## while循环

### 格式

* while循环语句格式：

初始化表达式①

while(布尔表达式②){

循环体③

步进表达式④

}

### 执行流程

* 执行流程

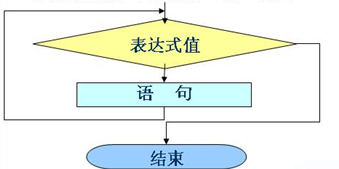
执行顺序：①②③④>②③④>②③④…②不满足为止。

①负责完成循环变量初始化。

②负责判断是否满足循环条件，不满足则跳出循环。

③具体执行的语句。

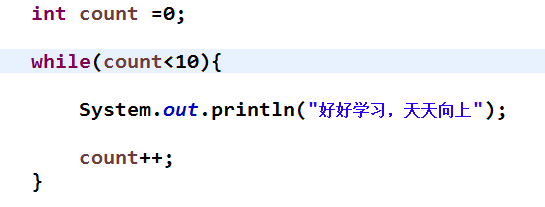
④循环后，循环变量的变化情况。



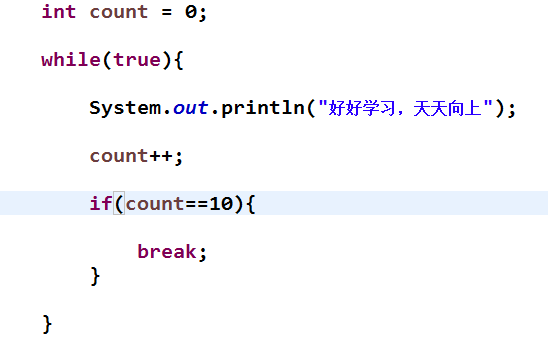
### 案例

#### 案例1：输出10次 “好好学习，天天向上”

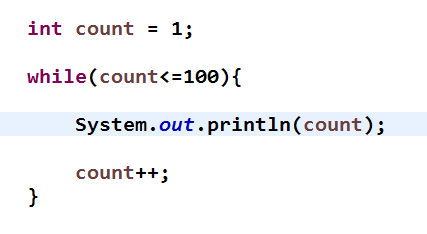
**方法一：**



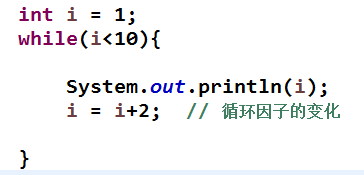
**方法二：（一定要有break，否则死循环）**



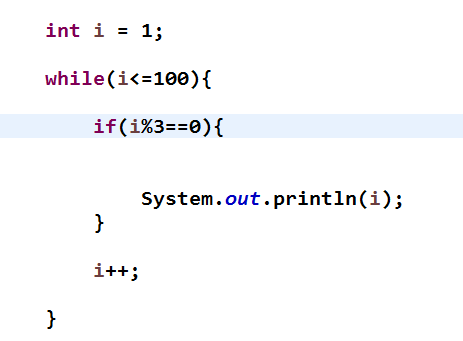
#### 案例2：输出1到100



#### 案例3：在控制台输出1,3,5,7,9

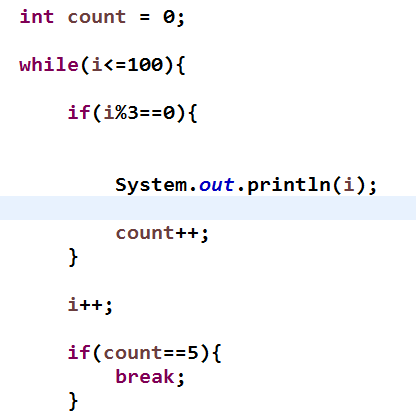


#### 案例4：输出1到100之间能被3整除的数

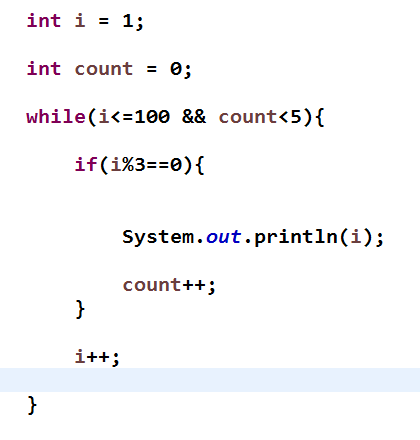


#### 案例5：输出1到100之间能被3整除的前5个数

**方法一：**



**方法二：**



#### **案例6：计算1+2+3+4+5+6+7+8+9+10 的值**

|  |
| --- |
| **int** sum = 0;  **int** i =1;  **while**(i<=10){  sum += i;  i++;  }  System.***out***.println(sum); |

