Certainly! Here are some resources for the points mentioned in the JavaScript Developer Skills Set List:

1. **Core JavaScript:**
   * MDN Web Docs - JavaScript Guide: <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide>
2. **ES6 (ECMAScript 2015) and Beyond:**
   * Exploring ES6 by Dr. Axel Rauschmayer: <https://exploringjs.com/es6/>
   * ES6 Features Overview: <https://www.javascripttutorial.net/es6/>
3. **DOM Manipulation:**
   * MDN Web Docs - Working with the DOM: <https://developer.mozilla.org/en-US/docs/Web/API/Document_Object_Model>
4. **Asynchronous Programming:**
   * JavaScript Promises: <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Using_promises>
   * Async/Await: <https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Asynchronous/Async_await>
5. **Client-Side Frameworks and Libraries:**
   * React: <https://reactjs.org/docs/getting-started.html>
   * Angular: <https://angular.io/docs>
   * Vue.js: <https://vuejs.org/v2/guide/>
6. **Frontend Development:**
   * MDN Web Docs - HTML: <https://developer.mozilla.org/en-US/docs/Web/HTML>
   * MDN Web Docs - CSS: <https://developer.mozilla.org/en-US/docs/Web/CSS>
7. **AJAX:**
   * AJAX Introduction: <https://www.w3schools.com/xml/ajax_intro.asp>
   * MDN Web Docs - Fetch API: <https://developer.mozilla.org/en-US/docs/Web/API/Fetch_API>
8. **RESTful APIs:**
   * MDN Web Docs - Using Fetch: <https://developer.mozilla.org/en-US/docs/Web/API/Fetch_API/Using_Fetch>
9. **JSON (JavaScript Object Notation):**
   * JSON Introduction: <https://www.json.org/json-en.html>
   * MDN Web Docs - JSON: <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/JSON>
10. **Package Managers:**
    * npm: <https://www.npmjs.com/get-npm>
    * yarn: <https://yarnpkg.com/getting-started>
11. **Bundlers:**
    * Webpack: <https://webpack.js.org/guides/getting-started/>
    * Parcel: <https://parceljs.org/getting_started.html>
12. **Version Control:**
    * Git Handbook: <https://guides.github.com/introduction/git-handbook/>
13. **Testing and Debugging:**
    * Jest: <https://jestjs.io/docs/getting-started>
    * MDN Web Docs - Debugging JavaScript: <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Statements/debugger>
14. **Cross-Browser Compatibility:**
    * MDN Web Docs - Cross-Browser Compatibility: <https://developer.mozilla.org/en-US/docs/Web/Progressive_web_apps/Cross-browser_compatibility>
15. **Performance Optimization:**
    * Google PageSpeed Insights: <https://developers.google.com/speed/pagespeed/insights/>
    * MDN Web Docs - Performance: <https://developer.mozilla.org/en-US/docs/Web/Performance>
16. **Security Best Practices:**
    * OWASP Top Ten: <https://owasp.org/www-project-top-ten/>
17. **Progressive Web Apps (PWAs):**
    * Google Developers - Progressive Web Apps: <https://developers.google.com/web/progressive-web-apps>
18. **Code Quality and Code Style:**
    * ESLint: <https://eslint.org/docs/user-guide/getting-started>
19. **Task Runners:**
    * Gulp: <https://gulpjs.com/docs/en/getting-started/quick-start>
20. **Responsive Design:**
    * MDN Web Docs - Responsive Web Design: <https://developer.mozilla.org/en-US/docs/Web/Progressive_web_apps/Responsive/responsive_design_building_blocks>
21. **Web Accessibility (A11y):**
    * Web Content Accessibility Guidelines (WCAG): <https://www.w3.org/WAI/standards-guidelines/wcag/>
22. **Cross-Platform Development:**
    * React Native: <https://reactnative.dev/docs/getting-started>
    * Ionic Framework: <https://ionicframework.com/getting-started>