

アクイア認定デベロッパー - Drupal 9 学習ガイド

本資料は「アクイア認定デベロッパー - Drupal 9」試験の学習ガイドです。このガイドでは、試験の構成とトピックの解説、試験で出題される問題の性質を理解することに役立ちます。

1. [試験概要](#)
 2. [試験取得までの道のり](#)
 3. [試験ドメイン](#)
 4. [自己評価](#)
 5. [試験の準備](#)
 6. [リソース集](#)
-

1. 試験概要

試験名 : アクイア認定デベロッパー - Drupal 9
試験時間 : 90分
受験料 : 250ドル
問題数 : 60問
合格ライン : 65%

アクイア認定デベロッパーは、基本的なウェブ概念、サイト構築、フロントエンド開発（テーマ作成）およびバックエンド開発（コーディング）の分野において、Drupal開発者としてのスキルおよび知識の検証を目的としています。この試験はDrupal 9をベースとしています。

この試験は、受験者の次の能力を検証します。

- Drupal 9をベースとしたデザイン、開発、デプロイ
- フロントエンド、バックエンド開発に関するDrupalコアの基本的なアーキテクチャに関するベストプラクティスの理解
- Drupal 9のフロントエンド、バックエンド開発を用いたアプリケーションの開発と保守
- 新規のDrupal 9モジュールとテーマの開発、実装
- 既存モジュール、テーマのカスタマイズと拡張

このレベルで必要となる基本的な知識およびスキルには、次のエリアおよび対象コンポーネントがすべて含まれます。

- Drupalテクノロジーを用いた専門的な経験
- Drupal サイトのセットアップと構成
- Drupalモジュールの開発、実装
- 新規テーマの構築や既存テーマのカスタマイズ：既存テーマ、HTML、CSS、JavaScript/JQueryに関する知識
- 既存モジュールのカスタマイズと拡張：既存モジュール、PHPに関する知識

2. 認定取得までの道のり

Drupal入門コースを受講する

試験を受験予定の全ての方は、Acquia AcademyのDrupal入門コースに[無料でアクセス](#)できます。Drupal開発者向けに次のコースを提供しています。Acquia Academyにログインしてご利用ください。

- [Drupal 9 Site Building](#)
- [Drupal Layout/Theming Training](#)
- [Drupal 9 Module Development](#)

コースを受講した後は、コースで学んだ概念を補強するために、自分で小さなウェブサイトを構築してみましょう。

実務経験を積む

コースを修了したからといって、認定試験を受ける準備ができているとは限りません。試験を受験するためには、いくつかの実務経験が必要です。少なくとも2~3のプロジェクトに取り組み、実務経験を積むことをお勧めします。

自己評価を行う

試験を受けることを決めたら、試験でカバーされている領域とトピック、およびそれらの相対的な重みを勉強することをお勧めします。また、試験の構成でカバーされているドメイン/トピックに対して、自分の得意分野と不得意分野の[自己評価](#)を行う必要があります。これは経験豊富なDrupal開発者にお勧めの出発点です。

学習ガイドを活用する

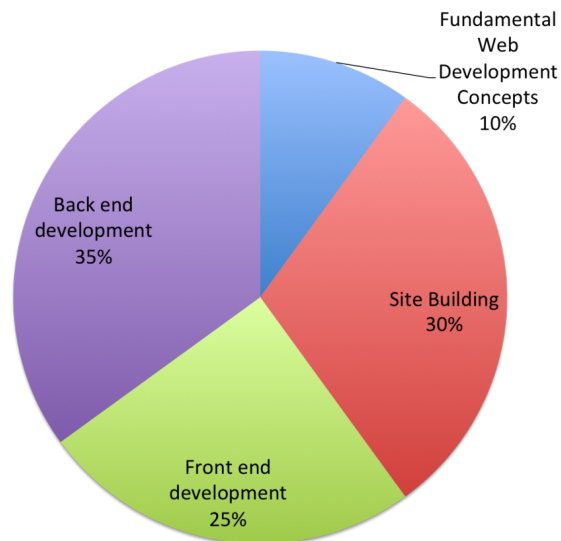
自己評価に基づいて、いくつかのドメインやトピックの知識や経験を得る必要があるかもしれません。Drupalを学ぶ最善の方法は、実際に作業したり、実験したりすることです。本ガイドの[試験の準備](#)や[リソース集](#)が参考になります。

