PHP 8.4 Обзор

Кочетов
Даниил

О чем поговорим?

php 8.4

- Новый функционал
- Обратная совместимость
- Переход
- Производительность

php 8.4

Новый функционал

Функциональный стиль вызова `new`

php 8.4

```php
// Старый стиль (с дополнительными скобками):
$area = (new Rectangle(5, 10))->calculateArea();

// Новый функциональный стиль в PHP 8.4:
$area = new Rectangle(5, 10)->calculateArea();
```

Хуки свойств

php 8.4

```php
class Person
{
    public string $fullName {
        get => $this->firstName . ' ' . $this->lastName;
    }

    public string $firstName {
        set => mb_ucfirst(strtolower($value));
    }

    public string $lastName {
        set {
            if (strlen($value) < 2) {
                throw new \InvalidArgumentException('Слишком короткая фамилия');
            }

            $this->lastName = $value;
        }
    }
}
```

```php
$p = new Person();
$p->firstName = 'пётр';
echo $p->firstName; // Выведет "Пётр"

$p->lastName = 'Петров';
echo $p->fullName; // Выведет "Пётр Петров"
```

Ассиметричный модификатор доступа

php 8.4

```php
class Software
{
    public private(set) string $version = '8.4';
}

$software = new Software();

echo $software->version;  // Выведет "8.4"

// $software->version = '8.5';  // Ошибка: нельзя изменять значение
```

Комбинирование хуков и асимметричного модификатора доступа

php 8.4

```
class Person
{
    private(set) string $name
    {
        get => ucfirst(strtolower(trim($this->name)));
    }
}
```

Ассиметричный модификатор доступа

PHP 8.4

- **public private(set)** // Можно
- **public protected(set)** // Можно
- **protected private(set)** // Можно
- **protected public(set)** // Нельзя
- **private public(set)** // Нельзя
- **private protected(set)** // Нельзя

php 8.4

Функции для работы с массивами

array_find()

php 8.4

Функция `array_find()` находит первый элемент массива, который удовлетворяет условию.

```php
$numbers = [1, 2, 3, 4, 5];
$found = array_find($numbers, fn($n) => $n > 3);
echo $found; // 4
```

array_find_key()

php 8.4

Функция `array_find_key()` находит первый ключ массива, который удовлетворяет условию.

```php
$numbers = ['a' => 1, 'b' => 2, 'c' => 3];
$key = array_find_key($numbers, fn($n) => $n === 2);
echo $key; // 'b'
```

array_find(), return null

php 8.4

`array_find()`, `array_find_key()` .Что если не нашли элемент

```php
$numbers = [1, 2, 3, 4, 5];
$found = array_find($numbers, fn($n) => $n > 5);
echo $found; // null
```

array_any()

php 8.4

Функция `array_any()` проверяет, существует ли хотя бы один элемент, который удовлетворяет условию.

```php
$numbers = [1, 2, 3, 4, 5];
$exists = array_any($numbers, fn($n) => $n > 3);
echo $exists ? 'True' : 'False'; // True
```

array_all()

php 8.4

Функция `array_all()` проверяет, удовлетворяют ли все элементы массива условию.

```php
$numbers = [1, 2, 3, 4, 5];
$all = array_all($numbers, fn($n) => $n > 0);
echo $all ? 'True' : 'False'; // True
```

Атрибут #[Deprecated]

php 8.4

```php
#[\Deprecated(message: "use safe_replacement() instead", since: "1.5")]
function unsafe_function()
{
   echo "This is unsafe", PHP_EOL;
}

unsafe_function();
```

Результат:

```
Deprecated: Function unsafe_function() is deprecated since 1.5,
use safe_replacement() instead in example.php on line 9
This is unsafe
```

Атрибут `#[Deprecated]` использование

php 8.4

```php
class MyClass {
    public string $version = '8.4';

    #[Deprecated("Use NEW_CONST instead")]
    public const OLD_CONST = 1;

    #[Deprecated("Use newMethod() instead")]
    public function oldMethod() {
        echo "This is an old method.\n";
    }
}
```

ext-Dom, Как было

php 8.4

```php
$dom = new DOMDocument();
$dom->loadHTML(
    <<<'HTML'
        <main>
            <article>PHP 8.4 is a feature-rich release!</article>
            <article class="featured">PHP 8.4 adds new DOM classes that are spec-compliant, keeping the old ones for.</article>
        </main>
    HTML,
    LIBXML_NOERROR,
);

$xpath = new DOMXPath($dom);
$node = $xpath->query(".//main/article[not(following-sibling::*)]")[0];
$classes = explode(" ", $node->className); // Simplified
var_dump(in_array("featured", $classes)); // bool(true)
```

ext-Dom, новые возможности

php 8.4

```
$dom = Dom\HTMLDocument::createFromString(
  <<<'HTML'
     <main>
         <article>PHP 8.4 is a feature-rich release!</article>
         <article class="featured">PHP 8.4 adds new DOM classes that are spec-compliant, keeping the old ones for.</article>
     </main>
     HTML,
LIBXML_NOERROR,
);

$node = $dom->querySelector('main > article:last-child');
var_dump($node->classList->contains("featured")); // bool(true)
```

