なんでいま静的解析なの？
PHPで学ぶ「静」と「動」

Why Static Analysis Now? Exploring “Static” and “Dynamic” in PHP

pixiv Inc.
USAMI Kenta

pixiv

2025-10-11 #phpcon_hiroshima
PHPカンファレンス広島2025

お前誰よ

- うさみけんた（@tadsan）/ **Zonu.EXE** / **にゃんだーすわん**
- **ピクシブ株式会社** Platform Div > WebTechnology Team **PHPer**
  - 2012年末から現職、APIとかCIとかいろいろなところを見つめてきました
- Emacs PHP Modeを開発しています（2017年-）
- プログラミング言語にちょっとこだわりのある素人（spcamp2010）
- 西日本とは縁もゆかりもなくて、広島に来たのは今日で三回目です！
- assert(count($tadsan->phpcons) >= **45**);

今回のお題

PHPカンファレンス広島2025

採択 2025/10/11 11:50〜 VAddyルーム レギュラートーク（20分）

なんでいま静的解析なの？ PHPで学ぶ「静」と「動」

 うさみけんた tadsan

☆ 3

PHPはWeb開発の現場でカジュアルに使われてきましたが、「正しく」使うのが難しい言語だと言えます。
20年以上の歴史を重ねてきたWeb開発の現場でもさまざまな言語が生まれてきましたが、PHPの存在感は未だに衰えてはいません。

そのような言語の中でも揺れ動いてきたのが「静的(static)」と「動的(dynamic)」という概念です。
PHPは一般に「動的型付けのスクリプト言語」のように分類されますが、異なる特徴を持った言語も多く登場しています。

本トークではPHPの静的性と動的性について紹介し、PHPの現在地点、そして「静的解析」という技術がどのように問題解決につながるかを紹介します。

static
dynamic

プログラミングを
やっていると
ちょくちょく出てくる

static
dynamic

static

dynamic

pixiv

静 static

動 dynamic

静とは

該当件数 : **696** 件

変化形 : 《複》**statics**

static

【形】

1. 静止した、動かない◆【反】**astatic**
2. 固定された、据え付けの
3. 変化がない、不活発な、停滞した
4. 《物理》静的な、静止の、定位の
5. 《物理》静力学の
6. 《電気》静電気の
   ・Dry air produced static electricity in the house. : 乾いた空気は家の中に静電気を引き起こしました。
7. 《電気》〔電波の〕雑音の
8. 《コ》〔メモリーが〕スタティック型の

引用:
英辞郎 on the WEB, 株式会社アルク, EDP
https://eowf.alc.co.jp/search?q=static
https://eowf.alc.co.jp/search?q=dynamic
(2025年10月11日 閲覧)

該当件数 : **696** 件

変化形 : 《複》**statics**

static

【形】

1. 静止した、動かない◆【反】**astatic**
2. 固定された、据え付けの
3. 変化がない、不活発な、停滞した
4. 《物理》静的な、静止の、定位の
5. 《物理》静力学の
6. 《電気》静電気の
   ・Dry air produced static electricity in the house. : 乾いた空気は家の中に静電気を引き起こしました。
7. 《電気》〔電波の〕雑音の
8. 《コ》〔メモリーが〕スタティック型の

引用:
英辞郎 on the WEB, 株式会社アルク, EDP
https://eowf.alc.co.jp/search?q=static
https://eowf.alc.co.jp/search?q=dynamic
(2025年10月11日 閲覧)

該当件数 : **696** 件

変化形 : 《複》**statics**

static

【形】

1. 静止した、動かない◆【反】**astatic**
2. 固定された、据え付けの
3. 変化がない、不活発な、停滞した
4. 《物理》静的な、静止の、定位の
5. 《物理》静力学の
6. 《電気》静電気の
   ・Dry air produced static electricity in the house. : 乾いた空気は家の中に静電気を引き起こしました。
7. 《電気》〔電波の〕雑音の
8. 《コ》〔メモリーが〕スタティック型の

該当件数 : **3161** 件

変化形 : 《複》**dynamics**

dynamic

【形】

1. 力強い、行動的な、動的な、活動的な、生き生きした、活力に満ちた
   ・Economies are dynamic in nature. : 経済は生き物である。
2. 動力の、力学の、動力学的な
3. 《コ》動的な

【名】

1. 〔政治的・社会的・心理的な〕原動力、活性化させる力
2. 〔拮抗する力などの〕動態、ダイナミクス

音声を聞く レベル 4、 発音 dainǽmik、 カナ ダイナミク、 変化 《複》dynamics、 分節 dy・nam・ic

引用:
英辞郎 on the WEB, 株式会社アルク, EDP
https://eowf.alc.co.jp/search?q=static
https://eowf.alc.co.jp/search?q=dynamic
(2025年10月11日 閲覧)

該当件数 : **696** 件

変化形 : 《複》**statics**

static

【形】

1. 静止した、動かない◆【反】**astatic**
2. 固定された、据え付けの
3. 変化がない、不活発な、停滞した
4. 《物理》静的な、静止の、定位の
5. 《物理》静力学の
6. 《電気》静電気の
   ・Dry air produced static electricity in the house. : 乾いた空気は家の中に静電気を引き起こしました。
7. 《電気》〔電波の〕雑音の
8. 《コ》〔メモリーが〕スタティック型の

該当件数 : **3161** 件

変化形 : 《複》**dynamics**

dynamic

【形】

1. 力強い、行動的な、動的な、活動的な、生き生きした、活力に満ちた
   ・Economies are dynamic in nature. : 経済は生き物である。
2. 動力の、力学の、動力学的な
3. 《コ》動的な

【名】

1. 〔政治的・社会的・心理的な〕原動力、活性化させる力
2. 〔拮抗する力などの〕動態、ダイナミクス

音声を聞く レベル 4、 発音 dainǽmik、 カナ ダイナミク、 変化 《複》dynamics、 分節 dy・nam・ic

引用:
英辞郎 on the WEB, 株式会社アルク, EDP
https://eowf.alc.co.jp/search?q=static
https://eowf.alc.co.jp/search?q=dynamic
(2025年10月11日 閲覧)

動とは

Adjective [edit]

dynamic (*comparative* **more dynamic**, *superlative* **most dynamic**)

1. Changing; active; in motion. [synonyms, antonym ▲]

   Synonyms: active, fluid, moving; *see also* Thesaurus:changeable, Thesaurus:in motion

   Antonym: static

   *The environment is **dynamic**, changing with the years and the seasons.*

   ***dynamic** economy*

2. Powerful; energetic. [synonyms ▲]

   Synonyms: energetic, powerful; *see also* Thesaurus:active

   *He was a **dynamic** and engaging speaker.*

3. Able to change and adapt.
4. (*music*) Having to do with the volume of sound.

   *The **dynamic** marking in bar 40 is forte.*

5. (*computing*) Happening at runtime instead of being predetermined at compile time. [antonym ▲]

   Antonym: static

   ***dynamic** allocation*

   ***dynamic** IP addresses*

   *the **dynamic** resizing of an array*

6. Pertaining to dynamics, the branch of mechanics concerned with the effects of forces on the motion of objects.
7. (*grammar*) Of a verb: not stative, but fientive; indicating continued or progressive action on the part of the subject.

引用：dynamic - Wiktionary, the free dictionary（2025年8月12日 10:33版）

Adjective [edit]

活動・動いている

dynamic (*comparative* **more dynamic**, *superlative* **most dyna**

1. Changing; active; in motion. [synonyms, antonym ▲]

   Synonyms: active, fluid, moving; *see also* Thesaurus:changeable, Thesaurus:in motion

   Antonym: static

   *The environment is **dynamic**, changing with the years and the seasons.*

   ***dynamic** economy*

2. Powerful; energetic. [synonyms ▲]

   Synonyms: energetic, powerful; *see also* Thesaurus:active

   *He was a **dynamic** and engaging speaker.*

3. Able to change and adapt.
4. (*music*) Having to do with the volume of sound.

   *The **dynamic** marking in bar 40 is forte.*

5. (*computing*) Happening at runtime instead of being predetermined at compile time. [antonym ▲]

   Antonym: static

   ***dynamic** allocation*

   ***dynamic** IP addresses*

   *the **dynamic** resizing of an array*

6. Pertaining to dynamics, the branch of mechanics concerned with the effects of forces on the motion of objects.
7. (*grammar*) Of a verb: not stative, but fientive; indicating continued or progressive action on the part of the subject.

引用：dynamic - Wiktionary, the free dictionary（2025年8月12日 10:33版）

Adjective [edit]

dynamic (*comparative* **more dynamic**, *superlative* **most dyna**

1. Changing; active; in motion. [synonyms, antonym ▲]

   Synonyms: active, fluid, moving; *see also* Thesaurus:changeable, Thesaurus:in motion

   Antonym: static

   *The environment is **dynamic**, changing with the years and the seasons.*

   ***dynamic** economy*

2. Powerful; energetic. [synonyms ▲]

   Synonyms: energetic, powerful; *see also* Thesaurus:active

   *He was a **dynamic** and engaging speaker.*

活動・動いている

力強い・
エネルギッシュ

3. Able to change and adapt.
4. (*music*) Having to do with the volume of sound.

   *The **dynamic** marking in bar 40 is forte.*

5. (*computing*) Happening at runtime instead of being predetermined at compile time. [antonym ▲]

   Antonym: static

   ***dynamic** allocation*

   ***dynamic** IP addresses*

   *the **dynamic** resizing of an array*

6. Pertaining to dynamics, the branch of mechanics concerned with the effects of forces on the motion of objects.
7. (*grammar*) Of a verb: not stative, but fientive; indicating continued or progressive action on the part of the subject.

