



## Vite против Nextjs: Какой фреймворк для веб-разработки выбрать?

### Описание

Если вы хотите создать пользовательский интерфейс для своих веб-приложений, то Vite и Next.js могут стать одними из тех фреймворков, которые вы можете рассмотреть. Эти два фреймворка имеют некоторые общие черты и во многом отличаются друг от друга, поэтому понять, какой из них лучше, может быть нелегко. В этой статье мы сравним Vite и Next.js, рассмотрим их особенности, сходства и различия и поможем вам принять взвешенное решение.

### Что такое Vite?

**Vite**  
Next Generation  
Frontend Tooling

Get ready for a development environment that  
can finally catch up with you.



Vite – это инструмент для сборки, который призван обеспечить быструю и

компактную разработку в области веб-разработки. Это фреймворк, ориентированный на мнение пользователей, с разумными настройками по умолчанию. Vite можно интегрировать с другими инструментами и фреймворками с помощью плагинов.

## Особенности Vite

- **Быстрый HMR:** функция горячей замены модулей (HMR) обеспечивает высокую скорость работы приложения Vite независимо от его размера.
- **Мгновенный запуск сервера:** Приложения Vite не требуют комплектации, так как обслуживают файлы по требованию через собственный ESM.
- **Оптимизированная сборка:** Vite поставляется с предварительно сконфигурированной сборкой rollup. Этот инструмент также поддерживает библиотечный режим и многостраничность.
- **Полностью типизированные API:** Vite можно использовать с JavaScript и TypeScript. Его API являются гибкими и программными.
- **Универсальные плагины:** В Vite реализован интерфейс плагинов rollup-superset, разделяемый между средами разработки и сборки.

## Плюсы использования Vite



- **Быстрое время сборки:** Vite представила новый подход, который не требует использования бандлера при разработке. Таким образом, разработчики получают достаточно времени, чтобы сосредоточиться на разработке, особенно при работе с крупными проектами.
- **Бесшовная интеграция:** Благодаря обширной экосистеме плагинов Vite можно интегрировать с современными инструментами и фреймворками фронтенд-разработки.
- **Живая разработка:** Vite запускает живой сервер, что позволяет видеть изменения, вносимые в код, в режиме реального времени. Такой подход позволяет легко отлаживать и выявлять ошибки на ранней стадии.
- **Поддержка современных веб-стандартов:** Vite использует современные API браузеров и собственные модули ES. Такой подход позволяет создавать проекты на основе современных практик и облегчает сопровождение кодовых баз.

## Минусы использования Vite

- **Небольшое сообщество:** Vite еще молод и не имеет большого сообщества по сравнению с конкурентами, такими как Next.js.

- **Совместимость с браузерами:** Vite ограничен стандартами современных браузеров. Таким образом, пользователи старых браузеров могут не воспользоваться этим инструментом.
- **Ориентация на рендеринг на стороне клиента:** Vite не имеет встроенного рендеринга на стороне сервера (SSR), который есть в таких альтернативах, как Next.js. Однако некоторые приложения, созданные с помощью Vite, можно настроить на SSR.

## Что такое Next.js?



Next.js – это веб-фреймворк React. Next.js позволяет разработчикам создавать полнофункциональные веб-приложения, используя новейшие возможности React. Фреймворк построен на основе React (самой популярной библиотеки JavaScript), Turbopack (бандлера, написанного на языке Rust и оптимизированного для JavaScript) и Speedy Web Compiler (расширяемого инструмента на основе Rust, который может использоваться для минификации и компиляции).

### Особенности Next.js

- **Динамическая потоковая передача HTML:** Next.js позволяет мгновенно

---

транслировать пользовательский интерфейс с сервера, который интегрирован с React Suspense и App Router.

