

004 数学函数、字符和字符串

作者: [pzs233](#)

原文链接: <https://ld246.com/article/1537246419906>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

```
<blockquote>
<p>本文为《Java 语言程序设计》第十版 章节笔记</p>
</blockquote>
<h2 id="4-1-常用的-Math-类方法-">4.1 常用的 Math 类方法: </h2>
<ul>
<li>三角函数方法 (trigonometric method)</li>
<li>指数函数方法 (exponent method)</li>
<li>服务方法 (service method)</li>
<li>还有两个常量: PI ( $\pi$ )、E (自然对数的底) </li>
</ul>
<hr>
<h3 id="4-1-1-三角函数方法-">4.1.1 三角函数方法: </h3>
<table>
<thead>
<tr>
<th>方法</th>
<th>描述</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>sin ( radians )</code></td>
<td>返回以弧度为单位的角度的三角正弦函数值</td>
</tr>
<tr>
<td><code>con ( radians )</code></td>
<td>返回以弧度为单位的角度的三角余弦函数值</td>
</tr>
<tr>
<td><code>tan ( radians )</code></td>
<td>返回以弧度为单位的角度的三角正切函数值</td>
</tr>
<tr>
<td><code>toRadians ( degree )</code></td>
<td>将以度为单位的角度值转换为以弧度表示</td>
</tr>
<tr>
<td><code>toDegrees ( radians )</code></td>
<td>将以弧度为单位的角度值转换为以角度表示</td>
</tr>
<tr>
<td><code>asin ( a )</code></td>
<td>返回以弧度为单位的角度的反三角正弦函数值</td>
</tr>
<tr>
<td><code>acos ( a )</code></td>
<td>返回以弧度为单位的角度的反三角余弦函数值</td>
</tr>
<tr>
<td><code>atan ( a )</code></td>
<td>返回以弧度为单位的角度的反三角正切函数值</td>
</tr>
</tbody>
</table>
```

<hr>
<h3 id="4-1-2-指数函数方法-">4.1.2 指数函数方法: </h3>

方法</th> <th>描述</th> </tr> </thead> <tbody> <tr> <td><code>exp(x)</code></td> <td>返回 e 的 x 次方</td> </tr> <tr> <td><code>log(x)</code></td> <td>返回 x 的自然底数</td> </tr> <tr> <td><code>log10(x)</code></td> <td>返回 x 的以 10 为底的对数</td> </tr> <tr> <td><code>pow(a, b)</code></td> <td>返回 a 的 b 次方</td> </tr> <tr> <td><code>sqrt(x)</code></td> <td>返回 a 的 b 次方</td> </tr> </tbody> </table>

