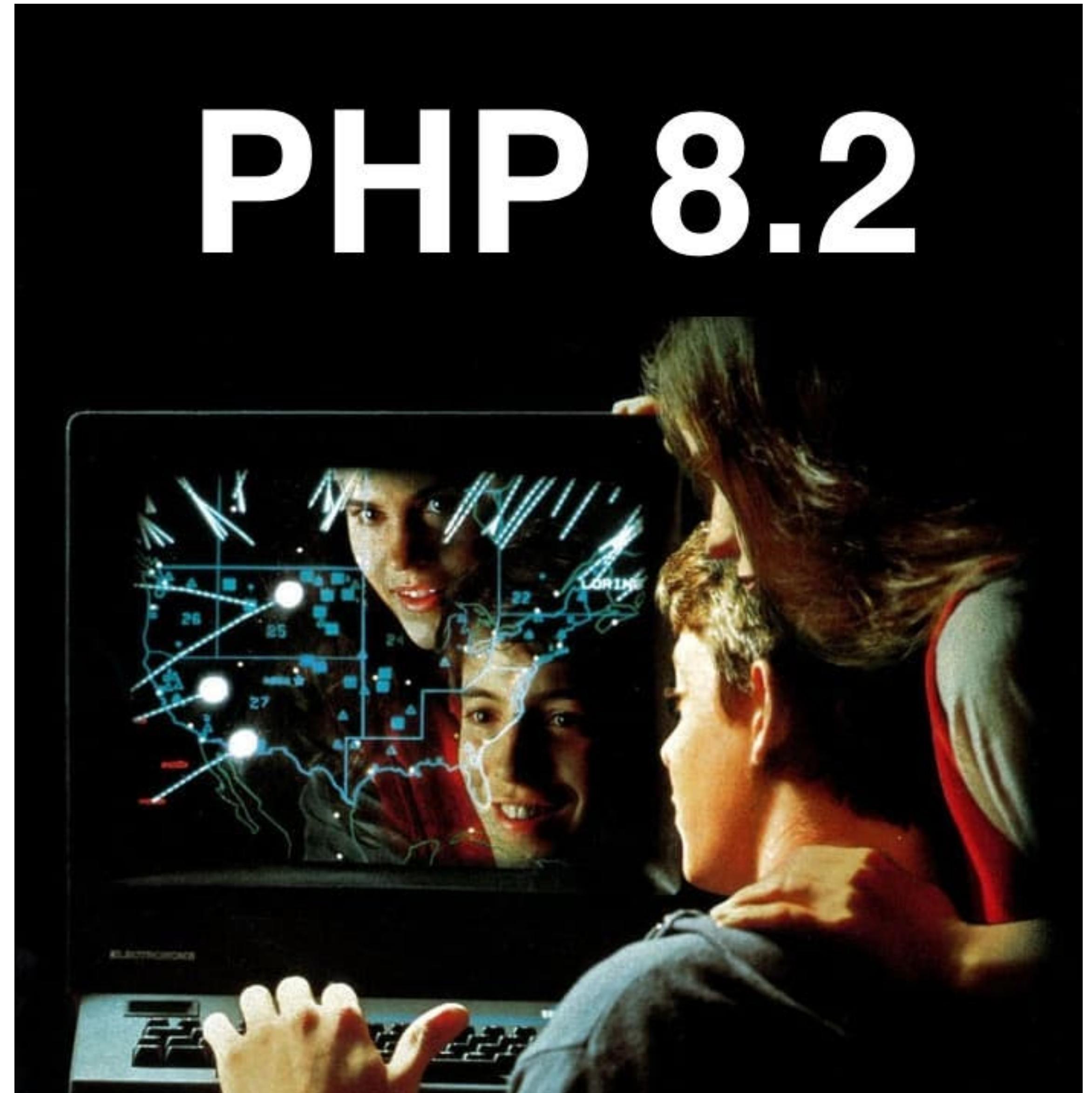


PHP 8.2

dnf module install php:remi-8.2 || apt install php8.2

No thanksgiving release

- November 24th: PHP 8.2 RC7
- December 8th: PHP 8.2.0



Currently supported versions

PHP 7 EOL

Branch	Initial Release		Active Support Until		Security Support Until	
7.4	28 Nov 2019	<i>2 years, 11 months ago</i>	28 Nov 2021	<i>11 months ago</i>	28 Nov 2022	<i>in 5 days</i>
8.0	26 Nov 2020	<i>1 year, 11 months ago</i>	26 Nov 2022	<i>in 3 days</i>	26 Nov 2023	<i>in 1 year</i>
8.1	25 Nov 2021	<i>11 months ago</i>	25 Nov 2023	<i>in 1 year</i>	25 Nov 2024	<i>in 2 years</i>

- Constants are supported in traits
- New curl_upkeep function
- New ini_parse_quantity function
- New memory_reset_peak_usage function
- No-capture modifier (/n) support in preg_* functions
- Curl:
CURLINFO_EFFECTIVE_METHOD support in curl_getinfo function
- AllowDynamicProperties Attribute

PHP 8.2

Deprecations

- Dynamic Properties are deprecated
- utf8_encode and utf8_decode functions deprecated
- Partially-supported callable are deprecated
- Mbstring: Base64, Uuencode, QPrint, and HTML Entity encodings are deprecated
- \${var} string interpolation deprecated

Voting (Request for Comments) - RFC

<https://wiki.php.net/rfc>

Fetch properties of enums in const expressions (View on PHP.net)

PHP 8.2

Add support for fetching properties of enums in constant expressions?

69 %

31 %

■ Yes: 24

■ No: 11

