addcslashes — 以 C 语言风格使用反斜线转义字符串中的字符

<?php

echo addcslashes('foo[]','o');

?>

//输出:f\o\o[] 括号内第一个参数表示要转义的字符,第2参数表示转换范围

addslashes — 使用反斜线引用字符串,

包括:单引号（'）双引号（"）反斜杠（\）NULL

<?php

$str = addslashes('Shanghai \is\ the "biggest" city in China.');

echo($str);//Shanghai \\is\\ the \"biggest\" city in China.

?>

bin2hex — 函数把包含数据的二进制字符串转换为十六进制值

字符串可通过使用 pack() 函数再转换回去。

<?php

$kk = "北京";

echo bin2hex($kk) . '<br>'; //e58c97e4baac

echo pack('H\*',bin2hex($kk)).'<br>'//北京

?>

hex2bin — 转换十六进制字符串为二进制字符串

<?php

echo hex2bin("48656c6c6f20576f726c6421").'<br>';//48656c6c6f20576f726c6421

echo bin2hex("Hello World!");//Hello World!

?>

chop — rtrim 的别名,从字符串右端移除字符：

<?php

$qwe="今天天气真好,风吹着\n很舒\n服\n\n";

echo chop($qwe).'<br>';//今天天气真好,风吹着 很舒 服

$qwe2=chop($qwe);

echo chop($qwe2,"很舒服" ).'<br>';//今天天气真好,风吹着 很舒

?>

chr — 通过ASCII值返回指定的字符

<?php

echo chr(65) . "<br>"; // 十进制 A

echo chr(065) . "<br>"; // 八进制值 5

echo chr(0x65) . "<br>"; // 十六进制值 e

?>

ord — 返回字符的 ASCII 码值

<?php

echo ord("A").'<br>';//65

echo ord("a").'<br>';//97

?>

chunk\_split — 将字符串分割成小块 注意:函数不会改变原字符串

语法:chunk\_split(string,length,end)

string 即要分割的字符串

length也可以不写,定义字符块的长度,默认值为76

end:字符串值,定义分割后每个字符串末尾内容,默认\r\n

<?php

$str = "1234567890";

echo chunk\_split($str,2).'<>';//12 34 56 78 90

echo chunk\_split($str,3,"@");//123@456@789@0@

?>

convert\_uudecode — 解码uuencode 编码的字符串

convert\_uuencode — 使用 uuencode 编码一个字符串

<?php

$str = "你好!";

// 对字符串进行编码

$str1 = convert\_uuencode($str);

echo $str1 . "<br>";//'Y+V@Y:6](0`` `

// 对字符串进行解码

$str2 = convert\_uudecode($str1);

echo $str2.'<br>';//你好!

echo $str;//你好!

?>

crypt — 单向字符串散列

crypt() 函数返回使用 DES、Blowfish 或 MD5 算法加密的字符串。

echo — 输出一个或多个字符串

区别在于:echo可以接受参数列表

print — 输出字符串

printf — 输出格式化字符串

<?php

$number = "2";

$str = "北京";

printf("在%s有 %u 百万辆摩拜单车。",$str,$number);

?>

explode — 使用一个字符串分割另一个字符串

<?php

$str = "Hello world. I love China!";

print\_r (explode(" ",$str));

/\*

Array (

[0] => Hello

[1] => world.

[2] => I

[3] => love

[4] => China!

)

\*/

?>

implode — 将一个一维数组的值转化为字符串

join — 别名 implode

<?php

$arr = array('Hello','World!','I','love','China!');

echo implode(" ",$arr);//Hello World! I love China!

?>

fprintf — 将格式化后的字符串写入到流

sprintf — sprintf() 函数把格式化的字符串写入变量中

vfprintf — 将格式化字符串写入流

vprintf — 输出格式化字符串

vsprintf — 返回格式化字符串

html\_entity\_decode — 把 HTML 实体转换为字符：

htmlentities — 将字符转换为 HTML 转义字符

htmlspecialchars\_decode — 将特殊的 HTML 实体转换回普通字符

htmlspecialchars — 将特殊字符转换为 HTML 实体

lcfirst — 使一个字符串的首字符小写

ucfirst — 使一个字符串的首字母大写

strtolower — 将字符串转化为小写

strtoupper — 将字符串转化为大写

levenshtein — 计算两个字符串之间的编辑距离

localeconv — 返回本地数字及货币格式信息。

money\_format — 将数字格式化成货币字符串

ltrim — 删除字符串左侧(开头)的空白字符（或其他字符）

rtrim — 删除字符串右侧(末尾)的空白字符（或者其他字符）

trim — 删除字符串首尾处的空白字符（或者其他字符）

<?php

$str = "Today is sun day!";

echo $str.'<br>';//Today is sun day!

