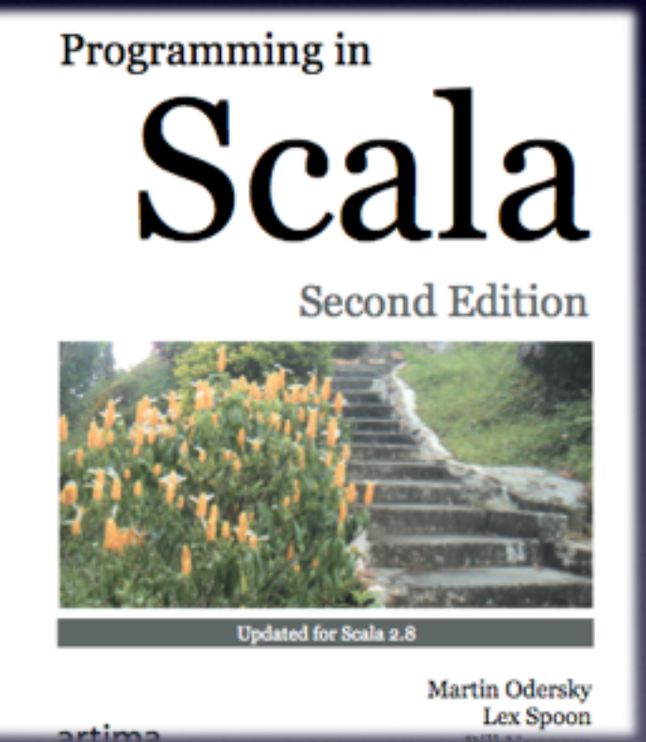
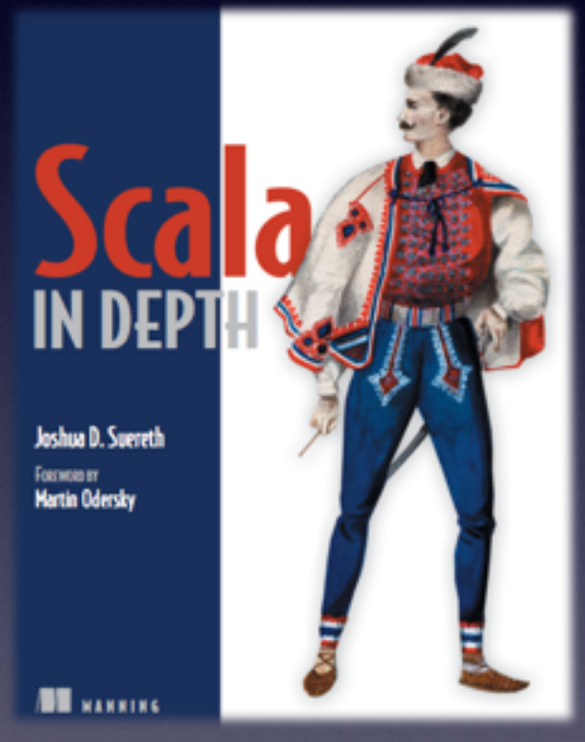
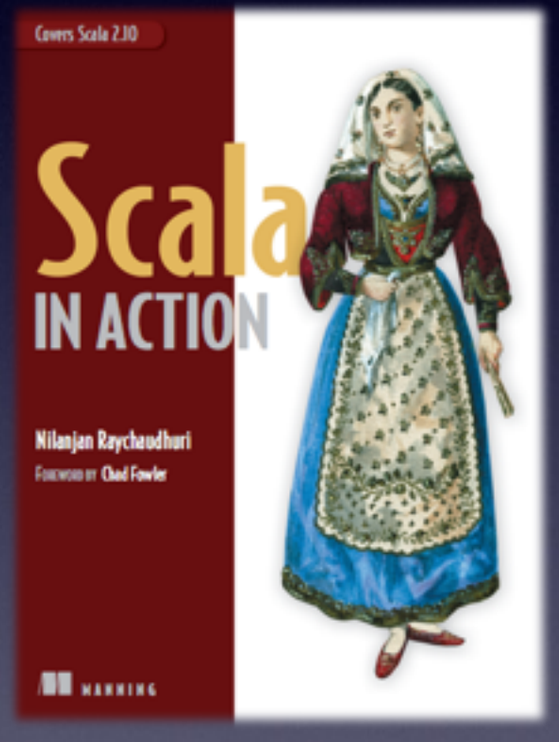
Scala-week1

陈 超
@CrazyJvm

推荐书籍







Object-Oriented Meets Functional

- 基于JVM的FP+OO
- 静态类型
- 和Java互操作

安装Scala

- 选择版本下载http://www.scala-lang.org/download/

注意点：

①提前安装好JDK，并设置好JAVA_HOME，将bin目录加进PATH环境变量中

②scala/bin目录必须包含在PATH环境变量中，建议设置SCALA_HOME

测试Scala解释器，直接在命令行中敲入scala后回车

scalac xxx.scala

scala xxx

IDE

- Eclipse http://scala-ide.org/
- intellij idea http://www.jetbrains.com/idea/download/

值与变量

- 值(val) : 赋值后不可变
  val 值名称 : 类型 = xxx
- 变量(var) : 赋值后可以改变
  var 变量名称 : 类型 = xxx

一般不需要显式指定类型，因为可以从赋值中推断出类型

常用类型

- Byte
- Char
- Short
- Int
- Long
- Float
- Double
- Boolean

并无基本类型与包装类型之分

方法定义

```
def 方法名(参数名 : 参数类型) : 返回类型 = {
    //block内最后一行为返回值
}
```

当返回值为Unit时可以定义为：

```
def 方法名(参数名 : 参数类型) {

}
```

方法注意

- 没有参数的方法可以不带圆括号访问
- Scala没有静态方法，通过object来实现

- 条件表达式(if)
- 循环表达式(for,while,to,until,Range没有continue与break)
- 语句终止(分行写时可以不用分号)

Lazy value

lazy val val_name = val_value

用到时才会去初始化

- 默认参数
- 带名参数(赋值时顺序可以定义时顺序不一致)
- 变长参数(：_*)

异常处理

```
try {
  block(redis)
} catch{
  case e : Exception => System.err.println(e)  //should use log in production
  case _ => //should never happen
}finally {
  this.close(pool, redis)
}
```

定长数组

- val array_name = new Array[T](length)
- val array_name = Array(“”,””)
- 通过( )访问，而不是[ ]

变长数组

- import scala.collection.mutable.ArrayBuffer
- val buff = ArrayBuffer[T]()
- +=/++=/insert/remove/toArray/sum/max/reverse

遍历数组

- for(i <- 0 until array_name.length)
- for(i <- array_name)
- 事实上会更多的使用map、filter等等来操作

Map(可变与不可变)

不可变 val age = Map(“Jack” -> 20,”Lucy” ->”18”)

可变 val age = scala.collection.mutable.Map(…)

初始化 val m = scala.collection.mutable.Map[String,Int]()

Map操作

- 取值 map(key),更好的方式map.getOrElse(key,default)
- 更新 map(key)=value / += / -=
- 迭代 for((k,v) <- map){ },也可单独取key或者value
  for((k,_) <- map) / for((-,v) <- map)
  也可使用map.keySet 或者 map.values

元组

- ( )里包含一系列的值
- 通过._取值，下标从1开始，例如 t._1
- 迭代 for(elem <- t.productIterator)

微博推荐

@邓草原

@hongjiang_wang

@zhongl

@福强王

@程序员老高

@诺铁

谢 谢

欢迎关注我的公众微信号 *ChinaScala*