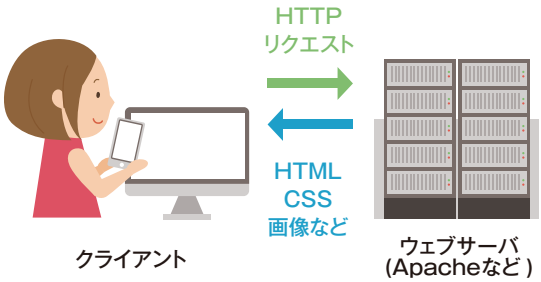
1-2 WordPressサイトを構成する要素

WordPressによってつくられるウェブサイトはどのような要素から構成されるのでしょうか。
WordPressによるウェブページ生成の仕組みを知るとともに、
WordPressサイトを設計するために必要な「コンテンツ管理」の考え方について説明します。

WordPressによるウェブページの生成

通常のウェブページを表示する仕組み

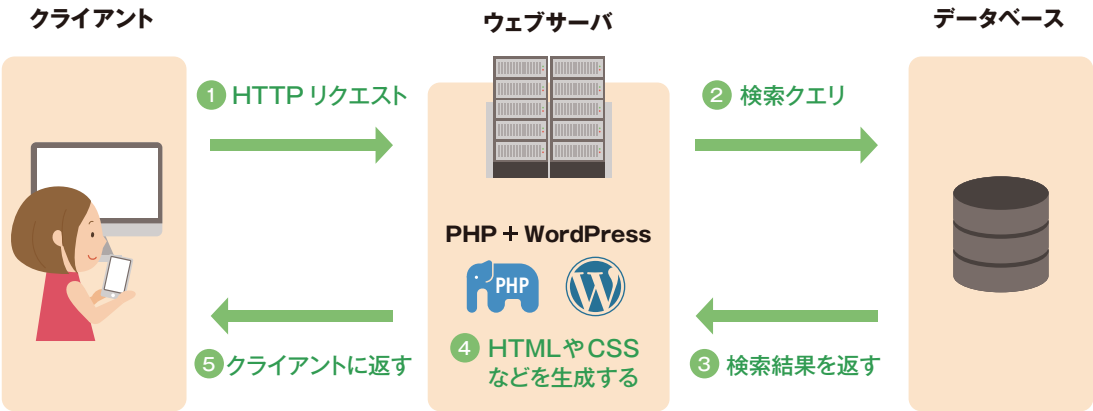
まずはブラウザが通常のウェブページを表示する仕組みを見ていきましょう。
登場人物はふたり、ウェブページのデータを提供するウェブサーバと、これを受けて表示するクライアント（この場合はウェブサイトを閲覧する端末のこと）です。
ウェブサイトを閲覧するとき、私たちはクライアントにインストールされたウェブブラウザ（ChromeやFirefox・Safari・Internet Explorer・Edgeなど）を通じてウェブサーバにアクセスをおこない、リクエストを送ります。
URLをブラウザに入力し、ウェブページ取得のリクエスト（HTTPリクエスト）を送信すると、ウェブサーバはこれを受けてHTMLや画像といったウェブページを構成する要素ファイルをクライアントに送り、それらがウェブブラウザによって解釈されてウェブページとして表示される、という流れです。



通常のウェブページの仕組み
クライアントがHTTPリクエストを送ると、ウェブサーバがHTMLや画像といったウェブページを構成する要素ファイルを送ります。

WordPressによって生成されるページを表示する仕組み

それに対して、WordPressの場合はどうでしょう。
登場人物は少し増えます。ウェブサーバの中にいる「WordPress（PHPファイル群）」と「データベース」です。
WordPressはPHPというプログラムによって動作するソフトウェアです。私たちの端末（クライアント）からのリクエストがWordPressがインストールされたウェブサーバに到達すると、まずサーバ内でWordPressによる処理が実行され、その実行結果としてHTMLが出力され、最後に私たちの端末（クライアント）に返されるという流れです。
またWordPressはPHPのほかに、データベースという仕組み（WordPressではおもにMySQLというデータベース管理システムを用います）を使ってページの内容や各種設定を保存しています。WordPressはクライアントからのリクエストに応じて、各種の情報をデータベースから引っ張り出し、出力されるHTML等に反映します。
つまり、WordPressはクライアントからのリクエスト（クエリ）ごとに、ウェブサーバの中でデータベースと連携してさまざまな処理をおこなうことでウェブページをその都度生成し、これをクライアントに返す、という仕事をしている、ということになります。よってWordPressが動作するには、ウェブサーバにPHPの実行環境とデータベースがインストールされており、さらにデータベースが利用できる状態になっていることが前提条件となります。



WordPressによるウェブページの仕組み
WordPressはクライアントからのリクエストごとに、ウェブサーバの中でデータベースと連携してさまざまな処理をおこなうことでウェブページをその都度生成し、これをクライアントに返す仕事をしています。

こうしたページの表示の仕組みを「ウェブページの動的生成」と呼びます。また「動的に生成されるページの集合」によって構成されるサイトを「動的なウェブサイト」と呼びます。これに対して、HTMLのみで構成されたサイトのように、あらかじめ用意されたウェブページをそのまま表示するものを「静的なウェブサイト」と呼ぶことがあります。
それでは、それぞれの要素についてもう少し詳しく見てみましょう。

CHECK!

PHPとは

PHPは「PHP: Hypertext Preprocessor（ピーエイチピー ハイパーテキスト プリプロセッサー）」の略で、サーバ側でHTMLを動的生成するための機能を多く備えた、汎用的なオープンソースプログラム言語です。

WordPressサイトを構成する要素

1.PHP

WordPressの本体はおもにPHPファイルの集合体ですが、このPHPファイル群（つまりWordPress）が動作するためにはウェブサーバにPHPの実行環境（以下単にPHPと呼びます）がインストールされていることが必要です。レンタルサーバを用いる場合は、そのサーバにPHPがインストールされているかをまずチェックしましょう。またWordPress5.5時点で、サーバにインストールされているPHPのバージョンは7.4以上が推奨されていますので、こちらもご注意ください。
またレンタルサーバによっては、PHPの動作を設定するファイル（php.ini）が編集できない場合があります。PHPの設定によるトラブル解決のために同設定をさわる必要が生じる場合がありますので、こちらにも注意したほうがいいでしょう。

