禅道

2022-2023年度

PHP综合现状分析报告

禅道项目管理软件团队 出品

www.zentao.net

PHP2022-2023年度相关数据

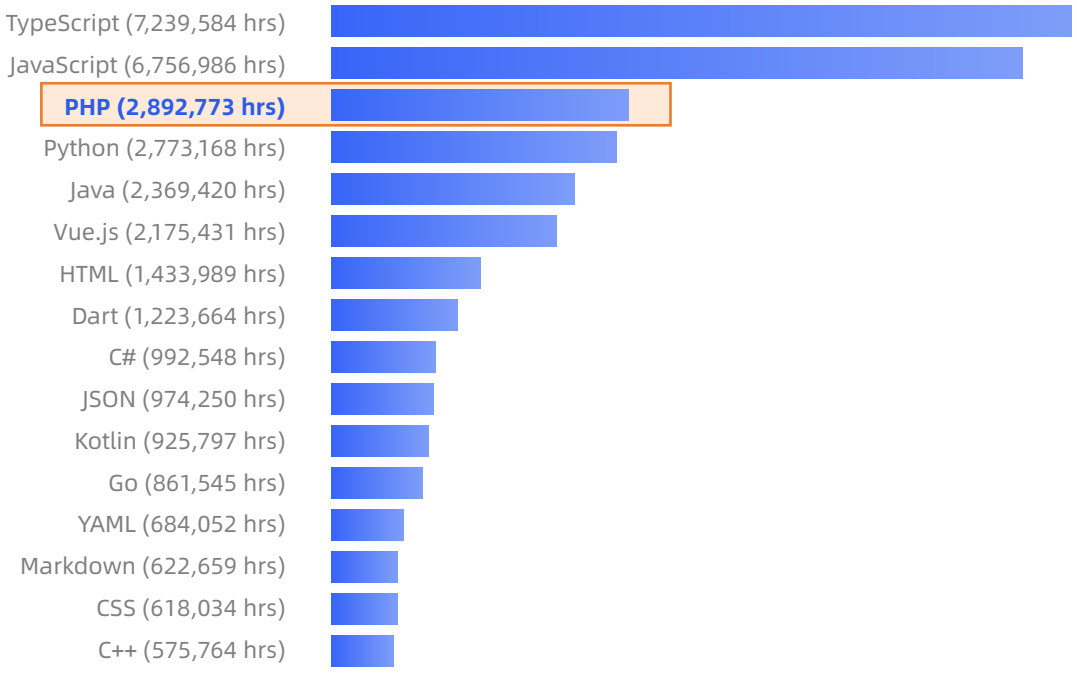
国内，PHP 这门有着近三十年历史的编程语言曾几度被唱衰，“PHP将亡”的言论伴随着“PHP 是最好的语言”的黑色幽默，多年来屡见不鲜。

那 PHP 的现状究竟如何？

为拨开 PHP 语言周遭的层层迷雾，禅道项目管理软件团队从近两年各方发布的 PHP 应用现状报告、2023年 PHP 最新动态以及对 PHP 社区用户的深入访谈等维度出发，整理编辑出了一份全面、真实、客观的《2022-2023年度 PHP 综合现状分析报告》。我们希望这份报告能帮助行业从业者更好地理解 PHP 语言的现状与趋势，为相关决策提供参考依据。

WakaTime 2022年度使用数据

编程时间管理工具 WakaTime 能够统计用户在不同语言、项目下的代码编写时长，包括使用的编辑器、编程语言、操作系统等数据。WakaTime 发布的年度使用数据显示，在2022年使用较为广泛的编程语言中，PHP 以较高的比例位居第三。



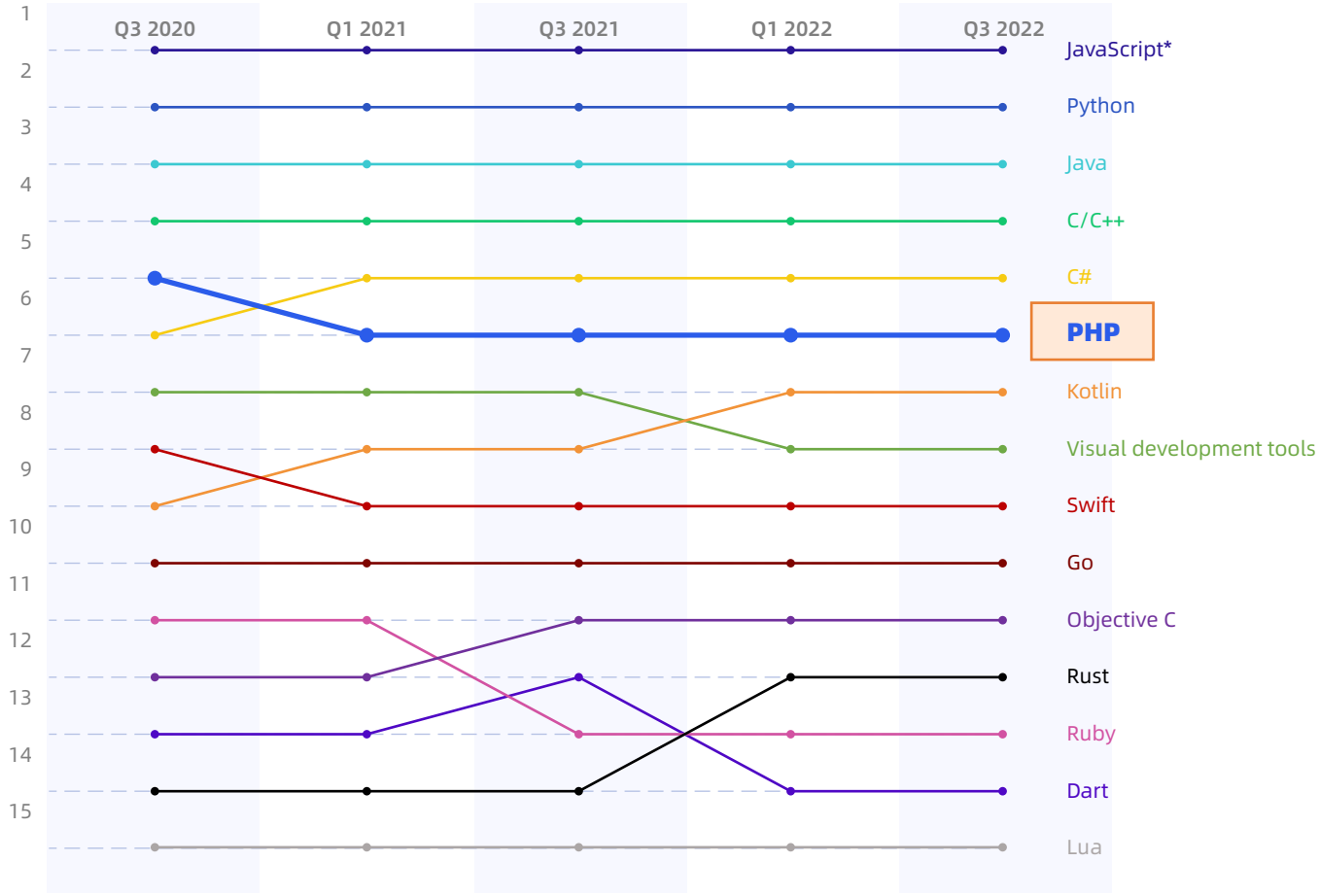
来源：https://wakatime.com/a-look-back-at-2022

