

アクイア認定 フロントエンドスペシャリスト - Drupal 9 学習ガイド

本資料は「アクイア認定フロントエンドスペシャリスト - 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 : アクイア認定フロントエンドスペシャリスト - Drupal 9
試験時間 : 90分
受験料 : 300ドル
問題数 : 60問
合格ライン : 68%

アクイア認定フロントエンドスペシャリストは、フロントエンド開発（テーマ作成）の分野におけるDrupal開発者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。

- フロントエンドに焦点を当てたDrupal 9ベースのソリューションの設計、開発、デプロイ
- サイト構築、テーマ開発におけるコアDrupal 9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
- 新規のDrupal 9テーマの開発、実装
- 既存のDrupal 9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

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

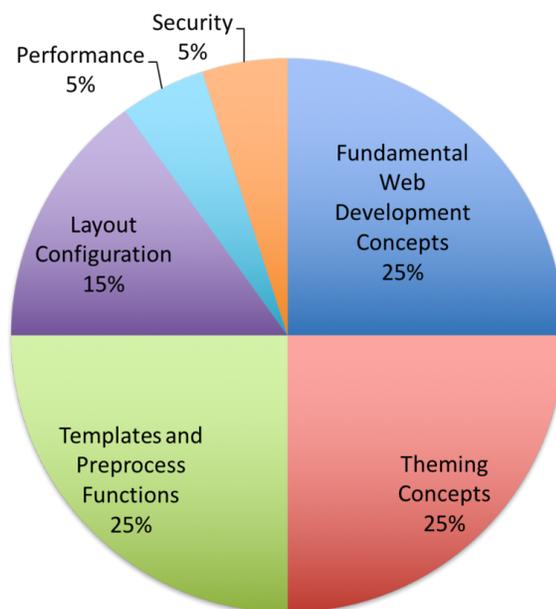
自己評価に基づいて、いくつかのドメインやトピックの知識や経験を得る必要があるかもしれません。Drupalを学ぶ最善の方法は、実際に作業したり、実験したりすることです。本ガイドの[試験の準備](#)や[リソース集](#)が参考になります。

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	25%
2.0 テーマ作成の概念	25%
3.0 テンプレートとプリプロセス関数	25%
4.0 レイアウト構成	15%
5.0 パフォーマンス	5%
6.0 セキュリティ	5%
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	PHPプログラミングの概念の理解	
1.3	JavaScriptおよびjQueryプログラミング概念を識別	
1.4	レスポンシブデザインの概念に関する知識を実証	
2	テーマ作成の概念	
2.1	テーマ内のカスタムリージョンの定義および使用能力を実証	
2.2	テーマ構成を使用する能力を実証	
2.3	スタイルシートの対応に関する知識を実証	
2.4	テーマでのJavaScriptの対応に関する知識を実証	
2.5	テーマでのブレイクポイントの対応に関する知識を実証	
2.6	ベーステーマからのサブテーマを構築する能力を実証	
3	テンプレートとプリプロセス関数	
3.1	Twig構文を使用する能力を実証	
3.2	マークアップ管理に関するコアテンプレートの構築およびカスタマイズ能力を実証	
3.3	テンプレートのsuggestionsに対応する能力を実証	
3.4	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	
3.5	Form AlterやTemplate Suggestions Alterを使用する能力を実証	
4	レイアウト構成	
4.1	レイアウト構築のためのブロックの作成および構成能力を実証	

4.2	コンテンツ一覧ページ、ブロック、フィードを構築するためのViewsの作成および構成能力を実証	
4.3	Layout Builderモジュールを使用してレイアウトを作成する能力を実証	
5	パフォーマンス	
5.1	サイト構成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	
5.2	カスタムテーマ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	
6	セキュリティ	
6.1	サイト構成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	
6.2	カスタムテーマ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	

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

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、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？
- どのテンプレート、テンプレートのsuggestionが使用されていますか？
- 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？

- どのブレイクポイントが使用されていますか？
- サブテーマを使用する場合、ベーステーマのどのプロパティが継承され、どのプロパティがサブテーマで一意になりますか？
- どんなTwig関数が使用されていますか？
- エンティティとフォームにカスタム表示モードと表示モードは定義されていますか？それらはどこでどのように使用されていますか？
- フロントエンドのパフォーマンスに関する問題はありますか？それらはどのように扱われましたか？

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。

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

1.1 HTMLおよびCSSに関する知識を実証—フロントエンドスペシャリストは、標準に準拠したHTML、CSSを記述する豊富な経験が必要です。また、HTML5とCSS3の仕様に精通し、最新の標準と実践を身につけている必要があります。

- [HTML5開発者ガイド](#)
- [CSS \(Mozilla Developer Network\)](#)

1.2 PHPプログラミングの概念の理解—PHPの基本的な構文、一般的なPHP関数、変数の取り扱いなどのPHP構文に関する知識が必要です。

- [PHPマニュアル](#)

1.3 JavaScriptおよびjQueryプログラミングの概念の理解—JavaScriptとjQueryを使った独自のUIコンポーネントの作成経験が豊富である必要があります。また、最新のベストプラクティスに従い、セキュリティやパフォーマンスに関する一般的な問題にも精通している必要があります。

- [JavaScriptドキュメント](#)
- [jQueryドキュメント](#)

1.4 レスポンシブデザインの概念に関する知識を実証—レスポンシブデザインの原則を理解する必要があります。Drupal 9はレスポンシブなモバイルファーストのアプローチを採用しています。

