The PHP Foundation

Laracon 2024

How many contributors were paid to work on the PHP language in 2021?

PHP’s Bus Factor

Dmitry Stogov

Nikita Popov

The "**bus factor**" is how many team members can unexpectedly leave before a project stalls.

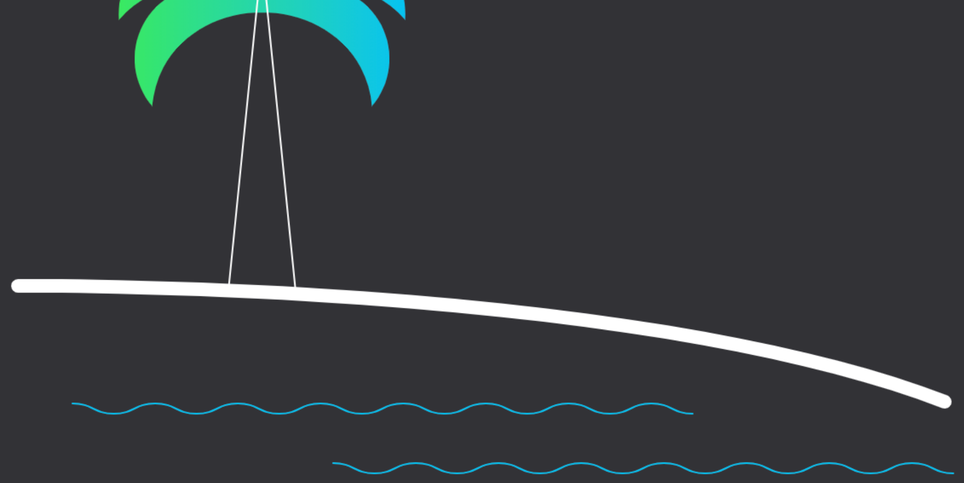
structure, making it easy for developers familiar with C, Perl, and similar languages to adopt.

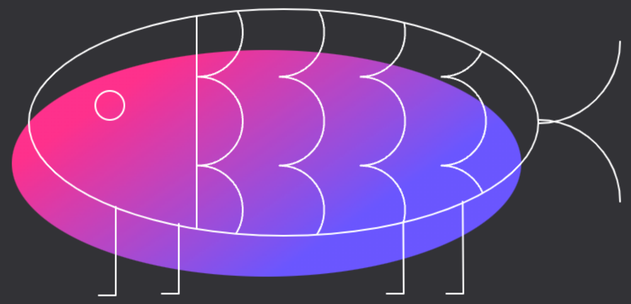
1995

Rasmus Lerdorf starts working on PHP

The very first incarnation of PHP is a simple set of Common Gateway Interface (CGI) binaries written in the C programming language (not Perl). Originally used for tracking visits to Rasmus Lerdorf's online resume, he named the suite of scripts "Personal Home Page Tools," more commonly referenced as "PHP Tools".

The Timeline data is opensource and available on GitHub. If you found a mistake, please feel free to create a pull-request.





1999

Zend company founded

The name Zend is a combination of Suraski's and Gutmans' forenames, Zeev and Andi. Zend has been a source of PHP innovations and development ever since its founding. It has contributed core improvements as well as userland packages and frameworks and development environment tooling.

ElePHPant Logo

Vincent Pontier creates an elePHPant logo for the PHP project, and it has been the adorable mascot of the PHP language ever since.

phpMyAdmin project was "born"

Tobias Ratschiller releases version 0.9.0 of the phpMyAdmin web administration tool for MySQL. For a long time it remained one of the primary tools for accessing MySQL and MySQL-like databases.

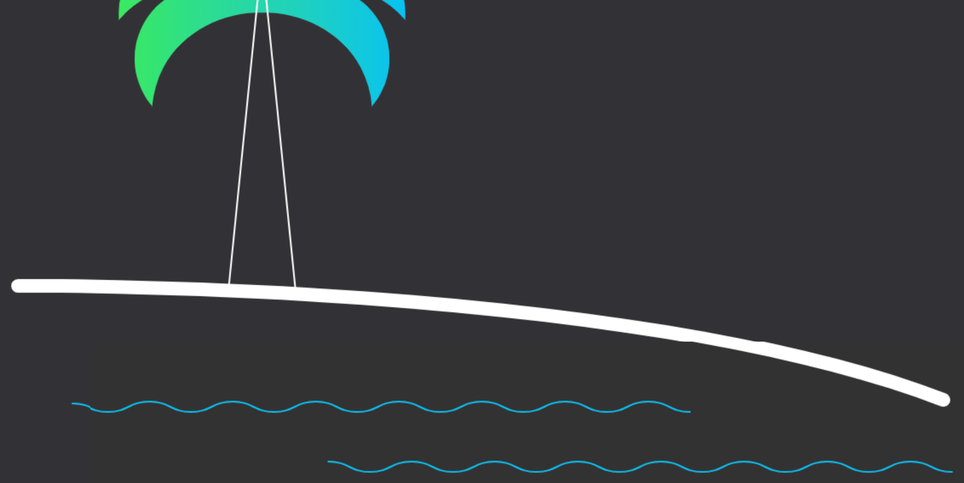
PHP 3.0

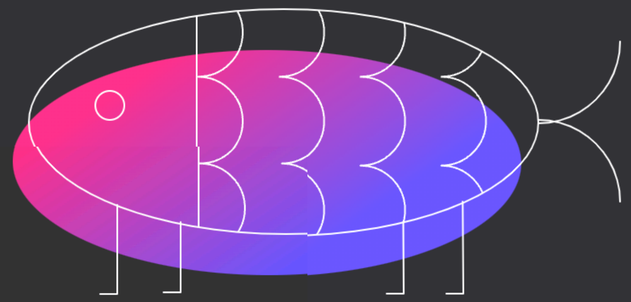
PHP 3.0 was the first version that closely resembles PHP as it exists today. Zeev and Andi rewrote the interpreter and introduced a new extension API which makes it a lot easier for the community to contribute new extensions. With many new developers from around the world on board, PHP 3 was announced by the new PHP Development Team.
The language's name changed to the recursive acronym "PHP: Hypertext Preprocessor". At its peak, PHP 3 was installed on approximately 10% of all web servers on the Internet.

PHP Logo

Netcraft Survey

First PHP books





Jan 1

1999

Zend company founded

The name Zend is a combination of Suraski's and Gutmans' forenames, Zeev and Andi. Zend has been a source of PHP innovations and development ever since its founding. It has contributed core improvements as well as userland packages and frameworks and development environment tooling.

ElePHPant Logo

Vincent Pontier creates an elePHPant logo for the PHP project, and it has been the adorable mascot of the PHP language ever since.

phpMyAdmin project was "born"

Tobias Ratschiller releases version 0.9.0 of the phpMyAdmin web administration tool for MySQL. For a long time it remained one of the primary tools for accessing MySQL and MySQL-like databases.

PHP 3.0

PHP 3.0 was the first version that closely resembles PHP as it exists today. Zeev and Andi rewrote the interpreter and introduced a new extension API which makes it a lot easier for the community to contribute new extensions. With many new developers from around the world on board, PHP 3 was announced by the new PHP Development Team.

The language's name changed to the recursive acronym "PHP: Hypertext Preprocessor". At its peak, PHP 3 was installed on approximately 10% of all web servers on the Internet.

PHP Logo

Netcraft Survey

First PHP books

Laravel 2

CakePHP 2

Composer

By 2011, all major languages already have their own package managers: npm, Bundler, pip, NuGet. Coming in last, PHP receives a state-of-the-art package management tool thanks to Jordi Boggiano, Nils Adermann, and other contributors. OHAI PHP DEVS, Y U NO PACKAGE?

Symfony 2 **Monolog initial release**

Laravel 1

Taylor Otwell creates Laravel as an attempt to provide a more advanced alternative to the CodeIgniter framework. It will go on to revolutionize the entire PHP landscape.

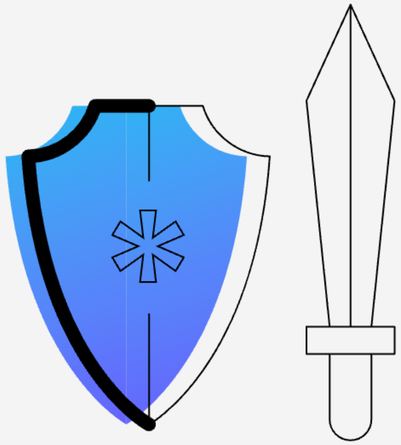
Guzzle 1.0

FuelPHP CodeIgniter 2

2011

Doctrine 2.0 released **FastCGI Process Manager (FPM)**

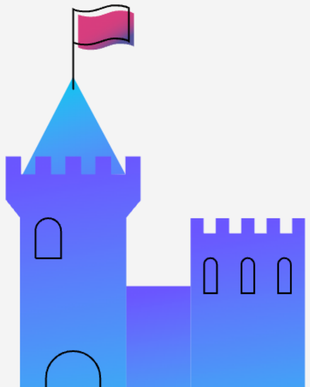
Slim 1



PSR-0 accepted

After receiving namespaces, PHP finally gets proper autoloading. Before this first PSR standard was accepted, PHP files were an insane mix of includes and requires.

Unofficial PHP Anthem



PhpStorm 1.0 Released

Previously known under its working

Laravel 2

CakePHP 2

Composer

By 2011, all major languages already have their own package managers: npm, Bundler, pip, NuGet. Coming in last, PHP receives a state-of-the-art package management tool thanks to Jordi Boggiano, Nils Adermann, and other contributors. OHAI PHP DEVS, Y U NO PACKAGE?

Symfony 2 Monolog initial release

Jun 9

Laravel 1

Taylor Otwell creates Laravel as an attempt to provide a more advanced alternative to the CodeIgniter framework. It will go on to revolutionize the entire PHP landscape.

Guzzle 1.0

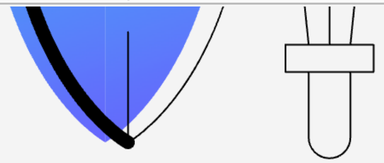
TaylorOtwell	2011-06-09 14:25:33	#1
The Creator From: Arkansas, USA Registered: 2011-06-06 Posts: 1,168	Get it now and enjoy a fresh new PHP framework! http://laravel.com/download/latest	

FuelPHP CodeIgniter 2

2011

Doctrine 2.0 released FastCGI Process Manager (FPM)

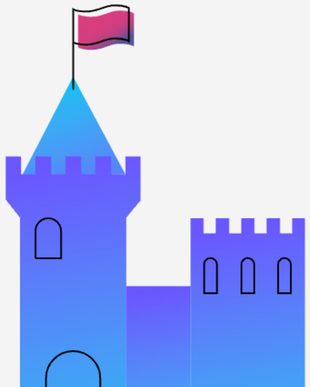
Slim 1



PSR-0 accepted

After receiving namespaces, PHP finally gets proper autoloading. Before this first PSR standard was accepted, PHP files were an insane mix of includes and requires.

Unofficial PHP Anthem



PhpStorm 1.0 Released

Previously known under its working

Laravel 2

Fix bug #61097 Memory leak in xmlrpc functions copying zvals
nikic committed 13 years ago

Fix bug #61097 Memory leak in xmlrpc functions copying zvals
nikic committed 13 years ago

Fix headers generated by phar
nikic committed 13 years ago

Fix headers generated by phar
nikic committed 13 years ago

Fix cli server tests broken by r323078
nikic committed 13 years ago

Jun 9

Composer

By 2011, all major languages already have their own package managers: npm, Bundler, pip, NuGet. Coming in last, PHP receives a state-of-the-art package management tool thanks to Jordi Boggiano, Nils Adermann, and other contributors. OHAI PHP DEVS, Y U NO PACKAGE?

Laravel 1

Taylor Otwell creates Laravel as an attempt to provide a more advanced alternative to the CodeIgniter framework. It will go on to revolutionize the entire PHP landscape.

TaylorOtwell 2011-06-09 14:25:33 #1

The Creator

From: Arkansas, USA

Registered: 2011-06-06

Posts: 1,168

Get it now and enjoy a fresh new PHP framework!

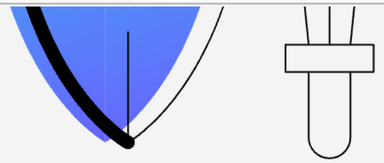
http://laravel.com/download/latest

2011

Doctrine 2.0 released

FastCGI Process Manager (FPM)

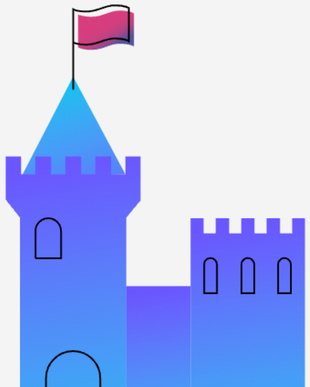
Slim 1



PSR-0 accepted

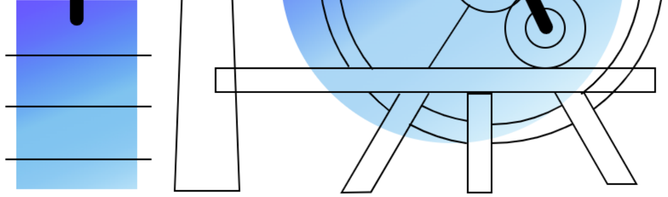
After receiving namespaces, PHP finally gets proper autoloading. Before this first PSR standard was accepted, PHP files were an insane mix of includes and requires.

Unofficial PHP Anthem



PhpStorm 1.0 Released

Previously known under its working



Composer 1.0 goes gold

PhpStorm 2016.1

Andi Gutmans leaves Zend

Zend Expressive 1.0

2016

PSR-6: Caching Interface accepted

Symfony 3 Drupal 8 released Magento 2

PHP 7.0

A major language renewal comes with the new Zend Engine 3 (PHPNG), numerous improvements and new features. It's also twice as fast as the previous PHP 5.6 release!

Rogue Wave acquires Zend

Zend, the company that had stood behind the language since PHP 3, is bought out by Rogue Wave.

A Field Guide to Elephpants started

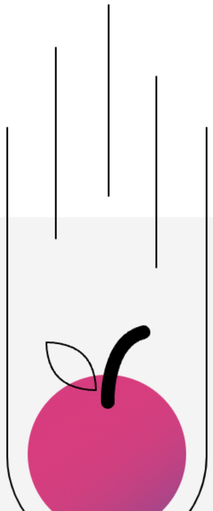
PHP turns 20

PSR-7 HTTP Message Interfaces accepted

The standard finally settled on how to deal with HTTP requests and responses the OOP way.

CodeIgniter 3 CakePHP 3

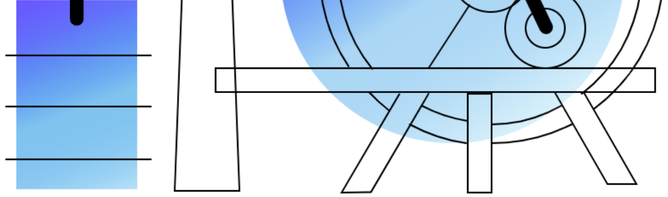
Zend JIT Open Sourced Laravel 5



2015

PSR-0 deprecated in favor of PSR-4 Yii 2.0

PHP-Parser 1.0



Composer 1.0 goes gold

PhpStorm 2016.1

Andi Gutmans leaves Zend

Zend Expressive 1.0

2016

PSR-6: Caching Interface accepted

Symfony 3 Drupal 8 released Magento 2

Oct 6

PHP 7.0

A major language renewal comes with the new Zend Engine 3 (PHPNG), numerous improvements and new features. It's also twice as fast as the previous PHP 5.6 release!

Rogue Wave acquires Zend

Zend, the company that had stood behind the language since PHP 3, is bought out by Rogue Wave.

A Field Guide to Elephpants started

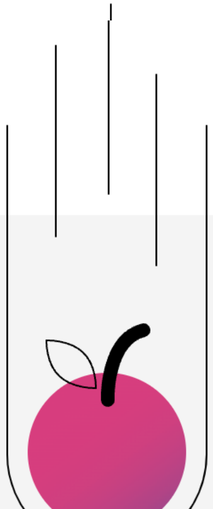
PHP turns 20

PSR-7 HTTP Message Interfaces accepted

The standard finally settled on how to deal with HTTP requests and responses the OOP way.

CodeIgniter 3 CakePHP 3

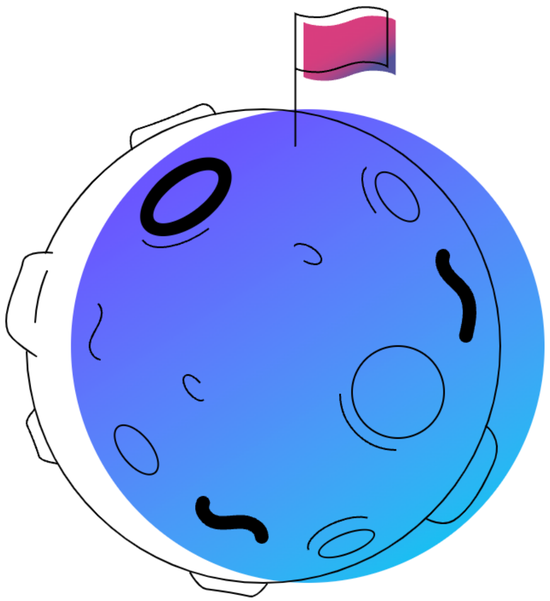
Zend JIT Open Sourced Laravel 5



2015

PSR-0 deprecated in favor of PSR-4 Yii 2.0

PHP-Parser 1.0



PHP 7.3

Updates for Heredoc and Nowdoc strings, trailing comma in arguments list, new `hrtime()` function, and improved garbage collection.

