The Languages and Frameworks You Should Learn in 2017

Martin Angelov December 8th, 2016

Tweet J'aime 1,8 K Partager G+1 30

The software development industry continues its relentless march forward. In 2016 we saw new releases of popular languages, frameworks and tools that give us more power and change the way we work. It is difficult to keep track of everything that is new, so at the end of every year we give you our take on what is important and what you should learn during the next twelve months.

The Trends

Progressive Web Apps

In 2016 we saw the rise of the Progressive Web App concept. It represents web applications that work offline and offer a native, app-like experience. They can be added to your smart device's homescreen and can even send you push notifications, bridging the gap with native mobile apps. We think that in 2017 PWA are going to become even more important and are well worth investigating. See our overview here.

The Bot Hype

Everybody is talking about bots right now. From platforms for running them, to frameworks for building them, the community is buzzing with activity (read our intro here). Bots are the new mobile apps, and if you hurry up you can catch the wave while everyone is excited. Once the novelty wears off, bots will probably be relegated to some boring role such as automated customer support. But hey, we can dream!

Consolidation of Frontend Frameworks

In the JavaScript community we have an incredible churn of frameworks and tools, with new ones being born almost every week. Until recently, the expectation was that the old tools would just be replaced by the new, but this is not what we saw in 2016. Instead, we saw the popular frameworks exchanging ideas and incorporating the innovations put forth by newcomers. So in 2017 it won't matter much which of the major JS frameworks you choose, their features are mostly comparable.

The Cloud

Companies and developers everywhere are embracing "the cloud". This is virtualized computer infrastructure that is available on demand and fully configurable from a control panel. The big three cloud providers are AWS, Google Cloud and Azure. Thanks to their ongoing competition prices have been falling, bringing it within the budgets of smaller companies and individual developers. Familiarizing yourself with the cloud workflow would be a good investment for 2017.

Machine Learning

Machine Learning (ML) has exploded in popularity during the last twelve months. And with the historic AlphaGo vs Lee Sedol match in March, it entered the mainstream. Smart computer systems that learn from raw data are revolutionizing the way we interact with our mobile devices. By the looks of it, ML will be an even bigger factor in 2017.

Languages

JavaScript continues its incredible pace of innovation. Catalyzed by the quick release schedules of web browsers, the JS standard is updated every year. The next edition, ES2017, is expected to be finalized in mid 2017. It will bring the dream feature of many JS developers – `async/await` for working with asynchronous functions. And thanks to Babel, you can write ES2017 in every browser even today.

TypeScript 2.1 was released in late 2016, bringing `async/await` for old browsers and improved type inference. TypeScript is a statically typed language which compiles to JavaScript. It adds powerful features like a classic OOP model and optional static typing to make large codebases easier to maintain. It is the preferred language for writing Angular 2 apps, and we recommend giving it a try. Here is our quick start guide about it.

C# 7.0 is expected in 2017 and will enhance an already excellent language. Microsoft surprised everyone when they introduced the open source Visual Studio Code editor and .Net Core. Both of these run on Linux, Windows and macOS and allow you to write fast and performant applications in C# (read more here). A vibrant community is forming around both of these tools, and we are confident there is an exciting year ahead of them.

Python 3.6 was released in December. It is solidifying its place as the scripting language of choice for devs, IT pros and scientists. It is suitable for automation, web development, machine learning and scientific computing. The Python 2/3 split has been an years-long struggle for the community, but these days you can confidently choose 3 and enjoy full library support. For those in need of extra performance, they can take a look at PyPy, an alternative JIT enabled Python runtime.

Ruby 2.3 was released earlier this year with a number of performance improvements. Ruby is also a good choice as a general purpose scripting language, but it shines when paired with Rails. The Ruby 3×3 initiative was announced, which will attempt to make the upcoming Ruby 3 release 3 times faster that the current version, opening the doors to using Ruby in more contexts.

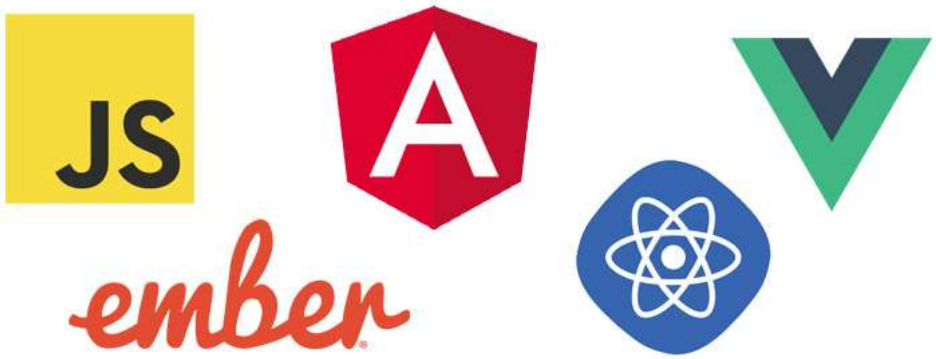
PHP 7.1 was released in December, and brings minor enhancements to the language. This builds upon the major performance improvements that were had in version 7.0 last year, turning PHP into a fast platform for building web applications. We recommend PHP The Right Way for good practices and a modern take on building web apps in the language.

