Spark Shell Important Functions — Usage and Meaning Cheat Sheet

## map

Meaning: Transform each element (1-to-1 transformation)

Usage:

val nums = sc.parallelize(1 to 5)  
val doubled = nums.map(x => x \* 2)

## filter

Meaning: Keep elements matching a condition

Usage:

val even = nums.filter(x => x % 2 == 0)

## flatMap

Meaning: Transform elements and flatten the results

Usage:

val words = lines.flatMap(line => line.split(" "))

## reduce

Meaning: Aggregate all elements to a single result

Usage:

val sum = nums.reduce(\_ + \_)

## reduceByKey

Meaning: Aggregate values by key

Usage:

val counts = pairs.reduceByKey(\_ + \_)

## groupByKey

Meaning: Group values with the same key

Usage:

val grouped = pairs.groupByKey()

## sortBy

Meaning: Sort elements by computed key

Usage:

val sorted = nums.sortBy(x => x)

## distinct

Meaning: Remove duplicate elements

Usage:

val distinctNums = nums.distinct()

## count

Meaning: Count total number of elements

Usage:

val total = nums.count()

## countByValue

Meaning: Count occurrences of each unique value

Usage:

val countMap = nums.countByValue()

## take

Meaning: Get first N elements

Usage:

val first3 = nums.take(3)

## collect

Meaning: Bring all RDD data to driver as array

Usage:

val arr = nums.collect()

## foreach

Meaning: Apply a function on each element

Usage:

nums.foreach(x => println(x))

## union

Meaning: Merge two RDDs

Usage:

val merged = rdd1.union(rdd2)

## sample

Meaning: Take a random sample from RDD

Usage:

val sample = nums.sample(false, 0.1)

## intersection

Meaning: Get common elements (no duplicates)

Usage:

val common = rdd1.intersection(rdd2)