**jsLint:**

jsLint homepage is an online quality-check tool, with lots of options for syntax analysis – a good way to quickly test your code.

jsLint is written in Javascript, and so it can be run anywhere where Javascript can be executed.

So, jsLint is to be a good utility, but it doesn’t have a quick-start and some time should be spent on research/tuning the utility for your own use.

**jSHint:**

jsHint is same as jsLint. Same web tool for code analysis, some options to check. But its homepage looks much better.

All necessary documentation and links are present and can be easily accessed without need for some waiting or approval.

Different platforms are also actively supported, plugins present for most of popular IDE’s and editors, local environments are presented by Rhino, Node, JavaScriptCore.

The main difference between jsHint and jsLint is that jsHint is customizable. You can enable/disable all of the style features. It’s not that strict as jsLint,

**.Result comparsion:**

jsLint shown maximum amount of errors by default. It’s unhappy with our variable naming conventions, use of unsafe Regexp, use of ‘arguments’ variable directly and lots of other things.

jsHint also caught some of same errors as jsLint, bt its depend on you.

**Conclusion:**

**jsLint**

Really thorough in searching different javascript bugs, insecurities and also checks syntax problems. The thing is, that this tools goes by strict set of rules, and if you agree with them and use jsLint right from the project start – it will work out .

**jsHint**

Very customizable. it can check both syntax problems and search for possible bugs, but can become much stricter if you need it.

If you need to set some custom code conventions for your team – this is the tool for you. You can always tune the set of rules however you want, and even make it as strict as jsLint**.**

* ***The most important difference is that JSHint is developed and supported by the JavaScript developer community and not by one very opinionated person.***

JSHint works in the browser as well as in environments like Node.js and Rhino, and has many options so you can control what errors you want to see.

## What It Does

**Here are some common mistakes that JavaScript Lint looks for:**

* Missing semicolons at the end of a line.
* Curly braces without an if, for, while, etc.
* Case statements in a switch that do not have a break statement.
* Statements that don't do anything**.**
* **Who uses JSHint?**

**Engineers from these companies and projects use JSHint:**

* [**Mozilla**](https://www.mozilla.org/)
* [**Wikipedia**](https://wikipedia.org/)
* [**Facebook**](https://facebook.com/)
* [**Twitter**](https://twitter.com/)
* [**Bootstrap**](http://getbootstrap.com/)
* [**Yahoo!**](https://yahoo.com/)
* [**jQuery**](http://jquery.com/)
* [**PDF.js**](http://mozilla.github.io/pdf.js)
* [**Adobe Brackets**](http://brackets.io/)
* [**Apache Cordova**](http://cordova.io/)
* [**RedHat**](http://redhat.com/)
* [**Cloud9**](http://c9.io/)