

# PHP против Node.js: Объяснение разницы

21.04.2023

PHP и Node.js – популярные технологии для программирования на стороне сервера. PHP – это универсальный язык сценариев для веб-разработки, используемый многими веб-сайтами и приложениями. Между тем, Node.js – это среда выполнения, которая позволяет разработчикам выполнять JavaScript на стороне сервера и отлично подходит для создания быстрых, масштабируемых и эффективных внутренних приложений. Несмотря на то, что это разные инструменты, Node.js и PHP работают в схожих технологических экосистемах и предоставляют схожие функциональные возможности. В результате выбор между ними не так прост. В этой статье вы узнаете о различиях между PHP и Node.js и о том, что вам нужно знать, прежде чем решить, какой из них использовать для вашего следующего проекта.

## История PHP

Созданный Расмусом Лерддорфом в 1993 году, PHP первоначально назывался “Personal Home Page”, а затем превратился в полноценный язык программирования под новым названием “PHP: Гипертекстовый препроцессор”. В первые дни существования Интернета разработчики в основном использовали PHP для программирования на стороне сервера. Однако он быстро стал известен как краеугольный камень веб-разработки, когда был объединен с другими технологиями для создания стека Linux, Apache, MySQL и PHP (LAMP).

Этот пакет программного обеспечения быстро завоевал популярность как мощная платформа с открытым исходным кодом для разработки и развертывания динамических веб-приложений. Влияние LAMP на веб-разработку было огромным. Его открытый исходный код позволил разработчикам свободно обмениваться

кодом и сотрудничать над проектами, что ускорило рост Интернета до огромных темпов. Сегодня PHP используется для всего – от веб-разработки и микросервисов до написания сценариев командной строки. Более 70 процентов всех существующих веб-сайтов созданы с использованием PHP.

## История Node.js

Компания Netscape создала JavaScript в 1995 году для добавления интерактивности веб-сайтам. Он быстро стал популярным выбором для разработки внешнего интерфейса и, как и PHP, со временем превратился в язык программирования. С тех пор он остается популярным выбором для внешней веб-разработки, а также для мобильных приложений и игр. Райан Дал представил Node.js в 2009 году как способ использования JavaScript на стороне сервера, что позволило создавать быстрые, масштабируемые и эффективные внутренние приложения.

С тех пор Node.js стал известен в области разработки внутренних приложений, его используют такие известные компании, как Netflix, eBay и PayPal. Это также открыло двери для разработчиков внешних приложений, которые могли использовать свои знания JavaScript для создания как внешних, так и внутренних приложений с помощью одного и того же языка. Рост экосистемы Node.js привел к разработке множества мощных JavaScript-фреймворков и библиотек, таких как Express.js и Socket.io, что еще больше упрощает создание сложных высокопроизводительных приложений, работающих в режиме реального времени.

## PHP против Node.js: как они сравниваются

Теперь, когда вы знакомы с базовой историей и использованием PHP и Node.js, давайте посмотрим, как PHP и Node.js сравниваются между собой. Мы сравним наиболее значимые

возможности и функциональность каждой технологии.

## **Простота использования**

Как правило, простота использования – это первое качество, на которое обращают внимание разработчики, когда пробуют новый язык или технологию. PHP считается простым в использовании языком для тех, кто имеет опыт работы с языками в стиле C. Он имеет простой синтаксис и обширную стандартную библиотеку, предоставляющую множество функций и возможностей. Разработчики с опытом работы с JavaScript обычно считают Node.js простым в использовании, потому что они могут изучить его относительно быстро. Как и PHP, он также имеет простой синтаксис и обширную экосистему библиотек и фреймворков, облегчающих создание веб-приложений.

## **Производительность**

Выбирая между PHP и Node.js, вы должны обратить внимание на производительность и то, как они обрабатывают запросы. PHP – синхронный язык, то есть он обрабатывает только один запрос за раз. Сервер будет обрабатывать задание до тех пор, пока оно не будет выполнено, прежде чем переходить к следующему запросу. Это может быть неэффективно, если у вас большой объем запросов, которые вы должны обрабатывать одновременно. В отличие от этого, Node.js по умолчанию является асинхронным языком и может обрабатывать несколько запросов одновременно. Когда вы делаете запрос, сервер передает его рабочему процессу и переходит к следующему запросу. Это делает Node.js более эффективным, чем PHP, при работе с большим объемом запросов, так как позволяет серверу обрабатывать несколько запросов одновременно.

## **Документация и поддержка сообщества**

PHP имеет полное онлайн-руководство, содержащее подробную информацию о синтаксисе, функциях, лучших практиках, а также примеры и объяснения использования PHP в различных контекстах.

Помимо официального руководства, в сети доступны многочисленные сторонние ресурсы и учебники. Node.js также имеет подробную онлайн-документацию, которая включает различные руководства с подробной информацией о синтаксисе, модулях, лучших практиках, примерах и вариантах использования. Помимо ресурсов официальной команды Node.js, в вашем распоряжении множество ресурсов и учебников с открытым исходным кодом. И PHP, и Node.js имеют большие сообщества разработчиков и пользователей, активных на таких форумах, как Stack Overflow. Сообщества разработчиков отлично подходят для поиска помощи по конкретным вопросам или для изучения обоих языков.

## Архитектура

PHP лучше всего подходит для традиционных веб-приложений. Он использует многопоточную, блокирующую архитектуру, то есть каждый запрос обрабатывается отдельным потоком. Это может сделать PHP более склонным к узким местам, особенно при обработке больших объемов данных. Node.js, с другой стороны, использует однопоточную, неблокирующую архитектуру, что позволяет ему обрабатывать несколько запросов одновременно, не создавая узких мест. Это делает его более подходящим для приложений, требующих высокого уровня параллелизма и обновлений в реальном времени.

## Примеры использования PHP и Node.js

PHP и Node.js имеют много сходств и несколько ключевых различий. Но, как всегда, главным решающим фактором является то, что нужно вашему проекту. В этом разделе рассматриваются некоторые общие сценарии использования, в которых вы можете предпочесть одну технологию другой.

## Безопасность

PHP – лучший выбор для проектов, требующих строгой

безопасности и соответствия требованиям. Он включает несколько встроенных функций безопасности, таких как санация ввода и кодирование вывода, которые могут помочь предотвратить распространенные уязвимости веб-приложений, например, в приложениях финансовых технологий (fintech), которые обрабатывают и хранят конфиденциальные данные пользователей. Кроме того, PHP имеет долгий послужной список обновлений и исправлений безопасности, что упрощает поддержку безопасного и совместимого приложения.

## **Приложения реального времени**

Для проектов, требующих общения в реальном времени, таких как приложения для чатов, многопользовательские игры или инструменты для совместной работы, Node.js является лучшим вариантом. Он включает встроенную поддержку WebSockets и имеет богатую экосистему библиотек и фреймворков, которые облегчают создание приложений реального времени.

## **Поддержка устаревшего кода или систем**

Как более старая технология, PHP поддерживается на более широком спектре платформ и сред. Это облегчает интеграцию со старыми или унаследованными системами и поддерживает совместимость со старым PHP-кодом или фреймворками.

## **Сильная поддержка нереляционных баз данных**

Node.js имеет сильную поддержку нереляционных баз данных и включает множество библиотек и инструментов для работы с данными, хранящимися в базах данных NoSQL. Это может облегчить создание приложений, требующих гибкого, масштабируемого хранения данных и работы с данными, которые плохо вписываются в традиционную реляционную схему.

## **Широкая поддержка систем управления контентом**

Несколько популярных систем управления контентом (CMS), включая Drupal и WordPress, построены на PHP, что делает его очевидным выбором для проектов, в которые необходимо интегрировать CMS. PHP позволяет вам интегрироваться с CMS или настраивать ее в соответствии с потребностями вашего проекта.

## **Поддержка современных функций Javascript**

По той же причине, по которой PHP лучше всего подходит для работы с сайтами WordPress, Node.js по своей сути предназначен для поддержки новейших фреймворков JavaScript. Node.js построен на базе JavaScript-движка V8 и регулярно обновляется последними функциями и улучшениями.

## **Преимущества и недостатки PHP и Node.js**

На данном этапе вы должны иметь представление о том, какая технология лучше подходит для вашего проекта. В заключение приведем основные сильные и слабые стороны двух технологий, которые мы только что обсудили.

### **Преимущества PHP**

- Несколько встроенных функций безопасности
- Регулярно обновляется новыми функциями и улучшениями
- Надежная поддержка CMS
- Сильная поддержка фреймворков для веб-разработки, веб-серверов, операционных систем и хостинговых платформ
- Широкий выбор инструментов и библиотек для веб-разработки

## Недостатки PHP

- Менее гибкий и динамичный, чем другие языки, такие как Java или C++
- Более низкая производительность для определенных типов приложений
- Не подходит для приложений реального времени или параллельного программирования
- Глобальное пространство имен усложняет управление и организацию больших кодовых баз PHP
- Относительно простая объектно-ориентированная модель, что может сделать его менее мощным и гибким, чем другие языки, такие как Java и C++.

## Преимущества Node.js

- Эффективная производительность
- Динамичность и гибкость для параллельного и параллельного программирования
- Front-end JavaScript разработчики могут легко освоить Node.js для создания своих back-end, что делает их более продуктивными за минимальное количество времени
- Встроенная поддержка websockets для создания приложений реального времени
- Хорошо подходит для создания микросервисов и бессерверных функций

## Недостатки Node.js

- Не подходит для веб-приложений, требующих отрисовки сложной графики
- Меньше встроенных функций безопасности
- Меньшая совместимость со старыми или устаревшими системами

- Отсутствует надежная поддержка CMS

## Какой язык лучше?

В конечном счете, и PHP, и Node.js отлично справляются со своими задачами, но при выборе между ними необходимо учитывать множество факторов. При создании приложений в реальном времени лучше использовать Node.js. В то же время PHP – это лучший вариант для интеграции или настройки любой популярной CMS для вашего проекта. Оба инструмента имеют низкий барьер входа, оживленные сообщества и легкодоступные учебные ресурсы. Это означает, что независимо от того, какой из них вы выберете, у вас будут все необходимые ресурсы, чтобы начать создавать успешные приложения.