

Элемент canvas

Элемент canvas

Canvas (холст) – элемент HTML5, предназначенный для создания **растрового** изображения при помощи Javascript.

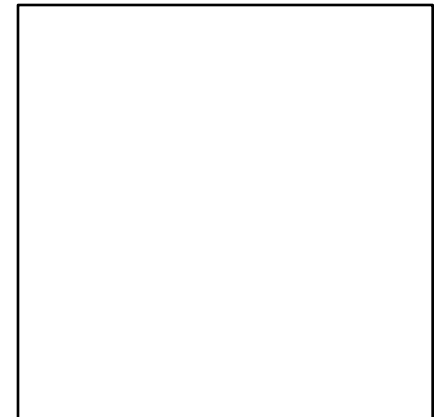
```
<canvas id="diagonal" width="200"  
height="200"></canvas>
```

Атрибуты:

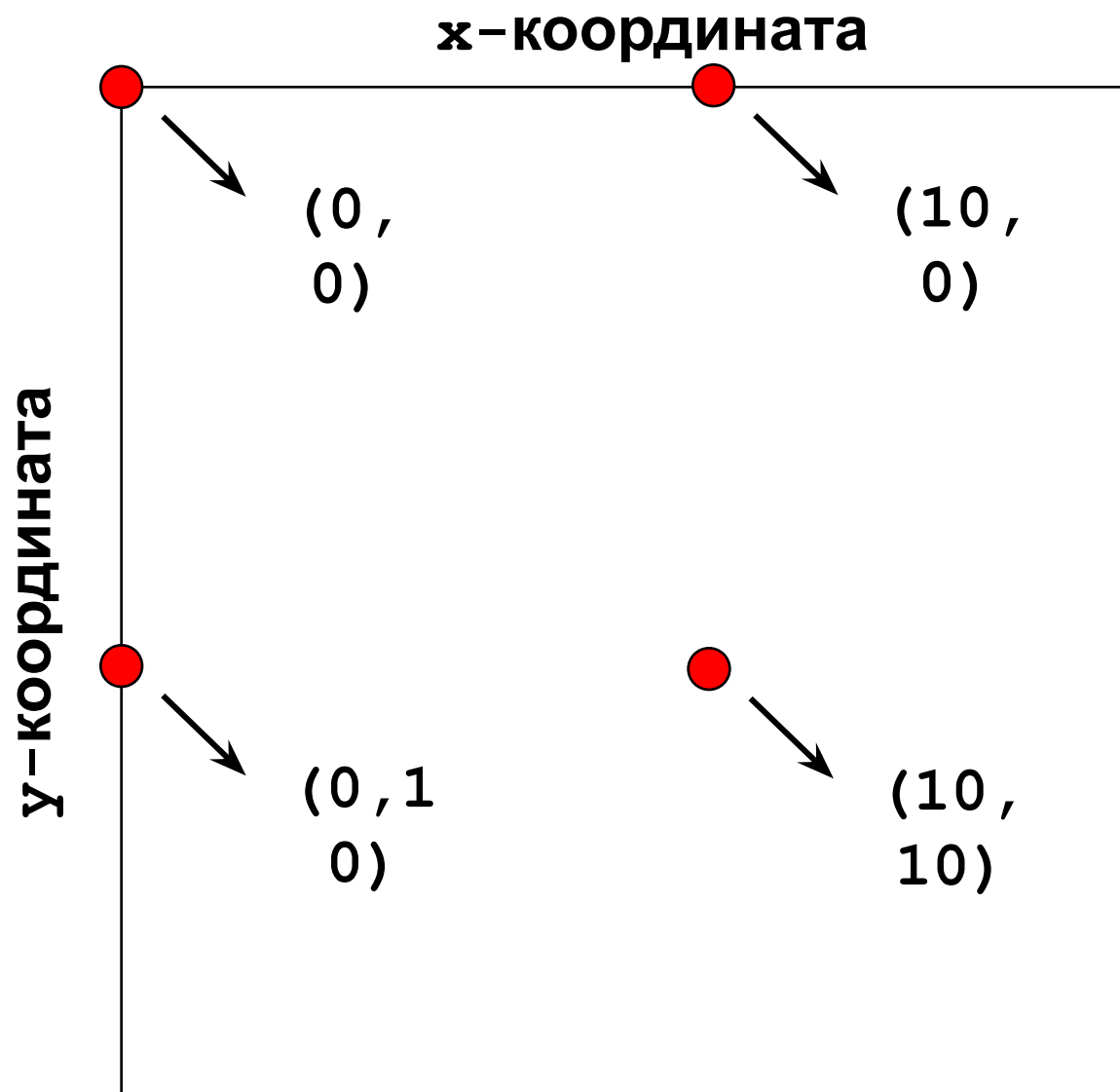
id – необходим для нахождения элемента canvas программным путем.

