Tuples and Maps

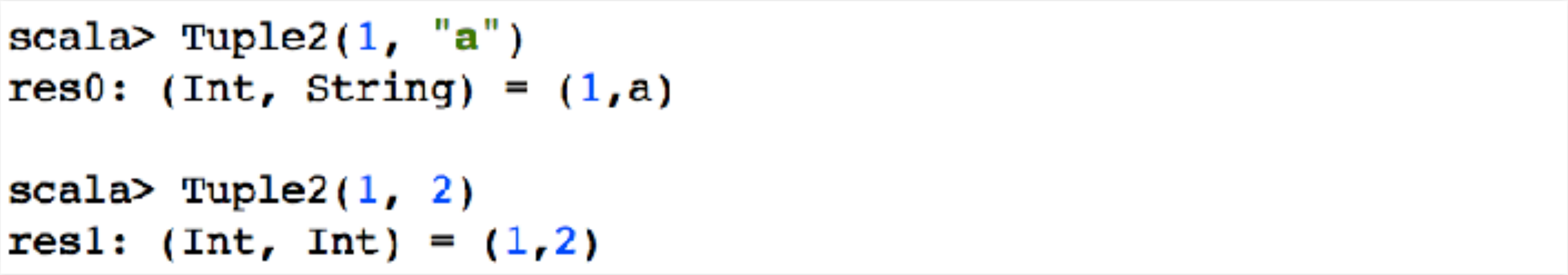
Lesson Objectives

- After completing this lesson, you should be able to:
  - Describe what a tuple is and how they are used
  - Outline how to deconstruct tuples
  - Describe the properties of a map

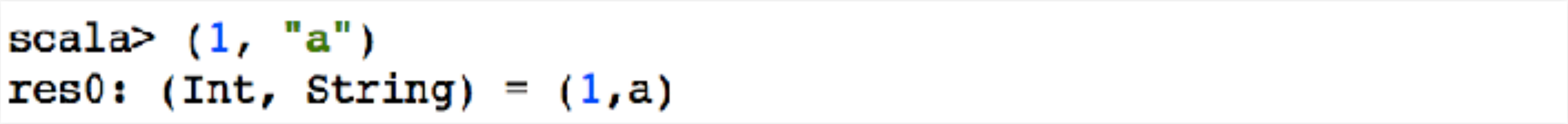
Tuples

- A loose aggregation of values into a single container
- Can have up to 22 values in Scala
- Are always used when you see parentheses wrapping data without a specific type

Tuples



```
scala> Tuple2(1, "a")
res0: (Int, String) = (1,a)

scala> Tuple2(1, 2)
res1: (Int, Int) = (1,2)
```

