

George Buckingham @ Tech Tonic February 2025

Is PHP dead?

Let's answer the question that won't die.

About Me

George Buckingham

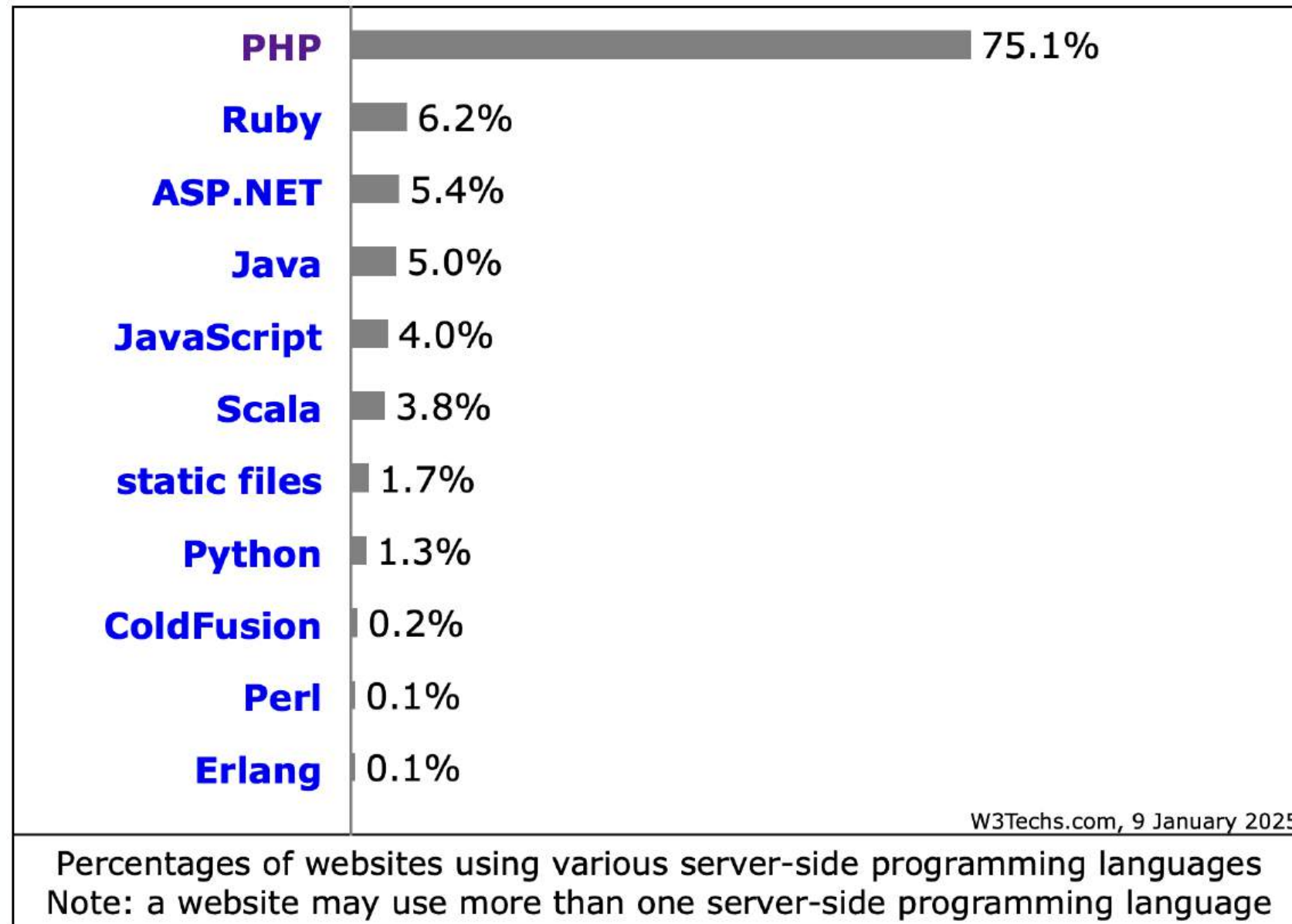
- Head of Development @ Yammayap
- Developing with PHP professionally for 15 years, dabbling for 20 years
- Focused on developing Laravel applications for past 10 years
- Outside of work you'll find me on a golf course, or travelling





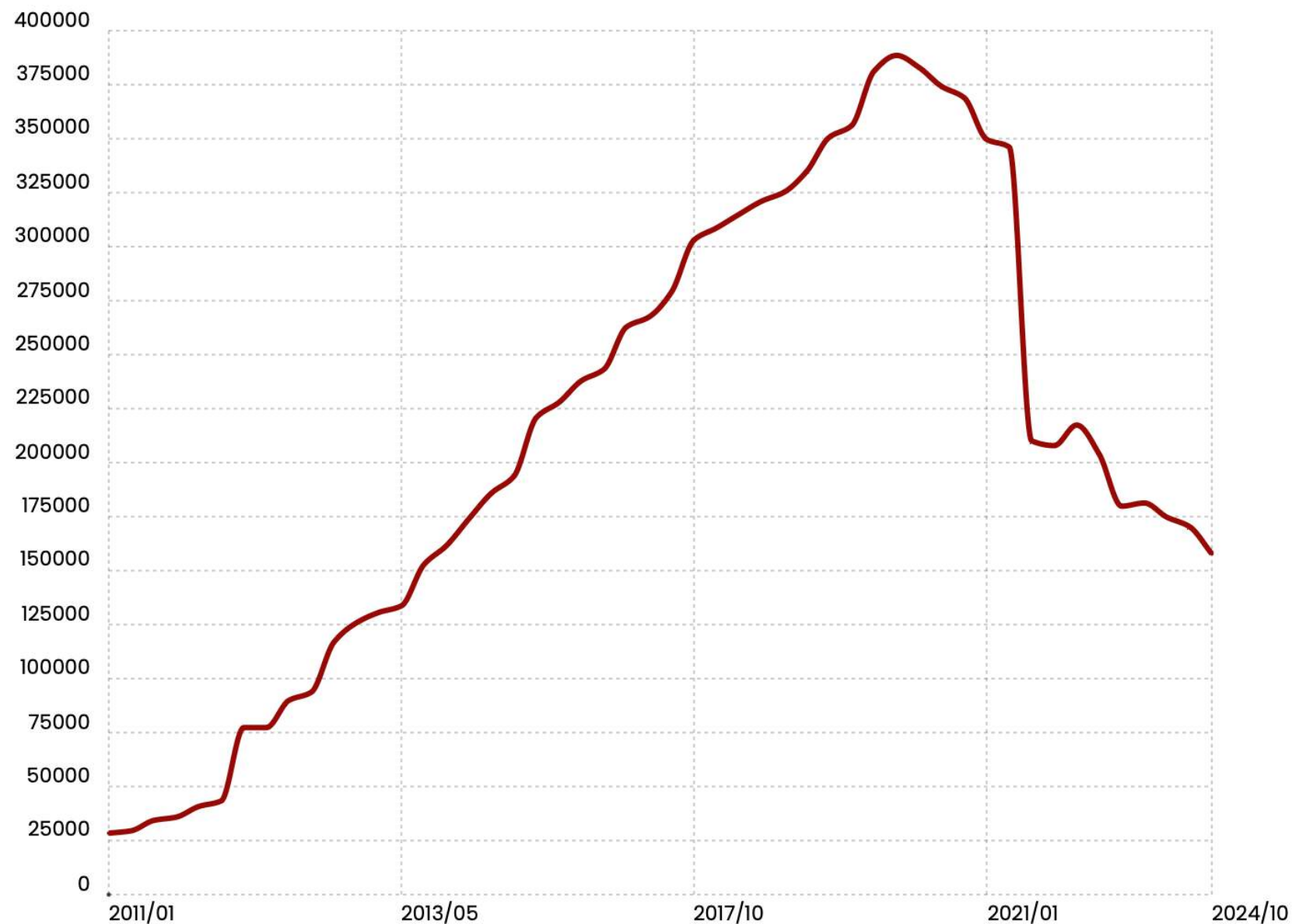
Is PHP dead?

