Arrays y objetos

Sprint 8

Realiza operaciones de comparación lógica:

- Los arrays pueden crearse con la sintaxis de array literal
- Todos los arrays tienen la propiedad `length` :
  - Esta devuelve el número de elementos en un array.
  - Puede emplearse para acceder al último elemento en un array:

```
const emptyArray = [];
const numbers = [1, 2, 3];
numbers[numbers.length - 3];  // 3
```

- Los elementos del array se pueden acceder o ser modificados por su índice mediante el operador de corchetes:

```
const numbers = [1, 2, 3];
numbers[0];                    // 1
numbers[0] = "a";
console.log(numbers);          // ["a", 2, 3]
```

Iteración sobre arrays

Por ejemplo, utilizando un bucle `for` para iterar a través de un array:

```
const numbers = [1, 2, 3];
for (let i = 0; i < numbers.length; i++) {
  console.log(numbers[i]);
}

// registra 1, 2 y 3 a la consola en líneas separadas
```

Un ejemplo análogo de usar un bucle `for..of` :

```
const numbers = [1, 2, 3];
for (let number of numbers) {
  console.log(number);
}

// registra 1, 2 y 3 a la consola en líneas separadas
```

Agregar y remover elementos del array

El método `push()`:

- Agrega un elemento al final de un array
- Devuelve la longitud del array

```
const letters = ["a", "b", "c"];
letters.push("d");                // 4
console.log(letters);             // ["a", "b", "c", "d"]
```

El método `pop()`:

- Elimina el último elemento de un array.
- Devuelve el elemento eliminado.

```
const letters = ["a", "b", "c"];
letters.pop();                    // "c"
console.log(letters);             // ["a", "b"]
```

Trabajar con objetos

Estructuras de datos que almacenan colecciones no ordenadas de pares clave:valor.

Crear un objeto con la sintaxis de objeto literal:

- claves y valores separados con dos puntos (:)
- a las claves se les debe dar nombres de variable JavaScript válidos o estar encerrados entre comillas
- los valores pueden ser cualquier valor JavaScript

```
const user = {
  name: "Sample User",
  "email address": "sample@example.com"
}
```

Puedes acceder y cambiar claves con:

- sintaxis de punto (`.`), si la clave tiene un nombre JavaScript válido
- sintaxis de corchetes (`[ ]`), si no

```
user.name;                                // "Sample User"
user.name = "New Name";
user.name;                                // "New Name"

user["email address"];                    // "kebab"
user["email address"] = "foo@bar.com";
user["email address"];                   // "foo@bar.com"
```

Métodos de array

Sprint 8

1.	**concat**	Fusiona arrays y elementos
2.	**push**	Añade elementos al final del array
3.	**pop**	Elimina el último elemento de un array y devuelve su valor
4.	**unshift**	Añade elementos al principio de un array
5.	**shift**	Elimina el primer elemento de un array y devuelve su valor
6.	**slice**	Copia una parte de un array y la devuelve como un nuevo array
7.	**splice**	Elimina los elementos especificados de un array y los sustituye por otros nuevos
8.	**forEach**	Itera sobre los elementos y ejecuta una función para cada uno de ellos
9.	**map**	Crea un nuevo array basado en el original ejecutando una función para cada elemento
10.	**filter**	Crea un nuevo array basado en el original filtrando los elementos según la comprobación proporcionada
11.	**some**	Comprueba si al menos un elemento del array pasa la comprobación proporcionada
12.	**every**	Comprueba que todos los elementos del array pasan la comprobación proporcionada
13.	**find**	Comprueba si hay un elemento determinado en el array
14.	**reduce**	Reduce un array a un solo valor
15.	**sort**	Ordena los elementos de un array según un determinado criterio

Cada método sirve para su propio propósito. Analiza la tarea en cuestión y decide cuál es la mejor para usar dadas las circunstancias. Esto no será fácil al principio, y a menudo tendrás que volver a la lista de métodos y recordar qué tipo de argumentos pueden tomar, y cómo funcionan en general. Con la práctica, estos conocimientos se grabarán en tu memoria y empezarás a elegir los métodos adecuados automáticamente.