java.text

**Class SimpleDateFormat**

* [java.lang.Object](https://docs.oracle.com/javase/7/docs/api/java/lang/Object.html)
  + [java.text.Format](https://docs.oracle.com/javase/7/docs/api/java/text/Format.html)
    - [java.text.DateFormat](https://docs.oracle.com/javase/7/docs/api/java/text/DateFormat.html)
      * java.text.SimpleDateFormat
* **All Implemented Interfaces:**

[Serializable](https://docs.oracle.com/javase/7/docs/api/java/io/Serializable.html), [Cloneable](https://docs.oracle.com/javase/7/docs/api/java/lang/Cloneable.html" \o "interface in java.lang)

public class **SimpleDateFormat**

extends [DateFormat](https://docs.oracle.com/javase/7/docs/api/java/text/DateFormat.html)

SimpleDateFormat is a concrete class for formatting and parsing dates in a locale-sensitive manner. It allows for formatting (date -> text), parsing (text -> date), and normalization.

SimpleDateFormat allows you to start by choosing any user-defined patterns for date-time formatting. However, you are encouraged to create a date-time formatter with eithergetTimeInstance, getDateInstance, or getDateTimeInstance in DateFormat. Each of these class methods can return a date/time formatter initialized with a default format pattern. You may modify the format pattern using the applyPattern methods as desired. For more information on using these methods, see [DateFormat](https://docs.oracle.com/javase/7/docs/api/java/text/DateFormat.html" \o "class in java.text).

**Date and Time Patterns**

Date and time formats are specified by *date and time pattern* strings. Within date and time pattern strings, unquoted letters from 'A' to 'Z' and from 'a' to 'z' are interpreted as pattern letters representing the components of a date or time string. Text can be quoted using single quotes (') to avoid interpretation. "''" represents a single quote. All other characters are not interpreted; they're simply copied into the output string during formatting or matched against the input string during parsing.

The following pattern letters are defined (all other characters from 'A' to 'Z' and from 'a' to 'z' are reserved):

|  |  |  |  |
| --- | --- | --- | --- |
| **Letter** | **Date or Time Component** | **Presentation** | **Examples** |
| G | Era designator | [Text](https://docs.oracle.com/javase/7/docs/api/java/text/SimpleDateFormat.html#text) | AD |
| y | Year | [Year](https://docs.oracle.com/javase/7/docs/api/java/text/SimpleDateFormat.html#year) | 1996; 96 |
| Y | Week year | [Year](https://docs.oracle.com/javase/7/docs/api/java/text/SimpleDateFormat.html#year) | 2009; 09 |
| M | Month in year | [Month](https://docs.oracle.com/javase/7/docs/api/java/text/SimpleDateFormat.html#month) | July; Jul; 07 |
| w | Week in year | [Number](https://docs.oracle.com/javase/7/docs/api/java/text/SimpleDateFormat.html#number) | 27 |
| W | Week in month | [Number](https://docs.oracle.com/javase/7/docs/api/java/text/SimpleDateFormat.html#number) | 2 |
| D | Day in year | [Number](https://docs.oracle.com/javase/7/docs/api/java/text/SimpleDateFormat.html#number) | 189 |
| d | Day in month | [Number](https://docs.oracle.com/javase/7/docs/api/java/text/SimpleDateFormat.html#number) | 10 |
| F | Day of week in month | [Number](https://docs.oracle.com/javase/7/docs/api/java/text/SimpleDateFormat.html" \l "number) | 2 |
| E | Day name in week | [Text](https://docs.oracle.com/javase/7/docs/api/java/text/SimpleDateFormat.html#text) | Tuesday; Tue |
| u | Day number of week (1 = Monday, ..., 7 = Sunday) | [Number](https://docs.oracle.com/javase/7/docs/api/java/text/SimpleDateFormat.html" \l "number) | 1 |
| a | Am/pm marker | [Text](https://docs.oracle.com/javase/7/docs/api/java/text/SimpleDateFormat.html#text) | PM |
| H | Hour in day (0-23) | [Number](https://docs.oracle.com/javase/7/docs/api/java/text/SimpleDateFormat.html#number) | 0 |
| k | Hour in day (1-24) | [Number](https://docs.oracle.com/javase/7/docs/api/java/text/SimpleDateFormat.html#number) | 24 |
| K | Hour in am/pm (0-11) | [Number](https://docs.oracle.com/javase/7/docs/api/java/text/SimpleDateFormat.html#number) | 0 |
| h | Hour in am/pm (1-12) | [Number](https://docs.oracle.com/javase/7/docs/api/java/text/SimpleDateFormat.html#number) | 12 |
| m | Minute in hour | [Number](https://docs.oracle.com/javase/7/docs/api/java/text/SimpleDateFormat.html#number) | 30 |
| s | Second in minute | [Number](https://docs.oracle.com/javase/7/docs/api/java/text/SimpleDateFormat.html#number) | 55 |
| S | Millisecond | [Number](https://docs.oracle.com/javase/7/docs/api/java/text/SimpleDateFormat.html#number) | 978 |
| z | Time zone | [General time zone](https://docs.oracle.com/javase/7/docs/api/java/text/SimpleDateFormat.html#timezone) | Pacific Standard Time; PST; GMT-08:00 |
| Z | Time zone | [RFC 822 time zone](https://docs.oracle.com/javase/7/docs/api/java/text/SimpleDateFormat.html#rfc822timezone) | -0800 |
| X | Time zone | [ISO 8601 time zone](https://docs.oracle.com/javase/7/docs/api/java/text/SimpleDateFormat.html#iso8601timezone) | -08; -0800; -08:00 |

