Step 1: Learn Basic JavaScript

Before you can become a senior JavaScript developer, you need to have a strong foundation in basic JavaScript. Here are some sub-steps to follow:

* Learn the basics of JavaScript syntax, data types, and control structures.
* Learn about functions, closures, and higher-order functions.
* Understand how to work with arrays and objects in JavaScript.

Resources:

* FreeCodeCamp JavaScript Course:

<https://www.freecodecamp.org/learn/javascript-algorithms-and-data-structures/basic-javascript/>

* Eloquent JavaScript Book: <https://eloquentjavascript.net/>

Step 2: Learn Front-End Technologies

Once you have a solid foundation in basic JavaScript, the next step is to learn front-end technologies such as HTML, CSS, and front-end frameworks. Here are some sub-steps to follow:

* Learn HTML and CSS to build static web pages.
* Learn a front-end framework such as React, Angular, or Vue.js.
* Learn about responsive design and how to build mobile-first web applications.

Resources:

* FreeCodeCamp HTML and CSS Course:

<https://www.freecodecamp.org/learn/responsive-web-design/>

* React Docs: <https://reactjs.org/docs/getting-started.html>
* Angular Docs: <https://angular.io/docs>
* Vue.js Docs: <https://vuejs.org/v2/guide/>

Step 3: Learn Back-End Technologies

After learning front-end technologies, the next step is to learn back-end technologies such as Node.js, Express.js, and databases. Here are some sub-steps to follow:

* Learn Node.js, which is a popular back-end runtime environment for JavaScript.
* Learn Express.js, which is a popular Node.js framework for building server-side applications.
* Learn about databases such as MongoDB, MySQL, or PostgreSQL.

Resources:

* Node.js Docs: <https://nodejs.org/en/docs/>
* Express.js Docs: <https://expressjs.com/>
* MongoDB Docs: https://docs.mongodb.com/

Step 4: Learn Advanced JavaScript Concepts

To become a senior JavaScript developer, you need to have a deep understanding of advanced JavaScript concepts. Here are some sub-steps to follow:

* Learn about closures, prototypes, async/await, generators, and Promises.
* Learn how to work with JavaScript modules and ES6 features.

Resources:

* You Don't Know JS Book Series: <https://github.com/getify/You-Dont-Know-JS>
* ES6 Features: <https://babeljs.io/docs/en/learn/>

Step 5: Learn Web Performance and Security

To build high-quality web applications, you need to understand how to optimize performance and ensure security. Here are some sub-steps to follow:

* Learn about web performance optimization techniques such as code splitting, caching, and lazy loading.
* Learn about web application security principles such as Cross-Site Scripting (XSS) and Cross-Site Request Forgery (CSRF) and how to prevent them.

Resources:

* Google Web Fundamentals: <https://developers.google.com/web/fundamentals>
* OWASP Top Ten Project: <https://owasp.org/Top10/>

Step 6: Learn DevOps Processes

To become a senior JavaScript developer, you need to understand DevOps processes such as continuous integration and deployment (CI/CD). Here are some sub-steps to follow:

* Learn about version control systems such as Git.
* Learn about CI/CD processes and tools such as Jenkins and Travis CI.

Resources:

* Git Docs: <https://git-scm.com/doc>
* Jenkins Docs: <https://www.jenkins.io/doc/>