Getting started with front-end development in 2018

This year is no different, with one exception. This is the year you'll make it!

Maciej Matuszewski Follow
Feb 11, 2018 · 8 min read

Photo by Jantine Doornbos on Unsplash

Web development is a great place to start your adventure with programming. The concepts are easier than in other parts of computer science such as machine learning, or systems programming. I remember how hard it was, starting out. I decided the time has come, that I share some of my knowledge. Hopefully, it will help you or somebody you know to gain the, much needed, motivation and push to get started!

Who am I, to talk about it?

Before I get to the point, let me tell you a bit about myself. Currently, I am working as a front-end developer at EL Passion (software house), and after hours I am writing articles and coding in React Native with my friends at **Applantic**. I was always interested in programming, but I really got into it about two years ago, when I got my first job. I was lucky enough to start without having any actual knowledge, but it meant I had to learn a lot, really quickly. Since then, I've been constantly learning great things.

A lot of people say that you need to know math really well to get into programming. That's not exactly true. The most important things you need, to start, are motivation and time. Math helps a lot when you get to the more advanced concepts, but to get started it isn't really necessary. I didn't excel at math, but I don't feel like it was a huge obstacle in becoming a front-end developer.

As you can see, I've only been programming for about two years, which isn't that much, but I feel like I can already make it easier by sharing my experiences over this time.

I remember that the hardest part was combining all the tutorials, into a more structured learning plan, that could teach me more than the bare minimum. Because of that, I prepared a simple roadmap, that might help you to get a better overview of basic technologies and things you should learn when getting started.

1. Start with the basics — HTML

Well, this is obvious, you always start with the basics. HTML is the backbone of the Internet. It's not a programming language per se, but it's a tool that every web developer should master.

What does HTML stand for?

H—Hyper
T— Text
M—Markup
L—Language

HTML is basically a set of building blocks, that we can use to define the structure of our website. When I explain HTML to people who don't know anything about web development, I like to refer to basic human biology.

HTML is the bones and the structure. It is the skeleton, of the human body. By itself, it can't really do a lot, but it defines a structure.

2. Make everything look better — CSS

After creating a basic website skeleton, we can get started with CSS. It stands for:

C—Cascading
S—Style
S—Sheets

CSS is responsible for the visual part. It allows us to specify how elements (HTML building blocks) should look. We can define elements sizes, colours, backgrounds and many many more.

Going back to my biological metaphor, we could tell that CSS is the characteristics of the person. It tells us about skin colour, eyes, height etc.
At this point we get a human being, that doesn't do anything yet, but has body (HTML) and look (CSS).

3. Start Building Things! (really important)

This step is essential. Don't worry that you don't know something. At this point, Google and StackOverflow should be your best friends. Whenever you don't know something, try to google it. You don't need javascript yet, so you can simply ignore it for now. HTML & CSS are the two skills that will land you your first job.

When starting out, it's a great idea to **copy** things that are popular. Try to use **HTML** & **CSS** to create a copy of a **Facebook post** or **Medium**

navigation. Go to **Airbnb** and copy their footer. It is essential, that you don't spend too much time on theory, and that you spend it coding.

I recently saw a tweet by **Chris Coyer** that really resonated with me:

Perfectionism will slow down the process of learning, that's why we should let ourselves fail sometimes, and gain more knowledge, quicker.

4. Learn about JavaScript

Next step, after building a few basic layouts, is JavaScript. It is a programming language created for Netscape corporation, by Brendan Eich, in a really short time. It was supposed to be used for simple things. Nobody could predict that it will be used in large-scale applications and basically in every existing environment it could be run on. Let's also get a few misconceptions out of the way:

1. **JavaScript ≠Java—**Those are two totally different languages
2. **jQuery is built with JavasScript—**It's not a different language. JQuery is a library built with JavaScript that's used to simplify and speed up development.
3. **JavaScrip = ECMAScript = ES2015/ES3/ES5/ES6…—** ECMAScript is a more technical name for JavaScript, but the original name is way more popular and used these days. ES2015 is the same ES6, which stands for the 6th version of ECMAScript. ES3,5,7 etc. are just different versions of Javascript

JavaScript evolved a lot over the years. There are a lot of libraries and frameworks that allow us to write it easier, better and faster. If any of those words are new to you, **do not worry! You will learn about everything when it's time.**

It’s easy to get weighed down by the number of new things you learn. One thing I learned when I was starting out, is that everything will become clear when it’s time. Consistency is the key. If you do something regularly you will achieve your goal. Something will ‘click’ in your brain, and you will get concepts that were impossible to understand just a week ago!

