemento actual.

Generalmente, el primer parámetro (función) es una función anónima de tipo flecha, pero puede ser una función definida.

A modo de ejemplo:

Unset

```
const personas = [
  { nombre: "Ana", edad: 25, ciudad: "Madrid" },
  { nombre: "Juan", edad: 16, ciudad: "Barcelona" },
  { nombre: "Luis", edad: 17, ciudad: "Madrid" }
];

const primerMadrid = personas.find(persona => persona.ciudad === "Madrid");
console.log(primerMadrid); // { nombre: "Ana", edad: 25, ciudad: "Madrid" }
```