使用String类的length（）方法可获取声明的字符串对象的长度

例如：System.out.println(“字符串的长度为：”+s.length());

String 类提供了indexOf()和lastIndexOf()方法来获取指定字符的索引位置，

IndexOf()是第一个字符，lastindexOf()是倒数第一的字符；

使用String 类的charAt()方法可获取指定索引处的字符，返回字符的索引

例如：Srting s=“hello world”;

char mychar2=s.charAt(6);

System.out.printfln(“字符串s中索引位置是6的字符为：”+mychar2);

去除字符串的前导、后尾空格 trim

去除字符串中的所有空格 StingTokenizer()和replaceAll()

String Tokenizer(String str,String delim)

replaceAll() str.replaceAll(String regex,String replacement)

本质是用“”替换“ ”

字符串替换

字符串替换就是新字符串替换掉原字符串中指定位置的字符串，生成一个新字符串

可以通过replace()和replaceFirst()等方法可以实现。

replace()方法

该方法用于替换所有与指定字符串相匹配的字符串。语法格式如下：

str.replace(String regex,String replacemeng)

replaceFirst()方法

该方法用于替换第一个出现的指定字符串，而后面出现的指定字符串则不会被替换。语法格式如下：

Str.replaceFirst(String regex,String replacement)

判断字符串

判断字符串是否相等有equals()和equalsIgnoreCase()两种方法，而又有各自的规定，是用不好就会出错，所以一定要区分好什么时候用哪种方法。

1. equals()方法

使用equals()方法对字符串进行比较时是严格区分大小写的，在此条件下，如果两个字符串仍具有相同的字符和长度，则返回true，否则返回false 即返回值为boolean类型

1. equalsIgoreCase()方法

使用equalsIgnoreCase()方法是在忽略大小写的 情况下比较两个字符串是否相等，如果两个字符串具有相同的字符和长度，则返回true，否则返回false

判断字符串的开始与结尾

String类的startsWith()与endsWith()方法分别用于判断字符串是否以指定的内容开始和结尾，它们的返回值都为boolean类型

str.startsWith(String prefix)

str.endsWith(String suffix)

字母大小与转换

String类提供了字母大小写转换的方法，即toLowerCase()和toUpperCase()方法

1.toLowerCase()方法

该方法将字符串中的大写字母转换为小写字母，如果原来就是小写字母，则不改变，最后返回一个新的字符串，字符长度与原字符长度相同。语法格式如下：

str.toLowerCase()

2.toUpperCase()方法

该方法将字符串中的小写字母转换为大写字母，如果原来就是大写字母，则不改变，最后返回一个新的字符串，字符长度与原字符长度相同。语法格式如下：

str.toUpperCase()

字符串分割

split()方法根据指定的分割符对字符串进行分割，并将分割后的结果存放在字符串数组中，split()方法提供了两种重载形式。

1. split(String sign)方法

该方法根据指定的分割符对字符串进行完全分割，语法格式如下：

str.split(String sign)

1. split(String sign,int limit)方法

str.split(String sign,int limit)

格式化字符串

格式化方法

在应用程序设计中，经常需要显示时间和日期。format()方法通过给定的特殊转换符作为参数实现对日期和时间的格式化。format()方法有以下两种重载形式。

1. format（String format,Object…args）方法

str.format(String format,Object…args)

format:格式化转换符

args:格式化字符串中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。此参数的数目是可变的，可以为0。

返回值：一个格式化的字符串。

1. format(Local I,String format,Object…args)方法

该方法使用指定的格式字符串和参数，格式化的新字符串使用参数所规定的的语言环境，返回格式化后的字符串。

str.format(Local I,String format,Object…args)

