

## パフォーマンス

推定値のため変動する可能性があります。[パフォーマンス スコアの計算](#)  
は、これらの指標を基に行っています。[計算ツールはこちら。](#)

▲ 0-49

50-89

90-100



指標

ビューを開く

First Contentful Paint

1.1 秒

Largest Contentful Paint

1.1 秒

Total Blocking Time

Cumulative Layout Shift

0 ミリ秒

0

Speed Index

1.1 秒



今年後半に、パフォーマンスの監査はインサイトに置き換えられます。[詳細およびフィード](#)

[監査に戻る](#)

バックについては、こちらにアクセスしてください。

次に関連する監査を表示: All [FCP](#) [LCP](#) [TBT](#)

インサイト

### ▲ レンダリングをブロックしているリクエスト — 推定される削減時間 150 ミリ秒 ^

リクエストがページの最初のレンダリングをブロックしているため、LCP が遅れる可能性があります。これらのネットワークリクエストを [遅らせるかオンライン化](#)すると、クリティカルパスから移動できます。 [FCP](#) [LCP](#)

URL	転送サイズ	継続時間
localhost <a href="#">自社</a>	0.0 KiB	150 ms
/_nuxt/entry.CN-Sic9a.css (localhost)	0.0 KiB	150 ms

### ▲ ネットワークの依存関係ツリー ^

チェーンの長さを縮小する、リソースのダウンロードサイズを抑える、不要なリソースのダウンロードを遅らせるなどしてページの読み込み速度を改善し、[クリティカルリクエストチェーンを回避](#)してください。 [LCP](#)

クリティカルパスの最大待ち時間: **50 ms**

最初の移動先

[http://localhost:3000 - 16 ms, 15.47 KiB](http://localhost:3000)

[/\\_nuxt/entry.CN-Sic9a.css \(localhost\) - 50 ms, 0.00 KiB](#)

[事前接続したオリジン](#)

[事前接続](#)のヒントは、ブラウザがページの読み込みの早い段階で接続を確立できるようにして、そのオリジンに対する最初のリクエストの実行時間を短縮します。以下は、ページから事前接続したオリジンです。

事前接続したオリジンはありません

### 事前接続の候補

最も重要なオリジンに[事前接続](#)のヒントを追加します。ヒントを追加するオリジンは 4 つ以下にすることをおすすめします。

事前接続の候補となるオリジンは他にありません

### ドキュメント リクエストのレイテンシ — 推定される削減サイズ 10 KiB

▲

最初のネットワーク リクエストは最も重要です。リダイレクトを回避し、サーバー応答を高速に保ち、テキスト圧縮を有効にして、レイテンシを削減します。 [FCP](#) [LCP](#)

リダイレクトなし

サーバーの応答が速い（観測値 3 ms）

圧縮の適用なし

#### ○ LCP の内訳

▲

各[サブパート](#)には、それに適した改善戦略があります。理想的なのは、LCP 時間のほとんどがリソースの読み込みに使われ、遅延に費やされることです。 [LCP](#)

### サブパート

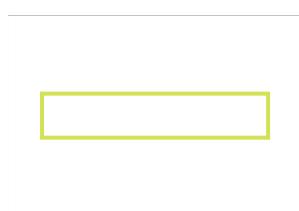
### 継続時間

Time to First Byte

0 ms

要素のレンダリングの遅延

90 ms



これらのインサイトは、Chrome DevTools のパフォーマンス パネルでも確認できます。詳しい情報を表示するには、[トレースを記録](#)してください。

### 診断

○ メインスレッドでタスクが長時間実行されないようにしてください — 長時間実行されているタスクが1件見つかりました



メインスレッドで長時間実行されているタスクを表示します。入力の遅延に最も影響しているタスクを特定する際に役立ちます。[長いメインスレッド タスクの回避方法の詳細](#) [TBT]

URL	開始時間	継続時間
localhost <span style="border: 1px solid blue; padding: 2px;">自社</span>		64 ms
http://localhost:3000	758 ms	64 ms

アプリケーションのパフォーマンスに関する詳細。これらの数値は、パフォーマンス スコアには直接影響しません。

合格した監査 (28) 非表示

#### 効率的なキャッシュ保存期間を使用する



キャッシュの有効期間を長くすると、再訪問したユーザーへのページの読み込み速度を向上できます。[詳細](#) [FCP] [LCP]

#### レイアウト シフトの原因



レイアウト シフトは、ユーザーの操作なしで要素が移動する場合に発生します。[レイアウト シフトの原因を調査](#)してください。

[CLS]

nuxt/image コンポーネントを使用して、width と height を明示的に指定します。[詳細](#)



#### DOM サイズを最適化する

DOM サイズが大きいと、スタイルの計算とレイアウトのリフレーに時間がかかり、ページの応答性に影響する可能性があります。DOM サイズが大きいと、メモリ使用量も増加します。[過度な DOM サイズの回避方法の詳細](#)

統計情報	要素	値
合計要素数		38
子の最大数		4
DOM の深さ		9

## 統計情報

## 要素

### 重複する JavaScript

^

重複する大きい JavaScript モジュールをバンドルから削除すると、ネットワーク アクティビティで不必要に消費されるデータ量を減らすことができます。 [FCP](#) [LCP](#)

### フォント表示

^

テキストの表示を統一するため、[font-display](#) を swap または optional に設定することを検討してください。swap をさらに最適化して、[フォント指標のオーバーライド](#)でレイアウト シフトを軽減できます。