- **Встроенные оптимизации:** При работе с Next.js вы можете воспользоваться автоматической оптимизацией изображений, скриптов и шрифтов для улучшения пользовательского интерфейса.
- **Серверные компоненты React:** Next.js построен на основе новейших функций React. Вы можете добавлять компоненты в приложение Next.js без передачи дополнительного JavaScript на стороне клиента, что означает увеличение скорости работы.
- **Расширенная маршрутизация и вложенные макеты:** Этот фреймворк позволяет создавать новые маршруты с помощью файловой системы. Next.js также поддерживает расширенные макеты пользовательского интерфейса и шаблоны маршрутизации.
- **Обработчики маршрутов:** Next.js позволяет разработчикам создавать конечные точки API, которые интегрируются со сторонними сервисами и потребляются из пользовательского интерфейса.

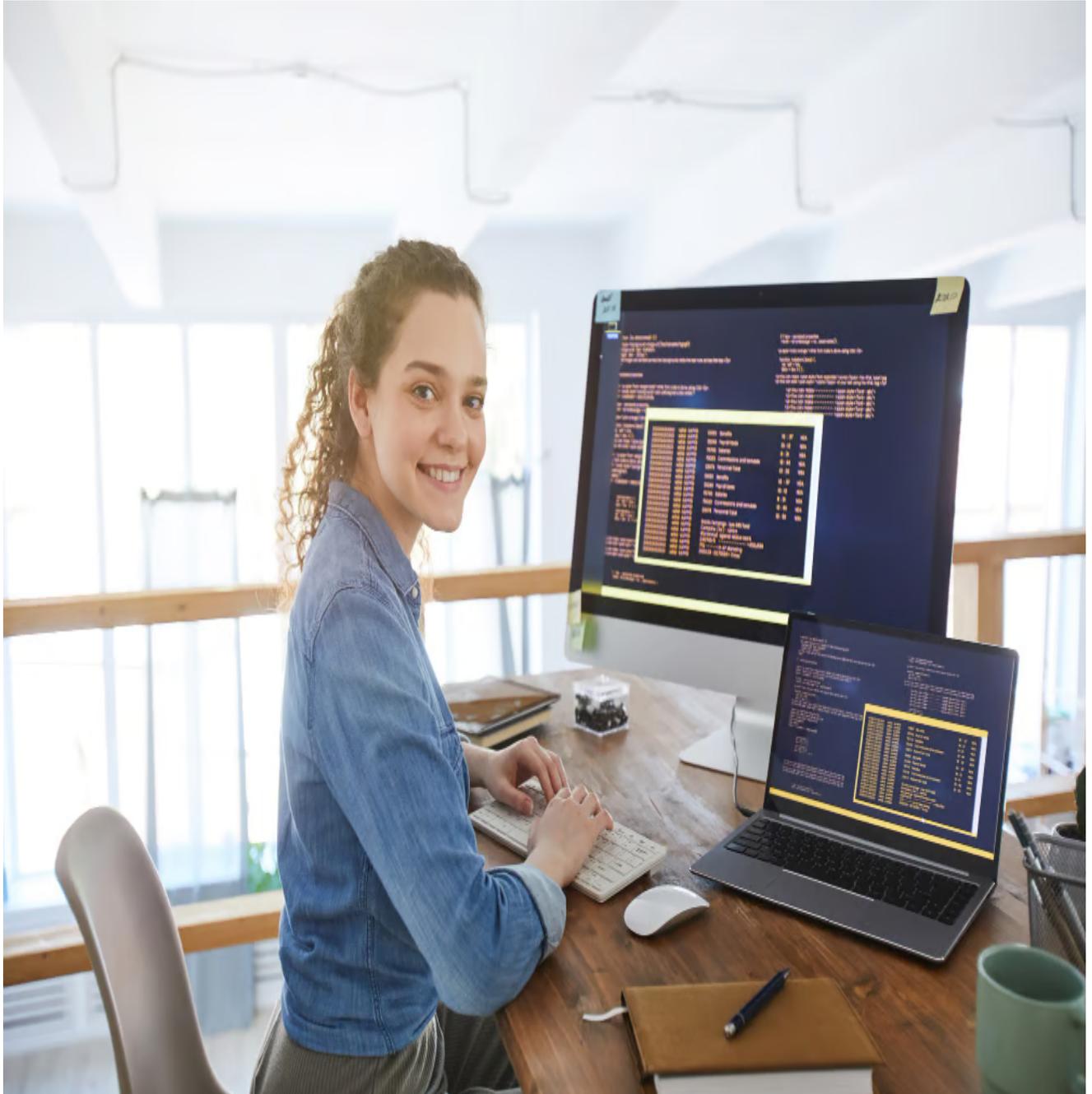
## Плюсы использования Next.js

- **Небольшая кривая обучения:** Next.js – это легкий в освоении фреймворк, особенно если вы пришли из React или простого JavaScript.
- **Предварительный рендеринг:** Next.js поддерживает рендеринг на стороне сервера. Техника SSR осуществляет предварительную загрузку данных во время сборки, что означает, что пользователю не нужно ждать загрузки данных при каждом их изменении. Такой подход делает Next.js быстрее, чем некоторые фреймворки, которые полагаются на рендеринг на стороне клиента.
- **Модульная архитектура и возможность повторного использования кода:** Приложение Next.js может быть разбито на небольшие блоки кода, объединенные в компоненты. Такая модульная архитектура позволяет легко поддерживать и повторно использовать код на разных страницах приложения.
- **Обширная экосистема:** Next.js имеет большое сообщество и инструменты, которые вы можете использовать для расширения функциональности вашего приложения. Фреймворк также построен на React, что делает возможным использование инструментов и библиотек React.

## Минусы использования Next.js

- **Мнение:** Next.js предлагает структурированный способ выполнения задач. Поэтому он может не подойти разработчикам, которые хотят полностью контролировать весь процесс разработки.
- **Управление состояниями:** Управление состояниями представляет собой серьезную проблему при создании приложений с динамическим контентом. Next.js не имеет встроенных функций управления состояниями и полагается на сторонние библиотеки, такие как Redux и MobX.

