|  |
| --- |
| **字段详细信息** |

**CASE\_INSENSITIVE\_ORDER**

public static final [Comparator](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/util/Comparator.html)<[String](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/lang/String.html)> **CASE\_INSENSITIVE\_ORDER**

一个对 String 对象进行排序的 Comparator，作用与 compareToIgnoreCase 相同。此比较器是可序列化的。

注意，Comparator *不* 考虑语言环境，因此可能导致在某些语言环境中的排序效果不理想。java.text 包提供 *Collator* 完成与语言环境有关的排序。

**从以下版本开始：**

1.2

**另请参见：**

[Collator.compare(String, String)](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/text/Collator.html#compare(java.lang.String, java.lang.String))

|  |
| --- |
| **构造方法详细信息** |

**String**

public **String**()

初始化一个新创建的 String 对象，使其表示一个空字符序列。注意，由于 String 是不可变的，所以无需使用此构造方法。

**String**

public **String**([String](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/lang/String.html) original)

初始化一个新创建的 String 对象，使其表示一个与参数相同的字符序列；换句话说，新创建的字符串是该参数字符串的副本。由于 String 是不可变的，所以无需使用此构造方法，除非需要 original 的显式副本。

**参数：**

original - 一个 String。

**String**

public **String**(char[] value)

分配一个新的 String，使其表示字符数组参数中当前包含的字符序列。该字符数组的内容已被复制；后续对字符数组的修改不会影响新创建的字符串。

**参数：**

value - 字符串的初始值

**String**

public **String**(char[] value,

int offset,

int count)

分配一个新的 String，它包含取自字符数组参数一个子数组的字符。offset 参数是子数组第一个字符的索引，count 参数指定子数组的长度。该子数组的内容已被复制；后续对字符数组的修改不会影响新创建的字符串。

**参数：**

value - 作为字符源的数组。

offset - 初始偏移量。

count - 长度。

**抛出：**

[IndexOutOfBoundsException](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/lang/IndexOutOfBoundsException.html) - 如果 offset 和 count 参数索引字符超出 value 数组的范围。

**String**

public **String**(int[] codePoints,

int offset,

int count)

分配一个新的 String，它包含 Unicode 代码点数组参数一个子数组的字符。offset 参数是该子数组第一个代码点的索引，count 参数指定子数组的长度。将该子数组的内容转换为 char；后续对 int 数组的修改不会影响新创建的字符串。

**参数：**

codePoints - 作为 Unicode 代码点的源的数组。

offset - 初始偏移量。

count - 长度。

**抛出：**

[IllegalArgumentException](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/lang/IllegalArgumentException.html) - 如果在 codePoints 中发现任何无效的 Unicode 代码点

[IndexOutOfBoundsException](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/lang/IndexOutOfBoundsException.html) - 如果 offset 和 count 参数索引字符超出 codePoints 数组的范围。

**从以下版本开始：**

1.5

**String**

[@Deprecated](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/lang/Deprecated.html)

public **String**(byte[] ascii,

int hibyte,

int offset,

int count)

**已过时。** *该方法无法将字节正确地转换为字符。从 JDK 1.1 开始，完成该转换的首选方法是使用带有* [*Charset*](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/nio/charset/Charset.html)*、字符集名称，或使用平台默认字符集的 String 构造方法。*

分配一个新的 String，它是根据一个 8 位整数值数组的子数组构造的。

offset 参数是该子数组的第一个 byte 的索引，count 参数指定子数组的长度。

子数组中的每个 byte 都按照上述方法转换为 char。

**参数：**

ascii - 要转换为字符的 byte。

hibyte - 每个 16 位 Unicode 代码单元的前 8 位。

offset - 初始偏移量。

count - 长度。

**抛出：**

[IndexOutOfBoundsException](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/lang/IndexOutOfBoundsException.html) - 如果 offset 或 count 参数无效。

**另请参见：**

[String(byte[], int)](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/lang/String.html#String(byte[], int)), [String(byte[], int, int, java.lang.String)](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/lang/String.html#String(byte[], int, int, java.lang.String)), [String(byte[], int, int, java.nio.charset.Charset)](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/lang/String.html#String(byte[], int, int, java.nio.charset.Charset)), [String(byte[], int, int)](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/lang/String.html#String(byte[], int, int)), [String(byte[], java.lang.String)](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/lang/String.html#String(byte[], java.lang.String)), [String(byte[], java.nio.charset.Charset)](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/lang/String.html#String(byte[], java.nio.charset.Charset)), [String(byte[])](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/lang/String.html#String(byte[]))

**String**

[@Deprecated](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/lang/Deprecated.html)

public **String**(byte[] ascii,

int hibyte)

**已过时。** *该方法无法将字节正确地转换为字符。从 JDK 1.1 开始，完成该转换的首选方法是使用带有* [*Charset*](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/nio/charset/Charset.html)*、字符集名称，或使用平台默认字符集的 String 构造方法。*

分配一个新的 String，它包含根据一个 8 位整数值数组构造的字符。所得字符串中的每个字符 *c* 都是根据 byte 数组中的相应组件 *b* 构造的，如下所示：

***c*** == (char)(((hibyte & 0xff) << 8)

| (***b*** & 0xff))

**参数：**

ascii - 要转换为字符的 byte。

hibyte - 每个 16 位 Unicode 代码单元的前 8 位。

**String**

public **String**(byte[] bytes,

int offset,

int length,

[String](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/lang/String.html) charsetName)

throws [UnsupportedEncodingException](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/io/UnsupportedEncodingException.html)

通过使用指定的字符集解码指定的 byte 子数组，构造一个新的 String。新 String 的长度是一个字符集函数，因此可能不等于子数组的长度。

当给定 byte 在给定字符集中无效的情况下，此构造方法的行为没有指定。如果需要对解码过程进行更多控制，则应该使用 [CharsetDecoder](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/nio/charset/CharsetDecoder.html) 类。

**参数：**

bytes - 要解码为字符的 byte

offset - 要解码的第一个 byte 的索引

length - 要解码的 byte 数

charsetName - 受支持 [charset](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/nio/charset/Charset.html) 的名称

**抛出：**

[UnsupportedEncodingException](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/io/UnsupportedEncodingException.html) - 如果指定的字符集不受支持

[IndexOutOfBoundsException](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/lang/IndexOutOfBoundsException.html) - 如果 offset 和 length 参数索引字符超出 bytes 数组的范围

**从以下版本开始：**

JDK1.1

**String**

public **String**(byte[] bytes,

int offset,

int length,

[Charset](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/nio/charset/Charset.html) charset)

通过使用指定的 [charset](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/nio/charset/Charset.html) 解码指定的 byte 子数组，构造一个新的 String。新 String 的长度是字符集的函数，因此可能不等于子数组的长度。

此方法总是使用此字符集的默认替代字符串替代错误输入 (malformed-input) 和不可映射字符 (unmappable-character) 序列。如果需要对解码过程进行更多控制，则应该使用 [CharsetDecoder](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/nio/charset/CharsetDecoder.html) 类。

**参数：**

bytes - 要解码为字符的 byte

offset - 要解码的第一个 byte 的索引

length - 要解码的 byte 数

charset - 用来解码 bytes 的 [charset](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/nio/charset/Charset.html)

**抛出：**

[IndexOutOfBoundsException](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/lang/IndexOutOfBoundsException.html) - 如果 offset 和 length 参数索引字符超出 bytes 数组的边界

**从以下版本开始：**

1.6

**String**

public **String**(byte[] bytes,

[String](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/lang/String.html) charsetName)

throws [UnsupportedEncodingException](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/io/UnsupportedEncodingException.html)