引用：dynamic - Wiktionary, the free dictionary（2025年8月12日 10:33版）

Adjective [edit]

dynamic (*comparative* **more dynamic**, *superlative* **most dyna**

1. Changing; active; in motion. [synonyms, antonym ▲]

   Synonyms: active, fluid, moving; *see also* Thesaurus:changeable, Thesaurus:in motion

   Antonym: static

   *The environment is **dynamic**, changing with the years and the seasons.*

   ***dynamic** economy*

2. Powerful; energetic. [synonyms ▲]

   Synonyms: energetic, powerful; *see also* Thesaurus:active

   *He was a **dynamic** and engaging speaker.*

3. Able to change and adapt.
4. (*music*) Having to do with the volume of sound.

   *The **dynamic** marking in bar 40 is forte.*

5. (*computing*) Happening at runtime instead of being predetermined at compile time. [antonym ▲]

   Antonym: static

   ***dynamic** allocation*

   ***dynamic** IP addresses*

   *the **dynamic** resizing of an array*

6. Pertaining to dynamics, the branch of mechanics concerned with the effects of forces on the motion of objects.
7. (*grammar*) Of a verb: not stative, but fientive; indicating continued or progressive action on the part of the subject.

活動・動いている

力強い・
エネルギッシュ

変化・適応できる

引用：dynamic - Wiktionary, the free dictionary（2025年8月12日 10:33版）

Adjective [edit]

dynamic (*comparative* **more dynamic**, *superlative* **most dyna**

1. Changing; active; in motion. [synonyms, antonym ▲]

   Synonyms: active, fluid, moving; *see also* Thesaurus:changeable, Thesaurus:in motion

   Antonym: static

   *The environment is **dynamic**, changing with the years and the seasons.*

   ***dynamic** economy*

2. Powerful; energetic. [synonyms ▲]

   Synonyms: energetic, powerful; *see also* Thesaurus:active

   *He was a **dynamic** and engaging speaker.*

3. Able to change and adapt.
4. (*music*) Having to do with the volume of sound.

   *The **dynamic** marking in bar 40 is forte.*

5. (*computing*) Happening at runtime instead of being predetermined at compile time. [antonym ▲]

   Antonym: static

   ***dynamic** allocation*

   ***dynamic** IP addresses*

   *the **dynamic** resizing of an array*

6. Pertaining to dynamics, the branch of mechanics concerned with the effects of forces on the motion of objects.
7. (*grammar*) Of a verb: not stative, but fientive; indicating continued or progressive action on the part of the subject.

活動・動いている

力強い・
エネルギッシュ

変化・適応できる

コンパイル時ではなく、
実行時に起こる

引用：dynamic - Wiktionary, the free dictionary（2025年8月12日 10:33版）

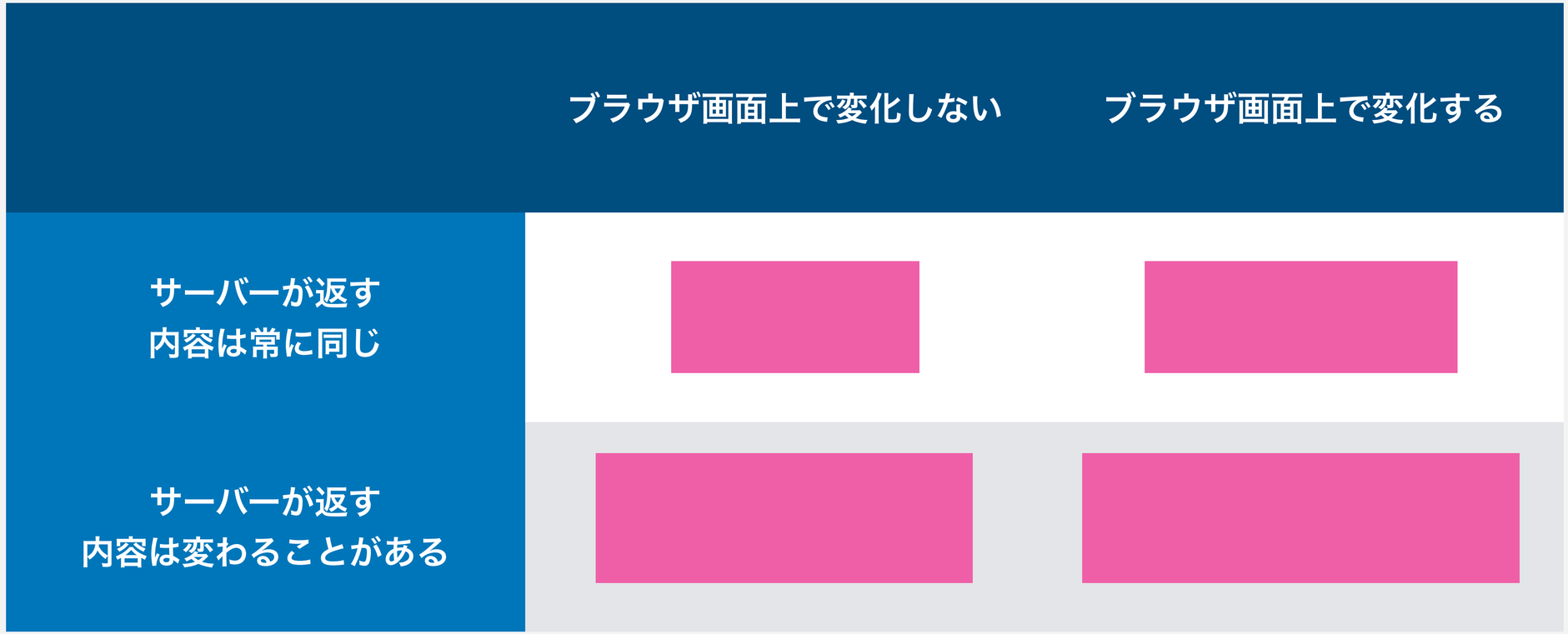
Webにおける静と動

Webページはいつ「変化」するか

- 永遠に変化しない／新しいファイルをアップロードしたら変化する
- 最初の表示は同じだが、人間が操作したら画面表示が変化する
- ページをリロードするごとに新しい内容が表示される
- 人間が操作するごとに新しい内容が表示される

整理してみましょう

Webページはどこで「変化」するか



pixiv

Webページはどこで「変化」するか

	ブラウザ画面上で変化しない	ブラウザ画面上で変化する
サーバーが返す 内容は常に同じ	静的ページ	
サーバーが返す 内容は変わることがある		

pixiv

Webページはどこで「変化」するか

	ブラウザ画面上で変化しない	ブラウザ画面上で変化する
サーバーが返す 内容は常に同じ	静的ページ	
サーバーが返す 内容は変わることがある	サーバーサイド テンプレートエンジン	

Webページはどこで「変化」するか

	ブラウザ画面上で変化しない	ブラウザ画面上で変化する
サーバーが返す 内容は常に同じ	静的ページ	シングルページ アプリケーション
サーバーが返す 内容は変わることがある	サーバーサイド テンプレートエンジン	

pixiv

Webページはどこで「変化」するか

	ブラウザ画面上で変化しない	ブラウザ画面上で変化する
サーバーが返す 内容は常に同じ	静的ページ	シングルページ アプリケーション
サーバーが返す 内容は変わることがある	サーバーサイド テンプレートエンジン	サーバーサイドレンダリング + hydration

Webアプリケーション分類

	ブラウザ画面上で変化しない	ブラウザ画面上で変化する
サーバーが返す 内容は常に同じ	静的ページ	シングルページ アプリケーション
サーバーが返す 内容は変わることがある	サーバーサイド テンプレートエンジン	サーバーサイドレンダリング + hydration

Webアプリケーション分類

	ブラウザ画面上で変化しない	ブラウザ画面上で変化する
サーバーが返す 内容は常に同じ	静的ページ	シングルページ アプリケーション
サーバーが返す 内容は変わることがある	サーバーサイド テンプレートエンジン	サーバーサイドレンダリング + hydration

Webアプリケーション分類

フロントエンド
アプリケーション

	ブラウザ画面上で変化しない	ブラウザ画面上で変化する
サーバーが返す 内容は常に同じ	静的ページ	シングルページ アプリケーション
サーバーが返す 内容は変わることがある	サーバーサイド テンプレートエンジン	サーバーサイドレンダリング + hydration

pixiv

Webアプリケーション分類

フロントエンド
アプリケーション

	ブラウザ画面上で変化しない	ブラウザ画面上で変化する
サーバーが返す 内容は常に同じ	静的ページ	シングルページ アプリケーション
サーバーが返す 内容は変わることがある	サーバーサイド テンプレートエンジン	サーバーサイドレンダリング + hydration

pixiv

Webアプリケーション分類

フロントエンド
アプリケーション

サーバーサイド
アプリケーション

	ブラウザ画面上で変化しない	ブラウザ画面上で変化する
サーバーが返す 内容は常に同じ	静的ページ	シングルページ アプリケーション
サーバーが返す 内容は変わることがある	サーバーサイド テンプレートエンジン	サーバーサイドレンダリング + hydration

