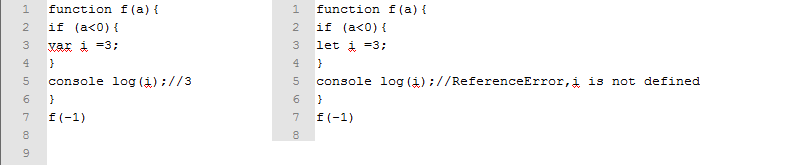
1. Области видимости переменных.

ES5 «Variables are initialised to undefined when created. A variable with an Initialiser is assigned the value of its AssignmentExpression when the VariableStatement is executed, not when the variable is created»

ES6 «A let and const declarations define variables are scoped to the running execution context’s LexicalEnvironment. The variables are created when their containing Lexical Environment is instantiated but may not be accessed in any way until the the variable’s LexicalBinding is evaluated. A variable definded by a LexicalBinding with an Initializer is assigned the value of its Initializer’s AssignmentExpression when the LexicalBinding is evaluated, not when the variable is created. If a LexicalBinding in a let declaration does not have an Initializer the variable is assigned the value undefinded when the LexicalBinding is evaluated»

Все переменные, объявленные с ключевым словом var будут видны в любом месте функции. Для объявления переменных с блочной областью видимости в новую версию было добавлено ключевое слово let.

На мой взгляд,это сделано для ограничения области видимости переменных, и, соответственно, повышения безопасности кода.



2.Функции (значение параметров по умолчанию)

ES5: « Function Definition »

ES6: « Functions and Generators :NOTE The LexicallyDeclaredNames of a FunctionBody does not include identifiers bound using var or function declarations. Simple parameter lists bind identifiers as VarDeclaredNames. Parameter lists that contain destructuring patterns, default value initialisers, or a rest parameter bind identifiers as LexicallyDeclaredNames. Multiple occurrences of the same Identifier in a FormalParamterList is only allowed for non-strict functions with simple parameter lists.»

Комментарий:

В ES6 добавилась возможность в функциях объявлять у параметров значения по умолчанию, то есть если функции не передается параметр, она присваивает его по умолчанию:

function setLevel(newLevel = 0) {...}

setLevel(); // newLevel = 0

setLevel(5); // newLevel = 5

setLevel(undefined); // newLevel = 0

# 3. Функции(TimeClip (time)).

ES5: « Return an implementation-dependent choice of either [ToInteger](http://www.ecma-international.org/ecma-262/5.1/#sec-9.4)(time) or [ToInteger](http://www.ecma-international.org/ecma-262/5.1/#sec-9.4)(time) + (+0). (Adding a positive zero converts −0 to +0.)»

ES6: « Dates never expose -0 as a time value»

Комментарий:

В версии 6 для большей понятности кода была устранена возможность, задавать время в отрицательном формате(-0).

4. Примитивные типы.

ES5: « Отсутствует»

ES6: « Added ToInt8 and similar abstract operation»

Комментарий:

Добавлено для расширения возможностей языка.