

PHP8 new features

reallyli

2020.12.16

Topic

PHP的**发展史**

PHP8的**新特性**

PHP的发展史



PHP1.0(1995)

- ▶ PHP 1.0: Personal HomePage Tool
- ▶ Written in Perl
- ▶ Solves Problems
- ▶ 首创: HTML和脚本融合在一起
- ▶ 开发维护效率大幅提升

```
<html><head><title>Form Example</title></head>
<body><h1>My Example Form</h1>
<form action="form.phtml" method="POST">
Name: <input type="text" name="name">
Age: <input type="text" name="age">
<br><input type="submit">
</form>
<if($name):?>
Hi <?echo $name?>, you are <?echo $age?> years old
<endif?>
</body></html>
```



```
<\oq\y><\ufwJ>
<zenpiti>
HT <zecho \nawew>` \o\ are <zecho \nep> \eis o\j
<tit(\nsame):?>
<\fO>
```

PHP2(1995-1997)

- ▶ PHP 2.0: PHP/FI(Form Interpreter)
- ▶ Written in C
- ▶ 还只是简单的Form解析
- ▶ 没有成熟的语法结构
- ▶ 加入了对mSQL的支持
- ▶ Netscape Navigator 2.0: LiveScript
- ▶ 1996: 50000个域名使用PHP

```
<!--getenv HTTP_USER_AGENT-->
<!--ifsubstr $exec_result Mozilla-->
    Hey, you are using Netscape!<p>
<!--endif-->

<!--sql database select * from table where user='\$username'-->
<!--ifless $numentries 1-->
    Sorry, that record does not exist<p>
<!--endif exit-->
    Welcome <!--$user-->!<p>
    You have <!--$index:0--> credits left in your account.<p>
Not even a language
<!--include /text/footer.html-->
```

<!--JPG/JPEG \F0X\FOOTER.PHP-->

Not even a language

PHP3(1998)

- ▶ PHP: PHP: Hypertext Preprocessor
- ▶ Andi, Zeev重写了Parser
- ▶ 终于是一门语言了(较完备的语法结构)
- ▶ 最关键的: 弱类型, 可扩展的语言



[PHP3] PHP 3.0 Release Announcement

 Rasmus Lerdorf Jun 7, 1998 at 1:03 am

PHP 3.0 is finally here! After 10 months of development PHP Version 3.0 is ready for production environments. Coupled with the release of the final version of Apache-1.3.0 today, the combination provides an unbeatable web hosting platform suitable for both small and large web sites.

PHP 3.0 is the latest evolution of the PHP/FI 2.0 language and it offers some dramatic enhancements. It is more than twice as fast, it uses less memory, it has a more consistent language implementation and a much wider set of expressions and language features than PHP/FI 2.0. An included conversion program will help you migrate your PHP2 scripts to PHP3.

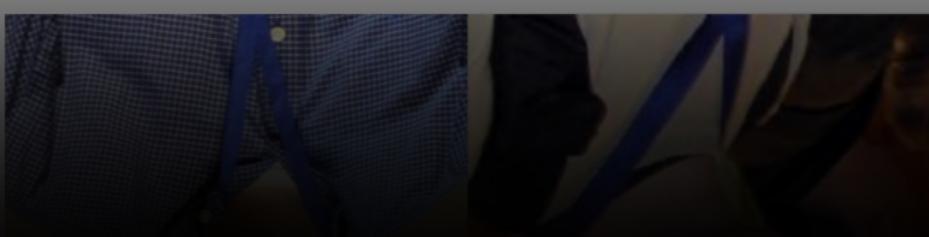
There are too many changes and enhancements to list each one here, but some of the key points are:

- * Persistent database connections for selected databases
- * New language constructs such as for loops and multi-dimensional arrays
- * A well-defined C API which lets you extend PHP yourself
- * Dynamically loadable modules
- * Run-time configuration of most PHP options

For a more detailed description of the differences between PHP2 and PHP3, see <http://www.php.net/changes.php3>

The people behind PHP3 are:

Zeev Suraski and Andi Gutmans who spearheaded the new and improved parser which is the heart and soul of PHP. Zeev also rewrote many of the database modules and implemented persistent connections.



Andi Gutmans and Zeev Suraski are the key figures behind PHP3. They spearheaded the development of the new parser, which is the core of PHP. They also rewrote many of the database modules and implemented persistent connections.

The people behind PHP3 are:

See <http://www.php.net/changes.php3>

PHP4(2000)

- ▶ Zend Engine 1.0
 - ▶ 基本的OO支持
 - ▶ 会话支持
 - ▶ 性能提升
 - ▶ 社区快速发展

PEAR

Main | Support | Documentation | Packages | Package Proposals | Developers

Home | News | Quality Assurance | The PEAR Group | Mirrors

PEAR - PHP Extension and Application Repository

» What is it?

PEAR is a framework and distribution system for PHP.

Sounds good? Perhaps you might be interested in the following notes:

User Contributed Notes [12 notes]

chernyshevsky at hotmail dot com 11 years ago

The easiest way to determine the character count of a UTF8 string is to pass the text through `utf8_decode()` first:

```
<?php
$length = strlen(utf8_decode($s));
?>
```

`utf8_decode()` converts characters that are not in ISO-8859-1 to '?', which, for the purpose of counting, is quite alright.

vcardillo at gmail dot com 3 years ago

I would like to demonstrate that you need more than just this function in order to truly test for an empty string. The reason being that `<?php strlen(null); ?>` will return 0. So how do you know if the value was null, or truly an empty string?

PHP4(2002)

- ▶ Yahoo! 从YScript迁移到了PHP
- ▶ Rasmus Lerdorf 也入职Yahoo!



[Home](#) → [Developer](#) → [Yahoo Goes PHP in Open Source Embrace](#)

Yahoo Goes PHP in Open Source Embrace

By [Ryan Naraine](#) | October 30, 2002
Page 1 of 1



Mega portal Yahoo has switched to the PHP scripting language for its back-end programming, a decision that's sure to bring smiles to the faces of open-source advocates.

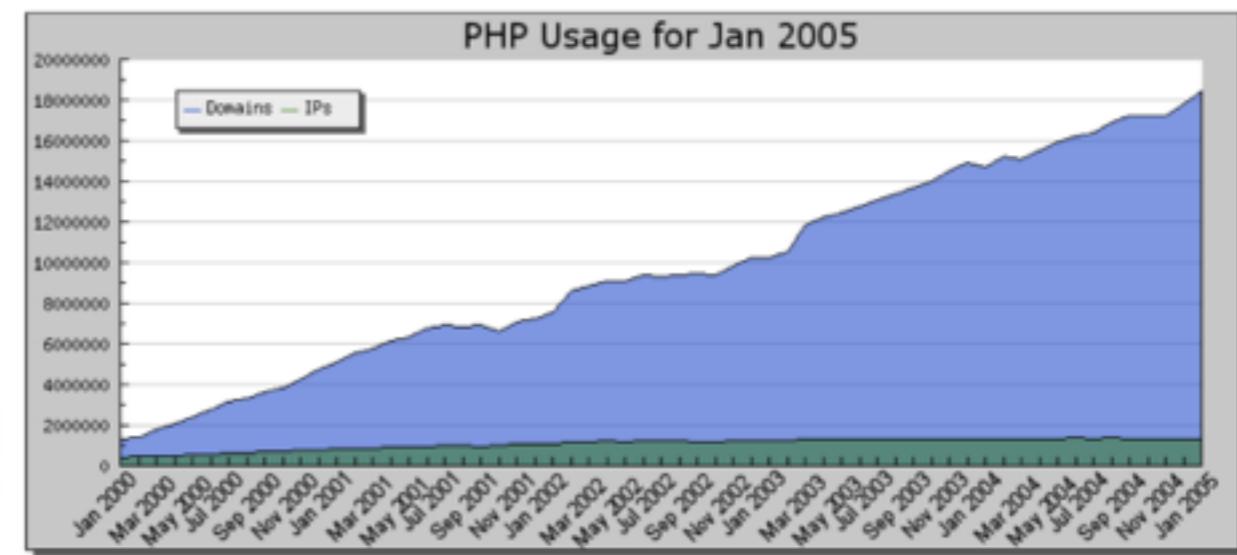
The Sunnyvale, Calif.-based Yahoo is undergoing a switch from a proprietary system written in C/C++ to PHP for its backend scripting, according to a company engineer in a [case study](#) presented this week at the PHPCon 2002 conference.