W3 Techs does't think so.



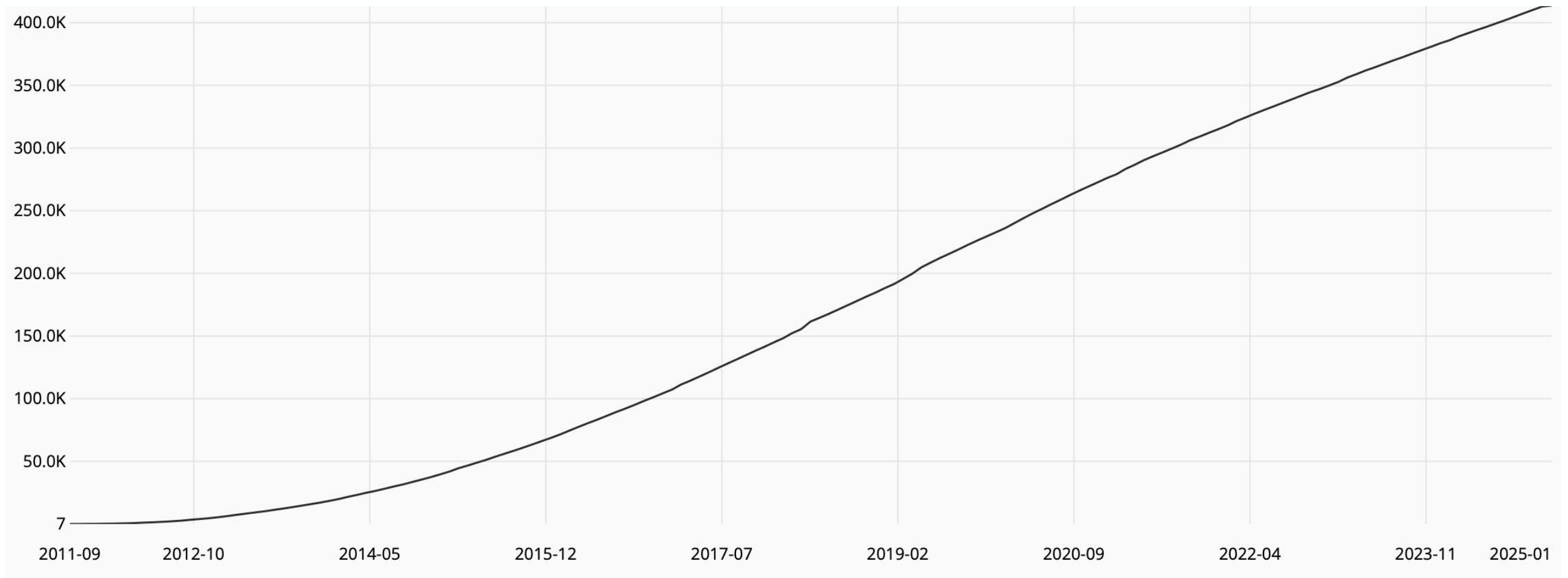
Source: https://w3techs.com/technologies/overview/programming_language

BuiltWith.com gives a different outlook.



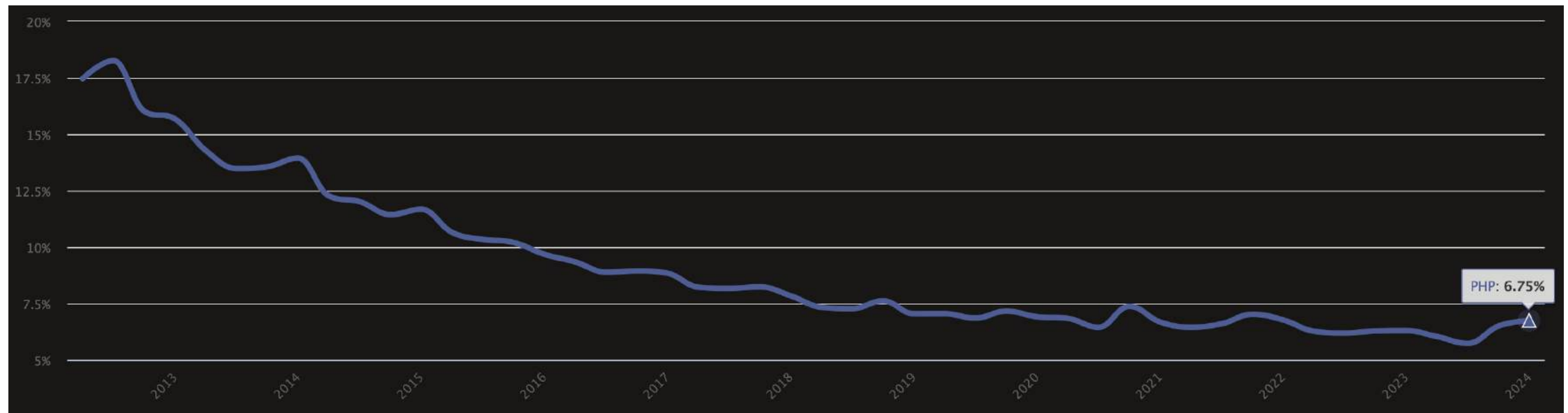
Source: <https://trends.builtwith.com/framework/PHP>

Packagist suggests the ecosystem is healthy.



Source: <https://packagist.org/statistics>

GitHub has a different take on it's popularity.



Source: https://madnight.github.io/githut/#/pull_requests/2024/1

**OH, PEOPLE CAN COME UP WITH STATISTICS
TO PROVE ANYTHING, KENT.**

SMART LITTLE



FORTY PERCENT OF ALL PEOPLE KNOW THAT!

“Virtually every feature in PHP is broken somehow. The language, the framework, the ecosystem, are all just bad. And I can’t even point out any single damning thing, because the damage is so systemic.”

