# SparkRDD

# 一、RDD概述

# 二、RDD转换和操作

## 2.1 RDD 创建

### 从集合创建RDD

sc.parallelize(1 to 9, 3) ===> ParallelCollectionRDD

### 从外部存储创建RDD

## sc.parallelize("file:///..../file.txt", 3) ===> ParallelCollectionRDD

## map

map是对RDD中的每个元素都执行一个指定的函数来产生一个新的RDD。**任何原RDD中的元素在新RDD中都有且只有一个元素与之对应。**

## mapPartitions

mapPartitions是map的一个变种。map的输入函数是应用于RDD中每个元素，而mapPartitions的输入函数是应用于每个分区，也就是把每个分区中的内容作为整体来处理的。

## mapValues

mapValues顾名思义就是输入函数应用于RDD中Kev-Value的Value，原RDD中的Key保持不变，与新的Value一起组成新的RDD中的元素。因此，该函数只适用于元素为KV对的RDD。

## mapWith

mapWith是map的另外一个变种，map只需要一个输入函数，而mapWith有两个输入函数。它的定义如下：

## flatMap

与map类似，区别是原RDD中的元素经map处理后只能生成一个元素，而原RDD中的元素经flatmap处理后可生成多个元素来构建新RDD。

## flatMapWith

flatMapWith与mapWith很类似，都是接收两个函数，一个函数把partitionIndex作为输入，输出是一个新类型A；另外一个函数是以二元组（T,A）作为输入，输出为一个序列，这些序列里面的元素组成了新的RDD。它的定义如下：

## flatMapValues

flatMapValues类似于mapValues，不同的在于flatMapValues应用于元素为KV对的RDD中Value。每个一元素的Value被输入函数映射为一系列的值，然后这些值再与原RDD中的Key组成一系列新的KV对。

## reduce

reduce将RDD中元素两两传递给输入函数，同时产生一个新的值，新产生的值与RDD中下一个元素再被传递给输入函数直到最后只有一个值为止。

## reduceByKey

顾名思义，reduceByKey就是对元素为KV对的RDD中Key相同的元素的Value进行reduce，因此，Key相同的多个元素的值被reduce为一个值，然后与原RDD中的Key组成一个新的KV对。