Canvas

Ref: developer.mozilla.org/en-US/docs/Web/API/Canvas_API/Tutorial/Basic_usage

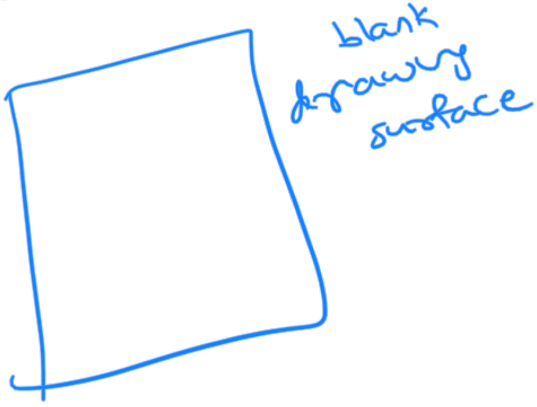
default

```
<canvas id = "mycanvas" width = "300"
                        height = "150">
        - - - - - -      </canvas>
```

image finally is scaled to fit layout size (CSS might distort -)

fallback content whatever is here will be displayed instead if browser doesn't support canvas

creates



script → access "rendering context" and draw on it

```
var canvas = document.getElementById
                        (`mycanvas);

    var ctx = canvas.getContext (`2d');
```

= canvas RenderingContext2D

type of context

script

sc

can check if canvas supported

```
if   canvas.getContext {
        var  ctx =  canvas.getContext ('2d');
                          :
 } else
 {
          no canvas code
 }
```

```
<html>
<body  onload = "draw();">
    <canvas> . - -    </canvas>
  </body>
 </html>
```

<u>JS</u>

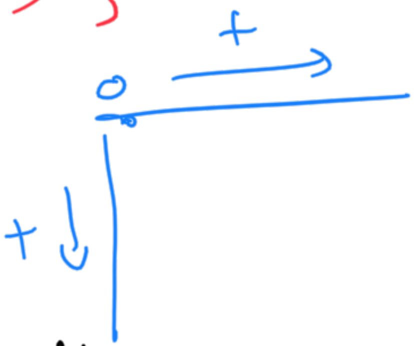
```
function  draw () {
 var   canvas  =  document. getElementbyId
                                  ('mycanvas);
    if   canvas.getContext {
            var  ctx =  canvas.getContext ('2d');
                             :
}

<style type = "text/css">
    canvas { border : 1px solid black; }
  </style>

}
```

Rectangles

```
ctx . fillStyle = 'rgb(200, 0, 0)';
ctx . fillRect (10, 10, 50, 50);
```

← top left corner ↓ (x, y, width, height)

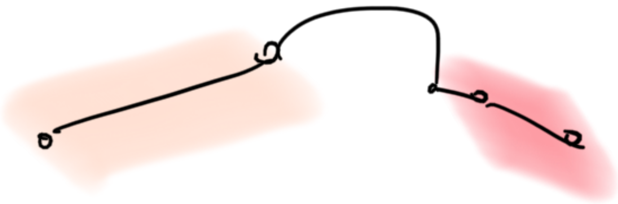
```
ctx . fillStyle = 'rgba(0, 0, 200, 0.5)';
ctx . fillRect (30, 30 50, 50);
```



strokeRect (x, y, width, ht) ← outline

clearRect (x, y, width, ht) ← clears an area, makes it transparent

Paths



create path
draw into path
stroke / fill the path, render it

```
ctx.beginPath();
ctx . moveTo (75, 50);
ctx . lineTo (100, 75);
```

starting path pos → moveTo

doesn't draw anything → moveTo

lineTo ← from current pos to

this point

ctx. lineTo (100, 25);

close path +

ctx. fill (); // ctx. stroke();

automatically closes path

closePath ← adds straight line to path going to start of current subpath

0

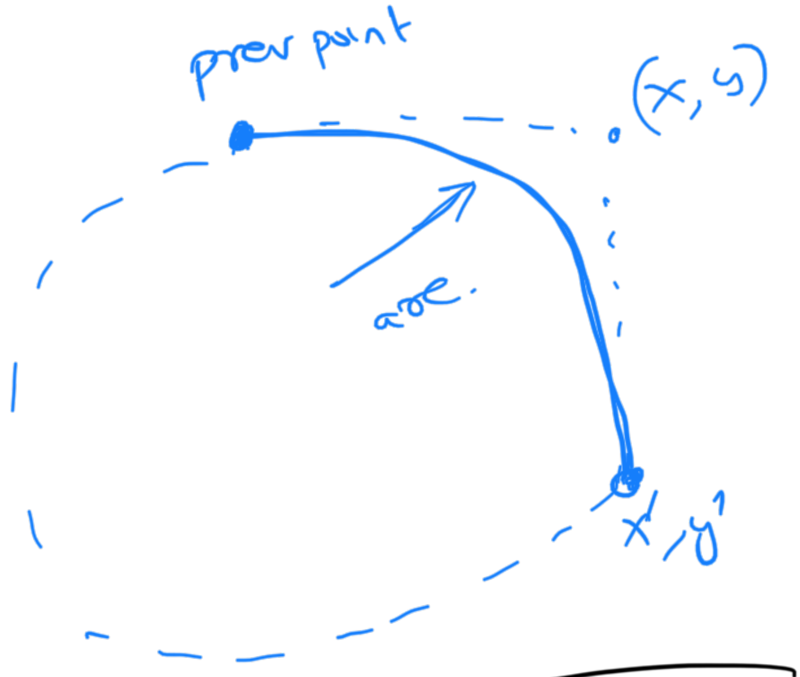
ctx. arc (x, y, radius, start Angle,
end Angle,
anticlockwise)

Bool

(eg)

Math. PI * 2

ctx. arcTo(x, y, x', y', radius)



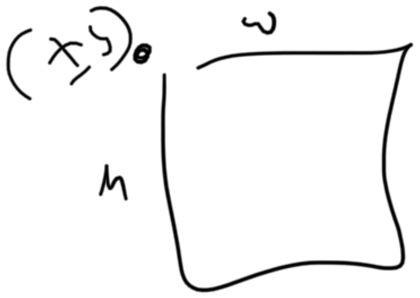
??

automatically moves to..

rect (x, y, width, height)

^

1



adds a
rectangular path
to an open path....

ctx. fillstyle = `rgba(r, g, b, a)';

0 → 1
transparent ↑ opaque

ctx. global Alpha = 0.2 ← globally alpha = 0.2

can also be

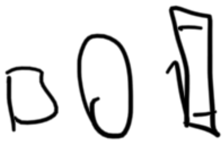
`rgba(255, 255, 255, ' + $\frac{i}{10}$ + ')';

↑ formatted string

line styles

lineWidth = value

lineCap = butt / round / square ...



line Join =