



HTML5 and CSS3 – The Future of the Web Programming



AL-FARABI KAZAKH NATIONAL UNIVERSITY



HTML5 Canvas

Sergio Luján Mora





Content

- Canvas
- Canvas reference



CANVAS



Canvas

- The canvas element provides an API for two-dimensional drawing—lines, fills, images, text, and so on
- The canvas is only a container for graphics, a script (e.g., JavaScript) must be used to actually draw the graphics
- Every pixel in the canvas can be controlled



Canvas

- Canvas element:
`<canvas></canvas>`
- Attributes:
 - height
 - width



```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8" />
<title>Canvas example</title>
<script>
function draw()
{
  var ctx =
document.querySelector("canvas").getContext("2d");
  // First square
  ctx.fillRect(10, 10, 50, 50);
  // Second square
  ctx.fillRect(100, 100, 50, 50);
}

window.onload = draw;
</script>
<body>
<canvas></canvas>
</body>
</html>
```





Canvas

- Any text inside the `<canvas>` element will be displayed in browsers that does not support `<canvas>`

```
<canvas>
<p>
Your browser doesn't support &lt;canvas&gt; element.
Please, download and use one of the following browsers:
</p>
<ul>
<li>Google Chrome</li>
<li>Mozilla Firefox</li>
<li>Opera</li>
</ul>
</canvas>
```



HTML5 & CSS3



Canvas

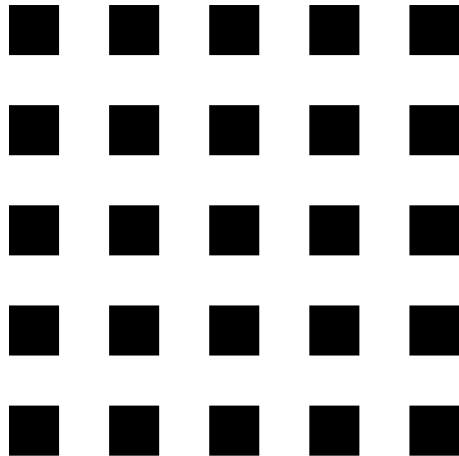
- `document.querySelector("canvas").getContext("2d");`
 - It gets the 2D context to allow you to draw
 - It provides methods to draw lines, boxes, circles, etc.
- `fillRect(x, y, width, height)`
 - Draws a filled rectangle using the color/style of the `fillStyle` attribute
 - The x and y coordinates start in the top left



Canvas

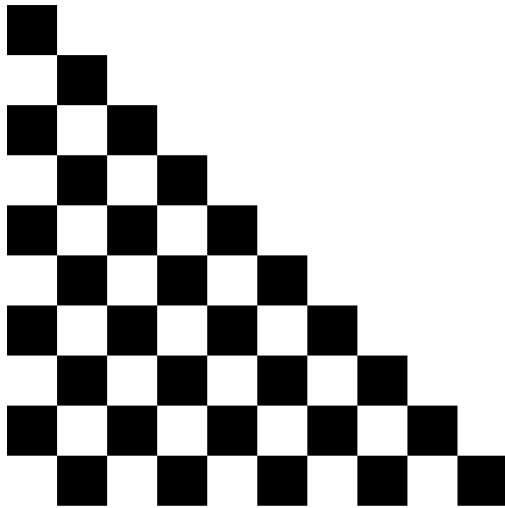
- Exercise:
 - Create the following pattern

HTML5 & CSS3



Canvas

- Exercise:
 - Create the following pattern



```
// Creates a solid square
ctx.fillStyle = "rgb(0, 255, 0)";
ctx.fillRect(10, 10, 50, 50);

// Draws an outline
ctx.strokeStyle = "rgb(0, 182, 0)";
ctx.lineWidth = 5;
ctx.strokeRect(9, 9, 52, 52);
```




```
var canvas = document.getElementById('example');  
  
var context = canvas.getContext('2d');  
  
context.fillStyle = "rgb(0,255,0)";  
context.fillRect (25, 25, 100, 100);  
  
context.fillStyle = "rgba(255,0,0, 0.6)";  
context.beginPath();  
context.arc(125,100,50,0,Math.PI*2,true);  
context.fill();  
  
context.fillStyle = "rgba(0,0,255,0.6)";  
context.beginPath();  
context.moveTo(125,100);  
context.lineTo(175,50);  
context.lineTo(225,150);  
context.fill();
```



```
var ctx =  
document.querySelector("canvas").getContext("2d");  
var img = document.createElement("img");  
  
// wait until the image has loaded  
img.onload = function () {  
    ctx.canvas.width = img.width;  
    ctx.canvas.height = img.height;  
    ctx.drawImage(img, 0, 0);  
};  
img.src = "sergio.jpg";
```



```
// Transformation
var pixels = ctx.getImageData(0, 0, img.width,
img.height);
for(var i = 0, n = pixels.data.length; i < n; i
+= 4) {
    // Red
    pixels.data[i+0] = 255 - pixels.data[i+0];
    // Green
    pixels.data[i+1] = 255 - pixels.data[i+2];
    // Blue
    pixels.data[i+2] = 255 - pixels.data[i+1];
}
ctx.putImageData(pixels, 0, 0);
```



```
// Transformation
var pixels = ctx.getImageData(0, 0, img.width,
img.height);
for(var i = 0, n = pixels.data.length; i < n; i
+= 4)
{
    total = (255 - pixels.data[i+0]
            + 255 - pixels.data[i+1]
            + 255 - pixels.data[i+2]) / 3;
    pixels.data[i+0] = total;
    pixels.data[i+1] = total;
    pixels.data[i+2] = total;
}
ctx.putImageData(pixels, 0, 0);
```



CANVAS REFERENCE



Canvas reference

- <http://simon.html5.org/dump/html5-canvas-cheat-sheet.html>





HTML5 & CSS3



Color and styles

- `fillStyle = color|style`
 - The fill-color of the drawing
- `strokeStyle = color|style`
 - The stroke-color of the drawing
- `lineWidth = number`
 - The width of the drawing stroke

HTML5 & CSS3



Color and styles

- `shadowColor = color`
 - The color of the shadow
- `shadowOffsetX = number`
 - The horizontal distance of the shadow
- `shadowOffsetY = number`
 - The vertical distance of the shadow
- `shadowBlur = number`
 - The size of the blurring effect



HTML5 & CSS3



Universitat d'Alacant
Universidad de Alicante



Path, curve, circle, and rectangle

- `fillRect(x, y, w, h)`
- `strokeRect(x, y, w, h)`
- `clearRect(x, y, w, h)`
- `rect(x, y, w, h)`
- `moveTo(x, y)`
- `lineTo(x, y)`

HTML5 & CSS3



Universitat d'Alacant
Universidad de Alicante



Path, curve, circle, and rectangle

- `arc(x, y, r, sAngle, eAngle, aClockwise)`
- `arcTo(x1, y1, x2, y2, radius)`