Lab 2. How does Javascript engine optimize javascript code?

Optimizing javascript code means minimizing repetitive codes which are not necessary in our program including declaration of repetitive functions, dead codes (no more used by our program) and it removes comments, white spaces and commas. So that when the optimized code executed, the program will be faster. The optimization process is done by javascript engine called v8 engine. The v8 engine optimizes the source code in different ways. Some of them could be explained shortly below:

- Compressing the code will minimize the file size and will have positive effect on page loading speed. Compressing the javascript code could be done by removing additional spaces, comments, line breaks and so forth.

- Avoiding a dead code which means the code that is not being executed is one way of optimizing the source code. On the example below, we can see that since the functions returns the one condition, the rest conditions other than the returned one, are dead or they will not be executed. Therefore, after optimization those dead codes will be removed and only the executed code will appear as it shown below.

function test(node) {  
 var parent = node.parentNode;  
 if (0) {  
   alert (“Hello from the other side"); // dead code.  
 } else {  
   alert( "We love Upwork" );  
 }  
 return;  
 alert( 1 );  
}

**After optimization:** function test(){alert("We love Upwork")}

-When the browser finds some Javascript files inside the html, since the javascript file located before the DOM, it will stop creating the DOM and start loading the javascript file and waits until it finishes loading the javascript file. Instead we can add “async” tag to the javascript file so that the javascript file and the DOM will load simultaneously. So that the tag made the DOM or the displayed content inside the body of the html, independent of the java script file which is placed in <script src=”javafile.js” async></script>.

-Replacing variables and functions on the return or console by their real values will optimize the number of declarations and call unnecessary repetitive calling of functions. And this kind of optimization is called variable inlining.

-Rounding trick: usually when we round numbers we use Math.floor() or Math.round() but using the operator “ ~~ “ is better because it is shorter and more efficient than those of two.

-Using shortcuts is more faster and save a lot of time. For example instead of using:

function checkName(name) {  
if (name) {  
return name;  
} else {  
return ‘Name is null’;  
}  
}

We can use: (optimized code)

function checkName(name) {  
return name || ‘Name is null’;  
}

Some resources and references includes:

[11 Tips to Optimize JavaScript And Improve Website Loading and Rendering Speeds | Upwork](https://www.upwork.com/resources/11-tips-to-optimize-javascript-and-improve-website-loading-and-rendering-speeds)

[7 Simple JavaScript Tips for Optimizing Your Code | by Amy J. Andrews | JavaScript in Plain English](https://javascript.plainenglish.io/7-simple-javascript-tips-for-optimizing-your-code-ba83d4d7da4e)

<https://mathiasbynens.be/notes/shapes-ics>

<https://www.digitalocean.com/community/tutorials/js-v8-engine>