### 強制リフロー

^

強制リフローは、DOM の状態の変更によってスタイルが無効化された後に JavaScript が幾何学的プロパティ (offsetWidth など) をクエリしたときに発生します。これにより、パフォーマンスが低下する可能性があります。詳しくは、[強制リフロー](#)とその緩和策をご覧ください。

### 画像配信を改善する

^

画像のダウンロード時間を短縮すると、ページ読み込みの体感時間と LCP を改善できます。[画像サイズの最適化に関する詳細](#) [FCP](#) [LCP](#)



nuxt/image コンポーネントを使用して、format="webp" を設定します。 [詳細](#)

### ○ INP の内訳

^

最も長いサブパートから調査を開始します。[遅延を最小限に抑えることができます](#)。処理期間を削減するには、[メインストリームのコスト \(通常は JS\) を最適化](#)します。

### ○ LCP リクエストの検出

^

LCP 画像を HTML からすぐに[検出できる](#)ようにし、[遅延読み込みを回避](#)して、LCP を最適化します



nuxt/image コンポーネントを使用して、LCP 画像の preload を指定します。 [詳細](#)

### 以前の JavaScript

^

ポリフィルと変換を使用すると、従来のブラウザで新しい JavaScript 機能を使用できるようになります。ただし、その機能の多くは最新ブラウザでは必要ありません。従来のブラウザをサポートする必要がある場合を除き、[Baseline](#) の機能をトランスペイルしないように JavaScript ビルドプロセスを変更することを検討してください。[ほとんどのサイトでトランスペイルせずに ES6+ コードをデプロイできる理由](#) [FCP](#) [LCP](#)

## 最新の HTTP

^

HTTP/2 と HTTP/3 には、多重化など、HTTP/1.1 と比べて多くのメリットがあります。[最新の HTTP の使用について](#)

[FCP] [LCP]

### サードパーティ

^

サードパーティのコードによって、読み込み速度が著しく低下する可能性があります。ページのコンテンツを優先させるには、[サードパーティのコードの読み込みを減らして遅らせます。](#)

### ビューポートをモバイル向けに最適化する

^

ビューポートがモバイル向けに最適化されていない場合、タップ操作が[最大 300 ミリ秒遅延](#)する可能性があります。

## meta

^

### オフスクリーン画像の遅延読み込み

^

オフスクリーンの非表示の画像は、重要なリソースをすべて読み込んだ後に遅れて読み込むようにして、操作可能になるまでの時間を短縮することをご検討ください。[オフスクリーンの画像を保留する方法の詳細](#) [FCP] [LCP]



nuxt/image コンポーネントを使用して、画面外の画像の `loading="lazy"` を設定します。[詳細](#)

^

### CSS の最小化

^

CSS ファイルを最小化すると、ネットワーク ペイロードのサイズを抑えることができます。[CSS を圧縮する方法の詳細](#)

[FCP] [LCP]

^

### JavaScript の最小化

^

JavaScript ファイルを最小化すると、ペイロード サイズとスクリプトの解析時間を抑えることができます。[JavaScript を圧縮する方法の詳細](#) [FCP] [LCP]

^

### 使用していない CSS の削減

^

スタイルシートから使用していないルールを削除して、スクロールせずに見える範囲のコンテンツで使用されていない CSS の読み込みを遅らせると、ネットワークの通信量を減らすことができます。[未使用的 CSS を削減する方法の詳細](#) [FCP]  
[LCP]

^

### 使用していない JavaScript の削減

^

使用していない JavaScript を削除して、必要になるまでスクリプトの読み込みを遅らせると、ネットワークの通信量を減らすことができます。[使用していない JavaScript を削除する方法の詳細](#) [FCP] [LCP]

## HTTP/2 を使用してください ^

HTTP/2 には、バイナリ ヘッダー、多重化など、HTTP/1.1 と比べて多くのメリットがあります。 [HTTP/2 の詳細](#) [LCP] [FCP]

## 最新ブラウザに従来の JavaScript を配信しないようにしてください ^

ポリフィルと変換を使用すると、従来のブラウザで新しい JavaScript 機能を使用できるようになります。ただし、その機能の多くは最新ブラウザでは必要ありません。従来のブラウザをサポートする必要がある場合を除き、[Baseline](#) の機能をトランスペイパイルしないように JavaScript ビルドプロセスを変更することを検討してください。[ほとんどのサイトでトランスペイパイルせずに ES6+ コードをデプロイできる理由](#) [FCP] [LCP]

## 過大なネットワーク ペイロードの回避 — 合計サイズは 21 KiB でした ^

ネットワーク ペイロードのサイズが大きいと、ユーザーの金銭的負担が大きくなり、多くの場合、読み込み時間が長くなります。 [ペイロード サイズを小さくする方法の詳細](#)

URL	転送サイズ
localhost <a href="#">自社</a>	21.0 KiB
http://localhost:3000	15.5 KiB
/favicon.ico (localhost)	4.5 KiB
...meta/14ade20e-ed05-4fd2-9413-9f5cf8b0ab30.json (localhost)	0.6 KiB
/_nuxt/xC2XOs-m.js (localhost)	0.5 KiB

## ○ カスタム速度の記録と計測 ^

User Timing API を使用してアプリを計測可能にし、主要なユーザー エクスペリエンスでのアプリの実際のパフォーマンスを測定できるようにしてください。 [カスタム速度マークの詳細](#)

## JavaScript の実行にかかる時間 — 0.0 秒 ^

JavaScript の解析、コンパイル、実行にかかる時間の短縮をご検討ください。JavaScript ペイロードのサイズを抑えるなどの方法があります。 [JavaScript の実行時間を短縮する方法の詳細](#) [TBT]