SlashData开发者报告（第23版）

开发者调查分析公司 SlashData 发布了2022年开发者报告（第23版）。报告显示，在过去24个月内 PHP 在编程语言的社区排名相对稳定在第五、六名。

编程语言社区排名

(Q3 2020 n=17,241 | Q1 2021 n=17,801 | Q3 2021 n=19,319 | Q1 2022 n=20,041 | Q3 2022 n=26,183)



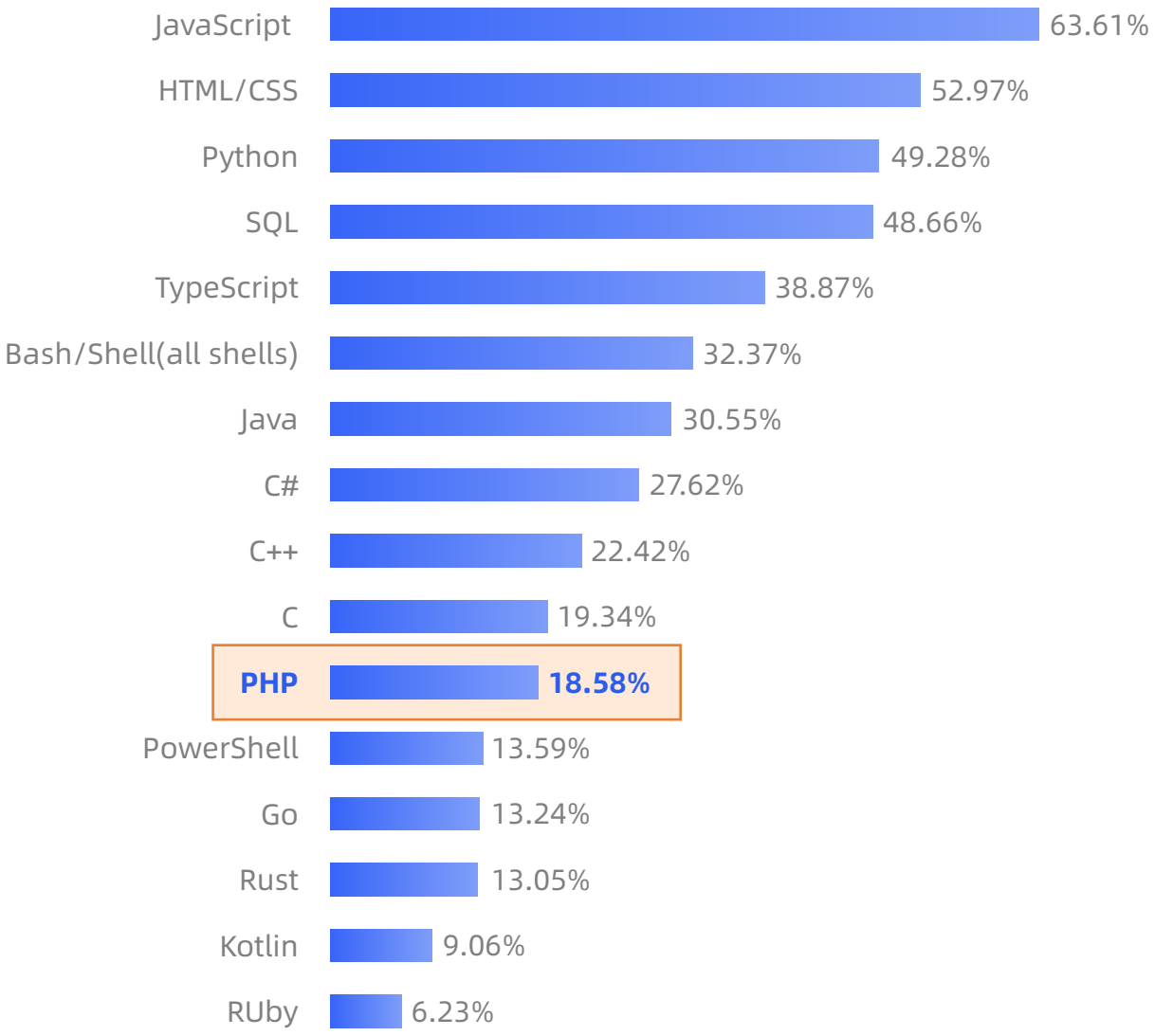
*JavaScript社区包括CoffeeScript与TypeScript

来源：https://slashdata-website-cms.s3.amazonaws.com/sample_reports/dsIe6JlZge_KsHWt.pdf