pixiv

静的ホスティング vs Webアプリ

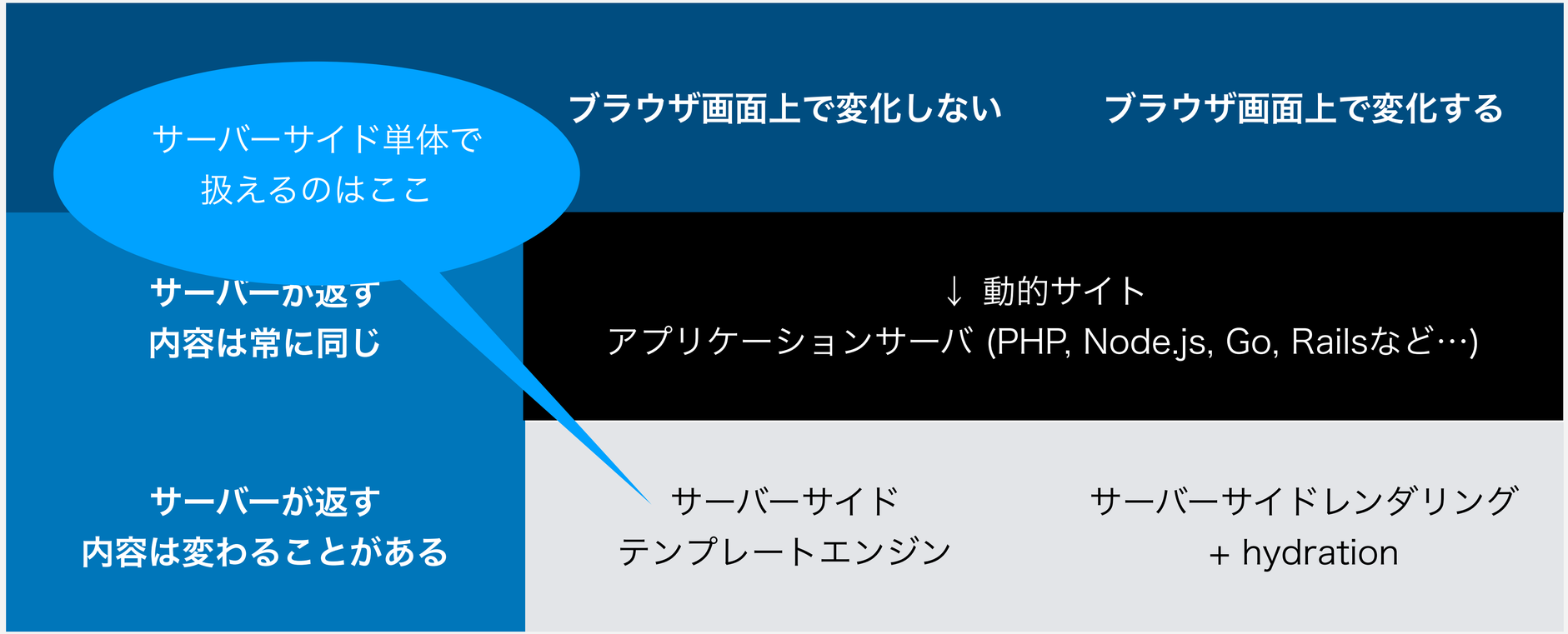
	ブラウザ画面上で変化しない	ブラウザ画面上で変化する
サーバーが返す 内容は常に同じ	静的ページ	シングルページ アプリケーション
サーバーが返す 内容は変わることがある	↑ 静的ホスティングサーバー (GitHub Pages, PHP/CGI非対応レンタルサーバ, HTTPサーバ単体)	

動的Webサーバ(PHP)

	ブラウザ画面上で変化しない	ブラウザ画面上で変化する
サーバーが返す 内容は常に同じ	↓ 動的サイト アプリケーションサーバ (PHP, Node.js, Go, Railsなど…)	
サーバーが返す 内容は変わることがある	サーバーサイド テンプレートエンジン	サーバーサイドレンダリング + hydration

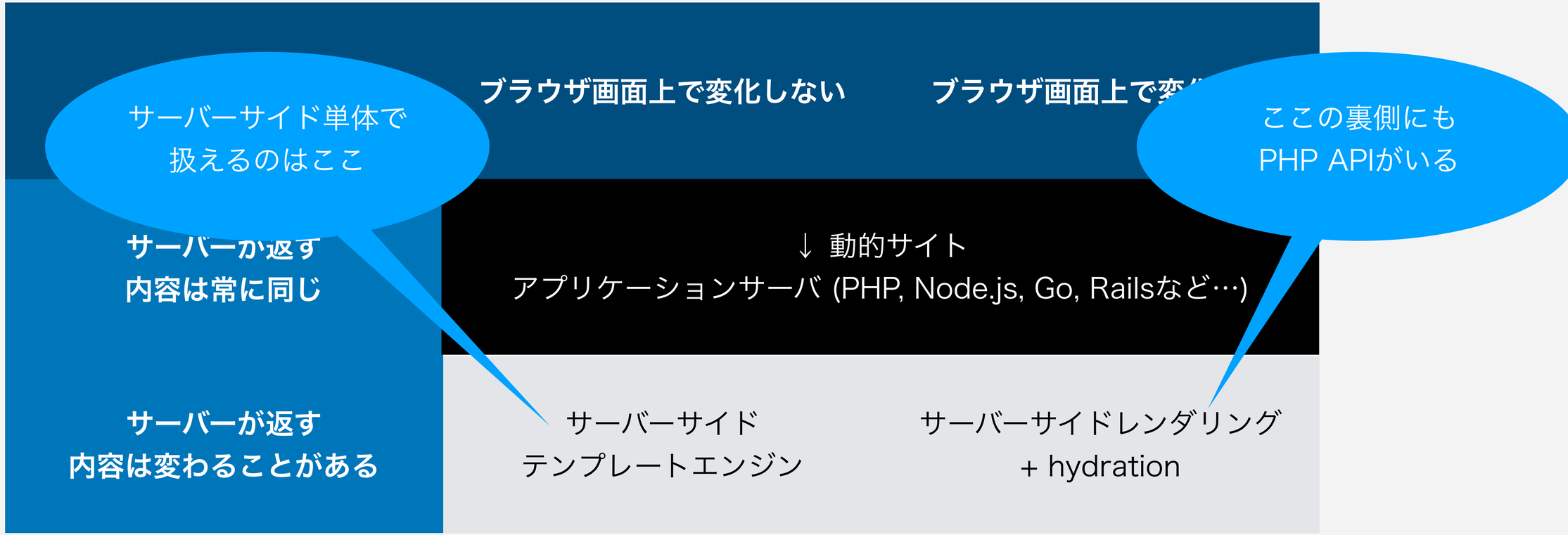
pixiv

動的Webサーバ(PHP)



pixiv

動的Webサーバ(PHP)



pixiv

Webアプリ vs Next.js (SSR)

	ブラウザ画面上で変化しない	ブラウザ画面上で変化する
サーバーが返す 内容は常に同じ	Next.jsのようなフレームワークは これに特化している	
サーバーが返す 内容は変わることがある	Vercel, AmplifyなどのPaaS	サーバーサイドレンダリング + hydration

Webアプリ vs Next.js (SSG/SPA)

	ブラウザ画面上で変化しない	ブラウザ画面上で変化する
サーバーが返す 内容は常に同じ	静的ページ (SSG/静的サイト生成)	シングルページ アプリケーション
サーバーが返す 内容は変わることがある	Next.jsは静的サイト開発にも使える	

pixiv

Webアプリケーションの静と動

- ここまでの説明はとても大雑把
- 現実のWebサイトでは静的な要素と動的な要素が複雑に入り混じる
  - 静的なHTML/JSファイルからAPIにリクエストしてページ表示するとか
- 静と動を適切に見極めて構成することで、開発効率やインフラコストの最適化
- 従来はサーバーサイドアプリケーションが必要だったものも、
  近年では外部サービス（mBaaS/IDaaS）に依存することもできる

静と動の境界を「ずらす」

	ブラウザ画面上で変化しない	ブラウザ画面上で変化する
サーバーが返す 内容は常に同じ	静的ページ	シングルページ アプリケーション
サーバーが返す 内容は変わることがある	サーバーサイド テンプレートエンジン	サーバーサイドレンダリング + hydration

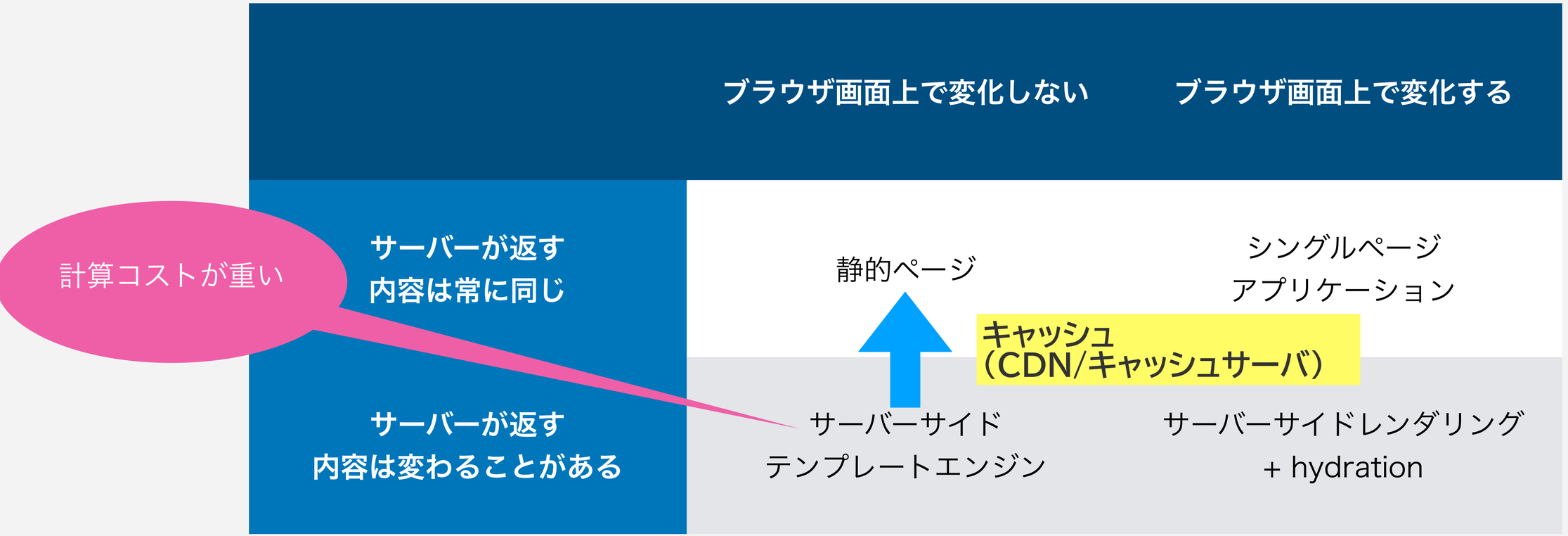
静と動の境界を「ずらす」

	ブラウザ画面上で変化しない	ブラウザ画面上で変化する
サーバーが返す 内容は常に同じ	静的ページ	シングルページ アプリケーション
サーバーが返す 内容は変わることがある	サーバーサイド テンプレートエンジン	サーバーサイドレンダリング + hydration