Java 9 is expected in 2017 and will come with welcome new features like a repl for evaluating code, HTTP 2.0 support and new APIs. There is a strong demand for talented Java developers and a breadth of exciting projects that use the language. If Java is not your thing, there are a number of JVM based languages like Kotlin and Scala that you can check out.

Swift 3 was released earlier this year. This is Apple's vision for a modern programming language that eases the development of apps on iOS and macOS. Swift is open source and has attracted a large community. Version 4 is planned for 2017, which will improve the language and introduce server APIs, making it a good choice for writing web apps and backends.

If you are looking for something more exciting, you can try out Crystal and Elixir, which both combine a friendly ruby-like syntax with superior performance. Or you can look into a functional language like Haskell or Clojure. Two other fast languages are Rust and Go which we recommend.

Learn one or more of these: JS (ES2017), TypeScript, C#, Python, Ruby, PHP7, Java/Kotlin/Scala.



Frontend

The web platform made two major advancements recently – Web Assembly and Service Workers. They open the gates for fast and performant web applications that bridge the gap with native compiled applications. Service Workers in particular are the enabling technology for Progressive Web Apps and bring support for Notifications to the web platform, with more APIs to follow in the future.

Angular.js 2 was released this year. The framework is backed by Google and is very popular with enterprises and large companies. It has a vast number of features that make writing everything from web to desktop and mobile apps possible. The framework is written in TypeScript, which is also the recommended language to write applications in. There is a lot to read about, but we think learning Angular 2 in 2017 would be a good investment.

Vue.js also saw its 2.0 release this year. It borrows the good ideas from Angular, React and Ember, and puts them into an easy to use package. It is also quite a bit leaner and faster than the first two. We suggest that you give it a try this year, by starting with one of our Vue.js tutorials.

Ember is another solid choice for a JavaScript framework. It supports data bindings, auto-updating templates, components and server-side rendering. One benefit that it has over its competitors, is that it is more mature and stable. Breaking changes are much less frequent and the community values backwards compatibility. This makes the framework a good choice for long-lived applications.

Two other frameworks that are worth a look are Aurelia and React. The ecosystem around React has grown considerably more complicated in the last year, making it difficult to recommend for beginners. But experienced devs can combine the library with GraphQL, Relay, Flux and Immutable.js into a comprehensive full stack solution.

No frontend compilation would be complete without mentioning Bootstrap. Version 4 is currently in Alpha and a release is expected in 2017. Notable changes are the new versatile *card* component and the flexbox grid (see our comparison with the regular grid here), which modernize the framework and make it a joy to work with.

SASS and LESS remain the two most popular CSS preprocessors today. Although vanilla CSS is finally getting support for variables, SASS and LESS are still superior with their support for mixins, functions and code organization. If you haven't already, take a look at our SASS and LESS quick start guides.

Learn one or more of these: Angular 2, Vue.js, Ember, Bootstrap, LESS/SASS.

Backend

There is plenty of choice for the backend, all coming down to your preference of a programming language or specific performance needs. An ongoing trend in web development is business logic to move away from the backend, turning that layer into an API which is consumed by the frontend and mobile apps. But a full stack framework is often simpler and faster to develop in, and is still a valid choice for a lot of web apps.

Node.js is the primary way for running JS outside the browser. It saw many new releases this year, which increased performance and added coverage for the entire ES6 spec. Node has frameworks for building fast APIs, servers, desktop apps and even robots, and a vast community creating every kind of module imaginable. Some frameworks that you may like to look into: Express, Koa, Next, Nodal.

PHP is a web language first and foremost, and has a large number of web frameworks to choose from. Thanks to its excellent documentation and features, Laravel has formed an active community. Zend Framework released version 3, which marks a great upgrade for this business oriented framework. Symfony also saw a lot of new releases this year, making it an even better choice as a full stack solution.

For **Ruby**, the Rails framework is the premier choice. Version 5.0 was released in 2016, bringing support for Web Sockets, API mode and more. Sinatra is also a good choice for small apps, with version 2.0 expected sometime in 2017.

Python has its own full stack/minimal framework combo in the form of Django and Flask. Django 1.10 was released in August introducing full text search for Postgres and an overhauled middleware layer.