I: 格式化过程中要应用的语言环境。如果1为null，则不进行本地化。

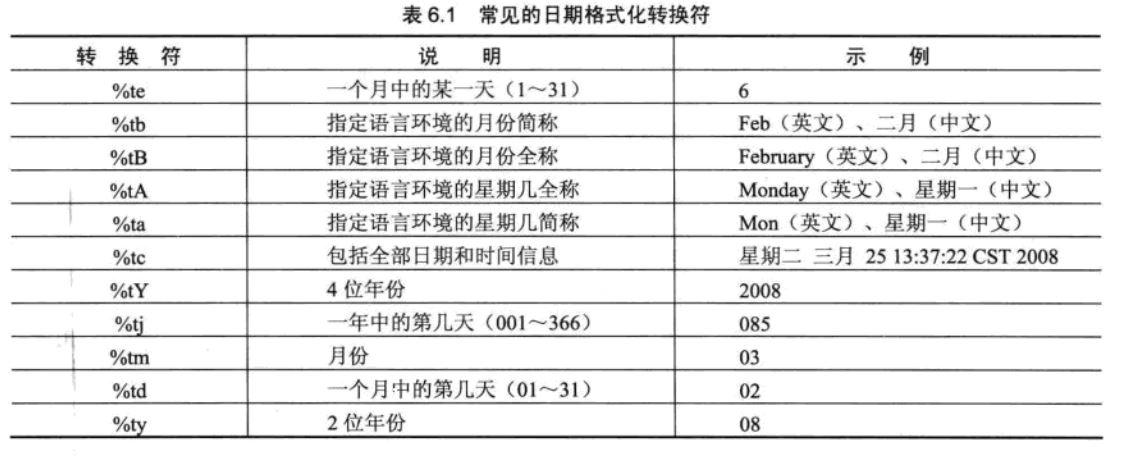
format: 格式化转化符。

args：格式化字符串中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。此参数的数目是可变的，可以为0。

返回值：一个格式化字符串。

日期格式化

使用format()方法对日期进行格式化时，会用到日期格式化转换符。



时间格式化

使用format()方法对时间进行格式化时，会用到时间格式化转换符。时间格式化转换符要比日期转换符更多，更精确，它可以将时间格式化为时、分、秒、毫秒。常用的时间格式化转换符为

日期时间组合格式化

因为日期与时间经常是同时出现的，所以格式化转换符还定义了各种日期和时间组合的格式。其中最常用的日期和时间的组合格式如表所示。



常规类型格式化

在程序设计过程中，经常需要对常规类型的数据进行格式化，如格式化为整数、格式化为科学计数表示等，在Java语言中可以使用常规类型的格式化转换符来实现。表列出了常规类型的格式化转换符。

正则表达式

判断时候符合正则表达式的方法

为了检查输入的数据是否满足某种格式，从JDK1.4开始可以使用String类的matches()方法进行判断。语法格式如下：

boolean matches(String regex)

regex:指定的正则表达式。

返回值：返回boolean类型

正则表达式的元字符

正则表达式是由一些含有特殊意义的字符组成的字符串，这些含有特殊意义的字符称为元字符。

正则表达式的限定符

在使用正则表达式时，如果需要某一类型的元字符多次输出，逐个输入就相当麻烦，这时可以使用正则表达式的限定元字符来重复次数。

例如

boolean bn2=s2.matches(“\\p{Upper}\\{Lower}\\p{Lower}\\p{Lower}\\d\\d\\d”);

可以写成

boolean bn=str.matches(“\\\p{Upper}\\p{Lower}{3}\\d{3}”);

方括号中元字符的含义

在正则表达式中还可以用方括号把多个字符括起来，方括号中各种正则表达式代表不同的含义。

字符串生成器

StringBuilder类

StringBuilder类即字符串生成器，新创建的StringBuilder对象初始容量是16个字符，可以自行指定初始长度，也可以动态地执行添加、删除和插入等字符串的编辑操作，大大提高了频繁增加字符串的效率。如果附加的字符超过可容纳的长度，则StringBuilder对象将自动增加长度以容纳被增加的字符。

StringBulider 类的构造方法有很多，主要是参数上的区别，这里主要介绍几种在编程中经常会用到的方法。

1. StringBuilder append(String str)方法

builder.append(String str);

builder:声明的字符串生成器。

str:追加到字符串生成器中的字符串。

返回值：返回字符串生成器的值，String型

1. StringBuilder append(StringBuffer sb)方法

该方法将参数sb指定的字符串缓存StringBuffer的值追加到字符串生成器中。

builder.append(StringBuffer sb);

builder:声明的字符串生成器。

sb：缓存中的值，追加到字符串生成器中。

返回值：返回将缓存的值追到字符串生成器后的值，String型。

1. StringBuilder insert(int offset, String str)方法

该方法将参数指定的字符串str添加到offset指定的位置。语法格式如下：

builder.insert(int offset, String str);

builder: 声明的字符串生成器

offset： 指定字符串添加的下标索引位置。

str：添加的字符串。

返回值： 返回添加字符串后的字符串生成器的值，String型

1. StringBuilder delete(int start, String end)方法

该方法移除字符串生成器中的子字符串，该子字符串从指定的start处开始，到指定的end处结束。

builder.delete(int start,String end);

builder: 声明的字符串生成器。

start: 删除子字符串的起始索引位置。

end：删除子字符串的结尾索引位置。

返回值：返回删除子字符串后的字符串生成器的值，String型

1. String toString()方法

该方法返回字符串生成器的字符串表示，即将字符串生成器转换为字符串，转换后字符串生成器的值不变。语法格式如下：

str.toString();

str:字符串生成器的值，转换字符串。

返回值：返回转换后的字符串。