2.データベース管理システム（MySQL/MariaDB）

WordPressサイトに必要な設定情報やコンテンツが格納されるデータベースを構成・管理するのがデータベース管理システムです。WordPressではMySQLまたはMariaDBというシステムが動作に必須です。レンタルサーバを利用するのであれば、PHPと並んでこのMySQL・またはMariaDBが利用できるウェブサーバであるか、またサーバが提供するDBシステムは推奨されているバージョンであるか、ということに注意してください。なお、どちらのデータベースを使用するかはおもにレンタルサーバの仕様によります。

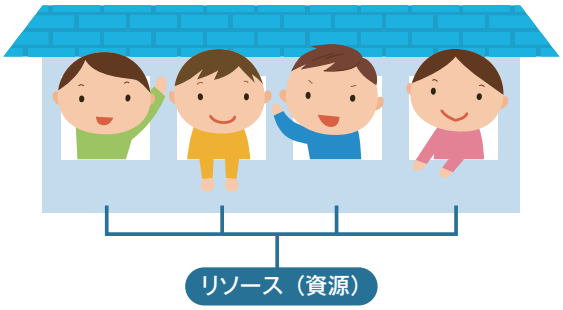
レンタルサーバの形態

一般的なレンタルサーバ会社で借りることができるサーバは、大きく分けて共有サーバと専用サーバがあります。つくりたいサイトの目的や将来のことを考えて選んでください。

共有サーバ

1つのサーバ機に複数の契約ユーザーが割り当てられ共有する形態です。家でたとえるなら集合住宅・マンションのイメージです。

- **メリット：安価に運用することができる。**
- **デメリット：同じサーバ機を共有する別ユーザーのトラフィック・負荷の増加により自分のサイトのパフォーマンスが落ちることがある。**



共有サーバはマンション

専用サーバ

1つのサーバ機に1契約ユーザーが占有する形態です。家でたとえるなら一戸建てのイメージです。

- **メリット：契約しているサーバの性能を占有することができるため、別ユーザーのトラフィック・負荷の増加による影響はないため、安定したパフォーマンスが期待できる。**
- **デメリット：導入・運用コストが高い。**

専用サーバは一戸建て

サポート体制とメール

サーバ会社やプランにより管理画面や設定方法は異なるため、電話の対応時間やメール対応の早さなどのサポート体制もサーバ選びの重要なポイントです。また、周りに利用者・情報の多いサーバを選ぶことで、情報を共有できるメリットもあります。
一般的なレンタルサーバではメールサーバも同時に利用できます。サーバ契約時に必要なメールアカウント利用数が足りているか、保存できるメールデータの容量が十分かも合わせて確認しておくとよいでしょう。

CHECK!

静的なサイトからの変更・リニューアル時の注意点

すでになんらかのサーバを借りている場合、同じ性能のウェブサーバで、静的な要素（HTML、CSS、JavaScript）で構成されたページと、WordPressなどPHPにより動的に生成されるページは、数十倍から数百倍のパフォーマンスの差があるため、静的なページで速度的に問題なく表示できている場合でも、WordPressを導入することにより大幅に遅くなってしまう場合があります（仕組みの概要は**2-2**で紹介）。そういった場合、サーバ選びの基準はアクセス数など状況・仕様の把握が必要になり値段だけで決めることは難しいため、サーバ会社やWordPressに詳しい人からアドバイスをもらうのがよいでしょう。

2-2 レンタルサーバ選びのポイント

WordPressを利用するために必要なレンタルサーバの環境について具体的に説明します。
また、ウェブサイトの住所といえるドメインを取得して利用する際の
注意についても知っておきましょう。

LAMP環境を確認する

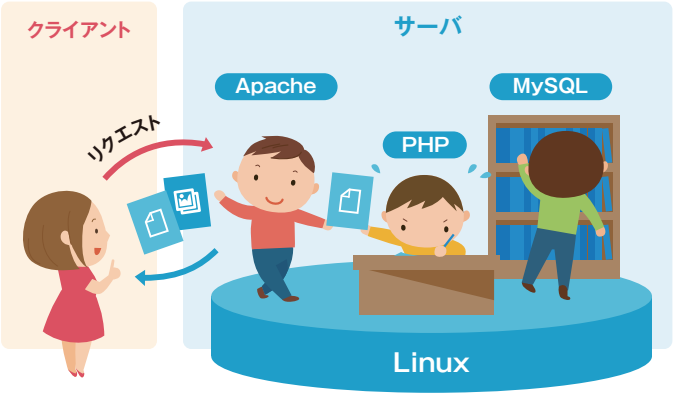
WordPressで必要とされる環境は、OS環境、アプリケーションの頭文字をとって
LAMP環境と呼ばれることがあります。レンタルサーバの選択にあたって、注意が必要な点を説明していきます。

L
Linux（OS）

A
Apache HTTP Server（ウェブサーバ）

M
MySQL（データベース）

P
PHP（プログラム言語）



LAMPがそろってWordPressが使えます。

ウェブサーバについて

代表的なウェブサーバは2つあります。共にオープンソースで開発がおこなわれているApache（アパッチ、正式にはApache HTTP Server）とNGINX（エンジンエックス）です。一般的なレンタルサーバではApacheが用いられていることが多く、WordPressのプラグインの多くもウェブサーバとしてApacheが想定されています。大規模・高速な処理が必要とされるサーバでは、最近はNGINXが用いられることも増えてきています。
ウェブサーバに関する機能は、使用するレンタルサーバにより使用の可否・設定できる項目が異なります。代表的なものに、「.htaccess」と呼ばれるファイルに設定を記述することで、アクセス制限をおこなう、特定のIPアドレスからのアクセス時にしか表示を許可しないようにする、「xxx.com」へのアクセスをwww付きの「www.xxx.com」に合わせる、などがあります。これらはサーバ会社から提供されている自分の契約配下の設定ファイルを変更することでおこないます。これらの仕様が必須である場合には事前にサーバ会社に確認し、無料期間があればテストをおこないましょう。
ウェブサーバのシェアではNGINXは現在Apacheに次いで2番目で急速に成長していますが、一般的なレンタル共有サーバでは用いられることは多くありません。設定方法がApacheとは異なることや、上記の「.htaccess」は使用できないため注意が必要です。

タグの設定

タグはカテゴリーとは異なり、親子関係の設定や記事URLに含めることなどができません。また、後述のウィジェット機能で「タグクラウド」（多くの記事についたタグ名が数多く並ぶ、タグから記事を探すためのインターフェイス）を表示させることができます。タグは記事を探してもらうためのキーワードとして、記事の内容に即したキーワードを2～3個設定するという利用法が一般的です。

