



Keeping Up with PHP

PHP Tek 2025

Andrew Woods

- ❖ Senior Software Engineer
- ❖ Live in NYC
- ❖ <https://joind.in/talk/9e966>



PHP is dead

Programmers of other languages since 2000

The State of PHP

2025 May 21 Wed

PHP 8.4 is the latest version.

PHP 8.3 on 2026 Jan 01 enter security fixes phase.

PHP 8.2 is already in security fixes phase.

PHP 8.1 will be on 2026 Jan 01.

PHP 8.0 went EOL 2023 Nov 26.



Are you
running an
EOL version
of PHP?



You're
running the
latest PHP?

Nice job,
buddy!

**PHP 8.1 first version with
2 years of security fixes**

The following code has been modified from its original version. It has been formatted to fit these slides.

me, just now

PHP 8.0-8.2

Early 8.x Foundations

PHP 8.0 New Features

- ❖ Named Arguments
- ❖ Constructor Property Promotions
- ❖ Attributes
- ❖ Union Types
- ❖ Nullsafe Operator

Named Arguments


```
<?php
```

```
$eric = new User('shocm', 'Eric Van Johnson', 'San Diego, CA');
```

```
$john = new User(  
    city: 'San Diego, CA',  
    realname: 'John Congdon',  
    username: 'j.congdon',  
);
```

Constructor Property Promotion


```
<?php
class User {
    protected string $username;

    public function __construct(
        string $username,
    ) {
        $this->username = $username;
    }
}
```

```
<?php
class User{
    public function __construct(
        protected string $username,
        protected string $realname,
        protected string $city,
    ) { // Empty Body }
}
```

Attributes

[\AllowDynamicProperties]

[\ReturnTypeWillChange]

[\SensitiveParameter]

#[SensitiveParameter]

Union Types

Prior to 8.0

```
/**
 * @param string | int $id
 *   A unique integer ID or login name to lookup a record.
 */
function findUserId( $id ){
    // Query the database for a user
}
```


Union Types with 8.0

```
/**  
 * @param string | int $id  
 *   A unique integer ID or login name to lookup a record.  
 */  
function findUserId(string | int $id){  
    // Query the database for a user  
}
```

PHP 8.1 New Features

- ❖ Enumerations
- ❖ Readonly Properties
- ❖ Pure Intersection Types
- ❖ Never return type
- ❖ Final class constants
- ❖ Fibers

Enumerations

```
enum WordPressRoles
{
    case Subscriber;
    case Contributor;
    case Author;
    case Editor;
    case Administrator;
    case SuperAdmin;
}
```



```
enum WordPressRoles
```

```
{
```

```
    case Subscriber = 'subscriber';
```

```
    case Contributor = 'contributor';
```

```
    case Author = 'author';
```

```
    case Editor = 'editor';
```

```
    case Administrator = 'administrator';
```

```
    case SuperAdmin = 'superadmin';
```

```
}
```

Pure Intersection Types

```
function displayRecords(Countable&Iterator $allUsers)
{
    // your code here
}
```

PHP 8.2 New Features

- ❖ Readonly Classes
- ❖ Disjunctive Normal Form Types
- ❖ Allow null, false, and true as standalone types
- ❖ Constants in Traits
- ❖ The Sensitive Parameter Attribute

Disjunctive Normal Form Types

**Allow null, false, and true
as standalone types**

PHP 8.3

Active until 31 Dec 2025

PHP 8.3 New Features

- ❖ Typed Class Constants
- ❖ Dynamic Class Constant Fetch
- ❖ New #[Override] attribute
- ❖ Deep Cloning of Readonly properties
- ❖ New json_validate() function
- ❖ Command line linter supports multiple files

Typed Class Constants

```
class PhpTek
{
    public const EVENT_TYPE = 'Conference';
    public const YEAR = 2025;
    public const LOCATION_CITY = 'Chicago, IL';
}
```

```
class PhpTek
{
    public const string EVENT_TYPE = 'Conference';
    public const int YEAR = 2025;
    public const string LOCATION_CITY = 'Chicago, IL';
}
```


Dynamic Class Constant Fetch

C :: {\$name}

```
class PhpTek
{
    public const string EVENT_TYPE = 'Conference';
    public const int YEAR = 2025;
    public const string LOCATION_CITY = 'Chicago, IL';
}

echo PhpTek::LOCATION_CITY . "\n";
```

```
class PhpTek
{
    public const string EVENT_TYPE = 'Conference';
    public const int YEAR = 2025;
    public const string LOCATION_CITY = 'Chicago, IL';
}

$field = 'LOCATION_CITY';
echo PhpTek::$field . "\n";
```

New #[Override] attribute

New json_validate() function

Command line linter
supports multiple files

PHP 8.4

Latest Version

PHP 8.4 New Features

- ❖ Property hooks
- ❖ Asymmetric Visibility
- ❖ New ext-dom features and HTML5 support
- ❖ Lazy Objects
- ❖ New `array_*()` functions
- ❖ Chaining new expressions without parentheses

