

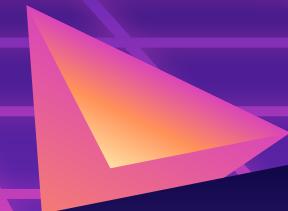
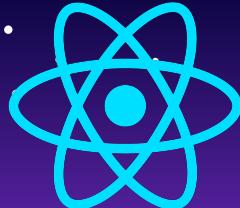


JS



Front-End Development With JavaScript

By Kaleb Dykema



Who am I?

**Former Audio
Engineer and
Musician**

Moved to Franklin, TN
in 2017



**Transitioned
to Dev in
2019**

Starting off doing
freelance work

What is front-end development?

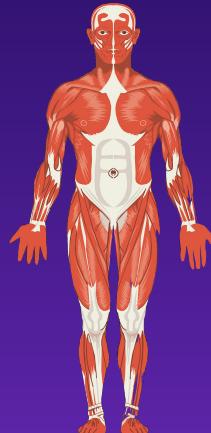
Defined by FreeCodeCamp: “A front-end developer is someone who works on software, like desktop or mobile websites, that users interact with. Specifically, a front-end developer spends time on the side of technology that the user touches and sees.” * <https://www.freecodecamp.org/news/what-is-front-end-development/>

Front-end development most popularly utilizes 3 technologies

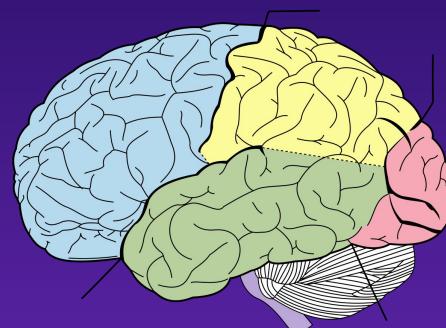
HTML



CSS



JavaScript



Choose an IDE (Integrated Development Environment)



VS Code

A general use text editor specialized for coding.
The most popular IDE.
This is free.



WebStorm

An IDE heavily specialized for JavaScript development.



Sublime

A general use text editor.
Differs from VS Code with better performance.



HTML

“HTML (HyperText Markup Language) is the most basic building block of the Web. It defines the meaning and structure of web content.” ^{1.}<https://developer.mozilla.org/en-US/docs/Web/HTML>

- HTML is the skeleton of websites.
- This is used to define the structure of a web page.
- It lacks much functionality and looks very bland without the use of CSS.

```
6 <body>
7   <header>
8     <h2>HTML Only Site</h2>
9     <nav>
10    <a href="#">index.html">Home</a>
11    <a href="#">about.html">About</a>
12    <a href="#">contact.html">Contact</a>
13  </nav>
14 </header>
15 <br>
16
17 <main>
18   <h3>This is my site!</h3>
19   <p>Lorem ipsum dolor sit amet consectetur, adipisicing elit. Quo voluptatem voluptas possimus. Eum laudantium, magnam voluptas veniam inventore aperiam. Neque excepturi iure odio
20      praesentium nulla incidunt modi quam repellendus doloribus.</p>
21 </main>
22 <br>
```

HTML Tidbit:

HTML is not a programming language. It's a markup language, which just means it's used for formatting a document.

Resources to Learn HTML

- [Traversy Media's HTML Crash Course](#) (~1 hour) - A great video for a quick overview of HTML.
- [Net Ninja's HTML & CSS Crash Course](#) (~1 hour) - The first three videos in this playlist cover HTML. If you're not a fan of Traversy's style, Net Ninja is a great alternative.
- [MDN Web Docs: HTML Basics](#) - A single article from a series of articles published on Mozilla Developer Network.



CSS



“Cascading Style Sheets (CSS) is a stylesheet language used to describe the presentation of a document written in HTML or XML.” <https://developer.mozilla.org/en-US/docs/Web/CSS>

- CSS is the muscles of websites.
- This is used to define the look and style of a web page.
- While it does have some tools to implement basic functionality in a site, it's still very limited.
- CSS is also used to make sites “responsive,” or scale appropriately depending on what device your looking at the site in (phone, tablet, desktop).

```
47  footer {  
48    width: 100%;  
49    background-color: var(--primaryOrange);  
50    display: flex;  
51    justify-content: center;  
52    align-items: center;  
53    height: 4rem;  
54    position: fixed;  
55    bottom: 0;  
56    font-weight: bold;  
57  }
```

Resources to Learn CSS

- [FreeCodeCamp Full CSS Course](#) (~11 hours) - A great video for a good, in-depth course covering CSS.
- [Net Ninja's HTML & CSS Crash Course](#) (~3 hours) - Videos four through eleven in this playlist covers CSS. A quicker way to get an introduction/overview to CSS.
- [Code Academy: Learn CSS](#) - An interactive course dedicated to learning CSS from the beginning.

