

Рисование с помощью элемента управления Canvas

Пример 1.

Определите сайт и создайте новую страницу primer1.html.

Перейдите в код страницы и внутри тега body определите элемент canvas ширина – 600, высота – 500, id="canvas".

```
<body>
<canvas id="canvas" width="600" height="500"> </canvas>
</body>
</html>
```

В разделе head нужно подготовить функцию, куда будут вписываться команды по рисованию на холсте. А в тег body вписать вызов этой функции.

```
<title>Untitled Document</title>
<script>
function loader()
{
}
</script>
</head>

<body onload="loader()">
<canvas id="canvas" width="600" height="500"> </canvas>
</body>
```

Запись onload="loader()" означает, что при загрузке страницы (событие onload) будет вызвана функция loader.

Введите выделенный код сценария для создания объекта, соответствующего элементу управления Canvas. Этот объект будет использоваться для доступа к элементу управления Canvas из сценария.

```
<script>
function loader()
{
var canvas = document.getElementById ('canvas');
var canvas1= canvas.getContext ('2d');
}
</script>
</head>

<body onload="loader()">
<canvas id="canvas" width="600" height="500"> </canvas>
```

Сначала обращаемся к холсту через его id – в первой строке. Потом получаем контекст отрисовки с помощью команды getContext.

Рисование прямоугольников

Полые прямоугольники можно рисовать с помощью функции strokeRect, а заполняемые с помощью функции fillRect. Обеим функциям в качестве аргументов передаются координаты (x, y) левого верхнего угла прямоугольника, а также его ширина и высота (w, h). Начало координат соответствует левому верхнему углу.

Цвет заполнения будем задавать с помощью атрибута fillStyle. В этом атрибуте задается цвет, указываемый с помощью функции rgba(), которой в качестве аргументов передаются значения красной, зеленой и синей составляющих выбираемого цвета, а также коэффициент видимости в пределах от 0-1.

В коде допишите следующее

```

var canvas = document.getElementById ('canvas');
var canvas1= canvas.getContext ('2d');
//прямоугольники
canvas1.fillStyle="rgba(0,0,200,1)";
canvas1.fillRect(30,30,75,70);
canvas1.fillStyle="rgba(200,200,0,1)";
canvas1.fillRect(70,50,55,70);
canvas1.fillStyle="rgba(200,0,0,1)";
canvas1.fillRect(90,50,75,50);
}
</script>

```

Сохраните страницу и посмотрите результат.
Должно получиться следующее.



Рисование штриховых рисунков

С помощью элемента управления Canvas можно рисовать штриховые рисунки. Этот процесс начинается с функции `beginPath()`, уведомляющей элемент управления Canvas о том, что создается фигура. А затем с помощью функций `moveTo()` и `lineTo()` задается исходное положение для рисования и фактически рисуются прямые линии.

По завершении рисования фигуры вызывается функция `closePath()`, замыкающая контур нарисованной фигуры. После этого фигура обводится с помощью функции `stroke()`.

Нарисует три треугольника.

Допишите следующее в коде.

```

canvas1.fillStyle="rgba(200,0,0,1)";
canvas1.fillRect(90,50,75,50);
//штриховые треугольники
canvas1.beginPath();
canvas1.strokeStyle="rgba(200,0,0,0.5)";
canvas1.moveTo(110,205);
canvas1.lineTo(110,125);
canvas1.lineTo(30,205);
canvas1.closePath();
canvas1.stroke();

canvas1.beginPath();
canvas1.moveTo(100,205);
canvas1.lineTo(100,125);
canvas1.lineTo(20,205);
canvas1.closePath();
canvas1.stroke();

canvas1.beginPath();
canvas1.moveTo(90,205);
canvas1.lineTo(90,125);
canvas1.lineTo(10,205);
canvas1.closePath();
canvas1.stroke();
}
</script>

```

Сохраните страницу и посмотрите результат работы.

Заполнение штриховых рисунков

Штриховые фигуры можно не только рисовать, но и заполнять выбранным цветом. Для заполнения фигуры будет использована функция fill().
Функция fill() заполняет нарисованную форму текущим цветом заполнения виртуального полотна, который задается в атрибуте fillStyle.

Добавьте следующий код в скрипт

```
//заполняемый треугольник
canvas1.fillStyle = "rgba(0,200,0,0.5)";
canvas1.beginPath();
canvas1.moveTo(225,25);
canvas1.lineTo(305,25);
canvas1.lineTo(225,105);
canvas1.closePath();
canvas1.fill();
}
</script>
```

Сохраните страницу и проверьте работу.

Рисование кривых Безье

Рисование линий не ограничивается только прямыми линиями. Используя функцию bezierCurveTo(), можно также рисовать кривые линии Безье.

Чтобы нарисовать форму сердечка добавьте в скрипт следующий код.

```
//форма сердечка
canvas1.fillStyle = "rgba(200,0,0,0.5)";
canvas1.beginPath();
canvas1.moveTo(72,250);
canvas1.bezierCurveTo(75,247,70,235,50,235);
canvas1.bezierCurveTo(20,235,20,275.5,20,272);
canvas1.bezierCurveTo(20,290,40,312,75,330);
canvas1.bezierCurveTo(110,312,130,290,130,272);
canvas1.bezierCurveTo(130,275.5,130,235,100,235);
canvas1.bezierCurveTo(85,235,75,247,75,250);
canvas1.closePath();
canvas1.fill();
}
</script>
```

Сохраните страницу и проверьте работу.

Рисование кривых второго порядка

Используя функцию quadraticCurveTo() можно рисовать кривые второго порядка.

Добавьте следующий код в скрипт.

```
//кривые второго порядка
canvas1.strokeStyle="rgba(0,0,0,1)";
canvas1.beginPath();
canvas1.moveTo(275,125);
canvas1.quadraticCurveTo(225,125,225,162);
canvas1.quadraticCurveTo(260,200,265,200);
canvas1.quadraticCurveTo(325,200,325,162);
canvas1.quadraticCurveTo(325,125,275,125);
canvas1.closePath();
canvas1.stroke();
```

Сохраните и посмотрите результат.

Рисование дуг

Рисование дуг осуществляется с помощью функции arc().

Добавьте следующий код.

```
// дуги
canvas1.beginPath();
canvas1.arc(275,275,50,0,Math.PI*2,true);
canvas1.moveTo(310,275);
canvas1.arc(275,275,35,0,0.75*Math.PI,false);
canvas1.moveTo(300,255);
canvas1.arc(265,255,35,0,0.5*Math.PI,false);
canvas1.moveTo(280,255);
canvas1.arc(245,255,35,0,0.2*Math.PI,false);
canvas1.closePath();
canvas1.stroke();

}
</script>
```

Сохраните страницу и проверьте результат.

Рисование текста

Для этого нужно сначала выбрать подходящий шрифт, а затем установить его типографские параметры в атрибуте font элемента управления Canvas.

Допишите следующий код

```
// текст
canvas1.font="italic 40px sans-serif";
canvas1.strokeText("Привет!", 50,400);
```

Сохраните страницу и проверьте работу.