

報道各位

2026年10月7日
株式会社インプレス

ブラウザだけで今すぐ始められる Nuxt 4 入門!

『サクッと動く! Vue.js & Nuxt 超入門 現場で役立つ Web アプリケーション構築のすべて』発行

技術の泉シリーズ 10 月の新刊

インプレスグループで IT・デザイン関連メディア事業を展開する株式会社インプレス(本社:東京都千代田区、代表取締役社長:高橋隆志)は、『サクッと動く! Vue.js & Nuxt 超入門 現場で役立つ Web アプリケーション構築のすべて』(著者:井口 湊太)をインプレス NextPublishing より発行いたします。

最新の知見を発信する『技術の泉シリーズ』は、「技術書典」や「技術書同人誌博覧会」をはじめとした各種即売会や、勉強会・LT 会などで頒布された技術同人誌を底本とした商業書籍を刊行し、技術同人誌の普及と発展に貢献することを目指します。

『サクッと動く! Vue.js & Nuxt 超入門 現場で役立つ Web アプリケーション構築のすべて』

<https://nextpublishing.jp/isbn/9784295605393>



著者:井口 湊太

小売希望価格:電子書籍版 1,800 円(税別)／印刷書籍版 2,000 円(税別)

電子書籍版フォーマット:EPUB3

印刷書籍版仕様:B5 版／カラー／本文 140 ページ

ISBN:978-4-295-60539-3

企画編集:合同会社技術の泉出版

発行:インプレス NextPublishing

発売:株式会社インプレス

発売日:2026 年 10 月 9 日

<<発行主旨>>

本書は、Vue とその上に構築されたフレームワークである Nuxt を用いた Web アプリケーション開発手法を解説する入門書です。JavaScript の変数、関数、条件分岐、ループ、配列、オブジェクトの基本的な扱い方を理解している読者を対象に、Vue や Nuxt の使用経験がない場合にも基礎から説明します。入門時の障壁となりがちな環境構築を後回しにし、ブラウザでコードを書きながら開発を始める方法と、Vue でデータを扱い表示する基本を学びます。続く章では Nuxt の主要な機能を活用したアプリケーションの構築を扱い、豊富なサンプルコードや TypeScript を使用したコード例を通じて内容を進めます。章末には学んだ内容を用いてアプリケーションや機能を実装する総合演習があり、付録には解答例を掲載し、最終章では Nuxt アプリケーションを公開する手順を説明します。

(インプレス NextPublishing は、株式会社インプレス R&D が開発したデジタルファースト型の出版モデルを承継し、幅広い出版企画を電子書籍＋オンデマンドによりスピーディで持続可能な形で実現しています。)

<<目次>>

まえがき

第 1 章 オンライン環境ではじめる Nuxt 開発

第 2 章 静的データのレンダリング

第 3 章 リアクティブデータの操作

第 4 章 ページルーティング

第 5 章 データフェッチと動的ルーティング

第 6 章 コンポーネントの作成と活用

第 7 章 状態管理

第 8 章 ライフサイクルと副作用

第 9 章 サーバーサイド処理の活用

第 10 章 アプリケーションの公開

ローカル開発環境の構築

総合演習の解答例

あとがき

<<著者紹介>>

井口 湊太

電気通信大学 I 類(情報系) 学生。Web フロントエンド開発を軸に、有志での学内向けツール開発や OSS の提供に携わっている。制作物は GitHub で 100 件以上のスターを獲得しているライブラリから、コミュニティの課題を解決するシステムまで幅広く、実用性を重視したソフトウェアの構築と公開に取り組んでいる。

<<内容紹介>>

第1章 オンライン環境ではじめるNuxt開発

第1章では、オンライン環境で簡単にNuxtの開発を始める方法を紹介します。その後にはVueとNuxtの基本的な特徴を学びます。

1.1 StackBlitzでNuxtを動かしてみよう

1.1.1 Nuxtとは？

Nuxtは、Webアプリケーションの画面を作るために使うJavaScriptのライブラリーである**Vue.js**に基くVueと記述、viewを同じ発想のベースにしにフレームワークです。Nuxtを使うことで、簡単に、かつ品質の高いWebアプリケーションを開発することができます。細かい特徴については次の節で詳しく説明しますが、まずは実際にNuxtを動かしてその便利さを体験してみましょう。