JavaScript



“JavaScript, often abbreviated as JS, is a programming language that is one of the core technologies of the World Wide Web, alongside HTML and CSS. As of 2022, 98% of websites use JavaScript on the client side for webpage behavior, often incorporating third-party libraries.” *<https://en.wikipedia.org/wiki/JavaScript>

- JavaScript is the brain of a website.
- This is used to add functionality to a website, such as the ability to grab data from a server and deliver it to a website. A great example of this is login systems.
- JavaScript is a high-level language, meaning many of the complexities of programming are abstracted away from developers, such as memory management.
- It has a robust system of libraries that other developers have created for anyone to use. These libraries can help accomplish various tasks in easier ways. Great examples of this are JQuery, React, or Lodash.

```
15  entries.forEach((entry) => {
16    // Reset errors
17    toggleError(entry[0], false)
18
19    // If we're missing any values, display the error and mark the form as invalid
20    if (!entry[1]) {
21      valid = false
22      toggleError(entry[0], true)
23    }
24  })
```

JS Tidbit:

JavaScript has nothing to do with Java. It was only called JavaScript for marketing purposes, since Java was popular back when JavaScript was created.

*<https://www.youtube.com/watch?v=XOmhtfTrRxc&t=125s>

Resources to Learn JavaScript

- [FreeCodeCamp Full JavaScript Course](#) (~8 hours) - A great video for a good, in-depth beginner course covering JavaScript.
- [Net Ninja's Modern JavaScript Course](#) (~5 hours) - A slightly shorter course from a creator that does a great job of teaching..
- [Traversy Media's JavaScript Crash Course](#) (~2 hours) - A great video for a quick overview of JavaScript.
- [MDN Web Docs: JavaScript Basics](#): A single article from a series of articles published on Mozilla Developer Network.

CSS Frameworks/Libraries



Bootstrap

Created by Twitter, intended to align with the CSS standards followed by the company.



Materialize

Created by Google, intended to align with their design standards used in their sites and apps.



Tailwind

Independently created, meant to be used more as a set of helper classes rather than define the look of your site.



Other Important Technologies



Git



Sass

SASS



Node/NPM

Git

“Git is a distributed version control system that tracks changes in any set of computer files, usually used for coordinating work among programmers collaboratively developing source code during software development.” *<https://en.wikipedia.org/wiki/Git>

- Git is most common associated with GitHub, a website where you can store your code. GitLab and BitBucket are two other options.
- This helps keep track of changes to your code over a longer period of time.
- You can use this to work on separate pieces of code from other developers on a team without interfering with each other’s work. This is commonly done with “feature branches.”
- While other version control software exists, almost everyone uses Git





SASS

“Sass is a CSS pre-processor. Sass reduces repetition of CSS and therefore saves time.” <https://www.w3schools.com/sass/>

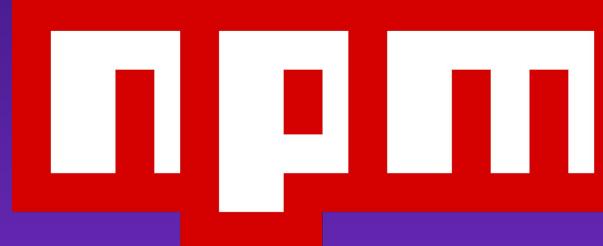
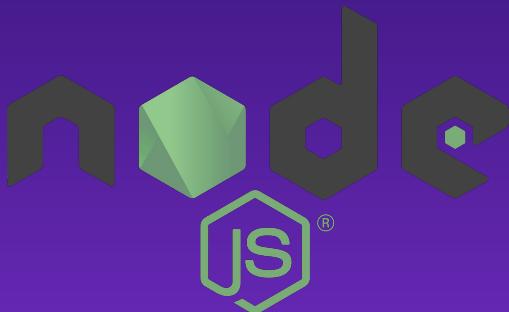
- SASS is an alternative to CSS to allows for cleaner CSS.
- This allows for nesting styles, something that is not available in CSS.
- Additionally, you can create functions and mixins to add more reusable pieces of styling that you can use throughout SASS
- Browsers cannot run SASS, so SASS is compiled down into CSS in order to run in a browser.



