# Recap of Front-end Development in 2017

- HTML 5.2 is done.
- It was a banner year for Vue.js in terms of adoption and popularity. No question about it.
- The great divide between a front-end HTML & CSS developer v.s. front-end application developer is realized/verbalized.
- Being a Front-end JavaScript developer who builds applications using web technologies continues to get better and worse.
- This year seemed fuller than most of app/framework solutions trying to contend with the mainstream JavaScript app tools (i.e. React, Angular, and Vue etc...) Let me list them for you. Moon, Marko, Hyperapp, Quasar Framework, POI, frint, BunnyJS, jsblocks, Sapper, Stimulus, Choo,
- This was the year that jsbin and jsfiddle evolved to things like StackBliz and codeSandbox. Making it dead simple to share a working app.
- React continues to be flattered by things like preact, inferno, nerv, dva, and rax.
- Cheatsheets got organized with devhints.io.
- We figured out that the correct pattern for an app boilerplate/cli tool is something very opinionated like React Create App with the ability to escape from it when needed.
- Most developers found that the combination of a really good code editor, eslint, and now prettier make writing code faster, easier, pleasurable.
- CSS Flexbox and Grid gain browser support and thus more developers are paying attention to both.
- We get, a headless chrome, finally.
- You no longer need Less or Sass to do amazing things with CSS.
- CSS revolutions/revolts are under way.
- JavaScript object explorer tools have arrived, JavaScript Array Explorer and JavaScript Object Explorer. This is a handy interface pattern for learning about JavaScript data types (e.g. Objects and Arrays) and their methods.
- The Chrome web browser dominates the market and people begin to fear the past might be repeating itself.
- Brave becomes the most pleasant and safest way to browser the internet.
- PhantomJS is no longer maintained, Headless Chrome and Puppeteer step in.
- Prettier comes from left field and becomes a staple for development.
- A whole lot of developers adopt static type checking for mostly subjective reasons or band wagon emotions. Some sell out completely to Typescript and the Microsoft way of doing things while others take on a slower approach with Flow. One thing is for sure, most developers don't need types, they are simply complicating already complex problems and solutions. Like most things, most of this trend is subjective dogma not

objective value.

- Static site generators & API CMS tools aka Headless CMS's are now on most developers radar.
- Web components still lurking and wait for significant traction by developers that might never come to be.
- JavaScript settled and CSS erupt and everyone will cry fatigue by this time next year.
- A lot of people stop doing CSS in CSS and move to CSS in JS when building application using component trees.
- Yarn seems to have filled a need, because a lot of people jump the npm ship. However, the real value of Yarn is the fact that it brings competition to NPM. Making npm better.
- A new video format for Interactive coding screencasts (recording of working in a live editor that you can edit too) becomes a real thing with Scrimba.
- Most people begin to see the correlation between component architectures and atomic design.
- And so it begins that ES modules will be part of the browser and if used a backup plan will be required (i.e. a bundle from something like webpack).
- MVC frameworks are on the outs.
- Developing and displaying React components outside of your applications is made popular by tools like Bluekit, Storybook, React Styleguidist, and bit.
- Getting a front-end job in 2017 is about experience, which is displayed from personal projects and a developers Github account.
- Preloading resources (CSS, JavaScript, Media etc..) from HTML documents arrives.
- Cypress arrives as a complete testing solution and hopefully testing will get better as end to end testing becomes the focus for app code.
- WebAssembly support now shipping in all major browsers
- Webpack dominates, and then competitors show up.
- React 16 aka fiber is released.
- React begins to rival jQuery in popularity in certain contexts.
- React clearly is the most used tool for building UI's with state.
- Facebook sheds its React BSD license for the MIT license (same for Jest, Flow, Immutable.js, and GraphQL)
- GraphQL got hot in 2017.
- Facebook continues to take charge in the development space with forthcoming tools like prepack.io.
- As expected ECMA-262 edition 8 is released.
- React Router finally stabilizes.
- All modern browsers pretty much now support ECMAScript 2015 (aka ES6).
- Async JavaScript functions start getting some serious attention and usage. Mostly because all modern browsers now support Async functions.
- Mobile development, still too hard. A strong rebellion advocating the web platform as a

solution to the pain gained momentum this year.

# In 2018 expect...

- Nothing will change or slow the usage or popularity of React for many years to come.
- GraphQL will replace a lot of REST API's this year.
- The web will continue to become more native-like with offline capabilities and seamless mobile experiences.
- HTML 5.3 is coming.
- Keep an eye on turbo, a blazing fast NPM client.
- Expect to learn and use CSS transforms 3d, CSS transitions, CSS flexbox, CSS filters, CSS grid
- JavaScript usage will continue to grow with no slowdown in sight.
- Still waiting on Web Assembly to peak. This will likely require tooling.
- Universal/isomorphic JavaScript solutions continue to evolve e.g. next.js and Sapper.
- Web components still lurk and wait for significant traction from developers.
- I believe the end is in sight for CSS pre-processors as PostCSS, CSSnext, and CSS in JS take over.
- Older server centric application patterns show up again but with a new spin. The pendulum could start to swinging away from strick SPA applications. People will begin to pull back on the complexity of single page applications and return to things like pjax (A mix of SPA and Server-side Rendering. See https://stimulusjs.org).
- Progressive Web Applications hopefully will catch fire. If they don't, I fear they never will. At least not in there current form.
- "Chatbots created on the basis of artificial intelligence and neural networks will continue to evolve helping to increase communication online. I wonder what it will lead to, but this is unconditional web development trends 2018". Nods.
- Vue.js usage will likely overtake all Angular usage.
- AR/AV, AI, and chat bots will continue to evolve and find there sweet spot.
- JavaScript Symbol and Generators will likely go unnoticed by most front-end developers.
- More developers will divorce themselves from plain JavaScript and try to marry another. But, just like in marital divorce one always takes most of the same problems with them to the greener grass and little actually changes. Preferences and values just get re-prioritized and history will repeat itself.
- Webpack 4 will happen, and be better, due to competition!
- Continued exploration for the ideal CSS solution for a tree of UI components will not cease.
- State management gets a reset and people start to simplify. Hopefully, this will be the year for solutions like mobx to shine.