

String Functions

H S Rana

CIT,UPES

February 3, 2014



Interrupting Loops

PHP's break statement allows you to break out of a loop at any point



Interrupting Loops

PHP's break statement allows you to break out of a loop at any point

```
<?php
$count = 0;
// loop 5 times
while ($count <= 4) {
    $count++;
    // when the counter hits 3
    // break out of the loop
    if ($count == 3) {
        break;
    }
    echo "This is iteration #$count <br/>";
}
?>
```



Skipping Loops

continue doesn't halt processing of the loop; it simply "jumps one" iteration.



Skipping Loops

continue doesn't halt processing of the loop; it simply "jumps one" iteration.

```
<?php
$count = 0;
// loop 5 times
while ($count <= 4) {
    $count++;
    // when the counter hits 3
    // skip to the next iteration
    if ($count == 3) {
        continue;
    }
    echo "This is iteration #$count <br/>";
}
?>
```



Checking for Empty String

The `empty()` function returns true if a string variable is “empty.”
Empty string variables are those with the values `' '`, `0`, `'0'`, or `NULL`.



Checking for Empty String

The `empty()` function returns true if a string variable is “empty.”
Empty string variables are those with the values `' '`, `0`, `'0'`, or `NULL`.

```
<?php
// test if string is empty
// output: true
$str = '';
echo (boolean) empty($str);
// output: true
$str = null;
echo (boolean) empty($str);
/ output: true
$str = '0';
echo (boolean) empty($str);
// output: true
unset($str);
echo (boolean)empty($str);
?>
```



strlen() function

The strlen() function returns the number of characters in a string.



strlen() function

The strlen() function returns the number of characters in a string.

```
<?php
// calculate length of string
// output: 17
$str = 'Welcome to Xanadu';
echo strlen($str);
?>
```



Reversing a string is as simple as calling the `strrev()` function,



Reversing a string is as simple as calling the strrev() function,

```
<?php
// reverse string
// output: 'pets llams en0'
$str = 'One small step';
echo strrev($str);
?>
```



String Repeatation

PHP offers the `str_repeat()` function, which accepts two arguments—the string to be repeated, and the number of times to repeat it.



String Repeatability

PHP offers the `str_repeat()` function, which accepts two arguments—the string to be repeated, and the number of times to repeat it.

```
<?php
// repeat string
// output: 'yoyoyo'
$str = 'yo';
echo str_repeat($str, 3);
?>
```



Working with Substrings

PHP also allows you to slice a string into smaller parts with the `substr()` function, which accepts three arguments: the original string, the position (offset) at which to start slicing, and the number of characters to return from the starting position.



Working with Substrings

PHP also allows you to slice a string into smaller parts with the `substr()` function, which accepts three arguments: the original string, the position (offset) at which to start slicing, and the number of characters to return from the starting position.

```
<?php
// extract substring
// output: 'come'
$str = 'Welcome to nowhere';
echo substr($str, 3, 4);
?>
```



Working with Substring

To extract a substring from the end of a string (rather than the beginning), pass `substr()` a negative offset.



Working with Substring

To extract a substring from the end of a string (rather than the beginning), pass `substr()` a negative offset.

```
<?php
// extract substring
// output: 'come here'
$str = 'Welcome to nowhere';
echo substr($str, 3, 5) . substr($str, -4, 4);
?>
```



Extracting Character

There are two ways to retrieve a single character of a string.



Extracting Character

There are two ways to retrieve a single character of a string.

```
<?php
// output: 'r'
$str = 'abracadabra';
echo substr($str,2,1);
?>
```

or

```
<?php
// output: 'r'
$str = 'abracadabra';
echo $str{2};
?>
```



Comparing String

If you need to compare two strings, the `strcmp()` function performs a case-sensitive comparison of two strings, returning a negative value if the first is “less” than the second, a positive value if it’s the other way around, and zero if both strings are “equal.”



Comparing String

If you need to compare two strings, the `strcmp()` function performs a case-sensitive comparison of two strings, returning a negative value if the first is “less” than the second, a positive value if it’s the other way around, and zero if both strings are “equal.”

```
<?php
// compare strings
$a = "hello";
$b = "hello";
$c = "hEllo";

// output: 0
echo strcmp($a, $b);

// output: 1
echo strcmp($a, $c);
?>
```



Counting Word in a String

PHP's `str_word_count()` function provides an easy way to count the number of words in a string.



Counting Word in a String

PHP's `str_word_count()` function provides an easy way to count the number of words in a string.

```
<?php
// count words
// output: 5
$str = "The name's Bond, James Bond";
echo str_word_count($str);
?>
```



String Replacement

PHP also has the `str_replace()` function, designed specifically to perform find-and-replace operations. This function accepts three arguments: the search term, the replacement term, and the string in which to perform the replacement.



String Replacement

PHP also has the `str_replace()` function, designed specifically to perform find-and-replace operations. This function accepts three arguments: the search term, the replacement term, and the string in which to perform the replacement.

```
<?php
// replace '@' with 'at'
// output: 'john at domain.net'
$str = 'john@domain.net';
echo str_replace('@', ' at ', $str);
?>
```



Formatting String

PHP's `trim()` function can be used to remove leading or trailing whitespace from a string; this is quite useful when processing data entered into a Web form.



Formatting String

PHP's `trim()` function can be used to remove leading or trailing whitespace from a string; this is quite useful when processing data entered into a Web form.

```
<?php
// remove leading and trailing whitespace
// output 'a b    c'
$str = '  a b    c  ';
echo trim($str);
?>
```



Changing Case

Changing the case of a string is as simple as calling the `strtolower()` or `strtoupper()` function.



Changing Case

Changing the case of a string is as simple as calling the `strtolower()` or `strtoupper()` function.

```
<?php
// change string case
$str = 'Yabba Dabba Doo';

// output: 'yabba dabba doo'
echo strtolower($str);

// output: 'YABBA DABBA DOO'
echo strtoupper($str);
?>
```