URL	合計 CPU 時間	スクリプトの評価	スクリプトの解析
localhost <a href="#">自社</a>	185 ms	10 ms	1 ms

URL	合計 CPU 時間	スクリプトの評価	スクリプトの解析
http://localhost:3000	185 ms	10 ms	1 ms
アトリビューション不可	51 ms	5 ms	0 ms
Unattributable	51 ms	5 ms	0 ms

## メインスレッド処理の最小化 — 0.2 秒

^

JavaScript の解析、コンパイル、実行にかかる時間を短縮することをご検討ください。JavaScript ペイロードのサイズを抑えるなどの方法があります。[メインスレッドでの作業を最小限に抑える方法の詳細](#) [TBT]

カテゴリ	かかった時間
Style & Layout	123 ms
Other	89 ms
Script Evaluation	16 ms
Rendering	6 ms
Parse HTML & CSS	4 ms
Script Parsing & Compilation	1 ms

## ○ ファサードでのサードパーティリソースの遅延読み込み

^

一部のサードパーティ埋め込みは遅延読み込みできます。必要になるまで、ファサードと置き換えることをご検討ください。[ファサードでサードパーティを保留にする方法の詳細](#) [TBT]

## スクロールパフォーマンスを高める受動的リスナーが使用されています

^

ページのスクロールパフォーマンスを高めるには、touch および wheel イベントリスナーを passive として指定することをご検討ください。[パッシブイベントリスナーの採用についての詳細](#)

## document.write() は使用されていません

^

接続速度が遅い環境のユーザーの場合、document.write() で動的に挿入される外部スクリプトによってページの読み込みが数十秒遅れる可能性があります。[document.write\(\) を回避する方法の詳細](#)

## ページでバックフォワードキャッシュの復元は妨げられませんでした

^

多くのナビゲーションは、前のページに戻るか、次に進むことで実行されます。バックフォワード キャッシュ (bfcache) を使用すると、こうした戻るナビゲーションを高速化できます。[bfcacheについて](#)



## ユーザー補助

これらのチェックにより、[ウェブアプリのユーザー補助機能の改善点](#)が明確になります。ただし、自動的に検出できる問題は一部に過ぎず、ウェブアプリのユーザー補助機能に問題がないことが保証されたわけではないため、[手動テスト](#)も実施することをおすすめします。

### おすすめの方法

▲ [user-scalable="no"] が `<meta name="viewport">` 要素で使用されているか、[maximum-scale] 属性が 5 未満に指定されています。



ズーム機能を無効にすると、画面の拡大操作を利用する視力の弱いユーザーがウェブページのコンテンツを確認できなくなります。[ビューポート メタタグの詳細](#)

#### 問題のある要素

meta

ここで、一般的なユーザー補助機能のおすすめの方法を確認できます。

### コントラスト

▲ 背景色と前景色には十分なコントラスト比がありません



低コントラストのテキストを使用すると、多くのユーザーは読むことが困難または不可能になります。[十分なカラー コントラストを確保する方法の詳細](#)

#### 問題のある要素

span



### 問題のある要素



div.container



span



div.container

ここには、コンテンツの読みやすさに関する改善点が表示されます。

### 多言語対応とローカライズ

#### ▲ `<html>` 要素に `[lang]` 属性が指定されていません

^

ページで `lang` 属性が指定されていない場合、スクリーン リーダーは、スクリーン リーダーの設定時にユーザーが選択したデフォルト言語がページで使用されているものと見なします。そのページでデフォルト言語が実際には使用されていない場合、スクリーン リーダーはページのテキストを正しく読み上げられない可能性があります。[lang 属性の詳細](#)

### 問題のある要素



html

ここには、地域ユーザー別のコンテンツの解釈に関する改善点が表示されます。

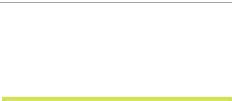
### 名前とラベル

#### ▲ フォームの要素にラベルが関連付けられていません

^

ラベルを使用すると、フォームの各コントロールが支援技術（スクリーン リーダーなど）によって正しく読み上げられるようになります。[フォーム要素のラベルの詳細](#)

### 問題のある要素



input.volume-slider

## 問題のある要素

ここには、アプリ内のコントロールのセマンティクスに関する改善点が表示されます。修正すると、支援技術（スクリーン リーダーなど）の利便性が向上する可能性があります。

### 個別の検証が必要な他の項目 (10)

非表示

#### ○ Interactive controls are keyboard focusable

^

Custom interactive controls are keyboard focusable and display a focus indicator. [Learn how to make custom controls focusable](#).

#### ○ Interactive elements indicate their purpose and state

^

Interactive elements, such as links and buttons, should indicate their state and be distinguishable from non-interactive elements. [Learn how to decorate interactive elements with affordance hints](#).

#### ○ The page has a logical tab order

^

Tabbing through the page follows the visual layout. Users cannot focus elements that are offscreen. [Learn more about logical tab ordering](#).

#### ○ Visual order on the page follows DOM order

^

DOM order matches the visual order, improving navigation for assistive technology. [Learn more about DOM and visual ordering](#).

#### ○ User focus is not accidentally trapped in a region

^

A user can tab into and out of any control or region without accidentally trapping their focus. [Learn how to avoid focus traps](#).

#### ○ The user's focus is directed to new content added to the page

^

If new content, such as a dialog, is added to the page, the user's focus is directed to it. [Learn how to direct focus to new content](#).

