Привет, Друг!

Ты уже познакомился с основными тегами html.

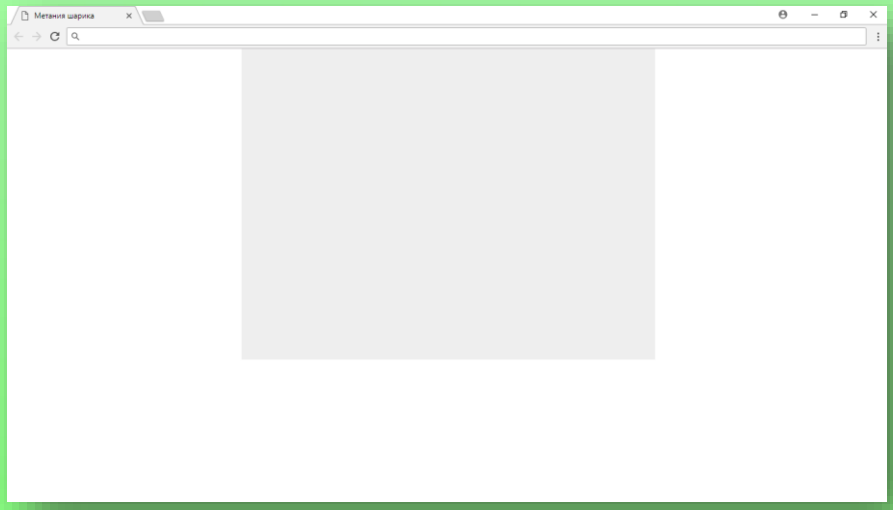
Выполняя это задание, ты откроешь для себя огромную новую ветвь веб-программирования - с помощью JavaScript.

Начнём рассматривать **Canvas** (с англ. – холст, канва) – это новый элемент HTML5, предназначенный для создания растровых изображений с помощью команд JavaScript.

С помощью этого элемента можно создавать всевозможные игры. Canvas поддерживает 3d и 2d графику. Мы начнём изучение с 2d. Будем двигаться от простого – к сложному.

Но, прежде, чем начать разрабатывать игры, давай вспомним, что такое координатная плоскость и научимся рисовать примитивы, с помощью JavaScript.

Создаем игровое поле



Вот так у меня выглядит поле, где мы будем осуществлять некоторые действия. Т.е. это некоторая область экрана, ширину и высоту которой мы зададим в нашем коде. Эта область может быть бесцветной. Но я предпочитаю закрашивать её на стадии разработки каким-то цветом, чтобы видеть границы.

Чтобы создать это поле canvas:

1. Создай пустую html-страницу, в заголовке страницы можешь написать «Canvas. Основы».
2. Внутри тега <head>, после тега </title>, впиши вот эти строки:

```
<style>
canvas { background: #eee; display: block; margin: 0 auto; }
</style>
```

Здесь мы задаем цвет нашему полю canvas и размещаем его в центре экрана. Это **CSS**, изучать его и разбираться будем немного позже.

3. Теперь в теле своего документа нужно создать сам элемент canvas. Это делается с помощью тега **<canvas>**, нужно обязательно задать **id** нашему полю. У меня это «**mycanvas**». По этому id в дальнейшем будем проводить все действия на холсте.

Создаем игровое поле

```
<!DOCTYPE html>
<html>
<head>
  <title>Canvas. Основы.</title>
  <style>
      canvas { background: #eee; display: block; margin: 0 auto; }
  </style>

</head>
<body>
<canvas id="myCanvas"></canvas>
```

Такой код получился. Если запустить данную страницу в Google Chrome, то ты увидишь в верхней части своей страницы небольшой серый прямоугольник. Далее нужно задать размеры этого прямоугольника. Это можно сделать с помощью атрибутов **width** и **height** тега Canvas или с помощью команд JavaScript позже (правильнее будет задать в JavaScript, чтобы потом можно было использовать значения размеров поля для размещения объектов. В следующий раз так и сделаем, а пока мы только начинаем знакомиться с Canvas, давай без изысков зададим размер здесь:

```
<canvas id="myCanvas" width='640' height='480'></canvas>
```

Теперь, при запуске, у тебя на экране поле, такое же, как на картинке в начале материала.