Stack Overflow 2023年开发者调查报告

面向程序员的问答网站 Stack Overflow 发布了2023年开发者调查报告，据称已有9万余名开发者参与了此次调查。

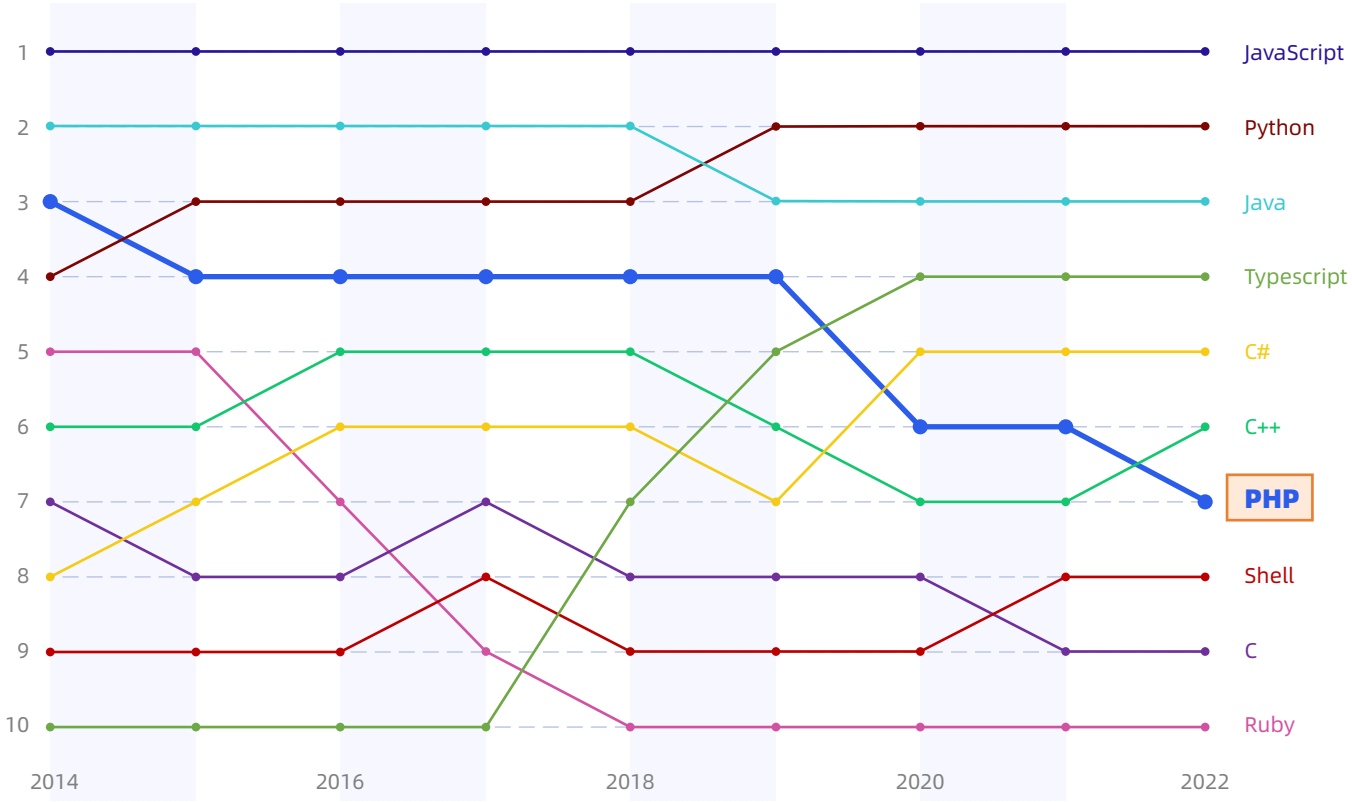
报告包含了受访开发者画像，以及关于开发技术、AI、职业、社区等方面的内容。此外，报告还总结了今年最流行的技术，其中 PHP 占比18.58%。



来源：https://survey.stackoverflow.co/2023/?utm_source=so-owned&utm_medium=blog&utm_campaign=dev-survey-results-2023&utm_content=survey-results

GitHub 2022年度Octoverse报告

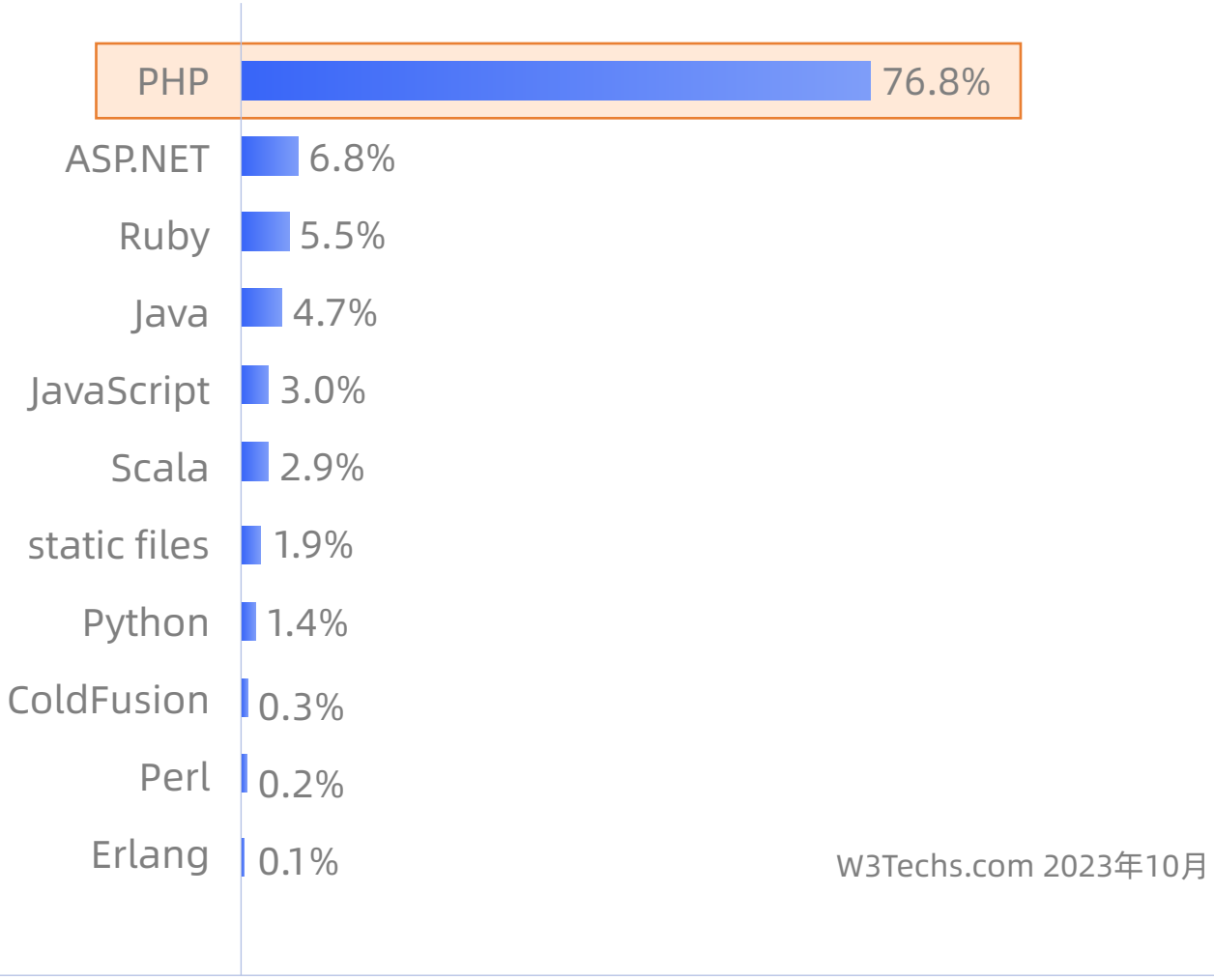
全球知名代码托管平台 GitHub 发布2022年度 Octoverse 报告，报告显示 PHP 使用量略有下降，但仍处于第七名。



来源：https://octoverse.github.com/2022/top-programming-languages

W3 Techs全球网站使用的服务器端编程语言分析

W3 Techs 发布的全球网站使用的服务器端编程语言分析报告显示，截至2023年10月，使用最为广泛的服务器端编程语言中 PHP 以76.8%的比例位居第一。



全球网站使用的服务器端编程语言分析
注：一个网站可能使用多种服务器端编程语言

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

TIOBE统计数据（2023年9月）

TIOBE Index 编程社区指数是衡量编程语言流行度的一个指标，基于全球熟练工程师的数量、课程和第三方供应商的数据生成，每月一更新。2023年9月份数据显示，PHP 在排行榜中位居第八，较为稳定。