記事にタグを設定する

タグの作成は、投稿の編集ページから直接作成・設定できます。編集画面右下に［タグ］というボックスがありますので、そこに設定したいタグを入力して Enter を押しましょう。ここでは記事内容に沿って「トランペット」「シンガー」と2つタグを設定してみてください。これまでと同じく記事を［更新］して設定完了です。

アイキャッチ画像の設定

ブログの記事一覧に、その記事を代表する画像がタイトルなどと合わせて表示されるパターンのデザインを数多く見かけます。こうした画像をWordPressでは「アイキャッチ画像」と呼びます。投稿の編集画面から記事にアイキャッチ画像を追加してみましょう。

記事にアイキャッチ画像を設定する

1 たんに本文に画像を追加しただけではアイキャッチ画像は表示されません。投稿の編集画面サイドバーの［アイキャッチ画像］ボックスから設定することができます。［アイキャッチ画像を設定］を押すと、画像の選択とアップロードができる画面が開きます。

2 ［メディアライブラリ］から先ほどアップロードしたルイ・アームストロングの画像を選択し❶、［アイキャッチ画像を設定］を押すと設定完了です❷。投稿の編集画面の［アイキャッチ画像］ボックスには設定した画像が表示されます。

3 記事を［更新］して表示すると、アイキャッチ画像が設定されています。なお、アイキャッチ画像の表示位置はテーマによって異なります。

4-3 ブロックエディターを使いこなそう

WordPressにはバージョン5.0.0からGutenberg（グーテンベルグ）とよばれる新しいブロックエディターが導入されました。
この「ブロックエディター」の基本的な使い方について学習しましょう。

ブロックエディターとは

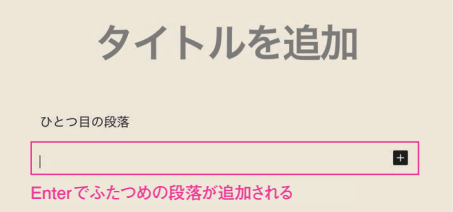
WordPress5.0.0より前のバージョンでは、1つのテキストエリアにコンテンツを入力するスタイルでした。しかしWordPress5.0.0からは、画像や段落を1つのブロックとして扱い、そのブロックを組み合わせたり並び替えたりしてコンテンツを作成するスタイルに切り替わりました。これによって、サイトでの表示に近い形で直感的にコンテンツを作成できるようになっています。

WordPress5.0.0より前のバージョンのエディタ。

ブロックの追加方法

新しく記事を作成したとき、初期設定では段落ブロックが1つ表示された状態です。

1 ［文章を入力、または / でブロックを選択］と表示された部分をクリックすると、［段落ブロック］として文章を入力することができます。この状態でEnterキーを押すと、下に新しく［段落ブロック］が追加されます。



2 ブロックとブロックの間にある［+］のアイコンをクリックすることで、ブロックの間に新しくブロックを追加することもできます。

［段落ブロック］以外の追加方法

Enterキーで追加したブロックはすべて［段落ブロック］として追加されます。段落以外のブロックを追加したい場合は、追加したい位置にある［+］のアイコンをクリックしましょう。すると追加できるブロックの候補が表示されます。
また、文章が入力されていない［段落ブロック］をクリックして、「/」（スラッシュ）を1文字目に入力すると、よく使われるブロックの候補が表示されます。ここで表示されたブロックをクリックすることで、その段落ブロックを指定した

7-1 ローカル開発環境の必要性

ローカル開発環境を構築する前に、まずはそもそもローカル開発環境とは何か、なぜ必要なのかについて見ていきましょう。さらに、本書で実際に利用することになるローカル開発環境構築ツール「MAMP」（マンプ）についても少し触れておきます。

ローカル開発環境とは

ローカル開発環境とは、本番サーバにサイトを公開する前に、手元のWindowsやMacなどの端末でサイトの表示や動作の確認をおこなうための環境です。ローカルという言葉のとおり、基本的にインターネット上に公開するものではありませんが、必要に応じて一時的にアクセス可能にすることはあります。

なぜローカルなのか

開発途中のサイトをインターネットに公開しないためには、本番環境とは別の開発環境が必要になります。ローカル開発環境以外には、リモートサーバに開発環境を構築する方法やクラウドサービスを使う方法などがあります。その中で、ローカルで開発環境をつくることにはいくつかのメリットがあります。

1.アップロードの時間が省略できる

リモート開発環境の場合、変更のたびにファイルをリモートサーバにアップロードする必要があります。しかし、ローカル開発環境では、ファイルを保存するだけで変更を確認することができます。これは開発のスピードアップに大いに貢献するでしょう。最近では多くのエディタが、ファイルを保存するだけで自動的に指定のサーバへファイルをアップロードする機能を備えていますが、ネットワークへの通信を考慮すれば、やはりローカル環境にはかなわないでしょう。

2.無料でできて制限がない

開発環境を構築できるクラウドサービスにおいては、ブラウザから少ない操作で簡単に開発環境を構築できるメリットはありますが、有料であったり、無料プランでは構築できるサイトの数に制限があるなどのデメリットもあります。

3.オフラインでも開発可能

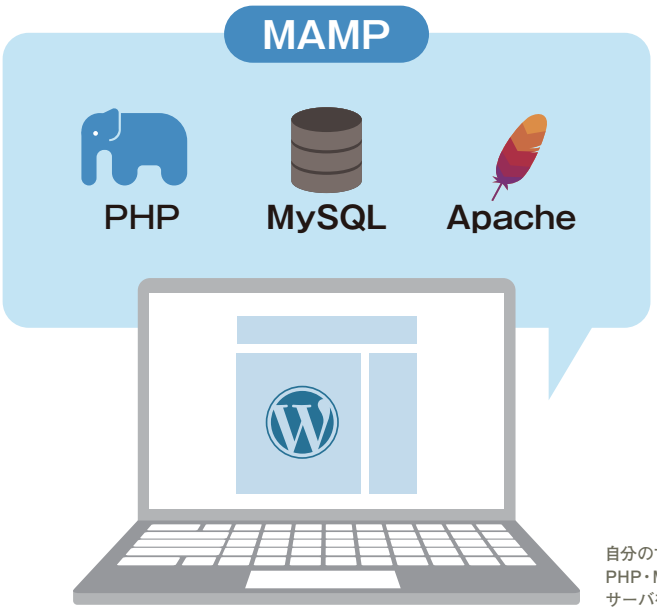
リモートやクラウドと違い、ローカル開発環境では、インターネットへアクセスできない環境でも、サイトの表示や基本的な開発が可能です。

