Node.js, Why?

Intro

At this point in the current world, especially in these past few months, the internet, more specifically, web sites have gone from being a niche thing that most people would never have heard of with clunky interfaces to now being a what is basically the main way the world currently operates with sites designed so well that even a toddler or an elderly person would find no hassle in traversing.

This has all become a strange reality and it took basically a single generation to go from no internet access to everything being accessible on the internet. This is in large part to the web development that has gone on behind the scenes over the past two decades or so mainly using things such as JavaScript to build something that everyone could use, operate and now even do their daily work with. This was by no means an easy task, especially as many who wrote the code for these websites had to think against their own savvy ideas of what may be right or wrong and what is necessary and is not as they weren't designing for themselves, it was for a regular person who more often than not did not possess the same level of knowledge that they had, they wanted to simply do their tasks and probably didn't think twice about what went on in the back-end that made this site so functional.

Web development especially has come on leaps and bounds over the course of the past couple of decades. The first example of websites being put in place for general public use would be awful by today's standards. This changed slightly when JavaScript was first introduced in the mid-nineties, but it was not until the next millennium where things really started to take up shape. Coding through JavaScript or HTML became more and more common and people did not need to be a total expert to create their own applications or websites. When the advent of the new century it brought internet usage far more into the mainstream and meant that everyone could use the internet, however this also obviously meant that web development became an even tougher task as you had to please a whole new set of people and consumers even further down the line. This meant that developers had to move on from older technologies or methodologies that may have been fine previously but were now considered outdated. This meant that if you learned and practiced one way, you were forced to adapt or else be left behind. Many were left behind as concepts and functionality quickly fell down the pecking order to what was now acceptable as the minimum requirement for a website, text boxes and buttons were no longer just acceptable, there had to be some more substance than that to survive in this new digital age on the internet.

The Beginnings of Node

Now onto what this means for Node.js, well firstly I will explain what it is. **(1)** It is a JavaScript runtime environment designed to build scalable network applications. It is also cross platform and open source working on the server-side meaning that you can execute JavaScript code on your machine instead of having to run it in browser like others. **(2)** It is relatively new in terms of the history of computers and the internet, created in 2009, for reference JavaScript itself is over twenty years old at this point so it's worth noting that there were many alternatives for many years before it's inception. However, since it was created it quickly became very popular over the years and has over **(3)** one billion downloads and many of the worlds biggest entities such as **(4)** NASA, Twitter and Intel all using it in some way or another to help in their day to day use. It, in turn, is also obviously very popular with the general public as in a survey taken **(5)** 51.4% of programmers use it ahead of .NET, .NET Core and Pandas, while **(6)** 67.7% of web developers preferring JavaScript for web development, over others such as HTML, SQL, Python or Java.

Node is not totally perfect

It goes without saying though that Node.js is obviously not perfect, of course it has it's issues and yes it is used by a perceived majority but there is still roughly a third or more of people who don't use it for whatever reason. Perhaps a person would prefer to use something that is multi-threaded or something that can handle CPU intensive tasks far better? After all, with Node it is not suited for large scale applications and it is probably not the ideal choice as it can quickly become too much to deal with. Similarly, with tasks that require the CPU it may run in to issues with the Event Loop doing too much and it would severely slow the whole thing down. This would obviously mean a poor experience for someone using Node to make any sort of large project as it would run into issues immediately. On top of this it can also be inconsistent with some unwanted changes and updates meaning you would have to fix some problems that previously was not even an issue, also having issues with support for other relational databases.

Why should you switch to Node?

