Building a Challenge Framework

Due on November 12th, 2015

Khan Academy Interview Project

Nasr Maswood

Choice of Framework/ Library

Framework

I was tempted to use JQuery or AngularJS as dependencies. JQuery is great for fishing out browser errors and the latter (which includes JQLite) is the framework where I have most of my experience. However, in the end seeing as this is a very small API I decided on VanillaJS.

Library

Esprima and Ace

I usually have a hard and fast rule for choosing between very similar libraries: Github Stars. While this may not be the most illuminating metric more stars generally means more community support, more documentation and an overall better implementation. Furthermore Ace had a certain company I am extremely fond of on their *Real World Users* page.

Outline of Implementation

Black List and White List

In order to test for black list/ white list compliance I simply traversed the parsed Javascript keeping track of every node[”type”] in a dictionary. Then I could easily check whether the type was being used in the code.

Rough Structure

To test whether a particular pattern was being used I traversed the tree as usual. However, if a node’s type was equal to the outer parameter of the pattern I was looking for (e.g. For in For → If) I would create a dictionary of all the types of that node’s children. If the inner, or the other parameter, was in that dictionary I set a corresponding Boolean to True.

Automatic Testing

Originally, I wanted to test the code on the desired parameters every single change in the document via an editor.on call. However, this turned out to be rather difficult. The status of the tests would update constantly and flash which was very annoying.Additionally, if one changed the tests, he would have to type in order to see the result of the tests : this is inconvenient. Therefore, I opted for an extremely simple fix and periodically call the tests every 7.5 seconds.

Design Choices

I really hate to see the text flashing every time the result of a test is shown. Therefore I included a ’manual’ setting where one clicks a check box and the periodic testing stops. Then one can test by hitting the ’push’ button. This leads to better performance and is much less annoying.

If I Had Infinite Time

- The css is not optimized for all screen types/sizes, in production this would not fly. However, it is also a very time consuming process...
- The output of the test should really be something like a bootstrap alert or something less intrusive. They keep coming in and out of the DOM and it is distracting.