**使用printf格式化日期**

printf 方法可以很轻松地格式化时间和日期。使用两个字母格式，它以 %t 开头并且以下面表格中的一个字母结尾。

%tY：输出四位数的年份，例如：2023

%ty：输出两位数的年份，例如：23

%tm：输出两位数的月份，例如：02

%tB：输出月份的全名，例如：February

%tb：输出月份的缩写，例如：Feb

%tA：输出星期的全名，例如：Wednesday

%ta：输出星期的缩写，例如：Wed

%td：输出两位数的日期，例如：24

%te：输出一位或两位数的日期，例如：24 或 02

%tH：输出24小时制的小时数，例如：23

%tI：输出12小时制的小时数，例如：11

%tM：输出分钟数，例如：45

%tS：输出秒数，例如：30

%tp：输出上午还是下午，例如：AM 或 PM

%tZ：输出时区，例如：GMT+08:00

转换符

说明

示例

%tc

包括全部日期和时间信息

星期六 十月 27 14:21:20 CST 2007

%tF

"年-月-日"格式

2007-10-27

%tD

"月/日/年"格式

10/27/07

%tr

"HH:MM:SS PM"格式（12时制）

02:25:51 下午

%tT

"HH:MM:SS"格式（24时制）

14:28:16

%tR

"HH:MM"格式（24时制）

14:28

更多 printf 解析可以参见：Java 格式化输出 printf 例子

实例

实例

import java.util.Date;

public class DateFormatExample {

public static void main(String[] args) {

Date date = new Date();

System.out.printf("%tY-%tm-%td %tH:%tM:%tS %tZ", date, date, date, date, date, date);

}

}

执行输出结果为：

2023-02-24 13:34:45 GMT+08:00

实例

import java.util.Date;

public class DateDemo {

public static void main(String[] args) {

// 初始化 Date 对象

Date date = new Date();

//c的使用

System.out.printf("全部日期和时间信息：%tc%n",date);

//f的使用

System.out.printf("年-月-日格式：%tF%n",date);

//d的使用

System.out.printf("月/日/年格式：%tD%n",date);

//r的使用

System.out.printf("HH:MM:SS PM格式（12时制）：%tr%n",date);

//t的使用

System.out.printf("HH:MM:SS格式（24时制）：%tT%n",date);

//R的使用

System.out.printf("HH:MM格式（24时制）：%tR",date);

}

}

以上实例编译运行结果如下:

全部日期和时间信息：星期一 九月 10 10:43:36 CST 2012

年-月-日格式：2012-09-10

月/日/年格式：09/10/12

HH:MM:SS PM格式（12时制）：10:43:36 上午

HH:MM:SS格式（24时制）：10:43:36

HH:MM格式（24时制）：10:43

如果你需要重复提供日期，那么利用这种方式来格式化它的每一部分就有点复杂了。因此，可以利用一个格式化字符串指出要被格式化的参数的索引。

索引必须紧跟在 % 后面，而且必须以 $ 结束。例如：

实例

import java.util.Date;

public class DateDemo {

public static void main(String[] args) {

// 初始化 Date 对象

Date date = new Date();

// 使用toString()显示日期和时间

System.out.printf("%1$s %2$tB %2$td, %2$tY",

"Due date:", date);

}

}

运行实例 »

以上实例编译运行结果如下:

Due date: February 09, 2014

或者，你可以使用 < 标志。它表明先前被格式化的参数要被再次使用。例如：

实例

import java.util.Date;

public class DateDemo {

public static void main(String[] args) {

// 初始化 Date 对象

Date date = new Date();

// 显示格式化时间

System.out.printf("%s %tB %<te, %<tY",

"Due date:", date);

}

}

运行实例 »

以上实例编译运行结果如下:

Due date: February 09, 2014

定义日期格式的转换符可以使日期通过指定的转换符生成新字符串。这些日期转换符如下所示：

实例

import java.util.\*;

public class DateDemo {

public static void main(String[] args) {

Date date=new Date();

//b的使用，月份简称

String str=String.format(Locale.US,"英文月份简称：%tb",date);

System.out.println(str);

System.out.printf("本地月份简称：%tb%n",date);

//B的使用，月份全称

str=String.format(Locale.US,"英文月份全称：%tB",date);

System.out.println(str);

System.out.printf("本地月份全称：%tB%n",date);

//a的使用，星期简称

str=String.format(Locale.US,"英文星期的简称：%ta",date);

System.out.println(str);

//A的使用，星期全称

System.out.printf("本地星期的简称：%tA%n",date);

//C的使用，年前两位

System.out.printf("年的前两位数字（不足两位前面补0）：%tC%n",date);

//y的使用，年后两位

System.out.printf("年的后两位数字（不足两位前面补0）：%ty%n",date);

//j的使用，一年的天数

System.out.printf("一年中的天数（即年的第几天）：%tj%n",date);

//m的使用，月份

System.out.printf("两位数字的月份（不足两位前面补0）：%tm%n",date);

//d的使用，日（二位，不够补零）

System.out.printf("两位数字的日（不足两位前面补0）：%td%n",date);

//e的使用，日（一位不补零）

System.out.printf("月份的日（前面不补0）：%te",date);

}

}

输出结果为：

英文月份简称：May

本地月份简称：五月

英文月份全称：May

本地月份全称：五月

英文星期的简称：Thu

本地星期的简称：星期四

年的前两位数字（不足两位前面补0）：20

年的后两位数字（不足两位前面补0）：17

一年中的天数（即年的第几天）：124

两位数字的月份（不足两位前面补0）：05

两位数字的日（不足两位前面补0）：04

月份的日（前面不补0）：4