Array 오브젝트(ES5)

프로퍼티 리스트(ES5)

이름	개요
Array 함수	
isArray()	배열 여부 반환; 배열이면 true, 아니면 false 반환
Array.prototype	
indexOf()	지정한 값에 일치하는 엘리먼트 인덱스 반환
lastIndexOf()	indexOf()와 같으며, 마지막 인덱스 반환
forEach()	배열을 반복하면서 콜백 함수 실행
every()	반환 값이 false일 때까지 콜백 함수 실행
some()	반환 값이 true일 때까지 콜백 함수 실행
filter()	콜백 함수에서 true를 반환한 엘리먼트 반환
map()	콜백 함수에서 반환한 값을 새로운 배열로 반환
reduce()	콜백 함수의 반환 값을 파라미터 값으로 사용
reduceRight()	reduce()와 같음. 단, 배열의 끝에서 앞으로 진행

isArray()

구분	데이터(값)
object	Array 오브젝트
파라미터	체크 대상
반환	배열: true, 아니면: false

- 체크 대상이 배열이면 **true**, 아니면 **false**
- isArray()는 함수 코드 1
- isArray() 함수가 필요한 이유 코드 2

index 처리 메소드

indexOf()

구분	데이터(값)
data	검색 대상
파라미터	검색할 값
	검색 시작 인덱스, 디폴트: 0
반환	검색된 인덱스

- 파라미터 값과 같은 엘리먼트의 인덱스 반환
  - 왼쪽에서 오른쪽으로 검색
  - 값이 같은 엘리먼트가 있으면 검색 종료 코드 1
  - 데이터 타입까지 체크 코드 2
- 두 번째 파라미터의 인덱스부터 검색 코드 3
- String과 Array의 indexOf() 차이 코드 4

lastIndexOf()

구분	데이터(값)
data	검색 대상
파라미터	검색할 값
	검색 시작 인덱스opt
반환	검색된 인덱스

- 파라미터 값과 같은 엘리먼트의
  - 마지막 인덱스 반환
  - 다른 처리 방법은 indexOf()와 같음 코드 1

콜백 함수를 가진 **Array** 메소드

지금부터 다루는 **7**개 메소드는
모두 콜백 함수를 호출합니다**.**
키워드는 시맨틱과 독립성입니다**.**

forEach()

구분	데이터(값)
data	반복 대상
파라미터	콜백 함수
	this로 참조할 오브젝트opt
반환	undefined

- 배열의 엘리먼트를 하나씩 읽어 가면서 콜백 함수 호출
- 콜백 함수 파라미터
  - 엘리먼트 값, 인덱스, 배열 전체 코드 1
- break, continue 사용 불가
  throw는 사용 가능
- 콜백 함수 분리(독립성) 코드 2
- this로 오브젝트 참조 코드 3

for()와 **forEach()** 차이

forEach()

- forEach()를 시작할 때 반복 범위 결정
- 엘리먼트를 추가하더라도 처리하지 않음 코드 1
- 현재 인덱스보다 큰 인덱스의 값을 변경하면 변경된 값을 사용 코드 2
  - 현재 인덱스보다 작은 인덱스의 값을 변경하면 처리하지 않음(당연 😀)
- 현재 인덱스보다 큰 인덱스의 엘리먼트를 삭제하면 배열에서 삭제되므로 반복에서 제외됨 코드 3
  - 추가는 처리하지 않지만, 삭제는 반영

for()와 forEach()

- forEach()는 시맨틱 접근
  - 처음부터 끝까지 반복한다는 시맨틱
  - 반복 중간에 끝나지 않는다는 시맨틱
  - 시맨틱으로 소스 코드의 가독성 향상
- for()는 함수 코드를 읽어야 알 수 있음
  - break, continue
- forEach()는 반복만 하며
  - 콜백 함수에서 기능 처리, this 사용 가능
- forEach()는 인덱스 0부터 시작
  - for()와 같이 인덱스 증가 값을 조정할 수 없음
  - 뒤에서 앞으로 읽을 수도 없음, 이것도 시맨틱 접근

프로그램은 코드가 아닌 시나리오로 풉니다.