#### ○ HTML5 landmark elements are used to improve navigation

^

Landmark elements (`<main>`, `<nav>`, etc.) are used to improve the keyboard navigation of the page for assistive technology. [Learn more about landmark elements](#).

- Offscreen content is hidden from assistive technology

^

Offscreen content is hidden with display: none or aria-hidden=true. [Learn how to properly hide offscreen content.](#)

- Custom controls have associated labels

^

Custom interactive controls have associated labels, provided by aria-label or aria-labelledby. [Learn more about custom controls and labels.](#)

- Custom controls have ARIA roles

^

Custom interactive controls have appropriate ARIA roles. [Learn how to add roles to custom controls.](#)

ここに、自動テストツールではカバーできない範囲に対処する項目が表示されます。詳しくは、[ユーザー補助機能の審査を実施する方法についてのガイド](#)をご覧ください。

## 合格した監査 (5)

非表示

ドキュメントの `<body>` に `[aria-hidden="true"]` は設定されていません

^

スクリーン リーダーなどの支援技術は、ドキュメントの `<body>` に `aria-hidden="true"` が設定されていると、正常に動作しません。[aria-hidden がドキュメントの本文に与える影響についての詳細](#)

## ボタンにユーザー補助機能名が指定されています

^

ボタンにユーザー補助機能名が指定されていない場合、スクリーン リーダーでは「ボタン」と読み上げられるため、スクリーン リーダーを使用しているユーザーはボタンの用途がわからず、使用することができなくなります。[ボタンにアクセスしやすくするための方法の詳細](#)

ドキュメントに `<title>` 要素が指定されています

^

タイトルを指定すると、スクリーン リーダーのユーザーがページの概要を把握できるようになります。検索エンジンの使用時には、検索語句に関連するページかどうかを判断するための重要な要素となります。[ドキュメントのタイトルに関する詳細](#)

## タップ ターゲットのサイズと間隔は十分な大きさです。

^

タップ ターゲットのサイズと間隔を十分な大きさにすると、小さいコントロールでもターゲットを選択しやすくなります。[タップ ターゲットの詳細](#)

## 見出し要素は降順になっています

^

見出しを適切なレベルの順序で配置すると、ページのセマンティック構造を伝えることができ、支援技術を使用した操作やコンテンツの把握が簡単になります。[見出しの順序の詳細](#)

該当なし (48)

非表示

○ [\[accesskey\] の値は一意です](#)

^

アクセキーは、ユーザーがページの特定の部分にすばやくフォーカスを移動するときに使います。正しく操作できるよう、各アクセキーは一意にする必要があります。[アクセキーの詳細](#)

○ [\[aria-\\*\] 属性は役割と一致しています](#)

^

各 ARIA role は、aria-\* 属性の特定のサブセットに対応しています。これらが一致しない場合、aria-\* 属性は無効になります。[ARIA 属性をロールと照合する方法の詳細](#)

○ [互換性のある要素にのみ ARIA ロールを使用する](#)

^

多くの HTML 要素は、特定の ARIA ロールのみを割り当てることができます。ARIA ロールを許可されていない場所で使用すると、ウェブページのアクセシビリティが妨げられる可能性があります。[ARIA ロールの詳細](#)

○ [button、link、menuitem 要素にユーザー補助機能向けの名前が設定されています](#)

^

要素にユーザー補助機能用の名前が設定されていない場合、スクリーン リーダーでは一般名で読み上げられるため、スクリーン リーダーのみを使用するユーザーには用途がわかりません。[コマンド要素にアクセスしやすくする方法の詳細](#)

○ [ARIA 属性は要素のロールで指定されたとおりに使用されています](#)

^

一部の ARIA 属性は、特定の条件下に限り要素で使用できます。[条件付き ARIA 属性の詳細](#)

○ [非推奨の ARIA ロールは使用されていませんでした](#)

^

非推奨の ARIA ロールは、支援技術によって正しく処理されない可能性があります。[非推奨の ARIA ロールの詳細](#)

○ [role="dialog" または role="alertdialog" が含まれる要素に、ユーザー補助機能用の名前が設定されています。](#)

^

ARIA ダイアログ要素にユーザー補助機能用の名前が設定されていない場合、スクリーン リーダーのユーザーはこれらの要素の用途を識別できません。[ARIA ダイアログ要素をアクセスしやすくする方法の詳細](#)

○ [\[aria-hidden="true"\] 要素にフォーカス可能な子要素は含まれていません](#)

^

フォーカス可能な子要素が [aria-hidden="true"] 要素内にある場合、スクリーン リーダーなどの支援技術を使用するユーザーはこれらの操作可能な要素を使用できません。[aria-hidden がフォーカス可能な要素に与える影響の詳細](#)

○ [ARIA 入力フィールドにユーザー補助機能向けの名前が設定されています](#)

^

入力フィールドにユーザー補助機能用の名前が設定されていない場合、スクリーン リーダーでは一般名で読み上げられるため、スクリーン リーダーのみを使用するユーザーには用途がわかりません。[入力フィールドのラベルの詳細](#)

- ARIA [meter](#) 要素にユーザー補助機能向けの名前が設定されています

^

メーター要素にユーザー補助機能用の名前が設定されていない場合、スクリーン リーダーでは一般名で読み上げられるため、スクリーン リーダーのみを使用するユーザーには用途がわかりません。[meter 要素に名前を付ける方法の詳細](#)

