WordPress AI自動更新システム 仕様書 & 導入マニュアル（汎用版）

**バージョン: 1.0 ／ 対象読者: AIやプログラミングの知識がない運用担当者**

このシステムは、**Claude Code（AIエージェント）**を使って、WordPressサイトの記事（コラム・Q&A など）を**半自動で作成・更新**するための仕組みです。この文書1つで「何ができるか」「どう入れるか」「どう使うか」「自分のサイト向けにどう変えるか」がわかるようにまとめています。

> 📌 ***困ったら、この文書を Claude Code に見せて「この部分が分かりません」と聞いてください。***
> *Claude Code があなたの代わりに調べたり、設定を手伝ったりできます。*

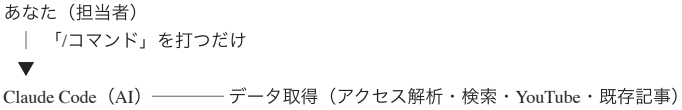
目次


- 基本の考え方（3分でわかる全体像）
- このシステムでできること
- 全体の仕組み（アーキテクチャ）
- 準備するもの（チェックリスト）
- 導入ステップ（バイステップ）
- 各機能の使い方
- 安全・コンプライアンス（重要）
- 自分のサイト向けにカスタマイズ
- 運用のコツ
- 引き継ぎ・鍵の所有権
- トラブルシューティング

- 付録A: コマンド早見表
- 付録B: .env 項目一覧
- 付録C: ファイル構成


## 0. 基本の考え方（3分でわかる全体像）

- **人が毎回文章を書く**のではなく、**AI（Claude Code）が記事の下書きを作る**仕組みです。
- 作られるのは基本的に**「下書き」**です。**公開は必ず人が内容を確認してから**行います（暴走・誤情報を防ぐため）。
- AIは、次のような「材料」をもとに記事を作ります。
- サイトのアクセス解析（どんな記事が読まれているか）
- 検索データ（どんな言葉で検索されているか）
- YouTubeの新着動画
- 既存の記事一覧（重複を避けるため）
- 記事だけでなく、**アイキャッチ画像**や**タグ**も自動で付けられます。
- **あなたがやること**は、基本的に「コマンドを打つ → 出てきた下書きを確認 → OKなら公開」だけです。



```
  │              └ 記事本文・画像・タグを自動生成
  ▼
WordPress に「下書き」として保存
  │
  ▼
あなたが確認 → 問題なければ「公開」
```

1. このシステムでできること

| | | |
|---|---|---|
| **季節記事の自動下書き** | アクセス解析＋検索データ＋「旬」から記事ネタを決め、コラムの下書きを作成 | /seasonal-articles |
| **YouTube取り込み** | 指定チャンネルの新着動画を、埋め込み記事として下書き化 | /youtube-import |
| **Q&A（FAQ）自動生成** | 既存と重複しない新しいQ&Aを下書き作成 | /faq-articles |
| **アイキャッチ画像の自動生成** | AIが記事テーマに合う写真を生成し、タイトル文字をのせて設定 | 各コマンドに内蔵／/faq-eyecatch-backfill ／/column-eyecatch-backfill |
| **タグの自動付与** | 記事内容に合うタグを、既存タグ優先で付与 | /faq-articles 他 |
| **既存記事の分析・改修** | 内容が薄い記事などを分析し、加筆改善（承認制で反映） | /faq-content-improve |
| **関連記事クラスタ（囲い込み）** | 深い記事＋周辺の関連記事を作り、内部リンクで束ねる | 上記の組み合わせ |

**すべて「下書き」で作成**します。公開は人の確認後です。

2. 全体の仕組み（アーキテクチャ）

このシステムは、次の3つの層でできています。

- **Claude Code（AIエージェント）**

- あなたが打つ「コマンド」を受けて、記事の企画・執筆・画像やタグの決定を行う頭脳。
- .claude/commands/ フォルダの「手順書（コマンド定義）」に沿って動きます。