Nikita Popov joins PhpStorm team

A well-known and long-time contributor to the PHP ecosystem joins JetBrains to work with the PhpStorm team on improving the PHP language and ecosystem.

Zeev Suraski leaves Zend

Zeev Suraski is the second of Zend's two founders to leave the company. Matthew Weier O'Phinney and Enrico Zimuel also leave.

2019

Laravel 6

PSR-12 Extended Coding Style Guide accepted

P++ discussion

Doctrine Company, LLC created

Zend Framework becomes Laminas

PSR-14 Event Dispatcher accepted

Narrow margins for RFCs abolished

PEAR disabled by default

Perforce acquires Rogue Wave Software, owner of Zend

The Xdebug Experience

PHP 5.6.40 – the last PHP 5 release

Symfony leaves PHP-FIG

PSR-18 HTTP Client accepted

PSR-17 HTTP Factories accepted

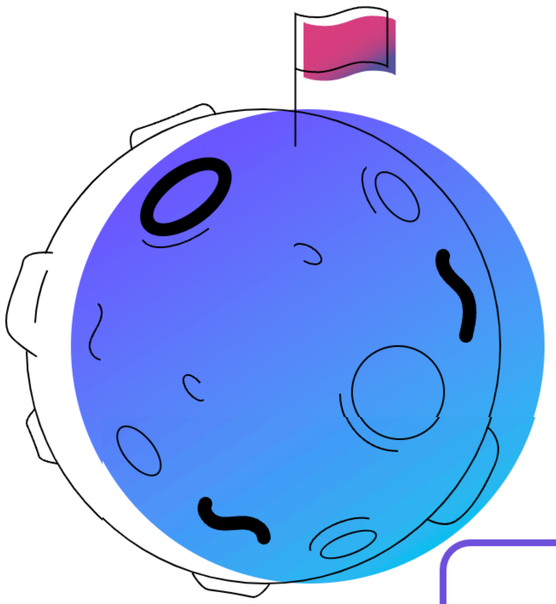
ReactPHP 1.0 LTS

Adobe acquires Magento

PHP FFI announced

Zend Expressive 3

Psalm 1.0



Laravel 6

PSR-12 Extended Coding Style Guide accepted

P++ discussion

Doctrine Company, LLC created

Zend Framework becomes Laminas

PSR-14 Event Dispatcher accepted

Narrow margins for RFCs abolished

PEAR disabled by default

Perforce acquires Rogue Wave Software, owner of Zend

The Xdebug Experience

PHP 5.6.40 – the last PHP 5 release

2019

Jan 11

Nikita Popov joins PhpStorm team

A well-known and long-time contributor to the PHP ecosystem joins JetBrains to work with the PhpStorm team on improving the PHP language and ecosystem.

PHP 7.3

Updates for Heredoc and Nowdoc strings, trailing comma in arguments list, new `hrtime()` function, and improved garbage collection.

Symfony leaves PHP-FIG

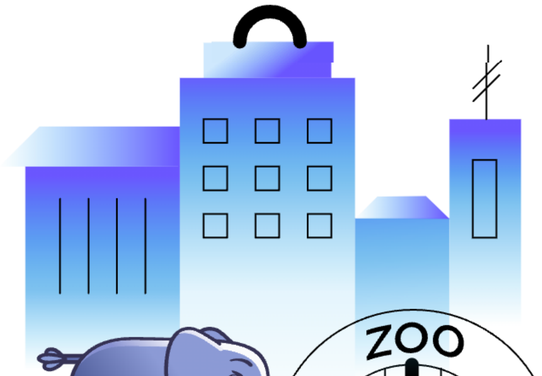
PSR-18 HTTP Client accepted

Zeev Suraski leaves Zend

Zeev Suraski is the second of Zend's two founders to leave the company. Matthew Weier O'Phinney and Enrico Zimuel also leave.

PSR-17 HTTP Factories accepted

ReactPHP 1.0 LTS



Adobe acquires Magento

PHP FFI announced

Zend Expressive 3

Psalm 1.0

Avoiding buses

Avoiding buses

May 4, 2021: Blog post by Joe Watkins

So, What ?

That number, 2. This is not an acceptable bus factor for a project the size and importance of PHP.

With every passing RFC, the project gets a little more complicated and has no more ability to maintain that complexity.

Avoiding buses

May 4, 2021: Blog post by Joe Watkins

So, What ?

That number, 2. This is not an acceptable bus factor for a project the size and importance of PHP.

With every passing RFC, the project gets a little more complicated and has no more ability to maintain that complexity.

May 20, 2021: JetBrains speaks to Joe

Avoiding buses

May 4, 2021: Blog post by Joe Watkins

So, What ?

That number, 2. This is not an acceptable bus factor for a project the size and importance of PHP.

With every passing RFC, the project gets a little more complicated and has no more ability to maintain that complexity.

May 20, 2021: JetBrains speaks to Joe

November, 2021: Nikita announces his leave

JetBrains

JetBrains

Does it make sense to look for a replacement

JetBrains

Does it make sense to look for a replacement
and pay for just one developer again?

JetBrains

Does it make sense to look for a replacement
and pay for just one developer again?

Or something more sustainable?...

The founding companies

The founding companies

Acquia
Automattic
Craft CMS
JetBrains
Laravel
PrestaShop
Private Packagist
Symfony
Tideways
Zend by Perforce

We support, advance, and develop the PHP language

The PHP Foundation

“The PHP Foundation's mission is to ensure the long-term prosperity of the PHP language.”

“The PHP Foundation's mission is to ensure the long-term prosperity of the PHP language.”

“The PHP Foundation's mission is to ensure the long-term prosperity of the PHP language.”

“The PHP Foundation's mission is to ensure the long-term prosperity of the PHP language.”

“The PHP Foundation's mission is to ensure the long-term prosperity of the PHP language.”

https://php.foundation

The PHP Foundation

“The language decisions are the matter of the PHP Internals community.”

https://php.foundation

The PHP Foundation

Funding PHP

The PHP Foundation

Funding PHP: Companies

JetBrains	Automattic	Craft CMS	Private Packagist	Cybozu	Tideways	Livesport s.r.o.	Acquia
Symfony Corp	Mercari Inc.	shopware AG	Aternos GmbH	Ardennes-étape	Zend by Perforce	Moodle	OP.GG
pixiv Inc.	Oro	Aligent Consulting	EC-CUBE	Spryker	Polcode	Laravel	BASE, Inc.

The PHP Foundation

Budget is public

https://opencollective.com/phpfoundation#category-BUDGET

$ TODAY'S BALANCE	TOTAL RAISED	TOTAL DISBURSED	ESTIMATED ANNUAL BUDGET
$764,163.94 USD	**$1,215,558.80** USD	**$451,394.86** USD	**$516,044.16** USD

Expenses

All time

EXPENSES PAID	AMOUNT DISBURSED
138	$450,559.05

TAGS	# OF EXPENSES	AMOUNT (USD)
Core Developer	131	$444,725.70
No Tag	4	$5,770.87

Contributions

All time

CONTRIBUTIONS RECEIVED	AMOUNT COLLECTED
1813	$903,533.59

TIERS	# OF CONTRIBUTIONS	AMOUNT (USD)
Recurring	948	$488,181.61
One-Time	865	$415,351.98

The team

The PHP Foundation

Team 2021-2023

6 Developers

The PHP Foundation

Team 2021-2023

6 Developers

Board

The PHP Foundation

Team 2021-2023

6 Developers

Board

Community

The PHP Foundation

Team 2024

10 Developers

Board

Community

The PHP Foundation

Who pays for PHP?

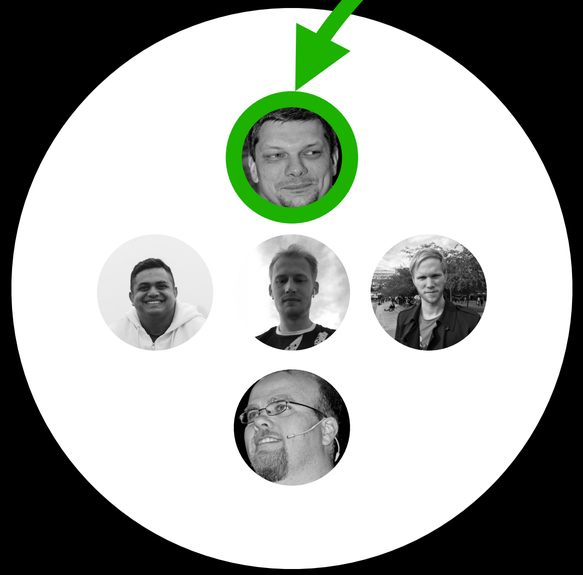
Paid by the Foundation

10 Developers

Paid by JetBrains

Board

Paid by Zend



Community

Language Impact

The PHP Foundation

Language Impact

Apr 1, 2022 – Dec 31, 2023

Contributions to master, line counts have been omitted because commit count exceeds 10,000.

iluuu1994	1,033 commits	#1
nielsdos	936 commits	#2
dstogov	566 commits	#3
Girgias	442 commits	#4
derickr	394 commits	#5
cmb69	354 commits	#6
bukka	351 commits	#7
devnexen	317 commits	#8
kocsismate	288 commits	#9
alexdowad	231 commits	#10
arnaud-lb	190 commits	#11
TimWolla	164 commits	#12

The PHP Foundation

Language Impact

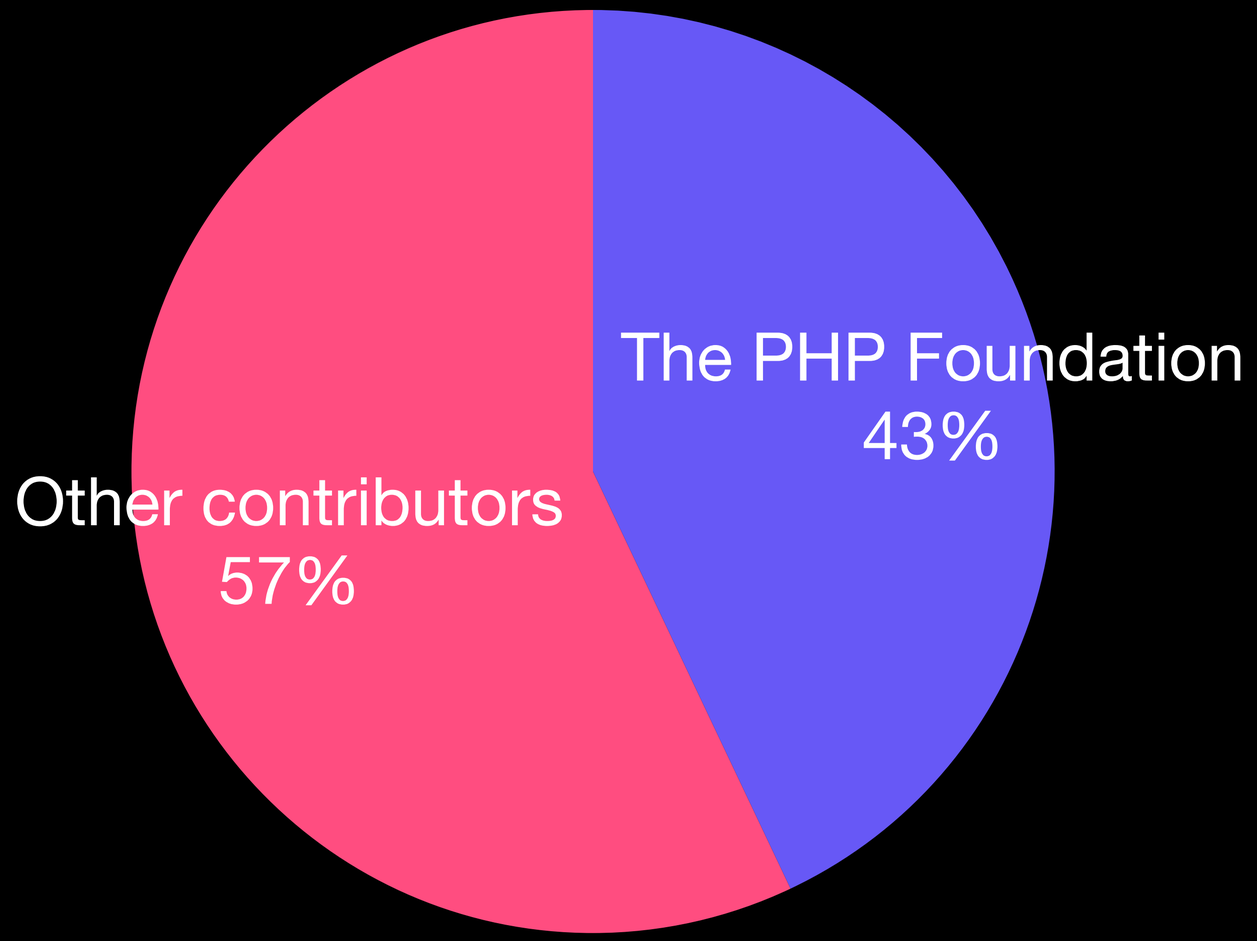
Apr 1, 2022 – Dec 31, 2023

Contributions to master, line counts have been omitted because commit count exceeds 10,000.

iluuu1994	1,033 commits	#1
nielsdos	936 commits	#2
dstogov	566 commits	#3
Girgias	442 commits	#4
derickr	394 commits	#5
cmb69	354 commits	#6
bukka	351 commits	#7
devnexen	317 commits	#8
kocsismate	288 commits	#9
alexdowad	231 commits	#10
arnaud-lb	190 commits	#11
TimWolla	164 commits	#12

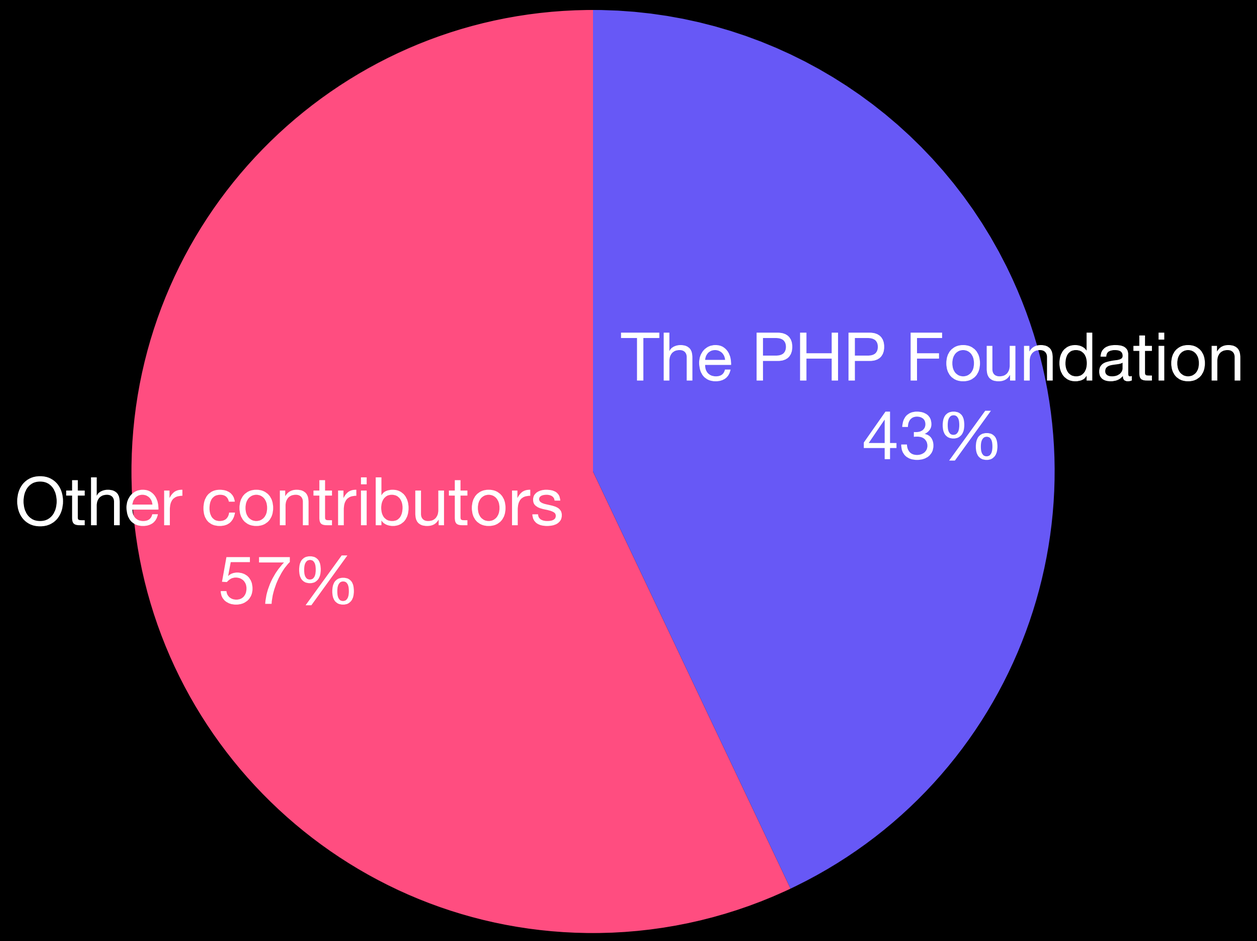
The PHP Foundation

Language Impact



The PHP Foundation

Language Impact



Share of commits in PHP repository

What devs are working on?