- ARIA [progressbar](#) 要素にユーザー補助機能向けの名前が設定されています

^

progressbar 要素にユーザー補助機能用の名前が設定されていない場合、スクリーン リーダーでは一般名で読み上げられるため、スクリーン リーダーのみを使用するユーザーには用途がわかりません。[progressbar 要素にラベルを付ける方法の詳細](#)

- 許可された ARIA 属性のみが要素で使用されています

^

禁止されているロールで ARIA 属性を使用すると、支援技術のユーザーに重要な情報が伝わらない可能性があります。[禁止されている ARIA ロールの詳細](#)

- [\[role\]](#) に必須の [\[aria-\\*\]](#) 属性がすべて指定されています

^

一部の ARIA ロールには、スクリーン リーダーに要素の状態を伝える必須の属性があります。[ロールと必須属性の詳細](#)

- ARIA [\[role\]](#) が指定されている要素に、特定の [\[role\]](#) を含む必要な子がすべて指定されています。

^

目的のユーザー補助機能を実行するには、一部の ARIA 親ロールに特定の子ロールを含める必要があります。[ロールと必要な子要素の詳細](#)

- [\[role\]](#) は必須の親要素に含まれています

^

想定されるユーザー補助機能が適切に動作するためには、一部の ARIA 子ロールは、特定の親ロールに含まれている必要があります。[ARIA ロールと必要な親要素の詳細](#)

- [\[role\]](#) の値は有効です

^

目的のユーザー補助機能を実行するには、ARIA ロールに有効な値を指定してください。[有効な ARIA ロールの詳細](#)

- [role=text](#) 属性が指定された要素にフォーカス可能な子孫がありません。

^

マークアップによってテキストノードの分割を [role=text](#) で囲むと、ナレーションで 1 つのフレーズとして扱うことができますが、要素のフォーカス可能な子孫は読み上げられません。[role=text 属性の詳細](#)

- ARIA 切り替えフィールドにユーザー補助機能向けの名前が設定されています

^

切り替えフィールドにユーザー補助機能用の名前が設定されていない場合、スクリーン リーダーでは一般名で読み上げられるため、スクリーン リーダーのみを使用するユーザーには用途がわかりません。[切り替えフィールドの詳細](#)

○ ARIA [tooltip](#) 要素にユーザー補助機能向けの名前が設定されています ^

ツールチップ要素にユーザー補助機能用の名前が設定されていない場合、スクリーン リーダーでは一般名で読み上げられるため、スクリーン リーダーのみを使用するユーザーには用途がわかりません。[tooltip 要素に名前を付ける方法の詳細](#)

○ ARIA [treeitem](#) 要素にユーザー補助機能向けの名前が設定されています ^

[treeitem](#) 要素にユーザー補助機能用の名前が設定されていない場合、スクリーン リーダーでは一般名で読み上げられるため、スクリーン リーダーのみを使用するユーザーには用途がわかりません。[treeitem 要素のラベル付けの詳細](#)

○ [\[aria-\\*\]](#) 属性に有効な値が指定されています ^

支援技術（スクリーン リーダーなど）で、無効な値が指定された ARIA 属性を解釈できません。[ARIA 属性で有効な値の詳細](#)

○ [\[aria-\\*\]](#) 属性は有効で、スペルミスもありません ^

支援技術（スクリーン リーダーなど）で、無効な名前が指定された ARIA 属性を解釈できません。[有効な ARIA 属性の詳細](#)

○ ページに見出し、スキップリンク、またはランドマーク領域が設定されています ^

重複するコンテンツをスキップする手段を追加すると、キーボードを使ったページの移動を効率化できます。[プロックの回避についての詳細](#)

○ [<dl>](#) には、正しく順序付けられた [<dt>](#) および [<dd>](#) グループ、[<script>](#)、[<template>](#)、または [<div>](#) 要素のみが含まれています。 ^

定義リストが適切にマークアップされていないと、スクリーン リーダーで、誤解を招く内容や不正確な内容が読み上げられる可能性があります。[定義リストを正しく構成する方法の詳細](#)

○ 定義リストの項目は [<dl>](#) 要素でラップされています ^

スクリーン リーダーで正しく読み上げられるようにするには、定義リストの項目（[<dt>](#) と [<dd>](#)）を親の [<dl>](#) 要素でラップする必要があります。[定義リストを正しく構成する方法の詳細](#)

○ ARIA ID は一意です ^

他のインスタンスが支援技術によって見落とされることのないように、ARIA ID の値は一意にする必要があります。[重複する ARIA ID の修正方法の詳細](#)

○ 複数のラベルが設定されたフォーム フィールドはありません ^

フォーム フィールドに複数のラベルが設定されている場合、スクリーン リーダーなどの支援技術で最初または最後のラベルだけ、もしくはすべてのラベルが読み上げられるため、混乱が生じる可能性があります。[フォームラベルの使用方法の詳細](#)

- <frame> または <iframe> の要素にタイトルが指定されています

^

スクリーン リーダーでは、フレームのコンテンツを説明するためにフレームのタイトルが使用されます。[フレームのタイトルの詳細](#)

- <html> 要素の [lang] 属性に有効な値が指定されています

^

有効な [BCP 47 言語](#) を指定すると、スクリーン リーダーでテキストが正しく読み上げられるようになります。[lang 属性の使用方法の詳細](#)

- <html> 要素に、[lang] 属性と同じベース言語の [xml:lang] 属性があります。

^

ウェブページで一貫性のある言語が指定されていない場合、スクリーン リーダーでページのテキストが正しく読み上げられないことがあります。[lang 属性の詳細](#)

