

C# ToString()方法一些特殊用法

一、取中文日期显示

1. 年月日时分

```
1 | currentTime.ToString("f"); //不显示秒
```

2. 年月

```
1 | currentTime.ToString("y");
```

3. 月日

```
1 | currentTime.ToString("m");
```

4. 格式为：2003-9-23

```
1 | currentTime.ToString("d");
```

5. 格式为：14:24

```
1 | currentTime.ToString("t");
```

二、字符型转换 转为字符串

代码	结果
12345.ToString("n");	12,345.00
12345.ToString("C");	¥ 12,345.00
12345.ToString("e");	1.234500e+004
12345.ToString("f4");	12345.0000
12345.ToString("x");	3039 (16进制)
12345.ToString("p");	1,234,500.00%

令DateTime.Now为2007-7-17 22:07:24

- 1. `DateTime.Now.ToString("yy - MM - dd");`
 - 处理后：07-07-17
- 2. `DateTime.Now.ToString("yy年MM月dd日");`
 - 处理后：07年07月17日（中文样式）

i 注：

格式	说明
d	月中的某一天。一位数的日期没有前导零
dd	月中的某一天。一位数的日期有一个前导零
ddd	周中某天的缩写名称，在 <code>AbbreviatedDayNames</code> 中定义
dddd	周中某天的完整名称，在 <code>DayNames</code> 中定义
M	月份数字。一位数的月份没有前导零
MM	月份数字。一位数的月份有一个前导零
MMM	月份的缩写名称，在 <code>AbbreviatedMonthNames</code> 中定义
MMMM	月份的完整名称，在 <code>MonthNames</code> 中定义
y	不包含纪元的年份。如果不包含纪元的年份小于 10，则显示不具有前导零的年份
yy	不包含纪元的年份。如果不包含纪元的年份小于 10，则显示具有前导零的年份
yyyy	包括纪元的四位数的年份
h	12 小时制的小时。一位数的小时数没有前导零
hh	12 小时制的小时。一位数的小时数有前导零
H	24 小时制的小时。一位数的小时数没有前导零
HH	24 小时制的小时。一位数的小时数有前导零
m	分钟。一位数的分钟数没有前导零
mm	分钟。一位数的分钟数有一个前导零
s	秒。一位数的秒数没有前导零
ss	秒。一位数的秒数有一个前导零



网上收集的DateTime技巧

大家在做报表或查询的时候都会有给用户预设一些可选的日期范围，如本年度销售额、本季度利润、本月新增客户

C#里内置的DateTime基本上都可以实现这些功能，巧用DateTime会使你处理这些事来变轻松多了

- 今天

```
1 DateTime.Now.Date.ToShortDateString();
```

- 昨天，就是今天的日期减一

```
1 DateTime.Now.AddDays(-1).ToShortDateString();
```

- 明天，同理，加一

```
1 DateTime.Now.AddDays(1).ToShortDateString();
```

- 本周(要知道本周的第一天就得先知道今天是星期几，从而得知本周的第一天就是几天前的那一天，要注意的是这里的每一周是从周日始至周六止

```
1 DateTime.Now.AddDays(Convert.ToDouble((0 -  
Convert.ToInt16(DateTime.Now.DayOfWeek))))).ToShortDateString();  
2 DateTime.Now.AddDays(Convert.ToDouble((6 -  
Convert.ToInt16(DateTime.Now.DayOfWeek))))).ToShortDateString();
```

- 如果你还不明白，再看一下中文显示星期几的方法就应该懂了
- 由于DayOfWeek返回的是数字的星期几，我们要把它转换成汉字方便我们阅读，有些人可能会用switch来一个一个个地对照，其实不用那么麻烦的

```
1 string[] Day = new string[] { "星期日", "星期一", "星期二", "星期三", "星期四", "星期五", "星期六" };  
2 Day[Convert.ToInt16(DateTime.Now.DayOfWeek)];
```

- 上周，同理，一个周是7天，上周就是本周再减去7天，下周也是一样

```
1 DateTime.Now.AddDays(Convert.ToDouble((0 - Convert.ToInt16(DateTime.Now.DayOfWeek))) -  
7).ToShortDateString();  
2 DateTime.Now.AddDays(Convert.ToDouble((6 - Convert.ToInt16(DateTime.Now.DayOfWeek))) -  
7).ToShortDateString();
```

- 下周

```
1 DateTime.Now.AddDays(Convert.ToDouble((0 - Convert.ToInt16(DateTime.Now.DayOfWeek))) + 7).ToShortDateString();
2 DateTime.Now.AddDays(Convert.ToDouble((6 - Convert.ToInt16(DateTime.Now.DayOfWeek))) + 7).ToShortDateString();
```