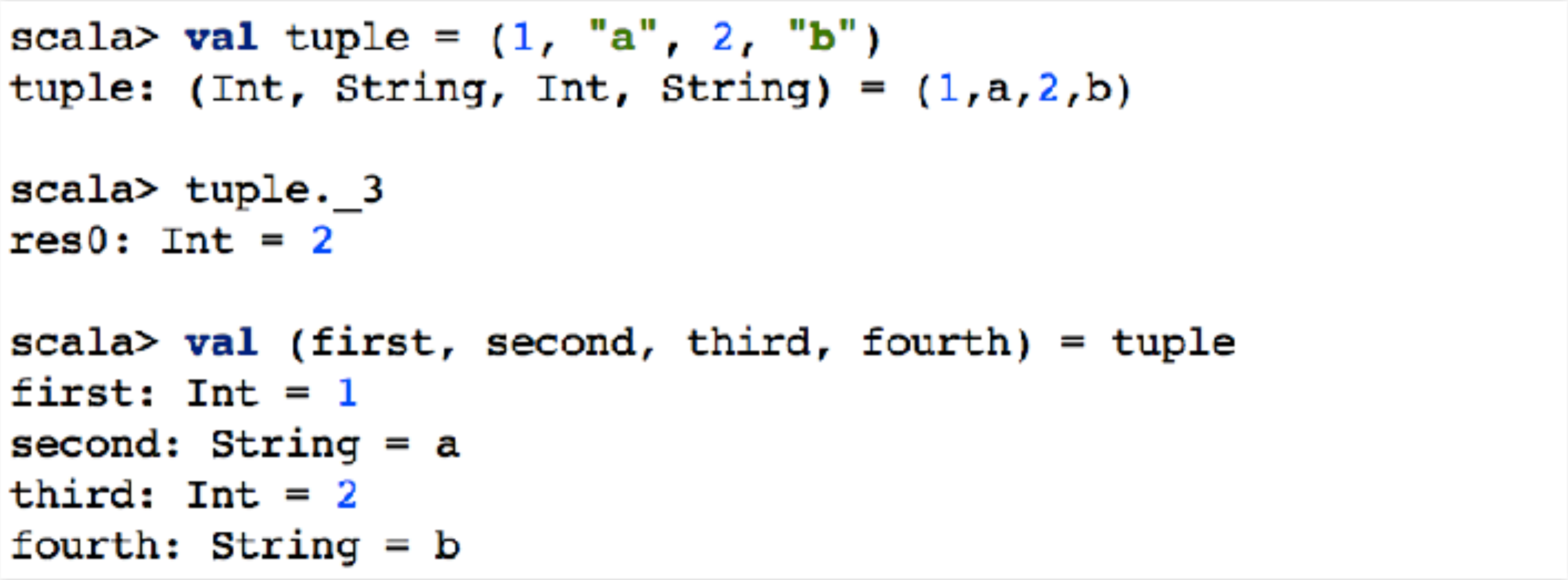


```
scala> (1, "a")
res0: (Int, String) = (1,a)
```

Tuples

- Can be accessed using a 1-based accessor for each value
- Can be deconstructed into names bound to each value in a tuple

Tuples

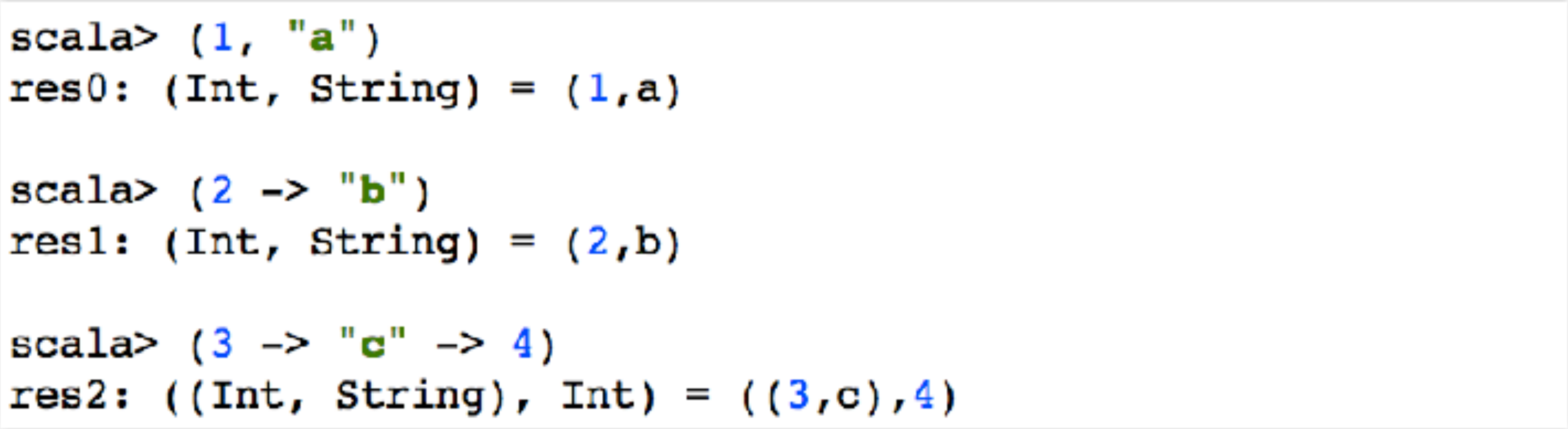


```
scala> val tuple = (1, "a", 2, "b")
tuple: (Int, String, Int, String) = (1,a,2,b)

scala> tuple._3
res0: Int = 2

scala> val (first, second, third, fourth) = tuple
first: Int = 1
second: String = a
third: Int = 2
fourth: String = b
```

