Scala:Zip函数族详解

在Scala中存在好几个Zip相关的函数，比如zip，zipAll，zipped 以及zipWithIndex等等。我们在代码中也经常看到这样的函数，这篇文章主要介绍一下这些函数的区别以及使用。

1、zip函数将传进来的两个参数中相应位置上的元素组成一个pair数组。如果其中一个参数元素比较长，那么多余的参数会被删掉。看下英文介绍吧：

Returns a list formed from this list and another iterable collection by combining corresponding elements in pairs. If one of the two collections is longer than the other, its remaining elements are ignored.

```
scala> val numbers = Seq(0, 1, 2, 3, 4)
numbers: Seq[Int] = List(0, 1, 2, 3, 4)

scala> val series = Seq(0, 1, 1, 2, 3)
series: Seq[Int] = List(0, 1, 1, 2, 3)

scala> numbers zip series
res24: Seq[(Int, Int)] = List((0,0), (1,1), (2,1), (3,2), (4,3))
```

2、zipAll
函数和上面的zip函数类似，但是如果其中一个元素个数比较少，那么将用默认的元素填充。

The zipAll method generates an iterable of pairs of corresponding elements from xs and ys, where the shorter sequence is extended to match the longer one by appending elements x or y

```
/**
 * User: 过往记忆
 * Date: 14-12-17
 * Time: 上午10:16
 * bolg:
 * 本文地址：/archives/1225.html
 * 过往记忆博客，专注于hadoop、hive、spark、shark、flume的技术博客，大量的干货
 * 过往记忆博客微信公共帐号：iteblog_hadoop
 */
scala> val xs = List(1, 2, 3)
```

```
xs: List[Int] = List(1, 2, 3)

scala> val ys = List('a', 'b')
ys: List[Char] = List(a, b)

scala> val zs = List("I", "II", "III", "IV")
zs: List1 = List(I, II, III, IV)

scala> val x = 0
x: Int = 0

scala> val y = '_'
y: Char = _

scala> val z = "_"
z: java.lang.String = _

scala> xs.zipAll(ys, x, y)
res30: List[(Int, Char)] = List((1,a), (2,b), (3,_))

scala> xs.zipAll(zs, x, z)
res31: List[(Int, java.lang.String)] = List((1,I), (2,II), (3,III), (0,IV))
```

3、zipped函数，这个不好翻译，自己看英文解释吧

The zipped method on tuples generalizes several common operations to work on multiple lists.

```
scala> val values = List.range(1, 5)
values: List[Int] = List(1, 2, 3, 4)

scala> (values, values).zipped toMap
res34: scala.collection.immutable.Map[Int,Int] = Map(1 -> 1, 2 -> 2, 3 -> 3, 4 -> 4)

scala> val sumOfSquares = (values, values).zipped map (_ * _) sum
sumOfSquares: Int = 30
```

4、zipWithIndex函数将元素和其所在的下表组成一个pair。

The zipWithIndex method pairs every element of a list with the position where it appears in the list.

```
scala> val series = Seq(0, 1, 1, 2, 3, 5, 8, 13)
series: Seq[Int] = List(0, 1, 1, 2, 3, 5, 8, 13)

scala> series.zipWithIndex
res35: Seq[(Int, Int)] = List((0,0), (1,1), (1,2), (2,3), (3,4), (5,5), (8,6), (13,7))
```

5、unzip函数可以将一个元组的列表转变成一个列表的元组

The unzip method changes back a list of tuples to a tuple of lists.

```
scala> val seriesIn = Seq(0, 1, 1, 2, 3, 5, 8, 13)
seriesIn: Seq[Int] = List(0, 1, 1, 2, 3, 5, 8, 13)

scala> val fibonacci = seriesIn.zipWithIndex
fibonacci: Seq[(Int, Int)] = List((0,0), (1,1), (1,2), (2,3), (3,4), (5,5), (8,6), (13,7))

scala> fibonacci.unzip
res46: (Seq[Int], Seq[Int]) = (List(0, 1, 1, 2, 3, 5, 8, 13),List(0, 1, 2, 3, 4, 5, 6, 7))

scala> val seriesOut = fibonacci.unzip _1
seriesOut: Seq[Int] = List(0, 1, 1, 2, 3, 5, 8, 13)

scala> val numbersOut = fibonacci.unzip _2
numbersOut: Seq[Int] = List(0, 1, 2, 3, 4, 5, 6, 7)
```