

# アクイア認定デベロッパー - Drupal 10 学習ガイド

本資料は「アクイア認定デベロッパー - Drupal 10」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 10  
**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 10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 10をベースとした設計、開発、デプロイ
- フロントエンド、バックエンド開発におけるDrupalコアの基本的なアーキテクチャのベストプラクティスの理解
- Drupal 10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 10モジュールとテーマの開発と実装
- 既存のモジュールと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
- 新規のDrupalモジュールの開発と実装
- 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サイトビルダー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 Site Building](#)
- [Drupal Layout/Theming Training](#)
- [Drupal 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サイトビルダーにお勧めの出発点です。

### 学習ガイドを活用する

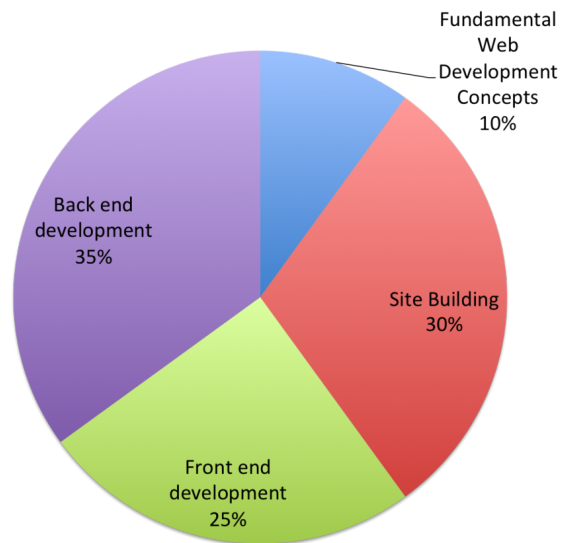
自己評価に基づいて、いくつかのドメインやトピックの知識や経験を得る必要があるかもしれません。Drupalを学ぶ最善の方法は、実際に作業したり、実験したりすることです。本ガイドの[試験の準備](#)や[リソース集](#)が参考になります。

### 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。

トピック	スコア
<b>1 基本的なウェブ開発概念</b>	
1.1 HTML、CSSに関する知識の実証	
1.2 JavaScript、jQueryのプログラミング概念の把握	
1.3 バージョン管理におけるGitの使用方法の実証	
<b>2 サイト構築</b>	
2.1 基本的なデータ構造の構築における、適切なフィールドとフィールド設定を持つ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、ビューモードを構築するためのディスプレイモードを設定する能力を実証	
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と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、メインおよび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 ンおよび代替ナビゲーションシステムを構築する能力を実証 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Viewsを作成し構成する能力を実証	
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 コア多言語機能を用いた多言語ウェブサイトを構築する能力を実証	
<b>3 フロントエンド開発 (テーマ作成)</b>	
3.1 カスタムテーマやサブテーマを作成する能力の実証	
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を定義するためのTwigテンプレートの構築や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プリプロセス関数の実装能力を実証	

<b>4 バックエンド開発 (コーディング)</b>	
4.1 コアとオブジェクト指向PHPを用いたコーディング能力の実証	
4.2 Drupal機能を拡張するためのDrupal APIを用いたカスタムモジュールの開発能力を実証	
4.3 コードを用いてデータを保存、取得する能力を実証	
4.4 他の主要なAPIを操作する能力を実証	
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 サイト構成やカスタムコードに起因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 サイト構成やカスタムコードに起因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実装能力を実証	

## 5. 試験の準備

準備が必要なトピックのリストができれば、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

Acquia Academyから、関連するトピックのコースを受講して知識をリフレッシュできます。また、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プリプロセス関数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準拠しなければなりません。

#### 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

トピック	リンク	参照
HTML、CSS、JS、PHPの学習リソース	<a href="#">Web technology for developers</a>	MDN Web Docs
Drupalでのライブラリ操作 (Javascript、Stylesheets)	<a href="#">Adding assets (CSS, JS) to a Drupal theme via *.libraries.yml</a>	Drupal.org
Drupal Javascript API	<a href="#">JavaScript API overview</a>	Drupal.org
Gitチュートリアル	<a href="#">Git Tutorials and Training</a>	Atlassian Git Tutorial
Gitの一般的なワークフロー	<a href="#">Comparing Git Workflows: What You Should Know</a>	Atlassian Git Tutorial

#### 2.0 サイト構築

トピック	リンク	参照
Drupalユーザーガイド	<a href="#">Drupal User Guide</a>	Drupal.org
コンテンツタイプ	<a href="#">Chapter 6. Setting Up Content Structure</a>	Drupal.org
タクソノミー	<a href="#">6.5. Concept: Taxonomy</a>	Drupal.org

ブロックの操作	<a href="#">Chapter 8. Blocks</a>	Drupal.org
メニューの操作	<a href="#">5.5. Concept: Menu</a>	Drupal.org
Viewsの操作	<a href="#">Chapter 9. Creating Listings with Views</a>	Drupal.org

### 3.0 フロントエンド開発 (テーマ作成)

トピック	リンク	参照
Drupalのテーマ開発	<a href="#">Theming Drupal</a>	Drupal.org

### 4.0 バックエンド開発 (コーディング)

トピック	リンク	参照
カスタムモジュールの作成	<a href="#">Creating modules</a>	Drupal.org
オブジェクト指向プログラミング	<a href="#">Getting Started - Background &amp; Prerequisites</a>	Drupal.org
名前空間	<a href="#">Namespaces</a>	Drupal.org
サービスと依存性注入	<a href="#">Services and dependency injection in Drupal 8+</a>	Drupal.org
Drupal API	<a href="#">Drupal APIs</a>	Drupal.org
Drupalコーディング規約	<a href="#">Coding standards</a>	Drupal.org
セキュアなコードの記述	<a href="#">Writing secure code for Drupal</a>	Drupal.org

本ガイドは以上になります。