Чтобы что-то изобразить на этом поле, открываем после тега </canvas> **<script></script>**, внутри него, в первую очередь нужно задать браузеру исходные параметры:

```
<script>
    var canvas = document.getElementById("myCanvas");
    var ctx = canvas.getContext("2d");
```

Первая переменная - **canvas** - это наш холст из текущего документа. Вторая переменная – **ctx** – указывает, что содержимое нашего поля будет в формате 2d.

Область для рисования готова. Разберем команды для рисования примитивов.

Рисуем прямоугольник

Для рисования любого примитива в JavaScript нужно сначала указать, что мы хотим начать рисование (как в Скрэтче «Опустить перо»). Для этого используем команду

```
ctx.beginPath();
```

Здесь ctx – «родитель», т.е. наш холст в формате 2d, а beginPath() – команда (с англ. – «начать путь»). Теперь можно указывать, что мы рисуем. Первый пример будет – прямоугольник. Указываем снова, что рисуем мы на холсте ctx, после точки пишем rect (от англ. rectangle – прямоугольник):

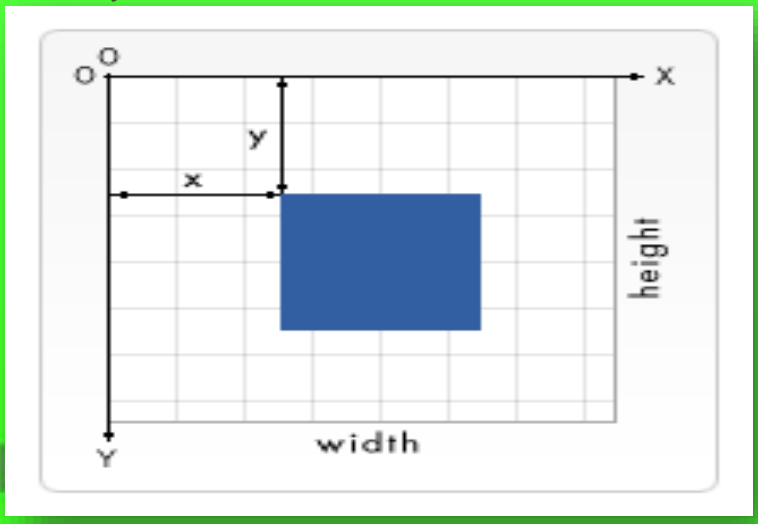
```
ctx.rect(20, 40, 30, 30);
```

В скобках указываем 4 параметра:

Первые два – координаты левого верхнего угла прямоугольника;

Последние два – ширина и длина прямоугольника, соответственно.

По поводу координат. Всё наше поле canvas представляет собой координатную плоскость с центром (точкой (0, 0)) в левом верхнем углу. Если двигаться вправо, то координата x возрастает, в крайней правой части холста x=640 (т.е. заданной ширине canvas). Координата y возрастает по мере движения вниз, в нижней части принимает значение 480 (т.е. height холста)



Рисуем примитивы. Прямоугольник

После того, как мы написали строку кода, описывающую наш прямоугольник, нам нужно выбрать стиль заливки. Это может быть:

- сплошной цвет ctx.fillStyle = 'цвет';
- цветной контур ctx.strokeStyle = 'цвет';

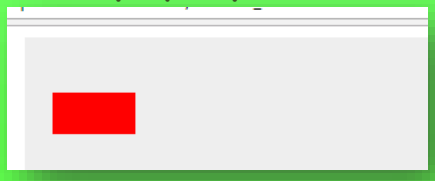
Я выбрала сплошную заливку красным цветом - '#FF0000'.

В следующих строках я заливаю свой прямоугольник - **ctx.fill()**; и завершаю рисование объекта – **ctx.closePath()** (с англ. – завершить путь).

Вот такой код у меня получился:

```
ctx.beginPath(); //начинаем рисовать объект
ctx.rect(20, 40, 60, 30); //указываем, ЧТО рисуем r
ctx.fillStyle = "#FF0000"; //стиль заливки и цвет
ctx.fill(); //заливаем
ctx.closePath(); //закончили рисовать объект
```

Открываю в браузере и получаю вот такое изображение:



Нужно заметить, что если ты собираешься рисовать несколько объектов одного цвета и стиля, то можно использовать вот такой вариант:

```
ctx.fillStyle = "#FF0000"; //стиль заливки и цвет
ctx.beginPath(); //начинаем рисовать объект
ctx.fillRect(20, 40, 60, 30); //указываем, ЧТО рисуем rect
ctx.closePath(); //закончили рисовать объект
        ctx.beginPath(); //начинаем рисовать объект
        ctx.fillRect(200, 60, 90, 80); //указываем, ЧТО ри
        ctx.closePath(); //закончили рисовать объект
```

В начале кода задаем цвет заливки, а потом используем команду **fillRect** – заполненный прямоугольник. Аналогично работает **strokeRect**, поэкспериментируй.