width – ширина холста

height – высота холста



Начало отсчета координат на холсте



Вычерчивание линии

```
<script>
function drawDiagonal() {
var canvas = document.getEleme
var context = canvas.getContex
context.beginPath();
context.moveTo(70, 140);
context.lineTo(140, 70);
context.stroke();
}
window.addEventListener("load", drawDiagonal,
true);
</script>
```

Получаем доступ

Задаётся

Начинается

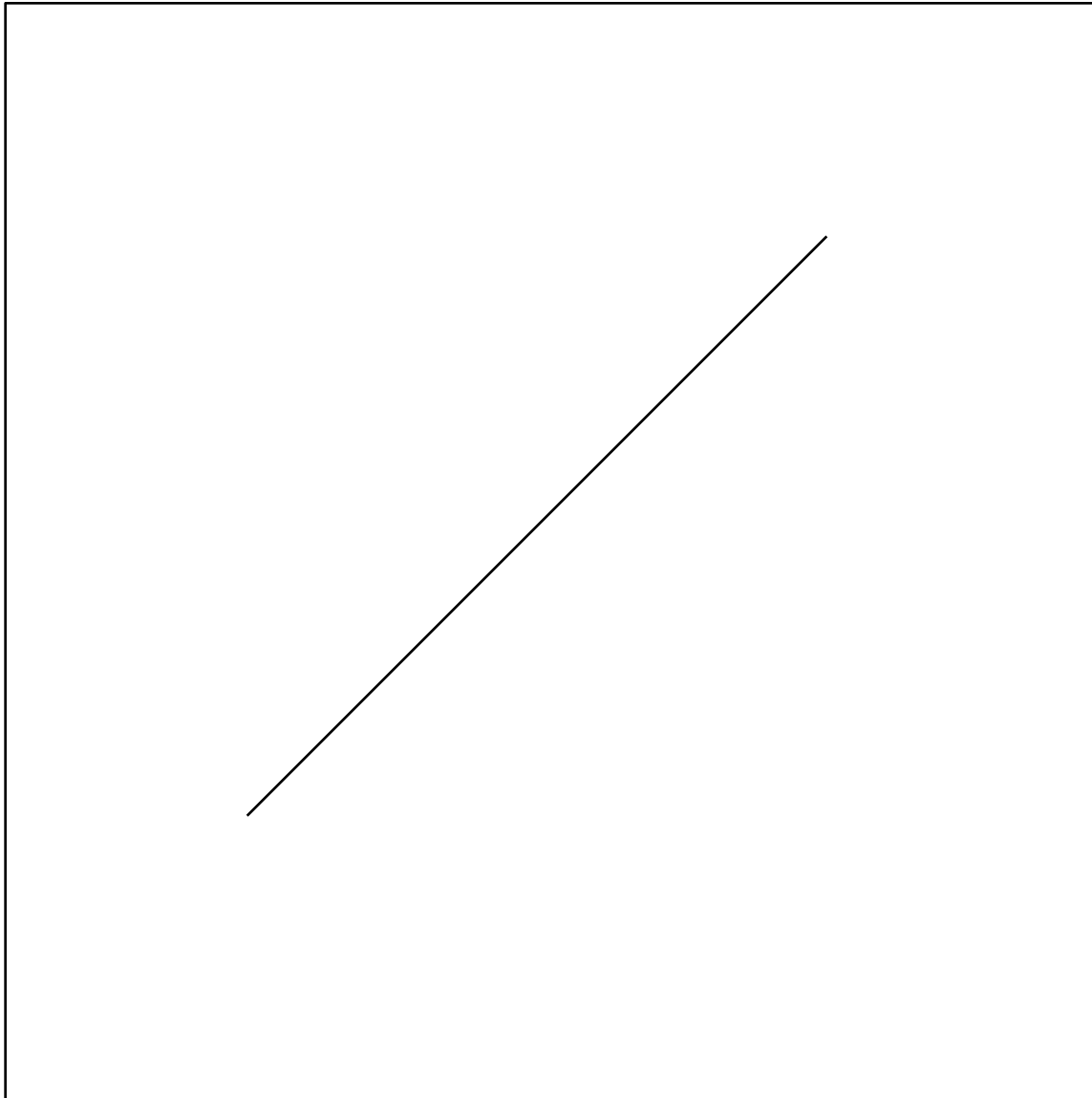
Передвижение в

Прорисовка линии до

Прорисовка
заданных линий

После загрузки окна будет
запущена функция
drawDiagonal

Вычерчивание линии. Результат



