Тема 8. Техника Canvas

Содержание:

1. Техника **Canvas**
2. Свойства объектов
3. Окружность
4. Задание.

Тег <canvas> определяется как «растровый холст, который может быть использован для отображения диаграмм, игровой графики или изображений на лету».

Холст это прямоугольная область на вашей странице, где с помощью JavaScript можно рисовать что пожелаете.

Техника **Canvas** позволяет создавать:

1. Графику для игры
2. Интерактивные визуализации
3. Графические эффекты для изображений
4. Работать с потоковым видео

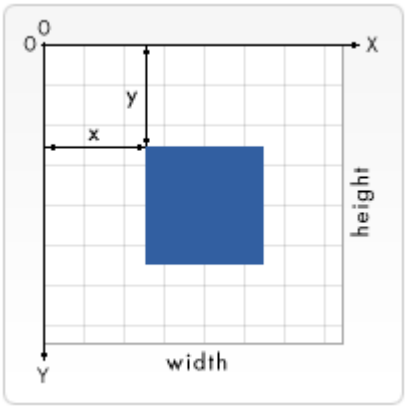
<canvas id="canv" width="300" height="225" style="border: 1px solid black"> </canvas> - если эти атрибуты отсутствуют, то ширина по умолчанию будет равна 300 пикселей, а высота 150 пикселей.

Перед тем как начать рисование, необходимо получить экземпляр объекта DOM для данной канвы. Это выполняется вызовом единственного метода getContext объекта HTMLCanvasElement:

var canvas1 = document.getElementById("canv"); *- обращение к объекту Canvas*
var ctx = canvas1.getContext('2d'); *- использование контекста 2d*

Canvas может иметь такие же атрибуты стиля как и у <img> (margin, border, background и т.д.). Если стиль не задан, то первоначально он полностью прозрачный.

```
<canvas id="canvas" style="border: 2px solid black" width="200" height="200">
</canvas>
```



Методы:

- **strokeRect(x, y, width, height)** - рисует контур прямоугольника с текущим стилем линии (контур).
- **fillRect(x, y, width, height)** - рисует закрашенный прямоугольник, заполненный текущим стилем заливки.
- **clearRect(x, y, width, height)** - удаляет пиксели в указанном прямоугольнике.

Свойства:

- **strokeStyle** – цвет обводки.
- **fillStyle** – цвет заливки.
- **lineWidth** – толщина линии обводки.

ctx.fillStyle = '#face8d';

var gradient = ctx.createLinearGradient(0, 0, 300, 150);
gradient.addColorStop(0, 'green');
gradient.addColorStop(0.5, 'rgba(0, 255, 255, 0.5) ');
gradient.addColorStop(1, 'black');

ctx.fillStyle = gradient;
ctx.fillRect(0, 0, 300, 150);

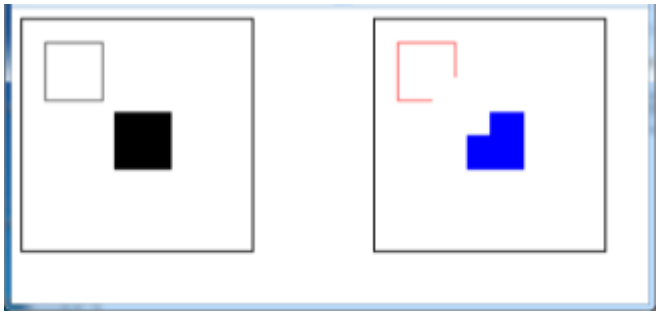
Нельзя присваивать значение свойства strokeStyle свойству fillStyle и наоборот.

Пример 1.

```
<html>
 <head>
   <script >
    function createImage()
    {
    var canvas = document.getElementById("canvas");
    var ctx = canvas.getContext("2d");
        ctx.strokeRect(20,20,50,50);
    ctx.fillRect(80, 80, 50, 50);
//      ctx.clearRect(50,50,50,50);
```

```
  var canvas2 = document.getElementById("canvas2");
  var ctx2 = canvas2.getContext("2d");
  ctx2.strokeStyle = "red";
  ctx2.strokeRect(20,20,50,50);
  ctx2.fillStyle = "blue";
  ctx2.fillRect(80, 80, 50, 50);
  ctx2.clearRect(50,50,50,50);
   }
  </script>

 </head>
 <body onload="createImage();">
   <canvas id="canvas" style="border: 2px solid black" width="200"
height="200"> </canvas>
  <canvas id="canvas2" style="margin-left: 100px; border: 2px solid black"
width="200" height="200"> </canvas>
 </body>
</html>
```



Контуры

Контур – последовательность линий.

`beginPath()` - создает новый контур. После создания используется в дальнейшем командами рисования при построении контуров.

`closePath()` - закрывает контур.
`stroke()` - рисует фигуру с внешней обводкой.

`fill()` - рисует фигуру с заливкой внутренней области.