Total number of votes cast: 35

 Readonly classes (View on PHP.net)

PHP 8.2

Add readonly classes as proposed?

80 %

20 %

■ Yes: 28

■ No: 7

Total number of votes cast: 35

Allow null and false as stand-alone types (View on PHP.net)

PHP 8.2

Accept Allow null and false as stand-alone types RFC?

100 %

■ Yes: 38

■ No: 0

Total number of votes cast: 38

Add true type (View on PHP.net)

PHP 8.2

Accept Add true type RFC?

100 %

■ Yes: 33

■ No: 0

Total number of votes cast: 33

Deprecate \${} string interpolation (View on PHP.net)

PHP 8.2

Deprecate \${} string interpolation in PHP 8.2 and remove them in PHP 9.0?

97 %

3 %

■ Yes: 31

■ No: 1

Total number of votes cast: 32

Locale-independent case conversion (View on PHP.net)

PHP 8.2

Use locale-independent case conversion for string functions as proposed?

97 %

3 %

■ Yes: 29

■ No: 1

Total number of votes cast: 30

Deprecate dynamic properties

Will throw an `ErrorException` in PHP 9.0

```
class BergenPhp {}  
$nix = new BergenPhp();  
$nix->beer = 'Hansa';
```

Property declared dynamically, this is deprecated starting from PHP 8.2

Add '#[AllowDynamicProperties]' to 'BergenPhp' ↗↔

More actions...

Readonly classes

syntactic sugar

```
readonly class Post82
{
    public function __construct(
        public string $title,
        public Author $author,
        public string $body,
        public DateTime $publishedAt,
    ) {}
}
```

#[]

