**package** 八月十号;

**import** java.util.Scanner;

**public** **class** test2 {

**public** **static** **void** f(Object x){

}

**public** **static** **void** main(String[] args) {

Scanner sc=**new** Scanner(System.***in***);

// Object a=1;

// Object b="sfsdada";

// f(1);

// f('A');

// f("Sjjdklsfjosaijfosiajfoerja");

// int n=sc.nextInt();

// for(int i=0;i<10;i++){

// if(i==n)continue;

// System.out.println(i);

// }

// System.out.println((int)('%'));

// int a=-3;

// int b=4;

// System.out.println(Math.sin(Math.PI/180));

//

// StringBuilder sb=new StringBuilder("");

// for(int i=0;i<26;i++){

// sb.append((char)('A'+i));

// System.out.println(sb);

// }

// String s="abcdefg";

//// int c=s.indexOf('d');

// char cc=s.charAt(3);

// System.out.println(cc);

**int** a[]=**new** **int**[10];

**int** b[][]=**new** **int**[10][10];

b[1][1]=1;

**for**(**int** i=2;i<10;i++){

**for**(**int** j=1;j<10;j++){

b[i][j]=b[i-1][j]+b[i-1][j-1];

}

}

**for**(**int** i=0;i<10;i++){

**for**(**int** j=0;j<10;j++){

**if**(b[i][j]==0)**continue**;

System.***out***.print(b[i][j]+" ");

}

System.***out***.println();

}

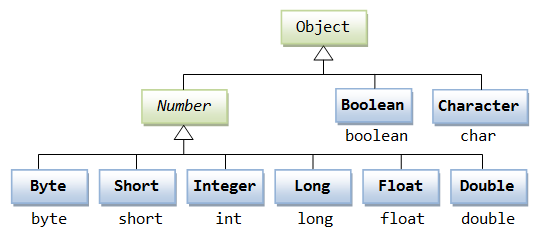
System.***out***.println(a[7]);

}

}

# Java Number & Math 类

一般地，当需要使用数字的时候，我们通常使用内置数据类型，如：**byte、int、long、double** 等。



# Java String 类

字符串广泛应用 在 Java 编程中，在 Java 中字符串属于对象，Java 提供了 String 类来创建和操作字符串。

## String 方法

下面是 String 类支持的方法，更多详细，参看 [Java String API](https://www.runoob.com/manual/jdk1.6/java/lang/String.html" \t "https://www.runoob.com/java/_blank) 文档:

