# 第四章 函数、字符和字符串

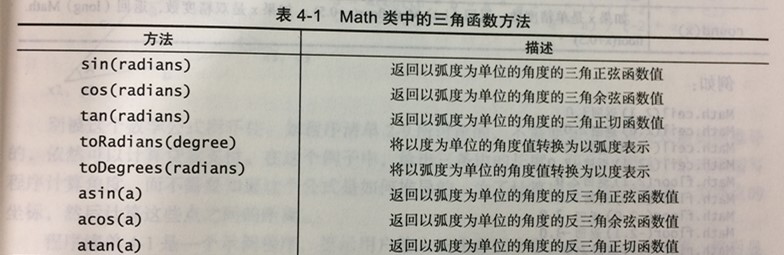
## 4.1引言

要点提示：重点介绍数学函数、字符和字符串对象，并使用它们来开发程序。

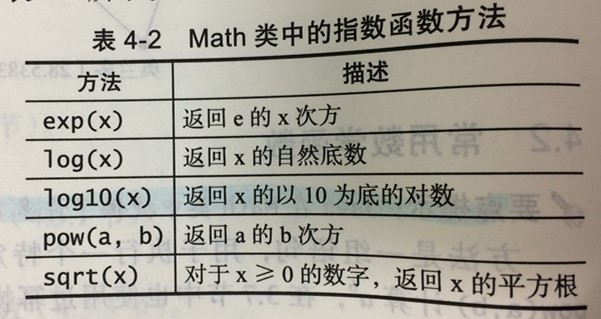
## 4.2常用的数学函数

主要有三角函数方法、指数函数方法、服务方法。服务方法包括：取整，求最小值，求最大值，求绝对值和随机方法。

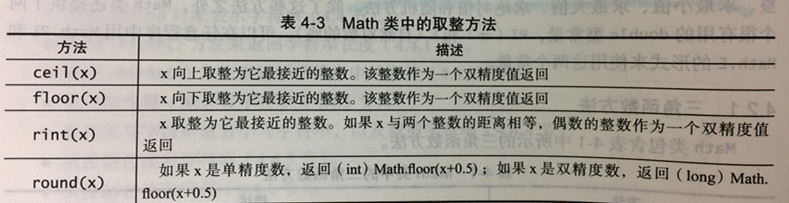
### 4.2.1三角函数方法



### 4.2.2指数函数方法



### 4.2.3取整方法



### 4.2.4 min、max和 abs方法

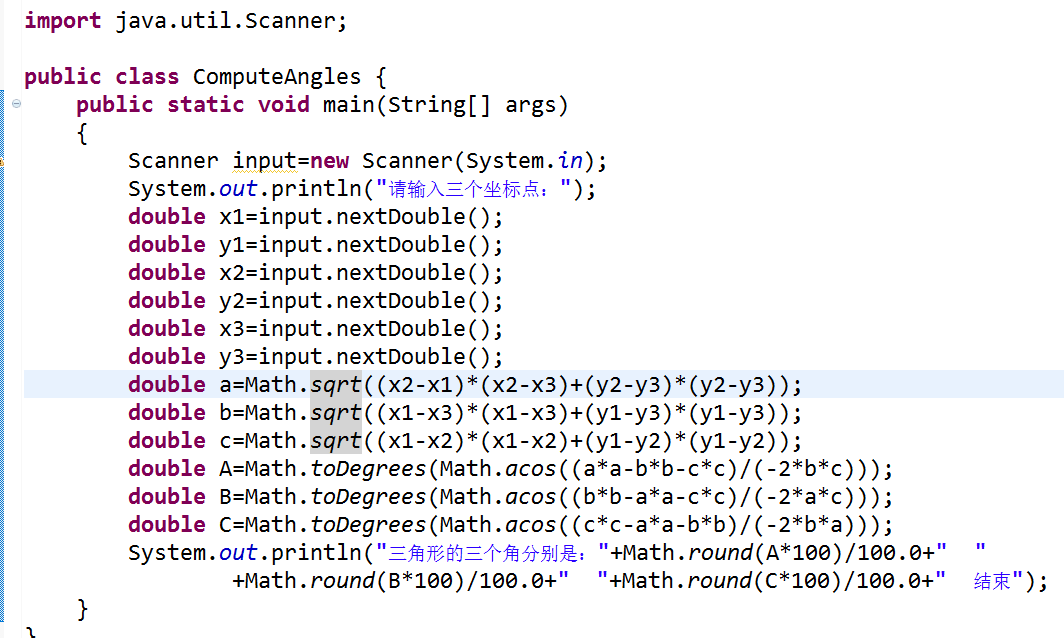
abs方法返回一个数（int、long、float或double型）的**绝对值**。