計算コストが重い

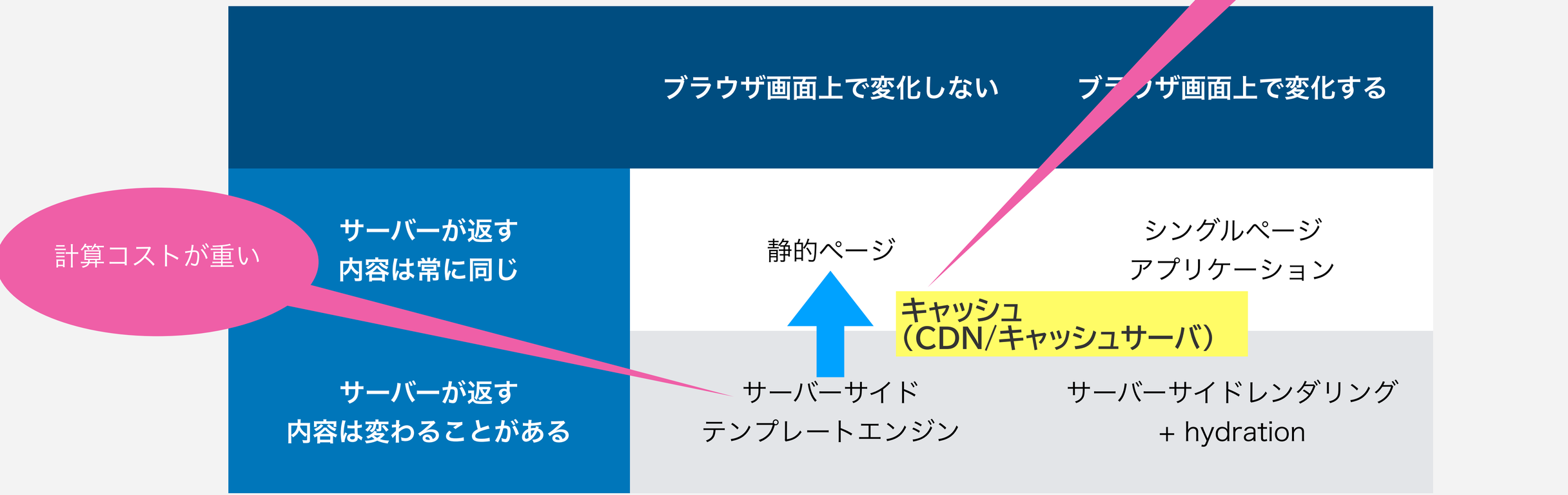
pixiv

静と動の境界を「ずらす」



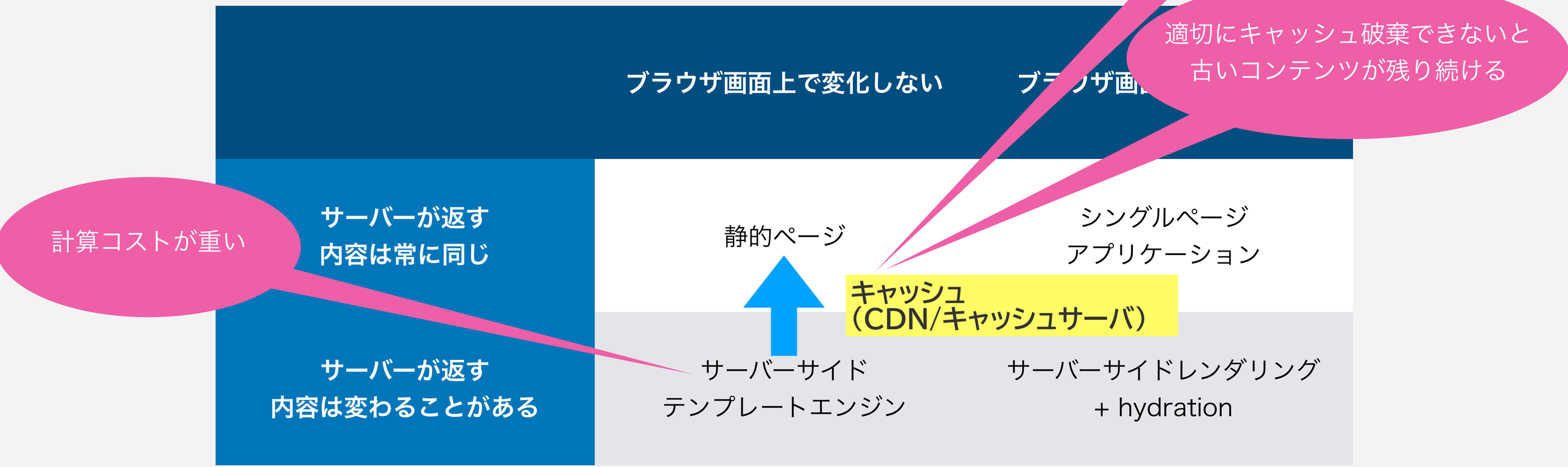
pixiv

静と動の境界を「ずらす」



pixiv

静と動の境界を「ずらす」



pixiv

静と動の境界を「ずらす」

	ブラウザ画面上で変化しない	ブラウザ画面上で変化する
サーバーが返す 内容は常に同じ	静的ページ	シングルページ アプリケーション
サーバーが返す 内容は変わることがある	サーバーサイド テンプレートエンジン	サーバーサイドレンダリング + hydration

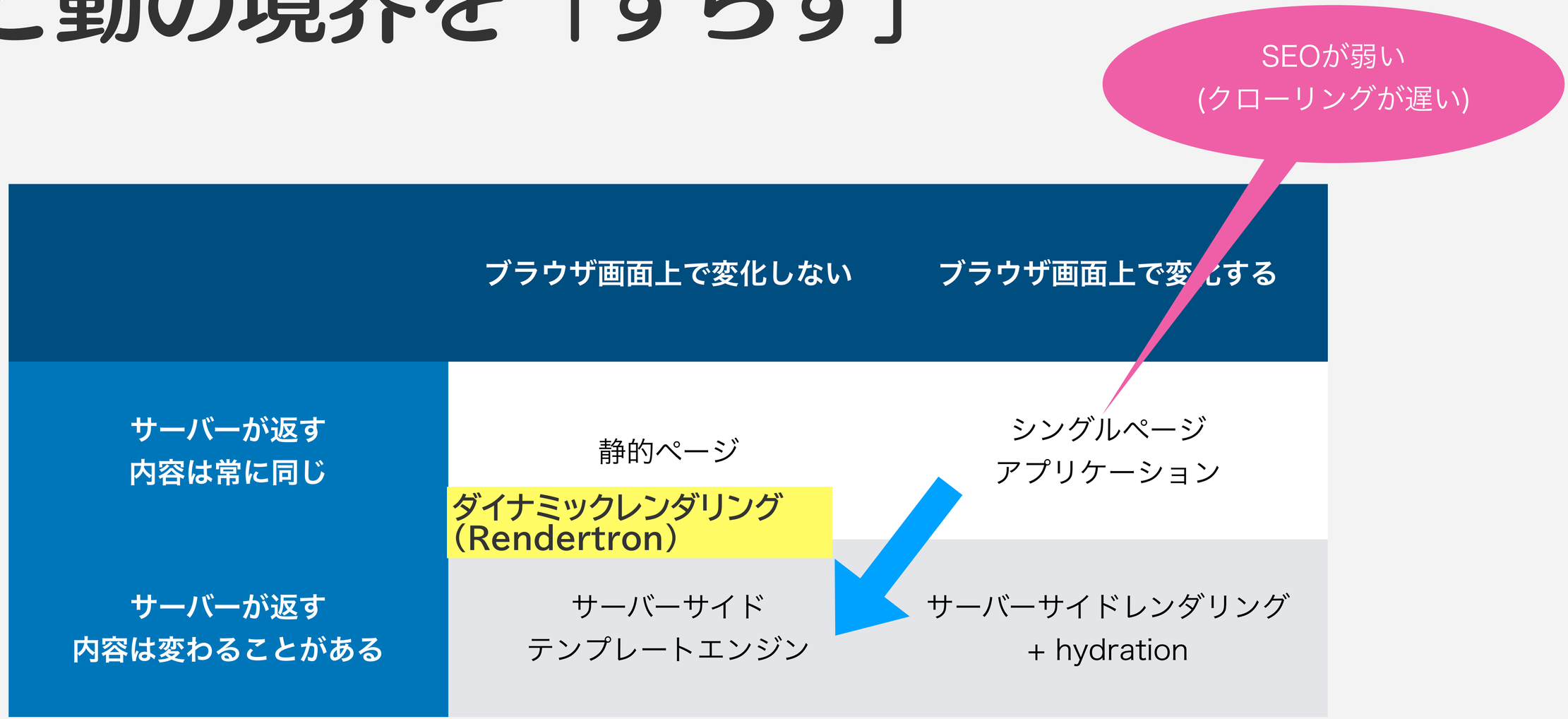
静と動の境界を「ずらす」

SEOが弱い
(クローリングが遅い)

	ブラウザ画面上で変化しない	ブラウザ画面上で変化する
サーバーが返す 内容は常に同じ	静的ページ	シングルページ アプリケーション
サーバーが返す 内容は変わることがある	サーバーサイド テンプレートエンジン	サーバーサイドレンダリング + hydration