Pattern letters are usually repeated, as their number determines the exact presentation:

* + **Text:** For formatting, if the number of pattern letters is 4 or more, the full form is used; otherwise a short or abbreviated form is used if available. For parsing, both forms are accepted, independent of the number of pattern letters.
  + **Number:** For formatting, the number of pattern letters is the minimum number of digits, and shorter numbers are zero-padded to this amount. For parsing, the number of pattern letters is ignored unless it's needed to separate two adjacent fields.
  + **Year:** If the formatter's [Calendar](https://docs.oracle.com/javase/7/docs/api/java/text/DateFormat.html#getCalendar()) is the Gregorian calendar, the following rules are applied.
    - For formatting, if the number of pattern letters is 2, the year is truncated to 2 digits; otherwise it is interpreted as a [number](https://docs.oracle.com/javase/7/docs/api/java/text/SimpleDateFormat.html#number).
    - For parsing, if the number of pattern letters is more than 2, the year is interpreted literally, regardless of the number of digits. So using the pattern "MM/dd/yyyy", "01/11/12" parses to Jan 11, 12 A.D.
    - For parsing with the abbreviated year pattern ("y" or "yy"), SimpleDateFormat must interpret the abbreviated year relative to some century. It does this by adjusting dates to be within 80 years before and 20 years after the time the SimpleDateFormat instance is created. For example, using a pattern of "MM/dd/yy" and a SimpleDateFormat instance created on Jan 1, 1997, the string "01/11/12" would be interpreted as Jan 11, 2012 while the string "05/04/64" would be interpreted as May 4, 1964. During parsing, only strings consisting of exactly two digits, as defined by [Character.isDigit(char)](https://docs.oracle.com/javase/7/docs/api/java/lang/Character.html" \l "isDigit(char)), will be parsed into the default century. Any other numeric string, such as a one digit string, a three or more digit string, or a two digit string that isn't all digits (for example, "-1"), is interpreted literally. So "01/02/3" or "01/02/003" are parsed, using the same pattern, as Jan 2, 3 AD. Likewise, "01/02/-3" is parsed as Jan 2, 4 BC.

Otherwise, calendar system specific forms are applied. For both formatting and parsing, if the number of pattern letters is 4 or more, a calendar specific [long form](https://docs.oracle.com/javase/7/docs/api/java/util/Calendar.html#LONG) is used. Otherwise, a calendar specific [short or abbreviated form](https://docs.oracle.com/javase/7/docs/api/java/util/Calendar.html#SHORT) is used.  
  
If week year 'Y' is specified and the [calendar](https://docs.oracle.com/javase/7/docs/api/java/text/DateFormat.html#getCalendar()) doesn't support any [week years](https://docs.oracle.com/javase/7/docs/api/java/util/GregorianCalendar.html#week_year), the calendar year ('y') is used instead. The support of week years can be tested with a call to [getCalendar()](https://docs.oracle.com/javase/7/docs/api/java/text/DateFormat.html" \l "getCalendar()).[isWeekDateSupported()](https://docs.oracle.com/javase/7/docs/api/java/util/Calendar.html#isWeekDateSupported()).

* + **Month:** If the number of pattern letters is 3 or more, the month is interpreted as [text](https://docs.oracle.com/javase/7/docs/api/java/text/SimpleDateFormat.html#text); otherwise, it is interpreted as a [number](https://docs.oracle.com/javase/7/docs/api/java/text/SimpleDateFormat.html#number).
  + **General time zone:** Time zones are interpreted as [text](https://docs.oracle.com/javase/7/docs/api/java/text/SimpleDateFormat.html#text) if they have names. For time zones representing a GMT offset value, the following syntax is used:
  + *GMTOffsetTimeZone:*
  + GMT *Sign* *Hours* : *Minutes*
  + *Sign:* one of
  + + -
  + *Hours:*
  + *Digit*
  + *Digit* *Digit*
  + *Minutes:*
  + *Digit* *Digit*
  + *Digit:* one of

0 1 2 3 4 5 6 7 8 9

*Hours* must be between 0 and 23, and *Minutes* must be between 00 and 59. The format is locale independent and digits must be taken from the Basic Latin block of the Unicode standard.

For parsing, [RFC 822 time zones](https://docs.oracle.com/javase/7/docs/api/java/text/SimpleDateFormat.html#rfc822timezone) are also accepted.

* + **RFC 822 time zone:** For formatting, the RFC 822 4-digit time zone format is used:
  + *RFC822TimeZone:*
  + *Sign* *TwoDigitHours* *Minutes*
  + *TwoDigitHours:*

*Digit Digit*

*TwoDigitHours* must be between 00 and 23. Other definitions are as for [general time zones](https://docs.oracle.com/javase/7/docs/api/java/text/SimpleDateFormat.html#timezone).

For parsing, [general time zones](https://docs.oracle.com/javase/7/docs/api/java/text/SimpleDateFormat.html#timezone) are also accepted.

* + **ISO 8601 Time zone:** The number of pattern letters designates the format for both formatting and parsing as follows:
  + *ISO8601TimeZone:*
  + *OneLetterISO8601TimeZone*
  + *TwoLetterISO8601TimeZone*
  + *ThreeLetterISO8601TimeZone*
  + *OneLetterISO8601TimeZone:*
  + *Sign* *TwoDigitHours*
  + Z
  + *TwoLetterISO8601TimeZone:*
  + *Sign* *TwoDigitHours* *Minutes*
  + Z
  + *ThreeLetterISO8601TimeZone:*
  + *Sign* *TwoDigitHours* : *Minutes*

Z

Other definitions are as for [general time zones](https://docs.oracle.com/javase/7/docs/api/java/text/SimpleDateFormat.html#timezone) or [RFC 822 time zones](https://docs.oracle.com/javase/7/docs/api/java/text/SimpleDateFormat.html#rfc822timezone).

