

27장. 배열

27.1 배열이란?

▼ 배열이란?

여러 개의 값을 순차적으로 나열한 자료구조

요소: 배열이 가지고 있는 값

인덱스: 배열에서 자신의 위치를 나타내는 0이상의 정수

▼ 배열 특징

1. 요소에 접근할 때 : 대괄호 표기법 사용
2. 배열의 길이를 나타내는 length 프로퍼티 → for문을 통해 순차적으로 요소 접근 가능
3. 배열은 객체 타입
4. 일반 객체와 차이점

구분	객체	배열
구조	프로퍼티 키와 프로퍼티 값	인덱스와 요소
값의 참조	프로퍼티 키	인덱스
값의 순서	X	O
length 프로퍼티	X	O

27.2 자바스크립트 배열은 배열이 아니다

▼ 밀집 배열이란?

배열의 요소는 하나의 데이터 타입으로 통일되어 있으며 서로 연속적으로 인접해 있다

▼ 희소 배열이란?

배열의 요소가 연속적으로 이어져 있지 않는 배열

▼ 자바스크립트의 배열이란?

일반적인 배열의 동작을 흉내 낸 특수한 객체

▼ 일반적인 배열과 자바스크립트 배열의 장단점

1. 장점

일반적인 배열은 인덱스로 요소에 빠르게 접근 가능

자바스크립트 배열은 요소를 삽입 또는 삭제하는 경우 빠른 성능 기대 가능

2. 단점

일반적인 배열은 요소를 삽입 또는 삭제하는 경우에는 비효율

자바스크립트 배열은 해시 테이블로 구현된 객체로 인덱스로 요소에 접근하는 경우 성능적인 면에서 느림

27.3 length 프로퍼티와 희소 배열

▼ length 특징

배열에 요소를 추가하거나 삭제하면 자동 갱신된다.

▼ 희소 배열 특징

희소 배열은 length와 배열 요소의 개수가 일치하지 않는다.

희소 배열의 length는 희소 배열의 실제 요소 개수보다 언제나 크다.

27.4 배열 생성

▼ 배열 리터럴

0개 이상의 요소를 쉼표로 구분하며 대괄호로 묶는다

객체 리터럴과 달리 프로퍼티 키가 없고 값만 존재

▼ Array 생성자 함수

전달된 인수가 1개이고 숫자인 경우 → length 프로퍼티 값이 인수인 배열을 생성

만약 전달된 인수가 범위를 벗어나면 RangeError가 발생

전달된 인수가 없는 경우 빈 배열을 생성 → 배열 리터럴 []과 같음

전달된 인수가 2개 이상이거나 숫자가 아닌 경우 인수를 요소로 갖는 배열 생성

▼ Array.of

전달된 인수가 1개이고 숫자이더라도 인수를 요소로 갖는 배열을 생성

▼ Array.from

유사 배열 객체 또는 이터러블 객체를 인수로 전달받아 배열로 변환하여 반환

27.5 배열 요소의 참조

27.6 배열 요소의 추가와 갱신

- 인덱스로 요소에 접근하여 명시적 값을 할당하지 않은 요소는 생성되지 않는다는 것에 주의
- 이미 요소가 존재하는 요소에 값을 재할당하면 요소값이 갱신
- 정수 이외의 값을 인덱스처럼 사용하면 요소가 생성되는 것이 아니라 프로퍼티가 생성

27.7 배열 요소의 삭제

▼ delete 연산자란?

객체의 프로퍼티를 삭제

하지만, 희소 배열이 되기 때문에 희소 배열을 만들지 않으면서 특정 요소를 완전히 삭제하려면 `Array.prototype.splice` 메서드를 사용

27.8 배열 메서드

▼ 배열 메서드란?

배열에는 원본 배열을 직접 변경하는 메서드와 원본 배열을 직접 변경하지 않고 새로운 배열을 생성하여 반환하는 메서드

- `Array.isArray` : 전달된 인수가 배열이면 `true`, 배열이 아니면 `false`를 반환
- `Array.prototype.indexOf` : 원본 배열에서 인수로 전달된 요소를 검색하여 인덱스를 반환 → 배열에 특정 요소가 존재하는지 확인할 때 유용
- `Array.prototype.push` : 모든 값을 원본 배열의 마지막 요소로 추가하고 변경된 `length` 프로퍼티 값을 반환
- `Array.prototype.pop` : 원본 배열에서 마지막 요소를 제거하고 제거한 요소를 반환

- `Array.prototype.unshift` : 모든 값을 원본 배열의 선두에 요소로 추가하고 변경된 `length` 프로퍼티 값을 반환
- `Array.prototype.shift` : 첫 번째 요소를 제거하고 제거한 요소를 반환 → 원본 배열이 빈 배열이면 `undefined` 반환
- `Array.prototype.concat` : 인수로 전달된 값들을 원본 배열의 마지막 요소로 추가한 새로운 배열을 반환
- `Array.prototype.splice` : 원본 배열의 중간에 요소를 추가하거나 중간에 있는 요소를 제거하는 경우 `splice` 메서드를 사용
- `Array.prototype.slice` : 인수로 전달된 범위의 요소들을 복사하여 배열로 반환
- `Array.prototype.join` : 원본 배열의 모든 요소를 문자열로 변환한 후, 구분자로 연결한 문자열을 반환
- `Array.prototype.reverse` : 원본 배열의 순서를 반대로 반환
- `Array.prototype.fill` : 인수로 전달받은 값을 배열의 처음부터 끝까지 요소로 채움
- `Array.prototype.includes` : 배열 내에 특정 요소가 포함되어 있는지 확인하여 `true` 또는 `false`를 반환
- `Array.prototype.flat` : 인수로 전달한 깊이만큼 재귀적으로 배열을 평탄화

27.9 배열 고차 함수

▼ 고차함수란?

함수를 인수로 전달받거나 함수를 반환하는 함수

- `Array.prototype.sort` : 원본 배열을 직접 변경하며 정렬된 배열을 반환
- `Array.prototype.forEach` : `for`문을 대체할 수 있는 고차 함수 → 자신의 내부에서 반복문을 실행
- `Array.prototype.map` : 자신을 호출한 배열의 모든 요소를 순회하면서 인수로 전달받은 콜백 함수를 반복 호출 → 콜백 함수의 반환 값들로 구성된 새로운 배열을 반환
- `Array.prototype.filter` : 자신을 호출한 배열의 모든 요소를 순회하면서 인수로 전달받은 콜백 함수를 반복 호출 → 콜백 함수의 반환값이 `true`인 요소로만 구성된 새로운 배열 반환
- `Array.prototype.reduce` : 자신을 호출한 배열을 모든 요소를 순회하며 인수로 전달받은 콜백 함수를 반복 호출

- `Array.prototype.some` : 자신을 호출한 배열의 요소를 순회하면서 인수로 전달된 콜백 함수를 호출
- `Array.prototype.every` : 인수로 전달된 콜백 함수를 호출 → 콜백 함수의 반환값이 모두 참이면 `true`, 한 번이라도 거짓이면 `false` 반환
- `Array.prototype.find` : 반환값이 `true`인 첫 번째 요소를 반환, `true`인 요소가 존재하지 않는다면 `undefined` 반환
- `Array.prototype.findIndex` : 반환값이 `true`인 첫 번째 요소의 인덱스를 반환, `true`인 요소가 존재하지 않으면 `-1`을 반환
- `Array.prototype.flatMap` : `map` 메서드를 통해 생성된 새로운 배열을 평탄화한다.