

*php* 8

# Content

- Overview
  - Version support
- What's new
  - Features / deprecations
  - JIT
- PHP 8.1



# Version support



# Supported Versions

<https://www.php.net/supported-versions.php>

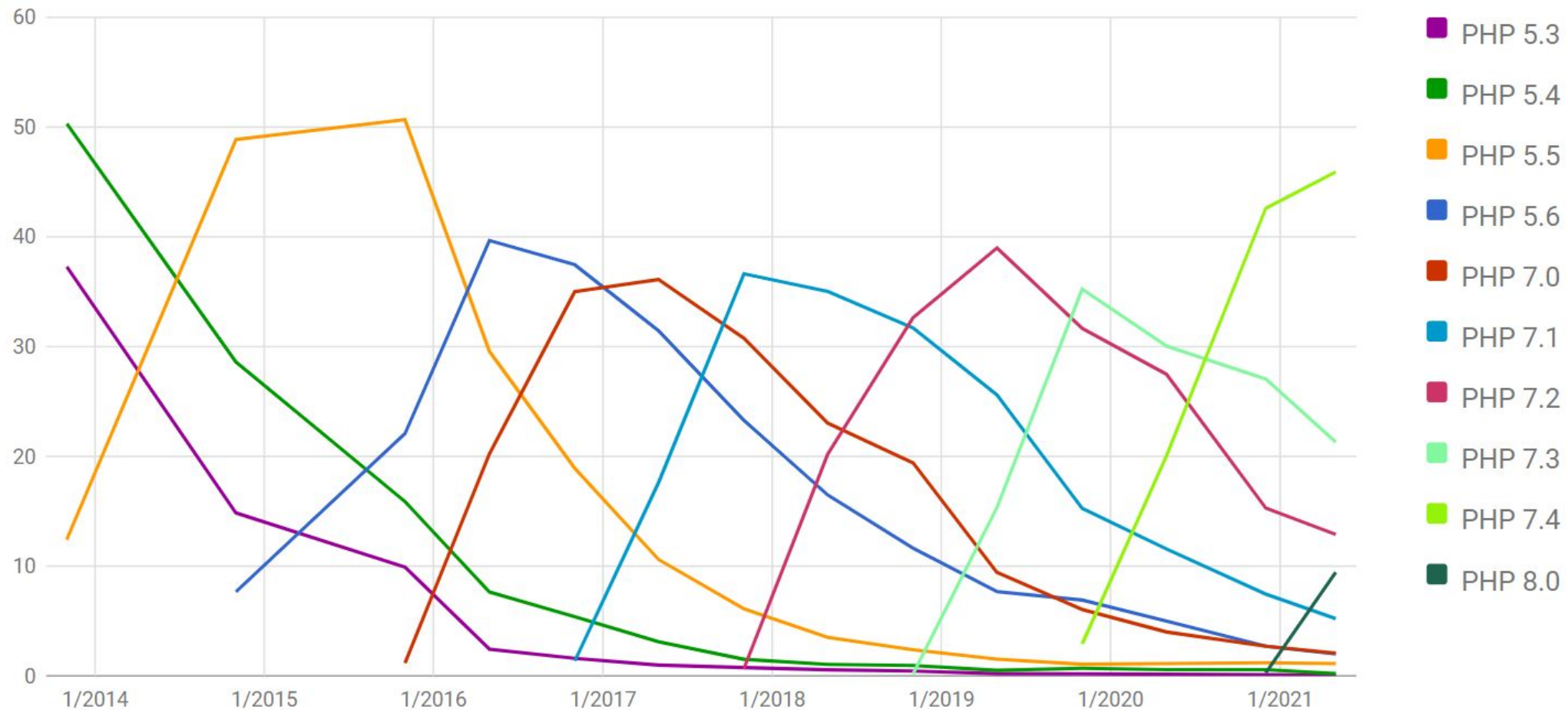


# PHP Versions Stats - 2020.2 Edition

<https://blog.packagist.com/php-versions-stats-2021-1-edition/>



PHP Versions (based on Composer usage data)



# PHP 8.0



- Released 26 nov 2020
- <https://wiki.php.net/todo/php80#timetable>
- <https://www.php.net/archive/2020.php#2020-11-26-3>
- Release managers
  - Sara Golemon
  - Gabriel Caruso

# 150 contributors

<https://php.watch/articles/php80-posters>



# Microsoft



- Will not provide support for PHP 8.0 on Windows
  - <https://news-web.php.net/php.internals/110907>
- Patches for EOL PHP versions
  - Backported into <https://github.com/Microsoft/php-src>
  - Pulled by Debian
  - I kid you not



**Practical**



# Hosting

- Combell shared hosting
  - Support since RC's <3



# 3th party tools



- Composer 1 + 2
- PhpStorm
  - Full support in version 2020.3
  - Released 3 dec 2020
  - <https://www.jetbrains.com/phpstorm/whatsnew/>
  - <https://www.youtube.com/watch?v=3OLKg7exBWc>
- Tools like PHPStan, Psalm, PHPUnit, GrumPHP, ... are ready

# Extensions

<https://blog.remirepo.net/post/2020/09/21/PHP-extensions-status-with-upcoming-PHP-8.0>

- apcu
- mongodb
- redis
- memcached
- amqp
- zmq
- imagick
- geoip
- ssh2
- Xdebug 3.0.0
- Blackfire
- ...