Javascript is a huge topic, and it is like a window to the world. If you learn it you will be able to create desktop, backed and even mobile applications. It’s important to know it well, but also remember that to land your first job in front-end development it’s not that important to know all the methods. HTML and CSS knowledge is the key to getting your first job. JS will be a big plus and you should know how to use it, but at the first stage, just knowing some syntax (language structure) and jQuery (library), will be enough.

5. Try to collaborate with others

Working with other people is essential to building great things. Julius Caesar couldn’t build his empire all by himself. Here are a few places where you can find people to work with on side projects, learn from or with, and hopefully work with.

1. Facebook groups—You will find great resources and fresh news here.
2. Meetups (meetup.com)—Great networking opportunities, enhances possibility to quicker find your first job.
3. Stack Overflow—Ask your questions, but only those that weren’t already answered! Great to build your online presence, and meet really smart people.
4. Github—This will probably be your collaboration hub when working on open source, or personal projects.

6. Get your first job

It just gets easier from here. If you land your first job as a junior front-end developer, you will have people to learn from, and you will constantly learn and use your skills.

You hear a lot about lack of developers. It's partly true because there is a huge lack of **skilled** developers. There are a lot of people that have some knowledge and want to become juniors, so there will be some competition when looking for your first job. From what I know there are two key things that will help you **a lot** in getting your first job.

1. **Attention to details**—This is the moment when you need to care about the quality of your code. When you send over code samples or provide a link to your GitHub account, make sure that the code is clean. HTML should be indented and formatted correctly, CSS shouldn't have a lot of comments, it should follow some of the good conventions (not essential, but helps a lot), like OOCSS or BEM. If you are asked to code something, make sure it's pixel perfect. Font size should be correct, spacing and font-weight also.
2. **Motivation to learn**—Show that you are willing to learn new things. Tell about what you are interested in, what are your goals. Show that you don't do it only for money. Curiosity and openness is a feature that makes great developers great. If you show that you have it, it will make you worth investing in.

7. Keep Learning

Of course, there is still a lot of things to learn. And I mean a shitload… Probably you will still feel overwhelmed by the number of topics that you hear about.

It's a good thing to know a bit about other aspects of web development, such as backend and databases. Understanding how the web works will clear out a lot of things for you.

Here are a few topics that are worth knowing:

1. **CSS preprocessors**—SASS, SCSS, LESS, PostCSS
2. **JS build systems**—Gulp, Webpack, Grunt
3. **JS Frameworks and Libraries**—Vue.js, Angular, React, Preact, Ember and many more…
4. **SVG on the web**
5. **Node.js**—backend development with JS

6. **CSS methodologies**—BEM, OOCSS, SMACSS, ITCSS and many more
7. **Semantically correct HTML**
8. **JS Design Patters**
9. **Mastering you code editor—**shortcuts and plugins

I strongly recommend that you check out this Web Developer Roadmap for the year 2018. It's full of great insights into the technologies of web development.

Learning Resources

The two amazing websites that still help me in my day-to-day work are **W3 Schools** and **Mozilla Developer Network (MDN).**

A lot of people say that **w3schools** are not a good source of information, but I disagree with that a bit. When I was getting started the explanations on this site were much more clear than the ones on **MDN.**

I would say that **w3schools** are better for the basic explanation when you are getting started and **MDN** is the ultimate go-to documentation that is very thorough and from is backed by the known companies such as Microsoft, Google, and Samsung.

When it comes to actual tutorials, I am not going to create yet another list of them because you can easily find that in google, but I can list a few websites that I've used and was happy with. You will find a lot of resources on this websites, so I won't repeat them in the following sections. You will find most of the tutorials you need on those sites.

- **Codecademy**
- **FreeCodeCamp**
- **Pluralsight (paid)**

"Fail Fast, Fail Often"

This quote is a popular silicon valley mindset. Some people say it’s true, some strongly disagree with that. In my opinion, at the beginning of your web development journey, following this mindset, will speed up your learning progress by an order of magnitude.

Remember that learning is a process, and just reading won’t help you that much. **Get your hands dirty, and stay consistent with your learning efforts!**

· · ·

Tap the 👏 button if you found this article useful!

About us

Applantic is a team of passionated software developers. We write our stories on Medium ✍ but you can find us on Instagram 📷 or Facebook ✍ & 📷 as well.

The author of this article is Maciej. Front-end developer, working on web applications in a software house. He is interested in many things, even some that are not related to computer science, such as sports and music. You can find him on GitHub or LinkedIn.

kamranahmedse/developer-roadmap

developer-roadmap - Roadmap to becoming a web developer in 2018

github.com