<hr>
<h3 id="4-1-3-取整方法-">4.1.3 取整方法: </h3>

方法</th> <th>描述</th> </tr> </thead> <tbody> <tr> <td><code>ceil(x)</code></td> <td>x 向上取整为它最接近的整数 (return double 类型) </td> </tr> <tr> <td><code>floor(x)</code></td> <td>x 向下取整为它最接近的整数 (return double 类型) </td> </tr> <tr> <td><code>rint(x)</code></td> <td>x 取整为它最接近的整数。如果 x 与两个整数的距离相等, 偶数的整数作为一个双精度值返回。 </td> </tr> </tbody> </table>
--

```

</tr>
<tr>
<td><code>round( x )</code></td>
<td>如果 x 是单精度数, 返回 <code>( int )Math.floor( x + 0.5 )</code> ; 如果 x 是双精度数,
回 <code>( long )Math.floor( x + 0.5 )</code></td>
</tr>
</tbody>
</table>
<hr>
<h3 id="4-1-4-min-max-和-abs-方法-">4.1.4 min、max 和 abs 方法: </h3>
<ul>
<li>min 和 max 方法用于返回两个数 (int、long、float 或 double 型) 的最小值和最大值。</li>
<li>abs 方法以返回一个数 (int、long、float 或 double 型) 的绝对值。</li>
</ul>
<hr>
<h3 id="4-1-5-random-方法-">4.1.5 random 方法: </h3>
<p><code>Math.random()</code> 方法, 生成大于等于 0.0 且小于 1.0 的 double 型随机数 (0.0
&lt;= Math.random() &lt; 1.0)。可以用它编写简单的表达式, 生成任意范围的随机数。</p>
<hr>
<h2 id="4-2-字符型数据类型和操作">4.2 字符型数据类型和操作</h2>
<h3 id="4-2-1-Unicode-和-ASCII码">4.2.1 Unicode 和 ASCII 码</h3>
<p>一个 16 位的 Unicode 码占两个字节, 用以 \u 开头的 4 位十六进制数表示, 范围从 '\u0000
到 '\uffff'。</p>
<h4 id="常用字符的-ASCII-码-">常用字符的 ASCII 码: </h4>
<table>
<thead>
<tr>
<th>字符</th>
<th align="center">十进制编码值</th>
<th>Uniclde 值</th>
</tr>
</thead>
<tbody>
<tr>
<td>'0' ~ '9'</td>
<td align="center">48 ~ 57</td>
<td>\u0030 ~ \u0039</td>
</tr>
<tr>
<td>'A' ~ 'Z'</td>
<td align="center">65 ~ 90</td>
<td>\u0041 ~ \u005A</td>
</tr>
<tr>
<td>'a' ~ 'z'</td>
<td align="center">97 ~ 122</td>
<td>\u0061 ~ \u007A</td>
</tr>
</tbody>
</table>
<h4 id="特殊字符的转义序列-">特殊字符的转义序列: </h4>
<table>
<thead>
<tr>

```

```

<th align="center">转义序列</th>
<th>名称</th>
<th align="center">Unicode 码</th>
<th align="center">十进制值</th>
</tr>
</thead>
<tbody>
<tr>
<td align="center">\b</td>
<td>退格键</td>
<td align="center">\u0008</td>
<td align="center">8</td>
</tr>
<tr>
<td align="center">\t</td>
<td>Tab 键</td>
<td align="center">\u0009</td>
<td align="center">9</td>
</tr>
<tr>
<td align="center">\n</td>
<td>换行符</td>
<td align="center">\u000A</td>
<td align="center">10</td>
</tr>
<tr>
<td align="center">\f</td>
<td>换页符</td>
<td align="center">\u000C</td>
<td align="center">12</td>
</tr>
<tr>
<td align="center">\r</td>
<td>回车符</td>
<td align="center">\u000D</td>
<td align="center">13</td>
</tr>
<tr>
<td align="center">\\</td>
<td>反斜杠</td>
<td align="center">\u005C</td>
<td align="center">92</td>
</tr>
<tr>
<td align="center">\"</td>
<td>双引号</td>
<td align="center">\u0022</td>
<td align="center">34</td>
</tr>
</tbody>
</table>
<h4 id="Java-的-Character类-提供了下列方法用于进行字符测试-">Java 的 Character 类 提供了
列方法用于进行字符测试： </h4>
<table>

```

方法	描述
<code>isDigit(ch)</code>	如果指定的字符是一个数字, 返回 true
<code>isLetter(ch)</code>	如果指定的字符是一个字母, 返回 true
<code>isLetterOrDigit(ch)</code>	如果指定的字符是一个字母或数字, 返回 true
<code>isLowerCase(ch)</code>	如果指定的字符是一个小写字母, 返回 true
<code>isUpperCase(ch)</code>	如果指定的字符是一个大写字母, 返回 true
<code>toLowerCase(ch)</code>	返回指定的字符的小写形式
<code>toUpperCase(ch)</code>	返回指定的字符的大写形式
编写代码, 产生随机的小写字母:	

第一次想到的: `(char)((int)(Math.random() * (123-97) + 97))`

最简洁的: `(int)(Math.random() * 26 + 'a')`

4.3 String 类型

char 类型只能表示一个字符。为了表示一串字符, 使用称为 String (字符串) 的数据类型。

String 对象的简单方法:

方法	描述
----	----

<code>length()</code>	返回字符串中的字符数
<code>charAt(index)</code>	返回字符串中指定位置的字符
<code>concat(s1)</code>	将本字符串和字符串 s1 连接，返回一个新的字符串
<code>toUpperCase()</code>	返回一个新的字符串，其中所有的字母都大写
<code>toLowerCase()</code>	返回一个新的字符串，其中所有的字母都小写
<code>trim()</code>	返回一个新的字符串，去掉了两边的空白字符

字符 ' ', \t, \f, \r, \n 被称为空白字符。

调用一个实例方法的语法是：`reference-Variable.methodName(arguments)`

调用一个静态方法的语法是：`ClassName.methodName(arguments)`

Scanner 对象上的 next()方法读取以空白字符结束的字符串（即' ', \t, \f, \r, \n）。可以使用 extLine() 方法读取一整行文本。nextLine() 方法读取以按下回车键为结束标志的字符串。

String类提供了以下方法-用于比较两个字符串-

方法	描述
<code>equals(s1)</code>	如果该字符串等于字符串 s1，返回 true
<code>equalsIgnoreCase(s1)</code>	如果该字符串等于 s1，返回 true，不区分大小写
<code>compareTo(s1)</code>	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、等于或小于 s1

</tr>
<tr>
<td> <code>compareToIgnoreCase(s1)</code> </td>
<td>和 compareTo 一样，但是比较时不区分大小写</td>
</tr>
<tr>
<td> <code>startsWith(prefix)</code> </td>
<td>如果字符串以特定的前缀开始，返回 true</td>
</tr>
<tr>
<td> <code>endsWith(suffix)</code> </td>
<td>如果字符串以特定的后缀结束，返回 true</td>
</tr>
<tr>
<td> <code>contains(s1)</code> </td>
<td>如果 s1 是该字符串的子字符串，返回 true</td>
</tr>
</tbody>
</table>

<p>操作符 == 只能检测 string1 和 string2 是否指向同一个对象，但它不会告诉你它们的内容是否同。</p>

<h4 id="String-类获取字符串的字符或者子串-">String 类获取字符串的字符或者子串: </h4>

<th>方法</th>	<th>描述</th>
<td> <code>substring(beginIndex)</code> </td>	<td>返回该字符串的子串，从特定位置 beginIndex 的字符开始到字符串的结尾</td>
<td> <code>substring(beginIndex, endIndex)</code> </td>	<td>返回该字符串的子串，从特定位置 beginIndex 的字符开始但下标为 endIndex-1 的字符。注，位于 endIndex 位置的字符不属于该子字符串</td>
<td> <code>indexOf(ch)</code> </td>	<td>返回字符串中出现的第一个 ch 的下标。如果没有匹配，返回 -1</td>
<td> <code>indexOf(ch, fromIndex)</code> </td>	<td>返回字符串中 fromIndex 之后的第一个 ch 的下标。如果没有匹配的，返回 -1</td>
<td> <code>indexOf(s)</code> </td>	<td>返回字符串中出现的第一个字符串 s 的下标。如果没有匹配的，返回 -1</td>
<td> <code>indexOf(s, fromIndex)</code> </td>	<td>返回字符串中 fromIndex 之后出现的第一个字符串 s 的下标。如果没有匹配的，返回 -1</td>

<code>lastIndexOf(ch)</code>	返回字符串中出现的最后一个 ch 的下标。如果没有匹配的，返回 -1
<code>lastIndexOf(ch, fromIndex)</code>	返回字符串中 fromIndex 之前出现的最后一个 ch 的下标。如果没有匹配的，返回 -1
<code>lastIndexOf(s)</code>	返回字符串中出现的最后一个字符串 s 的下标。如果没有匹配的，返回 -1
<code>lastIndexOf(s, fromIndex)</code>	返回字符串中 fromIndex 之前出现的最后一个字符串 s 的下标。如果没有匹配的，返回 -1

数值型字符串转换为数值-

`int intValue = Integer.parseInt(intString);`
`double doubleValue = Double.parseDouble(doubleString);`