|  |  |
| --- | --- |
| **SN(序号)** | **方法描述** |
| 1 | [char charAt(int index)](https://www.runoob.com/java/java-string-charat.html" \t "https://www.runoob.com/java/_blank) 返回指定索引处的 char 值。 |
| 2 | [int compareTo(Object o)](https://www.runoob.com/java/java-string-compareto.html" \t "https://www.runoob.com/java/_blank) 把这个字符串和另一个对象比较。 |
| 3 | [int compareTo(String anotherString)](https://www.runoob.com/java/java-string-compareto.html" \t "https://www.runoob.com/java/_blank) 按字典顺序比较两个字符串。 |
| 4 | [int compareToIgnoreCase(String str)](https://www.runoob.com/java/java-string-comparetoignorecase.html" \t "https://www.runoob.com/java/_blank) 按字典顺序比较两个字符串，不考虑大小写。 |
| 5 | [String concat(String str)](https://www.runoob.com/java/java-string-concat.html" \t "https://www.runoob.com/java/_blank) 将指定字符串连接到此字符串的结尾。 |
| 6 | [boolean contentEquals(StringBuffer sb)](https://www.runoob.com/java/java-string-contentequals.html" \t "https://www.runoob.com/java/_blank) 当且仅当字符串与指定的StringBuffer有相同顺序的字符时候返回真。 |
| 7 | [static String copyValueOf(char[] data)](https://www.runoob.com/java/java-string-copyvalueof.html" \t "https://www.runoob.com/java/_blank) 返回指定数组中表示该字符序列的 String。 |
| 8 | [static String copyValueOf(char[] data, int offset, int count)](https://www.runoob.com/java/java-string-copyvalueof.html" \t "https://www.runoob.com/java/_blank) 返回指定数组中表示该字符序列的 String。 |
| 9 | [boolean endsWith(String suffix)](https://www.runoob.com/java/java-string-endswith.html" \t "https://www.runoob.com/java/_blank) 测试此字符串是否以指定的后缀结束。 |
| 10 | [boolean equals(Object anObject)](https://www.runoob.com/java/java-string-equals.html" \t "https://www.runoob.com/java/_blank) 将此字符串与指定的对象比较。 |
| 11 | [boolean equalsIgnoreCase(String anotherString)](https://www.runoob.com/java/java-string-equalsignorecase.html" \t "https://www.runoob.com/java/_blank) 将此 String 与另一个 String 比较，不考虑大小写。 |
| 12 | [byte[] getBytes()](https://www.runoob.com/java/java-string-getbytes.html" \t "https://www.runoob.com/java/_blank)  使用平台的默认字符集将此 String 编码为 byte 序列，并将结果存储到一个新的 byte 数组中。 |
| 13 | [byte[] getBytes(String charsetName)](https://www.runoob.com/java/java-string-getbytes.html" \t "https://www.runoob.com/java/_blank) 使用指定的字符集将此 String 编码为 byte 序列，并将结果存储到一个新的 byte 数组中。 |
| 14 | [void getChars(int srcBegin, int srcEnd, char[] dst, int dstBegin)](https://www.runoob.com/java/java-string-getchars.html" \t "https://www.runoob.com/java/_blank) 将字符从此字符串复制到目标字符数组。 |
| 15 | [int hashCode()](https://www.runoob.com/java/java-string-hashcode.html" \t "https://www.runoob.com/java/_blank) 返回此字符串的哈希码。 |
| 16 | [int indexOf(int ch)](https://www.runoob.com/java/java-string-indexof.html" \t "https://www.runoob.com/java/_blank) 返回指定字符在此字符串中第一次出现处的索引。 |
| 17 | [int indexOf(int ch, int fromIndex)](https://www.runoob.com/java/java-string-indexof.html" \t "https://www.runoob.com/java/_blank) 返回在此字符串中第一次出现指定字符处的索引，从指定的索引开始搜索。 |
| 18 | [int indexOf(String str)](https://www.runoob.com/java/java-string-indexof.html" \t "https://www.runoob.com/java/_blank)  返回指定子字符串在此字符串中第一次出现处的索引。 |
| 19 | [int indexOf(String str, int fromIndex)](https://www.runoob.com/java/java-string-indexof.html" \t "https://www.runoob.com/java/_blank) 返回指定子字符串在此字符串中第一次出现处的索引，从指定的索引开始。 |
| 20 | [String intern()](https://www.runoob.com/java/java-string-intern.html" \t "https://www.runoob.com/java/_blank)  返回字符串对象的规范化表示形式。 |
| 21 | [int lastIndexOf(int ch)](https://www.runoob.com/java/java-string-lastindexof.html" \t "https://www.runoob.com/java/_blank)  返回指定字符在此字符串中最后一次出现处的索引。 |
| 22 | [int lastIndexOf(int ch, int fromIndex)](https://www.runoob.com/java/java-string-lastindexof.html" \t "https://www.runoob.com/java/_blank) 返回指定字符在此字符串中最后一次出现处的索引，从指定的索引处开始进行反向搜索。 |
| 23 | [int lastIndexOf(String str)](https://www.runoob.com/java/java-string-lastindexof.html" \t "https://www.runoob.com/java/_blank) 返回指定子字符串在此字符串中最右边出现处的索引。 |
