

## CHANGE CASE OF A CHARACTER OR STRING

PHP provides functions that allow you to change the case of text in a string. For example, you can use the `strtolower` function to change all the text in a string to lowercase. To change all the text in a string to uppercase, use the `strtoupper` function. Changing all the text in a string to the same case can make the text easier to search and manage.

The `ucfirst` function allows you to change the first character in a string to uppercase. PHP will apply the `ucfirst` function to the first character in a string, even if the first character is not a letter. For example, if the first character in a string is a number, the string will not display any changes.

Use the `ucwords` function to change the first character of each word in a string to uppercase. PHP will consider any character or group of characters that directly follows a whitespace character, such as a space or carriage return, to be a word. For more information about whitespace characters, see page 112.

Each function takes the name of the variable that stores the string you want to change as its argument. If the string has not been assigned to a variable, you can use the string, enclosed in quotation marks, as the argument.

The `strtolower`, `strtoupper`, `ucfirst` and `ucwords` functions can be used together to apply the results of one case function to another case function. This lets you change the case of a string in more than one way. For example, using the `ucwords` function will have no apparent effect if the string contains all uppercase characters. To ensure that only the first letter of every word is uppercase in the string, you can first apply the `strtolower` function and then apply the `ucwords` function to the string, such as `ucwords(strtolower($changeCase))`.

### CHANGE CASE OF A CHARACTER OR STRING

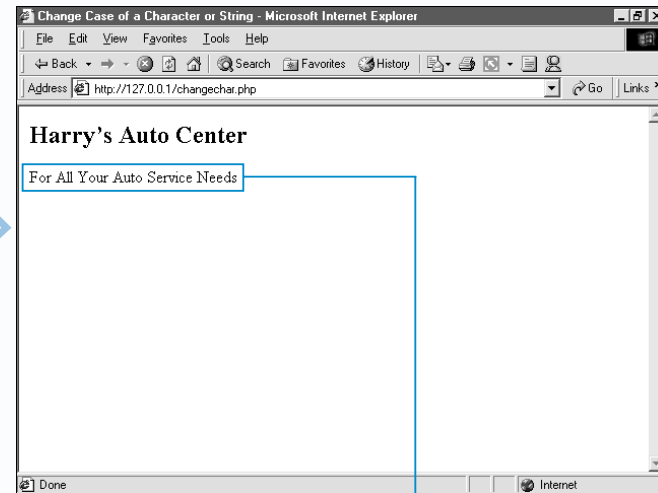
```

Untitled - Notepad
File Edit Search Help
<html>
<head>
<title>Change Case of a Character or String</title>
</head>
<body>

<h2>Harry's Auto Center</h2>

<?php
$wrongCase = "for all your auto service needs";
print ucwords($wrongCase);
?>
</body>
</html>

```



1 Type the code that creates a variable containing the string you want to change.

2 Type the name of the function you want to use to change the case of characters in the string (`strtolower`, `strtoupper`, `ucfirst` or `ucwords`) followed by `()`.

3 Between the parentheses, type the name of the variable that stores the string.

4 Type the code that uses the result of the function.

5 Display the PHP page in a Web browser.

The Web browser displays the result of changing the case of characters in a string.

## USING ASCII VALUES

ASCII (American Standard Code for Information Interchange) is a numbering system that assigns numerical values to letters, numbers and other characters. Many programs save, read and exchange data using the ASCII numbering system. Using the `ord` and `chr` functions allows you to convert between ASCII values and characters.

The `ord` function converts a specified character to an ASCII value. If multiple characters are specified, the `ord` function will return the ASCII value of only the first character. You may find that you need to convert characters to their ASCII values in order to format and manipulate data. For example, if you use a PHP page to write data to a file, you may need to convert formatting characters such as horizontal tabs to their ASCII values before you can write the characters to the file.

The `chr` function converts a specified ASCII value to a character. PHP allows you to manipulate the results of the `chr` function as you would manipulate strings. For example,

you can join characters returned by the `chr` function together using the concatenation operator (`.`). Converting ASCII values to characters in a PHP page is useful for including characters in your PHP page that do not appear on your keyboard, such as mathematical symbols.

The standard ASCII characters are numbered from 0 to 127. When using the extended ASCII character set, the characters are numbered up to 255. You can visit the [www.asciitable.com](http://www.asciitable.com) Web site to view a complete chart of the extended ASCII character set.

All characters on a computer, including non-displayable characters and whitespace characters, such as the tab and newline characters, have an ASCII value. To convert a non-displayable or whitespace character to an ASCII value, you must provide the escape sequence for the character. For a list of commonly used escape sequences, see the top of page 42.

### USING ASCII VALUES

```

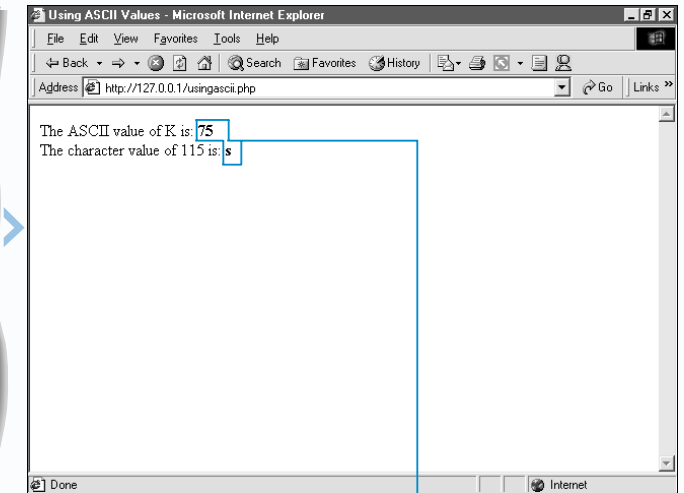
Untitled - Notepad
File Edit Search Help
<html>
<head>
<title>Using ASCII Values</title>
</head>
<body>

<?php
print "The ASCII value of K is: ";
print "<b>" . ord("K") . "</b><br>";

print "The character value of 115 is: ";
print "<b>" . chr(115) . "</b>";

?>
</body>
</html>

```



1 To convert a character to an ASCII value, type `ord()`.

2 Between the parentheses, type the character you want to convert enclosed in quotation marks.

3 To convert an ASCII value to a character, type `chr()`.

4 Between the parentheses, type the ASCII value you want to convert.

5 Type the code that uses the `ord` and `chr` functions.

6 Display the PHP page in a Web browser.

The Web browser displays the results of using the `ord` and `chr` functions.

