

# 9일간의 자바스크립트

이제부터 진짜 시작

# 함수형 프로그래밍이란?

## Functional Programming (FP)

- 흔히 명령형(Imperative), 절차적(Procedural) 프로그래밍과 대비되는 개념
- 함수를 엮어서 프로그래밍 한다는 의미이다
- 명령형 <-> 선언형  
절차적 <-> 함수형

# 왜 함수형인가

- 사실 함수형 프로그래밍 언어는 생각보다 일찍 등장했다. (LISP, 두번째 프로그래밍 언어)
- 그 때는 뜨지 못했지만 지금은 함수형 프로그래밍이라는 것이 뜨기 시작했다.
  - 하드웨어 성능, 컴파일러가 좋아졌다.
  - 동시, 병렬성과 관련된 기술이 발전 됐다.
    - 특히 병렬을 다루면서 야기된 문제가 같은 메모리에 있는 값을 서로 동시에 수정하려고 하는 상황이 발생하는데, 함수형 프로그래밍의 특징이 이와 반대로 바뀌지 않는 변수를 지향하고 있다.

# 왜 함수형인가

- 함수형으로 작성을 하면 코드가 간결해진다.
- 자연어로 말하자면 동사를 여러개 만들고, 그 동사를 조합해서 코드를 작성하는 것이다.
  - 명령형은 '어떻게'에 초점이 맞춰져 있다면, 선언형 프로그래밍은 '무엇을'에 집중
- 함수형 프로그래밍 언어는 현대적이다. (최신 연구결과가 많이 반영)
- 부작용이 없어서 안전하다.

# 사실 이미 함수형 프로그래밍을 해봤다

- 재귀도 함수형 프로그래밍의 일부!

# 잠깐 쉬어가는 타임

- 화살표 함수 복습

# 왜 자바스크립트인가

## 조금 어려운 개념이 많이 등장합니다

- 함수형 프로그래밍을 할 수 있는 조건은 다음과 같다.
  - 1급 함수: 함수가 1급 객체여야 한다.
  - 순수 함수: 부작용이 없어야 한다. -> 멍등성. 불변성
  - 재귀 함수
  - 커링
- 이 같은 조건을 쉽게 만족시킬 수 있으면서 배우기 쉬운 언어가 자바스크립트이다.

# 1급 객체

- 변수에 대입할 수 있다
- 인수로 전달할 수 있다
- 반환값으로 쓸 수 있다
- 동적으로 할당이 가능하다
- 항상 정해진 이름이 있지 않다. (그 자체로 고유하다)



# 또 다른 함수형 언어

- LISP
- F#: 망해간대요..
- Scala!!
- Elixir: 분산 처리에 특화
- Haskell: 너무 어렵다 -> 그래도 함수형 프로그래밍을 제대로 해보고 싶으면 꼭 하세요 꼭
  - 진짜 유일한 함수형 언어
  - 정적 타이핑 (안 맞으면 컴파일이 안 됨)



# 함수형 프로그래밍을 잘 하는 방법

- 모든 데이터는 변경이 안 된다고 생각한다

- 함수는 순수 함수로 만드는 연습

- 코드가 간결하지만 읽는 연습이 필요하다.

- 함수 개념이 항상 먼저다.

- 함수들을 분리하고, 추상화 시키고, 조합시키는 것이 필요하다.

- 함수형 프로그래밍에서 함수는 수학에서 말하는 함수랑 거의 일치한다.

$$Program(x) = (f \circ g \circ h \circ i)(x)$$

$$Program(x) = f(g(h(i(x))))$$

# 함수형 프로그래밍의 단점

- 모든 것을 함수형 프로그래밍으로 작성할 수는 없다.
- 입출력이 그런 예이다. (하스켈은 그 마저..)
- 개념이 무척 어렵다.