pixiv

静と動の境界を「ずらす」

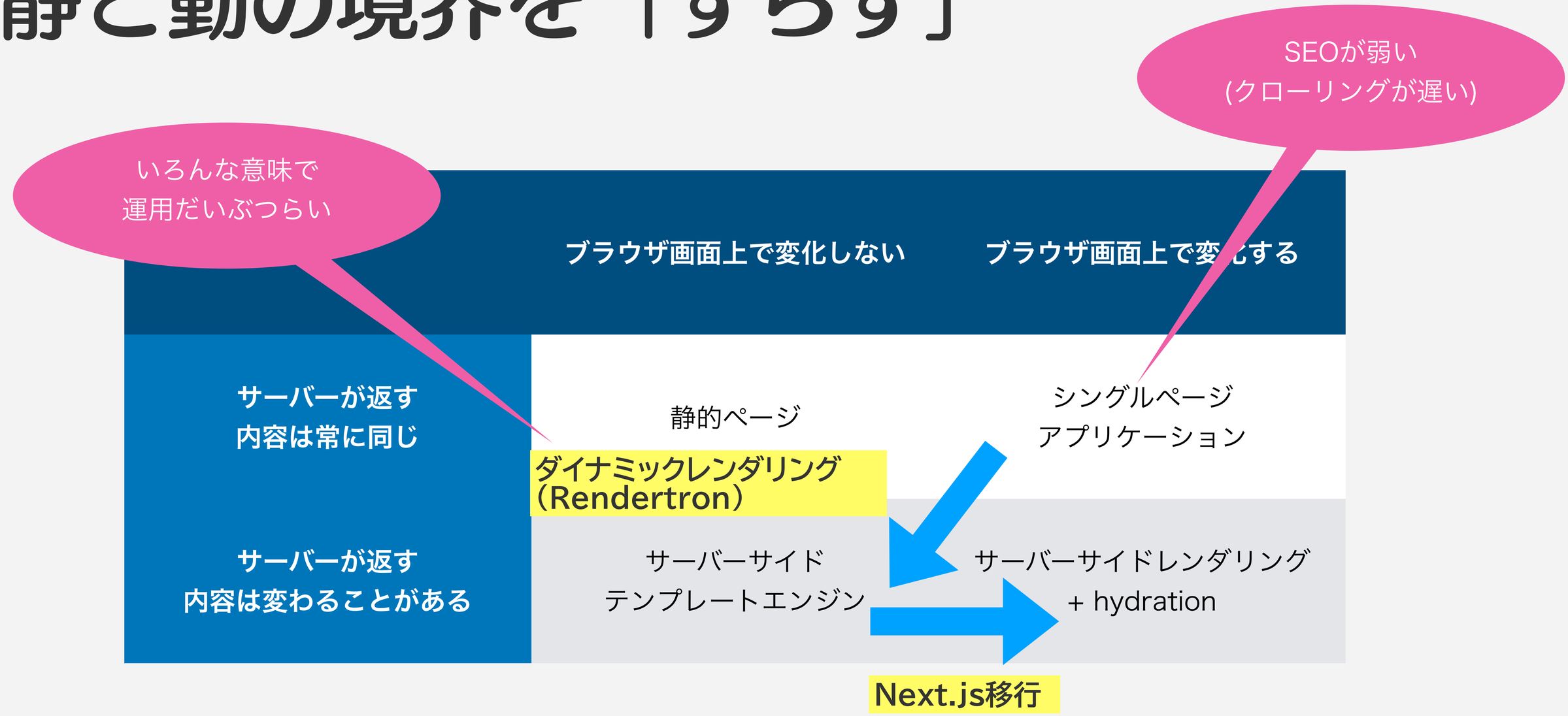


pixiv

静と動の境界を「ずらす」

pixiv

静と動の境界を「ずらす」



pixiv

静と動

pixiv

動くものと
変わらぬものがある

開発における静と動

プログラムにも
静と動がある

PHP (プログラミング言語)

119 個の言語版

ページ　ノート　　閲覧　ソースを編集　履歴を表示　ツール

出典: フリー百科事典『ウィキペディア（Wikipedia）』

PHP（ピー・エイチ・ピー）は、 "The PHP Group" によってコミュニティベースで開発[2]されているオープンソースの汎用プログラミング言語およびその公式の処理系であり、特にサーバーサイドで動的なウェブページを作成するための機能を多く備えていることを特徴とする[3]。 名称の **PHP** は再帰的頭字語として、 "PHP: Hypertext Preprocessor" を意味[4][5]するとされており、「PHPはHTMLのプリプロセッサである」とPHP自身を再帰的に説明している。

概要 [ソースを編集]

PHPはラスマス・ラードフが個人的にCで開発していたCGIプログラムである "Personal Home Page Tools" （短縮されて "PHP Tools" と呼ばれていた）を起源とする[5]。 元々はラードフ自身のWebサイトで簡単な動的Webページを作成するために用いられていたが、その後データベースへのアクセス機能などを追加したPHP Toolsを1995年にGPLの下で公開した[6]。 オープンソースライセンスの下で公開されたことにより同ツールの利用者が増加し、機能の追加を行う開発者たちの貢献もあって、幾度かの大きなバージョンアップを経て今日に至っている。 PHPの再帰的頭字語が PHP: Hypertext Preprocessor となったのは2017年現在の文法の基礎が確立したPHP 3から[5]である。

先に述べたように、PHPは動的なWebページを生成するツールを起源としているため、公式の処理系にはWebアプリケーション開発に関する機能が豊富に組み込まれている。 元々PHPはプログラミング言語と言えるものではなく、単にテンプレート的な処理を行うだけであったが、度重なる機能追加やコードの書き直しにより、2017年現在リリースされているPHP 5やPHP 7は目的によらず汎用的に使うことの出来るスクリプト言語となっている。 特にApache HTTP ServerやnginxといったWebサーバーソフトウェアから動作させるスクリプト言語として選択されてサーバーサイドWebアプリケーション開発に利用されることが多い。

PHP

PHPのロゴ

パラダイム	命令型プログラミング、関数型プログラミング、オブジェクト指向プログラミング、手続き型プログラミング、リフレクション
登場時期	1995年6月8日 (30年前)
開発者	ラスマス・ラードフ、アンディ・ガトマンズ、PHP Group、ゼンド・テクノロジーズ、ゼーブ・スラスキー
最新リリース	8.4.13[1] / 2025年9月25日 (8日前)
型付け	強い動的型付け
主な処理系	PHP, HHVM, Phalanger
方言	Hack
影響を受けた言語	C++、Perl、C言語、Java、Tcl、HyperText Markup Language、JavaScript
プラットフォーム	Unix系、Microsoft

引用：PHP（プログラミング言語）- Wikipedia（2025年6月22日 14:23版）

PHP (プログラミング言語)

119 個の言語版

ページ　ノート　　閲覧　ソースを編集　履歴を表示　ツール

出典: フリー百科事典『ウィキペディア（Wikipedia）』

PHP（ピー・エイチ・ピー）は、 "The PHP Group" によってコミュニティベースで開発[2]されているオープンソースの汎用プログラミング言語およびその公式の処理系であり、特にサーバーサイドで動的なウェブページを作成するための機能を多く備えていることを特徴とする[3]。 名称の **PHP** は再帰的頭字語として、 "PHP: Hypertext Preprocessor" を意味[4][5]するとされており、「PHPはHTMLのプリプロセッサである」とPHP自身を再帰的に説明している。

概要 [ソースを編集]

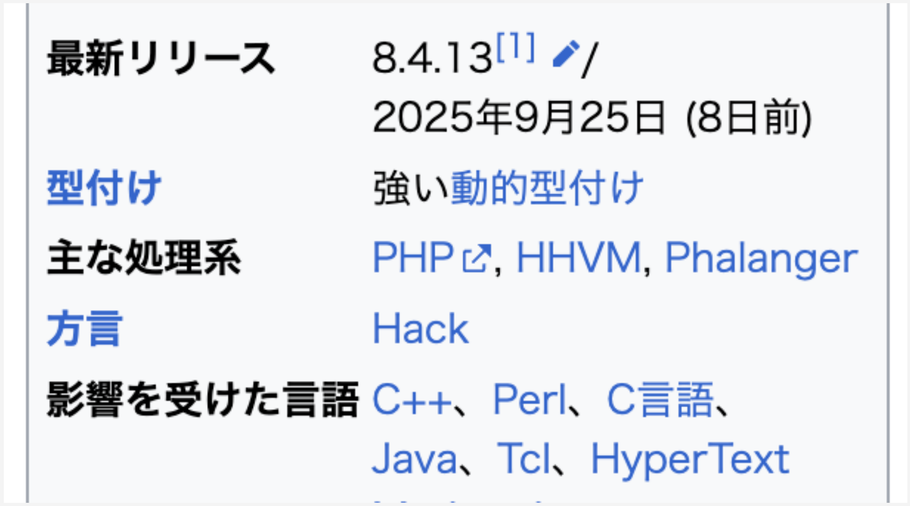
PHPはラスマス・ラードフが個人的にCで開発していたCGIプログラムである "Personal Home Page Tools" （短縮されて "PHP Tools" と呼ばれていた）を起源とする[5]。 元々はラードフ自身のWebサイトで簡単な動的Webページを作成するために用いられていたが、その後データベースへのアクセス機能などを追加したPHP Toolsを1995年にGPLの下で公開した[6]。 オープンソースライセンスの下で公開されたことにより同ツールの利用者が増加し、機能の追加を行う開発者たちの貢献もあって、幾度かの大きなバージョンアップを経て今日に至っている。 PHPの再帰的頭字語が PHP: Hypertext Preprocessor となったのは2017年現在の文法の基礎が確立したPHP 3から[5]である。

先に述べたように、PHPは動的なWebページを生成するツールを起源としているため、公式の処理系にはWebアプリケーション開発に関する機能が豊富に組み込まれている。 元々PHPはプログラミング言語と言えるものではなく、単にテンプレート的な処理を行うだけであったが、度重なる機能追加やコードの書き直しにより、2017年現在リリースされているPHP 5やPHP 7は目的によらず汎用的に使うことの出来るスクリプト言語となっている。 特にApache HTTP ServerやnginxといったWebサーバーソフトウェアから動作させるスクリプト言語として選択されてサーバーサイドWebアプリケーション開発に利用されることが多い。

PHP

PHPのロゴ

パラダイム	命令型プログラミング、関数型プログラミング、オブジェクト指向プログラミング、手続き型プログラミング、リフレクション
登場時期	1995年6月8日 (30年前)
開発者	ラスマス・ラードフ、アンディ・ガトマンズ、PHP Group、ゼンド・テクノロジーズ、ゼーブ・スラスキー
最新リリース	8.4.13[1] / 2025年9月25日 (8日前)
型付け	強い動的型付け
主な処理系	PHP, HHVM, Phalanger
方言	Hack
影響を受けた言語	C++、Perl、C言語、Java、Tcl、HyperText Markup Language、JavaScript
プラットフォーム	Unix系、Microsoft



引用：PHP（プログラミング言語）- Wikipedia（2025年6月22日 14:23版）

pixiv

PHP (プログラミング言語)

119 個の言語版

ページ　ノート　　閲覧　ソースを編集　履歴を表示　ツール

出典: フリー百科事典『ウィキペディア（Wikipedia）』

PHP（ピー・エイチ・ピー）は、 "The PHP Group" によってコミュニティベースで開発[2]されているオープンソースの汎用プログラミング言語およびその公式の処理系であり、特にサーバーサイドで動的なウェブページを作成するための機能を多く備えていることを特徴とする[3]。 名称の **PHP** は再帰的頭字語として、 "PHP: Hypertext Preprocessor" を意味[4][5]するとされており、「PHPはHTMLのプリプロセッサである」とPHP自身を再帰的に説明している。

