4주차 스터디

7장 배열

작성자 : 전솔

7장 배열

7 배열

배열 : 정렬된 값의 집합.

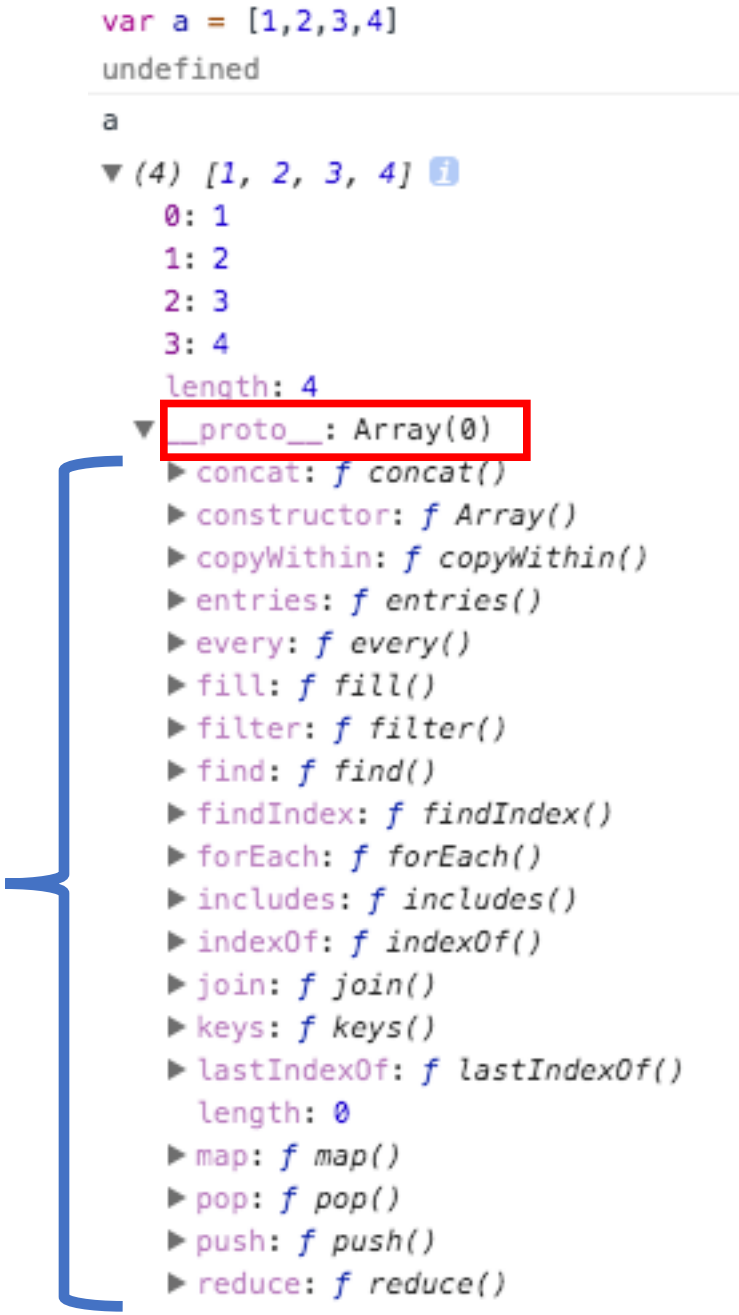
- length 프로퍼티는 배열에 새 원소가 추가될 때마다 자동으로 갱신됨.
- length 값을 임의로 설정함으로써 배열의 크기를 줄일 수 있음.
- 배열은 Array.prototype에 정의된 유용한 매서드들을 상속 함
- 배열의 class 속성 값은 "Array"로 설정됨

자바 스크립트 배열은 자바스크립트 객체의 특별한 형태

- 배열의 인덱스는 프로퍼티 이름이 정수

배열은 Array.prototype 의 프로퍼티들을 상속 받음

- Array.prototype에는 배열을 다루는 여러 메서드가 정의 되어 있음



7.1 배열 만들기

1. 리터럴

- var a = [1,2,3,4]

2. Array() 생성자 사용

- var a = new Array();
- var a = new Array(10); // 크기가 10인 배열 만들기
- var a = new Array(5,4,3,2,1);

=> 배열의 생성자를 사용하는 것보다 배열 리터럴을 사용하자!

7.3 희소 배열

희소 배열

- 배열에 속한 원소의 위치가 연속적이지 않은 배열

```
a = new Array(5);
▶ (5) [empty × 5]
a[1000] = 0;
0
a.length
1001
```

- Length > 원소의 개수
- 보통 배열보다 일반적으로 느리고, 메모리를 많이 사용할 뿐 아니라, 원소를 찾는데 걸리는 시간이 일반 객체의 속성값을 찾는 시간만큼 오래 걸림.
- 실무에서 쓰일 대부분의 자바스크립트배열은 빈 원소가 없는 배열

7.5 다차원 배열

다차원 배열

- 자바 스크립트는 진정한 의미에서의 다차원 배열을 지원하지는 않음
- 배열의 배열을 사용해 다차원 배열을 흉내
- 구구단 표 만들어 보기(연습) – 7*5 = 35 출력

7.8 배열 메서드(ECMAScript 3 기준)

