

The background is a vibrant, stylized illustration of an underwater scene. It features a blue central area representing water, surrounded by green and red sections. On the left, there is a large, abstract red shape resembling coral. In the upper right, a school of yellow fish swims. Below them, several red circles of varying sizes represent bubbles. At the bottom left, there is a green, leafy plant-like shape. On the bottom right, there are three horizontal red lines. The overall style is flat and modern.

PHP 8 in a Nutshell

Amit D. Merchant

amitmerchant.com

Table of Contents

Introduction	5
PHP 8	6
Union Types	7
The Nullsafe Operator	10
Constructor Property Promotion	12
New String functions	15
Non-capturing Exception Catches	19
The Mixed Type	20
Match Expressions	26
Named Arguments	30
Attributes	33
The ::class keyword	40
Throw as an Expression	42
The get_debug_type() function	44
PHP 8.1	47
Readonly Properties	48
Native Enumerations	52
Fibers or Coroutines	56
Intersection Types	59
First-class callables	61
New in initializers	64
Array unpacking with string keys	67
The array_is_list() function	69
The Never Return Type	72

Final class constants	75
PHP 8.2	77
Readonly classes	78
Null and false as standalone types	81
The true type	83
Using constants in traits	85
Disjunctive Normal Form Types	92
Dynamic Properties Depreciation	94
Fetch properties of enums in const expressions	96
Redacting properties in backtraces	98
New Random Extension	99
The new MySQLi execute_query() method	100
Some minor improvements	101
PHP 8.3	103
The json_validate() function	104
Improved unserialize() error handling	106
New methods in the Randomizer class	108
Fetch class constants dynamically	112
Improved Date/Time Exceptions	114
Typed Constants	117
Readonly amendments	119
The #[\Override] attribute	122
Arbitrary static variable initializers	126
Make unserialize() emit a warning for trailing bytes	128
A new mb_str_pad function	129
Miscellaneous improvements	131
Deprecations in PHP 8.3	133

Typed Constants

The perpetual longing of making the type system of PHP more robust is still going on. And going in that direction, PHP 8.3 has introduced [typed constants](#).

Essentially, up until now, you could not specify the type of constants. But with PHP 8.3, it won't be the case anymore.

So, from PHP 8.3, you would be able to specify a type to class, interface, trait, as well as enum constants.

Here are some examples of how you can use typed constants.

```
enum Car
{
    const string NAME = "Car"; // Car::NAME is a string
}

trait Base
{
    const string NAME = "Base"; // Base::NAME is a string
}

interface Adapter
{
    const string NAME = Car::NAME; // Adapter::NAME is a string as
    well
}

class Audi implements Adapter
{
    use Base;

    const string NAME = Car::NAME; // Audi::NAME must also be a string
}

class Tesla extends Audi
{
    const string NAME = "Model X"; // Tesla::NAME must also be a
    string, but the value can change
}
```

Constant values have to match the type of the class constant. In case it does not, a `TypeError` will be thrown.

Apart from this, like all the other type checks (property types), constant type checks are always performed in the strict mode.

While some may argue that why should constants have types? Well, in my opinion, typed constants are not necessary since constants are immutable by default. But by making them typed, it improves the code readability and makes the code self-documenting.

This is a sample from "PHP 8 in a Nutshell" by Amit D. Merchant.

For more information, [Click here](#).