(fancy highlights only)

Gina Peter Banyard

- 🚧 New core autoloading mechanism with support for function autoloading

The PHP Foundation

Ilija Tovilo

- 🚧 Property hooks
- 🚧 Pattern matching

The PHP Foundation

Arnaud Le Blanc

- 🚧 Lazy objects
- 🚧 Generics

The PHP Foundation

Máté Kocsis

- 🚧 Clone with

The PHP Foundation

Jakub Zelenka

- 🚧 JSONSchema support

The PHP Foundation

Derick Rethans

- 🚧 New tool for installing extensions

The PHP Foundation

Saki Takamachi

Shivam Mathur

James Titcumb

David Carlier

The PHP Foundation

The PHP Foundation

Language Impact: maintenance

The PHP Foundation

Language Impact: maintenance

- Triaging issues and PR reviews
- Performance improvements
- Documentation
- Servers maintenance
- Build infrastructure

The PHP Foundation

Language Impact: security

The PHP Foundation

Language Impact: security

Security Audit for PHP

The PHP Foundation

Language Impact: security

Security Audit for PHP

Commissioned by:

Sovereign Tech Fund

The PHP Foundation

Language Impact: security

Security Audit for PHP

Commissioned by:

Sovereign Tech Fund

Partnered with:

Goals 2024

The PHP Foundation

Goals 2024

The PHP Foundation

Goals 2024

- More financial stability

The PHP Foundation

Goals 2024

- More financial stability
- Shape the roadmap for PHP

The PHP Foundation

Goals 2024

- More financial stability
- Shape the roadmap for PHP Foundation

How do we shape the roadmap?

1. Advisory board

1. Advisory board

Ardennes-étape
Aternos GmbH
Composer
CraftCMS
JetBrains
Laravel
Les-Tilleuls.coop
Moodle
PrestaShop
Shopware
Symfony
Tideways
WordPress
Zend by Perforce

1. Advisory board
2. Analyse competitors

Rust

Kotlin

Go

JavaScript

Rust

Rust

- Official toolchain management with `rustup`

Rust

- Official toolchain management with `rustup`

`rustc` — Compiler.
`cargo` — Package manager and build tool.
`clippy` — Lint tool that with extra checks.
`rustfmt` — Tool for formatting code.
`rust-docs` — Local copy of the documentation.
`rust-analyzer` — Language server for editors.

Rust

- Official toolchain management with `rustup`

Kotlin

Go

JavaScript

Go

Go

- Backward compatibility

“There will not be a Go 2 that breaks Go 1 programs.”

“...prioritizing compatibility was the most important design decision we made...”

“Boring is good. Boring is stable. Boring means being able to focus on your work, not on what’s different about Go.”

Rust

- Official toolchain management with `rustup`

Kotlin

Go

- Backward compatibility

JavaScript

Kotlin

Kotlin

- Classes are final by default 😱

```
class Foo { }
class Bar(): Foo() { }
```

```
This type is final, so it cannot be inherited from
```

Kotlin

- Classes are final by default 😱

```
class Foo { }

class Bar(): Foo() { }
```

```
This type is final, so it cannot be inherited from
```

Kotlin

- Classes are final by default 😱
- Class extension (Laravel macro)

```
class Foo { }

fun Foo.myMethod() {
    println("Hello, Laracon!")
}
```

Rust

- Official toolchain management with `rustup`

Kotlin

- Classes are final by default 😱
- Class extension (Laravel macro)

Go

- Backward compatibility

JavaScript

JavaScript

Node.js' stability index

Stability: 0 - Deprecated. The feature may emit warnings. Backward compatibility is not guaranteed.

Stability: 1 - Experimental. Non-backward compatible changes or removal may occur in any future release. Use of the feature is not recommended in production environments.

Stability: 2 - Stable. Compatibility with the Composer ecosystem is a high priority.

Stability: 3 - Legacy. Although this feature is unlikely to be removed, it is no longer actively maintained, and other alternatives are available.

JavaScript

- Experimental features

Rust

- Official toolchain management with `rustup`

Kotlin

- Classes are final by default 😱
- Class extension (Laravel macro)

Go

- Backward compatibility

JavaScript

- Experimental features

1. Advisory board
2. Analyse competitors

1. Advisory board
2. Analyse competitors
3. **Measure**

Usage statistics for PHP

Usage statistics for PHP

- Telemetry in PHP 🤡

Usage statistics for PHP

- Telemetry in PHP 🤡
- Composer plugin 🤔

Usage statistics for PHP

- Telemetry in PHP 🤡
- Composer plugin 🤔

```
github.com/pronskiy/php-stats

$ php phpstats ./test_projects/
 17/17 [▓▓▓
```

Usage statistics for PHP

- Telemetry in PHP 🤡
- Composer plugin 🤔

Usage statistics for PHP

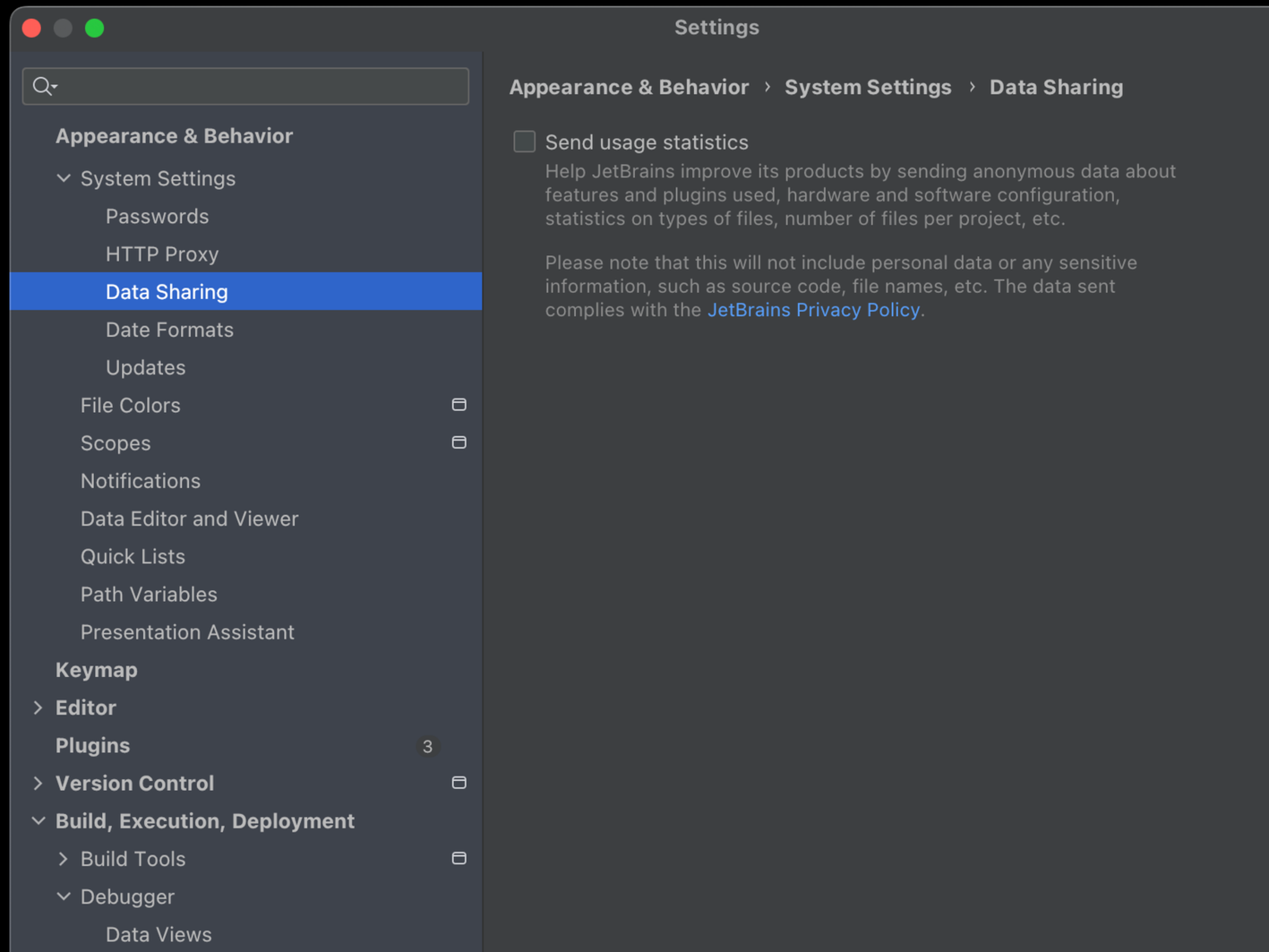
- Telemetry in PHP 🤡
- Composer plugin 🤔
- Scan OSS projects 👌

Usage statistics for PHP

- Telemetry in PHP 🤡
- Composer plugin 🤔
- Scan OSS projects 👌
- PhpStorm 👈

Usage statistics for PHP

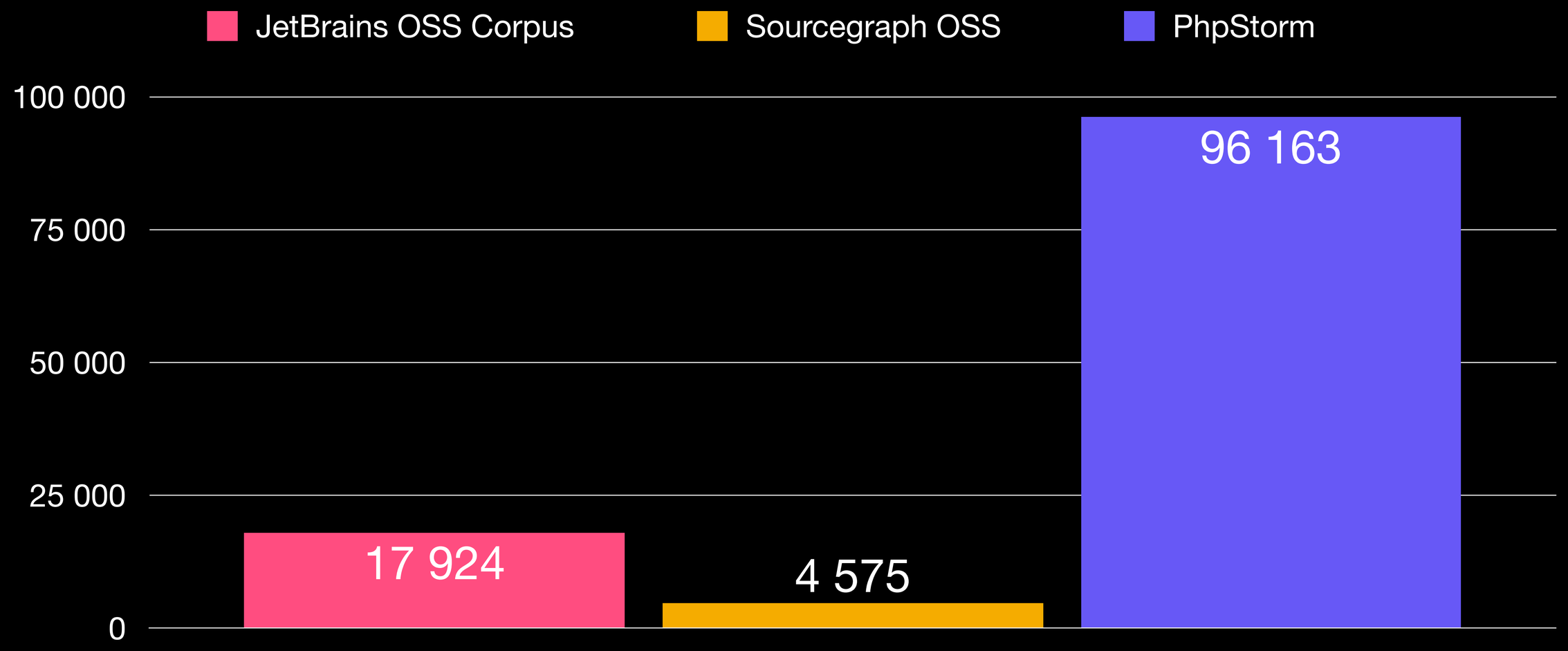
- Telemetry in PHP 🤡
- Composer plugin 🤔
- Scan OSS projects 👌
- PhpStorm 👈



How many projects use a certain new feature from PHP 8.*?

Repositories analysed

Repositories analysed



PhpStorm Analytics

Event statistics

My reports

Product

PhpStorm

Version

2023.2 major, Release

Group

Can't find?

PHP Configuration

Event (9 available)

PHP Feature Usage

Date range

Last 4 weeks

Columns and slicing

Selected columns

PHP Feature

PHP Feature Usage in PhpStorm per projects

Quick guide

28 Dec 2023 — 24 Jan 2024, External active users

Filter report

Sample size: 96 163 projects

#	PHP Feature	Ratio of Projects	Projects
1	SHORT_ARRAY_SYNTAX	66.1%	63 577
2	RETURN_TYPES	45.2%	43 499
3	COALESCE_OPERATOR	44.2%	42 497
4	CLASS_NAME_CONST	44.2%	42 459
5	TRAILING_COMMA_IN_FUNCTION_CALL	44.1%	42 431
6	ARRAY_DEREFERENCING	43.9%	42 199
7	SCALAR_TYPE_HINTS	42.6%	40 937
8	RETURN_VOID	36.9%	35 533

Regex search

https://sourcegraph.com/search

```
context:global file:\.php$ count:all repohascommitafter:01/08/2020 \#\[\\?AllowDynamicProperties\]
```

Aa .* []

6.1k results in 60.12s

Actions

Filters

WordPress/WordPress › wp-includes/class-wp.php ★ 18.2k

```
7   */
8   #[AllowDynamicProperties]
9   class WP {
```

WordPress/WordPress › wp-includes/class-wpdb.php ★ 18.2k

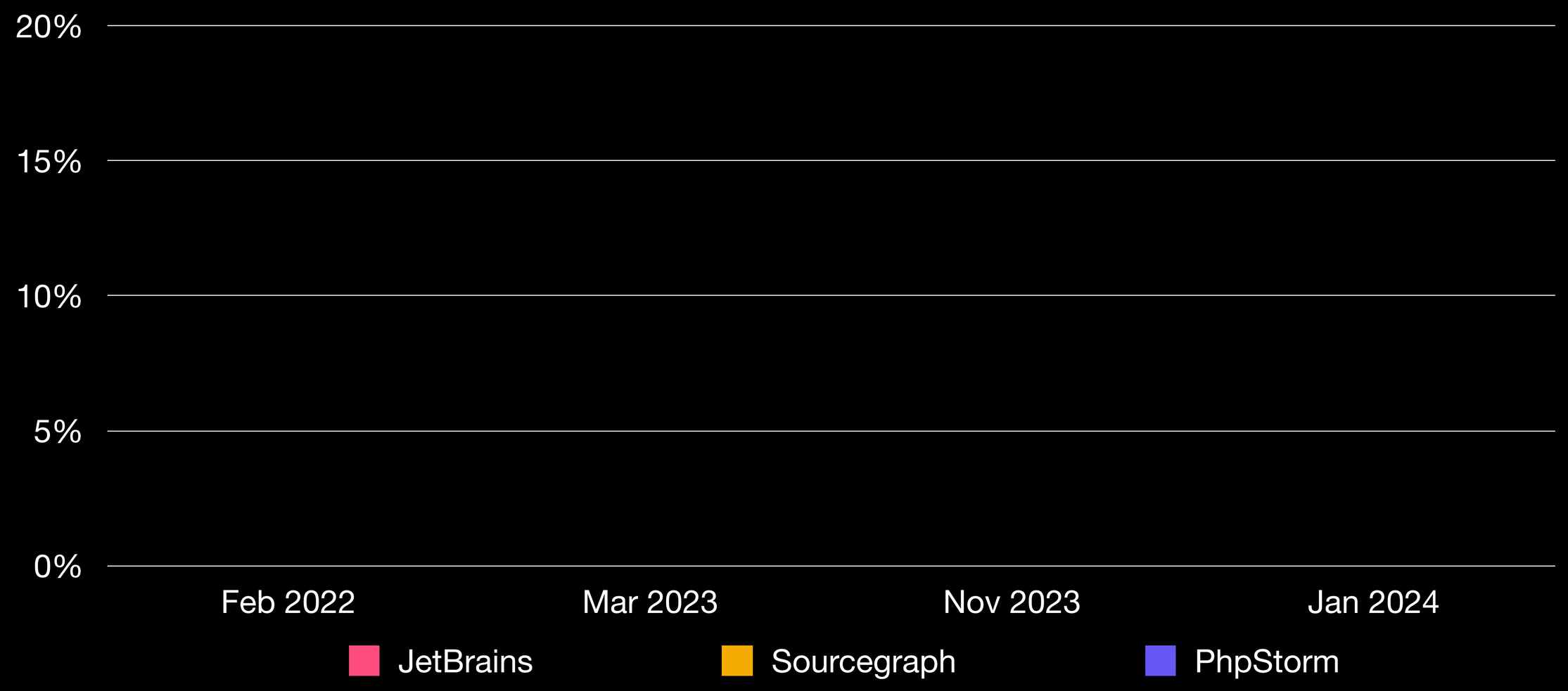
```
53  */
54  #[AllowDynamicProperties]
55  class wpdb {
```

