

#数组 声明方式: `type[] array ;`//一般使用这种, 更体现数组是一种类型的概念 `type array[]`; 理解: 数组也是一种类型; 定义时不能在[]内规定长度, 因为这是引用类型, 必须要用new来进行初始化。

### 初始化

静态初始化: 由程序员显式指定每个数组元素的初始值, 由系统决定长度。例如: `int[] Array = new int[]{5,6,7}; int[] Array2 = {6,7,8,9};`//简约形式, 只有在声明时一起进行初始化才能用这种形式 数组的长度使用length属性!

动态初始化: 程序员只规定长度, 系统分配初始值。例如: `int[] Array = new int[5];` foreach循环使用形式: `for(type x : array/collection){}` 系统自动将数组或者集合中的值赋值给变量X中。

多维数组 声明格式: `type[][] arrayName;`//二维数组 `arrayName = new arrayName[length][];` //先初始化一维的长度, 然后逐一初始化二维 `type[][] arrayName = new arrayName[length][length2];`//同时初始化 如果想要二维以上, 则可以先定义一个Object[]类型数组, 因为其元素是 Object所以可以继续指向Object[]数组。

Arrays工具类 静态方法:

`int binarySearch(type[] a,type key)`:用二分法查询key元素所在的索引, 要求数组必须按升序排列 `int binarySearch(type[] a,int from,int to,type key)`:区间查找, 同样要升序; `type[] copyOf(type[] a,int length)`:赋值一个新数组, 从前面开始复制 (如果长度小于a的长度), 如果大于, 则补充默认值。  
`type[] copyOfRange(type[] a,int from,int to)`:复制一个区间; `boolean equals(type[] a, type[] b)`:长度相同, 数组元素相同为true; `void fill(type[] a,type val)`:把数组中的所有元素赋值为val; `void fill(type[] a,int from,int to,type val)`:区间赋值; `void sort(type[] a)`:排序, 最终升序排列。collections.sort方法也是最终使用这个排序。源码解析: <https://blog.csdn.net/u013935164/article/details/80303830>和 <https://www.cnblogs.com/warehouse/p/9343195.html> `String toString(type[] a)`:把元素连起来, 用逗号隔开。

### 使用Array类创建数组

`static Object newInstance(Class<?> com, int...length)`:静态方法, 直接使用**Array.newInstance**创建数组。第一个参数是数组类型, 后面的参数的个数代表了维度, 参数的大小代表了数组的长度。注意返回值为**Object**, 需要强制类型转换 如:

```
//一维数组,String类型的, 长度为10的数组
Array.newInstance(String.class,10);
//三维数组, String类型的, 长度分别为3,8,10的数组, 即a[3][8][10]
Array.newInstance(String.class,3,8,10);
```