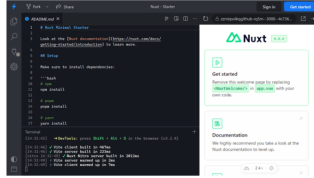
1.1.2 StackBlitzとは？

StackBlitz(<https://stackblitz.com/>)は、同名の企業が提供している、ブラウザ上で動作するオンライン開発環境です。開発環境のセットアップが不要で、すぐにコードを書いて試すことができます。Nuxtプロジェクトも、前例な環境構築なしに数秒で起動できるため、初めてNuxtに触れた人でも簡単に始められます。なお、ローカル環境で開発したい方向けは、ローカル環境での開発環境構築手順を別途で紹介しています。

1.1.3 Nuxtプロジェクトを作成する

Nuxtの開発チームが提供しているnuxt.newというサービスを使うと、StackBlitz上に簡単にNuxtプロジェクトを作成できます。<https://nuxt.new/v4>にアクセスしてください。自動的にStackBlitz上でNuxtのプロジェクトが作成されます。ページを開いて数秒待つかと、左側にはソースコードが、右側にはNuxtアプリケーションのプレビュー画面が表示されるはずです。

図1.1 StackBlitzで作成したNuxtのプロジェクト

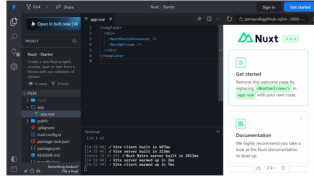


1.1.4 プロジェクトの構成をみる

作成されたプロジェクトには、いくつかのファイルが用意されています。その中でも最も重要なのが、app.vueファイルです。

左側のファイルツリーからappディレクトリ内にあるapp.vueを開いてみましょう。

図1.2 app.vueを開いた状態



StackBlitz で Nuxt 開発を即体験

本書の内容を踏まえ、この見開きでは、第1章では、オンライン環境のStackBlitzを使って、面倒な環境構築なしに数十秒でNuxtプロジェクトを起動する方法を解説します。作成されたapp.vueファイルの中身を確認し、コードを書き換えることでホットリロードによる表示更新を実際に体験できます。

第2章 静的データのレンダリング

第2章では、Vueにおいて値を画面に表示する基本的な方法を説明します。

2.1 マスタッシュ構文

マスタッシュ構文(mustache syntax)は、VueやNuxtでテンプレート内にデータを表示するための基本的な方法です。マスタッシュ構文では、{{ }}の中にJavaScriptの式(expression)を記述します。記述した式は、テンプレートが描画(rendering)される際に評価され、その結果がテキストとしてそのまま表示されます。ここで、式を評価するとは、JavaScriptエンジンがその表現を解釈、実行して、具体的な値を算出することを指します。計算式や関数の呼び出し、変数の参照など、さまざまなJavaScriptの表現が式として使用できます。

例えば、以下のようにテンプレート内でマスタッシュ構文を使うことができます。

```
1 <script setup lang="ts">
2 const name = "太郎";
3 const isAdmin = true;
4 </script>
5
6 <template>
7 <div>
8   <p>こんにちは、{{ name }}さん！</p>
9   <p>今日の日は {{ new Date().toLocaleDateString() }} 日です。</p>
10  <p>管理者の権限は {{ isAdmin ? "あり" : "なし" }} です。</p>
11 </div>
12 </template>
```

2.2 マスタッシュ構文が使えない場面

2.2.1 式でないもの(条件・繰り返し文など)

