# 数组

1. 概念

内存中一段连续的存储空间，存储的值必须是相同数据类型

特点：数据类型一致，长度不变

相关概念：

数组中每个数据格称为数组元素

对每个元素的操作称为对数组的访问

如果需要访问数组的元素，需要使用下标

下标自动生成，从0开始，一次+1，每个下标对应一个元素

数组中有n个元素，n为数组的长度,数组长度=最大下标+1

数组有默认值，整数类型：0 小数类型：0.0 布尔类型：false 字符类型：空格 字符串：null

1. 语法
   1. 声明数组

数据类型[] 数组名；//常用

数据类型 []数组名；

数据类型 数组名[];

* 1. 创建数组

1. 先定义数组，再分配空间

数据类型[] 数组名;

数组名 = new 数据类型[长度];

1. 定义数组并分配空间

数据类型[] 数组名=new 数据类型[长度];

1. 定义数组并赋值

数据类型[] 数组名=new 数据类型[]{value1,value2,value3...};

1. 定义数组并赋值，不能换行写

数组类型[] 数组名={value1,value2,value3,...}；

* 1. 访问元素

数组名[下标]

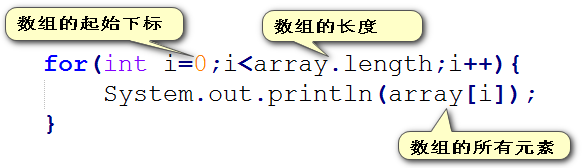
访问下标大于数组的最大下标出现：java.lang.ArrayIndexOutofBoundsException

C) 访问数组的长度

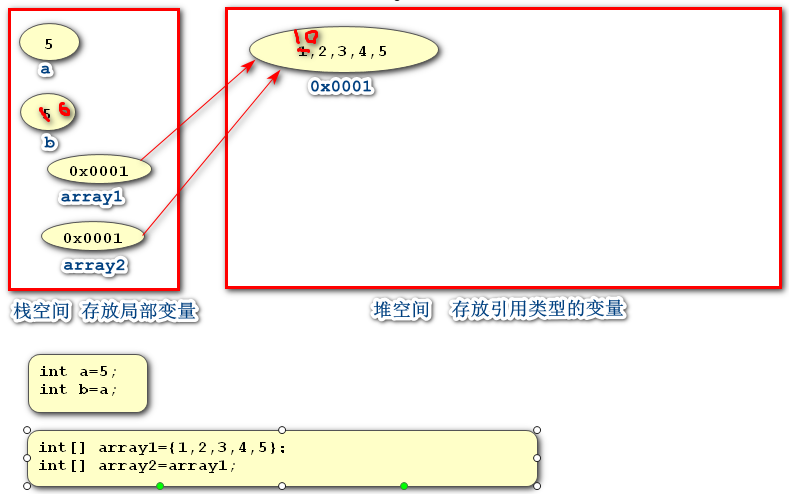
数组名.length

1. 数组遍历

对数组元素的依次访问



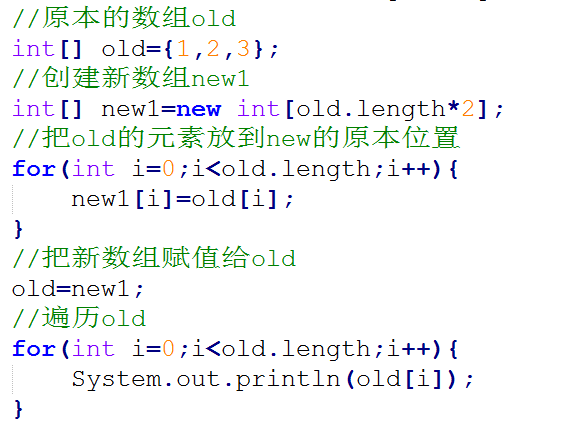
1. 数组的内存结构



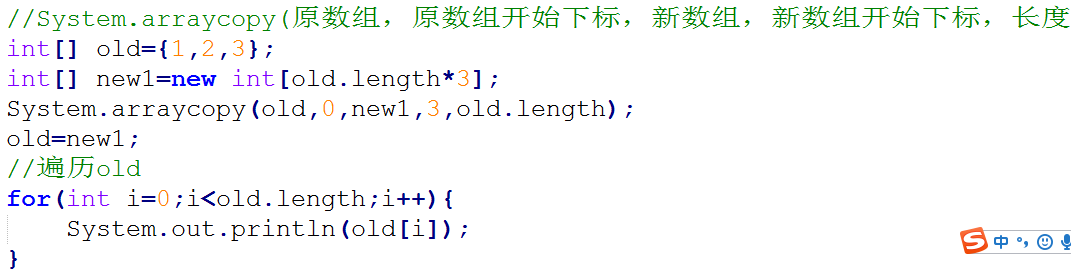
1. 数组扩容

数组大搬家

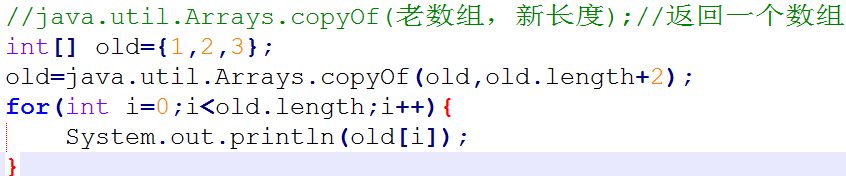
方式一：循环将原数组的元素赋值给新数组



方式二：System.arraycopy(原数组，原数组开始下标，新数组，新数组开始下标，长度)



方式三：java.util.Arrays.copyOf(老数组，新长度);//返回一个数组



1. 可变长参数

概念：可以接收多个相同类型的实参

语法：修饰符 返回值类型 函数名（数据类型... 变量名）{}

放在参数表的最后,可变长参数只能有一个

使用：和数组一样