Рисование путей – 1

`beginPath()` – создает путь

`closePath()` – закрывает путь

`stroke()` – рисует границу фигуры

`fill()` – рисует закрашенную фигуру

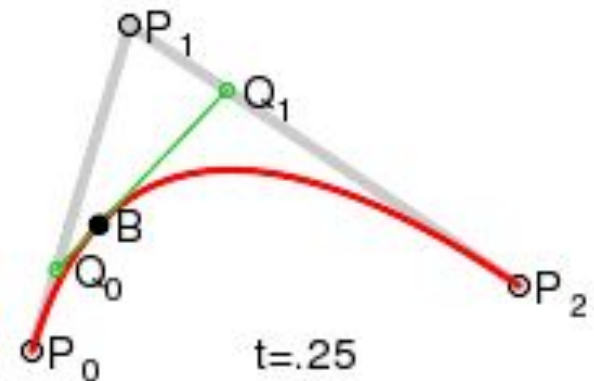
`moveTo(x, y)` – перенос в точку (x, y)

`lineTo(x, y)` – отрисовывает линию из текущей точки в точку (x, y)

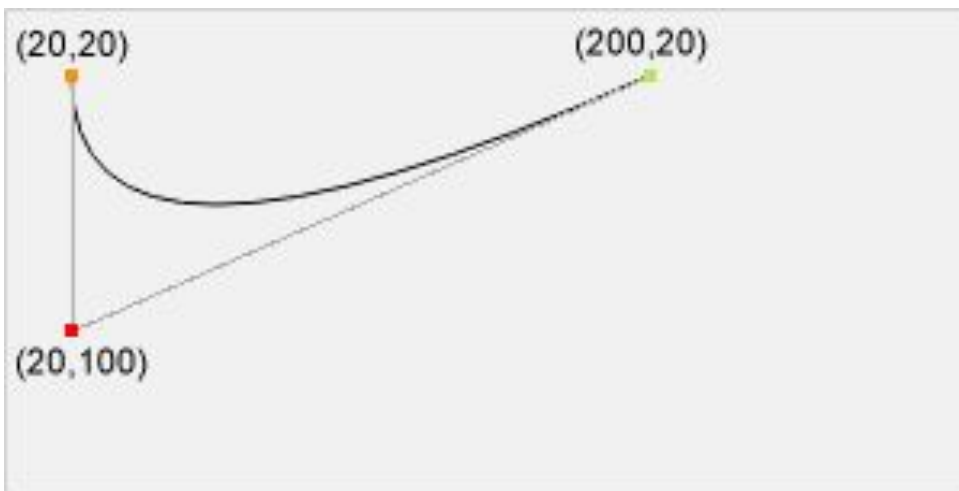
`arc(x, y, radius, startAngle, endAngle, anticlockwise)` – рисует дугу с центром в точке (x, y), радиусом `radius`, `startAngle` и `endAngle` определяют точки начала и конца дуги в радианах, `anticlockwise` логический параметр (по часовой стрелке или против).