For formatting, if the offset value from GMT is 0, "Z" is produced. If the number of pattern letters is 1, any fraction of an hour is ignored. For example, if the pattern is "X" and the time zone is "GMT+05:30", "+05" is produced.

For parsing, "Z" is parsed as the UTC time zone designator. [General time zones](https://docs.oracle.com/javase/7/docs/api/java/text/SimpleDateFormat.html" \l "timezone) are *not* accepted.

If the number of pattern letters is 4 or more, [IllegalArgumentException](https://docs.oracle.com/javase/7/docs/api/java/lang/IllegalArgumentException.html" \o "class in java.lang) is thrown when constructing a SimpleDateFormat or [applying a pattern](https://docs.oracle.com/javase/7/docs/api/java/text/SimpleDateFormat.html#applyPattern(java.lang.String)).

SimpleDateFormat also supports *localized date and time pattern* strings. In these strings, the pattern letters described above may be replaced with other, locale dependent, pattern letters.SimpleDateFormat does not deal with the localization of text other than the pattern letters; that's up to the client of the class.

**Examples**

The following examples show how date and time patterns are interpreted in the U.S. locale. The given date and time are 2001-07-04 12:08:56 local time in the U.S. Pacific Time time zone.

|  |  |
| --- | --- |
| **Date and Time Pattern** | **Result** |
| "yyyy.MM.dd G 'at' HH:mm:ss z" | 2001.07.04 AD at 12:08:56 PDT |
| "EEE, MMM d, ''yy" | Wed, Jul 4, '01 |
| "h:mm a" | 12:08 PM |
| "hh 'o''clock' a, zzzz" | 12 o'clock PM, Pacific Daylight Time |
| "K:mm a, z" | 0:08 PM, PDT |
| "yyyyy.MMMMM.dd GGG hh:mm aaa" | 02001.July.04 AD 12:08 PM |
| "EEE, d MMM yyyy HH:mm:ss Z" | Wed, 4 Jul 2001 12:08:56 -0700 |
| "yyMMddHHmmssZ" | 010704120856-0700 |
| "yyyy-MM-dd'T'HH:mm:ss.SSSZ" | 2001-07-04T12:08:56.235-0700 |
| "yyyy-MM-dd'T'HH:mm:ss.SSSXXX" | 2001-07-04T12:08:56.235-07:00 |
| "YYYY-'W'ww-u" | 2001-W27-3 |

**Synchronization**

Date formats are not synchronized. It is recommended to create separate format instances for each thread. If multiple threads access a format concurrently, it must be synchronized externally.

**See Also:**

[Java Tutorial](http://java.sun.com/docs/books/tutorial/i18n/format/simpleDateFormat.html), [Calendar](https://docs.oracle.com/javase/7/docs/api/java/util/Calendar.html), [TimeZone](https://docs.oracle.com/javase/7/docs/api/java/util/TimeZone.html), [DateFormat](https://docs.oracle.com/javase/7/docs/api/java/text/DateFormat.html), [DateFormatSymbols](https://docs.oracle.com/javase/7/docs/api/java/text/DateFormatSymbols.html), [Serialized Form](https://docs.oracle.com/javase/7/docs/api/serialized-form.html#java.text.SimpleDateFormat)

* + **Nested Class Summary**
    - **Nested classes/interfaces inherited from class java.text.**[**DateFormat**](https://docs.oracle.com/javase/7/docs/api/java/text/DateFormat.html)

[DateFormat.Field](https://docs.oracle.com/javase/7/docs/api/java/text/DateFormat.Field.html)

* + **Field Summary**
    - **Fields inherited from class java.text.**[**DateFormat**](https://docs.oracle.com/javase/7/docs/api/java/text/DateFormat.html)