さまざまなローカル開発環境構築ツール

先に説明したように、WordPressサイトをブラウザで表示可能にするためにはウェブサーバやMySQLなど、さまざまなツールが必要です。もちろんそれらを手動でひとつひとつ用意することも可能ですが、かなりの専門知識が必要です。今日では、その煩雑な構築作業を手助けしてくれるツールが数多く存在します。MAMP・XAMPP・Local by Flywheel・VVV・VCCW・Wockerなど、例をあげるときりがありませんが、本書ではWindowsとMacの両方で利用でき、かつ初心者でも扱いやすいMAMPを用いることにします。

MAMPとは

MAMPとは、ローカル上にウェブサーバやMySQLサーバなどを構築してくれるアプリケーションです。そのネーミングは元々Macintosh・Apache・MySQL・PHPの頭文字に由来しますが、いまではMac以外にWindows版も用意されており、さらに、ウェブサーバはApacheのほかにNGINXが、プログラミング言語はPHP以外にPythonやPerlも用意されています。



自分のマシンの中にApache・PHP・MySQLが入った仮想サーバを用意できます。

初心者に使いやすいMAMP

MAMPはWordPressを動かすのに最低限必要なツールをわかりやすいセットアップ画面経由でまとめて用意してくれます。もちろん、ほかの方法やツールで同様な環境を構築することも可能ですが、WindowsかMacのどちらかでしか利用できなかったり、それぞれのツールを個別でインストールする必要があったり、初心者にはとっつきにくい、いわゆる「黒い画面」と呼ばれているCUI（キャラクターユーザーインターフェイス）あるいはCLI（コマンドラインインターフェイス）を使う必要が生じるなどの懸念があります。WindowsとMacの両方で使えることとGUI（グラフィカルユーザーインターフェイス）が用意されており初心者に扱いやすい点から、本書ではMAMPを採用しました。また、MAMPと同様なツールとして、XAMPP（ザンプ）というものもあります。こちらもWindowsとMacのどちらでも利用できますが、本書ではインターフェイスのわかりやすさからMAMPを選択しました。

基本的にMAMPは無料で利用できますが、有料版のMAMP PROもあります。MAMP PROでは、簡単にバーチャルホストの設定ができ、複数サイトの管理などに便利な機能が加えられています。興味がある人は、まずは14日間の試用期間で試してみるといいでしょう。なお、本書で扱う内容としては無料版で十分です。

COLUMN

バーチャルホストとは

ひとつのサーバで複数のドメインを運用する技術のことです。MAMPによって構成されたローカルサーバにさまざまなドメインを割り当てることで、複数のサイトをローカル環境で運用することが可能になります。

STEP 03　静的サイトを用意しよう

easiest-wp-html

オリジナルテーマを作成する際、いきなりPHPを使ってコーディングしていくにはテーマ作成にかなり慣れている必要があります。本書では初心者にもわかりやすいように、まずはHTML・CSS・JavaScriptなどでコーディングされた静的サイトをサンプルとして用意し、それをテーマ化していく方法を採用します。

静的サイトのデータはダウンロードしたサンプルデータの中に含まれています。ご自身のテーマ化したいサイトを用いてもかまいませんが、WordPress テーマにはいくつかのルールがあるので、はじめての場合はサンプルデータを利用した方が混乱が少ないでしょう。

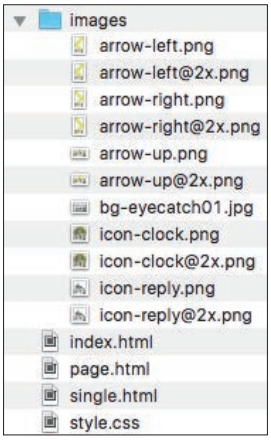
サイトイメージの確認

まず、テーマ化する前に、どういうサイトになるかイメージしやすいよう、完成時のものを確認してみましょう。サンプルサイトはフォトグラファーのポートフォリオサイトをイメージしています。ヘッダーにサイトタイトルとグローバルナビゲーションがあり、メインは2カラムで右側にサイドバー、フッターにはコピーライトが入ります。ヘッダー、フッター、サイドバーは基本的にすべてのページで共通とします。

ファイル構成

サイトのイメージを確認できたら、基になる静的サイトのデータが入っている「easiest-wp-html」フォルダを開いてファイル構成を見てみましょう。画像は「images」フォルダにまとめられており、投稿ページ、アーカイブページなどを想定したHTMLファイルとスタイルシートの「style.css」が入っています。

HTMLとCSSのファイル名は、混乱を避けるため、WordPressテーマのテンプレート名に合わせたものになっています。今後、自分でオリジナルな静的サイトを用意してそこからテーマを作成する際も参考にするとよいでしょう。



静的サイトのファイル構成

サンプルサイト完成イメージ

本書テーマのサンプルサイト　CHECK!

サンプルテーマの完成イメージとして、ダウンロードファイルとは別にサンプルサイトを用意しています。Lesson 15まで終えた状態のオリジナルテーマが公開されていますので、参考にしてください。
https://easiest-wp.com

8-2 PHPの基礎知識

PHPを扱うにあたっての基礎知識を学習しましょう。
プログラミングにはじめてふれる人は難しく感じることと思いますが、
はじめからすべてを理解する必要はありません。
あとのレッスンで困ったときに、ここに戻ってきてください。

PHPの概要

PHPは、広く使われているオープンソースの汎用プログラミング言語です。HTMLに埋め込む形で記述することができ、とくにウェブ開発に適しています。PHPという名称は、“PHP: Hypertext Preprocessor”の頭文字で、「ハイパーテキスト（HTML）のプリプロセッサ」という意味です。つまり、PHPファイルは読み込まれたときにHTMLを動的に生成するものであると考えてください。

基本構文

PHPタグ

PHPのコードをHTMLに埋め込むことができるとはどういう感じなのでしょうか。試しに「hello.php」という名前のファイルを作成して次のようにコードを記載し、MAMPの「htdocs」フォルダ直下に置いてください。

PHPはファイルを解析して、PHPの開始タグ`<?php`と終了タグ`?>`を探します。タグが見つかると、PHPはコードの実行を開始したり終了したりします。

ブラウザでhttp://localhost:8888/hello.phpにアクセスすると、「Hello, world!」が表示されているのが確認できます。上記ファイルのPHPタグに囲まれた部分では、`echo`命令により`<p>Hello, world!</p>`が出力され、それ以外の部分は通常のHTMLとして解釈されます。ブラウザからソースを確認すると、下の内容のHTMLがPHPによって出力されたことが確認できます。すべての確認が終わったら、hello.phpは削除してかまいません。

ソースコード hello.php

```
<html>
<head>
  <title>PHP Test</title>
</head>
<body>
  <?php echo '<p>Hello, world!</p>'; ?>
</body>
</html>
```

ソースコード hello.phpで出力されるHTML

```
<html>
<head>
  <title>PHP Test</title>
</head>
<body>
  <p>Hello, world!</p>
</body>
</html>
```

COLUMN

ブラウザでソースを確認するには

ブラウザにより異なりますが、ページを右クリックして表示される［（ページの）ソースを表示］をクリックするか、Macの場合はcommand＋option＋U、Windowsの場合はCtrl＋UあるいはF12（開発者ツール）で表示されます。

CHECK!

