回顾：

1. 循环: 反复多次执行相同或相似的代码
2. 三要素：

循环变量：在整个循环过程中所反复改变的那个数

1. 循环结构:

1) while: 先判断后执行，有可能一次都不执行

2) do..while: 先执行后判断，至少执行一次

要素1与要素三相同时，首选do..while

3) for: 与次数相关，应用率最高

正课：

1. 循环结构:

1) while:

2) do..while:

3) for: 与次数相关，应用率最高

1. break: 跳出循环

continue: 跳过循环体中剩余语句而进入下一次循环

1. 三种循环如何选择:

1) 先看循环是否与次数相关:

1.1) 若与次数相关-------------直接上for

1.2) 若与次数无关，再看要素1与要素3是否相同:

1.2.1) 若要素1和要素3相同-----------------直接上do..while

1.2.2) 若要素1和要素3不同-----------------直接上while

1. 嵌套循环:

1) 循环中套循环，常常多行多列时使用，一般外层控制行，内层控制列

2) 执行过程: 外层循环走一次，内层循环走所有次

3) 建议: 嵌套层数越少越好，能用一层就不用两层，能用两层就不用三层，

若业务必须通过三层以上的循环来解决，说明设计有问题

4) break只能跳出一层循环

1. 数组:

1) 相同数据类型元素的集合

2) 是一种数据类型(引用类型)------引用类型周五开始讲

3) 数组的声明:

int[ ] arr = new int[10];

4) 数组的初始化:

int[ ] arr = new int[4]; //0,0,0,0

int[ ] arr = {2,5,8,9}; //2,5,8,9

int[ ] arr = new int[ ]{2,5,8,9}; //2,5,8,9

int[ ] arr;

arr = {2,5,8,9}; //编译错误，此方式只能声明同时初始化

arr = new int[ ]{2,5,8,9}; //正确

5) 数组的访问:

5.1) 通过(数组名.length)可以获取数组的长度(元素的个数)

int[ ] arr = new int[10];

System.out.println(arr.length); //10

5.2) 通过数组的下标/索引来访问数组中的元素

下标从0开始，最大到(数组的长度-1)

int[ ] arr = new int[3];

arr[0] = 100; //第一个元素赋值为100

arr[1] = 200; //第二个元素赋值为200

arr[2] = 300; //第三个元素赋值为300

//arr[3] = 400; //数组下标越界异常(程序运行时发生的错误)

System.out.println( arr[arr.length-1] ); //输出最后一个元素的值

6) 数组的遍历:

int[ ] arr = new int[10];

for(int i=0; i<arr.length ; i++){

arr[i] = 100;

}

for(int i=0 ; i<arr.length ; i++){

System.out.println(arr[i]);

}

int[ ] arr = new int[10];

for( int i=0; i<arr.length; i++ ){

arr[i] = 100; //arr[i]表示arr中的每一个元素

}

for(int i=0 ; i<arr.length ; i++){

System.out.println(arr[i]);

}

System.out.println(arr[0]);

System.out.println(arr[1]);

System.out.println(arr[2]);

System.out.println(arr[3]);

System.out.println(arr[4]);

System.out.println(arr[5]);

System.out.println(arr[6]);

System.out.println(arr[7]);

System.out.println(arr[8]);

System.out.println(arr[9]);

arr[0] = 100;

arr[1] = 100;

arr[2] = 100;

arr[3] = 100;

arr[4] = 100;

arr[5] = 100;

arr[6] = 100;

arr[7] = 100;

arr[8] = 100;

arr[9] = 100;

int[ ] arr = new int[3]; //0,0,0

//给第二个数赋值为100

arr[1] = 100;

arr[0]---------代表arr中的第一个元素(int)

arr[1]---------代表arr中的第二个元素(int)

arr[2]---------……………………….三个元素(int)

int a = 0;

int b = 0;

int c = 0;

//给第二个数赋值为100

b = 100;

下课休息15分钟---------------4:15准时回来

int a = 5;

//声明整型数组a，包含5个元素

//每个元素都是int类型，默认值0(byte,short,int,long,char)

int[ ] a = new int[5]; //0，0，0，0，0

//声明浮点型数组d，包含10个元素

//每个元素都是double型，默认值为0.0(float,double)

double[ ] d = new double[10];

//声明布尔型数组b，包含26个元素，

//每个元素都是boolean型，默认值为false(boolean)

boolean[ ] b = new boolean[26];

int a ; //声明整型变量a

int[ ] a ; //声明整型数组变量a

double[ ] d; //浮点型数组

boolean[ ] b; //布尔型数组

char[ ] c; //字符型数组

下课休息15分钟-------------------3:06准时上课

数组 元素

衣柜 衣服

橱柜 厨具

鞋柜 鞋

书柜 书

数组里面的数-------元素

数组------数的组合，一堆数

周二------所有语言基础知识讲完了

周三------案例时间

通过break跳出所有层循环，只是你自己想当然的想法-----实际应用中没有这样的需求

6000000次

for( int i=1;i<=100;i++ ){

for(int j=1;j<=200;j++){

for(int k=1;k<=300;k++){

if( boolean条件 ){

break; //跳出k层循环

}

输出aaa

}

}

}

i=1

j=1

k走30次

j=2

k 走30次

…

j=20

k走30次

j走20次

i=2

j走20次

…

i=10

j走20次

\t: 水平制表位，固定占8位

1\*1=1

12=2 22=4

13=3. 23=6. 3\*3=9

0到99之间的随机数

Math.random()---------------0.0到0.9999999999999…

\*100------------------------0.0到99.999999999999…

(int)-------------------------0到99

int score = 0; //得分

for( int i=1 ; i<=10 ; i++ ){ //10次

1.出题 2.答题 3.判题

}

System.out.println(“得分为:”+score );

(1)45+67=?

算吧!

112

答对了

(2)78+4=?

算吧!

83

答错了

(3)45+67=?

算吧!

112

答对了

(4)78+4=?

算吧!

83

答错了

(5)45+67=?

算吧!

112

答对了

(6)78+4=?

算吧!

83

答错了

(7)45+67=?

算吧!

112

答对了

(8)78+4=?

算吧!

83

答错了

(9)45+67=?

算吧!

112

答对了

(10)78+4=?

算吧!

83

答错了

总分为: 50