[AM\_PM\_FIELD](https://docs.oracle.com/javase/7/docs/api/java/text/DateFormat.html#AM_PM_FIELD), [calendar](https://docs.oracle.com/javase/7/docs/api/java/text/DateFormat.html#calendar), [DATE\_FIELD](https://docs.oracle.com/javase/7/docs/api/java/text/DateFormat.html#DATE_FIELD), [DAY\_OF\_WEEK\_FIELD](https://docs.oracle.com/javase/7/docs/api/java/text/DateFormat.html#DAY_OF_WEEK_FIELD), [DAY\_OF\_WEEK\_IN\_MONTH\_FIELD](https://docs.oracle.com/javase/7/docs/api/java/text/DateFormat.html#DAY_OF_WEEK_IN_MONTH_FIELD), [DAY\_OF\_YEAR\_FIELD](https://docs.oracle.com/javase/7/docs/api/java/text/DateFormat.html#DAY_OF_YEAR_FIELD), [DEFAULT](https://docs.oracle.com/javase/7/docs/api/java/text/DateFormat.html#DEFAULT), [ERA\_FIELD](https://docs.oracle.com/javase/7/docs/api/java/text/DateFormat.html#ERA_FIELD), [FULL](https://docs.oracle.com/javase/7/docs/api/java/text/DateFormat.html#FULL), [HOUR\_OF\_DAY0\_FIELD](https://docs.oracle.com/javase/7/docs/api/java/text/DateFormat.html#HOUR_OF_DAY0_FIELD), [HOUR\_OF\_DAY1\_FIELD](https://docs.oracle.com/javase/7/docs/api/java/text/DateFormat.html#HOUR_OF_DAY1_FIELD), [HOUR0\_FIELD](https://docs.oracle.com/javase/7/docs/api/java/text/DateFormat.html#HOUR0_FIELD), [HOUR1\_FIELD](https://docs.oracle.com/javase/7/docs/api/java/text/DateFormat.html#HOUR1_FIELD), [LONG](https://docs.oracle.com/javase/7/docs/api/java/text/DateFormat.html#LONG), [MEDIUM](https://docs.oracle.com/javase/7/docs/api/java/text/DateFormat.html#MEDIUM), [MILLISECOND\_FIELD](https://docs.oracle.com/javase/7/docs/api/java/text/DateFormat.html#MILLISECOND_FIELD), [MINUTE\_FIELD](https://docs.oracle.com/javase/7/docs/api/java/text/DateFormat.html#MINUTE_FIELD), [MONTH\_FIELD](https://docs.oracle.com/javase/7/docs/api/java/text/DateFormat.html#MONTH_FIELD), [numberFormat](https://docs.oracle.com/javase/7/docs/api/java/text/DateFormat.html#numberFormat), [SECOND\_FIELD](https://docs.oracle.com/javase/7/docs/api/java/text/DateFormat.html#SECOND_FIELD), [SHORT](https://docs.oracle.com/javase/7/docs/api/java/text/DateFormat.html#SHORT), [TIMEZONE\_FIELD](https://docs.oracle.com/javase/7/docs/api/java/text/DateFormat.html#TIMEZONE_FIELD), [WEEK\_OF\_MONTH\_FIELD](https://docs.oracle.com/javase/7/docs/api/java/text/DateFormat.html#WEEK_OF_MONTH_FIELD), [WEEK\_OF\_YEAR\_FIELD](https://docs.oracle.com/javase/7/docs/api/java/text/DateFormat.html#WEEK_OF_YEAR_FIELD), [YEAR\_FIELD](https://docs.oracle.com/javase/7/docs/api/java/text/DateFormat.html#YEAR_FIELD)

* + **Constructor Summary**

|  |
| --- |
| **Constructors** |
| **Constructor and Description** |
| [**SimpleDateFormat**](https://docs.oracle.com/javase/7/docs/api/java/text/SimpleDateFormat.html#SimpleDateFormat())()  Constructs a SimpleDateFormat using the default pattern and date format symbols for the default locale. |
| **[SimpleDateFormat](https://docs.oracle.com/javase/7/docs/api/java/text/SimpleDateFormat.html" \l "SimpleDateFormat(java.lang.String))**([**String**](https://docs.oracle.com/javase/7/docs/api/java/lang/String.html) pattern)  Constructs a SimpleDateFormat using the given pattern and the default date format symbols for the default locale. |
| **[SimpleDateFormat](https://docs.oracle.com/javase/7/docs/api/java/text/SimpleDateFormat.html" \l "SimpleDateFormat(java.lang.String,%20java.text.DateFormatSymbols))**([**String**](https://docs.oracle.com/javase/7/docs/api/java/lang/String.html) pattern, [**DateFormatSymbols**](https://docs.oracle.com/javase/7/docs/api/java/text/DateFormatSymbols.html) formatSymbols)  Constructs a SimpleDateFormat using the given pattern and date format symbols. |
| **[SimpleDateFormat](https://docs.oracle.com/javase/7/docs/api/java/text/SimpleDateFormat.html" \l "SimpleDateFormat(java.lang.String,%20java.util.Locale))**([**String**](https://docs.oracle.com/javase/7/docs/api/java/lang/String.html) pattern, [**Locale**](https://docs.oracle.com/javase/7/docs/api/java/util/Locale.html) locale)  Constructs a SimpleDateFormat using the given pattern and the default date format symbols for the given locale. |

