



- [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

[Основы перечислений »](#)
[« Перечисления](#)

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

Change language: Russian

Обзор перечислений

(PHP 8 >= 8.1.0)

Перечисления, или Enums, разрешают разработчику определять настраиваемый тип, который ограничен одним значением из дискретного числа возможных. Это может быть особенно полезно при определении модели предметной области, поскольку даёт «сделать неверные состояния недопустимыми».

Перечисления появляются во многих языках программирования с разным набором функций и операций. В PHP перечисления — это вид объектов. Сами перечисления — это классы, а все его возможные случаи — это объекты одного экземпляра этого класса. То есть варианты перечисления — допустимые объекты и их можно использовать везде, где можно использовать объект, включая проверки типов.

Самый популярный пример перечисления — встроенный логический тип, он перечисляем и допускает значения **true** и **false**. Перечисления разрешают разработчикам определять свои произвольно надёжные перечисления.

[+add a note](#)

User Contributed Notes 1 note

[up](#)

[down](#)

2

[Hayley Watson](#) ¶

10 months ago

It's important to notice that the description here doesn't describe Enum values as being constants. Some languages, like C, C++, and C#, think of an enum as a list of named integers, and working with enums in those languages is just working with integers. PHP does not do that.

In the same way that PHP's booleans are their own type and not just constants with integer values 1 and 0, an Enum is its own type. It's not a bunch of integers (or strings) that have been given names, and shouldn't be thought of or used as such.

An Enum only needs to be backed by some primitive value like an integer or string if you need to communicate it outside the program (a message, UI, db storage, wire protocol, etc) where it has to be converted from/to its native PHP value (this is addressed again on the Backed Enumerations page). If you're converting back and forth between Enums and their backing values inside your own program to get anything done then you might be missing the point of Enums.

You might want more structure to your Enum than just a pure set of distinct but otherwise nondescript values (you might at least want them to be ordered). But all that carry-on should be encapsulated within the Enum class itself via additional methods.

[+add a note](#)

- [Перечисления](#)
 - [Обзор перечислений](#)
 - [Основы перечислений](#)
 - [Типизированные перечисления](#)
 - [Методы перечислений](#)
 - [Статические методы перечислений](#)
 - [Константы перечислений](#)
 - [Трейты](#)
 - [Значения перечисления в постоянных выражениях](#)
 - [Отличия от объектов](#)
 - [Список значений](#)
 - [Сериализация](#)
 - [Почему перечисления не расширяемы](#)
 - [Примеры](#)

- [Copyright © 2001-2024 The PHP Group](#)
- [My PHP.net](#)
- [Contact](#)

- [Other PHP.net sites](#)
- [Privacy policy](#)