通过使用指定的 [charset](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/nio/charset/Charset.html) 解码指定的 byte 数组，构造一个新的 String。新 String 的长度是字符集的函数，因此可能不等于 byte 数组的长度。

当给定 byte 在给定字符集中无效的情况下，此构造方法的行为没有指定。如果需要对解码过程进行更多控制，则应该使用 [CharsetDecoder](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/nio/charset/CharsetDecoder.html) 类。

**参数：**

bytes - 要解码为字符的 byte

charsetName - 受支持的 [charset](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/nio/charset/Charset.html) 的名称

**抛出：**

[UnsupportedEncodingException](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/io/UnsupportedEncodingException.html) - 如果指定字符集不受支持

**从以下版本开始：**

JDK1.1

**String**

public **String**(byte[] bytes,

[Charset](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/nio/charset/Charset.html) charset)

通过使用指定的 [charset](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/nio/charset/Charset.html) 解码指定的 byte 数组，构造一个新的 String。新 String 的长度是字符集的函数，因此可能不等于 byte 数组的长度。

此方法总是使用此字符集的默认替代字符串替代错误输入和不可映射字符序列。如果需要对解码过程进行更多控制，则应该使用 [CharsetDecoder](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/nio/charset/CharsetDecoder.html) 类。

**参数：**

bytes - 要解码为字符的 byte

charset - 要用来解码 bytes 的 [charset](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/nio/charset/Charset.html)

**从以下版本开始：**

1.6

**String**

public **String**(byte[] bytes,

int offset,

int length)

通过使用平台的默认字符集解码指定的 byte 子数组，构造一个新的 String。新 String 的长度是字符集的函数，因此可能不等于该子数组的长度。

当给定 byte 在给定字符集中无效的情况下，此构造方法的行为没有指定。如果需要对解码过程进行更多控制，则应该使用 [CharsetDecoder](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/nio/charset/CharsetDecoder.html) 类。

**参数：**

bytes - 要解码为字符的 byte

offset - 要解码的第一个 byte 的索引

length - 要解码的 byte 数

**抛出：**

[IndexOutOfBoundsException](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/lang/IndexOutOfBoundsException.html) - 如果 offset 和 length 参数索引字符超出 bytes 数组的范围

**从以下版本开始：**

JDK1.1

**String**

public **String**(byte[] bytes)

通过使用平台的默认字符集解码指定的 byte 数组，构造一个新的 String。新 String 的长度是字符集的函数，因此可能不等于 byte 数组的长度。

当给定 byte 在给定字符集中无效的情况下，此构造方法的行为没有指定。如果需要对解码过程进行更多控制，则应该使用 [CharsetDecoder](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/nio/charset/CharsetDecoder.html) 类。

**参数：**

bytes - 要解码为字符的 byte

**从以下版本开始：**

JDK1.1

**String**

public **String**([StringBuffer](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/lang/StringBuffer.html) buffer)

分配一个新的字符串，它包含字符串缓冲区参数中当前包含的字符序列。该字符串缓冲区的内容已被复制，后续对它的修改不会影响新创建的字符串。

**参数：**

buffer - 一个 StringBuffer

**String**

public **String**([StringBuilder](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/lang/StringBuilder.html) builder)

分配一个新的字符串，它包含字符串生成器参数中当前包含的字符序列。该字符串生成器的内容已被复制，后续对它的修改不会影响新创建的字符串。

提供此构造方法是为了简化到 StringBuilder 的迁移。通过 toString 方法从字符串生成器中获取字符串可能运行的更快，因此通常作为首选。

**参数：**

builder - 一个 StringBuilder

**从以下版本开始：**

1.5

|  |
| --- |
| **方法详细信息** |

**length**

public int **length**()

返回此字符串的长度。长度等于字符串中 [Unicode 代码单元](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/lang/Character.html#unicode)的数量。

**指定者：**

接口 [CharSequence](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/lang/CharSequence.html) 中的 [length](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/lang/CharSequence.html#length())

**返回：**

此对象表示的字符序列的长度。

**isEmpty**

public boolean **isEmpty**()

当且仅当 [length()](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/lang/String.html#length()) 为 0 时返回 true。

**返回：**

如果 [length()](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/lang/String.html#length()) 为 0，则返回 true；否则返回 false。

**从以下版本开始：**

1.6

**charAt**

public char **charAt**(int index)

返回指定索引处的 char 值。索引范围为从 0 到 length() - 1。序列的第一个 char 值位于索引 0 处，第二个位于索引 1 处，依此类推，这类似于数组索引。

如果索引指定的 char 值是[代理项](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/lang/Character.html#unicode)，则返回代理项值。

**指定者：**

接口 [CharSequence](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/lang/CharSequence.html) 中的 [charAt](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/lang/CharSequence.html#charAt(int))

**参数：**

index - char 值的索引。

**返回：**

此字符串指定索引处的 char 值。第一个 char 值位于索引 0 处。

**抛出：**

[IndexOutOfBoundsException](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/lang/IndexOutOfBoundsException.html) - 如果 index 参数为负或小于此字符串的长度。

**codePointAt**

public int **codePointAt**(int index)

返回指定索引处的字符（Unicode 代码点）。索引引用 char 值（Unicode 代码单元），其范围从 0 到 [length()](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/lang/String.html#length()) - 1。

如果给定索引指定的 char 值属于高代理项范围，则后续索引小于此 String 的长度；如果后续索引处的 char 值属于低代理项范围，则返回该代理项对相应的增补代码点。否则，返回给定索引处的 char 值。

**参数：**

index - char 值的索引

**返回：**

index 处字符的代码点值

**抛出：**

[IndexOutOfBoundsException](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/lang/IndexOutOfBoundsException.html) - 如果 index 参数为负或小于此字符串的长度。

**从以下版本开始：**

1.5

**codePointBefore**

public int **codePointBefore**(int index)

返回指定索引之前的字符（Unicode 代码点）。索引引用 char 值（Unicode 代码单元），其范围从 1 到 [length](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/lang/CharSequence.html#length())。

如果 (index - 1) 处的 char 值属于低代理项范围，则 (index - 2) 为非负；如果 (index - 2) 处的 char 值属于高低理项范围，则返回该代理项对的增补代码点值。如果 index - 1 处的 char 值是未配对的低（高）代理项，则返回代理项值。

**参数：**

index - 应返回的代码点之后的索引

**返回：**

给定索引前面的 Unicode 代码点。

**抛出：**

[IndexOutOfBoundsException](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/lang/IndexOutOfBoundsException.html) - 如果 index 参数小于 1 或大于此字符串的长度。

**从以下版本开始：**

1.5

**codePointCount**

public int **codePointCount**(int beginIndex,

int endIndex)

返回此 String 的指定文本范围中的 Unicode 代码点数。文本范围始于指定的 beginIndex，一直到索引 endIndex - 1 处的 char。因此，该文本范围的长度（用 char 表示）是 endIndex-beginIndex。该文本范围内每个未配对的代理项计为一个代码点。

**参数：**

beginIndex - 文本范围的第一个 char 的索引。

endIndex - 文本范围的最后一个 char 之后的索引。

**返回：**

指定文本范围中 Unicode 代码点的数量

**抛出：**

[IndexOutOfBoundsException](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/lang/IndexOutOfBoundsException.html) - 如果 beginIndex 为负，或 endIndex 大于此 String 的长度，或 beginIndex 大于 endIndex。

**从以下版本开始：**

1.5

**offsetByCodePoints**

public int **offsetByCodePoints**(int index,

int codePointOffset)

