



FULL SAIL
UNIVERSITY

HTML5

Canvas & More

**Web Interaction
& Animation**

Web Interaction & Animation

Web Design & Development Bachelor of Science Degree



Understanding HTML5 Canvas

Review of Canvas

- Canvas element provides web pages with a drawing surface for all kinds of graphics - shapes, images, text, etc..
- 2D drawing platform within the browser
- Uses nothing more than JavaScript and HTML - No Plugins
- Extensible through a JavaScript API
- Created by Apple for dashboard widgets, but now openly developed as a W3C Spec
- Think of it as a piece of paper



Understanding HTML5 Canvas

Immediate Mode

Canvas is an **IMMEDIATE MODE** bitmapped area of browser screen that can be manipulated by you through JavaScript.

- Canvas completely redraws bitmapped screen on every frame using Canvas API
- It's “on-the-fly” once its drawn, its there.
- Flash, Silverlight, SVG use retained mode



Understanding HTML5 Canvas

Multiple?

Pages can have more than one Canvas, when they do they can overlap.

- One layer could be the background
- Next layer could be characters
- So on and so on....



FULL SAIL
UNIVERSITY

Understanding HTML5 Canvas

What's it good for?

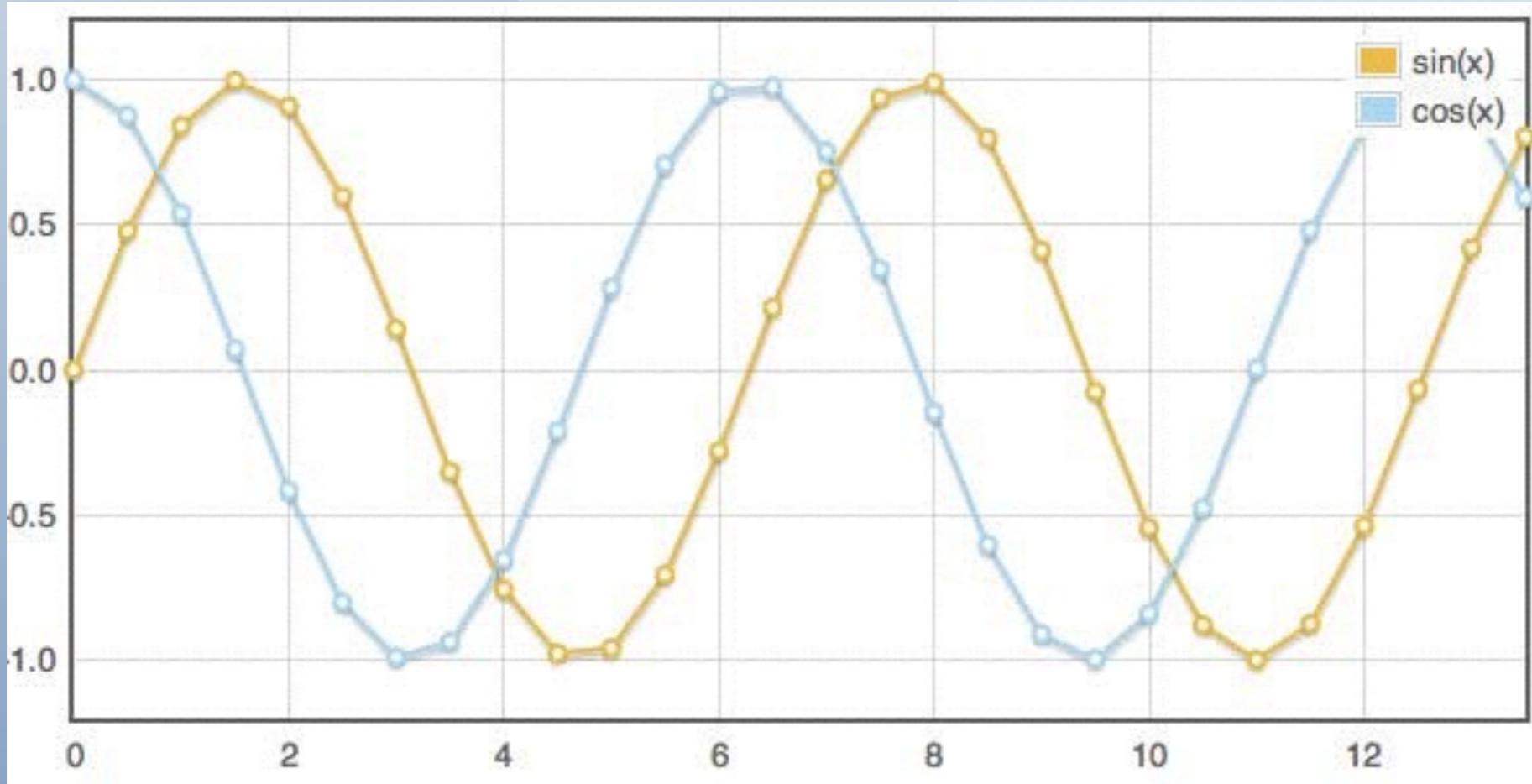
Canvas is great for dynamic visual media. Such as....



