

코틀린은 무엇인가?

코틀린 기초강좌

강사: 배정만

강의목차

- 프로그래밍 언어
- 왜 코틀린인가?
- 코틀린 맛보기

프로그래밍 언어

1세대: 71세대

2세대: 어셈블리어

3세대

C언어 (메모리 관리 X)
Java, C# (메모리 관리 O)

4세대(?)

절차형 + 함수형 = 동시성(쉽게) +
Tool Support/Dynamic debug (휴먼 에러를 줄여 줌)
Python, Ruby, Swift, Kotlin ...

5MICH(?)

왜 코틀린인가?

왜 코틀린인가?

- 메모리 관리를 넘어
- 최근 언어 트렌드 반영

C, C++



Java, Python



Swift, Kotlin

코틀린 코드

What does it look like?

Concise, simple and very easy to read (and write)

```
package hello
```

```
fun main(args: Array<String>) {  
    println("Hello World!")  
}
```


코틀린의 철학 1

- 간결함

- 개발시 코드의 양을 대폭 줄임

코틀린의 철학 2

- 안전함

- NullPointerException에서 자유로움

코틀린의 철학 3

- 다양한 운용 가능
 - Android, JVM 컴파일 가능
 - Javascript, iOS 개발 가능

코틀린의 철학 4

- 개발 도구 친화적임
 - IntelliJ 개발사에서 직접 개발
 - 다양한 코드 보조 도구 활용 가능

코틀린의 특징 1

● 정적 타이핑

성능

신뢰성

유지보수성

도구지원

코틀린의 특징 2

• 함수형 프로그래밍

일급함수

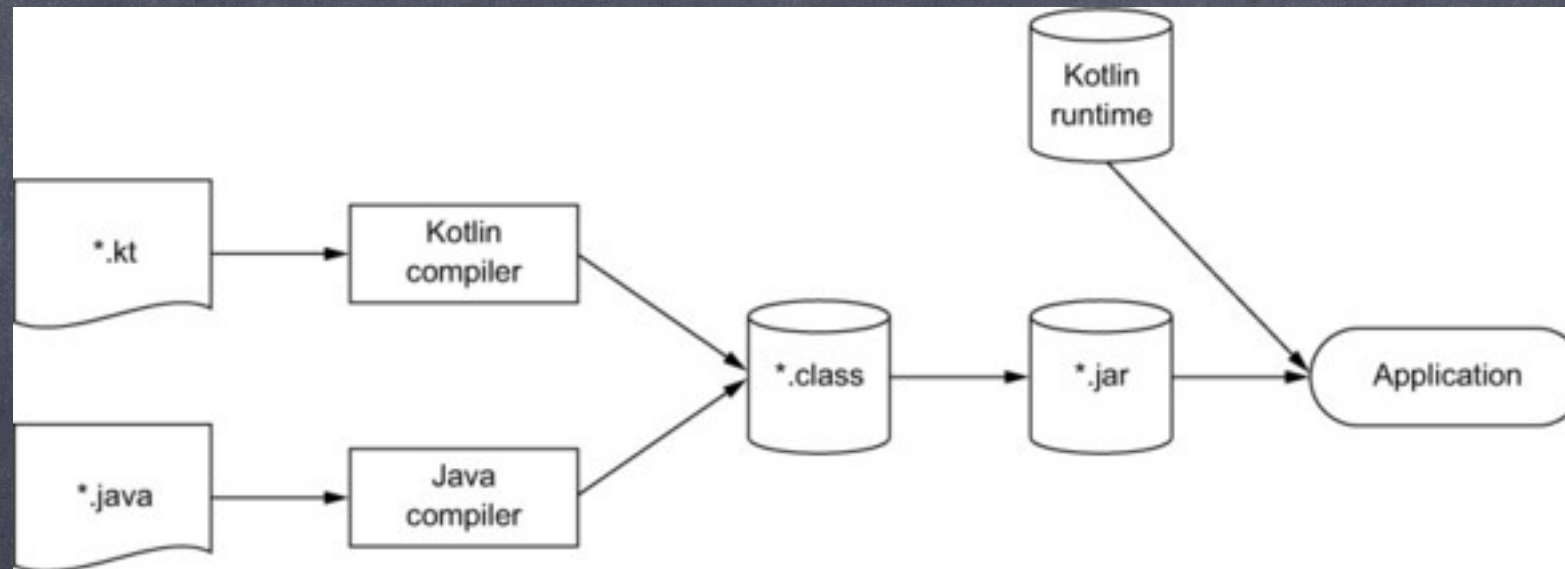
부작용 없음

불변성

코틀린의 특징 3

• 무료 & 오픈소스

코틀린 빌드 프로세스



👁 코틀린 확장자 .kt

코틀린 맛보기

Main 함수

함수이름

파라미터

리턴타입

```
fun max (a : Int, b : Int): Int {  
  return if (a > b) a else b  
}
```

함수바디

더 간결한 함수

```
fun max (a: Int, b: Int) = if (a > b) a else b
```


코틀린 학습트리