2023年9月	2022年9月	对比上月	编程语言	流行度	变化率
1	1		Python	14.16%	-1.58%
2	2		C	11.27%	-2.70%
3	4	▲	C++	10.65%	+0.90%
4	3	▼	Java	9.49%	-2.23%
5	5		C#	7.31%	+2.42%
6	7	▲	JavaScript	3.30%	+0.48%
7	6	▼	Visual Basic	2.22%	-2.18%
8	10	▲	PHP	1.55%	-0.13%
9	8		Assembly language	1.53%	-0.96%
10	9	▼	SQL	1.44%	-0.57%
11	15	⏫	Fortran	1.28%	+0.26%
12	12		Go	1.19%	+0.03%
13	14	▲	MATLAB	1.19%	+0.13%
14	22	⏫	Scratch	1.08%	+0.51%
15	13	▼	Delphi/Object Pascal	1.02%	-0.07%
16	16		Swift	1.00%	+0.02%
17	26	⏫	Rust	0.97%	+0.47%
18	18		R	0.97%	+0.02%
19	20	▲	Ruby	0.95%	+0.30%
20	34	⏫	Kotlin	0.90%	+0.59%

来源：https://www.tiobe.com/tiobe-index/

PYPL编程语言流行指数

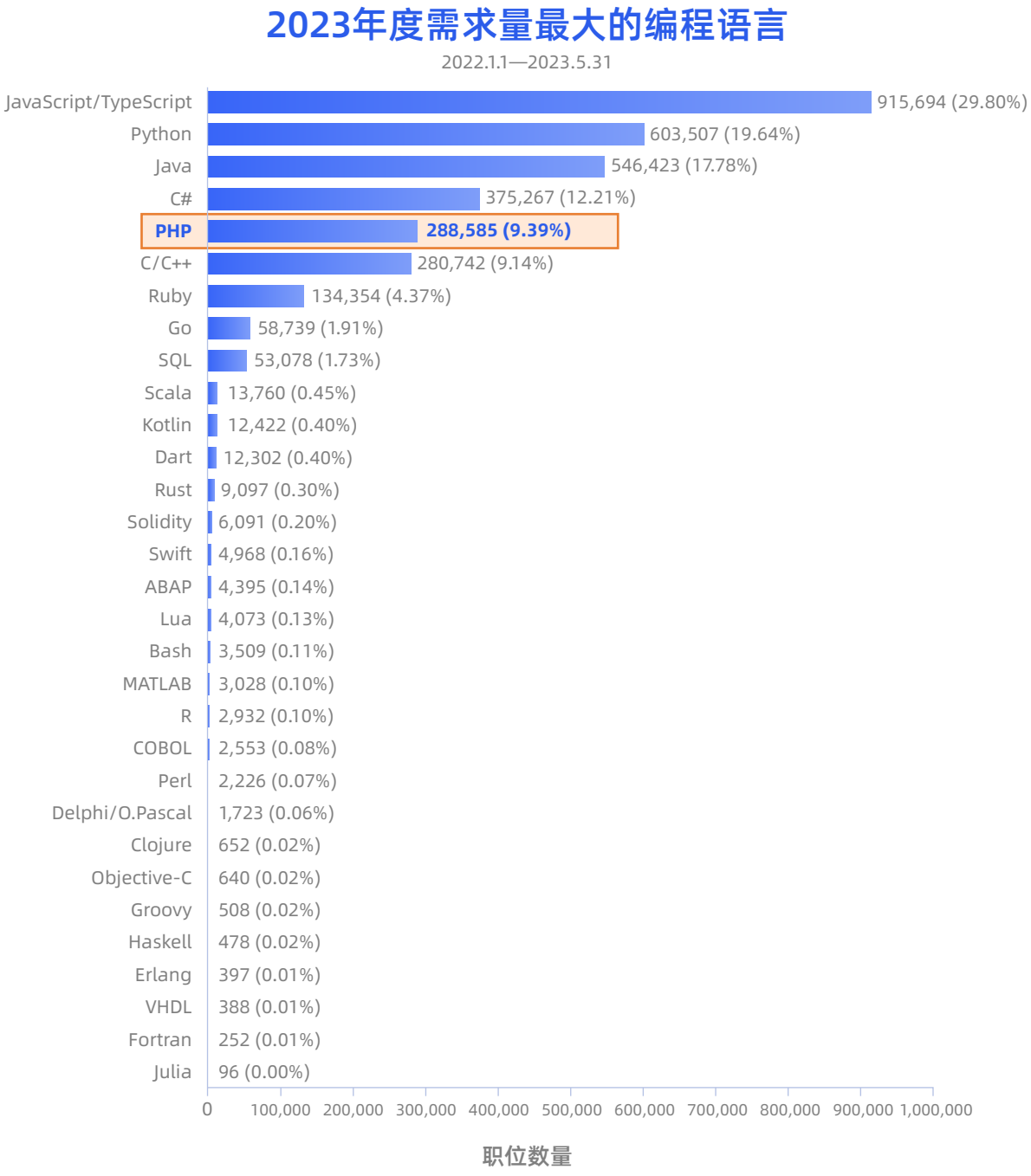
PYPL 编程语言流行指数是基于 Google 引擎中的用户搜索语言教程频率进行统计得出的。这一指标表明：搜索的语言教程越多，该语言被认为越受欢迎。从数据中可知，PHP 以4.86%的占比位居第六。

排名	语言	流行度	趋势
1	Python	28.05%	+0.1%
2	Java	15.88%	-1.0%
3	JavaScript	9.27%	-0.3%
4	C#	6.79%	-0.2%
5	C/C++	6.59%	+0.3%
6	PHP	4.86%	-0.4%
7	R	4.45%	+0.4%
8	TypeScript	2.93%	+0.1%
9	Swift	2.69%	+0.7%
10	Objective-C	2.29%	+0.2

来源：https://pypl.github.io/PYPL.html

DevJobsScanner报告

DevJobsScanner 发布了2023年度（2022.1-2023.5）需求量最大的编程语言数据，该报告调查了市面上共计1400万个开发人员职位，筛选了有明确编程语言需求的职位，最终得出了2023年度需求量最大的八种编程语言，其中 PHP 位居第五。



来源：https://www.devjobsscanner.com/blog/top-8-most-demanded-programming-languages/

国内招聘网站编程岗位现状

通过国内各大招聘网站编程岗位可以看出，PHP 依然是招聘热门岗位：

PHP 开发工程师岗位在国内招聘市场上需求量很大。许多公司和组织仍然依赖 PHP 来开发和维护 Web 应用程序。

此外，PHP 编程语言岗位的需求广泛分布在各个行业中，包括互联网公司、电子商务平台、IT咨询公司、传统企业等等。

PHP基金会2022年度报告

PHP 基金会2022年度报告中提及，截至2022年底，PHP 基金会成员结构为：10名志愿管理员和6名兼职付费开发人员。他们为 PHP 的发展做出了巨大的贡献。

