

04

## 04-1. 함수형 프로그래밍이란

*Functional Programming*

# 함수형 프로그래밍이란?

---

## 함수형 프로그래밍 정의

- 절차지향 프로그래밍 : 알고리즘과 로직을 중심으로 문제를 해결하기 위함을 주 목적으로 하는 프로그래밍 기법. 대표적 언어가 C, Pascal 등
- 객체지향 프로그래밍 : 클래스 선언을 최 우선으로 하는 프로그래밍 기법. 데이터와 데이터를 처리할 메서드를 한데 묶어 객체를 만들고 객체를 조합해서 프로그래밍 작성. 객체를 만들기 위한 추상화, 캡슐화, 상속성, 다형성등의 다양한 개념 제공. 대표적 언어로 자바, C++ 등
- 함수형 프로그래밍 : 함수 선언을 최 우선으로 하는 프로그래밍 기법. 데이터의 흐름에 초점을 맞추지 않고 함수의 선언과 선언된 함수의 유기적인 흐름을 주 목적으로 함. 대표적인 언어로 코틀린, 스칼라, 스위프트 등

# 함수형 프로그래밍이란?

---

## 함수형 프로그래밍의 주요 원칙

- First Class Citizen 으로서의 함수
- 순수함수 (Pure Function) 으로 정의되는 함수

## 함수형 프로그래밍에서의 데이터

- 데이터는 변경되지 않으며 프로그램의 상태만을 표현한다. (immutable state로서의 데이터)
- 함수에서 데이터는 변경하는 것이 아니라 새로운 데이터를 만들어 리턴한다.

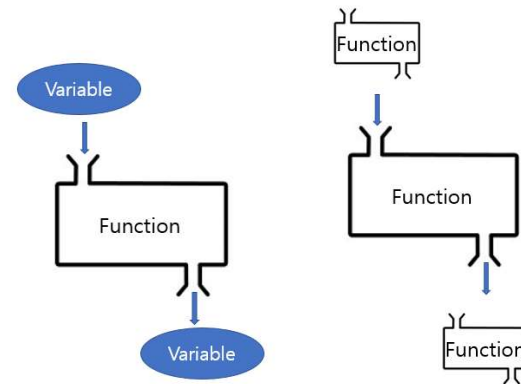
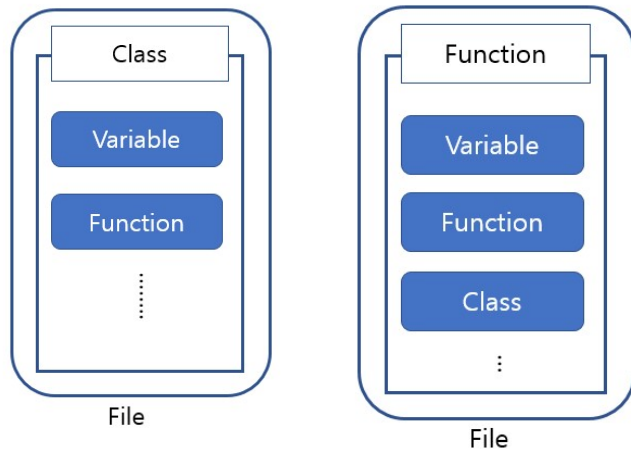
## 함수형 프로그래밍의 이점

- 코드가 간결하여 개발생산성 및 유지보수성이 증대된다.
- 동시성 작업을 좀더 쉽고 안전하게 구현할수 있다.

# 함수형 프로그래밍이란?

## 일급 객체로서의 함수

- 프로그램의 top - level에 함수를 정의
- 모든 구성요소를 함수안에 작성
- 함수를 변수처럼 이용



# 함수형 프로그래밍이란?

---

## 순수 함수로 정의되는 함수

- 순수 함수는 Side-Effect이 발생하지 않는 함수
- 함수 외부의 다른 값을 변경하지 않는다.
- 함수 내부에서 별도의 입력/출력이 발생하지 않는다. (파일, 데이터베이스, 네트워크등)

```
fun some(a: Int): Int {  
    return (Math.random()*a).toInt()  
}
```

```
fun some1(a: String): Boolean {  
    try {  
        val file=File("a.txt")  
        val out=FileWriter(file);  
        out.write(a)  
        out.flush()  
        return true  
    }catch (e: Exception){  
        return false  
    }  
}
```

# 함수형 프로그래밍이란?

---

## 순수 함수로 정의되는 함수

```
fun some2(a: Int, b: Int): Int {  
    return a+b  
}
```

```
fun some3(): Int {  
    return 10  
}
```

```
fun some4(a: Int, b: Int) {  
    val result=a + b  
}
```

# 함수형 프로그래밍이란?

---

## 코틀린에서 일급 객체로서의 함수

- 다양한 구성요소를 포함하는 함수

```
fun superFun(){
    val superData="hello"
    fun subFun1(){
        println("subFun1() .. superData : ${superData}")
    }
    fun subFun2(a: Int, b: Int): Int{
        subFun1()
        return a+b
    }

    class SubClass {
        fun classFun(){
            println("classFun() .. superData : ${superData}")
        }
    }
    subFun1()
    SubClass().classFun()
}
```

# 함수형 프로그래밍이란?

---

## 변수처럼 이용되는 함수

```
val dataVal=10

val funVal=fun someFun() {//error
    //.....
}
```

```
val funVal1= { x1: Int ->
    println("hello world")
    x1 * 10
}
funVal1(10)
```

```
fun someFun() {
    println("i am some Function")
}

val funVal2=::someFun

funVal2()
```





# 감사합니다

단단히 마음먹고 떠난 사람은  
산꼭대기에 도착할 수 있다.  
산은 올라가는 사람에게만 정복된다.



윌리엄 셰익스피어  
William Shakespeare