

Рассмотрение возможности веб-рисования на HTML5 с помощью canvas

Кочитов Михаил Евгеньевич

Приамурский государственный университет им. Шолом-Алейхема

студент

Аннотация

В данной статье рассматривается возможность веб-рисования на HTML5 с помощью canvas. Для веб-рисования будут использованы язык разметки HTML5, который с пятой версии начал поддерживать canvas и язык программирования JavaScript. В статье будут продемонстрированы собственные рисунки, нарисованные с помощью canvas в браузере.

Ключевые слова: веб-рисование, HTML5, JavaScript, рисунок, рисование, canvas

Consider the possibility of web drawing on HTML5 using the canvas

Kochitov Mikhail Evgenevich

Sholom-Aleichem Priamursky State University

student

Abstract

This article discusses the possibility of web-drawing on HTML5 using canvas. For web drawing, the markup language HTML5 will be used, which since the fifth version has started to support the canvas and the JavaScript programming language. The article will showcase your own drawings drawn using the canvas in the browser.

Keywords: web-drawing, HTML5, JavaScript, drawing, drawing, canvas

В настоящее время графические редакторы обрели большую популярность. Они обрели огромное количество инструментов, чтобы создать с нуля свое изображение или рисунок либо переделать оригинальную фотографию в наиболее красивую, придав ей несколько хороших фильтров. Однако с выходом новой версии языка разметки HTML5 появилась возможность создавать графические редакторы в самом браузере, благодаря введенному инструменту под названием canvas.

Целью данной статьи является возможность веб-рисования на HTML5 с помощью инструмента canvas, добавленного с пятой версии HTML. Само веб-рисование будет происходить с помощью языка программирования JavaScript.

В статье М.Н. Дядевой рассматривается использование библиотеки Raphael для создания SVG – графики [1]. В. Дронов в книге рассматривает процесс создания современных веб-сайтов на HTML5, CSS3, также

рассмотрел средства HTML5 для программного рисования [2]. Рассматривая книгу Э. Фримен и Э. Робсон можно увидеть глубокое изучение программирования HTML5, включая подробное 2D-рисование [3]. В статье П.С. Ляпина, Р.М. Мельничук и А.Д. Финогорова рассматриваются графические редакторы для схемотехнического проектирования [4]. Н.Н. Нуромская, И.К. Попова и О.А. Шестакова в статье рассматривают методику работы с графическим редактором GIMP [5].

Перед началом рисования необходимо создать HTML страницу и в ней написать код, представленный на рисунке ниже

```
<html>
  <head>
    <meta charset="utf-8">
    <title>Веб-рисование на HTML5</title>
  </head>
  <body>
    <h2>Веб-рисование на HTML5</h2>
    <canvas id="paint_block" style="border: 1px solid black"></canvas>
  </body>
</html>
```

Рис. 1. HTML код с добавленным тегом canvas

На рисунке 1 изображен HTML код, в котором добавлен тег под названием canvas. Благодаря этому тегу на веб-странице в браузере создается холст (электронный лист бумаги) на котором в дальнейшем и будет происходить веб-рисование. Для canvas обязательно задавать идентификатор или класс, чтобы можно было на нем рисовать, используя язык программирования JavaScript. В нашем случае canvas был присвоен ID – «paint block».

Далее необходимо скриптом JavaScript инициализировать сам canvas, чтобы можно было на нем начать уже что-то рисовать, для этого напишем несколько строк кода

```
<html>
  <head>
    <meta charset="utf-8">
    <title>Веб-рисование на HTML5</title>
  </head>
  <body>
    <h2>Веб-рисование на HTML5</h2>
    <canvas id="paint_block" style="border: 1px solid black"></canvas>

    <script>
      var canv = document.getElementById("paint_block");
      var ctx = canv.getContext("2d");
      canv.width = 500;
      canv.height = 500;
    </script>
  </body>
</html>
```

Рис. 2. Код инициализации canvas на JavaScript

Глядя на рисунок 2 можно увидеть инициализацию canvas скриптом. Для инициализации нужно сначала в переменную `canv` присвоить HTML элемент под ID – «paint_block», а после применения функции `getContext("2d")` сама переменная станет в роли canvas, и теперь через эту переменную можно рисовать что-либо на самом холсте. Параметры `width` и `height` задают изначально ширину и высоту самому холсту.