### for 和while的区别

|  |
| --- |
| **public** **static** **void** main(String[] args) {  **for** (**int** x = 0; x < 5; x++) {  System.*out*.println("hello java");  }  System.*out*.println(x);  //x cannot be resolved to a variable  **int** y = 0;  **while** (y < 5) {  System.*out*.println("hello world");  y++;  }  System.*out*.println(y);  } |

**注意：**

解释 x 为什么会找不到,注意了变量的作用域,也就是变量的作用范围。x 只在 for 循环的大括号内有效,出了这个区域，就无效了.在内存中就消失了。x消失后,仍要访问它,肯定会报错的。

y 就不一样了，y 是定义在while 外的。while循环完毕仍有效 while的初始化 动作在外边，循环结束后y 仍然存在。

当定义的y 只作为循环增量存在的话的，循环完毕后y就没有用了，但是y还是占着一块内存。所以，如果定义的变量只作为循环增量存在的话，就用for 循环可以节约内存。

其实for 和while 是可以互换的。

**最后总结：**

|  |
| --- |
| 1、for里面的两个表达式运行的顺序，初始化表达式只读一次，判断循环条件，为真就执行循环体，然后再执行循环后的操作表达式，接着继续判断循环条件，重复找个过程，直到条件不满足为止。  2、while与for可以互换，区别在于for为了循环而定义的变量在for循环结束时就在内存中释放。而while循环使用的变量在循环结束后还可以继续使用。  3、最简单无限循环格式：while(true) , for(;;),无限循环存在的原因是并不知道循环多少次，而是根据某些条件，来控制循环。推荐使用while(true) |

## do while循环

* do...while循环格式

初始化表达式①

do{

循环体③

步进表达式④

}while(布尔表达式②);

* 执行流程

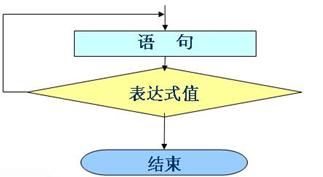
执行顺序：①③④>②③④>②③④…②不满足为止。

①负责完成循环变量初始化。

②负责判断是否满足循环条件，不满足则跳出循环。

③具体执行的语句

④循环后，循环变量的变化情况



* 注意事项

1、循环条件表达式必须是boolean类型

2、while循环条件后面，要有一个分号

* 代码演示

|  |
| --- |
| **int** i = 1;  **do**{  System.***out***.println(i);  i++;  }**while**(i<=100); |

* while 循环和do while循环的区别

while:先判断条件，只有条件满足才执行循环体。

do while: 先执行循环体，再判断条件，条件满足，再继续执行循环体。

简单一句话：do while：无论条件是否满足，循环体至少执行一次。

## 多重循环（嵌套循环）

### 结构：

|  |
| --- |
| for(int i=1;i<=10;i++){  for(int j=1;j<=10;j++){  }  } |

### 说明

1. 在一次大的循环中，每次的循环内容（循环体）又是一个复杂的操作，是重复的操作，就需要在大循环中，定义一个小循环。称为循环的嵌套。
2. 格式：

for ( 初始化语句1; 条件表达式1; 初始化变量1的自增) {

for ( 初始化语句2 ; 条件表达式2 ; 初始化变量2的自增 ) {

循环体语句;

}

}

1. 执行流程：

1、初始化语句1

2、判断条件表达式1，计算如果为false，外层循环结束

3、判断条件表达式1，计算如果为true，初始化语句2

4、计算条件表达式2，计算结果为false，内循环结束，执行初始化变量1的自增，回到第2步

5、计算条件表达式2，计算结果为true，执行循环体语句，

6、执行初始化变量2的自增，回到第4步

### 示例

public static void main(String[] args) {

for (int i = 1; i <= 3; i++) {

for (int j = 1; j <= 5; j++) {

System.out.println("i=" + i + ", j=" + j);

}

}

}

}

### 案例

#### 案例1：输出图形：

**\*\*\*\*\***

**\*\*\*\*\***

**\*\*\*\*\***

**\*\*\*\*\***

|  |
| --- |
| **public** **static** **void** main(String[] args) {    //死去活来法(常量变异法)  /\*  System.out.println("\*\*\*\*\*");  System.out.println("\*\*\*\*\*");  System.out.println("\*\*\*\*\*");  System.out.println("\*\*\*\*\*");  \*/        **for**(**int** i=1;i<=4;i++){ // 输出4行  /\*  System.out.print("\*");  System.out.print("\*");  System.out.print("\*");  System.out.print("\*");  System.out.print("\*");  \*/    **for**(**int** j=1;j<=5;j++){ // 每行输出5个\*  System.***out***.print("\*");  }    System.***out***.println(); // 输入一个换行  }    // 总结： 双重循环，输出的内容是平面。外层循环控制输出多少行，内层循环控制每行的内容  } |

