The Languages And Frameworks That You Should Learn In 2015

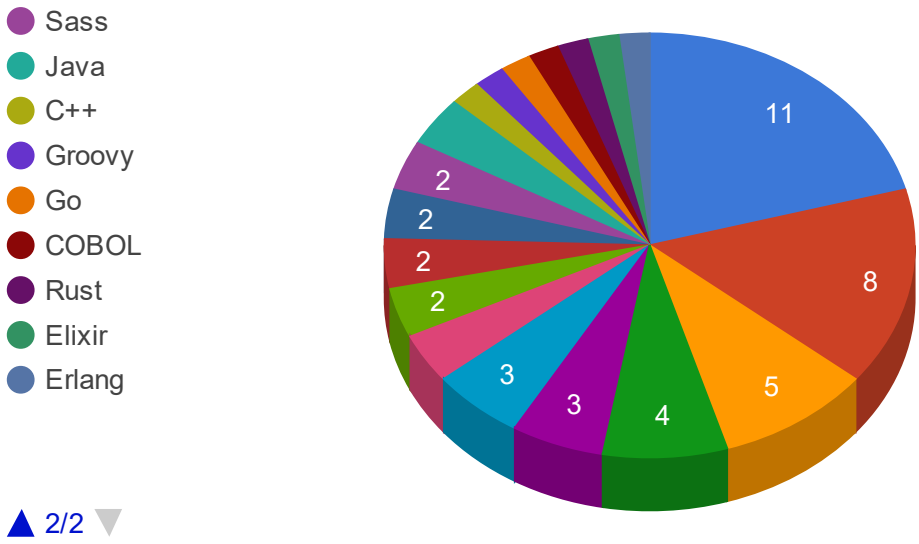
Martin Angelov December 16th, 2014

Last week, we asked you which were the languages and frameworks you were looking forward to learn in 2015. 47 of you replied, and here are your answers.

Languages / platforms

Node.js is the winner here, with PHP second, and JavaScript – third. There is an amazing community forming around Node.js, so it isn't a surprise that a lot of you are excited about it. If you know

JavaScript, you are already half the way to building web apps in Node.js.

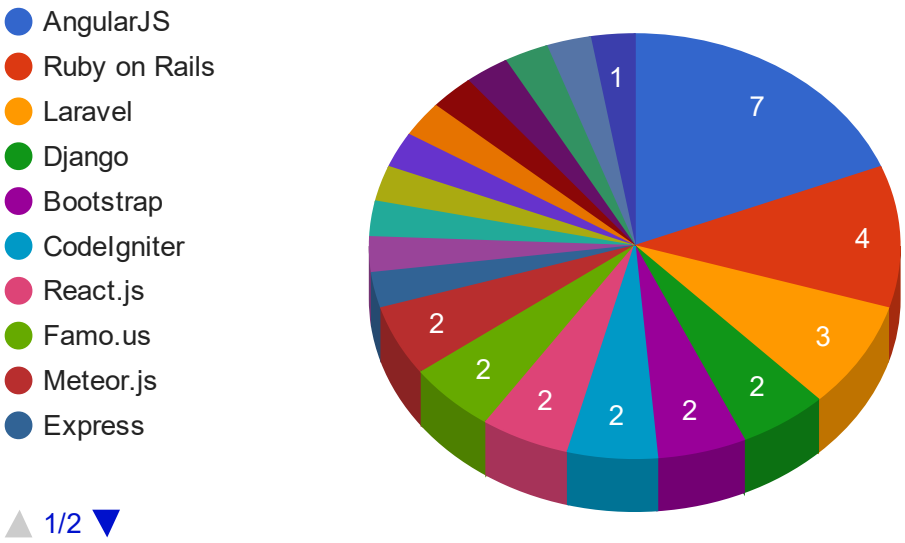


This is only half of the picture. Here are the frameworks that our readers are looking forward to picking up.

Frameworks

AngularJS takes the lead in the frameworks category. Large companies and enterprises have adopted Angular, which drives the demand for coders skilled in the framework. The fact that it is backed by some of Google's best engineers also helps it reach the top spot.

This is important to consider since if there is alot of support for the framework, then we should use it.



To help you make your choice, we've prepared an overview of the above technologies and more!

Here is what you should learn in 2015

Libraries and frameworks come and go, so it is risky to put the effort to learn every new thing that comes along. But here are our suggestions for languages and frameworks that we believe will stick around in the long run and are worth learning. They are all popular, have large communities, and give a lot of career opportunities.

1. JavaScript is everywhere

If you are doing web development, JavaScript is a language that you should know, regardless of what other language you write your backends in. These days you can use JS in the browser, on the server, in mobile apps and even on programmable hardware. ES6 will bring much needed improvements and will make the language even more powerful and easy to write. It is also a good idea to learn about Bower and npm, and also tools like jshint and jscs for code style and issue reporting.

