



- [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

[Объяснение ссылок »](#)

[« Чтение атрибутов с помощью Reflection API](#)

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

Change language: Russian

Объявление классов атрибутов

Хотя и нет строго требования, лучше выполнять рекомендацию — создавать класс для каждого атрибута. В самом простом случае необходимо создать пустой класс с атрибутом `#[Attribute]`, класс которого можно импортировать из глобального пространства имён через оператор `use`.

Пример #1 Простой класс с атрибутом

```
<?php

namespace Example;

use Attribute;

#[Attribute]
class MyAttribute
{
}
```

Чтобы ограничить типы сущностей, на которые можно будет нацелить атрибут, необходимо в момент объявления атрибута `#[Attribute]` передать в качестве первого аргумента битовую маску.

Пример #2 Спецификация указания целей, которым атрибут может быть присвоен

```
<?php

namespace Example;

use Attribute;

#[Attribute(Attribute::TARGET_METHOD | Attribute::TARGET_FUNCTION)]
class MyAttribute
{
}
```

После такого декларирования попытка присвоить атрибут **MyAttribute** другой сущности, тип которой отличается от метода или функции, приведёт к выбрасыванию исключения при вызове [ReflectionAttribute::newInstance\(\)](#).

Можно указать следующие цели:

- `Attribute::TARGET_CLASS`
- `Attribute::TARGET_FUNCTION`
- `Attribute::TARGET_METHOD`
- `Attribute::TARGET_PROPERTY`
- `Attribute::TARGET_CLASS_CONSTANT`
- `Attribute::TARGET_PARAMETER`
- `Attribute::TARGET_ALL`

По умолчанию атрибут можно присвоить сущности только один раз. Присвоить одинаковые атрибуты одной сущности можно, если объявить атрибут `#[Attribute]` с флагом `Attribute::IS_REPEATABLE` в битовой маске.

Пример #3 Применение константы IS_REPEATABLE при объявлении атрибута для разрешения его многократного присваивания

```
<?php

namespace Example;

use Attribute;

#[Attribute(Attribute::TARGET_METHOD | Attribute::TARGET_FUNCTION | Attribute::IS_REPEATABLE)]
class MyAttribute
{
}
```

```
}
```

[+add a note](#)

User Contributed Notes 1 note

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

7

[esdras-schonevald ¶](#)

2 years ago

```
#! Require PHP >= 8.0
```

```
#! This is a Sample
```

```
<?php
```

```
declare(strict_types = 1);
```

```
#[Attribute]
```

```
class Foo
```

```
{
```

```
function __construct(){
```

```
echo "Running " . __METHOD__ . PHP_EOL;
```

```
}
```

```
}
```

```
#[Attribute(Attribute::TARGET_CLASS|Attribute::IS_REPEATABLE)]
```

```
class Bar {
```

```
function __construct(?string ...$args){
```

```
echo "Running " . __METHOD__ ,
```

```
" args: " . implode(", ", $args) . PHP_EOL;
```

```
}
```

```
}
```

```
#[Attribute(Attribute::TARGET_ALL)]
```

```
class Baz {
```

```
function __construct(
```

```
private string $parameter
```

```
){
```

```
echo "Running " . __METHOD__ ,
```

```
" arg: " . $this->parameter . PHP_EOL;
```

```
}
```

```
}
```

```
#[Foo] // [0]
```

```
#[Bar] // [1]
```

```
#[Bar("Banana")] // [2]
```

```
#[Bar("Banana", "Apple", "Lemon", "Grape")] // [3]
```

```
#[Baz("The Only One")] // [4]
```

```
class Qux
```

```
{
```

```
}
```

```
// Getting class attribute with ReflectionClass
```

```
$ref = new ReflectionClass(Qux::class);
```

```
$attrs = $ref->getAttributes(); // Array of attributes
```

```
$attrs[0]->newInstance(); // "Running Foo::__construct"
```

```
$attrs[1]->newInstance(); // "Running Bar::__construct args: "
```

```
$attrs[2]->newInstance(); // "Running Bar::__construct args: Banana"
```

```
$attrs[3]->newInstance(); // "Running Bar::__construct args: Banana, Apple, Lemon, Grape"
```

```
$attrs[4]->newInstance(); // "Running Baz::__construct arg: The Only One"
```

[+ add a note](#)

- [Атрибуты](#)
 - [Введение в атрибуты](#)
 - [Синтаксис атрибутов](#)
 - [Чтение атрибутов с помощью Reflection API](#)
 - [Объявление классов атрибутов](#)
- [Copyright © 2001-2024 The PHP Group](#)
- [My PHP.net](#)
- [Contact](#)
- [Other PHP.net sites](#)
- [Privacy policy](#)