PHP 基金会2022年的主要重点是加强 PHP 核心的维护，计划2023年进一步扩展开发团队，支撑 PHP 的更多工作。

2023年，基金会的目标：

组织目标：

- 扩展基础开发人员团队
- 推动基金会的社区运作与发展
- 改善沟通，为赞助商提供明确的利益
- 探索战略合作伙伴关系和营销机会

技术目标：

- 持续维护、开发PHP核心架构
- 提高基金会整理的想法和意见稿的质量
- 为PHP的更新发展制定更长远的路线图和愿景

2022年，PHP基金会得到了组织和个人的资金支持，目标是向尽可能多的核心开发人员支付有竞争力的薪酬。该集体收到了来自组织的580,000美元捐款，此外，还有1400人以个人支持或捐赠的名义为 PHP 基金会筹集资金。

从财务报告可以看出，PHP 基金会拥有健康的财务状况，对下一年的财务计划也很明确：2023年，PHP 财务支出会包括因开发人员参与 PHP 项目时间增加而产生的人力成本，以及与 OpenCollective 平台合作的运营成本等。

同时，PHP 基金会也通过 OpenCollective 平台持续公开日常收入与支出（截止2023年10月23日）：

企业和个人的捐款总额（扣除手续费前）：**1,200,719.86美元**

捐款合计（扣除手续费后）：**1,050,038.63美元**

目前已支出：**384,600.94美元**
当前余额：**664,719.1美元**

2023年计划支出：**553,137.81美元**

基于以上信息，我们可以得出如下结论：

PHP 基金会正处于正常、稳定的运作之中，且对未来也有着良好的计划和展望。

从上述报告可以看出，PHP 这门有着近三十年历史的编程语言曾几度被唱衰，“PHP 将亡”的言论伴随着“PHP 是最好的语言”的黑色幽默，多年来一直屡见不鲜，但时至今日，这门语言一直坚挺着。新语言层出不穷，老牌语言起起伏伏，各类趋势变化万千，而 PHP 颇有“敌军围困万千重，我自岿然不动”之势。

PHP应用现状概览

Zend by Perforce 就 PHP 的应用现状发布了“2023 PHP 形势报告”。该报告基于2022年10月至12月的匿名调查，对开发人员最常用的 PHP 工具、正在构建的内容以及所面临的挑战等多方面内容进行了统计分析。

PHP开发的优先级

报告显示，对于 PHP 开发的优先级这一问题，大多数开发团队（46%）优先考虑在当前的项目中构建新特性。而安全则是另一项重点关注的维度：19%的团队认为安全是团队持续开发的首要任务。提高应用程序性能（15%）和提高代码质量（14%）也是 PHP 开发团队重点关注的维度。

构建/部署什么类型的PHP应用

对于“大家构建或部署什么类型的PHP应用”问题，66.6%的受访者表示他们正在构建服务或API，60.5%的受访者正在构建内部的业务应用，还有43.6%的受访者通过PHP构建自己的内容管理系统（CMS）。PHP 正在快速适应产生业务价值的能力。

PHP的版本

参与报告调查的受访者中，近三分之二的人使用的版本是 PHP 7.4，还有近三成的人使用更“古老”的版本——PHP 5.6 和 PHP 5.0。关于版本升级的问题，受访者认为，在升级中最耗时的部分是重构，其次分别是测试、基础架构配置、规划和合规性更新。

来源：https://www.zend.com/blog/2023-php-report-overview

除此之外，PHP 生态系统正在经历大规模的现代化改造。有77%的受访者表示有采用容器化技术的计划，还有62%的受访者透露有计划采用编排技术。容器化已经成为一种趋势，有57.5%的受访者正在应用，较2022年30%的比例有所提升。

PHP 在云平台上的部署也要多于本地部署，受访者中有46%部署在 AWS，其次分别是占比19.5%的 Google Cloud Platform（GCP）、占比17.2%的 Digital Ocean，以及占比14.2%的 Azure。本地部署的占比则为36.8%。此外，关系数据库、Web API 等则是生产 PHP 应用程序的常见集成对象。

在这份报告的最后，也对PHP未来进行了展望：

自从 PHP 7 的发布和采用了一个可预测的发布周期以来，该语言已经取得了巨大的进步。

PHP 2023年最新进展

1月3日，Swow v1.0.0 发布

Swow 是一个使用 PHP 和 C 编写的高性能纯协程网络通信引擎，它致力于使用最小 C 核心及多数 PHP 代码，以支持 PHP 高性能网络编程。

项目初衷

尽管现在 PHP 协程生态已经有了非常多样的选择，但所有的协程框架、库都存在着一个非常致命的问题，即它们都是从异步框架演变而来，并非开始就自底向上为协程设计，且出于一些兼容性方面的考量，往往会牺牲掉很多本应具备的能力，又或是囿于历史包袱而无法做出合理的设计改变。

因此，Swow 项目致力于打造一个自下向上专为协程设计的高性能、高可控、易兼容的引擎，依托架构设计优势，全面释放协程技术的真正实力。此为 Swow 及其团队成员的初衷，Swow 项目成员也将持续投入并为此付出努力，同时欢迎大家加入项目中，参与开源建设。

设计理念

Swow 的最小 C 核心设计决定了它在保障关键性能之外，更多的优势是追求二次开发能力与可调试性。

来源：https://www.oschina.net/news/223626

Swow 内核与 PHP 代码的无缝融合运作使得开发者的编程效率与灵活性能得到极大的提升。

Swow 提供了丰富多样的调试机制与强大且低门槛的调试能力，最大程度地确保开发者免受 Bug 困扰，普通开发者也能通过工具的加持具备接近专家级别的 Debug 能力。

总的来说，Swow 的未来充满了想象空间，它为 PHP 提供了远超以往的可能性，我们可以藉由 Swow 的能力去想、去做更多未曾深思过的事情。

1月3日，Hyperf 3.0发布，开启PHP新时代

Hyperf 是基于 Swoole 4.4+ 实现的高性能、高灵活性的 PHP 协程框架，内置协程服务器及大量常用的组件，性能较传统基于 PHP-FPM 的框架有质的提升，提供超高性能的同时，也保持着极其灵活的可扩展性，标准组件均基于 PSR 标准实现，基于强大的依赖注入设计，保证了绝大部分组件或类都是可替换与可复用的。

项目初衷

尽管现在基于 PHP 语言开发的框架处于一个百花齐放的时代，但仍旧未能看到一个优雅的设计与超高性能共存的完美框架，亦没有看到一个真正为 PHP 微服务铺路的框架，此为 Hyperf 及其团队成员的初衷。Hyperf 团队也将持续投入并为此付出努力，欢迎大家加入其中，参与开源建设。

Hyperf 3.0新时代

最新发布的 Hyperf 3.0 带来了包含原生注解、分布式事务、Swow 网络引擎、SDB 协程调试器等很多非常有意思的新能力，其中一些新能力是 PHP 领域里面前所未有的。Hyperf 3.0 还做了大量的优化和调整。