`moveTo(x, y)` - перемещает перо в точку с координатами x и y.
`lineTo(x, y)` - рисует линию с текущей позиции до позиции, определенной x и y.

Последовательность создания фигур с использованием контура:

1. Создать контур: `beginPath()`;
2. Установить исходную точку: `moveTo(x, y);`
3. Нарисовать контур из линий: `lineTo(x, y)`;
4. Закрыть контур: `closePath()`;
5. Обвести или залить контур: `stroke()` или `fill()`;

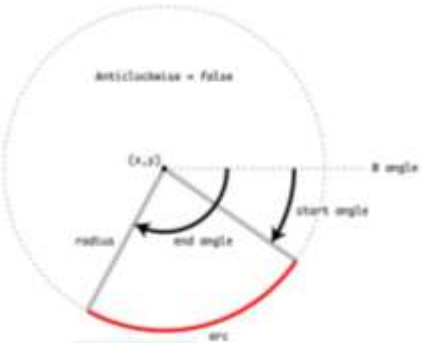
Пример2 (lab20-03.html):

```
<script>
  var ctx = document.querySelector("canvas").getContext("2d");
  ctx.beginPath();
  for (var y = 10; y < 400; y += 10) {
    ctx.moveTo(10, y);
    ctx.lineTo(90, y);
  }
  ctx.stroke();
</script>
```

Рисование дуг

Для рисования дуг используется метод arc:

```
arc(x, y, r, sA, eA, a);
```



Методу передаются

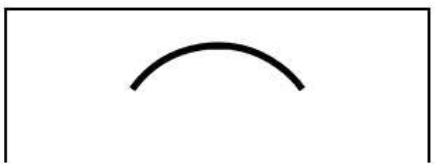
- x и y - центр окружности;
- r - радиус окружности;
- sA - начальный угол;
- eA - конечный угол;
- a - направление.

Рассмотрим на следующем примере:

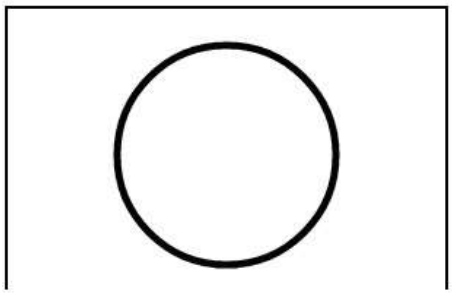
```
function createImage()
  {
  var canvas = document.getElementById("canvas");
  var ctx = canvas.getContext("2d");
  ctx.beginPath();
  ctx.arc(100, 100, 75, 2*Math.PI, 1.8 * Math.PI, false);
  ctx.lineWidth = 5;
  ctx.strokeStyle = "black";
  ctx.stroke();

   }
```

В результате получим следующую дугу:



Частным случаем дуги является окружность, изменив одну строку предыдущего примера, а именно значения начального и конечного угла для метода arc (на ноль и $2 * \pi$ соответственно), получим следующий результат:



Вывод текста на canvas

```
ctx.font = '30px Tahoma';
ctx.textBaseline = 'hanging';
ctx.fillText("Привет!", 0, 10);
```

Интернет- ресурсы:

http://htmlbook.ru/html5/canvas - Давайте порисуем

http://ruseller.com/lessons.php?rub=1&id=1010 – Введение в Canvas

http://ruseller.com/lessons.php?rub=1&id=1122 - Введение в Canvas: продвинутые техники рисования

https://studyjavascript.blogspot.com/2017/08/js30-8.html - Изучаем JavaScript

- https://developer.mozilla.org/en/HTML/Canvas
- http://www.w3schools.com/html5/html5_canvas.asp
- http://uroki-html.ru/html5/html5_canvas.php

2. Контекст отрисовки
   - http://www.w3.org/TR/2dcontext/
3. Работа с цветами в canvas:

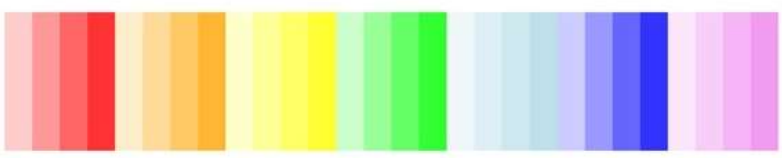
- http://true-coder.ru/javascript/delaem-mir-canvas-yarche-primenenie-stilej-cvetov-i-tenej.html
- http://saitcreate.ru/primenenie-stilej-i-cvetov/
- https://professorweb.ru/my/html/html5/level1/html5_index.php - элемент Canvas (прозрачность и рисование)

Практическая работа №8. Работа с Canvas.

Задание

Разместить элемент <canvas> на странице. Создать файл стилей для <canvas>. При помощи *JavaScript* – функции реализовать генерацию следующих изображений на странице:

1.



2. Написать скрипт для размещения 30 шариков размером от 20 до 40px на холсте во весь экран:

```
var width = canvas.width = window.innerWidth;
var height = canvas.height = window.innerHeight;
```