Tuple2

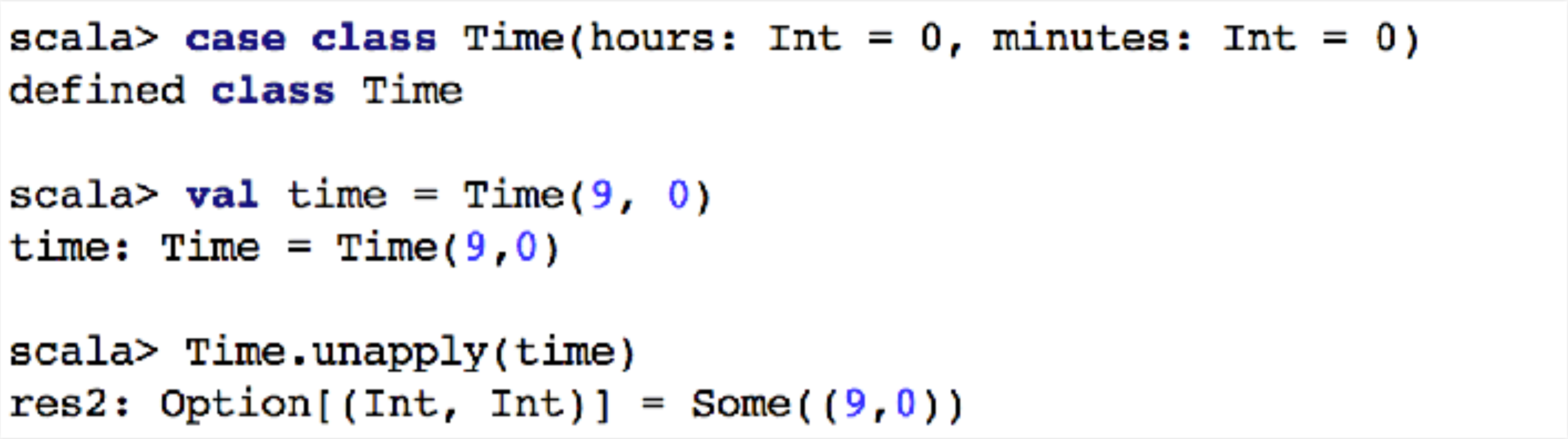
- Frequently called a pair
- Have a unique syntax for values

Tuple2



```
scala> (1, "a")
res0: (Int, String) = (1,a)

scala> (2 -> "b")
res1: (Int, String) = (2,b)

scala> (3 -> "c" -> 4)
res2: ((Int, String), Int) = ((3,c),4)
```

