

# How To Learn Web Development

Part 1: Beginning front end development

Spyros Argalias

December 30, 2018

# Contents

|          |  |          |
|----------|--|----------|
| <b>1</b> | <b>Fields of Programming</b>                         | <b>1</b> |
| <b>2</b> | <b>Why web development?</b>                          | <b>1</b> |
| <b>3</b> | <b>What is web development?</b>                      | <b>1</b> |
| <b>4</b> | <b>Components of Web Development</b>                 | <b>2</b> |
| 4.1      | Front end development . . . . .                      | 2        |
| 4.2      | Back end developmen . . . . .                        | 2        |
| 4.3      | Dev ops . . . . .                                    | 2        |
| 4.4      | Which should you start with? . . . . .               | 3        |
| <b>5</b> | <b>How to learn web development</b>                  | <b>3</b> |
| 5.1      | A bad plan . . . . .                                 | 3        |
| 5.2      | Learn in iterations . . . . .                        | 4        |
| <b>6</b> | <b>Web development learning plan</b>                 | <b>4</b> |
| 6.1      | Introduction to web development . . . . .            | 4        |
| 6.2      | Learning HTML and CSS . . . . .                      | 4        |
| 6.2.1    | The three resources you need . . . . .               | 5        |
| 6.2.2    | How to go through the resources . . . . .            | 5        |
| 6.2.3    | Bonus resources . . . . .                            | 6        |
| 6.3      | JavaScript . . . . .                                 | 6        |
| <b>7</b> | <b>Next steps, preparing for a job</b>               | <b>7</b> |
| 7.1      | Build a website for your portfolio . . . . .         | 7        |
| 7.2      | Portfolio and practice . . . . .                     | 8        |
| 7.2.1    | 100 Days CSS Challenge . . . . .                     | 8        |
| 7.3      | At some point, you'll also want to look at . . . . . | 8        |
| <b>8</b> | <b>Final note</b>                                    | <b>9</b> |

### **Abstract**

So you want to learn web development? This post attempts to go through the best way to learn web development as a complete beginner. It gives you guidelines on how to learn web development, what topics to learn in which order and why, and general principles on learning. It also includes suggested pathways and resources. The source for this article can be found at:

<https://dev.to/sargalias/how-to-learn-web-development-2bee>

# 1 Fields of Programming

Before we begin, note that there are different fields of programming. Some examples are:

- Web development
- Games development
- Data science
- Systems programming
- And many more

Exactly which you decide to go for will depend on your goals and interests. If you're not specifically set on one, then web development is a good one to start with.

# 2 Why web development?

There are several reasons why web development is probably the best programming field to get started with:

- Web development is generally considered easier than other fields such as games development and data science.
- Web development teaches you programming just as well as any other kind of development does. It can be a great start to your career regardless of which field you end up in.
- Web development is trending right now. Application development is moving more and more towards web technologies. We have Progressive Web Apps (PWAs) which are becoming more prominent and replacing native apps for many businesses. Also it's possible to make native apps with technologies such as Electron, React Native, etc. These are cross-platform and use web technologies. In short, web development is a very good field right now.
- Web development is a great career. You can potentially get your first job as a front end developer very quickly. Some jobs only require HTML and CSS which are quite fast to learn.

# 3 What is web development?

As it sounds, web development concerns with making websites. But as mentioned above, technologies used in web development can be used for much more than that. They can be used to make Progressive Web Applications, native applications, data visualisations, games and much more.

## 4 Components of Web Development

Web development (and programming in general) can be broadly classified into 3 main parts:

- Front end development
- Back end development
- Dev ops

### 4.1 Front end development

Front end development is basically what a website looks like and what interactivity it has.

- It includes HTML, CSS and JavaScript.
- HTML is the structure and content of a website.
- CSS are the styles of the website, what the website looks like.
- JavaScript affects the interactivity of the website. For example you can use JavaScript to make the website do something special when a button is clicked.

### 4.2 Back end developmen

Back end development happens on the server. The usual functions of the back end are things like:

- Processing forms and payments (e.g. when you make a purchase and submit payment details on a website).
- Returning search results (e.g. when you search for products on Amazon).
- Creating accounts (e.g. when you sign up to a new website).
- And also serving the web page to the web browser.

