

9일간의 자바스크립트

배열 메소드

함수형 복습

뭐... 하고 있는게 함수형 프로그래밍이긴 하지만

- map
 - 배열에 각각 함수를 적용시킨다
- reduce
 - 배열에 각각 함수를 적용시키면서 합친다.
- filter
 - 배열의 원소 중에서 함수의 결과가 참인 원소만 만든다.

추상화란?

- map, reduce, filter가 전부 추상화를 한 함수들이다.
- 쉽게 말해서 추상화란 공통적인 부분을 따로 분리하는 거라고 보면 된다.
- 객체 지향에서 말하는 추상화는 사각형, 삼각형, 원을 도형으로 추상화 하는 것이다.
- 함수형에서 말하는 추상화는 함수가 하는 공통적인 일을 분리하는 것이다.

함수형 복습

- `map()`, `reduce()`, `filter()`로 절차적으로 짠 것을 다시 풀어봅시다!
 - list에서 홀수를 `length`만큼 뽑아서 제공한 후 모두 더하기
 - 배열의 길이 구하기
 - `sum`, `product`를 `reduce`를 사용해서 구현하기

배열 메소드

사용하는 방법이 달라졌어요!

- map, reduce, filter 함수 대신에 배열 메소드를 사용
- 이는 자바스크립트에 포함된 기능으로 쓸 때마다 직접 구현하지 않아도 된다.
 - 단점은 배열에서만 사용할 수 있다.

배열 메소드

함수형으로 만들어보기 전에..

- 배열을 다루는 더 다양한 메소드가 존재합니다
- find: 찾기 (하나만)
- every: 모두 함수 조건을 만족하는가? (all)
- some: 하나라도 조건을 만족하는가? (any)
- flat: 펴기
- slice: 자르기
- splice: 자르고 집어넣기... 별로 추천하진 않아요 -> immutable을 만족하지 않음