Michael Radwin, who is co-leading the PHP crusade at Yahoo, told the conference that Yahoo's size and complicate server-side software made the switch to PHP a no-brainer, pointing to the huge savings cost associated with migrating to an open-source platform.

He said Yahoo, which serves more than 1.5 billion page views a day, had already adopted open-source software like Perl, Apache and FreeBSD to run its 74 Web properties, which includes 25 international sites.

PHP5(2004)

- ▶ Zend Engine 2.0
- ▶ 更好的OO支持
- ▶ PDO的引入
- ▶ 性能提升
- ▶ 社区快速成长



Framework?(04-08)

- ▶ Use or Not Use Framework



Rasmus Lerdorf: PHP Frameworks? Think Again.

David Peterson August 29, 2008 +1

[f](#) Facebook [t](#) Twitter

Was this helpful?



This is the first time I have heard [Rasmus Lerdorf](#) speak and it was entertaining to say the least. Refreshing would another way to describe it, I enjoy hearing real opinions and not holding back – Rasmus doesn't hold back.



Just a short background, Rasmus Lerdorf is the creator of PHP and still continues as a core developer to the PHP project.

CodeIgniter



developer to the PHP project
Just a short background, Rasmus Lerdorf is the creator of PHP and still continues as a core

Just a short background, Rasmus Lerdorf is the creator of PHP and still continues as a core

PHP6(2005)

- ▶ Unicodes支持
- ▶ 然而....



Laruence V G
PHP官方开发组核心成员

+ 关注

好吧, 很多人问为啥没有PHP6了, 我来解释下吧: 在很久很久以前, 有一群人, 创建了一个 PHP6的项目, 主要的目的是为PHP引擎增加Unicode支持. 当时开发者们同时维护5和6的开发, 慢慢的大家发现新功能都等着提交给6, 而6因为开发速度慢, 导致很多新特性没法提交, 状态很不理想. 再后来6就没人开发了. 于是.

2014-8-29 13:21 来自微博 weibo.com

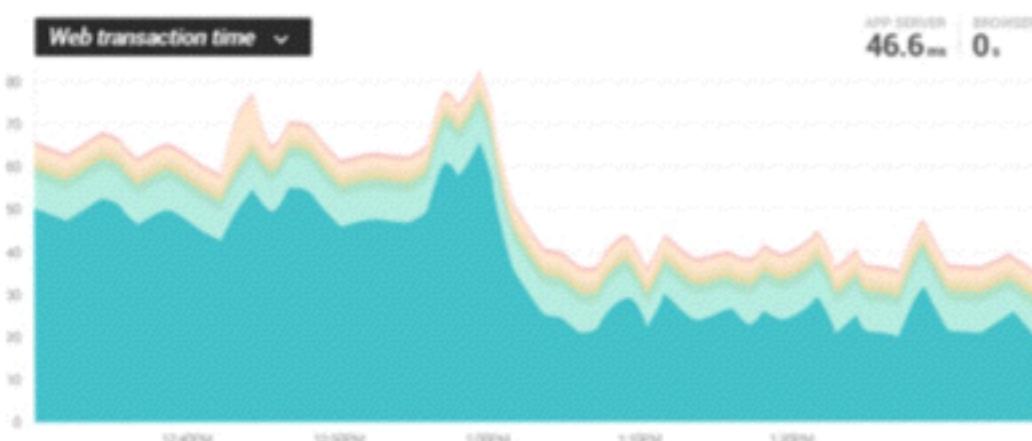
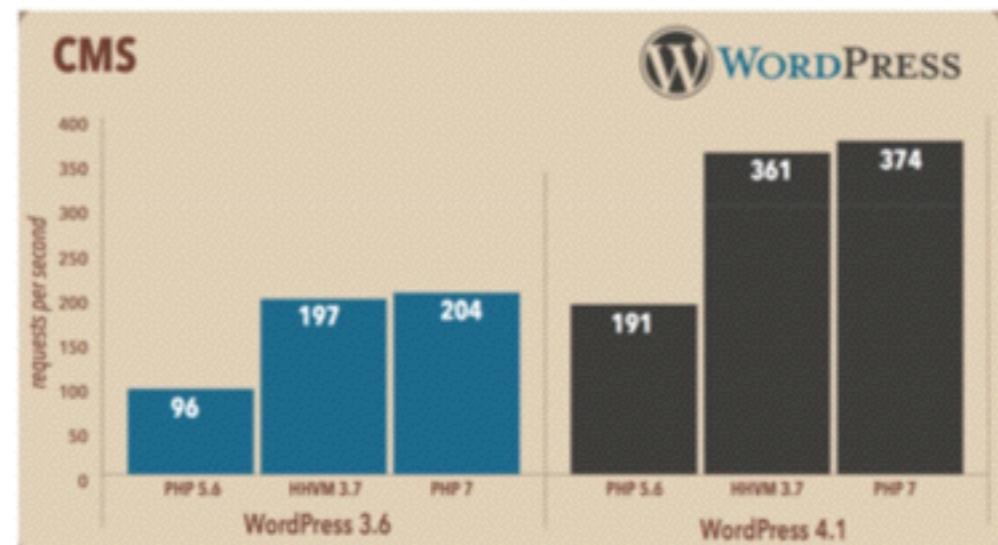
Performance(2010+)

- ▶ HipHop
- ▶ JPHP
- ▶ Zephir
- ▶ Yaf
- ▶ Phalcon

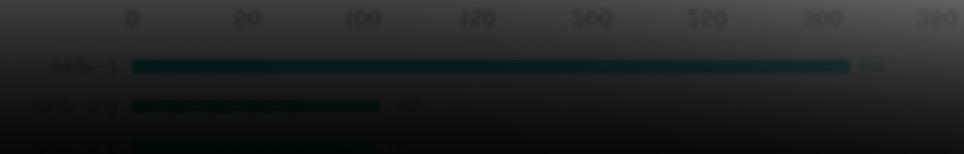
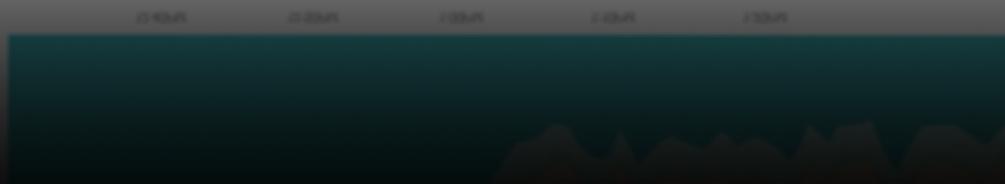
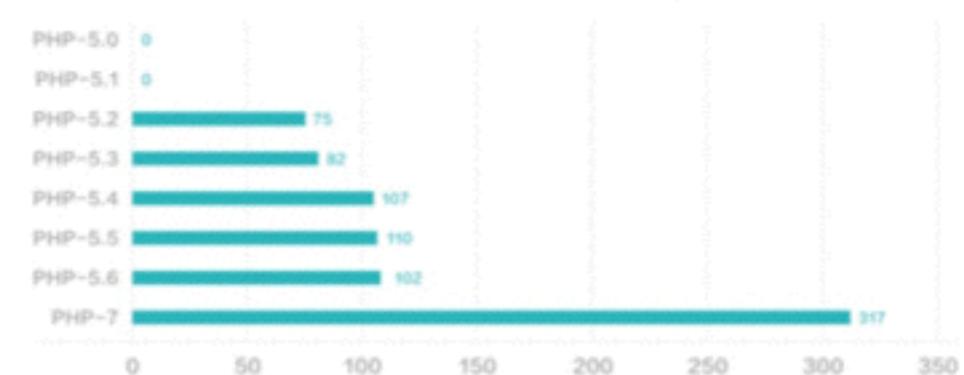


PHP7(2014)

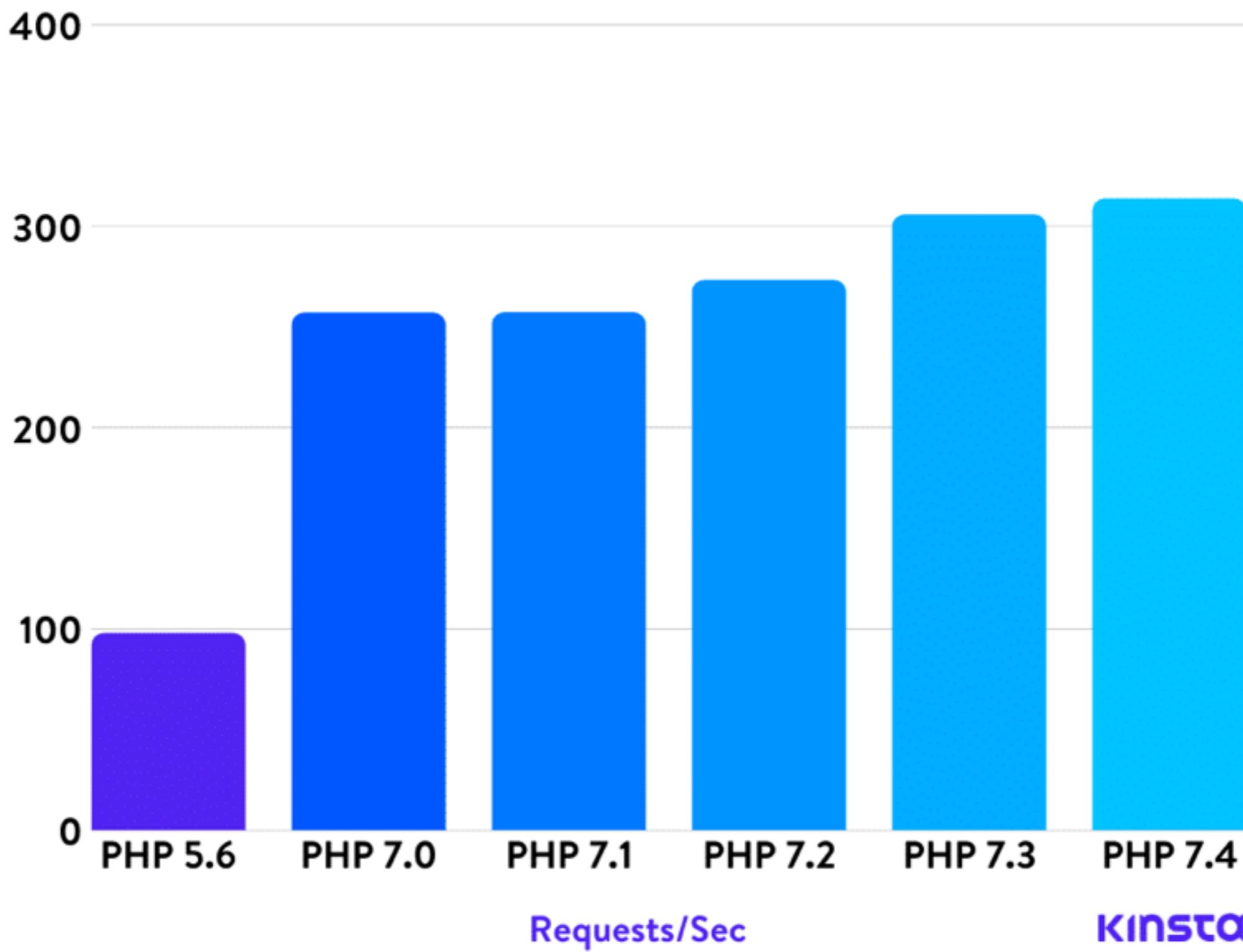
- ▶ Zend Engine 3.0
- ▶ 基于PHP5.5 Opcache JIT项目
- ▶ 最大的一次重构,历时一年多开发
- ▶ PHP最大的性能提升版本



wordpress 3.6 home page qps



WordPress 5.3



Requests/Sec

Kinsta

Requests/Sec

Kinsta

PHP8(2020)

A blue rectangular banner with white text. At the top is the PHP logo with the number 8. Below it is the word "Released!". A paragraph of text follows, and at the bottom is a button with the text "Go update to PHP 8!"

PHP 8

Released!

PHP 8.0 is a major update of the PHP language. It contains many new features and optimizations including named arguments, union types, attributes, constructor property promotion, match expression, nullsafe operator, JIT, and improvements in the type system, error handling, and consistency.

Go update to PHP 8!

PHP8于11月26号发行
从很多方面来看， PHP8都是很
重要的一次突破
加入了很多新的特性，根据语
义化版本规范，有相当多的向
下不兼容的变化

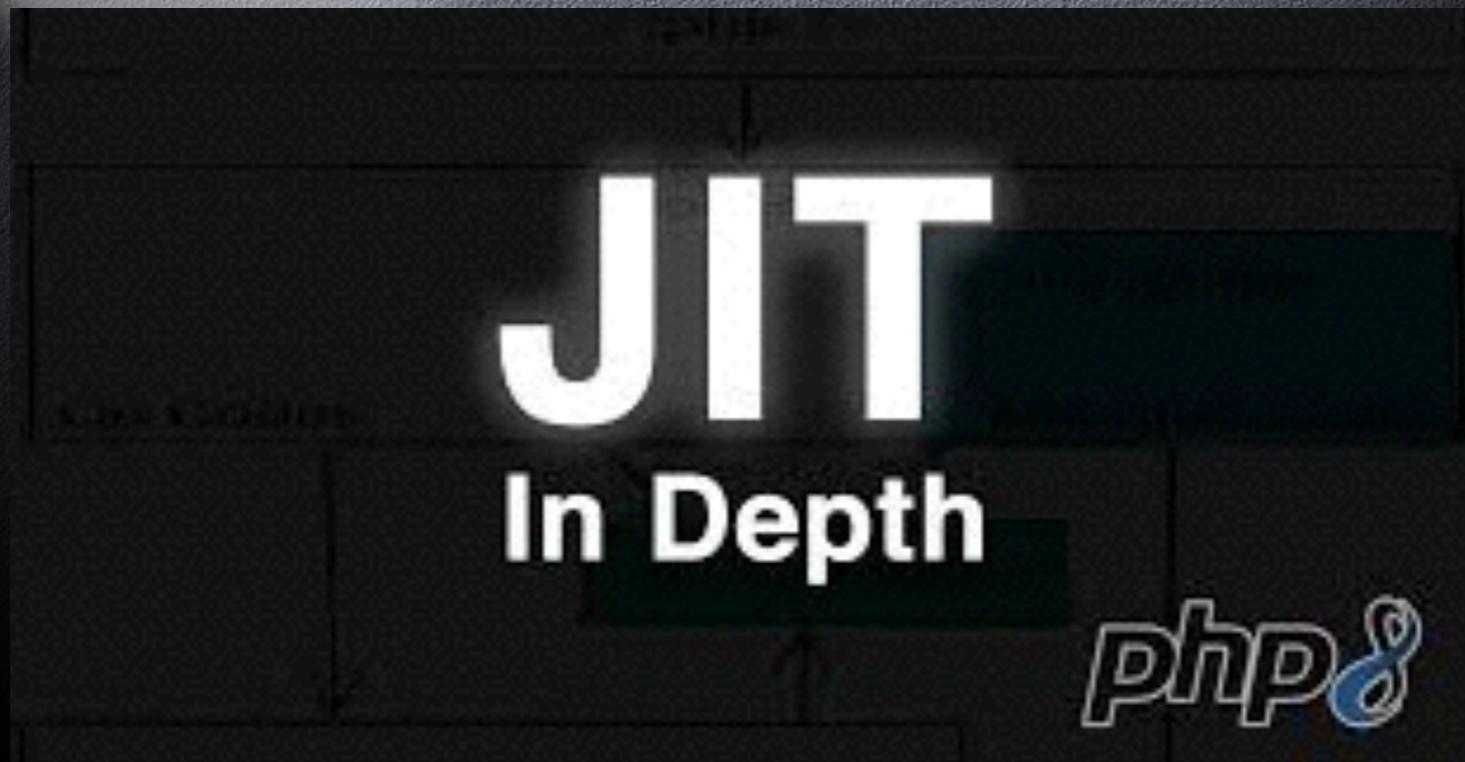
Features

- JIT Compiler
- Attributes
- Union Types
- Mixed Types
- Match Expression
- Nullsafe Operator

...



JIT

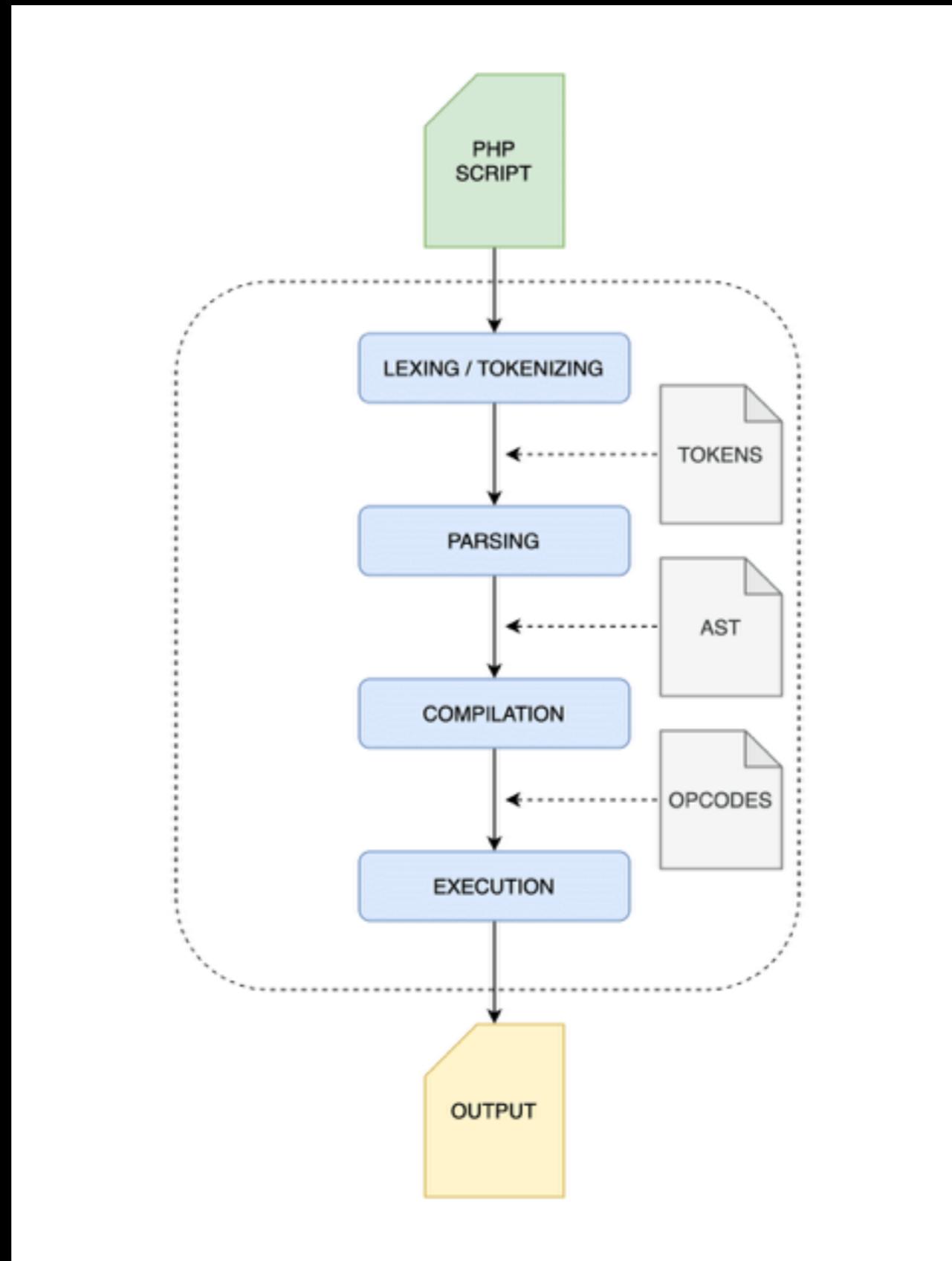


php&

bvb9

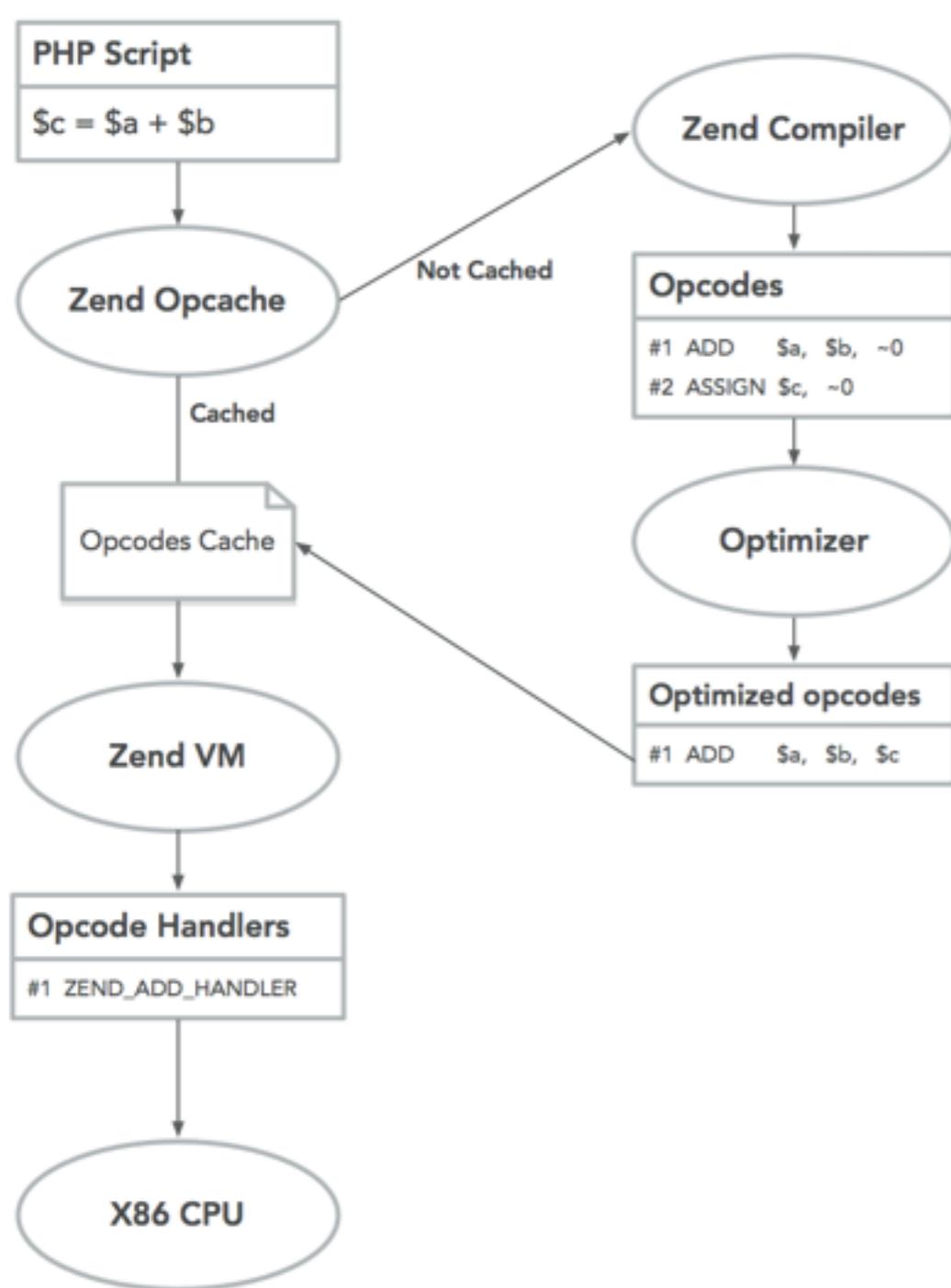
III Debru

让 PHP 代码编译成原生的机器
码，而不是在 Zend 虚拟机上执
行
目标就是提升运行时的性能



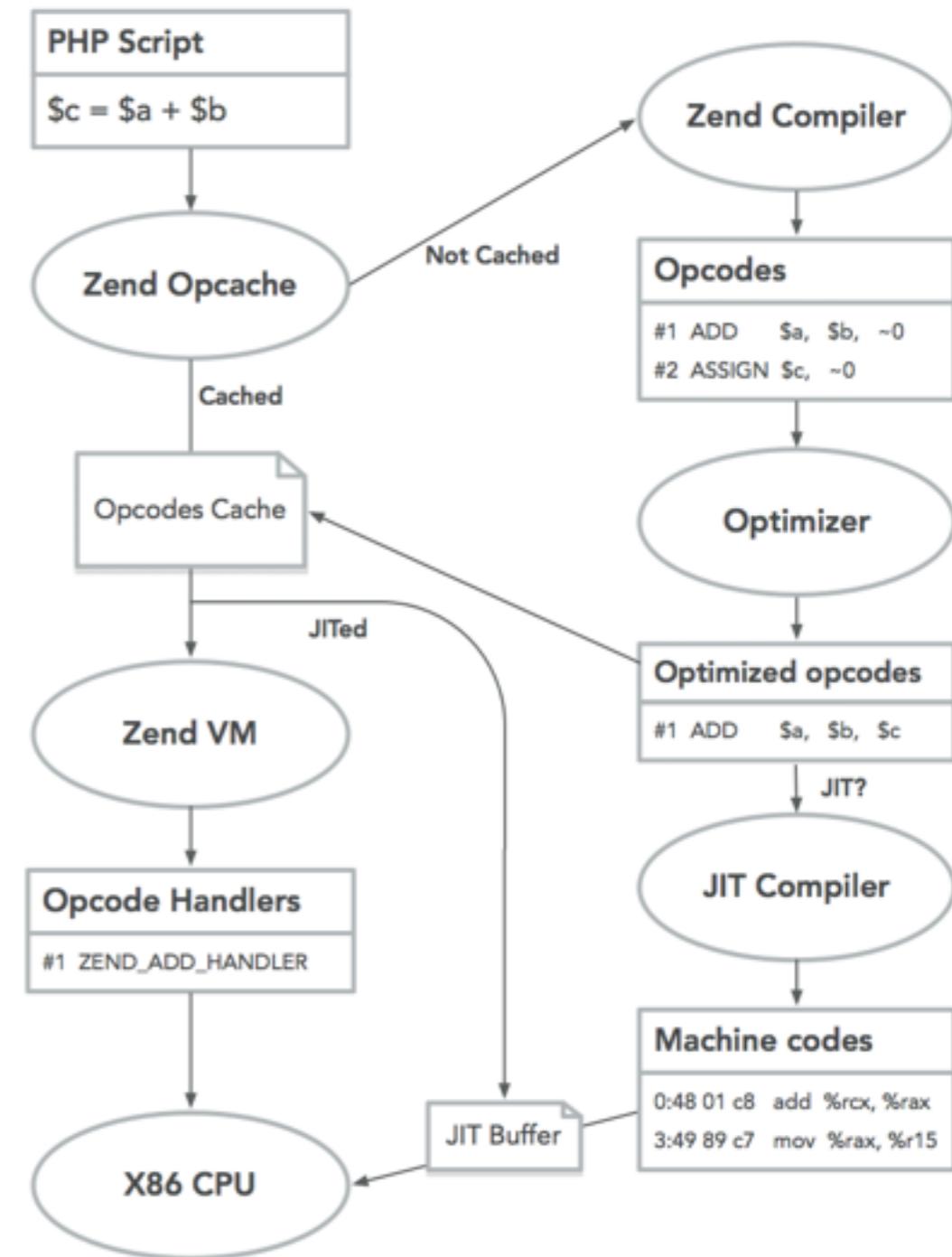
JIT 在 Opcache 优化之后的基础上，结合 Runtime 的信息再次优化，直接生成机器码

PHP8 的 JIT 目前是在 Opcache 之中提供的



PHP With Opcache

PHP 7.0 ын Опсаче



PHP With Opcache Plus JIT

PHP 7.2 ын Опсаче һәм JIT

laruence.com

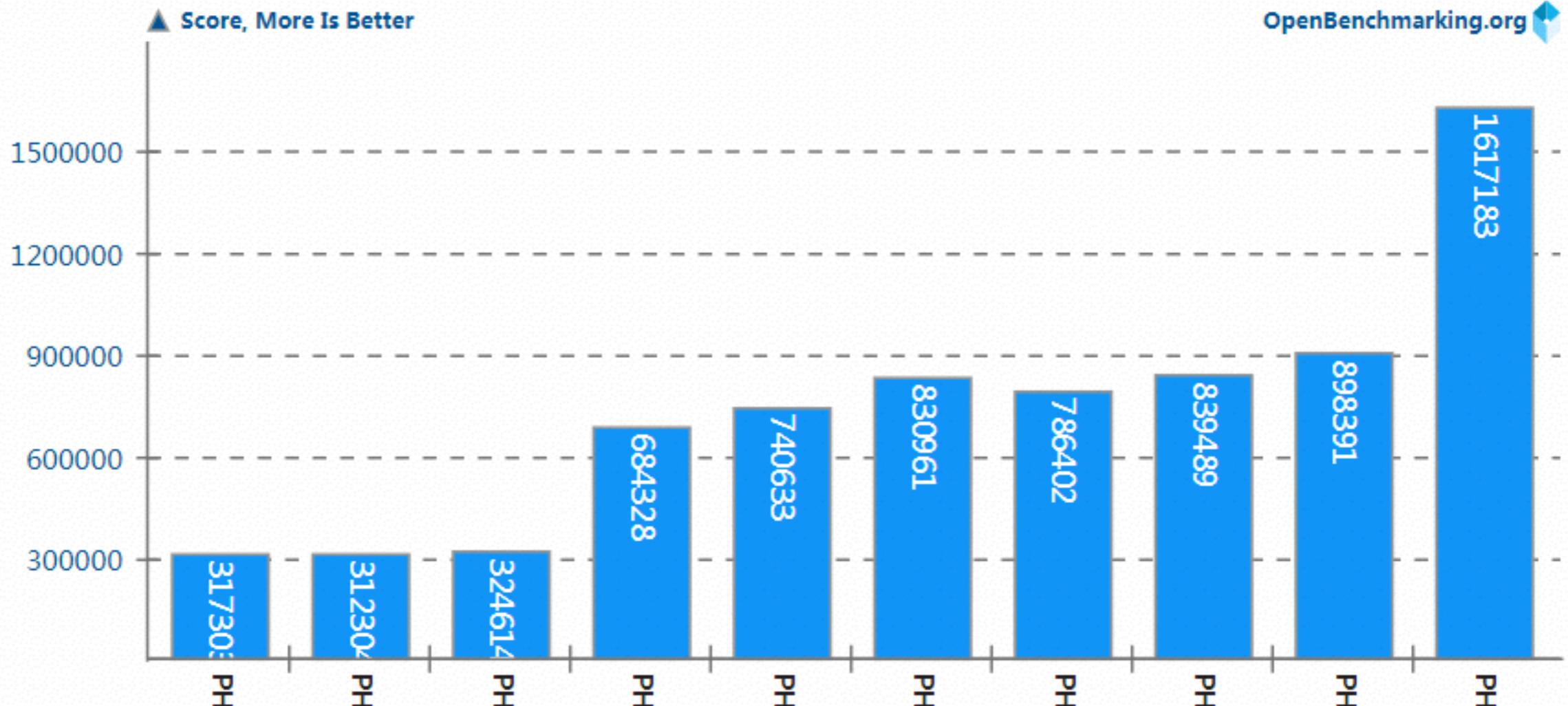
JIT 7.2.0 ын Опсаче

PHPBench v0.8.1

PHP Benchmark Suite



OpenBenchmarking.org



Attributes



新的注解 (Attributes) 取代了
现有的 PHP doc 的注释
(Annotation)

Use **structured metadata** with
PHP's native syntax.

```
// 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) { /* ... */ }
}

// Photo by Muhammad Putra Arienda on Unsplash
```



Named Arguments



允许基于参数名称传递参数到函数，而不是参数所在的位置
函数参数就可以自解释并且与顺序无关，并且允许跳过默认值



```
declare(strict_types=1);

// Using positional arguments
array_fill(0, 100, 50);

// Using named arguments
array_fill(value: 50, count: 100, start_key: 0);

htmlspecialchars($string, double_encode: false);
// Same as
htmlspecialchars($string, ENT_COMPAT | ENT_HTML401, 'UTF-8', false);
```

```
declare(strict_types=1);

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

new Point(x: 2.1, y: 3.3, z: 4.2);
```



