

به نام خدا

کارگاه آموزشی :

مزیت‌های زبان kotlin نسبت به جاوا

هادی بیگی (hdbeigy@gmail.com)

همایش گنو/لینوکس و نرم افزار های آزاد

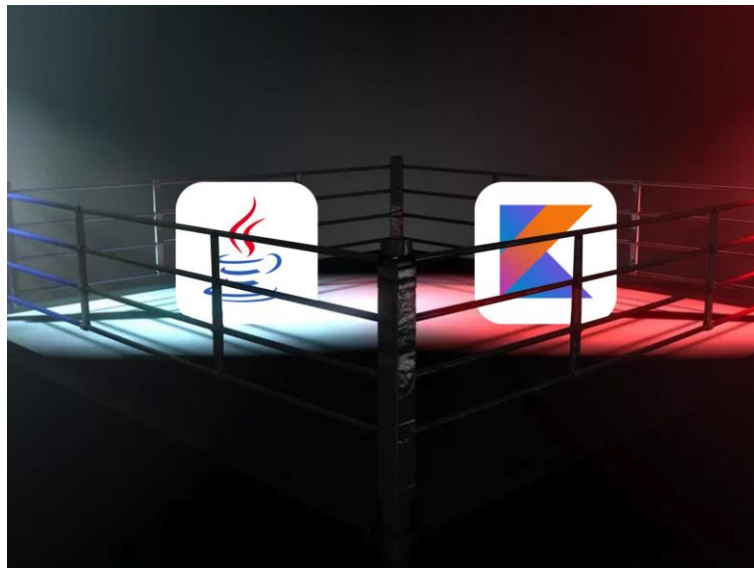
جشن انتشار اوبونتو ۱۷,۰۴

تابستان 96

هدف این کارگاه معرفی زبان کوتلین به عنوان زبان آینده ی توسعه ی اندروید به آن دسته از توسعه دهندگان است که هنوز برای این تغییر شک دارند.

## کوتلین؟ یعنی چی؟

- کوتلین یک زبان برنامه‌نویسی از نوع ایستا است که بر روی ماشین مجازی جاوا (JVM) اجرا می‌شود. همچنین می‌تواند به سورس کد جاوا اسکرپت نیز کامپایل شود.
- اولین توسعه ی ان از تیم برنامه‌نویسان JetBrains واقع در سنت پترزبورگ روسیه است (این نام از روی جزیره کوتلین در نزدیکی سنت پترزبورگ می‌آید). کوتلین در ژانویه سال ۲۰۱۲ به عنوان زبان ماه انتخاب شد
- با وجود آنکه از نظر syntax با جاوا سازگار نیست، کوتلین طراحی شده تا با کد جاوا همکاری داشته باشد و متکی بر کد جاوا از کتابخانه کلاس جاوا است، مانند مجموعه فریم ورک‌ها.
- مشابه با Scala و Pascal, Haxe, PL/SQL, F#, Go



-

## چند اصل ساده ولی جالب! :

### Datatypes :

Type	Bit width
Double	64
Float	32
Long	64
Int	32
Short	16
Byte	8

String

Boolean

....

### Explicitly Conversions :

- `toByte(): Byte`
- `toShort(): Short`
- `toInt(): Int`
- `toLong(): Long`
- `toFloat(): Float`
- `toDouble(): Double`
- `toChar(): Char`

### Arrays :

```
// Creates an Array<String> with values ["0", "1", "4", "9", "16"]
val asc = Array(5, { i -> (i * i).toString() })

val x: IntArray = intArrayOf(1, 2, 3)
x[0] = x[1] + x[2]
```

## Control Flow:

If:

```
// Traditional usage
var max = a
if (a < b) max = b

// With else
var max: Int
if (a > b) {
    max = a
} else {
    max = b
}

// As expression
val max = if (a > b) a else b
```

When:

```
when (x) {
    1 -> print("x == 1")
    2 -> print("x == 2")
    else -> { // Note the block
        print("x is neither 1 nor 2")
    }
}
```

```
when (x) {
    0, 1 -> print("x == 0 or x == 1")
    else -> print("otherwise")
}
```

For :

```
for (i in array.indices) {
    print(array[i])
}

for ((index, value) in array.withIndex()) {
    println("the element at $index is $value")
}
```

While – do While :

```
while (x > 0) {
    x--
}

do {
    val y = retrieveData()
} while (y != null) // y is visible here!
```

آنچه کوتلین را جذاب میکند!! :

Null safety

```
var a: String = "abc"
a = null // compilation error

var b: String? = "abc"
b = null // ok
```

Elvis operator:

```
val l: Int = if (b != null) b.length else -1

val l = b?.length ?: -1
```

Safe Casts:

```
val aInt: Int? = a as? Int
```

Extension Functions:

```
fun MutableList<Int>.swap(index1: Int, index2: Int) {
    val tmp = this[index1] // 'this' corresponds to the list
    this[index1] = this[index2]
    this[index2] = tmp
}
```