# FORMAT A STRING

The `printf` function allows you to format strings for output. The `printf` function requires two arguments—a format string and a list of values you want to format. A format string contains the formatting rules that will be applied to the values. Each formatting rule in the format string must have a corresponding value in the list.

Formatting rules are specified using a percentage symbol (%) and a type specifier that indicates the type of formatting to be applied. Before including the type specifier, you may also indicate padding, alignment, width and precision specifiers, in that order, to further format the string. These specifiers are useful for lining up text and numbers, improving the appearance of data in text-based Web browsers.

The padding specifier indicates the character you want to use to pad the result. The padding specifier consists of a single quotation mark (') followed by the character you want to use. By default, spaces are used for padding.

The result of the `printf` function is normally right-justified, but you can include the alignment specifier (-) to signify that the result should be left-justified.

You can also include a width specifier to indicate a minimum width. For example, the formatting rule `%8s` formats a string so it will have a minimum width of eight characters.

A precision specifier indicates how many decimal digits should be displayed. The precision specifier is represented by a dot (.) followed by the number of decimal digits. For example, the formatting rule `%.2f` formats a floating-point number with a precision of two decimal places.

A format string can also contain characters you do not want to format. For example, to include text in a string with formatted numbers, type the text in the format string. PHP will replace the formatting rules in the format string with the formatted numbers and then output the text and formatted numbers.

## Extra

The `sprintf` function uses the same formatting rules as the `printf` function. However, the `sprintf` function returns the formatted string as a result that can be assigned to a variable instead of automatically outputting the formatted string.

### TYPE THIS:

```
$name = "David";
$rate = 10.2369;
$info = sprintf("%s's hourly rate is $%.2f.", $name, $rate);
print "$info<br>";
```

### RESULT:

David's hourly rate is \$10.24.

You can use the following type specifiers to indicate the type of formatting you want to apply.

TYPE SPECIFIER:	MEANING:
b	Format as a binary number.
c	Format as a character specified by the ASCII value.
d	Format as a decimal number.
e	Format as a floating-point number, using exponential notation.
f	Format as a floating-point number.
o	Format as an octal number.
s	Format as a string.
x	Format as a hexadecimal number with lowercase letters.
X	Format as a hexadecimal number with uppercase letters.

## FORMAT A STRING

```
<html>
<head>
<title>Format a String</title>
</head>
<body>

<h3>Employee Information</h3>
<pre>

<?php
$staff = array (
    array("Jill", 13.4567, 10),
    array("Maureen", 9.4823, 5),
    array("David", 10.2369, 12)
);

printf("Name: %-8s Rate: %'*.5.2f Hours: %3d \n");
?>

</pre>
</body>
</html>
```

- 1 To format a string for output, type `printf()`.
- 2 Between the parentheses, type the formatting rules you want to use enclosed in quotation marks.

Note: You may also include text you do not want to format within the quotation marks.

```
<html>
<head>
<title>Format a String</title>
</head>
<body>

<h3>Employee Information</h3>
<pre>

<?php
$staff = array (
    array("Jill", 13.4567, 10),
    array("Maureen", 9.4823, 5),
    array("David", 10.2369, 12)
);

printf("Name: %-8s Rate: %'*.5.2f Hours: %3d \n", $name);
?>

</pre>
</body>
</html>
```

- 3 Type a comma followed by the name of the variable you want to use for the first formatting rule you specified.
- 4 You may specify a value instead of a variable. String values must be enclosed in quotation marks.

```
<html>
<head>
<title>Format a String</title>
</head>
<body>

<h3>Employee Information</h3>
<pre>

<?php
$staff = array (
    array("Jill", 13.4567, 10),
    array("Maureen", 9.4823, 5),
    array("David", 10.2369, 12)
);

foreach ($staff as $employee)
{
    list($name, $rate, $hours) = $employee;
    printf("Name: %-8s Rate: %'*.5.2f Hours: %3d \n", $name, $rate, $hours);
}
?>

</pre>
</body>
</html>
```

- 4 Repeat step 3 for each formatting rule you specified.
- 5 Type the code that uses the `printf` function.

```
Format a String - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search Favorites History
Address http://127.0.0.1/formatstring.php Go Links
Employee Information
Name: Jill Rate: ***13.46 Hours: 10
Name: Maureen Rate: ***9.48 Hours: 5
Name: David Rate: ***10.24 Hours: 12
```

- 6 Display the PHP page in a Web browser.
- 7 The Web browser displays the result of formatting a string.

## FIND THE LENGTH OF A STRING

PHP's `strlen` function allows you to determine the number of characters in a string. The `strlen` function is often used to find the length of a string stored in a variable. The `strlen` function is especially useful for finding the length of a string before it is processed. For example, you can use the `strlen` function to check whether a password provided by a user has the required minimum number of characters or whether the data entered by a user is within a specific character limit before allowing the user to continue using your Web site.

Determining the length of a string is also useful if you want to iterate through each character in a string using a loop. You would determine the length of the string and use the length as the condition for ending the loop.

If the string for which you want to determine the length contains spaces or punctuation marks, the `strlen` function will count the spaces and punctuation marks as characters. PHP evaluates any escape sequences in a string, such as tabs (`\t`) or newlines (`\n`), before counting the number of characters, so each escape sequence is counted as one character. For example, the `strlen` function would count six characters in the string "hello\n".

If you use the `strlen` function with a numeric value, PHP will evaluate the number as if it were a string. If the number includes a decimal point, the decimal point will also be counted as a character. For example, the `strlen` function would count nine characters in the number 123456.78.

### Extra

The `strpos` function allows you to determine the length of a portion of a string. To use the `strpos` function, you must specify the characters you want to count in the string, called a mask. The `strpos` function returns the length of the portion of the string that contains the characters specified in the mask. The `strpos` function will stop executing when it encounters the first character that is not specified in the mask.

