### L1.IDEA部署Kotlin环境

## IDEA环境部署

### 1.1 下载IDEA编译器下载最新版的就可以了

<https://www.jetbrains.com/idea/>

### 1.2 安装IDEA插件

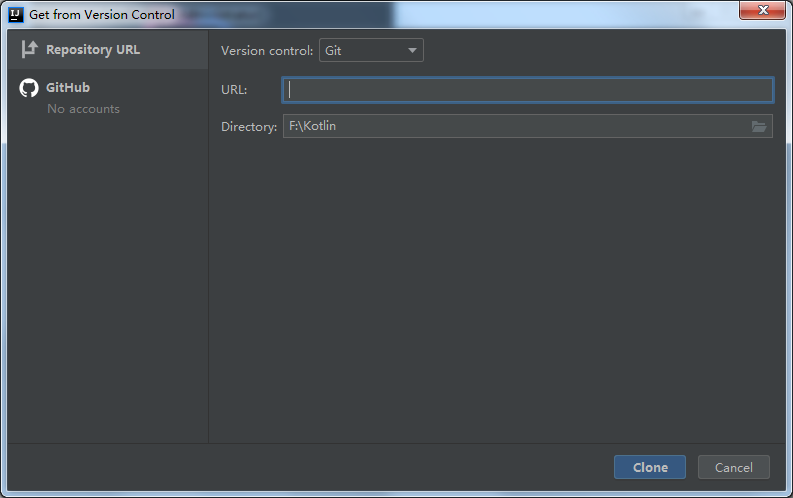
<https://zhile.io/2020/11/18/jetbrains-eval-reset-da33a93d.html>

