



- [Downloads](#)
- [Documentation](#)
- [Get Involved](#)
- [Help](#)
- 

[Dutch PHP Conference 2024](#)

## [Getting Started](#)

[Introduction](#)

[A simple tutorial](#)

## [Language Reference](#)

[Basic syntax](#)

[Types](#)

[Variables](#)

[Constants](#)

[Expressions](#)

[Operators](#)

[Control Structures](#)

[Functions](#)

[Classes and Objects](#)

[Namespaces](#)

[Enumerations](#)

[Errors](#)

[Exceptions](#)

[Fibers](#)

[Generators](#)

[Attributes](#)

[References Explained](#)

[Predefined Variables](#)

[Predefined Exceptions](#)

[Predefined Interfaces and Classes](#)

[Predefined Attributes](#)

[Context options and parameters](#)

[Supported Protocols and Wrappers](#)

## [Security](#)

[Introduction](#)

[General considerations](#)

[Installed as CGI binary](#)

[Installed as an Apache module](#)

[Session Security](#)

[Filesystem Security](#)

[Database Security](#)

[Error Reporting](#)

[User Submitted Data](#)

[Hiding PHP](#)

[Keeping Current](#)

## [Features](#)

[HTTP authentication with PHP](#)

[Cookies](#)

[Sessions](#)

[Dealing with XForms](#)

[Handling file uploads](#)

[Using remote files](#)

[Connection handling](#)

[Persistent Database Connections](#)

[Command line usage](#)

[Garbage Collection](#)  
[DTrace Dynamic Tracing](#)

## [Function Reference](#)

[Affecting PHP's Behaviour](#)  
[Audio Formats Manipulation](#)  
[Authentication Services](#)  
[Command Line Specific Extensions](#)  
[Compression and Archive Extensions](#)  
[Cryptography Extensions](#)  
[Database Extensions](#)  
[Date and Time Related Extensions](#)  
[File System Related Extensions](#)  
[Human Language and Character Encoding Support](#)  
[Image Processing and Generation](#)  
[Mail Related Extensions](#)  
[Mathematical Extensions](#)  
[Non-Text MIME Output](#)  
[Process Control Extensions](#)  
[Other Basic Extensions](#)  
[Other Services](#)  
[Search Engine Extensions](#)  
[Server Specific Extensions](#)  
[Session Extensions](#)  
[Text Processing](#)  
[Variable and Type Related Extensions](#)  
[Web Services](#)  
[Windows Only Extensions](#)  
[XML Manipulation](#)  
[GUI Extensions](#)

## Keyboard Shortcuts

?	This help
j	Next menu item
k	Previous menu item
g p	Previous man page
g n	Next man page
G	Scroll to bottom
g g	Scroll to top
g h	Goto homepage
g s	Goto search (current page)
/	Focus search box

[Массивы »](#)

[« Логика](#)

- [Руководство по PHP](#)
- [Справочник языка](#)
- [Операторы](#)

Change language: Russian

# Строковые операторы

В PHP есть два оператора для работы со строками (string). Первый — оператор конкатенации («.»), который возвращает строку, представляющую собой соединение левого и правого аргумента. Второй — оператор присваивания с конкатенацией ('.='), который присоединяет правый аргумент к левому. Подробно об этом рассказано в разделе «[Операторы присваивания](#)».

```
<?php

$a = "Привет, ";
$b = $a . "Мир!"; // $b теперь содержит строку «Привет, Мир!»

$a = "Привет, ";
$a .= "Мир!"; // $a теперь содержит строку «Привет, Мир!»
```

## Смотрите также

- [Строки](#)
- [Функции для работы со строками](#)

[+add a note](#)

## User Contributed Notes 6 notes

[up](#)  
[down](#)  
240  
[K.Alex ¶](#)  
**11 years ago**  
As for me, curly braces serve good substitution for concatenation, and they are quicker to type and code looks cleaner. Remember to use double quotes (" ") as their content is parced by php, because in single quotes (' ') you'll get literal name of variable provided:

```
<?php

$a = '12345';

// This works:
echo "qwe{$a}rty"; // qwe12345rty, using braces
echo "qwe" . $a . "rty"; // qwe12345rty, concatenation used

// Does not work:
echo 'qwe{$a}rty'; // qwe{$a}rty, single quotes are not parsed
echo "qwe$aarty"; // qwe, because $a became $arty, which is undefined
```

?>  
[up](#)  
[down](#)  
164  
[anders dot benke at telia dot com ¶](#)  
**19 years ago**

A word of caution - the dot operator has the same precedence as + and -, which can yield unexpected results.

Example:

```
<php

$var = 3;

echo "Result: " . $var + 3;

?>
```

The above will print out "3" instead of "Result: 6", since first the string "Result3" is created and this is then added to 3 yielding 3, non-empty non-numeric strings being converted to 0.

To print "Result: 6", use parantheses to alter precedence:

```
<php
$var = 3;

echo "Result: " . ($var + 3);
?>
```

[up](#)  
[down](#)

108

[Stephen Clay](#)

18 years ago

```
<?php
"${str1}${str2}${str3}"; // one concat = fast
$str1. $str2. $str3; // two concats = slow
?>
```

Use double quotes to concat more than two strings instead of multiple '.' operators. PHP is forced to re-concatenate with every '.' operator.

[up](#)  
[down](#)

89

[hexidecimalgadget at hotmail dot com](#)

15 years ago

If you attempt to add numbers with a concatenation operator, your result will be the result of those numbers as strings.

```
<?php

echo "thr"."ee"; //prints the string "three"
echo "twe" . "lve"; //prints the string "twelve"
echo 1 . 2; //prints the string "12"
echo 1.2; //prints the number 1.2
echo 1+2; //prints the number 3
```

[up](#)  
[down](#)

5

[biziclop](#)

1 year ago

Some bitwise operators (the and, or, xor and not operators: & | ^ ~ ) also work with strings too since PHP4, so you don't have to loop through strings and do chr(ord(\$s[i])) like things.

See the documentation of the bitwise operators: <https://www.php.net/operators.bitwise>

```
<?php var_dump(
('23456787654' ^ 'zVXYyhoXDYP'), // 'Hello_World'
('(!($)^!)@$_' | '@ddhfIvn2H$'), // 'hello_world'
('{ } ~ | o ! Wo { }' & 'Lgmno|Wovmf'), // 'Hello World'
(~'<0-14)(98' & '}}}}}}}}}') // 'AMPLITUDE'
); ?>
```

Live demo: <https://3v41.org/MnFeb>

[up](#)  
[down](#)

36

[mariusads::at::helpedia.com](#)

15 years ago

Be careful so that you don't type "." instead of ";" at the end of a line.

It took me more than 30 minutes to debug a long script because of something like this:

```
<?
echo 'a'.
$c = 'x';
echo 'b';
echo 'c';
?>
```

The output is "axbc", because of the dot on the first line.

[+add a note](#)

- [Операторы](#)
  - [Приоритет](#)
  - [Арифметика](#)
  - [Инкремент и декремент](#)
  - [Присваивание](#)
  - [Побитовые операторы](#)
  - [Сравнение](#)
  - [Управление ошибками](#)
  - [Исполнение](#)
  - [Логика](#)
  - [Строки](#)
  - [Массивы](#)
  - [Проверка типа](#)
- [Copyright © 2001-2024 The PHP Group](#)
- [My PHP.net](#)
- [Contact](#)
- [Other PHP.net sites](#)
- [Privacy policy](#)

