Engineer

WordPress

ウェブサイト用のオープンソースのコンテンツ管理システム

どのようなものか

WordPressは、オープンソースのコンテンツ管理システムです。PHPで動き、サイトの投稿、固定ページ、メディアの記録、利用者、設定をMySQLまたはMariaDBのデータベースに保存し、要求に応じてページを組み立てます。アップロードしたメディアファイル本体は、通常、データベースの外に置きます。名前はWPと略されることが多く、同じ接頭辞を自身のファイルやフォルダー名にも使います（wp-config.php、wp-content、wp-admin）。GPLv2以降のライセンスで公開されており、ソフトウェアのダウンロード、実行、変更に料金はかかりません。

WordPressサイトには3つの層があります。コアは、全員がダウンロードして更新する共通のコードです。テーマは表示を、プラグインは振る舞いを担当し、プロジェクト自身のテーマハンドブックも、この境界を示しています。どちらも、コアの中で別のコードを動かせる、名前付きの箇所であるフックを通して機能を加えます。フックにはアクションとフィルターがあります。アクションは、投稿の保存時などの時点で実行され、何も返しません。フィルターは値を受け取り、変更して返します。次の例は、自動生成する投稿の抜粋を、初期設定の55語から30語に短くします。

```
add_filter( 'excerpt_length', function ( $length ) {
    return 30;
} );
```

テーマの形は、時代とともに変わってきました。クラシックテーマは、2005年のWordPress 1.5で導入されたPHPのテンプレート方式を使い、今も広く利用されています。現在の方式はブロックテーマです。テンプレートはブロックで構成するHTMLファイルで、サイトエディターから編集でき、色、フォント、間隔はtheme.jsonで管理します。ハイブリッドテーマは、ブロック機能の一部を取り入れたクラシックテーマです。

コンテンツは、ブロックエディター自身も使うREST APIを通して、JSONとして外部に取り出せます。サイトの/wp-json/にリクエストすると、提供する全ルートの一覧が返ります。公開コンテンツはログインせずに読めますが、非公開のものには認証が必要です。どのプログラムからでも、次のように最新の3投稿を取得できます。

```
curl -s 'https://example.com/wp-json/wp/v2/posts?per_page=3'
```

実行環境として、プロジェクトはPHP 8.3以上、MySQL 8.0以上またはMariaDB 10.11以上、HTTPSを推奨し、ウェブサーバーにはApacheかNginxを勧めています。JoomlaとDrupalも同種のコンテンツ管理システムです。ほかに比較すべき選択肢は、Djangoなどのフレームワーク上の独自アプリケーションと、静的サイトジェネレーターです。

概念を判断に生かす

「WordPressとは？」を実際の判断に生かしましょう。その概念を、開発または購入するシステムに結び付けて考えるお手伝いをします。 表示が速く、検索で見つけやすく、文章を変えるたびに料金が発生せず、自社の担当者が編集できるサイトをつくります。

サービスを見る お問い合わせ →

用語の背後にある前提を検証する

解説とリンク先の実績を参考に、プロジェクトについての質問を整理してください。ツールを選ぶ前に、前提を一緒に検討できます。 HTML/HTML5でWebアプリケーション100個を開発 — 100 構築したWebアプリケーション

HTML/HTML5でWebアプリケーション100個を開発 お問い合わせ →

なぜ重要なのか

WordPressを選ぶ際の本当の判断は、公開後に誰がサイトを更新し、その間に誰が安全を保つかです。WordPressは、技術担当ではない人にも、一通りの編集画面を提供します。開発担当に依頼を出さず、ページの公開、画像の差し替え、利用者の追加、メニューの作り直しができます。毎週変わるサイトでは、開発者を必要としない編集が、最も安く済みます。

ライセンスは無料でも、サイトは無料ではありません。ソフトウェアにはGPLが適用され、wordpress.orgは、プラグインとテーマも同じライセンスを引き継ぐ派生物だという立場です。支出は別の場所に移ります。PHPとデータベースを動かすホスティング、テーマを選び、設定し、テストする時間、そしてコア、プラグイン、テーマ、PHPのバージョンを新しく保つ作業です。最後の費用はサイトが存在する間ずっと続くため、見積もりには誰が担うかを明記すべきです。

保守が必要なのは、WordPressが稼働中のアプリケーションだからです。キャッシュを使わずにページを生成すると、PHPがデータベースにアクセスします。フルページキャッシュなら、保存済みのページを返せます。プラグインのコードはコアと同じ権限で動き、誰かが別の防御を前に置かない限り、ログイン画面はインターネットに公開されています。コアのバックグラウンド更新は3.7で導入されました。5.6以降の新規インストールでは、設定や更新対象となる条件に左右されますが、初期状態でマイナー更新とメジャー更新の両方が有効です。5.6より前のインストールは、変更しない限り以前の設定を保ちます。PHPには別の更新日程があります。

ページや投稿とはまったく異なるデータや業務の流れには、独自実装が適しています。ただし、その編集手順も作り、保守しなければなりません。静的サイトは、リクエストの処理経路からPHPとデータベースのアプリケーションを取り除きますが、サーバーやビルドツールの保守は残ります。通常、編集者はファイルとビルド工程を通して作業します。WordPressが適するのは、その間の広い範囲です。所有者が自ら運営する、比較的標準的なサイトで、編集画面の価値が保守の負担に見合う場合です。コードがGPLで、コンテンツは通常のSQLデータベースにあり、REST APIから投稿、固定ページ、分類をJSONで取得できるため、乗り換えを妨げる制約も小さく済みます。

