

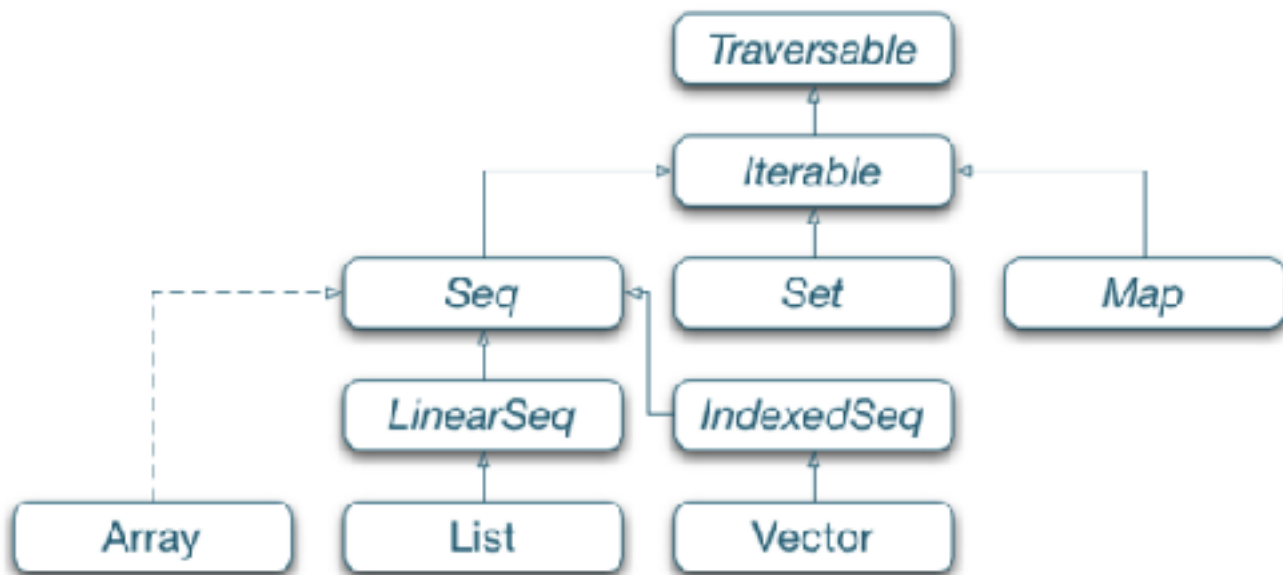
Collections Overview



Lesson Objectives

- After completing this lesson, you should be able to:
 - Understand the structure of the Scala collections hierarchy
 - Describe how to apply functions to data in collections
 - Outline the basics of structural sharing
 - Illustrate the performance characteristics of different data structures

Scala Collections



Higher Order Functions

- Scala is a functional programming language
- We apply functions to data inside of containers
- There are many higher order functions available to you across the collections library

Higher Order Functions

```
scala> 1 to 10
res0: scala.collection.immutable.Range.Inclusive =
  Range(1, 2, 3, 4, 5, 6, 7, 8, 9, 10)

scala> res0.map(n => n + 1)
res1: scala.collection.immutable.IndexedSeq[Int] =
  Vector(2, 3, 4, 5, 6, 7, 8, 9, 10, 11)
```

Structural Sharing

- When you create a collection, it is an aggregation of references to individual values in the Java heap
- If you use immutable collections of immutable values, those references can be shared between collection instances

Structural Sharing

```
scala> List(1, 2, 3, 4, 5)
res0: List[Int] = List(1, 2, 3, 4, 5)

scala> res0 :+ 6
res1: List[Int] = List(1, 2, 3, 4, 5, 6)

scala> 0 +: res1
res2: List[Int] = List(0, 1, 2, 3, 4, 5, 6)
```

Performance

COLLECTIONS

Performance Characteristics

The previous explanations have made it clear that different collection types have different performance characteristics. That's often the primary reason for picking one collection type over another. You can see the performance characteristics of some common operations on collections summarized in the following two tables.

Performance characteristics of sequence types:

| | head | tail | apply | update | prepend | append | insert |
|------------------|------|------|-------|--------|---------|--------|--------|
| immutable | | | | | | | |
| List | O | O | L | L | O | L | - |
| Stream | O | O | L | L | O | L | - |
| Vector | eO | eO | eO | eO | eO | eO | - |
| Stack | O | O | L | L | e | O | L |
| Queue | eO | eO | L | L | L | O | - |
| Range | O | O | O | - | - | - | - |
| Traversable | O | L | O | L | L | L | - |

<http://docs.scala-lang.org/overviews/collections/performance-characteristics.html>



Lesson Summary

- Having completing this lesson, you should be able to:
 - Understand the structure of the Scala collections hierarchy
 - Describe how to apply functions to data in collections
 - Outline the basics of structural sharing
 - Illustrate the performance characteristics of different data structures