返回此 String 中从给定的 index 处偏移 codePointOffset 个代码点的索引。文本范围内由 index 和 codePointOffset 给定的未配对代理项各计为一个代码点。

**参数：**

index - 要偏移的索引

codePointOffset - 代码点中的偏移量

**返回：**

String 的索引

**抛出：**

[IndexOutOfBoundsException](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/lang/IndexOutOfBoundsException.html) - 如果 index 为负或大于此 String 的长度；或者 codePointOffset 为正，且以 index 开头子字符串的代码点比 codePointOffset 少；如果 codePointOffset 为负，且 index 前面子字符串的代码点比 codePointOffset 的绝对值少。

**从以下版本开始：**

1.5

**getChars**

public void **getChars**(int srcBegin,

int srcEnd,

char[] dst,

int dstBegin)

将字符从此字符串复制到目标字符数组。

要复制的第一个字符位于索引 srcBegin 处；要复制的最后一个字符位于索引 srcEnd-1 处（因此要复制的字符总数是 srcEnd-srcBegin）。要复制到 dst 子数组的字符从索引 dstBegin 处开始，并结束于索引：

dstbegin + (srcEnd-srcBegin) - 1

**参数：**

srcBegin - 字符串中要复制的第一个字符的索引。

srcEnd - 字符串中要复制的最后一个字符之后的索引。

dst - 目标数组。

dstBegin - 目标数组中的起始偏移量。

**抛出：**

[IndexOutOfBoundsException](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/lang/IndexOutOfBoundsException.html) - 如果下列任何一项为 true：

* srcBegin 为负。
* srcBegin 大于 srcEnd
* srcEnd 大于此字符串的长度
* dstBegin 为负
* dstBegin+(srcEnd-srcBegin) 大于 dst.length

**getBytes**

[@Deprecated](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/lang/Deprecated.html)

public void **getBytes**(int srcBegin,int srcEnd,byte[] dst,int dstBegin)

**已过时。** *该方法无法将字符正确转换为字节。从 JDK 1.1 起，完成该转换的首选方法是通过* [*getBytes()*](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/lang/String.html#getBytes()) *方法，该方法使用平台的默认字符集。*

将字符从此字符串复制到目标 byte 数组中。每个 byte 接收相应字符的 8 个低位。不复制每个字符的高位，它们不参与任何方式的转换。

要复制的第一个字符位于索引 srcBegin 处；要复制的最后一个字符位于索引 srcEnd-1 处。要复制的字符总数为 srcEnd-srcBegin。将转换为 byte 的字符复制到 dst 的子数组中，从索引 dstBegin 处开始，并结束于索引：

dstbegin + (srcEnd-srcBegin) - 1

**参数：**

srcBegin - 字符串中要复制的第一个字符的索引

srcEnd - 字符串中要复制的最后一个字符之后的索引

dst - 目标数组

dstBegin - 目标数组中的起始偏移量

**抛出：**

[IndexOutOfBoundsException](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/lang/IndexOutOfBoundsException.html) - 如果下列任何一项为 true：

* srcBegin 为负
* srcBegin 大于 srcEnd
* srcEnd 大于此 String 的长度
* dstBegin 为负
* dstBegin+(srcEnd-srcBegin) 大于 dst.length

**getBytes**

public byte[] **getBytes**([String](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/lang/String.html) charsetName)

throws [UnsupportedEncodingException](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/io/UnsupportedEncodingException.html)

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。

当此字符串不能使用给定的字符集编码时，此方法的行为没有指定。如果需要对编码过程进行更多控制，则应该使用 [CharsetEncoder](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/nio/charset/CharsetEncoder.html) 类。

**参数：**

charsetName - 受支持的 [charset](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/nio/charset/Charset.html) 名称

**返回：**

所得 byte 数组

**抛出：**

[UnsupportedEncodingException](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/io/UnsupportedEncodingException.html) - 如果指定的字符集不受支持

**从以下版本开始：**

JDK1.1

**getBytes**

public byte[] **getBytes**([Charset](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/nio/charset/Charset.html) charset)

使用给定的 [charset](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/nio/charset/Charset.html) 将此 String 编码到 byte 序列，并将结果存储到新的 byte 数组。

此方法总是使用此字符集的默认替代 byte 数组替代错误输入和不可映射字符序列。如果需要对编码过程进行更多控制，则应该使用 [CharsetEncoder](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/nio/charset/CharsetEncoder.html) 类。

**参数：**

charset - 用于编码 String 的 [Charset](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/nio/charset/Charset.html)

**返回：**

所得 byte 数组

**从以下版本开始：**

1.6

**getBytes**

public byte[] **getBytes**()

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。

当此字符串不能使用默认的字符集编码时，此方法的行为没有指定。如果需要对编码过程进行更多控制，则应该使用 [CharsetEncoder](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/nio/charset/CharsetEncoder.html) 类。

**返回：**

所得 byte 数组

**从以下版本开始：**

JDK1.1

**equals**

public boolean **equals**([Object](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/lang/Object.html) anObject)

将此字符串与指定的对象比较。当且仅当该参数不为 null，并且是与此对象表示相同字符序列的 String 对象时，结果才为 true。

**覆盖：**

类 [Object](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/lang/Object.html) 中的 [equals](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/lang/Object.html#equals(java.lang.Object))

**参数：**

anObject - 与此 String 进行比较的对象。

**返回：**

如果给定对象表示的 String 与此 String 相等，则返回 true；否则返回 false。

**另请参见：**

[compareTo(String)](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/lang/String.html#compareTo(java.lang.String)), [equalsIgnoreCase(String)](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/lang/String.html#equalsIgnoreCase(java.lang.String))

**contentEquals**

public boolean **contentEquals**([StringBuffer](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/lang/StringBuffer.html) sb)

将此字符串与指定的 StringBuffer 比较。当且仅当此 String 与指定 StringBuffer 表示相同的字符序列时，结果才为 true。

**参数：**

sb - 要与此 String 比较的 StringBuffer。

**返回：**

如果此 String 与指定 StringBuffer 表示相同的字符序列，则返回 true；否则返回 false。

**从以下版本开始：**

1.4

**contentEquals**

public boolean **contentEquals**([CharSequence](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/lang/CharSequence.html) cs)

将此字符串与指定的 CharSequence 比较。当且仅当此 String 与指定序列表示相同的 char 值序列时，结果才为 true。

**参数：**

cs - 要与此 String 比较的序列

**返回：**

如果此 String 与指定序列表示相同的 char 值序列，则返回 true；否则返回 false。

**从以下版本开始：**

1.5

**equalsIgnoreCase**

public boolean **equalsIgnoreCase**([String](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/lang/String.html) anotherString)

将此 String 与另一个 String 比较，不考虑大小写。如果两个字符串的长度相同，并且其中的相应字符都相等（忽略大小写），则认为这两个字符串是相等的。

在忽略大小写的情况下，如果下列至少一项为 true，则认为 c1 和 c2 这两个字符相同。

* 这两个字符相同（使用 == 运算符进行比较）。
* 对每个字符应用方法 [Character.toUpperCase(char)](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/lang/Character.html#toUpperCase(char)) 生成相同的结果。
* 对每个字符应用方法 [Character.toLowerCase(char)](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/lang/Character.html#toLowerCase(char)) 生成相同的结果。

**参数：**

anotherString - 与此 String 进行比较的 String。

**返回：**

如果参数不为 null，且这两个 String 相等（忽略大小写），则返回 true；否则返回 false。

**另请参见：**

[equals(Object)](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/lang/String.html#equals(java.lang.Object))

**compareTo**

public int **compareTo**([String](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/lang/String.html) anotherString)

