

Sets

Sets

- Sets are another basic abstraction in the Scala collections
- A set is written analogously to a sequence:

```
val fruit = Set("apple", "banana", "pear")  
val s = (1 to 6).toSet
```
- Most operations on sequences are also available on sets:

```
s.map(_ + 2) // Set(3, 4, 5, 6, 7, 8) – not particularly in this order  
fruit.filter(_.startsWith("app")) // Set("apple")  
s.nonEmpty // true
```

Sets vs Sequences

The principal differences between sets and sequences are:

1. Sets are unordered; the elements of a set do not have a predefined order in which they appear in the set
2. Sets do not have duplicate elements:

```
s.map(_ / 2) // Set(0, 1, 2, 3)
```
3. The fundamental operation on sets is **contains**:

```
s.contains(5) // true
```