Тема 3.9

«Функции PHP по работе с датой и временем»

Вступление

Обработка даты и времени – важная задача для многих приложений. В PHP для её решения существует несколько простых и мощных функций, которые мы сейчас рассмотрим.

Unixtime

Для начала – немного информации о том, что такое Unixtime, в котором PHP работает с датой и временем.

Unixtime – это значение времени, выраженное в количестве секунд, прошедших с 1 января 1970 года 00:00:00.

Unixtime – integer, а потому прекрасно переносится между различными операционными системами и аппаратными платформами, операции с ним выполняются быстро и удобно.

К недостаткам Unixtime относится то, что его максимальное значение в 32-битных платформах не очень велико. Однако, с переходом на 64-битные платформы эта проблема отпадает автоматически.

Получение текущего значения времени

int time (void) – возвращает текущее значение времени в формате Unixtime.

Внимание! Эта функция возвращает текущее значение времени НА СЕРВЕРЕ. Т.е., например, если вы сидите в Минске, а сервер размещён в Нью-Йорке, вы получите нью-йоркское время, а не минское!

Пример:

```
echo time(); // 1232995956
```

Получение текущего значения времени

mixed microtime ([bool get_as_float]) – возвращает текущее значение времени с микросекундами.

Эта функция доступна только на операционных системах, в которых есть системная функция gettimeofday().

При вызове без необязательного параметра, возвращается строка в формате "msec sec", где sec – время в формате Unixtime, а msec – это дробная часть (микросекунды).

Если передан аргумент get_as_float, равный TRUE, функция microtime() возвращает дробное число.

Замечание: аргумент get_as_float появился в PHP 5.0.0.

Пример:

```
function getmicrotime()
 {
  list($usec, $sec) = explode(" ", microtime());
  return ((float)$usec + (float)$sec);
 }
$time_start = getmicrotime();
```

Эту функцию удобно использовать для определения времени выполнения тех или иных операций.

Unixtime «в человеческом виде»

Поскольку человеку «не очень удобно» воспринимать время в виде одного целого числа, в PHP есть функция, позволяющая преобразовать это число в удобный для человека (или для некоторых программ) формат.

string date (string format [, int timestamp]) – возвращает время, отформатированное в соответствии с аргументом format, используя метку времени, заданную аргументом timestamp или текущее системное время, если аргумент timestamp не задан.

Замечание: для большинства операционных систем допустимыми являются даты с 13 декабря 1901, 20:45:54 GMT по 19 января 2038, 03:14:07 GMT. Эти даты соответствуют минимальному и максимальному значению 32-битового целого со знаком. Для Windows допустимы даты с 01-01-1970 по 19-01-2038.

Аргумент format представляет собой строку, в которой перечисленные ниже значения трактуются как соответствующие «плейсхолдеры» (placeholders) (если это понятие вам не знакомо, – подождите, мы рассмотрим его в теме, посвящённой разделению дизайна и кода).

Unixtime «в человеческом виде»

Символ в строке format	Описание	Пример возвращаемого значения
a	Ante meridiem или Post meridiem в нижнем регистре	am или pm
A	Ante meridiem или Post meridiem в верхнем регистре	AM или PM
B	Время в стадарте Swatch Internet	От 000 до 999
c	Дата в формате ISO 8601 (добавлено в PHP 5)	2004-02-12T15:19:21+00:00
d	День месяца, 2 цифры с ведущими нулями	от 01 до 31
D	Сокращенное наименование дня недели, 3 символа	от Mon до Sun
F	Полное наименование месяца, например January или March	от January до December
g	Часы в 12-часовом формате без ведущих нулей	От 1 до 12
G	Часы в 24-часовом формате без ведущих нулей	От 0 до 23
h	Часы в 12-часовом формате с ведущими нулями	От 01 до 12
H	Часы в 24-часовом формате с ведущими нулями	От 00 до 23
i	Минуты с ведущими нулями	00 to 59
I (заглавная i)	Признак летнего времени	1, если дата соответствует летнему времени, иначе 0 otherwise.
j	День месяца без ведущих нулей	От 1 до 31
l (строчная 'L')	Полное наименование дня недели	От Sunday до Saturday
L	Признак високосного года	1, если год високосный, иначе 0.
m	Порядковый номер месяца с ведущими нулями	От 01 до 12
M	Сокращенное наименование месяца, 3 символа	От Jan до Dec