<https://eev.ee/blog/2012/04/09/php-a-fractal-of-bad-design/>

8:43

Google

PHP Tutorials

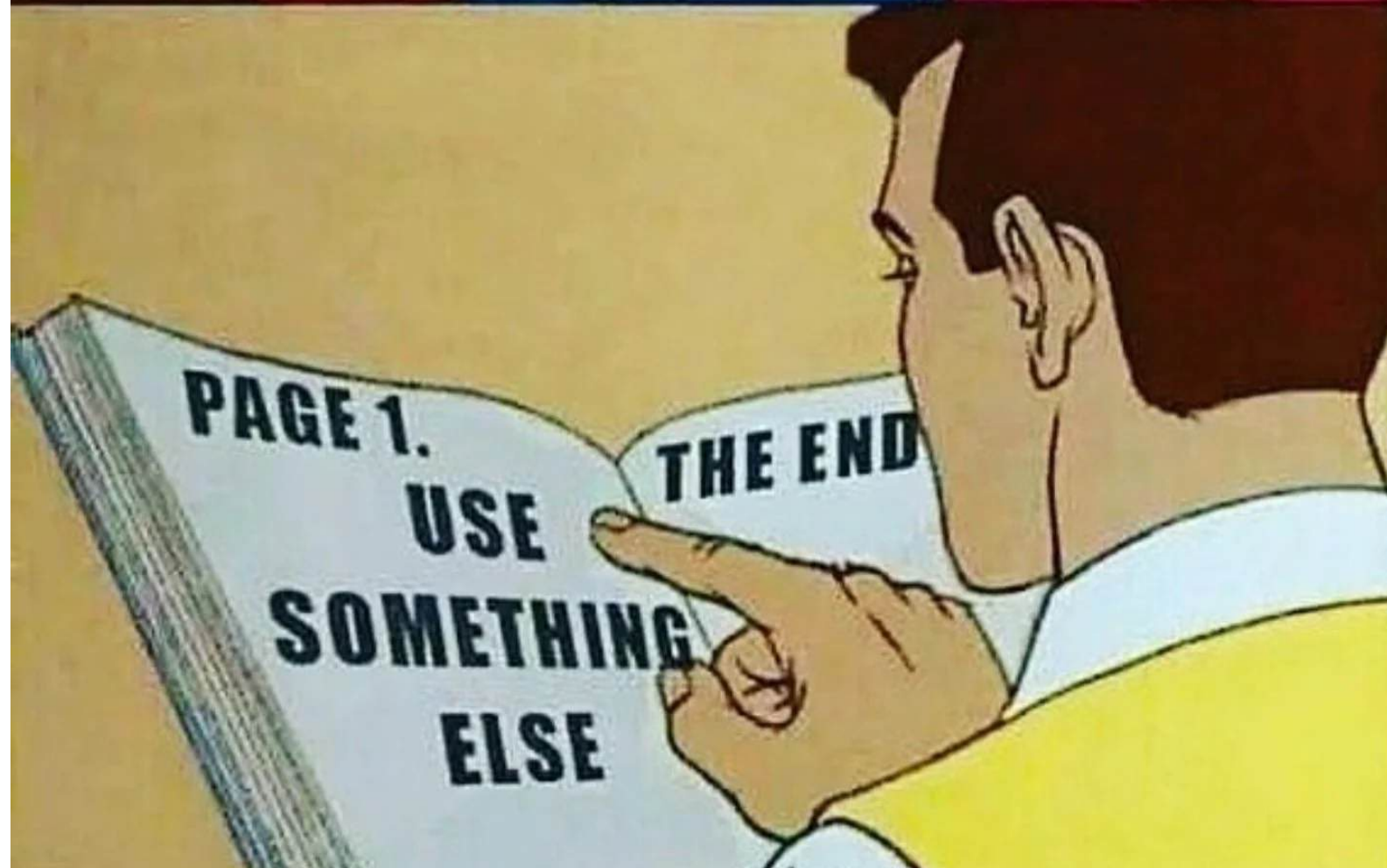
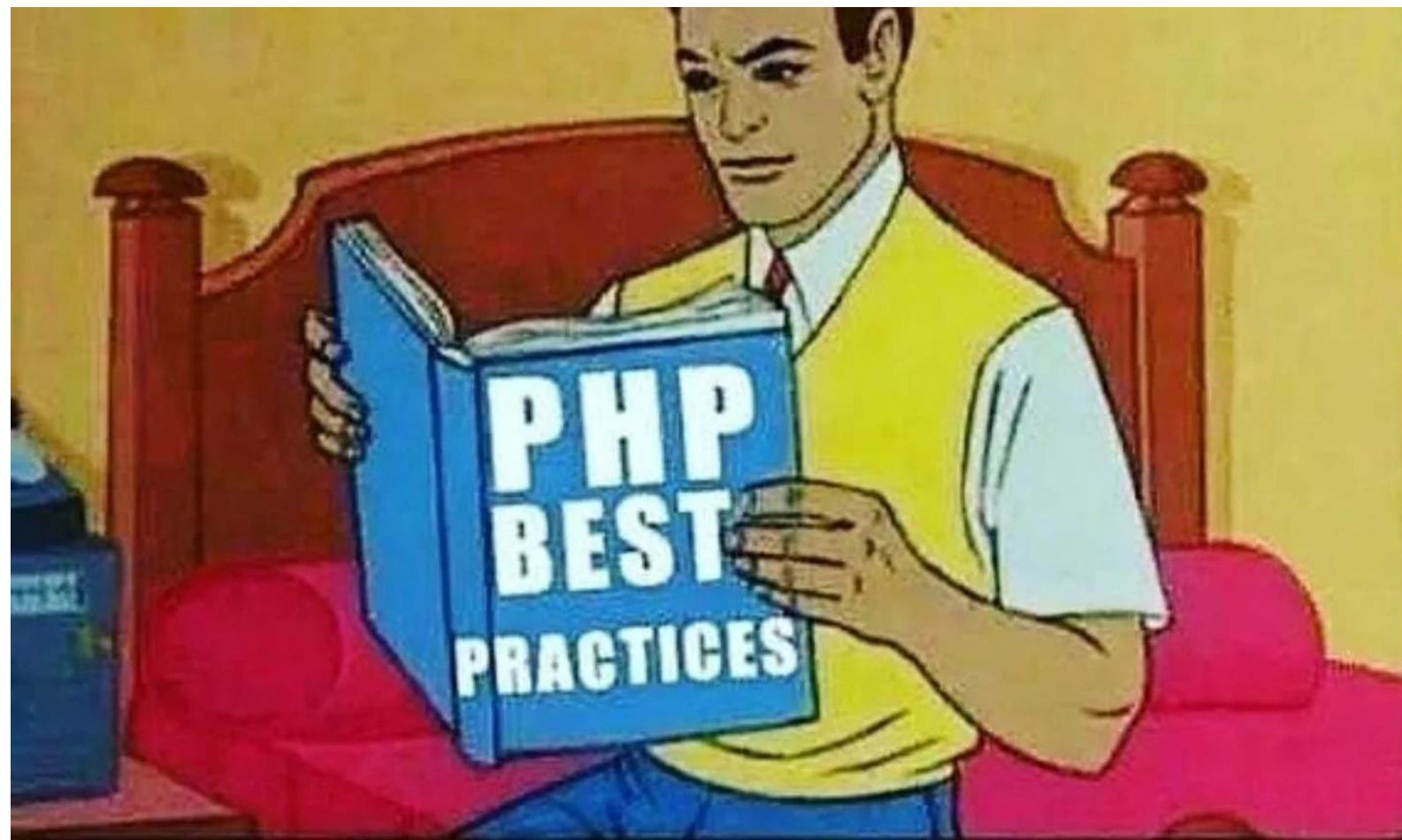