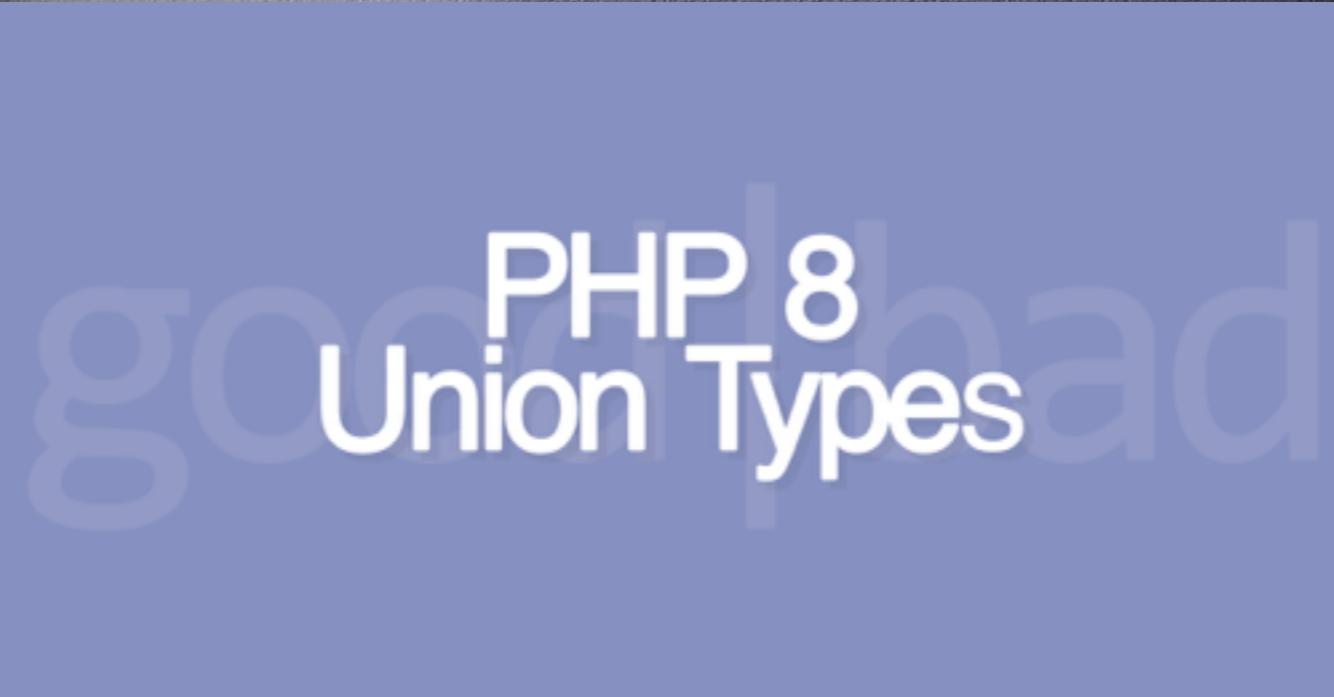
```
declare(strict_types=1);

function acceptsAnything(...$args)
{
    var_dump($args);
}
acceptsAnything(1, 2, x: 3, y: 4);
```

T O Y



Union Types



PHP 8
Union Types

允许一个变量拥有多个类型的
值，而不是一个
使用联合类型声明来代替类型
组合的PHPDoc注释

```
declare(strict_types=1);

// PHP 7
class Number {
    /** @var int|float */
    private $number;

    /**
     * @param float|int $number
     */
    public function __construct($number) {
        $this->number = $number;
    }
}

new Number('NaN'); // Ok

// PHP 8
class Number {
    public function __construct(
        private int|float $number
    ) {}
}

new Number('NaN'); // TypeError
```



```
<?php declare(strict_types=1);

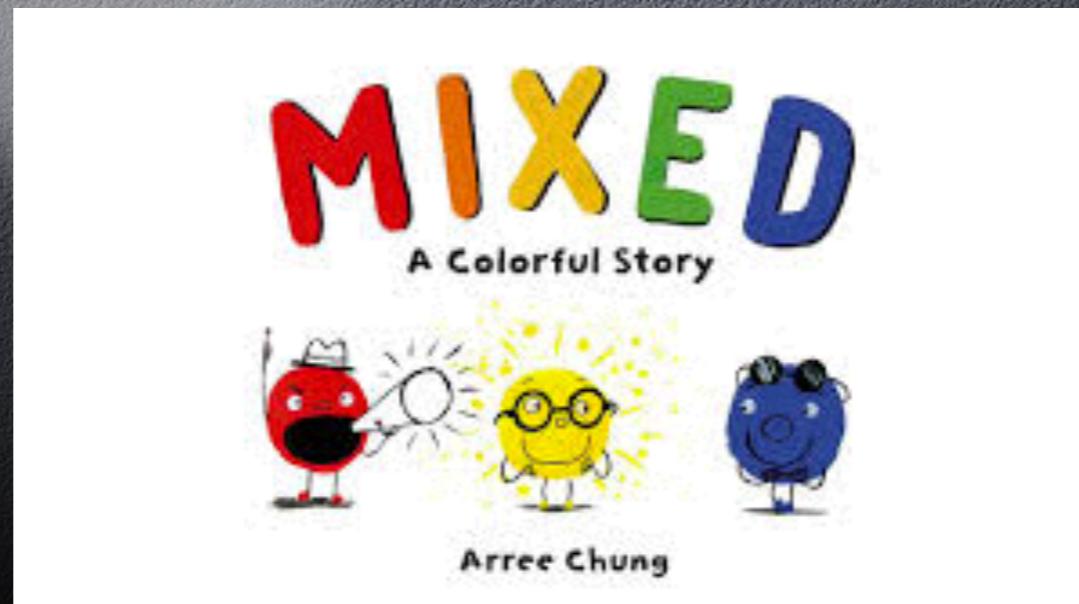
function test(int|float|bool $arg) {
    var_dump($arg);
}

test(45);          // int(45)
test(45.8);        // float(45.8)
test("45");        // TypeError
test("45.8");      // TypeError
test("");           // TypeError
test("X");          // TypeError
test([]);           // TypeError
```

```
test([]);           // TypeError
test("X");          // TypeError
```



Mixed Types



Arree Chung

VILLAGE COUNCIL

等价于 array|bool|callable|int|
float|null|object|resource|string
Mixed 类型与完全指定空类型
非常相似， 它允许明确地说这
个参数确实接受任意类型



```
declare(strict_types=1);

function debug_function(mixed ...$data)
{
    var_dump($data);
}

debug_function(1, 'string', []);
```



Static Return Types

Static Typing: Why Do It?



使用 static 关键字标识某个方法
返回该方法当前所属的类
通常为了表达后期静态绑定语
义



```
declare(strict_types=1);

class Test {
    public function doWhatever(): static {
        // Do whatever.
        return $this;
    }
}
```

I O X



Match Expression



Match 是一个表达式，表示其结果可以存储在变量中或返回
Match 分支仅支持单行表达式，不需要中断，声明匹配进行严格比较

```
● ● ●  
  
declare(strict_types=1);  
  
// 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",  
};  
//> This is what I expected
```



Nullsafe Operator



使用带有新的 `nullsafe` 运算符的
调用链来代替空检查条件

当对链中一个元素的求值失败
时，整个链的执行将中止，并
且整个链的求值为空

```
● ● ●  
declare(strict_types=1);  
  
// 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;
```

Constructor Promotion

Constructor in PHP

function __construct()

//constructor_function __



function __construct()

//constructor_function

www.educba.com

www.educba.com

function __construct()

//constructor_function

更少的样板代码来定义和初始化属性

加强版构造函数提供了方便的语法来编写完全相同的代码，
自动生成对应的属性

```
declare(strict_types=1);

// 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,
    ) {}
}
```





str_contains etc.

