Performance

Overview

This module includes some methods by which a programmer can improve the performance of their javascript code.

Reduce Activity in Loops

Loops are often used in programming. Each statement in a loop, including the for statement, is executed for each loop iteration.

Statements or assignments that can be placed outside the loop will make the loop run faster.

Example :

```
for ( let i = 0 ; i < arr.length ;  i++ ) {
        console.log( “Hello” ) ;
}
```

Improved :

```
let len = arr.length ;
for ( let i = 0 ; i < len  ;  i++ ) {
        console.log( “Hello” ) ;
}
```

- ❖ **It is better to store the length of the array in a variable rather than getting the length every time.**

Avoid Unnecessary Variables

Don't create new variables if you don't plan to save values because it increases the space complexity of your code.

Example :

```
var z =  x + y ;
console.log( z ) ;
```

Improved:

```
console.log( x + y ) ;
```

Avoid Using with

Avoid using the **with** keyword. It has a negative effect on speed. It also clutters up JavaScript scopes.

The with keyword is not allowed in strict mode.

Avoid using new keyword

Create objects and arrays using brackets rather than using new keyword
new keyword slows down the javascript code.

Example :	var arr = **new** Array( ) ;
Improved:	var arr = [ ] ;