

The Unofficial Guide to Becoming a Front End Developer.



Stefan Charlebois [Follow](#)

Aug 16, 2017 · 7 min read



Its pretty common to hear the title of Front End Developer being thrown around these days. The funny part about front end is that 9 out of 10 people seem to consider it different things. This became a big problem for me when I was trying to learn the ropes. I spent hours and hours digging around for articles, tutorials, videos, ebooks; anything I could get my hands on to learn more. What I found was that I always felt like I knew less than when I started.

One thing I had going for me was a software development background. I took a few extremely introductory courses on HTML and CSS in college and knew that these were the foundations of Web Development. As I started learning more advanced HTML and CSS I found all these other technologies popping up: JavaScript, jQuery, React, Rails, Backbone, Node JS, PHP...WTF is all this? They must be pretty important because there are multiple tutorials, articles and courses posted on them everyday.

I would sit and make lists of what technology I had to learn next and never really had a clue as to what they would do for me. Everyone is talking about them so I have to know them right? OMG! How am I going to learn them all?!?

Wrong...The web is a never ending list of flavour of the week technologies that are here one day and gone the next. 95% of it is just noise that distracts you from learning and actually building something. I am not going to say that I don't get excited by some of these technologies, but I've learned not to let it distract me.

Ok...So, if you can ignore the majority of technologies then what ones should you pay attention to?

The Basics

1. HTML

This is not really an option folks. HTML is the foundation of any website and has been since Al Gore invented the Internet (get the ref?). It describes the content of the page to the web browser, which then renders it for your users viewing pleasure. HTML can be used to add text, images, audio, video, forms, tables, and guestbooks (shout out to Geocities!) to your website. The great thing about HTML is that it's easy to learn and you can write it in any text editor. Did I mention the best editors are free? Keep that precious money for beer, or coffee, or *insert thing I spend my money on*.

2. CSS

Do you have room for another love / hate relationship in your life? No? Too bad. CSS is also a cornerstone of web development and its going to be your best friend and worst enemy all in the same day. HTML

describes what's on the page and CSS describes how it should look. You want a hot pink button that bounces when you hover? CSS can do that for you. You want a 4 column grid that dynamically resizes, and is compatible in Internet Explorer 6? Well...It can do that too, but you may tear a bit of hair out. The bottom line is CSS is easy enough to pick up, but is challenging to master. You can already do some amazing things with it and the features are growing daily. Stick with this and it will pay off. You may need to use that pay off for new hair, but at least you learned something right?

Learning Material

Below is a link to the course I used to get my feet wet with HTML and CSS. The course instructor Jeffrey Wray not only teaches everything you should know at the start, but also explains why things work the way they do. I highly recommend this course...and no, I don't work for the company and am not secretly Jeffrey Wray.



3. Javascript

Ok this is where things become a bit more fun. So, we've got our content on the page and it's got a nice coat of paint on it thanks to CSS, but it's still going to feel pretty static. Let's bring this thing to life! I want to click a button and have dollar bills start raining down the page. JavaScript can help you make it rain. JavaScript can be used for triggering events, inserting / removing elements from our page, creating pop ups, and yes—"making it rain". It's very flexible and is an incredibly hot topic in the web world right now. When you start down

your journey to mastering Javascript pay close attention, as if it were your first crush in grade school.

Warning: There will be a learning curve if you've never coded before, but the great thing is the foundational skills you'll learn are applicable to any programming language. Its a pretty easy language to learn the foundational topics with.

Learning Material

There are a ton of courses out there, but I didn't really follow a specific course to learn JavaScript. I learned through random videos, blog posts, tutorials etc...

Since I'm such a nice guy I did some quick searching and found a site that has beginner to advanced tutorials. After a quick skim it looks like it has some good information to get you started.