試験を受ける

この段階で、あなたは試験を受ける準備ができているはずですが、[アクイア認定 受験方法](#)のページにアクセスして受験方法を確認します。

3. 試験ドメイン

ドメイン	配分
1.0 基本的なウェブ開発概念	10%
2.0 サイト構築	30%
3.0 フロントエンド開発 (テーマ作成)	25%
4.0 バックエンド開発 (コーディング)	35%
合計	100%



4. 自己評価

試験のドメインとトピックに関して、スキルと専門知識の自己評価を行うことをお勧めします。各トピックに対して、あなたのスキルを0～10段階でランク付けします。知識や経験が全くない場合は0を、そのトピックを完全にマスターしている場合は10とします。

ドメイン	トピック	スコア
1	基本的なウェブ開発概念	
1.1	HTMLおよびCSSに関する知識を実証	
1.2	JavaScriptおよびjQueryプログラミング概念を識別	
1.3	バージョン管理におけるGitの使用方法を実証	
2	サイト構築	
2.1	基本的なデータ構成の構築における適切なフィールドおよびフィールド設定によるエンティティタイプを作成し構成する能力を実証	
2.2	コアエンティティのカスタムフォームおよびビューモードを構築するためのディスプレイモードの設定能力を実証	
2.3	コンテンツの分類と整理のためのタクソノミーボキャブラリーおよびタームを作成し、使用する能力を実証	
2.4	ブロックタイプの設定、ブロックライブラリの管理、ブロックレイアウトの設定を行う能力を実証	
2.5	メニューを使用して、メインナビゲーションシステムと代替ナビゲーションシステムを構築する能力を実証	
2.6	コンテンツリストページ、ブロック、フィードを構築するためのビューの作成と設定の能力を実証	
2.7	サイト構成をエクスポートするための構成管理機能を使用する能力を実証	
2.8	コア多言語対応機能を用いた多言語対応ウェブサイトの構築能力を実証	
3	フロントエンド開発 (テーマ作成)	
3.1	カスタムテーマまたはサブテーマの作成能力の実証	
3.2	テーマ作成の概念に関する知識の実証	
3.3	Twig構文を使用する能力の実証	

3.4	レイアウトコンテンツ定義のためのTwigテンプレートの構築またはオーバーライドする能力の実証	
3.5	カスタム出力をオーバーライドするためのテンプレート前処理関数の記述能力を実証	
4	バックエンド開発 (コーディング)	
4.1	コアまたはオブジェクト指向型PHPを用いたコーディング能力を実証	
4.2	Drupal機能拡張のためのDrupal APIを使用したカスタムモジュール開発能力を実証	
4.3	コードを用いたデータの保存、検索能力を実証	
4.4	その他必要なAPIの対応能力を実証	
4.5	Drupalコーディング規約を用いたコーディング能力を実証	
4.6	サイト構成またはカスタムコードにより発生するサイトパフォーマンスの問題の分析、解決能力を実証	
4.7	サイト構成またはカスタムコードにより発生するセキュリティの問題の分析、解決能力を実証	
4.8	コアのテストフレームワークを用いたテスト記述能力を実証	

5. 試験の準備

準備が必要なトピックのリストができたら、次の2つの課題に直面することになります。

1. トピックのための文書や知識のリソースを探して勉強する
2. 学ぶべきトピックを網羅した十分な実践的なケーススタディを見つける

知識リソース

入門的なDrupalトレーニングを、ドキュメントとオンラインリソースで補うのは良い考え方です。本ガイドの[リソース集](#)に、各学習領域のオンラインリソースとドキュメントのリストがあります。