8:48

Google

Alternatives of PHP







**So where does the
negative opinion & hate
come from?**

Type system


```
1 <?php
2
3 $foo = 0;
4
5 switch( $foo ) {
6     case 'test':
7         echo 'A';
8         break;
9     case 0:
10        echo 'B';
11        break;
12    default:
13        echo 'C';
14        break;
15 }
```

```
1 <?php
2
3 $foo = 0;
4
5 switch( $foo ) {
6     case 'test':
7         echo 'A';
8         break;
9     case 0:
10        echo 'B';
11        break;
12    default:
13        echo 'C';
14        break;
15 }
```

```
// PHP >= 8.0
'B'
```



```
1 <?php
2
3 $foo = 0;
4
5 switch( $foo ) {
6     case 'test':
7         echo 'A';
8         break;
9     case 0:
10        echo 'B';
11        break;
12    default:
13        echo 'C';
14        break;
15 }
```

```
// PHP <= 7.4
'A'
```

```
1 <?php
```

```
2
```

```
3 "0e462097431906509019562988736854" === "0e830400451993494058024219903391";
```



```
1 <?php
```

```
2
```

```
3 "0e462097431906509019562988736854" === "0e830400451993494058024219903391";
```

```
4
```

```
5 // false
```

```
1 <?php
```

```
2
```

```
3 "0e462097431906509019562988736854" == "0e830400451993494058024219903391";
```

```
1 <?php
```

```
2
```

```
3 "0e462097431906509019562988736854" == "0e830400451993494058024219903391";
```

```
4
```

```
5 // true
```



```
1 <?php
2
3 md5( '240610708' ) // '0e462097431906509019562988736854'
4 md5( 'QNKCDZO' ) // '0e830400451993494058024219903391'
5
6 md5( '240610708' ) == md5( 'QNKCDZO' )
7 // true
```

```
1 <?php
```

```
2
```

```
3 declare(strict_types=1);
```

Unexpected behaviour

Return Values

Returns true on success or false on failure.

Return Values

Returns true on success or false on failure.

Caution However, if libgd fails to output the image, this function returns true.

The DateTimeInterface interface

Predefined Constants

[DateTimeInterface::ISO8601](#) [string](#)

[DATE_ISO8601](#)

ISO-8601 (example: 2005-08-15T15:52:01+0000)

The DateTimeInterface interface

Predefined Constants

[DateTimeInterface::ISO8601](#) [string](#)

[DATE_ISO8601](#)

ISO-8601 (example: 2005-08-15T15:52:01+0000)

Note: This format is not compatible with ISO-8601, but is left this way for backward compatibility reasons.

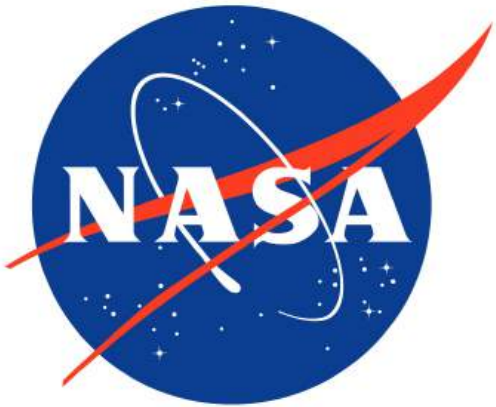
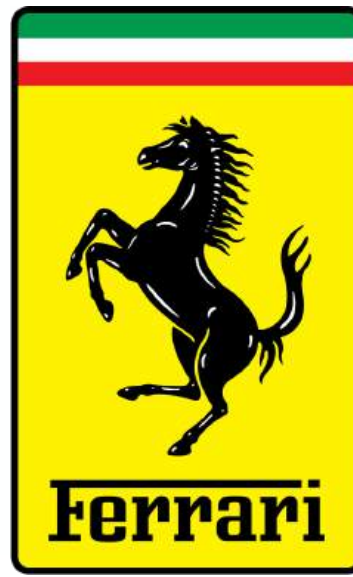
Let's talk functions

- `nl2br()` / `strtoupper()`
- `strpos()` / `str_replace()`
- `file_get_contents()`
- `fopen()`
- `filemtime()`

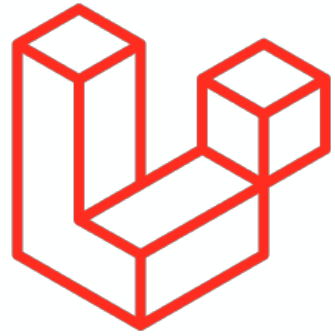

```
strpos(  
    $subject,  
    $search  
)
```

```
str_replace(  
    $search,  
    $replace,  
    $subject  
)
```

