# Java 数组

## 一、什么是Java中的数组？

数组是一种数据结构，可以存储多个相同类型的元素；在Java中，数组是一种对象，它可以存储基本数据类型和对象类型。

## 二、数组有哪些常见的操作？

* 创建数组：使用new操作符创建一个新的数组对象。

* 访问元素：使用索引访问数组中的元素。

* 遍历数组：使用循环结构遍历数组中的所有元素。

* 复制数组：使用System.arraycopy()方法或Arrays.copyOf()方法将一个数组复制到另一个数组中。

* 排序数组：使用Arrays.sort()方法对数组进行排序。

* 搜索数组：使用Arrays.binarySearch()方法在一个已排序的数组中搜索一个元素。

## 三、数组的五种赋值方法和使用方法

#### 声明数组

int[] arr;   
//开辟三个空间  
arr = new int[3];   
//向数组中赋值，0、1、2是下标   
arr[0] = 30;   
arr[1] = 97;   
arr[2] = 65;

#### 声明数组并且分配空间

int[] arr = new int[3];   
//开辟三个空间   
//向数组中赋值   
arr[0] = 30;   
arr[1] = 97;   
arr[2] = 65;

#### 声明数组同时赋值(1)

int[] arr = new int[]{30, 97, 65};

#### 声明数组同时赋值(2)

int[] arr = {30, 97, 65};

#### 从控制台输入向数组赋值

int[] arr = new int[3];   
System.out.println("向数组中存入三个数据：");   
//arr.length是数组的长度   
for (int i = 0; i < arr.length; i++) {   
//i是下标   
 arr[i] = input.nextInt();  
 }   
//把数组中的值输出   
for (int i = 0; i < arr.length; i++) {   
//输出数组中的值   
 System.out.println(arr[i]);   
}

## 四、求总和平均

int i = 0; int zong = 0;   
int[] arr = new int[3];   
for (i = 0; i < arr.length; i++) {  
 System.out.print("输入第" + (i + 1) + "位同学的成绩：");   
 arr[i] = input.nextInt();  
 zong = zong + arr[i];   
}   
System.out.println(i + "位同学的总分是：" + zong + "平均分是：" + zong \* 1.0 / arr.length);

## 五、求数组中最大值最小值

int[] arr = { 90, 100, 98, 67, 45, 79 };  
 int max = arr[0];  
 int min = arr[0];  
 for (int i = 1; i < arr.length; i++) {  
 if (arr[i] > max) {  
 max = arr[i];   
}  
 if (arr[i] < min) {  
 min = arr[i];   
}   
}   
System.out.println("最大值为：" + max); System.out.println("最小值为：" + min);

## 六、对数组进行升序排序

数组升序输出  
 int[] arr1 = { 90, 100, 98, 67, 45, 79 };   
 Arrays.sort(arr1);  
 for (int i = 0; i < arr1.length; i++) {   
 System.out.println(arr1[i]);   
 }   
  
数组降序输出   
int[] arr2 = {90, 100, 98, 67, 45, 79};   
Arrays.sort(arr2);   
for (int i = arr2.length - 1; i >= 0; i--) {   
 System.out.println(arr2[i]);   
}

## 六、不调用方法进行数组排序

int[] array = {10, 30, 17, 6, 29, 20};  
int temp; for(int i = 0;i<array.length -1;i++){  
 for (int j = 0; j < array.length - 1 - i; j++) {  
 if (array[j] < array[j + 1]) {  
 //<为降序，>为升序   
 temp = array[j];  
 array[j] = array[j + 1];  
 array[j + 1] = temp;  
 }  
 }  
}   
for(int i :array){  
 System.out.print(i + "\t");  
}

## 七、Arrays 类常用方法

Arrays类位于 java.util 包中， import java.util.Arrays;

int[] array1 = {10, 20, 16, 19, 5};  
 int[] array2 = {10, 20, 16, 19, 5};  
 //boolean equals(array1, array2)比较两个数组是否相等   
System.out.println(Arrays.equals(array1, array2));   
//打印true   
  
//String toString(array)将数组转换成字符串显示值  
 System.out.println(Arrays.toString(array1));   
//打印[10, 20, 16, 19, 5]   
  
//void fill(array, val)将数组的所有值填充为val Arrays.fill(array2, 20); System.out.println(Arrays.toString(array2));  
 //打印[20, 20, 20, 20, 20]   
  
//数组copyOf(array, newLength)将array复制成指定长度的新数组 int[] array3 =Arrays.copyOf(array1, 10);   
System.out.println(Arrays.toString(array3));   
//打印[10, 20, 16, 19, 5, 0, 0, 0, 0, 0]   
  
//int binarySearch(array, key)在array数组中查找指定的值，并返回其索引，前提是数组是有序的 Arrays.sort(array1);   
System.out.println(Arrays.binarySearch(array1, 16));   
//打印2