

Xdebug を Windows で使う

Ver 1.5

Seiichi Nukayama

2025 年 4 月 23 日

目次

1	Xdebug を Windows にインストールする	1
1.1	準備作業	1
1.2	xdebug.org での作業	1
1.3	ローカル PC での作業	1
2	VSCode で Xdebug を使う (Windows)	2
2.1	準備作業	2
2.2	拡張機能 "PHP Debug" をインストールする。	2
2.3	launch.json の設定	3
2.4	動作させてみる	4

1 Xdebug を Windows にインストールする

1.1 準備作業

1.1.1 phpinfo() を出力する

"C:¥xampp¥htdocs" フォルダの中に、"phpinfo.php" を作成し、以下の内容とする。

phpinfo.php

```
1 <?php
2 phpinfo();
```

このファイルをブラウザで開く。

<http://localhost/phpinfo.php>

1.2 xdebug.org での作業

1.2.1 ブラウザの phpinfo() の内容をすべてコピーして、メモリに記憶させる。

1.2.2 <https://xdebug.org/wizard> にアクセスし、四角の領域に貼り付ける。

"Analyze my phpinfo() output" ボタンをクリックすると、ページが開くので、「Instructions」の内容にしたがって、作業をすることになる。

1.3 ローカル PC での作業

1.3.1 php_xdebug-3.4.2-8.2-ts-vs16-x86_64.dll をダウンロードする。

Chrome の場合は、「未確認のダウンロードをブロックしました」というメッセージが表示される。

その場合は、右の ">" をクリックする。

開いたダイアログで「未確認のファイルをダウンロードする」をクリックする。これでダウンロードできる。

1.3.2 ダウンロードした上記のファイルを C:¥xampp¥php¥ext¥に移動して、ファイル名を "php_xdebug.dll" とする。

1.3.3 C:¥xampp¥php¥php.ini を開き、末尾に以下のように記述する。

```
1 [Xdebug]
2 zend_extension = xdebug
3 xdebug.mode = develop,debug
4 xdebug.start_with_request = yes
```

develop — var_dump() の出力がていねいになる。
debug — コードを 1 行ずつ実行できるようになる。

1.3.4 xampp コントロールパネルで Apache を再起動する。

Apache のボタンを"STOP" し、そのあと"START" する。

2 VSCode で Xdebug を使う (Windows)

2.1 準備作業

動作を確認するためのコードを以下のように作成。

```
./sampleApp
  index.php
  xdebuginfo.php
```

sampleApp フォルダで VSCode を開き、以下のコードを書く。

index.php

```
1 <?php
2 $name = '佐藤 明';
3 $age = 28;
4 $sex = '男';
5 ?>
6 <!DOCTYPE html>
7 <html lang="en">
8 <head>
9   <meta charset="UTF-8">
10  <meta name="viewport" content="width=device-width, initial-scale=1.0">
11  <title>Document</title>
12 </head>
13 <body>
14   <h1>人物紹介</h1>
15   <dl>
16     <dt>名前</dt>
17     <dd><?= $name ?></dd>
18     <dt>年齢</dt>
19     <dd><?= $age ?></dd>
20     <dt>性別</dt>
21     <dd><?= $sex ?></dd>
22   </dl>
23 </body>
24 </html>
```

xdebuginfo.php

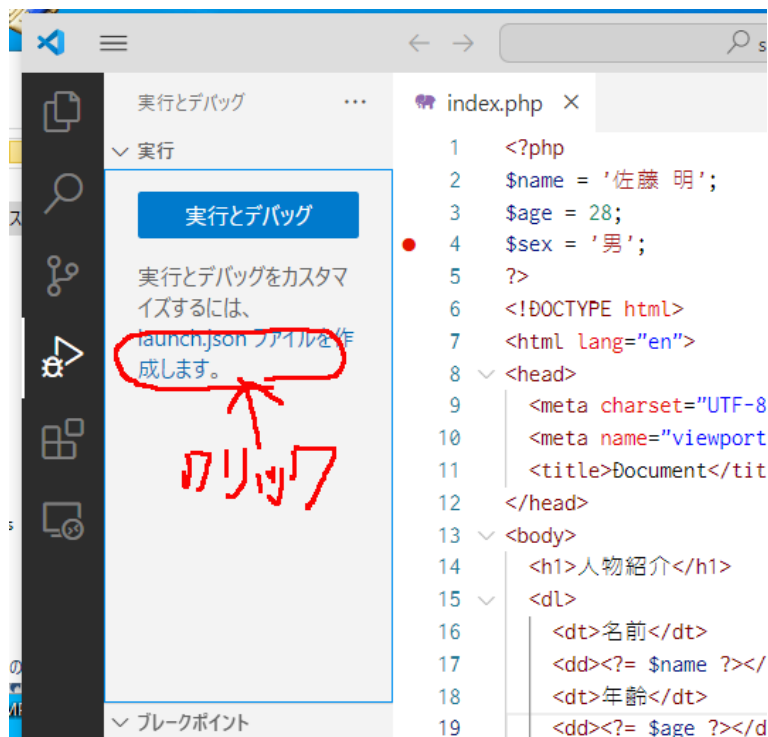
```
1 <?php
2 xdebug_info();
```

2.2 拡張機能 "PHP Debug" をインストールする。

```
PHP Debug
Xdebug xdebug.org
Debug support for PHP with Xdebug
```