Объектно-ориентированный API для BCMath

php 8.4

```php
$num1 = '0.12345';
$num2 = 2;
$result = bcadd($num1, $num2, 5);

echo $result; // '2.12345'
var_dump(bccomp($num1, $num2) > 0); // false
```

```php
use BcMath\Number;

$num1 = new Number('0.12345');
$num2 = new Number('2');
$result = $num1 + $num2;

echo $result; // '2.12345'
var_dump($num1 > $num2); // false
```

Ленивая инициализация объектов

php 8.4

```php
class Example
{
    public function __construct(private int $data) {}

    // ...
}

$initializer = static function (Example $ghost): void {
    $data = ...; // Получение данных или зависимостей

    $ghost->__construct($data); // Инициализация объекта
};

$reflector = new ReflectionClass(Example::class);
$object = $reflector->newLazyGhost($initializer);
```

request_parse_body

php 8.4

```php
[$_POST, $_FILES] = request_parse_body([
    'post_max_size' => '10M',
    'upload_max_filesize' => '10M',
]);
```

RequestParseBodyException

ValueError

grapheme_str_split

php 8.4

```php
$string = "👨‍👩‍👧‍👦";

$array = grapheme_str_split($string);
print_r($array);

$array2 = grapheme_str_split($string, 2);
print_r($array2);
```

```
Array
(
    [0] => 👨
    [1] => 👩
    [2] => 👧
    [3] => 👦
)
Array
(
    [0] => 👨👩
    [1] => 👧👦
)
```

Изменения, которые ломают обратную

СОВМЕСТИМОСТЬ

`exit` и `die`: новое поведение

php 8.4

```php
declare(strict_types=1);
exit(3.14);
```

PHP Fatal error: Uncaught TypeError: exit(): Argument #1 ($status) must be of type string|int, float given

Типы параметров, неявно допускающие значение null объявлены устаревшими

php 8.4

```php
function greet(string $name = null): void
{
    echo "Hello, " . ($name ?? "Guest") . "!\n";
}

greet(); // Deprecated: Implicitly allowing null for parameter $name
```

Типы параметров, неявно допускающие значение null объявлены устаревшими

php 8.4

```php
function greet(?string $name = null): void
{
    echo "Hello, " . ($name ?? "Guest") . "!\n";
}

greet(); // Hello, Guest!
```

Обновление round

php 8.4

```
round(1.1, precision: 0, mode: INCORRECT_MODE); // ValueError
round(1.1, precision: 0, mode: PHP_ROUND_CEILING); // 2
round(1.1, precision: 1, mode: PHP_ROUND_FLOOR); // -1
round(1.9, precision: 0, mode: PHP_ROUND_TOWARD_ZERO); // 1
round(1.9,  precision: 0, mode: PHP_ROUND_AWAY_FROM_ZERO); // 2
```

Обновления `opcache` и `JIT`

php 8.4

По умолчанию до 8.4:

`opcache.jit_buffer_size=0`

`opcache.jit=tracing`

По умолчанию в 8.4:

`opcache.jit_buffer_size=64M`

`opcache.jit=disable`

Сравнение производительности

php 8.4

PHP Performance Test

PHP Version: 8.3.14

JIT Enabled: No

Math Benchmark: 573.10 ms (average over 10 runs)
Array Benchmark: 101.48 ms (average over 10 runs)
String Benchmark: 452.48 ms (average over 10 runs)

PHP Performance Test

PHP Version: 8.4.1

JIT Enabled: No

Math Benchmark: 522.80 ms (average over 10 runs)
Array Benchmark: 101.61 ms (average over 10 runs)
String Benchmark: 453.33 ms (average over 10 runs)

PHP Performance Test

PHP Version: 8.3.14

JIT Enabled: Yes

Math Benchmark: 390.02 ms (average over 10 runs)
Array Benchmark: 88.64 ms (average over 10 runs)
String Benchmark: 439.26 ms (average over 10 runs)

PHP Performance Test

PHP Version: 8.4.1

JIT Enabled: Yes

Math Benchmark: 367.84 ms (average over 10 runs)
Array Benchmark: 91.38 ms (average over 10 runs)
String Benchmark: 421.07 ms (average over 10 runs)

Переход на 8.4, Ректор

php 8.4

```php
$rectorConfig->rules([
    ExplicitNullableParamTypeRector::class,
    RoundingModeEnumRector::class
]);
```

Выводы

php 8.4

Branch	Initial Release		Active Support Until		Security Support Until	
8.1	25 Nov 2021	*3 years, 1 month ago*	25 Nov 2023	*1 year, 1 month ago*	31 Dec 2025	*in 11 months*
8.2	8 Dec 2022	*2 years, 1 month ago*	31 Dec 2024	*20 days ago*	31 Dec 2026	*in 1 year, 11 months*
8.3	23 Nov 2023	*1 year, 1 month ago*	31 Dec 2025	*in 11 months*	31 Dec 2027	*in 2 years, 11 months*
8.4	21 Nov 2024	*1 month ago*	31 Dec 2026	*in 1 year, 11 months*	31 Dec 2028	*in 3 years, 11 months*

php 8.4

Вопросы