



- [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

[Чтение атрибутов с помощью Reflection API »](#)
[« Введение в атрибуты](#)

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

Change language: Russian

Синтаксис атрибутов

Синтаксис атрибутов состоит из нескольких частей. Во-первых, декларация атрибута начинается с символа `#[` и заканчивается символом `]`. Во-вторых, внутри перечисляют один или несколько разделённых запятой атрибутов. Имена атрибутов могут быть неполными, полными и абсолютными, как описано в разделе [Использование пространства имён: основы](#). Аргументы атрибутов необязательны, но если они есть, то их заключают в круглые скобки `()`. Аргументы атрибутов могут быть либо конкретными значениями (литералами), либо константными выражениями. Аргументы можно записывать как позиционным, так и именованным синтаксисом.

Когда Reflection API запрашивает экземпляр класса атрибута, имя атрибута трактуется как имя запрашиваемого класса, а аргументы атрибута передаются в конструктор этого класса. Поэтому для каждого атрибута необходимо создавать класс.

Пример #1 Синтаксис атрибутов

```
<?php
// a.php

namespace MyExample;

use Attribute;

#[Attribute]
class MyAttribute
{
    const VALUE = 'value';

    private $value;

    public function __construct($value = null)
    {
        $this->value = $value;
    }
}

// b.php

namespace Another;

use MyExample\MyAttribute;

#[MyAttribute]
#[\MyExample\MyAttribute]
#[MyAttribute(1234)]
#[MyAttribute(value: 1234)]
#[MyAttribute(MyAttribute::VALUE)]
#[MyAttribute(array("key" => "value"))]
#[MyAttribute(100 + 200)]
class Thing
{
}

#[MyAttribute(1234), MyAttribute(5678)]
class AnotherThing
{
}
```

[+add a note](#)

User Contributed Notes 1 note

[up](#)

[down](#)

1
[yarns dot purport0n at icloud dot com ¶](#)

1 month ago

It wasn't obvious to me for a while but you can subclass attributes

<https://3v4l.org/TrMTe>

<?php

```
#[Attribute(Attribute::TARGET_PROPERTY)]
class PropertyAttributes
{
    public function __construct(
        public readonly ?string $name = null,
        public readonly ?string $label = null,
    ) {}
}

#[Attribute(Attribute::TARGET_PROPERTY)]
class IntegerPropertyAttributes extends PropertyAttributes
{
    public function __construct(
        ?string $name = null,
        ?string $label = null,
        public readonly ?int $default = null,
        public readonly ?int $min = null,
        public readonly ?int $max = null,
        public readonly ?int $step = null,
    ) {
        parent::__construct($name, $label);
    }
}

#[Attribute(Attribute::TARGET_PROPERTY)]
class FloatPropertyAttributes extends PropertyAttributes
{
    public function __construct(
        ?string $name = null,
        ?string $label = null,
        public readonly ?float $default = null,
        public readonly ?float $min = null,
        public readonly ?float $max = null,
    ) {
        parent::__construct($name, $label);
    }
}

class MyClass
{
    #[IntegerPropertyAttributes('prop', 'property: ', 5, 0, 10, 1)]
    public int $prop;
}

$refl = new ReflectionProperty('MyClass', 'prop');
$attributes = $refl->getAttributes();

foreach ($attributes as $attribute) {
    var_dump($attribute->getName());
    var_dump($attribute->getArguments());
    var_dump($attribute->newInstance());
}
```

?>

[+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](#)

