



# PhpStorm 超絶技巧

## 40分集中講義

株式会社サムライズム 山本 裕介

#phpconkagawa

# 株式会社サムライズム

- ・ 2013年設立
- ・ JetBrainsの日本の総代理店
- ・ メール、Slackによる日本語テクニカルサポート
- ・ 製品トレーニング
- ・ プラグイン開発

 Window Resizer

 Clclone

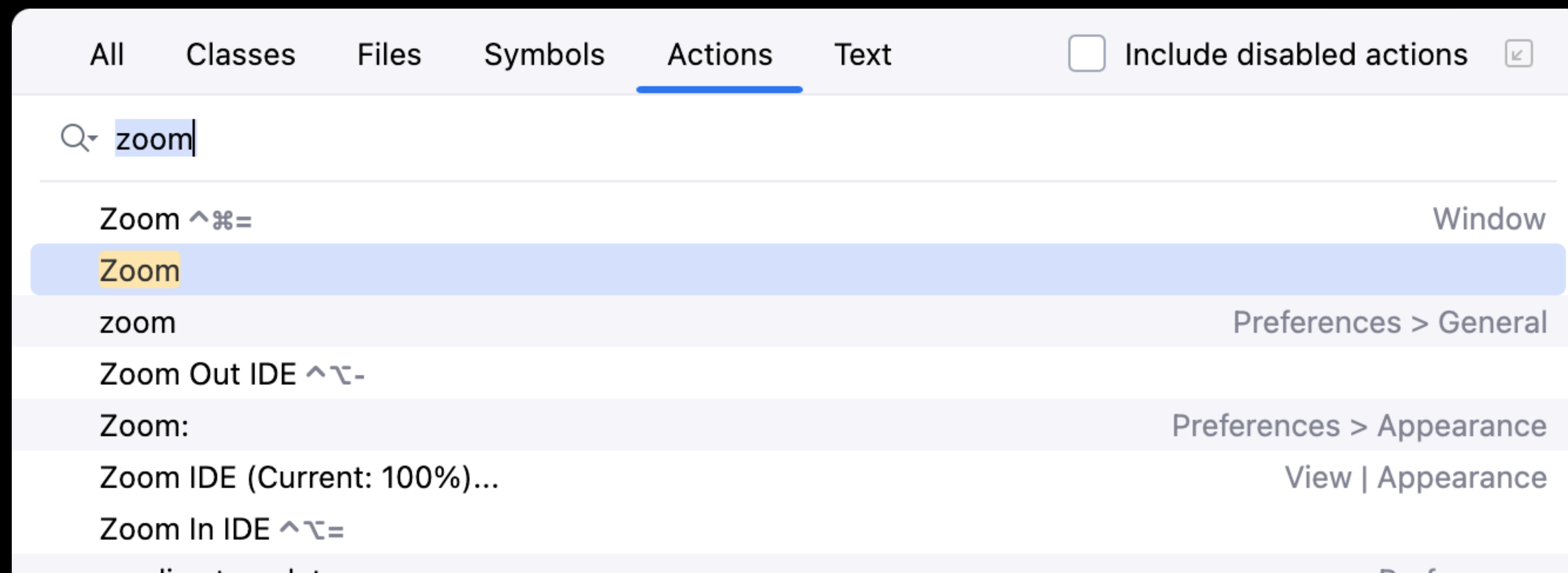
 TamaCat



# 画面制御 - 超絶技巧

# 画面のズーム

	Mac	Win/Linux
縮小	Ctrl + Opt + -	Alt + Shift + -
拡大	<del>Ctrl + Opt + =</del>	<del>Alt + Shift + =</del>
100%に戻す	Ctrl + Opt + 0	Alt + Shift + 0



# 画面レイアウトの制御

	Mac	Win/Linux
エディタ最大化/元に戻す	Shift + Cmd + F12	Ctrl + Shift + F12
レイアウトを保存	Window > Layouts > Save Changes in Current Layout	
レイアウトを復帰	Shift + F12	

# ウインドウのアレンジ


- Window Resizerプラグイン(無料)
- ウインドウサイズを揃えるのに便利
- 16:9、4:3、左半分、右半分などに調整可

The screenshot shows the IntelliJ IDEA Settings window with the 'Plugins' tab selected. The search bar contains 'window resizer'. The 'Marketplace' tab is active, showing a list of search results. The 'Window Resizer' plugin by Samuraism Inc. is highlighted. The 'User Interface' section of the plugin details is visible, showing the 'Window Resizer' title, version 1.6.0, and an 'Install' button. Below the title, there are tabs for 'Reviews', 'Additional Info', 'Overview', and 'What's New'. The 'Overview' tab is selected, displaying a list of predefined orientations/sizes: Maximum / Left / Right / Top / Bottom, 1920 x 1080, 1600 x 900, 1280 x 720, 854 x 480, 640 x 360, 426 x 240 (16:9), and 1200 x 760.

The screenshot shows the 'Window Resizer' plugin settings. The 'Resize' section is expanded, showing a list of predefined window sizes and orientations. The '16:9 1600 x 900' option is selected. The list includes:

- 16:9 1280 x 720
- 16:9 1920 x 1080
- 16:9 1600 x 900 (selected)
- 16:9 1280 x 720
- 16:9 854 x 480
- 16:9 640 x 360
- 16:9 426 x 240
- 16:9 1200 x 760
- 4:3 1920 x 1440
- 4:3 1600 x 1200
- 4:3 1280 x 960