Unixtime «в человеческом виде»

Символ в строке format	Описание	Пример возвращаемого значения
n	Порядковый номер месяца без ведущих нулей	От 1 до 12
O	Разница с временем по Гринвичу в часах	Например: +0200
r	Дата в формате RFC 2822	Например: Thu, 21 Dec 2000 16:01:07 +0200
s	Секунды с ведущими нулями	От 00 до 59
S	Английский суффикс порядкового числительного дня месяца, 2 символа	st, nd, rd или th. Применяется совместно с j
t	Количество дней в месяце	От 28 до 31
T	Временная зона на сервере	Примеры: EST, MDT ...
U	Unixtime	См. описание функции time()
w	Порядковый номер дня недели	От 0 (воскресенье) до 6 (суббота)
W	Порядковый номер недели года по ISO-8601, первый день недели - понедельник (добавлено в PHP 4.1.0)	Например: 42 (42-я неделя года)
Y	Порядковый номер года, 4 цифры	Примеры: 1999, 2003
y	Номер года, 2 цифры	Примеры: 99, 03
z	Порядковый номер дня в году (нумерация с 0)	От 0 до 365
Z	Смещение временной зоны в секундах. Для временных зон западнее UTC это отрицательное число, восточнее UTC - положительное.	От -43200 до 43200

Unixtime «в человеческом виде»

Пример:

```
echo date (“Y.m.d H:i:s”); // 2009.01.26 21:03:34
```

Примечание: чтобы не рисковать, в качестве формата представления времени лучше передавать только соответствующие буквы и разделители, а всё остальное «приклеить» вне аргумента функции date().

Пример:

```
echo “Сегодня ”.date(“Y.m.d”).” время ”.date (“H:i:s”);
```

«Человеческое время» в Unixtime

Для обратного преобразования (из «человеческого» представления времени в Unixtime) существует функция mktime().

int mktime ([int hour [, int minute [, int second [, int month [, int day [, int year [, int is_dst]]]]]]]) – возвращает Unixtime, соответствующий указанным дате и времени.

Аргументы этой функции могут быть опущены в порядке справа налево. В этом случае их значения по умолчанию равны соответствующим компонентам локальной даты/времени.

Аргумент is_dst может быть установлен в 1, если заданной дате соответствует летнее время, 0 в противном случае, или -1 (значение по умолчанию), если неизвестно, действует ли летнее время на заданную дату. В последнем случае PHP пытается определить это самостоятельно.

Начиная с версии 5.1.0 этот параметр более не рекомендуется к использованию. Вместо этого рекомендуется устанавливать соответствующую временную зону.

«Человеческое время» в Unixtime

Функцию mktime() удобно использовать для выполнения арифметических операций с датами, так как она вычисляет верные значения при некорректных аргументах.

Например, в следующем примере каждая строка выведет "Jan-01-1998".

Пример:

```
echo date("M-d-Y", mktime(0, 0, 0, 12, 32, 1997));
echo date("M-d-Y", mktime(0, 0, 0, 13, 1, 1997));
echo date("M-d-Y", mktime(0, 0, 0, 1, 1, 1998));
echo date("M-d-Y", mktime(0, 0, 0, 1, 1, 98));
```

Параметр year может быть двух- или четырёхзначным числом. Значения от 0 до 69 соответствуют 2000-2069, а 70-99 соответствуют 1970-1999 (в большинстве современных систем, где время представляется 32-битным целым со знаком, допустимыми являются значения year между 1901 и 2038).

