

## Лабораторная работа № 2. Элемент Canvas

Цель: изучить возможности рисования на веб-странице.

### Теория

Элемент `<canvas>` позволяет рисовать на веб-страницах произвольные фигуры с помощью JavaScript (или других клиентских скриптов). Сам по себе canvas ничего не рисует, это холст, который предоставляет возможности для рисования. Создать canvas можно так:

```
<canvas id='draw' width='300' height='200'>...</canvas>.
```

Для создания прямоугольников:

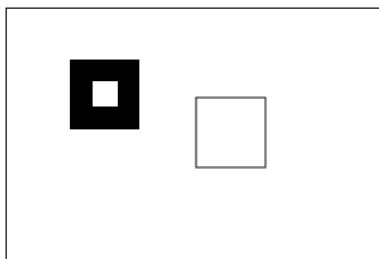
**fillRect(x,y,ширина,высота)** – рисует закрашенный прямоугольник;

**strokeRect(x,y,ширина,высота)** – не закрашенный прямоугольник;

**clearRect(x,y,ширина,высота)** – очищает указанную зону (x и y – величина смещения прямоугольника от верхнего левого угла холста в пикселях).

В следующем примере на экран выводятся два прямоугольника: закрашенный с очищенной зоной внутри и не закрашенный. Первые две строки являются стандартными для рисования любого объекта в canvas.

```
var canvas = document.getElementById("draw");  
var x=canvas.getContext("2d");  
x.fillRect(50,40,55,55);  
x.strokeRect(150,70,55,55);  
x.clearRect(68,57,20,20);
```



### Рисование составных фигур

Составные фигуры состоят из нескольких соединенных простых объектов.

**moveTo(x,y)** – устанавливает координаты точки, из которой начнется рисование следующего объекта;

**lineTo(x,y)** – рисует прямую линию;

**arc(x,y,радиус,нач\_угол,конеч\_угол)** – рисует круг. Угол необходимо задавать в радианах, (радианы=(Math.PI/180)\*градусы);

**rect(x, y, ширина, высота)** – рисует прямоугольник.

```
x.beginPath();  
x.moveTo(20,20);  
x.lineTo(70,70);  
x.lineTo(20,70);  
x.closePath();  
x.fill();
```

Для раскрашивания нарисованных в canvas фигур предусмотрены свойства: **fillStyle** (применяется к закрашенным фигурам) и **strokeStyle** (к незакрашенным фигурам).

```
x.fillStyle="green";  
x.strokeStyle="#FF45FF"  
x.fillStyle="rgb(255,73,73)"  
x.fillStyle="rgba(0,0,0,0.5)" //(0.5 – прозрачность)
```

Для оформления линий используются: **lineWidth** – ширина линии, **lineCap** – кончики линий (round – закругленные) , **lineJoin** – сглаживание стыков двух линий (round).

### Градиент

**CreateLinearGradient(x1,y1,x2,y2)** – создать линейный градиент, x1 и y1 координаты начальной, x2 и y2 – конечной точки градиента. После создания градиента надо указать цвета перехода с помощью метода **addColorStop(точка,цвет)**.

```
var grad =x.createLinearGradient(0,0,0,150);  
grad.addColorStop(0.0,'black');  
grad.addColorStop(1.0,'white');  
x.fillStyle=grad;  
x.fillRect(20,20,200,200);
```

### Создание теней

**shadowOffsetX** – смещение тени от объекта по горизонтали (может быть отрицательным).

**shadowOffsetY** – смещение тени по вертикали (может быть отрицательным)

**shadowBlur** – величина размытия тени

**shadowColor** – цвет тени (по умолчанию черный).

```
x.shadowOffsetX=3;  
x.shadowOffsetY=3;  
x.shadowBlur=5;  
x.shadowColor='black';  
x.fillStyle='#ffaa00';  
x.fillRect(50,40,55,55);
```

## Текст

Метод **fillText("текст",x,y)** позволяет отображать в элементе canvas произвольный текст, который может быть оформлен с помощью свойства font.

```
x.font='15px Verdana';  
x.fillStyle='#60016d';  
x.fillText("Я студент ВГТУ", 10, 40);
```

## Вставка изображений

В canvas могут быть вставлены изображения форматов PNG, GIF и JPEG с помощью метода **drawImage('ссылка на картинку',x,y)**

```
<img id='image1' src='ris.jpg' style='display:none;' />  
<canvas id='draw' width='400' height='300' style='border:1px  
solid'></canvas>  
<script>  
Var img=document.getElementById('image1');  
var canvas=document.getElementById("draw")  
var x=canvas.getContext("2d");  
x.drawImage(img,10,10);  
</script>
```

## Задания для лабораторной работы № 2

**Задание 1.** Создайте гистограмму, используя элемент Canvas.

Порядок выполнения:

1. Создайте чистый лист для рисования. Определите функцию (например, **drawgist**).
2. Используя **moveTo()** и **lineTo()**, начертите оси.
3. Используя **fillText()**, добавьте обозначения. Задайте тексту размер, начертание, шрифт.
4. Объектами гистограммы будут служить прямоугольники - **fillRect()**. Нарисуйте их необходимое количество над горизонтальной осью. Задайте различные цвета, примените градиентную заливку, заливку изображением (функция **createPattern()**).
5. Добавьте детали к графику для лучшей визуализации информации.

**Задание 2.** Создайте диаграмму, используя элемент Canvas.

Порядок выполнения:

1. Перед началом написания кода, просчитайте необходимые значения градусов (процентов), опираясь на данные.

2. Определите функцию (например, *drawdiagr*).
3. Используя *arc()*, нарисуйте круг-основу.
4. Нарисуйте сектора. Для этого удобно создать универсальную функцию, которая, получая значения центра окружности, радиуса, начального и конечного угла, направления, будет рисовать необходимую часть диаграммы. Функция также должна заливать объект цветом.

```
x.fillStyle = "black";  
x.beginPath();  
x.arc(100, 100, 50, -Math.PI/180*90, 0, false);  
x.lineTo(100,100);  
x.fill();
```

5. Используя *fillText()*, добавьте обозначения. Задайте тексту размер, начертание, шрифт. К надписи названия диаграммы добавьте тень.

#### Примеры тематики:

- Количество лет, проведенных в конкретном учебном заведении (школа, колледж, университет и т.п.)
- Режим дня – сколько часов занимает какое-либо действие (сон, учеба, досуг)
- Посещение различных стран и количество проведенных в них дней
- Место рождения одноклассников по областям Беларуси и их количество.
- Количество одноклассников, рожденных зимой, весной, летом, осенью.
- Изучаемые дисциплины, отведенное на них число часов.
- Темы лабораторных работ и их количество.

## Примеры выполнения:

