



Node.js vs. PHP: A Relative Study With all the Useful Aspects



PHP and Node.js are the highly endorsed back-end technologies leveraged in web development endeavors. Both have participated in and performed server-side development, and Node.js even serves both server-side and client-side development.

PHP has been assisting developers for three decades, and it even now empowers most of the web. Node.js is comparatively new in contrast yet growing quickly and turning a popular selection due to its full-stack development abilities.

Even though both technologies were launched at different times and in diverse tech scenarios, there are some prominent resemblance factors between PHP and Node.js. The Node.js vs. PHP comparison is about benefits, disadvantages, performance, coding, database, community support, and other practical aspects such as when to select PHP over Node.js and vice versa.

Why Use Node.js?

As per Stack Overflow's 2021 Developer Survey, [33.91% of all respondent developers](#) said they leverage Node.js for all-embracing development in 2021, and it is predicted the number will increase in the coming years.

That is because Node.js delivers an inclusive series of features such as full-stack JS, Non-blocking code, swift and Scalable, strong back-end connection, and a massive ecosystem with worldwide user base support.

Who Uses Node.js?

Since its launch in 2009, Node.js has created a significant role for numerous business and industry leaders. Today, all sorts of businesses, from startups to enterprise companies, depend

on Node.js.

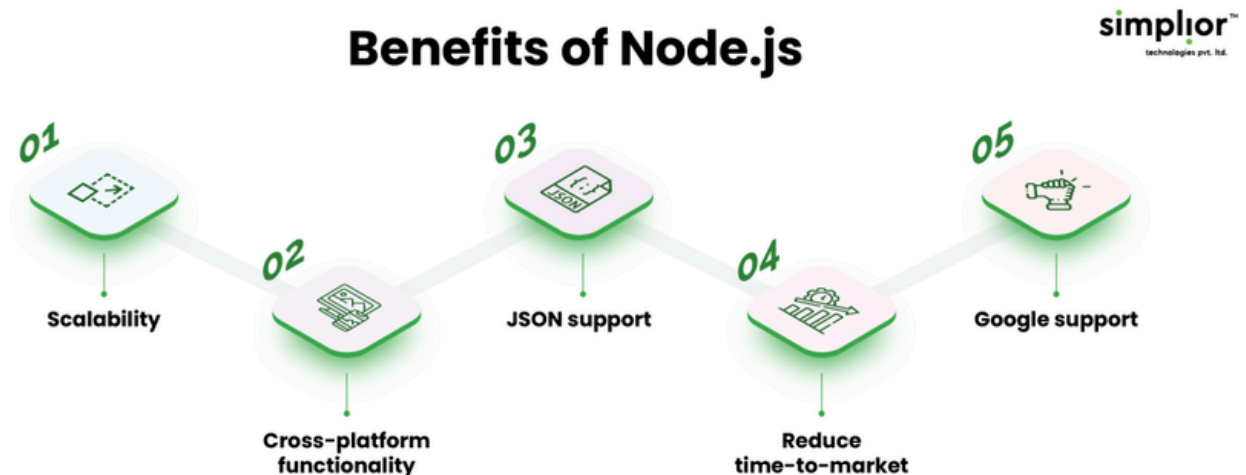
Let us get to know some of the industry's known names that leverage Node.js:



Benefits & Disadvantages of Node.js

In the current scenario, Node.js is one of the most extensively utilized software development technologies due to its front-end and back-end development and cross-platform abilities.

Benefits of Node.js



1) Scalability

Node.js modules offer load balancing over multiple CPU cores, making it straightforward to accomplish desired outcomes through smaller-sized modules without wearing your RAM. Besides, Node.js enables a non-blocking event-loop architecture that delivers outstanding scalability and entitles the server to perform requests in real-time.

Moreover, Node.js assigns apps to numerous teams and facilitates you to state and allot tasks concurrently. As a result, you get a precise, higher-performing framework leveraging fewer

resources.

2) Cross-platform functionality

Programmers can blend NW.js and Electron to build real-time cross-platform applications and don't require to write diverse code for numerous platforms.

Consequently, you don't require to be anxious if your programming team lacks a thorough understanding of C# or Objective-C, Node.js can craft native apps without them. Likewise, Google tools can be exceptional support for programmers for coding and debugging.

