GRAPHICS

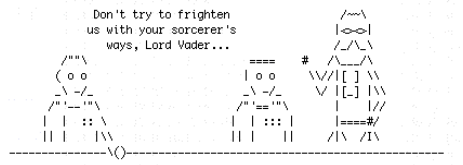
Drawing shapes with the <canvas> tag

Once upon a time, graphics were limited to external images

- jpgs, gifs, pngs

```
<img src="VaderAttacks.jpg" />
```

- There was no way to create dynamic or original graphics in HTML



But in swoops SVG and Canvas!

Which is better?

<svg>	<canvas>
• Vector • Elements are part of the DOM • Therefore, easy for events and slow for animations	• Raster • Only the canvas is part of the DOM • Therefore, hard for events and fast for animation

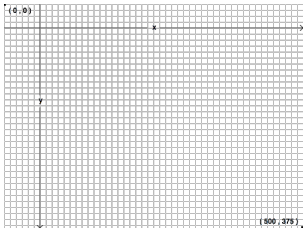
<canvas> joined us with HTML5

- Canvas allows drawing just about anything

```
<canvas id="cv"
width="500"
height="500" />
```

The canvas is laid out with coordinates

- The origin is in the upper-left
- Units are pixels



We need some JavaScript to draw on the canvas

```
function drawOnCanvas() {
  var canvas = document.getElementById("cv");
  var ctx = canvas.getContext("2d");
  ctx.fillRect(50, 25, 150, 100);
}
```

fillRect(x, y, width, height) draws a rectangle

- It uses the current fill style

fillStyle and strokeStyle tell us what kind of paint we're using when we draw

CSS color **Pattern** **Gradient**

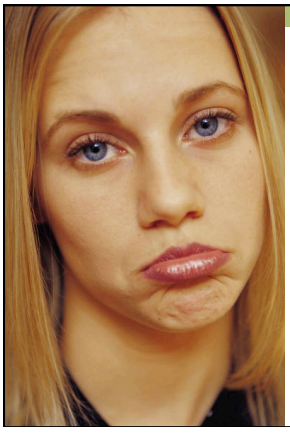
strokeRect(x, y, width, height) draws a rectangle with the current stroke style

- fillRect draws it and fills it. strokeRect only draws the lines

We add text to our drawings with fillText()

- context.fillText(textstring, x, y);

```
context.font = "bold 12px arial";
context.fillText("Liz", 200, 40);
context.fillText("Lemon", 200, 60);
```



CSS styling is not available to the text on canvases

- We can set a font, color and size
- But we cannot do it with styles
- No:
  - padding
  - margins
  - wrapping

font can be anything you would put in a CSS font rule

font style
font variant
font weight
font size
line height
font family

A *path* is a series of strokes

- To make a path, you ...
1. beginPath();
2. // Draw lines, circles, curves, whatever
3. stroke();
4. closePath();
- This groups your strokes

moveTo(x,y) says "Pick the brush up off the canvas and re-place it elsewhere"

- Merely moves the pen

```
ctx.moveTo(100, 10);
```

lineTo(x,y) draws a line from your current position to this new position.

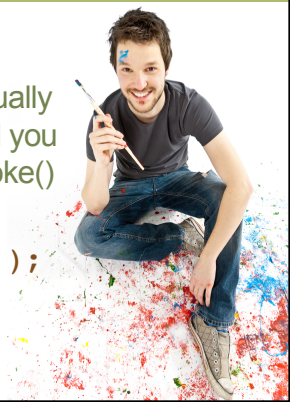
- Draws a line

```
ctx.lineTo(200, 10);
ctx.lineTo(200, 110);
```

- Basically says "Keep the brush down and move directly to this new position."

But nothing actually gets drawn until you tell it to with stroke()

```
ctx.stroke();
```



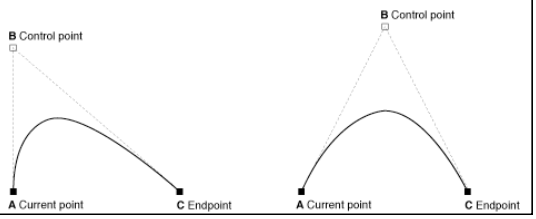
You can draw arcs & circles

ctx.arc(x, y, radius, startAngle, endAngle, direction);

- Where:
  - x, y: the center of the circle
  - startAngle, endAngle: The angle in radians that the arc starts and ends. 0=straight left
  - direction: true=counterclockwise & false=clockwise

There are some other ways to draw curves

- ctx.quadraticCurveTo(cpx,cpy,x,y);
- ctx.bezierCurveTo(cp1x,cp1y,cp2x,cp2y,x,y);
- ctx.arcTo(x1,y1,x2,y2,radius);



Sometimes, though, we just need to put an image in our canvas

drawImage() allows us to include an existing jpg, gif, or png

- Draw image starting at (x, y) and scale it to (w, h)

```
ctx.drawImage(theImage, x, y, w, h);
```

- Example:

```
var vader = new Image();
vader.addEventListener("load",function () {
  ctx.drawImage(vader, x, y);
});
vader.src = "images/vaderAttacks.jpg";
```

tl;dr

- HTML5 canvas suddenly makes it possible to create drawings, games and other RIA without Flash, Silverlight or other plug-ins
- The low-level access to the canvas's context allows you to draw just about anything
- We can include text and images in addition to rectangles, lines, and arcs