**ECMAScript 2015 Goodies – Part 2 – const**

This is the second article in ECMAScript 2015 Goodies. In the first article, we have covered new keyword in the language – let, which allows us to define a true block level variable in the program. In this short article, we will cover another new keyword – const. As the name suggests, const is used to define a read-only named constant.

Following code snippets demonstrates how constants behave in different scenarios

A constant must contain an initialization statement before used

const pi; // SyntaxError

console.log(pi);

The value of a constant cannot be changed through re-assignment and a constant cannot be re-declared.

const pi = 3.14;

pi = 3.1415; //Syntax Error

A constant cannot share its name with a function or a variable in the same scope.

const pi = 3.14;

// var pi = "hi"; // Error

Similar to let, constants are block scoped and they does not hoist in a declared scope.

(function foo() {

{ // Explicit Block

const company = "Microsoft";

console.log(company);

}

console.log(company); // Error

console.log(pi); // Error

const pi = 3.14;

console.log(pi);

}());

And finally, it works on object as well!

const env = {

"language": "JavaScript",

"tool": "SublimeText"

};

console.log(env.language);

Useful, isn’t it?