



- [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

[Система типов »](#)

[« Типы](#)

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

Change language: Russian

Введение

У каждого выражения в PHP один из следующих встроенных типов в зависимости от его значения:

- null
- bool
- int
- float (floating-point number)
- string
- array
- object
- [callable](#)
- resource

PHP – динамически типизированный язык, что означает, что по умолчанию нет необходимости указывать тип переменной, так как он будет определён во время выполнения. Однако можно статически типизировать некоторые аспекты языка, используя [декларации типов](#).

Типы ограничивают тип операций, которые могут быть выполнены над ними. Однако, если выражение/переменная используется в операции, которую не поддерживает её тип, PHP попытается [преобразовать](#) значение в тип, который поддерживает операцию. Этот процесс зависит от контекста, в котором используется значение. Для получения дополнительной информации смотрите раздел [Манипуляции с типами](#).

Подсказка

[Таблицы сравнения типов](#) также могут быть полезны, поскольку в них представлены различные примеры сравнения значений разных типов.

Замечание: Можно заставить выражение оцениваться с определённым типом, используя [приведение типов](#). Переменная также может быть приведена к определённому типу с помощью функции [settype\(\)](#).

Чтобы проверить значение и тип [выражения](#), используйте функцию [var_dump\(\)](#). Чтобы получить тип [выражения](#), используйте функцию [get_debug_type\(\)](#). Однако, чтобы проверить, является ли выражение определённым типом, используйте функции `is_type`.

```
<?php
$a_bool = true; // логическое значение
$a_str = "foo"; // строка
$a_str2 = 'foo'; // строка
$a_int = 12; // целое число

echo get_debug_type($a_bool), "\n";
echo get_debug_type($a_str), "\n";

// Если это целое число, увеличить на четыре
if (is_int($a_int)) {
    $a_int += 4;
}
var_dump($a_int);

// Если $a_bool - это строка, вывести её
if (is_string($a_bool)) {
    echo "Строка: $a_bool";
}
?>
```

Результат выполнения приведённого примера в PHP 8:

```
bool
string
int(16)
```

Замечание: Вместо функции [get_debug_type\(\)](#), которая была недоступна до PHP 8.0.0, вызывали функцию

[gettype\(\)](#). Однако она не возвращает канонические имена типов.

[+ 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](#)