| 24 | [int lastIndexOf(String str, int fromIndex)](https://www.runoob.com/java/java-string-lastindexof.html" \t "https://www.runoob.com/java/_blank)  返回指定子字符串在此字符串中最后一次出现处的索引，从指定的索引开始反向搜索。 |
| 25 | [int length()](https://www.runoob.com/java/java-string-length.html" \t "https://www.runoob.com/java/_blank) 返回此字符串的长度。 |
| 26 | [boolean matches(String regex)](https://www.runoob.com/java/java-string-matches.html" \t "https://www.runoob.com/java/_blank) 告知此字符串是否匹配给定的正则表达式。 |
| 27 | [boolean regionMatches(boolean ignoreCase, int toffset, String other, int ooffset, int len)](https://www.runoob.com/java/java-string-regionmatches.html" \t "https://www.runoob.com/java/_blank) 测试两个字符串区域是否相等。 |
| 28 | [boolean regionMatches(int toffset, String other, int ooffset, int len)](https://www.runoob.com/java/java-string-regionmatches.html" \t "https://www.runoob.com/java/_blank) 测试两个字符串区域是否相等。 |
| 29 | [String replace(char oldChar, char newChar)](https://www.runoob.com/java/java-string-replace.html" \t "https://www.runoob.com/java/_blank) 返回一个新的字符串，它是通过用 newChar 替换此字符串中出现的所有 oldChar 得到的。 |
| 30 | [String replaceAll(String regex, String replacement)](https://www.runoob.com/java/java-string-replaceall.html" \t "https://www.runoob.com/java/_blank) 使用给定的 replacement 替换此字符串所有匹配给定的正则表达式的子字符串。 |
| 31 | [String replaceFirst(String regex, String replacement)](https://www.runoob.com/java/java-string-replacefirst.html" \t "https://www.runoob.com/java/_blank)  使用给定的 replacement 替换此字符串匹配给定的正则表达式的第一个子字符串。 |
| 32 | [String[] split(String regex)](https://www.runoob.com/java/java-string-split.html" \t "https://www.runoob.com/java/_blank) 根据给定正则表达式的匹配拆分此字符串。 |
| 33 | [String[] split(String regex, int limit)](https://www.runoob.com/java/java-string-split.html" \t "https://www.runoob.com/java/_blank) 根据匹配给定的正则表达式来拆分此字符串。 |
| 34 | [boolean startsWith(String prefix)](https://www.runoob.com/java/java-string-startswith.html" \t "https://www.runoob.com/java/_blank) 测试此字符串是否以指定的前缀开始。 |
| 35 | [boolean startsWith(String prefix, int toffset)](https://www.runoob.com/java/java-string-startswith.html" \t "https://www.runoob.com/java/_blank) 测试此字符串从指定索引开始的子字符串是否以指定前缀开始。 |
| 36 | [CharSequence subSequence(int beginIndex, int endIndex)](https://www.runoob.com/java/java-string-subsequence.html" \t "https://www.runoob.com/java/_blank)  返回一个新的字符序列，它是此序列的一个子序列。 |
| 37 | [String substring(int beginIndex)](https://www.runoob.com/java/java-string-substring.html" \t "https://www.runoob.com/java/_blank) 返回一个新的字符串，它是此字符串的一个子字符串。 |
| 38 | [String substring(int beginIndex, int endIndex)](https://www.runoob.com/java/java-string-substring.html" \t "https://www.runoob.com/java/_blank) 返回一个新字符串，它是此字符串的一个子字符串。 |
| 39 | [char[] toCharArray()](https://www.runoob.com/java/java-string-tochararray.html" \t "https://www.runoob.com/java/_blank) 将此字符串转换为一个新的字符数组。 |
| 40 | [String toLowerCase()](https://www.runoob.com/java/java-string-tolowercase.html" \t "https://www.runoob.com/java/_blank) 使用默认语言环境的规则将此 String 中的所有字符都转换为小写。 |
| 41 | [String toLowerCase(Locale locale)](https://www.runoob.com/java/java-string-tolowercase.html" \t "https://www.runoob.com/java/_blank)  使用给定 Locale 的规则将此 String 中的所有字符都转换为小写。 |
| 42 | [String toString()](https://www.runoob.com/java/java-string-tostring.html" \t "https://www.runoob.com/java/_blank)  返回此对象本身（它已经是一个字符串！）。 |
| 43 | [String toUpperCase()](https://www.runoob.com/java/java-string-touppercase.html" \t "https://www.runoob.com/java/_blank) 使用默认语言环境的规则将此 String 中的所有字符都转换为大写。 |
| 44 | [String toUpperCase(Locale locale)](https://www.runoob.com/java/java-string-touppercase.html" \t "https://www.runoob.com/java/_blank) 使用给定 Locale 的规则将此 String 中的所有字符都转换为大写。 |
| 45 | [String trim()](https://www.runoob.com/java/java-string-trim.html" \t "https://www.runoob.com/java/_blank) 返回字符串的副本，忽略前导空白和尾部空白。 |
| 46 | [static String valueOf(primitive data type x)](https://www.runoob.com/java/java-string-valueof.html" \t "https://www.runoob.com/java/_blank) 返回给定data type类型x参数的字符串表示形式。 |