Even with all that though there are some major advantages to using Node. As I mentioned previously it is open-source and uses JavaScript as it is language. This opens it up to the vast majority of people interested in using it for Web Development, it's open-source meaning that any person can use it on any system for free, like Windows, Mac and Linux, with JavaScript being used it also means that it's very accessible due to the fact that you don't need to learn any new language to be able to use it from the front end or the back end, meaning it's all very easy and approachable to get into even for a beginner. On top of this it's also asynchronous, meaning the multiple operations you may be running can all be used at the same time and nothing is being blocked off, it also has the benefit of a great community behind it which obviously helps as if someone were to be having issues with something involving Node there is thousands of other people who use it that contribute throughout the internet on things such as GitHub or Stack Overflow to help out with these kinds of issues. This shows a great many advantages that, although may be somewhat present with others, are a really vital part of what makes switching to Node a much easier pill to swallow as there is a support system in place for any teething problems that may occur for either new or experienced users who've made the choice to migrate to Node.

There are also many other advantages that can be said for Node such as it being generally fast due to it's v8 engine developed by Google for Chrome but many other advantages can also be seen as more of a choice of preference for a developer and other options may be preferred because of that. For many it's ideal to start off with Node as a first choice as it has many desirable aspects and advantages associated with it with not many major drawbacks unless you're aiming very large but for a beginner this obviously shouldn't be the aim. Like I said for a beginner there is no need to learn anything other than JavaScript which can be a massive bonus as a project may be daunting enough as it is, never mind when you'd need to learn a whole new language just for it such as HTML for example.

For more experienced people who perhaps are more used to other languages or environments it still may not even be a bad idea to alternate and even change fully over to Node. After all, as I previously mentioned it is among the majority when programmers were surveyed on what they use and what language they use and there have been over one billion downloads to go with that too. Obviously, it is an individual's choice but with so many already choosing it in such a relatively short life so far then surely that shows something for its quality. These people may well be used to something different, but the industry of Computers and the Internet has generally been about advancing and evolving past what was used previously, it's why new parts are made for computers every year, why software is being constantly updated and why websites generally refresh their looks and functionality repeatedly, it's all adapting to something newer which is basically what switching to Node is like.

Of course it's not ideal in every single case but nothing is, large scale projects and things like that are probably best off kept elsewhere as Node won't be able to handle things like that but

I would imagine the majority of cases won't have that issue and that is arguable Node's biggest flaw so if you can overcome that there's generally not too much else wrong with the choice aside from some minor quality of life things you may or may not run into, that mostly depends on the person and what they want from the environment themselves.

Conclusion

In conclusion, Node.js is a great choice to go to for most people. It is still in its infancy in terms of age and if you examine the timeline of events it has really grown a great deal in the past decade and has been routinely updated in the past few years especially, showing the support it has from it's creators but also from the community behind it who are also using it day in day out from small scale single person projects to massive global entities also using it within their own infrastructures to great effect and success. I hope I've shown that while Node can be flawed in some aspects it can be used to great effect when implemented in the appropriate scenarios and many people are either considering or even have already chosen to move their environments over to Node and are clearly enjoying the experience thus far. Obviously people should use the environment that they feel the most comfortable with and what they know best but with some of the largest forces using it regularly versus alternatives I think it is fair to say that Node's popularity will probably only grow and if you were going to aspire for a career going forward with Web Development it would perhaps be wise to get the most acclimated to what could be the most used and most common environment and writing with JavaScript.

Until another competitor comes along that can perhaps offer something that could support more large scale projects, which Node isn't suited to, while also offering the same sort of functionality that Node does, I don't think that it will be a viable option to go with a different environment. As I pointed out though, tech is always evolving and a competitor could arrive at any point, however with Node itself being so relatively young it is hard to see any time soon.

References

(1.) Node

https://nodejs.org/en/about/

(2.) Node

https://nodejs.dev/learn/a-brief-history-of-nodejs

(3.) Node, 2018

https://nodejs.medium.com/more-than-a-billion-downloads-of-node-js-952a8a98eb42

(4.) Karolina Gawron, October 2018

https://www.monterail.com/blog/nodejs-development-enterprises

(5.) https://www.statista.com/statistics/793840/worldwide-developer-survey-most-used-frameworks/

(6.) https://www.statista.com/statistics/793628/worldwide-developer-survey-most-used-languages/