Теперь нарисуем что-нибудь, для этого напишем некоторый код, представленный на рисунке ниже

```
<script>
var canv = document.getElementById("paint_block");
var ctx = canv.getContext("2d");
canv.width = 500;
canv.height = 500;

ctx.fillStyle = "blue";
ctx.fillRect(0, 0, 500, 500);
ctx.fillStyle = "green";
ctx.fillRect(0, 400, 500, 100);
ctx.fillStyle = "grey";
ctx.fillRect(100, 200, 300, 200);
ctx.fillStyle = "grey";
ctx.fillRect(280, 5, 40, 100);
ctx.fillStyle = "brown";
ctx.fillRect(80, 170, 340, 30);
ctx.fillStyle = "brown";
ctx.fillRect(100, 140, 300, 30);
ctx.fillStyle = "brown";
ctx.fillRect(120, 110, 260, 30);
ctx.fillStyle = "brown";
ctx.fillRect(140, 80, 220, 30);
ctx.fillStyle = "brown";
ctx.fillRect(160, 50, 180, 30);
ctx.fillStyle = "brown";
ctx.fillRect(210, 280, 70, 120);
ctx.fillStyle = "black";
ctx.fillRect(260, 340, 10, 10);
ctx.fillStyle = "white";
ctx.fillRect(120, 280, 60, 60);
ctx.fillStyle = "white";
ctx.fillRect(310, 280, 60, 60);
</script>
```

Рис. 3. Код рисования на холсте заполненными цветами прямоугольниками на JavaScript

На рисунке 3, представлен код, в котором видна функции `fillStyle` и `fillRect`, теперь рассмотрим эти функции подробнее. Функция `fillStyle` устанавливает цвет кисти заливки, значением этой функции указывается ключевое слово зарегистрированных цветов, RGB палитра либо HEX код. Функция `fillRect` рисует заполненным цветом прямоугольник на холст с указанной длиной, шириной и координатами откуда начать его рисовать. Для этой функции имеется четыре обязательных параметра. Первый параметр – это X координата верхнего левого угла прямоугольника, откуда и будет происходить рисование, второй параметр – тоже самое что и первый параметр только стартовая координата Y, третий параметр – ширина прямоугольника, четвертый параметр – длина прямоугольника. Если третий и четвертый параметр будут иметь одинаковые значения, то этот прямоугольник будет являться квадратом, что логично. Теперь глянем на сам холст, что из заполненных прямоугольников можно подобное нарисовать