# Packagist



- Nearly 300 000 packages <https://packagist.org/statistics>
  - Not obvious what works, lot of packages don't even have PHP version requirements
  - Major version bump, greater BC risk!
  - Trial en error
- Composer version requirements
  - <https://twitter.com/nicolasgrekas/status/1263023258938548225>
  - <https://blog.jetbrains.com/phpstorm/2020/09/video-nikita-popov-and-nikolas-grekas-on-symfony-php-8-compatibility/>

# Frameworks



# Frameworks

- Symfony
  - All supported branches
- Doctrine
  - DBAL, ORM, migrations
- Drupal
  - From 9.1 <https://www.drupal.org/project/drupal/issues/3109885>
- WordPress
  - From 5.6 <https://make.wordpress.org/core/2020/10/06/call-for-testing-php-8-0/>
- Laravel
  - Supported in 6.x <https://github.com/laravel/framework/pull/33388>
- Shopware
  - Planned for 5.7 and 6.4 <https://twitter.com/mitelg/status/1330179068428480513>
- ApiPlatform
  - Support since 2.5.8 <https://github.com/api-platform/core/releases/tag/v2.5.8>

# Features





## 7.x added nice features!

- 7.0 <http://php.net/manual/en/migration70.php>
  - operators (<=>, ??), scalar en return type hints, Throwable / Error, anonymous classes, ...
- 7.1 <http://php.net/manual/en/migration71.php>
  - nullable types, void functions, symmetric array destructuring, class constant visibility, iterable, multi-catch exception handling, ...
- 7.2 <http://php.net/manual/en/migration72.php>
  - object type, abstract method overriding, Sodium, Argon2, extended string types for PDO, ...
- 7.3 <http://php.net/manual/en/migration73.php>
  - more flexible here- / nowdoc syntax, trailing commas in calls, array\_key\_first / \_last, is\_countable, ...
- 7.4 <http://php.net/manual/en/migration74.php>
  - typed properties, arrow functions, covariant return types, contravariant argument types, null coalescing assignment operator, custom object serialization, ext/ffi, preloading, ...

## 5.x added nice features!

- 5.3 <http://php.net/manual/en/migration53.php>
  - namespaces, closures, goto, mysqlnd, late static binding (self/static), ...
- 5.4 <http://php.net/manual/en/migration54.php>
  - short array syntax ([ ]), internal webserver, traits, short echo tags (<?=>), ...
- 5.5 <http://php.net/manual/en/migration55.php>
  - generators, finally, OPcache, DateTimeImmutable, password hashing api, array\_column, ...
- 5.6 <http://php.net/manual/en/migration56.php>
  - > 2GB uploads, variadict functions + argument unpacking (...), constant scalar expressions, phpdbg, ...

# New in 8.0

<https://www.php.net/releases/8.0/en.php>

[https://www2.slideshare.net/nikita\\_ppv/whats-new-in-php-80-239762987](https://www2.slideshare.net/nikita_ppv/whats-new-in-php-80-239762987)



# Constructor property promotion

[https://wiki.php.net/rfc/constructor\\_promotion](https://wiki.php.net/rfc/constructor_promotion)