* + **Method Summary**

|  |  |
| --- | --- |
| **Methods** | |
| **Modifier and Type** | **Method and Description** |
| void | [**applyLocalizedPattern**](https://docs.oracle.com/javase/7/docs/api/java/text/SimpleDateFormat.html#applyLocalizedPattern(java.lang.String))(**[String](https://docs.oracle.com/javase/7/docs/api/java/lang/String.html" \o "class in java.lang)** pattern)  Applies the given localized pattern string to this date format. |
| void | [**applyPattern**](https://docs.oracle.com/javase/7/docs/api/java/text/SimpleDateFormat.html#applyPattern(java.lang.String))([**String**](https://docs.oracle.com/javase/7/docs/api/java/lang/String.html) pattern)  Applies the given pattern string to this date format. |
| **[Object](https://docs.oracle.com/javase/7/docs/api/java/lang/Object.html" \o "class in java.lang)** | [**clone**](https://docs.oracle.com/javase/7/docs/api/java/text/SimpleDateFormat.html#clone())()  Creates a copy of this SimpleDateFormat. |
| boolean | [**equals**](https://docs.oracle.com/javase/7/docs/api/java/text/SimpleDateFormat.html#equals(java.lang.Object))([**Object**](https://docs.oracle.com/javase/7/docs/api/java/lang/Object.html) obj)  Compares the given object with this SimpleDateFormat for equality. |
| **[StringBuffer](https://docs.oracle.com/javase/7/docs/api/java/lang/StringBuffer.html" \o "class in java.lang)** | [**format**](https://docs.oracle.com/javase/7/docs/api/java/text/SimpleDateFormat.html#format(java.util.Date,%20java.lang.StringBuffer,%20java.text.FieldPosition))([**Date**](https://docs.oracle.com/javase/7/docs/api/java/util/Date.html) date, [**StringBuffer**](https://docs.oracle.com/javase/7/docs/api/java/lang/StringBuffer.html) toAppendTo, **[FieldPosition](https://docs.oracle.com/javase/7/docs/api/java/text/FieldPosition.html" \o "class in java.text)** pos)  Formats the given Date into a date/time string and appends the result to the given StringBuffer. |
| **[AttributedCharacterIterator](https://docs.oracle.com/javase/7/docs/api/java/text/AttributedCharacterIterator.html" \o "interface in java.text)** | [**formatToCharacterIterator**](https://docs.oracle.com/javase/7/docs/api/java/text/SimpleDateFormat.html#formatToCharacterIterator(java.lang.Object))([**Object**](https://docs.oracle.com/javase/7/docs/api/java/lang/Object.html) obj)  Formats an Object producing an AttributedCharacterIterator. |
| **[Date](https://docs.oracle.com/javase/7/docs/api/java/util/Date.html" \o "class in java.util)** | [**get2DigitYearStart**](https://docs.oracle.com/javase/7/docs/api/java/text/SimpleDateFormat.html#get2DigitYearStart())()  Returns the beginning date of the 100-year period 2-digit years are interpreted as being within. |
| **[DateFormatSymbols](https://docs.oracle.com/javase/7/docs/api/java/text/DateFormatSymbols.html" \o "class in java.text)** | [**getDateFormatSymbols**](https://docs.oracle.com/javase/7/docs/api/java/text/SimpleDateFormat.html#getDateFormatSymbols())()  Gets a copy of the date and time format symbols of this date format. |
| int | [**hashCode**](https://docs.oracle.com/javase/7/docs/api/java/text/SimpleDateFormat.html#hashCode())()  Returns the hash code value for this SimpleDateFormat object. |
| **[Date](https://docs.oracle.com/javase/7/docs/api/java/util/Date.html" \o "class in java.util)** | [**parse**](https://docs.oracle.com/javase/7/docs/api/java/text/SimpleDateFormat.html#parse(java.lang.String,%20java.text.ParsePosition))([**String**](https://docs.oracle.com/javase/7/docs/api/java/lang/String.html) text, **[ParsePosition](https://docs.oracle.com/javase/7/docs/api/java/text/ParsePosition.html" \o "class in java.text)** pos)  Parses text from a string to produce a Date. |
| void | [**set2DigitYearStart**](https://docs.oracle.com/javase/7/docs/api/java/text/SimpleDateFormat.html#set2DigitYearStart(java.util.Date))([**Date**](https://docs.oracle.com/javase/7/docs/api/java/util/Date.html) startDate)  Sets the 100-year period 2-digit years will be interpreted as being in to begin on the date the user specifies. |
| void | [**setDateFormatSymbols**](https://docs.oracle.com/javase/7/docs/api/java/text/SimpleDateFormat.html#setDateFormatSymbols(java.text.DateFormatSymbols))(**[DateFormatSymbols](https://docs.oracle.com/javase/7/docs/api/java/text/DateFormatSymbols.html" \o "class in java.text)** newFormatSymbols)  Sets the date and time format symbols of this date format. |
| **[String](https://docs.oracle.com/javase/7/docs/api/java/lang/String.html" \o "class in java.lang)** | [**toLocalizedPattern**](https://docs.oracle.com/javase/7/docs/api/java/text/SimpleDateFormat.html#toLocalizedPattern())()  Returns a localized pattern string describing this date format. |
| **[String](https://docs.oracle.com/javase/7/docs/api/java/lang/String.html" \o "class in java.lang)** | [**toPattern**](https://docs.oracle.com/javase/7/docs/api/java/text/SimpleDateFormat.html#toPattern())()  Returns a pattern string describing this date format. |

* + - **Methods inherited from class java.text.**[**DateFormat**](https://docs.oracle.com/javase/7/docs/api/java/text/DateFormat.html)

[format](https://docs.oracle.com/javase/7/docs/api/java/text/DateFormat.html#format(java.util.Date)), [format](https://docs.oracle.com/javase/7/docs/api/java/text/DateFormat.html#format(java.lang.Object,%20java.lang.StringBuffer,%20java.text.FieldPosition)), [getAvailableLocales](https://docs.oracle.com/javase/7/docs/api/java/text/DateFormat.html#getAvailableLocales()), [getCalendar](https://docs.oracle.com/javase/7/docs/api/java/text/DateFormat.html#getCalendar()), [getDateInstance](https://docs.oracle.com/javase/7/docs/api/java/text/DateFormat.html#getDateInstance()), [getDateInstance](https://docs.oracle.com/javase/7/docs/api/java/text/DateFormat.html#getDateInstance(int)), [getDateInstance](https://docs.oracle.com/javase/7/docs/api/java/text/DateFormat.html#getDateInstance(int,%20java.util.Locale)), [getDateTimeInstance](https://docs.oracle.com/javase/7/docs/api/java/text/DateFormat.html#getDateTimeInstance()), [getDateTimeInstance](https://docs.oracle.com/javase/7/docs/api/java/text/DateFormat.html#getDateTimeInstance(int,%20int)), [getDateTimeInstance](https://docs.oracle.com/javase/7/docs/api/java/text/DateFormat.html#getDateTimeInstance(int,%20int,%20java.util.Locale)), [getInstance](https://docs.oracle.com/javase/7/docs/api/java/text/DateFormat.html#getInstance()), [getNumberFormat](https://docs.oracle.com/javase/7/docs/api/java/text/DateFormat.html#getNumberFormat()), [getTimeInstance](https://docs.oracle.com/javase/7/docs/api/java/text/DateFormat.html#getTimeInstance()), [getTimeInstance](https://docs.oracle.com/javase/7/docs/api/java/text/DateFormat.html#getTimeInstance(int)), [getTimeInstance](https://docs.oracle.com/javase/7/docs/api/java/text/DateFormat.html#getTimeInstance(int,%20java.util.Locale)), [getTimeZone](https://docs.oracle.com/javase/7/docs/api/java/text/DateFormat.html#getTimeZone()), [isLenient](https://docs.oracle.com/javase/7/docs/api/java/text/DateFormat.html#isLenient()), [parse](https://docs.oracle.com/javase/7/docs/api/java/text/DateFormat.html#parse(java.lang.String)), [parseObject](https://docs.oracle.com/javase/7/docs/api/java/text/DateFormat.html#parseObject(java.lang.String,%20java.text.ParsePosition)), [setCalendar](https://docs.oracle.com/javase/7/docs/api/java/text/DateFormat.html#setCalendar(java.util.Calendar)), [setLenient](https://docs.oracle.com/javase/7/docs/api/java/text/DateFormat.html#setLenient(boolean)), [setNumberFormat](https://docs.oracle.com/javase/7/docs/api/java/text/DateFormat.html#setNumberFormat(java.text.NumberFormat)), [setTimeZone](https://docs.oracle.com/javase/7/docs/api/java/text/DateFormat.html#setTimeZone(java.util.TimeZone))