ファイル終端におけるPHP終了タグ

ファイル終端におけるPHP終了タグはオプション（任意）ですが、省略することが推奨されています。終了タグのあとに余分な空白や改行があると、予期せぬ挙動を引き起こす場合があるので、これを防ぐためです。本書ではファイル終端のPHP終了タグをすべて省略します。

9-2 記事の一覧を表示させよう

記事の一覧を表示するにはWordPressのループを利用します。
実際にオリジナルテーマを書き換える前に、
まずはループとはどういう仕組みなのかについて理解しましょう。

ループを理解しよう

ループとは、テンプレートファイル（**10-1**参照）内で投稿内容を表示するためにWordPressが用意している仕組みです。記事の一覧ページや記事の詳細ページなど、記事内容を出力するページでは必ず使用されます。ループ内では指定された条件に従って管理画面から投稿された投稿内容を出力することができ、ループ内に書き込んだHTMLやPHPなどのコードは、繰り返し処理されます。ループの繰り返す回数は、表示設定ページの「1ページに表示する最大投稿数」やテンプレートファイルの種類などによって変わります。まずはもっともシンプルなループの例を見てみましょう。

ソースコード

```
<?php if ( have_posts() ) : ?>// ❶
  ここは投稿がある場合、ループ前に一度だけ処理されます。
  <?php while ( have_posts() ) : ?>// ❷
    <?php the_post(); ?>// ❸
    ここは繰り返し処理されます。
  <?php endwhile; ?>// ❹
  ここは投稿がある場合、ループ後に一度のみ処理されます。
<?php else : ?> // ❺
  ここは投稿がない場合、一度のみ処理されます。
<?php endif; ?> // ❻
```

❶if (have_posts()) :

最初の行はWordPressの`have_posts()`関数を使った基本的なPHPの`if`文です。`have_posts()`関数は真偽値を返す関数であり、要求にマッチする投稿がある場合は`true`（真）を、投稿がない場合は`false`（偽）を返します。したがって、もし投稿がある場合は`else`までのコードが実行されますが、投稿がない場合は`else`から`endif`までのコードが実行されます。次の行は`if`文の「投稿がある場合」の中かつループの前です。ループの開始前に一度だけ処理されます。たとえば、投稿の一覧に`ul`、`li`を使う場合、ここに`<ul>`を入れるといいでしょう。

❷while (have_posts()) :

次の行ではPHPの`while`文を利用しています。`while`は与えられた条件が真である限り、`endwhile`までのコードを繰り返し処理します。つまり投稿がある場合、WordPressの投稿設定「1ページに表示する最大投稿数」の設定がデフォルトの「10」であれば、ループの中身が最大で10回処理されます。

❸the_post();

`the_post()`はループ内で参照する投稿を順番にセットアップしてくれるWordPress関数です。この関数をループ内で実行することにより、投稿のあらゆるデータを取得できるようになります。WordPressにはループ内でのみ使用できるテンプレートタグ（投稿したさまざまな記事データを出力するためのWordPress関数をとくにこう呼びます）が数多くあります。たとえば、タイトルを出力する`the_title()`、投稿内容を出力する`the_content()`、投稿ページへのURLを出力する`the_permalink()`など。これらをループ内で正しく機能させるには、`while ( have_posts() )`によってループを開始したあと必ず`the_post()`を実行しましょう。

❹endwhile;

`endwhile`でループを終了したあと`else`までの間は、ループ後に一度のみ処理されますので、ループ前に`<ul>`を開始している場合はここに`</ul>`を入れるといいでしょう。

❺else:

`else`から`endif`の間は表示する投稿がない場合に一度のみ処理されますので、ここに「投稿がありません」などと記述するといいでしょう。

❻endif;

最後に`endif`で`if`文を閉じます。忘れるとエラーが生じてページが表示されなくなるので気をつけましょう。

STEP 01 オリジナルテーマの記事一覧をループに書き換えよう

ループについて理解したところで、実際にオリジナルテーマにループを実装しましょう。
テーマのindex.phpを開き記事一覧の場所を探します。
サンプルテーマでは35行目あたりの`<ul class="archive">`から91行目あたりの`</ul>`までが記事一覧です。

1 まずは`<ul class="archive">`の上に`<?php if ( have_posts() ) : ?>`を追加します。この際、`<ul class="archive" />`全体のインデントを1つ下げると見やすくなります。

ソースコード After

```
<div class="main-column">
  <h1 class="box-heading box-heading-main-col">Blog</h1>
  <div class="box-content">

    <?php if ( have_posts() ) : ?>

      <ul class="archive">
        <li class="item-archive">
          <div class="time-and-thumb-archive">
```

2 93行目あたりの`</ul>`の下に、`<?php else : ?>`と`<?php endif; ?>`を追加します。これで、記事がある場合はこれまでどおり`<ul class="archive" />`の中身が表示され、記事がない場合は「投稿がありません」と表示されます。

ソースコード After

```
    </ul>

  <?php else : ?>

    <p>投稿がありません。</p>

  <?php endif; ?>

</div>
<nav class="pagination">
```

3 続いて、33行目あたりの`<ul class="archive">`の下の`<li />`は、同じものが5回繰り返されていますので4つを削除し、1つのみを残した状態で全体的にインデントを1つ下げましょう。その上に`<?php while ( have_posts() ) : ?>`を追加し、さらにその下に`<?php the_post(); ?>`を追加します。最後に`</li>`と`</ul>`の間に`<?php endwhile; ?>`を追加します。