**Check if String Contains
Substring using
str_contains in PHP 8.0**

- str_contains
- str_starts_with
- str_ends_with



```
declare(strict_types=1);

function str_contains(
    string $haystack, string $needle
): bool {}

function str_starts_with(
    string $haystack, string $needle
): bool {}

function str_ends_with(
    string $haystack, string $needle
): bool {}

// Instead of doing this
if (strpos('string with lots of words', 'words') !== false) { /* ... */ }
// You can now do this
if (str_contains('string with lots of words', 'words')) { /* ... */ }

str_starts_with('haystack', 'hay'); // true
str_ends_with('haystack', 'stack'); // true
```

Throw expression

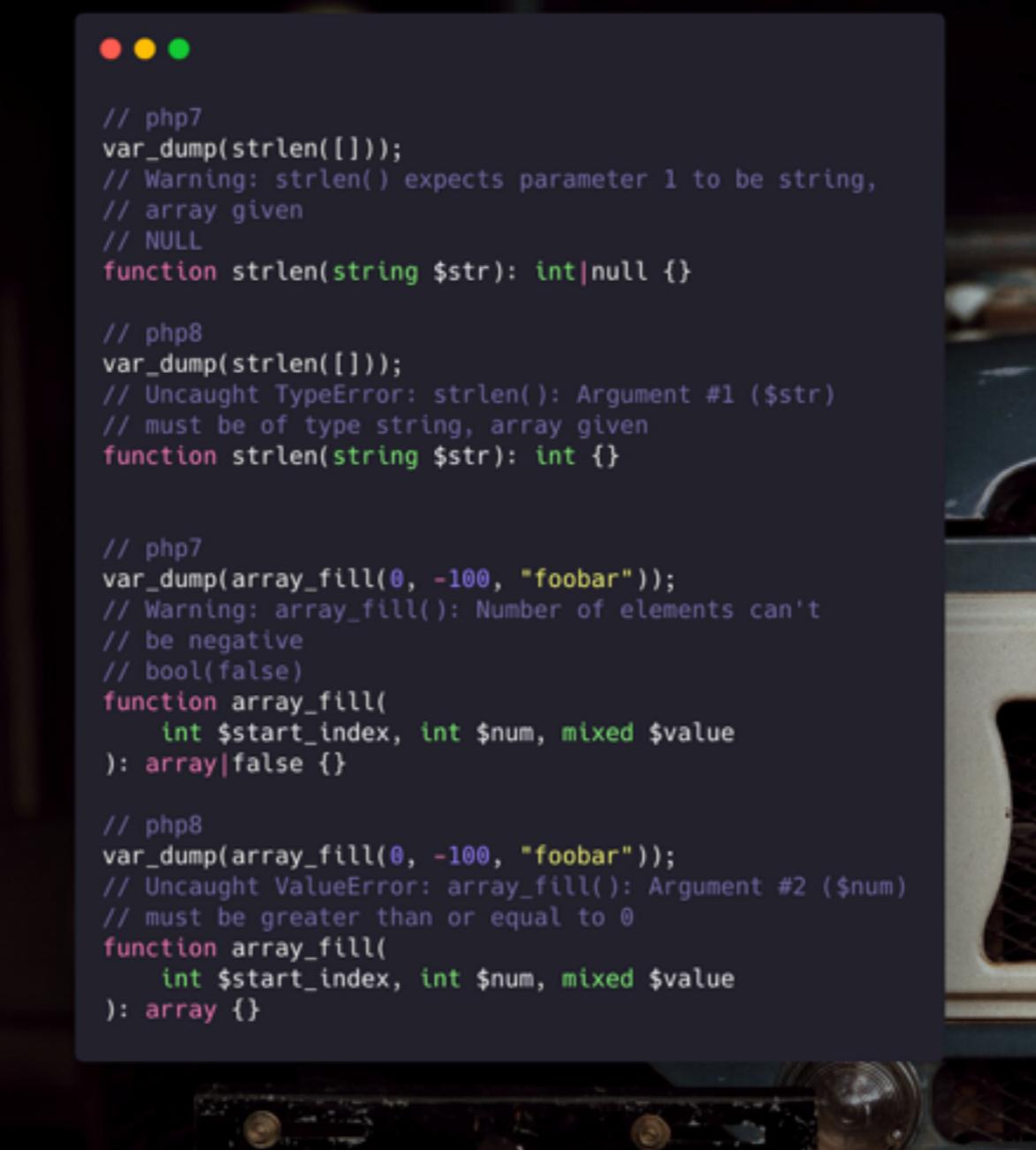


```
$triggerError = fn () => throw new MyError();  
  
$foo = $bar['offset'] ?? throw new OffsetDoesNotExist('offset');
```

Non-capturing catches

```
//so instead of this:  
  
try {  
    // Something goes wrong  
} catch (MySpecialException $exception) {  
    Log::error("Something went wrong");  
}  
  
//You can now do this:  
  
try {  
    // Something goes wrong  
} catch (MySpecialException) {  
    Log::error("Something went wrong");  
}
```

Type I Value Error



```
// php7
var_dump(strlen([]));
// Warning: strlen() expects parameter 1 to be string,
// array given
// NULL
function strlen(string $str): int|null {}

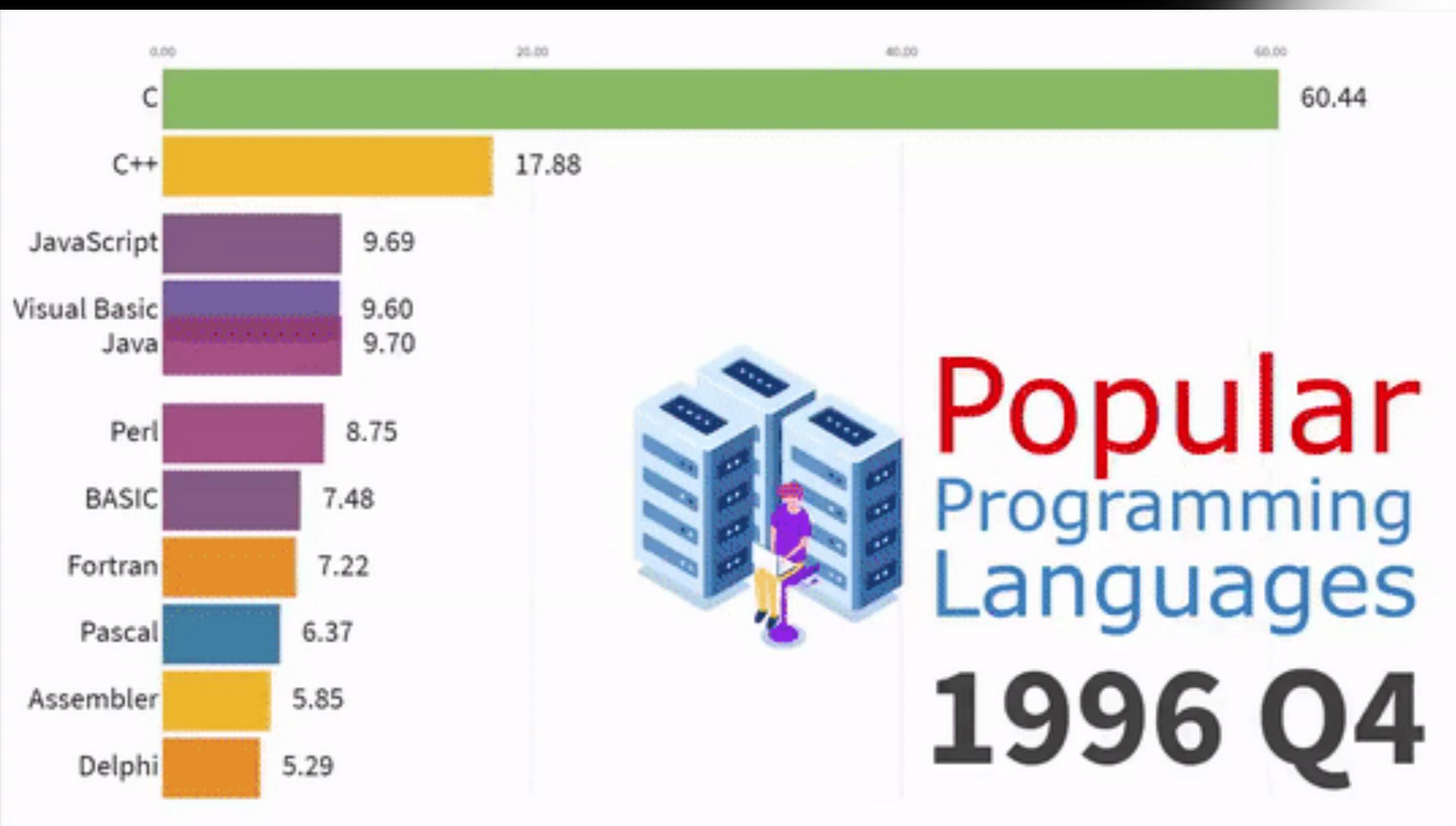
// php8
var_dump(strlen([]));
// Uncaught TypeError: strlen(): Argument #1 ($str)
// must be of type string, array given
function strlen(string $str): int {}

// php7
var_dump(array_fill(0, -100, "foobar"));
// Warning: array_fill(): Number of elements can't
// be negative
// bool(false)
function array_fill(
    int $start_index, int $num, mixed $value
): array|false {}

// php8
var_dump(array_fill(0, -100, "foobar"));
// Uncaught ValueError: array_fill(): Argument #2 ($num)
// must be greater than or equal to 0
function array_fill(
    int $start_index, int $num, mixed $value
): array {}
```

And much much more...

