

## ■ 자바스크립트 배열

많은량의 데이터를 저장하고 사용할 때 배열을 사용한다

배열이름의 기준으로부터 인덱스를 통해 각 요소에 접근할 수 있다

인덱스는 0부터 시작함

배열은 데이터를 순서대로 저장하고 다루는 데 사용되는 객체로, 유용한 메서드(기능)를 제공한다

### 1. 배열리터럴로 생성하기

`let evens = [ 2,4,6];` //변수 3개 대신 배열을 사용할 수 있다.

`let empty=[];`

`console.log ( evens[0]);`

`console.log ( evens[1]) ;`

`console.log ( evens[2]);`

### 2. Array 생성자로 생성하기

`let evens = new Array( 2,4,6);`

`let empty= new Array();`

`console.log ( evens[0]);`

`console.log ( evens[1]) ;`

`console.log ( evens[2]);`

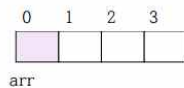
length프로퍼티 :배열의 길이, 최대인덱스 +1

c,java 같은 프로그래밍 언어의 배열요소는 메모리의 연속된 공간에 차례대로 배치됨

따라서 인덱스를 지정하면 인덱스가 가리키는 요소를 매우 빠르게 읽거나 쓸 수 있다

( 연속적이다 , 요소의 크기가 같다 가 보장되어야 함.. )

변수4개와 배열을 사용 예 ( 2,4,6,8 이라는 짝수값을 저장한다고 하자



```
a=2;  
b=4;  
c=6;  
d=8;  
arr[0]=2;  
arr[1]=4;  
arr[2]=6;  
arr[3]=8;
```

자바스크립트 배열은 Array객체이며 객체로 배열의 기능을 가상으로 흉내 낸 것이다.  
 자바스크립트 배열은 객체이므로 **속성과 다양한 기능**(함수형태임)을 제공한다.

자바스크립트 배열 특징:

- 인덱스 **기반(0부터 시작)**으로 요소에 접근
- 요소의 자료형이 혼합 가능.
- **배열의 길이는 length 속성**으로 확인

## ■ 배열의메서드

### 1. 탐색 및 확인

메서드	설명	반환값
<b>find()</b>	<b>조건에 맞는 첫 번째 요소 반환</b>	<b>요소 또는 undefined</b>
<b>findIndex()</b>	<b>조건에 맞는 첫 번째 요소의 인덱스 반환</b>	<b>인덱스 또는 -1</b>
<b>includes()</b>	<b>배열에 특정 값이 있는지 확인</b>	<b>true 또는 false</b>
<b>indexOf()</b>	<b>특정 값의 첫 번째 인덱스 반환</b>	<b>인덱스 또는 -1</b>
<b>lastIndexOf()</b>	<b>특정 값의 마지막 인덱스 반환</b>	<b>인덱스 또는 -1</b>

### 2. 배열 조작

메서드	설명	반환값
<b>push()</b>	<b>배열 끝에 요소 추가</b>	<b>새로운 배열 길이</b>
<b>pop()</b>	<b>배열 끝의 요소 제거</b>	<b>제거된 요소</b>
<b>unshift()</b>	<b>배열 앞에 요소 추가</b>	<b>새로운 배열 길이</b>
<b>shift()</b>	<b>배열 앞의 요소 제거</b>	<b>제거된 요소</b>
<b>splice()</b>	<b>특정 위치에서 요소 추가/제거/교체</b>	<b>제거된 요소 배열</b>
<b>concat()</b>	<b>배열 합치기</b>	<b>새 배열</b>

### 3. 배열 순회

메서드	설명	반환값
<b>forEach()</b>	<b>각 요소에 대해 함수를 실행</b>	<b>undefined</b>
<b>map()</b>	<b>각 요소를 변환하여 새 배열 반환</b>	<b>새 배열</b>
<b>filter()</b>	<b>조건에 맞는 요소로 새 배열 생성</b>	<b>새 배열</b>
<b>reduce()</b>	<b>배열의 값을 누적하여 단일 값 반환</b>	<b>누적된 값</b>

### 4. 정렬 및 변형

메서드	설명	반환값
<b>sort()</b>	<b>배열 요소 정렬 (기본은 문자열 순서)</b>	<b>정렬된 배열</b>
<b>reverse()</b>	<b>배열 요소 순서 반전</b>	<b>반전된 배열</b>
<b>join()</b>	<b>배열 요소를 문자열로 결합</b>	<b>문자열</b>
<b>flat()</b>	<b>중첩 배열을 평탄화</b>	<b>평탄화된 배열</b>

### 5. 생성 및 채우기

메서드	설명	반환값
<b>Array.from()</b>	<b>유사 배열/반복 가능한 객체를 배열로 변환</b>	<b>새 배열</b>
<b>Array.of()</b>	<b>인수로 배열 생성</b>	<b>새 배열</b>
<b>fill()</b>	<b>배열을 특정 값으로 채움</b>	<b>수정된 배열</b>

### 6. 기타

메서드	설명	반환값
<b>isArray()</b>	<b>값이 배열인지 확인</b>	<b>true 또는 false</b>
<b>toString()</b>	<b>배열을 문자열로 변환</b>	<b>문자열</b>

■ 배열의 문자열로 변환하는 매서드 `toString()` vs `join()`

`toString()` : 배열의 모든 요소를 쉼표(,)로 연결하여 단순 문자열로 반환

`join()` : 배열의 모든 요소를 지정한 구분자로 연결하여 문자열로 반환

■ `toString()` 예시

```
const arr = [1, 2, 3];  
console.log(arr.toString()); // "1,2,3"
```

■ `join()` 예시

```
const arr = [1, 2, 3];  
console.log(arr.join());           // "1,2,3" (기본값 쉼표 사용)  
console.log(arr.join('-'));        // "1-2-3" (구분자를 하이픈으로 지정)  
console.log(arr.join(' | '));      // "1 | 2 | 3"  
console.log(arr.join(''));         // "123" (구분자를 빈 문자열로 지정)
```