3) JSON support

Back-end technologies like PHP and Ruby on Rails use JSON format for interaction. Node.js does the similar stuff without translating amongst binary models and rather smartly leverages JavaScript.

So, crafting RESTful APIs for No-SQL database support turns straightforward, like MongoDB. Additionally, the JavaScript environment offers simple data transfer connectivity with data transfer benchmarks, boosting effectiveness.

4) Reduce time-to-market

As Node.js is full-stack and swift, you can reduce the total programming time. Besides, your development team has access to a series of Node.js libraries, ready programs, and precise assets from GitHub, which lessens development time and challenges. As a result, you are more likely to reduce your time to market.

5) Google support

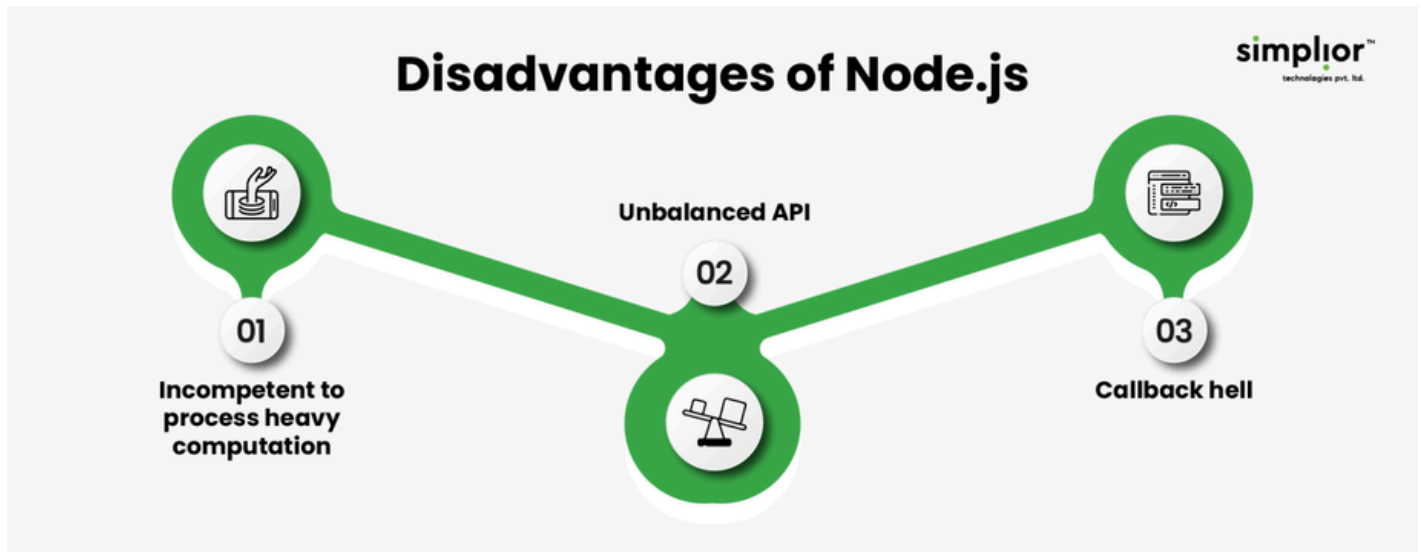
Node.js is produced on Google's V8 JavaScript engine and works as both front-end and back-end development tools. Likewise, the Google V8 engine enables Node.js to operate multiple JavaScript scripts in the front-end simultaneously. Consequently, it's straightforward to find faults in the front end and the back-end while dealing with Node.js.

Google lists Node.js as one of the swiftest and most productive runtime system engines. Also, Google Cloud provides JavaScript programmers tools to craft cloud-native apps. You can leverage SDKs, in-IDE assistance, and native runtime backing on Cloud functions for a

quicker application advancement procedure. Moreover, Google Cloud can execute your app from the initial stages to the end.

Disadvantages of Node.js

With benefits, Node.js even has some drawbacks.



1) Incompetent to process heavy computation

Node.js doesn't back multi-threaded development. Nonetheless, it can manage suggestively more decent apps than Ruby, even if it is not envisioned executing long-running and heavy calculations.

