

Java Script

СИМВОЛЫ

- Unicode
 - Чувствительность к регистру
 - Любое количество табуляций и пробелов
 - Комментарии:
 - // и до конца строки
 - Текст между /* */
-

Идентификаторы

- ❑ Первый символ: буква / _ / \$
 - ❑ Остальные: буква / цифра / _ / \$
 - ✓ Применения \$ лучше избегать
 - ✓ Идентификаторы не должны совпадать с ключевыми словами (break, case, for, null, true и т.д.)
-

Типы данных

□ Числа

целые и вещественные представляются одинаково,
аналог в C double

□ Строки

"str", 'str'

□ Логический тип

константы true, false

□ Составной (object)

частный случай – массивы

□ Два тривиальных типа: null, undefined

Переменные

JS нетипизированный язык, т.е. в переменной может быть что угодно

Можно не объявлять, но тогда переменная будет обязательно глобальной

Объявление с помощью var:

```
var i;      var j, k;      var n=0, l=1;  
for (var i=0; i<10; i++) document.write(i, "<br>");
```

Преобразование типов

Преобразование типов происходит автоматически

```
a="10";
```

```
b=a-0; //итог число
```

```
c=a+0; //итог строка
```

```
0, null, undefined => false
```

```
1, "hello" => true
```

```
false => 0
```

```
true => 1
```

Массивы

- ❑ Массивы содержат в себе что угодно, в том числе и ссылки на другие массивы.
 - ❑ Элементы массива не обязательно имеют один и тот же тип
 - ❑ Массив имеет любое количество элементов.
 - ❑ Массивы могут быть разреженными
-

Определение массива

```
var empty = []; //пустой массив
var primes = [2,3,4,5,6]; //массив с пятью числовыми элементами
var misc=[1.1, true, "a"]; //3 элемента разных типов
var count = [1, ,3]; //массив из 3х элементов, средний не определен
```

С ПОМОЩЬЮ КОНСТРУКТОРА Array()

```
var a = new Array();
var a=new Array(5, 4, 2, "testing", true)
var a=new Array(10);
```

Обход непрерывного массива производим как обычно, разреженный надо проверять:

```
for(var i=0; i < a.length; i++)
    if (a[i] != undefined) делаем что-то...
```

HTML5, canvas

```
<html>
<body>
<canvas id='draw' width='400' height='300' style='border:1px solid'></canvas>
<script type='text/javascript'>

var canvas=document.getElementById("draw")
var x=canvas.getContext("2d");

x.fillRect(50,100,15,15);

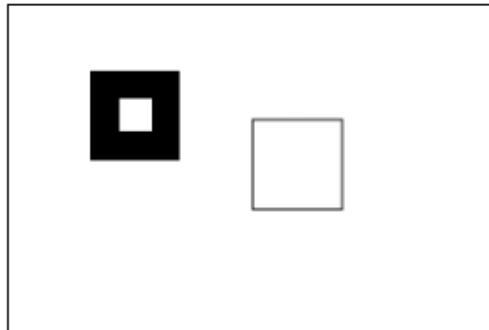
</script>
</body>
</html>
```

HTML5, рисование

Метод	Описание
<code>fillRect(x,y,ширина,высота)</code>	Рисует закрашенный прямоугольник.
<code>strokeRect(x,y,ширина,высота)</code>	Рисует не закрашенный прямоугольник.
<code>clearRect(x,y,ширина,высота)</code>	Очищает указанную зону делая ее полностью прозрачной.
<code>beginPath()</code>	Начало сложной фигуры
<code>closePath()</code>	Автоматически завершает фигуру, соединяя конечную точку с начальной
<code>stroke()</code>	Фигура не закрашена
<code>fill()</code>	Фигура закрашена
<code>moveTo(x,y)</code>	Устанавливает координаты точки, из которой начнется рисование следующего объекта.
<code>lineTo(x,y)</code>	Рисует прямую линию
<code>rect(x, y, ширина, высота)</code>	Рисует прямоугольник
<code>arc(x,y,R,нач_угол,конеч_угол)</code>	Рисует круг (Угол в радианах)

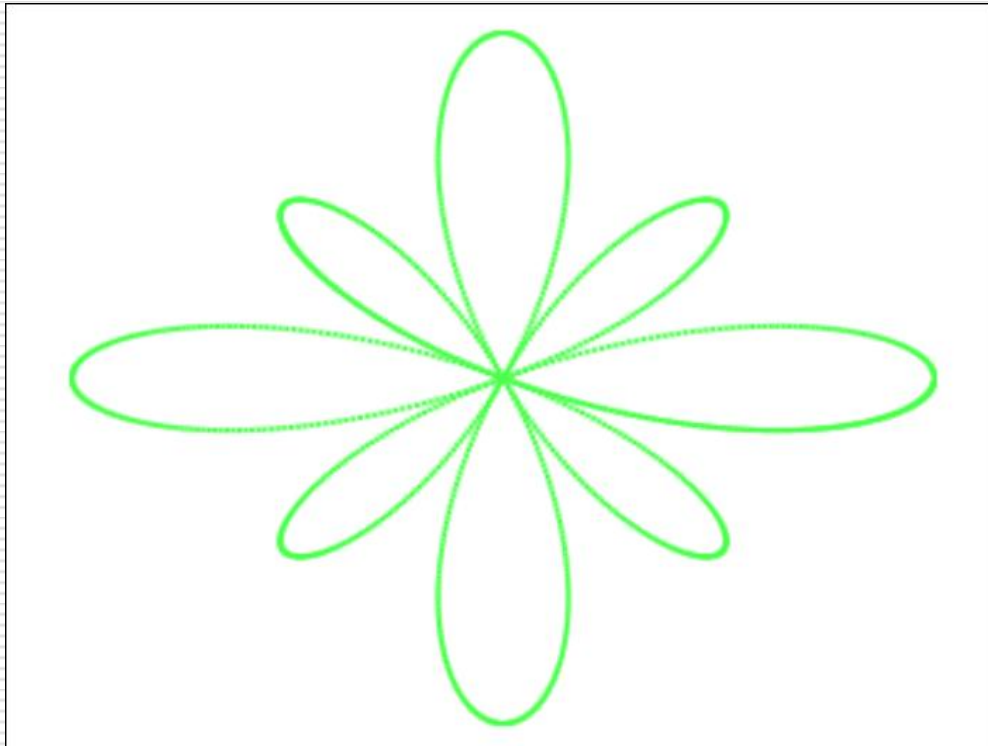
HTML5, прямоугольники

```
var canvas=document.getElementById("draw") // находим нужный холст  
var x=canvas.getContext("2d") // обращаемся ко встроенному объекту, который  
    содержит различные методы для рисования (первые два шага являются  
    стандартными для рисования любого объекта в canvas)  
x.fillRect(50,40,55,55) // рисуем закрашенный прямоугольник  
x.strokeRect(150,70,55,55) // рисуем не закрашенный прямоугольник  
x.clearRect(68,57,20,20) // очищаем зону в закрашенном прямоугольнике
```



$$R = 0.5 + 3 * \text{Math.cos}(4 * t)$$

$$R = 0.5 + 3 * \text{Math.cos}(4 * t)$$



Другие функции

□ $R = \mathbf{k} * t * \sin(\mathbf{n} * t)$

□ $R = \mathbf{1} + \mathbf{k} * \cos(\mathbf{n} * t)$

□ $R = t^2 + \mathbf{2} * t + \mathbf{3}$
