



- [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

[Присваивание »](#)

[« Арифметика](#)

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

Change language: Russian

Операторы инкремента и декремента

PHP поддерживает префиксные и постфиксные операторы инкремента и декремента. Эти унарные операторы разрешают увеличивать или уменьшать значение на единицу.

Операторы инкремента и декремента		
Пример	Название	Действие
++\$a	Префиксный инкремент	Увеличивает <i>\$a</i> на единицу, затем возвращает значение <i>\$a</i> .
\$a++	Постфиксный инкремент	Возвращает значение <i>\$a</i> , затем увеличивает <i>\$a</i> на единицу.
--\$a	Префиксный декремент	Уменьшает <i>\$a</i> на единицу, затем возвращает значение <i>\$a</i> .
\$a--	Постфиксный декремент	Возвращает значение <i>\$a</i> , затем уменьшает <i>\$a</i> на единицу.

Приведём пример простого скрипта:

```
<?php

echo 'Постфиксный инкремент:', PHP_EOL;
$a = 5;
var_dump($a++);
var_dump($a);

echo 'Префиксный инкремент:', PHP_EOL;
$a = 5;
var_dump(++$a);
var_dump($a);

echo 'Постфиксный декремент:', PHP_EOL;
$a = 5;
var_dump($a--);
var_dump($a);

echo 'Префиксный декремент:', PHP_EOL;
$a = 5;
var_dump(--$a);
var_dump($a);
```

Результат выполнения приведённого примера:

```
Постфиксный инкремент:
int(5)
int(6)
Префиксный инкремент:
int(6)
int(6)
Постфиксный декремент:
int(5)
int(4)
Префиксный декремент:
int(4)
int(4)
```

Внимание

Операторы инкремента и декремента не влияют на логические значения (bool). Начиная с PHP 8.3.0 вызывается ошибка уровня **E_WARNING**, потому что в будущем это будет неявно приведено к значению целого числа (int).

Оператор декремента не влияет на значения null. Начиная с PHP 8.3.0 вызывается ошибка уровня **E_WARNING**, потому что в будущем это будет неявно приведено к значению целого числа (int).

Оператор декремента не влияет на не- [числовые строки](#). Начиная с PHP 8.3.0 вызывается ошибка уровня **E_WARNING**, потому что в будущем будет выбрасываться исключение [TypeError](#).

Замечание:

Внутренние объекты, которые поддерживают перегрузку сложения и/или вычитания, могут также быть инкрементированы и/или декрементированы. Один такой внутренний объект — это [GMP](#).

Функция увеличения строки PERL

Внимание

Начиная с PHP 8.3.0 эта функция мягко устарела. Вместо этого вызывают функцию [str_increment\(\)](#).

В PHP можно инкрементировать не- [числовую строку](#). Строка должна быть буквенно-цифровой последовательностью в кодировке ASCII. Буквы увеличиваются до следующей буквы, при достижении буквы z инкремент переносится в следующий разряд слева. Например, выражение `$a = 'Z'; $a++;` превратит значение переменной `$a` в значение «AA».

Пример #1 Пример увеличения строки PERL

```
<?php
```

```
echo '== Буквенные строки ==' . PHP_EOL;
$s = 'W';
for ($n=0; $n<6; $n++) {
echo ++$s . PHP_EOL;
}
// Буквенно-цифровые строки ведут себя иначе
echo '== Буквенно-цифровые строки ==' . PHP_EOL;
$d = 'A8';
for ($n=0; $n<6; $n++) {
echo ++$d . PHP_EOL;
}
$d = 'A08';
for ($n=0; $n<6; $n++) {
echo ++$d . PHP_EOL;
}
```

Результат выполнения приведённого примера:

```
== Буквенные строки ==
X
Y
Z
AA
AB
AC
== Буквенно-цифровые строки ==
A9
B0
B1
B2
B3
B4
A09
A10
A11
A12
A13
A14
```

Внимание

Если буквенно-цифровая строка может быть интерпретирована как [числовая строка](#), она будет приведена к типу `int` или `float`. Эта проблема встаёт особенно остро со строками, которые выглядят как числа с плавающей точкой, записанные в научной нотации. Функция [str_increment\(\)](#) не страдает от эти неявных приведений типов.

Пример #2 Буквенно-цифровая строка, преобразованная в число с плавающей точкой

```
<?php
```

```
$s = "5d9";
var_dump(++$s);
var_dump(++$s);
```

Результат выполнения приведённого примера:

```
string(3) "5e0"  
float(6)
```

Так происходит потому, что значение «5e0» было интерпретировано как число с плавающей точкой (float) и приведено к значению 5.0 перед началом увеличения.

[+ add a note](#)

User Contributed Notes 1 note

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

69
[hartmut at php dot net ¶](#)
11 years ago

Note that

```
$a="9D9"; var_dump(++$a); => string(3) "9E0"
```

but counting onwards from there

```
$a="9E0"; var_dump(++$a); => float(10)
```

this is due to "9E0" being interpreted as a string representation of the float constant 9E0 (or 9e0), and thus evalutes to $9 * 10^0 = 9$ (in a float context)

[+ 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](#)