- 本月,很多人都会说本月的第一天嘛肯定是1号,最后一天就是下个月一号再减一天。当然这是对的

- 一般的写法

```
1 DateTime.Now.Year.ToString() + DateTime.Now.Month.ToString() + "1"; //第一天
2 DateTime.Parse(DateTime.Now.Year.ToString() + DateTime.Now.Month.ToString() + "1").AddMonths(1).AddDays(-1).ToShortDateString(); //最后一天
```

- 巧用C#里ToString的字符格式化更简便

```
1 DateTime.Now.ToString("yyyy-MM-01");
2 DateTime.Parse(DateTime.Now.ToString("yyyy-MM-01")).AddMonths(1).AddDays(-1).ToShortDateString();
```

- 上个月, 减去一个月份

```
1 DateTime.Parse(DateTime.Now.ToString("yyyy-MM-01")).AddMonths(-1).ToShortDateString();
2 DateTime.Parse(DateTime.Now.ToString("yyyy-MM-01")).AddDays(-1).ToShortDateString();
```

- 下个月, 加去一个月份

```
1 DateTime.Parse(DateTime.Now.ToString("yyyy-MM-01")).AddMonths(1).ToShortDateString();
2 DateTime.Parse(DateTime.Now.ToString("yyyy-MM-01")).AddMonths(2).AddDays(-1).ToShortDateString();
```

- 7天后

```
1 DateTime.Now.Date.ToShortDateString();
2 DateTime.Now.AddDays(7).ToShortDateString();
```

- 7天前

```
1 DateTime.Now.AddDays(-7).ToShortDateString();
2 DateTime.Now.Date.ToShortDateString();
```

- 本年度, 用ToString的字符格式化我们也很容易地算出本年度的第一天和最后一天

```
1 DateTime.Parse(DateTime.Now.ToString("yyyy-01-01")).ToShortDateString();
2 DateTime.Parse(DateTime.Now.ToString("yyyy-01-01")).AddYears(1).AddDays(-1).ToShortDateString();
```

- 上年度, 不用再解释了吧

```
1 DateTime.Parse(DateTime.Now.ToString("yyyy-01-01")).AddYears(-1).ToShortDateString();
2 DateTime.Parse(DateTime.Now.ToString("yyyy-01-01")).AddDays(-1).ToShortDateString();
```

- 下年度

```
1 DateTime.Parse(DateTime.Now.ToString("yyyy-01-01")).AddYears(1).ToShortDateString();
2 DateTime.Parse(DateTime.Now.ToString("yyyy-01-01")).AddYears(2).AddDays(-1).ToShortDateString();
```

- 本季度，很多人都会觉得这里难点，需要写个长长的过程来判断。其实不用的，我们都知道一年四个季度，一个季度三个月
- 首先我们先把日期推到本季度第一个月，然后这个月的第一天就是本季度的第一天了

```
1 DateTime.Now.AddMonths(0 - ((DateTime.Now.Month - 1) % 3)).ToString("yyyy-MM-01");
```

- 同理，本季度的最后一天就是下季度的第一天减一

```
1 DateTime.Parse(DateTime.Now.AddMonths(3 - ((DateTime.Now.Month - 1) % 3)).ToString("yyyy-MM-01")).AddDays(-1).ToShortDateString();
```

- 下季度

```
1 DateTime.Now.AddMonths(3 - ((DateTime.Now.Month - 1) % 3)).ToString("yyyy-MM-01");
2 DateTime.Parse(DateTime.Now.AddMonths(6 - ((DateTime.Now.Month - 1) % 3)).ToString("yyyy-MM-01")).AddDays(-1).ToShortDateString();
```

- 上季度

```
1 DateTime.Now.AddMonths(-3 - ((DateTime.Now.Month - 1) % 3)).ToString("yyyy-MM-01");
2 DateTime.Parse(DateTime.Now.AddMonths(0 - ((DateTime.Now.Month - 1) % 3)).ToString("yyyy-MM-01")).AddDays(-1).ToShortDateString();
```

格式化数值：有时，我们可能需要将数值以一定的格式来呈现，就需要对数值进行格式化。我们使用格式字符串指定格式。格式字符串采用以下形式：**Axx**，其中 **A** 为格式说明符，指定格式化类型，**xx** 为精度说明符，控制格式化输出的有效位数或小数位数。

格式说明符	说明	示例	输出
C	货币	2.5.ToString("C")	¥ 2.50
D	十进制数	25.ToString("D5")	00025
E	科学型	25000.ToString("E")	2.500000E+005
F	固定点	25.ToString("F2")	25.00
G	常规	2.5.ToString("G")	2.5
N	数字	2500000.ToString("N")	2,500,000.00
X	十六进制	255.ToString("X")	FF



十六进制(hex)显示 in C# `.ToString("X2")`

```
byte tempbyte=0xaa;
```

```
MessageBox.Show (tempbyte.ToString("X2"));
```