PHP 7

```
class Point
{
    public float $x;
    public float $y;
    public float $z;

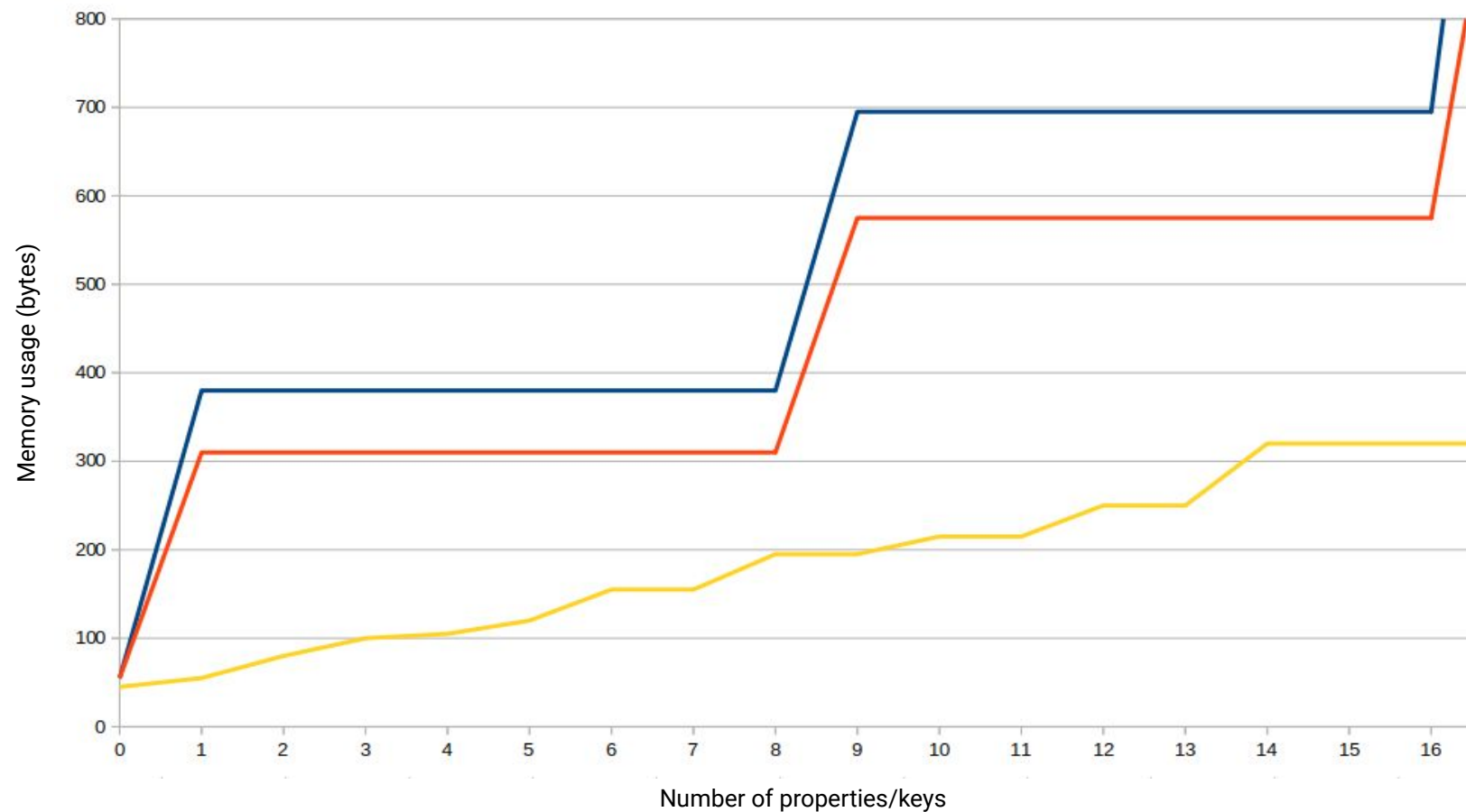
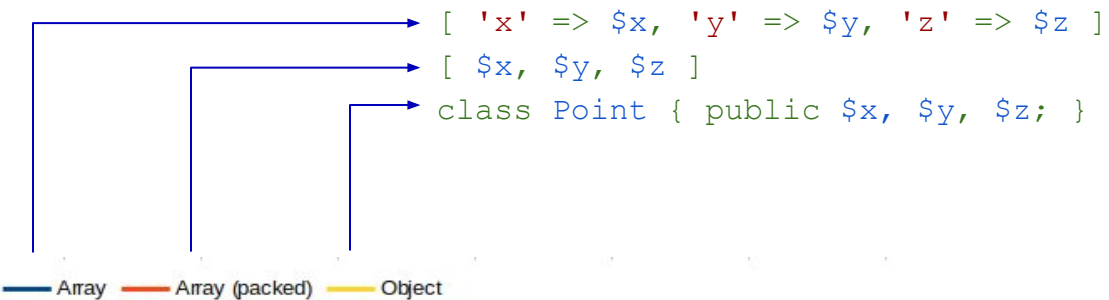
    public function __construct(
        float $x = 0.0,
        float $y = 0.0,
        float $z = 0.0
    ) {
        $this->x = $x;
        $this->y = $y;
        $this->z = $z;
    }
}
```

PHP 8

```
class Point
{
    public function __construct(
        public float $x = 0.0,
        public float $y = 0.0,
        public float $z = 0.0,
    ) {}
}
```

# Memory usage

<https://youtu.be/JBWgvUrb-q8>



# Tailing comma in parameter list

[https://wiki.php.net/rfc/trailing\\_comma\\_in\\_parameter\\_list](https://wiki.php.net/rfc/trailing_comma_in_parameter_list)



PHP 8

```
class Point
{
    public function __construct(
        public float $x = 0.0,
        public float $y = 0.0,
        public float $z = 0.0,
    ) {}
}
```

# Named arguments

[https://wiki.php.net/rfc/named\\_params](https://wiki.php.net/rfc/named_params)



PHP 7

```
htmlspecialchars(  
    $string,  
    ENT_COMPAT | ENT_HTML401,  
    'UTF-8',  
    false  
);  
  
array_fill(0, 100, 50);
```

PHP 8

```
htmlspecialchars(  
    $string,  
    double_encode: false  
);  
  
array_fill(  
    value: 50,  
    count: 100,  
    start_index: 0,  
);
```

# call\_user\_func\_array()

<https://www.php.net/manual/en/migration80.incompatible.php>



PHP 7

```
function name($fName, $lName) {
    return "{$fName} {$lName}";
}

call_user_func_array(
    'name', ['fName' => 'John', 'lName' => 'Doe']
);
// John Doe

call_user_func_array(
    'name', ['lName' => 'Doe', 'fName' => 'John']
);
// Doe John

call_user_func_array(
    'name', ['f_name' => 'John', 'l_name' => 'Doe']
);
// John Doe
```

PHP 8

```
function name($fName, $lName) {
    return "{$fName} {$lName}";
}

call_user_func_array(
    'name', ['fName' => 'John', 'lName' => 'Doe']
);
// John Doe

call_user_func_array(
    'name', ['lName' => 'Doe', 'fName' => 'John']
);
// John Doe

call_user_func_array(
    'name', ['f_name' => 'John', 'l_name' => 'Doe']
);
// Uncaught Error: Unknown named parameter $f_name
```



# Attributes

[https://wiki.php.net/rfc/attributes\\_v2](https://wiki.php.net/rfc/attributes_v2)



PHP 7

```
class PostsController
{
    /**
     * @Route("/api/posts/{id}", methods={"GET"})
     */
    public function get($id) { /* ... */ }
}
```

PHP 8

```
class PostsController
{
    #[Route("/api/posts/{id}", methods: ["GET"])]
    public function get($id) { /* ... */ }
}
```

# Attributes

[https://wiki.php.net/rfc/attributes\\_v2](https://wiki.php.net/rfc/attributes_v2)



- Support in Symfony <https://symfony.com/blog/new-in-symfony-5-2-php-8-attributes>
- Support in ApiPlatform <https://github.com/api-platform/core/pull/3851>
- Doctrine annotations <https://github.com/doctrine/orm/pull/8266> (draft)
- PhpStorm attributes package <https://github.com/JetBrains/phpstorm-attributes>
- ...

# Mixed type

[https://wiki.php.net/rfc/mixed\\_type\\_v2](https://wiki.php.net/rfc/mixed_type_v2)



PHP 7

```
/**
 * @var mixed $foo
 *
 * @return mixed
 */
function foo($foo)
{
    return $foo;
}
```

PHP 8

```
function foo(mixed $foo): mixed
{
    return $foo;
}
```

# Union types

[https://wiki.php.net/rfc/union\\_types\\_v2](https://wiki.php.net/rfc/union_types_v2)



PHP 7

```
/** @var int|string */
function foo($foo): void
{
    /* ... */
}

/** @return int|false */
function bar(?string $bar)
{
    /* ... */
}
```

PHP 8

```
function foo(int|string $foo): void
{
    /* ... */
}

function bar(?string $bar): int|false
{
    /* ... */
}
```

# Match expression

[https://wiki.php.net/rfc/match\\_expression\\_v2](https://wiki.php.net/rfc/match_expression_v2)



PHP 7

```
switch (8.0) {
    case '8.0':
        $result = "Oh no!";
        break;
    case 8.0:
        $result = "This is what I expected";
        break;
}
echo $result;
//> Oh no!
```

PHP 8

```
echo match (8.0) {
    '8.0' => "Oh no!",
    8.0 => "This is what I expected",
    // default => throw new UnhandledMatchError(),
};
//> This is what I expected
```

# Nullsafe operator

[https://wiki.php.net/rfc/nullsafe\\_operator](https://wiki.php.net/rfc/nullsafe_operator)



PHP 7

```
$country = null;

if ($session !== null) {
    $user = $session->user;

    if ($user !== null) {
        $address = $user->getAddress();

        if ($address !== null) {
            $country = $address->country;
        }
    }
}
```

PHP 8

```
$country = $session?->user?->getAddress()?->country;
```

# Clean Code

A Handbook of Agile Software Craftsmanship



Chapter 7: Error Handling

```
try {  
    $country = $session->user()->address()->country();  
} catch (\Throwable) {  
    throw new NoCountryException();  
}
```

# Non-capturing catches

[https://wiki.php.net/rfc/non-capturing\\_catches](https://wiki.php.net/rfc/non-capturing_catches)



PHP 7

```
try {
    $country = $session->user()->address()->country();
} catch (\Throwable $e) {
    throw new NoCountryException();
}
```

PHP 8

```
try {
    $country = $session->user()->address()->country();
} catch (\Throwable) {
    throw new NoCountryException();
}
```



# String to Number Comparison

[https://wiki.php.net/rfc/string\\_to\\_number\\_comparison](https://wiki.php.net/rfc/string_to_number_comparison)



Comparison	PHP 7	PHP 8
<code>0 == "0"</code>	TRUE	TRUE
<code>0 == "0.0"</code>	TRUE	TRUE
<code>0 == "foo"</code>	TRUE	FALSE
<code>0 == ""</code>	TRUE	FALSE
<code>42 == " 42"</code>	TRUE	TRUE
<code>42 == "42foo"</code>	TRUE	FALSE

# Consistent type errors for internal functions

[https://wiki.php.net/rfc/consistent\\_type\\_errors](https://wiki.php.net/rfc/consistent_type_errors)



PHP 7

```
strlen([]); // Warning: strlen() expects parameter 1 to be string, array given  
array_chunk([], -1); // Warning: array_chunk(): Size parameter expected to be greater than 0
```

