

# Code Formatting and Syntax Review Tool for JavaScript Code

Samprity Kashyap  
Electrical and Communication  
Engineering Department, UBC  
samprity.kashyap@alumni.ubc.ca

Harshdeep Brar  
Electrical and Communication  
Engineering Department, UBC  
brar32@ece.ubc.ca

Arun Marria  
Electrical and Communication  
Engineering Department, UBC  
arun.marria@ece.ubc.ca

## ABSTRACT

In the paper, we describe the implementation and the benefit of a code formatting and syntax review tool for JavaScript. The tool can be used to detect syntax errors and format the code as per the team's coding conventions. We plan to integrate the tool as a google chrome extension.

## 1. PROBLEM STATEMENT

Detection of syntax errors is a painstaking task while writing JavaScript code. Further, programmers generally find implementation of a team's coding conventions as a time consuming task

## 2. MOTIVATION

A tool to detect syntax errors in JavaScript can make the code writing and debugging task easier for a programmer. Following the coding conventions can enhance the readability of the code and make bug detection easier. Further, it can also help in version control by making it easier to visualize the changes made in the code. A tool that detects absence of implementation of coding conventions and implements them automatically and/or reminds the programmer to implement them can enhance programmer's productivity.

## 3. OBJECTIVE

We plan to implement a static analysis tool that detects syntax errors in JavaScript such as:

- Missing semicolons, commas, braces, etc.
- Switch construct with case statements without a break statement
- Redundant statements
- Number of parameters in a method

In terms of implementing code conventions, we plan to include features such as:

- Maximum length of line
- Indentation
- Different text color for methods and variables

## 4. TIMELINE

January 31 – Complete literature survey

February 15 – Syntax error detection code implementation

March 1 – Coding convention code implementation

March 10 – Testing and user feedback

March 21 – Complete conference paper

## 5. RESOURCES

We are planning to use JavaScript, HTML, CSS to implement the tool as a google chrome extension or a standalone application

## 6. REFERENCES

- [1] <http://jshint.com>
- [2] <https://developers.google.com/closure/compiler/>
- [3] <http://surelogic.com/jsure.html>

The heading of a section should be in Times New Roman 12-point bold in all-capitals flush left with an additional 6-points of white space above the section head. Sections and subsequent subsections should be numbered and flush left. For a section head and a subsection head together (such as Section 3 and subsection 3.1), use no additional space above the subsection head.

## 6.1 Subsections

The heading of subsections should be in Times New Roman 12-point bold with only the initial letters capitalized. (Note: For subsections and subsubsections, a word like *the* or *a* is not capitalized unless it is the first word of the header.)

### 6.1.1 Subsubsections

The heading for subsubsections should be in Times New Roman 11-point italic with initial letters capitalized and 6-points of white space above the subsubsection head.

#### 6.1.1.1 Subsubsections

The heading for subsubsections should be in Times New Roman 11-point italic with initial letters capitalized.

#### 6.1.1.2 Subsubsections

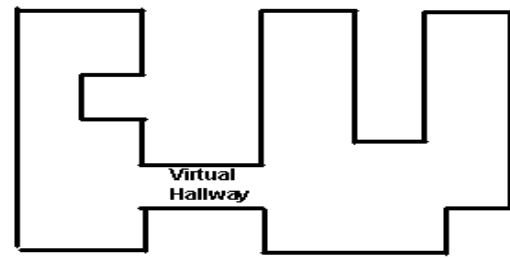
The heading for subsubsections should be in Times New Roman 11-point italic with initial letters capitalized.

## 7. ACKNOWLEDGMENTS

Our thanks to ACM SIGCHI for allowing us to modify templates they had developed.

## 8. REFERENCES

- [4] Bowman, M., Debray, S. K., and Peterson, L. L. 1993. Reasoning about naming systems. *ACM Trans. Program. Lang. Syst.* 15, 5 (Nov. 1993), 795-825. DOI=<http://doi.acm.org/10.1145/161468.16147>.
- [5] Ding, W. and Marchionini, G. 1997. *A Study on Video Browsing Strategies*. Technical Report. University of Maryland at College Park.
- [6] Fröhlich, B. and Plate, J. 2000. The cubic mouse: a new device for three-dimensional input. In *Proceedings of the*



**Figure 1. Insert caption to place caption below figure.**

*SIGCHI Conference on Human Factors in Computing Systems* (The Hague, The Netherlands, April 01 - 06, 2000). CHI '00. ACM, New York, NY, 526-531. DOI=<http://doi.acm.org/10.1145/332040.332491>.

- [7] Tavel, P. 2007. *Modeling and Simulation Design*. AK Peters Ltd., Natick, MA.
- [8] Sannella, M. J. 1994. *Constraint Satisfaction and Debugging for Interactive User Interfaces*. Doctoral Thesis. UMI Order Number: UMI Order No. GAX95-09398., University of Washington.
- [9] Forman, G. 2003. An extensive empirical study of feature selection metrics for text classification. *J. Mach. Learn. Res.* 3 (Mar. 2003), 1289-1305.
- [10] Brown, L. D., Hua, H., and Gao, C. 2003. A widget framework for augmented interaction in SCAPE. In *Proceedings of the 16th Annual ACM Symposium on User Interface Software and Technology* (Vancouver, Canada, November 02 - 05, 2003). UIST '03. ACM, New York, NY, 1-10. DOI=<http://doi.acm.org/10.1145/964696.964697>.
- [11] Yu, Y. T. and Lau, M. F. 2006. A comparison of MC/DC, MUMCUT and several other coverage criteria for logical decisions. *J. Syst. Softw.* 79, 5 (May. 2006), 577-590. DOI=<http://dx.doi.org/10.1016/j.jss.2005.05.030>.
- [12] Spector, A. Z. 1989. Achieving application requirements. In *Distributed Systems*, S. Mullender, Ed. ACM Press Frontier Series. ACM, New York, NY, 19-33. DOI=<http://doi.acm.org/10.1145/90417.90738>.

**Columns on Last Page Should Be Made As Close As Possible to Equal Length**