- 画像要素に [alt] 属性が指定されています

^

説明的要素は、簡潔でわかりやすい代替テキストにする必要があります。装飾的要素は、alt 属性が空の場合は無視される可能性があります。[alt 属性の詳細](#)

- 画像要素に、重複する内容の [alt] 属性は含まれていません。

^

説明的要素は、簡潔でわかりやすい代替テキストにする必要があります。リンクや画像の近くのテキストとまったく同じ代替テキストを指定すると、スクリーン リーダーで同じ内容が 2 回読み上げられるため、混乱を招く可能性があります。[alt 属性の詳細](#)

- 入力ボタンに識別可能なテキストがあります。

^

識別可能およびアクセス可能なテキストを入力ボタンに追加すると、スクリーン リーダーのユーザーが入力ボタンの目的を把握するのに役立ちます。[入力ボタンの詳細](#)

- <input type="image"> 要素に [alt] テキストが指定されています

^

画像を <input> ボタンとして使用している場合は、代替テキストを指定すると、スクリーン リーダーのユーザーがボタンの用途を理解しやすくなります。[入力画像の代替テキストの詳細](#)

- リンクは色に依存せずに識別可能です。

^

低コントラストのテキストを使用すると、多くのユーザーは読むことが困難または不可能になります。識別可能なリンクテキストを使用すると、ロービジョンのユーザーの利便性が向上します。[リンクを識別可能にする方法の詳細](#)

- リンクに識別可能な名前が指定されています

^

識別可能、フォーカス可能な一意のリンクテキスト（および画像をリンクとして使用している場合はその代替テキスト）を

使用すると、スクリーン リーダーでのナビゲーションの操作性が向上します。[リンクをアクセス可能にする方法の詳細](#)

- リストには、`<li>` 要素と、スクリプト対応要素（`<script>` と `<template>`）のみが含まれています。

^

スクリーン リーダーでは、特殊な方法でリストが読み上げられます。適切に読み上げられるようにするには、正しいリスト構造を指定する必要があります。[適切なリスト構造についての詳細](#)

- リスト項目（`<li>`）は `<ul>`、`<ol>`、`<menu>` の親要素に含まれています

^

スクリーン リーダーで正しく読み上げられるようにするには、リスト項目（`<li>`）を親の `<ul>`、`<ol>`、`<menu>` に含める必要があります。[適切なリスト構造についての詳細](#)

- ドキュメントで `<meta http-equiv="refresh">` が使用されていません

^

ユーザーはページが自動的に更新されると思っていないため、自動更新によってフォーカスがページ上部に戻ると、ユーザーの利便性が低下する可能性があります。[メタタグの更新の詳細](#)

- `<object>` 要素に代替テキストが指定されています

^

スクリーン リーダーは、テキスト以外のコンテンツを解釈できません。`<object>` 要素に代替テキストを追加すると、スクリーン リーダーを使用するユーザーが意味を把握するのに役立ちます。[object 要素の代替テキストの詳細](#)

- 一部の要素にラベル要素が関連付けられています。

^

効果的なラベルのないフォーム要素は、スクリーン リーダーのユーザーにとって不便となる可能性があります。[select 要素の詳細](#)

- スキップリンクはフォーカス可能です。

^

スキップリンクを含めると、メインのコンテンツにスキップできるようになり、ユーザーの手間が省けます。[スキップリンクの詳細](#)

- `[tabindex]` に 0 より大きい値を指定している要素はありません

^

値が 0 より大きい場合は、明示的なナビゲーション順序を示します。技術的には有効ですが、多くの場合、支援技術を使用しているユーザーにとって利便性が低下します。[tabindex 属性の詳細](#)

- 表の `summary` 属性と `<caption>` の内容は異なっています。

^

`summary` 属性では表構造を説明し、`<caption>` には画面に表示されるタイトルを指定します。表のマークアップを正確に行うと、スクリーン リーダーを使用するユーザーに役立ちます。[summary と caption の詳細](#)

- `<table>` 要素で `[headers]` 属性を使用しているセルは、同じ表内の表セルを参照しています。

^

スクリーン リーダーには、表内の移動を補助する機能があります。[headers] 属性を使用している <td> セルが同じ表の他のセルのみを参照するように設定すると、スクリーン リーダーの利便性が向上する可能性があります。[headers 属性の詳細](#)

- <th> 要素および [role="columnheader"/"rowheader"] が指定された要素に、記述されたデータセルがあります。

^

スクリーン リーダーには、表内の移動を補助する機能があります。表のヘッダーが常に一部のセルを参照するように設定すると、スクリーン リーダーの利便性が向上する可能性があります。[テーブル ヘッダーの詳細](#)

- [lang] 属性に有効な値が指定されています

^

要素に有効な [BCP 47 言語](#) を指定すると、スクリーン リーダーでテキストが正しく読み上げられるようになります。[lang 属性の使用方法の詳細](#)

- <video> 要素に [kind="captions"] が指定された <track> 要素が含まれています

^

動画に字幕を設定すると、聴覚障がいのあるユーザーが情報にアクセスしやすくなります。[動画の字幕についての詳細](#)



## おすすめの方法

### 全般

- ▲ ブラウザのエラーがコンソールに記録されました

^

コンソールに記録されたエラーは未解決の問題を表します。これらはネットワーク リクエストの失敗や他のブラウザの問題が原因で表示される可能性があります。[コンソールの診断監査でのこのエラーの詳細](#)

ソース	説明
localhost <a href="#">自社</a>	:3000/_nuxt/xC2XOs-m.js:1 Failed to load resource: the server responded with a status of 500 (Server Error)
14ade20e-ed05-4fd2-9...9f5cf8b0ab30.json:1	Failed to load resource: the server responded with a status of 500 (Server Error)

## ソース

## 説明

(インデックス) :1

Refused to apply style from 'http://localhost:3000/\_nuxt/entry.CN-Sic9a.css' because its MIME type ('application/json') is not a supported stylesheet MIME type, and strict MIME checking is enabled.