## Vite против Next.js: Глубокое сравнение



Vite и Next.js имеют ряд общих черт, таких как поддержка современного JavaScript, открытость исходного кода, оптимизация сборки и наличие серверов разработки. Несмотря на то, что оба продукта могут быть использованы для фронтенд-разработки, они различаются по следующим параметрам;

## **Опыт разработки**

**Vite** – это эффективная и быстрая среда разработки. С помощью Vite можно создавать приложения для таких фреймворков, как React, Svelte, Vue.js и Preact.

Этот инструмент имеет интуитивно понятную систему конфигурирования и известен быстрой перестройкой. Vite также имеет отличные инструменты отладки и тестирования, которые делают работу разработчиков более продуктивной.

**Next.js** – это комплексное решение для создания React-приложений. Встроенные функции, такие как генерация статических сайтов и разделение кода, позволяют легко поддерживать код и создавать быстрые приложения. Отлаживать приложения на Next.js очень просто, особенно если в качестве языка выбран TypeScript. Автоматическая генерация маршрутов – замечательная функция, которая может повысить производительность труда разработчиков.

## Рендеринг

**Next.js** позволяет разработчикам на уровне компонентов выбирать, на каком сервере или на клиенте будут отрисовываться их страницы. Маршрутизатор приложений по умолчанию отображает компоненты на сервере. Next.js предлагает варианты рендеринга ‘Static’ и ‘Dynamic’. При статическом рендеринге приложение Next.js по-разному рендерит серверные и клиентские компоненты. С другой стороны, при динамическом рендеринге серверные и клиентские компоненты отображаются на сервере во время запроса.

**Vite** можно использовать для настройки приложений для различных фреймворков. Выбор фреймворка определяет подход к рендерингу, который будет использоваться в Vite. В данном инструменте реализована встроенная поддержка рендеринга на стороне сервера для Vue 3 и React. При использовании Vite в производственной среде целесообразно использовать его в качестве промежуточного ПО.

## Производительность

**Vite** имеет быстрый процесс сборки и сервер разработки. Этот инструмент сборки опирается на собственные модули ES, что обеспечивает горячую замену модулей (Hot Module Replacement, HMR). Эти возможности обеспечивают Vite отзывчивую разработку.