Equally, multi-threaded intensive computations can reduce productivity and performance levels. Henceforth, Node.js is apt for single-threaded multifaceted systems; otherwise, it will consume unused processing time and power.

2) Unbalanced API

The Node.js Application Programming Interface (API) deviates frequently and stays disconcerted. Besides, with each fresh API release, numerous backward-incompatible alterations force the programmers to change their access code to cope with the newest version of the NodeJS-based APIs.

3) Callback hell

JavaScript deeply counts on callbacks due to its asynchronous nature. Consequently, a function is allocated that performs the tasks one after one from an extended queue. Every queued activity in the background with callbacks generates an immense callback called “callback hell.” It trims down the overall quality of the programming, and as one callback is nested within another, callbacks in numerous levels lead to a tough coding scenario.

Why Use PHP?

As per Stack Overflow’s 2021 Developer Survey, [21.98% of respondent developers leveraged PHP](#) for heavy programming in 2021. Many web specialists projected that PHP would be a dead development language; however, it withstood the critical test of time.

A substantial number of the user base are still depending on PHP and some of its exceptional functionalities, which comprise server-side scripting, cross-platform access, web compatibility, database connectivity, and HTML-friendly features.

Who Uses PHP?

PHP owns the wide-ranging online world. It empowers more than 78% of website portals. There is no debate that PHP’s usage in the development space stays on a higher side. Also, it is still a significant technology for some of the largest worldwide technology organizations.

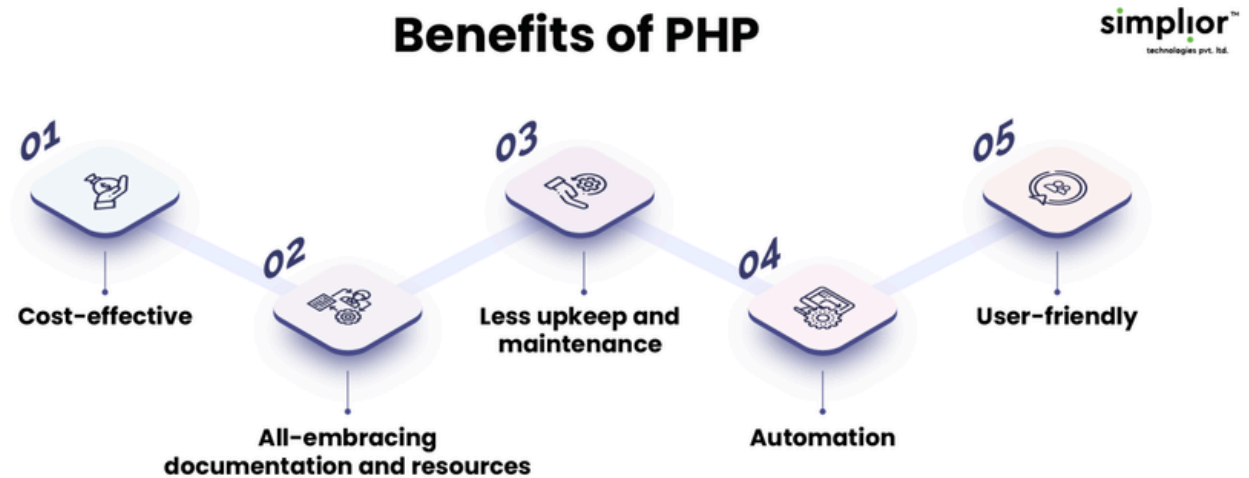
We have listed some of the topmost organizations that still depend on PHP:



Benefits and Disadvantages of PHP

Like with any programming language, PHP has a series of benefits and drawbacks, despite its supremacy in the marketplace. We will first explore the PHP benefits.

Benefits of PHP



1) Cost-effective

PHP is a free and open-source development tool. Numerous frameworks, comprising WordPress, Drupal, Laravel, and CodeIgniter, augment PHP's functionalities and make for a fitting development environment.

2) All-embracing documentation and resources

You can access several tutorials, manuals, references, and demo projects around PHP programming. Whether you are a novice developer or an expert, these materials will back you when you are stuck.