PHP 8

```
strlen([]); // TypeError: strlen(): Argument #1 ($str) must be of type string, array given  
array_chunk([], -1); // TypeError: array_chunk(): Argument #2 ($length) must be greater than 0
```

# Weak maps

[https://wiki.php.net/rfc/weak\\_maps](https://wiki.php.net/rfc/weak_maps)



PHP 8

```
class FooBar
{
    private WeakMap $cache;

    public function getSomethingWithCaching(object $obj)
    {
        return $this->cache[$obj] ??= $this->computeSomethingExpensive($obj);
    }

    // ...
}
```

# Concatenation precedence

[https://wiki.php.net/rfc/concatenation\\_precedence](https://wiki.php.net/rfc/concatenation_precedence)



PHP 7

```
echo "sum: " . $a + $b;  
// echo ("sum: " . $a) + $b;
```

PHP 8

```
echo "sum: " . $a + $b;  
// echo "sum: " . ($a + $b);
```

# Resource to object migration



- Phasing out resources
- Replaced by opaque objects
- Watch out with `is_resource()` calls!
- Curl, Gd, Sockets, OpenSSL, XMLWriter, XML extensions done
- More to come

# Others

<https://www.php.net/manual/en/migration80.php>

- New Stringable interface <https://wiki.php.net/rfc/stringable>
- New `str_contains()`, `str_starts_with()`, `str_ends_with()` functions
- The `@` operator no longer silences fatal errors.
- Allow `::class` on objects.
- `static` return type
- `throw` is an expression
- Reclassified engine warnings
- `ext/json` is always available
- `xmlrpc` moved to `pecl`
- ...



