



천천히 잘 읽어봐라

자바스크립트의 배열은 다양한 데이터 유형의 값들을 순차적으로 저장하는 데 사용되는 자료 구조다. 배열의 각 요소는 인덱스를 통해 접근할 수 있으며, 자바스크립트 배열은 동적으로 크기가 조정될 수 있다. 배열의 기본 작업에는 요소 추가, 삭제, 탐색 등이 있다.

자바스크립트 배열 사례들

1. 요소 추가 및 삭제

- `push()` : 배열의 끝에 요소 추가
- `pop()` : 배열의 끝 요소 제거
- `shift()` : 배열의 첫 요소 제거
- `unshift()` : 배열의 시작 부분에 요소 추가

2. 배열 탐색 및 변환

- `find()` : 조건에 맞는 첫 번째 요소 반환
- `filter()` : 조건에 맞는 모든 요소를 포함하는 새 배열 반환
- `map()` : 모든 요소에 대해 함수를 실행하고 결과를 포함하는 새 배열 반환

3. 배열 합치기 및 분할

- `concat()` : 배열을 합침

- `slice()`: 배열의 일부를 추출하여 새 배열 생성

4. 배열 정렬

- `sort()`: 배열의 요소를 정렬
- `reverse()`: 배열의 순서를 뒤집음

예시 코드 및 결과값

1. 요소 추가 및 삭제

```
let fruits = ["Apple", "Banana"];
fruits.push("Orange"); // Orange 추가
fruits.pop(); // 마지막 요소 Orange 제거
```

결과: `["Apple", "Banana"]`

2. 배열 탐색 및 변환

```
let numbers = [1, 2, 3, 4, 5];
let evenNumbers = numbers.filter(num => num % 2 === 0);
```

결과: `[2, 4]` (짝수만 포함하는 배열)

3. 배열 합치기 및 분할

```
let arr1 = [1, 2, 3];
let arr2 = [4, 5, 6];
let combined = arr1.concat(arr2);
```

결과: `[1, 2, 3, 4, 5, 6]` (두 배열의 결합)

4. 배열 정렬

```
let arr = [3, 1, 4, 1, 5, 9];
arr.sort();
```

결과: `[1, 1, 3, 4, 5, 9]` (오름차순으로 정렬된 배열)

1. 요소 추가 및 삭제

push()

```
let fruits = ["Apple", "Banana"];  
fruits.push("Orange");
```

결과: ["Apple", "Banana", "Orange"] (Orange가 추가됨)

pop()

```
fruits.pop();
```

결과: ["Apple", "Banana"] (마지막 요소 Orange 제거됨)

shift()

```
fruits.shift();
```

결과: ["Banana"] (첫 번째 요소 Apple 제거됨)

unshift()

```
fruits.unshift("Strawberry");
```

결과: ["Strawberry", "Banana"] (Strawberry가 배열의 시작 부분에 추가됨)

2. 배열 탐색 및 변환

find()

```
let numbers = [5, 12, 8, 130, 44];  
let found = numbers.find(element => element > 10);
```

결과: 12 (10보다 큰 첫 번째 요소 반환)

filter()

```
let filtered = numbers.filter(num => num > 10);
```

결과: `[12, 130, 44]` (10보다 큰 모든 요소 반환)

`map()`

```
let mapResult = numbers.map(num => num * 2);
```

결과: `[10, 24, 16, 260, 88]` (각 요소를 2배로 만든 새 배열 반환)

3. 배열 합치기 및 분할

`concat()`

```
let arr1 = ["a", "b", "c"];
let arr2 = ["d", "e", "f"];
let combined = arr1.concat(arr2);
```

결과: `["a", "b", "c", "d", "e", "f"]` (두 배열이 결합된 결과)

`slice()`

```
let sliced = combined.slice(2, 5);
```

결과: `["c", "d", "e"]` (인덱스 2부터 4까지의 요소를 추출한 새 배열)

4. 배열 정렬

`sort()`

```
let unsortedNumbers = [3, 1, 4, 1, 5, 9];
unsortedNumbers.sort((a, b) => a - b);
```

결과: `[1, 1, 3, 4, 5, 9]` (오름차순으로 정렬된 배열)

`reverse()`

```
let reversed = unsortedNumbers.reverse();
```

결과: `[9, 5, 4, 3, 1, 1]` (배열의 순서가 뒤집힌 결과)