3) Less upkeep and maintenance

PHP frameworks back the model-view-controller (MVC) architecture and ease web app development and sustenance. For instance, you can leverage MVC architecture to divide a web app into diverse models, views, and controllers. You may even utilize a PHP MVC framework to detach the app's UI and business logic.

4) Automation

You can leverage PHP bindings for programmer browser-based automation test setups. The framework is flexible, and the Selenium web driver performs competently in all significant browsers.

Additionally, PHP frameworks' functions and tools assist you in automating your web development activities such as caching, authentication, session handling, and URL mapping without costing you in competencies.

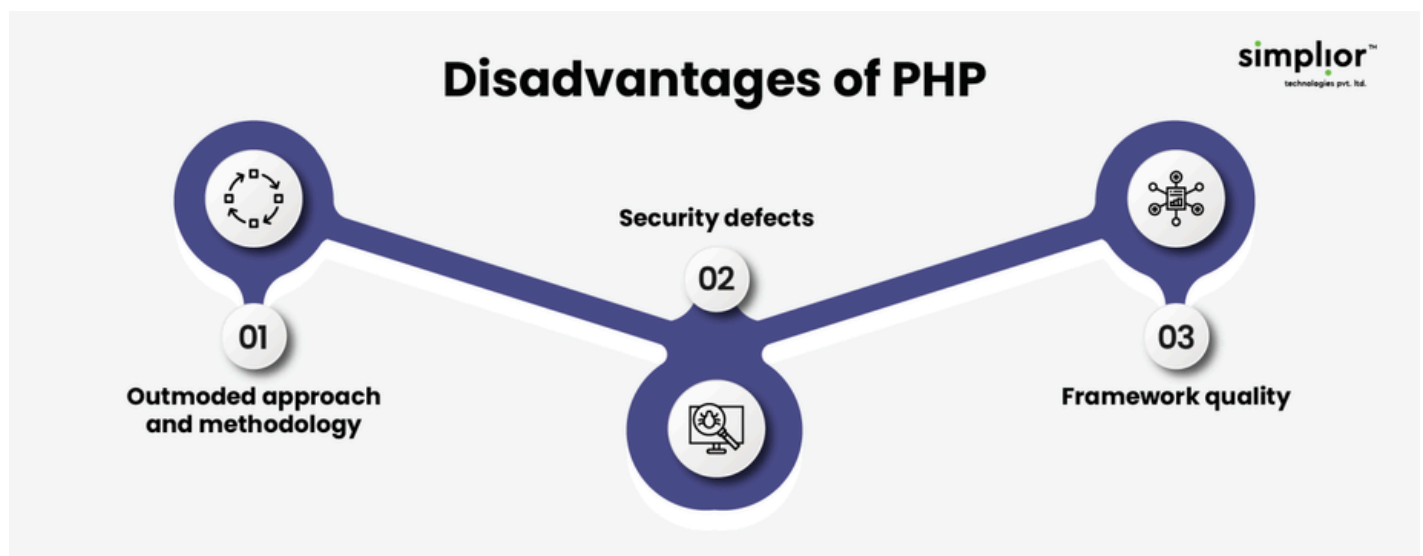
5) User-friendly

PHP has a quicker learning curve, which you can prevail rapidly. If you know a principal language like C, you can swiftly catch on to PHP for app programming. Even if PHP is your foremost development language, you can enable PHP on any OS or install it on servers with minimal effort.

Most user bases choose PHP because it is easy to learn, easy to work with, and extensively used. Many development languages are complex and scary to learners, but PHP is superiorly organized, clear, and seamless.

Disadvantages of PHP

With benefits, PHP even has some drawbacks.



1) Outmoded approach and methodology

Machine learning is advancing in popularity and use. Also, languages such as Python are more appealing to the user base.

But though PHP has its libraries, it cannot contest Python's Scikit-learn, Theano, and TensorFlow libraries to craft web applications enabled by machine learning or comparable

amendment of its legacy methodology.

As an outcome, PHP is not the precise development solution if your application will need advanced technologies like machine learning.

2) Security defects

PHP has a negative reputation for security due to the low access fence for new developers. Less experienced programmers craft websites and applications with probable vulnerabilities.

