# 课堂笔记

## 控制语句

**if**、while、for、foreach的用法

### if语句

if (x > 0) 1 else 0

赋值给变量：

var a = if (x > 0) 1 else 0

if(x>0) {

1

} else if (x==0)

{0}

elise -1

不需要分号，可以加上分号，根据你的习惯。

如果一行中写多个表达式，则用分号。

### 循环语句

**while循环**

var a = 0

var sum = 0

while(a < 10){

sum += a

a += 1 //必须有跳出循环的控制，否则是死循环

}

**for循环**

没有Java中 **for(int i=0 ; i<=9 ; i++)** 的格式，格式如下：

for( i <- 表达式)

{

// to do

}

for(i <- 1 to 10)

{

println(i)

{

var a= “abcde”

for(i <- 0 until a.length if i!=2)

{

println(a(i))

{

Scala 没提供continue和break，怎办？

**import** scala.util.control.Breaks.\_

**object** IForWhile {

**def** main(args: Array[*String*]): Unit = {

**var** i = 0

**var** sum = 0

breakable{

**while**(i <= 10)

{

sum += i

i += 1

**if(i==2) { break }**

}

}

println(sum)

}

}

until 和 to的区别：

until是<, to是<=

**foreach用法举例：**

用法灵活，常用用法如下。

用法1、

var i=0

0 until 5 foreach {

i += \_

...

}

()和{} 都可以，通常多行的时候用{}

**\_ 代表每个元素**

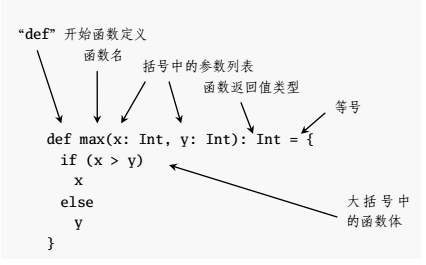
用法2、

(0 until 10).foreach(println(\_))

(0 until 10) foreach(i => println(i))

(0 until str.length()).foreach(s => println(s))

## 函数



def hello(name : String ) : String = {

"hello:" + name

}

def用于定义函数

不需要return，**{} 里最后一行就是返回值**，可以用return，便于读。

Unit 对应Java里void，就是不返回结果

无参数的函数：

def helloScala() ()可以不要

{

}

调用：helloScala 或 helloScala()

匿名函数：

(x:int , y:int)=> x+y

val add = (x:Int , y:Int)=> x+y

调用add(1,2)

def add2(x:Int)(y:Int)=x+y

调用add2(1)(2)

变长参数：

def printEvery(c:String\*)=

{

c.foreach(x => println(x))

}

调用：printEvery("a","b","c") 。 \*代表多个同类型

函数**默认参数**：

def func(a:String , b:String =”bvalue”, c:String=”cValue”) =

{  
 ...

}

调用方式：

func(“sdf”)

func(“sdf”,”wer”)

func(“sdf”,”wer”,”245”)

例如：

**def** func(a:*String* , b:*String* ="bvalue", c:*String*="cValue") =

{

println(a+b+c)

}

func("1")

func("1","sdd")

func("1","sdd","end")

如下三个是**带名参数**：

func("1",c="end")

func("1",b="end")

func("1",c="45",b="end")

## 过程 procedure

有时不需要函数返回结果，返回类型可以是Unit，定义时去掉{}前的=，如下：

**def** printEvery(c:*String*\*) //此处无=，相当于:Unit=

{

c.foreach(x => println(x))

}

printEvery("1","c","b")

## 异常处理

**try** {

//首先执行这里，如果报错则执行catch部分

} **catch** {

**case** t: *Throwable* => t.printStackTrace() // **TODO**: handle error

**case** ex:IOException => println("自定义内容")

}

**finally**

{

// 必执行的一部分

}

eclipse里录入try后按组合键：alt+/ 可以提示。

## 打包部署

配置Scala和Java的环境变量。

确保Path里能访问到scala、java、jar命令。

编译一个类：

scalac Test.scala // 由于我们是用Eclipse自动编译，所以这个命令基本不用

scala Test //手工执行一个类

Scala工程打包：

cd bin

**jar** -cvf a.jar ./

运行a.jar：

scala -cp a.jar com.cloudy.scala.lesson.Test 123