**Nobody has ever used
PHP have they?**



Foundation of the web



Laravel



Symfony



yii framework

Slim 

a micro framework for PHP



Magento®



Joomla!®



moodle



Drupal™



statamic



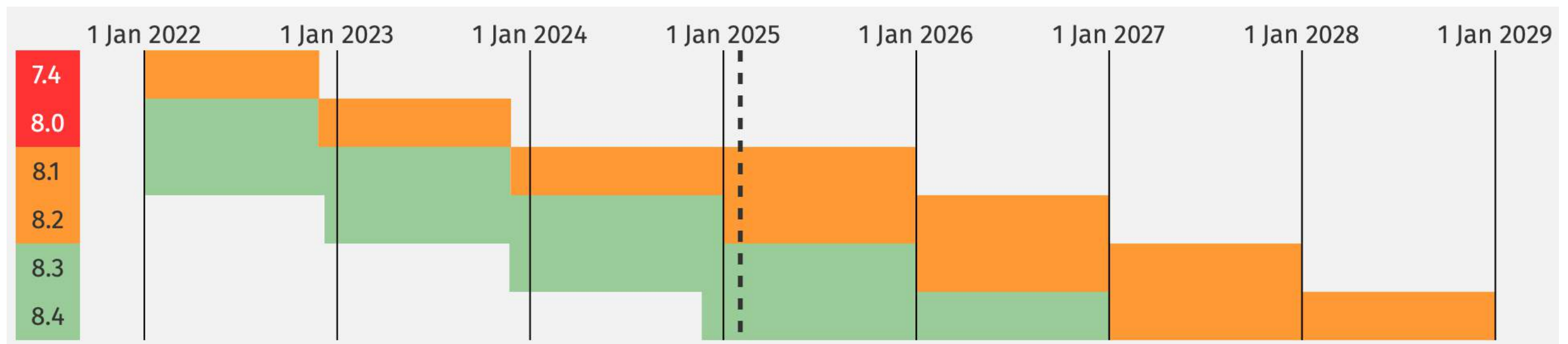
WORDPRESS

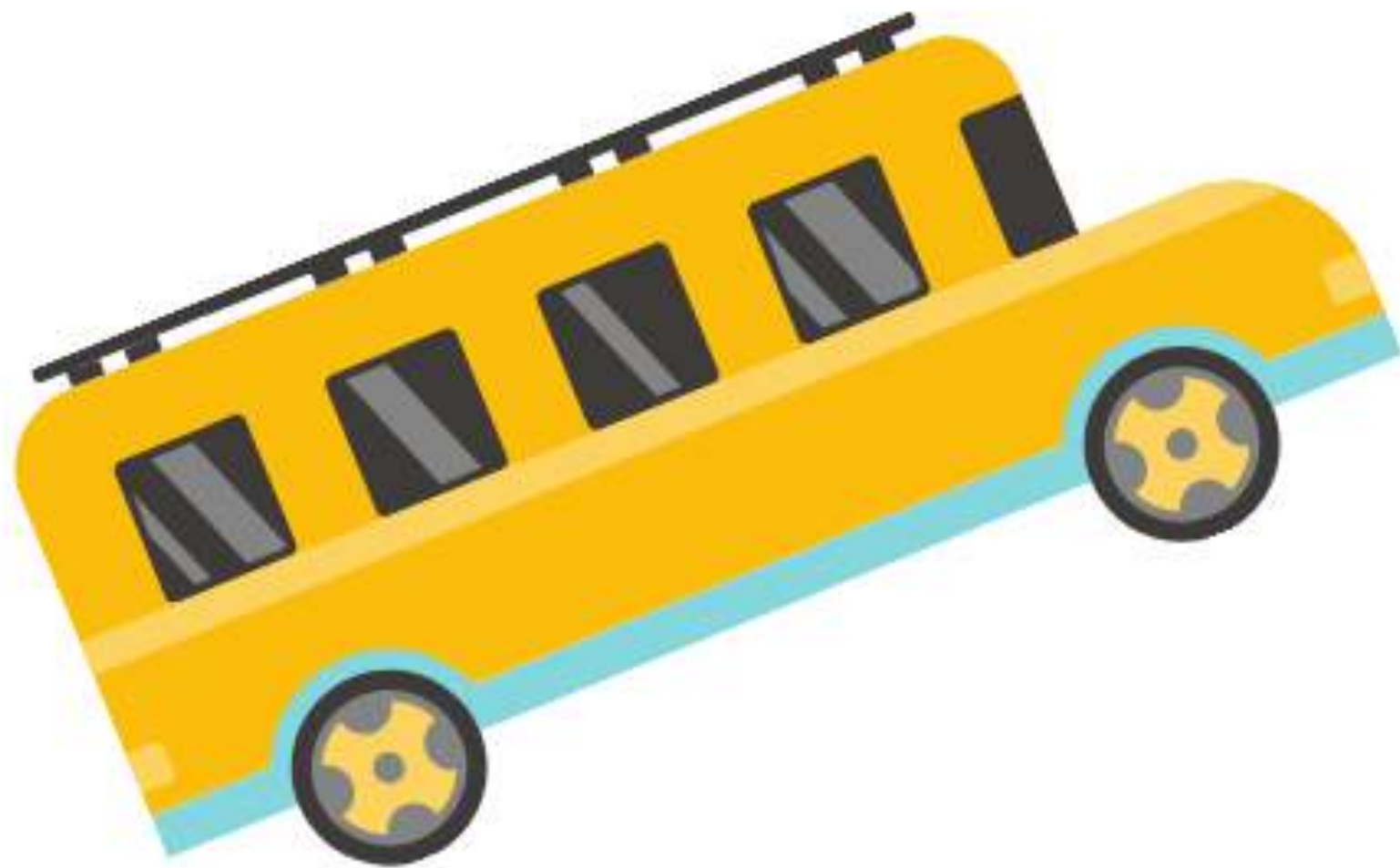


COMMERCE


```
web.php
Route::get('/funding', function (Accel $accel) {
    return $accel->fund(Laravel: '57M');
});
```


How is PHP managed?





I'M THE ONLY DEV.!

The PHP Foundation

Community



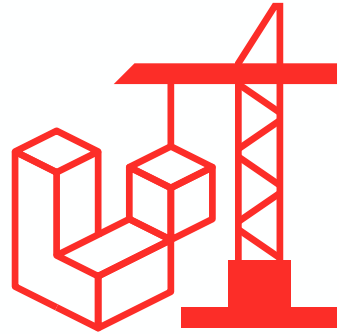
Source: <https://php.ug/>



PHP CONFERENCE JAPAN 2024



Local development



Laravel Valet

Visual Studio Code



PHP
MONITOR



Development tools



PHPUnit



PHP-CS-Fixer



Laravel Pint

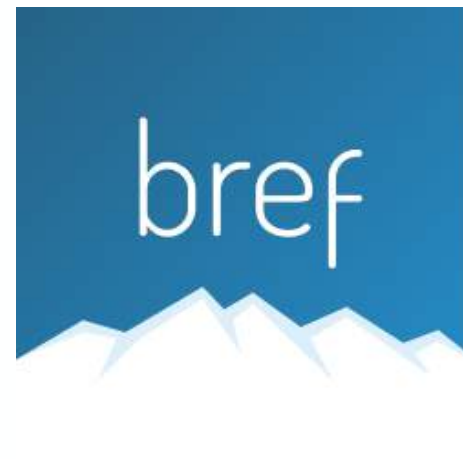
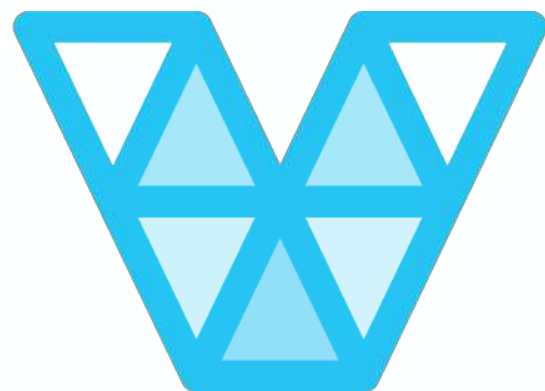
Psalm



Infrastructure



RoadRunner



**What can you do with
PHP?**



Browse catalogue or

Search catalogue



Home > Allocations > Race

Order

Daniel Ricciardo

Donington Park

Racing Driver

Change

Allocation
Race

All products >

Boots
1 remaining

Ear Plugs
3 remaining

Gloves
2 remaining

Helmet
2 remaining

Racing Glasses
2 remaining

Gender

Colour

EN 1149 - Anti Static ?



Eye on ?



This allocation will reset at the end of each season.

Boots

View usage

1 item

Remaining of 2



B43.78802-GY

MONTI S3 SRC Safety Shoe (Grey)

Monti S3 SRC Safety Shoe



1 in stock

Option

Footwear Size (UK): 9 (€62.64 ex VAT)

Reason

New issue

Quantity

1

Add to Basket

€62.64 (ex VAT) ~~€62.64 (inc VAT)~~



B43.78950-GY

YOUTH S3 SRC Safety Shoe (Grey)

Youth s3 src.