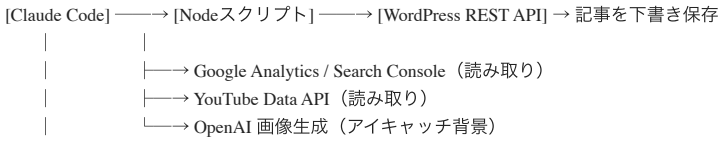
- **Nodeスクリプト（データの出し入れ）**

- 記事の材料を取ってきたり、WordPressに投稿したりする小さなプログラム群（scripts/ フォルダ）。
- **記事作成のコア部分は追加インストール不要**（パソコンにNodeが入っていれば動きます）。
- アイキャッチ画像の合成だけ、画像用の部品を1回だけ入れます（後述）。

- **外部サービス（API）**

- WordPress（記事の保存先）
- Google Analytics 4 / Search Console（アクセス・検索データ／季節記事で使用）
- YouTube Data API（動画取り込みで使用）

- OpenAI（アイキャッチの背景画像生成で使用）




**大事なポイント：** 秘密の鍵（パスワードやAPIキー）は、.env という1つのファイルにだけ書きます。
このファイルは外部（GitHubなど）には公開されません。

3. 準備するもの（チェックリスト）

導入前に、次のものを用意します（機能ごとに必要なものが違います）。

**必須（全機能共通）**

- [ ] パソコン（Windows / Mac）
- [ ] Claude Code（インストール済み）
- [ ] Node.js の LTS版（node --version で 18 以上）
- [ ] このシステムのフォルダ一式
- [ ] WordPressサイト（**REST APIが有効**であること。多くのサイトは既定で有効）
- [ ] WordPressの **アプリケーションパスワード**（投稿用。取得方法は後述）

**機能ごとに必要（使う機能だけ用意すればOK）**

- [ ] 季節記事 → **GA4 と Search Console のサービスアカウント（JSON）**、GA4プロパティID、サイトURL
- [ ] YouTube取り込み → **YouTube Data API キー**、対象チャンネルID
- [ ] アイキャッチ生成 → **OpenAI APIキー** ＋ 画像用の部品（npm install・1回だけ）

💡 ***鍵はすべて「自社（御社）名義」で作成することを強くおすすめします***（→9. 引き継ぎ）。

4. 導入ステップ（バイステップ）

ステップ 4-1. Claude Code と Node.js を用意する

- claude.com/claude-code の案内に沿って Claude Code を入れる。
- nodejs.org から「LTS」を入れる。
- ターミナル（Macは「ターミナル」、Windowsは「PowerShell」）で node --version と打ち、v18 以上が出ればOK。

ステップ 4-2. フォルダを置く

- システムのフォルダ一式を、分かりやすい場所（例：書類フォルダ）に置く。
- そのフォルダを Claude Code で開く。
- まず動作確認（鍵はまだ不要）：ターミナルで npm test と打ち、テストがすべて通ればプログラム本体は正常です。

ステップ 4-3. 鍵（.env）を設定する

- フォルダ内の .env.example をコピーして、名前を .env にする（例：cp .env.example .env）。
- .env をテキストエディタで開き、必要な値を入れる（→ 付録B）。
- 各鍵の取り方：

**(A) WordPress アプリケーションパスワード**

- WordPress管理画面 →「ユーザー」→「プロフィール」→ 下部の「アプリケーションパスワード」で発行。
- 発行された文字列（スペース入り）と、そのユーザー名を .env の WP_USER / WP_APP_PASSWORD に入れる。
- サイトのURLを WP_BASE_URL に入れる（例：https://example.com）。

**(B) GA4 ＋ Search Console のサービスアカウント**（季節記事を使う場合）

- 詳しくは **同梱の「GA4-SearchConsole 設定手順」** を参照（このシステムに含まれる別ファイル）。
- 要点：Google Cloud で「サービスアカウント」を作り、JSONキーを発行。そのメールを **GA4（閲覧者）** と

**Search Console（制限付き）** に追加。GA4プロパティID（数字）とサイトURLを控える。

**(C) YouTube Data API キー**（YouTube取り込みを使う場合）

- Google Cloud で「YouTube Data API v3」を有効化 → APIキーを発行。
- .env の YT_API_KEY に入れ、YT_CHANNEL_ID に対象チャンネルのID（UC...）を入れる。

**(D) OpenAI APIキー**（アイキャッチ画像を使う場合）

- OpenAIの管理画面 でAPIキーを発行し、.env の OPENAI_API_KEY に入れる。
- 画像1枚あたりおおよそ **$0.02〜0.2**（従量課金）。
- この機能だけ、フォルダ内で **`npm install` を1回だけ**実行して画像用の部品を入れます。