**Next.js** использует рендеринг на стороне сервера, который предварительно рендерит страницы, тем самым улучшая время начальной загрузки. Благодаря функции SSR этот фреймворк демонстрирует высокую скорость работы в веб-

приложениях, обслуживающих динамический контент.

## Гибкость

**Vite** как инструмент сборки может использоваться с Vue.js, React и Svelte. Однако этот фреймворк спроектирован так, чтобы быть легким, но обеспечивать быструю среду разработки. Vite имеет высокую модульность и позволяет разработчикам выбирать только те части или компоненты, которые необходимы им в приложении. Созданное на Vite приложение можно усовершенствовать, интегрировав его с другими инструментами.

**Next.js** разработан как комплексное решение для React-приложений. Этот фреймворк обладает гибкими возможностями кэширования и рендеринга. Next.js позволяет выбирать среду рендеринга (клиентскую или серверную) на уровне компонентов.

## Развертывание

**Vite** поддерживает статический и бессерверный хостинг. При создании веб-приложения с помощью Vite генерируются статические файлы с HTML, CSS и JavaScript, которые можно разместить на статических хостингах, таких как Vercel, Netlify или GitHub Pages.

Приложения **Next.js** поддерживают статическое, серверное и бессерверное развертывание. Для статического развертывания можно использовать Netlify, Vercel или GitHub Pages. Next.js также поддерживает самостоятельное развертывание, для чего можно использовать Docker или Node.js. Для развертывания приложений Next.js доступны такие бессерверные варианты, как Azure Static Web Apps, AWS Amplify, Firebase и Cloudflare Pages.

## Сообщество

**Vite** является более новым фреймворком, так как он был выпущен в 2020 году. Поэтому его сообщество невелико, но продолжает расти.

**Next.js** был создан в 2016 году. У него большое сообщество и обширная экосистема библиотек и инструментов. Next.js также пользуется поддержкой со стороны более крупного сообщества React.

## Когда использовать Vite и когда использовать Next.js



Как видно, и Vite, и Next.js имеют свои достоинства и недостатки. Однако в некоторых случаях Vite может оказаться лучше, чем Next.js, и наоборот;

### Когда следует использовать Vite

- **Нужен быстрый инструмент:** Vite быстро генерирует проекты, поскольку не требует компоновки. Функция Hot Module Replacement (HMR) позволяет

разработчикам видеть изменения в коде без ручной перезагрузки.

- **Хотите получить разнообразный инструмент:** С помощью Vite можно генерировать приложения React, Vue.js, Svelte и Preact. Просто выберите нужный вам шаблон и Vite за считанные минуты создаст ваше приложение.

## Когда использовать Next.js

- **Вам нужен фреймворк с большой экосистемой:** Next.js существует уже более семи лет и завоевал большую популярность. Вы можете использовать множество ресурсов, инструментов и плагинов для интеграции со сторонними инструментами и библиотеками.
- **Хотите получить выгоду от SEO:** Функция рендеринга на стороне сервера в Next.js упрощает обнаружение и индексацию вашего сайта веб-гусеницами. Предварительный рендеринг HTML на сервере повышает скорость загрузки, что сказывается на удобстве работы пользователей.
- **Вам нужен гибкий фреймворк:** Next.js позволяет переключаться между статической генерацией сайта (SSG) и рендерингом на стороне сервера (SSR). Функция SSR позволяет использовать функцию предварительной загрузки, которая подходит для страниц, содержащих динамические данные. Для статических страниц можно также выбрать рендеринг SSG.

## Заключение

Мы рассмотрели все, что необходимо знать в споре между Vite и Next.js. Эти два фреймворка не являются прямыми конкурентами, поскольку Vite – это инструмент для сборки, а Next.js – это фреймворк React. Окончательный выбор фреймворка будет зависеть от характера проекта, который вам необходимо создать, и ваших предпочтений.

### Дата Создания

13.07.2023