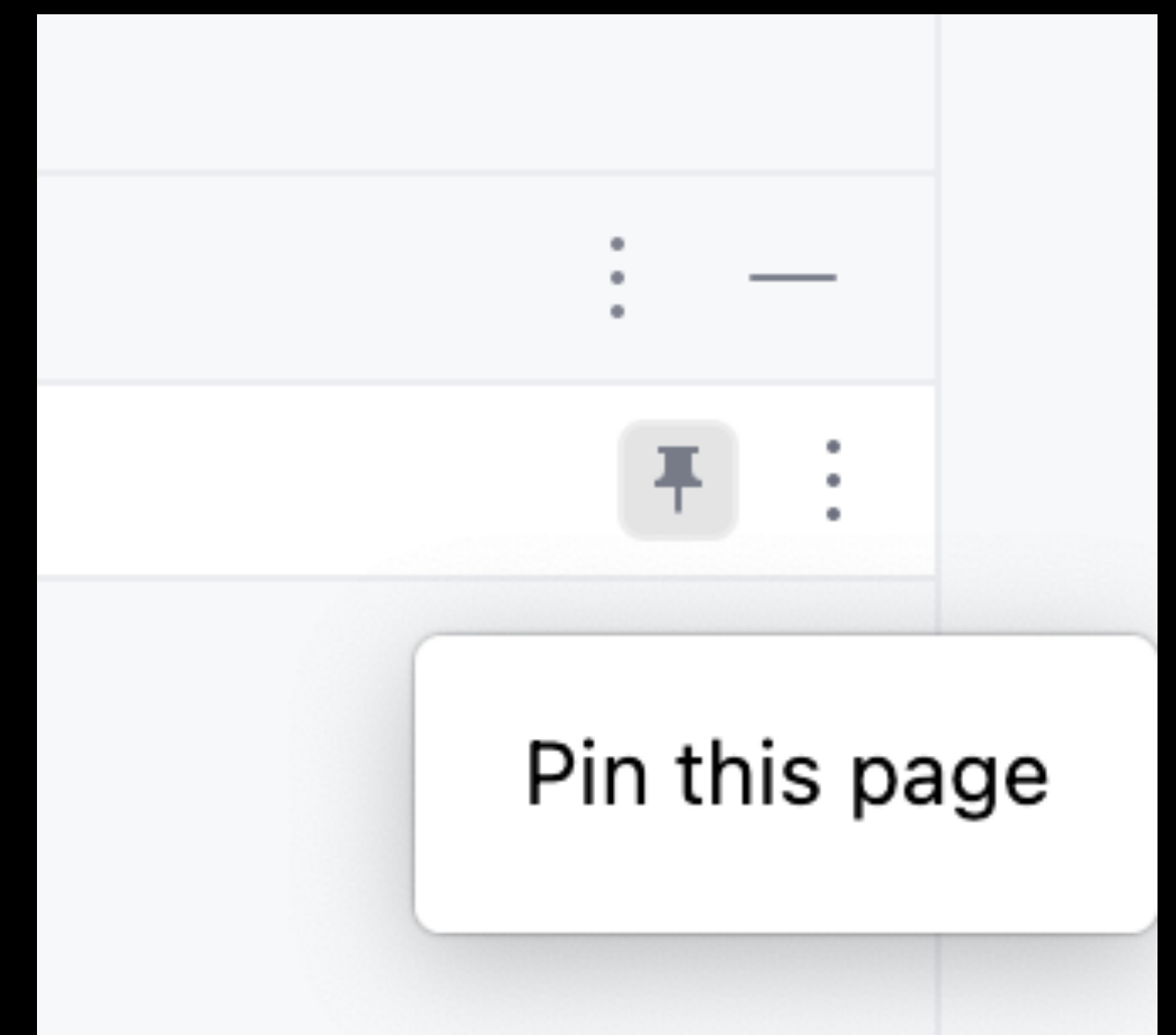
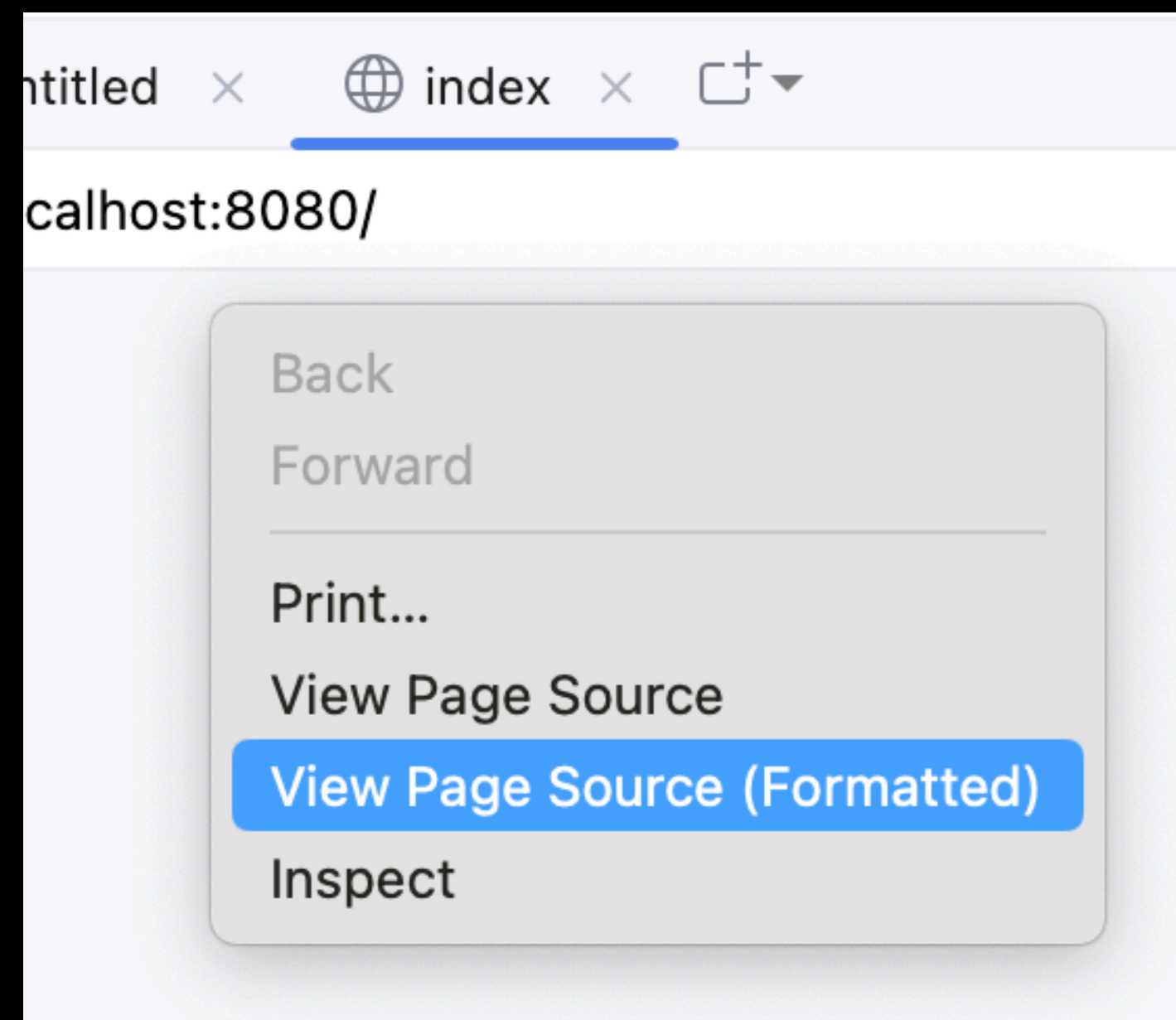
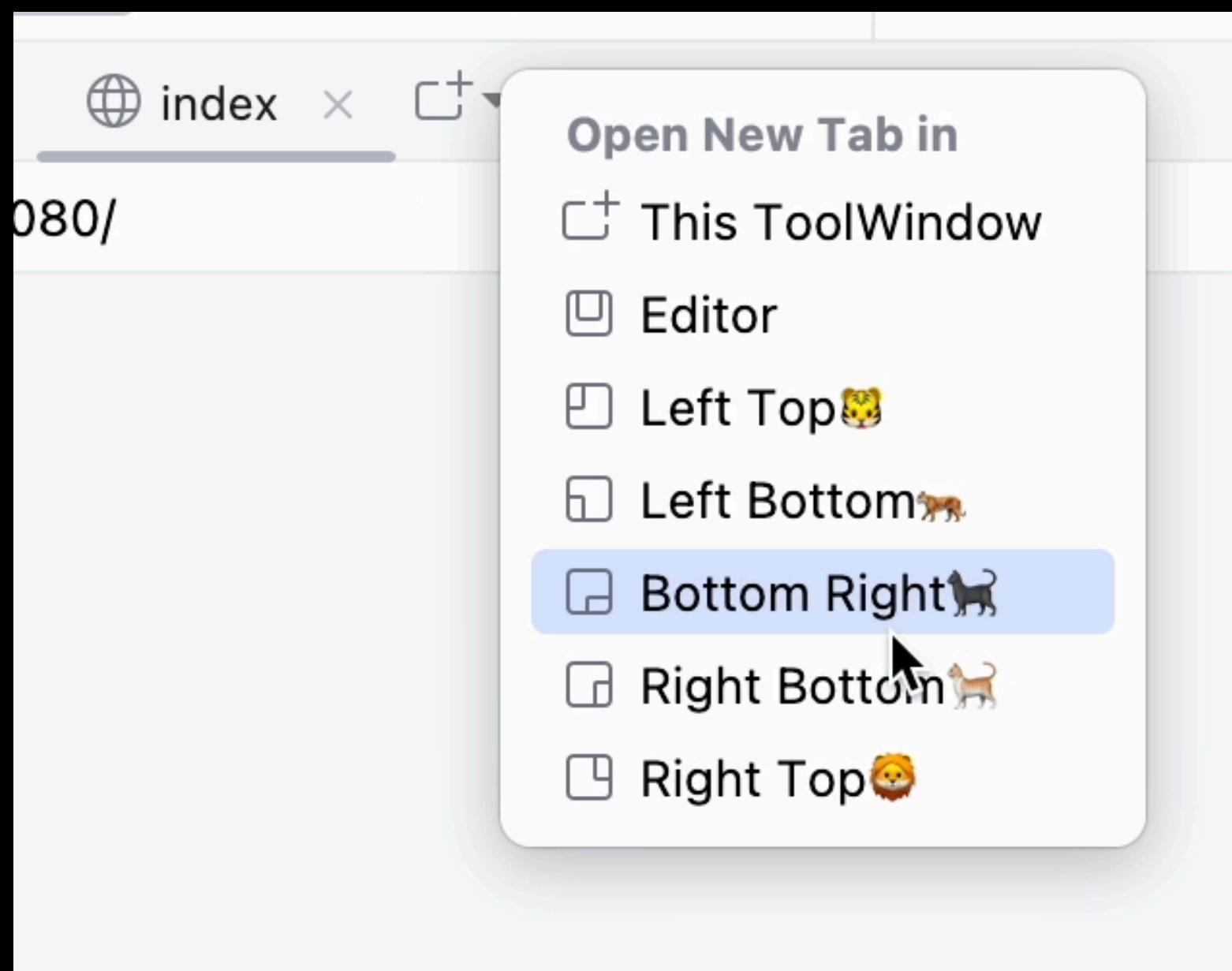
# Webアプリケーション開発するなら

- 弊社開発「TamaCat」プラグイン 
- PhpStorm 内にWebページを表示  
(Chromiumベース)
- DevTools など開発者向けの機能も



# TamaCat 🐾の主な機能

- 複数のツールウインドウ、エディタでページを表示
- 最初からフォーマットされた状態でソースを標示
- タブを固定(プロジェクトを開き直してもページが開かれる)