Рисуем примитивы. Прямая линия.

Следующие пара команд:

moveTo(x,y) – перемещает виртуальный «карандаш» в точку с координатами x и y;

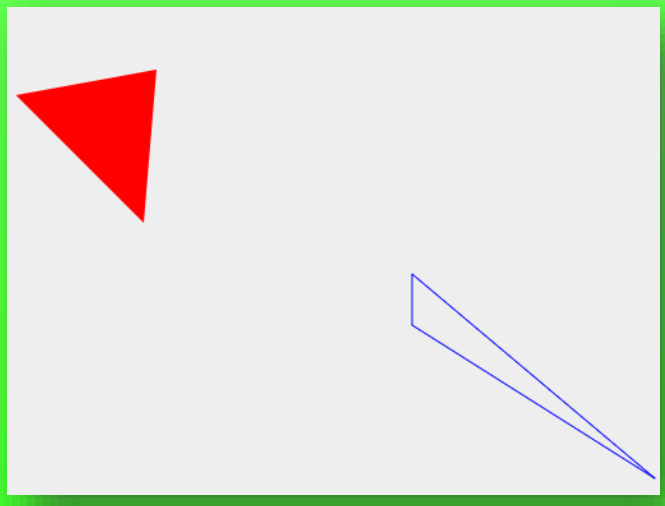
lineTo(x,y) – рисует прямую линию из текущей точки в точку с координатами x и y. Чтобы линия стала видима пользователю, необходимо задать **strokeStyle** = 'цвет' и после **lineTo(x,y)** написать **stroke()**.

Казалось бы, всё просто, но есть некоторые фишки:

```
ctx.beginPath(); //начинаем рисовать объект
ctx.moveTo(200,60);
ctx.lineTo(90, 80);
ctx.lineTo(190, 180);
ctx.lineTo(200, 60);
ctx.fill();
ctx.closePath(); //закончили рисовать объект
```

С помощью вот такого кода, можно получить закрашенный треугольник.

Если ты хочешь получить не закрашенный объект, не забудь в конце указать **stroke()**.



Ещё одна интересная команда, которая тебе пригодится – **lineWidth** - толщина линии ☺

```
ctx.lineWidth = 3;
```

Рисуем примитивы. Дуга, окружность, круг.

Следующая команда:

arc(x,y,r,a1,a2,s) – создает дугу окружности;

x, y – координаты центра окружности;

r – радиус окружности;

a1 – начальный угол (в радианах);

a2 – конечны угол (в радианах);

s – направление рисования дуги окружности (true – против часовой стрелки, false – по часовой стрелке), параметр необязателен, по умолчанию, false.

Остановимся на новом для тебя слове **Радиан**.

Ты знаешь, что углы измеряются в градусах. Но второй вариант единицы измерения – радианы. В математике есть понятие – число Пи, оно равно 3,14 рад. Длина любой окружности – два Пи, а если в градусах – 360.

Если что-то непонятно, попробуй почитать тематический материал в интернете, либо спроси ведущего.

В JavaScript есть встроенный объект Math, содержащий основные математически константы и функции. Число Пи записывается так: Math.PI.

Получается, что нарисовать окружность можно вот так:

```
ctx.beginPath();
ctx.arc(350,200,150,2*Math.PI,0);
ctx.stroke();
ctx.closePath();
```

В результате увидим окружность радиусом 150 пикселей, в центре с точкой (350,200), а рисует её компьютер от угла 2*Пи (это 360 градусов) до угла 0, т.е. полностью. Если вместо 0, мы напишем Math.PI, что получится?

Нарисуй вот такого персонажа с помощью arc.

Задания

Мы разобрали основные примитивы Canvas, пришла пора самостоятельных заданий.

Первое задание - Найди в интернете и изучи самостоятельно, как нарисовать эллипс (овал).

Второе задание – нарисуй с помощью JavaScript и Canvas домик. Обязательные условия – использование ВСЕХ изученных примитивов (прямоугольник, линия, окружность, овал).

У меня получилось так:

Дополни свой рисунок каким-то интересным объектом, получи бонус от ведущего.

СОФТИУМ

Остались вопросы, спроси ведущего! ☺