**TYPE THIS:**

```
$amount = "\$259 dollars for hotel expenses.";
$mask = "1234567890\$";
print strpos($amount, $mask);
```

**RESULT:**

4

The `strcspn` function is similar to the `strpos` function, except the `strcspn` function works in reverse by returning the length of the portion of a string which does not contain the characters specified in a mask. When the `strcspn` function encounters a character specified in the mask, the function will stop executing.

**TYPE THIS:**

```
$filename = "\n\n\r\t\t##my_webpage.html";
$mask = "abcdefghijklmnopqrstuvwxyz_.";
print strcspn($filename, $mask);
```

**RESULT:**

10

### FIND THE LENGTH OF A STRING

```
Untitled - Notepad
File Edit Search Help
<html>
<head>
<title>Find the Length of a String</title>
</head>
<body>
<h2>Welcome to the ABC Corporation Web Site.</h2>
<?php
$password = "pass";
strlen()
?>
</body>
</html>
```

1 Type the code that creates a string variable and assigns it a value.

2 To determine the length of the string, type `strlen()`.

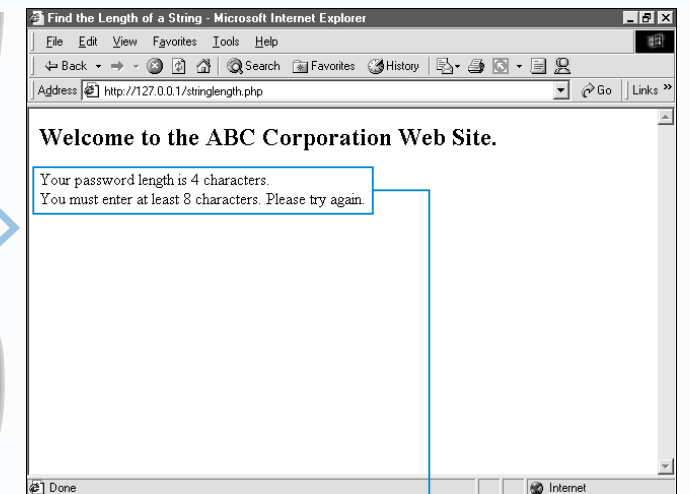
```
Untitled - Notepad
File Edit Search Help
<html>
<head>
<title>Find the Length of a String</title>
</head>
<body>
<h2>Welcome to the ABC Corporation Web Site.</h2>
<?php
$password = "pass";
strlen($password)
?>
</body>
</html>
```

3 Between the parentheses, type the name of the variable that stores the string for which you want to determine the length.

You can also type a string between the parentheses. Strings must be enclosed in quotation marks.

```
Untitled - Notepad
File Edit Search Help
<html>
<head>
<title>Find the Length of a String</title>
</head>
<body>
<h2>Welcome to the ABC Corporation Web Site.</h2>
<?php
$password = "pass";
if (strlen($password) < 8)
{
    print "Your password length is " . strlen($password) . " characters.<br>";
    print "You must enter at least 8 characters. Please try again.";
}
else
{
    print "You have entered a valid password.";
}
?>
</body>
</html>
```

4 Type the code that uses the `strlen` function.



5 Display the PHP page in a Web browser.

The Web browser displays the result of finding the length of a string.

## DIVIDE A STRING

You can use the `strtok` function to divide a string based on one or more characters, called tokens, you specify. This process is also referred to as tokenizing a string. When the `strtok` function is first called, it requires two arguments—the string you want to divide and the tokens you want to use to divide the string. The `strtok` function then returns a segment of the string from the beginning of the string to the position of the first token.

You may specify one or more characters to be used as tokens. If you specify an empty string as a token, the `strtok` function will simply return the entire string. The `strtok` function will also return the entire string if the string does not contain any of the tokens you specified.

After the first time the `strtok` function is called in a PHP page, PHP will keep track of the remaining string that has not yet been divided. When the `strtok` function is called again in the page, you need to indicate only the tokens. You do not need to specify the string you want to divide again.

The remaining string will then be processed and the `strtok` function will return a segment of the string from the last token found to the next token in the string.

If you want to access every segment of a string, you may want to call the `strtok` function once and then use a while loop to execute subsequent calls to the `strtok` function. The while loop should be set to terminate when the `strtok` function returns an empty string.

The `strtok` function works well for processing strings with simple structures. For example, the `strtok` function is useful for counting the number of words in a sentence or for processing a database file in which the data is separated by tab characters.

### Extra

When using a while loop to execute the `strtok` function, if two tokens are adjacent in the string, the function will return an empty string and the while loop may end before the rest of the string is processed. To avoid this type of problem, you should use a strict comparison for `FALSE`.

#### Example:

```
while (($word = strtok(" .")) !== FALSE)
{
    print $word . "<br>";
}
```

The `explode` function may be used to divide a string and return an array of strings. This function typically requires two arguments—a string to be used as a separator and the string to be divided. You may also specify an optional third argument that indicates the maximum number of elements you want returned.

#### TYPE THIS:

```
$fruitsString = "apple orange peach grape melon";
$fruitsArray = explode(" ", $fruitsString, 5);
foreach ($fruitsArray as $value)
{
    print "$value : ";
}
```

The `implode` function may be used to join the elements in an array into one string. When calling this function, you need to specify a string, which will be used to join the array elements, and the array containing the elements to be joined.

#### TYPE THIS:

```
$fruitsArray = array ("apple", "orange", "peach",
                    "grape", "melon");
$fruitsString = implode(" and ", $fruitsArray);
print $fruitsString;
```

#### RESULT:

apple and orange and peach and grape and melon

#### RESULT:

apple : orange : peach : grape : melon :

### DIVIDE A STRING

```
Untitled - Notepad
File Edit Search Help
<html>
<head>
<title>Divide a String</title>
</head>
<body>
<?php
$sentence = "The Internet is the largest computer network in the world.";
strtok($sentence, " .");
?>
</body>
</html>
```

**1** Create a variable that contains a string you want to divide.

**2** To divide the string, type `strtok()`.

**3** Between the parentheses, type the name of the variable that contains the string, followed by a comma.

**4** Type the characters you want to use to divide the string, enclosed in quotation marks.

```
Untitled - Notepad
File Edit Search Help
<html>
<head>
<title>Divide a String</title>
</head>
<body>
<?php
$sentence = "The Internet is the largest computer network in the world.";
$word = strtok($sentence, " .");
print $word . "<br>";
?>
</body>
</html>
```

**5** Type the code that assigns the result of the `strtok` function to a variable.

**6** Type the code that uses the result of the `strtok` function.

■ The `strtok` function returns a segment of the string up to the first token.

```
Untitled - Notepad
File Edit Search Help
<html>
<head>
<title>Divide a String</title>
</head>
<body>
<?php
$sentence = "The Internet is the largest computer network in the world.";
$word = strtok($sentence, " .");
print $word . "<br>";
while ($word = strtok(" ."))
{
    print $word . "<br>";
}
?>
</body>
</html>
```

**7** To access the subsequent segments of the string, type the code that creates a while loop and uses the results of the `strtok` function.