# 検索 - 超絶技巧

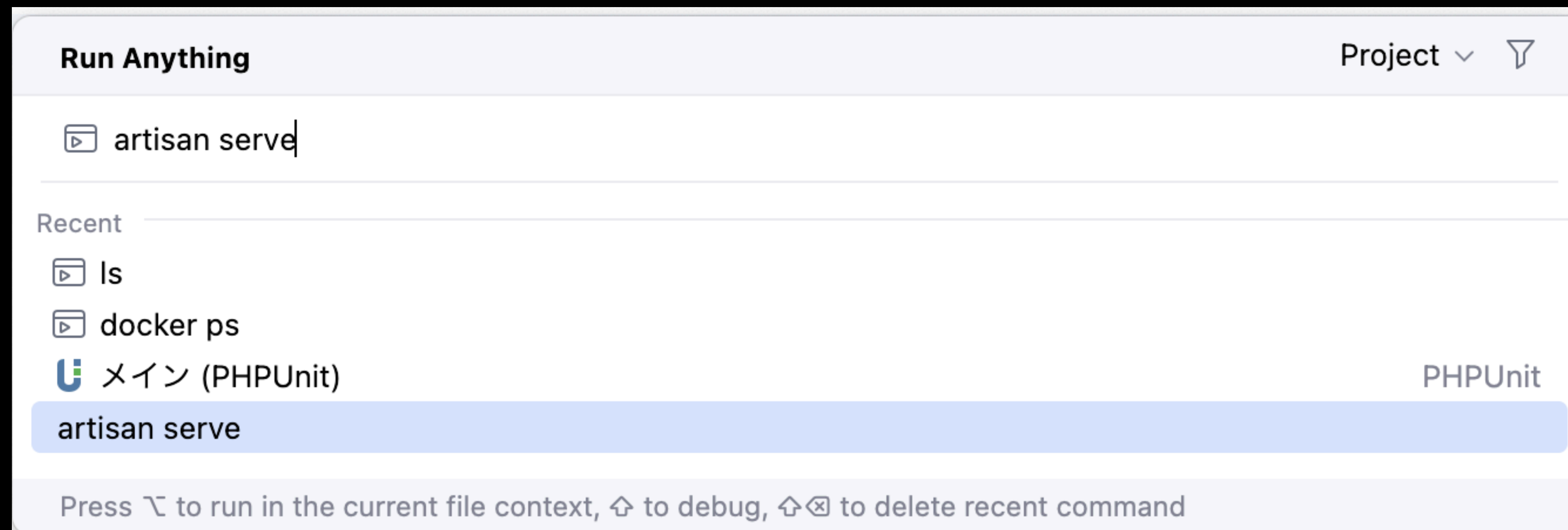
# 検索超絶技巧

	Mac	Win/Linux
どこでも検索 Search Everywhere		Shift * 2



# 検索/ファイルオープン超絶技巧

	Mac	Win/Linux
なんでも実行 Run Anything		Ctrl * 2



	Mac	Win/Linux
クラス名で開く	Cmd + O	Ctrl + N
シンボル名で開く メソッド/フィールド名など	Opt + Cmd + O	Shift + Ctrl + Alt + N
ファイル名で開く	Shift + Cmd + O	Shift + Ctrl + N
アクション名を開く	Shift + Cmd + A	Shift + Ctrl + A

# ナビゲーション - 超絶技巧

# ナビゲーション超絶技巧

	Mac	Win/Linux
プロジェクトツールウインドウ	Cmd + 1	Alt + 1
VCSツールウインドウ	Cmd + 9	Alt + 9
エディタ	ESC	

# マウスやトラックパッドでタブを切り替えるのはやめよう

The image shows a screenshot of an IDE (likely PhpStorm) with a code editor and a settings dialog. The code editor displays the following PHP code:

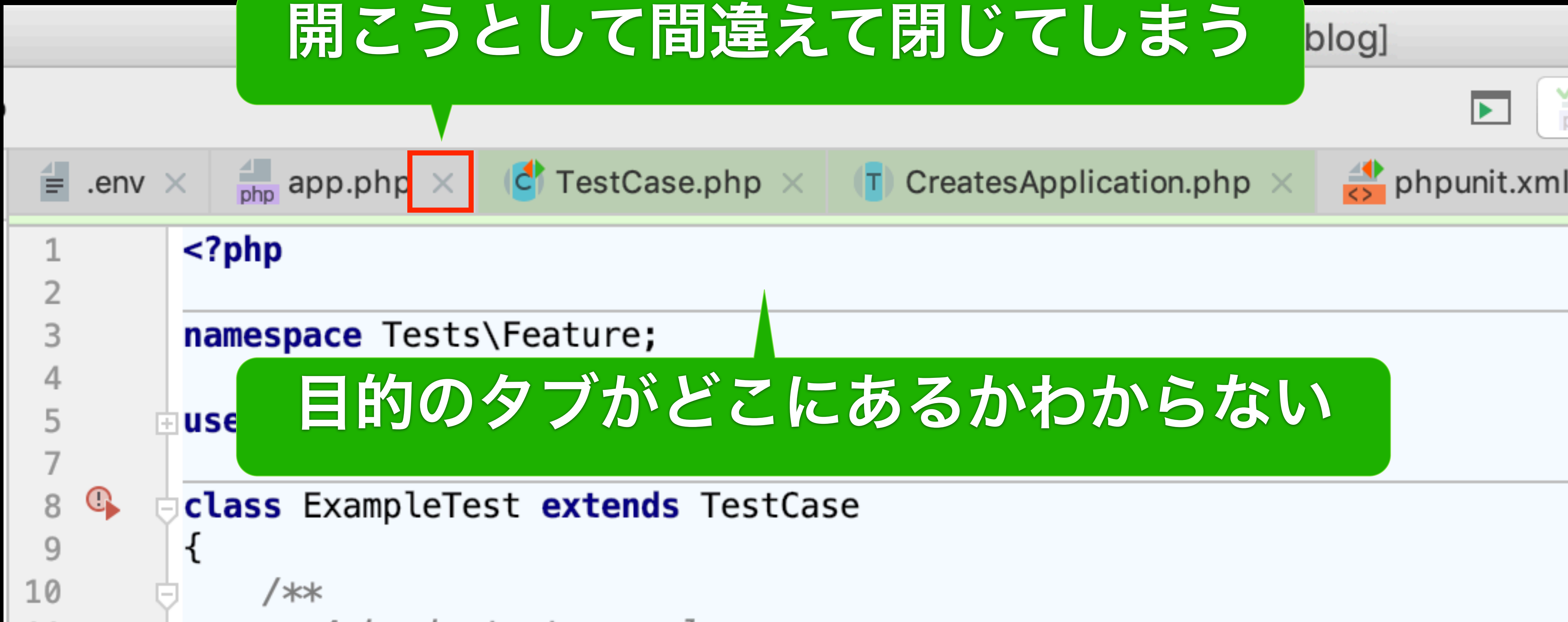
```
1 <?php
2
3 namespace Tests\Feature;
4
5 use ...
6
7
8 class ExampleTest extends TestCase
9 {
10     /**
11      * A basic test example.
12      *
13      * @return void
14      */
```

The settings dialog is open to the "Editor > General > Editor Tabs" section. The "Tab placement" dropdown is set to "None". The "Show tabs in:" section has "One row" selected. The "None" option in the "Show tabs in:" dropdown is highlighted.

The IDE interface also shows several open tabs: `.env`, `app.php`, `TestCase.php`, `CreatesApplication.php`, `phpunit.xml`, and `Feature/ExampleTest.php`. A red line is drawn across the tabs, indicating the action of switching tabs.

# タブの問題

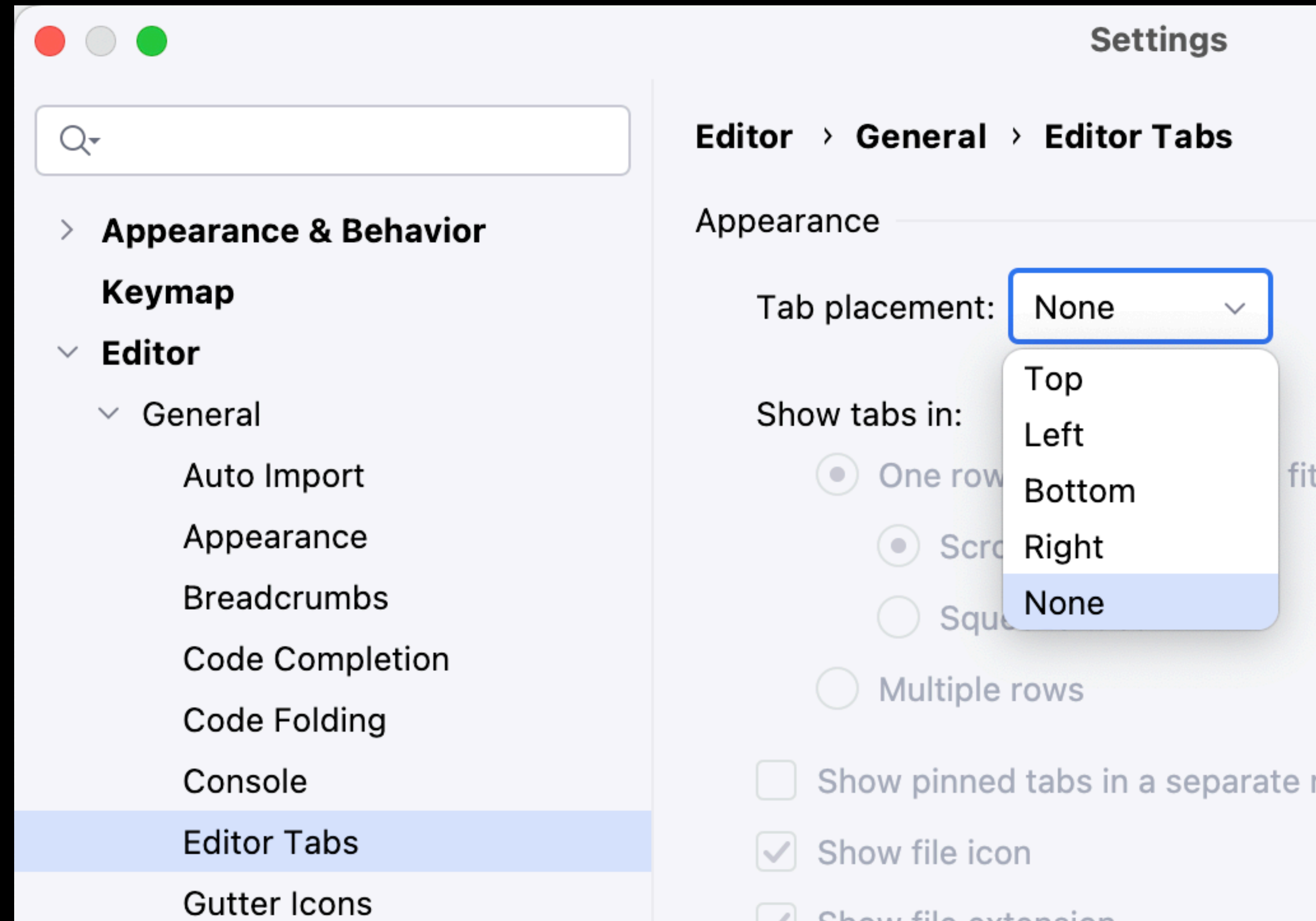
開こうとして間違えて閉じてしまう



目的のタブがどこにあるかわからない



# タブを非表示にして画面をスッキリ



# ナビゲーション超絶技巧

	Mac	Win/Linux
最近使用したファイル	Cmd + E	Ctrl + E
最近使用した箇所	Shift + Cmd + E	Shift + Ctrl + E
1つ前のファイル	Ctrl + Tab	
2つ前のファイル	Ctrl + Tab * 2	



