PERL - DATE & TIME

http://www.tutorialspoint.com/perl/perl_date_time.htm

This chapter will give you the basic understanding on how to process and manipulate dates and times in Perl.

Current Date & Time

Let's start with **localtime** function, which returns values for the current date and time if given no arguments. Following is the 9-element list returned by the localtime function while using in list context −

```
sec,     # seconds of minutes from 0 to 61
min,     # minutes of hour from 0 to 59
hour,    # hours of day from 0 to 24
mday,    # day of month from 1 to 31
mon,     # month of year from 0 to 11
year,    # year since 1900
wday,    # days since sunday
yday,    # days since January 1st
isdst    # hours of daylight savings time
```

Try the following example to print different elements returned by localtime function −

```
#!/usr/local/bin/perl

@months = qw( Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec );
@days = qw(Sun Mon Tue Wed Thu Fri Sat Sun);

($sec,$min,$hour,$mday,$mon,$year,$wday,$yday,$isdst) = localtime();
print "$mday $months[$mon] $days[$wday]\n";
```

When the above code is executed, it produces the following result −

```
16 Feb Sat
```

If you will use localtime function in scalar context, then it will return date and time from the current time zone set in the system. Try the following example to print current date and time in full format −

```
#!/usr/local/bin/perl

$datestring = localtime();
print "Local date and time $datestring\n";
```

When the above code is executed, it produces the following result −

```
Local date and time Sat Feb 16 06:50:45 2013
```

GMT Time

The function **gmtime** works just like localtime function but the returned values are localized for the standard Greenwich time zone. When called in list context, $isdst, the last value returned by gmtime, is always 0 . There is no Daylight Saving Time in GMT.

You should make a note on the fact that localtime will return the current local time on the machine that runs the script and gmtime will return the universal Greenwich Mean Time, or GMT *orUTC*.

Try the following example to print the current date and time but on GMT scale −

```
#!/usr/local/bin/perl
```

```
$datestring = gmtime();
print "GMT date and time $datestring\n";
```

When the above code is executed, it produces the following result −

```
GMT date and time Sat Feb 16 13:50:45 2013
```

Format Date & Time:

You can use localtime function to get a list of 9-elements and later you can use the **printf** function to format date and time based on your requirements as follows −

```
#!/usr/local/bin/perl

($sec,$min,$hour,$mday,$mon,$year,$wday,$yday,$isdst) = localtime();

printf("Time Format - HH:MM:SS\n");
printf("%02d:%02d:%02d", $hour, $min, $sec);
```

When the above code is executed, it produces the following result −

```
Time Format - HH:MM:SS
06:58:52
```

Epoch time

You can use the time function to get epoch time, i.e. the numbers of seconds that have elapsed since a given date, in Unix is January 1, 1970.

```
#!/usr/local/bin/perl

$epoc = time();

print "Number of seconds since Jan 1, 1970 - $epoc\n";
```

When the above code is executed, it produces the following result −

```
Number of seconds since Jan 1, 1970 - 1361022130
```

You can convert a given number of seconds into date and time string as follows −

```
#!/usr/local/bin/perl

$datestring = localtime();
print "Current date and time $datestring\n";

$epoc = time();
$epoc = $epoc - 12 * 60 * 60;   # one day before of current date.

$datestring = localtime($epoc);
print "Yesterday's date and time $datestring\n";
```

When the above code is executed, it produces the following result −

```
Current date and time Sat Feb 16 07:05:39 2013
Yesterday's date and time Fri Feb 15 19:05:39 2013
```

POSIX Function strftime

You can use the POSIX function **strftime** to format date and time with the help of the following table. Please note that the specifiers marked with an asterisk ∗ are locale-dependent.

Specifier	Replaced by	Example
%a	Abbreviated weekday name *	Thu
%A	Full weekday name *	Thursday
%b	Abbreviated month name *	Aug
%B	Full month name *	August
%c	Date and time representation *	Thu Aug 23 14:55:02 2001
%C	Year divided by 100 and truncated to integer (00-99)	20
%d	Day of the month, zero-padded (01-31)	23
%D	Short MM/DD/YY date, equivalent to %m/%d/%y	08/23/01
%e	Day of the month, space-padded (1-31)	23
%F	Short YYYY-MM-DD date, equivalent to %Y-%m-%d	2001-08-23
%g	Week-based year, last two digits (00-99)	01
%g	Week-based year	2001
%h	Abbreviated month name * (same as %b)	Aug
%H	Hour in 24h format (00-23)	14
%I	Hour in 12h format (01-12)	02
%j	Day of the year (001-366)	235
%m	Month as a decimal number (01-12)	08
%M	Minute (00-59)	55
%n	New-line character ('\n')	
%p	AM or PM designation	PM
%r	12-hour clock time *	02:55:02 pm
%R	24-hour HH:MM time, equivalent to %H:%M	14:55
%S	Second (00-61)	02
%t	Horizontal-tab character ('\t')	
%T	ISO 8601 time format (HH:MM:SS), equivalent to %H:%M:%S	14:55
%u	ISO 8601 weekday as number with Monday as 1 (1-7)	4
%U	Week number with the first Sunday as the first day of week one (00-53)	33
%V	ISO 8601 week number (00-53)	34
%w	Weekday as a decimal number with Sunday as 0 (0-6)	4
%W	Week number with the first Monday as the first day of week one (00-53)	34
%x	Date representation *	08/23/01

%X	Time representation *	14:55:02
%y	Year, last two digits (00-99)	01
%Y	Year	2001
%z	ISO 8601 offset from UTC in timezone $1minute = 1, 1hour = 100$	+100
	If timezone cannot be termined, no characters	
%Z	Timezone name or abbreviation *	CDT
	If timezone cannot be termined, no characters	
%%	A % sign	%

Let's check the following example to understand the usage −

```
#!/usr/local/bin/perl
use POSIX qw(strftime);

$datestring = strftime "%a %b %e %H:%M:%S %Y", localtime;
printf("date and time - $datestring\n");

# or for GMT formatted appropriately for your locale:
$datestring = strftime "%a %b %e %H:%M:%S %Y", gmtime;
printf("date and time - $datestring\n");
```

When the above code is executed, it produces the following result −

```
date and time - Sat Feb 16 07:10:23 2013
date and time - Sat Feb 16 14:10:23 2013
```

Loading [MathJax]/jax/output/HTML-CSS/jax.js