■ After the first call to the `strtok` function, you do not need to specify the variable that contains the string you want to divide.

```
Divide a String - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Forward Stop Home Search Favorites History
Address http://127.0.0.1/dividestring.php
The
Internet
is
the
largest
computer
network
in
the
world
```

**8** Display the PHP page in a Web browser.

■ The Web browser displays the result of using the `strtok` function to divide a sentence.



## TRIM A STRING

There are several functions used for trimming whitespace characters from strings before the strings are used or processed. Whitespace characters, which include spaces, carriage returns (`\r`), newlines (`\n`), tabs (`\t`), vertical tabs (`\013`) and null characters (`\0`), are normally invisible when displayed on a Web page, but may cause problems when included in data to be processed.

To remove whitespace characters from the end of a string, you can use the `rtrim` function. The `ltrim` function is used to remove whitespace characters from the beginning of a string. To remove whitespace characters from the beginning and the end of a string, you can use the `trim` function.

The `rtrim`, `ltrim` and `trim` functions all take the string to be trimmed as an argument and return the string with the appropriate whitespace characters removed. The returned string can be directly displayed to a Web browser or assigned to a variable, which can then be further processed in the script.

These functions are useful for cleaning up user input from a form, since the data received from forms can sometimes be unpredictable. For example, when a user is asked to enter data into a form, a user may inadvertently type extra spaces. Trimming the extra spaces from the beginning and end of the string may help ensure that the data is processed properly.

You may also use these functions to remove formatting from text that is read from a file or will be written to a file. For example, text files that come from different operating systems, such as UNIX or Windows, may be formatted differently. Removing whitespace characters from a line of text before it is processed or written to a file can help avoid potential errors.

### Extra

If you want to verify that the whitespace characters have been removed from a string, you may use the `strlen` function to count the number of characters in the original string and the number of characters in the resulting string.

#### TYPE THIS:

```
$original = "Welcome to my Web page!\0\t\r\n";
$trimmed = rtrim($original);
print "Length of original string: " . strlen($original);
print "<br>Length of trimmed string: " . strlen($trimmed);
```

#### RESULT:

Length of original string: 28  
Length of trimmed string: 23

PHP does not have a function that removes whitespace characters from the middle of a string, but you can use the `eregi_replace` function to replace adjacent whitespace characters in a string with a specified character or string. The `eregi_replace` function

requires three arguments—a pattern containing the characters to be matched and replaced, a replacement character or string and a string that contains characters to be replaced. The `eregi_replace` function cannot be used to replace null characters (`\0`) in a string.

#### TYPE THIS:

```
$original = " This \t is\n\n a string . . .";
$replaced = eregi_replace("[\n\t\013\r ]+", " ", $original);
print "<b>Before:</b><pre>$original</pre>";
print "<b>After:</b><pre>$replaced</pre>";
```

#### RESULT:

##### Before:

```
This      is
a string . . .
```

##### After:

```
This is a string . . .
```

### TRIM A STRING

```
Untitled - Notepad
File Edit Search Help

<html>
<head>
<title>Trim a String</title>
</head>
<body>

<?php
$message1 = "Welcome to my Web page! \t\t\n";
$message2 = "\n\r My name is Martine.";
$message3 = "\n\r My e-mail address is martine@abccorp.com \t\t\n";
print rtrim($message1) . "<br>";
?>

</body>
</html>
```

**1** Type the code that creates variables that contain strings you want to trim.

**2** To remove whitespace characters from the end of a string, type `rtrim()`.

**3** Between the parentheses, type the name of the variable that contains the string you want to trim.

**4** Type the code that uses the result of the `rtrim` function.

```
Untitled - Notepad
File Edit Search Help

<html>
<head>
<title>Trim a String</title>
</head>
<body>

<?php
$message1 = "Welcome to my Web page! \t\t\n";
$message2 = "\n\r My name is Martine.";
$message3 = "\n\r My e-mail address is martine@abccorp.com \t\t\n";
print rtrim($message1) . "<br>";
print ltrim($message2) . "<br>";
?>

</body>
</html>
```

**5** To remove whitespace characters from the beginning of a string, type `ltrim()`.

**6** Between the parentheses, type the name of the variable that contains the string you want to trim.

**7** Type the code that uses the result of the `ltrim` function.

```
Untitled - Notepad
File Edit Search Help

<html>
<head>
<title>Trim a String</title>
</head>
<body>

<?php
$message1 = "Welcome to my Web page! \t\t\n";
$message2 = "\n\r My name is Martine.";
$message3 = "\n\r My e-mail address is martine@abccorp.com \t\t\n";
print rtrim($message1) . "<br>";
print ltrim($message2) . "<br>";
print trim($message3) . "<br>";
?>

</body>
</html>
```

**8** To remove whitespace characters from the beginning and end of a string, type `trim()`.

**9** Between the parentheses, type the name of the variable that contains the string you want to trim.

**10** Type the code that uses the result of the `trim` function.

```
Trim a String - Microsoft Internet Explorer
File Edit View Favorites Tools Help

Back Forward Stop Home Search Favorites History
Address http://127.0.0.1/trimstring.php Go Links

Welcome to my Web page!
My name is Martine.
My e-mail address is martine@abccorp.com

Done Internet
```

**11** Display the PHP page in a Web browser.

The Web browser displays the results of trimming strings.

## COMPARE STRINGS

You can use the `strcmp` function to compare two strings based on the ASCII (American Standard Code for Information Interchange) values of the characters in each string. For more information about ASCII values, see page 105.