実際の活用例

当社の業務では、WordPressは、目的に合わせて選ぶ複数の答えの1つです。100のウェブアプリケーションを開発した実績は、2003年から2014年の業務と、選択の基準を説明しています。独自のものが必要な顧客には、HTML/HTML5、CSS/SCSS、Djangoで構築しました。自ら管理できる、より標準的なサイトが必要な場合は、WordPressかJoomlaのほうが早く、理にかなっていました。

その実績が重視するのは、ツールではなく判断です。必要のない独自開発を思いとどまってもらうことも、本当に必要なら独自開発を選んでもらうことも、業務の一部でした。

見積もりを依頼する前に発注者が答えられる問いにすると、その判断は次のようになります。

- 開発者を介さずに毎週編集するのか、それとも変更は年に数回なのか。
- 内容は固定ページ、投稿、分類、メディアに収まるのか、それとも独自のデータモデルと規則が必要なのか。
- サイトが存在する間、誰がコア、プラグイン、PHPを最新に保つのか。

- 手入れされた少数のプラグインで運営するのか、それとも数十をつなぎ合わせてアプリケーションのようにするのか。

毎週の編集、仕組みに合う内容、担当の保守者、少数のプラグインという条件なら、WordPressなどのCMSが候補になります。特殊なデータや、最新に保つ担当者の不在は、独自実装か静的サイトを選ぶ理由になります。

問題が起きるところ

普通の更新でサイトが壊れる

プラグインを更新したところ、ページにエラーが出たり、管理画面が開かなくなったりします。長年にわたってプラグインが増え、作者が放棄したものもあり、それぞれが全面的なアクセス権で動く第三者のPHPコードとなり、そのうち2つが衝突した例です。ハードニングのハンドブックが求めるように、数を抑え、使わなくなったプラグインは削除します。更新前には毎回バックアップを取り、メジャー更新は先に複製環境で試し、作者が今もリリースを続けているプラグインを優先します。

更新やデザイン変更で独自の変更が消える

数か月動いていた変更が、テーマ更新の日に消えたり、業務に必要な機能が、サイトの見た目を変えたときになくなったりします。前者は、他人のテーマのファイルを直接編集しており、更新で置き換えられた場合です。テーマハンドブックも、テーマのfunctions.phpに直接追加した関数は、次の更新で消えると警告しています。後者は、サイトに必須の振る舞いをテーマに入れていたため、テーマと一緒に失われた場合です。他人のテーマへの変更は子テーマに置き、機能はコアのフックを使う小さな独自プラグインに置きます。テーマには表示だけを担わせます。

管理者のパスワードからサーバー上のコード実行に進まれる

管理者1人のパスワードを盗んだ者は、投稿の編集から独自のPHPの実行へ進めます。管理画面に、テーマとプラグインのファイルを編集する機能があるためです。wp-config.phpの1行で、このエディターを無効にできます。全利用者からedit_themes、edit_plugins、edit_filesの権限を取り除くのと同じ効果があります。

```
define( 'DISALLOW_FILE_EDIT', true );
```

これは複数ある経路の1つを閉じるだけです。ハンドブックも、不正なファイルをアップロードできる攻撃者は防げないと明記しています。強いパスワード、全管理者の2段階認証、業務に必要な最小限の管理者アカウントと組み合わせます。

現行リリースから取り残される

管理画面には何か月も新バージョンの通知が出ているのに、サイトのメジャーリリースやPHPの版が、現行版から大きく遅れます。コアのマイナー更新だけを有効にしたサイトは、自動で修正を受け続けながら、メジャーバージョンとPHPが古くなり、依存するプラグインが新しい版を要求する段階に至ることがあります。5.6以降の新規インストールではメジャー更新も初期状態で有効ですが、設定で変えられます。メジャー更新の予定を組み、同時にプロジェクトの要件ページでPHPのバージョンも確認し、両方を複製環境で試してから、本番サイトに適用します。

参考資料

- WordPress文書 — 投稿、メディアから、インストールと保守まで、サイト運営の公式マニュアル。
- 動作要件 — プロジェクトが現在推奨するPHP、MySQL、MariaDBのバージョン。
- WordPressのアップグレード — バックグラウンド更新、ワンクリック更新、手動更新と、その前に取るバックアップ。
- WordPressのハードニング — ファイル権限、データベース権限、ファイルエディターを止める設定、プラグインの整理。
- プラグインハンドブック: フック — コアを変えずにWordPressの動作を変える、アクションとフィルター。
- REST APIハンドブック — ブロックエディターが使うAPIのルート、エンドポイント、認証。

用語集

実際の制約に合わせて選ぶ

業務の流れ、既存のインターフェース、変更を必要とする利用者をお知らせください。検証可能な納品内容と、お客様のチームが保守を続けられる引き継ぎ内容を定められます。

料金を見る お問い合わせ →

用語を実務に結び付ける

「WordPressとは？」がお客様の組織にどう当てはまるかをご相談ください。現在のシステムの具体例が一つあれば、話を始められます。

サービスを見る

welcome@engineer.company +45 50 10 61 84 デンマーク、コペンハーゲン

このページを持ち歩く

連絡先カード

心を込めて丁寧に制作： D.

CVR 45969703 · Tagensvej 105C, 3. th, 2200 København N

著作権 © 2025 – 現在 · Engineer ApS · すべての権利を留保します。

印刷元： https://engineer.company/ja/glossary/wordpress/