ソースコード After

```
<?php if ( have_posts() ) : ?>

  <ul class="archive">

    <?php while ( have_posts() ) : ?>

      <?php the_post(); ?>

      <li class="item-archive">
        <div class="time-and-thumb-archive">
          <time class="pub-date" datetime="2017-04-01T23:59:99+09:00">2017年4月1日</time>
          <p class="thumb thumb-archive"><a href="single.html"><img src="http://placehold.it/
            190x130"></a></p>
        </div>
        <div class="data-archive">
          <p class="list-categories-archive"><a href="archive.html">カテゴリー名</a></p>
          <h2 class="title-archive"><a href="single.html">記事タイトル記事タイトル
          記事タイトル記事タイトル</a></h2>
        <p class="list-tags-archive">タグ: <a href="archive.html">タグ名</a>,
          <a href="archive.html">タグ名</a>, <a href="archive.html">タグ名</a>,
            <a href="archive.html">タグ名</a></p>
        </div>
      </li>

    <?php endwhile; ?>

  </ul>

<?php else : ?>
```

11-1 カスタマイズAPIとは

**各種テンプレートファイル作成が終わり、
WordPressを使った動的サイトの全ページが正しく表示されるようになりました。
これで公開してもテーマとして通用しますが、
さらにこのテーマに独自のカスタマイズ機能を追加してみましょう。**

テーマオプションを提供する方法

Lesson05から何度か利用したWordPressのテーマカスタマイザー（[外観] → [カスタマイズ] 画面）は、ユーザーに対して、テーマやサイトを、テンプレートファイルをさわらずしてカスタマイズするための統一されたインターフェイスを提供してくれます。これによりテーマの設定・色調・ウィジェット・メニューなど、テーマやサイトに対するさまざまな変更をライブプレビューつきでおこなうことができます。
このテーマカスタマイザーの独自のオプションを今回のオリジナルテーマに追加します。Lesson10までテーマとしては完成といってもかまわないのですが、さらに管理画面からできるテーマ独自のカスタマイズを追加するのがこのレッスンの目的です。これによってこのテーマのユーザーはソースコードにさわることなく外観をアレンジすることができるようになります。

テーマカスタマイザー

テーマカスタマイザーを変更するカスタマイズAPI

開発者はカスタマイズAPIを利用することで、よりパワフルでインタラクティブなカスタマイズ設定をテーマやプラグインに加えることができます。WordPressでは、テーマに対してオプション設定を提供するのに、カスタマイズAPIを利用することがもっとも正統な方法です。

COLUMN

APIとは

API（Application Programing Interface、アプリケーションプログラミングインターフェイス）とは、プログラミングの際にアプリケーションの機能を利用するためのインターフェイス（窓口）です。手間を省くため、簡潔にプログラミングできるよう、さまざまな関数、変数、仕様などが用意されています。

完成イメージ

まずこれからおこなうことの概略を説明しましょう。今回の目標はテーマカスタマイザーを利用して、「フロントページに表示させるコンテンツを複数の固定ページから選べるようにする」ことです。
通常、固定フロントページを設定している場合、サイトのフロントページにはその設定された固定ページのコンテンツのみが表示されます。そこを今回、オリジナルテーマのオプションとして、最大5つの固定ページを選んで、フロントページに表示させることを可能にします。
さらに固定フロントページで「投稿ページ」として設定したページ（本書では固定ページの「ブログ」が該当します）を選んだ場合、固定ページの本文ではなく、最新の投稿を5つ表示するようにします。

具体的な課題

- **管理メニュー［外観］→［カスタマイズ］の中に［Theme Options］というメニュー項目を新たに追加して表示させるようにします。**
- **［Theme Options］を押すとオプションの設定画面に移り、そこには「Front Page Content #」（#は1～5の数字）という5つのメニューボックスを表示させます。**
- **各メニューボックスからは任意の固定ページを選べるようにします。**
- **フロントページのテンプレートファイルを新たに用意して、そのフロントページに選んだ固定ページを順に表示させるようにします。表示させるのは固定ページのタイトルと本文です。**
- **固定ページ「ブログ」を選んだ場合だけは、固定ページの本文の代わりに最新の投稿を5つ表示するようにします。**
- **固定ページは最大5つまで表示できますが、メニューボックスで選択しないかぎり表示しません。**

これだけの機能を実装するのは大変そうに思えますが、WordPressのカスタマイズAPIを使うと比較的簡単に実現できます。

カスタマイズAPIのほかに、1～5のメニューボックスや選んだ固定ページの繰り返し表示には`for`ループを利用し、選択した固定ページの有無の判断にはすでに学んだ`if`文を用いて条件分岐をおこないます。

新しいWordPress関数やアクションも利用するので、総合的な学習になります。最初はすべてを理解できなくてもかまいませんので、まずテーマカスタマイザーを完成させてみて、わからないところは順を追ってCodexなどで調べて理解を深めていってください。

ここでは、このレッスンの鍵となるカスタマイズAPIについて説明しておきましょう。

フロントページ　front-page.php

13-1 検索エンジンとSNSへの施策

本書では検索エンジンによるサイトの認証・サイトマップの公開・SNSとの連携という3つの切り口からサイトの集客につながる設定を紹介します。まずはそれぞれの切り口について解説します。

サイト認証の必要性

サイトを訪れるユーザーは検索エンジンからキーワードで探して来ることも多いでしょう。検索エンジンに認識されないことはサイトへの集客の面で大変不利です。そこでまず主要な検索エンジンにあなたがサイトの所有者であるという認証を受け、検索をされたときに適切に表示されるように管理することがサイト公開後の第一歩です。

GoogleやBingでは、公開したサイトの所有権が確かにあなたにあることを、アカウントに紐づけることで証明することができるようになっています。これをおこなうことで、検索からあなたのサイトに訪れた人々のデータをある程度把握できるようになります。

また、検索エンジンのクローラーがサイトのクロールをおこなった際に発生したエラーを確認できたり、次に説明するサイトマップを送信することができます。これによって検索順位がアップするわけではありませんが、エラー情報から必要な対策を確認でき、検索エンジンからサイトを訪れてもらいやすくなります。セキュリティの評価も同時におこなわれます。マルウェア検出やフィッシングサイトとしての評価がされていないか、定期的に情報を確認するようにしましょう。

サイトの所有を証明するためにはいくつかの方法がありますが、Jetpackプラグインでは出力されるHTML内に検索エンジンアカウント固有の「メタタグ」を埋め込むことができるようになっています。このメタタグがあることで、検索エンジンは所有者を確認できるというわけです。

サイトマップとは

GoogleやBingなどの検索エンジンは「ロボット型検索エンジン」と呼ばれ、ウェブを周期的（サイトの規模や更新頻度により周期はさまざま）に巡回し、サイトの情報を収集しています。このとき、検索エンジンのクローラーがすべてのページを順に巡回することは、それぞれのサーバリソース・ネットワークの利用率から見ても効率的ではありません。