### 1.3 获取教程项目

完成以上部署后获取教程项目：

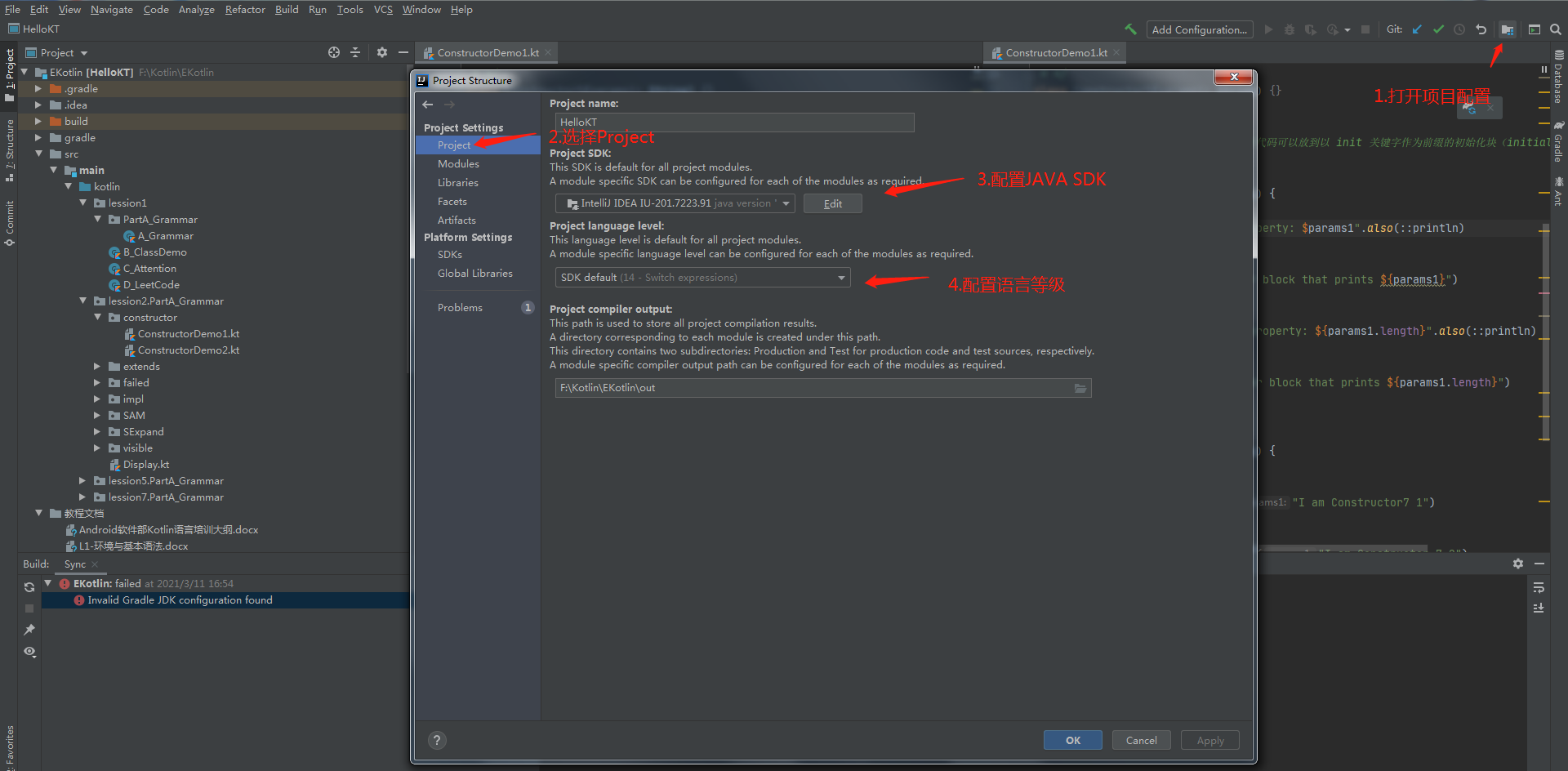


输入: <https://github.com/JustGank/EKotlin.git>



正常Build获取即可。

### 1.4 获取完成配置项目



配置完成后项目不报错，代表环境部署成功。

此时就具备了Kotlin语法的实验环境。

## 二、基本语法介绍

此部分作为Kotlin的大体语法展示单元，主要是展示Kotlin在常规编程中的使用方式。此部分讲解时需要体现特性和重点，部分细节可以放到后面单元专项中进行讲解。

### 2.1 包的定义与导入

包的声明应处于源文件顶部：

package my.demo

import kotlin.text.\*

此部分和JAVA类似，不用特殊关注

### 2.2 程序的入口点

Kotlin 应用程序的入口点是 main 函数。

fun main() {

println("Hello world!")

}

每一个文件中都可以有 fun main()的存在。这样可以很方便的让我们在每个文件内部完成展示和调试。

### 2.3函数

带有两个 Int 参数、返回 Int 的函数：

fun sum1(a: Int, b: Int): Int {

return a + b

}

Kotlin是先声明变量名，再声明变量类型。

将表达式作为函数体、返回值类型自动推断的函数：

fun sum2(a: Int, b: Int) = a + b

函数返回无意义的值,此处可以理解为Void,Unit也可以不写：

fun printSum(a: Int, b: Int): Unit {

println("sum of $a and $b is ${a + b}")

}

### 2.4变量

定义只读局部变量使用关键字 val 定义。**只能为其赋值一次。**

val a: Int = 1 // 立即赋值

val b = 2 // 自动推断出 `Int` 类型

val c: Int // 如果没有初始值类型不能省略

c = 3 // 明确赋值

可重新赋值的变量使用 var 关键字：

var x = 5 // 自动推断出 `Int` 类型

x += 1

顶层变量：

val PI = 3.14

var x = 0

fun incrementX() {

x += 1

}

### 2.5 注释

与大多数现代语言一样，Kotlin 支持单行（或行末）与多行（块）注释。

// 这是一个行注释

/\* 这是一个多行的

块注释。 \*/

Kotlin 中的块注释可以嵌套。

/\* 注释从这里开始

/\* 包含嵌套的注释 \*/

并且在这里结束。 \*/

### 2.6 字符串模板

此处就是$引用，直接引用变量时使用$即可，使用变量的某个变量时可以使用 {} 将要输入的内容包起来。

var a = 1

// 模板中的简单名称：

val s1 = "a is $a"

a = 2

// 模板中的任意表达式：

val s2 = "${s1.replace("is", "was")}, but now is $a"

### 2.7 条件表达式

下面两种写法等价：

fun maxOf(a: Int, b: Int): Int {

if (a > b) {

return a

} else {

return b

}

}

fun maxOf(a: Int, b: Int) = if (a > b) a else b（简单逻辑更推荐）