alvin alexander

Scala tuple examples and syntax

By Alvin Alexander. Last updated: Jan 18, 2016

Scala FAQ: Can you share some examples of using tuples in Scala?

table of contents [hide]

Getting started with tuples

A Scala *tuple* is a class that can contain a miscellaneous collection of elements. I like to think of them as a little bag or container you can use to hold things and pass them around.

You create a tuple with the following syntax, enclosing its elements in parentheses. Here's a tuple that contains an `Int` and a `String`:

```
val stuff = (42, "fish")
```

This creates a specific instance of a tuple called a `Tuple2`, which we can demonstrate in the REPL:

```
scala> val stuff = (42, "fish")
stuff: (Int, java.lang.String) = (42,fish)

scala> stuff.getClass
res0: java.lang.Class[_ <: (Int, java.lang.String)] = class scala.Tuple2
```

A tuple isn't actually a collection; it's a series of classes named `Tuple2`, `Tuple3`, etc., through `Tuple22`. You don't have to worry about that detail, other than knowing that you can have anywhere from two to twenty-two items in a tuple. (And in my opinion, if you have twenty-two miscellaneous items in a bag, you should probably re-think your design.)

Accessing tuple elements

You can access tuple elements using an underscore syntax. The first element is accessed with `_1`, the second element with `_2`, and so on, like this:

```
scala> val things = ("a", 1, 3.5)
things: (java.lang.String, Int, Double) = (a,1,3.5)

scala> println(things._1)
a

scala> println(things._2)
1

scala> println(things._3)
3.5
```

Use variable names to access tuple elements

When referring to a Scala tuple you can also assign names to the elements in the tuple. I like to do this when returning miscellaneous elements from a method. To demonstrate the syntax, let's create a very

more scala

general

recursion examples
using match like switch
current date/time
if/then/else
ternary operator
for loop and yield

curly brace packaging
add methods to existing classes
spring framework dependency injection

classes and methods
creating javabeans
importing java code
multiple constructors
named and default parameters
calling methods
class casting
equivalent of java .class
rename classes on import
private primary constructor

try/catch/finally
try, catch, and finally syntax
declare a null var before try/catch

collections
mutable arrays
string arrays
convert array to string
data types
convert java collections to scala
multidimensional arrays (2D array)
iterating over lists (foreach, for)
iterating over maps
convert array to string with mkString

list
list, foreach, and for
merging lists
creating lists
list examples
add elements to a list
the filter method

map
map class examples
iterating over maps

tuple
tuple examples
map tuples in anonymous function

strings
multiline strings
string arrays
string formatting
convert array to string
split string example

simple method that returns a tuple:

```
def getUserInfo = ("Al", 42, 200.0)
```

Now we can call that method, and assign the tuple results directly to variables, like this:

```
val(name, age, weight) = getUserInfo
```

Here's what this looks like in the REPL:

```
scala> def getUserInfo = ("Al", 42, 200.0)
getUserInfo: (java.lang.String, Int, Double)

scala> val(name, age, weight) = getUserInfo
name: java.lang.String = Al
age: Int = 42
weight: Double = 200.0
```

It's shown in the REPL results, but we'll further confirm that we can indeed access the values by variable name:

```
scala> name
res4: java.lang.String = Al

scala> age
res5: Int = 42

scala> weight
res6: Double = 200.0
```

That's pretty nice.

In a cool, related feature, if you only want to access some of the elements, you can ignore the others by using an underscore placeholder for the elements you want to ignore. Imagine you want to ignore the `weight` in our example:

```
scala> val(name, age, _) = getUserInfo
name: java.lang.String = Al
age: Int = 42
```

Or suppose you want to ignore the `age` and `weight`:

```
scala> val(name, _, _) = getUserInfo
name: java.lang.String = Al
```

Again, that's good stuff.

Iterating over a Scala tuple

As mentioned, a tuple is not a collection; it doesn't descend from any of the collection traits or classes. However, you can treat it a little bit like a collection by using its `productIterator` method.

