



- [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

[Возврат к глобальному пространству »](#)  
[« Псевдонимирование и импорт](#)

- [Руководство по PHP](#)
- [Справочник языка](#)
- [Пространства имён](#)

Change language: Russian

# Глобальное пространство

(PHP 5 >= 5.3.0, PHP 7, PHP 8)

Без определения пространства имён определения классов и функций помещаются в глобальном пространстве — так же, как это было в PHP до введения пространств имён. Добавление префикса \ к именам указывает, что требуется имя из глобального пространства, даже в контексте пространства имён.

## Пример #1 Спецификация глобального пространства

```
<?php

namespace A\B\C;

/* Функция — это A\B\C\fopen */
function fopen() {
    /* ... */
    $f = \fopen(...); // Вызов глобальной функции fopen
    return $f;
}

?>
```

[+add a note](#)

## User Contributed Notes 6 notes

[up](#)  
[down](#)  
58

[routinet ¶](#)  
**12 years ago**

Included files will default to the global namespace.

```
<?php
//test.php
namespace test {
include 'test1.inc';
echo '- ',__NAMESPACE__,'-<br />';
}
?>
```

```
<?php
//test1.inc
echo '- ',__NAMESPACE__,'-<br />';
?>
```

Results of test.php:

--  
-test-  
[up](#)  
[down](#)  
3

[PhonexSegovia at gmail dot com ¶](#)  
**1 year ago**

Note that variables aren't part of the namespace so they are always global (or scoped to function, etc.) and can't be accessed the same way as other namespace stuff.

So no:

```
namespace Foo;
```

```
$var = "hello";  
echo \Foo\ $var;
```

[up](#)

[down](#)

14

[nobody at example dot com ¶](#)

**8 years ago**

In namespaced context the Exception class needs to be prefixed with global prefix operator.

```
<?php
```

```
namespace hey\ho\lets\go;
```

```
class MyClass  
{  
    public function failToCatch()  
    {  
        try {  
            $thing = somethingThrowingAnException();  
        } catch (Exception $ex) {  
            // Not caught  
        }  
    }  
}
```

```
public function succeedToCatch()  
{  
    try {  
        $thing = somethingThrowingAnException();  
    } catch (\Exception $ex) {  
        // This is now caught  
    }  
}
```

[up](#)

[down](#)

1

[dmc60 at cam dot ac dot uk ¶](#)

**26 days ago**

To define an associative array (hash) so that it is part of the namespace, instead of going into the global namespace, just declare it as const, instead of as a variable.

This is handy for lookup tables, config settings, etc.

```
// Example. Instead of writing:
```

```
$my_datatypes = [  
    "sterility" => [  
        "xlsx" => [  
            "Sample Type",  
            "Run Pass/Fail",  
            "Result"  
        ],  
        "db" => [  
            "SampleType",  
            "RunPassFail",  
            "Result"  
        ]  
    ]  
];
```

```
// ...declare the lookup table like this:
```

```
const MY_DATATYPES = [  
  "sterility" => [  
    "xlsx" => [  
      "Sample Type",  
      "Run Pass/Fail",  
      "Result"  
    ],  
    "db" => [  
      "SampleType",  
      "RunPassFail",  
      "Result"  
    ]  
  ]  
];
```

// ...and it will be declared within the current namespace.

[up](#)

[down](#)

-29

[hileon at gmail dot com ¶](#)

**8 years ago**

This design of access globe name is bad, at least inconvenient.

In case that you have some legacy code, and try to copy and re-use it under some namespace. You have to add prefix "\" on any global function call.

[up](#)

[down](#)

-43

[xmarcos at gmail dot com ¶](#)

**11 years ago**

That's the expected behavior, you have to declare the namespace at the top of the file to "extend" it.

If you include a global namespaced file, it will operate on the global namespace.

[+ add a note](#)

- [Пространства имён](#)
  - [Обзор](#)
  - [Пространства имён](#)
  - [Подпространства имён](#)
  - [Несколько пространств имён в одном файле](#)
  - [Основы](#)
  - [Пространства имён и динамические особенности языка](#)
  - [Ключевое слово namespace и константа \\_\\_NAMESPACE\\_\\_](#)
  - [Псевдонимирование и импорт](#)
  - [Глобальное пространство](#)
  - [Возврат к глобальному пространству](#)
  - [Правила разрешения имён](#)
  - [FAQ](#)
- [Copyright © 2001-2024 The PHP Group](#)
- [My PHP.net](#)
- [Contact](#)
- [Other PHP.net sites](#)
- [Privacy policy](#)