按字典顺序比较两个字符串。该比较基于字符串中各个字符的 Unicode 值。按字典顺序将此 String 对象表示的字符序列与参数字符串所表示的字符序列进行比较。如果按字典顺序此 String 对象位于参数字符串之前，则比较结果为一个负整数。如果按字典顺序此 String 对象位于参数字符串之后，则比较结果为一个正整数。如果这两个字符串相等，则结果为 0；compareTo 只在方法 [equals(Object)](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/lang/String.html#equals(java.lang.Object)) 返回 true 时才返回 0。

这是字典排序的定义。如果这两个字符串不同，那么它们要么在某个索引处的字符不同（该索引对二者均为有效索引），要么长度不同，或者同时具备这两种情况。如果它们在一个或多个索引位置上的字符不同，假设 *k* 是这类索引的最小值；则在位置 *k* 上具有较小值的那个字符串（使用 < 运算符确定），其字典顺序在其他字符串之前。在这种情况下，compareTo 返回这两个字符串在位置 k 处两个char 值的差，即值：

this.charAt(k)-anotherString.charAt(k)

如果没有字符不同的索引位置，则较短字符串的字典顺序在较长字符串之前。在这种情况下，compareTo 返回这两个字符串长度的差，即值：

this.length()-anotherString.length()

**指定者：**

接口 [Comparable](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/lang/Comparable.html)<[String](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/lang/String.html)> 中的 [compareTo](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/lang/Comparable.html#compareTo(T))

**参数：**

anotherString - 要比较的 String。

**返回：**

如果参数字符串等于此字符串，则返回值 0；如果此字符串按字典顺序小于字符串参数，则返回一个小于 0 的值；如果此字符串按字典顺序大于字符串参数，则返回一个大于 0 的值。

**compareToIgnoreCase**

public int **compareToIgnoreCase**([String](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/lang/String.html) str)

按字典顺序比较两个字符串，不考虑大小写。此方法返回一个整数，其符号与使用规范化的字符串调用 compareTo 所得符号相同，规范化字符串的大小写差异已通过对每个字符调用 Character.toLowerCase(Character.toUpperCase(character)) 消除。

注意，此方法*不* 考虑语言环境，因此可能导致在某些语言环境中的排序效果不理想。java.text 包提供 *Collators* 完成与语言环境有关的排序。

**参数：**

str - 要比较的 String。

**返回：**

根据指定 String 大于、等于还是小于此 String（不考虑大小写），分别返回一个负整数、0 或一个正整数。

**从以下版本开始：**

1.2

**另请参见：**

[Collator.compare(String, String)](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/text/Collator.html#compare(java.lang.String, java.lang.String))

**regionMatches**

public boolean **regionMatches**(int toffset,

[String](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/lang/String.html) other,

int ooffset,

int len)

测试两个字符串区域是否相等。

将此 String 对象的一个子字符串与参数 other 的一个子字符串进行比较。如果这两个子字符串表示相同的字符序列，则结果为 true。要比较的此 String 对象的子字符串从索引 toffset 处开始，长度为 len。要比较的 other 的子字符串从索引 ooffset 处开始，长度为 len。当且仅当下列至少一项为 true 时，结果才为 false ：

* toffset 为负。
* ooffset 为负。
* toffset+len 大于此 String 对象的长度。
* ooffset+len 大于另一个参数的长度。
* 存在某个小于 len 的非负整数 *k*，它满足：this.charAt(toffset+*k*) != other.charAt(ooffset+*k*)

**参数：**

toffset - 字符串中子区域的起始偏移量。

other - 字符串参数。

ooffset - 字符串参数中子区域的起始偏移量。

len - 要比较的字符数。

**返回：**

如果此字符串的指定子区域完全匹配字符串参数的指定子区域，则返回 true；否则返回 false。

**regionMatches**

public boolean **regionMatches**(boolean ignoreCase,

int toffset,

[String](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/lang/String.html) other,

int ooffset,

int len)

测试两个字符串区域是否相等。

将此 String 对象的子字符串与参数 other 的子字符串进行比较。如果这两个子字符串表示相同的字符序列，则结果为 true，当且仅当 ignoreCase 为 true 时忽略大小写。要比较的此 String 对象的子字符串从索引 toffset 处开始，长度为 len。要比较的 other 的子字符串从索引 ooffset 处开始，长度为 len。当且仅当下列至少一项为 true 时，结果才为 false：

* toffset 为负。
* ooffset 为负。
* toffset+len 大于此 String 对象的长度。
* ooffset+len 大于另一个参数的长度。
* ignoreCase 为 false，且存在某个小于 len 的非负整数 *k*，即：
* this.charAt(toffset+k) != other.charAt(ooffset+k)

* ignoreCase 为 true，且存在某个小于 len 的非负整数 *k*，即：
* Character.toLowerCase(this.charAt(toffset+k)) !=
* Character.toLowerCase(other.charAt(ooffset+k))

以及：

Character.toUpperCase(this.charAt(toffset+k)) !=

Character.toUpperCase(other.charAt(ooffset+k))

**参数：**

ignoreCase - 如果为 true，则比较字符时忽略大小写。

toffset - 此字符串中子区域的起始偏移量。

other - 字符串参数。

toffset - 字符串参数中子区域的起始偏移量。

len - 要比较的字符数。

**返回：**

如果此字符串的指定子区域匹配字符串参数的指定子区域，则返回 true；否则返回 false。是否完全匹配或考虑大小写取决于 ignoreCase 参数。

**startsWith**

public boolean **startsWith**([String](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/lang/String.html) prefix,

int toffset)

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。

**参数：**

prefix - 前缀。

toffset - 在此字符串中开始查找的位置。

**返回：**

如果参数表示的字符序列是此对象从索引 toffset 处开始的子字符串前缀，则返回 true；否则返回 false。如果 toffset 为负或大于此 String 对象的长度，则结果为 false；否则结果与以下表达式的结果相同：

this.substring(toffset).startsWith(prefix)

**startsWith**

public boolean **startsWith**([String](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/lang/String.html) prefix)

测试此字符串是否以指定的前缀开始。

**参数：**

prefix - 前缀。

**返回：**

如果参数表示的字符序列是此字符串表示的字符序列的前缀，则返回 true；否则返回 false。还要注意，如果参数是空字符串，或者等于此 String 对象（用 [equals(Object)](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/lang/String.html#equals(java.lang.Object)) 方法确定），则返回 true。

**从以下版本开始：**

1. 0

**endsWith**

public boolean **endsWith**([String](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/lang/String.html) suffix)

测试此字符串是否以指定的后缀结束。

**参数：**

suffix - 后缀。

**返回：**

如果参数表示的字符序列是此对象表示的字符序列的后缀，则返回 true；否则返回 false。注意，如果参数是空字符串，或者等于此 String 对象（用 [equals(Object)](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/lang/String.html#equals(java.lang.Object)) 方法确定），则结果为 true。

**hashCode**

public int **hashCode**()

返回此字符串的哈希码。String 对象的哈希码根据以下公式计算：

s[0]\*31^(n-1) + s[1]\*31^(n-2) + ... + s[n-1]

使用 int 算法，这里 s[i] 是字符串的第 *i* 个字符，n 是字符串的长度，^ 表示求幂。（空字符串的哈希值为 0。）

**覆盖：**

类 [Object](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/lang/Object.html) 中的 [hashCode](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/lang/Object.html#hashCode())

**返回：**

此对象的哈希码值。

**另请参见：**

[Object.equals(java.lang.Object)](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/lang/Object.html#equals(java.lang.Object)), [Hashtable](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/util/Hashtable.html)

**indexOf**

public int **indexOf**(int ch)