数值转换为字符串-

`String s = number + "";`

4.4 格式化控制台输出

可以使用 `System.out.printf()` 方法在控制台上显示格式化输出。

调用这个方法的格式是：
`System.out.printf(format, item1, item2, ..., itemk);`
`format` 是指一个由子串和格式标识符构成的字符串。格式标识符指定每个条目应如何显示。

常用的格式标识符：

标识符	输出	示例
<code>%b</code>	布尔值	true 或 false
<code>%c</code>	字符	'a'
<code>%d</code>		

```

<td align="center">十进制整数</td>
<td align="center">200</td>
</tr>
<tr>
<td align="center">%f</td>
<td align="center">浮点数</td>
<td align="center">45.54000</td>
</tr>
<tr>
<td align="center">%e</td>
<td align="center">标准科学计数法形式的数</td>
<td align="center">4.5500e+01</td>
</tr>
<tr>
<td align="center">%s</td>
<td align="center">字符串</td>
<td align="center">"Java is cool" </td>
</tr>
</tbody>
</table>

```

<p>e.g:</p>

```

<pre> <code class="language-java highlight-chroma"> <span class="highlight-line"> <span class="highlight-cl"> <span class="highlight-kt">int</span> <span class="highlight-n">count</span> /span> <span class="highlight-o">=</span> <span class="highlight-mi">5</span><span class="highlight-o">>;</span>
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> <span class="highlight-kt">double</span> <span class="highlight-n">amount</span> <span class="highlight-o">=</span> <span class="highlight-mf">45.56</span><span class="highlight-o">>;</span>
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> <span class="highlight-n">System</span> <span class="highlight-o">.</span><span class="highlight-na">out</span> <span class="highlight-o">.</span><span class="highlight-na">printf</span> <span class="highlight-o">(</span><span class="highlight-s">"count is %d and amount is %f"</span></span> <span class="highlight-o">,</span><span class="highlight-n">count</span><span class="highlight-o">,</span><span class="highlight-n">amount</span><span class="highlight-o">)</span>;</span>
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> <span class="highlight-c1">// Display count is 5 and amount is 45.560000</span>
</span> </span> </span></code></pre>

```

<p>条目与标识符必须在次序、数量和类型上匹配。默认情况下，浮点值显示小数点后 6 位数字。可在标识符中指定宽度和精度。</p>

<p>指定宽度和精度的例子:</p>

```

<table>
<thead>
<tr>
<th>举例</th>
<th>输出</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>%5c</code></td>
<td>输出字符并在这个字符条目前加 4 个空格</td>
</tr>

```

```

<tr>
<td><code>%6b</code></td>
<td>输出布尔值，在 false 前加一个空格，在 true 前加两个空格</td>
</tr>
<tr>
<td><code>%5d</code></td>
<td>输出整数条目，宽度至少为 5。如果该条目的数字位数小于 5，就在数字前面加空格。如果该条目的位数大于 5，则自动增加宽度</td>
</tr>
<tr>
<td><code>%10.2f</code></td>
<td>输出的浮点条目宽度至少为 10，包括小数点和小数点后两位数字。这样，给小数点前分配了 7 数字。如果该条目小数点前的位数小于 7，就在数字前面加空格。如果该条目小数点前的位数大于 7 则自动增加宽度</td>
</tr>
<tr>
<td><code>%10.2e</code></td>
<td>输出的浮点条目的宽度至少为 10，包括小数点、小数点后两位数字和指数部分。如果按科学计法显示的数字位数小于 10，就给数字前加空格</td>
</tr>
<tr>
<td><code>%12s</code></td>
<td>输出的字符串宽度至少为 12 个字符。如果该字符串条目小于 12 个字符，就在该字符串前加空。如果该字符串的条目多于 12 个字符，则自动增加宽度</td>
</tr>
</tbody>
</table>
<p>默认情况下，输出是右对齐的。可以在格式标识符中放一个负号 (-)，表明该条目在特定区域输出是左对齐的。</p>

```