



1. Laravelとは

Laravelとは

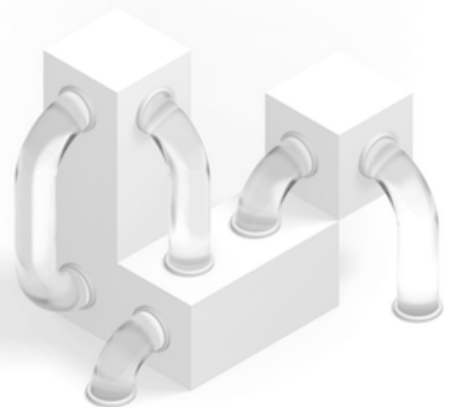
Laravel は **Syhmphony** をベースにしたPHPフレームワーク 1つです。後発なため使いやすさや機能が豊富で、世界的にもシェアの高いフレームワークです。

**Laravel**[Documentation](#)[Forge](#)[Ecosystem ▾](#)[News](#)[Partners](#)[🔍 Search Docs \(Press '/'\)](#)

Vapor can scale your applications! Sign up today! →

The PHP Framework for Web Artisans

Laravel is a web application framework with expressive, elegant syntax. We've already laid the foundation — freeing you to create without sweating the small things.

[Documentation](#)[Watch Laracasts](#)

公式サイト

- [Laravel（公式）](#)
- [Laravel（日本）](#)

Laravel の基本機能

Laravel は、[MVC（コントローラー・ビュー・モデル）](#)、[ルーティング](#)、[ORM（オーアールマッパー）](#)などのWeb開発に必要な基本的な機能を搭載しています。また、[PHP](#) ベースのコマンド [artisan](#) で開発の効率化をしたり、大規模開発で利用されるデザインパターンにも考慮した設計になっています。

PHPインストール（初学者向け）

PHP のインストールは色々な方法があります。初学者でわからない人は [XAMPP](#) や [MAMP](#) でインストールしてみてください。

[XAMPP・MAMPで開発環境構築](#) [🔗](#)

XAMPP（Win初学者向け）

Windowsの初心者は、Web開発環境ツール [XAMPP](#) を利用するのが便利です。

MAMP（Mac初学者向け）

Macの初心者は、Web開発環境ツール [MAMP](#) を利用するのが便利です。

PHPインストール（Homebrew）

Mac では [Homebrew](#) を使うと [PHP](#) をインストールできます。[Homebrew はあらかじめインストールしておく必要があります。](#)

[Homebrew（公式）](#) [🔗](#)

php モジュール検索

[brew コマンドで php を検索](#)すると、[PHP](#) に関するモジュール一覧が確認できます。

```
% brew search php
==> Formulae
brew-php-switcher  php-cs-fixer  php@7.3  phpbrew  phpmyadmin
php                php-cs-fixer@2  php@7.4  phplint  phpstan
```

```
php-code-sniffer  php@7.2  php@8.0  phpmd  phpunit

==> Casks
eclipse-php
```

php モジュールインストール

モジュールの中から **PHP8.0** をインストールします。

```
% brew install php@8.0
```

PHPパスの設定

zsh の場合

ターミナルで `~/.zshrc` に、**PHP** のパスを追記して読み込みます。

```
% echo 'export PATH="/opt/homebrew/opt/php@8.0/bin:$PATH"' >> ~/.zshrc
% echo 'export PATH="/opt/homebrew/opt/php@8.0/sbin:$PATH"' >> ~/.zshrc
% source ~/.zshrc
```

bash の場合

ターミナルで `~/.bashrc` に **PHP** のパスを追記して読み込みます。

```
% echo 'export PATH="/opt/homebrew/opt/php@8.0/bin:$PATH"' >> ~/.bashrc
% echo 'export PATH="/opt/homebrew/opt/php@8.0/sbin:$PATH"' >> ~/.bashrc
% source ~/.bashrc
```

PHP コマンドの確認

PHP のバージョンを確認します。

```
% php -v
PHP 8.1.1 (cli) (built: Dec 17 2021 23:49:52) (NTS)
```

PHPインストール（仮想PC）

本運用のサーバは Linux で稼働させるのが一般的で、現場では Docker、VirtualBox、WSL といった仮想PCをローカル環境で構築するのがの主流です。ただし、仮想PCや Linuxの知識が必要になるので、初学者にはハードルは高くなります。

Linux のコマンド例

例えば、Linux (Ubunt) に PHP と関連モジュールを以下のコマンドでインストールします。

```
% sudo apt install php libapache2-mod-php php-mysql
```

その他、WebサーバやDBなども個別にインストールする必要があり、Docker に関してはコンテナの設定も必要になります。

Composer とは

Composer は、サードパーティ (vendor) のプログラムを使ってPHP開発するためのモジュール管理ツールです。Laravel の開発では PHP と Composer をローカル環境にインストールする必要があります。

[Composer \(公式\)](#)



A Dependency Manager for PHP

Latest: **2.0.7** ([changelog](#))

Composer 2.0 is now available! [Read our announcement!](#)

Composerインストール (Win)

Windows では Composer のインストーラをダウンロードできます。

[Composer: Installation - Windows](#)

Installation - Windows

Using the Installer

This is the easiest way to get Composer set up on your machine.

Download and run [Composer-Setup.exe](#). It will install Composer. You can then call `composer` from any directory in your command prompt.

Note: Close your current terminal. Test usage with `composer` loaded when the terminal starts.

Composer のバージョン確認

PowerShell などでも Composer のバージョンを確認してみましょう。

```
% composer --version
Composer version 2.1.12
```

Composerインストール (Mac)

Homebrew でインストールする場合

Macの場合 Homebrew でのインストールが簡単です。ターミナルで Composer をインストールします。

```
% brew install composer
```

PHPコマンドでインストールする場合

MacやLinuxでは、ターミナルを利用してインストールもできます。

```
% php -r "copy('https://getcomposer.org/installer', 'composer-setup.php');"
% php -r "if (hash_file('sha384', 'composer-setup.php') === 'baf1608c33254d00611ac1705c1d9958c817a1a33bce370c05
% php composer-setup.php
% php -r "unlink('composer-setup.php');"
% sudo mv composer.phar /usr/local/bin/composer
```

Composer のバージョン確認

Composer のバージョンを確認してみましょう。

```
% composer --version
Composer version 2.1.12
```

composer の動作確認ができました。これで Laravel プロジェクトを作成する準備ができました。

Composer のパスが設定されない場合

Homebrew で Composer のパスが設定されない場合、composer リンクを設定します。

```
% brew link composer
```

サードパーティのパス設定 (Mac)

Mac や Linux では composer のサードパーティのパスを利用するのに環境変数を設定しておきます。

zsh の場合

ターミナルで .zshrc に vendor パスを追記します。

```
% echo 'export PATH=$HOME/.composer/vendor/bin:$PATH' >> ~/.zshrc
% source ~/.zshrc
```

bash の場合

ターミナルで .bashrc に vendor パスを追記します。

```
% echo 'export PATH=$HOME/.composer/vendor/bin:$PATH' >> ~/.bashrc
% source ~/.bashrc
```

当サイトの教材をはじめとするコンテンツ（テキスト、画像等）の無断転載・無断使用を固く禁じます。これらのコンテンツについて権利者の許可なく複製、転用等する事は法律で禁止されています。尚、当ウェブサイトの内容をWeb、雑誌、書籍等へ転載、掲載する場合は「ロジコヤ」までご連絡ください。