The **Java** ecosystem also has popular web frameworks to choose from. Play and Spark are two solid choices, and as a bonus they can be used with Scala as well.

For the enthusiasts there is also Phoenix, which is written in **Elixir** and attempts to be a feature complete alternative to Rails with superior performance. If Elixir is one of the languages you would like to learn in 2017, give Phoenix a try.

Learn one of these: A full stack backend framework, a micro framework.

Databases

PostgreSQL saw two whole releases this year – 9.5 and 9.6. They brought the long awaited UPSERT functionality that we know from MySQL (aka `ON DUPLICATE KEY UPDATE`), better full text search and speed improvements thanks to parallel queries, more efficient replication, aggregation, indexing and sorting. Postgres is used for massive, terabyte scale datasets, as well as for busy web apps, and these optimizations are welcome.

MySQL 8.0 is going to be the next major release of the database. It is expected sometime in 2017 and it will bring a lot of improvements to the system. MySQL is still the most popular database management system and the entire industry benefits from these new releases.

For NoSQL fans, we can recommend CouchDB. It is a fast and scalable JSON storage system which exposes a REST-ful HTTP API. The database is easy to use and offers great performance. PouchDB is a spiritual counterpart to CouchDB that works entirely in the browser and can sync with Couch. This allows you to use Pouch in an offline ready web app, and get automatic syncing once internet connectivity is available.

Redis is our favorite key value store. It is small, fast and versatile. You can use it as a smart memcache alternative, as a NoSQL data store or a process messaging and synchronization channel. It offers a large number of data structures to choose from, and the upcoming 4.0 release will have a module system and improved replication.

Learn one of these: Postgres, MySQL, CouchDB, Redis.

Tools

Yarn is an alternative package manager for Node.js which is developed by Facebook. It is an upgrade over the npm command line tool and provides faster installs, better security and deterministic builds. It still uses the npm package registry as its backend, so you have access to the same incredible ecosystem of JavaScript modules. Yarn is compatible with the `package.json` format that npm uses, and is just a quick install away.

The two most popular open source code editors – Visual Studio Code and Atom have seen an incredible amount of innovation in the past 12 months. Both of these projects are built using web technologies and have attracted huge communities of fans. The editors have plugins available which bring syntax checking, linting and refactoring tools for a large number of languages.

Git is the most popular source code version control system out there. It is serverless and you can turn any folder on your computer into a repository. If you wish to share code, you have many options like GitLab, Bitbucket and Github, to name a few. For 2017 we suggest that you familiarize yourself with the git command line, as it will come in handy more times than you think.

Desktop applications are not dead yet. Even though web apps are becoming more and more capable, sometimes you need powerful capabilities and APIs that are simply not available to the web platform. With tools like **Electron** and **NW.js** you can write desktop applications by using web technologies. You get full access to the operating system and the breadth of modules available to npm. To learn more about these tools, read our tutorials about Electron and NW.js.

A recent trend in software team organization is to have developers who are in charge of their own software deployment. Also called DevOps, this leads to quicker releases and faster fixes of issues in production. Developers with operations experience are highly valued by companies, so familiarity with the technologies that enable it is going to be a huge plus from now on. Some of the tools that we recommend are Ansible and Docker. Experience with the Linux command line and basic system administration skills will also serve you well.

Try out one or more of these: Yarn, Git, Visual Studio Code, Electron, Ansible, Docker.

Tech

The **cloud** has won over the entire software industry, with large companies closing down their datacenters and moving their entire infrastructure there. The three main platforms are AWS, Google Cloud and Azure. All three have powerful, ever expanding feature sets, including virtual machines, hosted databases, machine learning services and more. Prices are going down rapidly, and the cloud is within reach of small companies and individual developers. For 2017, it would be a good learning experience to deploy a side project to one of these providers.

Artificial Intelligence was the buzzword of 2016. Speech recognition and image classification are only two of the user facing applications of the technology, with machines reaching and even surpassing human level performance. There are a lot of startups that apply AI and Machine Learning to new domains. And a lot of open source projects were released like Google's Tensor Flow and Microsoft's Cognitive Toolkit. Machine Learning is a very math-heavy topic, and for those just starting out there are comprehensive online courses available.

Virtual Reality (VR) and **Augmented Reality** (AR) have been around for a while, but finally the technology is mature enough to offer a compelling experience. Facebook (Oculus Rift), Google (Daydream) and Microsoft (Windows Holographic) all have virtual reality platforms that welcome third party developers. VR headsets still face challenges like eliminating nausea and offering compelling use cases outside of gaming, but they are getting there.

Learn one of these: Cloud deployment, a Machine Learning library, VR Development.

Article License **Privacy Policy** **Contact Form** **Advertise**

Zine EOOD © 2009-2017