

Scala中的下划线使用总结

1、方法转化为函数

例如：

```
def m1(x:Int,y:Int)=x*y  
val f1=m1 _
```

```
scala> def m1(x:Int,y:Int)=x*y  
m1: (x: Int, y: Int)Int  
  
scala> val f1 =m1  
f1: (Int, Int) => Int = <function2>  
  
scala>
```

2、集合中的每一个元素

例如：

```
val list=List(1,2,3,4)  
val list1=list.map(_ * 10)
```

```
scala> val list=List(1,2,3,4)  
list: List[Int] = List(1, 2, 3, 4)  
  
scala> list.map(_ * 10)  
res1: List[Int] = List(10, 20, 30, 40)  
  
scala>
```

3、获取元组Tuple中的元素

例如：

```
val t=("hadoop",3.14,100)  
t._1  
t._2  
t._3
```

```
scala> val t=("hadoop",3.14,100)  
t: (String, Double, Int) = (hadoop, 3.14, 100)  
  
scala> t._1  
res2: String = hadoop  
  
scala> t._2  
res3: Double = 3.14  
  
scala> t._3  
res4: Int = 100
```

4、模式匹配

例如：

```
val word="hadoop"
val result =word match{
    case "hadoop" => 1
    case "spark"  => 2
    case _        => 0    //以上都没有匹配到才会被执行
}
```

```
scala> val word="hadoop"
word: String = hadoop

scala> val result =word match{
    case "hadoop" => 1
    case "spark"  => 2
    case _        => 0
  }
result: Int = 1
```

5、队列

例如：

```
val list=List(1,2,3,4)
list match{
    case List(_,_*) =>1
    case _ =>2
}
```

```
scala> val list=List(1,2,3,4)
list: List[Int] = List(1, 2, 3, 4)

scala> list match{
    case List(_,_*) =>1
    case _ =>2
  }
res6: Int = 1
```

6、导包引入的时候

例如：

```
import scala.collection.mutable._
表示引入的时候将scala.collection.mutable包下面所有的类都导入
```

```
scala> import scala.collection.mutable._
import scala.collection.mutable._
```

7、初始化变量

例如：

```
var name:String= _
```

//在这里，name也可以声明为null，例：var name:String=null。这里的下划线和null的作用是一样的。

```
var age:Int= _
```

//在这里，age也可以声明为0，例：var age:Int=0。这里的下划线和0的作用是一样的。

```
scala> var name:String= _  
name: String = null
```

```
scala> var age:Int= _  
age: Int = 0
```