Comparing the values of strings can be useful if you want to sort the strings into a specific order. For example, you can compare the strings "cap" and "cat" and then put the strings in order. Based on a comparison of the ASCII values of each character in the strings, the string "cap" would come before the string "cat". It is important to remember that uppercase characters have different ASCII values than lowercase characters.

The `strcmp` function takes the strings you want to compare or the names of the variables storing the strings as its arguments. After performing a case-sensitive comparison, the `strcmp` function returns an integer value indicating which of the strings has a greater value. If the first string has a greater value than the second string, the

function returns a value greater than zero, typically 1. If the second string has a greater value than the first string, the function returns a value less than zero, typically -1. If the two strings have an equal value, the function returns zero.

If one entire string exactly matches the beginning of the other string, the `strcmp` function returns the number of additional characters found in the longer string. If you compare an empty string to a non-empty string, the number of characters in the non-empty string is returned. Whether the number is positive or negative depends on which of the two strings is longer. If the first string is longer, a positive number is returned. If the second string is longer, a negative number is returned.

The `strcmp` function is often used with an `if` statement to test which string has the greater value and then perform an action.

### Extra

The `strncmp` function can be used to compare a specific number of characters at the beginning of two strings. This function takes three arguments—the names of the two variables storing the strings you want to compare and the number of characters to be compared. The `strncmp` function, similar to the `strcmp` function, performs a case-sensitive comparison and returns an integer indicating which of the strings has a greater value or indicating that the strings have the same value.

#### TYPE THIS:

```
$word1 = "banana";
$word2 = "bandit";

$result = strncmp($word1, $word2, 3);
if ($result == 0)
{
    print "The beginning of the word <b>$word1</b> is the same";
    print " as the beginning of the word <b>$word2</b>.";
}
else
{
    print "The words are different.";
}
```

#### RESULT:

The beginning of the word **banana** is the same as the beginning of the word **bandit**.

The `strcasecmp` and `strncasecmp` functions can be used to perform case-insensitive string comparisons. When these functions are used, the case of characters in the strings being compared is ignored. To use the `strncasecmp` function, you must indicate the number of characters you want to compare at the beginning of the strings.

#### TYPE THIS:

```
$firstWord = "hello";
$secondWord = "HELLO";
$result = strcasecmp($firstWord, $secondWord);

if ($result == 0)
{
    print "The words are the same.";
}
else
{
    print "The words are different.";
}
```

#### RESULT:

The words are the same.

### COMPARE STRINGS

```
<html>
<head>
<title>Compare Two Strings</title>
</head>
<body>

<?php
$firstWord = "peach";
$secondWord = "mango";
strcmp();
?>

</body>
</html>
```

1 Type the code that creates the first string variable you want to use in a comparison.

2 Type the code that creates the second string variable you want to use in a comparison.

3 To compare the strings, type `strcmp()`.

```
<?php
$firstWord = "peach";
$secondWord = "mango";
strcmp($firstWord, $secondWord);
?>

</body>
</html>
```

4 Between the parentheses, type the name of the first string variable followed by a comma.

5 Type the name of the second string variable.

```
<?php
$firstWord = "peach";
$secondWord = "mango";

$result = strcmp($firstWord, $secondWord);
if ($result >= 1)
{
    $statement = "comes after";
}
else if ($result <= -1)
{
    $statement = "comes before";
}
else
{
    $statement = "is the same as";
}

print "The word <b>$firstWord</b> $statement ";
print "the word <b>$secondWord</b>.<br>";
```

6 Type the code that uses the `strcmp` function.

The word **peach** comes after the word **mango**.

7 Display the PHP page in a Web browser.

8 The Web browser displays the result of comparing strings.

# EXTRACT OR REPLACE A SUBSTRING

The `substr` function allows you to access a specific part of a string, called a substring.

You must specify the name of the variable containing the string from which you want to extract a substring. The offset, or position, of the substring within the string must also be specified. If the offset value is a positive number, PHP finds the starting position of the substring by counting the number of characters from the beginning of the string. Characters in a string are numbered starting at zero (0). The `substr` function will return a substring that begins at the position you specified and ends at the end of the string.

If the offset value you specify is a negative number, PHP finds the starting position of the substring by counting the number of characters from the end of

the string. When counting characters from the end of a string, PHP starts the count at 1, not zero (0).

The `substr_replace` function allows you to specify replacement text for a substring and return the modified string. You must specify the name of the string variable in which you want to replace text. The new text you want to use and a positive or negative offset of the substring must also be specified.

You can also specify the length of the substring you want to replace. If a positive number is used to specify the length, PHP replaces that number of characters with the replacement text. If a negative number is used, PHP stops replacing text that number of characters from the end of the string. The replacement text does not have to be the same length as the substring.

## Extra

If you do not want to extract a substring from the offset to the end of the string, you can include an argument in the `substr` function to specify the number of characters you want to extract. If a negative number is specified, PHP removes that number of characters from the end of the substring.

### TYPE THIS:

```
$file = "my_webpage.htm";
print "Filename: " . substr($file, 0, 10);
```

### RESULT:

Filename: my\_webpage

The `strstr` function can be used to extract a substring from the first occurrence of a specified character to the end of a string. To use this function, you must specify the name of the string variable from which you want to extract a substring and the character to be matched.

### TYPE THIS:

```
$email = "martine@maran.com";
$domain = strstr($email, "@");
print $domain;
```

### RESULT:

@maran.com

The `strrchr` function can be used to extract a substring from the last occurrence of a specified character to the end of a string. To use this function, you must specify the name of the string variable from which you want to extract a substring and the character to be matched.