💡 ***JSONを `.env` に1行で貼るのが難しいときは、ファイルを Claude Code に渡して***
*「このJSONを .env の GA_SERVICE_ACCOUNT_JSON に設定して」と頼めば、AIが設定してくれます。*

ステップ 4-4. 動作確認

- **鍵なしで確認**（形だけ）：各スクリプトは SAMPLE=1（ダミーデータ）や DRY=1（送信しない）で試せます。

例：SAMPLE=1 node scripts/youtube-fetch.mjs

- **実接続で確認**：鍵を入れたら、node scripts/ga-fetch.mjs などを実行してデータが表示されればOK。
- エラーが出たら、**日本語のメッセージをそのまま Claude Code に貼って**相談してください（多くは .env の入れ間違い）。

5. 各機能の使い方

いずれも、Claude Code をフォルダで開いて、記載のコマンドを打つだけです。

5-1. 季節記事を作る /seasonal-articles

- Claude Code で /seasonal-articles と打つ。
- AIがアクセス解析・検索データ・「今〜数週間先の旬」から**記事ネタを1〜3本**提案します。
- 提案されたテーマを確認し、本文が生成され、**コラムの下書き**として保存されます。
- （OpenAIキーがあれば）続けて**アイキャッチ**も自動生成・設定されます。
- 管理画面の「下書き」で内容を確認し、問題なければ公開。

5-2. YouTube取り込み /youtube-import

- /youtube-import と打つ。
- AIが指定チャンネルの**新着動画**を検知し、埋め込み記事の**下書き**を作ります。

- 動画の埋め込み＋「見どころ」＋概要を、動画の概要欄から自動で整形します。

- 同じ動画を二重に投稿しないよう、処理済みは自動で記録されます。
- 下書きを確認して公開。

5-3. Q&A（FAQ）を作る /faq-articles

- /faq-articles と打つ。
- AIが**既存のQ&Aをすべて確認**し、**重複しない新しい質問**を1〜3本決めます。
- 回答本文を、既存のフォーマットに合わせて生成（**下書き**）。
- タグも自動で付きます。（OpenAIキーがあればアイキャッチも）
- 下書きを確認して公開。

5-4. アイキャッチを一括で付ける

- FAQに付ける：/faq-eyecatch-backfill
- コラムに付ける：/column-eyecatch-backfill

- コマンドを打つと、AIが**アイキャッチ未設定の記事**を一覧化します。
- 記事ごとに、短い見出し＋テーマに合う背景画像を生成して設定します（1回に数件ずつ）。
- まず数件で仕上がりを確認 → 問題なければ残りも実行。

- ⚠ 公開中の記事は**即反映**されます（画像を足すだけの低リスク変更）。

5-5. タグを付ける

- 新規記事作成時に自動で付きます。既存記事にまとめて付けたい場合も、AIに

「タグ未設定の記事にタグを付けて」と頼めば、既存タグを優先して付与します。

5-6. 既存記事を充実させる /faq-content-improve

- /faq-content-improve と打つ。
- AIが**内容が薄い記事**などを分析し、「課題＋改訂案」をレポートで提示。
- あなたが**反映してよい記事を選ぶ**（承認）。
- 承認した記事だけ、本文が更新されます（即反映・WordPressのリビジョンで元に戻せます）。

5-7. 関連記事クラスタ（囲い込み）

- 1本の深い記事（ハブ）と、その周辺の関連記事（スポーク）を作り、**内部リンクで束ねる**手法です。
- 手順の例：

- ハブになる記事を作る（例：文字起こしや社長の話をもとにコラム化）。
- 関連する短めのFAQ・コラムを数本作り、**各記事からハブへリンク**。
- ハブ記事にも「関連記事」欄を作り、スポークへリンク（双方向）。

