



Array Functions

array_diff (arr1, arr2 ...)
array_filter (arr, function)
array_flip (arr)
array_intersect (arr1, arr2 ...)
array_merge (arr1, arr2 ...)
array_pop (arr)
array_push (arr, var1, var2 ...)
array_reverse (arr)
array_search (needle, arr)
array_walk (arr, function)
count (count)
in_array (needle, haystack)

String Functions

crypt (str, salt)
explode (sep, str)
implode (glue, arr)
nl2br (str)
sprintf (frmt, args)
strip_tags (str, allowed_tags)
str_replace (search, replace, str)
strpos (str, needle)
strrev (str)
strstr (str, needle)
strtolower (str)
strtoupper (str)
substr (string, start, len)

Filesystem Functions

clearstatcache ()
copy (source, dest)
fclose (handle)
fgets (handle, len)
file (file)
filetime (file)
filesize (file)
file_exists (file)
fopen (file, mode)
fread (handle, len)
fwrite (handle, str)
readfile (file)

fopen() Modes

r	Read
r+	Read and write, prepend
w	Write, truncate
w+	Read and write, truncate
a	Write, append
a+	Read and write, append

Regular Expression Functions

ereg (pattern, str)
split (pattern, str)
ereg_replace (pattern, replace, str)
preg_grep (pattern, arr)
preg_match (patten, str)
preg_match_all (pattern, str, arr)
preg_replace (pattern, replace, str)
preg_split (pattern, str)

Regular Expressions Syntax

^	Start of string
\$	End of string
.	Any single character
(a b)	a or b
(...)	Group section
[abc]	Item in range (a, b or c)
[^abc]	Not in range (not a, b or c)
\s	White space
a?	Zero or one of a
a*	Zero or more of a
a*?	Zero or more of a, ungreedy
a+	One or more of a
a+?	One or more of a, ungreedy
a{3}	Exactly 3 of a
a{3,}	3 or more of a
a{3,6}	3 to 6 of a
a{3,6}?	3 to 6 of a, ungreedy
\	Escape character
[[:punct:]]	Any punctuation symbol
[[:space:]]	Any space character
[[:blank:]]	Space or tab

PCRE Modifiers

i	Case-insensitive
s	Period matches newline
m	^ and \$ match lines
U	Ungreedy matching
e	Evaluate replacement
x	Pattern over several lines

Date and Time Functions

date (format, timestamp)
mktime (hr, min, sec, month, day, yr)
strtotime (str)
time ()

Available free from
AddedBytes.com

Date Formatting

Y	4 digit year (2008)
y	2 digit year (08)
F	Long month (January)
M	Short month (Jan)
m	Month ⁴ (01 to 12)
n	Month (1 to 12)
D	Short day name (Mon)
l	Long day name (Monday)
d	Day ⁴ (01 to 31)
j	Day (1 to 31)
h	12 Hour ⁴ (01 to 12)
g	12 Hour (1 ro 12)
H	24 Hour ⁴ (00 to 23)
G	24 Hour (0 to 23)
i	Minutes ⁴ (00 to 59)
s	Seconds ⁴ (00 to 59)
w	Day of week ¹ (0 to 6)
z	Day of year (0 to 365)
W	Week of year ² (1 to 53)
t	Days in month (28 to 31)
a	am or pm
A	AM or PM
B	Swatch Internet Time (000 to 999)
S	Ordinal Suffix (st, nd, rd, th)
T	Timezone of machine (GMT)
Z	Timezone offset (seconds)
O	Difference to GMT (hours) (+0200)
I	Daylight saving (1 or 0)
L	Leap year (1 or 0)
U	Seconds since Epoch ³
c	ISO 8601 (PHP 5)
r	RFC 2822

- 0 is Sunday, 6 is Saturday.
- Week that overlaps two years belongs to year that contains most days of that week. Hence week number for 1st January of a given year can be 53 if week belongs to previous year.
`date("W", mktime(0, 0, 0, 12, 8, $year))` always gives correct number of weeks in `$year`.
- The Epoch is the 1st January 1970.
- With leading zeroes