



1부

1. Hello world

```
fun main()
    println("Hello, world")
}
```

- java 와 비교해서 식 뒤 세미클론 필요 없음
- System.out.println 말고 println 쓰면 됨

var, val, 데이터 타입

```
// 상수 정의 타입을 지정해도 안해도 된다.
val 식별자: 타입 = 초기화식
val 식별자 = 초기화식

// 변수 정의 타입을 지정해도 안해도 된다.
var 식별자: 타입 = 초기화식
var 식별자 = 초기화식

// 데이터 타입 변경시 에러 발생
var num = 20
// 에러가 발생
num = 20.5

// String Type 은 ak (삼중 큰따옴표 String, raw String) 이 추가되었다.
val str = """i wanna backend dev
in NAVER FINANCIAL
"""

// auto boxing 도 해준다.
val num1 = 1;
val num2 = 1 + 1.2;
// num2 = 2.2
```

- java 와 비교하면 데이터 타입을 **타입 추론** 해줌으로 편함
- 하지만 변수에 타입이 변경되는 경우를 주의

함수

```
// 기본식 - 블록 본문
// Type, ReturnType 생략 가능
fun functionName(args: Type1, args: Type2, ...): ReturnType {
    // 생략
    return result;
}

// 축약식 - 식 본문
// Type, ReturnType 생략 가능
fun functionName(args: Type1, args: Type2, ...): ReturnType = 결과 식
```

- java 와 비교하면 결과 식이 추가되었고, 접근 제어자, return type 을 지정하지 않아도 됨.

if

```
fun main() {
    val num = 10
    val result = if (num > 100) 1 else 5
}

// 함수와 결합하면 아래 처럼 사용 가능하다.
fun text(exp: Boolean): String = if (exp) "정답" else "오답"
```

- java 와 비교하면 별 다른게 없지만 위 처럼 초기화시 사용 가능

문자열 템플릿

```
// 문자열("") 에 $(식별자(변수, 상수), 식)를 넣으면 값이 치환
fun main() {
    val answer = 42
    println("Found $answer!")
    // Found 42!

    val condition = true
    println("${if (condition) 'a' else 'b'}")
    // a

    val x = 11
    println("$x + 4 = ${x + 4}")
    // 11 + 4 = 15
}
```

- java 와 비교하면 \$식별자, \$식으로 넣을 수 있다.

수 타입

```
fun main() {
    // Int 타입 추론
```

```

val i = 0
// L을 사용해 Long 타입
val l1 = 0L
// 명시적 타입 지정
val l2: Long = 0
println("$l1 $l2")
}

```

- java 와 별 다르게 없다.

불리언

- java 와 별 다르게 없다.

while 로 반복하기

- java 와 별 다르게 없다.

루프와 범위

```

fun showRange(r: IntProgression) {
    for (i in r) {
        print("$i ")
    }
    println()
}

fun main(){
    // 1 2 3 4 5
    showRange(1..5)
    // 0 1 2 3 4
    showRange(0 until 5)
    // 5 4 3 2 1
    showRange(5 downTo 1)
    // 0 2 4 6 8
    showRange(0..9 step 2)
    // 0 3 6 9
    showRange(0 until 10 step 3)
    // 9 6 3
    showRange(9 downTo 2 step 3)
}

```

- java 와 비교하면 **..**, **until**, **downTo**, **step**

in 키워드

```

fun main() {
    val percent = 35
    // true
    println(percent in 1..100)
}

```

```
// true
println("ab" in "aa".. "az")
}
```

- DB in 같은 느낌이다.

식과 문

```
fun unitFun() = Unit

fun main() {
    // kotlin.Unit
    println(unitFun())
    // 42
    val u1: Unit = println(42)
    // kotlin.Unit
    println(u1)
    // 0
    val u2 = println(0)
    // kotlin.Unit
    println(u2)
}
```

- 연습문제