# 함수의 종류

- 우리가 잘 아는 함수 형태  $f(x), v(t)$
- prefix  $\sin \theta$
- postfix  $n!$

# 실전

- 명령형 코드와 함수형 코드를 비교

# 어떻게 쓰이는가?

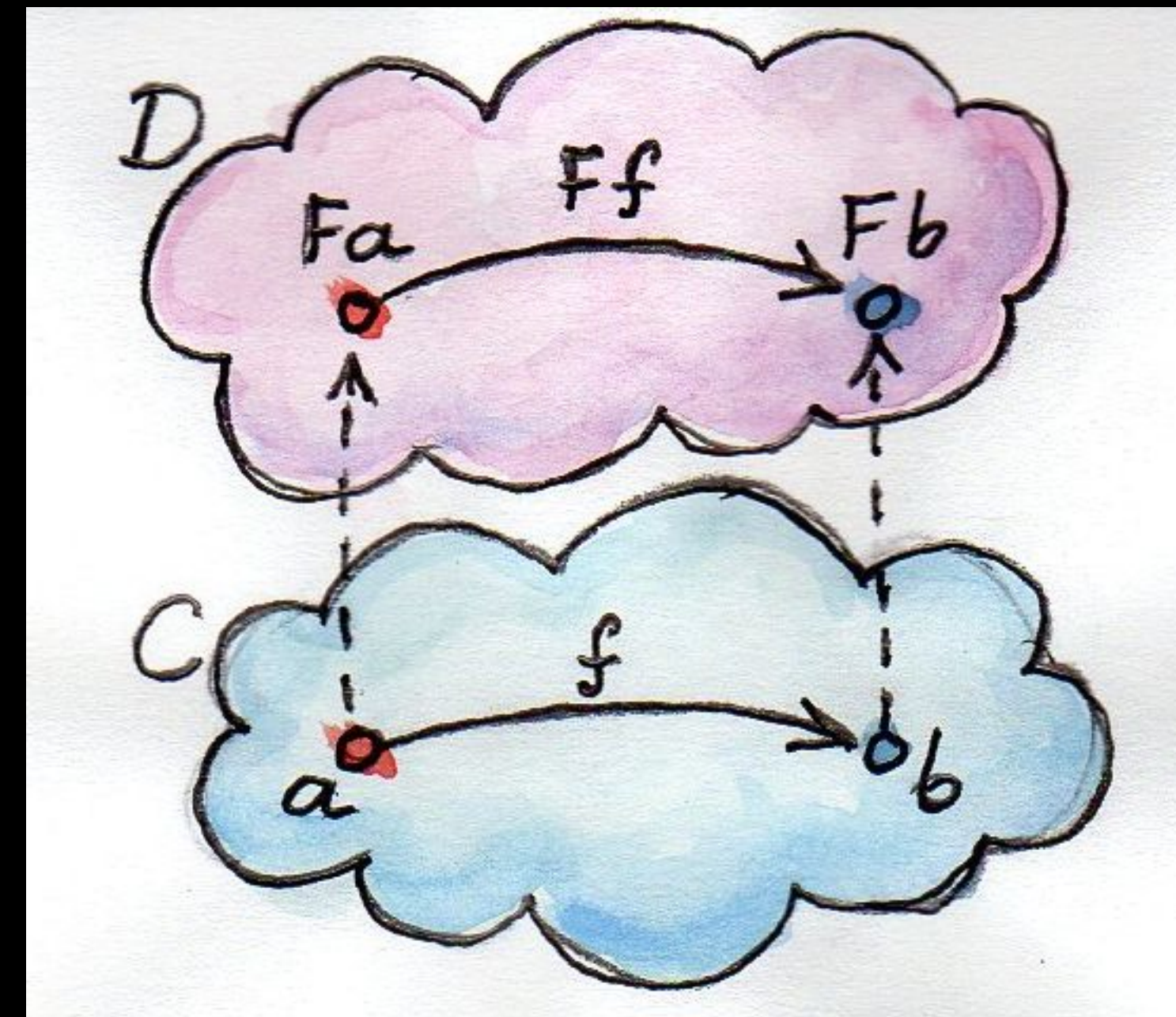
- 많이 쓰인다.
- 특히 프론트엔드, 리액트를 하는데 함수형 프로그래밍은 거의 필수라고 본다.
  - 함수형의 특징 불변성에 기반

# 직접! 만들어 보면서

- High Order function을 만들어 봅시다.
- High Order function이란, 함수를 인자로 받아서 함수를 반환하는 함수

# map

- 매핑, 함자(functor), 함수(function)
- 똑같은 함수를 적용시키는 것이라고 보면 된다.





# reduce

다른 언어에서는 fold

- 다른 언어 처럼 접는다는 의미이다.
- 또는 차원을 줄인다는 의미도 될 수 있다.
- 배열(1차원)에 reduce를 쓰면 0차원인 하나의 값으로 나올 것이다.

# filter

- 어떤 함수를 받아서 그것을 만족하는 것만 반환하는 함수이다.

# length

- 어떤 순회 가능한 객체(Iterable)의 길이를 구하는 함수
- 가능한 꼬리 재귀를 사용 (함수 그 자체만의 실행 결과가 반환되게끔)

끝

$$\begin{aligned} \textit{Program}(x) &= (f \circ g \circ h \circ i)(x) \\ \textit{Program}(x) &= f(g(h(i(x)))) \end{aligned}$$

# 내일은

- 한글날!
- 선택적으로 들으시면 됩니다.
- 오늘 구현한 map, reduce, filter 말고 go라는 함수에 대해, 느긋한 연산에 대해 볼 예정입니다.