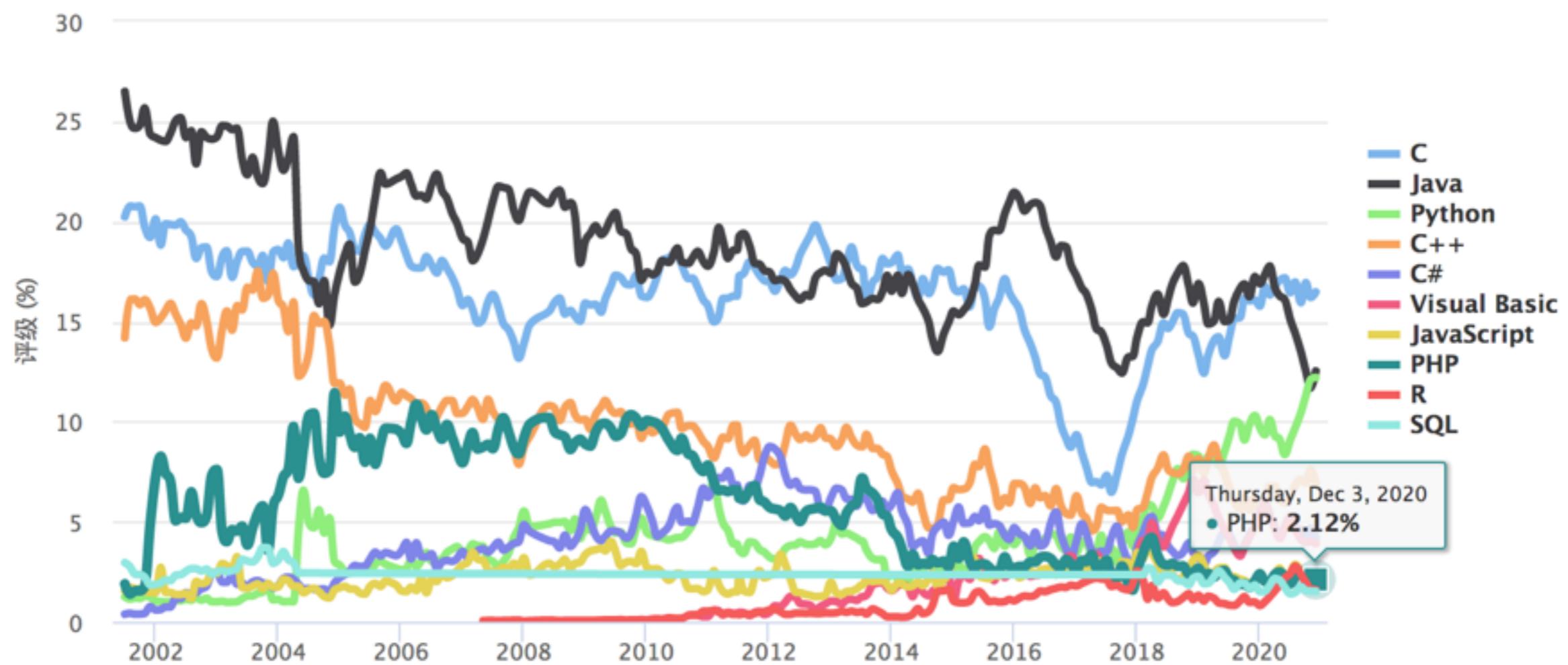
Trending



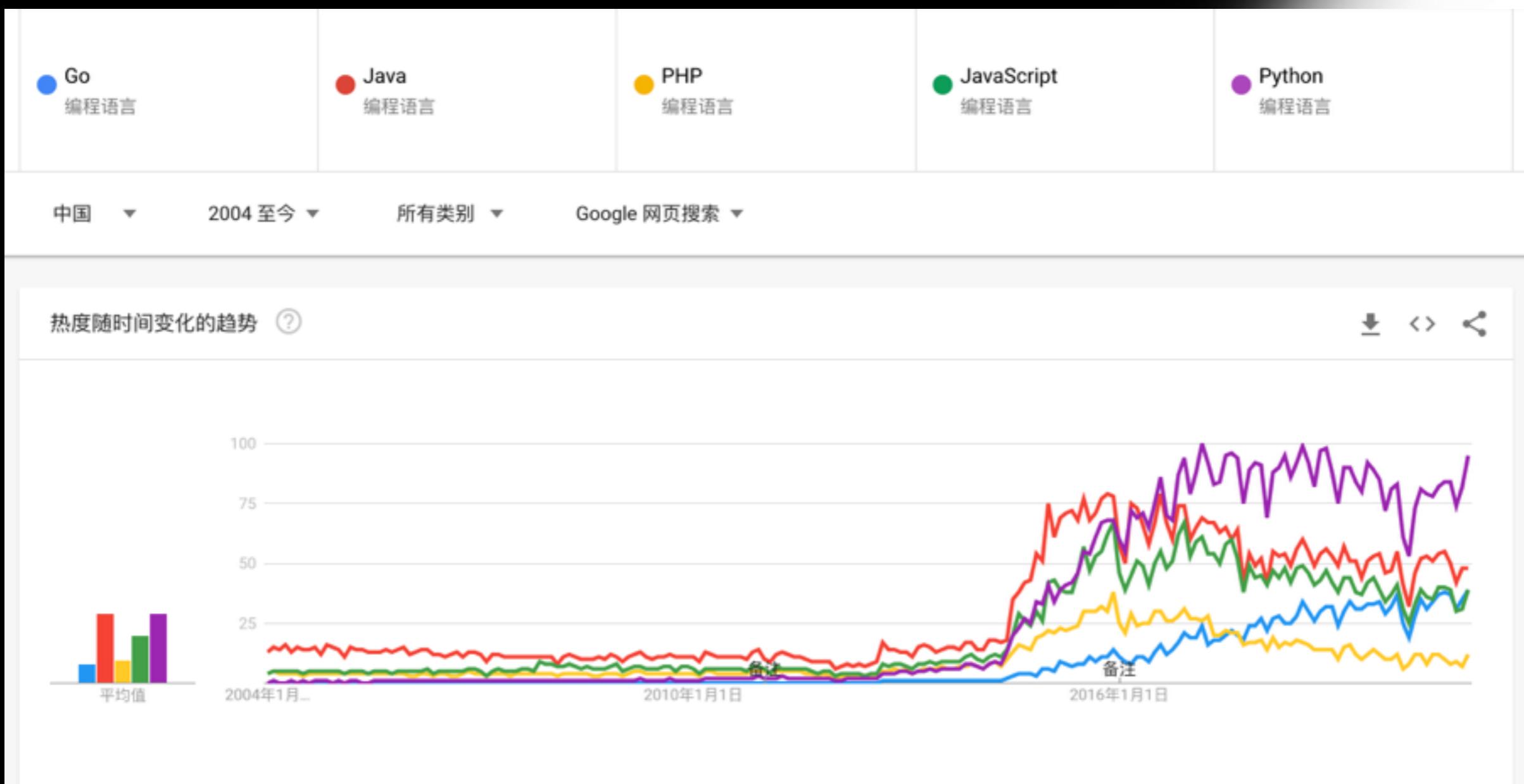
TIOBE

TOP 10 编程语言的走势图

Source: www.tiobe.com



Google Trends



THANKS

Q & A