Subsequently, these results often lead to PHP's image as a less safe and secure development language. Nevertheless, if you comprehend PHP and are a proficient developer, you will receive robust web security levels.

3) Framework quality

Most prevalent PHP frameworks are open source and free to leverage. So, web programmers utilize these frameworks without increasing overall project costs.

But not all PHP frameworks provide similar strengths or functionalities in the community and appeal to many programmers. As an outcome, some PHP frameworks lack precise support and accessible resources.

Node.js vs. PHP: A Head-to-Head Assessment

#1 Coding

If the syntax is simple and better engineered, you will be capable of doing more with minimal code.

- **Node.js**

It seems that Node.js needs more lines of code to perform the same functionality as PHP. Appreciatively, you essentially don't need to remember the syntax of diverse languages while coding as the complete development setting is developed with JavaScript.

So Node.js may sometimes need comparatively longer lines of code, but it runs with merely one language.

- **PHP**

It may seem from the outside that PHP needs lesser lines of code to execute functionality and is thus simpler. However, then you require to be fully assured of how operators and generators perform in Apache HTTP Server, MySQL, Linux, and PHP itself.

#2 Swiftness

This metric indicates the performance and rapidity intrinsic to both leading technologies. Higher speediness of execution means a swifter development procedure and more gainful projects.

- **Node.js**

Node.js is amongst few asynchronous development settings and environments. This, being a significant discriminator, offers Node.js a competitive edge over other development language. Being asynchronous means that it requires not waiting for any modules to be effectively performed prior to loading the subsequent one. Such performance radically trims down the downtime for the web application and makes for a unified user experience.

- **PHP**

Like most conventional technologies, PHP runs on synchronicity. That means each module is performed in the code-specified order. If one of the modules is not performed well, the substantial ones will not initiate until it is thoroughly completed.

However, if you still want to continue successfully with PHP, you can pair it with HHVM Virtual Machine leveraged to perform and execute PHP web applications.

It can increase the performance levels of your PHP web application. Even so, Node.js is still a quicker substitute.

#4 Frameworks

Frameworks assist you in concentrating on the fine details of your project as the most frequently utilized code is offered with libraries, functions, and APIs. The more widespread the framework, the lesser the redundant code you require to create.

- **Node.js**

With frameworks such as Derby, Express, Meteor, and Sails, Node.js has seen an immense extension of its libraries in a comparatively brief time. While these frameworks can increase performance, trimming down development time and resource usage, PHP's frameworks beat those of Node.js by an enormous margin.

- **PHP**

PHP has a superior library of frameworks with niche marketplaces of its own. Many development companies focus explicitly on forms of PHP frameworks such as Laravel, CodeIgniter, Phalcon, and CakePHP.

#5 Database

Here, we are comparing how suitably the technology is able to connect with the database along with the category of the database. Traditional databases enable the storage of the data in a navigational or hierarchical setup.

- **Node.js**

Since Node.js has matured with the arrival of NoSQL databases, it is highly synchronized with all structures of databases.

With JavaScript Object Notation, an exclusive format of data leveraged while engaging with databases – Node.js can work successfully with NoSQL databases such as CouchDB and MongoDB. It even backs graph databases and has considerable libraries to better access SQL databases.

- **PHP**

Being the old technology, PHP is developed to work with conventional and relational databases such as MySQL and MariaDB. While it is probable to precisely import libraries to work with NoSQL databases, the procedure is monotonous and consumes a large amount of processing time.

#6 Modules

Modules possess a specific functionality set and are frequently interchangeable, offering the program its precise 'modularity.'

- **Node.js**

Some PHP supporters often criticize that the Node.js projects are not as steady with modules as their PHP complements. As the community is still mounting, quality-control systems are hardly in place.

As a portion of the control, the Node.js package registry has acquainted the npm-audit resourcefulness. With the assistance of this, you can now check every package and module for nasty code.

- **PHP**

Some PHP supporters also have articulated annoyance toward receiving lackluster projects. This has provoked them to migrate to other platforms specifically because the inherent functionalities and modules of PHP can be easily imported.

#7 Community backing

The power of a community decides what sort of updates come to numerous frameworks, libraries, and projects introduced with respect to every technology.