# Compatibility

- Static analyzers (phpstan)
- PHPCompatibility/PHPCompatibility

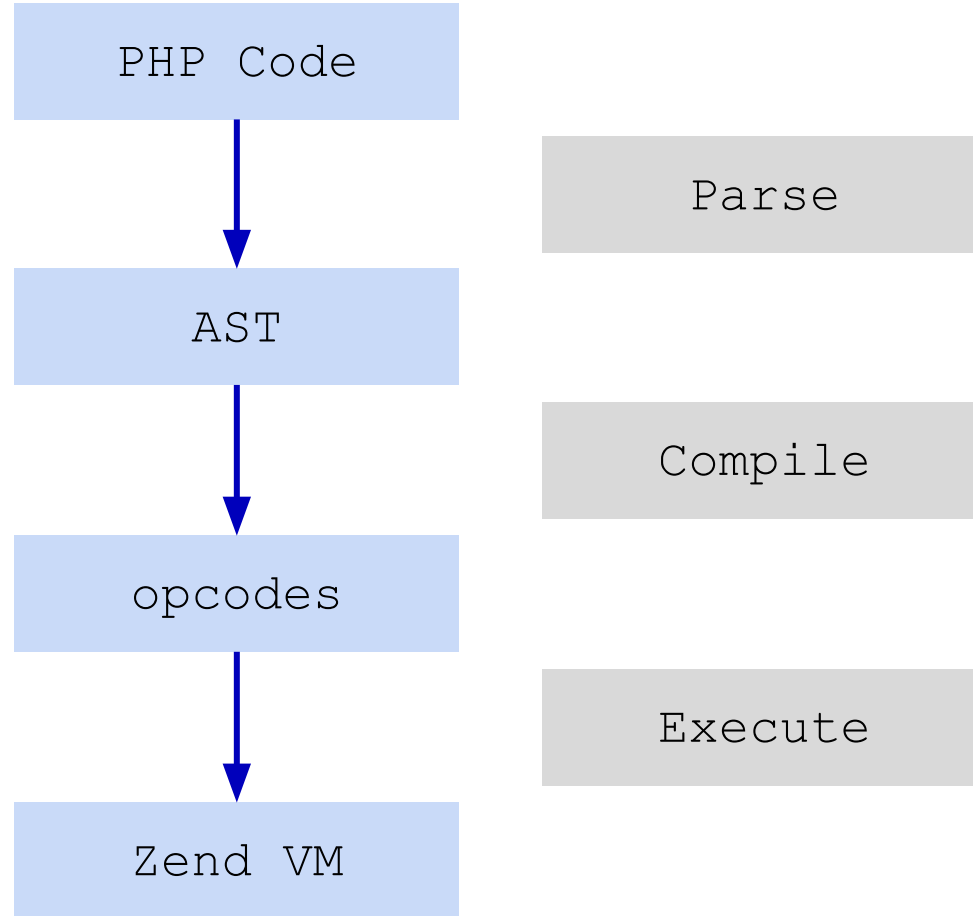


**JIT**

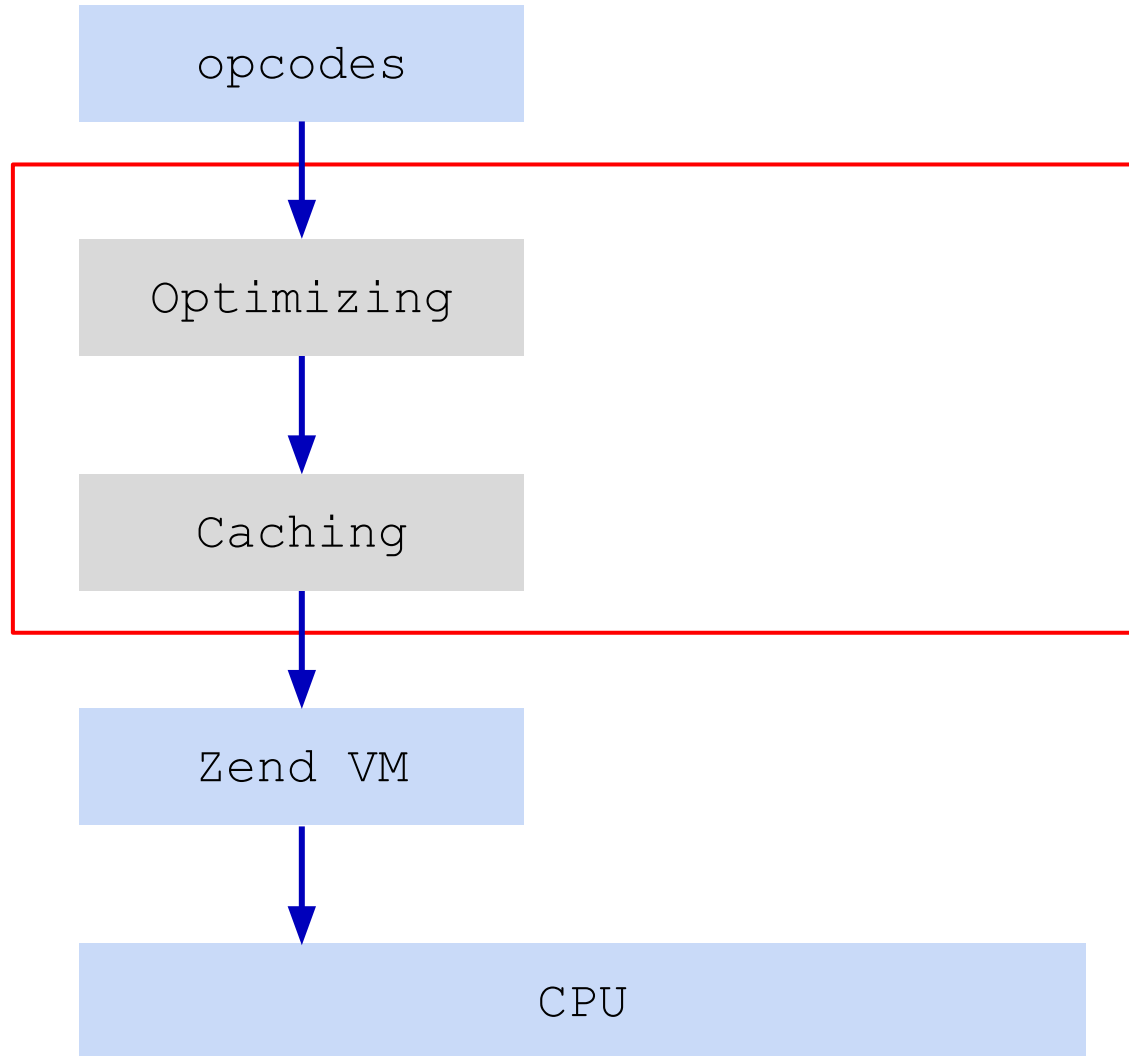




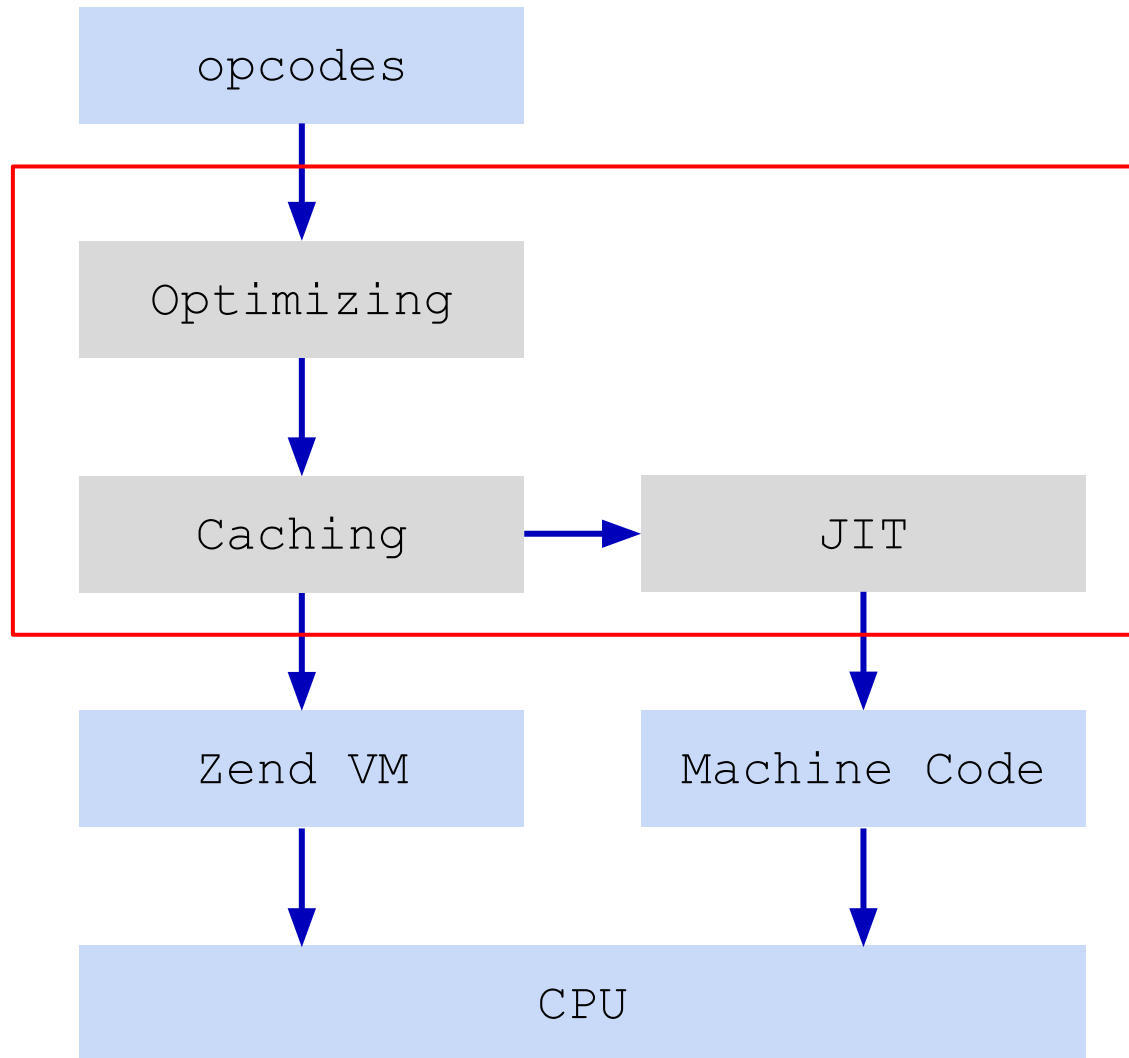
# How is PHP compiled



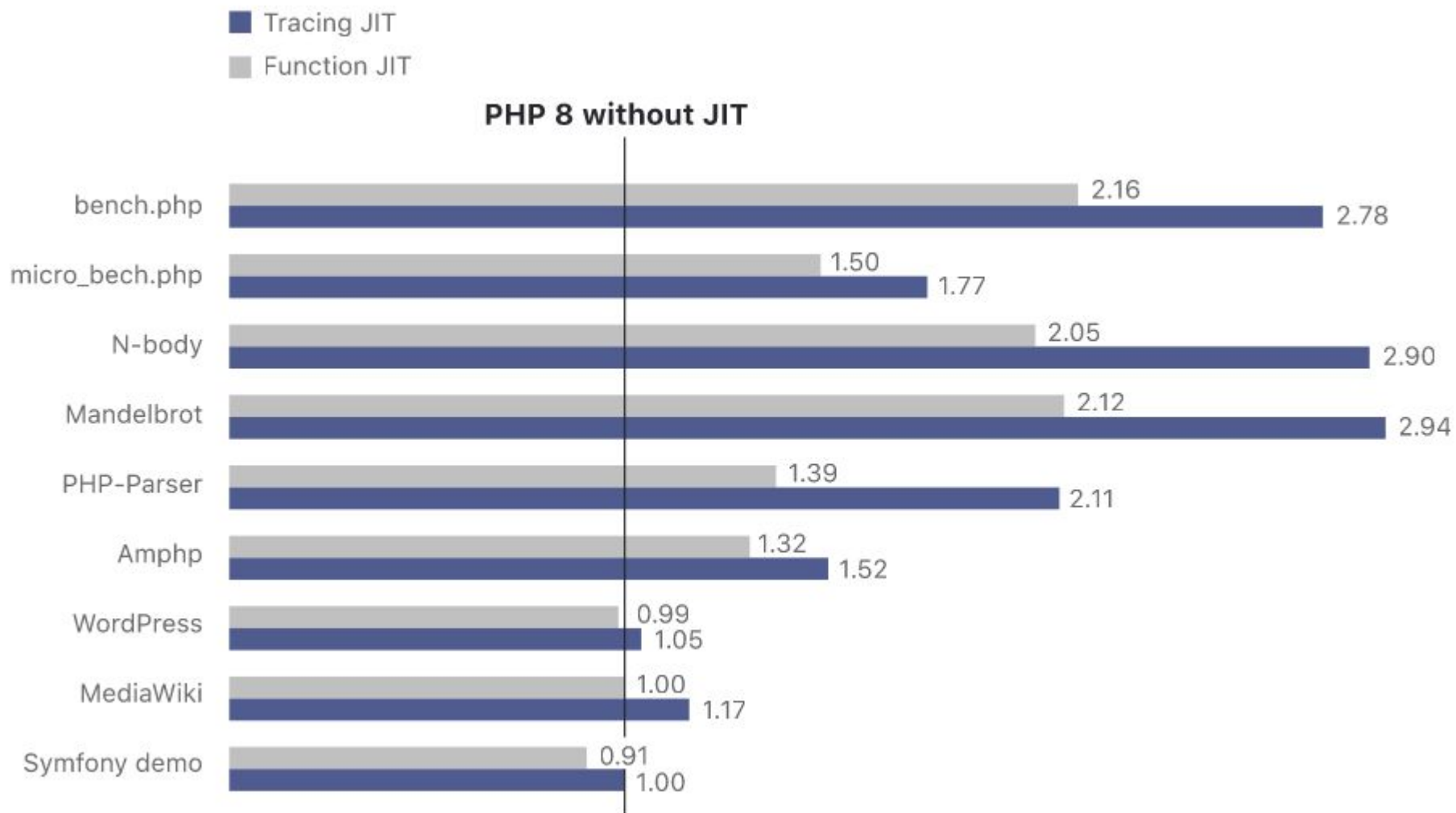
# ext/opcache



# ext/opcache



# Speed



# JIT

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

<https://blog.krakjoe.ninja/2021/05/avoiding-busses.html>

- Pro
  - Enables new types of workloads on PHP
  - CPU intensive workloads
- Con
  - No benefits for typical web workload
  - Stability
  - Maintenance
    - Developed by Dmitry Stogov
    - Understood by ~3 people



# JIT configuration

<https://www.php.net/manual/en/opcache.configuration.php>



- `opcache.jit_buffer_size=128M`
- `opcache.jit` (CRTO)
- Lots of other knobs in `opcache.jit_*` settings
- Deep dive into JIT configuration
  - [https://www.slideshare.net/nikita\\_ppv/justintime-compiler-in-php-8](https://www.slideshare.net/nikita_ppv/justintime-compiler-in-php-8)

## o (optimization level)

- 0: No JIT.