Unapply Deconstructs a Case Class



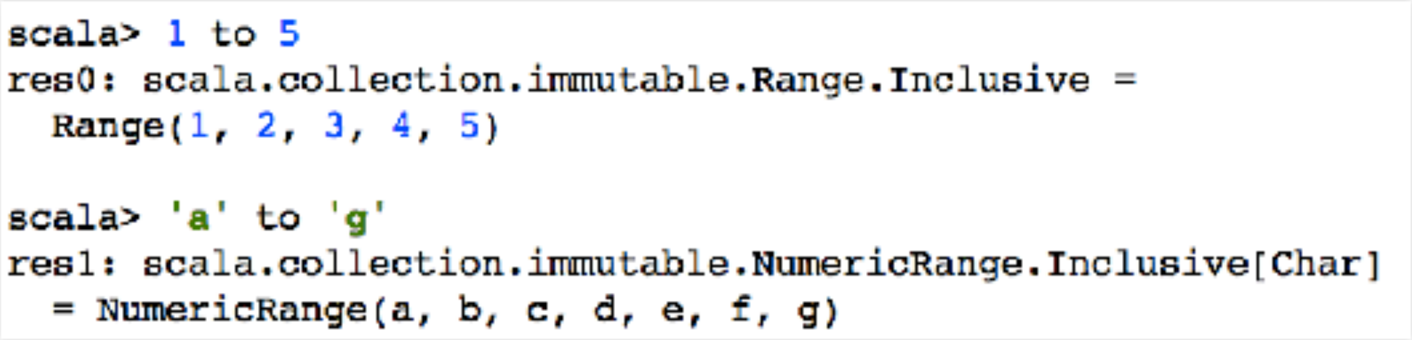
```
scala> case class Time(hours: Int = 0, minutes: Int = 0)
defined class Time

scala> val time = Time(9, 0)
time: Time = Time(9,0)

scala> Time.unapply(time)
res2: Option[(Int, Int)] = Some((9,0))
```

Maps

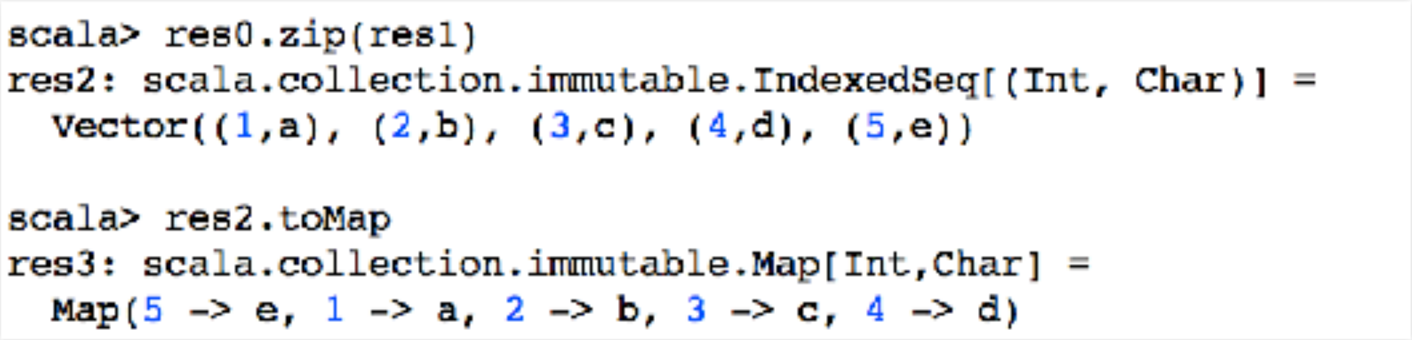
- A grouping of data by key to value, which are tuple “entries”
- Allows you to index values by a specific key for fast access
- Common implementations: HashMap, TreeMap