- **Node.js**

As Node.js is comparatively new, its community is smaller equated to PHP. Yet the projects backed and built by the community lean more toward the programming requirements of today. Most of the projects focus on adding exclusive features to Node.js rather than working as a library to import functionalities from other development languages.

- **PHP**

PHP has a large-sized community and extensive projects under its belt. However, one significant challenge here is that the latest projects built by the community merely appear monotonous when equated to Node.js.

At first, the projects they crafted were received well for adding to the platform functionalities already in other languages. However, those multiple similar projects have now led some PHP supporters to feel the requirement for more thought-provoking projects to come their way.

#8 Request Handling

This functionality is the measure of how rapidly a technology processes and enables client-side requests. Performing the request is undoubtedly critical, but it also must be executed leveraging minimal resources and time. Error handling would assist us in understanding which technology best optimizes the user experience.

- **Node.js**

Thanks to the asynchronous processing of numerous requests, the system isn't delaying other processes to end prior processes to the subsequent one starts. CPU and RAM times are also not unused. The mere concern here is that if an unanswered mistake of one request isn't enabled in a decided time, it could delay other requests generating a pan-system fault.

- **PHP**

PHP manages a single request at a time. Unsurprisingly, it eats more resources on the CPU and the RAM. Meanwhile, it is on the contradictory side of the gamut with this constraint, and each request is performed in a silo.

This scenario averts cross-contamination of request errors but also generates a lag in the system with its one-request-at-a-time tactic.

Node.js vs. PHP: Side-by-Side Evaluation Table

Let us explore a side-by-side assessment of Node.js vs. PHP and inspect some of the core programming parameters:

Node.js vs. PHP

	Node.js	PHP
Category	Javascript runtime environment	Programming language
Programming	Long	Concise
Engine	Google's v8 javascript	Zend engine
Package manager	Composer package manager	Node package manager
Execution speed	Swifter and light in weight	Slower compared to node.js
Execution	Asynchronous	Synchronous
Concurrency	Event-driven non-blocking i/o	Multi-threaded blocking i/o
Performance levels	Quicker	Slower
Database	Any of the databases	Relational and conventional
Community	Small-sized but dynamic	Massive community

When to Select Node.js Over PHP?

If you are exploring the following functionalities, Node.js can offer optimal value to your application development projects:

- **Development Proficiency**

If you are building a dynamic single-page app in Node.js for simplicity of development and optimum productivity.

- **Higher Speed and Reliable Callback from Servers**

Web apps built leveraging Node.js tend to perform healthier while dependably sending requests to the server. Its asynchronous architecture enables non-blocking execution which is effective for any project that desires swiftness.

- **Real-Time Data Sets**

As the data-transmission rapidity from the server to the client-side application is optimum, Node.js is seamless for web applications that require real-time data functionality.

When to Select PHP Over Node.js?

PHP would be the better technology for your development stack if you deal with the following requirements for your web application:

- **Centralized Server and No Scaling Desires**

If your business plan is to assign a precise centralized server to your web application that is not scaled across an extensive series of servers, PHP is the ideal addition to your technology stack. Also, you can leverage it with Linux, Apache, and MySQL.

- **Need Portability**

While PHP certainly puts a tab on the servers you can efficiently connect with, it also offers you portability amongst servers. That is why you can practically port your web applications to any server with Apache, IIS, and database-backing functionalities.

Also, if you leverage PHP with content management systems (CMSs) such as WordPress or Drupal, you can have your website portal up and operate in no time.

Key Takeaways

There is no benchmarked procedure to detect if Node.js is healthier than PHP or vice versa. However, rather than attempting to measure which technology outperforms the other, it would serve you well first to map out your project needs and then choose the technology that precisely fits it.

In addition to technology significance, also pay superior consideration to the resources you possess. If your team is more capable of one technology over the other, it is sensible to inquire if the proficiency can pay off for the probable downsides of the technology.

If you are exploring opting for Node.js or PHP technology in your next application development project, in that case, you must hire our developers from Simplior to leverage all the needed functionalities and improve your services with [Node.js or PHP development](#).

This Blog is originally published on : [Simplior Technologie's Blogs](#)