* + - **Methods inherited from class java.text.**[**Format**](https://docs.oracle.com/javase/7/docs/api/java/text/Format.html)

[format](https://docs.oracle.com/javase/7/docs/api/java/text/Format.html#format(java.lang.Object)), [parseObject](https://docs.oracle.com/javase/7/docs/api/java/text/Format.html" \l "parseObject(java.lang.String))

* + - **Methods inherited from class java.lang.**[**Object**](https://docs.oracle.com/javase/7/docs/api/java/lang/Object.html)

[finalize](https://docs.oracle.com/javase/7/docs/api/java/lang/Object.html#finalize()), [getClass](https://docs.oracle.com/javase/7/docs/api/java/lang/Object.html#getClass()), [notify](https://docs.oracle.com/javase/7/docs/api/java/lang/Object.html#notify()), [notifyAll](https://docs.oracle.com/javase/7/docs/api/java/lang/Object.html#notifyAll()), [toString](https://docs.oracle.com/javase/7/docs/api/java/lang/Object.html#toString()), [wait](https://docs.oracle.com/javase/7/docs/api/java/lang/Object.html#wait()), [wait](https://docs.oracle.com/javase/7/docs/api/java/lang/Object.html#wait(long)), [wait](https://docs.oracle.com/javase/7/docs/api/java/lang/Object.html#wait(long,%20int))

* + **Constructor Detail**
    - **SimpleDateFormat**

public SimpleDateFormat()

Constructs a SimpleDateFormat using the default pattern and date format symbols for the default locale. **Note:** This constructor may not support all locales. For full coverage, use the factory methods in the [DateFormat](https://docs.oracle.com/javase/7/docs/api/java/text/DateFormat.html" \o "class in java.text) class.

* + - **SimpleDateFormat**

public SimpleDateFormat([String](https://docs.oracle.com/javase/7/docs/api/java/lang/String.html" \o "class in java.lang) pattern)

Constructs a SimpleDateFormat using the given pattern and the default date format symbols for the default locale. **Note:** This constructor may not support all locales. For full coverage, use the factory methods in the [DateFormat](https://docs.oracle.com/javase/7/docs/api/java/text/DateFormat.html" \o "class in java.text) class.

**Parameters:**

pattern - the pattern describing the date and time format

**Throws:**

[NullPointerException](https://docs.oracle.com/javase/7/docs/api/java/lang/NullPointerException.html" \o "class in java.lang) - if the given pattern is null

[IllegalArgumentException](https://docs.oracle.com/javase/7/docs/api/java/lang/IllegalArgumentException.html) - if the given pattern is invalid

* + - **SimpleDateFormat**
    - public SimpleDateFormat([String](https://docs.oracle.com/javase/7/docs/api/java/lang/String.html" \o "class in java.lang) pattern,

[Locale](https://docs.oracle.com/javase/7/docs/api/java/util/Locale.html) locale)

Constructs a SimpleDateFormat using the given pattern and the default date format symbols for the given locale. **Note:** This constructor may not support all locales. For full coverage, use the factory methods in the [DateFormat](https://docs.oracle.com/javase/7/docs/api/java/text/DateFormat.html" \o "class in java.text) class.

**Parameters:**

pattern - the pattern describing the date and time format

locale - the locale whose date format symbols should be used

**Throws:**

[NullPointerException](https://docs.oracle.com/javase/7/docs/api/java/lang/NullPointerException.html" \o "class in java.lang) - if the given pattern or locale is null

[IllegalArgumentException](https://docs.oracle.com/javase/7/docs/api/java/lang/IllegalArgumentException.html) - if the given pattern is invalid

* + - **SimpleDateFormat**
    - public SimpleDateFormat([String](https://docs.oracle.com/javase/7/docs/api/java/lang/String.html" \o "class in java.lang) pattern,

[DateFormatSymbols](https://docs.oracle.com/javase/7/docs/api/java/text/DateFormatSymbols.html) formatSymbols)

Constructs a SimpleDateFormat using the given pattern and date format symbols.

**Parameters:**

pattern - the pattern describing the date and time format

formatSymbols - the date format symbols to be used for formatting

**Throws:**

[NullPointerException](https://docs.oracle.com/javase/7/docs/api/java/lang/NullPointerException.html" \o "class in java.lang) - if the given pattern or formatSymbols is null

[IllegalArgumentException](https://docs.oracle.com/javase/7/docs/api/java/lang/IllegalArgumentException.html) - if the given pattern is invalid

* + **Method Detail**
    - **set2DigitYearStart**

public void set2DigitYearStart([Date](https://docs.oracle.com/javase/7/docs/api/java/util/Date.html) startDate)

Sets the 100-year period 2-digit years will be interpreted as being in to begin on the date the user specifies.

