

Módulo Javascript

Arrays

Objetivos: Conocer los Arrays en javascript, para qué sirven y cómo utilizarlos.

Un array es un objeto que almacena una lista de valores. Cada valor se identifica por un índice numérico, que comienza en cero. Los arrays en JavaScript pueden contener cualquier tipo de datos, incluyendo otros arrays y objetos.

Para crear un array en JavaScript, se puede utilizar la sintaxis literal de array, que consiste en corchetes `[]` y los valores separados por comas:

JavaScript

```
const miArray = [1, 2, 3, "hola", true];
```

También se pueden crear arrays vacíos e ir añadiendo valores después:

JavaScript

```
const miArray = [];  
miArray.push(1);  
miArray.push(2);  
miArray.push("hola");
```

Los arrays tienen una propiedad `length` que indica el número de elementos que contiene:

JavaScript

```
console.log(miArray.length); // 3
```

Para acceder a un elemento específico de un array, se utiliza su índice numérico, entre corchetes:

JavaScript

```
console.log(miArray[0]); // 1  
console.log(miArray[2]); // "hola"
```

También se pueden modificar los valores de un array accediendo a ellos directamente por su índice:

JavaScript

```
miArray[1] = "adiós";  
console.log(miArray); // [1, "adiós", "hola"]
```

Los arrays en JavaScript tienen muchos métodos útiles para manipular su contenido. Algunos ejemplos son:

- `push()` y `unshift()`: añaden un elemento al final o al principio del array, respectivamente.

JavaScript

```
const miArray = [1, 2, 3];  
miArray.push(4);  
miArray.unshift(0);  
console.log(miArray); // [0, 1, 2, 3, 4]
```

- `pop()` y `shift()`: eliminan el último o el primer elemento del array, respectivamente.

JavaScript

```
const miArray = [0, 1, 2, 3, 4];  
miArray.pop();  
miArray.shift();  
console.log(miArray); // [1, 2, 3]
```

- `slice()`: devuelve una copia de una porción del array, desde el índice indicado hasta el índice anterior al segundo parámetro indicado (opcional).

JavaScript

```
const miArray = [0, 1, 2, 3, 4];  
const miSubArray = miArray.slice(1, 3);  
console.log(miSubArray); // [1, 2]
```

- `splice()`: elimina o añade elementos al array, comenzando desde el índice indicado, y opcionalmente indicando el número de elementos a eliminar y los elementos a añadir.

JavaScript

```
const miArray = [0, 1, 2, 3, 4];  
miArray.splice(2, 1);  
console.log(miArray); // [0, 1, 3, 4]  
miArray.splice(1, 0, "hola", "mundo");  
console.log(miArray); // [0, "hola", "mundo", 1, 3, 4]
```

- `map()`: devuelve un nuevo array con los resultados de aplicar una función a cada elemento del array original.

JavaScript

```
const miArray = [1, 2, 3];  
const miArrayDoble = miArray.map(x => x * 2);
```

```
console.log(miArrayDoble); // [2, 4,
```