JavaScriptの式でないもの(文(statement)は、マスタッシュ構文内で使用できません。例えば、if文やfor文などの制御構文は式ではないため、マスタッシュ構文内で使用することはできません。これらの制御構文は、Vueでは特別な構文を使って表現しますが、それについては後の節で詳しく解説します。

リスト2.2 マスタッシュ構文で使えない例(式でないもの)

```
1 <template>
2 <div>
3   <p>{{ if (isAdmin) { "管理者です"; } }}</p>
4   <p>{{
5     for (let i = 0; i < 5; i++) { `{{i}} 番目の要素です` } }}
6   </p>
7 </div>
8 </template>
```

2.2.2 属性値の中

マスタッシュ構文は、HTML属性値の中では使用できません。例えば、以下のようなコードは正しく動作しません。

リスト2.3 属性値の中でのマスタッシュ構文の例

```
1 <script setup lang="ts">
2 const imageUrl = "https://example.com/image.jpg";
3 </script>
4
5 <template>
6 <div>
7   
8 </div>
9 </template>
```

出力されるHTMLは、とはならず、のままとなり、src属性に正しいURLが設定されません。式を用いた属性の指定にも専用の構文が用意されているため、次の節で解説します。

2.2.3 エスケープ処理

マスタッシュ構文では、式の結果に含まれるHTMLタグは自動的に無効化されます。

リスト2.4 エスケープの例

```
1 <script setup lang="ts">
2 const rhtml = "<strong>これは文字になるぞ！</strong>";
3 </script>
4
5 <template>
6 <div>
7   <p>{{ rhtml }}</p>
```

制御構文不可の制約下でv-bindを使う方法

第2章では、Vueで値を画面に表示する基本を解説します。ただしif文やfor文などの制御構文は使用できません。またHTML属性値内でも使えず、エスケープ処理によりHTMLタグは無効化されます。これらの制約に対し、v-bindディレクティブを用いて属性に式を紐づける方法も紹介しています。

第3章 リアクティブデータの操作

第2章では、変化しないデータを表示する方法について説明しました。第3章では、ユーザー入力に応じてデータを変更する方法について説明します。

3.1 リアクティブとは

リアクティブ(*reactive*)とは、「変化に応じて自動的に反応する」という意味です。VueやNextでは、データが変化した際に自動的に画面が更新される仕組みを指しています。例えば、ユーザーがボタンをクリックしてデータを変更した場合、その変化に応じて画面の表示内容も自動的に更新されます。

3.1.1 refでリアクティブなデータを作成する

JavaScriptにおける変数は`let`や`const`で宣言しますが、これらの変数はリアクティブではありません。VueやNextでリアクティブなデータを作成するには、`ref`関数を使用します。`ref`関数は、Vueが提供するリアクティブシステムの一部であり、リアクティブなデータを簡単に作成できます。

リスト3.1 refでリアクティブなデータを作成する例

```
1: <script setup lang="ts">
2: import { ref } from "vue";
3:
4: const count = ref(0);
5: </script>
```

上記の例では、`ref()`を使ってリアクティブなデータ `count` を作成しています。`ref`関数の引数には、初期値を渡します。この場合、`count`の初期値は0です。`count`はリアクティブなデータであり、その値が変更されると、関連するUIも自動的に更新されます。

補足: ref関数の自動インポート

リスト3.2 ref関数のインポート文

```
1: import { ref } from "vue";
```

この例は、Vueから `ref` 関数を読み込んで使用することを示しています。`ref` 関数は Vue の中心機能の一部であり、`vue-demi` ライブラリから `import { ref } from "vue"` することで利用可能になります。ただし、Nextの標準設定では、Vueの基本的な機能は自動的にインポートされるため、`ref` 関数も自動的にインポートする必要はありません。そのため、当該のコードでは、`ref` 関数のインポート行を省略して説明します。

