

# Scala-week3

陈超  
@CrazyJvm



# 值函数

- `val valFunName = funName _`

# 嵌套函数

- def套def

# 匿名函数

- (参数名：参数类型) => 表达式
- 可以将匿名函数赋给val或者var



# 函数作为参数

- `fun(函数)`
- `def fun( f : (Int,Int) => Int ) : Int`

```
def fun( f : (Int,Int) => Int ) : Int = f(1,2)
```

```
fun((x : Int , y : Int) => x + y)
```

# 闭包

- 闭包 = 代码 + 用到的非局部变量

# 参数简化与类型推导

- `list.map((x : Int) => x + 1)`
- `list.map((x) => x + 1)`
- `list.map(x => x + 1)`
- `list.map(_ + 1)`

# partial function

- `def add(x : Int, y : Int, z : Int)`
- `def _add = add(_ : Int ,_ : Int , 0)`



# Currying

- `def add(x : Int)(y : Int) = x + y`
- `add(1)_`

# 重要高阶函数

- map
- filter
- reduce/reduceLeft/reduceRight
- fold/foldLeft/foldRight

# By-name parameters

- 定义与例子
- lazy evaluation

# 集合

- Seq
- Set
- Map



- Scala的集合操作与Spark的操作是平行的！！！！

# 可变与不可变集合

- `scala.collection.mutable.Map`
- `scala.collection.immutable.Map`
- `scala.collection.mutable.List`
- `scala.collection.immutables.List`

# Range

- `Range(start : Int, end : Int, step : Int)`
- 提供了`apply`方法
- `step`可为正或者负，但不能为0

# List

- List 由Nil或者head + tail构成,其中tail又是一个List
- new-value :: list
- 构建: List(1,2,3) 或者 1 :: 2 :: 3 :: Nil
- 基于List的模式匹配
- 利用迭代或者递归来处理List



# ListBuffer(可变)

- `+=/+++=/-=-/- -=`

# immutable Set

- `+/++`
- `-/-`
- `scala.collection.immutable.SortedSet`

# mutable Set

- `++` `+=` `--` `-=` 都会创建一个新的Set
- `++` `+=` `--` `-=` 不会
- `scala.collection.mutable.SortedSet`(trait)
- `TreeSet` & `BitSet`

# 重要方法

- map
- foreach / take
- filter
- flatten/flatMap
- reduce/fold reduceLeft/foldLeft or Right
- sum/max/min/count
- zip



# 谢 谢

欢迎关注我的公众微信号 *ChinaScala*