原生注解（Attribute）

PHP 8.1、8.2 的发布，给 PHP 带来了很多新的特性，其中与 Hyperf 最为相关的就是PHP的原生注解（Attribute）了，Hyperf 3.0 放弃了过往采用的基于注释解析的注解功能实现，转而采用 PHP 的原生注解。当然对于依赖的 PHP 版本，Hyperf 也将最低要求调整为 PHP 8.0 版本。

分布式事务

在过去的一年里，Hyperf 团队也为 PHP 孵化了两个分布式事务组件，并贡献到对应的开源社区，对应 DTM（首个基于 Go 语言实现的流行分布式事务管理器）与 Seata（由阿里巴巴开源的流行分布式事务管理器）两款主流的开源分布式事务管理器，分别是 dtm-php/dtm-client 和 seata/-seata-php。其中 dtm-php 是实现了 dtm 完整功能的分布式事务客户端，已支持 TCC 模式、Saga、XA、二阶段消息模式的分布式事务模式，并分别实现了与 DTM Server 通过 HTTP 协议或 gRPC 协议通讯，该客户端可安全运行于 PHP-FPM 和 Swoole 协程环境中，更是对 Hyperf 框架做了更加易用的功能支持，可应用于生产环境中；而 seata-php 仍在开发迭代中，尚未能用于生产环境，也希望能有更多人参与进来共同迭代。

2月14日，Laravel 10发布

Laravel 10 现已发布，包括最低 PHP 8.1 版本、新的 Laravel Pennant 包、可调用验证规则、本机类型声明等.....

Laravel 10放弃对PHP 8.0的支持

Laravel 框架将在 Laravel 10 中放弃对 PHP 8 及更低版本的支持。所需的 PHP 最低版本是 PHP 8.1 以上的版本。查看 Laravel 9.x 和 master 之间的差异，我们可以期待看到框架中使用的了 PHP 8.1 的特性，例如只读属性。

Laravel Pennant

Laravel Pennant 是 Laravel 团队创建的一个包，它将随 Laravel 10 一起发布，并为用户的应用程序提供特性标志。

特性标志能够帮助用户更有信心地逐步推出新的应用程序功能、对新的界面进行 A/B 测试、补充基于主干的开发策略等等。

这个包是核心团队提供的官方包系列中的最新包，也是构建良好且经过测试的包，能够为用户提供更多强大的功能。

来源：http://laravel.p2hp.com/

Laravel的进程层

Laravel Process 服务让“实现与测试和运行 CLI 进程一起工作”的梦想成为可能。

```php
use Illuminate\Support\Facades\Process;

$result = Process::run('ls -la');

$result->successful();
$result->failed();
$result->exitCode();
$result->output();
$result->errorOutput();
$result->throw();
$result->throwIf($condition);
```

Process层包括开箱即用的丰富功能，例如：

在运行之前构建流程实例的 Fluent 流程方法	异步进程	通过 fake() 实现丰富的测试功能
在收到输出时进行处理	进程池	防止出现测试期间的杂散过程

测试过程从未如此简单。

Laravel 10骨架中的原生类型声明

在 Laravel 10 中，应用程序骨架代码将具有原生类型声明。这意味着框架生成的用户空间中的任何代码都将具有类型提示和返回类型。

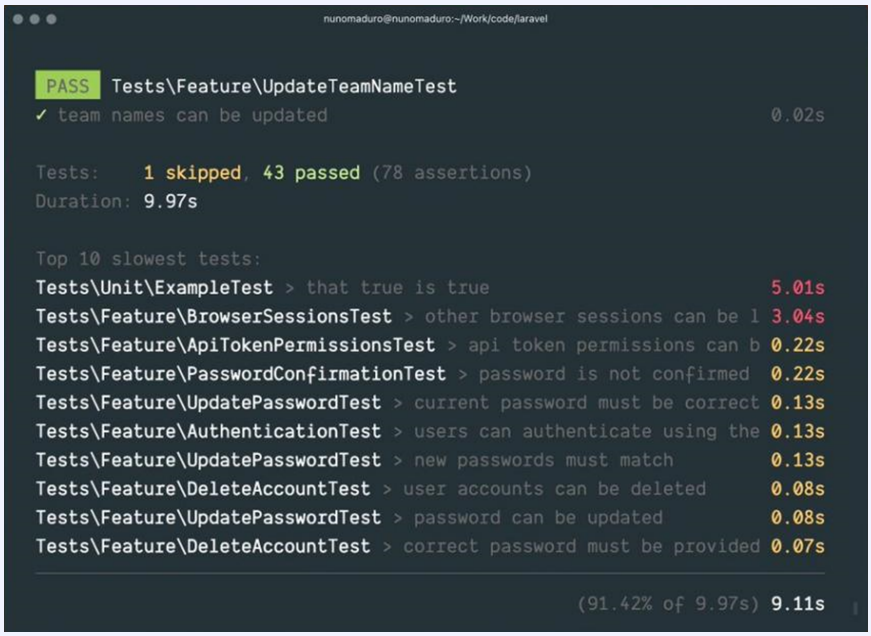
可调用验证规则是默认的

从 Laravel 10 开始，可调用验证规则已成为默认的规则。当用户通过 artisan 创建新的验证规则时，我们可以预想到如下结果：

```
# Laravel 9 creates a rule class that implements the
# Illuminate\Contracts\Validation\Rule interface
artisan make:rule Uppercase

# Laravel 9 flag to create an invokable and implicit rule
artisan make:rule Uppercase --invokable
artisan make:rule Uppercase --invokable --implicit

# Laravel 10 creates an invokable rule by default
artisan make:rule Uppercase

# Laravel 10 implicit rule
artisan make:rule Uppercase --implicit
```

测试的配置文件选项

Laravel 10 的一个新特性是一个 --profile 选项，它可以让用户轻松找到应用程序中所有缓慢的测试。



这个 --profile 选项有助于维持快速测试的水平，修复慢速测试。

新的字符串密码助手

Str::password　方法可以生成给定长度的安全随机密码。密码将由字母、数字、符号和空格的组合组成。默认情况下，密码长度为32个字符：

```php
use Illuminate\Support\Str;

$password = Str::password();

// 'EbJo2vE-AS:U,$%_gkrV4n,q~1xy/-_4'

$password = Str::password(12);

// 'qwuar>#V|i]N'
```

8月31日，PHP 8.3.0 RC1发布

PHP 8.3.0 发布了第一个 Release Current（候选版本）。按照发布计划，RC阶段会有6个版本更新，现已分别于9月14日发布 RC2、9月28日发布 RC 3、10月12日发布 RC 4 版本，最终将于11月23日发布正式 PHP 8.3.0 版本。