#### 案例2：输出图形：

**\***

**\*\*\***

**\*\*\*\*\***

**\*\*\*\*\*\*\***

**\*\*\*\*\*\*\*\*\***

|  |
| --- |
| /\*  -----------------------  \*  \*\*\*  \*\*\*\*\*  \*\*\*\*\*\*\*  \*\*\*\*\*\*\*\*\*  行数： 1 2 3 4 5 i  空格数： 4 3 2 1 0 j = 5-i  星数： 1 3 5 7 9 k = 2\*i-1    -----------------------  \*/  **public** **static** **void** main(String[] args) {    **for**(**int** i=1;i<=5;i++){ // 控制行数    **for**(**int** j=1;j<=5-i;j++){ // 负责输出本行的空格    System.***out***.print(" ");  }    **for**(**int** k=1;k<=2\*i-1;k++){ // 负责输出本行的\*  System.***out***.print("\*");  }    System.***out***.println(); // 换行  }    } |

#### 案例3：输出图形：

**\***

**\* \***

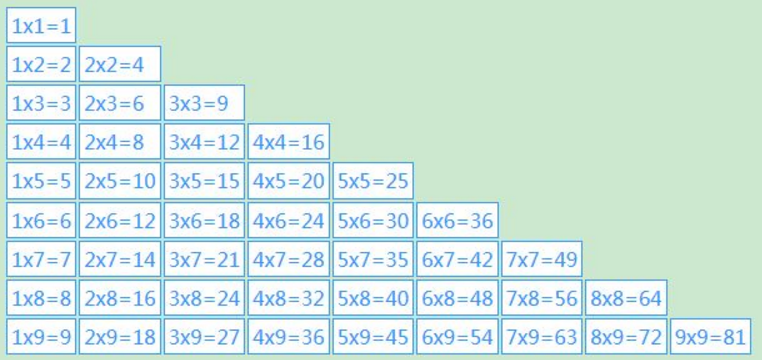
**\* \* \***

**\* \* \* \***

**\* \* \* \* \***

|  |
| --- |
| /\*  ---------------------  \*  \* \*  \* \* \*  \* \* \* \*  \* \* \* \* \*  ---------------------  行： 1 2 3 4 5 i  空： 4 3 2 1 0 j = 5-i  星： 1 2 3 4 5 k = i  \*/    **public** **static** **void** main(String[] args) {    **for**(**int** i=1;i<=5;i++){    **for**(**int** j=1;j<=5-i;j++){    System.***out***.print(" ");  }    **for**(**int** k=1;k<=i;k++){  System.***out***.print("\* ");  }    System.***out***.println();  }    } |
| **package** package04;  /\*\*  \* **@author** liulei    \* ----> i = 0, 前面4个空格，1个\*  \* \* ----> i = 1, 前面3个空格，2个\*  \* \* \* ----> i = 2, 前面2个空格，3个\*  \* \* \* \* ----> i = 3, 前面1个空格，4个\*  \* \* \* \* \* ----> i = 4, 前面0个空格，5个\*  \*/  **public** **class** Demo07 {  **public** **static** **void** main(String[] args) {    **for**(**int** i = 0;i < 5;i ++) {    **for**(**int** j = 0;j < 4 - i;j ++) {  System.***out***.print(" ");  }    **for**(**int** j = 0;j < i + 1;j ++) {  **if**(j != i) {  System.***out***.print("\* ");  }**else** {  System.***out***.println("\*");  }  }    // System.out.println();    }  }  } |

#### 案例4：输出99乘法表：



|  |
| --- |
| **public** **static** **void** main(String[] args) {    **for**(**int** i = 1;i <= 9;i ++ ) {  **for**(**int** j = 1;j <= i;j ++) {  System.***out***.print(j + "x" + i + "=" + i \* j + " ");  }  System.***out***.println();  }  } |

#### 案例5：输出100 到 1000之间的所有素数

|  |
| --- |
| **public** **static** **void** main(String[] args) {    **for**(**int** i = 100;i <= 1000;i ++) {    **boolean** flag = **true**;  **for**(**int** j = 2;j < i;j ++) {  **if**(i % j == 0) {  flag = **false**;  **break**;  }  }  **if**(flag) {  System.***out***.println(i);  }  }  } |