- [メディアクエリ：幅とデバイス幅](#)
- [レスポンシブデザイン](#)

2.0 テーマ作成の概念

2.1 テーマ内のカスタムリージョンの定義および使用能力を実証—カスタムリージョンを定義し、適切なテンプレートでそれらのリージョンを出力し、リージョンを出力する際に条件付きで表示させるロジックを作成した経験が求められます。

- [テーマにリージョンを追加する](#)
- [page.html.twig \(API\)](#)

2.2 テーマ構成を使用する能力を実証—テーマ設定に関連する管理設定に精通している必要があります。カスタムコードを使用して独自のテーマ設定を作成できる必要があります。

- [高度なテーマ設定の作成](#)

2.3 スタイルシートの対応に関する知識を実証—CSSとDrupalのベストプラクティスに沿った記述と整理方法でCSSを作成する必要があります。独自のスタイルシートをテーマに追加する方法、Drupalがスタイルシートを集約する方法、Drupalがスタイルシートを読み込む順番、スタイルシートを条件付きで読み込む方法を理解する必要があります。

- [CSS](#)
- [CSSとJavaScript](#)

2.4 テーマでのJavaScriptの対応に関する知識を実証—独自のJavaScriptとjQueryを記述した経験を持っていることが求められます。DrupalのJavaScript APIを使用し、DrupalのJavaScriptコーディング規約に精通する必要があります。またJavaScriptを条件付きで読み込む方法を理解する必要があります。

- [CSSとJavaScript](#)
- [JavaScript API](#)

2.5 テーマでのブレイクポイントの対応に関する知識を実証—サイト全体で利用できる共通のブレイクポイントを定義し、使用方法を理解する必要があります。またCSSメディアクエリに精通し、さまざまなデバイスをターゲットにしたメディアクエリの構築に精通している必要があります。

- [Drupalでのブレイクポイントの操作](#)
- [メディアクエリの使用](#)

2.6 ベーステーマからのサブテーマを構築する能力を実証—様々なコアテーマとコントリビュートテーマのベーステーマの長所と短所を理解する必要があります。プロジェクトに適したベーステーマを採用し、ベーステーマのアセットを利用し、上書きし、拡張して適切にサブテーマ化する方法を理解する必要があります。

- [サブテーマの作成](#)

3.0 テンプレートとプリプロセス関数

3.1 Twig構文を使用する能力を実証—関数やフィルタ、Drupal固有の関数の使用方法、変数の取り扱いなど、基本的～高度なTwig構文に関する知識が必要です。

- [Twig公式ドキュメント](#)
- [DrupalのTwig](#)
- [Twigテンプレートの関数](#)
- [テンプレートデザイナーのためのTwig](#)
- [フィルタ - Twigテンプレート変数の変更](#)

3.2 マークアップ管理に関するコアテンプレートの構築およびカスタマイズ能力を実証—プロジェクトに必要なマークアップを出力するためのテンプレートの使い方、名前の付け方、作成方法に関する知識が必要です。

- [Twigテンプレートの操作](#)
- [Twigテンプレートの命名規則](#)
- [高度なテーマ設定の作成](#)

3.3 テンプレートのsuggestionsに対応する能力を実証—テンプレートsuggestionsを処理する、独自のsuggestionを追加する、または要素に対して特定のテンプレートsuggestionを定義するなど、Drupalによってより良い方法が提供されています。

- [Twigテンプレートのデバッグ](#)

3.4 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—前処理関数を使用してテンプレートに必要なデータを上書きしたり、改善したり、追加データを取得するなどの方法を理解する必要があります。

- [hook_preprocess_HOOK 関数](#)

4.0 レイアウト構成

4.1 レイアウト構築のためのブロックの作成および構成能力を実証—DrupalのUIやカスタムコードでブロックを作成・管理することに精通する必要があります。Drupalに組み込まれているブロックシステムを利用して柔軟なレイアウトを作成するためのベストプラクティスがあります。

- [Drupalユーザーガイド：Blocks](#)

4.2 コンテンツ一覧ページ、ブロック、フィードを構築するためのViewsの作成および構成能力を実証—Viewsを使用したコンテンツの表示に精通する必要があります。Viewsの表示方法の種類やオプション、それぞれの長所と短所について熟知する必要があります。

- [Drupalユーザーガイド：Views](#)

4.3 Layout Builderモジュールを使用してレイアウトを作成する能力を実証—DrupalコアのLayout Builderモジュールを利用して、コンテンツタイプごと、またはノードごとにレイアウトを設定する方法に熟知する必要があります。

- [Drupalユーザーガイド：Layout Builder](#)

5.0 パフォーマンス

5.2 カスタムテーマ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—このトピックでは、様々なテーマを作成する際のパフォーマンスに関する一般的な落とし穴、PHP/JavaScript/CSSの効率的なオプション、Drupalのテーマで遅いコードをデバッグするためのテクニックについてがカバーされます。

- [フロントエンドパフォーマンスガイド](#)

6.0 セキュリティ

6.2 カスタムテーマ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—このトピックにはテーマレイヤにおける適切なセキュリティにまつわるベストプラクティスが含まれます。PHPとTwigの処理におけるデータのサニタイズ技術も含まれます。レンダリングサイクルの各レイヤにおけるセキュリティプロトコルの知識が必要です。

- [Drupalのセキュリティ](#)

本ガイドは以上になります。