返回指定字符在此字符串中第一次出现处的索引。如果在此 String 对象表示的字符序列中出现值为 ch 的字符，则返回第一次出现该字符的索引（以 Unicode 代码单元表示）。对于 0 到 0xFFFF（包括 0 和 0xFFFF）范围内的 ch 的值，返回值是

this.charAt(*k*) == ch

为 true 的最小 *k* 值。对于其他 ch 值，返回值是

this.codePointAt(*k*) == ch

为 true 最小 *k* 值。无论哪种情况，如果此字符串中没有这样的字符，则返回 -1。

**参数：**

ch - 一个字符（Unicode 代码点）。

**返回：**

在此对象表示的字符序列中第一次出现该字符的索引；如果未出现该字符，则返回 -1。

**indexOf**

public int **indexOf**(int ch,

int fromIndex)

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。

在此 String 对象表示的字符序列中，如果带有值 ch 的字符的索引不小于 fromIndex，则返回第一次出现该值的索引。对于 0 到 0xFFFF（包括 0 和 0xFFFF）范围内的 ch 值，返回值是

(this.charAt(*k*) == ch) && (*k* >= fromIndex)

为 true 的最小 *k* 值。对于其他 ch 值，返回值是

(this.codePointAt(*k*) == ch) && (*k* >= fromIndex)

为 true 的最小 *k* 值。无论哪种情况，如果此字符串中 fromIndex 或之后的位置没有这样的字符出现，则返回 -1。

fromIndex 的值没有限制。如果它为负，则与它为 0 的效果同样：将搜索整个字符串。如果它大于此字符串的长度，则与它等于此字符串长度的效果相同：返回 -1。

所有索引都在 char 值中指定（Unicode 代码单元）。

**参数：**

ch - 一个字符（Unicode 代码点）。

fromIndex - 开始搜索的索引。

**返回：**

在此对象表示的字符序列中第一次出现的大于或等于 fromIndex 的字符的索引；如果未出现该字符，则返回 -1。

**lastIndexOf**

public int **lastIndexOf**(int ch)

返回指定字符在此字符串中最后一次出现处的索引。对于 0 到 0xFFFF（包括 0 和 0xFFFF）范围内的 ch 的值，返回的索引（Unicode 代码单元）是

this.charAt(*k*) == ch

为 true 最大 *k* 值。对于其他 ch 值，返回值是

this.codePointAt(*k*) == ch

为 true 的最大 *k* 值。无论哪种情况，如果此字符串中没有这样的字符出现，则返回 -1。从最后一个字符开始反向搜索此 String。

**参数：**

ch - 一个字符（Unicode 代码点）。

**返回：**

在此对象表示的字符序列中最后一次出现该字符的索引；如果未出现该字符，则返回 -1。

**lastIndexOf**

public int **lastIndexOf**(int ch,

int fromIndex)

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。对于 0 到 0xFFFF（包括 0 和 0xFFFF）范围内的 ch 值，返回的索引是

(this.charAt(*k*) == ch) && (*k* <= fromIndex)

为 true 的最大 *k* 值。对于 ch 的其他值，返回值是

(this.codePointAt(*k*) == ch) && (*k* <= fromIndex)

为 true 的最大 *k* 值。无论哪种情况，如果此字符串中 fromIndex 或之前的位置没有这样的字符出现，则返回 -1。

所有的索引都以 char 值指定（Unicode 代码单元）。

**参数：**

ch - 一个字符（Unicode 代码点）。

fromIndex - 开始搜索的索引。fromIndex 的值没有限制。如果它大于等于此字符串的长度，则与它小于此字符串长度减 1 的效果相同：将搜索整个字符串。如果它为负，则与它为 -1 的效果相同：返回 -1。

**返回：**

在此对象表示的字符序列（小于等于 fromIndex）中最后一次出现该字符的索引；如果在该点之前未出现该字符，则返回 -1。

**indexOf**

public int **indexOf**([String](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/lang/String.html) str)

返回指定子字符串在此字符串中第一次出现处的索引。返回的整数是

this.startsWith(str, *k*)

为 true 的最小 *k* 值。

**参数：**

str - 任意字符串。

**返回：**

如果字符串参数作为一个子字符串在此对象中出现，则返回第一个这种子字符串的第一个字符的索引；如果它不作为一个子字符串出现，则返回 -1。

**indexOf**

public int **indexOf**([String](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/lang/String.html) str,

int fromIndex)

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。返回的整数是满足下式的最小 k 值：

k >= Math.min(fromIndex, this.length()) && this.startsWith(str, k)

如果不存在这样的 *k* 值，则返回 -1。

**参数：**

str - 要搜索的子字符串。

fromIndex - 开始搜索的索引位置。

**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。

**lastIndexOf**

public int **lastIndexOf**([String](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/lang/String.html) str)

返回指定子字符串在此字符串中最右边出现处的索引。将最右边的空字符串 "" 视为出现在索引值 this.length() 处。返回的索引是

this.startsWith(str, k)

为 true 的最大 *k* 值。

**参数：**

str - 要搜索的子字符串。

**返回：**

如果字符串参数作为一个子字符串在此对象中出现一次或多次，则返回最后一个这种子字符串的第一个字符。如果它不作为一个子字符串出现，则返回 -1。

**lastIndexOf**

public int **lastIndexOf**([String](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/lang/String.html) str,

int fromIndex)

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。返回的整数是满足下式的最大 *k* 值：

k <= Math.min(fromIndex,this.length()) && this.startsWith(str, k)

如果不存在这样的 *k* 值，则返回 -1。

**参数：**

str - 要搜索的子字符串。

fromIndex - 开始搜索的索引位置。

**返回：**

指定子字符串在此字符串中最后一次出现处的索引。

**substring**

public [String](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/lang/String.html) **substring**(int beginIndex)

返回一个新的字符串，它是此字符串的一个子字符串。该子字符串从指定索引处的字符开始，直到此字符串末尾。

示例：

"unhappy".substring(2) returns "happy"

"Harbison".substring(3) returns "bison"

"emptiness".substring(9) returns "" (an empty string)

**参数：**

beginIndex - 起始索引（包括）。

**返回：**

指定的子字符串。

**抛出：**

[IndexOutOfBoundsException](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/lang/IndexOutOfBoundsException.html) - 如果 beginIndex 为负或大于此 String 对象的长度。

**substring**

public [String](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/lang/String.html) **substring**(int beginIndex,

int endIndex)

返回一个新字符串，它是此字符串的一个子字符串。该子字符串从指定的 beginIndex 处开始，直到索引 endIndex - 1 处的字符。因此，该子字符串的长度为 endIndex-beginIndex。

示例：

"hamburger".substring(4, 8) returns "urge"

"smiles".substring(1, 5) returns "mile"

**参数：**

beginIndex - 起始索引（包括）。

endIndex - 结束索引（不包括）。

**返回：**

指定的子字符串。

**抛出：**

[IndexOutOfBoundsException](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/lang/IndexOutOfBoundsException.html) - 如果 beginIndex 为负，或 endIndex 大于此 String 对象的长度，或 beginIndex 大于 endIndex。

**subSequence**

public [CharSequence](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/lang/CharSequence.html) **subSequence**(int beginIndex,

int endIndex)

返回一个新的字符序列，它是此序列的一个子序列。

此方法这种形式的调用：

str.subSequence(begin, end)

与以下调用的行为完全相同：

str.substring(begin, end)

定义此方法使 String 类能够实现 [CharSequence](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/lang/CharSequence.html) 接口。

**指定者：**

接口 [CharSequence](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/lang/CharSequence.html) 中的 [subSequence](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/lang/CharSequence.html#subSequence(int, int))