New Classes, Enums, and Interfaces

- ❖ #[Deprecated] Attribute
- ❖ RoundingMode Enum
- ❖ PDO driver specific subclasses (PDO_MYSQL, PDO_PGSQL, PDO_SQLite)
- ❖ BcMath\Number class

PHP 8.4 Deprecations

- ❖ Implicitly nullable parameter
- ❖ Raising 0 to a negative number
- ❖ Using an underscore as a class name
- ❖ `E_STRICT` constant is deprecated
- ❖ Using `trigger_error()` with `E_USER_ERROR`

Property hooks

```
class GeoPoint {  
    public string $latitude;  
    public string $longitude;  
}
```

```
class GeoPoint {  
    public string $latitude;  
    public string $longitude;  
}
```

```
$esb = new GeoPoint();  
$esb->latitude = 40.748440;  
$esb->longitude = -73.985664;
```

```
var_dump($esb) . "\n";
```



```
class GeoPoint {  
    public string $latitude;  
    public string $longitude;  
    public string $coordinates {  
        get => $this->latitude . ", " . $this->longitude;  
    }  
}  
  
$esb = new GeoPoint();  
$esb->latitude = 40.748440;  
$esb->longitude = -73.985664;  
  
echo "Empire State Building: {$esb->coordinates} \n";
```



```
class GeoPoint {  
    public function __construct(  
        public string $latitude,  
        public string $longitude,  
    ) { /* Empty body */ }  
    public string $coordinates {  
        get => $this->latitude . ", " . $this->longitude;  
    }  
}  
  
$esb = new GeoPoint(40.748440, -73.985664);  
  
echo "Empire State Building: {$esb->coordinates} \n";
```

```
class GeoPoint {  
    public function __construct(  
        public string $latitude,  
        public string $longitude,  
    ) { /* Empty body */ }  
    public string $coordinates {  
        get => $this->latitude . ", " . $this->longitude;  
    }  
}  
  
$esb = new GeoPoint(  
    'longitude': -73.985664,  
    'latitude': 40.748440,  
);  
  
echo "Empire State Building: {$esb->coordinates} \n";
```

Asymmetric Visibility

```
class Application {  
    public protected(set) string $name;  
    public function __construct(string $name) {  
        $this->name = $name;  
    }  
}
```

```
$app = new Application('Drupal');
```

```
$app->name = "WordPress"; // cause a Fatal error
```

```
echo "Application: {$app->name}\n";
```

#[Deprecated] Attribute

```
#[\Deprecated]
```

```
function getUserByEmail($email) {
```

```
    // well written code here.
```

```
}
```

```
function getUser() {
```

```
    return getUserByEmail('awoods@example.com');
```

```
}
```



```
#[\Deprecated(  
    message: "use getUserByUsername() instead",  
    since: "1.5"  
)]  
function getUserByEmail($email) {  
    // well written code here.  
}
```


New ext-dom features and HTML5 support

```
$dom = Dom\HTMLDocument::createFromstring(
    <<<'HTML'
    <main>
    <article>
    <h1>PHP Tek 2025</h1>
    <p>This is my article.
        There are many like it but this one is mine.</p>
    </article>
    </main>
    HTML,
    LIBXML_NOERROR,
);
```

```
$dom = Dom\HTMLDocument::createFromstring($content);  
/* Your amazing code here */  
echo $dom->saveHTML() . "\n";
```

New array_*() functions

`array_find()`

Returns first element to satisfy a callback

`array_find_key()`

similar but returns the key

`array_any()`

Checks if at least one element satisfies a callback function

`array_all()`

Checks if all elements satisfy a callback function

PDO driver specific subclasses

Pdo\Dblib

,Pdo\Firebird,

Pdo\MySql,

Pdo\Odbc,

Pdo\Pgsql,^a

nd **Pdo\Sqlite**

PD0 :: connect()

**Chaining new expressions
without parentheses**

```
class GeoPoint
{
    public function __construct(
        public string $latitude,
        public string $longitude,
    ) { /* Empty body */
    }

    public string $coordinates {
        get => $this->latitude . ", " . $this->longitude;
    }
}
```

```
echo (new GeoPoint(40.748440, -73.985664)) ->coordinates;
```

```
echo new GeoPoint(40.748440, -73.985664) ->coordinates;
```

```
define('TEK_TIME', 'Y M d D H:i e');
```

```
echo new DateTime(  
    timezone: new DateTimeZone('America/Chicago')  
)->format(TEK_TIME);
```

```
// 2025 May 18 Sun 09:46 America/Chicago
```

Wrap Up

<https://www.php.net/manual/en/appendices.php>

Thank You

❖ **Andrew Woods**

❖ @awoodsnet@phpc.social

❖ <https://andrewwoods.net>

❖ <https://joind.in/talk/9e966>



► WiFi Sponsored By:  **typesense**
<https://typesense.org>



► Thanks to all of our Sponsors



<https://osmihelp.org>