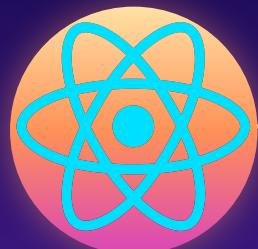
Node/NPM

“ Node.js is a back-end JavaScript runtime environment, runs on the V8 JavaScript Engine, and executes JavaScript code outside a web browser.” *<https://en.wikipedia.org/wiki/Node.js>

- Node.js allows use of JavaScript outside of the browser.
- It can be used to create command line interfaces (software you run in a terminal) or backend APIs.
- Included in the install for Node is NPM, or Node Package manager, which you can use to install and manage third party libraries in a project or a computer.
- This includes a unique API, or code you can use that is unique to Node and not available in regular JavaScript. For example, the “fs” module that is included with Node allows you to manipulate a file system on a machine.
- Node does not have access to many browser APIs, such as the DOM API, which is used to create, read, update, or delete HTML and CSS in a website.



Single Page Application Frameworks



React



Vue



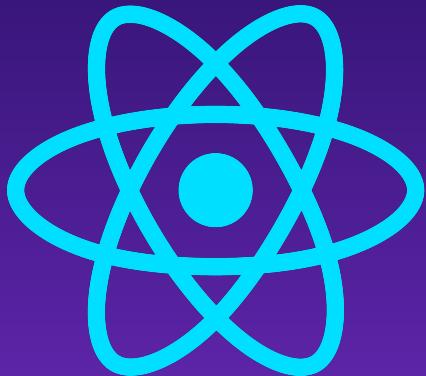
Angular



Svelte

React

- Created by Facebook/Meta
- Most popular front-end framework. Most front-end jobs are for React developers.
- Very lightweight in functionality, with the expectation that you'll build on top of it with other libraries for specific uses, such as React Router and Redux.
- Utilizes JSX, which is a syntax used to write JavaScript mixed with your HTML.
- Medium in difficulty for beginners when compared to other frameworks.





Vue



- Created by Evan You
 - Very popular outside of enterprise use. However, there are still plenty of Vue jobs available.
 - Lightweight component library, similar to React, with the expectation that you'll build on top of it with other libraries for specific uses, such as Vue Router and Pinia.
 - Rather than utilize JSX by default (you can still set this up), it uses custom “bindings” to apply data to the visible piece of site and receive data back from it.
 - Very easy in difficulty for beginners when compared to other frameworks. This can largely be attributed to their amazing documentation and official tutorial.
- 



Angular



- Created by Google
- Very popular for enterprise use. This is less popular outside of enterprise use when compared to other frameworks.
- Extremely fleshed out. Includes APIs and libraries you need to do almost anything you would need to do with a front-end framework
- Very structured in how you have to work with it. While the other frameworks are more of libraries, providing you lots of freedom around how to accomplish tasks, Angular provides you with more structure and “strict” ways of doing thing.
- Hard in difficulty for beginners when compared to other frameworks.
- NOTE: Rather than using JavaScript, this uses TypeScript, which we'll talk about later.



Svelte

- Created by Rich Harris
- Very popular for personal use. This is still really new, so not many jobs exist for it yet. However, it's climbing in popularity very quickly.
- Svelte is just a component library, but there's also SvelteKit, which is a full fledged framework that provides tools to perform many tasks you would need to do with a front-end framework.
- Has a focus on making it feel like working with vanilla HTML, CSS, and Javascript, unlike the other major frameworks.
- Great on performance compared to the other frameworks.
- Starts easy, becomes medium in difficulty for beginners when compared to other frameworks. It's easy to learn the basics for, but can become harder to learn the more advanced pieces.



Bundlers and Toolkits



Webpack

The most popular option.
Easy to use and configure,
but doesn't perform well at
scale.



Rollup

Far from beginner friendly.
Hard to use and configure,
but can be much more
powerful than Webpack.



Vite

Easiest to use and fast. A
recent creation that has
quickly gained popularity.
Can use without a
configuration for basic sites.



Gulp

A toolkit to automate various
JS-related tasks, such as
compiling SASS into CSS. Very
manual, and not a bundler
unlike the rest.