### TYPE THIS:

```
$url = "http://www.abccorp.com/Sales2000";
$directory = strrchr($url, "/");
print $directory;
```

### RESULT:

/Sales2000

## EXTRACT OR REPLACE A SUBSTRING

```
Untitled - Notepad
File Edit Search Help
<html>
<head>
<title>Extract or Replace a Substring</title>
</head>
<body>

<?php
$file = "my_webpage.htm";
print "File Type: " . substr($file, -4);
print "<br>\n";
?>

</body>
</html>
```

1 Type the code that creates a string variable and assigns it a value.

2 To extract a substring, type `substr()`.

3 Between the parentheses, type the name of the string variable from which you want to extract a substring.

4 Type a comma followed by the offset of the substring you want to extract.

5 Type the code that uses the `substr` function.

```
Untitled - Notepad
File Edit Search Help
<html>
<head>
<title>Extract or Replace a Substring</title>
</head>
<body>

<?php
$file = "my_webpage.htm";
print "File Type: " . substr($file, -4);
print "<br>\n";
substr_replace($file, "index");
?>

</body>
</html>
```

### REPLACE A SUBSTRING

6 To replace a substring in a string, type `substr_replace()`.

7 Between the parentheses, type the name of the string variable in which you want to replace text.

8 Type a comma followed by the replacement text enclosed in quotation marks.

```
Untitled - Notepad
File Edit Search Help
<html>
<head>
<title>Extract or Replace a Substring</title>
</head>
<body>

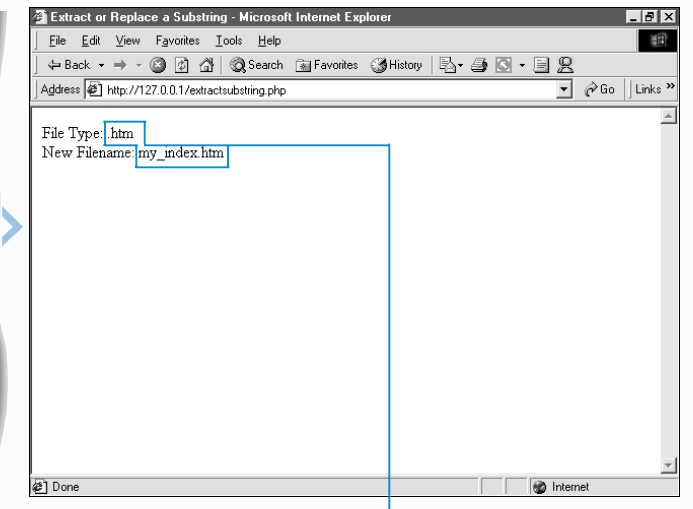
<?php
$file = "my_webpage.htm";
print "File Type: " . substr($file, -4);
print "<br>\n";
print "New Filename: " . substr_replace($file, "index", 3, 7);
?>

</body>
</html>
```

9 Type a comma followed by the offset of the substring you want to replace.

10 Type a comma followed by the length of the substring you want to replace.

11 Type the code that uses the `substr_replace` function.



12 Display the PHP page in a Web browser.

The Web browser displays the results of extracting and replacing substrings.

# SEARCH FOR A SUBSTRING OR CHARACTER

The `strpos` and `strrpos` functions allow you to search a string to find a substring or character you specify.

The `strpos` function allows you to search a string for the first occurrence of a substring and return the offset, or position, of the substring. If the substring cannot be found in the string, a value of false is returned.

To use the `strpos` function, you must specify the name of the variable containing the string you want to search and the substring you want to search for. The substring must be enclosed in quotation marks.

Although not required, PHP allows you to specify a starting offset when searching for a substring. When determining the starting offset, keep in mind that the characters in a string are numbered starting at zero (0). It is important to note that even if you specify a starting offset, the position of the substring returned by the function will still be relative to the beginning of the string.

The `strrpos` function allows you to search a string for the last occurrence of a character and return the offset of the character. If the character cannot be found in the string, a value of false is returned.

The `strrpos` function takes two arguments—the name of the variable containing the string you want to search and the character you want to search for, enclosed in quotation marks. If you specify more than one character, only the first character will be used in the search.

The `strpos` and `strrpos` functions are often used with an `if` statement to test whether the specified substring or character was found and then perform an action. When using an `if` statement, PHP could confuse the zero (0) position in a string with a value of false. To prevent this type of problem in your code, you should use the Identical to comparison operator (`===`) in the `if` statement.

## Apply It

You may want to use the `substr` function to extract a substring that is a specific number of characters from a substring or character you find using the `strpos` or `strrpos` function. In the example below, the extension is extracted from a filename.

The `strrpos` function is used to find the last occurrence of a dot (.) in the string. The return value is then increased by 1 and the new value is used as the offset where the `substr` function will extract the extension.

### TYPE THIS:

```
$file = "my_webpage.php";
$offset = strrpos($file, ".") + 1;
print "The file type is: " . substr($file, $offset);
```

### RESULT:

The file type is: php

You can use the `strpos` function in a `while` statement to find all the instances of a substring in a string. In the example below, the offset is advanced by one position every time the `while` loop is iterated.

### TYPE THIS:

```
$users = "Barry Prueett, Mary Corder, Ted Cains, Mary Bednarek, Mary Hicks";
$offset = 0;

while ($position !== false)
{
    $position = strpos($users, "Mary", $offset);
    if ($position === false) break;
    print "Mary found at position: $position<br>";
    $offset = $position + 1;
}
```

### RESULT:

Mary found at position: 14  
Mary found at position: 38  
Mary found at position: 53

## SEARCH FOR A SUBSTRING OR CHARACTER

```
Untitled - Notepad
File Edit Search Help
<html>
<head>
<title>Search for a Substring or Character</title>
</head>
<body>
<?php
$url = "http://www.maran.com/books/visual";
strpos($url, "maran");
?>
</body>
</html>
```

- 1 Type the code that creates a variable that stores the string you want to search.
- 2 To search for the first occurrence of a substring in a string, type `strpos()`.

- 3 Between the parentheses, type the name of the string variable you want to search followed by a comma.
- 4 Type the substring you want to search for, enclosed in quotation marks.

