

JavaScript Array Methods

- description : JavaScript Array Methods
- author : 오션
- email : shlim@repia.com
- lastupdate : 2021-04-27

The source of this article

[JavaScript Array Methods](#)

JavaScript 메소드 `toString()`은 쉼표로 구분된 배열을 배열 값의 문자열로 변환합니다.

```
let fruits = ["Banana", "Orange", "Apple", "Mango"];
document.getElementById("demo").innerHTML = fruits.toString(); //
Banana,Orange,Apple,Mango
```

`join()` 메서드는 또한 모든 배열 요소를 문자열로 결합합니다.

`join()` 메서드는 `toString()`처럼 동작하지만 추가로 구분자(separator)를 지정할 수 있습니다.

```
let fruits = ["Banana", "Orange", "Apple", "Mango"];
document.getElementById("demo").innerHTML = fruits.join(" * ");
// Banana * Orange * Apple * Mango
```

Popping and Pushing

배열로 작업할 때, 쉽게 요소를 제거하고 새 요소를 추가 할 수 있습니다.

이것이 popping과 pushing 입니다.

배열에서 항목을 꺼내거나(popping), 배열로 항목을 넣습니다(push).

Popping

`pop()` 메서드는 배열에서 마지막 요소를 제거합니다.

```
let fruits = ["Banana", "Orange", "Apple", "Mango"];
```

```
document.getElementById("demo1").innerHTML = fruits; //
Banana, Orange, Apple, Mango
fruits.pop(); // remove Mango
document.getElementById("demo2").innerHTML = fruits; // Banana, Orange, Apple
```

pop() 메서드는 빠져나온(popped out) 마지막 요소를 반환합니다.

```
let fruits = ["Banana", "Orange", "Apple", "Mango"];
document.getElementById("demo1").innerHTML = fruits; //
Banana, Orange, Apple, Mango
document.getElementById("demo2").innerHTML = fruits.pop(); // Mango
document.getElementById("demo3").innerHTML = fruits; //
Banana, Orange, Apple
```

Pushing

push() 메서드는 배열의 마지막에 새 요소를 추가합니다.

```
let fruits = ["Banana", "Orange", "Apple", "Mango"];
document.getElementById("demo").innerHTML = fruits;

function myFunction() {
  fruits.push("Kiwi");
  document.getElementById("demo").innerHTML = fruits; //
Banana, Orange, Apple, Mango, Kiwi
}
```

push() 메서드는 새로운 배열의 길이를 반환합니다:

```
let fruits = ["Banana", "Orange", "Apple", "Mango"];
document.getElementById("demo1").innerHTML = fruits; //
Banana, Orange, Apple, Mango

function myFunction() {
  document.getElementById("demo2").innerHTML = fruits.push("Kiwi"); // 5
  document.getElementById("demo1").innerHTML = fruits; //
Banana, Orange, Apple, Mango, Kiwi
}
```

Shifting Elements

Shifting은 popping과 동일하며, 마지막 요소 대신에 첫 번째 요소를 작업합니다.

`shift()` 메서드는 첫 번째 배열 요소를 제거하고, 다른 모든 요소들을 더 낮은 인덱스로(왼쪽으로) “이동” 시킵니다.

```
let fruits = ["Banana", "Orange", "Apple", "Mango"];
document.getElementById("demo1").innerHTML = fruits; //
Banana, Orange, Apple, Mango
fruits.shift(); // remove first element "Banana" from fruits
document.getElementById("demo2").innerHTML = fruits; // Orange, Apple, Mango
```

`shift()` 메서드는 첫 번째 배열 요소가 제거된 문자열을 반환합니다:

```
let fruits = ["Banana", "Orange", "Apple", "Mango"];
document.getElementById("demo1").innerHTML = fruits; //
Banana, Orange, Apple, Mango
document.getElementById("demo2").innerHTML = fruits.shift(); // Banana
document.getElementById("demo3").innerHTML = fruits; //
Orange, Apple, Mango
```

`unshift()` 메서드는 배열의 시작 부분에 새로운 요소를 추가하고, 기존의 요소들을 배열의 오른쪽으로 이동시킵니다(“unshifts” older elements):

```
let fruits = ["Banana", "Orange", "Apple", "Mango"];
document.getElementById("demo").innerHTML = fruits;
//Banana, Orange, Apple, Mango

function myFunction() {
  fruits.unshift("Lemon");
  document.getElementById("demo").innerHTML = fruits; //
Lemon, Banana, Orange, Apple, Mango
}
```

`unshift()` 메서드는 새로운 배열의 길이를 반환합니다.

```
let fruits = ["Banana", "Orange", "Apple", "Mango"];
document.getElementById("demo1").innerHTML = fruits; //
Banana, Orange, Apple, Mango
document.getElementById("demo2").innerHTML = fruits.unshift("Lemon"); // 5
document.getElementById("demo3").innerHTML = fruits; //
Lemon, Banana, Orange, Apple, Mango
```

Changing Elements

인덱스 번호(index number)를 사용하여 배열의 요소들에 액세스합니다.

배열의 인덱스는 0 부터 시작합니다. [0]은 첫 번째 배열 요소이고, [1]은 두 번째, [2]은 세 번째...

```
let fruits = ["Banana", "Orange", "Apple", "Mango"];
document.getElementById("demo1").innerHTML = fruits; //
Banana, Orange, Apple, Mango
fruits[0] = "Kiwi"; // index[0]의 Banana를 "Kiwi"로 교체
document.getElementById("demo2").innerHTML = fruits; //
Kiwi, Orange, Apple, Mango
```

length 속성은 배열에 새로운 요소를 쉽게 추가하는 방법을 제공합니다.

```
let fruits = ["Banana", "Orange", "Apple", "Mango"];
document.getElementById("demo").innerHTML = fruits; //
Banana, Orange, Apple, Mango

function myFunction() {
  fruits[fruits.length] = "Kiwi"; // fruits 배열에 새 요소 'Kiwi'를 배열의 마지막
  에 추가
  document.getElementById("demo").innerHTML = fruits; //
  Banana, Orange, Apple, Mango, Kiwi
}
```

Deleting Elements

JavaScript 배열은 오브젝트이므로, JavaScript 연산자 delete를 사용하여 요소를 삭제할 수 있습니다:

delete를 사용하는 것은 배열에 정의되지 않은 구멍(holes)를 남길 수도 있습니다. 대신에 pop() 또는 shift()를 사용하세요.

[오션, Javascript, Array, Methods](#)

From:

<http://rwiki.repia.com/> - 문제를 잘 정의하는 것은 문제를 절반 해결한 것이다. - 2023.12

Permanent link:

[http://rwiki.repia.com/doku.php?id=wiki:javascript:javascript_note:js_array_methods
&rev=1619486527](http://rwiki.repia.com/doku.php?id=wiki:javascript:javascript_note:js_array_methods&rev=1619486527) 

Last update: **2022/03/10 19:52**