Understanding HTML5 Canvas

What is it used for?

- Data Visualization





FULL SAIL
UNIVERSITY

Understanding HTML5 Canvas

What is it used for?

- Animated Graphics



Web Interaction & Animation

Web Design & Development Bachelor of Science Degree

**Web Interaction
& Animation**

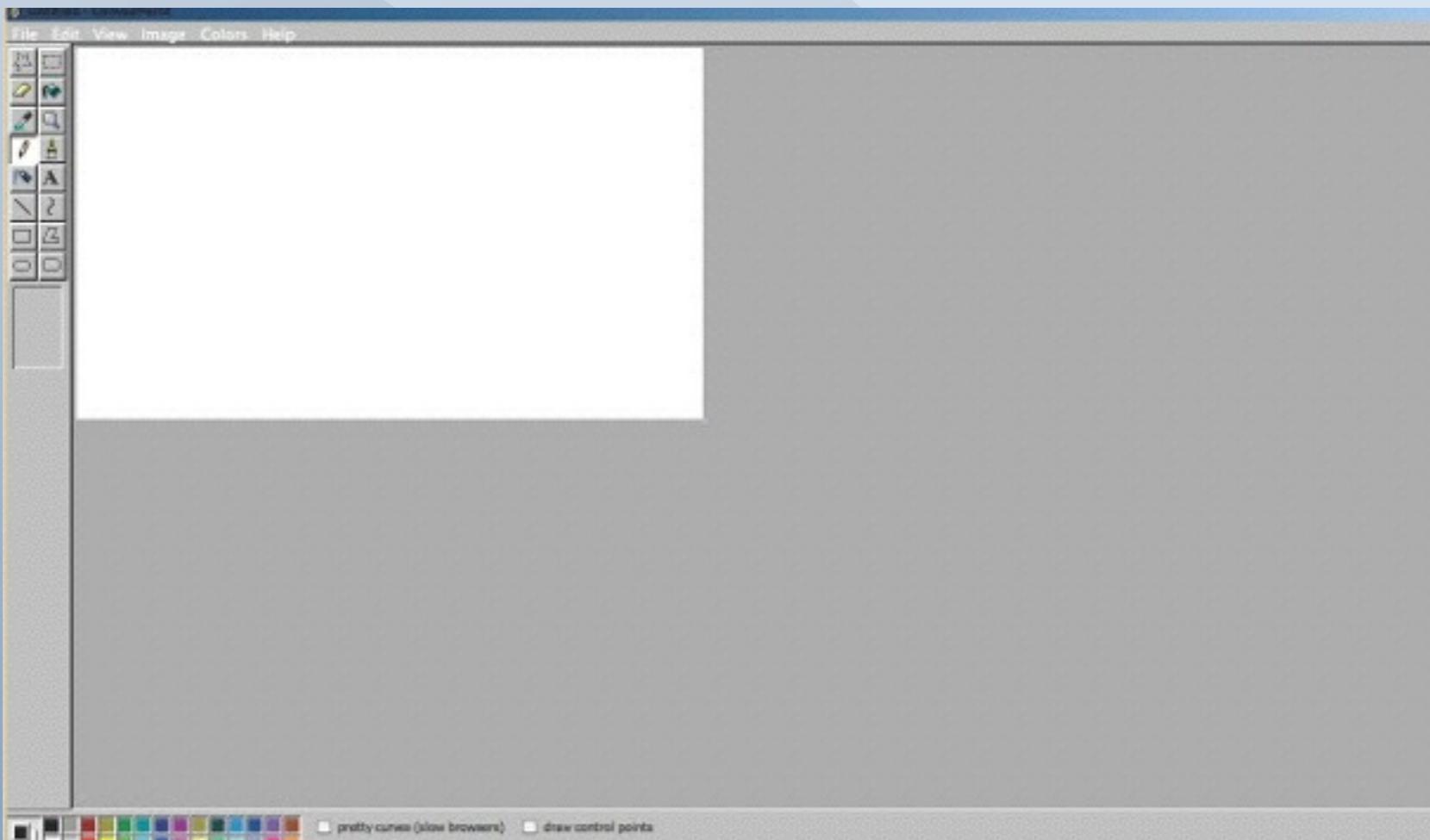


FULL SAIL
UNIVERSITY

Understanding HTML5 Canvas

What is it used for?

