

# 日期 (Date)

@M了个J  
李明杰

<https://github.com/CoderMJLee>

<http://cnblogs.com/mjios>



实力IT教育 www.520it.com

码拉松



# Date

- `java.util.Date` 是开发中经常用到的日期处理类（注意：不是 `java.sql.Date`）
- 包含了年、月、日、时、分、秒等信息

```
// date1 和 date2 都代表当前时间
Date date1 = new Date();
Date date2 = new Date(System.currentTimeMillis());

// Tue Dec 24 12:30:00 CST 2019
System.out.println(date1);
// Tue Dec 24 12:30:00 CST 2019
System.out.println(date2);
```

- `System.currentTimeMillis` 返回的是从 1970-01-01 00:00:00 GMT 开始到现在经历的毫秒数
- 1970-01-01 00:00:00 GMT、1970-01-01 08:00:00 CST代表的是同一个时间
- ✓ GMT (Greenwich Mean Time) : 格林尼治时间
- ✓ CST (China Standard Time UT+8:00) : 中国标准时间

# Date 常用方法

```
Date d1 = new Date();  
Date d2 = new Date();  
  
// 设置毫秒数  
d1.setTime(1000);  
d2.setTime(2000);  
  
// 获取毫秒数  
d1.getTime() // 1000  
d2.getTime() // 2000  
  
// 比较时间  
d2.after(d1) // true  
d1.before(d2) // true  
d1.compareTo(d2) // -1
```

# SimpleDateFormat

- java.text.SimpleDateFormat 常用来进行日期的格式化处理

```
SimpleDateFormat fmt = new SimpleDateFormat("yyyy年MM月dd日 HH:mm:ss");  
// 利用日期对象生成字符串  
String str = fmt.format(new Date());  
// 2019年12月24日 15:40:53  
System.out.println(str);  
  
// 解析字符串为日期对象  
Date date = fmt.parse(str);  
// Tue Dec 24 15:40:53 CST 2019  
System.out.println(date);
```

# SimpleDateFormat 的模式字母

字母	元素	示例
G	Era 标志符	AD
y	年	1996; 96
M	年中的月份	July; Jul; 07
w	年中的周数	27
W	月份中的周数	2
D	年中的天数	189
d	月份中的天数	10
F	月份中的星期	2
E	星期中的天数	Tuesday; Tue
a	Am/pm 标记	PM

字母	元素	示例
H	一天中的小时数 (0-23)	0
k	一天中的小时数 (1-24)	24
K	am/pm 中的小时数 (0-11)	0
h	am/pm 中的小时数 (1-12)	12
m	小时中的分钟数	30
s	分钟中的秒数	55
S	毫秒数	978
z	时区	Pacific Standard Time; PST; GMT-08:00
Z	时区	-0800

# Calendar

- java.util.Calendar 也是开发中经常用到的日期处理类
- ▣ 功能比 Date 更加丰富，Date 中很多过期的方法都迁移到了 Calendar 中

```
// 表示当前时间
Calendar c = Calendar.getInstance();
c.get(Calendar.YEAR) // 年
// 月（取值范围[0, 11]，0是1月，11是12月）
c.get(Calendar.MONTH)
// 一月中的第几天（取值范围[1, 31]）
c.get(Calendar.DAY_OF_MONTH)
// 一周中的第几天（取值范围[1, 7]，1是星期天，2是星期一，...，7是星期六）
c.get(Calendar.DAY_OF_WEEK)
// 一年中的第几天（取值范围[1, 366]）
c.get(Calendar.DAY_OF_YEAR)
c.get(Calendar.HOUR) // 时
c.get(Calendar.MINUTE) // 分
c.get(Calendar.SECOND) // 秒
c.get(Calendar.MILLISECOND) // 毫秒
```

# Calendar 的常用方法

```
Calendar c = Calendar.getInstance();  
// 2019年7月6日  
c.set(2019, 06, 06);  
// 2019年7月11日  
c.add(Calendar.DAY_OF_MONTH, 5);  
// 2019年9月11日  
c.add(Calendar.MONTH, 2);  
  
// 设置Date对象  
c.setTime(new Date());  
// 获得Date对象  
c.getTime()  
  
// 设置毫秒数  
c.setTimeInMillis(System.currentTimeMillis());  
// 获得毫秒数  
c.getTimeInMillis()
```

# 打印格式化（很少用）

转换符	作用
tB	month (本地化)
td, te	day (若只有 1 位数字, td 会在前面补 0)
ty, tY	ty = 2-digit year, tY = 4-digit year
tl	hour (12 小时制)
tM	minute (若只有 1 位数字, 在前面补 0)
tp	am/pm (本地化)
tm	month (若只有 1 位数字, 在前面补 0)
tD	%tm%td%ty

```
Calendar c = Calendar.getInstance();
Date date = new Date();
System.out.format("%tB %te, %tY%n", date, date, date); // 十二月 24, 2019
System.out.format("%tl:%tM %tp%n", c, c, c); // 4:32 下午
System.out.format("%tD%n", c); // 12/24/19
```