3.1.2 refで作成したデータの操作

`ref`で作成したリアクティブなデータは、`.value` プロパティを通じてアクセスおよび操作します。

リスト3.3 refで作成したデータの操作例

```
1: <script setup lang="ts">
2: const count = ref(0);
3:
4: // 1秒ごとに増やす
5: onMounted(() => { // ブラウザで表示されたときに実行(後の章で解説)
6:   setInterval(() => {
7:     count.value += 1;
8:   }, 1000);
9: });
10: </script>
11:
12: <template>
13: <div>
14:   <span>カウンタ: {{ count }}</span>
15: </div>
16: </template>
```

上記の例では、`count.value` を使ってリアクティブなデータ `count` の値にアクセスし、1秒ごとにその値を1ずつ増加させています。代入するだけで表示が自動的に更新されていることを確認してください。また、スクリプト置(`script`)タグ内では、`value`が必要ですが、**テンプレート置**(`template`)タグ内では、`value`を省略して `count` と記述するだけで値が参照できます。

では、次の例を見てみましょう。

リスト3.4 値の変更とref値

```
1: <script setup lang="ts">
2: let count2 = 0;
3: onMounted(() => {
4:   setInterval(() => {
5:     count2 += 1;
6:   }, 1000);
```

refとv-on で画面更新とイベント処理

書籍全体の内容を踏まえ、この見開きでは、`ref` 関数でリアクティブなデータを作成し、`.value` プロパティで値を操作して画面を自動更新する仕組みを説明します。また、イベントを検知して処理を実行する `v-on` ディレクティブの使い方や、イベント引数の取得方法もサンプルコードを用いて紹介します。

第6章 コンポーネントの作成と活用

第6章では、ページの部品をコンポーネントに分割して作成・活用する方法について学びます。

6.1 コンポーネントの基本概念

6.1.1 コンポーネントとは

Webフロントエンド開発における**コンポーネント**(*component*)とは、**UIの一部を独立した部品として切り出し、再利用可能なもの**を指します。例えば、ボタン、ナビゲーションバー、カードなどがコンポーネントとしてよく使われます。さらに大い単位では、特定の機能を持つフォームやUIジュエルの全体なども、コンポーネントとして定義されることがあります。より広義には、今章で説明したテンプレート置とインポート置のコンポーネントとして扱われます。**同じデザインや機能を保持しUI要素を複数の場所で使用する場合や、複雑なUIを分割して管理しやすくしたい場合に**、コンポーネントは非常に有用です。

コンポーネントを使用する主な利点は、以下の通りです。

- **再利用性の向上**：一度作成したコンポーネントは、アプリケーション内の複数の場所で再利用できます。これにより、コードの重複を減らし、開発効率を向上させることができます。
- **保守性の向上**：コンポーネントは独立しているため、特定のコンポーネントを修正しても、他の部分に影響を与えにくくなります。これにより、バグの修正や機能追加が容易になります。
- **可読性の向上**：コンポーネントを使用することで、コードが整理され、構造が明確になります。これにより、他の開発者に渡す自分のコードを認識したときにも内容を理解しやすくなります。

6.1.2 コンポーネントの作成

Nextでは、コンポーネントを `components` ディレクトリ内に作成します。例えば、`components` ディレクトリ内に `MyButton.vue` というファイルを作成すると、そのコンポーネントをアプリケーション内で使用できるようになります。

リスト6.1 コンポーネントの例

```
1: <template>
2:   <button class="my-button">My Button</button>
3: </template>
4:
5: <style scoped>
6:   .my-button {
7:     background-color: #4285f4;
```

```
8:   color: white;
9:   border: none;
10:   padding: 8.5em 1em;
11:   border-radius: 0.25em;
12:   cursor: pointer;
13: }
14: </style>
```

ここでは、シンプルをボタンコンポーネントを作成しました。`<template>`タグ内にコンポーネントのHTML構造を記述し、`<style scoped>`タグ内にスタイルを定義しています。`scoped`属性を付けることで、このスタイルはこの**コンポーネント内でのみ適用**され、他のコンポーネントには影響しません。

6.1.3 コンポーネントの使用

作成したコンポーネントは、他のコンポーネントやページで使用できます。使用するには、コンポーネントを `/>` の形で記述します。Nextの自動インポート機能により、`components` ディレクトリ内のコンポーネントは自動的に利用可能になります。