Рисование путей – 2

`quadraticCurveTo(p1x, p1y, p2x, p2y)` –
квардатичная кривая Безье

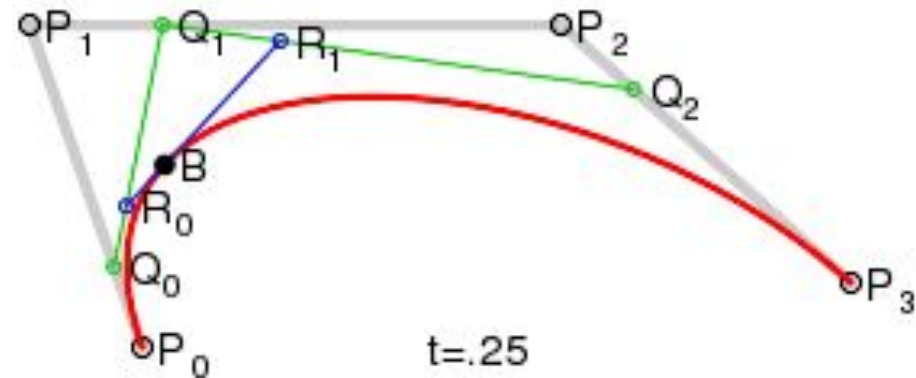


```
ctx.moveTo(20,20);
ctx.quadraticCurveTo(20,100,200,20);
```

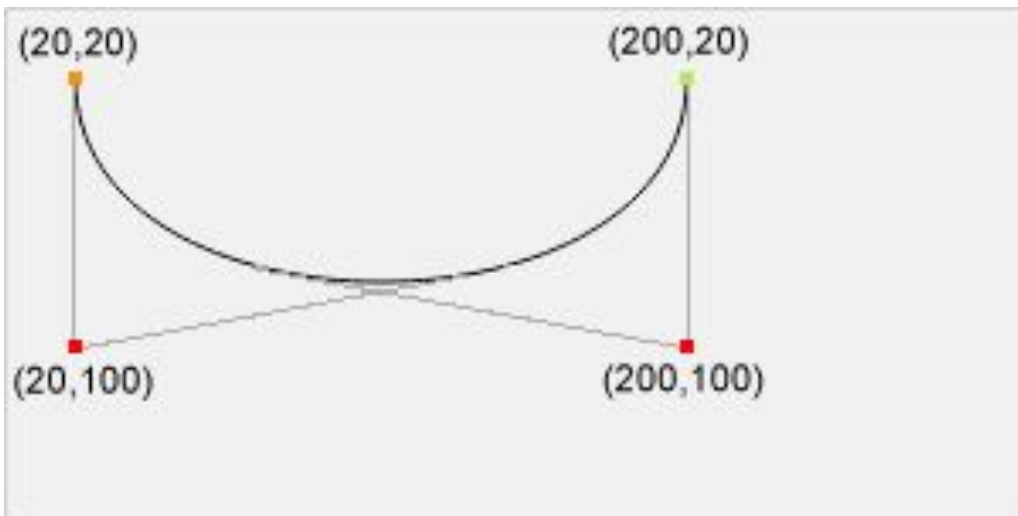


Рисование путей – 3

`bezierCurveTo(p1x, p1y, p2x, p2y, p3x, p3y)` –
кубическая кривая Безье



```
ctx.moveTo(20,20);
ctx.bezierCurveTo(20,100,200,100,200,20);
```



Градиенты

`createLinearGradient(x0, y0, x1, y1)` – создает линейный градиент из точки `(x0, y0)` в точку `(x1, y1)`

`createRadialGradient(x0, y0, r0, x1, y1, r1)` – создает радиальный градиент из точки `(x0, y0)` с радиусом `r0` в точку `(x1, y1)` с радиусом `r1`

`addColorStop(stop, color)` – указывает цвет и позицию в градиенте (`stop` от 0 до 1)

```
grd = ctx.createLinearGradient(0, 0, 170, 0);  
grd.addColorStop(0, "yellow");  
grd.addColorStop(1, "blue");
```