- 検索でスポークに入った読者を、深いハブ記事へ自然に誘導できます。

6. 安全・コンプライアンス（重要）

- **下書き運用が基本**：AIが作るのは下書き。**公開は必ず人が内容を確認**してから行う。
- **薬機法・景品表示法**（健康・美容・食品・医療などに関わるサイトの場合）：
- 「病気が治る」「免疫が上がる」「デトックス」等の**効能・治療効果の断定は書かない**。
- 効果に触れるときは「一般に」「〜と言われています」「個人差があります」等で**断定を避ける**。
- 「最高」「No.1」「絶対」等の**根拠のない最上級・比較は書かない**。
- このガードレールは、記事生成コマンドの手順書（.claude/commands/）に組み込みます（→ 7. カスタマイズ）。
- **事実・引用**：数値・データ・引用文献は**実在するものだけ**を使う（AIに捏造させない）。

引用を載せる場合は、公開前に**内容の裏取り**をする。

- **著者・発言**：特定の人物の発言・仮説として書く場合は、公開前に本人・担当の確認を取る。

7. 自分のサイト向けにカスタマイズする

このシステムは、設定と手順書を書き換えるだけで、**別のサイト・別の業種**に使えます。
主に変えるのは次の箇所です。

| | |
|---|---|
| .env | サイトURL・各鍵・対象チャンネル・GA4プロパティ・カテゴリID など（サイト固有の値） |
| .claude/commands/*.md | 各機能の手順書。**トーン（文体）・業種・ガードレール（薬機法など）・投稿タイプ・カテゴリ**をここで指定 |
| 投稿タイプ | 記事の入れ先（post＝通常投稿/コラム、faq＝Q&A など）。type で指定 |
| カテゴリ／タグ | 既定カテゴリID（WP_CATEGORY_ID）、タグ名の付け方 |
| アイキャッチの画風 | 画像生成プロンプトの方向性（例：写実的な畑／料理／店舗 など） |
| 対象 | YouTubeチャンネルID、季節記事の解析期間 など |

**カスタマイズの進め方（初心者向け）**

- まず .env をそのサイトの値に変える。
- .claude/commands/ の手順書を開き、**会社名・商品名・業種・文体・守るべきルール**を、そのサイト向けに書き換える。

- 難しければ、Claude Code に「このサイトは〇〇業なので、手順書のトーンとガードレールを合わせて」と頼めば手伝ってくれます。

- 少量（1〜2記事）で試して、仕上がりを見て微調整する。

💡 ***業種別ガードレールの例***
*- 健康・食品・化粧品 → 薬機法・景表法（効能断定NG）*
*- 金融・投資 → 断定的な利益表現・元本保証の示唆NG*
*- 医療・クリニック → 医療広告ガイドライン準拠（体験談・ビフォーアフターの扱いに注意）*

8. 運用のコツ

- **繁忙期でも回る仕組みに**：毎回ゼロから手で書くのではなく、コマンドで下書きを量産→確認、の流れにする。
- **少量ずつ確認**：一度に大量に公開せず、数件ずつ確認して品質を担保する。
- **1本の記事を多方面へ**：作った記事を、SNS・LINE・メールなど**複数の場所に展開**すると波及力が増す（別途連携が必要）。
- **重複を避ける**：新規Q&A・季節記事は、AIが既存記事を見て重複を避けるので、テーマがかぶりにくい。
- **記録を残す**：作成済みテーマ・処理済み動画は自動で記録され、二重作成を防ぐ。

9. 引き継ぎ・鍵の所有権（契約・担当交代に備えて）

- **鍵（APIキー・サービスアカウント・アプリパスワード）は、すべて「自社（サイト運営者）名義」で作成**する。
- 制作者・外注先のアカウントで作ると、契約終了時にサイトの自動更新が止まる／外部が自社データにアクセスできる、という問題が起きます。
- 引き継ぎ時のチェック：

- .env の全ての鍵が自社名義になっているか確認。
- 作業用PCの .env（テスト用の鍵が入っている場合）を削除、または鍵を失効。
- コードのリポジトリ（GitHub等）を使っている場合は、外部担当の権限を外す／自社アカウントへ移管。

