



专注于商业智能BI和大数据的垂直社区平台

Spark初级课程

讲师：平常心

lazy懒值

当val被声明为lazy时，初始化将被推迟，只有在第一次使用时才会取值，或者变量对应的表达式才会发生计算。

lazy的特性对于开销比较大的初始化语句，特别耗时的计算操作特别有用，比如打开文件进行IO，网络IO等。

- **import** scala.io.Source._
lazy val lines = fromFile("F:\\vm_linux\\bigdata\\vmware.log").mkString

即使文件不存在，也不会报错，只有第一个使用变量时会报错，证明了表达式计算的lazy特性。

```
val lines = fromFile("F:\\vm_linux\\bigdata\\vmware0.log").mkString //直接赋值，文件不存在报错
```

```
val lines = fromFile("F:\\vm_linux\\bigdata\\vmware0.log").mkString
```

```
val split: Array[String] = lines.split("\\n") //第一次调用赋值
```

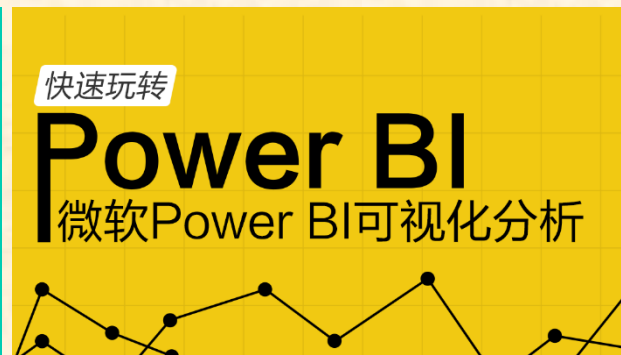
```
for(line <- split){  
  println(line)  
}
```

异常

在Scala中，异常处理和捕获机制与java非常相似。如果有异常发生，catch字句是按次序捕捉的。因此，在catch字句中，越具体的异常越要靠前，越普遍的异常越靠后。如果抛出的异常不在catch字句中，该异常则无法处理，会被升级到调用者处。

```
import scala.io.Source._  
try {  
  val file = fromFile("F:\\vm_linux\\bigdata\\vmware.log")  
} catch {  
  case ex: FileNotFoundException => println("Missing file exception")  
  case ioex: IOException => println("IO Exception")  
}
```


更多商业智能BI和大数据精品视频尽在 www.hellobi.com



BI、商业智能
数据挖掘 大数据
数据分析
R Python
机器学习
Tableau
QLIKVIEW
Hive Hadoop
BIWORK
BAO胖子 seng
曹浩 贝克汉姆