### 4.2.5 random方法

生成大于等于0.0且小于1.0的double类型随机数（0.0<=Math.random()<1.0），可以使用它编写简单的表达式，生成任意范围的随机数。

### 4.2.6 示例学习：计算三角形的度数

示例程序：提示用户输入三角形三个顶点的x和y坐标值，然后显示三个角。



## 4.3字符数据类型和操作

1.字符类型用单引号括住，例如 char letter =’A’;

字符串类型用双引号括住，例如 char soul=”A”;

在这里，第二个A表示字符串，第一个A则表示字符。

### 4.3.1Unicode和ASCII码

### 4.3.2特殊字符的转义序列

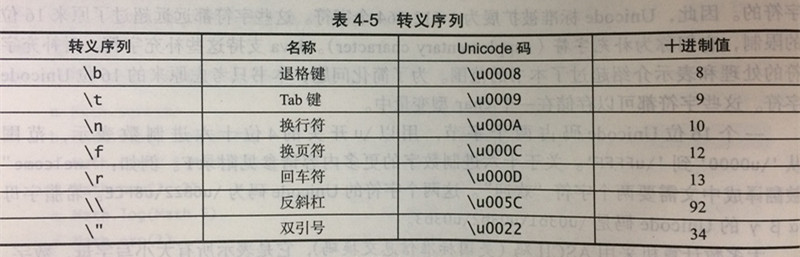
1.：System.out.println(“he said “java is fun””);

答案是不能，这条语句有语法错误。编译器会认为第二个引号字符就是这个字符串的结束标志，而不知道如何处理剩余的字符。

可以以下方式输出带引号的消息：

System.out.println(“He said \”java is fun\” “);

2.反斜杠\被称为转义字符。



4.3.3 字符型数据与数值型数据之间的转换

4.3.4 字符的比较与测试

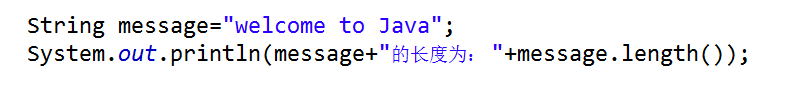
## 4.4 String类型

1.要点提示：字符串是一个字符序列。

2.String 对象的简单方法

|  |  |
| --- | --- |
| 方法 | 描述 |
| length() | 返回字符串中的字符数 |
| charAt(index) | 返回字符串s中指定位置的字符 |
| Concat(s1) | 将本字符串和字符串s1相连，返回一个新字符串 |
| toUpperCase() | 返回一个新字符串，其中所有的字母大写 |
| toLowerCase() | 返回一个新字符串，其中所有的字母小写 |
| trim() | 返回一个新字符串，去掉两边的空白字符 |

### 4.4.1求字符串长度



为方便起见，Java允许在不创建新变量的情况下，使用字符串直接引用字符串。

例如：它返回15。

“”表示空字符串，并且“”.length()为0。

### 4.4.2从字符串中获取字符

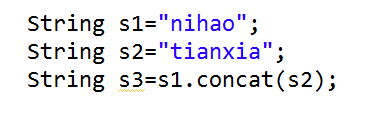
1.方法s.charAt(index)可以用于提取字符串s中的某个特定字符，其中下表index的取值范围在0-s.length()-1之间。

2.警告：在字符串s中越界访问是一种常见的程序设计错误。为了避免此类错误，要确保使用的下标不会超过s.length()-1。

### 4.4.3连接字符串

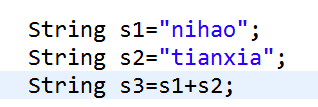
1.可以使用concat方法连接两个字符串。

例如：如下的语句将是s1和s2连接成s3:



因为字符串连接在程序设计中应用十分广泛，所以Java提供了一种实现字符串连接的简便方法。可以使用加号（+）连接两个或者多个字符串。

因此，以上语句可以改写为：



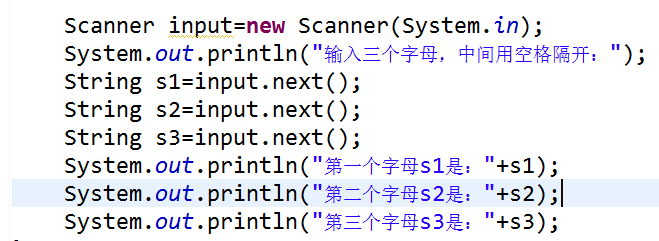
### 4.4.4字符串的转换

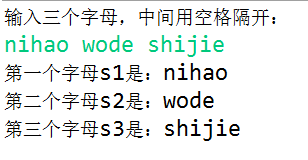
1.方法返回一个新字符串，其中所有字母小写；方法返回一个新字符串，其中所有字母大写。

2.方法通过删除字符串两端的空白字符返回一个新的字符串。

### 4.4.5从控制台读取字符串

1.next()方法读取以空白字符结束的字符串





2.nextLine()方法读取一整行文字

重要警告：为了避免输入错误，不要在nextByte()、nextShort()、nextInt()、nextLong()、nextFloat()、nextDouble()和next()之后使用nextLine()。

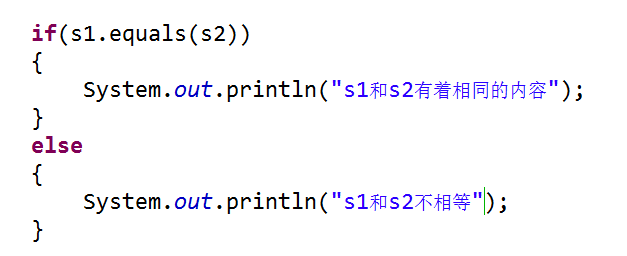
### 4.4.6从控制台读取字符

### 4.4.7字符串比较

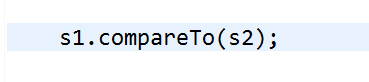
String对象的比较方法

|  |  |
| --- | --- |
| 方法 | 描述 |
| equals(s1) | 如果该字符串等于字符串s1,返回true |
| equalsIgnoreCase(s1) | 如果该字符串等于字符串s1,返回true，不区分大小写 |
| compareTo(s1) | 返回一个大于0，等于0，小于0的整数。表明一个字符串是否大于，等于或者小于s1 |
| compareToIgnoreCase(s1) | 和compareTo一样，除了比较是区分大小写的之外 |
| startWith(prefix) | 如果字符串以特定的前缀开始，返回true |
| endWith(suffix) | 如果字符串以特定的后缀结束，返回true |
| contain(s1) | 如果s1是该字符串的字符串，返回true |

1. 操作符==只能检测string1和string2是否指向同一个对象，但是无法判断其内容是否相等。可以用以下代码判断其是否相等：



也可以使用如下代码对两个字符串进行比较：



如果s1=s2，返回0。

如果s1<s2，返回<0。

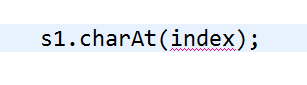
如果s1>s2，返回>0。

1. 示例程序



### 4.4.8获得子字符串

1.方法



可用于提取字符串s1中的某个特定字符。也可以使用String的类中的

方法从字符串中提取子串。

1. String类包含的获取子串的方法

|  |  |
| --- | --- |
| 方法 | 描述 |
| substring(beginIndex) | 返回该字符串的子串，从特定的beginIndex的字符开始到字符串的结尾 |
| substring(beginIndex,endIndex) | 返回该字符串的子串，从特定的位置beginIndex的字符开始到下标为endIndex-1的字符 |