概要 [ソースを編集]

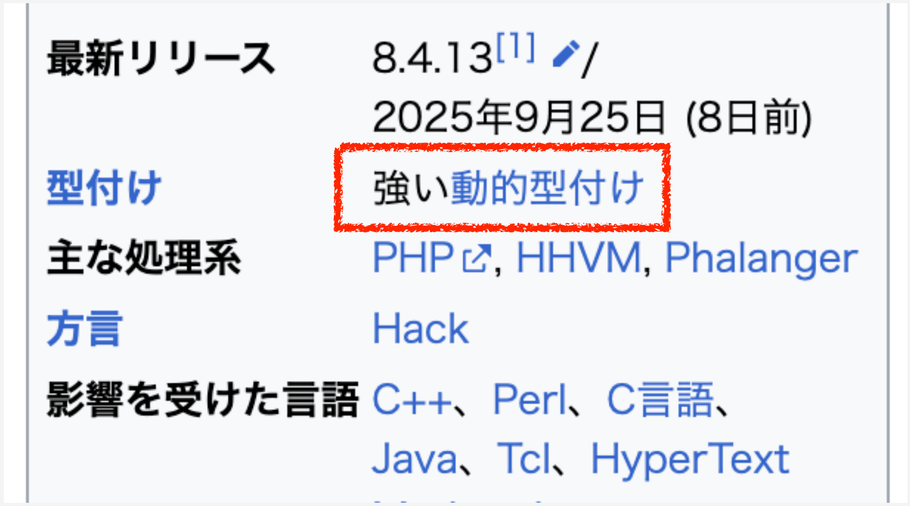
PHPはラスマス・ラードフが個人的にCで開発していたCGIプログラムである "Personal Home Page Tools" （短縮されて "PHP Tools" と呼ばれていた）を起源とする[5]。 元々はラードフ自身のWebサイトで簡単な動的Webページを作成するために用いられていたが、その後データベースへのアクセス機能などを追加したPHP Toolsを1995年にGPLの下で公開した[6]。 オープンソースライセンスの下で公開されたことにより同ツールの利用者が増加し、機能の追加を行う開発者たちの貢献もあって、幾度かの大きなバージョンアップを経て今日に至っている。 PHPの再帰的頭字語が PHP: Hypertext Preprocessor となったのは2017年現在の文法の基礎が確立したPHP 3から[5]である。

先に述べたように、PHPは動的なWebページを生成するツールを起源としているため、公式の処理系にはWebアプリケーション開発に関する機能が豊富に組み込まれている。 元々PHPはプログラミング言語と言えるものではなく、単にテンプレート的な処理を行うだけであったが、度重なる機能追加やコードの書き直しにより、2017年現在リリースされているPHP 5やPHP 7は目的によらず汎用的に使うことの出来るスクリプト言語となっている。 特にApache HTTP ServerやnginxといったWebサーバーソフトウェアから動作させるスクリプト言語として選択されてサーバーサイドWebアプリケーション開発に利用されることが多い。

PHP

PHPのロゴ

パラダイム	命令型プログラミング、関数型プログラミング、オブジェクト指向プログラミング、手続き型プログラミング、リフレクション
登場時期	1995年6月8日 (30年前)
開発者	ラスマス・ラードフ、アンディ・ガトマンズ、PHP Group、ゼンド・テクノロジーズ、ゼーブ・スラスキー
最新リリース	8.4.13[1] / 2025年9月25日 (8日前)
型付け	強い動的型付け
主な処理系	PHP, HHVM, Phalanger
方言	Hack
影響を受けた言語	C++、Perl、C言語、Java、Tcl、HyperText Markup Language、JavaScript
プラットフォーム	Unix系、Microsoft



引用：PHP（プログラミング言語）- Wikipedia（2025年6月22日 14:23版）

pixiv

C言語

123 個の言語版

ページ　ノート　　閲覧　ソースを編集　履歴を表示　ツール

出典: フリー百科事典『ウィキペディア（Wikipedia）』

C言語（シーげんご、英: C programming language）は、1972年にAT&Tベル研究所のデニス・リッチーが主体となって開発した汎用プログラミング言語である。英語圏では「C language」または単に「C」と呼ばれることが多い。日本でも文書や文脈によっては同様に「C」と呼ぶことがある。制御構文などに高水準言語の特徴を持ちながら、ハードウェア寄りの記述も可能な低水準言語の特徴も併せ持つ。基幹系システムや、動作環境の資源制約が厳しい、あるいは実行速度性能が要求されるソフトウェアの開発に用いられることが多い。後発のC++やJava、C#など、「C系」と呼ばれる派生言語の始祖でもある[注釈 1]。

ANSI、ISO、またJISにより言語仕様が標準規格化されている。

特徴 [ソースを編集]

この節に雑多な内容が羅列されています。 事項を箇条書きで列挙しただけの節は、本文として組み入れるか、または整理・除去する必要があります。(2019年8月)

Cには他のプログラミング言語と比較して、特筆すべきいくつかの特徴がある。

利点 [ソースを編集]

- 構造化プログラミングのパラダイムに対応した高水準の手続き型言語である。ハードウェアの直接的な制御ができる機能を備えつつ、機械語やアセンブリ言語（アセンブラ）のような低水準言語と比較して、ソースコードの再利用性やメンテナンス性に優れており、目的に応じたプログラムの変更や拡張が容易である。

C言語

C言語のロゴ

パラダイム	命令型プログラミング、構造化プログラミング、手続き型プログラミング
登場時期	1972年 (53年前).
開発者	ベル研究所、デニス・リッチー、米国国家規格協会、国際標準化機構、ケン・トンプソン
最新リリース	ISO/IEC 9899:2024/ 2024年10月31日 (11か月前)
型付け	弱い静的型付け
主な処理系	GCC, Clang, Visual C++, Intel C++ Compiler

引用：C言語 - Wikipedia（2025年9月16日 22:41版）

Java

120 個の言語版

ページ　ノート　　閲覧　ソースを編集　履歴を表示　ツール

出典: フリー百科事典『ウィキペディア（Wikipedia）』

この項目では、プログラミング言語について説明しています。その他の用法については「ジャバ」をご覧ください。

「JavaScript」とは異なります。

この記事は検証可能な参考文献や出典が全く示されていないか、不十分です。 出典を追加して記事の信頼性向上にご協力ください。（このテンプレートの使い方）

出典検索?: "Java" – ニュース・書籍・スカラー・CiNii・J-STAGE・NDL・dlib.jp・ジャパンサーチ・TWL *(2019年3月)*

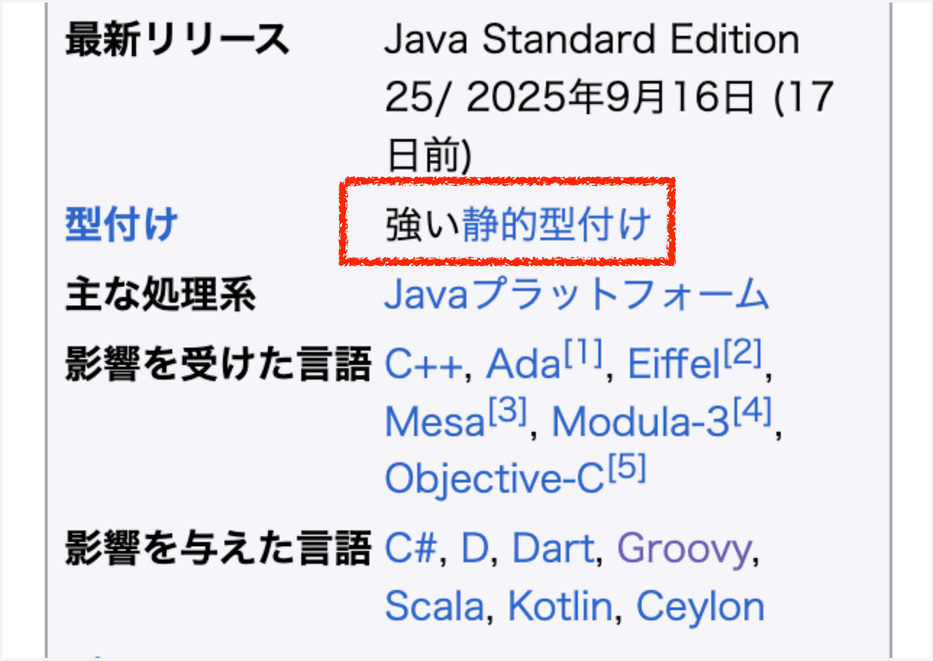
Java（ジャバ、ジャヴァ）は、汎用プログラミング言語とソフトウェアプラットフォームの双方を指している総称ブランドである[6]。オラクルおよびその関連会社の登録商標である。1996年にサン・マイクロシステムズによって市場リリースされ、2010年に同社がオラクルに吸収合併された事によりJavaの版権もそちらに移行した。

プログラミング言語Javaは、C++に類似の構文、クラスベースのオブジェクト指向、マルチスレッド、ガベージコレクション、コンポーネントベース、分散コンピューティングといった特徴を持ち、平易性重視のプログラム書式による堅牢性と、仮想マシン上での実行によるセキュリティ性およびプラットフォーム非依存性が理念とされている。Javaプラットフォームは、Javaプログラムの実行環境または、実行環境と開発環境の双方を統合したソフトウェアであり、ビジネスサーバ、モバイル機器、組み込みシステム、スマートカードといった様々なハードウェア環境に対応したソフトウェア形態で提供されている。その中枢技術であるJava仮想マシンは各プラットフォーム環境間の違いを吸収しながら、Javaプログラムの適切な共通動作を実現する機能を備えている[7]。このテクノロジは「*write once, run anywhere*」と標榜されていた[8]。

2019年の時点でGitHubによると[9]、Javaは特にクライアント/サーバモデルのWebアプリケーションで使用されている最も人気の高いプログラミング言語の1つであり[9]、全世界でおよそ900万人の開発者がいるとレポートされている[10]。最新バージョンは、2025年3月にリリ