 A screenshot of the HTML Dog website. The header features the logo 'HTML Dog' with a stylized dog icon. The main navigation bar includes 'Home', 'Tutorials', and 'JavaScript Tutorials'. Below the header, a sidebar on the left lists categories: 'Tutorials' (HTML, CSS, JavaScript), 'Techniques', 'Examples', 'References' (HTML Tags, CSS Properties), and 'Web Hosting'. The main content area is titled 'JavaScript Tutorials' and contains a brief introduction: 'While, generally speaking, HTML is for content and CSS is for presentation, JavaScript is for interactivity.' It also notes that JavaScript is used for interactivity on large applications and games. A section titled 'JavaScript Beginner Tutorial' provides an introduction to programming through JavaScript, listing topics like 'Making Stuff Happen', 'Variables and Data', 'Doing Math', 'Logic', 'Conditional', 'Looping', 'Functions', 'Objects', and 'Arrays'.

Click the image to visit the site

Once you feel you have a good understanding of the basics of JavaScript, I would highly recommend a **FREE CODING COURSE** called [JavaScript30](#) by [Wes Bos](#). I've gone through this course myself, and even after years of using JavaScript, I learned a lot of cool little tricks. It's split up into small daily projects that teach you about different subjects in JavaScript. The best part is the projects are actually fun to build and really well done. [Wes Bos](#) is also a great instructor and offers courses on a bunch of other hot front end topics.



Click to go to [JavaScript30.com Website](https://javascript30.com)

If you are finding you need a bit more guidance than the quick tutorials above, I would recommend taking a full course on JavaScript. There are many great online code schools and I've linked to a few of their JavaScript courses below.

1. [Free Code Camp](#)
2. [Code Academy](#)
3. [Treehouse](#)
4. [Lynda](#)

Once you have a grasp on the basics I would highly recommend bookmarking Mozilla's developer network site on Javascript. I use this on an almost daily basis and its helped me countless times

[Mozilla Developer Network—Javascript](#)

4. Javascript Frameworks

So, we know the foundations of Javascript, were making some interactive elements on our pages, but were ready to take things to the next level. This is where things can get really blurry for someone just hopping into the game.

The Javascript ecosystem changes extremely rapidly, and can quickly have your head spinning with flavour of the week technologies. I have

been working in the ecosystem for 5+ years and still have to put on my blinders to avoid insanity.

Frameworks typically help developers by giving them a foundation for the structure of their website or web application. Some are very strict and opinionated, where others are very loose and give a very high level guideline to follow. I am a firm believer that there is not a single framework that makes sense for every project, but I will name some of the most popular players in the game below.

Note: The descriptions below have been taken directly from each framework's website and might not mean much now, but as you advance these will help take your skill set to the next level. Don't stress over them now, just make a note to read up on them once you are comfortable with Javascript.

1. React.js

React makes it painless to create interactive UIs. Design simple views for each state in your application, and React will efficiently update and render just the right components when your data changes.

<https://facebook.github.io/react/>

2. Angular

Angular is a platform that makes it easy to build applications with the web. Angular combines declarative templates, dependency injection, end to end tooling, and integrated best practices to solve development challenges. Angular empowers developers to build applications that live on the web, mobile, or the desktop

<https://angular.io/>

3. Vue.js

Vue (pronounced /vju:/, like view) is a progressive framework for building user interfaces. Unlike other monolithic frameworks, Vue is designed from the ground up to be incrementally adoptable.

The core library is focused on the view layer only, and is very easy to pick up and integrate with other libraries or existing projects.

<https://vuejs.org/>

4. Ember.js

Ember is a JavaScript front-end framework designed to help you build websites with rich and complex user interactions. It does so

by providing developers both with many features that are essential to manage complexity in modern web applications, as well as an integrated development toolkit that enables rapid iteration.

<https://www.emberjs.com/>

As I mentioned, don't let these stress you out now. When you are comfortable with Javascript and move onto working on more complex websites and applications, these will be valuable tools to have.

Now get to work...

As you may have noticed, this isn't meant to be an extremely intricate guide on becoming a front end developer, but a rough path on how I became one myself. I have been doing front end for almost 8 years now, and I'm still learning everyday. Don't let blog posts, tweets and other peoples skill set discourage you from focusing on the basics first. Many people become overwhelmed by all the technologies out there and lose focus on the most important thing...The basics.

HTML, CSS, and Javascript are more than enough to land your first gig. There are endless resources online to learn all three of these technologies proficiently for little to no cost. Focus on these, put together some nice portfolio pieces, and get those resumes onto desks!

Until next time folks...