VSCode 左端の "実行とデバッグ" を選択すると、VSCode 左に "実行とデバッグ" ボタンが表示されるの

で、その下の "launch.json" の作成を選択する。



2.3 launch.json の設定

以下の内容の launch.json が自動で作成される。

特に、"Listen for Xdebug" の項目が大事。

.vscode/launch.json

```
1 {
2   // IntelliSense を使用して利用可能な属性を学べます。
3   // 既存の属性の説明をホバーして表示します。
4   // 詳細情報は次を確認してください: https://go.microsoft.com/fwlink/?linkid=830387
5   "version": "0.2.0",
6   "configurations": [
7
8     {
9       "name": "Listen for Xdebug",
10      "type": "php",
11      "request": "launch",
12      "port": 9003
13    },
14    {
15      "name": "Launch currently open script",
16      "type": "php",
17      "request": "launch",
18      "program": "${file}",
19      "cwd": "${fileDirname}",
20      "port": 0,
```

```

21     "runtimeArgs": [
22         "-dxdebug.start_with_request=yes"
23     ],
24     "env": {
25         "XDEBUG_MODE": "debug,develop",
26         "XDEBUG_CONFIG": "client_port=${port}"
27     }
28 },
29 {
30     "name": "Launch Built-in web server",
31     "type": "php",
32     "request": "launch",
33     "runtimeArgs": [
34         "-dxdebug.mode=debug",
35         "-dxdebug.start_with_request=yes",
36         "-S",
37         "localhost:0"
38     ],
39     "program": "",
40     "cwd": "${workspaceRoot}",
41     "port": 9003,
42     "serverReadyAction": {
43         "pattern": "Development Server \\(http://localhost:([0-9]+)\\) started",
44         "uriFormat": "http://localhost:%s",
45         "action": "openExternally"
46     }
47 }
48 ]
49 }

```

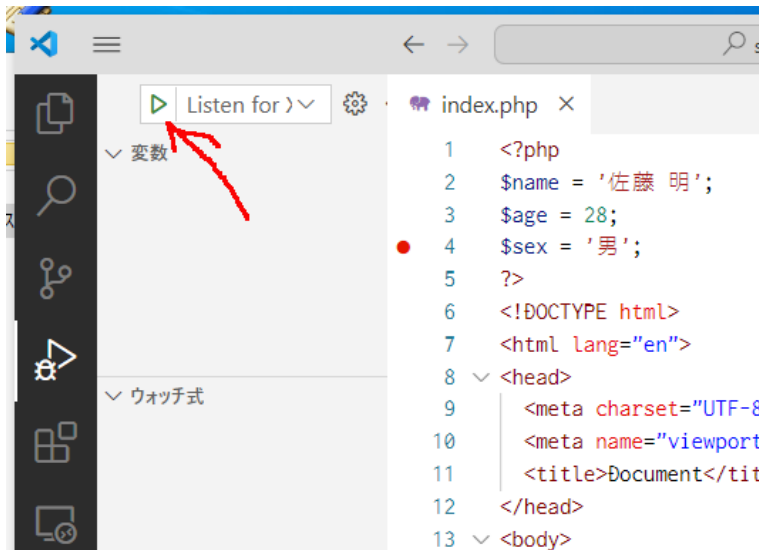
2.4 動作させてみる

2.4.1 ブレークポイントの設定

index.php を選択し、コードのどこかにブレークポイントを設定する。
行番号の左をクリックすれば OK。

2.4.2 デバッグ開始

“実行とデバッグ” を選択し、“デバッグの開始” をクリックする。



2.4.3 PHP server を起動、あるいはリロード

php サーバーを動作させると、自動でブラウザが開くが、何も表示されない。ブラウザは動作し続けている状態となる。

php サーバー以外のサーバーを使っている場合

もし、php の簡易サーバー機能を使って、“sampleApp” フォルダの中で“php -S localhost:8888”とされているならば、ブラウザで“http://localhost:8888/”にアクセスするとよい。

もし、sampleApp フォルダを “C:\xampp\htdocs\” においているならば、ブラウザで “http://localhost/sampleApp/” にアクセスするとよい。

いずれの場合も何も表示されずに、ブラウザが動作している状態となる。

2.4.4 VSCode に戻ると、ブレークポイントで実行が止まっている。

あとは、step over で 1 行ずつ実行すればよい。