echo ltrim($str,"Today") . '<br>';//is sun day!

echo rtrim($str,"day!") . '<br>';//Today is sun

echo trim($str,"Today day!");//is sun

?>

md5\_file — 计算指定文件的 MD5 散列值

md5 — 计算字符串的 MD5 散列值

nl\_langinfo — nl\_langinfo() 函数返回具体的本地信息,不适用windows平台

nl2br — 在字符串所有新行之前插入 HTML 换行标记

<?php

echo nl2br("one \nday.\n anotnher\n day!");

?>

/\*

one

day.

anotnher

day!

\*/

number\_format — 以千位分隔符方式格式化一个数字

number\_format(number,decimals,decimalpoint,separator)

1.number:要格式化的数字,如果未设置其他参数，则数字会被格式化为不带小数点且 以逗号（,）作为千位分隔符

2.可选。规定多少个小数。如果设置了该参数，则使用点号（.）作为小数点来格式化数字。

3.可选。规定用作小数点的字符串。

4.可选。规定用作千位分隔符的字符串。仅使用该参数的第一个字符。比如 "xxx" 仅输出 "x"。

注释：如果设置了该参数，那么所有其他参数都是必需的。

<?php

echo number\_format("123456789")."<br>";//123,456,789

echo number\_format("123456789",4)."<br>";//123,456,789.0000

echo number\_format("123456789" , 2 , "," , "?");//123?456?789,00

?>

parse\_str — 将字符串解析成多个变量

<?php

parse\_str("name=kk&age=123");

echo $name."<br>"; //kk

echo $age.'<br>'; //123

parse\_str("name=kk&age=123",$myArray);

print\_r($myArray); //Array ( [name] => kk [age] => 123 )

?>

quoted\_printable\_decode — 将 quoted-printable 字符串转换为 8-bit 字符串

quoted\_printable\_encode — 将 8-bit 字符串转换成 quoted-printable 字符串

quotemeta — 转义元字符集

quotemeta() 函数在字符串中某些预定义的字符前添加反斜杠

预定义的字符：

句号（.） 反斜杠（\） 加号（+） 星号（\*）

问号（?） 方括号（[]） 脱字号（^） 美元符号（$）

圆括号（()）

$str1 = "1 + 1 = 2";

$str2 = "1 \* 1 = 1";

$str3 = "Could you borrow me 5$?";

$str4 = "Are you not e&";

$str5 = "The caret [ ^ ] Looks like a hat!";

echo quotemeta($str1)."<br>"; //1 \+ 1 = 2

echo quotemeta($str2)."<br>"; //1 \\* 1 = 1

echo quotemeta($str3)."<br>"; //Could you borrow me 5\$\?

echo quotemeta($str4)."<br>"; //Are you not e&

echo quotemeta($str5)."<br>"; //The caret \[ \^ \] Looks like a hat!

setlocale — 设置地区信息

sha1\_file — 计算文件的 sha1 散列值

sha1 — 计算字符串的 sha1 散列值

similar\_text — 计算两个字符串的相似度

<?php

$q1="aaaaa";

$q2="bbbab";

echo similar\_text($q1,$q2);//1

?>

soundex — 计算字符串的 soundex 键。

sscanf — 根据指定格式解析输入的字符

str\_ireplace —替换字符串中的一些字符（不区分大小写）。

str\_replace — 子字符串替换(区分大小写)

<?php

echo str\_ireplace("WORLD","kk","Hello world!");//Hello kk!

?>

str\_pad — 使用另一个字符串填充字符串为指定长度

str\_pad(string,length,pad\_string,pad\_type)

1.必需。规定要填充的字符串

2.必需。规定新的字符串长度。如果该值小于字符串的原始长度，则不进行任何操作。

3.可选。规定供填充使用的字符串。默认是空白。

4.可选。规定填充字符串的哪边。

可能的值：

STR\_PAD\_BOTH - 填充字符串的两侧。如果不是偶数，则右侧获得额外的填充。

STR\_PAD\_LEFT - 填充字符串的左侧。

STR\_PAD\_RIGHT - 填充字符串的右侧。默认。

<?php

$str = "Hello World";

echo str\_pad($str,15,"@");//Hello World@@@@

?>

str\_repeat — 重复一个字符串

<?php

echo str\_repeat("qwe",5);//qweqweqweqweqwe

?>

str\_rot13 — 对字符串执行 ROT13 转换

ROT13 编码把每一个字母在字母表中向前移动 13 个字母。数字和非字母字符保持不变。

str\_shuffle — 随机打乱一个字符串,每次刷新页面结果都不同

<?php

echo str\_shuffle("today is sun day");