サイトマップは、クローラーがサイトの構造・更新情報などを効率よく収集できるようにする仕組みであり、通常「sitemap.xml」というファイルにまとめられます。サイトマップには本の目次のように、最終アップロード日・通常の更新頻度・読み込むURLの優先度などが記されます。検索エンジンに示すためにサイトマップを作成・設置してあげれば、あなたのサイトをクローラーがより効率的に巡回できるというわけです。

日々更新されるサイトのサイトマップを手動で更新し続けることは大変困難です。Jetpackプラグインの機能を使ってサイトマップの作成を自動化することができます。なお、より細かい設定を必要とする場合は、別のプラグインやテーマのカスタマイズで対応できます。これもWordPressの魅力です。

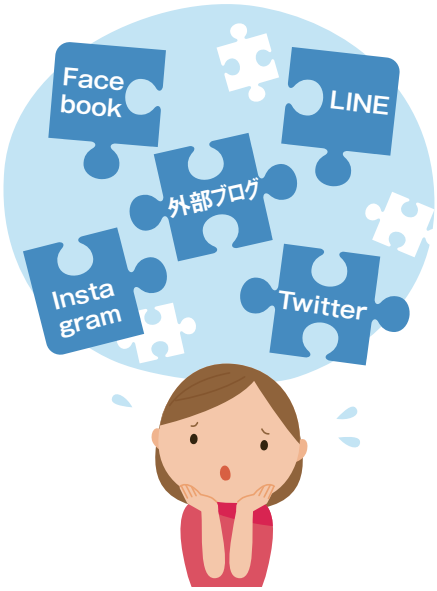
COLUMN

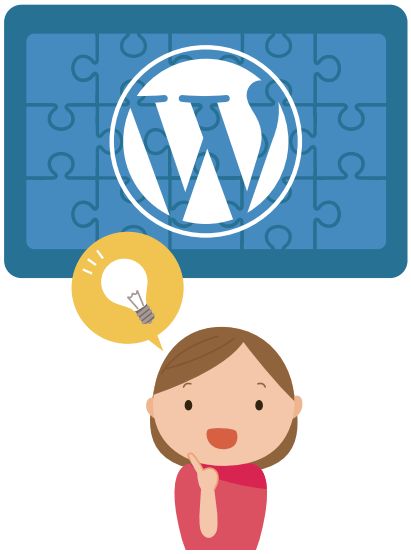
SEOプラグインとサイトマップ

6-2で紹介した「SEO SIMPLE PACK」にはサイトマップ生成機能は含まれていませんので、つづく**13-3**でJetpackを使ってこれをおこないます。海外製のSEOプラグインで有名な「Yoast SEO」「All in One SEO Pack」にはサイトマップ生成機能が搭載されています。

ソーシャル・ネットワーキング・サービスの利用と連携

いまやサイトの集客にソーシャル・ネットワーキング・サービス（SNS）の活用は欠かせません。WordPressではJetpackを利用して新規の記事を投稿したことをさまざまなSNSへ自動的にシェアすることができます。多くのSNSやレンタルブログに情報を無秩序に分散させるのではなく、あなたのサイトに発信したい情報を集約しながら浸透力の高いSNSから鮮度の高い情報を発信し、サイトへの流入をはかることができるというわけです。また、あなたのサイトを見た人がSNSへシェアしやすい環境を整えておくことも依然重要です。SNSへのシェアを促すインターフェイスもJetpackで簡単に実現することが可能です。





無料ブログ・ウェブ作成サービスとの違いとWordPressで得られるメリット

CHECK!

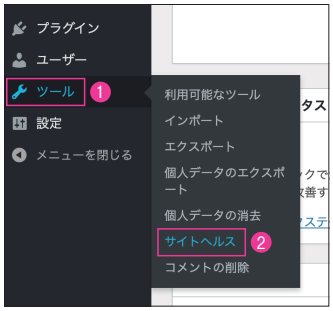
現在は無料で簡単に使用できるさまざまなブログ・ウェブ作成サービスがありますが、バックアップが手元にとれないサービスでは蓄積したコンテンツを失ってしまうことがあるかもしれません。WordPressサイトを運用すると手間は増えるかもしれませんが、すべてのデータが自分の管理で運用できるのは大きなメリットです。万一契約しているサーバでサービスの終了や不具合が発生しても、同じWordPressで別のサーバに引越しして継続的な情報発信ができます（WordPressサイトのバックアップ方法はLesson14で紹介します）。ただコンテンツを作成するだけではなく、継続的な運用についても考慮に入れてプラットフォームを選択するようにしましょう。

14-1 サイトヘルスで状況を確認しよう

WordPressの5.2より別途インストールが必要なプラグインであったサイトヘルスが標準の機能として搭載されました。ダッシュボードから簡単にWordPressの動作しているサーバ環境、プラグイン・テーマのバージョンや使用状況を手軽に把握することができます。

サイトヘルスで動作状況を確認する

1 サイトヘルスを確認するには、管理メニューから［ツール］❶→［サイトヘルス］❷を選択します。



2 サイトヘルスステータスが表示されます。今回は停止中のプラグイン、テーマを削除するという改善点が表示されています。詳細は各項目の右にある▽をクリックすることで表示されます。

3 ［情報］タブを選択し❶、利用環境の確認をおこないましょう。ここではWordPressウェブサイトの構成に関するすべての詳細が表示されています。すべての情報を一覧にしてエクスポートするには、［サイト情報をクリップボードにコピー］ボタンを押してクリップボードにコピーします❷。その後、テキストファイルに貼りつけて端末に保存することができます。

COLUMN

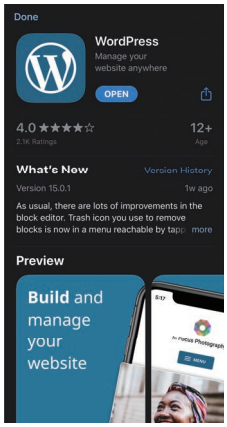
停止中のテーマの削除について

有効にしているテーマに問題が発生した際、WordPressは自動的にデフォルトテーマTwenty Twentyを使用するため、デフォルトテーマは削除しないようにしておきましょう。

14-2 スマホアプリでWordPress管理

WordPressのモバイルアプリを利用すれば、
記事やページの投稿、編集だけでなく、統計情報の確認や、
アクセスの急増やコメントがついた際にアプリにプッシュ通知がされるため、
パソコンが手元にない場合でも手軽にサイトの管理をおこなうことができます。

