# 语法

|  |
| --- |
| scala 中所有的对象的超类是Any。Any又分为AnyVal和AnyRef。  AnyVal 相当于java中基础数据类型。  AnyRef 相当于对象引用。可以引用scala的对象，也可以引用java的对象  var 变量 val 常量  val name : String =”lisi”;  val name = “lisi”;  println（）  println(f”name= $name”); f允许使用插值器。通过$+变量直接在字符串中取值。  Println(s”1 + 1= { 1 + 1}”); s与f类似。允许使用插值器。还可以使用 表达式  Println(s” name = ${stu.name}”) $+表达式 获取对象的属性  大括号的最后一行 为返回值  条件表达式  val s = if ( i > 10) {  i  }else {  0  }  循环  类似java增强for循环  for( item <- array){  println(item)  }  类似java普通循环  for(index <- 0 to array.length -1 ){  println(array(index))  }  to可换成until，表示左闭右开区间  数组取数，通过（）+下标来取 数据的元素  yield 关键字  将 循环中的每次的返回 形成一个新的数组。  如  for（item <- 1 to 10）yeild item+1  返回 2-11的数组  操作符重载成方法  如： + - \* / to until map  var a=1+2 可写成var a=1.+2  1 to 10 可写成 1.to(10)  array map (\_.1)  定义方法  1、def 方法名 （传参）：返回类型=｛方法体｝  2、def sayHello = println(“hello”) 可以在sayHello在定义时没有加（），调用时也不能加（）  方法转函数: 方法名 \_  如：add \_  定义函数  1、（传参名：类型 ...） => 方法体  （a : int, b : int ）=> a+b  如：val add =（a : int, b : int ）=> a+b  2、（传参类型） => 返回类型 =>(传参名) => 方法体  如： val add : (Int, Int) => Int =(x,y) => x+y  可变参数  Def (a :int , b Int \*):Int ={  for(item :<- b){  }  }  参数默认值，调用时，若不传参数则使用默认值  def add (a:Int = 6,b:Int=6) ={}  或者指定传名  add(b=9,a=9);  add(b=9);  高阶函数：  将其它函数作为参数 或 函数的返回值是一个函数  部分参数应用函数：  如  def add(a:Int,b:Int) = a+b  var add1 =(x:Int) =>add(1,x)  此时，add1也是一个函数。调用 add1(4) 就等 调用 aad(1+4)  函数柯里化  将一个函数的多个参数 拆开。如：  def add(a:Int ,b:Int)=a+b 柯里化 def add(a:Int)(b:Int)=a+b ,调用时add(1)(2)  以上柯里化函数可以 写成 def add(a:Int)=(b:Int) =>a+b  调用时，var add1=add(1) 返回一个函数，再调用add1(3) 返回4.  偏函数，是一个类 PartialFunction[String,Int], String，int是可以是其它类型，String是传入的类型，Int是返回的类型  def f1: PartialFunction[] ={ case “a” => 97 case \_ => 0}  使用 f1(“a”) 返回97 |

数组，集合

|  |
| --- |
| 定义  var x:Array[String] =new Array[String](3)  或 var y = new Array[String](3)  Val z= Array[Int](1,2,3)  修改元素内容  Z(0)= 100  添加内容  z += (200) 报错  数组长度不可变，内容可变  数组的map方法 映射：即对每个元素进行操作 ，返回新的数组 flatten将数组里面的多个数组元素转化一个数组  flatmap  foreach 遍历  groupBy 分组，返回一个map ，map[String,Array]如：array.groupBy( x => x)  第一个x表示每个元素。第二个x，表示返回的分组，即返回map的Key |
| mapValues( ) 对每个value进行映射。返回一个map，该map的key仍是原map的key，value是映射后的值 |