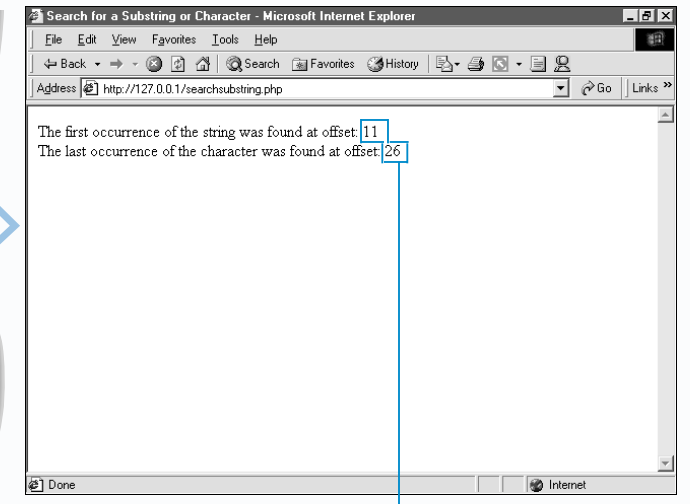
```
Untitled - Notepad
File Edit Search Help
<html>
<head>
<title>Search for a Substring or Character</title>
</head>
<body>
<?php
$url = "http://www.maran.com/books/visual";
$string = strpos($url, "maran", 7);
if ($string === false)
{
    print "The string was not found.<br>";
}
else
{
    print "The first occurrence of the string ";
    print "was found at offset: $string<br>";
}
?>
```

- 5 To specify a starting position for the search, type a comma followed by the offset in the string where you want to start the search.
- 6 Type the code that uses the `strpos` function.

```
Untitled - Notepad
File Edit Search Help
if ($string === false)
{
    print "The string was not found.<br>";
}
else
{
    print "The first occurrence of the string ";
    print "was found at offset: $string<br>";
}

$char = strrpos($url, "/");
if ($char === false)
{
    print "The character was not found.";
}
else
{
    print "The last occurrence of the character ";
    print "was found at offset: $char";
}
```

- 7 To search for the last occurrence of a character in a string, type `strrpos()`.
- 8 Between the parentheses, type the name of the string variable you want to search followed by a comma.
- 9 Type the character you want to search for, enclosed in quotation marks.
- 10 Type the code that uses the `strrpos` function.



- 11 Display the PHP page in a Web browser.
- The Web browser displays the results of searching for a substring and a character in a string.



## REPLACE DATA IN A STRING

You can use the `str_replace` function to replace all the occurrences of data in a string with another string. The `str_replace` function requires three arguments—the data to be replaced, the replacement string and the string that contains the data to be replaced. The result of the `str_replace` function can be directly displayed to a Web browser or assigned to a variable that can be further processed.

The `str_replace` function is useful for editing data that is read from a text file or will be written to a file. For example, you may want to correct the spelling of a name or update an e-mail address before processing the information.

You may want to replace specific characters in a string with another set of characters. This can be accomplished using the `strtr` function. To use the `strtr` function, you must first specify the string to be modified. You then specify the set of characters that you want to replace and the replacement characters. The last two parameters should

have the same number of characters. If the two parameters do not match in length, the longer parameter will be truncated to match the number of characters in the shorter parameter.

The `strtr` function is useful for correcting data received from a form or a text file. For example, if a user enters data containing accented letters into a form, you may want to replace the accented letters with regular letters before processing the data.

When the `strtr` function is used, each character in the string is examined in sequence and replaced accordingly. After the string has been completely processed, no further replacements are performed.

You may also use the `substr_replace` function to replace data in a string. For information about the `substr_replace` function, see page 116.

### Extra

The `str_replace` function is case-sensitive and will only replace portions of the string that perfectly match the data to be replaced. To perform a case-insensitive replacement of data, you can use the `eregi_replace` function, which takes the same arguments as the `str_replace` function.

#### TYPE THIS:

```
$text = "Mary had a little lAmb, little lamB, little LaMb.";
print eregi_replace("lamb", "sheep", $text);
```

#### RESULT:

Mary had a little sheep, little sheep, little sheep.

The `strtr` function may also be used to replace multiple words in a string. Instead of specifying the sets of characters to be searched and replaced, you can specify an associative array as the second argument of the `strtr` function. The instances of the array keys found in the string are replaced by the corresponding array values.

#### TYPE THIS:

```
$oldList = "apple, banana, melon, grape";
$replacement = array("apple" => "melon", "melon" => "mango");
$newList = strtr($oldList, $replacement);
print "Old list: $oldList<br>";
print "New list: $newList<br>";
```

#### RESULT:

Old list: apple, banana, melon, grape  
New list: melon, banana, mango, grape

### REPLACE DATA IN A STRING

```
Untitled - Notepad
File Edit Search Help
<html>
<head>
<title>Replace Data in a String</title>
</head>
<body>
<?php
$text1 = "This Web site contains html documents, such as index.html";
$text2 = "this is a secret message";
str_replace()
?>
</body>
</html>
```

#### USING THE `str_replace` FUNCTION

1 Type the code that creates variables that contain strings in which you want to replace data.

2 To replace data in a string, type `str_replace()`.

```
Untitled - Notepad
File Edit Search Help
<html>
<head>
<title>Replace Data in a String</title>
</head>
<body>
<?php
$text1 = "This Web site contains html documents, such as index.html";
$text2 = "this is a secret message";
print str_replace("html", "xml", $text1) . "<p>";
?>
</body>
</html>
```

3 Between the parentheses, type the data you want to replace, enclosed in quotation marks.

4 Type a comma followed by the replacement string, enclosed in quotation marks.

5 Type a comma followed by the name of the string variable in which you want to replace data.

6 Type the code that uses the result of the `str_replace` function.

```
Untitled - Notepad
File Edit Search Help
<html>
<head>
<title>Replace Data in a String</title>
</head>
<body>
<?php
$text1 = "This Web site contains html documents, such as index.html";
$text2 = "this is a secret message";
print str_replace("html", "xml", $text1) . "<p>";
print "The following line has been scrambled: ";
print strtr($text2, "aceghimrst", "tsrmihgeca");
?>
</body>
</html>
```

#### USING THE `strtr` FUNCTION

7 To replace characters in a string, type `strtr()`.

8 Between the parentheses, type the name of the string variable in which you want to replace characters.

9 Type a comma followed by the characters you want to replace, enclosed in quotation marks.

10 Type a comma followed by the replacement characters, enclosed in quotation marks.

11 Type the code that uses the result of the `strtr` function.

```
Replace Data in a String - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Forward Stop Home Search Favorites History
Address http://127.0.0.1/replacedata.php Go Links
This Web site contains xml documents, such as index.xml
The following line has been scrambled: ahhc hc t crsera grectmr
Done Internet
```

12 Display the PHP page in a Web browser.

The Web browser displays the results of replacing data and characters in a string.

