吴雪峰@THOUGHTWORKS 2014.06.08

SCALA 语法糖

CASE CLASS

- equals
- toString
- copy

APPLY

- List(1,2,3) = List.apply(1,2,3)
- List(1)(0) = List.apply(1).apply(0)

UPDATE

- Map(1 -> "a")(1) = "b"
- scala.collection.mutable.Map(1 -> "a")(1) = "b"

_

- Placeholder syntax: List(1, 2, 3) map (_ + 2)

- Wildcard patterns: Some(5) match { case Some(_) => println("Yes") }

- Wildcard imports: import java.util._

```
class Test {
  private var x0: Int = 0
  def x = x0
  def x_=(a: Int) = x0 = a
}
```

```
scala> val t = new Test
t: Test = Test@4166d6d3

scala> t.x = 1
t.x: Int = 1
```

```scala
class Underscores {
 import collection.{ Map =>_ , _ }

 var count : Int = _

 def sum = (_:Int) + (_:Int)
 //could be defined with multiple argument lists
 def sum2(a:Int)(b:Int) = a+b
 def offset = sum2(count) _

 def sizeOf(l:Traversable[_]) : Unit = l match {
   case it:Iterable[Int] => count = (0/:it)(_ + _)
   case s:Seq[_] =>  s.foreach( _ => count = count+1)
   case _ => println(offset(l.size))
 }
}
```

1 2 3 4 5 6