Option

Footwear Size (UK): 9 (€73.08 ex VAT)

Reason

New issue

Schedule 02-02-2025 to 16-03-2025

Seat Planner

Change Dates

Filter Applied

Clear Filters

	Sun 2 Feb	Mon 3 Feb	Tue 4 Feb	Wed 5 Feb	Thu 6 Feb	Fri 7 Feb	Sat 8 Feb	Sun 9 Feb	Mon 10 Feb	Tue 11 Feb	Wed 12 Feb	Thu 13 Feb	Fri 14 Feb	Sat 15 Feb	Sun 16 Feb	Mon 17 Feb	Tue 18 Feb	Wed 19 Feb	Thu 20 Feb	Fri 21 Feb	Sat 22 Feb	Sun 23 Feb	Mon 24 Feb	Tue 25 Feb	Wed 26 Feb	Thu 27 Feb	Fri 28 Feb	Sat 1 Mar	
Lighting Artist (Lead) Ben Hartley										V A U X	V A U X	V A U X	V A U X			V A U X	V A U X	V A U X	V A U X	V A U X			V A U X	V A U X	V A U X	V A U X	V A U X		
Lighting Artist (Lead) Suzanne Dupont		D U S T 2	D U S T 2	D U S T 2	D U S T 2	D U S T 2			D U S T 2	D U S T 2	D U S T 2	D U S T 2	D U S T 2			D U S T 2	D U S T 2	D U S T 2	D U S T 2	D U S T 2			A S I 3	A S I 3	A S I 3	A S I 3	A S I 3		
Lighting Artist (Senior) Suzanne Falar		L O T	L O T	L O T	L O T	L O T			L O T	L O T	L O T	L O T	L O T			L O T	L O T	L O T	L O T	L O T			!	!	!	!	!		
Lighting Artist (Lead) James Joseph Parsons		F O X	F O X	F O X	F O X	F O X			F O X	F O X	F O X	F O X	F O X			F O X	F O X	F O X	F O X	F O X			F O X	F O X	F O X	F O X	F O X		
Lighting Artist (Senior) Matthew Spencer		A S I 3	A S I 3	A S I 3	A S I 3	A S I 3			A S I 3	A S I 3	A S I 3	A S I 3	A S I 3			A S I 3	A S I 3	A S I 3	A S I 3	A S I 3			A S I 3	A S I 3	A S I 3	A S I 3	A S I 3		
Lighting Artist (Senior) Nick Langford		A S I 3	A S I 3	A S I 3	A S I 3	A S I 3			A S I 3	A S I 3	A S I 3	A S I 3	A S I 3			A S I 3	A S I 3	A S I 3	A S I 3	A S I 3			A S I 3	A S I 3	A S I 3	A S I 3	A S I 3		
11: Compositing																													
Composer (Lead) Antonio Ramos		D U S T 2	D U S T 2	D U S T 2	D U S T 2	D U S T 2			D U S T 2	D U S T 2	D U S T 2	D U S T 2	D U S T 2																
Composer (Mid) Chetan Gaur																								V A U X	V A U X	V A U X	V A U X	V A U X	
Composer (Senior) Chris Callahan		L O T	L O T	L O T	L O T	L O T			L O T	L O T	L O T	L O T	L O T																
Composer (Lead) Elena Estevez Santos		A S I 3	A S I 3	A S I 3	A S I 3				A S I 3	A S I 3	A S I 3	A S I 3				A S I 3	A S I 3	A S I 3	A S I 3				A S I 3	A S I 3	A S I 3	A S I 3			
Composer (Senior) Gilles Collart		!	!	!	!	!			D U S T 2	D U S T 2	D U S T 2	D U S T 2	D U S T 2																
Composer (Junior)		D U S T 2	D U S T 2	D U S T 2	D U S T 2	D U S T 2			D U S T 2	D U S T 2	D U S T 2	D U S T 2	D U S T 2			D U S T 2	D U S T 2	D U S T 2											

ORDER 99999: BOURNEMOUTH BH23

Download PDF

Order Processing

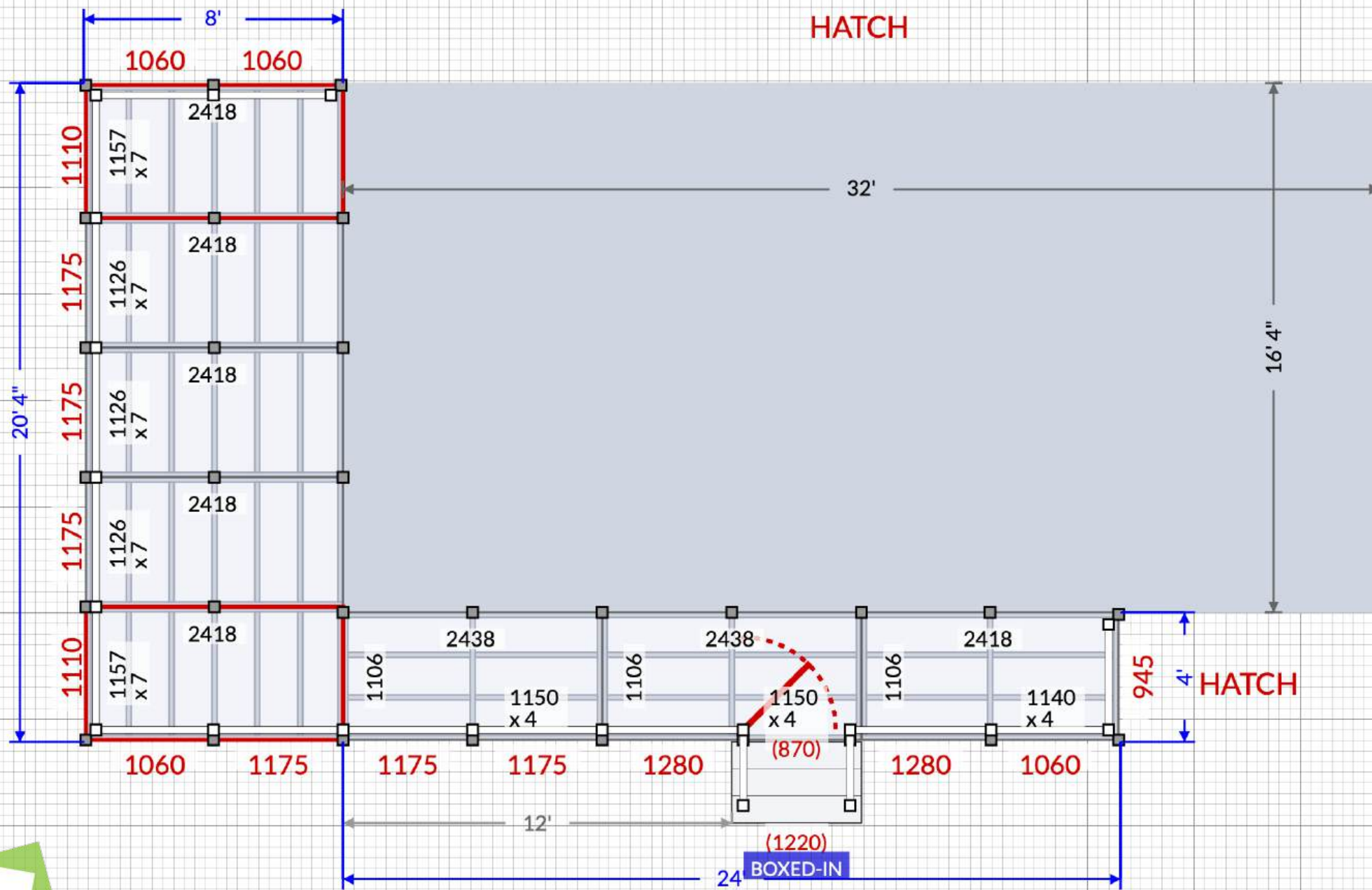
Build Drawing

Kit List

Quote

Fitting

Redraw Build Drawing





449

Plot Details

Plot Report

Proposed plot area	Plot type	Existing building area	Existing land area	Building coverage
7,911.1 sq ft	Rear garden	740.6 sq ft	17,164.7 sq ft	4.3%