Java

パラダイム	オブジェクト指向, コンポーネントベース, リフレクティブ, ジェネリック, 関数型, 並行プログラミング
登場時期	1995年5月23日α版 1995年秋β版 1996年1月23日ver1.0
設計者	Java Community Process
開発者	サン・マイクロシステムズ、オラクル、ジェームズ・ゴスリン
最新リリース	Java Standard Edition 25/ 2025年9月16日 (17日前)
型付け	強い静的型付け
主な処理系	Javaプラットフォーム
影響を受けた言語	C++, Ada[1], Eiffel[2], Mesa[3], Modula-3[4], Objective-C[5]



引用：Java - Wikipedia（2025年9月24日 16:18版）

静的型付き言語(statically typed language)

- 型検査 = プログラム(ソースコード)が型の制約を満たすか調べる
  - 例: 関数呼び出しに正しい型を渡せているか
- 静的型付きの言語はプログラムの実行前に型検査合格を前提とする
  - 型がついたとき、実行時にその種のエラーが起こらない性質を「健全である」あるいは「型安全である」という
- コンパイルの過程で型検査し、実行時に型情報を持たない言語も多い
- 静的型付けと呼ぶのはおすすめしない（が、よく言われている）

動的言語(dynamic language)

- プログラムの実行前に型検査合格を前提としない言語
- 実行時に「値」と「型タグ」をセットで扱い、処理しながら逐次型検査を行なう
- 実行時にソースコードを評価する（eval）など柔軟な機能を提供する
  - 原理上どんな文字列も実行できるので、無限の可能性を秘めている
- 動的型付けと呼ぶのはおすすめしない（が、よく言われている）

無限の彼方へ
さあ行くぞ！

pixiv

無限とか言われても困る

品質管理 vs 俺たち

- ソフトウェアを安定させるには、その挙動を予測可能な範囲に収めたい
- 例:
  メソッドに期待する形式の引数を渡せば、エラーなく実行されてほしい
  - メソッドに同じ引数を渡せば、決まった結果になってほしい
  - わざわざ実装にジャンプしなくても、引数の形式がわかりやすくなってほしい
- そこで、動的解析 ＝ ユニットテスト ＝ 動かして試してみる

PHP vs 型付け

- PHPは根本的に動的言語であり、ほとんどの特徴が当てはまる
- ただしPHPには型宣言があり、型宣言通りの値が渡されなければエラー
- 他言語でも型宣言は存在するが、実行時に保証されない口約束であることも
  - Python, TypeScript…
  - 一方でPHPでは動かしてみれば型宣言に反しているかがわかる

PHP 5.x

```php
/**
 * @param string $name
 * @param int $age
 * @return void
 */
function register_user($name, $age) {
    if (!is_string($name) ||
        !is_int($age)
    ) {
        throw new Exception('引数エラー');
    }

    // DB insert処理
}
```

PHP 5.x

```php
/**
 * @param string $name
 * @param int $age
 * @return void
 */
function register_user($name, $age) {
    if (!is_string($name) ||
        !is_int($age)
    ) {
        throw new Exception('引数エラー');
    }

    // DB insert処理
}
```

PHP 7.x

```php
function register_user(string $name, int $age): void
{
    // DB insert処理
}
```


pixiv

PHP 5.x

```php
/**
 * @param string $name
 * @param int $age
 * @return void
 */
function register_user($name, $age) {
    if (!is_string($name) ||
        !is_int($age)
    ) {
        throw new Exception('引数エラー');
    }

    // DB insert処理
}
```

PHP 7.x

```php
function register_user(string $name, int $age): void
{
    // DB insert処理
}
```

安全性を落とさず
本質的な実装に集中できる

pixiv

```php
function register_user(string $name, int $age): void
{
    // DB insert処理
}

register_user(123, 'PHP Taro');
```


pixiv

```php
function register_user(string $name, int $age): void
{
    // DB insert処理
}

register_user(123, 'PHP Taro');
```

コーディングしながら
即座に間違いに気付きたい

pixiv

```php
function register_user(string $name, int $age): void
{
    // DB insert処理
}

register_user(123, 'PHP Taro');
```

コーディングしながら
即座に間違いに気付きたい

pixiv

```php
function register_user(string $name, int $age): void
{
    // DB insert処理
}

register_user(123, 'PHP Taro');
```

コーディングしながら
即座に間違いに気付きたい

Found 2 errors

Line	Error
8	Parameter #1 $name of function register_user expects string, int given.
8	Parameter #2 $age of function register_user expects int, string given.

pixiv

PHPStan

- **PHPStan** - PHP Static Analysis Tool
  - フルネームで呼んでいる人は見たことがない
  - PHPで書かれたPHPの静的解析ツール
- チェコ人の**Ondřej Mirtes**が個人で開発している
  - 今年になってPHPStan s.r.o(有限会社)として法人化した

