JAVASCRIPT

ES6 Features-1

Sreekanth M. E.

Freelance Trainer & Consultant

http://www.SreekanthME.com

JS HISTORY

ECMAScript 5 (ES5) or ECMAScript 2009

ECMAScript 6 (ES6) or ECMAScript 2015

ECMAScript 7 (ES7) or ECMAScript 2017

ES7 is the latest version of JS.

Many of the new ES6 / ES7 features are syntactic sugar.

In computer science, syntactic sugar is syntax within a programming language that is designed to make things easier (for humans) to read or to express.

A construct in a language is called syntactic sugar if it can be removed from the language without any effect on what the language can do

Almost all browsers, in use today, support ES5.

But ES6 support is mixed.

Tools like Babel can be used to write JS code in ES6 and still ensure it runs in all browsers.

LET & CONST

The let statement declares a variable, optionally initializing it to a value.

The const statement declares a constant, mandatorily initialized to a value.

The value of a constant cannot change through re-assignment, and it cannot be redeclared.

Let & const are block-scoped. They only exists within the current block.

var is function-scoped.

When an object is made a constant, assigning another object will cause an error as expected.

But, it is still possible to modify its properties.

Use Object.freeze() method to prevent modification.

With let (unlike var), variables must be both declared and initialized before they can be accessed.

Otherwise an error occurs.

TEMPLATE LITERALS

Template Literals provide syntactic sugar for constructing strings.

Template literals are enclosed by the back-tick (` `) (grave accent) character instead of double or single quotes.

Template literals can contain place holders which are indicated by the Dollar sign and curly braces (${expression}).

Template literals support Multi-line strings.

DEFAULT PARAMETERS

Default function parameters allow parameters to be initialized with default values if no value or undefined is passed

```
function foo(x=0, y=0) {
    ...
}
```

ENHANCED OBJECT LITERALS

PROPERTY VALUE SHORTHAND

The property value shorthand allows writing only the property name(key), if the property name matches an existing variable name .

```
let obj = { x, y };
```

METHOD DEFINITION SHORTHAND

The method definition shorthand allows writing only the method name(key) without the function keyword.

ES6:

```
obj = {
    foo (a, b) {
        …
    },
}
```

ES5:

```
obj = {
    foo: function (a, b) {
        …
    },
};
```

COMPUTED PROPERTY KEYS

ES6 allows property keys of object literals to use expressions, making them computed property keys.

END OF CHAPTER

APPENDIX