Local planning authority
Runnymede

Electoral ward
Virginia Water

Plot attributes

- 🏠 Has outbuildings (site)
- 🏠 No outbuildings (plot)
- 🏰 Not a listed building
- 🔥 Review fire service access

Area attributes

- 💧 Not at risk of flooding
- 🗑️ Not in a historic landfill area
- 👤 Not in an SSSI area
- 🌳 Not in a conservation area

+

-

▲

Map view

Satellite view

With overlay

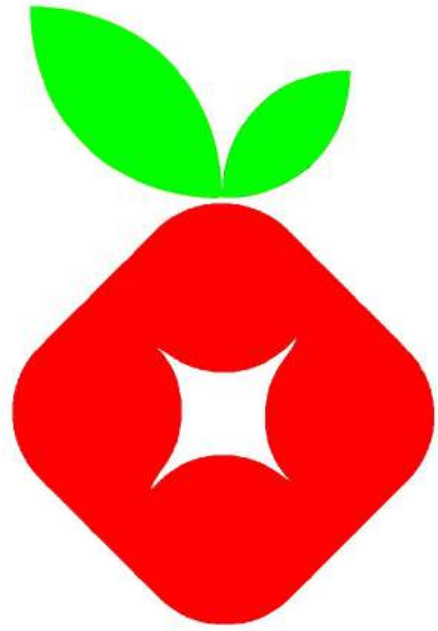
Map Layers	Show / Hide
■ Existing land	
■ Existing building	
■ Potential plot	
■ Access route	
■ Outbuildings	<input checked="" type="checkbox"/>
■ Nearby flood areas	<input checked="" type="checkbox"/>
■ Nearby SSSIs	<input checked="" type="checkbox"/>
■ Nearby conservation areas	<input checked="" type="checkbox"/>
■ Nearby historic landfill areas	<input checked="" type="checkbox"/>
■ Nearby listed buildings	<input checked="" type="checkbox"/>

Hide Map Key

© Mapbox © OpenStreetMap Improve this map

Plot feedback

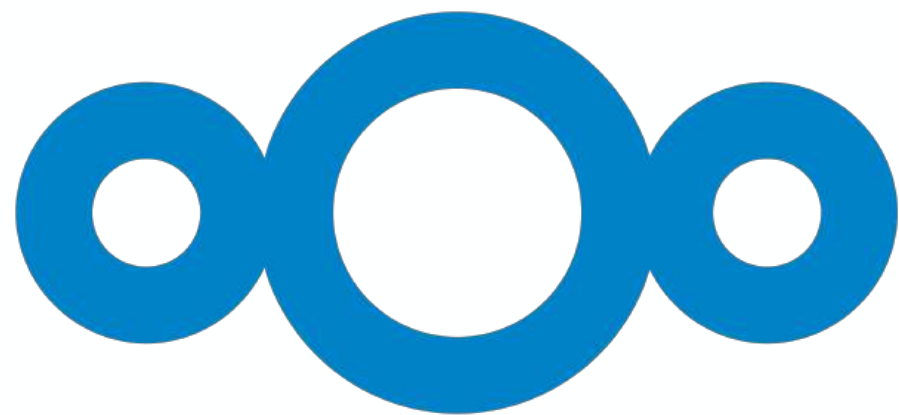
Next: Define building



Pi-Hole



Pixelfed



Nextcloud

fathom
analytics/

Beyond the browser

⌘1

php

github-prompts on `main` via `via` v8.3.4
> php artisan github

Interact with GitHub using Laravel Prompts!

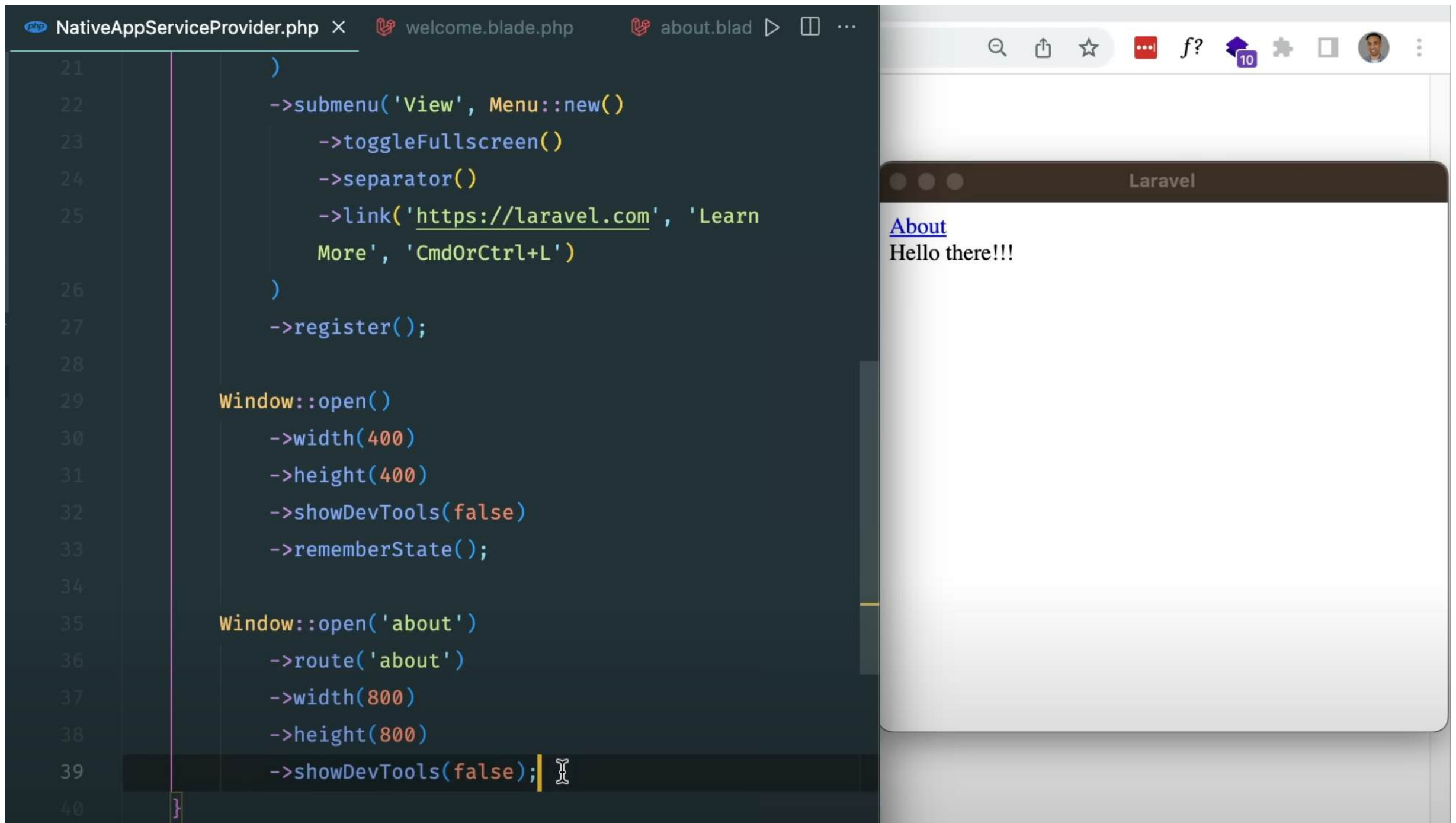
Menu:

List your public GitHub repositories

Select a repository:

- > ● laravel-mailcoach-sdk
- abuse-ipdb-laravel
- artisan.page
- ash-jc-allen
- Awesome-Black-Friday-Cyber-Monday

NativePHP

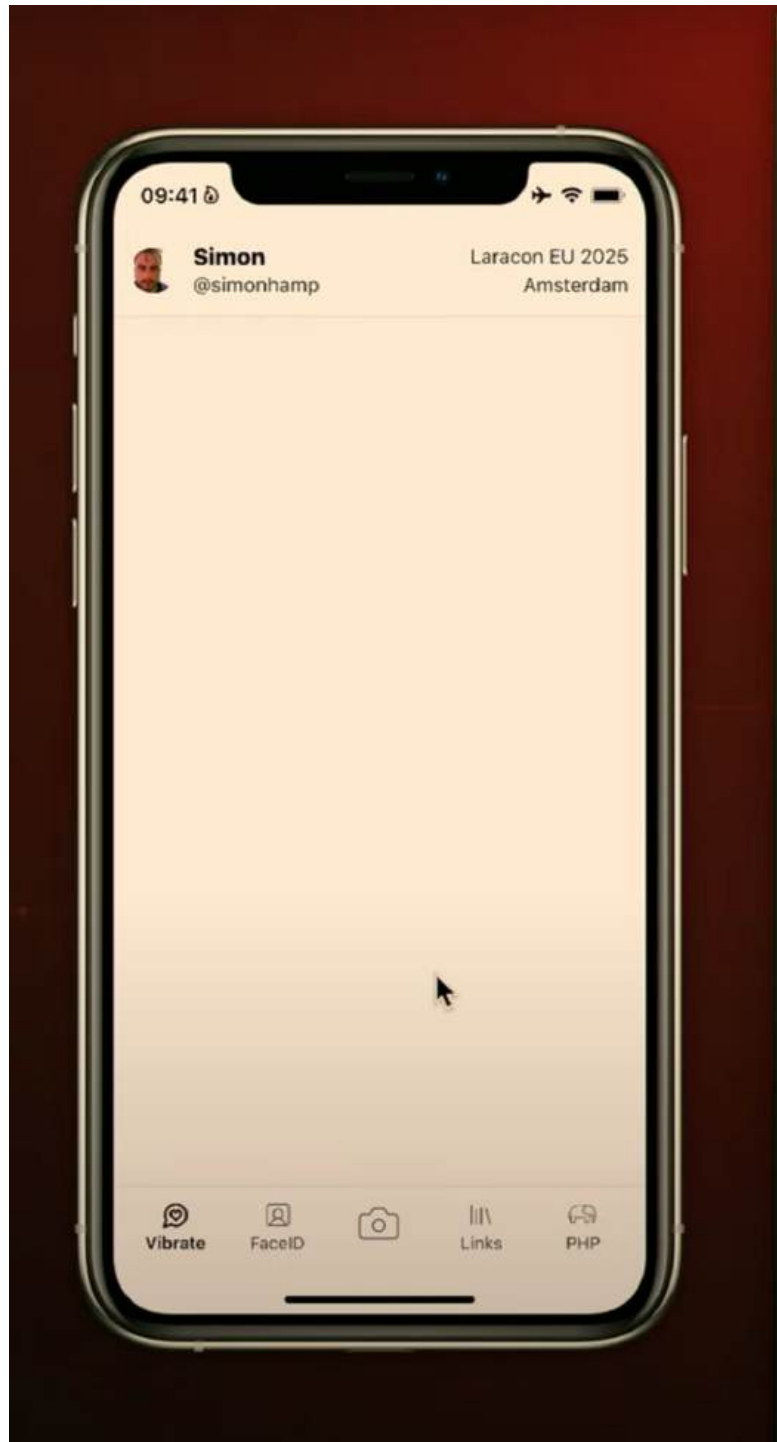


The image shows a development environment with a code editor on the left and a browser window on the right. The code editor displays the following PHP code in `NativeAppServiceProvider.php`:

```
21     )
22     ->submenu('View', Menu::new()
23         ->toggleFullscreen()
24         ->separator()
25         ->link('https://laravel.com', 'Learn
26           More', 'CmdOrCtrl+L')
27     )
28     ->register();
29
29     Window::open()
30         ->width(400)
31         ->height(400)
32         ->showDevTools(false)
33         ->rememberState();
34
35     Window::open('about')
36         ->route('about')
37         ->width(800)
38         ->height(800)
39         ->showDevTools(false);
40 }
```

The browser window, titled "Laravel", shows the rendered output of the application. It features a blue link labeled "About" and the text "Hello there!!!".

NativePHP



```
demo main Main
1 <?php
2
3 namespace App\Livewire;
4 |
5 use Livewire\Attributes\On;
6 use Livewire\Attributes\Reactive;
7 use Livewire\Component;
8
9 no usages
10 class Demo extends Component
11 {
12     #[Reactive]
13     public $current;
```



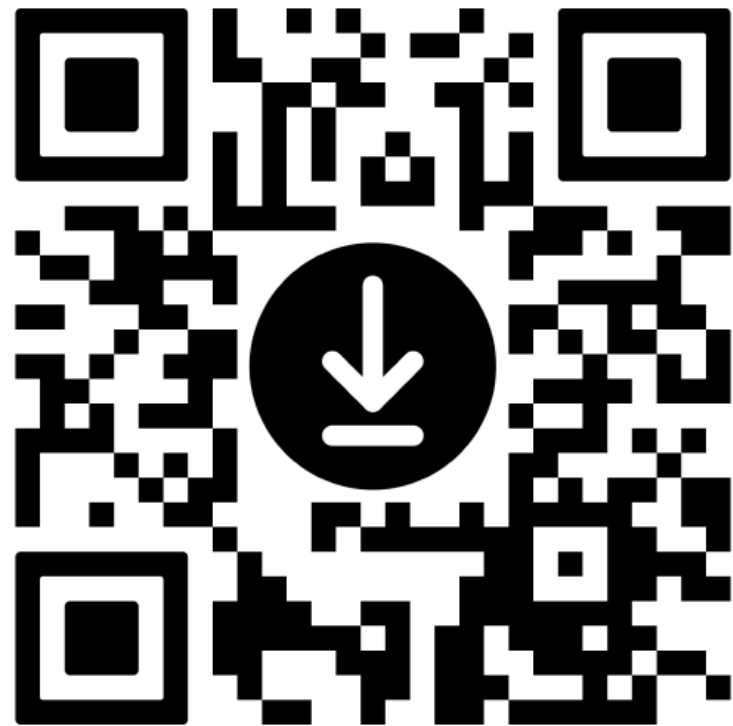


Is PHP dead?

THEY WERE ALL LYING ABOUT PHP



Thanks for listening.



< Download slides

<https://georgebuckingham.com/talks>



yammayap

We're hiring for a senior PHP developer >

<https://yammayap.com/jobs>