Веб-рисование на HTML5

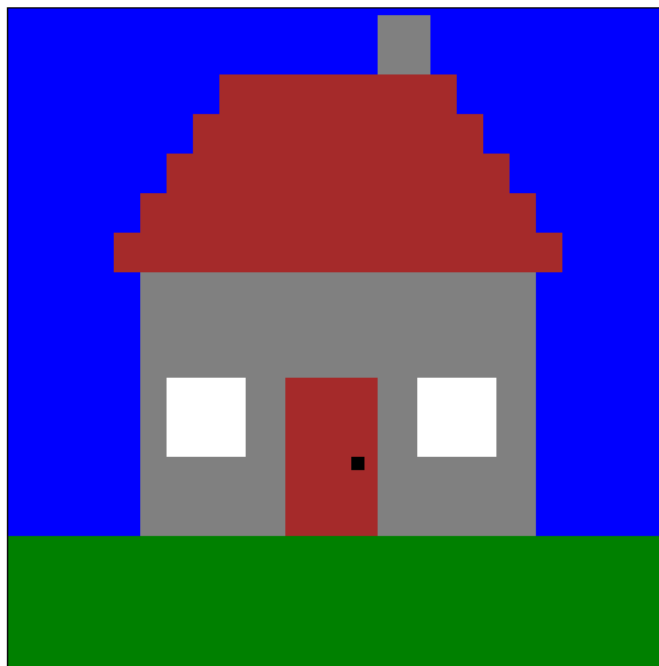


Рис. 4. Рисунок домика на canvas

Как видно на рисунке 4 изображен домик из заполненных цветами прямоугольников, довольно неплохой рисунок получился, теперь пойдём дальше и рассмотрим ещё функции для рисования на canvas. Новый код представлен на рисунке ниже

```
<script>
  var canv = document.getElementById("paint_block");
  var ctx = canv.getContext("2d");
  canv.width = 500;
  canv.height = 500;

  ctx.strokeStyle = "black";
  ctx.strokeRect(230, 230, 30, 30);
  ctx.strokeStyle = "blue";
  ctx.strokeRect(200, 200, 90, 90);
  ctx.strokeStyle = "green";
  ctx.strokeRect(170, 170, 150, 150);
  ctx.strokeStyle = "grey";
  ctx.strokeRect(140, 140, 210, 210);
  ctx.strokeStyle = "brown";
  ctx.strokeRect(110, 110, 270, 270);
  ctx.strokeStyle = "red";
  ctx.strokeRect(80, 80, 330, 330);
  ctx.strokeStyle = "orange";
  ctx.strokeRect(50, 50, 390, 390);
  ctx.strokeStyle = "violet";
  ctx.strokeRect(20, 20, 450, 450);
</script>
```

Рис. 5. Код рисования на холсте контурами прямоугольников на JavaScript

Глядя на рисунок 5, можно увидеть код и две новые функции `strokeStyle` и `strokeRect`, теперь рассмотрим их подробнее. Функция `strokeStyle` – это тоже самое что и `fillStyle` только здесь устанавливается цвет

кисти контура. Функция `strokeRect` похожа также как и `fillRect`, только на холсте рисует незаполненный цветом заливки прямоугольник с контурами. Параметры такие же, что и в `fillStyle` – стартовые X Y координаты, ширина и высота прямоугольника с контуром.

Теперь посмотрим, что можно с помощью незаполненных цветом прямоугольников с контурами нарисовать