WordPress/WordPress › wp-includes/class-wp-http.php ★ 18.2k

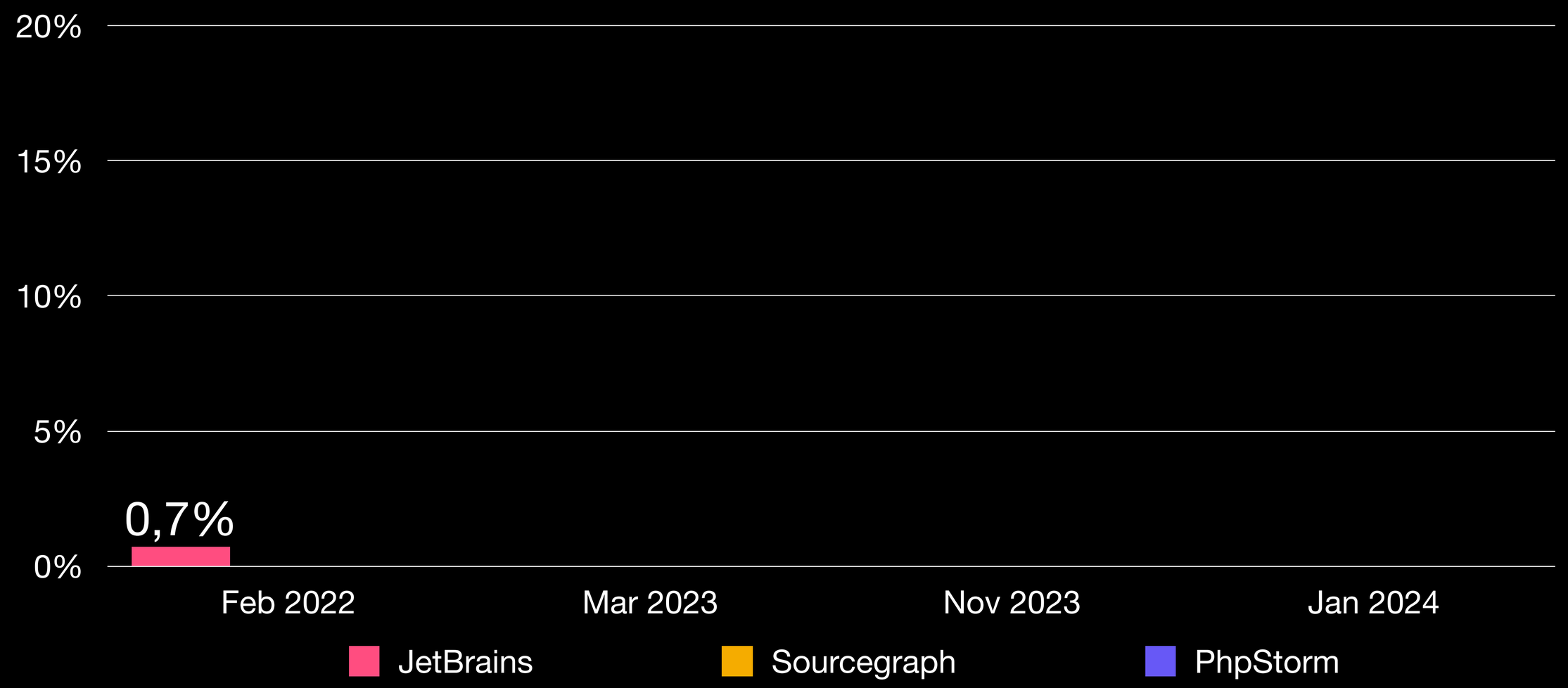
How many projects use:

How many projects use: Enums

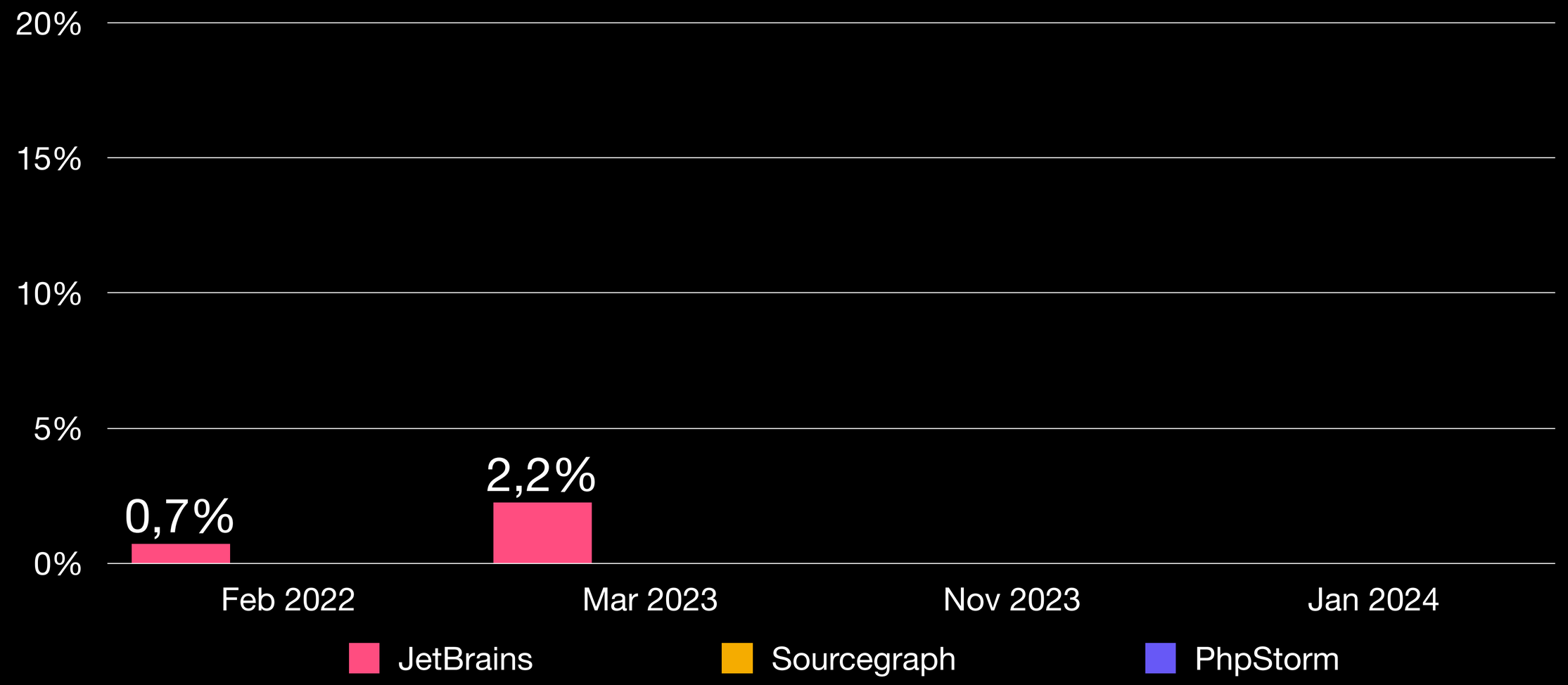
How many projects use: Enums



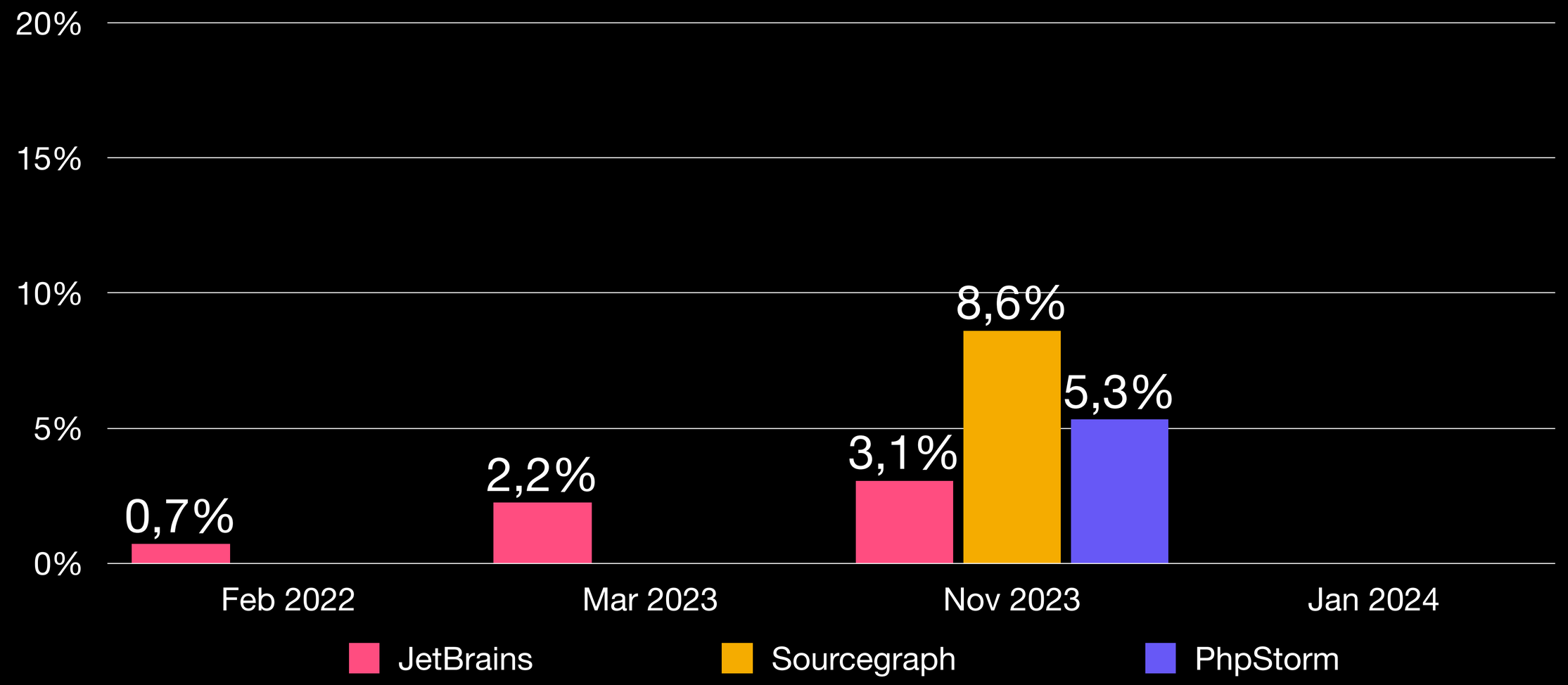
How many projects use: Enums



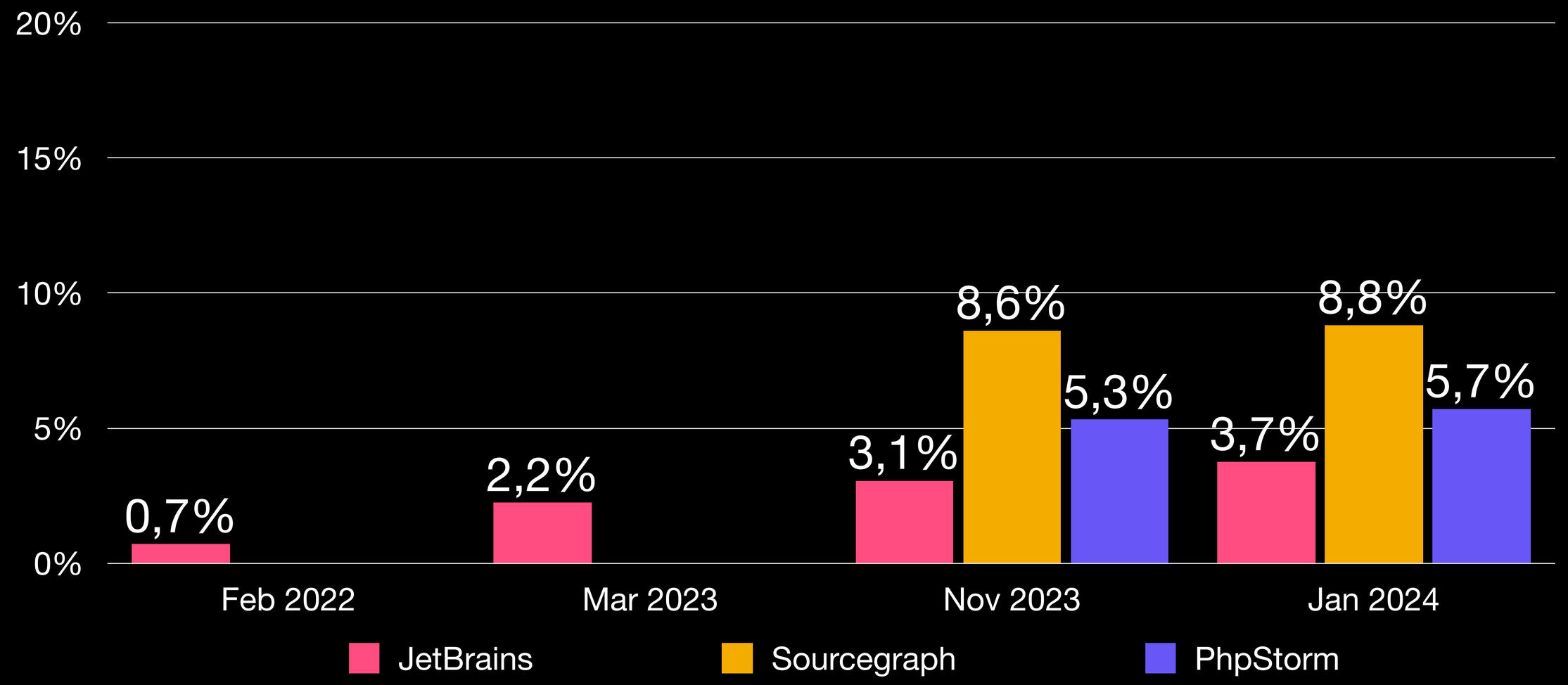
How many projects use: Enums



How many projects use: Enums

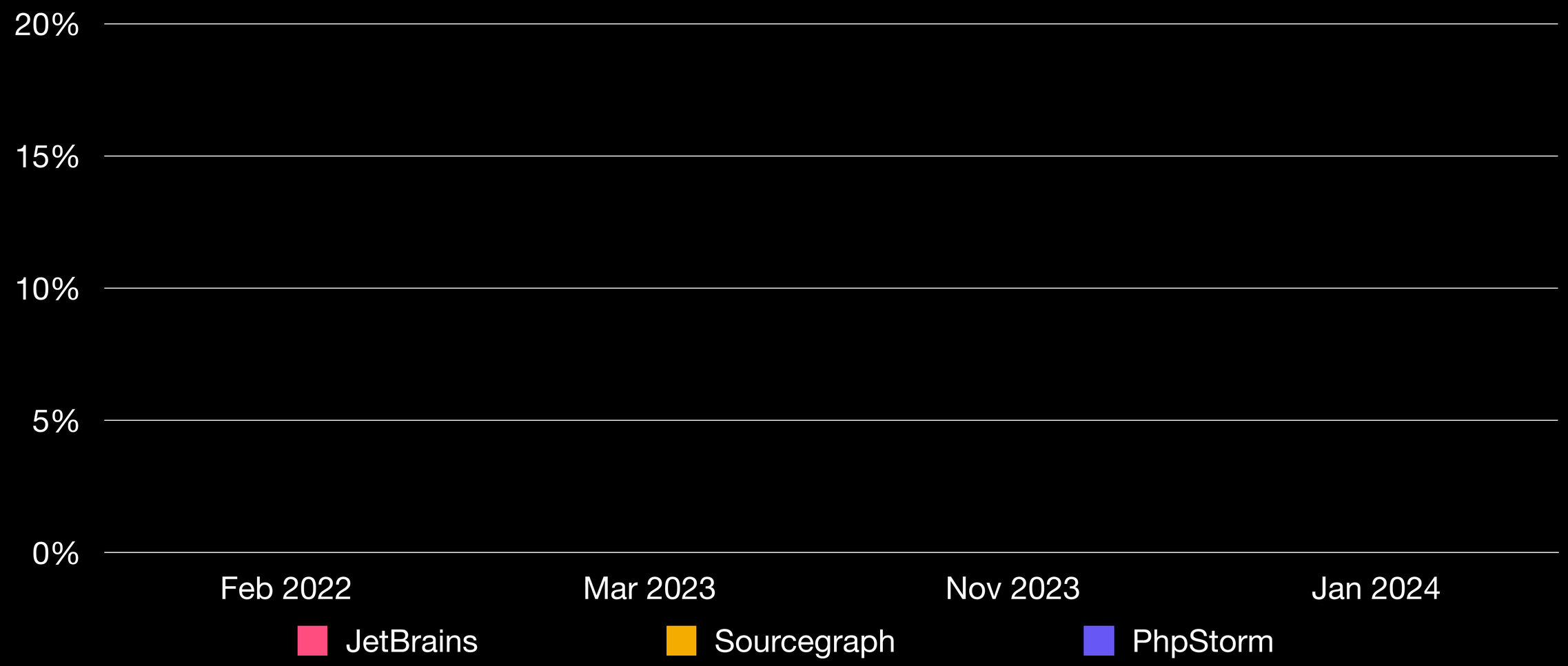


How many projects use: Enums

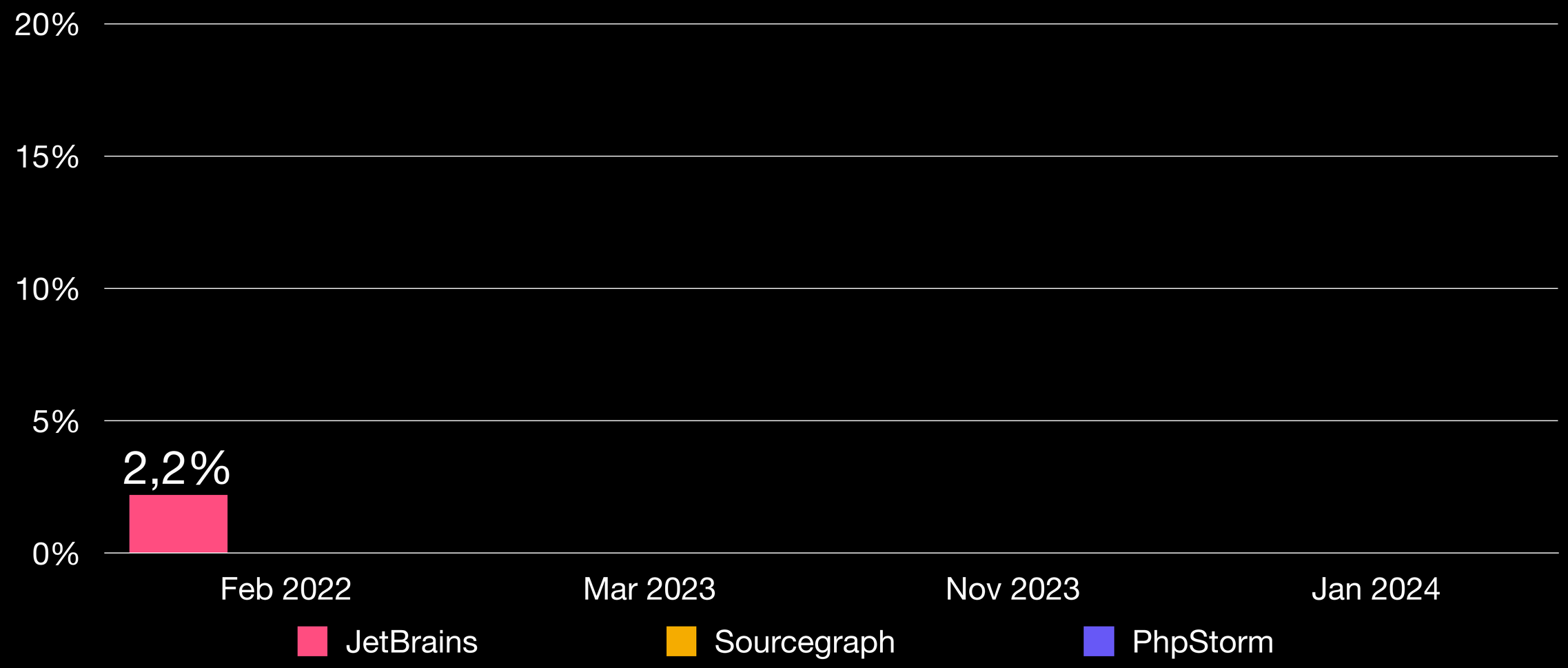


How many projects use: Readonly

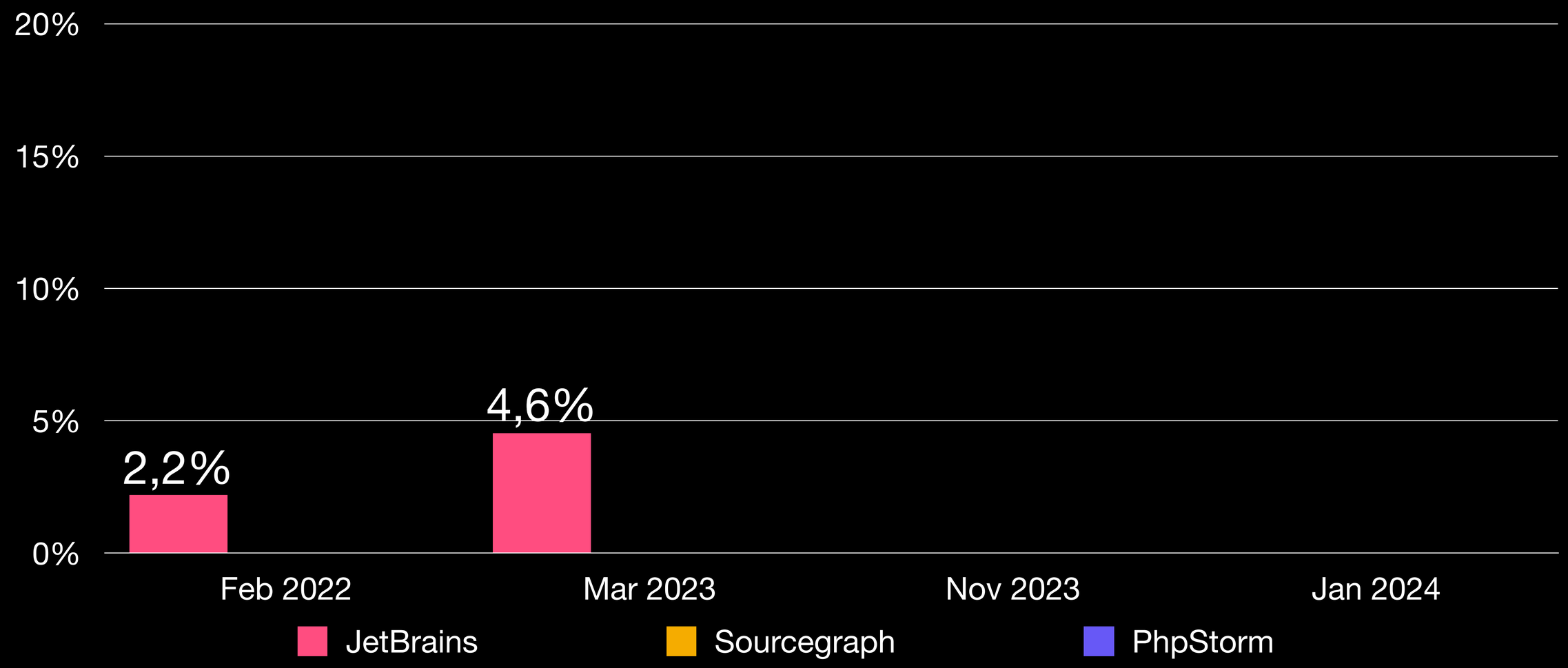
How many projects use: Readonly



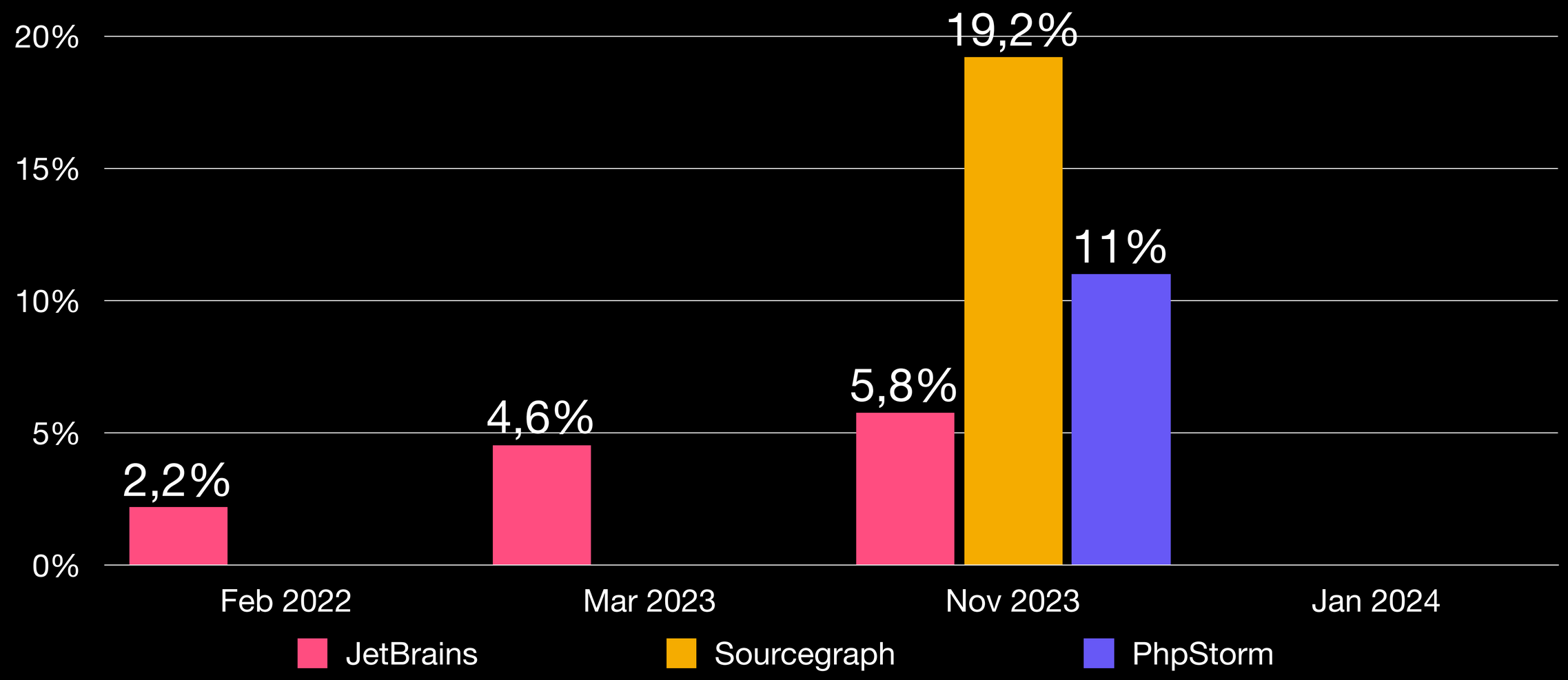
How many projects use: Readonly



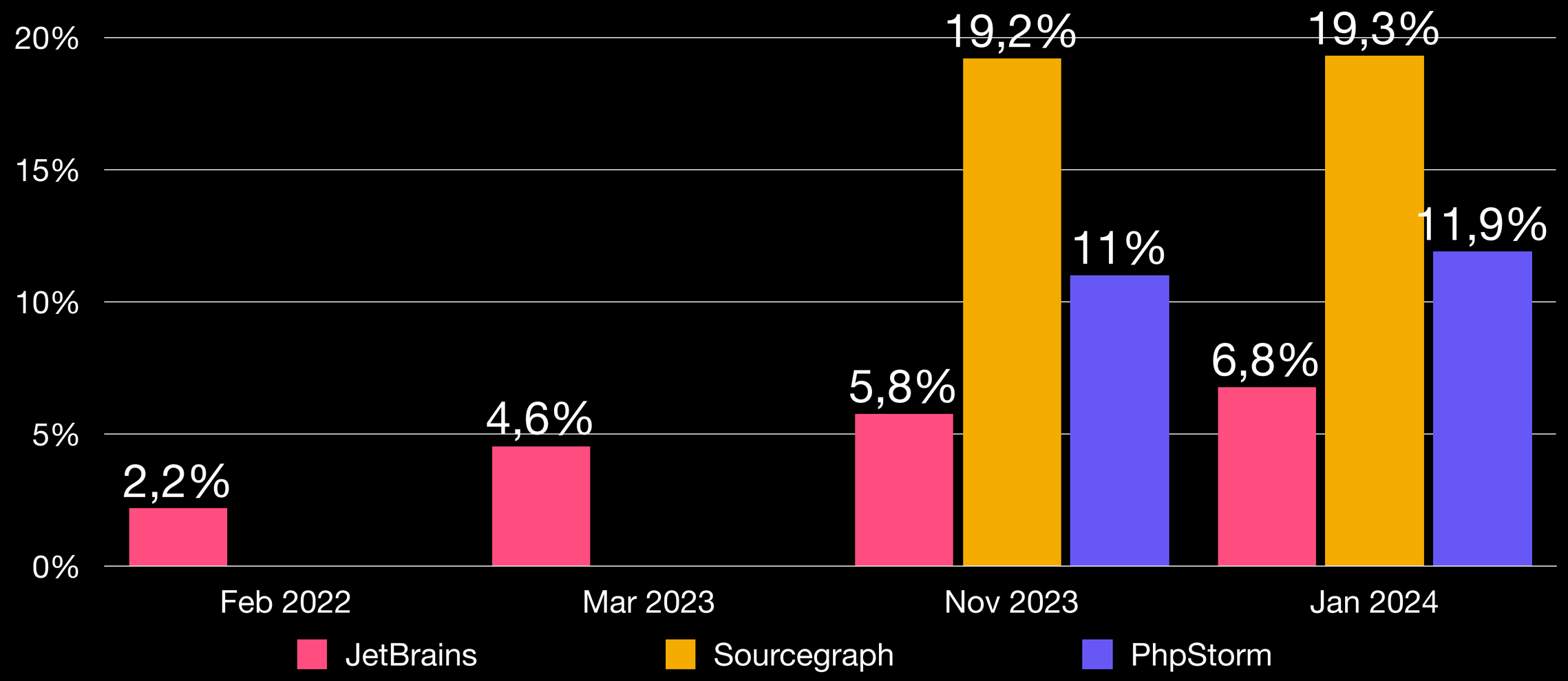
How many projects use: Readonly



How many projects use: Readonly

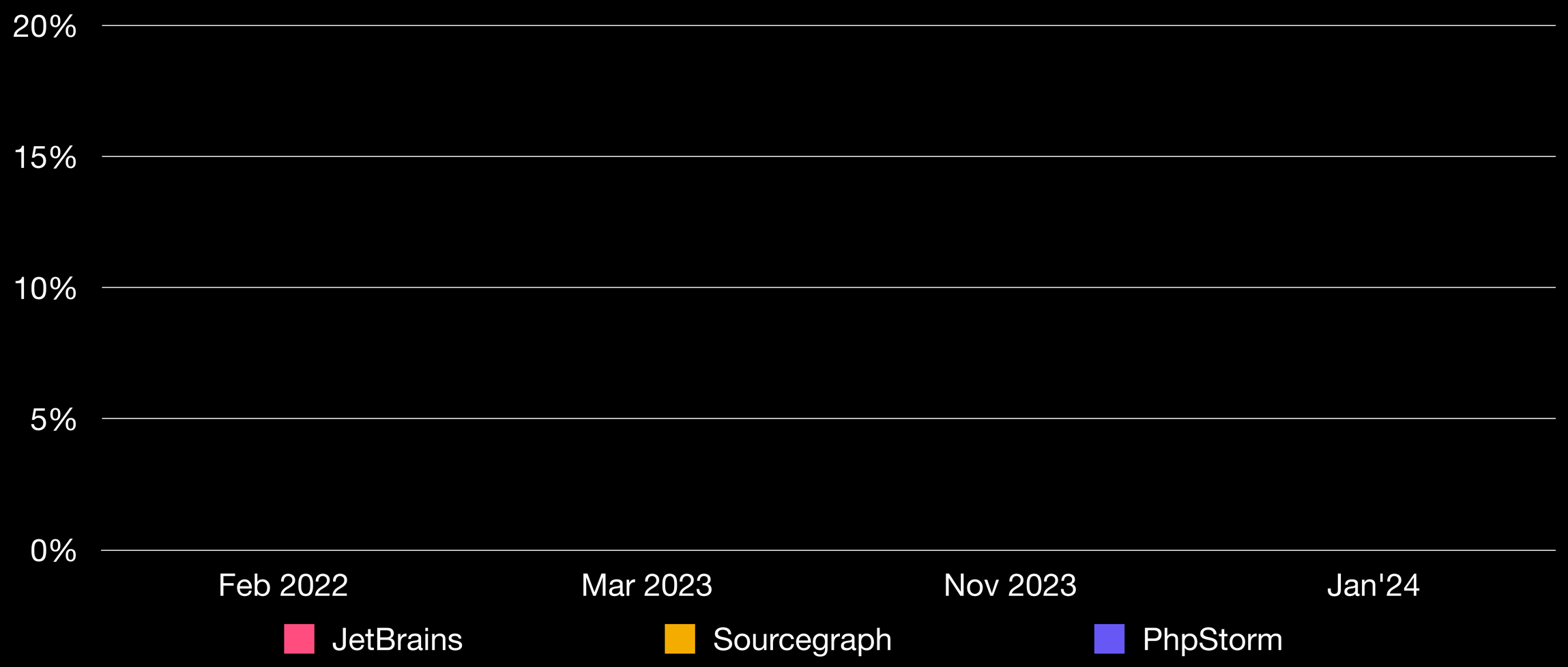


How many projects use: Readonly

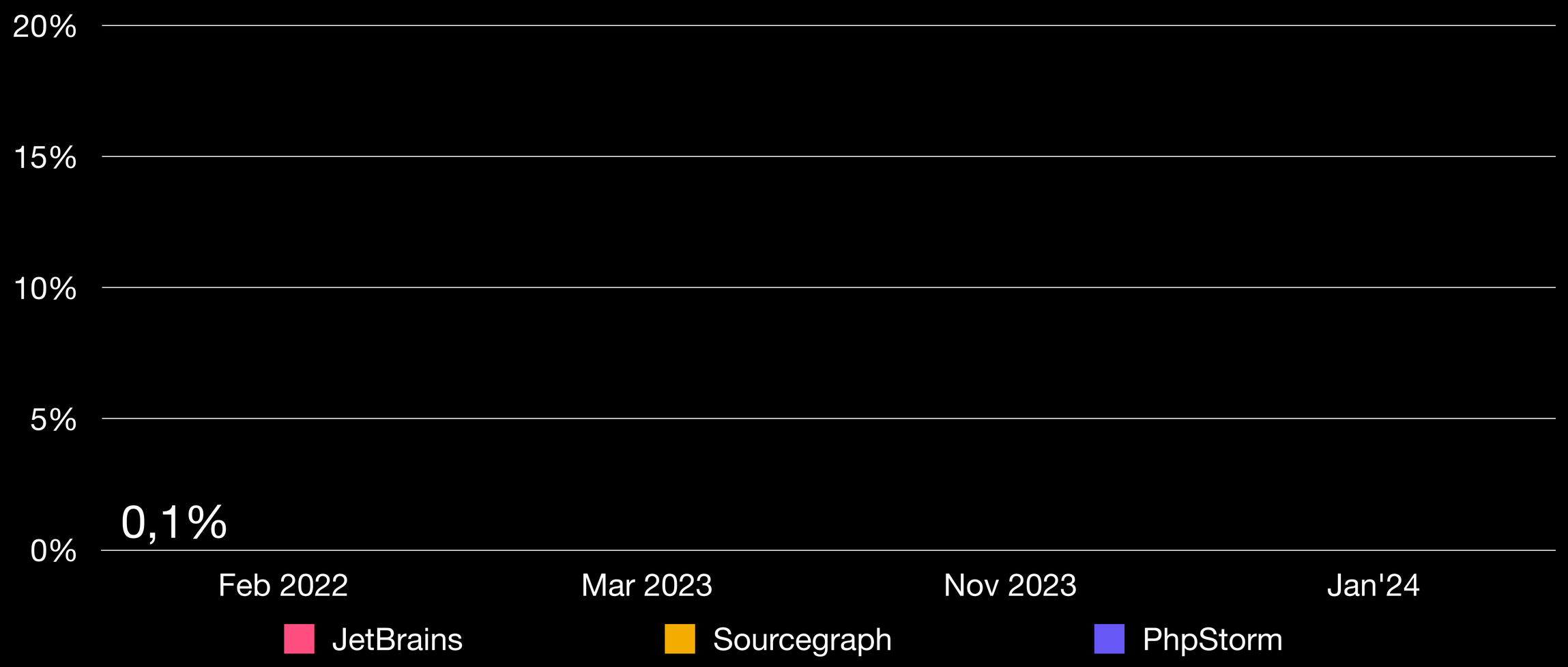


How many projects use: Fibers

How many projects use: Fibers

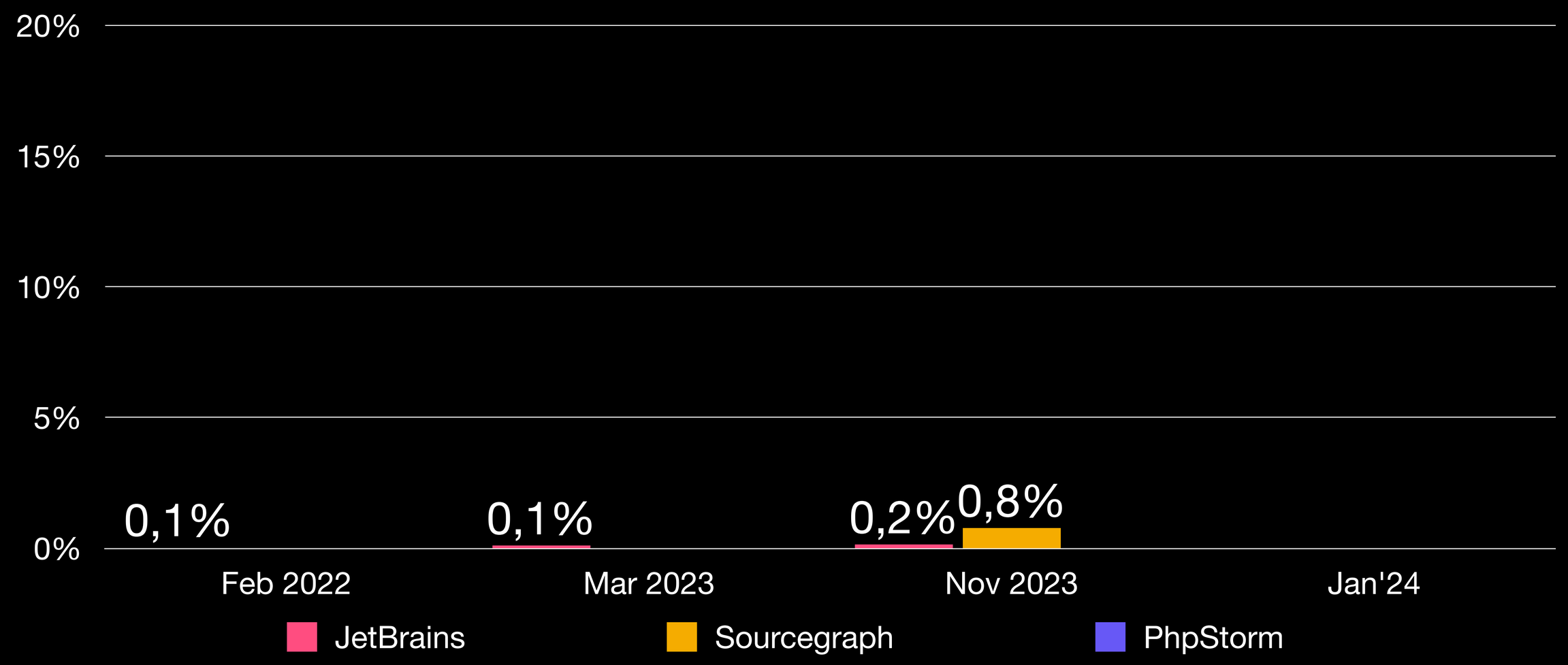