**参数：**

beginIndex - 起始索引（包括）。

endIndex - 结束索引（不包括）。

**返回：**

指定子序列。

**抛出：**

[IndexOutOfBoundsException](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/lang/IndexOutOfBoundsException.html) - 如果 beginIndex 或 endIndex 为负，如果 endIndex 大于 length() 或 beginIndex 大于 startIndex

**从以下版本开始：**

1.4

**concat**

public [String](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/lang/String.html) **concat**([String](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/lang/String.html) str)

将指定字符串连接到此字符串的结尾。

如果参数字符串的长度为 0，则返回此 String 对象。否则，创建一个新的 String 对象，用来表示由此 String 对象表示的字符序列和参数字符串表示的字符序列连接而成的字符序列。

示例：

"cares".concat("s") returns "caress"

"to".concat("get").concat("her") returns "together"

**参数：**

str - 连接到此 String 结尾的 String。

**返回：**

一个字符串，它表示在此对象字符后连接字符串参数字符而成的字符。

**replace**

public [String](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/lang/String.html) **replace**(char oldChar,

char newChar)

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。

如果 oldChar 在此 String 对象表示的字符序列中没有出现，则返回对此 String 对象的引用。否则，创建一个新的 String 对象，它所表示的字符序列除了所有的 oldChar 都被替换为 newChar 之外，与此 String 对象表示的字符序列相同。

示例：

"mesquite in your cellar".replace('e', 'o')

returns "mosquito in your collar"

"the war of baronets".replace('r', 'y')

returns "the way of bayonets"

"sparring with a purple porpoise".replace('p', 't')

returns "starring with a turtle tortoise"

"JonL".replace('q', 'x') returns "JonL" (no change)

**参数：**

oldChar - 原字符。

newChar - 新字符。

**返回：**

一个从此字符串派生的字符串，它将此字符串中的所有 oldChar 替代为 newChar。

**matches**

public boolean **matches**([String](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/lang/String.html) regex)

告知此字符串是否匹配给定的[正则表达式](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/util/regex/Pattern.html#sum)。

调用此方法的 *str*.matches(*regex*) 形式与以下表达式产生的结果完全相同：

[Pattern](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/util/regex/Pattern.html).[matches](mk:@MSITStore:F:\\java资料\\JDK_API_1_6_zh_CN.CHM::/java/util/regex/Pattern.html" \l "matches(java.lang.String, java.lang.CharSequence))(*regex*, *str*)

**参数：**

regex - 用来匹配此字符串的正则表达式

**返回：**

当且仅当此字符串匹配给定的正则表达式时，返回 true

**抛出：**

[PatternSyntaxException](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/util/regex/PatternSyntaxException.html) - 如果正则表达式的语法无效

**从以下版本开始：**

1.4

**另请参见：**

[Pattern](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/util/regex/Pattern.html)

**contains**

public boolean **contains**([CharSequence](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/lang/CharSequence.html) s)

当且仅当此字符串包含指定的 char 值序列时，返回 true。

**参数：**

s - 要搜索的序列

**返回：**

如果此字符串包含 s，则返回 true，否则返回 false

**抛出：**

[NullPointerException](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/lang/NullPointerException.html) - 如果 s 为 null

**从以下版本开始：**

1.5

**replaceFirst**

public [String](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/lang/String.html) **replaceFirst**([String](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/lang/String.html) regex,

[String](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/lang/String.html) replacement)

使用给定的 replacement 替换此字符串匹配给定的[正则表达式](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/util/regex/Pattern.html#sum)的第一个子字符串。

调用此方法的 *str*.replaceFirst(*regex*, *repl*) 形式与以下表达式产生的结果完全相同：

[Pattern](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/util/regex/Pattern.html).[compile](mk:@MSITStore:F:\\java资料\\JDK_API_1_6_zh_CN.CHM::/java/util/regex/Pattern.html" \l "compile(java.lang.String))(*regex*).[matcher](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/util/regex/Pattern.html#matcher(java.lang.CharSequence))(*str*).[replaceFirst](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/util/regex/Matcher.html#replaceFirst(java.lang.String))(*repl*)

注意，在替代字符串中使用反斜杠 (\) 和美元符号 ($) 与将其视为字面值替代字符串所得的结果可能不同；请参阅 [Matcher.replaceFirst(java.lang.String)](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/util/regex/Matcher.html#replaceFirst(java.lang.String))。如有需要，可使用 [Matcher.quoteReplacement(java.lang.String)](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/util/regex/Matcher.html#quoteReplacement(java.lang.String)) 取消这些字符的特殊含义。

**参数：**

regex - 用来匹配此字符串的正则表达式

replacement - 用来替换第一个匹配项的字符串

**返回：**

所得 String

**抛出：**

[PatternSyntaxException](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/util/regex/PatternSyntaxException.html) - 如果正则表达式的语法无效

**从以下版本开始：**

1.4

**另请参见：**

[Pattern](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/util/regex/Pattern.html)

**replaceAll**

public [String](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/lang/String.html) **replaceAll**([String](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/lang/String.html) regex,

[String](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/lang/String.html) replacement)

使用给定的 replacement 替换此字符串所有匹配给定的[正则表达式](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/util/regex/Pattern.html#sum)的子字符串。

调用此方法的 *str*.replaceAll(*regex*, *repl*) 形式与以下表达式产生的结果完全相同：

[Pattern](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/util/regex/Pattern.html).[compile](mk:@MSITStore:F:\\java资料\\JDK_API_1_6_zh_CN.CHM::/java/util/regex/Pattern.html" \l "compile(java.lang.String))(*regex*).[matcher](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/util/regex/Pattern.html#matcher(java.lang.CharSequence))(*str*).[replaceAll](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/util/regex/Matcher.html#replaceAll(java.lang.String))(*repl*)

注意，在替代字符串中使用反斜杠 (\) 和美元符号 ($) 与将其视为字面值替代字符串所得的结果可能不同；请参阅 [Matcher.replaceAll](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/util/regex/Matcher.html#replaceAll(java.lang.String))。如有需要，可使用 [Matcher.quoteReplacement(java.lang.String)](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/util/regex/Matcher.html#quoteReplacement(java.lang.String)) 取消这些字符的特殊含义。

**参数：**

regex - 用来匹配此字符串的正则表达式

replacement - 用来替换每个匹配项的字符串

**返回：**

所得 String

**抛出：**

[PatternSyntaxException](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/util/regex/PatternSyntaxException.html) - 如果正则表达式的语法无效

**从以下版本开始：**

1.4

**另请参见：**

[Pattern](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/util/regex/Pattern.html)

**replace**

public [String](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/lang/String.html) **replace**([CharSequence](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/lang/CharSequence.html) target,

[CharSequence](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/lang/CharSequence.html) replacement)

使用指定的字面值替换序列替换此字符串所有匹配字面值目标序列的子字符串。该替换从字符串的开头朝末尾执行，例如，用 "b" 替换字符串 "aaa" 中的 "aa" 将生成 "ba" 而不是 "ab"。

**参数：**

target - 要被替换的 char 值序列

replacement - char 值的替换序列

**返回：**

所得 String

**抛出：**

[NullPointerException](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/lang/NullPointerException.html) - 如果 target 或 replacement 为 null。

**从以下版本开始：**

1.5

**split**

public [String](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/lang/String.html)[] **split**([String](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/lang/String.html) regex,

int limit)

根据匹配给定的[正则表达式](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/util/regex/Pattern.html#sum)来拆分此字符串。