○ JavaScript ライブラリが検出されました ^

ページで検出されたすべてのフロントエンドの JavaScript ライブラリです。 [JavaScript ライブラリ検出の診断監査についての詳細](#)

## 名前

## バージョン

Nuxt.js

## 信頼性と安全性

○ CSP が XSS 攻撃を防げるよう設定されているか確認する ^

厳格なコンテンツ セキュリティ ポリシー (CSP) を設定すると、クロスサイト スクリプティング (XSS) 攻撃のリスクを大幅に軽減できます。 [CSP を使用して XSS を防ぐ方法の詳細](#)

## 説明

## ディレクティブ

## 重大度

適用されている CSP はありません

高

○ 強力な HSTS ポリシーを使用する ^

HSTS ヘッダーを実装すると、HTTP 接続のダウングレードやスニッフィング攻撃のリスクが大幅に軽減されます。低い max-age から段階的にロールアウトすることをおすすめします。 [強力な HSTS ポリシーの使用に関する詳細](#)

## 説明

## ディレクティブ

## 重大度

HSTS ヘッダーはありません

高

○ COOP でオリジン分離が適切に行われていることを確認する ^

Cross-Origin-Opener-Policy (COOP) を使用すると、ポップアップなどの他のドキュメントから最上位のウィンドウを分離できます。 [COOP ヘッダーの実装に関する詳細](#)

説明	ディレクティブ	重大度
COOP ヘッダーはありません		高

○ XFO または CSP を使用してクリックジャッキングを軽減する ^

X-Frame-Options (XFO) ヘッダーまたは Content-Security-Policy (CSP) ヘッダーの frame-ancestors ディレクティブは、ページを埋め込むことができる場所を制御します。これらを使用して、一部またはすべてのサイトでページの埋め込みをブロックすることで、クリックジャッキング攻撃を軽減できます。[クリックジャッキングを軽減する方法について](#)

説明	重大度
フレーム制御ポリシーが見つかりません	高

○ Trusted Types を使用して DOM ベースの XSS に対処する ^

Content-Security-Policy (CSP) ヘッダーの require-trusted-types-for ディレクティブは、DOM XSS シンク関数に渡されるデータの制御をユーザー エージェントに指示します。[Trusted Types を使用して DOM ベースの XSS に対処する方法の詳細](#)

説明	重大度
Trusted Types ディレクティブを含む `Content-Security-Policy` ヘッダーが見つかりません	高

合格した監査 (14) 非表示

HTTPS を使用しています ^

すべてのサイトは、機密性の高い情報を扱っていない場合でも、HTTPS で保護する必要があります。たとえば、[混合コンテンツ](#) (HTTPS で送信されたリクエストに対して一部のリソースを HTTP で読み込む) などは使用しないようにします。HTTPS は、侵入者があなたのアプリとユーザー間の通信を改ざんしたり、傍受したりするのを防ぎます。HTTPS は、HTTP/2 や多くの新しいウェブ プラットフォーム API を使用するための前提条件となります。[HTTPS についての詳細](#)

サポートを終了した API は使用されていません ^

非推奨の API は最終的にブラウザから削除されます。[非推奨 API の詳細](#)

サードパーティ Cookie は使用されていません ^

サードパーティ Cookie は、状況によってはブロックされることがあります。[サードパーティ Cookie の制限に備える方法について](#)

ユーザーは入力フィールドに貼り付けできます ^

ユーザーの利便性が損なわれるため、入力フィールドへの貼り付けを禁止することは推奨されません。パスワード マネージャーがブロックされるため、セキュリティの低下にもつながります。詳しくは、[ユーザーフレンドリーな入力フィールド](#)をご覧ください。

ページの読み込み時に位置情報の許可はリクエストされません ^

サイトから脈絡なしに位置情報の許可を求められると、ユーザーは不信感を抱き、困惑します。リクエストはユーザーの操作と関連付けて行うようにしてください。[位置情報の権限に関する詳細](#)

ページの読み込み時に通知の許可はリクエストされません ^

サイトから脈絡なしに通知の送信許可を求められると、ユーザーは不信感を抱き、困惑します。リクエストはユーザーの操作と関連付けて行うようにしてください。[通知の許可を責任を持って取得する方法の詳細](#)

正しいアスペクト比の画像が表示されています ^

画像は本来のアスペクト比で表示する必要があります。[画像のアスペクト比についての詳細](#)

画像が適切な解像度で表示されています ^

画像をできるだけ鮮明に表示するには、画像の実サイズを、表示するサイズとピクセル比に比例させる必要があります。[レスポンシブな画像の表示方法の詳細](#)

`width` または `initial-scale` を指定した `<meta name="viewport">` タグがあります ^

`<meta name="viewport">` を使用すると、モバイル画面サイズ向けにアプリを最適化できるほか、[ユーザー入力の遅延を300ミリ秒](#)短縮できます。[ビューポート メタタグの使用の詳細](#)