リスト6.2 コンポーネントの例

```
1: <template>
2:   <div>
3:     <div>Welcome to My Next App!</div>
4:     <my-button />
5:   </div>
6: </template>
```

上記の例では、`MyButton` コンポーネントをトップページで使用しています。

6.1.4 スロットの使用

今回の例のように、コンポーネントの表示内容が事前に決まっていることは少なく、**コンポーネントの利用者が表示内容を指定**できるようにすることが一般的です。そのために、Vueでは**スロット**(`slot`)という仕組みが用意されています。スロットを使用すると、コンポーネントの特定の部分に、利用者が任意の内容を挿入できるようになります。

リスト6.3 UI要素とコンポーネント例

```
1: <template>
2:   <button class="my-button">
3:     <slot />
4:   </button>
5: </template>
```

Nuxt でコンポーネント分割とスロット活用

第6章では、UIを部品として切り出すコンポーネントの作成方法を解説します。`components` ディレクトリ内にファイルを作成し、自動インポート機能で利用する手順を示します。また、表示内容を指定できるスロットや、親から子へデータを渡す `Props` の仕組みも、サンプルコードを通じて確認できます。

<<販売ストア>>

電子書籍：

Amazon Kindle ストア、楽天 kobo イブックスストア、Apple Books、紀伊國屋書店 Kinoppy、Google Play Store、honto 電子書籍ストア、Sony Reader Store、BookLive!、BOOK☆WALKER、BOOK TECH

印刷書籍：

Amazon.co.jp

※ 各ストアでの販売は準備が整いしだい開始されます。

※ 全国の一般書店からもご注文いただけます。

<<技術の泉シリーズについて>>

『技術の泉シリーズ』は、技術者の知見のアウトプットである技術同人誌を底本とした、2017 年創刊の技術書シリーズです。NextPublishing によるスピーディな編集制作とプリントオンデマンドによる 1 冊からの印刷製本により、技術の変化に追従しつつ返品や品切れのないサステナブルな出版モデルを特徴としています。本シリーズを通じて、エンジニアの“知の結晶”である技術同人誌の世界に、より多くの方が触れていただくきっかけとなることを目指しています。



【株式会社インプレス】 <https://www.impress.co.jp/>

シリーズ累計 8,000 万部突破のパソコン解説書「できる」シリーズ、デザイン・グラフィック関連の書籍ブランド「MdN」、「デジタルカメラマガジン」等の定期雑誌、IT 関連の専門メディアとして国内最大級のアクセスを誇るデジタル総合ニュースサービス「Impress Watch シリーズ」等のコンシューマ向けメディア、「IT Leaders」をはじめとする企業向け IT 関連メディアなどを総合的に展開・運営する事業会社です。IT・デザイン関連出版メディア事業、およびデジタルメディア＆サービス事業を幅広く展開しています。

【インプレスグループ】 <https://www.impressholdings.com/>

株式会社インプレスホールディングス(本社: 東京都千代田区、代表取締役: 塚本由紀)を持株会社とするメディアグループ。「IT・デザイン」「音楽」「山岳・自然」「航空・鉄道」「モバイルサービス」「学術・理工学」を主要テーマに専門性の高いメディア&サービスおよびソリューション事業を展開しています。さらに、コンテンツビジネスのプラットフォーム開発・運営も手がけています。

【NextPublishing】

NextPublishing は、インプレス R&D が開発した電子出版プラットフォーム(またはメソッド)の名称です。電子書籍と印刷書籍の同時制作、プリント・オンデマンド(POD)による品切れ解消などの伝統的出版の課題を解決しています。これにより、伝統的出版では経済的に困難な多品種少部数の出版を可能にし、優秀な個人や組織が持つ多様な知の流通を目指しています。

【本件に関するお問合せ先】

株式会社インプレス : PubX 室

E-mail: np-info@impress.co.jp