PHP 8.3.0更新一览表

PHP 8.3.0带来的新变化如下：

常量支持类型化（Typed class constants）
支持获取动态的类常量(Dynamic class constant)和枚举成员(Enum member)
添加json_validate()函数，用于验证JSON
新增Random扩展
添加mb_str_pad()
实现#[\Override]属性
添加更多PHP Sockets选项
增加对cURL 7.87及以下版本的新cURL选项和常量的支持
支持匿名只读类（Anonymous read-only classes）
支持在数组中使用负数索引（Negative indices）

期待 PHP 8.3.0 带来更多的改进。

来源：https://www.php.net/

PHP社区用户访谈

开源先锋鲁飞：跟禅道一起，做PHP铁杆粉丝

网名沈唁，GitHub：@sy-records，PHPMQTT作者，Docsify、Hyperf、PHP、Swoole、ThinkPHP等开源项目开发组成员，入选2022年中国开源先锋33人。

都说“PHP是最好的语言”，你是怎么看待这个说法的？

“PHP 是最好的语言”这个梗来自于PHP官方手册中，到现在为止，感觉成为了一个讽刺 PHPer 的梗。在我看来，没有任何一种编程语言是“最好的语言”，每种编程语言都有它的优点和缺点，这些因素取决于该语言的设计目的、开发者需求和编写的应用程序类型等多种因素。

PHP 具有简单易学、丰富的开发工具和库等优点，常用于Web开发，特别是有了 Composer 之后，Composer 给 PHP 带来了更好的依赖关系管理和更方便的开发体验，使得 PHP 开发人员能够更专注于应用程序的功能和业务逻辑的开发。

像其他编程语言一样，它也有一些缺点，例如在安全性方面可能需要多加注意与维护、不能轻松地支持多线程和异步编程等高级特性、同时也存在一些争议性的设计决策和语言特性。

但我们也可以使用像 Swoole、Swow、Fiber 等这种扩展来弥补一些它的不足之处.....

总之，没有一种编程语言可以适用于所有情况，也没有一种编程语言是绝对“最好”的。选择使用哪种编程语言应该取决于开发者的需求、团队的技术栈和开发项目的特点。

你是从什么时候接触PHP的，又是为什么选择了PHP？

接触 PHP 应该算是从15、16年开始，那会还比较流行微商城的系统，就打算自己研究怎么做一个，于是乎就研究起了 PHP。

至于为什么选择 PHP 吧，也是因为它看起来比较简单，上手轻松，一入 PHP 深似海，从此就走上了撸码的道路。

后来开始为开源做贡献之后，就接触到了 Swoole、ThinkPHP 等这些比较优秀的项目，也和一群志同道合的人一起研发 Hyperf，我们也是一直围绕 PHP 的生态，为 PHP 生态做贡献。

当前公司用PHP的项目现状是怎样的？

目前就职于禅道软件，带了一个部门研发新的禅道版本（感兴趣的话可以期待一下）。**禅道可以说算是 PHP 的铁杆粉丝了**，禅道软件是一个项目管理软件，到目前为止依旧在支持 PHP 的低版本，至今未抛弃还在使用低版本PHP的用户。当然也有项目使用 PHP 8.x 系列，开始拥抱 PHP 高版本，使用强类型、协程等功能。

禅道软件也在PHP生态中做出了自己的贡献：

1. 禅道项目管理：基于自研的PHP框架，研发出功能强大、专业且通用、开源且免费的项目管理软件；
2. phpmicro：micro 自执行 SAPI 提供了 php“自执行文件”的可能性；
3. PHPSciter：PHPSciter 项目集成了跨平台的 GUI 开发框架 Sciter，旨在为广大的 PHP 开发者提供一套便利的 GUI 开发框架。
4. Phiwrapper：Phiwrapper 是一个跨平台应用程序打包方案。使用它可以将 PHP 的运行环境以及 PHP 的代码打包成一个可执行文件，并且可以在 Windows、Linux 和 MacOS 上面运行。
5. 自研 ZenTools、zbox.....

禅道软件也希望尽自己的一份力量，让 PHP 更加实用！

注意到当前Swow、Hyperf的作者都很年轻（91年-97年不等），你认为可能的原因是什么？

PHP 从1995年对外发布第一个版本，至今也28年之久了。

以我个人来说，我是在15、16年开始接触的 PHP，那会儿对于一个语言的衰落和昌盛还不是很了解和在乎，当时主要接触的是应用层，就觉得那些写底层代码的大佬很厉害，想着有一天能跟他们接触接触多好。

在机缘巧合之下，我接触到了 Swoole 团队，这会才开始接触国内 PHP 的底层大佬们，通过参与 Swoole 开源项目，获得更多的编程经验，同时也开始参与其他的开源项目。

年轻一代的加入，更愿意和能够承受风险去创新，也有更多的时间去学习、使用和掌握新技术，也更容易适应新的技术变化，在共建的同时也造就了一些非常活跃的框架、包和平台，更加丰富了 PHP 语言的生态环境。

年龄并不能决定一个人的工作能力或者他们对技术的理解深度，而年轻一代的加入也是在前人的基础上继续添砖加瓦，为 PHP 社区做出贡献，和社区成员们一起推动 PHP 社区的发展和进步。

Hyperf核心作者李铭昕：如果PHP发展不行，那我就让他变得行

李铭昕 Hyperf核心作者之一，前KnowYourself技术负责人

都说“PHP是最好的语言”，你是怎么看待这个说法的？

现在我们都把这句话当作一个梗，其实这句话最开始是官方自己定位的原话，当时通常是说在 Web上，脱离上下文直接下这个结论确实容易找麻烦，所以后来大家都把它当作一个梗来说。

但是在我看来的话，因为我用的语言比较多，至少写过5种语言，或者说包括大学的 C 和 C++ 是7种，但是到接触 PHP 之后就一直主要在用 PHP，因为 PHP 确实是我自认为从性能跟开发效率上来说的最优解。回到刚才那句话，要真说最好的，对我个人来说 PHP 确实是最好的，在这大概7种语言里，我写到现在，认为 PHP 确实是写起来最舒服的。

从公司层面来说，提及性能，会认为 Java 的性能更好，但 Java 吃内存也非常高。PHP 基于 Swow/Swoole 框架和 Java 基于 Spring 框架来比的话，性能上不吃亏，但开发速度却快多了。同样的，Golang 的性能也很不错，在开发效率上还是跟 PHP 有差距。

除此之外，PHP 也容易上手，学习门槛较低，像我是看了一周 PHP 的语法，就可以上手写了，而且写得很好。

所以对我个人来说，尽管这是个梗，我还是认可这句话的。

你是从什么时候接触PHP的，又是为什么选择了PHP？

我现在毕业有10年了，只有2年的时间写 .NET，后面8年都是写 PHP 。Java 和 Go 是穿插进行的，Java大概写了1年，Go 的话做了3年左右，但这两个对我来说都是辅助性语言，是用来辅助我写 PHP 的。