Веб-рисование на HTML5

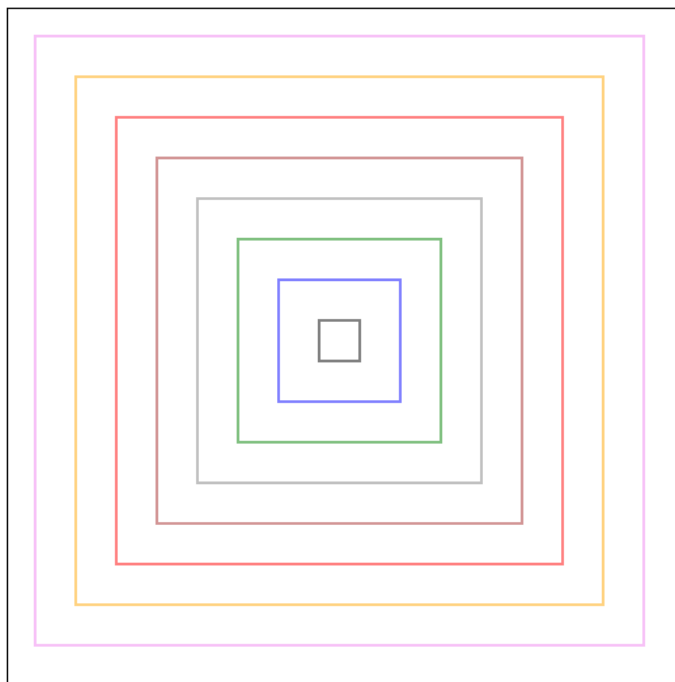


Рис. 6. Рисунок квадратов с разными цветами контура и размера на canvas

На рисунке 6, как видно изображены квадраты разных размеров и разных цветов контуры. Рисунок немного похож на пирамиду видом сверху. Теперь рассмотрим еще функции на canvas. Другой код, представлен на рисунке ниже

```
<script>
var canv = document.getElementById("paint_block");
var ctx = canv.getContext("2d");
canv.width = 500;
canv.height = 500;

ctx.beginPath();

ctx.moveTo(250, 60);
ctx.lineTo(350, 440);
ctx.lineTo(70, 200);
ctx.lineTo(420, 200);
ctx.lineTo(150, 440);
ctx.lineTo(250, 60);

ctx.strokeStyle = "blue";

ctx.moveTo(80, 100);
ctx.arc(80, 100, 50, 1 * Math.PI, 2/5 * Math.PI, false);
ctx.moveTo(420, 100);
ctx.arc(420, 100, 50, 1/4 * Math.PI, 2/4 * Math.PI, true);
ctx.moveTo(80, 400);
ctx.arc(80, 400, 50, 1/4 * Math.PI, 3/4 * Math.PI, false);
ctx.moveTo(420, 400);
ctx.arc(420, 400, 50, 1/4 * Math.PI, 11/6 * Math.PI, false);

ctx.fillStyle = "green";

ctx.stroke();
ctx.fill();
</script>
```

Рис. 7. Код рисования на холсте линий и дуг на JavaScript

На рисунке 7 представлен код с возможностью рисовать различные линии и дуги, тем самым введя новые функции – `beginPath`, `moveTo`, `lineTo`, `arc`, `stroke` и `fill` теперь рассмотрим эти функции подробнее. Функция `beginPath` дает команду canvas начала построения рисования линий последовательно. Функция `moveTo` программно перемещает «курсор» на указанные координаты в параметрах этой функции, `lineTo` – создает линию от координат начала, установленных функцией `moveTo` до координат конца, указанных в самой функции `lineTo`. Функция `arc`, которая рисует окружности и дуги немного сложная и имеет много параметров: первый и второй параметр – координаты начала рисования, третий параметр – радиус окружности или дуги, четвертый параметр – начальный угол, пятый параметр – конечный угол и шестой параметр указание направления рисования. Углы в четвертом и пятом параметрах указываются в радианах, поэтому был использован метод `Math.PI` который является числом «Пи» равным приблизительно 3.14. Последние две функции `stroke` и `fill` собственно рисуют линии и дуги на сам холст с контуром и заливкой.

Теперь увидим на рисунке, что можно подобное нарисовать с помощью линий и окружностей, а также дуг

