Linting tool for JavaScript - Jshint

JSHint is a program that flags suspicious usage in programs written in JavaScript. **JSHint** is a static code analysis tool used in software development for checking if JavaScript source code complies with coding rules.

It was forked from JSLint project, as it was felt that the original did not allow enough customization options. There is also an internet version available at its official website in which users can paste code to run the application online. A command-line version of JSHint, distributed as a Node.JS module, makes it possible to automate one's "linting" process and integrate JSHint into the website's development workflow.

It is very flexible so you can easily adjust it to your particular coding guidelines and the environment you expect your code to execute in.

JSHint scans a program written in JavaScript and reports about commonly made mistakes and potential bugs. The potential problem could be a syntax error, a bug due to implicit type conversion, a leaking variable or something else.

Official Documentation: http://jshint.com/docs/

It support Plugins for text editors and IDEs like sublime, vim Emacs.

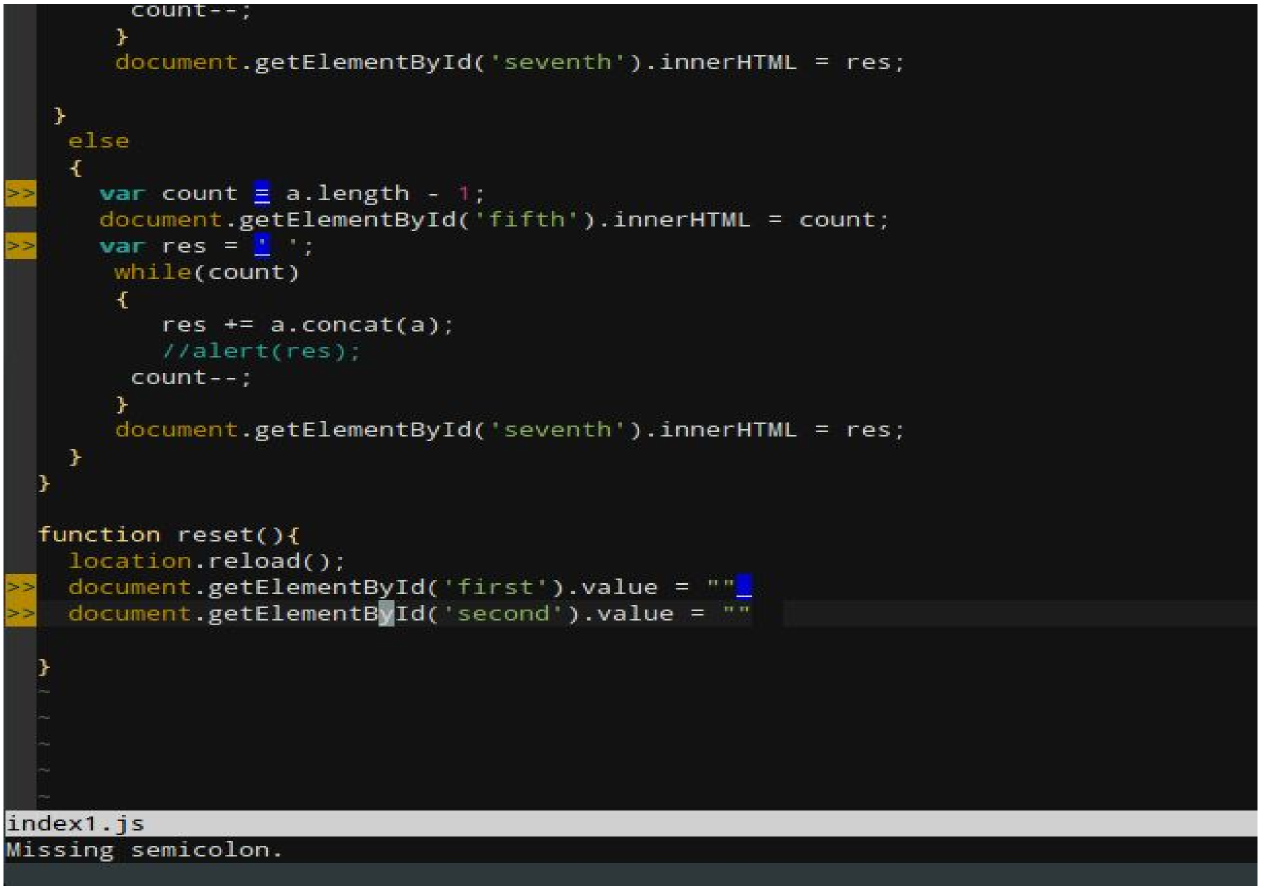
Well written documentation to install node.js before installing jshint:
https://www.digitalocean.com/community/tutorials/how-to-install-node-js-on-an-ubuntu-14-04-server