一开始选择PHP是基于刚刚说到的性能和效率吗？

那倒不是（笑），那个时候懂啥。就是从第一家公司跳到第二家公司的时候，第二家公司就是写 PHP 的。因为那个时候 PHP 蛮火的，虽然我是零基础，但跳过去公司也要了，基本上我自己一学就入门了。当时的项目就是做网页，还比较传统。自从在这家公司写PHP之后，我再跳槽也是找 PHP 的工作，一直到现在。

当前公司用PHP的项目现状是怎样的？

其实我不是只写 PHP，公司 Golang 和 Node.js 的项目也都是我在写，但主要业务全部是用 PHP 来开发的。Node.js 是祖传遗留项目的问题，有几个小服务是用 Golang 写的。Hyperf 在开源之初我们公司就在使用了，当时 Hyperf 对于阿里云上传的模块，还没有处理好，我们就先用 Golang 写了一部分服务，处理阿里云上传之类的事务。现在已经不需要 Golang 服务了，可以完全用 PHP 自己来写。之前我们还是有 Golang 场景的，有时候做运维方面的工作或者开发一些小工具给业务或产品人员来使用，如果是用 PHP 开发的话他们本地需要有环境来跑，会受到一些限制。当时没有 Swow/Swoole 这些生态，就用 Golang 来写，打包成二进制进行分发，再让他们去本地执行一些脚本。

发展到后来，尤其是现在，这些基本被淘汰了，Hyperf 开源了一个 box 组件，可以直接把 PHP 代码打包成二进制，放到 Windows、Linux、Mac 下直接运行。原来这部分用 Golang 来写的业务现在也不太需要了。

目前 know Yourself 公众号和“月食”App 的后端都是用 PHP 来开发的，后端的接口反馈数据都是 PHP 提供的。

是怎么想到开发Hyperf框架的？

我四五年前刚到 know Yourself 的时候，项目问题还比较多。像我们公众号发测试活动之类的，公众号一推送，用户进去是白屏，高峰期非常卡。当时每次推活动先买机器，活动开始前先买10台云服务机器，把项目铺上，流量结束后再把机器撤下来。

现在是整个跑在 Swarm 集群上，需要扩容的话直接进行一个指令就自动扩容了，完全基于 Docker，很正规了。

但当时没办法，找了很多种解决方案都不太行。最先研究的框架是 Phalcon，这是一个基于 C 扩展写的框架，当时给那个框架提 Pr，熟悉了 GitHub 的开源流程。这个项目对 Pr 的处理速度比较慢，跟老外沟通也稍微有点费劲，于是又看国内的框架。当时 Swoole 生态比较看好的是 Swoft 框架，所以当时第一个新项目就使用了 Swoft 框架，我也因为在使用过程中，给 Swoft 修改了大量的 Bug，后来才有幸可以进入 Swoft 开发组。但是后来，还是因为 Swoft 设计理念与我有些偏差，所以才想到跟黄朝晖一起出来重新搞了 Hyperf 框架。

Hyperf框架的现状如何？

Hyperf 作者目前有10人左右，是这四年里慢慢发展的，有的人贡献多了，我们就给拉进来，相当于转个正，成为我们的核心作者了。1月刚发布了 Hyperf 的3.0版本，带来了包含原生注解、分布式事务、Swow/Swoole 网络引擎、SDB 协程调试器等很多非常有意思的新能力。

之后 Hyperf 主要方向是丰富生态，因为主要基调是确定的。真要看的话跟 Laravel 的设计上是很像的，对标 Java 的 Spring Core 的整套体系。因为现在的PHP生态说实话确实还不够，Hyperf 的目标就是**丰富生态，弥补微服务生态的不足**。

也就是说Hyperf的所有核心人员都是国内的，自始至终是国产原生的开源项目。

对，目前核心贡献者也都是国内开发者。现在 Hyperf 在 GitHub 上有5.1k的Star，在国内外都得到了一定的应用和推广。我自己是很喜欢写代码的，所以对我来说开源不是负担，是一件值得让我坚持的事情。

我手上也还有其他的开源项目，也会帮忙维护，比如我会帮忙改 Easy-WeChat 这个项目的一些东西，但不在开发组里面。整体来说参加开源工作蛮多的，都是围绕 Hyperf 来的，一些 Hyperf 的周边项目是别人在维护，但有时候断更的时候只能我去维护。如果再搞几个和 Hyperf 同样量级的开源项目，我这边时间确实不够，没有办法全职做开源。

你怎么看待PHP的现状？对PHP未来发展有什么展望？

PHP 这个问题，我从上家公司离职的时候，当时的上司就跟我说“你再跳槽去一家 PHP 的公司，PHP 在大厂也没有什么职位，只能一直在小厂中去跳。PHP 的这点东西也不多，你想想后面怎么发展”。我的回答很简单：“**PHP 发展不行，我就让他发展好就可以了**。”

一直到现在，我们还是有这么一波人——黄朝晖、Twosee（陈曹奇昊）等都在坚守，这一波人其实换语言真的分分钟就换走了，语言对我们来说没有很高的壁垒。这些人里，91年、94年、95年、95年的都有，很年轻的一群人。为什么守着 PHP 呢，还是铆着一股劲，还是有一部分感情，如

果它不行，那我们让它变得行。所以现在我们一直还在打造 Hyperf 的生态，想把它扩大、占领市场。

PHP 的未来发展肯定是越来越好的，但现在还是需要有更多的人来做，把大家的思想转变过来。比如前段时间我们 CEO 就问我：为什么不用 Java，别人都在用 Java。我的回答很简单，就是便宜。**要达到相同的性能，PHP 的人工和服务器都更便宜**，这就非常直观。不需要说别的，我们用PHP的性能也很OK，这就是一种思想转变。

这一路我是被动、甚至被逼出来使用 PHP，又来开发 Hyperf 的，但写到现在我真的觉得**很好，不后悔**。

报告小结

从此次 PHP 综合分析报告中，我们可以得出如下结论：

1. PHP 凭借其容易上手、高运行效率、高跨平台性等优势，在过去一直是主流编程语言，最近几年，PHP 在国外的流行度也有所上升。这说明，PHP 在国际范围内仍具有广泛的影响力，也在不断地吸引着众多开发者加入学习与实践。

2. 在发展方面，PHP 核心团队一直保持着稳健的风格，不断进行渐进式的改进，未来可期。

3. 此外，PHP 的高开发效率以及与 Swoole 等高性能解决方案的结合，让 PHP 的开发效率也得到了进一步提升，使得 PHP 越来越受到创业团队的青睐。

合作社区


ALLData


重庆软件园
CHONGQING SOFTWARE PARK


FASTADMIN


FSCN


开源
江湖


开源社
kaiyuanshe


Measures 麦哲思


敏航管理咨询


msup


OSCHINA
oschina.net


渠成开源社区
QuCheng Open Sourse


新知 商业新知
每天用新进阶 工作轻松财富多


TesterHome


自由 开放 分享
testwo测试窝


VIPTEST


中国DevOps社区

2022-2023年度PHP综合现状分析报告

www.zentao.net

禅道官网

禅道项目管理软件团队 出品