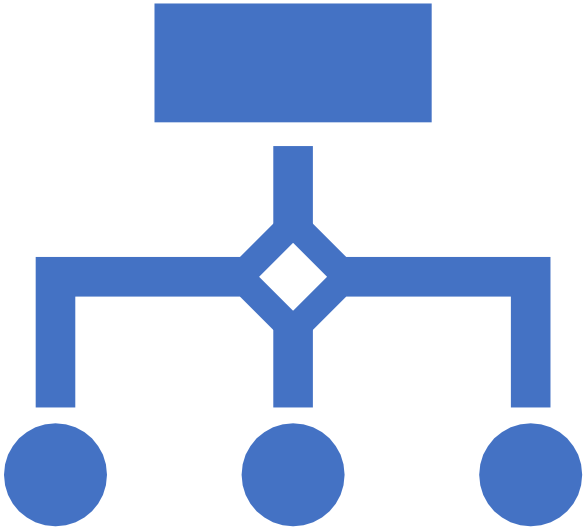
Packages and Imports

BY Pablo Duarte Tzuc

What is it?



Package in Scala is a mechanism to encapsulate a group of classes, sub packages, traits and package objects. It basically provides namespace to put our code in a different files and directories.

Putting code in packages

You can place code into named packages in Scala in two ways

First way:

```
package bobsrockets.navigation
class Navigator // into package we have the class
//"Navigator"
```

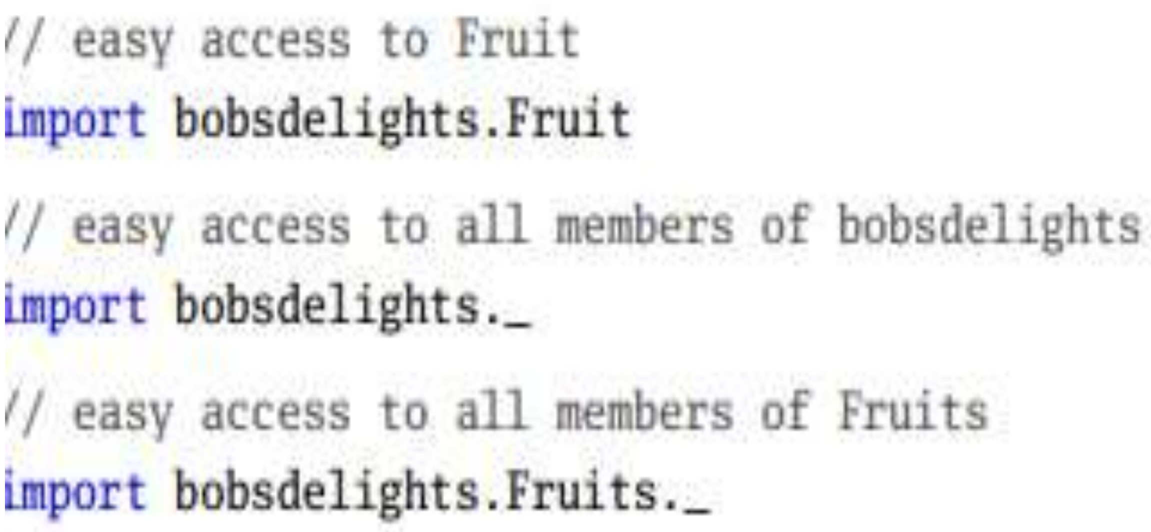
Second way:

```
package bobsrockets.navigation {
class Navigator//we can specify with brackets that inside
our //package we have
}                // the class "Navigator"
```

Imports

In Scala, packages and their members can be imported using import clauses. Imported items can then be accessed by a simple name like File, as opposed to requiring a qualified name like java

Example



```
// easy access to Fruit
import bobsdelights.Fruit
// easy access to all members of bobsdelights
import bobsdelights._
// easy access to all members of Fruits
import bobsdelights.Fruits._
```

```
package bobsdelights //create out package
abstract class Fruit(//abstract class
    val name: String, // variables inmutables
    val color: String
)
object Fruits { //we have our object fruits
    object Apple extends Fruit("apple", "red")
    object Orange extends Fruit("orange", "orange")
    object Pear extends Fruit("pear", "yellowish")
    val menu = List(Apple, Orange, Pear)
}
// we have ready our package for import
```

Scala’s flexible imports

```
def showFruit(fruit: Fruit) = {

import fruit._ //Imports into Scala can appear
anywhere, not //just at the beginning of a
//compilation unit.

println(name + "s are " + color)
}
```

Another flexibility that scale allows us is that you can hide a member of your package, this is useful to avoid ambiguities.

```
import Fruits.{Pear => _, _}// it mean that the
pear is renowden to  "_" so of this  form is very
useful for avoid //ambiguities
```

```
import Notebooks._
import Fruits.{Apple => _, _}
```

This would import all Notebooks and all Fruits, except for Apple.

Implicit imports

These packages already have predefined classes and common objects definitions of types, methods and implicit conversions that are commonly used in Scala programs.

```
import java.lang._ // everything in the java.lang
//package

import scala._ // everything in the scala package

import Predef._ // everything in the Predef
object
```

Private members

private is visible only inside the class or object that contains the member definition

```
class Outer {
    class Inner {// in the class inner are the
//private member
        private def f() = { println("f") }
        class InnerMost {
            f() // OK
        }
    }
(new Inner).f() // error: f is not accessible
}
```

Protected members

In Scala, a protected member is only accessible from subclasses of the class in which the member is defined

```
package p {
    class Super {
     protected def f() = { println("f") }
}
class Sub extends Super {
    f()
}
class Other {
    (new Super).f() // error: f is not accessible
    }
}
```

Public members

Public members can be accessed from anywhere

```
class Example {
    var a:Int=7
}
object access extends App{
    var e=new Example()
    e.a=8
    println(e.a)
}
```