### 4.3 Dev ops

Dev ops are “peripherals” to web development and other things that increase efficiency of development. For example web hosting, source control, and other things.

## 4.4 Which should you start with?

In my opinion, unless you’ve got a strong preference for back end development, you should start with front end development.

Dev ops isn’t really development, it’s more like peripherals to development, so you don’t start with that so to speak.

Which leaves back end or front end. You can start with either of those, however there are several pros to starting with front end development.

- There are good third party solutions to back end development, such as Google Firebase. You can use these to make complete applications. You could even make a static website or application, which doesn’t even need a back end other than a basic server to send the page. However, other than templates or having someone else do it for you, you can’t really avoid front end development.
- Front end development is easier than back end development. HTML and CSS are definitely easier than JavaScript, or other programming languages, to pick up.
- Learning back end development is much easier after learning front end development. JavaScript is a full programming language. Knowing any programming language makes your second one much easier and faster to learn. Also, if you choose Node as your back end language (which is a great choice), that’s just JavaScript which you’ll already have learned.

## 5 How to learn web development

To learn anything, the most important thing you need to do is to put the time in. In my experience, putting in the time seems much more important than having a good learning plan.

But of course, having a good learning plan can massively accelerate the process.

A bit further down, I list what in my experience is the best path to learning web development, along with the best resources.

But first, let’s talk about what is a bad way to learn web development.

### 5.1 A bad plan

If you search forums and such for the best way to learn web development, you’ll get many answers telling you to “just build stuff”. This is a good thing to do, but as a complete beginner it’s not enough to just build things and learn as you go.

For starters, as a beginner, you don’t know what you don’t know. Until you encounter something by accident, you won’t know it exists and there will be gaps in your knowledge.

So you also need to learn from thorough courses that will expose you to everything you need to know.

And also build stuff along with that.

## 5.2 Learn in iterations

You won't learn everything the first time you encounter it

To be a good web developer there are many things you need to know. When starting out it's not enough to know about something deeply, rather you have to know a little about a lot of things. You need to have a complete base of knowledge.

That's why learning in iterations is best. For example when you go through JavaScript the first time, you'll learn the basics of it. While you may have questions about how some things work behind the scenes, not all of those should be answered right away. You would just be wasting too much time learning about obscure features without having a solid base-level of knowledge.

The most efficient way to learn is in iterations. Learn the basics of a technology. Practice with it for a while and build stuff. Then come back some time later (a few weeks or months), and learn some more. Next time you'll learn about it a level deeper. And so on.

Don't worry, this will feel like a natural process after a while. You'll basically be learning more and more, but instead of learning a topic completely at once, you'll just be learning more about it over time.

## 6 Web development learning plan

### 6.1 Introduction to web development

I suggest starting with the course: Front end masters – Complete Intro To Web Development

This is an 11 hour course (so not too long by web development course standards) that briefly goes over all the important topics of front end web development.

Will you be an expert web developer after doing this course? No. But the benefit of this course is that it will expose you to all the technologies you need to know, so when you do the next courses, things will “click” much more easily. You'll already have encountered them, they'll make sense, and you'll be able to learn them deeper than otherwise.

This is the learn in iterations principle mentioned above.

Don't try to learn this course 100%, it's here as a quick introduction and exposure for you to learn some of the basics.

### 6.2 Learning HTML and CSS

The first step in front end development is learning HTML. This is the easiest of the 3 to learn (HTML, CSS, JavaScript).

HTML is basically just using the correct element for what you want to do. That's about it. For example, if you're writing a paragraph, use the paragraph tag. If you're writing a heading / title, use the heading tag, and so on.

CSS is a bit more difficult, but not too bad. With CSS you basically declare the style of the HTML, how it appears visually. E.g. a declaration of "color: red;" makes the text color appear red. Overall it's also fairly easy to learn the basics of it.

### **6.2.1 The three resources you need**

1. The Web Developer Bootcamp by Colt Steele
2. MDN tutorials
3. Free code camp

Also, always remember to google anything you're stuck on. Google is a programmer's best friend.

1. The Web Developer Bootcamp Probably the best course for beginners at web development.

It teaches you the basics of HTML, CSS, JavaScript, and then continues further with back end and many more topics. Overall it's an amazing course and I highly recommend it. It's the course I started with and it helped me land my first job. Also it has projects, and the teachers and assistants are always there to answer any questions you might have.

