

PHP Tutorial-String Manipulation

PHP String Manipulation

PHP provides a rich set of functions to manipulate strings. In this topic, we will discuss some common functions used by PHP developers to remove spaces from a string, count the number of characters of a string, convert a string to contain upper case or lower case letters, split a string or join strings, get substrings from a string, compare strings, search for a substring in a string, and replace and old substring with a new substring of a string, etc.

trim, ltrim, and rtrim functions

trim, ltrim, and rtrim functions are used to remove space from a string.

-trim(String) removes leading and trailing space from the string.

-ltrim(String) removes leading spaces.

-rtrim(String) removes trailing spaces.

Example:

```
<?php
$str=" PHP web development ";
echo trim($str)."<br>";//remove spaces at the beginning and the end
echo ltrim($str)."<br>";//remove spaces at the beginning only
echo rtrim($str)."<br>";//remove spaces at the end only
?>
```

strlen() function

The strlen(String) function is used to count the number of characters of a string. In the example below, we use this function to count length of the password text input by a user in the login.php page. If the length of the password text is less than 8 characters, the error message will display. In contrast, the user will be redirected to the welcome.php page.

Example:

login.php

```
<html>
<head>
<title>Password length checking</title>
<?php
$pa=$_GET['password'];
if(strlen($pa)<8) echo "<font color='red'>Your password must be more than 8 characters.</font>";
else header("location: welcome.php");
?>
</head>
<body>
```

```
<form action="testlib.php" method="get">
<table>
<tr>
<td>User name:</td><td> <input type="text" name="username" ></td>
</tr>
<tr>
<td>Password: </td><td><input type="text" name="password" ></td>
</tr>
<tr>
<td><input type="submit" name="sub" value="submit"></td>
</tr>
</table>
</form>

</body>
</html>
```

welcome.php

```
<?php
echo "Welcome!";

?>
```

PHP String Manipulation

strtolower, strtoupper, ucfirst, ucwords function

The strtolower, strtoupper, ucfirst, and ucwords functions are used to change change cases of a string:

- strtolower(String) changes a string to lowercase.
- strtoupper(String) changes a string to uppercase.
- ucfirst(String) capitalizes the first character of a string.
- ucwords(String) capitalizes the first character of each word in a string.

Example:

```
<?php
$str="PHP is powerful.";
echo strtolower($str)."<br>";
echo strtoupper($str)."<br>";
echo ucfirst($str)."<br>";
echo ucwords($str)."<br>";
?>
```

strcmp() and strcasecmp() functions

The `strcmp(String1,String2)` compares `String1` with `String2`. It returns less than zero if `String1` is less than `String2`. If `String1` is greater than `String2` it return greater than zero. If both strings are equal, it returns 0. This function compares two strings in case-sensitive manner. If you want to compare two strings without case-sensitivity, you can use `strcasecmp()` instead.

Example:

```
<?php
$str1="php";
$str2="PHP";
if(strcmp($str1,$str2)<0) echo "str1 is less than str2.";
else if(strcmp($str1,$str2)>0) echo "str1 is greater than str2.";
else echo "Both strings are equal.";
?>
```

split() and join() functions

The `split(Separator_char, String)` function is used to split a string in to an array of strings by a separating character. In this example, we split the string "PHP is powerful." by space character.

Example:

```
<?php
$str="PHP is powerful.";
$strarr=split(' ', $str);
$limit=count($strarr);
for($i=0;$i<$limit;$i++)
echo $strarr[$i]."<br>";
?>
```

In contrast, the `join(Joiner_char)` is used to combine an array of strings to become a single longer string.

Example:

```
<?php
$strarr=array(0=>"PHP",1=>"JavaScript",2=>"CSS");
$st=join(', ', $strarr);
echo $st;
?>
```

substr() function

The `substr()` method has two main forms:

-`substr(String, Start)` returns a substring from the `Start` position to the end of the string.

-`substr(String,Start,Length)` returns a substring from the `Start` position in which the length of the substring is equal to `Length`.

Exampe:

```
<?php
$str="PHP programming";
$substr1=substr($str,4);
$substr2=substr($str,0,3);
echo $substr1."<br>";
echo $substr2;
?>
```

strpos() and str_replace() functions

The strpos(String, String_to_find) returns the position of the String_to_find in the String. The str_replace(Old_string,New_string,String) is used to replace the Old_string with the New_string. In the example below, we create a simple find-and-replace word program that allows a user to find and replace words in a textarea box.

Example:

```
<html>
<head>
<title>Find and Replace</title>

<?php
$str1=$_GET['txtmessage'];
global $strnew;
$str_to_find=$_GET['txtfind'];
$str_to_replace=$_GET['txtreplace'];
$pos=strpos($str1,$str_to_find);//find the word to replace
if($pos>=0)//The word is found. Then make replacement.
$strnew=str_replace($str_to_find,$str_to_replace,$str1);

?>

</head>
<body>

<form action="welcome.php" method="get">
<table>
<tr>
<?php

//assign the new string to the textarea
echo "<td><textarea name='txtmessage' rows='18' cols='100'>$strnew</textarea></td>";
?>

</tr></table>
<table>
<tr>
<td align='left'>Find what:</td><td align='left'> <input type="text" name="txtfind" ></td>
</tr>
```

```
<tr>  
<td align='left'>Replace with: </td><td align='left'><input type="text" name="txtreplace" ></td>  
<td><input type="submit" name="sub" value="submit"></td>  
</tr>
```

```
</table>
```

```
</form>
```

```
</body>
```

```
</html>
```