Here's how you can iterate over the elements in a tuple:

```
scala> val t = ("Al", 42, 200.0)
t: (java.lang.String, Int, Double) = (Al,42,200.0)

scala> t.productIterator.foreach(println)
Al
42
200.0
```

The tuple toString method

The tuple `toString` method gives you a nice representation of a tuple:

```
scala> t.toString
res9: java.lang.String = (Al,42,200.0)

scala> println(t.toString)
(Al,42,200.0)
```

Creating a tuple with ->

In another cool feature, you can create a tuple using this syntax:

```
1 -> "a"
```

This creates a `Tuple2`, which we can demonstrate in the REPL:

```
scala> 1 -> "a"
res1: (Int, java.lang.String) = (1,a)

scala> res11.getClass
res2: java.lang.Class[_ <: (Int, java.lang.String)] = class scala.Tuple2
```

You'll see this syntax a lot when creating maps:

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

- creating small forms
- (2D) extract rows and columns
- convert objects to json
- run play on different port
- populate data on startup
- using map in template
- template comments
- template functions
- 404 and 500 errors
- mapping field validators
- web service request with timeout
- play console commands/help
- testing web services with curl
- logout, destroy session
- read cookies

web services

- rest client using apache httpclient
- twitter client example
- json parsing using lift-json
- json array parsing using lift-json
- lift framework form examples

xml

- create xml literal
- generate dynamic xml
- xml – pretty printing
- xml – save to file
- xml – serialize, deserialize
- xml – load a file
- xml – load a url
- xml – xpath searching
- searching xml namespaces, xpath
- xml – extract data from nodes
- xml – extract data from arrays
- xml – parsing, tags
- xml – using match expressions
- xml – many examples

build, testing, and debugging

- sbt documentation (pdf)
- show sbt history
- scalatest - installing
- scalatest - writing tdd tests
- scalatest - writing bdd tests
- scalatest - given/when/then with bdd
- scalatest - test suite
- scalatest - expected, actual
- scalatest - mark test as pending
- scalatest - testing exceptions
- scalatest - tagging tests

Scala Cookbook
Alvin Alexander
Best Price $10.87
or Buy New $36.81
Buy from amazon.com

Privacy Information

Scala tuples – Summary

If you needed information on how to use a Scala *tuple*, I hope these examples have been helpful. Here are a few links to the tuple classes mentioned:

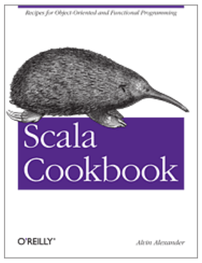
- Tuple2
- Tuple3
- Tuple22

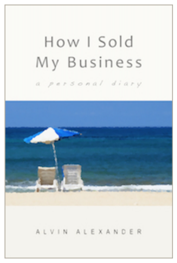
tags: collection examples iterate iterator scala scala syntax tuple tuple2 tuple22 tuple3

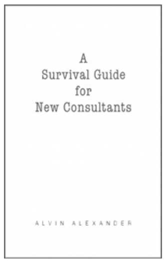
related

- Scala Tuples, for when you just need a bag of things
- Scala - Using tuples in an anonymous function
- A Java tuple class
- How to define Scala methods that return multiple items (tuples)
- Scala - How to rename a class when you import it (a 'rename on import' syntax example)
- How to create a mutable Set in Scala

books i've written







new

- How to use multiple generators in Scala 'for' expressions (loops)
- When you want to store static text in an Android file
- A Java FIFO queue class
- A Java tuple class
- A Java method to calculate the NFL Passer Rating

scalatest - disabling tests
scalatest - mock objects
scalatest - running in eclipse

categories

- android (45)
- best practices (63)
- career (50)
- cvs (27)
- design (33)
- drupal (91)
- eclipse (6)
- funny (3)
- gadgets (108)
- git (15)
- intellij (4)
- java (429)
- jdbc (26)
- swing (74)
- jsp (9)
- latex (26)
- linux/unix (289)
- mac os x (315)
- mysql (54)
- ooa/ood (11)
- perl (156)
- php (97)
- postgresql (17)
- ruby (56)
- scala (336)
- sencha (23)
- servlets (10)
- svn (6)
- technology (84)
- testing (13)
- uml (24)

- An Android method to center text when using Canvas drawText
- An Android cheat sheet (my notes, main concepts)
- How to show an HTML WebView in an Android AlertDialog
- A Java method to determine if an integer is between a range
- How do I round a float or double to an integer in Java?

more

alvin's blog

Post new comment

Your name:
Anonymous

E-mail:

The content of this field is kept private and will not be shown publicly.

Homepage:

Subject:

Comment: *

[✓] Notify me when new comments are posted

[● All comments] ○ Replies to my comment

Preview

java
java applets
java faqs
misc content
java source code
test projects
lejos

perl
perl faqs
programs
perl recipes
perl tutorials

unix
man (help) pages
unix by example
tutorials

source code warehouse
java examples
drupal examples

misc
privacy policy
terms & conditions
subscribe
unsubscribe
wincvs tutorial
function point analysis (fpa)
fpa tutorial

other
mobile website
rss feed
my photos
life in alaska
how i sold my business
living in talkeetna, alaska
my bookmarks
inspirational quotes
source code snippets