此方法返回的数组包含此字符串的子字符串，每个子字符串都由另一个匹配给定表达式的子字符串终止，或者由此字符串末尾终止。数组中的子字符串按它们在此字符串中出现的顺序排列。如果表达式不匹配输入的任何部分，那么所得数组只具有一个元素，即此字符串。

limit 参数控制模式应用的次数，因此影响所得数组的长度。如果该限制 *n* 大于 0，则模式将被最多应用 *n* - 1 次，数组的长度将不会大于 *n*，而且数组的最后一项将包含所有超出最后匹配的定界符的输入。如果 *n* 为非正，那么模式将被应用尽可能多的次数，而且数组可以是任何长度。如果 *n* 为 0，那么模式将被应用尽可能多的次数，数组可以是任何长度，并且结尾空字符串将被丢弃。

例如，字符串 "boo:and:foo" 使用这些参数可生成以下结果：

|  |  |  |
| --- | --- | --- |
| **Regex** | **Limit** | **结果** |
| : | 2 | { "boo", "and:foo" } |
| : | 5 | { "boo", "and", "foo" } |
| : | -2 | { "boo", "and", "foo" } |
| o | 5 | { "b", "", ":and:f", "", "" } |
| o | -2 | { "b", "", ":and:f", "", "" } |
| o | 0 | { "b", "", ":and:f" } |

调用此方法的 *str.*split(*regex*, *n*) 形式与以下表达式产生的结果完全相同：

[Pattern](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/util/regex/Pattern.html).[compile](mk:@MSITStore:F:\\java资料\\JDK_API_1_6_zh_CN.CHM::/java/util/regex/Pattern.html" \l "compile(java.lang.String))(*regex*).[split](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/util/regex/Pattern.html#split(java.lang.CharSequence, int))(*str*, *n*)

**参数：**

regex - 定界正则表达式

limit - 结果阈值，如上所述

**返回：**

字符串数组，它是根据给定正则表达式的匹配拆分此字符串确定的

**抛出：**

[PatternSyntaxException](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/util/regex/PatternSyntaxException.html) - 如果正则表达式的语法无效

**从以下版本开始：**

1.4

**另请参见：**

[Pattern](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/util/regex/Pattern.html)

**split**

public [String](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/lang/String.html)[] **split**([String](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/lang/String.html) regex)

根据给定[正则表达式](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/util/regex/Pattern.html#sum)的匹配拆分此字符串。

该方法的作用就像是使用给定的表达式和限制参数 0 来调用两参数 [split](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/lang/String.html#split(java.lang.String, int)) 方法。因此，所得数组中不包括结尾空字符串。

例如，字符串 "boo:and:foo" 使用这些表达式可生成以下结果：

|  |  |
| --- | --- |
| **Regex** | **结果** |
| : | { "boo", "and", "foo" } |
| o | { "b", "", ":and:f" } |

**参数：**

regex - 定界正则表达式

**返回：**

字符串数组，它是根据给定正则表达式的匹配拆分此字符串确定的

**抛出：**

[PatternSyntaxException](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/util/regex/PatternSyntaxException.html) - 如果正则表达式的语法无效

**从以下版本开始：**

1.4

**另请参见：**

[Pattern](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/util/regex/Pattern.html)

**toLowerCase**

public [String](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/lang/String.html) **toLowerCase**([Locale](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/util/Locale.html) locale)

使用给定 Locale 的规则将此 String 中的所有字符都转换为小写。大小写映射关系基于 [Character](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/lang/Character.html) 类指定的 Unicode 标准版。由于大小写映射关系并不总是 1:1 的字符映射关系，因此所得 String 的长度可能不同于原 String。

下表中给出了几个小写映射关系的示例：

|  |  |  |  |
| --- | --- | --- | --- |
| **语言环境的代码** | **大写字母** | **小写字母** | **描述** |
| tr (Turkish) | \u0130 | \u0069 | 大写字母 I，上面有点 -> 小写字母 i |
| tr (Turkish) | \u0049 | \u0131 | 大写字母 I -> 小写字母 i，无点 |
| (all) | French Fries | french fries | 将字符串中的所有字符都小写 |
| (all) |  |  | 将字符串中的所有字符都小写 |

**参数：**

locale - 使用此语言环境的大小写转换规则

**返回：**

要转换为小写的 String。

**从以下版本开始：**

1.1

**另请参见：**

[toLowerCase()](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/lang/String.html#toLowerCase()), [toUpperCase()](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/lang/String.html#toUpperCase()), [toUpperCase(Locale)](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/lang/String.html#toUpperCase(java.util.Locale))

**toLowerCase**

public [String](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/lang/String.html) **toLowerCase**()

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。这等效于调用 toLowerCase(Locale.getDefault())。

**注：** 此方法与语言环境有关，如果用于应独立于语言环境解释的字符串，则可能生成不可预料的结果。示例有编程语言标识符、协议键、HTML 标记。例如，"TITLE".toLowerCase() 在 Turkish（土耳其语）语言环境中返回 "t?tle"，其中“?”是 LATIN SMALL LETTER DOTLESS I 字符。对于与语言环境有关的字符，要获得正确的结果，请使用 toLowerCase(Locale.ENGLISH)。

**返回：**

要转换为小写的 String。

**另请参见：**

[toLowerCase(Locale)](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/lang/String.html#toLowerCase(java.util.Locale))

**toUpperCase**

public [String](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/lang/String.html) **toUpperCase**([Locale](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/util/Locale.html) locale)

使用给定 Locale 的规则将此 String 中的所有字符都转换为大写。大小写映射关系基于 [Character](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/lang/Character.html) 类指定的 Unicode 标准版。由于大小写映射关系并不总是 1:1 的字符映射关系，因此所得 String 的长度可能不同于原 String。

下表中给出了几个与语言环境有关和 1:M 大小写映射关系的一些示例。

|  |  |  |  |
| --- | --- | --- | --- |
| **语言环境的代码** | **小写** | **大写** | **描述** |
| tr (Turkish) | \u0069 | \u0130 | 小写字母 i -> 大写字母 I，上面有点 |
| tr (Turkish) | \u0131 | \u0049 | 小写字母 i，无点 -> 大写字母 I |
| (all) | \u00df | \u0053 \u0053 | 小写字母 sharp s -> 两个字母：SS |
| (all) | Fahrvergnügen | FAHRVERGNÜN |  |

**参数：**

locale - 使用此语言环境的大小写转换规则

**返回：**

要转换为大写的 String。

**从以下版本开始：**

1.1

**另请参见：**

[toUpperCase()](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/lang/String.html#toUpperCase()), [toLowerCase()](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/lang/String.html#toLowerCase()), [toLowerCase(Locale)](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/lang/String.html#toLowerCase(java.util.Locale))

**toUpperCase**

public [String](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/lang/String.html) **toUpperCase**()

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。此方法等效于 toUpperCase(Locale.getDefault())。

**注：** 此方法与语言环境有关，如果用于应独立于语言环境解释的字符串，则可能生成不可预料的结果。示例有编程语言标识符、协议键、HTML 标记。例如，"title".toUpperCase() 在 Turkish（土耳其语）语言环境中返回 "T?TLE"，其中“?”是 LATIN CAPITAL LETTER I WITH DOT ABOVE 字符。对于与语言环境有关的字符，要获得正确的结果，请使用 toUpperCase(Locale.ENGLISH)。

**返回：**

要转换为大写的 String。

**另请参见：**

[toUpperCase(Locale)](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/lang/String.html#toUpperCase(java.util.Locale))

**trim**

public [String](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/lang/String.html) **trim**()

返回字符串的副本，忽略前导空白和尾部空白。

如果此 String 对象表示一个空字符序列，或者此 String 对象表示的字符序列的第一个和最后一个字符的代码都大于 '\u0020'（空格字符），则返回对此 String 对象的引用。

