**Debugging**

**JavaScript typically involves identifying and resolving errors or unexpected behavior in your code. Here are several techniques and tools you can use for debugging JavaScript:**

**Using console.log():**

**The simplest and most common way to debug JavaScript code is by using console.log() statements to print out variable values, object properties, or any other information you want to inspect.**

a = 5;

b = 6;

c = a + b;

console.log(c);

**Browser Developer Tools:**

**Most modern web browsers come with built-in developer tools that include powerful debugging capabilities. You can use features like breakpoints, step-by-step execution, watch expressions, and the console to inspect and debug your JavaScript code.**

**In Chrome, you can access Developer Tools by right-clicking on the page and selecting "Inspect" or pressing Ctrl + Shift + I (Windows/Linux) or Cmd + Opt + I (Mac).**

**Debugger Statement:**

**You can insert the debugger statement into your code to pause execution at a specific point and allow you to inspect the program's state using browser developer tools.**

*function* myFunction() {

    // Some code

    debugger;

    // More code

}

*let* x = 15 \* 5;

debugger;

console.log(x);

**Using debugger Keyword:**

**Similar to the debugger statement, you can use the debugger keyword directly in your code. When the browser encounters this keyword, it will pause execution and allow you to debug.**

**Error Messages:**

**Pay attention to error messages in the browser console or the Node.js terminal. They often provide valuable information about what went wrong and where the error occurred.**

**Network Tab:**

**If your JavaScript code interacts with server-side components, the network tab in browser developer tools can help you debug issues related to network requests and responses.**

**Linters:**

**Use JavaScript linting tools like ESLint or JSHint to catch potential errors, enforce coding conventions, and improve code quality.**

**Unit Testing:**

**Write unit tests for your JavaScript code using testing frameworks like Jest, Mocha, or Jasmine. Unit tests help you identify and fix bugs early in the development process.**

**By using a combination of these techniques and tools, you can effectively debug your JavaScript code and ensure its correctness and reliability.**