

# How FP Deals With Effects

---

\* PoolC 양제성

Source:  2024/01/24 (Wed)

# 목차

---

## 1st Session

1. 함수형 프로그래밍 Intro
  - Overall Structure
  - Historical Review (CS + Math)
2. SW 엔지니어링의 목표
  - SW Maintainability
  - FP vs OOP vs PP
3. FP는 정말 순수한가?
  - Purity of Functions
  - File I/O Scenario

*Basic Haskell Knowledge*

## 2nd Session

1. 함수 합성을 위한 도구들
  - Partial Application
  - Kleisli Composition
2. ...중 하나인 모나드
  - Functor to Monad
  - IO Monad
3. 부수 효과의 관리
  - Action / Calculation / Data
  - Preventing Action Propagation

FP is all about **composing pure functions**.

```
int main(void) {  
    f(); g(); h(); ..  
}
```

[Procedural Programming]

VS

$f(g(h(\dots)))$

[Functional Programming]

FP is all about **composing pure functions**.  How?

```
int main(void) {  
    f(); g(); h(); ..  
}
```

[Procedural Programming]

VS

$f(g(h(\dots)))$

[Functional Programming]