## 随机数

### 概述

Java中提供了Math类中的random()方法，用来生成一个0到1之间的随机数 [0,1)

另外Java通过了一个Random类专门用来生成随机数

### 使用方式

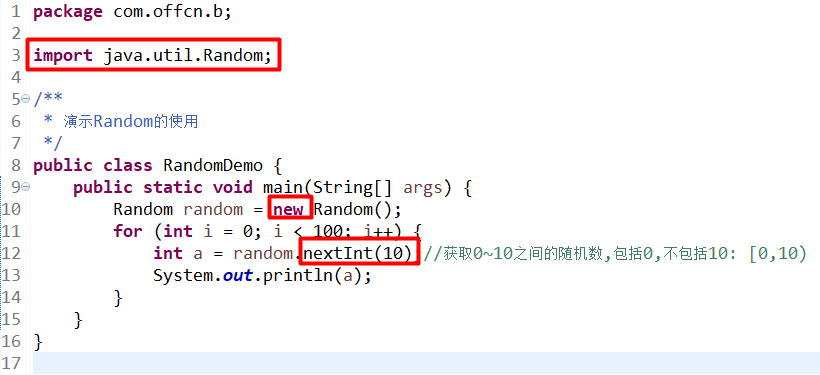
Math 类（不常用） ： double x = Math.random();

Random类：

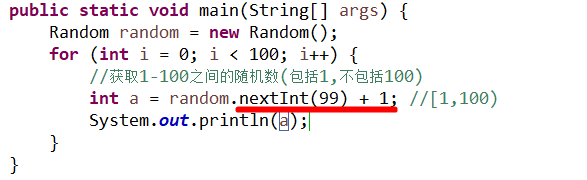
1. 创建实例格式：Random 变量名 = new Random();
2. import导包：所属包java.util.Random
3. 调用方法int number = r.nextInt(10);// [0,10) 产生的数据在0到10之间，包括0，不包括10。

### 示例代码

* + 获取0-10之间的随机数(包括0,不包括10)



* + 获取1-100之间的随机数(包括1,不包括100)



### 练习

猜数字小游戏案例:系统产生一个1-100之间的随机数，请猜出这个数是多少。10次机会

思路:

1.系统生成

2.用户输入

3.比较

4.做其他的额外功能...

* + 代码实现



## String类型入门

### 概念

字符串：由字符组成的串

### 定义

定义：String s = “abcd”；

String s = “”;空串

由于字符串在java中的使用频率非常高，所以java为String类型提供了类似于基本数据类型的定义方式。

获取字符串的长度(由多少个字符组成)

### 方法

### str.equals(“asdasd”)

1 获取字符串长度

int length = s.length(); （注意：.length( ) 是一个方法，有括号）

2 获取字符串中的某个字符

获取字符串中某个位置的字符(下标索引从0开始)，得到char类型的值。

char c = s.charAt(0); ------- ‘a’

### 案例

#### **案例1：获取字符串的第一个字符**

|  |
| --- |
| **char** c1 = str.charAt(0); |

#### **案例2：获取字符串的最后一个字符**

|  |
| --- |
| **char** last= str.charAt(str.length()-1); |

#### 案例3：分行输出字符串的每个字符

|  |
| --- |
| String str = "Helloworld";  **for**(**int** i=0;i<str.length();i++){  System.***out***.println(str.charAt(i));  } |

#### 案例4：String str = "aaa bbb ccc"; 去掉所有空格

|  |
| --- |
| /\*  \* 思路： 1 先拿一个空串  \* 2 遍历原来的字符串（取出每个字符串），判断取出的字符是不是空格  \* 3 如果不是空格，串到空串上。如果是空格，不要  \* \*/    // 去掉字符串中的所有空格  String str = "aaa bbb ccc ddd";  String result = ""; // 空串  **for**(**int** i=0;i<str.length();i++){  **char** ch = str.charAt(i);  **if**(ch!=' '){  result =result+ch;  }  }    System.***out***.println(str);  System.***out***.println(result); |

#### 案例5：反转字符串

|  |
| --- |
| String str = "abcdefg";  String result = "";  /\*  for(int i=str.length()-1;i>=0;i--){  result = result+str.charAt(i);  }  System.out.println(result);  \*/    /\*  for(int i=0;i<str.length();i++){  result =result+str.charAt(str.length()-1-i);  }  System.out.println(result);  \*/    **for**(**int** i=0;i<str.length();i++){  **char** c = str.charAt(i);  result = c+result; // result= result +....  }    System.***out***.println(result); |