

- [읽기](#)
- [페이지 편집](#)
- [히스토리](#)

카페북 검색

제목+내용

reduce

업데이트
2020.06.16. 19:16
참여저자
[성빈](#), [도미 doami2005](#) [상세보기](#)
댓글
[바로가기](#)

[책갈피](#)
<https://cafe.naver.com/nameyee/book5105233/15340> 주소복사

reduce(callback (함수), initialValue(초깃값))
총 2개의 인자가 들어갈 수 있습니다.

callback (함수)
배열의 각 요소에 대해 실행할 함수
아래 4개의 인자가 들어갈 수 있습니다.

• accumulator (누적값)
- 반환값을 누적합니다.

• currentValue (현재값)
- 처리할 현재 요소

• currentIndex (현재인덱스)
- 처리할 현재 요소의 인덱스.

initialValue 를 제공한 경우 0, 아니면 1로 시작합니다.

• array (배열)
- reduce 를 호출한 배열

initialValue (초깃값)
callback 의 최초 호출에서 첫번째 인수에 제공하는 값
생략하면 배열의 첫번째 요소를 사용합니다.

작동 원리
[1, 2, 3, 4].reduce((a, b, c, d) => a + b) 일 때,

callback	a	b	c	d
1번째 호출	0	1	1	[1, 2, 3, 4]
2번째 호출	1	2	2	[1, 2, 3, 4]
3번째 호출	3	3	3	[1, 2, 3, 4]
4번째 호출	6	4	4	[1, 2, 3, 4]

결과 : 10

arr = [{n:1}, {n:3}, {n:2}]

이 배열에서 모든 객체의 n의 값들을 더하고 싶다면,

arr.reduce((a, b) => a+n+b.n)

으로 하시면 안됩니다.

그 이유는 원리를 보시면 알 수 있습니다.

초기값을 지정하지 않았으므로 {n:1} 부터 순회를 시작하여

[{n:1}, {n:3}, {n:2}]

1+3
(1+3).n+2

이런식으로 되어 NaN(Not a Number) 이라는 결과가 나오게 됩니다.

그러면 초기값을 설정한다면?

arr.reduce((a, b) => a.n+b.n, 0) 이거도,

arr.reduce((a, b) => a.n+b.n, {n:0}) 이거도
절대 되지 않습니다.



어차피 도중에 (1+2).n+3 가 나올 수 밖에 없기 때문입니다.

따라서 arr.reduce((a, b) => a+b.n, 0)

이런식으로 만들어 준다면
0+1
0+1+2
0+1+2+3
결과 : 6
이 나오는 걸 확인할 수 있습니다.





카톡봇을 만들 때에는 2개의 인자만 넣어 배열의 모든 요소를 더할 때 말고는 거의 사용하지 않습니다.

reduceRight 이라는 함수도 있으며,
reduce 와 차이점은 순회를 오른쪽부터 왼쪽으로 합니다.



댓글영역

댓글 [이동](#) [등록순](#) | 조회수 48 |

 |  [인쇄](#) [신고](#)

- [▲ top](#)
- [↗이전글](#) [Array\(배열\)](#) [1]
- [↘다음글](#) [파싱](#) [1]

도미 doami2005	2020.04.05.
도미 doami2005	2020.04.08.