Lesson 73 JS strict mode

Strict mode enables certain JS code to behave differently semantically in order to make it more **secure**. As such, browsers that do not support the use of strict mode execute code written in strict mode differently than the browsers that do.

Use of strict mode:

strict mode makes several changes to the way JS code generally behaves. It makes certain mistakes, which are otherwise ignored, and throws errors instead. It also performs optimization by fixing mistakes to make code run faster and prevents the use of certain syntax. The details of these changes are provided below.

- It is not allowed to create global variables by mistake. All variables and objects must be declared before use.
- It is not permissible to delete a variable or function.
- Assigning values to variables that would be silently ignored in normal circumstances turns into an error. For instance, assigning a value to NaN would throw an error.
- It ensures that all parameters of a function are unique.

To write JS code in strict mode, use the code below :

```
'use strict';
```

The effect of using strict mode for some of the cases above is presented in the code below :

```
'use strict';

num1 = 23; // will throw an error since num1 is not declared

var eval = 10 + 3; /* Error since eval can not be used as a variable name.*/

var undefined = 100; /* raises an error since assigning a value to a non-writable variable*/

console.log(this); // produces 'undefined'.

var octal_num = 067; /* gives invalid number error, cannot use octal literals preceding with a 0 */

delete num1; // cannot delete a local variable in strict mode
```