# Java StringBuffer 和 StringBuilder 类

当对字符串进行修改的时候，需要使用 StringBuffer 和 StringBuilder 类。

和 String 类不同的是，StringBuffer 和 StringBuilder 类的对象能够被多次的修改，并且不产生新的未使用对象。

StringBuilder 类在 Java 5 中被提出，它和 StringBuffer 之间的最大不同在于 StringBuilder 的方法不是线程安全的（不能同步访问）。

由于 StringBuilder 相较于 StringBuffer 有速度优势，所以多数情况下建议使用 StringBuilder 类。然而在应用程序要求线程安全的情况下，则必须使用 StringBuffer 类。

## Java Math 类

Java 的 Math 包含了用于执行基本数学运算的属性和方法，如初等指数、对数、平方根和三角函数。

Math 的方法都被定义为 static 形式，通过 Math 类可以在主函数中直接调用。

|  |
| --- |
| **序号** |
| 1 | [xxxValue()](https://www.runoob.com/java/number-xxxvalue.html" \t "https://www.runoob.com/java/_blank) 将 Number 对象转换为xxx数据类型的值并返回。 |
| 3 | [equals()](https://www.runoob.com/java/number-equals.html" \t "https://www.runoob.com/java/_blank) 判断number对象是否与参数相等。 |
| 4 | [valueOf()](https://www.runoob.com/java/number-valueof.html" \t "https://www.runoob.com/java/_blank) 返回一个 Number 对象指定的内置数据类型 |
| 5 | [toString()](https://www.runoob.com/java/number-tostring.html" \t "https://www.runoob.com/java/_blank) 以字符串形式返回值。 |
| 6 | [parseInt()](https://www.runoob.com/java/number-parseInt.html" \t "https://www.runoob.com/java/_blank) 将字符串解析为int类型。 |
| 7 | [abs()](https://www.runoob.com/java/number-abs.html" \t "https://www.runoob.com/java/_blank) 返回参数的绝对值。 |
| 8 | [ceil()](https://www.runoob.com/java/number-ceil.html" \t "https://www.runoob.com/java/_blank) 返回大于等于( >= )给定参数的的最小整数，类型为双精度浮点型。 |
| 9 | [floor()](https://www.runoob.com/java/number-floor.html" \t "https://www.runoob.com/java/_blank) 返回小于等于（<=）给定参数的最大整数 。 |
| 10 | [rint()](https://www.runoob.com/java/number-rint.html" \t "https://www.runoob.com/java/_blank) 返回与参数最接近的整数。返回类型为double。 |
| 11 | [round()](https://www.runoob.com/java/number-round.html" \t "https://www.runoob.com/java/_blank) 它表示**四舍五入**，算法为 **Math.floor(x+0.5)**，即将原来的数字加上 0.5 后再向下取整，所以，Math.round(11.5) 的结果为12，Math.round(-11.5) 的结果为-11。 |
| 12 | [min()](https://www.runoob.com/java/number-min.html" \t "https://www.runoob.com/java/_blank) 返回两个参数中的最小值。 |
| 13 | [max()](https://www.runoob.com/java/number-max.html" \t "https://www.runoob.com/java/_blank) 返回两个参数中的最大值。 |
| 14 | [exp()](https://www.runoob.com/java/number-exp.html" \t "https://www.runoob.com/java/_blank) 返回自然数底数e的参数次方。 |
| 15 | [log()](https://www.runoob.com/java/number-log.html" \t "https://www.runoob.com/java/_blank) 返回参数的自然数底数的对数值。 |
| 16 | [pow()](https://www.runoob.com/java/number-pow.html" \t "https://www.runoob.com/java/_blank) 返回第一个参数的第二个参数次方。 |
| 17 | [sqrt()](https://www.runoob.com/java/number-sqrt.html" \t "https://www.runoob.com/java/_blank) 求参数的算术平方根。 |
| 18 | [sin()](https://www.runoob.com/java/number-sin.html" \t "https://www.runoob.com/java/_blank) 求指定double类型参数的正弦值。 |
| 19 | [cos()](https://www.runoob.com/java/number-cos.html" \t "https://www.runoob.com/java/_blank) 求指定double类型参数的余弦值。 |
| 20 | [tan()](https://www.runoob.com/java/number-tan.html" \t "https://www.runoob.com/java/_blank) 求指定double类型参数的正切值。 |
| 21 | [asin()](https://www.runoob.com/java/number-asin.html" \t "https://www.runoob.com/java/_blank) 求指定double类型参数的反正弦值。 |
| 22 | [acos()](https://www.runoob.com/java/number-acos.html" \t "https://www.runoob.com/java/_blank) 求指定double类型参数的反余弦值。 |
| 23 | [atan()](https://www.runoob.com/java/number-atan.html" \t "https://www.runoob.com/java/_blank) 求指定double类型参数的反正切值。 |
| 24 | [atan2()](https://www.runoob.com/java/number-atan2.html" \t "https://www.runoob.com/java/_blank) 将笛卡尔坐标转换为极坐标，并返回极坐标的角度值。 |
| 25 | [toDegrees()](https://www.runoob.com/java/number-todegrees.html" \t "https://www.runoob.com/java/_blank) 将参数转化为角度。 |
| 26 | [toRadians()](https://www.runoob.com/java/number-toradians.html" \t "https://www.runoob.com/java/_blank) 将角度转换为弧度。 |
| 27 | [random()](https://www.runoob.com/java/number-random.html" \t "https://www.runoob.com/java/_blank) 返回一个随机数。 |

# Java 数组

数组对于每一门编程语言来说都是重要的数据结构之一，当然不同语言对数组的实现及处理也不尽相同。

Java 语言中提供的数组是用来存储固定大小的同类型元素。

你可以声明一个数组变量，如 numbers[100] 来代替直接声明 100 个独立变量 number0，number1，....，number99。

本教程将为大家介绍 Java 数组的声明、创建和初始化，并给出其对应的代码。