Стандарт EcmaScript 5:

<http://www.ecma-international.org/ecma-262/5.1/>

Изменения в EcmaScript 6:

<http://wiki.ecmascript.org/doku.php?id=harmony:specification_drafts>

Ваша задача:

Посмотреть как изменится стандарт в отношении определенных частей языка, а именно:

1. примитивных типов
2. примитивных объектов
3. функций
4. массивов
5. циклов
6. ООП

Правила ответа:

1. Можно выбирать абсолютно любой пункт (или несколько).
2. Следите за тем, что уже ответили ваши коллеги (повторы не засчитываются).
3. Цитируете ES5 (как было), затем ES6 (как стало), снабжаете комментариями – что изменилось, зачем нужно, какие из этого следствия. Стараетесь «встроить» изменения в общую концепцию языка.

Пример оформления:

1. Примитивные типы

ES5: «» - цитата из ES5

ES6: «» - цитата из ES6

Ваш комментарий.

#### 3. Функции

#### Именованные параметры функций

В функциях также появилась возможность указывать именованные параметры:

function foo({ from, to = 10 }) {

...

}

foo({ from: 1, to: 5 });

foo({ to: 5, from: 1 });

foo({ from: 1 });

Именованные параметры можно комбинировать с обычным (позиционными параметрами):

function foo(positional, { named1, named2 }) {

...

}

foo(123, { named1: 'abc', named2: 'def' })

foo(123, { named2: 'def', named1: 'abc' })

6.ООП

В ECMAScript 6 появятся классы:

class Person {

constructor(name) {

this.name = name;

}

describe() {

return "Person called " + this.name; }

}

Теперь можно использовать эти классы:

let jane = new Employee("Jane", "CTO");

jane instanceof Person; // true

Также появились статические методы

class Point {

constructor(x, y) {

this.x = x; this.y = y;

}

static zero() {

return new Point(0, 0);

}

}

В ECMAScript 5 как таковых классов не было, схема классов строилась на основе прототипов.

function Person(name) {

this.name = name;

}

Person.prototype.describe = function () {

return "Person called " + this.name;

};

5.Циклы

Как вы знаете, цикл for-in в JavaScript итерирует по всем полям объекта (включая наследованных). Итерирование по массивам возможно, но не желательно:

let arr = [ "привет", "пока" ];

arr.notAnIndex = 123;

Array.prototype.protoProp = 456;

for(var x in arr) {

console.log(x); // Напечатает привет, пока, notAnIndex, protoProp }

В ECMAScript 6 появится цикл for-of, выводятся только элементы массива:

for(var x of arr) {

console.log(x); // Напечатает только привет, пока

}

**4.Массивы**

В ECMAScript 6 появятся стрелочные функции:

let rmp = [ 10, 20, 30 ].map(x => x \* 3);  
Код выше эквивалентен этому:

let rmp = [ 10, 20, 30 ].map(function (x) { return x \* 3 });

Отличие от стандартных функций, в том, что this привязывается к вышестоящему контексту, также нельзя обращаться к переменной arguments и их нельзя использовать как конструкторы.