**Parameters:**

startDate - During parsing, two digit years will be placed in the range startDate to startDate + 100 years.

**Since:**

1.2

**See Also:**

[get2DigitYearStart()](https://docs.oracle.com/javase/7/docs/api/java/text/SimpleDateFormat.html#get2DigitYearStart())

* + - **get2DigitYearStart**

public [Date](https://docs.oracle.com/javase/7/docs/api/java/util/Date.html) get2DigitYearStart()

Returns the beginning date of the 100-year period 2-digit years are interpreted as being within.

**Returns:**

the start of the 100-year period into which two digit years are parsed

**Since:**

1.2

**See Also:**

[set2DigitYearStart(java.util.Date)](https://docs.oracle.com/javase/7/docs/api/java/text/SimpleDateFormat.html" \l "set2DigitYearStart(java.util.Date))

* + - **format**
    - public [StringBuffer](https://docs.oracle.com/javase/7/docs/api/java/lang/StringBuffer.html" \o "class in java.lang) format([Date](https://docs.oracle.com/javase/7/docs/api/java/util/Date.html) date,
    - [StringBuffer](https://docs.oracle.com/javase/7/docs/api/java/lang/StringBuffer.html) toAppendTo,

[FieldPosition](https://docs.oracle.com/javase/7/docs/api/java/text/FieldPosition.html) pos)

Formats the given Date into a date/time string and appends the result to the given StringBuffer.

**Specified by:**

[format](https://docs.oracle.com/javase/7/docs/api/java/text/DateFormat.html" \l "format(java.util.Date,%20java.lang.StringBuffer,%20java.text.FieldPosition)) in class [DateFormat](https://docs.oracle.com/javase/7/docs/api/java/text/DateFormat.html" \o "class in java.text)

**Parameters:**

date - the date-time value to be formatted into a date-time string.

toAppendTo - where the new date-time text is to be appended.

pos - the formatting position. On input: an alignment field, if desired. On output: the offsets of the alignment field.

**Returns:**

the formatted date-time string.

**Throws:**

[NullPointerException](https://docs.oracle.com/javase/7/docs/api/java/lang/NullPointerException.html" \o "class in java.lang) - if the given date is null.

* + - **formatToCharacterIterator**

public [AttributedCharacterIterator](https://docs.oracle.com/javase/7/docs/api/java/text/AttributedCharacterIterator.html) formatToCharacterIterator([Object](https://docs.oracle.com/javase/7/docs/api/java/lang/Object.html) obj)

Formats an Object producing an AttributedCharacterIterator. You can use the returned AttributedCharacterIterator to build the resulting String, as well as to determine information about the resulting String.

Each attribute key of the AttributedCharacterIterator will be of type DateFormat.Field, with the corresponding attribute value being the same as the attribute key.

**Overrides:**

[formatToCharacterIterator](https://docs.oracle.com/javase/7/docs/api/java/text/Format.html" \l "formatToCharacterIterator(java.lang.Object)) in class [Format](https://docs.oracle.com/javase/7/docs/api/java/text/Format.html)

**Parameters:**

obj - The object to format

**Returns:**

AttributedCharacterIterator describing the formatted value.

**Throws:**

[NullPointerException](https://docs.oracle.com/javase/7/docs/api/java/lang/NullPointerException.html" \o "class in java.lang) - if obj is null.

[IllegalArgumentException](https://docs.oracle.com/javase/7/docs/api/java/lang/IllegalArgumentException.html) - if the Format cannot format the given object, or if the Format's pattern string is invalid.

**Since:**

1.4

* + - **parse**
    - public [Date](https://docs.oracle.com/javase/7/docs/api/java/util/Date.html) parse([String](https://docs.oracle.com/javase/7/docs/api/java/lang/String.html) text,

[ParsePosition](https://docs.oracle.com/javase/7/docs/api/java/text/ParsePosition.html) pos)

Parses text from a string to produce a Date.

The method attempts to parse text starting at the index given by pos. If parsing succeeds, then the index of pos is updated to the index after the last character used (parsing does not necessarily use all characters up to the end of the string), and the parsed date is returned. The updated pos can be used to indicate the starting point for the next call to this method. If an error occurs, then the index of pos is not changed, the error index of pos is set to the index of the character where the error occurred, and null is returned.

This parsing operation uses the [calendar](https://docs.oracle.com/javase/7/docs/api/java/text/DateFormat.html#calendar) to produce a Date. All of the calendar's date-time fields are [cleared](https://docs.oracle.com/javase/7/docs/api/java/util/Calendar.html#clear()) before parsing, and the calendar's default values of the date-time fields are used for any missing date-time information. For example, the year value of the parsed Date is 1970 with [GregorianCalendar](https://docs.oracle.com/javase/7/docs/api/java/util/GregorianCalendar.html" \o "class in java.util) if no year value is given from the parsing operation. The TimeZone value may be overwritten, depending on the given pattern and the time zone value in text. Any TimeZone value that has previously been set by a call to[setTimeZone](https://docs.oracle.com/javase/7/docs/api/java/text/DateFormat.html#setTimeZone(java.util.TimeZone)) may need to be restored for further operations.

**Specified by:**

[parse](https://docs.oracle.com/javase/7/docs/api/java/text/DateFormat.html" \l "parse(java.lang.String,%20java.text.ParsePosition)) in class [DateFormat](https://docs.oracle.com/javase/7/docs/api/java/text/DateFormat.html" \o "class in java.text)

**Parameters:**

text - A String, part of which should be parsed.

pos - A ParsePosition object with index and error index information as described above.

