**Browser errors were logged to the console**

May 2, 2019 • Updated Aug 28, 2019

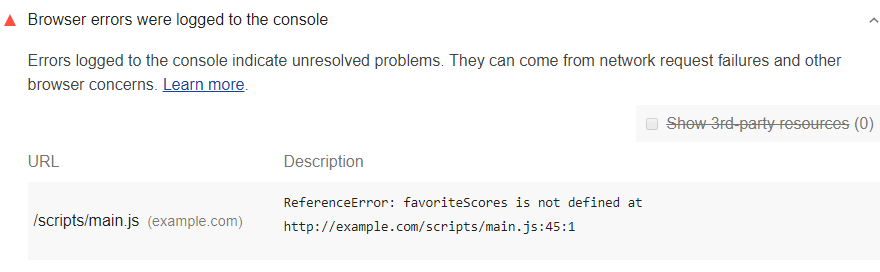
Appears in: [Best Practices audits](https://web.dev/lighthouse-best-practices)

Most browsers ship with built-in developer tools. These developer tools usually include a [console](https://developers.google.com/web/tools/chrome-devtools/console/). The console gives you information about the page that's currently running.

Messages logged in the console come from either the web developers who built the page or the browser itself. All console messages have a severity level: Verbose, Info, Warning, or Error. An Error message means there's a problem on your page that you need to resolve.

**How the Lighthouse browser error audit fails** [**#**](https://web.dev/errors-in-console/?utm_source=lighthouse&utm_medium=devtools#how-the-lighthouse-browser-error-audit-fails)

[Lighthouse](https://developers.google.com/web/tools/lighthouse/) flags all browser errors logged to the console:



Each Best Practices audit is weighted equally in the Lighthouse Best Practices Score. Learn more in [The Best Practices score](https://developers.google.com/web/tools/lighthouse/v3/scoring#best-practices).

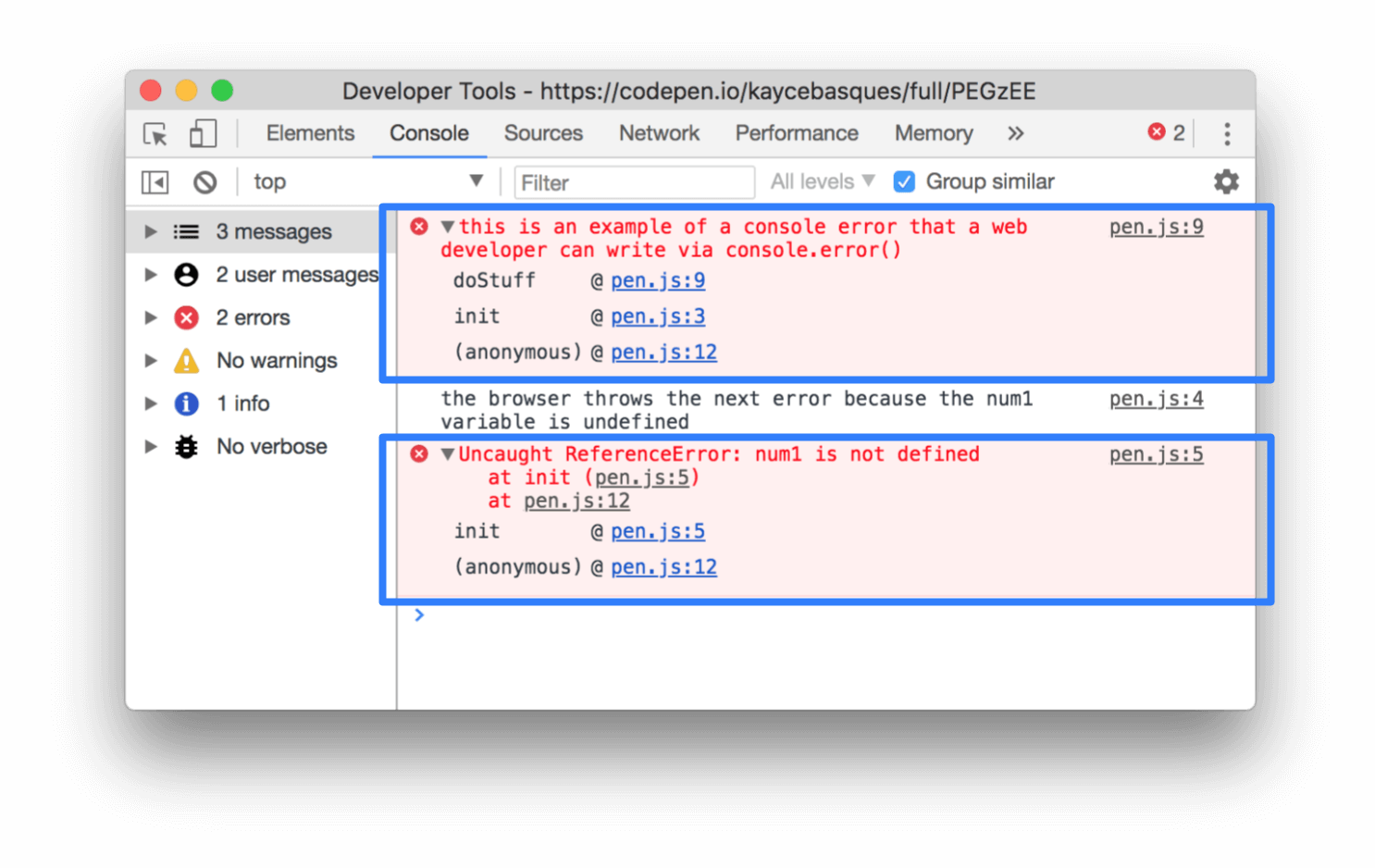
**How to fix browser errors** [**#**](https://web.dev/errors-in-console/?utm_source=lighthouse&utm_medium=devtools#how-to-fix-browser-errors)

Fix each browser error that Lighthouse reports to ensure that your page runs as expected for all your users.

Chrome DevTools includes a couple tools to help you track down the cause of errors:

* Below the text of each error, the DevTools Console shows the [call stack](https://developer.mozilla.org/docs/Glossary/Call_stack) that caused the problematic code to execute.
* A link at the top-right of each error shows you the code that caused the error.

For example, this screenshot shows a page with two errors:



In the example above, the first error comes from a web developer via a call to [console.error()](https://developer.chrome.com/docs/devtools/console/api/#error). The second error comes from the browser and indicates that a variable used in one of the page's scripts does not exist.

Below the text of each error, the DevTools Console indicates the call stack in which the error appears. For example, for the first error the Console indicates that an (anonymous) function called the init function, which called the doStuff function. Clicking the pen.js:9 link in the top-right of that error shows you the relevant code.

Reviewing the relevant code for each error in this way can help you identify and resolve possible problems.

If you can't figure out the cause of an error, try entering the error text into a search engine. If you can't find solutions to your problem, try asking a question on [Stack Overflow](https://stackoverflow.com).

If you can't fix an error, consider wrapping it in a [try…catch](https://developer.mozilla.org/docs/Web/JavaScript/Reference/Statements/try...catch) statement to explicitly indicate in the code that you're aware of the issue. You can also use the catch block to handle the error more gracefully.