ドキュメントで判読可能なフォントサイズが使用されています — 判読可能なテキスト: 80.85% ^

12 px より小さいフォントサイズは小さすぎて判読できず、モバイルユーザーには「ピンチしてズーム」の操作が必要になります。60% を超えるページテキストでフォント サイズが 12 px 以上になるようにしてください。[判読可能なフォントサイズの詳細](#)

ソース	選択ツール	ページテキストの割合	フォントサイズ
localhost <span style="border: 1px solid blue; padding: 2px;">自社</span>			
	(インデックス) :1 .hint-desc[data-v-97ca75e3]	19.15%	11px
アトリビューション不可			

ソース	選択ツール	ページテキストの割合	フォントサイズ
判読可能なテキスト		80.85%	≥ 12px

ページに HTML doctype が指定されています ^

doctype を指定すると、ブラウザは後方互換モードに切り替えることができなくなります。 [doctype 宣言の詳細](#)

文字セットを適切に定義する ^

文字エンコードの宣言が必要です。これには、HTML の最初の 1,024 バイトか Content-Type HTTP レスポンス ヘッダーの中で <meta> タグを使用します。 [文字エンコードの宣言についての詳細](#)

Chrome デベロッパー ツールの [Issues] パネルに記録されている問題はありません ^

Chrome デベロッパー ツールの [Issues] パネルに記録された問題は未解決のものです。ネットワーク リクエストの失敗、不十分なセキュリティ コントロールや、他のブラウザの問題が原因で表示される可能性があります。各問題について詳しくは、Chrome DevTools の [Issues] パネルを開いてご確認ください。

ページで有効なソースマップが使用されています ^

ソースマップは軽量化したコードを元のソースコードに変換するもので、デベロッパーが本番環境でデバッグする際に役立ちます。また、Lighthouse で詳しい分析情報を提供することも可能になります。このようなメリットをふまえ、ソースマップの導入をご検討ください。 [ソースマップの詳細](#)

該当なし (1) 非表示

○ HTTP トラフィックは HTTPS にリダイレクトされます ^

すべてのユーザーに安全なウェブ機能を提供できるように、すべての HTTP トラフィックが HTTPS にリダイレクトされることを確認してください。 [詳細](#)



## SEO

これらのチェックを行うことで、ページが検索エンジン最適化の基本的な推奨事項に沿っていることを確認できます。検索ランキングに影響する可能性のある要素には、Lighthouse で考慮されている要素以外に、[ウェブに関する主な指標](#)のスコアなどもあります。 [Google 検索の基本事項の詳細](#)

## 個別の検証が必要な他の項目 (1)

非表示

- 構造化データが有効です

^

[構造化データ テストツール](#)と[構造化データ用 Linter](#)を実行して構造化データを検証してください。[構造化データの詳細](#)

サイトでこれらの他の検証ツールを実行し、SEO のその他のおすすめの方法をご確認ください。

## 合格した監査 (8)

非表示

ページのインデックス登録はブロックされていません

^

ページのクロールを許可しない場合、検索エンジンはそのページを検索結果に追加できません。[クローラディレクティブの詳細](#)

ドキュメントに `<title>` 要素が指定されています

^

タイトルを指定すると、スクリーンリーダーのユーザーがページの概要を把握できるようになります。検索エンジンの使用時には、検索語句に関連するページかどうかを判断するための重要な要素となります。[ドキュメントのタイトルに関する詳細](#)

ドキュメントにメタ ディスクリプションが指定されています

^

メタ ディスクリプションを検索結果に追加すると、ページコンテンツの内容を簡潔にまとめることができます。[メタ ディスクリプションの詳細](#)

ページに適切な HTTP ステータス コードが指定されています

^

HTTP ステータス コードが正しくないページはインデックスに適切に登録されていない可能性があります。[HTTP ステータス コードの詳細](#)

リンクにわかりやすいテキストが設定されています

^

リンクテキストをわかりやすくすると、検索エンジンがコンテンツを認識しやすくなります。[リンクにアクセスしやすくなる方法の詳細](#)

リンクはクロール可能です

^

検索エンジンでは、ウェブサイトのクロールにリンクの `href` 属性を使用する場合があります。アンカー要素の `href` 属性に適切なリンク先が設定されていて、ウェブサイトの他のページを見つけられるようになっていることを確認してください。[リンクをクロール可能にする方法の詳細](#)

robots.txt は有効です

^

robots.txt ファイルの形式が間違っていると、ウェブサイトのクロールやインデックス登録について指定した設定をクローラが認識できない可能性があります。 [robots.txtについての詳細](#)

ドキュメントに有効な `hreflang` が指定されています ^

`hreflang` リンクを利用して、所定の言語や地域の検索結果に掲載する必要があるページのバージョンを検索エンジンに伝えます。 [hreflangの詳細](#)

該当なし (2) 非表示

- 画像要素に `[alt]` 属性が指定されています ^

説明的要素は、簡潔でわかりやすい代替テキストにする必要があります。装飾的要素は、`alt` 属性が空の場合は無視される可能性があります。 [alt 属性の詳細](#)

- ドキュメントに有効な `rel=canonical` が指定されています ^

正規リンクで、検索結果に表示する URL を指定します。 [正規リンクの詳細](#)

 Captured at 2025年10月27日  
17:42 JST

 最初のページ読み込み

 Moto G Power のエミュレーション  
with Lighthouse 12.8.2

 低速 4G スロットリング

 1ページのみのセッション  
用 Chromium 141.0.0.0  
with devtools

Generated by **Lighthouse** 12.8.2 | [問題を提出](#)