Search

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

Ресурсы »
« Объекты

- Руководство по PHP
- Справочник языка
- Типы

Change language: Russian

# Перечисления

(PHP 8 >= 8.1.0)

## Основы перечислений

Перечисления — это ограничивающий слой над классами и константами классов, предназначенный для предоставления способа определения закрытого набора возможных значений для типа.

```
<?php
enum Suit
{
case Hearts;
case Diamonds;
case Clubs;
case Spades;
}

function do_stuff(Suit $s)
{
// ...
}

do_stuff(Suit::Spades);
?>
```

Полное описание смотрите в главе о перечислениях.

## Приведение типов

Если перечисление (enum) преобразовывается в объект (object), оно не изменяется. Если перечисление (enum) преобразовывается в массив (array), то создаётся массив с одним ключом `name` (для простых перечислений) или массив с двумя ключами `name` и `value` (для типизированных перечислений). Все остальные приведения типов приведут к ошибке.



## User Contributed Notes 1 note


2
***esdras-schonevald*** ¶
**1 year ago**
https://gist.github.com/esdras-schonevald/71a6730e6191c5e9c053e2f65b839eec

```
<?php

declare(strict_types=1);

/**
* This is a sample
* How to use Enum to create a custom exception cases
* PHP 8.1^
*/

enum MyExceptionCase {
case InvalidMethod;
case InvalidProperty;
case Timeout;
}
```

```
class MyException extends Exception {
function __construct(private MyExceptionCase $case){
match($case){
MyExceptionCase::InvalidMethod => parent::__construct("Bad Request - Invalid Method", 400),
MyExceptionCase::InvalidProperty => parent::__construct("Bad Request - Invalid Property", 400),
MyExceptionCase::Timeout => parent::__construct("Bad Request - Timeout", 400)
};
}
}

// Testing my custom exception class
try {
throw new MyException(MyExceptionCase::InvalidMethod);
} catch (MyException $myE) {
echo $myE->getMessage(); // Bad Request - Invalid Method
}
```