



- [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 »](#)

[« Строки](#)

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

Change language: Russian

Числовые строки

В PHP строка (string) считается числовой, если её можно интерпретировать как целое (int) число или как число с плавающей точкой (float).

Формально с PHP 8.0.0:

```
WHITESPACES    \s*
LNUM           [0-9]+
DNUM           ([0-9]*[\.]{LNUM}) | ({LNUM}[\.][0-9]*)
EXPONENT_DNUM  (({LNUM} | {DNUM}) [eE][+-]? {LNUM})
INT_NUM_STRING {WHITESPACES} [+]? {LNUM} {WHITESPACES}
FLOAT_NUM_STRING {WHITESPACES} [+]? ({DNUM} | {EXPONENT_DNUM}) {WHITESPACES}
NUM_STRING     ({INT_NUM_STRING} | {FLOAT_NUM_STRING})
```

В PHP также присутствует концепция *префиксной* числовой строки. Это строка, которая начинается как числовая и продолжается любыми другими символами.

Замечание:

Любая строка, содержащая букву *e* (без учёта регистра), ограниченную цифрами, будет восприниматься как число, выраженное в научной нотации. Это может привести к неожиданным результатам.

```
<?php
```

```
var_dump("0D1" == "000"); // false, «0D1» — не научная нотация
var_dump("0E1" == "000"); // true, «0E1» — это 0 * (10 ^ 1) или 0
var_dump("2E1" == "020"); // true, «2E1» — это 2 * (10 ^ 1) или 20
?>
```

Строки, используемые в числовых контекстах

Когда строку необходимо использовать в качестве числа (например арифметические операции, декларация целочисленного типа, и т. д.), используется следующий алгоритм действий:

1. Если строка числовая, представляет целое число и не превышает максимально допустимого значения для типа `int` (определённого в `PHP_INT_MAX`), то она приводится к типу `int`. Иначе она приводится к типу `float`.
2. Если в заданном контексте дозвоительно использовать префиксную числовую строку, то, если начало строки представляет целое число и не превышает максимально допустимого значения для типа `int` (определённого в `PHP_INT_MAX`), то она приводится к типу `int`. Иначе она приводится к типу `float`. Также, в этом случае, выдаётся ошибка уровня `E_WARNING`.
3. Если строка не числовая - выбрасывается исключение [TypeError](#).

Поведение до PHP 8.0.0

До PHP 8.0.0 строка считалась числовой только в случае, если она *начиналась* с пробельных символов. Если она *завершалась* пробельными символами — она считалась префиксной числовой.

До PHP 8.0.0, когда строку необходимо использовать как число, использовался тот же алгоритм, что и описан выше, но с некоторыми отличиями:

- Использование префиксной числовой строки вызывало ошибку уровня `E_NOTICE`, а не `E_WARNING`.
- Если строка не являлась числовой, вызывалась ошибка уровня `E_WARNING`, а сама строка приводилась к числу `0`.

До PHP 7.1.0 не вызывалась ошибка уровня ни `E_NOTICE`, ни `E_WARNING`.

```
<?php
```

```
$foo = 1 + "10.5"; // $foo — число с плавающей точкой (11.5)
$foo = 1 + "-1.3e3"; // $foo — число с плавающей точкой (-1299)
$foo = 1 + "bob-1.3e3"; // TypeError начиная с PHP 8.0.0. Ранее $foo принималось за целое число (1)
$foo = 1 + "bob3"; // TypeError начиная с PHP 8.0.0, Ранее $foo принималось за целое число (1)
$foo = 1 + "10 Small Pigs"; // $foo — целое (11). В PHP 8.0.0 выдаётся ошибка уровня E_WARNING, а в более ранних версиях — уровня E_NOTICE
$foo = 4 + "10.2 Little Piggies"; // $foo — число с плавающей точкой (14.2). В PHP 8.0.0 выдаётся ошибка уровня E_WARNING, а в более ранних версиях — уровня E_NOTICE
```

```
$foo = "10.0 pigs " + 1; // $foo – число с плавающей точкой (11). В PHP 8.0.0 выдаётся ошибка уровня E_WARNING, а в более ранних версиях – уровня E_NOTICE
$foo = "10.0 pigs " + 1.0; // $foo – число с плавающей точкой (11). В PHP 8.0.0 выдаётся ошибка уровня E_WARNING, а в более ранних версиях – уровня E_NOTICE
?>
```

[+add a note](#)

User Contributed Notes

There are no user contributed notes for this page.

- [Типы](#)
 - [Введение](#)
 - [Система типов](#)
 - [NULL](#)
 - [Логические значения](#)
 - [Целые числа](#)
 - [Числа с плавающей точкой](#)
 - [Строки](#)
 - [Числовые строки](#)
 - [Массивы](#)
 - [Объекты](#)
 - [Перечисления](#)
 - [Ресурсы](#)
 - [Callable и callback-функции](#)
 - [Mixed](#)
 - [Void](#)
 - [Never](#)
 - [Относительные типы классов](#)
 - [Типы значений](#)
 - [Итерируемые значения](#)
 - [Объявления типов](#)
 - [Манипуляции с типами](#)
- [Copyright © 2001-2024 The PHP Group](#)
- [My PHP.net](#)
- [Contact](#)
- [Other PHP.net sites](#)
- [Privacy policy](#)