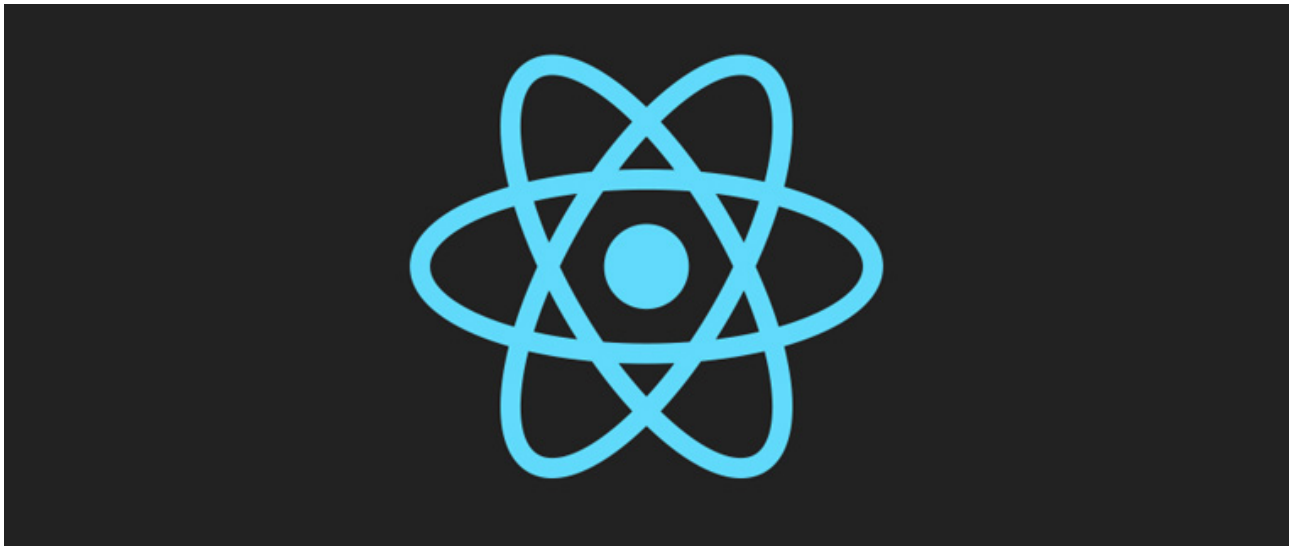
Probably an important tool, especially to make the front end as responsive as possible.

2. AngularJS

Good to note since a developer that uses this tool can deliver content really quickly. This is great for the client and for our design since it is easy to use. Furthermore, it is well documented, which is why we consider using it.

AngularJS is a JavaScript framework by Google, which quickly established itself as the enterprise way of building powerful web apps. With that recognition comes an increased demand for programmers experienced with the framework, and as a result, you will find it mentioned on the list of requirements of nearly every JavaScript-related job ad. But don't jump into it just yet. It has been said that a large rewrite and rethinking of Angular is coming soon, so it would be better if you wait until version 2.0 is released before picking it up. See our guide with angularjs examples.

3. React



React is the newest entrant in this list, but it proved itself a practical realization of the idea for reusable web components. The library is developed by Facebook and provides very fast performance thanks to its virtual DOM, and can be easily plugged into existing projects. It also has a very active community

that develops all kinds of components. In our opinion, React has a lot of potential and is the framework to watch (and learn) in 2015. See our quick react tutorial.

Having reusable web components is important because it can make the website look as nice as possible and consistent.

4. Node.js

With Node.js you can develop networked server applications in JavaScript. It can be used for simple website backends using a framework like Express, API endpoints, websocket servers or even torrent clients. Node has an incredibly active community and surpassed every other language by module count this year. If you are a beginner, we recommend trying some of the interactive tutorials at NodeSchool.

5. NoSQL databases

These databases are really good and really quick to learn and they make more sense than the SQL since it is more object oriented. I would recommend we use this for our design.

Databases which need neither tables nor SQL are highly valued by today's web developers and we believe these databases will only become more popular next year. The two noteworthy choices are Mongodb and Redis. It is much easier to get started with one of these databases than with MySQL and Postgres. But don't get fooled into thinking that NoSQL databases are a perfect replacement – in some situations a classic relational database will make your development easier even if it takes more effort to set up.

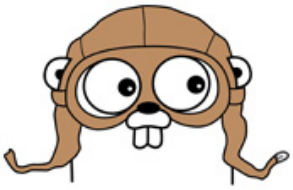
6. Less/Sass/Stylus

There is a lot to dislike about CSS. It is too easy to end up with an unwieldy 1000 line css file which is hard to navigate and change. To solve this, there are languages like Less, Sass and Stylus which are compiled to CSS and offer things like variables, macros and other goodies that will help you write better code. You can learn one of these in a single afternoon.

7. Exciting new frameworks

Meteor is a radically new approach to web application development which blurs the boundaries between front end and back end. It allows you to write reall-time apps, and has a rapidly growing community writing packages for it. Hood.ie is a smaller contender, but offers a novel approach. It handles the backend for you, so you can concentrate entirely on the front end of your application.

8. Exciting new languages



For the language nerds out there, here are some treats. Golang, Rust and Elixir are gaining momentum in programming circles and are used in situations which demand extremely high performance. We don't recommend moving your development to one of these just yet, but you might want to do the interactive tutorials that are provided on their websites.

9. A classic full stack framework


Even though single page applications are gaining popularity, there is still a huge demand for classic server-side web apps. Ruby on Rails, Django, Laravel, Play, ASP.NET are the top full-stack frameworks at the moment. But any solid MVC framework will do wonders to your productivity if you take the time to study it.

10. The old guard

There is a large collection of established languages and platforms that are still in demand – Java, .NET, Python, Ruby. They have large communities and will look good on any CV. They all have their pros and cons, but it doesn't hurt to create small side projects in one of them every now and then. This is something that no programming course or tutorial will teach you and you will quickly get a feel whether that language fits with your way of work.

11. Don't forget these

These are good to look at for front end since it is easy for implementation and they make the product really good

PHP, WordPress, and jQuery are still a perfectly valid way to create a website. WordPress has outgrown its blog platform past, and is now a powerful CMS/framework for developing a wide range of web applications. If you are a designer you should consider picking these technologies up. If you decide to go with PHP for your backend needs, don't forget about the good practices in PHP the right way. Also, if you haven't already, take a look at Bootstrap – it will help you write frontend code. There are also lots of bootstrap plugins you can choose from.

PHP is also a good option to consider since it has been around for so long that there are alot of resources to look at.