# あらゆるポップアップはタイプして絞り込み可

The image displays two side-by-side screenshots of an IDE's file explorer, illustrating a search function. A large green arrow points from the left screenshot to the right one.

**Left Screenshot:** The 'Recent Files' list is shown. The file 'Feature/ExampleTest.php' is selected and highlighted in blue. Other files in the list include 'Response.php', 'TestCase.php', 'ResponseTrait.php', 'storage/app/.gitignore', 'Unit/ExampleTest.php', '.env', 'config/app.php', 'CreatesApplication.php', 'phpunit.xml', 'index.php', 'Dockerfile', '.htaccess', and '503.svg'. The path at the bottom is '~/.laravel/blog/tests/Feature'.

**Right Screenshot:** The search bar at the top left contains the text 'Search for: test'. The 'Recent Files' list is filtered to show only files containing 'test': 'Feature/ExampleTest.php', 'TestCase.php', and 'Unit/ExampleTest.php'. The file 'TestCase.php' is selected and highlighted in blue. The text 'Nothing to show' is visible in the main area. The path at the bottom is '~/.laravel/blog/tests'.

# ナビゲーション超絶技巧

	Mac	Win/Linux
定義箇所へ	Cmd + B	Ctrl + B
利用箇所ポップアップ	Opt + Cmd + F7	Ctrl + Alt + F7
利用箇所ポップアップ	(定義箇所) Cmd + B	(定義箇所) Ctrl + B

# ナビゲーション超絶技巧

	Mac	Win/Linux
スーパークラスへ	Cmd + U	Ctrl + U
子クラス(実装)へ	Opt + Cmd + B	Ctrl + Alt + B
指定行へ	Cmd + L	Ctrl + G
指定クラス/指定行へ	(検索ポップアップで) 検索キー:[行番号]	


# 補完超絶技巧

- TabとEnterの補完の違いを理解

# 補完超絶技巧

- TabとEnterの補完の違いを理解

```
$response->assertStatus( status: 200);
```



# 補完超絶技巧

- TabとEnterの補完の違いを理解

```
$response->assertStatus( status: 200);  
m assertCookie(cookieName :  
m assertCookieExpired(cookie  
m assertCookieMissing(cookie  
m assertCookieNotExpired(coo  
m assertDontSee(value : stri  
m assertDontSeeText(value :  
m assertExactJson(data : arr  
m assertForbidden()  
m assertHeader(headerName :
```

- Enterで補完すると…

- Tabで補完すると…



# 補完超絶技巧

- TabとEnterの補完の違いを理解

```
$response->assertStatus( status: 200);  
m assertCookie(cookieName :  
m assertCookieExpired(cookie  
m assertCookieMissing(cookie  
m assertCookieNotExpired(coo  
m assertDontSee(value : stri  
m assertDontSeeText(value :  
m assertExactJson(data : arr  
m assertForbidden()  
m assertHeader(headerName :
```

- Enterで補完すると…

```
$response->assertCookie()tStatus(200);
```

- Tabで補完すると…

# 補完超絶技巧

- TabとEnterの補完の違いを理解

```
$response->assertStatus( status: 200);  
m assertCookie(cookieName :  
m assertCookieExpired(cookie  
m assertCookieMissing(cookie  
m assertCookieNotExpired(coo  
m assertDontSee(value : stri  
m assertDontSeeText(value :  
m assertExactJson(data : arr  
m assertForbidden()  
m assertHeader(headerName :
```

- Enterで補完すると…

```
$response->assertCookie()tStatus(200);
```

- Tabで補完すると…

```
$response->assertCookie( cookieName: 200);
```



# 補完超絶技巧

- TabとEnterの補完の違いを理解
- 基本、Tabで補完

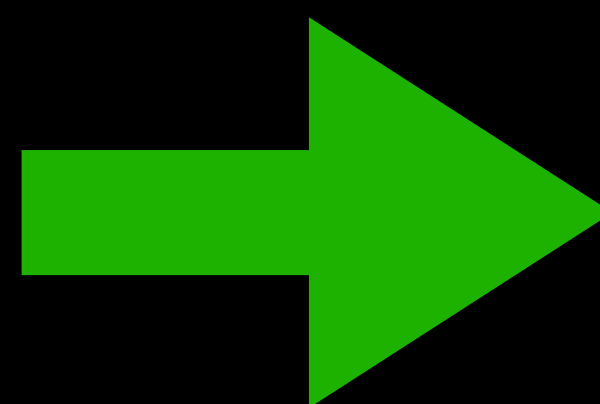
# 補完 - 超絶技巧

# 補完超絶技巧

- ・セミコロンは書かなくていいです

	Mac	Win/Linux
ステートメント補完	Shift + Cmd + Enter	Shift + Ctrl + Enter

```
echo "1+1 = ".(1+1|
```



```
echo "1+1 = " . (1 + 1);
```

# 補完超絶技巧

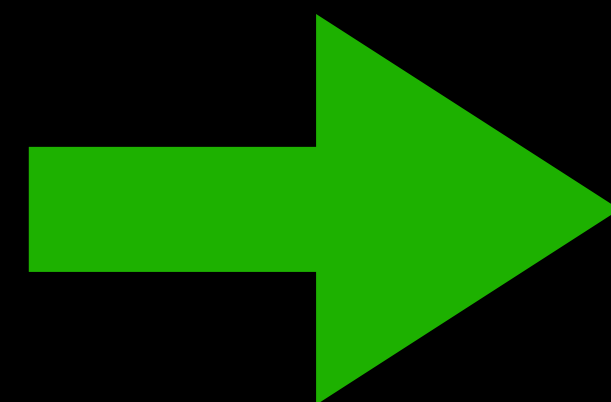
- Live Templateを使いこなせ
- “**fun**” + tab → “function (){}”



# 補完超絶技巧

- Live Templateを使いこなせ
- “**fori**” + tab → “for (\$i = 0; \$i < ; \$i++) {”

```
1 <?php
2 fori|
```



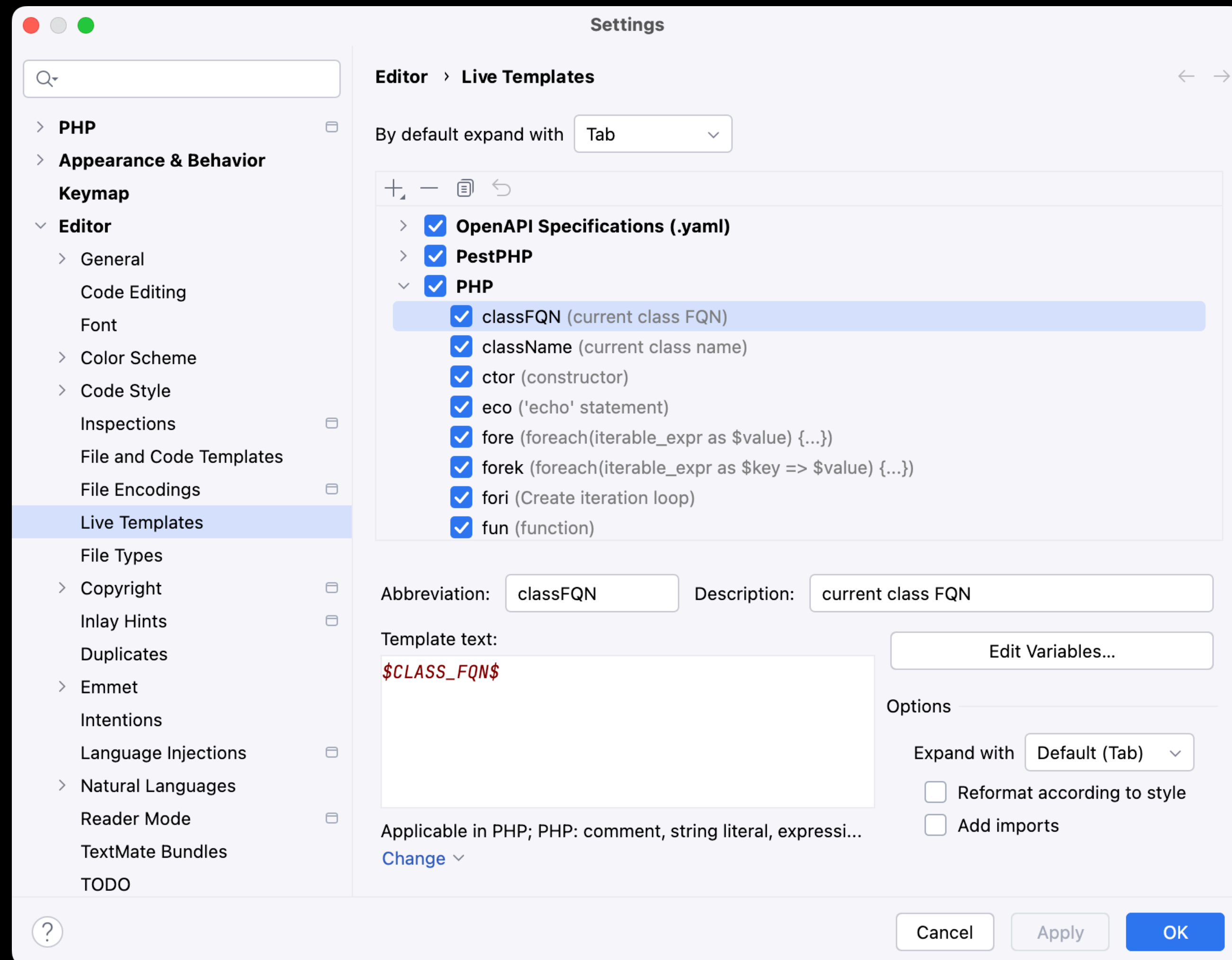
```
1 <?php
2 for ($i = 0; $i < ; $i++) {
3
4 }
```

