

# Perl 时间日期

本章节我们为大家介绍 Perl 语言对时间日期的处理。

Perl中处理时间的函数有如下几种：

- 1、**time()** 函数：返回从1970年1月1日起累计的秒数
- 2、**localtime()** 函数：获取本地时区时间
- 3、**gmtime()** 函数：获取格林威治时间

## 当前时间和日期

接下来让我们看下 **localtime()** 函数，该函数在没有参数的情况下返回当前的时间和日期。

以下 9 个符号代表不同的时间日期参数：

sec,	# 秒, 0 到 61
min,	# 分钟, 0 到 59
hour,	# 小时, 0 到 24
mday,	# 天, 1 到 31
mon,	# 月, 0 到 11
year,	# 年, 从 1900 开始
wday,	# 星期几, 0-6, 0表示周日
yday,	# 一年中的第几天, 0-364, 365
isdst	# 如果夏令时有效, 则为真

实例演示如下：

### 实例

```
#!/usr/bin/perl
@months = qw( 一月 二月 三月 四月 五月 六月 七月 八月 九月 十月 十一月 十二月 );
@days = qw( 星期天 星期一 星期二 星期三 星期四 星期五 星期六 );
($sec,$min,$hour,$mday,$mon,$year,$wday,$yday,$isdst) = localtime();
print "$mday $months[$mon] $days[$wday]\n";
```

以上实例执行输出结果为：

```
12 六月 星期天
```

如果直接调用 **localtime()**，它返回系统当期设置时区的时间，实例如下：

### 实例

```
#!/usr/bin/perl
$datestring = localtime();
print "时间日期为: $datestring\n";
```

以上实例执行输出结果为：

```
时间日期为：Sun Jun 12 11:27:31 2016
```

## 格林威治时间 (GMT)

函数 **gmtime()** 与 **localtime()** 类似，但它返回标准格林威治时间。

### 实例

```
#!/usr/bin/perl
$local_datestring = localtime();
print "本地时间日期为：$local_datestring\n";
$gmt_datestring = gmtime();
print "GMT 时间日期为：$gmt_datestring\n";
```

以上实例执行输出结果为：

```
本地时间日期为：Sun Jun 12 11:32:14 2016
GMT 时间日期为：Sun Jun 12 03:32:14 2016
```

从实例中我们可以看出，中国的时间和格林威治时间相差了8小时。

## 格式化日期和时间

我们可以使用 **localtime()** 函数的 9 个时间元素来输出需要制定的格式时间。格式化输出使用 **printf()** 函数：

### 实例

```
#!/usr/bin/perl
($sec,$min,$hour,$mday,$mon,$year,$wday,$yday,$isdst) = localtime();
printf("格式化时间：HH:MM:SS\n");
printf("%02d:%02d:%02d", $hour, $min, $sec);
```

以上实例执行输出结果为：

```
格式化时间：HH:MM:SS
11:35:23
```

## 新纪元时间(Epoch Time)

我们可以使用 **time()** 函数来获取新纪元时间，该函数返回从1970年1月1日起累计的秒数。实例如下：

### 实例

```
#!/usr/bin/perl
$epoc = time();
print "从1970年1月1日起累计的秒数为：$epoc\n";
```

以上实例执行输出结果为：

```
从1970年1月1日起累计的秒数为：1465702883
```

我们可以输出一个我们想要的时间格式：

实例

```
#!/usr/bin/perl
($sec,$min,$hour,$mday,$mon,$year,$wday,$yday,$isdst) = localtime();
print "当前时间和日期：";
printf("%d-%d-%d %d:%d:%d",$year+1900,$mon+1,$mday,$hour,$min,$sec);
print "\n";
$epoc = time();
$epoc = $epoc - 24 * 60 * 60; # 一天前的时间秒数
($sec,$min,$hour,$mday,$mon,$year,$wday,$yday,$isdst) = localtime($epoc);
print "昨天时间和日期：";
printf("%d-%d-%d %d:%d:%d",$year+1900,$mon+1,$mday,$hour,$min,$sec);
print "\n";
```

以上实例执行输出结果为：

```
当前时间和日期：2017-3-15 12:47:54
昨天时间和日期：2017-3-14 12:47:54
```

POSIX 函数 strftime()

函数 strftime() 可以将时间格式化为我们想要的格式。

下表列出了一些格式化的符号，\* 号表示想要依赖本地时间：

符号	描述	实例
%a	星期几的简称 ( Sun..Sat ) *	Thu
%A	星期几的全称 ( Sunday..Saturday ) *	Thursday
%b	月的简称 ( Jan..Dec ) *	Aug
%B	月的全称 ( January..December ) *	August
%c	日期和时间 *	Thu Aug 23 14:55:02 2001
%C	年份除于100，并取整 (00-99)	20
%d	一个月的第几天 (01-31)	23
%D	日期, MM/DD/YY 相等于 %m/%d/%y	08/23/01

%e	一个月的第几天，使用空格填充个位数 ( 1-31)	23
%F	YYYY-MM-DD 的简写类似于 %Y-%m-%d	2001-08-23
%g	年份的最后两位数 (00-99)	01
%g	年	2001
%h	月的简称 * (和%b选项相同)	Aug
%H	24 小时制 (00-23)	14
%I	12 小时制 (01-12)	02
%j	一年的第几天 (001-366)	235
%m	月 (01-12)	08
%M	分钟 (00-59)	55
%n	新行 ('\n')	
%p	显示出AM或PM	PM
%r	时间 ( hh : mm : ss AM或PM ) ， 12小时 *	02:55:02 pm
%R	24 小时 HH:MM 时间格式, 等于 %H:%M	14:55
%S	秒数 (00-61)	02
%t	水平制表符 ('\t')	
%T	时间 ( 24小时制 ) ( hh:mm:ss ) ， 等于 %H:%M:%S	14:55
%u	ISO 8601 的星期几格式，星期一为1 (1-7)	4
%U	一年中的第几周，星期天为第一天(00-53)	33
%V	ISO 8601 第几周 (00-53)	34
%w	一个星期的第几天 ( 0代表星期天 ) (0-6)	4
%W	一年的第几个星期，星期一为第一天 (00-53)	34
%x	显示日期的格式 ( mm/dd/yy ) *	08/23/01

%X	显示时间格式 *	14:55:02
%y	年, 两位数 (00-99)	01
%Y	年	2001
%z	ISO 8601与UTC的时区偏移(1 minute=1, 1 hour=100)	+100
%Z	当前时区的名称,如"中国标准时间" *	CDT
%%	% 符号	%

实例

```
#!/usr/bin/perl
use POSIX qw(strftime);
$datestring = strftime "%Y-%m-%d %H:%M:%S", localtime;
printf("时间日期 - $datestring\n");
# GMT 格式化时间日期
$datestring = strftime "%Y-%m-%d %H:%M:%S", gmtime;
printf("时间日期 - $datestring\n");
```

以上实例执行输出结果为：

```
时间日期 - 2016-06-12 12:15:13
时间日期 - 2016-06-12 04:15:13
```

← Perl 运算符

Perl 子程序(函数) →

 点我分享笔记