# WORK WITH HTML TEXT

The `htmlspecialchars` and `htmlentities` functions allow you to convert special characters in a string to their character entity reference. The character entity reference consists of an ampersand, the name of the character and a semicolon. You can convert characters in order to force a Web browser to display HTML code as plain text on a PHP page instead of processing the code. This is useful when you want to block HTML code a user enters in a form that might interfere with the processing of a Web page.

The `htmlspecialchars` function converts only the most common HTML characters to their character entity reference values. The `htmlentities` function is similar to the `htmlspecialchars` function, except the `htmlentities` function converts all the characters that have an equivalent character entity reference in HTML.

To use the `htmlspecialchars` and `htmlentities` functions, you specify the string to be converted or a variable that stores the string. Each function will return a string with the corresponding character entity references for the HTML characters.

You can use the `get_html_translation_table` function to display a list of the characters that the `htmlspecialchars` or `htmlentities` function will convert. As an argument, you specify either `HTML_SPECIALCHARS` or `HTML_ENTITIES`. The function returns an associative array that has the actual character to be converted as the key and the character entity reference as the value for each element.

PHP allows you to use an optional second argument with the `htmlspecialchars`, `htmlentities` and `get_html_translation_table` functions to specify how quotation marks should be handled. By default, PHP will use the `ENT_COMPAT` value, which converts only double quotation marks to their HTML entity reference. To convert both double and single quotation marks, use the `ENT_QUOTES` value. If you specify the `ENT_NOQUOTES` value, neither double nor single quotation marks will be converted.

## Extra

You can use the `strip_tags` function to remove all HTML formatting from a file or to prevent code a user inputs from being processed. The `strip_tags` function removes all the HTML and PHP tags from a string and then returns the result as plain text that can be easily processed or manipulated.

### TYPE THIS:

```
$htmlSource = "<html><head><title>ABC Corporation</title></head><body><h1>Welcome to Our Web Site.</h1></body></html>";
$rawText = strip_tags($htmlSource);
print $rawText;
```

### RESULT:

ABC Corporation Welcome to Our Web Site.

The `nl2br` function allows you to convert newline characters (`\n`) in a string to HTML `<br>` tags. For example, you can use the `nl2br` function to convert a newline character a user types in a text area on a form to a `<br>` tag that a Web browser can process and display as a line break on a Web page. The `nl2br` function takes as its argument the string that contains the newline character to be converted or a variable that stores the string.

### TYPE THIS:

```
$userInput = "Hello!\nMy name is Tom!";
$formattedText = nl2br($userInput);
print $formattedText;
```

### RESULT:

Hello!  
My name is Tom!

## WORK WITH HTML TEXT

```
<html>
<head>
<title>Work with HTML Text</title>
</head>
<body>

<h2>Welcome to my Web page.</h2>

<?php
$userInput = "Hello! My name is <font size = \'7\'>Tom!</font>";
htmlspecialchars($userInput);
?>

<p><hr></p>

<?php
?>

</body>
</html>
```

### CONVERT HTML CHARACTERS

1 Type the code that creates a variable that contains HTML code.

2 To convert HTML characters to their character entity reference, type `htmlspecialchars()` or `htmlentities()`.

3 Between the parentheses, type the name of the variable that contains HTML code.

```
<html>
<head>
<title>Work with HTML Text</title>
</head>
<body>

<h2>Welcome to my Web page.</h2>

<?php
$userInput = "Hello! My name is <font size = \'7\'>Tom!</font>";
print htmlspecialchars($userInput, ENT_NOQUOTES);
?>

<p><hr></p>

<?php
get_html_translation_table(HTML_SPECIALCHARS)
?>

</body>
</html>
```

4 To specify how you want the function to convert quotes, type a comma followed by the value you want to use (`ENT_NOQUOTES`, `ENT_COMPAT` or `ENT_QUOTES`).

5 Type the code that uses the results of the conversions.

### DISPLAY THE TRANSLATION TABLE

6 To view the characters that will be converted, type `get_html_translation_table()`.

7 Between the parentheses, type the value for the translation table you want to display (`HTML_SPECIALCHARS` or `HTML_ENTITIES`).

```
<h2>Welcome to my Web page.</h2>

<?php
$userInput = "Hello! My name is <font size = \'7\'>Tom!</font>";
print htmlspecialchars($userInput, ENT_NOQUOTES);
?>

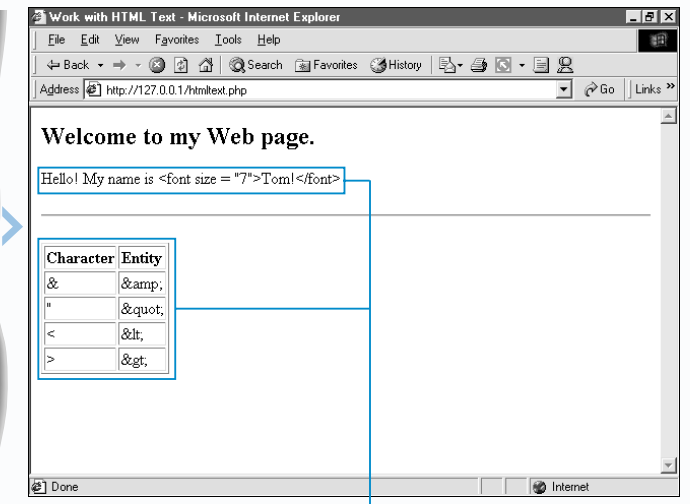
<p><hr></p>

<?php
print "<table border = \'1\'> ";
print "<tr><th>Character</th><th>Entity</th></tr> ";
foreach(get_html_translation_table(HTML_SPECIALCHARS) as $character)
{
    $entity = htmlspecialchars($character);
    print "<tr><td>$character</td><td>$entity</td></tr>";
}
print "</table>";
?>
```

8 To display the character entity reference values in a translation table, type a variable name followed by `htmlspecialchars()`.

9 Between the parentheses, type the name of the variable that stores the characters that will be converted.

10 Type the code that formats the result of the `get_html_translation_table` function in a table.



11 Display the PHP page in a Web browser.

The Web browser displays the result of converting HTML characters and displays a translation table for the function you specified.