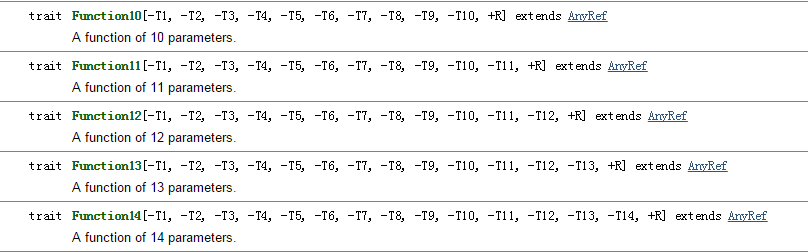
在scala中能够定义函数。定义的函数可以有返回值，也可以没有返回值。没有返回值的叫做过程，有返回值的叫做函数。在语法上的区别是是否有等号。

Scala中既有函数也有方法，大多数情况下我们都可以不去理会他们之间的区别。但是有时候我们必须要了解他们之间的不同。

Scala中的方法跟Java的方法一样，方法是组成类的一部分。方法有名字、类型签名，有时方法上还有注解，以及方法的功能

实现代码（字节码）。

Scala中的函数是一个完整的对象。Scala中用22个特质（trait）抽象出了函数的概念。这22特质从Function1到Function22

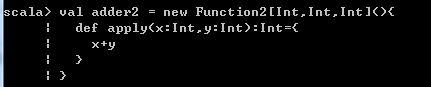


如上图中的Function10代表的是：有10个形参，返回值为R（协变）的函数。

Scala中的函数其实就是继承了这些Trait的类的对象，如：我们通过函数字面量定义一个函数

http://images.cnitblog.com/blog2015/508475/201504/131329257456696.png

其实上述函数的定义方式跟如下定义方式等同：



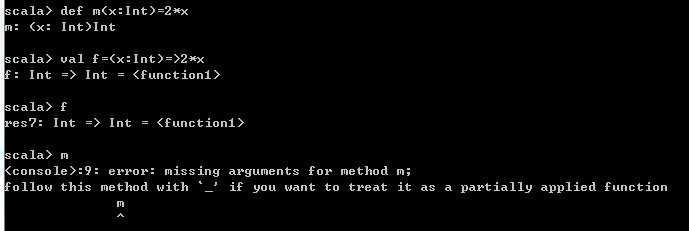
由于Function2是特质，不能直接new。上述new Function2[Int,Int,Int](){}其实是定义并实例化一个实现了Function2特质的类的对象。

apply是scala中的语法糖：对一个对象obj上调用obj()，scala编译器会转换为obj.apply()；在一个类clazz上调用clazz(),scala编译器会转

换为clazz\_company\_obj.apply()，其中clazz\_company\_obj为clazz的伴生对象。

具体的差异，总结为如下几点：

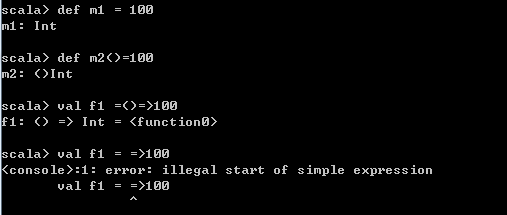
1.方法不能作为单独的表达式而存在（参数为空的方法除外），而函数可以。如：



 在如上的例子中，我们首先定义了一个方法m，接着有定义了一个函数f。接着我们把函数名（函数值）当作最终表达式来用，由于f本身就是

一个对象（实现了FunctionN特质的对象），所以这种使用方式是完全正确的。但是我们把方法名当成最终表达式来使用的话，就会出错。

2.函数必须要有参数列表，而方法可以没有参数列表

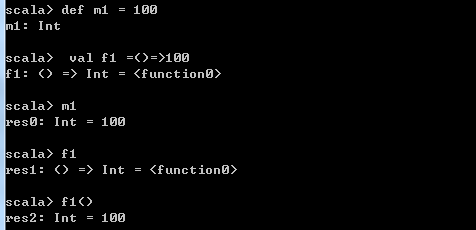


 在如上的例子中，m1方法接受零个参数，所以可以省略参数列表。而函数不能省略参数列表

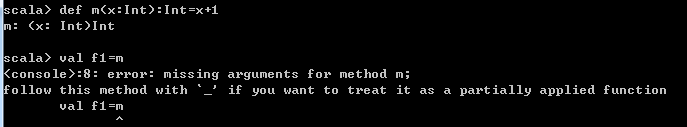
3.方法名是方法条用，而函数名只是代表函数对象本身

 这个比较容易理解。因为保存函数字面量的变量（又称为函数名或者函数值）本身就是实现了FunctionN特质的类的对象，要调用对象的apply

方法，就需要使用obj()的语法。所以函数名后面加括号才是调用函数。如下：



4.在需要函数的地方，如果传递一个方法，会自动进行ETA展开（把方法转换为函数）



如果我们直接把一个方法赋值给变量会报错。如果我们指定变量的类型就是函数，那么就可以通过编译，如下：

http://images.cnitblog.com/blog2015/508475/201504/131416455573783.png

当然我们也可以强制把一个方法转换给函数，这就用到了scala中的部分应用函数：

http://images.cnitblog.com/blog2015/508475/201504/131418082768279.png

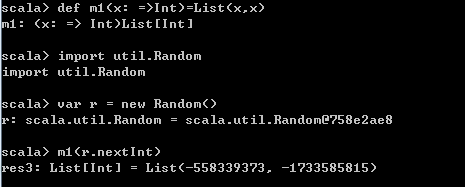
5.传名参数本质上是个方法

 传名参数实质上是一个参数列表为空的方法，如下：

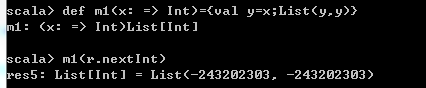
http://images.cnitblog.com/blog2015/508475/201504/131422203399106.png

如上代码实际上定义了一个方法m1，m1的参数是个传名参数（方法）。由于对于参数为空的方法来说，方法名就是方法调用

，所以List(x,x)实际上是进行了两次方法调用。



由于List(x,x)是进行了两次方法调用，所以得到两个不同的值。



如果我们稍微修改一下函数的m1的定义，把x先缓存起来，结果就会跟以前大不一样。