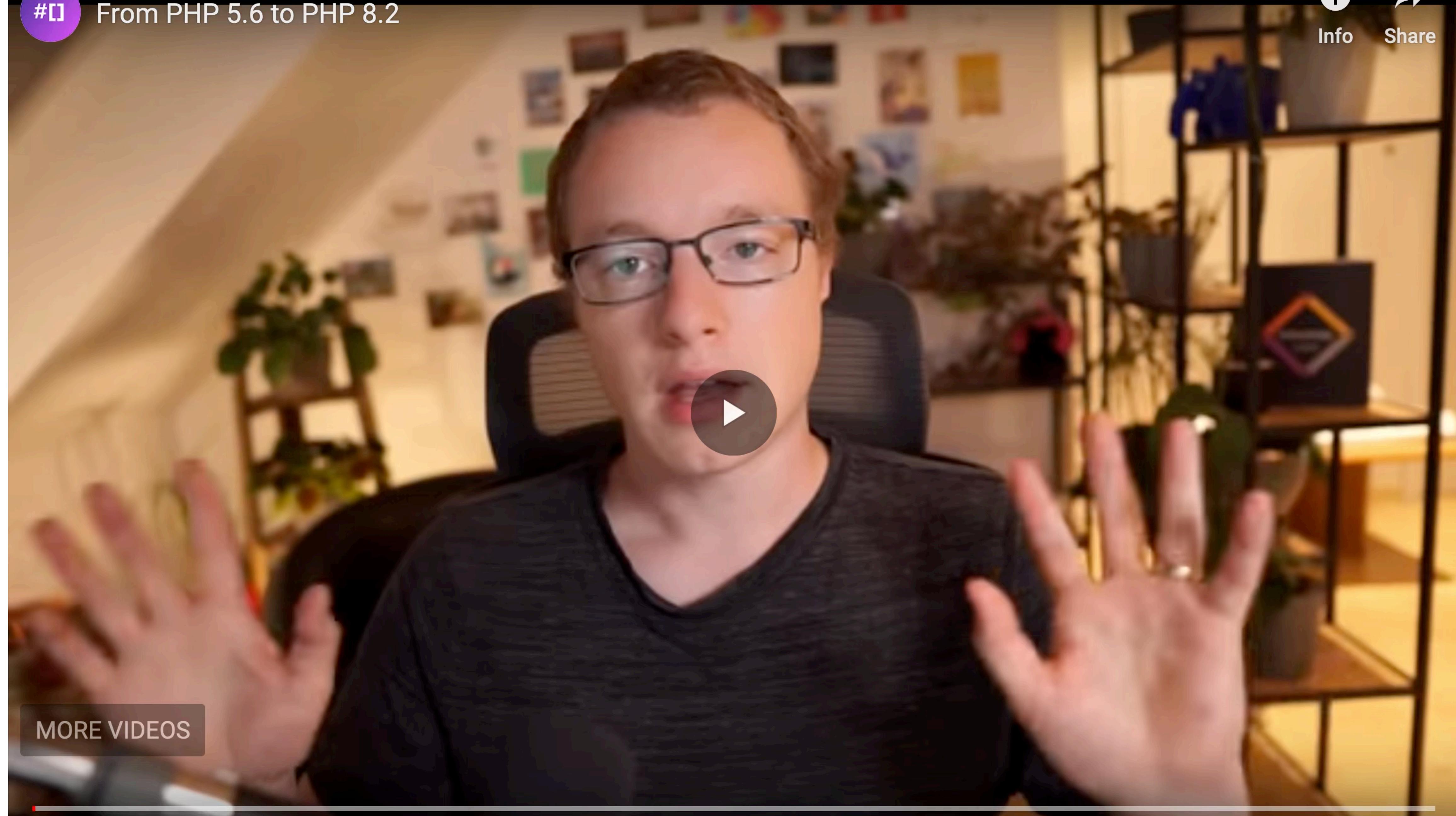
From PHP 5.6 to PHP 8.2



Info



Share



MORE VIDEOS



0:00 / 1:55



YouTube



<https://stitcher.io>

Security improvements

- New PHP extension named random.
- New \Random namespace
- Existing functions still available in global namespace.