?>

str\_split — 将字符串转换为数组,有2个参数,str表示要分割的字符串,length规定每个数组的长度

<?php

print\_r(str\_split("qqqwwweeerrrr",3));

?>

/\*

Array

(

[0] => qqq

[1] => www

[2] => eee

[3] => rrr

[4] => r

)

\*/

str\_word\_count — 返回字符串中单词的使用情况

str\_word\_count(string,return,char)

1.string:必需。规定要检查的字符串。

2.return:可选。规定 str\_word\_count() 函数的返回值。

可能的值：

0 - 默认。返回找到的单词的数目。

1 - 返回包含字符串中的单词的数组。

2 - 返回一个数组，其中的键名是单词在字符串中的位置，键值是实际的单词。

3.可选。规定被视为单词的特殊字符。

<?php

print\_r(str\_word\_count("I love China & good study!",1));

print\_r(str\_word\_count("I love China & good study!",1,"&"));

?>

/\* Array

(

[0] => I

[1] => love

[2] => China

[3] => good

[4] => study

)

Array

(

[0] => I

[1] => love

[2] => China

[3] => &

[4] => good

[5] => study

)

\*/

strcasecmp(str1,str2) — 二进制安全比较字符串（不区分大小写）

如果 str1 小于 str2 返回 < 0； 如果 str1 大于 str2 返回 > 0；如果两者相等，返回 0

<?php

echo strcasecmp("beijing","BEIJING").'<br>';//0

echo strcasecmp("beijing","BEI").'<br>';//4

echo strcasecmp("beijing","JING");//-8

?>

strchr(string,search,before\_search);搜索字符串在另一字符串中的第一次出现

1.string:搜索字符串在另一字符串中的第一次出现

2.search:必需。规定所搜索的字符串。

如果该参数是数字，则搜索匹配该数字对应的 ASCII 值的字符。

3.before\_search:可选。默认值为 "false" 的布尔值。

如果设置为 "true"，它将返回 search 参数第一次出现之前的字符串部分。

<?php

echo strchr("Hello world!","world").'<br>'; //world!

echo strchr("Hello world!",111).'<br>'; //o world!

echo strchr("Hello world!","world",true); //Hello

?>

strcmp(str1,str2) — 二进制安全字符串比较

本函数返回：

0 - 如果两个字符串相等

<0 - 如果 str1 小于 str2

>0 - 如果 str1 大于 str2

对大小写敏感

<?php

echo strcmp("Hello world!","Hello world!") . '<br>'; // 0(两字符串相等)

echo strcmp("Hello world!","Hello") . '<br>'; // 7

echo strcmp("Hello world!","Hello world! Hello!") . '<br>'; // -7

echo strcmp("Hello","Hello") . '<br>';//0

echo strcmp("Hello","hELLo");//-1

?>

strcoll — 基于区域设置的字符串比较

strcspn — 获取不匹配遮罩的起始子字符串的长度

strip\_tags — 从字符串中去除 HTML 和 PHP 标记

stripcslashes — 反引用一个使用 addcslashes 转义的字符串

stripos — 查找字符串首次出现的位置（不区分大小写）

stripslashes — 反引用一个引用字符串

stristr — strstr 函数的忽略大小写版本

strlen — 获取字符串长度

strnatcasecmp — 使用“自然顺序”算法比较字符串（不区分大小写）

strnatcmp — 使用自然排序算法比较字符串

strncasecmp — 二进制安全比较字符串开头的若干个字符（不区分大小写）

strncmp — 二进制安全比较字符串开头的若干个字符

strpbrk — 在字符串中查找一组字符的任何一个字符

strpos — 查找字符串首次出现的位置

strrchr — 查找指定字符在字符串中的最后一次出现

strrev — 反转字符串

strripos — 计算指定字符串在目标字符串中最后一次出现的位置（不区分大小写）

strrpos — 计算指定字符串在目标字符串中最后一次出现的位置

strspn — 计算字符串中全部字符都存在于指定字符集合中的第一段子串的长度。

strstr — 查找字符串的首次出现

strtok — 标记分割字符串

strtr — 转换指定字符

substr\_compare — 二进制安全比较字符串（从偏移位置比较指定长度）

substr\_count — 计算字串出现的次数

substr\_replace — 替换字符串的子串

substr — 返回字符串的子串

ucwords — 将字符串中每个单词的首字母转换为大写

wordwrap — 打断字符串为指定数量的字串

2017/05/07 21:44