BYU-Idaho Online Learning
Video Transcript

What You Need to Know to be a Front-End Developer in 2018

[One speaker.]

[Porth speaks to the camera, different shots of the studio she's in are shown.]

Treasure Porth: Trends in front-end development are always changing. What worked yesterday may not work today. This can make being your front-end developer pretty challenging, but what's most challenging about being your front-end developer is also what's most exciting: you're always learning new technologies. So if you're an aspiring front-end developer or you're learning front-end technology, here's what you need to know to be a front-end developer in 2018.

[Stock Image of an office, then animation of front end and back end areas.]

First up, what is a front-end developer? There are generally two parts to any web application. The front-end, or client side of the web app is everything you see and interact with in the browser as a user of the application. Then there's the back-end, or server side of the web application where the application's data is saved, updated, manipulated, and sent back to the client side.

Think of your favorite video site. When you click on that video of waterskiing puppies, the video is retrieved from a database and served you by a server. That's the backend. The front end displays it for you in the browser, allowing you to interact with the video by providing buttons to blow up the video to full screen, pause, rewind, or share with your grandma. *[Animation shows these things.]*

As a web developer, you'll likely need to have some knowledge of how the front and back ends of an application interact with each other. However, many companies split these roles into two careers: a front-end developer and a back-end developer.

Back-end developers use languages like Java, Python, Ruby, and PHP to build the behind the scenes functionality of a web site, such as storing and retrieving user data or securely accepting and storing credit card numbers.

A front-end developer is a generic term for someone who builds the user facing parts of websites and applications, using HTML, CSS, and JavaScript. Nearly every business uses a website or application to communicate to its customers. The world needs the skills of a front-end developer. In this video I'm going to talk to you about what you need to know to be a more general but still awesome front-end developer.

[Screen summarizes her words over stock images of developers and animations.]

To be a front-end developer you'll need to learn many different technologies, but the main technologies are HTML, CSS, and the script. HTML, or Hypertext Markup Language is the foundation of every web page. Without it, everything on the internet would be a shapeless wall of text. With HTML, you provide the web browser with information about how your content is structured by defining its different parts. For example, you surround the content of your web pages within different HTML tags to tell the browser which parts are headings, sidebars, a footer, or main area of the site.

One of the goals of HTML is to provide good semantic markup. That is: provide information about what kind of content the HTML tag contains. You use HTML to define which elements are links, list items, buttons, paragraphs, checkboxes, radio boxes, and so on. This not only helps you style a page with CSS, but search engines use this information to understand your pages, and semantic markup helps accessibility tools like screen readers make more sense of your content.

CSS, or Cascading Style Sheets is a design language with a lot of power. CSS helps us describe the presentation of web pages. We use CSS to define how HTML should look and be laid out. For example, we can use CSS to tell the browser that we want our web page to display in a certain number of rows or columns or how much space should be between each element. We'd also use CSS to change the color or font size of a button. Make it wider or taller, bigger or smaller, place it on the left, right or middle of the webpage, and so on. Part of writing good CSS means making sure designs are consistent across different devices. We use CSS to make web pages responsive. That means that they look good and function well on any screen size or device.

CSS seems deceptively easy at first, but it can quickly become challenging. The nuances of formatting and styling are tough, and you've got thousands of options at your disposal. You may need to consult a ton of different references and resources to get things right. It requires a lot of practice.

Javascript is the first programming language that you're likely to learn as a front-end developer. It's a continuously evolving language that's used nearly everywhere these days. On the front-end, back-end, on mobile, and even desktop devices. Like everything else it can be challenging but fun to learn.

Generally front-end developers write JavaScript for the browser using the Dom, or Document Object Model. The Document Object model is like a representation of your web page or application that allows JavaScript and the browser to communicate. Each HTML element on your web page is represented by the Dom and you can use JavaScript to manipulate those Dom elements. For example, say you have some text on your web page that reads “click me.” You'd use HTML tags to define that text as a button. With CSS, you'd control what that button looks like. Finally, with JavaScript, you'd write code that triggers the page or application to do something every time the button is clicked, like play a video or submit a comment to a social media post.

If you're serious about being a front-end developer, JavaScript is a language you'll want to master. These are some of the main tools and technologies that you'll use on a daily basis as a front-end developer.

[Screen reads: Front End Tools, Text Editor, Browser Development Tools, Version Control.]

The text editor is where you'll spend most of your time. It's where you write the code for a given site or app. Microsoft's Visual Studio Code, Atom, or Sublime Text are popular choices in 2018. Don't worry too much about which one you should use. You can try them all and figure out which one you like best. Developer tools for Google Chrome and Firefox lets you look at any website and see how it's structured. Dev tools help you evaluate and debug your applications, step through JavaScript code, and manipulate and experiment with your application CSS.

