

SCALA COLLECTIONS

http://www.tutorialspoint.com/scala/scala_collections.htm

Copyright © tutorialspoint.com

Scala has a rich set of collection library. Collections are containers of things. Those containers can be sequenced, linear sets of items like List, Tuple, Option, Map, etc. The collections may have an arbitrary number of elements or be bounded to zero or one element *e. g. , Option*.

Collections may be **strict** or **lazy**. Lazy collections have elements that may not consume memory until they are accessed, like **Ranges**. Additionally, collections may be **mutable** *thecontentsofthereferencemaychange* or **immutable** *thethingthatareferencereferstoisneverchanged*. Note that immutable collections may contain mutable items.

For some problems, mutable collections work better, and for others, immutable collections work better. When in doubt, it is better to start with an immutable collection and change it later if you need mutable ones.

This chapter gives details of the most commonly used collection types and most frequently used operations over those collections.

SN Collections with Description

1 [Scala Lists](#)

Scala's List[T] is a linked list of type T.

2 [Scala Sets](#)

A set is a collection of pairwise different elements of the same type.

3 [Scala Maps](#)

A Map is a collection of key/value pairs. Any value can be retrieved based on its key.

4 [Scala Tuples](#)

Unlike an array or list, a tuple can hold objects with different types.

5 [Scala Options](#)

Option[T] provides a container for zero or one element of a given type.

6 [Scala Iterators](#)

An iterator is not a collection, but rather a way to access the elements of a collection one by one.

Example:

Following code snippet is a simple example to define all the above type of collections:

```
// Define List of integers.
val x = List(1,2,3,4)

// Define a set.
var x = Set(1,3,5,7)

// Define a map.
```

```
val x = Map("one" -> 1, "two" -> 2, "three" -> 3)

// Create a tuple of two elements.
val x = (10, "Scala")

// Define an option
val x:Option[Int] = Some(5)
```

Loading [Mathjax]/jax/output/HTML-CSS/fonts/TeX/fontdata.js