**Returns:**

A Date parsed from the string. In case of error, returns null.

**Throws:**

[NullPointerException](https://docs.oracle.com/javase/7/docs/api/java/lang/NullPointerException.html" \o "class in java.lang) - if text or pos is null.

* + - **toPattern**

public [String](https://docs.oracle.com/javase/7/docs/api/java/lang/String.html" \o "class in java.lang) toPattern()

Returns a pattern string describing this date format.

**Returns:**

a pattern string describing this date format.

* + - **toLocalizedPattern**

public [String](https://docs.oracle.com/javase/7/docs/api/java/lang/String.html" \o "class in java.lang) toLocalizedPattern()

Returns a localized pattern string describing this date format.

**Returns:**

a localized pattern string describing this date format.

* + - **applyPattern**

public void applyPattern([String](https://docs.oracle.com/javase/7/docs/api/java/lang/String.html) pattern)

Applies the given pattern string to this date format.

**Parameters:**

pattern - the new date and time pattern for this date format

**Throws:**

[NullPointerException](https://docs.oracle.com/javase/7/docs/api/java/lang/NullPointerException.html" \o "class in java.lang) - if the given pattern is null

[IllegalArgumentException](https://docs.oracle.com/javase/7/docs/api/java/lang/IllegalArgumentException.html) - if the given pattern is invalid

* + - **applyLocalizedPattern**

public void applyLocalizedPattern([String](https://docs.oracle.com/javase/7/docs/api/java/lang/String.html) pattern)

Applies the given localized pattern string to this date format.

**Parameters:**

pattern - a String to be mapped to the new date and time format pattern for this format

**Throws:**

[NullPointerException](https://docs.oracle.com/javase/7/docs/api/java/lang/NullPointerException.html" \o "class in java.lang) - if the given pattern is null

[IllegalArgumentException](https://docs.oracle.com/javase/7/docs/api/java/lang/IllegalArgumentException.html) - if the given pattern is invalid

* + - **getDateFormatSymbols**

public [DateFormatSymbols](https://docs.oracle.com/javase/7/docs/api/java/text/DateFormatSymbols.html" \o "class in java.text) getDateFormatSymbols()

Gets a copy of the date and time format symbols of this date format.

**Returns:**

the date and time format symbols of this date format

**See Also:**

[setDateFormatSymbols(java.text.DateFormatSymbols)](https://docs.oracle.com/javase/7/docs/api/java/text/SimpleDateFormat.html" \l "setDateFormatSymbols(java.text.DateFormatSymbols))

* + - **setDateFormatSymbols**

public void setDateFormatSymbols([DateFormatSymbols](https://docs.oracle.com/javase/7/docs/api/java/text/DateFormatSymbols.html) newFormatSymbols)

Sets the date and time format symbols of this date format.

**Parameters:**

newFormatSymbols - the new date and time format symbols

**Throws:**

[NullPointerException](https://docs.oracle.com/javase/7/docs/api/java/lang/NullPointerException.html" \o "class in java.lang) - if the given newFormatSymbols is null

**See Also:**

[getDateFormatSymbols()](https://docs.oracle.com/javase/7/docs/api/java/text/SimpleDateFormat.html#getDateFormatSymbols())

* + - **clone**

public [Object](https://docs.oracle.com/javase/7/docs/api/java/lang/Object.html" \o "class in java.lang) clone()

Creates a copy of this SimpleDateFormat. This also clones the format's date format symbols.

**Overrides:**

[clone](https://docs.oracle.com/javase/7/docs/api/java/text/DateFormat.html" \l "clone()) in class [DateFormat](https://docs.oracle.com/javase/7/docs/api/java/text/DateFormat.html" \o "class in java.text)

**Returns:**

a clone of this SimpleDateFormat

**See Also:**

[Cloneable](https://docs.oracle.com/javase/7/docs/api/java/lang/Cloneable.html" \o "interface in java.lang)

* + - **hashCode**

public int hashCode()

Returns the hash code value for this SimpleDateFormat object.

**Overrides:**

[hashCode](https://docs.oracle.com/javase/7/docs/api/java/text/DateFormat.html" \l "hashCode()) in class [DateFormat](https://docs.oracle.com/javase/7/docs/api/java/text/DateFormat.html" \o "class in java.text)

**Returns:**

the hash code value for this SimpleDateFormat object.

**See Also:**

[Object.equals(java.lang.Object)](https://docs.oracle.com/javase/7/docs/api/java/lang/Object.html" \l "equals(java.lang.Object)), [System.identityHashCode(java.lang.Object)](https://docs.oracle.com/javase/7/docs/api/java/lang/System.html#identityHashCode(java.lang.Object))

* + - **equals**

public boolean equals([Object](https://docs.oracle.com/javase/7/docs/api/java/lang/Object.html) obj)

Compares the given object with this SimpleDateFormat for equality.

**Overrides:**

[equals](https://docs.oracle.com/javase/7/docs/api/java/text/DateFormat.html" \l "equals(java.lang.Object)) in class [DateFormat](https://docs.oracle.com/javase/7/docs/api/java/text/DateFormat.html" \o "class in java.text)

**Parameters:**

obj - the reference object with which to compare.

**Returns:**

true if the given object is equal to this SimpleDateFormat

**See Also:**

[Object.hashCode()](https://docs.oracle.com/javase/7/docs/api/java/lang/Object.html#hashCode()), [HashMap](https://docs.oracle.com/javase/7/docs/api/java/util/HashMap.html" \o "class in java.util)