- 1: Minimal JIT (call standard VM handlers).
- 2: Inline VM handlers.
- 3: Use type inference.
- 4: Use call graph.
- 5: Optimize whole script.

## T (trigger)

- 0: Compile all functions on script load.
- 1: Compile functions on first execution.
- 2: Profile functions on first request and compile the hottest functions afterwards.
- 3: Profile on the fly and compile hot functions.
- 4: Currently unused.
- 5: Use tracing JIT. Profile on the fly and compile traces for hot code segments.

## R (register allocation)

- 0: Don't perform register allocation.
- 1: Perform block-local register allocation.
- 2: Perform global register allocation.

## c (CPU-specific optimization flags)

- 0: Disable CPU-specific optimization.
- 1: Enable use of AVX, if the CPU supports it.

# PHP 8.1

Imagine tomorro\_



# PHP 8.1



- ETA nov/2021
  - No timetable yet <https://wiki.php.net/todo>
  - RM selection in april <https://externals.io/message/113334#113335>
- RFC's
  - Implemented [https://wiki.php.net/rfc#php\\_81](https://wiki.php.net/rfc#php_81)
  - Draft [https://wiki.php.net/rfc#in\\_draft](https://wiki.php.net/rfc#in_draft)
- Follow
  - <https://php.watch/versions/8.1>
  - <https://phpinternals.news/> (Derick Rethans)
  - PHP internals mailing list / <https://externals.io/>
  - <https://www.reddit.com/r/PHP/> (new moderators!)
- Twitter
  - <https://twitter.com/PHPRFCBot>
  - <https://twitter.com/derickr>
  - [https://twitter.com/nikita\\_ppv](https://twitter.com/nikita_ppv)



# Thanks!

Questions?

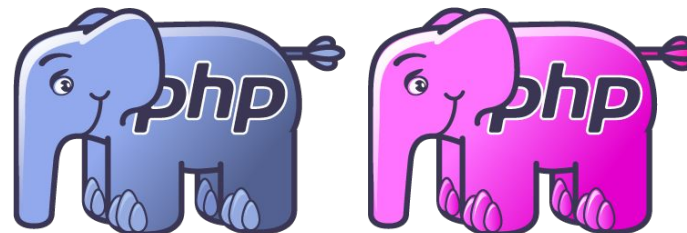
<https://ctors.net/php8.pdf>



# Hi!



- whoami
  - Tom Van Looy - tvlooy - tom@ctors.net
- Join our team at Intracto
  - SRE - Herentals
  - <https://www.intracto.com/nl-be/jobs/linux-systeembeheerder-met-php-ervaring>
- We're a group <https://intracto.group/campuses/>
  - Antwerp, Brussels, Ghent, Herentals
  - Amsterdam, Rotterdam, Utrecht, 's-Hertogenbosch, Eindhoven
- Side project <https://www.secretsantaorganizer.com/>
  - Open-source on it's own <https://github.com/intracto/secretsanta>
  - Sponsoring open-source projects like Xdebug, PhpStan, Debian packaging, ...



**intracto**