スマートフォンでWordPressサイトを管理しよう



モバイル管理アプリのインストール

WordPressを開発しているAutomattic社から、モバイル用の管理アプリがリリースされています。iPhone/iPadはApp Store、Android端末はGoogle Playからアプリをインストールしましょう。使っているデバイスに応じて、それぞれのストアから「WordPress」で検索してインストールします。機能的に大きな違いはありませんので、ここではiPhoneのアプリ画面で紹介をおこないます。

iPhone：https://apps.apple.com/jp/app/wordpress/id335703880
Android：https://play.google.com/store/apps/details?id=org.wordpress.android

管理サイトを追加する

WordPressの記事、ページの投稿、編集はアプリの設定画面からサイトを追加することで可能になります。

1

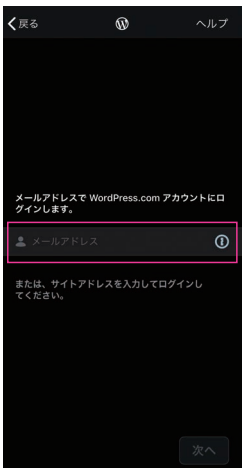
アプリを起動します。起動画面で［ログイン］をタップします。

2

WordPressサイトの管理者として、Jetpackに登録したアカウントで認証をおこなうので、［WordPress.comで続ける］をタップします。

3

つづいて、登録時に用いたメールアドレス、パスワードを入力し［次へ］をタップします。



これでスマホアプリからの利用が可能になりました。

15-3 ブロックエディターを拡張しよう

WordPressのブロックエディターにはさまざまなブロックが用意されていますが、
実際にウェブサイトをつくるなかで不足を感じることもあるでしょう。
ここでは、プラグインによって手軽に便利なブロックを追加したり、
オリジナルのブロックをつくってみましょう。

STEP 01　プラグインを使ってブロックを追加しよう

sample-data ▶ Lesson15

Lesson04で学んだとおりブロックエディターにはさまざまなブロックが備わっていますが、制作ニーズに応じて、さまざまなブロックを追加したい人もいるでしょう。まずはお手軽にプラグインを使って、よく使われそうなブロックを追加する方法を見てみましょう。

プラグイン「VK Blocks」の導入

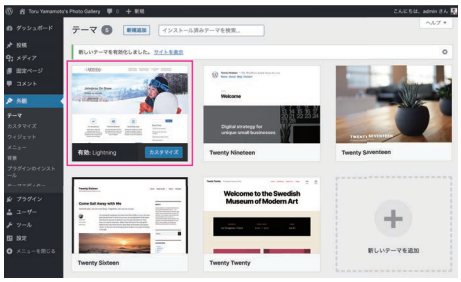
ブロックを追加するためのプラグインはいくつかありますが、今回はLesson05で紹介したテーマ「Lightning」と親和性が高いブロック追加プラグイン「VK Blocks」を例にしてブロックの追加をしてみましょう。

COLUMN

カスタムフィールドを紹介していないのはなぜ？

タイトルや本文以外に、定形のコンテンツを入れる枠をつくりたいときに多用されたのがカスタムフィールドという仕組みです。このカスタムフィールドは「Advanced Custom Fields」などのプラグインとともに、WordPressのコンテンツ管理に欠かせないものとして扱われてきました。
しかしカスタムフィールドに入力したデータの出力には原則としてテーマのカスタマイズが必須であり、テーマを変更するとカスタムフィールドのコンテンツは出力されなくなるケースがほとんどでした。これはLesson01で述べた「見た目とコンテンツの分離」という観点で望ましくありません。そこで本書では簡易なカスタムブロックを用いてコンテンツを管理する方法を紹介しています。

1 管理メニュー［外観］から、テーマを「Lightning」に変更しましょう。「VK Blocks」はどのテーマでも利用することができますが、ブロックのスタイルが「Lightning」に最適化されています。



2 つづいて［プラグイン］から「VK Blocks」をインストール・有効化します。

3 ［投稿］→［新規作成］から新しい投稿を作成し、タイトルを「VK Blocks のサンプル」としましょう❶。エディタの ▦ ボタン→［すべて表示］ボタンを押してブロック挿入パネルを出し、［VK Blocks］セクションから［新FAQ］ブロックを押します❷。

4 ［新FAQ］ブロックが追加されますので、「Q」と「A」にそれぞれ文章を打ち込んでみましょう。こうしてQ&Aコンテンツが作成できました。繰り返し［新FAQ］ブロックを使用することで「よくある質問集」風のページを作成することができます。

5 つづいてインタビュー記事などでよく使われる、吹き出し表現を用いた会話風のコンテンツをつくってみます。エディタの ⊞ ボタンを押してブロック挿入パネルを出し、［VK Blocks］セクションから［吹き出し］ブロックを追加します。

6 話者の顔画像・話者の名前・そしてセリフの内容を入力します。ひとつできたら、画像を参考に、その下にもうひとつ［吹き出し］ブロックをつくり、会話している風のコンテンツをつくってみましょう。サンプルデータに話者の顔を表す画像を2種類用意していますので、こちらを利用してください。なお画面右のブロックオプションから位置の左右・吹き出しのタイプを変更可能です。

CHECK！

ブロックを追加するプラグイン

ほかにも、ブロックを追加するプラグインは多く存在しています。VK Blocksと同じく国内の開発者が制作したテーマに対応している「Snow Monkey Blocks」や、広く人気を集めている「Stackable」などがありますので、ぜひトライしてみてください。

STEP 02　プラグインでオリジナルブロックをつくってみよう

sample-data ▶ Lesson15

VK Blocksで追加できるブロックはあくまでプラグインが提供するものだけでしたが、管理者自身がブロックをデザインして制作するにはどうすればいいのでしょうか。今回はプラグイン「Genesis Custom Blocks」を用いて、簡単なオリジナルブロックをつくってみます。

Gutenberg Block API

CHECK！

WordPressではGutenberg Block APIというオリジナルブロックを制作するための仕組みが提供されていますが、これを使いこなすにはReactというJavaScriptライブラリを利用する必要があり、かなり難しい作業となるため、本書では取り扱いません。

Genesis Custom Blocksの導入と入力

1 ［プラグイン］から「Genesis Custom Blocks」をインストール・有効化すると、管理メニューに［Custom Blocks］が追加されます。［Custom Blocks］→［All Blocks］に移動しましょう。