実践的なケーススタディ

2つ目の課題は、トピックをカバーする十分な実用的な事例を見つけることです。ここでは、準備のための事例を見つけるためのいくつかのアイデアと提案を紹介します。

サイト構築

あなたやあなたのチームが過去1年間に取り組んだDrupalのプロジェクトをいくつか（3～5個）レビューしてみてください。サイトで使用されているコンテンツタイプ、フィールド設定、表示設定、ボキャブラリー、ブロック、メニュー、Viewsを確認してください。

プロジェクトについて、以下を考えてみましょう。

- なぜそのような構成になっているのか？
- 何を変更したり、改善することができますか？
- クライアントによって提起された問題やサポート要求は何ですか？
- 初期設定にどのような変更を加えたのか、またその理由は何ですか？

フロントエンド開発

あなたのチームが過去1年間に開発したカスタムテーマを見つけてください。可能であれば、それらがベースになっているソースデザインやワイヤーフレームを入手してください。テンプレートファイルとTwigの構文、CSSとJSファイルのレビュー、テーマに実装されている前処理機能を確認してください。

- テーマはデザイン通りに実装されていますか？そうでない場合、どのような変更が必要だったのか、その理由は何ですか？
- テーマリージョンは最適ですか？冗長性はありますか？
- テーマの設定はありますか？
- テーマ内のCSSとJavaScriptのファイルはどのように呼び出されますか？
- 前処理関数は使用されていますか？それはなぜですか？

もしあなたのチームが開発したテーマにアクセスできない場合は、公開されているテーマのソースコードを調べてみるのも良いでしょう。自分でテーマを作成すれば、さらに知識が磨かれます。

バックエンド開発

バックエンド開発やモジュール開発に携わったことがない場合、この試験の準備は困難な作業になる可能性があります。非常に複雑なモジュールや、要件やユースケースに固有である可能性があるため、あなたのチームが開発したモジュールのコードを調査することは役に立つ場合と役に立たない場合があります。この試験では、DrupalのAPI、コアプラグインシステム、およびそれらのアプリケーションを総合的に理解する必要があります。

このドメインの準備のために、モジュール開発の有償トレーニングに参加したり、シンプルなモジュールを自作することをお勧めします。

独学で学習している場合は、モジュールのサンプルコードが詰まっている[Examples for Developersモジュール](#)も参考になると思います。このモジュールでは、4.1項と4.2項でカバーされているDrupal APIやコンセプトについて、高度にドキュメント化された作業コードのスニペットを提供しています。

6. リソース集

Drupal開発のために知っておくべきツール

Drupalコーディング規約

Drupalコミュニティで適用されている[コーディングのベストプラクティスと規約の概要](#)です。

これらのコーディング規約は、Drupalの開発中にコードが生成される可能性のあるすべての領域について詳細に記述されており、コミュニティで広く確立されたベストプラクティスを表しています。これらのベストプラクティスに従うことで、開発がよりスムーズに進み、Drupalコアやモジュールへの貢献がより容易に受け入れられるようになります。アクイアのテクニカルサポートを利用してプロジェクトに取り組んでいる場合、これらはプルリクエストが遵守しなければならないコーディング基準です。

Drush

[Drush](#)はDrupalサイト開発の基本的なスキルと考えられています。

トピック別 参照資料

1.0 基本的なウェブ開発概念

- [HTML、CSS、JS、PHPの学習教材](#)
- [Drupalでのライブラリ操作 \(Javascript、Stylesheets\)](#)
- [Drupal Javascript API](#)
- [Gitチュートリアル](#)
- [Gitの一般的なワークフロー](#)

2.0 サイト構築

- [Drupalユーザーガイド](#)
- [コンテンツタイプ](#)
- [タクソノミー](#)
- [ブロック](#)
- [メニュー](#)
- [Views](#)

3.0 フロントエンド開発 (テーマ作成)

- [Drupalのテーマ開発](#)

4.0 バックエンド開発 (コーディング)

- [カスタムモジュールの作成](#)
 - [オブジェクト指向プログラミング](#)
 - [名前空間](#)
 - [サービスと依存性注入](#)
 - [Drupal API](#)
 - [Drupalコーディング規約](#)
 - [セキュアなコード](#)
-

本ガイドは以上になります。