1. **join()**

- 배열의 모든 원소를 문자열로 변환하고, 변환한 문자들을 이어 붙인 결과를 반환

2. **split()**

- join()과 반대로 작동
- 문자열을 조각들로 분리하고, 이 조각들을 원소로 하는 배열을 생성

3. **reverse()**

- 배열의 원소 순서를 반대로 뒤집어 반환
- 이미 존재하는 배열안에서 원소들의 순서를 뒤 바꿈

4. **sort()**

- 배열 안의 원소들을 정렬하여 반환

7.8 배열 메서드(ECMAScript 3 기준)

5. concat()

- 기존 배열의 모든 원소에 concat() 메서드의 전달 인자들을 추가한 새로운 배열을 반환

6. slice()

- 부분 배열을 반환
- 처음 인덱스(포함) ~ 마지막 인덱스(포함x)
- 기존 배열 변경 x

7. splice()

- 배열의 원소를 삽입하거나 원소를 제거하려 할 때 범용적으로 사용 할 수 있는 메서드
- 호출 배열 변경 o

8. push()

- 하나 이상의 원소들을 배열의 끝부분에 이어 붙이고, 배열의 새로운 length값을 반환

7.8 배열 메서드(ECMAScript 3 기준)

9. pop()

- push()와 반대로 작동
- 배열의 마지막 원소를 제거하고 배열의 length값을 감소 시킨 후, 배열에서 제거한 원소들을 반환
- push()와 pop() 모두 배열 그 자체를 변화

10. Unshift, shift()

- push() 와 pop()과 유사하게 동작하는데, 배열의 끝이 아니라 배열의 맨 앞에서 원소를 추가하고 제거

11. toString()

- 배열의 모든 원소를 문자열로 변환하고 이 문자열들을 쉼표로 분리한 목록을 반환
- 호출 배열 변경 o

12. toLocaleString()

- 배열이 각 원소들을 그 원소의 toLocaleString() 메서드를 사용해 변환하고, 변환된 문자열들을 지역화된(또는 각 구현체가 정의하는) 구분자 문자열로 연결하여 반환
- https://opentutorials.org/course/50/177

7.9 배열 메서드(ECMAScript 5 기준)

1. forEach()

- 배열을 순회하는 메서드
- 기존 값 변경

```
> var data = [1,2,3,4,5];
<· undefined
> data.forEach(function(v,i,a){a[i] = v + 1})
<· undefined
> data
<· ▶ (5) [2, 3, 4, 5, 6]
```

=> v : 원소 값, I : 원소의 인덱스 값, a : 배열 그 자체

2. map()

- 배열의 각 원소를 메서드의 첫 번째 전달인자로 지정한 함수에 전달하고, 해당 함수의 반환값을 배열에 담아 반환
- 기존 값 변경 x

```
a = [1,2,3,4];
▶ (4) [1, 2, 3, 4]
b = a.map(function(x) {return x*x});
▶ (4) [1, 4, 9, 16]
```

=> X : 원소 값

7.9 배열 메서드(ECMAScript 5 기준)

3. reduce()

- 인자로 주어진 함수를 사용하여 배열의 원소들을 하나의 값으로 결합

```
> a = [1,2,3,4,5]
▶ (5) [1, 2, 3, 4, 5]
> var sum = a.reduce(function(x,y) {return x+y},0);
undefined
> sum
15
> var sum = a.reduce(function(x,y) {return x+y},1);
undefined
> sum
16
> var sum = a.reduce(function(x,y) {return x+y},2);
undefined
> sum
17
```

7.9 배열 메서드(ECMAScript 5 기준)

4. index(), lastIndexOf()

- 배열의 원소 중에서 특정한 값을 찾음

```
> a = [0,1,2,1,0];
< ▶ (5) [0, 1, 2, 1, 0]
> a.indexOf(1)
< 1
> a.lastIndexOf(1)
< 3
> a.indexOf(3)
< -1
```

7.11 유사 배열 객체

유사배열

- 배열과 유사한 객체
- length 프로퍼티와 양의 정수 이름의 프로퍼티가 있는 객체는 일종의 배열로 취급 할 수 있음.
- 수많은 배열 알고리즘이 기존 배열뿐 아니라 유사 배열 객체에서도 잘 작동함.
- 문자열은 배열과 유사객체가 아님.
- 유사 배열은 Array.prototype을 상속받지 않기 때문에 배열 메서드를 해당 객체의 메서드로 호출 할 수는 없음
  - 단, Function.call 메서드를 통해서 간접적으로 호출 할 수 있음.

```
var a = {"0":"a","1":"b","2":"c",length:3}
undefined
Array.prototype.join.call(a,"+")
"a+b+c"
```

7.12 문자열을 배열처럼 사용하기

문자열은 읽기 전용 배열처럼 동작

- 각 문자는 charAt() 메서드나 []로 접근 가능

```
var s = "test";
undefined
s.charAt(0);
"t"
s[1]
"e"
```

- 문자열은 변하지 않는 값이라서, 읽기 전용 배열로만 다룰 수 있음
  - Push() , sort(), reverse(), splice() 와 값은 배열 메서드는 배열을 직접 수정하므로 문자열에는 동작하지 않는다.