[kotlin基础入门](http://192.168.1.220:8090/pages/viewpage.action?pageId=3112997)

## 1、kotlin简介

Kotlin 是一个用于现代多平台应用的静态编程语言，由 [JetBrains](https://baike.baidu.com/item/JetBrains) 开发。  
Kotlin可以编译成Java字节码，也可以编译成JavaScript，方便在没有JVM的设备上运行。

## 2、kotlin特点

Kotlin，类似 Xtend 一样，旨在提供一种更好的 Java 而非重建整个新平台。这两种语言都向下编译为[字节码](https://baike.baidu.com/item/%E5%AD%97%E8%8A%82%E7%A0%81/9953683)（虽然 Xtend 是首先转换成相应的 Java 代码，再让 Java 编译器完成繁重的工作），而且两者都引入了函数和扩展函数（在某个有限范围内静态地增加一个新方法到某个已有类型的能力）。Xtend 是基于 Eclipse 的，而 Kotlin 是基于 IntelliJ 的，两者都提供无界面构建。能够首先演变到其他 IDE 的语言有可能成为最后的赢家。

## 3、基础语法

因kotlin可以无缝对接所有语言,所以在java中存在的jdk在kotlin中全部可以无障碍调用

## 4、变量

关键字 var:开启一个变量,在未赋值或者未进行变量类型确认前var 后面代入的变量名为Any类型(类似为java的object类型,但是在kotlin中var不仅仅为object类型,也可以为方法的变量属性)

举例:

var a = "a" //a为一个String类型的变量,a的属性值为"a"

var a = Student() //a为一个student对象类型,获取了student的所有属性对象

var a:String? = null //a必须为一个可为null的String类型,后续赋值不允许赋入非String的其他属性

var a  =null //a等于一个null,不具备任何类型的变量,它只为null,且后续也无法赋值除null以外的属性,如需赋予全部属性则需改写该初始化为 var a :Any? = null

关键字 val :开启一个加了final关键字修饰的属性,只允许在代码层进行一次初始化赋值,后期无法对该属性进行变更。

val id :Int = “1000”

## 5、方法

方法的关键符号为: {  }

一个空的大括号即为一个方法,方法可以直接写入var或val属性中: var a ={ }

一般方法定义: 一个名字叫做a的方法,返回值为String类型

fun a:String{  
    return ""  
}

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

return if(a>b) a else b

}

[kotlin入门扩展](http://192.168.1.220:8090/pages/viewpage.action?pageId=3112993)

print("") //日志打印

println("") //日志打印换行

class类反射java类参考java方法

var a = { return "a" }  a的类型为String类型,在有返回值方法赋值给属性时,属性的类型为返回值类型

switch语法:

在kotlin也有与java内swicth语法相同的方法,而且功能比java的switch语法更加强大,我们使用when

## 一般样例

|  |
| --- |
| when(a){      is "a" ->{          print("这是a")      }      is "b" ->{          print("这是b")      }  } |

## 2、进阶样例

|  |
| --- |
| var a:Any? = null  a = "a"  when(a){      is String ->{          print("a是String变量")      }      is Int ->{          print("a是int变量")      }      else ->{          print("不知道a是什么变量")      }  } |

## 3、循环遍历:

kotlin的遍历方法从jdk1.8版本开始借鉴,使用更加高效的便利方法

|  |
| --- |
| [map.forEach](http://192.168.1.220:8090/pages/map.forEach) { t, u ->    }  [collection.forEach](http://192.168.1.220:8090/pages/collection.forEach) {    } |