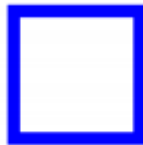


The background features abstract geometric shapes in shades of orange and yellow, primarily concentrated on the right side and bottom, creating a modern, layered effect. The text "More Canvas" is centered in a bold, orange, sans-serif font.

More Canvas

Line Width

```
let cx =  
document.getElementById("canvas");  
let ctx = cx.getContext("2d");  
ctx.strokeStyle = "blue";  
ctx.strokeRect(5 , 5 , 50 , 50);  
ctx.lineWidth = 5;  
ctx.strokeRect(135 , 5 , 50 , 50) ;
```



Text

- ▶ `fillText()`
 - ▶ Draws letters filled with current `fillStyle` color
- ▶ `strokeText()`
 - ▶ Draws outline of letters

Example

```
cx.font = "40px Georgia ";  
cx.fillStyle = "orange";  
cx.fillText ("I can draw text , too !" , 10 ,  
50) ;
```

```
cx.strokeStyle = "purple";  
cx.strokeText("Fancy text is fancy", 10, 100);
```

```
cx.fillStyle="yellow";  
cx.strokeStyle="blue";  
cx.fillText("EXTRA FANCY", 10, 150);  
cx.strokeText("EXTRA FANCY", 10, 150);
```

I can draw text , too !

Fancy text is fancy

EXTRA FANCY

Loops and Canvas

- ▶ Have you noticed yourself writing the same code over and over again when drawing on canvas?
- ▶ Why not use loops instead!

Example

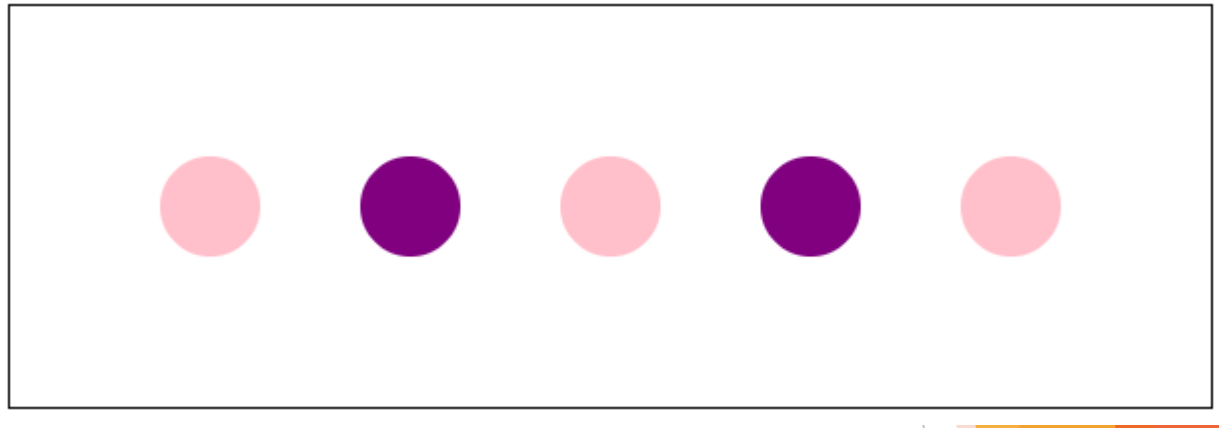
- ▶ What if I want to draw 5 purple balls on the canvas, and space them 100px apart?

```
var r = 25;  
var x = 100, y=100;  
cx.fillStyle = "purple";  
for(var i = 1; i <= 5; i++){  
    cx.beginPath();  
    cx.arc(x, y, r, 0, 2*Math.PI);  
    cx.fill();  
    x += 100;  
}
```



Example

```
var r = 25;  
var x = 100, y=100;  
  
for(var i = 1; i <= 5; i++){  
    if(i % 2 === 0)  
        cx.fillStyle = "purple";  
    else cx.fillStyle = "pink";  
  
    cx.beginPath();  
    cx.arc(x, y, r, 0, 2*Math.PI);  
    cx.fill();  
    x += 100;  
}
```



More Complex Example

```
var radius = 50;
cx.beginPath();
cx.arc(200, 200, radius, 0, 2*Math.PI);
cx.stroke();

cx.save();
cx.translate(200,200);

for(var i = 0; i < 8; i++){
    cx.beginPath();
    cx.arc(0,100, radius,0, 2*Math.PI);
    cx.stroke();
    cx.rotate(45*Math.PI/180);
}
cx.restore();
```