Веб-рисование на HTML5

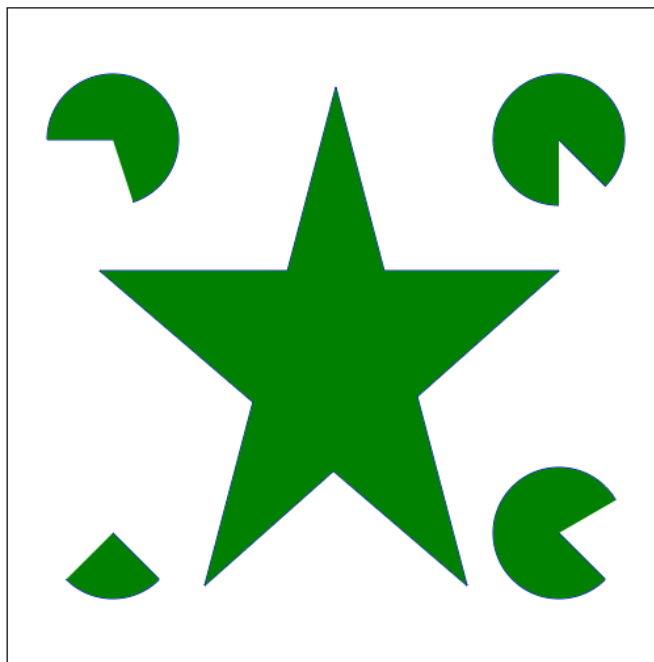


Рис. 8. Рисунок звезды и окружностей с вырезанными кусками на canvas

На рисунке 8 изображена звезда, нарисованная линиями и окружности с вырезанными кусочками. Фигуры имеют цвет контура черный, а цвет заливки – зеленый. Теперь осталось рассмотреть еще две функции их код представлен на рисунке ниже

```
<script>
var canv = document.getElementById("paint_block");
var ctx = canv.getContext("2d");
canv.width = 500;
canv.height = 500;

ctx.beginPath();

ctx.moveTo(10, 15);
ctx.bezierCurveTo(100, 100, 200, 100, 200, 200);
ctx.moveTo(10, 15);
ctx.bezierCurveTo(100, 100, 200, 300, 200, 200);
ctx.moveTo(10, 15);
ctx.bezierCurveTo(75, 55, 175, 20, 250, 15);
ctx.moveTo(10, 15);
ctx.quadraticCurveTo(100, 100, 250, 15);
ctx.moveTo(400, 400);
ctx.bezierCurveTo(100, 200, 350, 200, 400, 450);
ctx.moveTo(200, 300);
ctx.quadraticCurveTo(240, 200, 320, 410);
ctx.stroke();
</script>
```

Рис. 9. Код рисования кривых Безье и квадратичных кривых на JavaScript

Глядя на рисунок 9 можно увидеть две функции это – `bezierCurveTo` и `quadraticCurveTo`. Функция `bezierCurveTo` рисует кривую Безье, ее первые четыре параметра это X Y координаты двух опорных (контрольных) точек для кривой Безье, последние пятый и шестой параметры задают координаты конца кривой Безье, `quadraticCurveTo` – рисует квадратичную кривую, первые два параметра являются X Y координатами опорной (контрольной) точки, остальные два – координаты конца квадратичной кривой.

После осталось увидеть, что можно нарисовать кривыми на холсте.

Веб-рисование на HTML5



Рис. 10. Рисунок кривых на canvas

На последнем рисунке 10 изображены кривые – Безье и квадратичные, с помощью этих кривых можно изобразить листочки стеблей, петли, необычные дуги и многое другое.

Таким образом, в статье была рассмотрена возможность веб-рисования на HTML5 с помощью `canvas`, используя язык программирования JavaScript. Также были рассмотрены рисунки, нарисованные на `canvas`, используя различные инструменты. Графические онлайн веб-редакторы используют `canvas`, избавляя пользователей от возможности установки программного обеспечения на устройства, только им необходим всего лишь современный браузер, имеющий поддержку HTML5 и доступ в интернет.

Библиографический список

1. Дядьева М.Н. Использование библиотеки Raphael для создания SVG – графики // В сборнике: Приоритеты и научное обеспечение технологического прогресса. Сборник статей Международной научно-практической конференции. 2016. С. 65-66.

2. Дронов В. HTML 5, CSS 3 и WEB 2.0. Разработка современных Web-сайтов. СПб, 2011.
3. Фримен Э., Робсон Э. Изучаем программирование на HTML5. СПб, 2012.
4. Ляпин П.С., Мельничук Р.М., Финогенов А.Д. Графические редакторы для схемотехнического проектирования // Вісник Національного технічного університету України "Київський політехнічний інститут". Серія: Інформатика, управління та обчислювальна техніка. 2012. № 55. С. 188-193.
5. Нуромская Н.Н., Попова И.К., Шестакова О.А. Методика работы с графическим редактором GIMP // Психолого-педагогический журнал Гаудеамус. 2010. Т. 2. № 16. С. 185-188.
6. CANVAS шаг за шагом: Основы URL: <https://habr.com/post/111308/> (дата обращения 14.07.2018)
7. Методы и свойства canvas URL: <https://webonrails.ru/post/102724754472964822/> (дата обращения 14.07.2018)