String Functions and Numaric Functions

H S Rana

CIT,UPES

February 10, 2014

ucfirst and ucword functions

You can also uppercase the first character of a string with the ucfirst() function, or format a string in “word case” with the ucwords() function.

ucfirst and ucword functions

You can also uppercase the first character of a string with the ucfirst() function, or format a string in "word case" with the ucwords() function.

```
<?php
// change string case
$str = 'the yellow brigands';

// output: 'The Yellow Brigands'
echo ucwords($str);

// output: 'The yellow brigands'
echo ucfirst($str);
?>
```

HTML String

addslashes() function, which automatically escapes special characters in a string with backslashes.

HTML String

addslashes() function, which automatically escapes special characters in a string with backslashes.

```
<?php
// escape special characters
// output: 'You\'re awake, aren\'t you?'
$str = "You're awake, aren't you?";
echo addslashes($str);
?>
```

HTML String

You can reverse the work done by addslashes() with the aptly named stripslashes() function, which removes all the backslashes from a string

HTML String

You can reverse the work done by addslashes() with the aptly named stripslashes() function, which removes all the backslashes from a string

```
<?php
// remove slashes
// output: 'John D'Souza says "Catch you later".'
$str = "John D\'Souza says \"Catch you later\".";
echo stripslashes($str);
?>
```

HTML String

The htmlentities() and htmlspecialchars() functions automatically convert special HTML symbols (like < and >) into their corresponding HTML representations (&lt; and &<hairline #>gt;). Similarly, the nl2br() function automatically replaces newline characters in a string with the corresponding HTML line break tag <br />.

HTML String

The htmlentities() and htmlspecialchars() functions automatically convert special HTML symbols (like < and >) into their corresponding HTML representations (&lt; and &<hairline #>gt;). Similarly, the nl2br() function automatically replaces newline characters in a string with the corresponding HTML line break tag <br />.

```
<?php
// replace with HTML entities
// output: '&lt;div width=&quot;200&quot;&gt;
//            This is a div&lt;/div&gt;'
$html = '<div width="200">This is a div</div>';
echo htmlentities($html);
// replace line breaks with <br/>s
// output: 'This is a bro<br />
//              ken line'
$lines = 'This is a bro
            ken line';
echo nl2br($lines);
```

HTML String

You can reverse the effect of the htmlentities() and htmlspecialchars() functions with the html_entity_decode() and htmlspecialchars_decode() functions.

HTML String

You can reverse the effect of the htmlentities() and htmlspecialchars() functions with the html_entity_decode() and htmlspecialchars_decode() functions.
the strip_tags() function searches for all HTML and PHP code within a string and removes it to generate a “clean” string.

HTML String

You can reverse the effect of the htmlentities() and htmlspecialchars() functions with the html_entity_decode() and htmlspecialchars_decode() functions.
the strip_tags() function searches for all HTML and PHP code within a string and removes it to generate a “clean” string.

```
<?php
// strip HTML tags from string
// output: 'Please log in again'
$html = '<div width="200">Please <strong>log in again
</strong></div>';
echo strip_tags($html);
?>
```

Using Numeric Function

. PHP offers the ceil() and floor() functions to round up and down a number.

Using Numeric Function

. PHP offers the ceil() and floor() functions to round up and down a number.

```
<?php
// round number up
// output: 19
$num = 19.7;
echo floor($num);
// round number down
// output: 20
echo ceil($num);
?>
```

Using Numeric Function

the abs() function, which returns the absolute value of a number.

Using Numeric Function

the abs() function, which returns the absolute value of a number.

```
<?php
// return absolute value of number
// output: 19.7
$num = -19.7;
echo abs($num);
?>
```

Using Numeric Function

the abs() function, which returns the absolute value of a number.

```
<?php
// return absolute value of number
// output: 19.7
$num = -19.7;
echo abs($num);
?>
```

The pow() function returns the value of a number raised to the power of another:

Using Numeric Function

the abs() function, which returns the absolute value of a number.

```
<?php
// return absolute value of number
// output: 19.7
$num = -19.7;
echo abs($num);
?>
```

The pow() function returns the value of a number raised to the power of another:

```
<?php
// calculate 4 ^ 3
// output: 64
echo pow(4,3);
?>
```

Using Numeric Function

he log() function calculates the natural or base-10 logarithm of a number, while the exp() function calculates the exponent of a number

Using Numeric Function

he log() function calculates the natural or base-10 logarithm of a number, while the exp() function calculates the exponent of a number

```
<?php
// calculate natural log of 100
// output: 2.30258509299
echo log(10);

// calculate log of 100, base 10
// output: 2
echo log(100,10);

// calculate exponent of 2.30258509299
// output: 9.99999999996
echo exp(2.30258509299);
?>
```

Using Numeric Function

rand() function automatically generates a random integer greater than 0. You can constrain it to a specific number range by providing optional limits as arguments.

Using Numeric Function

rand() function automatically generates a random integer greater than 0. You can constrain it to a specific number range by providing optional limits as arguments.

```
<?php
// generate a random number
echo rand();

// generate a random number between 10 and 99
echo rand(10,99);
?>
```

Using Numeric Function

PHP comes with built-in functions for converting between binary, decimal, octal, and hexadecimal bases. Here's an example which demonstrates the bindec(), decbin(), decoct(), dechex(), hexdec(), and octdec() functions in action:

Using Numeric Function

PHP comes with built-in functions for converting between binary, decimal, octal, and hexadecimal bases. Here's an example which demonstrates the bindec(), decbin(), decoct(), dechex(), hexdec(), and octdec() functions in action:

```
<?php
// convert to binary
echo decbin(8);// output: 1000
// convert to hexadecimal
echo dechex(8);  // output: 8
// convert to octal
echo decoct(8); // output: 10
// convert from octal
echo octdec(10); // output: 8
// convert from hexadecimal
echo hexdec(65); // output: 101
// convert from binary
echo bindec(1000); // output: 8
```

Using Numeric Function

PHP offers the number_format() function, which accepts four arguments: the number to be formatted, the number of decimal places to display, the character to use instead of a decimal point, and the character to use to separate grouped thousands (usually a comma).

Using Numeric Function

PHP offers the number_format() function, which accepts four arguments: the number to be formatted, the number of decimal places to display, the character to use instead of a decimal point, and the character to use to separate grouped thousands (usually a comma).

```
<?php
// format number (with defaults)
// output: 1,106,483
$num = 1106482.5843;
echo number_format($num);

// format number (with custom separators)
// output: 1?106?482*584
echo number_format($num,3,'*','?');
?>
```