[코딩 시간]

- 목적: 함수 호출 시간 측정
- [요구사항]
- 함수 코드가 없는 빈 함수 작성
  함수 이름: check()
- 배열에 1부터 1,000,000까지 작성
- forEach()로 배열을 반복하면서
  check() 함수 호출
  즉 1,000,000번 check() 함수 호출
- 반복이 끝나면 실행 시간을 출력하세요.
  종료 시각 - 시작 시각 힌트

true, false를 반환하는 메소드

every()

구분	데이터(값)
data	반복 대상
파라미터	콜백 함수
	this로 참조할 오브젝트opt
반환	true, false

- forEach()처럼 시맨틱 접근
- 배열의 엘리먼트를 하나씩 읽어가면서
  - false를 반환할 때까지 콜백 함수 호출
  - 즉, false가 반환되면 반복 종료
  - false를 반환하지 않으면 true 반환 코드 1
- false가 되는 조건이
  배열의 앞에 있을 때 효율성 높음

some()

구분	데이터(값)
data	반복 대상
파라미터	콜백 함수
	this로 참조할 오브젝트opt
반환	true, false

- every()처럼 시맨틱 접근
- 단, true를 반환할 때까지 콜백 함수 호출
  - 즉, true가 반환되면 반복 자동 종료
  - true를 반환하지 않으면 false 반환 코드 1
- true가 되는 조건이
  배열의 앞에 있을 때 효율성 높음

필터, 매핑

filter()

구분	데이터(값)
data	반복 대상
파라미터	콜백 함수
	this로 참조할 오브젝트opt
반환	[true일 때의 엘리먼트]

- forEach()처럼 시맨틱 접근
- 배열의 엘리먼트를 하나씩 읽어가면서
  - 콜백 함수에서 true를 반환하면
    현재 읽은 엘리먼트를 반환 코드 1
- 조건에 맞는 엘리먼트를 추려낼 때 유용

map()

구분	데이터(값)
data	반복 대상
파라미터	콜백 함수
	this로 참조할 오브젝트opt
반환	[콜백 함수에서 반환한 엘리먼트]

- forEach()처럼 시맨틱 접근
- 배열의 엘리먼트를 하나씩 읽어가면서
  - 콜백 함수에서 반환한 값을 새로운 배열에 첨부하여 반환 코드 1

반환 값을 파라미터 값으로 사용

reduce()

구분	데이터(값)
data	반복 대상
파라미터	콜백 함수
	초깃값opt
반환	콜백 함수에서 반환한 값

- forEach()처럼 시맨틱 접근
- 배열 끝까지 콜백 함수 호출
  - 파라미터 작성 여부에 따라 처리가 다름

reduce()

- 콜백 함수만 작성한 경우
  - 즉, 파라미터를 하나만 작성
- 처음 콜백 함수를 호출할 때
  - 인덱스 [0]의 값을 직전 값에 설정
  - 인덱스 [1]의 값을 현재 값에 설정
  - 인덱스에 1을 설정
- 두 번째로 콜백 함수를 호출할 때
  - 콜백 함수에서 반환된 값을 직전 값에 설정
  - 인덱스 [2]의 값을 현재 값에 설정

[코드 1] 첫 번째 파라미터만 작성

```
var value = [1, 3, 5, 7];
var fn = function(prev, curr, index, all){
  log(prev + "," + curr);
  return prev + curr;
};
var result = value.reduce(fn);
log("결과:", result);
```

1. 4번이 아니라 3번 반복한 것은
   처음 시작할 때 인덱스가 1이기 때문입니다.

[실행 결과]

```
1,3
4,5
9,7
결과:16
```

reduce()

- 두 번째 파라미터를 작성한 경우
- 처음 콜백 함수를 호출할 때
  - 두 번째 파라미터 값을 직전 값에 설정
  - 인덱스 [0]의 값을 현재 값에 설정
  - 인덱스에 0을 설정
- 두 번째로 콜백 함수를 호출할 때
  - 콜백 함수에서 반환된 값을 직전 값에 설정
  - 인덱스 [1]의 값을 현재 값에 설정

[코드 1] 두 번째 파라미터 작성

```
var value = [1, 3, 5];
var fn = function(prev, curr, index, all){
  log(prev + "," + curr);
  return prev + curr;
};
var result = value.reduce(fn, 7);
log("반환:", result);
```

1. 두 번째 파라미터에 초깃값으로 7을 작성
2. 처음 콜백 함수를 호출할 때
   두 번째 파라미터 값 7을 prev에 설정
   prev:7, curr:1, index:0, 반환값:8
3. 두 번째 콜백 함수를 호출할 때
   prev:8, curr:3, index:1, 반환값:11

[실행 결과]

```
7,1
8,3
11,5
반환:16
```

reduceRight()

구분	데이터(값)
data	반복 대상
파라미터	콜백 함수
	초깃값opt
반환	콜백 함수에서 반환한 값

- reduce()와 처리 방법 같음
- 배열 끝에서 앞으로 하나씩 읽어가면서
  - 콜백 함수에서 반환한 값을 반환 코드 1