Git is a popular version control system. Version control systems help you to track and save changes to your project, so that you can easily manage different versions and features. You can also revert to previous versions of your projects or work on a single codebase as part of a team. Github is a specific service that allows you to share your code online, as well as fork, or make copies of other existing code bases, so you can work on them yourself or see how they're made. Let's talk about a few more technologies that are common for front-end developers to learn.

[Screen reads: Other Important Technologies, Sass/CSS Preprocessors, Javascript Libraries, JS Frameworks, Command Line CLI, Module Loading/Bundling Tools, WordPress, Image Editors, Wireframe Tools.]

Once you have a good understanding of CSS, you'll be ready to learn Sass. Sass is what's known as a CSS preprocessor. Basically, it adds useful features to CSS that make writing, organizing, and maintaining the CSS easier, a lot easier.

[Screen shows examples of each website talked about.]

Let's talk about JavaScript libraries and frameworks. jQuery has been around since 2006 and is still widely used on a lot of websites. It's not the newest library but still worth knowing as a front-end developer, as it lets you quickly and easily add complex JavaScript interactions to your webpages.

Newer JavaScript libraries and frameworks include React, Angular, and Vue.js. These are all designed to make it easier to create and maintain complex web applications. Typically those that display large amounts of data, or complex data, have complex user interfaces, or are made up of many different UI elements and components.

Twitter Bootstrap is a design framework that provides a variety of design templates for common website and web application components, like typography, forms, buttons, and navigation menus. By adding pre-styled bootstrap CSS classes to your HTML, you can quickly layout and style your web pages. Bootstrap makes it easy to drop in and set up common user interface components like photo carousels and modal windows.

Learning how to use the terminal, or the command line, will give you greater control over the functionality of your computer. It'll help you install different software packages and libraries, as well as use version control systems, like Git, more quickly and efficiently than using a graphical user interface. No matter what kind of development you do, front-end or back-end, knowledge of the terminal will help you succeed.

[Different animations of the bundling tools appear as they are discussed.]

As your projects grow larger and more complex, you'll find that they're made up of many different parts, frameworks, modules, libraries, and tooling. Bundling tools like Web Pack and Browserify with Gulp help you minify and bundle your code so it can be loaded quickly and efficiently into the browser, helping your applications load faster.

25% of the web is WordPress. WordPress is a content management system that is free and open-source. WordPress is extremely versatile, and while it has a reputation as a blogging platform, you can extend the functionality of any wordpress site by using any of the thousands of free and open source plugins.

Photoshop is the behemoth when it comes to editing images for the web, but Sketch is a great lightweight tool that is designed for creating vector images. Sketch is a modern tool built for the web. Both are paid tools that require a perpetual license.

Wireframes are your plans or sketches for websites before you begin to build them. Both Balsamiq and Figma have cloud-based tools that let you create and share an idea in its early stages before you commit the time to creating it in full. As a front-end developer, you'll likely build webpages and applications based on a design or wireframe provided by a web designer, rather than designing directly with these tools yourself. There can be some overlap, however, so it's helpful for a front-end developer to know the basics of using these technologies.

This was a long list and you may be asking yourself, “How will I ever learn all this stuff?” It can feel daunting especially when the tools and technologies of 2018 are likely to change in 2019 and beyond. Just know that it's normal to feel this way.

Take things one step at a time, create a learning strategy, and commit to learning and practicing as much as you can. The concepts, principles, and tools that you learn from one programming language or framework will make it much much easier to learn similar concepts, principles, and tools for another language or framework.

So don't feel like you need to learn everything all at once. There are tons of resources for learning how to be a front-end developer - podcasts, video tutorials, books, apps, conferences, meetups - just tons of things. And while it's a challenge to filter out what's most valuable, as most of these resources are going to be talking about things you won't understand just yet, immerse yourself in the language and take it all in.

Concentrate at first by getting comfortable with the basics: HTML, CSS, and JavaScript, and the rest will come in time. Learning a new technology is hard and one of the hardest things about approaching a career like front-end development is figuring out what you need to know when you don't know what you need to know.

I'm a teacher at Treehouse, an online school where you learn at your own pace. At $25 a month, you get access to the front-end web development track, which has on-demand content that includes videos, quizzes, and code challenges that will teach you HTML, CSS, JavaScript, and more, everything you need to know to start out as a front-end web developer. Click the link at the top of the description of this video to start your seven day free trial at Treehouse and get started on the front-end web development track and I'll see you there. I teach the first course.

[End of video.]