Good JavaScript Practices

Avoid global variables

Global variables can be overwritten by other scripts, they can also be significantly slower to access. The best practice is to us local variables and closures.

Avoid using eval()

Using eval() represents some security concerns, and can also have an inpact on performance. In most cases the use of eval() is avoidable

Declare variables at the top of code

Declaring variables at the top of each function or script gives a code that is cleaner, and easier to manage.

Treat primitive values as such, not as objects

Declarations like ***var y = new String("some string")*** will slow down the code, also as JS objects cannot be compared, this can lead to further problems.

Use === instead of == wherever possible

The operator ***===*** does not convert the operands to matching types before comparsion, therefore is faser than using ***==*** . The same goes for ***!==*** instead of ***!=***

Keep loops fast by reducing the code inside

Avoid placing assignments and statements inside a loop that could be done before entering the loop. This can have a great effect on the speed of the code, as some operations will only be done once instead of at every iteration. For the same reason ***for(var a=0;a<b.length;a++){}*** will perform slower than ***for(var a=b.length-1;a>=0;a--){}***