否则，若字符串中没有代码大于 '\u0020' 的字符，则创建并返回一个表示空字符串的新 String 对象。

否则，假定 *k* 为字符串中代码大于 '\u0020' 的第一个字符的索引，*m* 为字符串中代码大于 '\u0020' 的最后一个字符的索引。创建一个新的 String 对象，它表示此字符串中从索引 *k* 处的字符开始，到索引 *m* 处的字符结束的子字符串，即 this.substring(*k*, *m*+1) 的结果。

此方法可用于截去字符串开头和末尾的空白（如上所述）。

**返回：**

此字符串移除了前导和尾部空白的副本；如果没有前导和尾部空白，则返回此字符串。

**toString**

public [String](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/lang/String.html) **toString**()

返回此对象本身（它已经是一个字符串！）。

**指定者：**

接口 [CharSequence](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/lang/CharSequence.html) 中的 [toString](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/lang/CharSequence.html#toString())

**覆盖：**

类 [Object](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/lang/Object.html) 中的 [toString](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/lang/Object.html#toString())

**返回：**

字符串本身。

**toCharArray**

public char[] **toCharArray**()

将此字符串转换为一个新的字符数组。

**返回：**

一个新分配的字符数组，它的长度是此字符串的长度，它的内容被初始化为包含此字符串表示的字符序列。

**format**

public static [String](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/lang/String.html) **format**([String](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/lang/String.html) format,

[Object](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/lang/Object.html)... args)

使用指定的格式字符串和参数返回一个格式化字符串。

始终使用 [Locale.getDefault()](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/util/Locale.html#getDefault()) 返回的语言环境。

**参数：**

format - [格式字符串](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/util/Formatter.html#syntax)

args - 格式字符串中由格式说明符引用的参数。如果还有格式说明符以外的参数，则忽略这些额外的参数。参数的数目是可变的，可以为 0。参数的最大数目受 [Java Virtual Machine Specification](http://java.sun.com/docs/books/vmspec/) 所定义的 Java 数组最大维度的限制。有关 null 参数的行为依赖于[转换](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/util/Formatter.html#syntax)。

**返回：**

一个格式化字符串

**抛出：**

IllegalFormatException - 如果格式字符串中包含非法语法、与给定的参数不兼容的格式说明符，格式字符串给定的参数不够，或者存在其他非法条件。有关所有可能的格式化错误的规范，请参阅 formatter 类规范的[详细信息](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/util/Formatter.html#detail) 一节。

[NullPointerException](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/lang/NullPointerException.html) - 如果 format 为 null

**从以下版本开始：**

1.5

**另请参见：**

[Formatter](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/util/Formatter.html)

**format**

public static [String](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/lang/String.html) **format**([Locale](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/util/Locale.html) l,

[String](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/lang/String.html) format,

[Object](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/lang/Object.html)... args)

使用指定的语言环境、格式字符串和参数返回一个格式化字符串。

**参数：**

l - 格式化过程中要应用的[语言环境](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/util/Locale.html)。如果 l 为 null，则不进行本地化。

format - [格式字符串](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/util/Formatter.html#syntax)

args - 格式字符串中由格式说明符引用的参数。如果还有格式说明符以外的参数，则忽略这些额外的参数。参数的数目是可变的，可以为 0。参数的最大数目受 [Java Virtual Machine Specification](http://java.sun.com/docs/books/vmspec/) 所定义的 Java 数组最大维度的限制。有关 null 参数的行为依赖于[转换](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/util/Formatter.html#syntax)。

**返回：**

一个格式化字符串

**抛出：**

IllegalFormatException - 如果格式字符串中包含非法语法、与给定参数不兼容的格式说明符，格式字符串给定的参数不够，或存在其他非法条件。有关所有可能的格式化错误的规范，请参阅 formatter 类规范的[详细信息](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/util/Formatter.html#detail) 一节。

[NullPointerException](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/lang/NullPointerException.html) - 如果 format 为 null

**从以下版本开始：**

1.5

**另请参见：**

[Formatter](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/util/Formatter.html)

**valueOf**

public static [String](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/lang/String.html) **valueOf**([Object](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/lang/Object.html) obj)

返回 Object 参数的字符串表示形式。

**参数：**

obj - 一个 Object。

**返回：**

如果参数为 null，则字符串等于 "null"；否则，返回 obj.toString() 的值。

**另请参见：**

[Object.toString()](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/lang/Object.html#toString())

**valueOf**

public static [String](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/lang/String.html) **valueOf**(char[] data)

返回 char 数组参数的字符串表示形式。字符数组的内容已被复制，后续修改不会影响新创建的字符串。

**参数：**

data - char 数组。

**返回：**

一个新分配的字符串，它表示包含在字符数组参数中的相同字符序列。

**valueOf**

public static [String](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/lang/String.html) **valueOf**(char[] data,

int offset,

int count)

返回 char 数组参数的特定子数组的字符串表示形式。

offset 参数是子数组的第一个字符的索引。count 参数指定子数组的长度。字符数组的内容已被复制，后续修改不会影响新创建的字符串。

**参数：**

data - 字符数组。

offset - String 值的初始偏移量。

count - String 值的长度。

**返回：**

一个字符串，它表示在字符数组参数的子数组中包含的字符序列。

**抛出：**

[IndexOutOfBoundsException](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/lang/IndexOutOfBoundsException.html) - 如果 offset 为负，count 为负，或者 offset+count 大于 data.length。

**copyValueOf**

public static [String](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/lang/String.html) **copyValueOf**(char[] data,

int offset,

int count)

返回指定数组中表示该字符序列的 String。

**参数：**

data - 字符数组。

offset - 子数组的初始偏移量。

count - 子数组的长度。

**返回：**

一个 String，它包含字符数组的指定子数组的字符。

**copyValueOf**

public static [String](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/lang/String.html) **copyValueOf**(char[] data)

返回指定数组中表示该字符序列的 String。

**参数：**

data - 字符数组。

**返回：**

一个 String，它包含字符数组的字符。

**valueOf**

public static [String](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/lang/String.html) **valueOf**(boolean b)

返回 boolean 参数的字符串表示形式。

**参数：**

b - 一个 boolean。

**返回：**

如果参数为 true，则返回一个等于 "true" 的字符串；否则，返回一个等于 "false" 的字符串。

**valueOf**

public static [String](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/lang/String.html) **valueOf**(char c)

返回 char 参数的字符串表示形式。

**参数：**

c - 一个 char。

**返回：**

一个长度为 1 的字符串，它包含参数 c 的单个字符。

**intern**

public [String](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/lang/String.html) **intern**()

返回字符串对象的规范化表示形式。

一个初始为空的字符串池，它由类 String 私有地维护。

当调用 intern 方法时，如果池已经包含一个等于此 String 对象的字符串（用 [equals(Object)](mk:@MSITStore:F:\java资料\JDK_API_1_6_zh_CN.CHM::/java/lang/String.html#equals(java.lang.Object)) 方法确定），则返回池中的字符串。否则，将此 String 对象添加到池中，并返回此 String 对象的引用。

它遵循以下规则：对于任意两个字符串 s 和 t，当且仅当 s.equals(t) 为 true 时，s.intern() == t.intern() 才为 true。

所有字面值字符串和字符串赋值常量表达式都使用 intern 方法进行操作。字符串字面值在 [Java Language Specification](http://java.sun.com/docs/books/jls/html/) 的 §3.10.5 定义。

**返回：**

一个字符串，内容与此字符串相同，但一定取自具有唯一字符串的池。