Typesafe

Maps



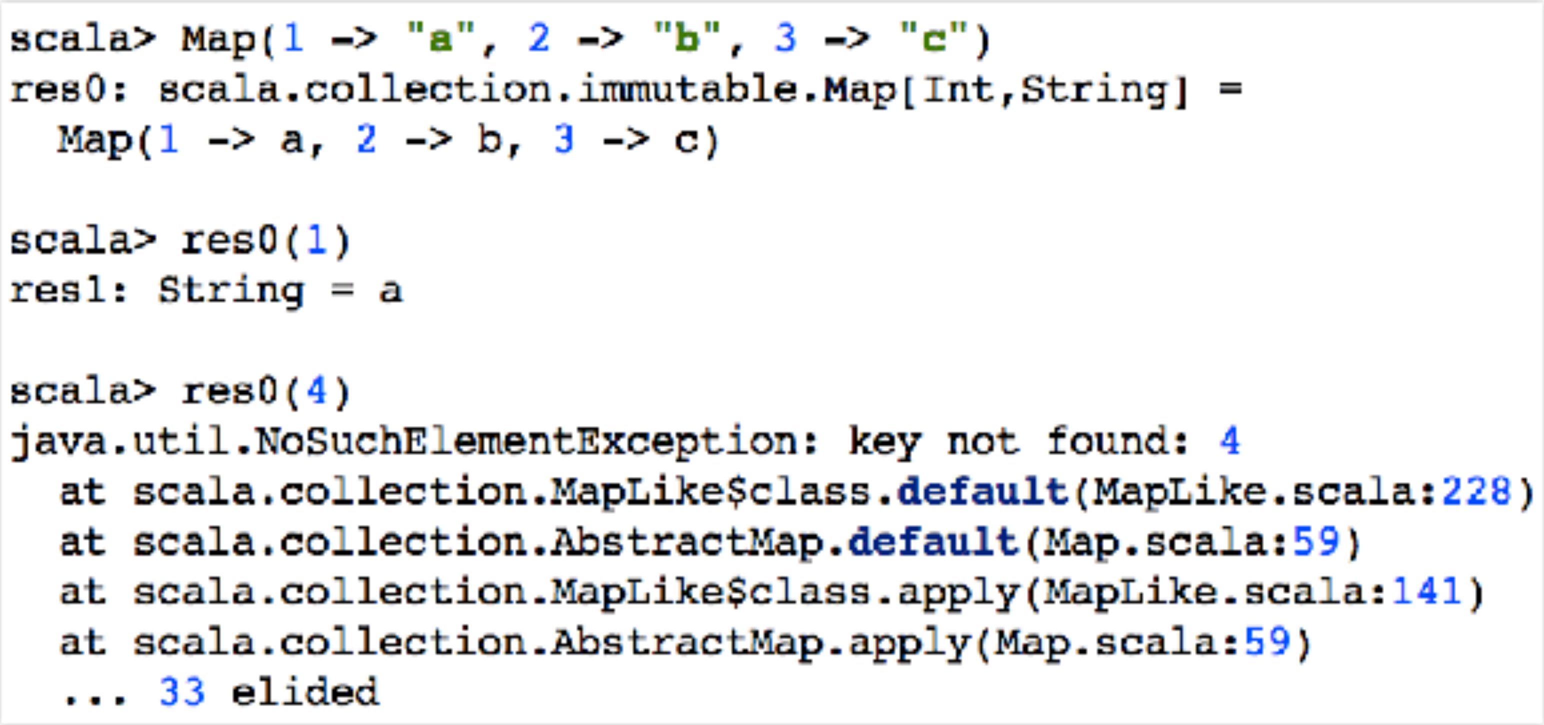
```
scala> 1 to 5
res0: scala.collection.immutable.Range.Inclusive =
  Range(1, 2, 3, 4, 5)

scala> 'a' to 'g'
res1: scala.collection.immutable.NumericRange.Inclusive[Char]
  = NumericRange(a, b, c, d, e, f, g)
```



```
scala> res0.zip(res1)
res2: scala.collection.immutable.IndexedSeq[(Int, Char)] =
  Vector((1,a), (2,b), (3,c), (4,d), (5,e))

scala> res2.toMap
res3: scala.collection.immutable.Map[Int,Char] =
  Map(5 -> e, 1 -> a, 2 -> b, 3 -> c, 4 -> d)
```

