1. **Примитивные типы**

В версии 5.1:

Предыдущие издания ECMAScript (относительно этой) включали дополнительный синтаксис и семантику для указания восьмеричных литералов и восьмеричных управляющих последовательностей. Они были удалены из этого издания ECMAScript. Эта пристройка представляет однообразный синтаксис и семантику для восьмеричных литералов и восьмеричных управляющих последовательностей для совместимости с некоторыми старыми ECMAScript программ.

В версии 6:

Восьмеричные управляющие последовательности не допускаются в строковых литералах шаблонов.

Комментарий:

Я считаю, что это необходимо было сделано, для лучшей читаемости кода ведь литералы должны быть понятны программисту, и один и тот же литерал, я считаю, в разных участках кода должен быть представлен в одной и той же кодировке.

1. **Нотация**

В версии 5.1:

JSON грамматика используется для перевода строки, которая описывает набор объектов ECMAScript в реальных объектах.

В версии 6:

Удалена грамматика JSON и заменена ссылками на ECMA-404.

Комментарий:

Я считаю, что замена была осуществлена, для упрощения структуры языка.

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