Колобов Александр

ДКИ-301

**Форум**

1. **Блочная область видимости**

**ES5**: « 12.2Variable Statement

A variable statement declares variables that are created as defined in 10.5. 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**: « 12.2 Declarations and the Variable Statement

12.2.1 Let and Const Declarations

NOTE A let and const declarations define variables that 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 variable’s LexicalBinding is evaluated. A variable defined by a LexicalBinding with an Initialiser is assigned the value of its Initialiser’s AssignmentExpression when the LexicalBinding is evaluated, not when the variable is created. If a LexicalBinding in a let declaration does not have an an Initialiser the variable is assigned thevalue undefined when the LexicalBinding is evaluated. »

**Комментарий**:

В ES5 присутствует функциональная область видимости, т.е. переменные, объявленные c помощью ключевого слова var, будут видны в любом месте функции (даже если они объявлены внутри блока):

function f(a) {

if (a < 0) {

var i = 3;

}

console.log(i); // 3

}

f(-1)

В ES6 же появилось ключевое слово let, которое позволяет объявлять переменные с блочной областью видимости, т.е. переменные видны только в блоке, где объявлены, это позволяет сделать код более безопасным:

function f(a) {

if (a < 0) {

let i = 3;

}

console.log(i); // ReferenceError: i is not defined

}

f(-1)

1. **Циклы**

**ES5:** «12.6.4The for-in Statement

The production *IterationStatement* **:** **for** **(***LeftHandSideExpression* **in***Expression***)** *Statement* »

**ES6:** « 12.6.4 The for-in and for-of Statements

Static Semantics

Static Semantics: Early Errors

*IterationStatement* :

for (*LeftHandSideExpression* in *Expression* ) *Statement*

for (*LeftHandSideExpression* of *AssignmentExpression* ) *Statement* »

**Комментарий:**

Был добавлен новый цикл for-of.

Цикл for-in итерирует по всем полям объекта (включая наследованные):

let arr = [ "blue", "green" ];

arr.notAnIndex = 123;

Array.prototype.protoProp = 456;

for(var x in arr) {

console.log(x); // Напечатает blue, green, notAnIndex, protoProp

}

В ES6 цикл for-of решает данную проблему:

for(var x of arr) {

console.log(x); // Напечатает только blue, green

}

1. **Функции (значение параметров по умолчанию)**

**ES5:** « 13Function Definition »

**ES6:** « 13 Functions and Generators

13.1 Function Definitions

...

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 в функциях добавилась возможность 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