Внимание: до PHP 5.1.0 отрицательные метки времени не поддерживались под всеми системами Windows. Следовательно, допустимые значения параметра year были между 1970 и 2038.

«Человеческое время» в Unixtime

Последний день любого месяца можно вычислить как «нулевой» день следующего месяца. Оба приведённых ниже примера выведут "Последний день в феврале 2000 г: 29".

Пример:

```
$lastday = mktime(0, 0, 0, 3, 0, 2000);
echo "Последний день в феврале 2000 г: ". $lastday;
$lastday = mktime(0, 0, 0, 4, -31, 2000);
echo "Последний день в феврале 2000 г: ". $lastday;
```

«Человеческое время» по-русски

string strftime (string format [, int timestamp]) – возвращает строку, отформатированную в соответствии с аргументом format, используя аргумент timestamp или текущее системное время, если этот аргумент не передан.

Названия месяцев, дней недели и другие строки, зависящие от языка, соответствуют текущей локали, установленной функцией setlocale().

В форматирующей строке распознаются следующие символы:

%a - сокращённое название дня недели в текущей локали;

%A - полное название дня недели в текущей локали;

%b - сокращенное название месяца недели в текущей локали;

%B - полное название месяца недели в текущей локали;

%c - предпочтительный формат даты и времени в текущей локали;

«Человеческое время» по-русски

%C - столетие (год, делённый на 100 и округлённый до целого, от 00 до 99);

%d - день месяца в виде десятичного числа (от 01 до 31);

%D - аналогично %m/%d/%y;

%e - день месяца в виде десятичного числа (если это одна цифра, то перед ней добавляется пробел (от ' 1' до '31'))

%g - подобно %G, но без столетия;

%G - год, 4-значное число, соответствующее номеру недели по ISO (см. %V);

%h - аналогично %b;

%H - номер часа от 00 до 23;

%I - номер часа от 01 до 12;

%j - номер дня в году (от 001 до 366);

%m - номер месяца (от 01 до 12);

%M – минуты;

%n - символ \n;

%p - am или pm, или соответствующие строки в текущей локали;

«Человеческое время» по-русски

%r - время в формате a.m. или p.m.;

%R - время в 24-часовом формате;

%S – секунды;

%t - символ табуляции \t;

%T - текущее время, аналогично %H:%M:%S;

%u - номер дня недели от 1 до 7, где 1 соответствует понедельнику;

%U - порядковый номер недели в текущем году, первым днём первой недели в году считается первое воскресенье года;

%V - порядковый номер недели в году по стандарту ISO 8601:1988 от 01 до 53, где 1 соответствует первой неделе в году, в которой как минимум 4 дня принадлежат этому году;

%W - порядковый номер недели в текущем году, первым днём первой недели в году считается первый понедельник года;

%w - номер дня недели, 0 соответствует воскресенью;

%x - предпочтительный формат даты без времени в текущей локали;

«Человеческое время» по-русски

%X - предпочтительный формат времени без даты в текущей локали;

%y - год без столетия (от 00 до 99);

%Y - год, включая столетие;

%Z - временная зона в виде смещения, аббревиатуры или полного наименования;

%% - символ %.

Замечание: strftime() использует функции операционной системы, поэтому отдельные форматирующие символы могут не работать в вашей операционной системе. Кроме того, не все платформы поддерживают отрицательные метки времени. Это значит, что %e, %T, %R и %D (а возможно и другие) и даты до Jan 1, 1970 не поддерживаются Windows, некоторыми версиями Linux и некоторыми другими операционными системами.

Список форматирующих символов, поддерживаемых Windows, можно найти на сайте MSDN.