\$i

Press ↵ or → to replace

# 補完超絶技巧

- 設定 > Live Templateより一覧可能



# 補完超絶技巧

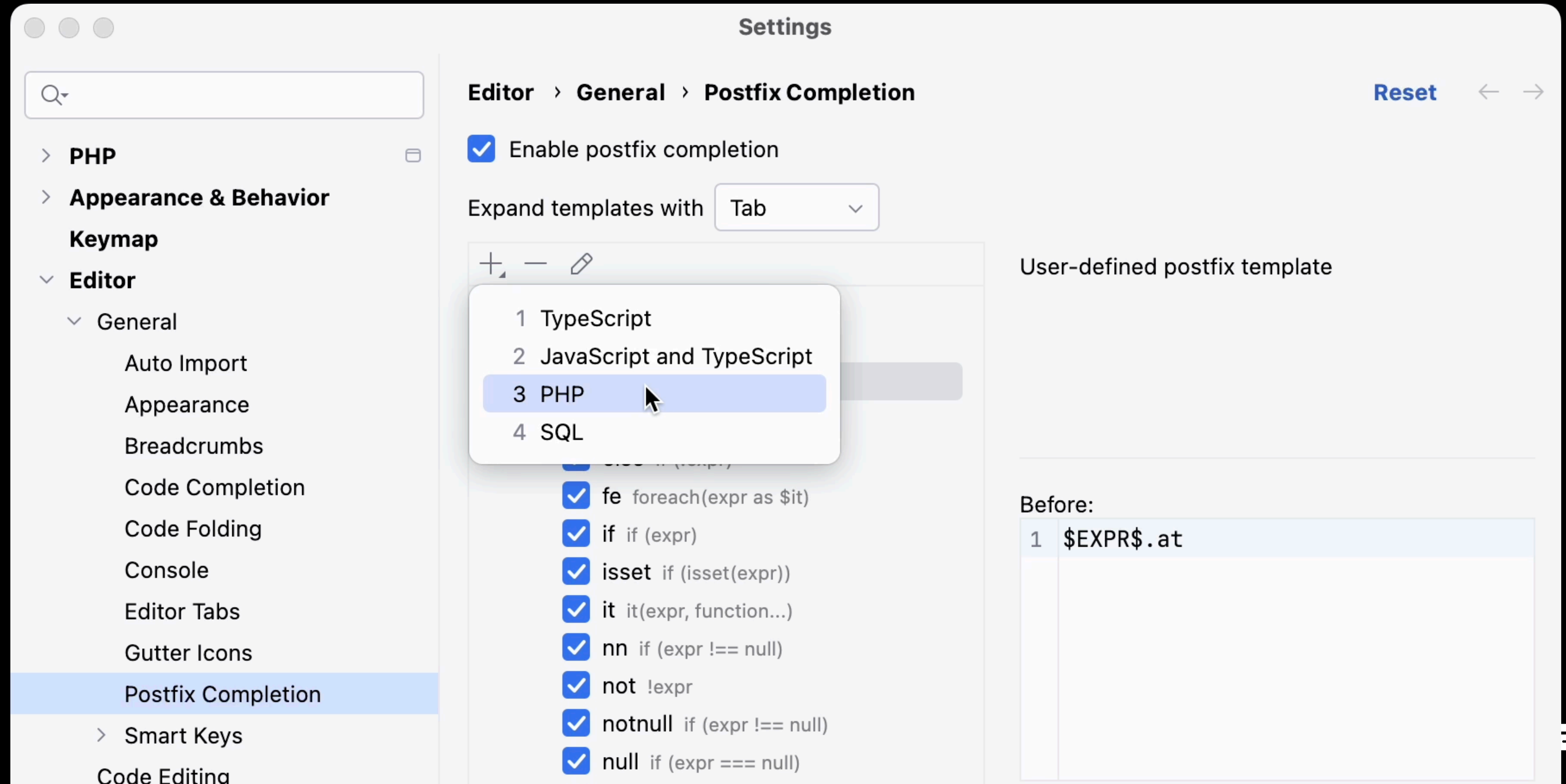
- 後置補完 (Postfix completion) を使いこなせ
- “式.**if**” + TAB → “if(式){”
- “式.**echo**” + TAB → “echo 式;”
- PhpStorm 「設定」内の「後置補完」より一覧可能

上から下へ、左から右へ、カーソルの戻りを  
極力抑えて流れるように書ける



# 補完超絶技巧

- Postfix completionはカスタム定義出来る！





# 補完超絶技巧

- 例えばユニットテストをスラスラ書けるカスタム定義

Edit 'at' Template

Key:

Apply to the topmost expression

```
$this->assertTrue($EXPR$);
```

Use \$EXPR\$ variable to refer target expression

Create New PHP Template

Key:

Apply to the topmost expression

```
$this->assertFalse($EXPR$);
```

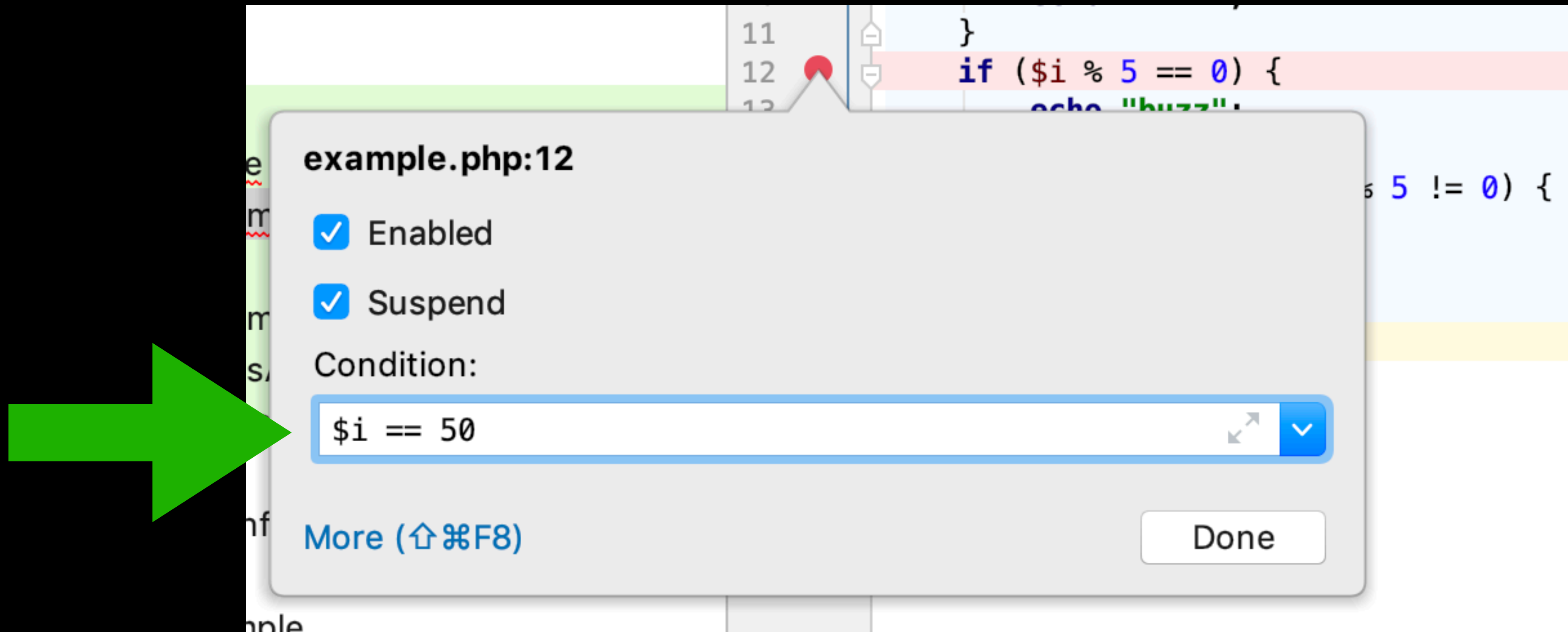
Use \$EXPR\$ variable to refer target expression