How many projects use: Fibers

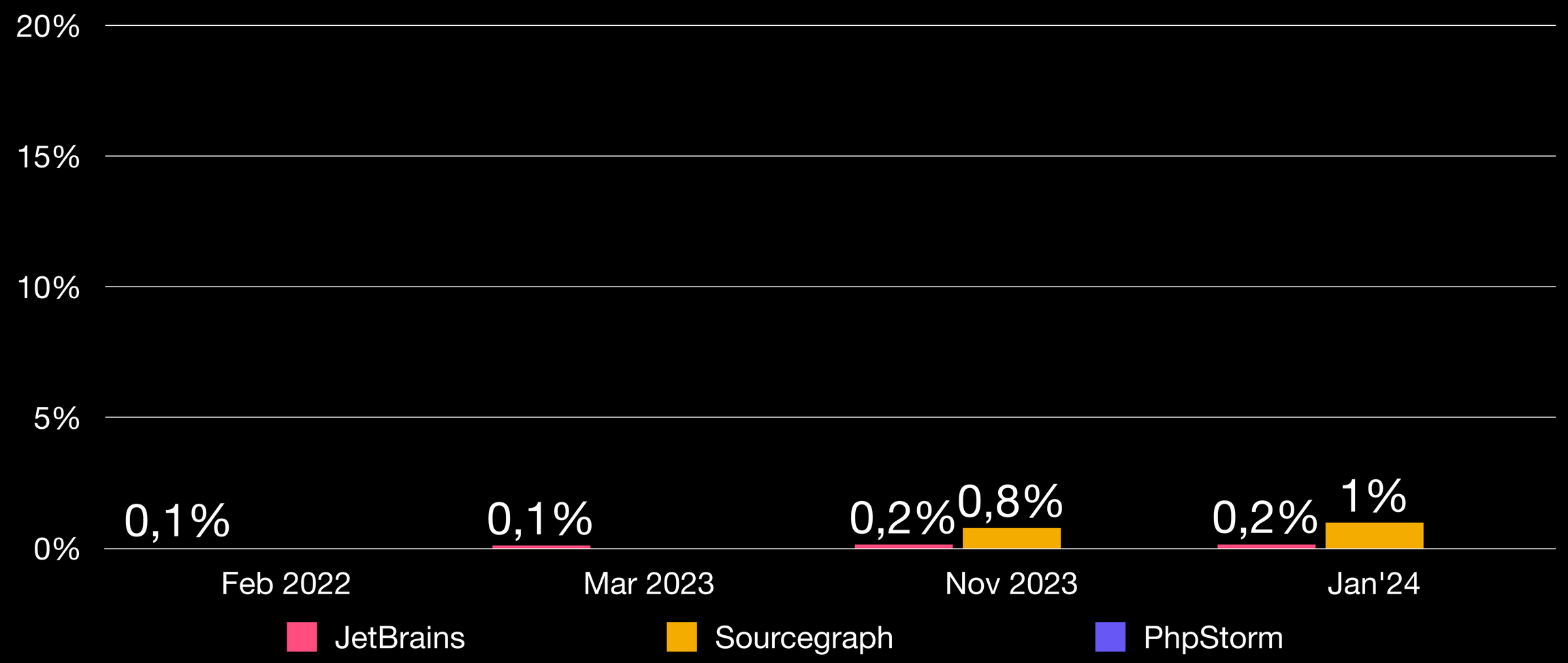


How many projects use: Fibers

How many projects use: Fibers

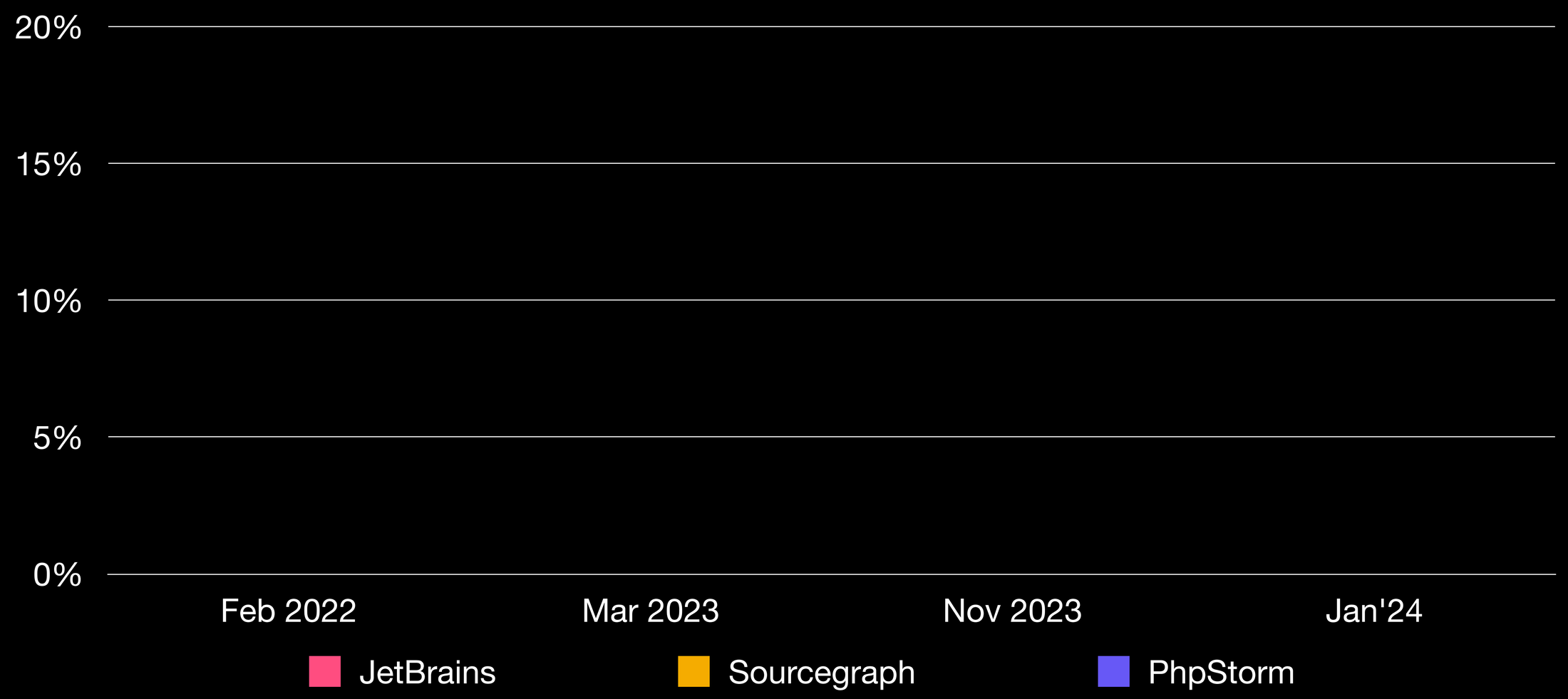


How many projects use: Fibers

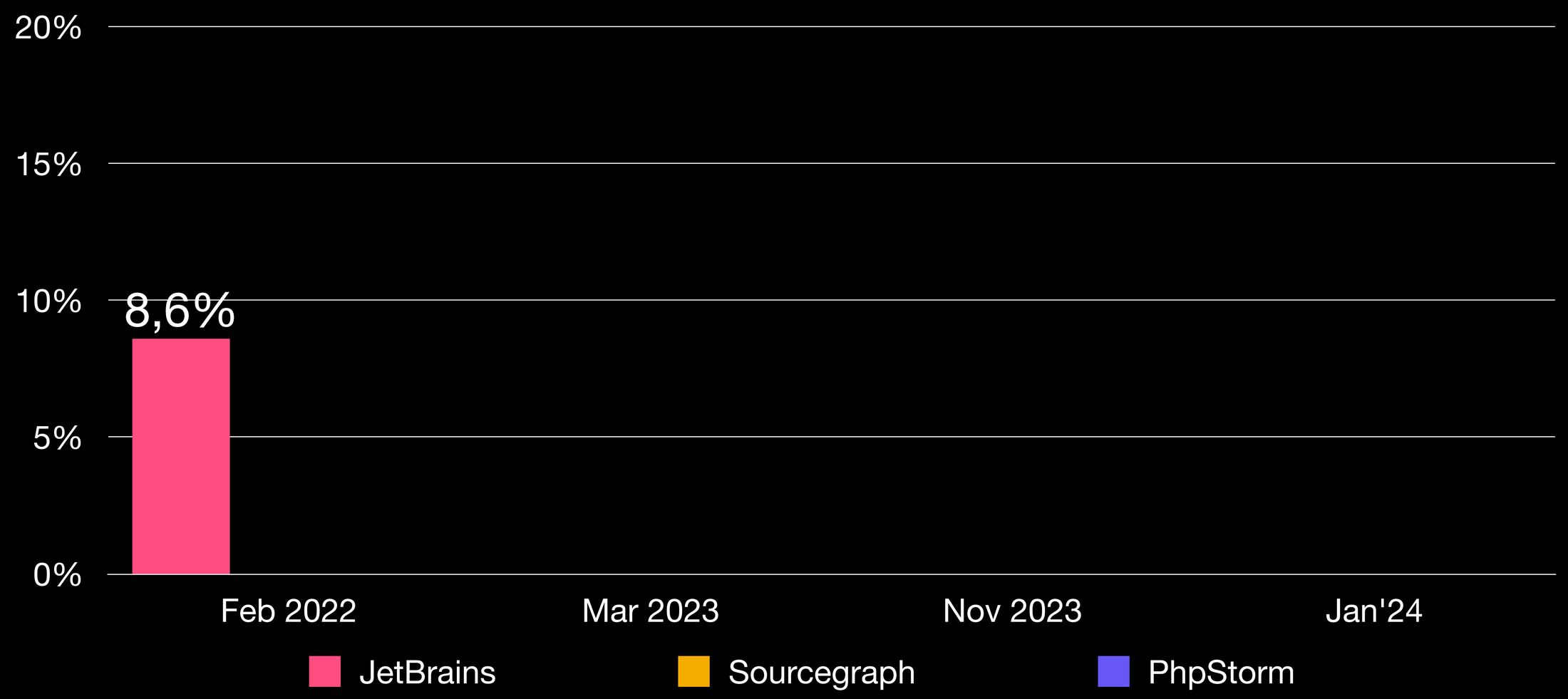


How many projects use: `#[Attributes]`

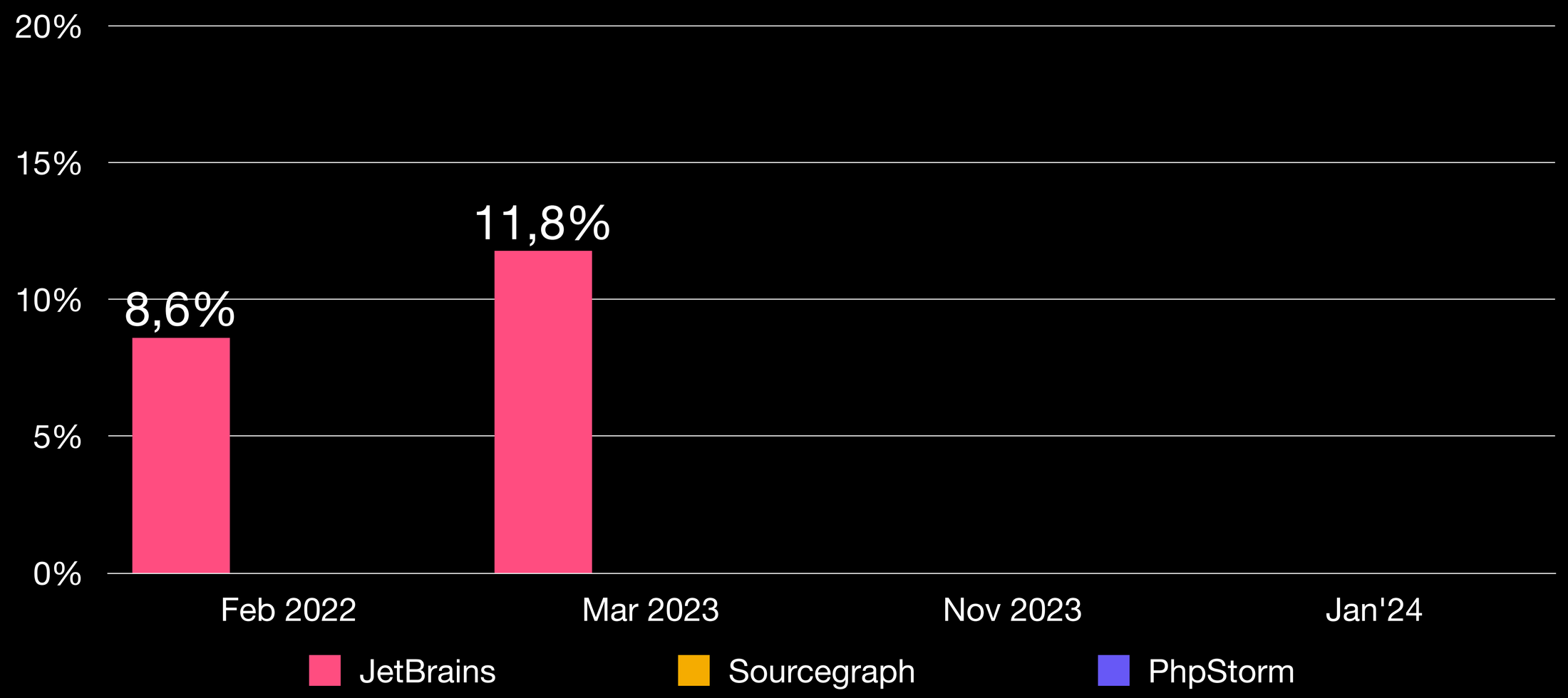
How many projects use: `#[Attributes]`



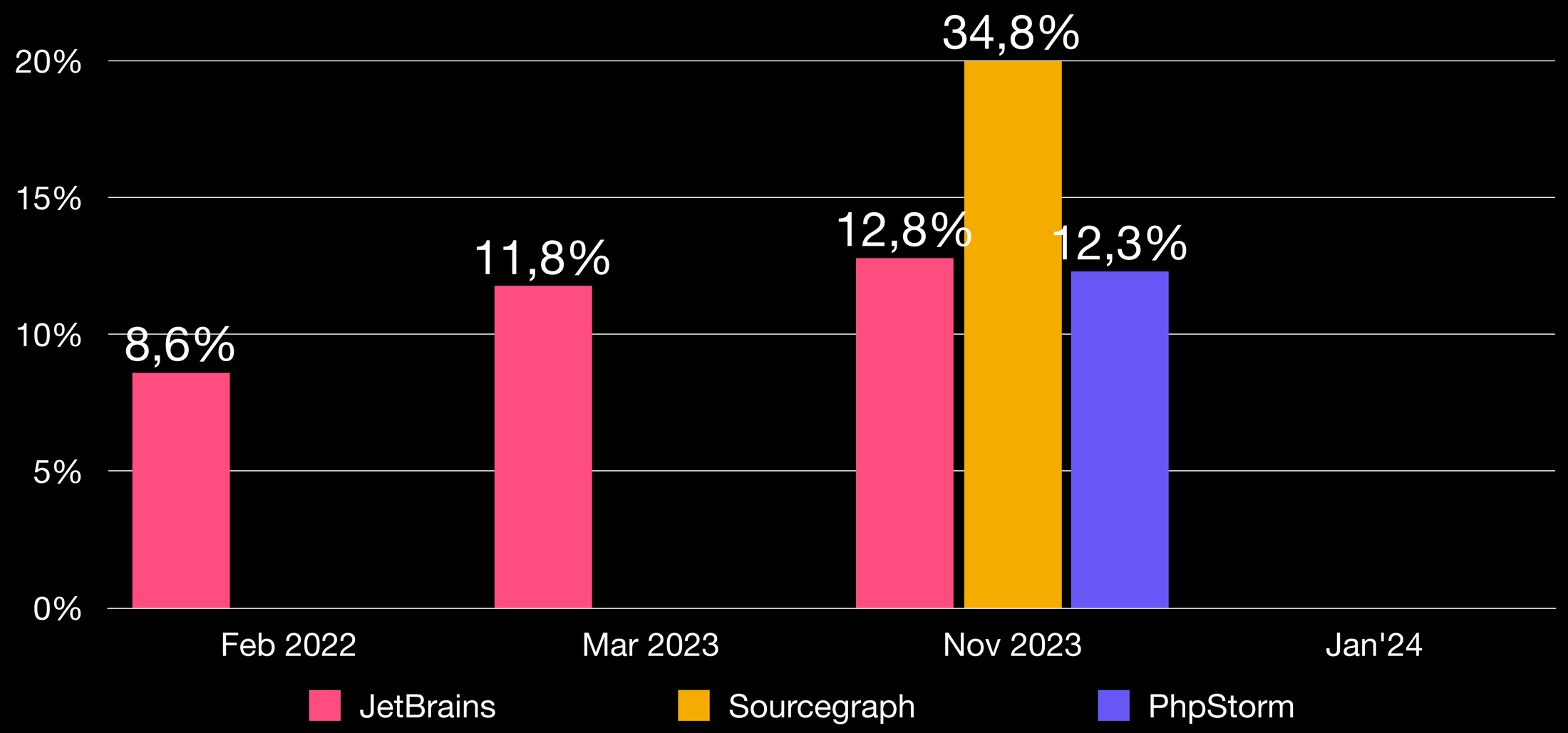
How many projects use: `#[Attributes]`



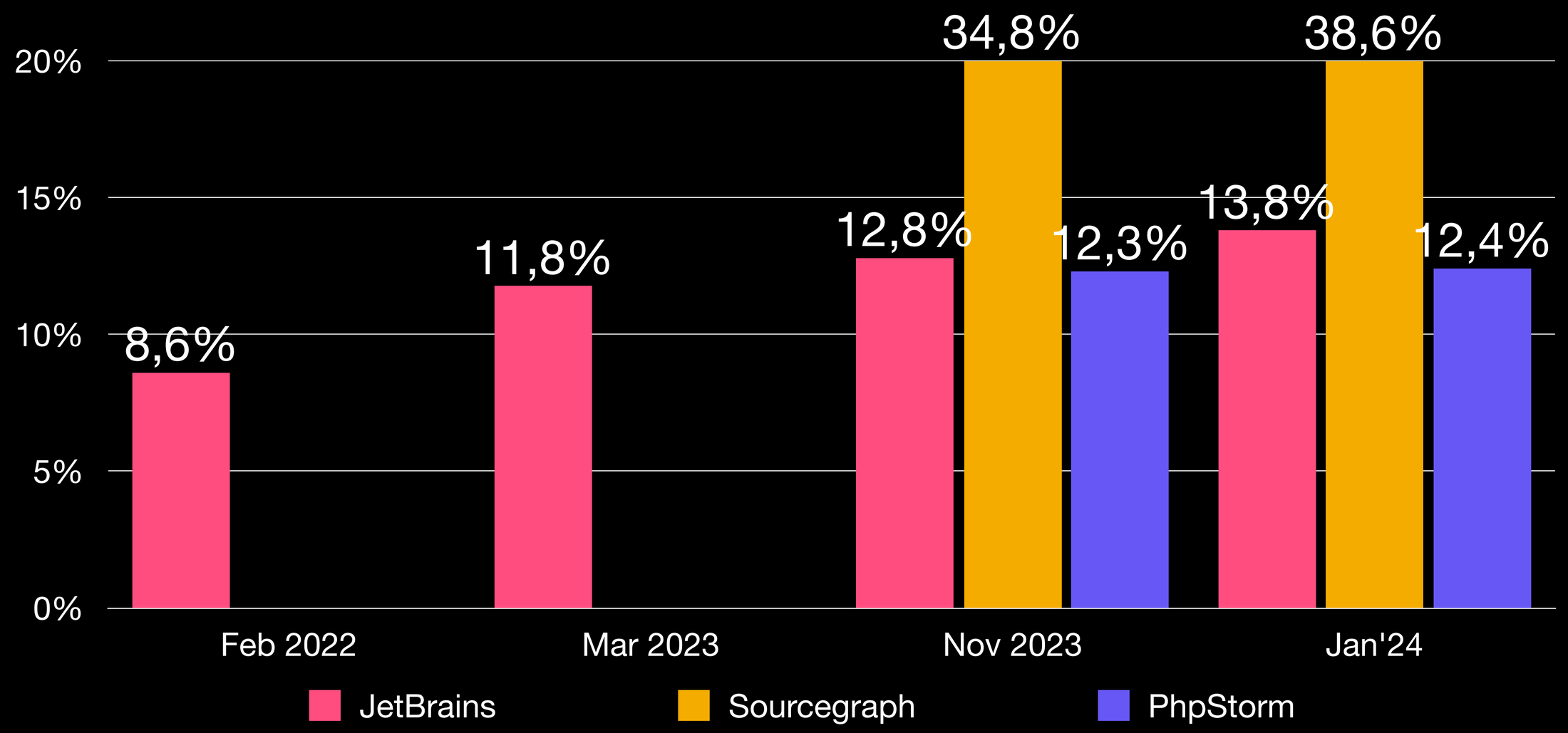
How many projects use: `#[Attributes]`



How many projects use: `#[Attributes]`



How many projects use: `#[Attributes]`



1. Advisory board
2. Analyse competitors
3. **Measure**

1. Advisory board
2. Analyse competitors
3. Measure
4. **Brainstorm**

What features do you need?

~~What features do you need?~~