«Человеческое время» по-русски

Пример использования функции strftime() с разными локалями:

```
setlocale(LC_TIME, "C");
echo strftime("%A");
setlocale(LC_TIME, "fi_FI");
echo strftime(" по-фински - %A,");
setlocale(LC_TIME, "fr_FR");
echo strftime(" по-французски - %A и");
setlocale(LC_TIME, "de_DE");
echo strftime(" по-немецки - %A.\n");
```

Этот пример будет работать, если на вашей системе установлены соответствующие локали.

Внимание: поскольку с локалями до сих пор существует много проблем, значительно надёжнее будет работать скрипт, в котором вы получите числовое значение интересующих вас величин и сами «вручную» преобразуете его к соответствующему словесному представлению.

Проверка даты

bool checkdate (int month, int day, int year) – возвращает TRUE если дата, заданная аргументами, является правильной; иначе возвращает FALSE.

Дата считается правильной, если:

- год лежит в диапазоне от 1 до 32767 включительно;
- месяц лежит в диапазоне от 1 до 12 включительно;
- день является допустимым номером дня для месяца, заданного аргументом month, принимая во внимание, что year может задавать високосный год.

Пример:

```
var_dump(checkdate(12, 31, 2000)); // bool(true)
var_dump(checkdate(2, 29, 2001)); // bool(false)
```

Восход, заход и временная зона

mixed date_sunrise (int timestamp [, int format [, float latitude [, float longitude [, float zenith [, float gmt_offset]]]]]) – возвращает время восхода солнца для указанного в формате Unixtime дня и указанного широтой и долготой места.

Список параметров:

timestamp – дата в формате Unixtime;

format – формат возвращаемого значения:

- SUNFUNCS_RET_STRING – строка: 16:46
- SUNFUNCS_RET_DOUBLE – дробное число: 16.78243132
- SUNFUNCS_RET_TIMESTAMP – целое число: 1095034606

latitude – широта;

longitude – долгота;

zenith – высота солнца над горизонтом;

gmtoffset – смещение временной зоны от Гринвича (в часах).

Восход, заход и временная зона

Пример:

```
/* Широта: 38.4 (север)
Долгота: 9 (запад)
Зенит: ~= 90
Смещение от Гринвича: +1 */
echo date_sunrise(time(), SUNFUNCS_RET_STRING, 38.4, -9, 90, 1);
```

mixed date_sunset (int timestamp [, int format [, float latitude [, float longitude [, float zenith [, float gmt_offset]]]]]) – возвращает время захода солнца для указанного в формате Unixtime дня и указанного широтой и долготой места. В остальном аналогична date_sunrise().

Восход, заход и временная зона

bool date_default_timezone_set (string timezone_identifier) – устанавливает временну**ю** зону, используемую по умолчанию всеми функциями работы с датой и временем.

Временну**ю** зону также можно указать директивой date.timezone в конфигурационном файле PHP.

Параметр timezone_identifier является значением временн**о**й зоны, например «UTC» или «Europe/Lisbon». См. список зон в документации по PHP.

Эта функция возвращает TRUE, если передана корректная временн**а**я зона, и FALSE в противном случае.

string date_default_timezone_get (void) – возвращает текущую временну**ю** зону, пытаясь определить её в следующем порядке:

- зона, установленная с помощью date_default_timezone_set();
- переменная окружения «TZ»;
- директива date.timezone в php.ini:
- «магическое угадывание», зависящее от операционной системы;
- если ничто из вышеназванного не помогло, возвращает «UTC».

Приостановка выполнения

int sleep (int seconds) – заставляет скрипт «заснуть» на указанное количество секунд.

Пример:

```
echo date('h:i:s'); // 21:31:23
sleep(10);
echo date('h:i:s'); // 21:31:33
```

void usleep (int micro_seconds) – заставляет скрипт «заснуть» на указанное количество микросекунд.

```
echo date('h:i:s'); // 21:13:28
usleep(2000000);
echo date('h:i:s'); // 21:13:30
```