2. MDN Probably the best website of all time for learning web development in general, especially HTML. It has pretty much everything you need to know about HTML, CSS, JavaScript and tons of other stuff. While the Web Developer Bootcamp is great for initial learning and getting you up to speed, MDN is much more complete and will teach you in incredible detail.

As a caveat, note that the detail in MDN is extreme and you shouldn't try to learn everything all at once. The first time around I recommend just going through the tutorials and looking up anything you're stuck with. For the most part ignore the references. Come back to those on your second iteration of learning HTML and CSS, or if you get stuck.

3. Free code camp Here you'll basically practice what you learn. It's not all the practice you need, but it's pretty good. The challenges are fairly good at reinforcing what you learned, and the projects are especially good.

### **6.2.2 How to go through the resources**

I recommend doing these three together. Specifically, do the HTML section of The Web Developer Bootcamp, then go through the HTML tutorials in MDN, then finish off with the HTML section on Free Code Camp.



The reason I recommend it in this way is because the Web Developer Bootcamp course teaches with video so it's nice and easy to learn from. Afterwards, the MDN tutorials are very thorough, so they'll teach you HTML in great detail. Finally Free Code Camp is more about practice than learning, so it's good to do also. With Free Code Camp, start the "Responsive Web Design Certification" and do the "Basic HTML and HTML5".

After you finish the HTML section of each resource, repeat the process for the CSS section. For Free Code Camp, finish the "Responsive Web Design Certification".

### 6.2.3 Bonus resources

When you Google stuff, these resources will come up quite often:

**CSS Tricks** a great resource on CSS. It's worth looking at if you're searching for something specifically about CSS.

**Stack Overflow** a resource about development in general. It's a website where people can ask questions and get them answered by other developers. If you have a question, it's very likely that someone else has had it in the past and it has been answered there.

## 6.3 JavaScript

The one course I would recommend for JavaScript is: The Modern JavaScript Bootcamp, by Andrew Mead

Andrew Mead is the best. I have taken some of his other courses and I can confidently say that they are the best courses I've ever done in my entire life on anything. I've left him glowing 5 star reviews. Also I can say with confidence that he is probably the biggest reason I got a great job as a beginner programmer and excelled there. I realise I must sound like a massive fanboy but it's true.

The best thing about this course is that, along with being one of the best JavaScript courses out there, it goes way beyond teaching you just JavaScript. It also teaches you how to set up your machine properly and how to use the command line. The command line is something very easy to learn and extremely helpful for a programmer, but unfortunately some people never learn to use it just because of how difficult it seems. This course takes care of this problem and gives you an invaluable tool for your career as a web developer. The course also touches on Webpack and Babel, which are things you'll definitely be looking at in the future, so it's good to get some initial exposure on these at this stage.

Finally, repeat the process you did for HTML and CSS with the three resources, this time doing the JavaScript sections from them.

- Keep going through the rest of the Web Developer Bootcamp course until you finish all the JavaScript sections. Stop when you reach the "Backend Basics" section. We'll come back to that in the future.

- With Free Code Camp, keep going until you finish the “JavaScript algorithms and data structures certification”.

## 7 Next steps, preparing for a job

After finishing the courses, you’ll have a very solid level of knowledge and you’ll also have had some great practice.

The next steps are to practice a bit more by making some more projects, and to create a portfolio.

The portfolio will be very helpful in helping you get your first job. The fact is that employers don’t know you or your skills personally. They only have your portfolio to judge that you’re able to do the work. So a portfolio is essential.

### 7.1 Build a website for your portfolio

You’ll need a website to host your portfolio on.

There are many ways to build a website:

- Static website (built with a static site generator such as Jekyll, Hugo, or another JavaScript-based one).
- CMS with a custom theme (such as WordPress with a custom theme).
- CMS with a premium theme.
- CMS with a page builder and theme builder (such as WordPress with Elementor Pro).
- Using a site builder such as Wix or Squarespace.

There are more options out there, and most of them are pretty good.

My recommendation: WordPress website with a page builder and theme builder.

My personal recommendation would be to build a WordPress website with Elementor Pro page builder and theme builder, along with the Astra theme. The reason for this is that it’s probably the easiest and fastest way to make a portfolio website for yourself.