TypeScript

- Created by Microsoft
- Adds static typing into JavaScript (such as strings, integers, and float) along with similar concepts such as interfaces, enums, tuples, and even custom types.
- TypeScript is compiled into JavaScript, as web browsers cannot run TypeScript directly.
- It's much better at detecting potential errors in your code and helps prevent many bugs that would be more likely to pop up in a JavaScript project.
- Quickly gaining popularity both in enterprise and personal use. There are lots of TypeScript developer jobs, and most JavaScript libraries are actually written in TypeScript nowadays.
- Self-documents your code when you use the features given to you by the language.
- Any JavaScript code is valid TypeScript code, but not the other way around. However, code made in TypeScript can still be used by JavaScript (through the use of bundlers and tooling), which is why you're able to use libraries written in TypeScript within JavaScript.
- Can be used with nearly all of the major frameworks and libraries.
- Easy to learn if you already know JavaScript. Not recommended for devs that don't know JavaScript



Other Tech, Tools, and Concepts

- Dev Tools
 - Chromium Browsers (Google Chrome, Edge, Brave, Opera)
 - Firefox
 - Safari
- Figma Basics
- Practice Algorithms Used in Coding Interviews
- Bash/Zsh or Cmd
- Docker
- Basic Backend Skills with Node.js
- Open Source
- Electron (for desktop applications)
- React Native (for mobile applications) and similar tools
- Jest (for unit tests)
- Cypress (for end-to-end and integration tests)
- GraphQL (REST API alternative)





Tips For Beginners

- Take your time, and take breaks.
- Take your time getting very used to CSS and JavaScript before learning any CSS or JavaScript frameworks.
- Don't rush into new technologies when you haven't finished learning the basics of a previous one.
- Ask questions and search up questions and solutions online.
- Once you start getting a good understanding of JavaScript and finished your basic learning, spend some time to learn about calculating cognitive complexity and then begin practicing interview problems on sites like leetcode.com.
- Persistence is key when/if looking for your first job. It will take a while, and you will apply to a ton of jobs before even hearing back. Networking and getting someone to refer you is your best option, if possible. Utilize the help of recruiters.
- Have a hobby and take care of yourself. "Living and breathing code" is hard to do and not sustainable for most people. You need to work on other things.

Learning Resources

- Code Demos Used in this Presentation:
<https://github.com/KalebDykema/FrontEndDevelopmentWithJavaScriptPresentation>
- MDN Web Docs: <https://developer.mozilla.org/>
- Stack Overflow: <https://stackoverflow.com/>
- FreeCodeCamp: <https://www.freecodecamp.org/> and
<https://www.youtube.com/@freecodecamp>
- CodeAcademy: <https://www.codecademy.com/>
- Traversy Media: <https://www.traversymedia.com/> and
<https://www.youtube.com/@TraversyMedia>
- Net Ninja: <https://netninja.dev/> and <https://www.youtube.com/@NetNinja>
- Fireship: <https://fireship.io/> and
<https://www.youtube.com/channel/UCsBjURrPoezykLs9EqgamOA>
- W3 Schools: <https://www.w3schools.com/>

Image Sources

Google Slide Theme - slidesgo.com

JavaScript, React, and Node Logos - logos-download.com

VS Code Logo - code.visualstudio.com

WebStorm Logo - logonoid.com

Sublime Logo - sublimetext.com

HTML, CSS, and SASS - wikipedia.org

Vue, Angular, and Jest Logos - freebiesupply.com

Skeleton and Brain Diagrams, Chrome, Edge, Opera, Firefox, Safari, Figma, Bash, Docker, and TypeScript Logos - commons.wikimedia.org

Muscles Diagram - medicalmediareview.com

NPM Logo - logolynx.com

Webpack Logo - betterprogramming.pub

Vite Logo - vitejs.dev

Gulp Logo - gulpjs.com

Rollup Logo - rollupjs.org

Bootstrap Logo - getbootstrap.com

Materialize Logo - seeklogo.com

Tailwind Logo - iconape.com

Brave Logo - brave.com

Cypress Logo - cypress.io

GraphQL - graphql.org

Contact Me

dykemakaleb@gmail.com

<https://www.linkedin.com/in/kalebdykema/>

<https://github.com/KalebDykema>

CREDITS: This presentation template was created by Slidesgo, including icons by Flaticon, and infographics & images by Freepik and illustrations by Stories.

