Packages and imports

See Scala for the Impatient, Chapter 7 for more details

- Scala's packages are like Java packages
- But there's also a lot more (that we won't cover)
- import org.scalatest.Matchers imports the Matchers member of the org.scalatest package.
- import org.scalatest._ imports the all members.
- import statements can go anywhere

Implicit conversions

See Scala for the Impatient, Chapter 21.4 for more details

The compiler looks for an implicit conversion when:

- the expected type differs from the inferred type
- an object does not contain an expected attribute

The compiler finds an implicit conversion when:

- a conversion is declared as implicit
- a conversion is in scope and is named with a single identifier
- a conversion is defined in the current class's companion object

The compiler does not look for an implicit conversion when:

- the code compiles without one
- the compiler has already performed one (for a given expression)
- it finds multiple conversions (i.e., conversion is ambiguous)

When you want to define implicit conversions, you'll probably want to include the following in the file that contains the implicits:

```scala
import scala.language.implicitConversions
```

Identifiers

See Scala for the Impatient, Chapter 11 for more details

A valid identifier (i.e., name) can include the following characters:

- Standard Unicode characters
- any ASCII character except:
  
  ( ) [ ] { } . , ; ' "

Precedence

See Scala for the Impatient, Chapter 11.5 for more details

Precedence determines how to decide which of two different operations to perform first.

In Scala, the first character of an operator's name determines its precedence, in increasing order as follows:

<table>
<thead>
<tr>
<th>(all letters)</th>
</tr>
</thead>
</table>
| | ^
| &
| < >
| = !
| :
| + -
| * / %

(all other special characters)

So, * has higher precedence than +, etc. Unsurprisingly, there are a few caveats:

- Assignment has lower precedence than anything else.
- Postfix operators have lower precedence than infix ones.

Associativity

See Scala for the Impatient, Chapter 11.6 for more details

Associativity determines how to decide which of two applications of the same operation to perform first.

In Scala, the last character of an operator's name determines its associativity, according to these rules:

- Operators that end with a colon : are right-associative.
- Assignment is right-associative.
- Every other operator is left-associative.