# デバツガ超絶技巧

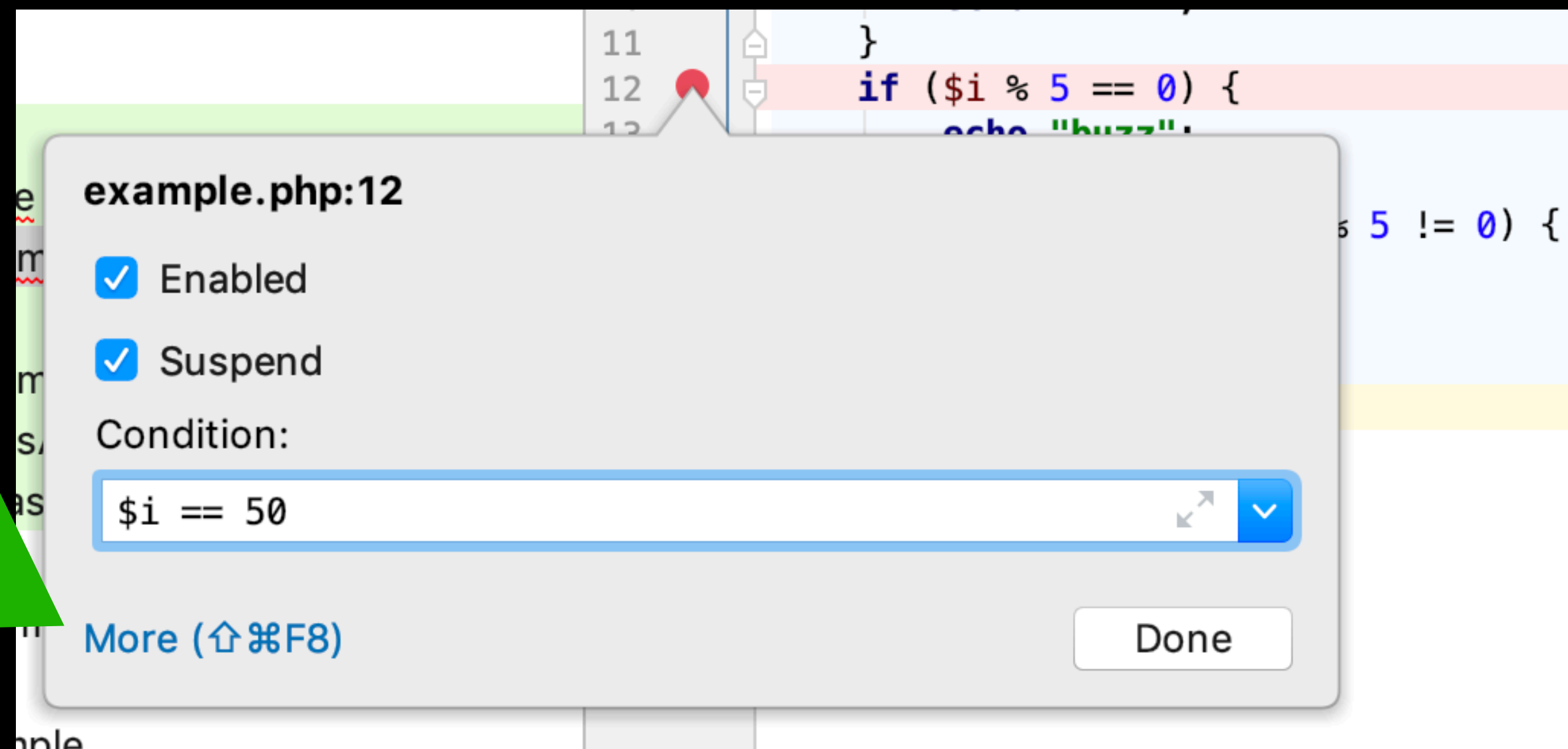
- ブレークポイントで止めるだけがデバツガじゃない
- echo でデバッグしなくても良い

# ブレークポイントを右クリック

- ・ 停止条件の設定



- 詳細設定



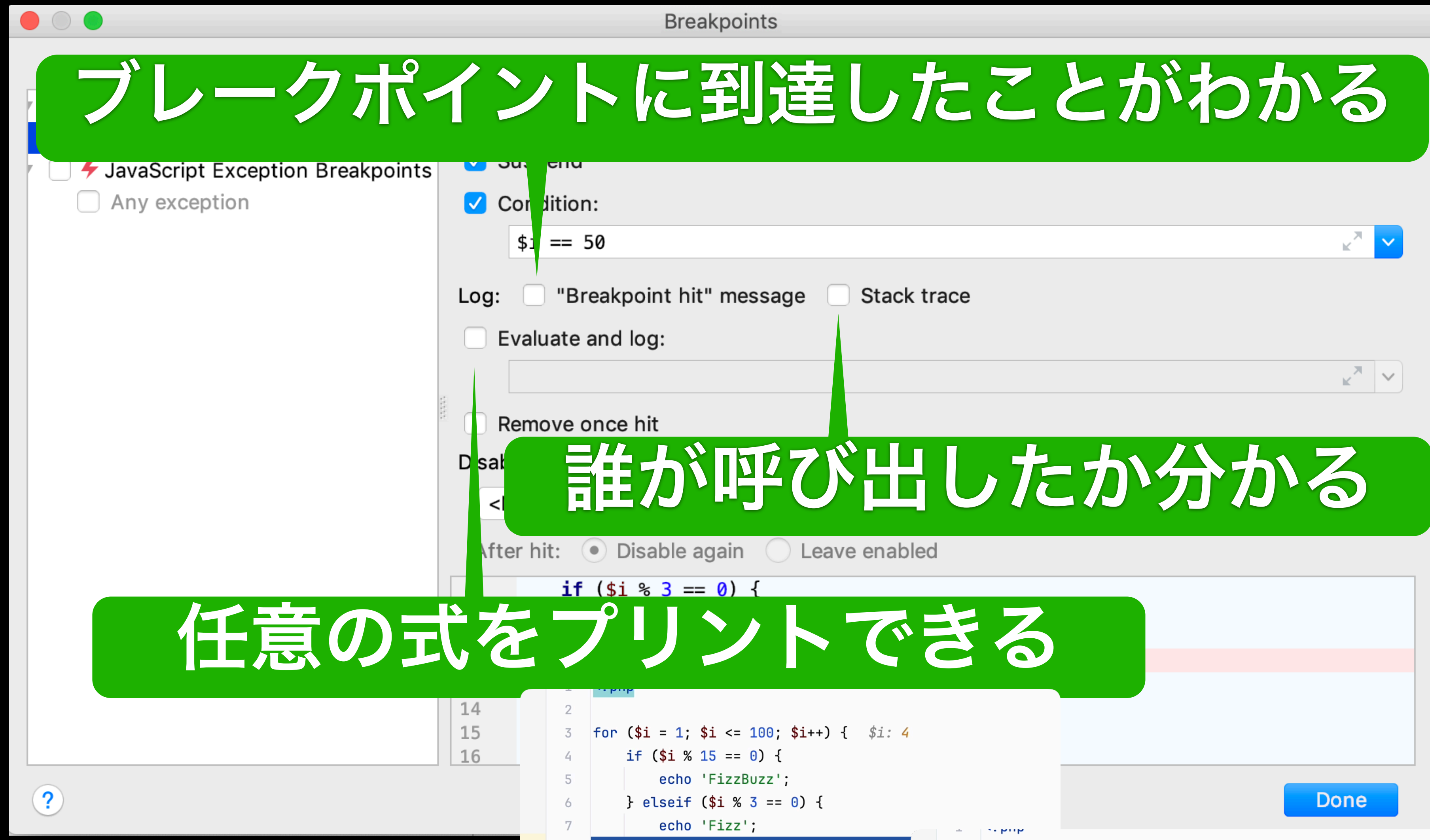
# 色々できるブレイクポイント

- ・ 停止条件の設定

ブレイクポイントに到達したことがわかる

誰が呼び出したか分かる

任意の式をプリントできる



# マウスで簡単制御

- マウスホバーで「ここまで実行」(Run to Cursor)

```
3 for ($i = 1; $i <= 100; $i++) { $i: 4
4     if ($i % 15 == 0) {
5         echo 'FizzBuzz';
6     } elseif ($i % 3 == 0) {
7         echo 'Fizz';
8     } elseif ($i % 5 == 0) { $i: 4
9         echo 'Buzz';
10    } else {
11        echo $i;
12    }
13 }
14 }
```

Run to Cursor `\F9`



# マウスで簡単制御

- マウスホバーで「実行再開」 (Resume Program)