- random_bytes function
- random_int function
- rand function
- getrandmax function
- srand function
- lcg_value function
- mt_rand function
- mt_getrandmax function
- mt_srand function
- MT_RAND_PHP constant
- MT_RAND_MT19937 constant

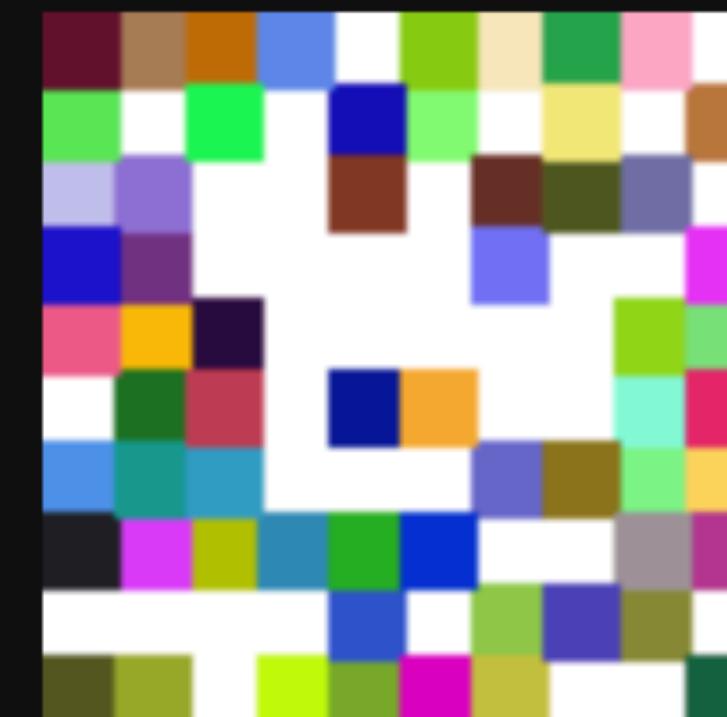
Visualizing

As long as the seed value is the same, the generated random number series remain the same.

As long as the seed value is the same, the generated random number series remain the same.

Changing the seed value even by one byte yields completely different series of random numbers.

Pcg0neseq128Xs1Rr64



Xoshiro256StarStar



```
$phpMeetup = new Random\Randomizer();
echo $phpMeetup->getInt(1, 15) . PHP_EOL;
echo $phpMeetup->shuffleBytes( bytes: 'Code in Bergen');
```

#[\SensitiveParameter] Attribute

Sensitive Parameter value redaction support

```
function passwordHash( string $password) {  
    password_hash($password, algo: 'Slitan');  
}  
  
passwordHash( password: 'hunter2');
```

```
PHP Fatal error:  
Uncaught ValueError: password_hash():  
Argument #2 ($algo) must be a valid password hashing algorithm in sensitive.php:11  
Stack trace:  
#0 sensitive.php(11): password_hash(Object(SensitiveParameterValue), 'Slitan')  
#1 sensitive.php(14): passwordHash('hunter2')  
#2 {main}  
... thrown in sensitive.php on line 11  
#1 sensitive.php(14): passwordHash('hunter2')
```



HAS ACCESS TO SENSITIVE DATA

HIDES PASSWORDS UNDER KEYBOARD

Traits

Constants are supported in traits

```
// PHP 8.1

trait VersionDependent {

    protected function ensureVersion(): void {
        if (self::CURRENT_VERSION < self::MIN_VERSION) {
            throw new Exception(message: 'Current version is too old');
        }
    }
}
```

```
class foo {
    use VersionDependent;
    const CURRENT_VERSION = 2.4;
}
```

```
trait VersionDependentPhp82 {
    protected const CURRENT_VERSION = 2.6;
    protected const MIN_VERSION = 2.5;

    protected function ensureVersion(): void {
        if (self::CURRENT_VERSION < self::MIN_VERSION) {
            throw new Exception(message: 'Current version is too old');
        }
    }
}
```

Type system history

PHP 7.0 added support for scalar types such as `string`, `int`, and `bool`

PHP 8.0 - introduction of Union Types, `false`

PHP 8.2 - introduction of `true`, as well as `null` and `false`

```
class · Foo
{
    · private · true · $processed;

    · public · function · process(string|int|true · $value): · true · {
        ·     ·     return · true;
    }
}
```



PHP 8.3

才

**Improve unserialize() error handling
json_validate function**

Nov 23 2023

<https://wiki.php.net/todo/php83>