Recall a recent time where PHP didn’t meet your expectations or created a significant hurdle? How did you overcome this issue?

It's always a tradeoff

vs

It's always a tradeoff

Clever

Readable

vs

It's always a tradeoff

Clever

Readable

```
$numbers = [...];
echo array_reduce(
    array_map(fn($n) => $n * $n, $numbers),
    fn($carry, $item) => $carry + $item,
    initial: 0
);
```

vs

It's always a tradeoff

Clever

Readable

```php
$numbers = [...];
echo array_reduce(
    array_map(fn($n) => $n * $n, $numbers),
    fn($carry, $item) => $carry + $item,
    initial: 0
);
```

vs

```php
$numbers = [...];
$sum = 0;
foreach ($numbers as $number) {
    $sum += $number * $number;
}
echo $sum;
```

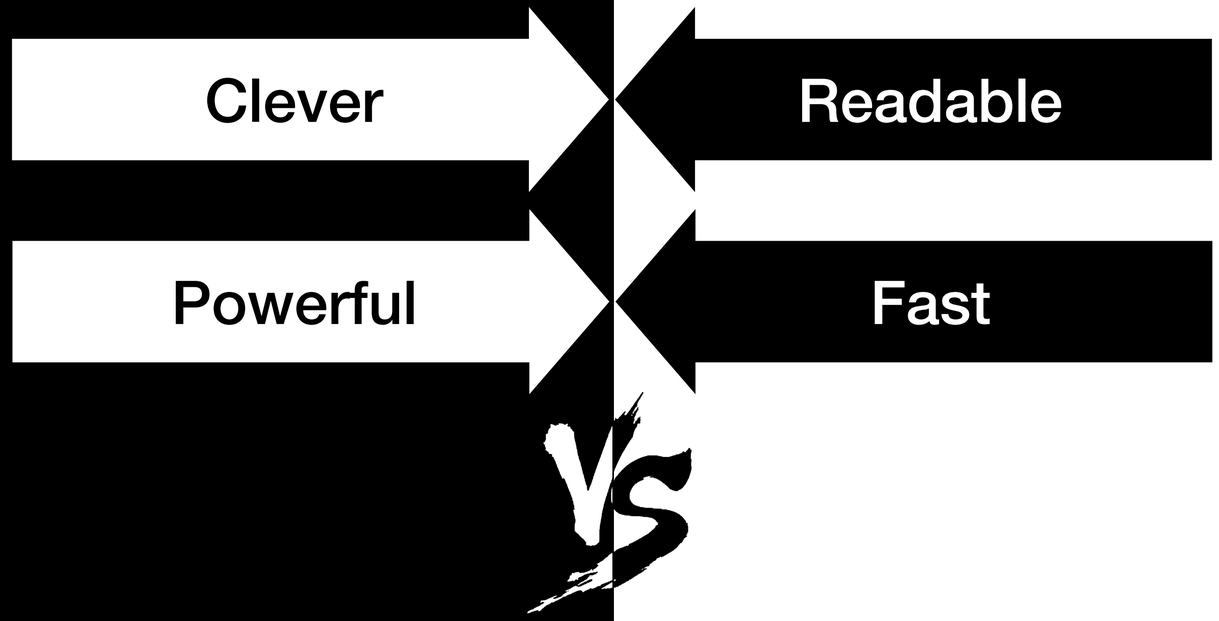
It's always a tradeoff

Clever

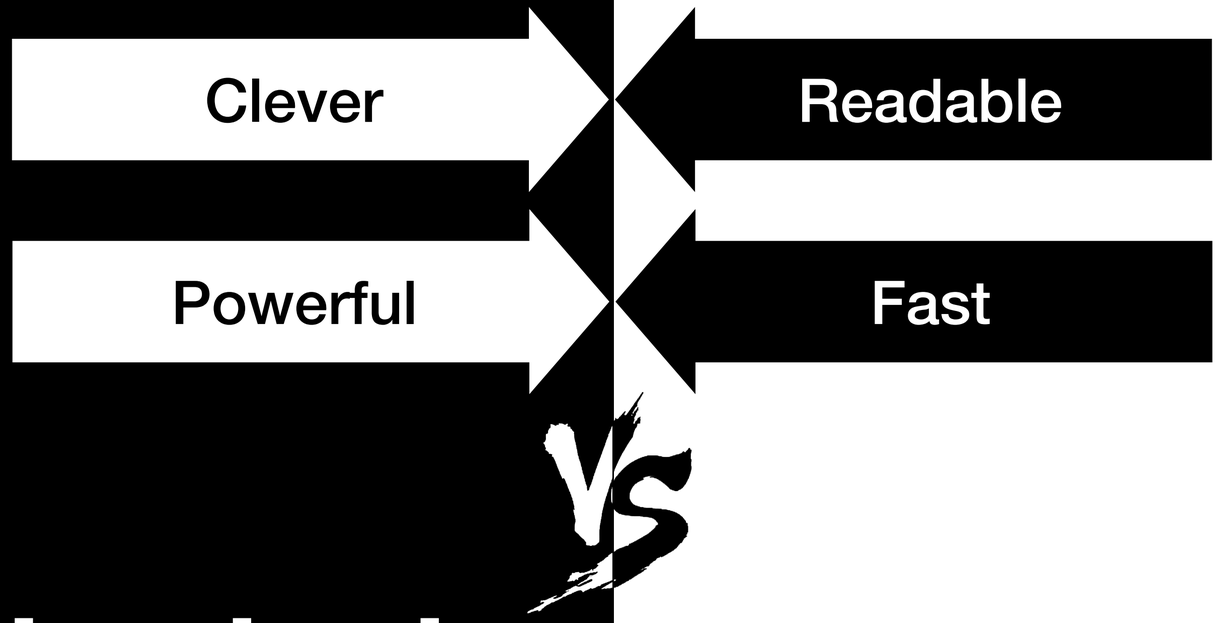
Readable

vs

It's always a tradeoff

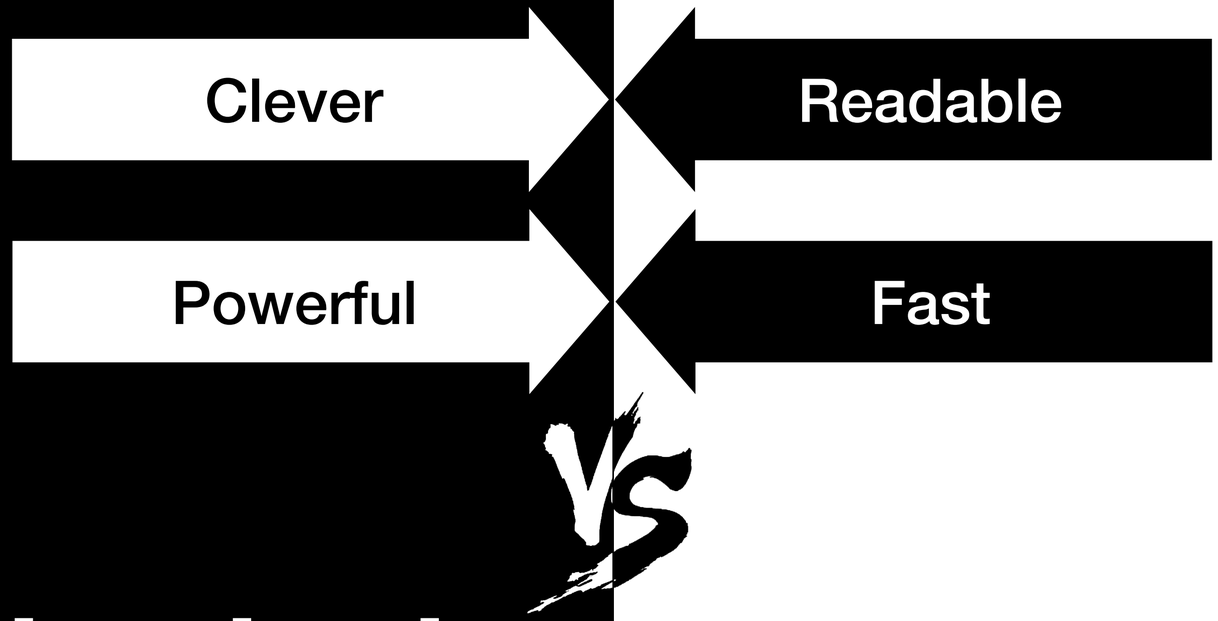


It's always a tradeoff



Runtime checked generics?