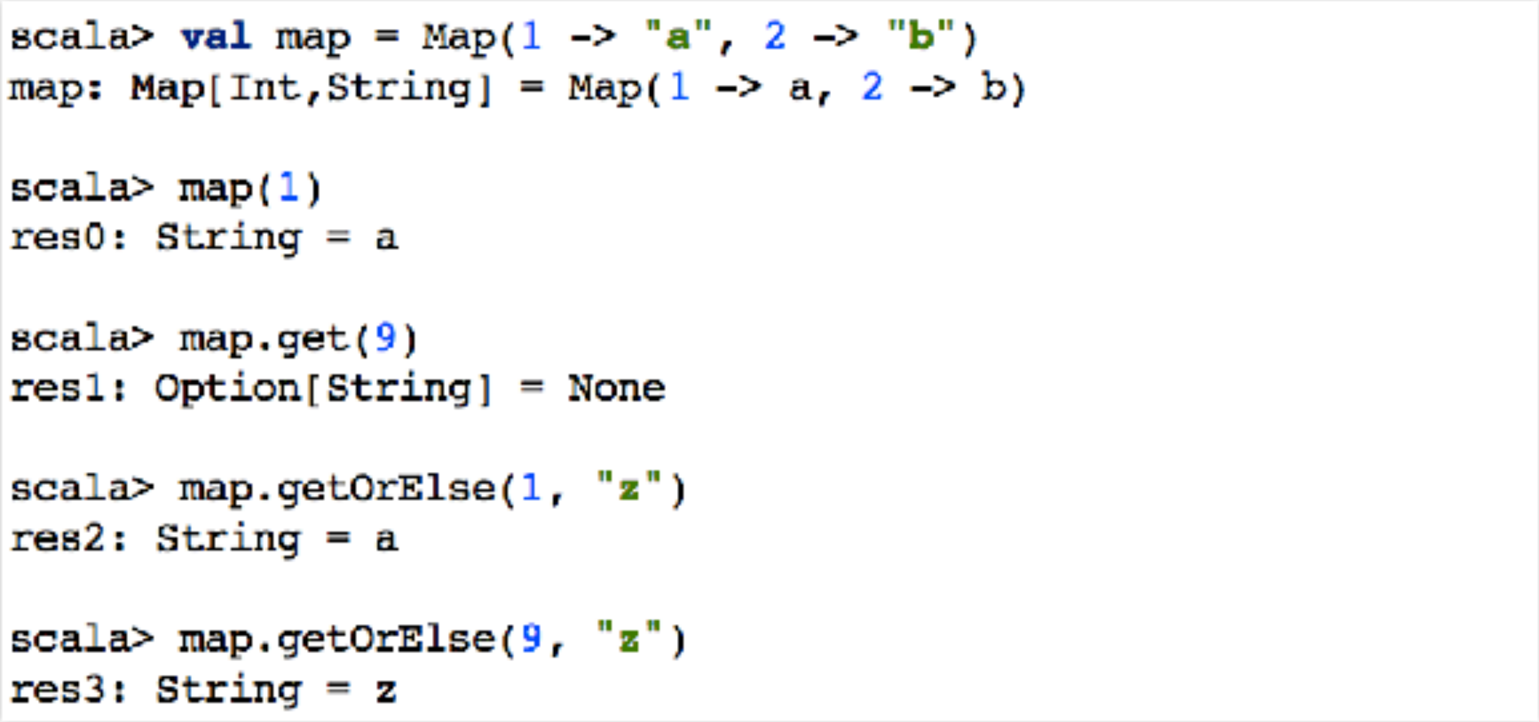
Typesafe

Maps



```
scala> Map(1 -> "a", 2 -> "b", 3 -> "c")
res0: scala.collection.immutable.Map[Int,String] =
  Map(1 -> a, 2 -> b, 3 -> c)

scala> res0(1)
res1: String = a

scala> res0(4)
java.util.NoSuchElementException: key not found: 4
  at scala.collection.MapLike$class.default(MapLike.scala:228)
  at scala.collection.AbstractMap.default(Map.scala:59)
  at scala.collection.MapLike$class.apply(MapLike.scala:141)
  at scala.collection.AbstractMap.apply(Map.scala:59)
  ... 33 elided
```

Maps



```
scala> val map = Map(1 -> "a", 2 -> "b")
map: Map[Int,String] = Map(1 -> a, 2 -> b)

scala> map(1)
res0: String = a

scala> map.get(9)
res1: Option[String] = None

scala> map.getOrElse(1, "z")
res2: String = a

scala> map.getOrElse(9, "z")
res3: String = z
```

Typesafe

Lesson Summary

- Having completing this lesson, you should be able to:
  - Describe what a tuple is and how they are used
  - Outline how to deconstruct tuples
  - Describe the properties of a map