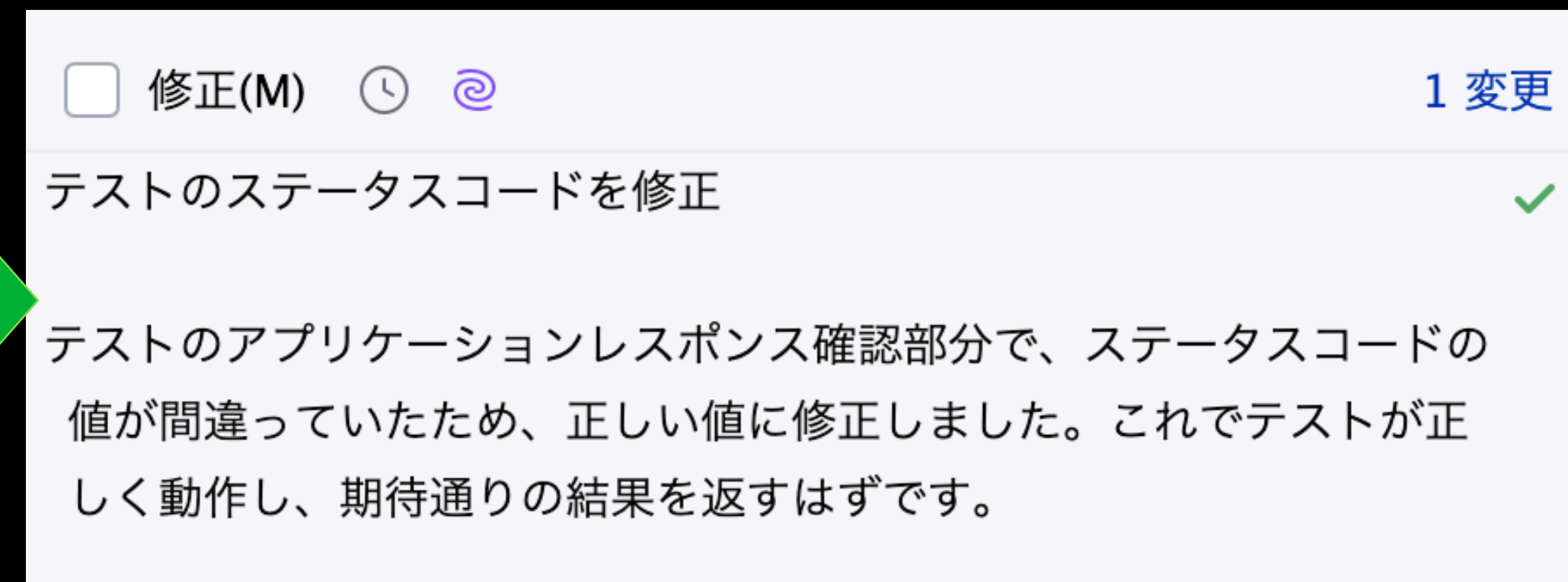
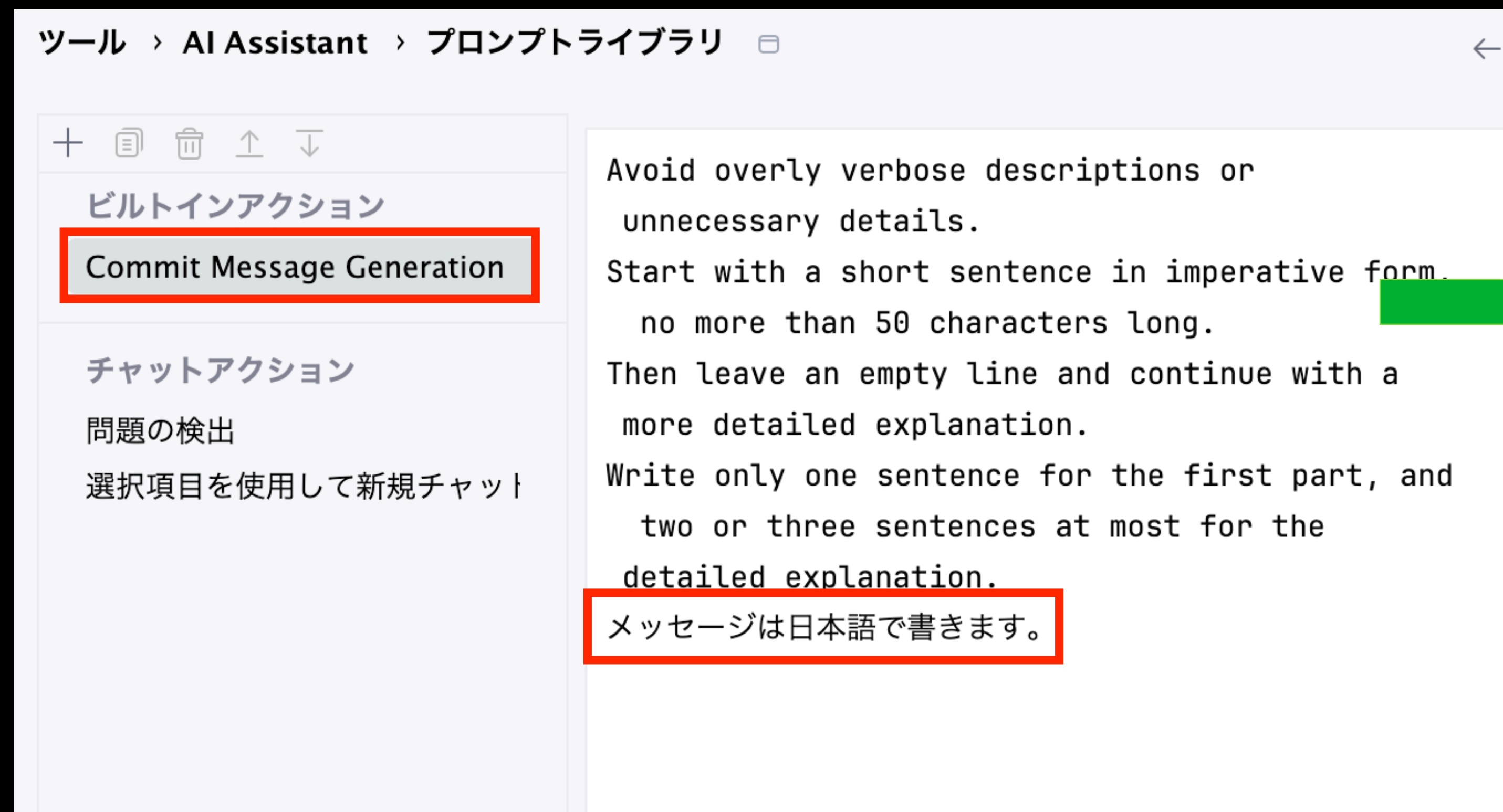
```
3  for ($i = 1; $i <= 100; $i++) {  $i: 4
4      if ($i % 15 == 0) {
5          echo 'FizzBuzz';
6      } elseif ($i % 3 == 0) {
7          echo 'Fizz';
8      } elseif ($i % 5 == 0) {
9          echo 'Buzz';
10     } else {
11         echo $i;  $i: 4
12     }
13
14 }
```

Resume Program ⌘⌘R



# PhpStormを使うと

- AI を使ってコミットメッセージを自動的に書ける
- PhpStorm 2024.1からは日本語で書いてくれる



# AI - 超絶技巧

# AI 機能

- リファクタリング
- コードの説明
- 別の言語に変換など

# AI 機能

- 右クリックして「AIアクション」

```
11 * A basic test example.  
12 *  
13 * @return void  
14 */  
15 <img alt="play button icon" data-bbox="50 470 65 485"/> mike_neck *  
16 public function test_the_applica  
17 {  
18     $response = $this->get( uri:  
19     $response->assertCookie( coo  
20 }  
21 }  
22
```

コンテキストアクションの表示

- AI アクション
- 切り取り ⌘X
- コピー ⌘C
- 貼り付け ⌘V
- 特殊なコピー / 貼り付け >
- 列選択モード ⇧⌘8
- 複数ファイル内検索
- 移動 >
- 折りたたみ >
- Google 検索

- コードの説明
- リファクタリングの提案
- 問題の検出
- 選択項目を使用して新規チャットを開く
- ドキュメントの記述
- コードの生成...
- ファイルを別の言語に変換
- プロンプトの追加...

# AI 機能

- 「diffを確認して適用」

The screenshot displays an IDE interface with a code editor on the left and an AI Assistant panel on the right. A green callout box highlights the text: 「AIのリファクタリング結果を比較し、必要に応じて適用できる」. The code editor shows PHP code with two lines (17 and 19) marked with checkmarks and red boxes, indicating they are ready for application. The AI Assistant panel shows the refactored code and a button labeled 「diffを確認して適用」, which is also highlighted with a red box.

ExampleTest.php リファクタリング: ×

すべて承認 統一 2分割

バージョン  
リファクタリングさ

11 11  
12 12  
13 13  
14 14  
15 15  
16 16  
17 ✓ \$response = \$this->get('/');  
17 const HOME\_ROUTE = '/';  
18 const EXPECTED\_COOKIE = 300;  
18 19  
19 ✓ \$response->assertCookie(300);  
20 public  
21 function test\_the\_application\_returns\_a\_successful\_response()  
22 {  
23 \$this->makeRequestAndAssertCookie();  
24 }  
25  
26 private  
27 function makeRequestAndAssertCookie()  
28 {  
29 \$response = \$this->get(self::HOME\_ROUTE);  
30 \$response->assertCookie(self::EXPECTED\_COOKIE);  
31 }

AI Assistant すべてのチャット 新規チャット

- Define your route '/' to a constant. This would be beneficial if you were to change the route in the future.
- Extract the entire logic of making a GET request and asserting the cookie to a well named method to enhance readability.