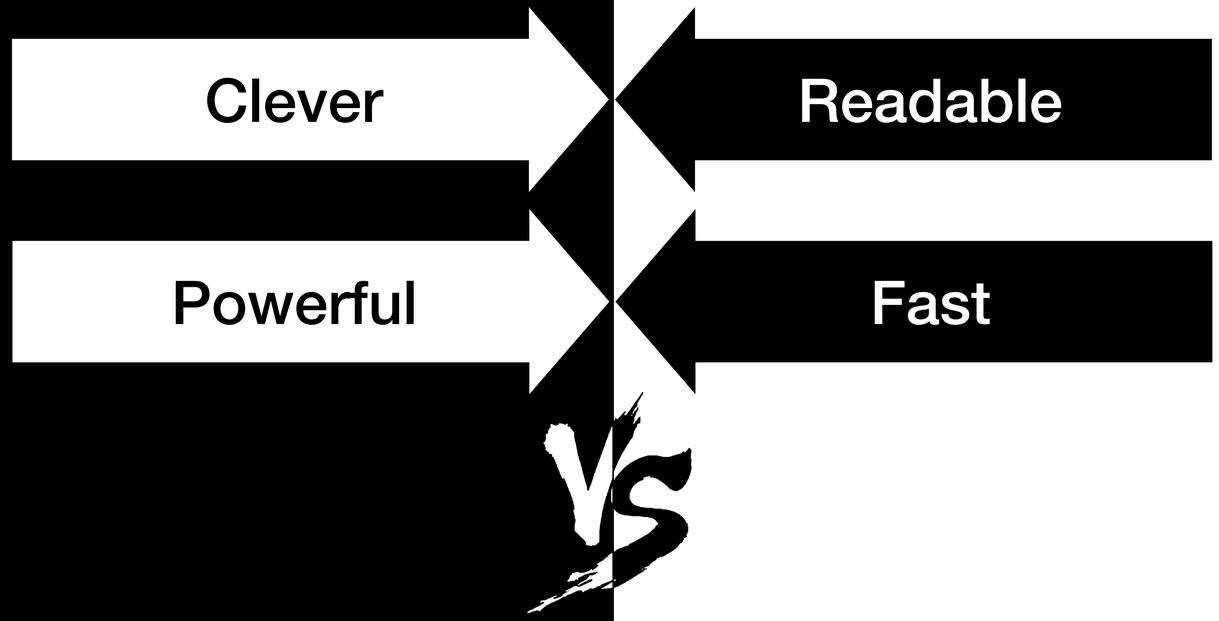
It's always a tradeoff



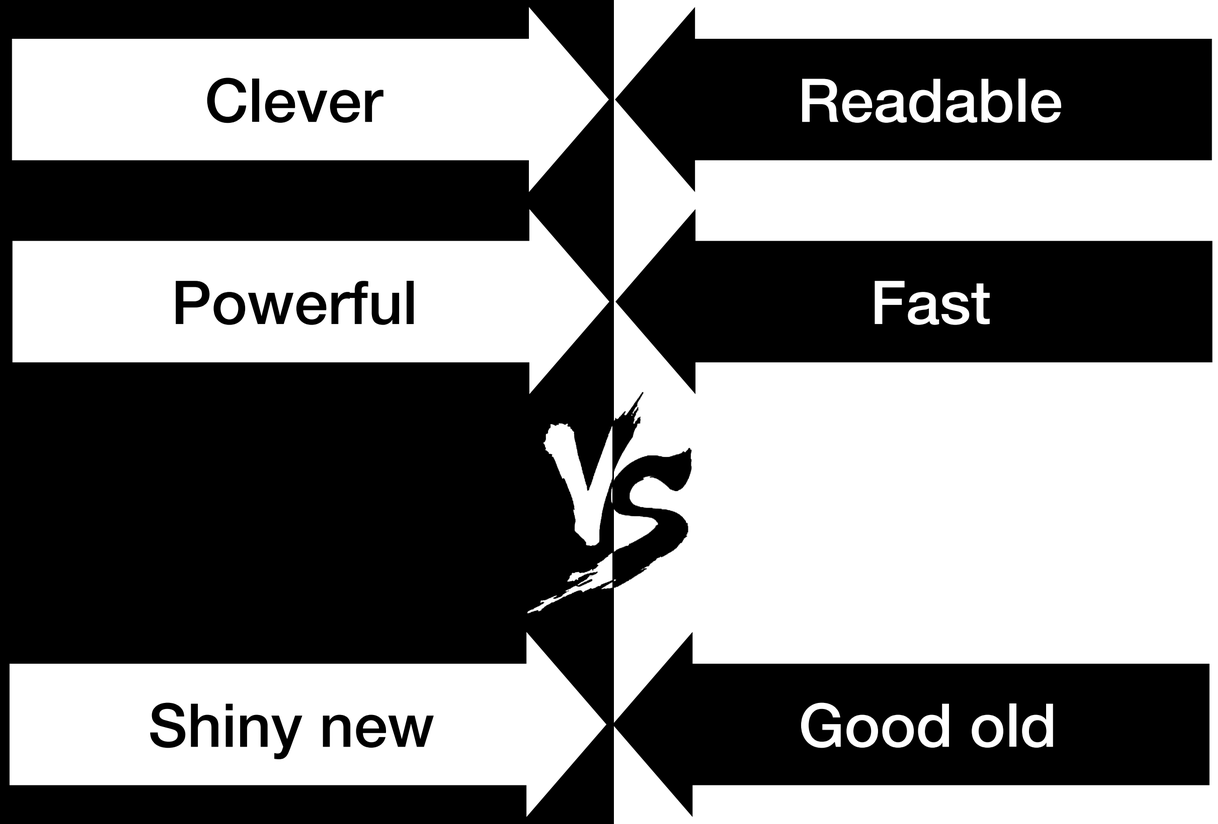
Runtime checked generics? *

* But with a 20% performance penalty?

It's always a tradeoff



It's always a tradeoff



It's always a tradeoff

```php
interface Logger {
    function log() { echo __METHOD__, "\n"; }
}
```

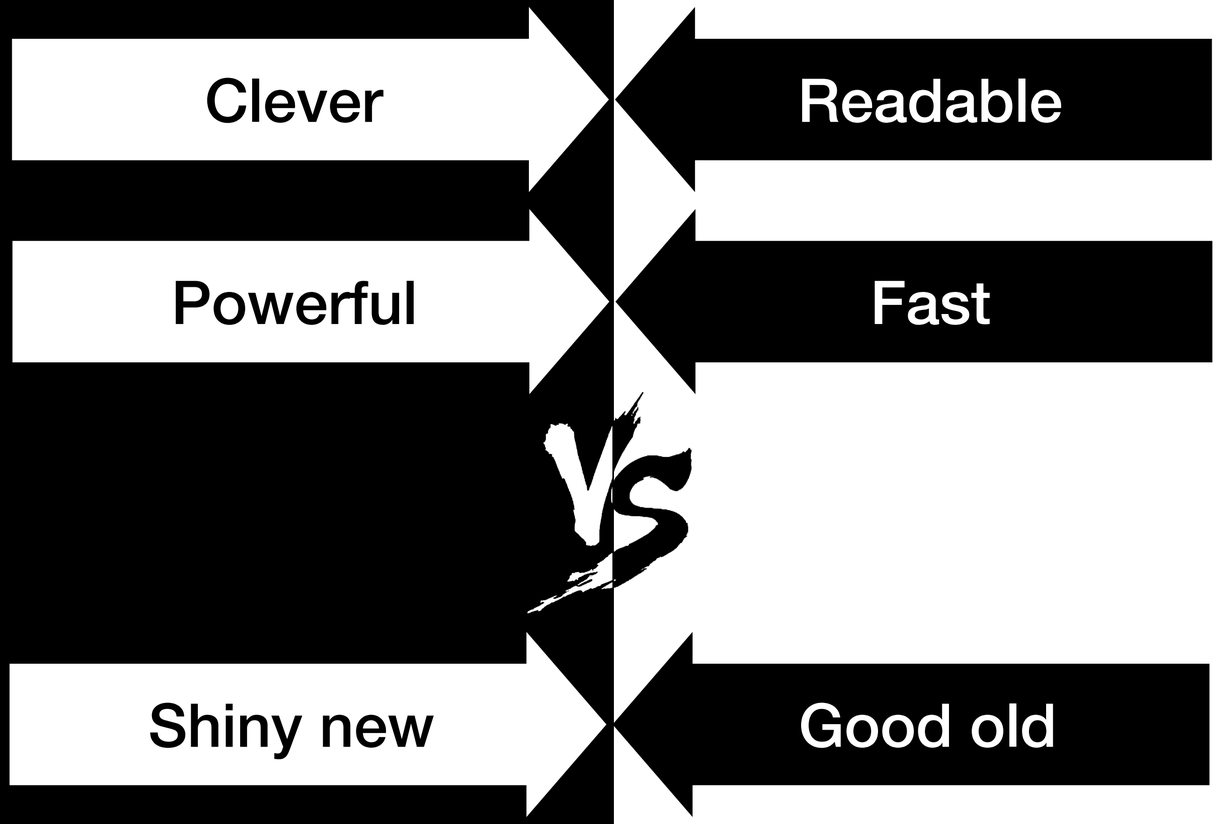
```php
interface Logger {
    function log();
}
trait LoggerTrait {
    function log() { echo __METHOD__, "\n"; }
}
```

VS

Shiny new

Good old

It's always a tradeoff



It's always a tradeoff

Clever	vs	Readable
Powerful		Fast
Shiny new		Good old
Complexity		Value

It's always a tradeoff

Clever	vs	Readable
Powerful		Fast
Shiny new		Good old
Complexity		Value

Tell and Show

My PHP Wishlist 😍

My PHP Wishlist 😍

- Official toolchain

My PHP Wishlist 😍

- Official toolchain

- Install any PHP version on any system and per project.
- Composer.
- Static analyser.

My PHP Wishlist 😍

- Official toolchain

My PHP Wishlist 😍

- Official toolchain
- Modern JSON, String, HTTP, etc

My PHP Wishlist 😍

- Official toolchain
- Modern JSON, String, HTTP, etc

```
// How to get the length of a string with emoji?

echo strlen( string: '👩‍❤️‍👩');      // 25
echo mb_strlen('👩‍❤️‍👩');        // 7

echo Str::length( value: '👩‍❤️‍👩'); // 7

echo grapheme_strlen('👩‍❤️‍👩');  // 1
```

My PHP Wishlist 😍

- Official toolchain
- Modern JSON, String, HTTP, etc
- Bundled extensions

My PHP Wishlist 😍

- Official toolchain
- Modern JSON, String, HTTP, etc
- Bundled extensions

- Swoole / swow

```
$http = new Swoole\Http\Server('127.0.0.1', 9501);
$http->set(['hook_flags' => SWOOLE_HOOK_ALL]);

$http->on('request', function ($request, $response) {
    $result = [];
    Co::join([
        go(function () use (&$result) {
            $result['google'] = file_get_contents( filename: "https://www.google.c
        }),
        go(function () use (&$result) {
            $result['bing'] = file_get_contents( filename: "https://www.bing.com/"
        })
    ]);
    $response->end(json_encode($result));
});

$http->start();
```

My PHP Wishlist 😍

- Official toolchain
- Modern JSON, String, HTTP, etc
- Bundled extensions

My PHP Wishlist 😍

- Official toolchain
- Modern JSON, String, HTTP, etc
- Bundled extensions
- Local readonly variables

My PHP Wishlist 😍

- Official toolchain
- Modern JSON, String, HTTP, etc
- Bundled extensions
- Local readonly variables

```
function foobar()
{
    readonly $local_const = '💜';

    echo $local_const;

    $local_const = '⚠️'; // Compile time error
}
```

My PHP Wishlist 😍

- Official toolchain
- Modern JSON, String, HTTP, etc
- Bundled extensions
- Local readonly variables

My PHP Wishlist 😍

- Official toolchain
- Modern JSON, String, HTTP, etc
- Bundled extensions
- Local readonly variables
- Multiline short closures

My PHP Wishlist 😍

- Official toolchain
- Modern JSON, String, HTTP, etc
- Bundled extensions
- Local readonly variables
- Multiline short closures
- Compiling PHP to binary

Your PHP Wishlists!

#myphpwishlist

Your PHP Wishlists!

#myphpwishlist

@brendt_gd

What I would change about PHP

If *you* could change something about PHP, without having to worry about backwards compatibility or breaking changes; what would it be? I know for sure I'd change a thing or two — or ten.

I like thinking about these topics. Not because I believe all of these things necessarily need be added to PHP, but because it's good to challenge our own views and critically think about language design.

Your PHP Wishlists!

#myphpwishlist

@brendt_gd

@sebdedeyne

What I would ch
about PHP

If *you* could change something about PHP, without
backwards compatibility or breaking changes; wha
sure I'd change a thing or two — or ten.

I like thinking about these topics. Not because I be
necessarily need be added to PHP, but because it
own views and critically think about language desi

PHP wishlist: The pipe operator

2023-02-16 #php #php-wishlist

Is it weird to have a favorite operator? Well, the pipe operator `|>` is mine. Not only does it look cool, it opens a world of possibilities for better code.

Your PHP Wishlists!

#myphpwishlist

@brendt_gd

What I would ch
about PHP

If *you* could change something about PHP, without
backwards compatibility or breaking changes; wha
sure I'd change a thing or two — or ten.

I like thinking about these topics. Not because I be
necessarily need be added to PHP, but because it
own views and critically think about language desi

@sebdedeyne

PHP wishlist: The pipe operator

2023-02-16 #php #php-wishlist

Is it weird to have a favorite operator? Well, the pipe operator `|>` is mine. Not only does it look cool, it opens a world of possibilities for better code.

@ryangjchandler

4 things I'd like to see in a future version of PHP

7TH DECEMBER 2022

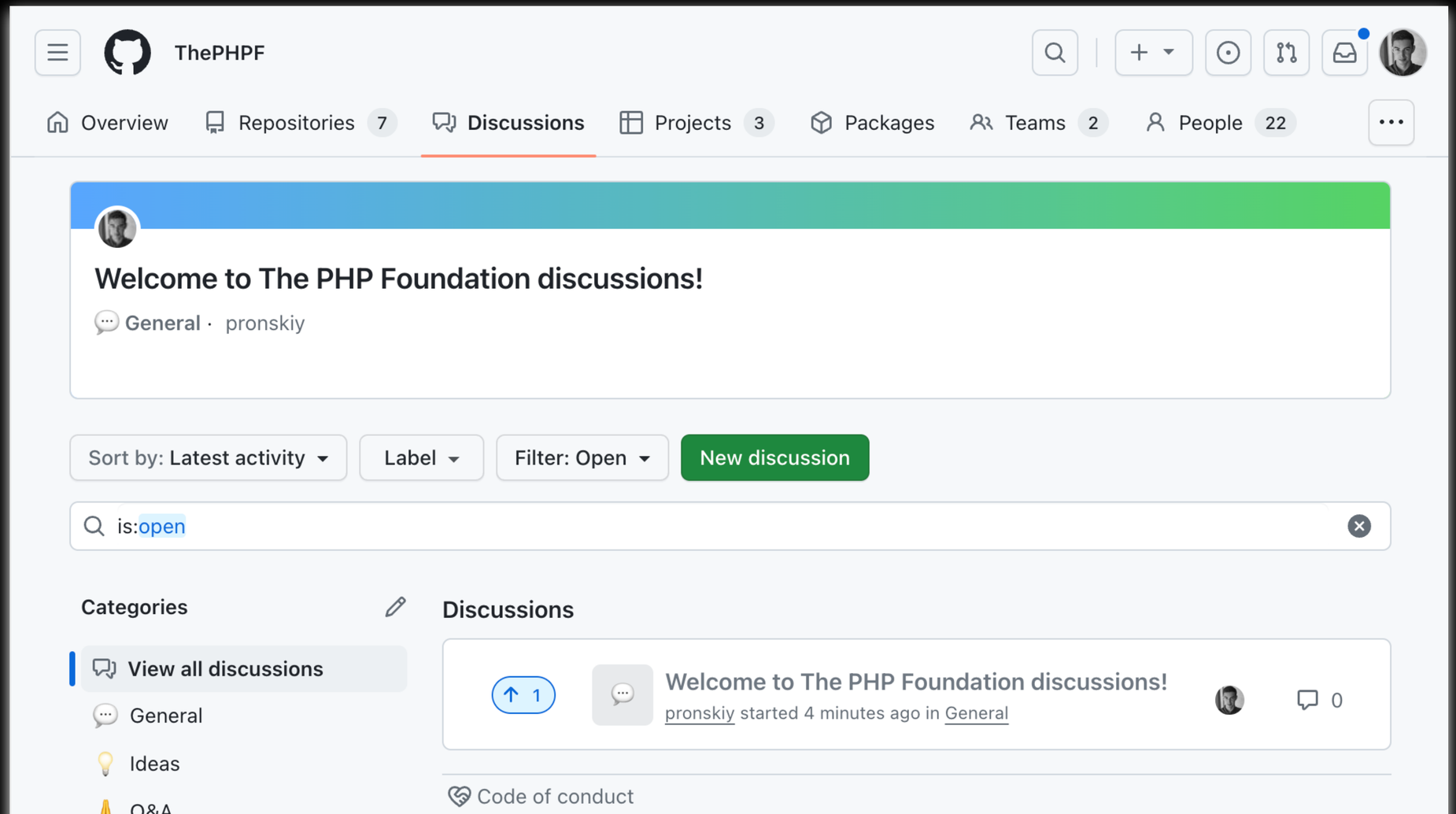
php

Table of Contents

- 1. Type aliases
- 2. Multi-line match arms
- 3. Generics
  Monomorphization
  Reificiation
  Type erasure
- 4. Pattern matching on value types

How can you help?

Get involved!



💜 Sponsor

opencollective.com/phpfoundation

🙏 More importantly!

Ask your company to join as a member

thephp.foundation/join

📣 Spread the word

thephp.foundation

@thephpf

Thank you 💜🐘

jb.gg/phpf

@pronskiy