### 4.4.9获取字符串中的字符或子串

1.String 类包含获取子串的方法

|  |  |
| --- | --- |
| 方法 | 描述 |
| indexOf(ch) | 返回字符串中出现第一个ch的下标。如果没有匹配，返回-1 |
| indexOf(ch，fromIndex) | 返回字符串fromIndex之后出现的第一个ch的下标。如果没有匹配，返回-1 |
| indexOf(s) | 返回字符串中出现第一个字符串s的下标。如果没有匹配则返回-1 |
| indexOf(s，fromIndex) | 返回字符串中from Index之后出现的第一个字符串s的下标。如果没有则返回-1 |
| lastIndexOf(ch) | 返回字符串中出现的最后一个ch的下标。如果没有，则返回-1 |
| lastIndexOf(ch，fromIndex) | 返回字符串中from Index之前出现的最后一个ch的下标。如果没有匹配，则返回-1 |
| lastIndexOf(s) | 返回字符串中出现最后一个字符串s的下标。如果没有匹配，则返回-1 |
| lastIndexOf(s，fromIndex) | 返回字符串中fromIndex之前出现的最后一个字符串s的下标。如果没有匹配，则返回-1 |

### 4.4.10字符串和数字间的转换

1.可以将数值型字符串转换为数值。



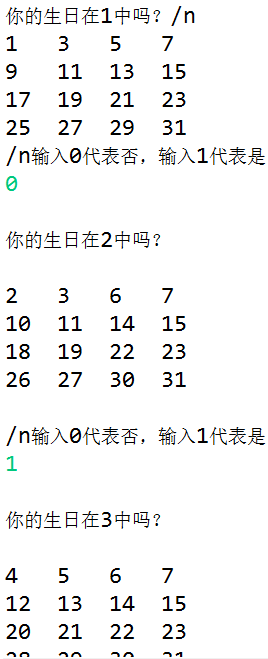
IntString是一个数值型字符串，例如：“123”

## 4.5示例学习

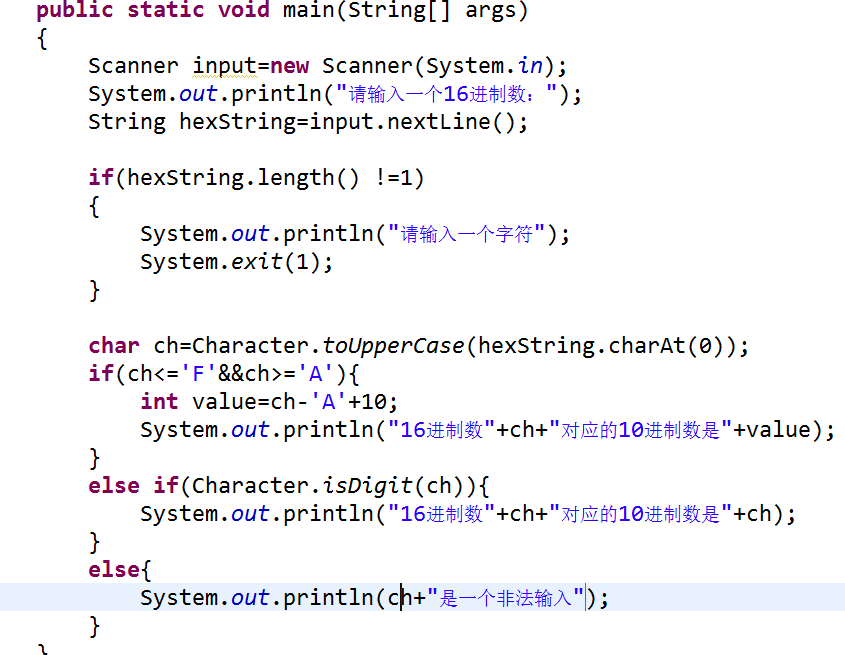
1.要点提示：字符串在编程中是非常基础的内容。使用字符串进行编程对学习Java编程非常关键。

### 4.5.1猜测生日

以下是猜测生日（月份中的天数）的一个程序，由于每一个数字阵列都是经过科学设计的，所以按照它来就行了（虽然有点马虎）



### 4.5.2将十六进制数转换成十进制数



### 4.5.3使用字符串修改彩票程序

修改之前的彩票程序，程序产生两个随机数字，并将它们连成一个字符串，这样的话，lottery包含两个随机数字。



## 4.6格式化控制台输出

