

HTML Canvas and OO Javascript

This lecture is based on materials from:



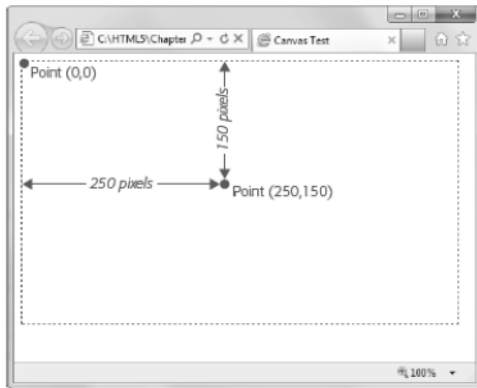
Eloquent JavaScript
A Modern Introduction to Programming
by Marijn Haverbeke



HTML5: The Missing Manual
The Book That Should Have Been in the Box
Matthew MacDonald
O'Reilly Media

Canvas

- A drawing surface



```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>Lines</title>

  <style>
    canvas {
      border: 1px dashed black;
    }
  </style>

  <script>

    window.onload = function() {
      var canvas = document.getElementById("drawingCanvas");
      var context = canvas.getContext("2d");

      context.strokeStyle = "blue";

      context.moveTo(250,50);
      context.lineTo(50,250);
      context.lineTo(450,250);
      context.closePath();

      // Draw the outline.
      context.stroke();
    };

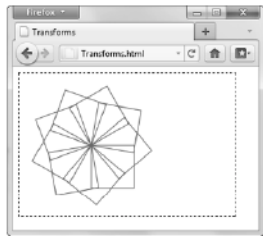
  </script>
</head>

<body>
  <canvas id="drawingCanvas" width="500" height="300"></canvas>
</body>
</html>
```



Transforms

- Shifts the canvas' coordinate system
- Functions:
 - `translate(x,y)`
 - `rotate(angle)`
 - `scale(w,h)`
 - `transform()`
 - `save()`
 - `restore()`





Drawing Images

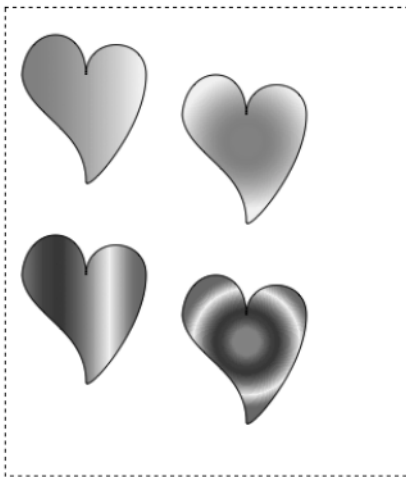
Step 1: Load image into `` tag

Step 2: Paint image on canvas

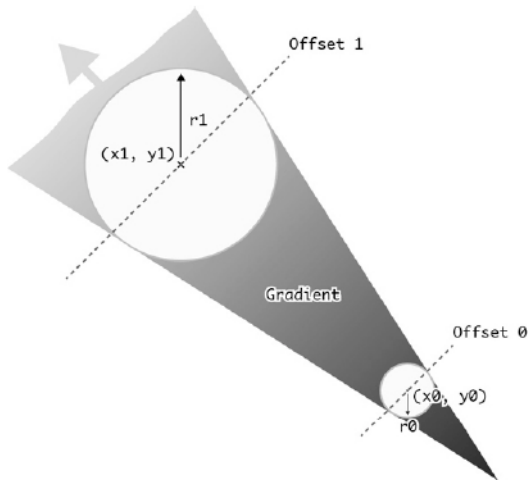
```
canvas.drawImage(img,x,y);
```

Caveat: step 2 should run only after
image has been loaded

Gradients and Bezier Curves



Radial Gradients





Arrays

```
var a = new Array();      // empty array  
var b = new Array("dog", 3, 8.4);  
var c = [2, 5, 'a', 'b'];
```

```
c[15] = "hello";  // implicit extension
```

Properties:

- length
- join()
- reverse()
- sort()
- concat()
- slice()
- splice()
- push() / pop()
- shift() / unshift()
- toString()



Objects

- Collection of properties and methods
- Creation
 - `var empty = {};`
 - `var cat = {color:"gray", name:"Spots"};`
- Adding properties/methods
 - `cat.age = 10;`
 - `cat.talk = function () { console.log("Cats don't talk!");}`
- Deleting properties/methods
 - `delete cat.age;`

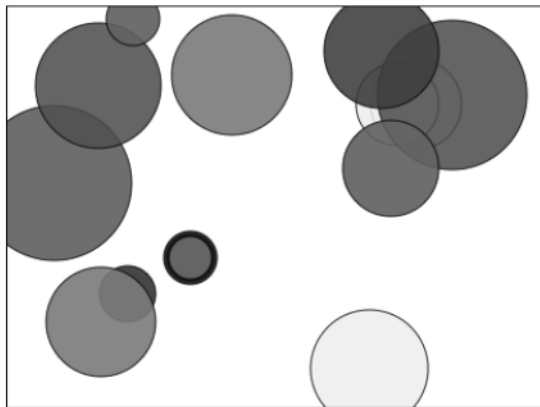


Object Constructors

- Any function can become an object constructor
- Instantiate an object by calling **new** on the function.

```
function Rectangle(w,h) {  
    this.width = w;  
    this.height = h;  
    this.area = function () {  
        return this.width * this.height;  
    };  
}  
  
var rec = new Rectangle(2,4);  
  
rec.area();  
rec["area"]();  
rec.newFunction = newPredeclaredFunc;
```

Interactive Circles



Add Circle

Clear Canvas



Method and Property Types

- Private variables
- Private functions
- Privileged methods
- Public properties
- Public methods
- Prototype properties
- Prototype methods
- Static properties



Prototype

- Javascript does not have a class construct
- Every object has a **prototype** property
- Use prototype to change type-wide definition



Privileged vs. Prototype

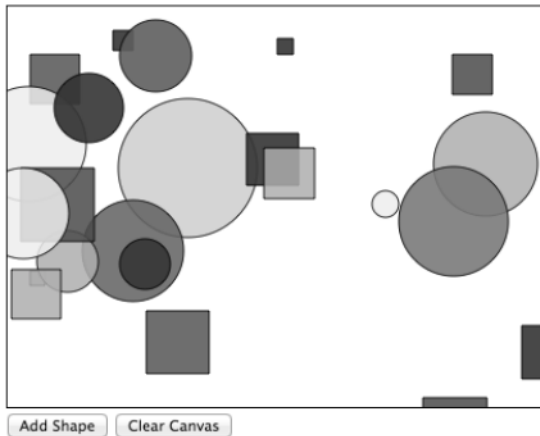
- Prototype methods
 - are static
 - one single instance shared among all definitions
- Privileged methods
 - recreated for each object
 - can access private data.



Inheritance

- Inherit by setting prototype to instance of parent object
- Calling parent constructor
 - `Parent.call(this, args....);`

Interactive Shapes





Saving the Picture in the Canvas

- Data URL
 - `var url = canvas.toDataURL("image/jpeg");`
 - Character-based representation of an image file
 - Can be assigned to an ``
- Binary pixel data
 - `var data = canvas.getImageData();`
- Store a list of steps

