



链滴

java 日期类

作者: [nerkeler](#)

原文链接: <https://ld246.com/article/1674011092416>

来源网站: [链滴](#)

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

Date

包路径: `package java.util`

```
public class Date
    implements java.io.Serializable, Cloneable, Comparable<Date>
```

常用构造方法

```
//源码中传入了系统当前时间
public Date() {
    this(System.currentTimeMillis());
}
```

```
//指定时间戳 Date(System.currentTimeMillis) 相当于Date()
public Date(long date) {
    fastTime = date;
}
```

已废弃构造方法 (Deprecated)

```
@Deprecated
public Date(int year, int month, int date) {
    this(year, month, date, 0, 0, 0);
}
```

```
@Deprecated
public Date(int year, int month, int date, int hrs, int min) {
    this(year, month, date, hrs, min, 0);
}
```

```
@Deprecated
public Date(int year, int month, int date, int hrs, int min, int sec)
```

常用方法

方法	描述
<code>boolean after(Date when)</code> 定日期之后	判断此日期是否在
<code>boolean before(Date when)</code> 指定日期之前	判断此日期是否
<code>int compareTo(Date anotherDate)</code> 期的顺序	比较两个
<code>boolean equals(Object obj)</code> 等性	比较两个日期的
<code>long getTime()</code> 00 GMT 以来, 此 Date 对象表示的毫秒数	返回自 1970 年 1 月 1 日 00:00

`String toString()` 把此 `Date` 对象转换为以下形式的 `String`: `dow mon dd hh:mm:ss zzz yyyy`。其中 `dow` 是一周中的某一天(`Sun`、`Mon`、`Tue`、`Wed`、`Thu`、`Fri` 及 `Sat`)

simpleDateFormat

```
public class SimpleDateFormat extends DateFormat
```

简介

`SimpleDateFormat` 是一个具体的类，用于以区域设置敏感的方式格式化和解析日期。它允许格式化日期文本，解析（文本日期）和归一化。

也就是将 `Date` 类型 转换为我们想要的以字符串形式输出日期格式。

日常使用

将字符串转化为日期

```
public void test1() throws ParseException {  
  
    String s = "2020-01-01";  
    SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd");  
    Date parse = sdf.parse(s);  
    System.out.println(parse);  
  
}
```

结果: `Wed Jan 01 00:00:00 CST 2020`

将日期格式化成字符串

```
@Test  
public void test2(){  
    Date date = new Date();  
    SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd hh:mm:ss");  
    String format = sdf.format(date);  
    System.out.println(format);  
  
}
```

结果: `2022-11-18 03:34:19`

时区设置

```
@Test  
public void test2(){  
    Date date = new Date();  
    SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd hh:mm:ss");
```

```
sdf.setTimeZone(TimeZone.getTimeZone("CTT")); //"CTT"等价于"Asia/Shanghai" x

String format = sdf.format(date);
System.out.println(format);

}
```

字符对应关系

字符	表示的意义	格式	
G	Era designator	Text	
D (公元)			
y	Year	Year	1992(yyyy)
92(yy)			
Y	Week year	Year	2009
09			
M	Month in year(context sensitive)		Month
uly;Jul;07			
L	Month in year(standalone form)		Month
uly;Jul;07			
w	Week in year	Number	
7			
W	Week in month	Number	2
D	Day in year	Number	
89			
d	Day in month	Number	
0			
F	Day of week in month	Number	
E	Day name in week	Text	
uesday; Tue			
u	Day number of week(1=Monday)	Number	
a	Am/pm marker	Number	
M			
H	Hour in day(0-23)	Number	0
k	Hour in day(1-24)	Number	
4			
K	Hour in am/pm(0-11)	Number	
h	Hour in am/pm(1-12)	Number	
2			
m	Minute in hour	Number	
0			
s	Second in minute	Number	

5		
S	Millisecond	Number
78		
z	Time zone	General time
acific Standard Time;PST;GMT-08:00		
Z	Time zone	RFC 822 time
0800		
X	Time zone	ISO 8601 time zone
08;-0800;-08:00		

- ****Y** : Y 表示 week in year 西方12月26日表示新年的第一天（圣诞节后一天）已经算是新的一年了
- ****

构造方法

SimpleDateFormat(String pattern) //构造时指定一个格式

SimpleDateFormat(String pattern, Locale locale) //构造一个 SimpleDateFormat使用给定的模式和给定的区域设置的默认日期格式符号。

SimpleDateFormat(String pattern, DateFormatSymbols formatSymbols) //使用给定的模式和日格式符号构造一个 SimpleDateFormat 。

常用方法

format //将给定的 **Date**成日期/时间字符串，并将结果附加到给定的 **String**
uffer 。

```
public StringBuffer format(Date date,
                           StringBuffer toAppendTo,
                           FieldPosition pos)
/**
参数
date - 要格式化为日期时间字符串的日期时间值。
toAppendTo - 要附加新的日期时间文本。
pos - 格式化位置。 在输入：对齐字段，如果需要。 输出：对齐字段的偏移量。
结果
格式化的日期时间字符串。
**/
```

parse // 从字符串中解析文本以产生一个**Date** 。

```
public Date parse(String text,
                  ParsePosition pos)
/**
参数
text - A String ， 其中一部分应解析。
pos - 具有 ParsePosition的索引和错误索引信息的 ParsePosition对象。
```

结果

一个Date从字符串中解析出来。万一出错，返回null。

**/

该方法尝试从pos给出的索引开始解析文本。如果解析成功，那么pos的索引将在使用最后一个字符后更新为索引（解析不一定使用字符串末尾的所有字符），并返回解析的日期。更新的pos可用于指下一次调用此方法的起始点。如果发生错误，则pos的索引不改变，pos的错误索引设置为发生错误的pos的索引，返回null。

这个解析操作使用calendar来生成一个Date。在解析之前，所有calendar的日期时间字段为cleared，日期时间字段的默认值为calendar用于任何缺少的日期时间信息。例如，被分析的年份值Date是170年GregorianCalendar如果没有年份值从解析操作说明。所述TimeZone值可以根据给定的模式在时区值被覆盖，text。任何TimeZone先前已通过调用设定值setTimeZone可能需要恢复的进一步行动。