Steps for installation: http://jshint.com/install/

FYI, You need to set proxy to install it through npm as mentioned below. It won’t allow plugins to install automatically behind the proxy.

```
npm config set proxy = "http://proxy.iiit.ac.in:8080"
npm config set https-proxy="http://proxy.iiit.ac.in:8080”
```

Screenshot of run time error in vim while saving the binary.js file with some syntax errors.



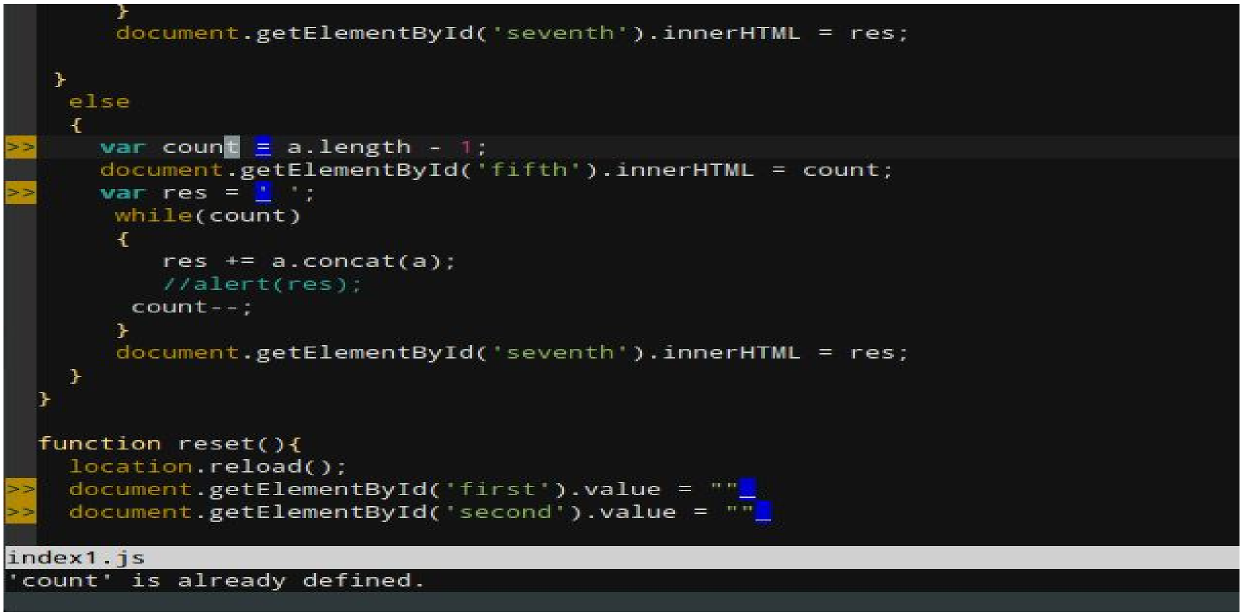
```
        count--;
      }
      document.getElementById('seventh').innerHTML = res;

   }
    else
    {
>>     var count = a.length - 1;
       document.getElementById('fifth').innerHTML = count;
>>     var res = ' ';
        while(count)
        {
           res += a.concat(a);
           //alert(res);
         count--;
        }
        document.getElementById('seventh').innerHTML = res;
    }
  }

  function reset(){
    location.reload();
>>  document.getElementById('first').value = ""
>>  document.getElementById('second').value = ""

  }
  ~
  ~
  ~
  ~
  ~
index1.js
Missing semicolon.
```

Count is defined globally.



```
        }
        document.getElementById('seventh').innerHTML = res;

   }
    else
    {
>>     var count = a.length - 1;
       document.getElementById('fifth').innerHTML = count;
>>     var res = ' ';
        while(count)
        {
           res += a.concat(a);
           //alert(res);
         count--;
        }
        document.getElementById('seventh').innerHTML = res;
    }
  }

  function reset(){
    location.reload();
>>  document.getElementById('first').value = ""
>>  document.getElementById('second').value = ""
index1.js
'count' is already defined.
```