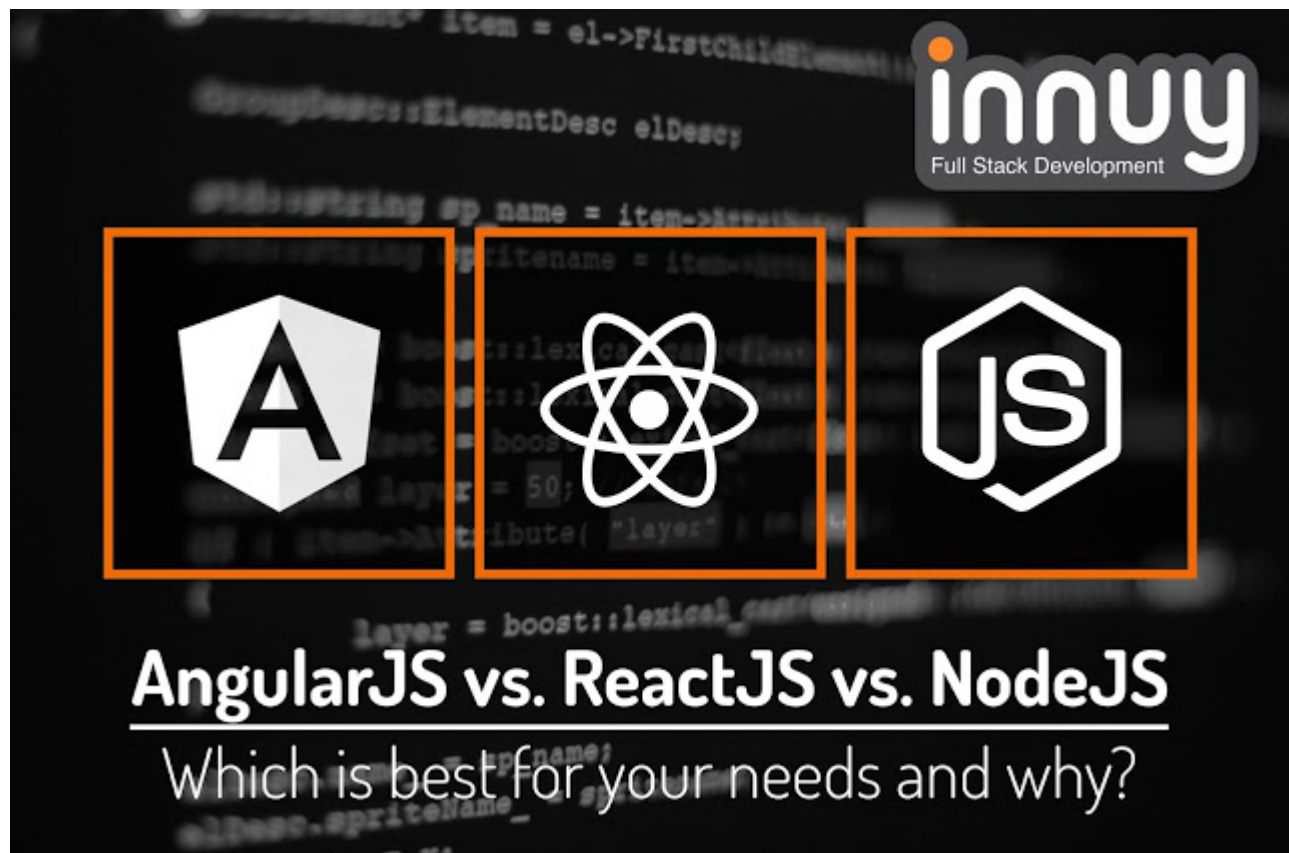


AngularJS vs. ReactJS vs. NodeJS: Which Is Best For Your Needs and Why?

on August 26, 2019



It's safe to say that Javascript has firmly taken its place as the de facto language of the Web. It's easy to learn, easy to use, and offers many opportunities for creating reusable libraries and frameworks to help speed up development and share resources. [AngularJS](#), [ReactJS](#), and [NodeJS](#), arguably the most popular frameworks, all rely on Javascript yet are functionally different – they each bring a certain set of strengths to the Web development arena to help companies launch fast, scalable apps and Web sites.

Which of these three is right for your needs, though? It's important to become familiar with each framework's capabilities and how they can work for you before you hire a company to develop your app. So let's take a high-level look at the major characteristics of AngularJS vs. ReactJS vs. NodeJS to help you figure out which one is best for you.

AngularJS



[AngularJS](#), is an open-source framework developed by Google and launched in 2009. The reasons it has become a leading Javascript tool, with a huge developer base, include the following:

- Development is simple and easy because AngularJS simply extends your HTML client-side view. Your HTML attributes dictate which Angular controllers are used for which elements.
- Angular's module-based development model is fantastic for single page applications. Developers can organize their code in separate Javascript files, keeping that code clean and neat.
- Automatic data synchronization is possible with Angular, which allows real-time testing throughout all stages of development using mock data.

AngularJS is best for you if:

- You're building fast, complex, and rich user interfaces
- You need clean, reusable code
- Your app needs to be tested and brought to market quickly

ReactJS



[React](#), is technically not a framework, but a library of Javascript tools and components. It's a favorite of gigantic Web sites like Facebook that need to display huge data records quickly.

- React's components are translated into design elements that can be used over and over again for new apps or interfaces.
- It's virtual DOM refreshes only the critical part of a page displaying a data set, rather than the entire page at once, making response time to user actions lightning fast.
- This in turn make a Javascript site uniquely SEO-friendly because the virtual DOM returned by React on the server side appears to the client side as a regular old HTML Web page.

ReactJS is best for you if:

- You need to build multiple apps quickly or want a plug-and-play library of design elements that can be used across your suite of apps
- You have a site where load speed is a critical part of your business model

NodeJS



[NodeJS](#) is technically considered a runtime environment rather than a framework. It's powered by the Google Chrome Javascript engine and can run server-side on various platforms including Windows, Linux, Unix, etc.

NodeJS is best for you if:

- You want to use Javascript on both the front and back end. Node.js on the server-side can listen to HTTP requests and directly read and write to a database. It also reduces the need to duplicate code on both the client and server sides.
- You will need the ability to scale your apps quickly and easily as your needs increase.
- You have a need for real-time video or audio streaming/encoding. Node's data streaming capabilities allow you to do this with no hit to performance.

As you can see, each of these frameworks is a solid choice, depending on what you need from them. It's important to have a clear understanding of your project, goals, and the technology at your disposal before you select the framework best for you.

If you need more guidance, however, [call us!](#) We can do a thorough exploration of your needs and make the best framework recommendation to meet them.