Here's the refactored code:

PHP diffを確認して適用

```
const HOME_ROUTE = '/';  
const EXPECTED_COOKIE = 300;  
  
public function  
test_the_application_returns_a_successful_response()  
{  
    $this->makeRequestAndAssertCookie();  
}  
  
private function makeRequestAndAssertCookie()  
{  
    $response = $this->get(self::HOME_ROUTE);  
    $response->assertCookie  
        (self::EXPECTED_COOKIE);  
}
```



# AI 機能

- お馴染みの「空気を読む」アクションより呼びだし可能

	Mac	Win/Linux
空気を読む	Opt + Enter	Alt + Enter
AIアクション (自分で割り当て)	Ctrl + Cmd + Enter	Ctrl + Alt + @

```
$response = $this->get( uri: '/' );
```

\$respo

- PHPDoc コメントの追加
- テンプレートデータ言語の選択
- 変数のインライン化
- @ AI アクション...

F1 を押すとプレビューに切り替わります

インテンション設定の編集

'AI アクション...' を無効にする

ショートカットの割り当て...

例えばMacなら  
「Ctrl + Cmd + Enter」を割り当て

# CI - 超絶技巧



# CI連携 - Clclone プラグイン



- CIサービス (GitHub Actions、 Jenkins、 CircleCI、 AWS CodeBuild、 GitLab) と PhpStorm を連携

- PhpStorm 内で


サムライズムから PhpStorm を購入  
した場合は Clclone プラグイン 付属

- ビルド完了通知！

- 失敗したテストケース一覧とローカル実行

# CI連携 - Clclone プラグイン

- ビルド完了通知！

 **GitHub Actions ワークフローが完了しました**  
失敗  
ツールウィンドウで開く

自分がプッシュしたビルドのみ  
完了通知をすることも可能

# CI連携 - Circle プラグイン

- 失敗したテストケース一覧とローカル実行

```
53 3) Tests\Unit\DataTest::testAdd with data set #1 (0, 1, 2)
54 Failed asserting that 1 is identical to 2.
55
56 /home/runner/work/my-first-php/my-first-php/tests/Unit/DataTest.php:1
57
58 4) Tests\Unit\DataTest::testAdd with data set #3 (1, 1, 3)
59 Failed asserting that 2 is identical to 3.
60
61 /home/runner/work/my-first-php/my-first-php/tests/Unit/DataTest.php:1
62
63 5) Tests\Unit\NormalFailureTest::test_normal_failure
64 Failed asserting that 3 matches expected 2.
65
66 /home/runner/work/my-first-php/my-first-php/tests/Unit/NormalFailureT
67
68 6) Tests\Unit\NormalFailureTest::testNormalFailure
69 Failed asserting that -1 matches expected 1.
70
71 /home/runner/work/my-first-php/my-first-php/tes
72
```

- 失敗したテスト
  - 1) Tests\Unit\NormalFailureTest::testExceptionInApp
  - 1) Tests\Unit\ArrayTest::testPush
  - 2) Tests\Unit\ArrayTest::testCompare
  - 3) Tests\Unit\DataTest::testAdd with data set #1 (0, 1, 2)
  - 4) Tests\Unit>DataTest::testAdd with data set #3 (1, 1, 3)
  - 5) Tests\Unit\NormalFailureTest::test\_normal\_failure
  - 6) Tests\Unit\NormalFailureTest::testNormalFailure
  - 1) Tests\Unit\ArrayTest::testNoAssertion

ソースへ移動

ローカルで実行

PhpStorm内で一覧表示

ダブルクリックで該当のソース行へジャンプ

# 節約 - 超絶技巧

割安にPhpStormを使いたい！

# 割安にPhpStormを使いたい！

- 日本で購入

税別価格	コマーシャル	パーソナル
	\$249 (38,769円)	\$99 (15,414円)
	32,370円	12,870円



# 割安にPhpStormを使いたい！

- ・サムライズムで購入

PhpStorm 税別価格	コマーシャル	パーソナル
直販	32,370円	12,870円
サムライズム	32,170円 (ライトプラン)	12,670円



# 割安にPhpStormを使いたい！

- ClcloneもTamaCatも使いたい！

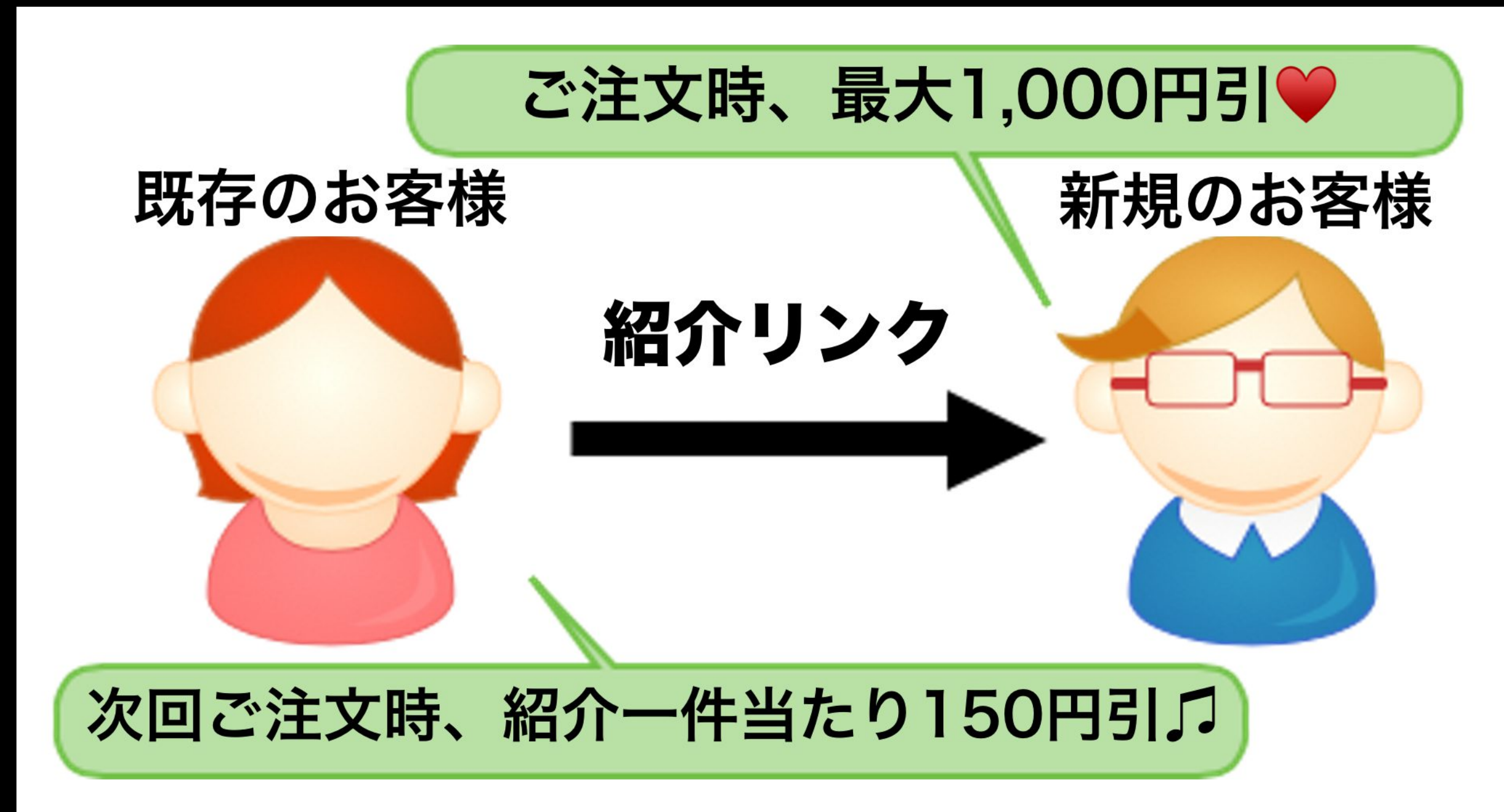
PhpStorm+TamaCat+Clclone 税別価格	コマーシャル	パーソナル
直販	53,170円	23,270円
サムライズム	35,000円 (レギュラープラン)	12,670円

# 割安にPhpStormを使いたい！

- JetBrains直販ライセンスをサムライズムで更新可能
- 継続ディスカウントも引き継がれます
- ブースでディスカウントクーポンお配りしてます

# 割安にPhpStormを使いたい！

- ・紹介プログラムで更新が割安に



<https://samuraism.com/referral-program>

- ・ブログ・ツイートで紹介すると効果絶大

# まとめ

- PhpStormのほとんどの操作はキーボードから行える
- 頻繁に行う操作はショートカットを覚えましょう
- git操作は PhpStorm から
- AI や CI 連携も活用してより便利に



# ブース





ご清聴ありがとうございました