PHPUnitと同じようにCIで使うもの…

pixiv

と考えてもいいが、
まずは小さく
使ってほしい

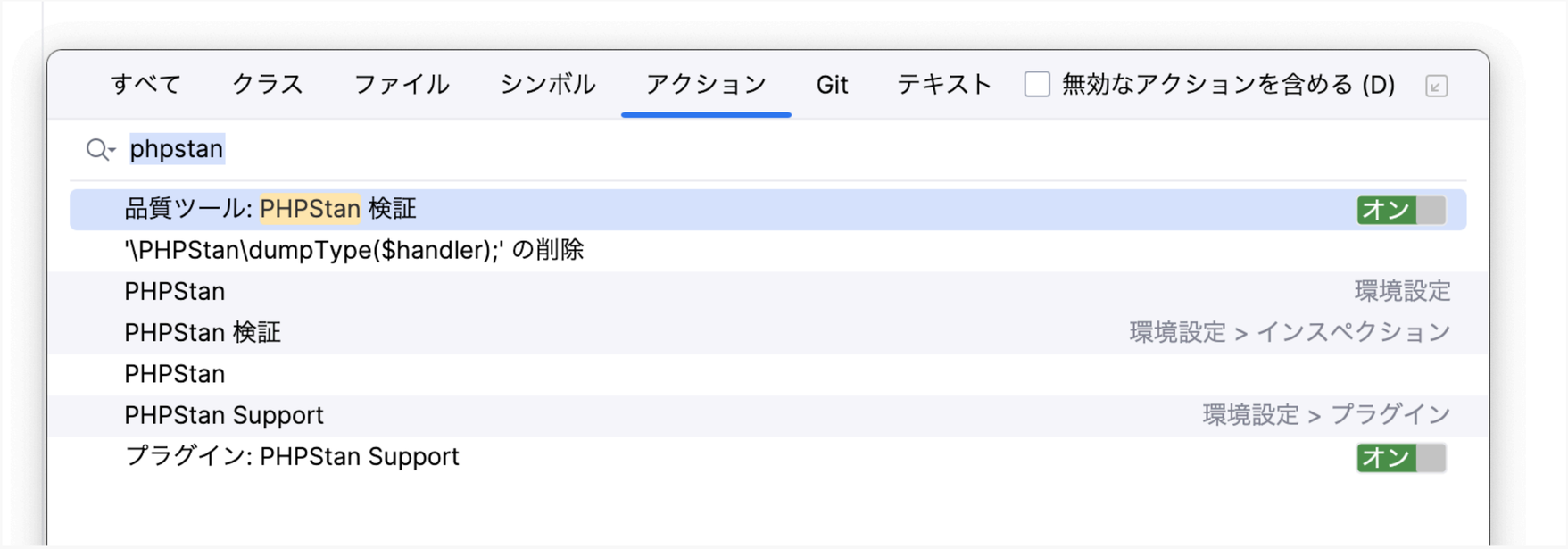
インストラクション
PHPStanを使い倒せ

PHPStanは
毎秒起動せよ

……どうやって？

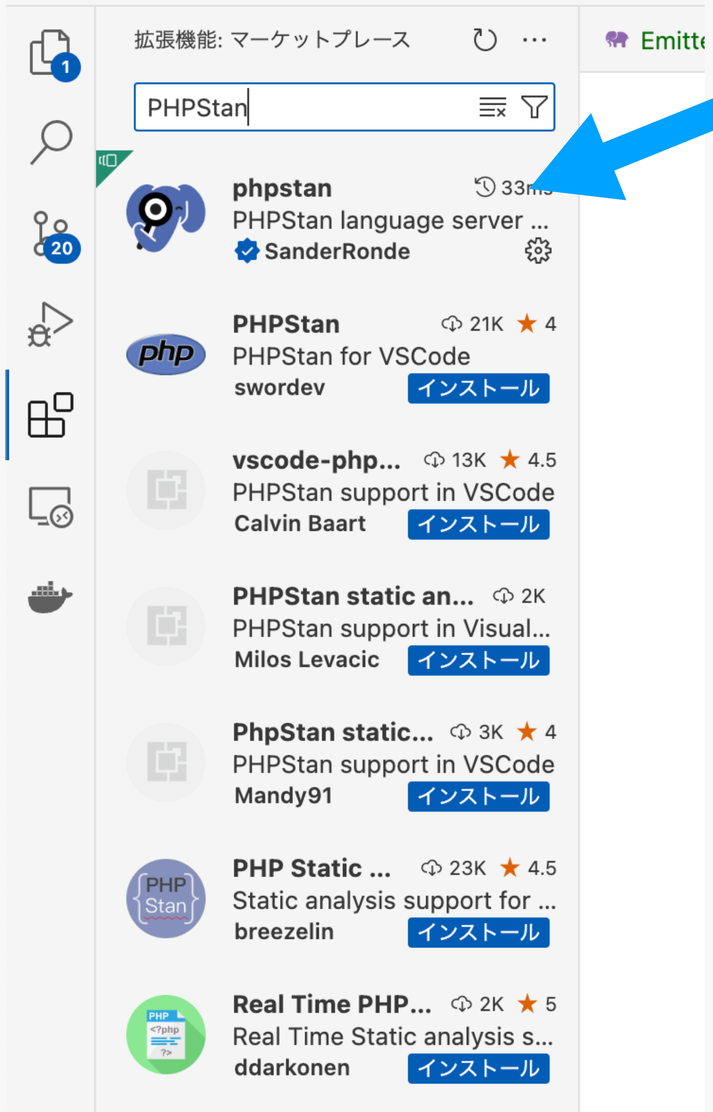
テキストエディタに組み込みましょう

PhpStormで簡単に有効化



pixiv

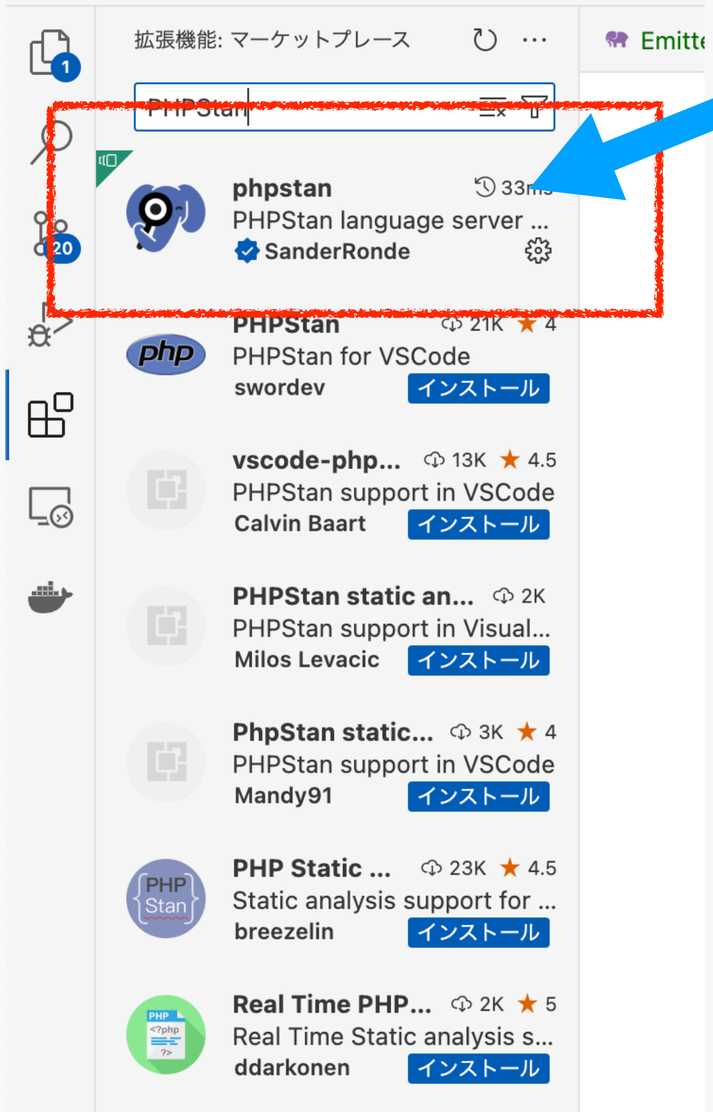
VS Codeの場合



おそらくSanderRondeが
いちばんよくできてる

pixiv

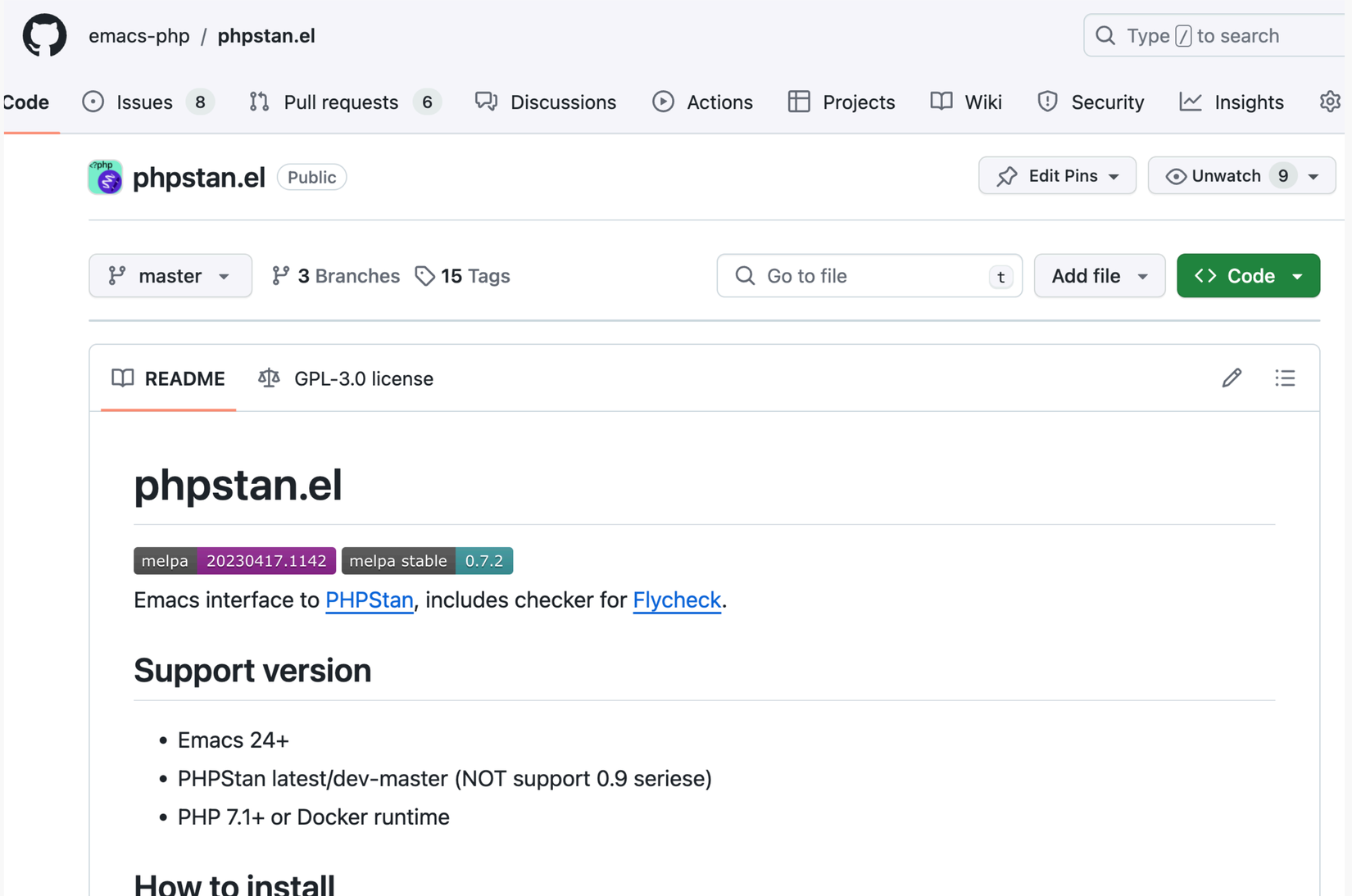
VS Codeの場合



おそらくSanderRondeが
いちばんよくできてる

pixiv

Emacsは私が作っています



pixiv

常時ペアプロ相手に
なってくれてる
ようなもの

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

/**
 * @param list<int> $a1
 * @param non-empty-array<string> $a2
 */
function f(array $a1, array $a2): void
{
    $a3 = range(1, 10);
    \PHPStan\dumpType(['a1' => array_chunk($a1, 3)]);
    \PHPStan\dumpType(['a2' => array_chunk($a2, 3)]);
    \PHPStan\dumpType(['a3' => array_chunk($a3, 3)]);
}

```

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

/**
 * @param list<int> $a1
 * @param non-empty-array<string> $a2
 */
function f(array $a1, array $a2): void
{
    $a3 = range(1, 10);
    \PHPStan\dumpType(['a1' => array_chunk($a1, 3)]);
    \PHPStan\dumpType(['a2' => array_chunk($a2, 3)]);
    \PHPStan\dumpType(['a3' => array_chunk($a3, 3)]);
}

```

Line	Error
10	Dumped type: array{a1: list<non-empty-list<int>>}
11	Dumped type: array{a2: non-empty-list<non-empty-list<string>>}
12	Dumped type: array{a3: array{array{1, 2, 3}, array{4, 5, 6}, array{7, 8, 9}, array{10}}}

\PHPStan\dumpType()

```
20
21      /** @return array{ServerRequestInterface, ?ResponseInterface} */
22      public function interceptRequest(ServerRequestInterface $request): array
23      {
24          $handler = new InterceptChecker();
25          foreach ($this->interceptors as $interceptor) {
26              \PHPStan\dumpType($interceptor);
27                                                                  est);
28
29                  return [$request, $response];
30              }
31
32              $request = $handler->getRequest();
33          }
34
```

phpstan: Dumped type: Psr\Http\Server\MiddlewareInterface

pixiv

PHPStanの基本機能を使えるようになろう

Zenn

投稿する

pixiv Publicationへの投稿

Edit

PHPStanクイックガイド2023

2023/09/17に公開 2024/11/07

PHP PHPStan Tech

PHPStan（PHP Static Analysis Tool）はコードを実行せずに検査できるツールです。本稿では業務アプリケーションにPHPStanを導入するまでに押さえておきたい事柄を記述します。

この記事の初出はPHPerKaigi 2023のパンフレット原稿です。

にゃんだーすわん

にゃーん

バッジを贈る

バッジを贈るとは →

pixiv

README GFDL-1.3 license

PHPStan型付けチュートリアル

PHPStanを使った型付けをWebブラウザ上で体験できるチュートリアルです。

コード編集中にリアルタイムでPHPStanのフィードバックを得て「**PHPStanとペアプログラミングをする**」感覚を獲得してください。

チュートリアル

1. 🌱 **PHPStan型付けチュートリアル 入門編**
2. 🔰 **PHPStan型付けチュートリアル 基礎編** ***(工事中)***

取り組み方

チュートリアルには節見出しごとに、以下のようなブロックがあります。

無限の可能性を
認識可能な範囲内に
縮めるのが型

実行しなくても
わかること
＝ 静的に確認したい

実行して
確かめたいこと
= PHPUnit

静と動

静的解析とテスト

どちらかではなく
両輪

PHPStanが
変だなと思ったら

phpusers-ja Slack

pixiv

続きはWebで