10. トラブルシューティング

| | |
|---|---|
| 〇〇 が未設定です | .env の該当項目が空、または .env の置き場所が違う（scripts と同じ階層に置く） |
| WordPress投稿失敗（401） | ユーザー名かアプリケーションパスワードが違う → 再発行 |
| WordPress投稿失敗（403 / rest_forbidden） | REST APIがブロックされている（セキュリティプラグイン・サーバー設定）。共有サーバーでは .htaccess に Authorizationヘッダを渡す設定が必要な場合あり |
| GA/GSC で権限エラー | サービスアカウントのメールを GA4（閲覧者）と Search Console（制限付き）の**両方**に追加したか確認。反映に数分かかることも |
| 数値が全部0 | GA4プロパティIDの取り違え（測定ID G-XXX を入れていないか） |
| 画像生成でエラー | OPENAI_API_KEY 未設定、課金上限、または npm install 未実行 |
| その他 | **エラーメッセージ全文をコピーして Claude Code に貼り、「このエラーを直したい」と相談** |

付録A: コマンド早見表

| | |
|---|---|
| /seasonal-articles | 季節記事の下書き作成（アクセス解析＋旬） |
| /youtube-import | YouTube新着の埋め込み記事化 |
| /faq-articles | 新規Q&Aの下書き作成（重複回避） |
| /faq-eyecatch-backfill | FAQのアイキャッチ一括付与 |
| /column-eyecatch-backfill | コラムのアイキャッチ一括付与 |
| /faq-content-improve | 既存記事の分析・改修（承認制） |

付録B: .env 項目一覧

```
# ── WordPress（全機能共通）──
WP_BASE_URL=https://example.com        # サイトのURL
WP_USER=                               # 投稿用ユーザー名
WP_APP_PASSWORD=                       # アプリケーションパスワード
WP_CATEGORY_ID=1                       # コラムなど既定カテゴリのID

# ── Google Analytics 4 ＋ Search Console（季節記事）──
GA_PROPERTY_ID=                        # GA4プロパティID（数字。G-XXXとは別）
GA_SERVICE_ACCOUNT_JSON=               # サービスアカウントJSONの中身（1行で）
GA_LOOKBACK_DAYS=30                    # GA取得期間（任意）
GSC_SITE_URL=https://example.com/      # 例: https://example.com/ または sc-domain:example.com
GSC_LOOKBACK_DAYS=28

# ── YouTube（取り込み）──
YT_API_KEY=                            # YouTube Data API v3 キー
YT_CHANNEL_ID=                         # 対象チャンネルID（UC...）
YT_MAX_NEW=5                           # 1回で処理する新着の最大件数

# ── OpenAI（アイキャッチ画像生成）──
OPENAI_API_KEY=                        # OpenAI APIキー（従量課金）
```

*.env は秘密情報のファイルです。**GitHub等に公開しない**設定（.gitignore）が入っています。*

付録C: ファイル構成（概要）

```
（プロジェクトフォルダ）/
├── .claude/commands/          … 各機能の手順書（AIが読む。カスタマイズはここ）
│   ├── seasonal-articles.md
│   ├── youtube-import.md
│   ├── faq-articles.md
│   ├── faq-eyecatch-backfill.md
│   ├── column-eyecatch-backfill.md
│   └── faq-content-improve.md
├── scripts/                   … データ取得・投稿のプログラム（依存ゼロ）
│   ├── ga-fetch.mjs           … GA4データ取得
│   ├── gsc-fetch.mjs          … Search Consoleデータ取得
│   ├── youtube-fetch.mjs      … YouTube新着取得（本文素材込み）
│   ├── faq-fetch.mjs          … 既存FAQ・タグ一覧（重複判定用）
│   ├── missing-eyecatch.mjs   … アイキャッチ未設定の記事一覧
│   ├── wp-post.mjs            … 下書き作成（type=post/faq）
│   ├── update-post.mjs        … 既存記事の更新（本文/タイトル/タグ）
│   ├── eyecatch.mjs           … アイキャッチ生成（AI背景＋文字合成＋設定）
│   └── lib/…                  … 共通部品（認証・整形など）
```

```
├── assets/fonts/            … アイキャッチ用の日本語フォント（同梱）
├── data/                    … 作成済みテーマ・処理済み動画の記録
├── docs/                    … 手順書・仕様書（この文書など）
├── .env / .env.example      … 秘密の鍵（.env は非公開）
├── SETUP.md                 … 初回セットアップ手順（画面つき）
└── README.md                … 概要
```

この文書について

- 特定のサイトに依存しない**汎用版**です。実際に使う際は、.env と .claude/commands/ を自社向けに設定してください。
- 不明点は、この文書ごと Claude Code に読ませて質問すると、あなたの環境に合わせた案内が受けられます。