Building a completely custom website is good, but it can take very long. Custom development just takes a lot of time. But with something like Elementor Pro you have ready-made templates and components to help you out. You can practically have a website in a few days.

Of course, if you’re up for the challenge, feel free to create a custom website. It will take longer, but the result will be custom made for you and it will be good practice.

Building a basic static website is free, and then you can host it on GitHub pages or Netlify (I recommend Netlify). For that you’ll definitely want to use a static site generator instead, such as Jekyll or Hugo. I recommend Jekyll.

Building it without one of those is possible but much less efficient except for not having to learn how to use a static site generator.

You can also use a CMS such as WordPress and build your own custom theme.

In terms of other site builders like Wix and Squarespace, I don't have any experience with them, so I can't comment.

In terms of hosting and domain names, I recommend hosting by Dreamhost or Siteground and domain names from Hover or Namecheap. I've had good personal experiences with all those services, and they're priced well.

## 7.2 Portfolio and practice

Use YouTube for project walkthroughs.

- Traversy Media is great and has tutorials on building web pages and components.
- Drew Ryan has tutorials on making complete web pages. They're very good and worth following along. He doesn't upload often but the content is relevant.

### 7.2.1 100 Days CSS Challenge

100 Days CSS Challenge

- These are not essential, but feel free to attempt some every once in a while to practice your CSS skills.
- Note that you'll never make these kinds of things in a real website, but the practice is good nonetheless and trains CSS skills you will use on normal websites.

Come up with your own ideas and search the internet for inspiration. Upload them on your website for employers to see.

## 7.3 At some point, you'll also want to look at

- Git
  - Read the tutorials on Atlassian.
- Webpack,
- Babel
- (and optionally Gulp)
  - These are difficult, especially Webpack, so follow along some YouTube tutorials and learn the basics. There is no need to learn these to a high level at this stage, just the basics will do.

- SASS
  - Again, just learn the basics of this for now.
  - The section on SASS on Free Code Camp’s “Front End Libraries” certification should be enough to start with.
  - You’ll also need to learn how to compile SASS into CSS in your projects. Your webpack learning above should take care of that.

## 8 Final note

These are a lot of things to go through, but afterwards you should have a great foundations as a web developer. Remember to take your time and enjoy the process.

If you found this useful, please leave a comment and a like :). I can make part 2 if it’s requested.

Good luck and study well!

## Index

### C

CMS, 7, 8  
Complete Intro To Web Development,  
4

#### courses

Complete Intro To Web Develop-  
ment, 4  
Front End Masters, 4

css, 4–6

CSS Tricks, 6

### D

domain names, 8  
Hover, 8  
Namecheap, 8  
Dreamhost, 8  
Drew Ryan, 8

### E

Elementor Pro, 7

### F

fields of programming, 1  
Free Code Camp, 5, 6  
Front End Masters, 4

### G

GitHub, 7

### H

Hover, 8  
html, 4, 5  
Hugo, 7

### J

javascript, 5, 6  
courses, 6  
Jekyll, 7

### M

MDN, 5  
Mead, Andrew, 6  
Modern JavaScript Bootcamp, The, 6

### N

Namecheap, 8  
Netlify, 7

### P

portfolio, 7, 8  
practice, 5  
projects, 7, 8

### R

Ryan, Drew, 8

### S

site builder  
Squarespace, 7  
Wix, 7  
Siteground, 8  
Squarespace, 7, 8  
Stack Overflow, 6  
static site generators, 7  
Steele, Colt, 5

### T

theme builder  
Elementor Pro, 7  
Traversy Media, 8  
tutorials  
MDN, 5

### W

Web Developer Bootcamp, The, 5, 6  
web development  
courses, 4, 5  
The Modern JavaScript Boot-  
camp, 6  
The Web Developer Bootcamp,  
5  
html and css, 4  
iterations, 4  
path, 3  
portfolio, 7  
process, 5  
projects, 7, 8  
Resources

- MDN, 5
- resources, 6
  - Free Code Camp, 5
  - MDN, 5
- website, 7
- web site
  - hosting, 7
    - domain names, 8
    - Dreamhost, 8
    - Siteground, 8
  - static site generators, 7
- website, 7
  - static, 7
- Wix, 7, 8
- WordPress, 7, 8

## **Y**

- YouTube, 8
  - Drew Ryan, 8
  - Traversy Media, 8