- Web Applications



Web Interaction & Animation

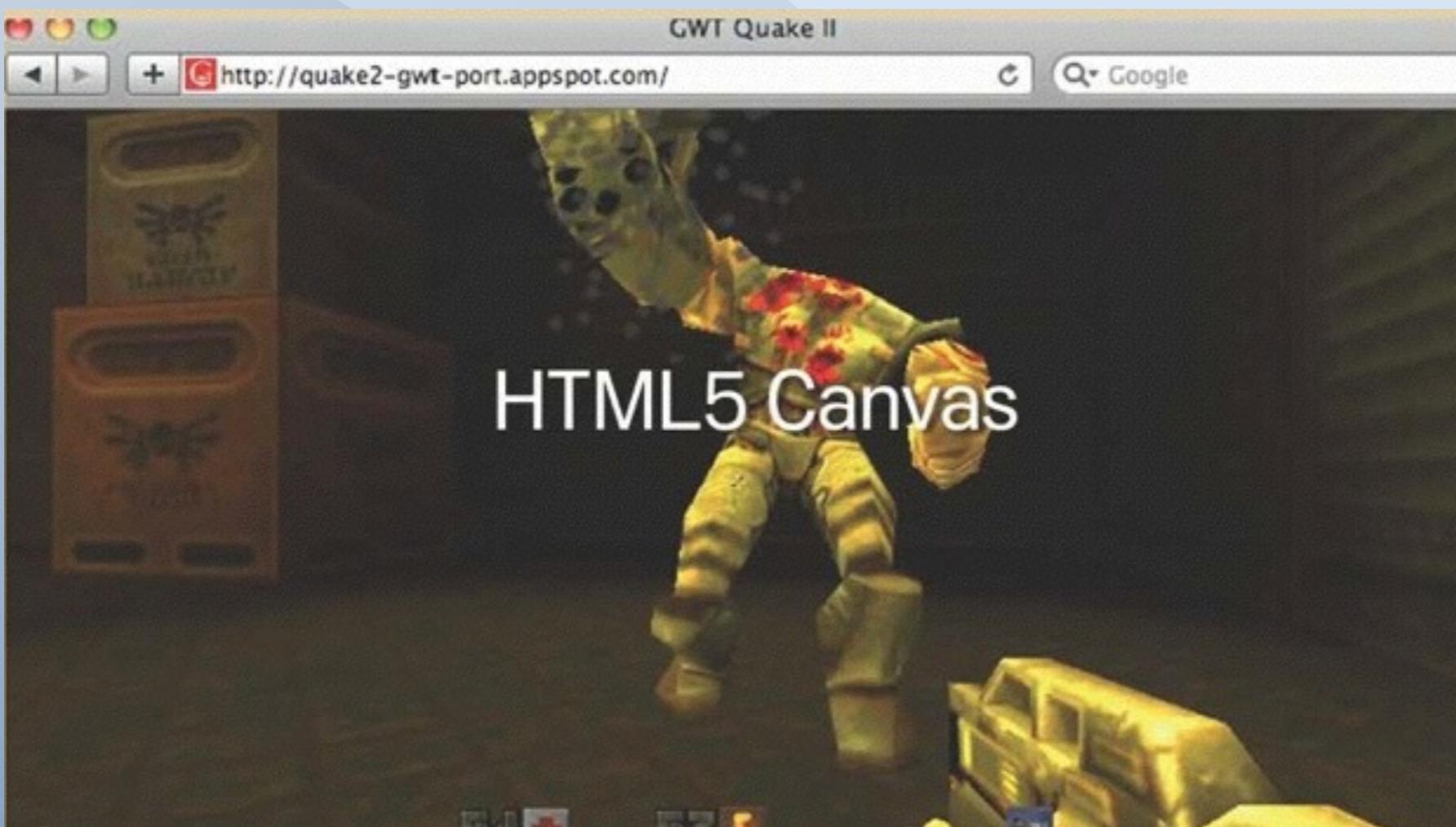
Web Interaction & Animation
Web Design & Development Bachelor of Science Degree



Understanding HTML5 Canvas

What is it used for?

- Games





Understanding HTML5 Canvas

Browser Support

- Most modern browsers support it
 - Safari
 - Chrome
 - Firefox
 - Opera
- Internet Explorer 9+



Understanding HTML5 Canvas

Browser Support

	Supported
Chrome	3.0+
Safari	3.0+
Firefox	3.0+
Opera	10.0+
IE*	9+
iOS	1.0+
Android	1.0

- Basic Canvas Support is in 70% of browsers
- <http://caniuse.com/>



Understanding HTML5 Canvas

Internet Explorer

- IE 8 and below is missing support for canvas
- Ployfills can be used for VML in IE7 and IE8
- Explorercanvas.js



Understanding HTML5 Canvas

Internet Explorer

- Everything is not supported!
- Limitations:
 - Only Linear Gradients
 - Patterns must repeat in both directions
 - Clipping paths are not supported
 - Non-Uniform scaling does not scale outlines correctly
 - Really bad performance!



Bitmap vs Vector

Graphic Types

- Canvas is a Bitmap system
 - Everything is drawn as a single flat picture
 - Changes require the whole picture to be redrawn
- SVG is a Vector System
 - Scalable Vector Graphics
 - Elements to be drawn are separate DOM objects
 - They can be manipulated individually



FULL SAIL
UNIVERSITY

Bitmap vs Vector

What should we use?

So which should we use?

BOTH!

It depends on what you are doing!



Canvas

What is it good for?

● Canvas - Low Level

- Faster at drawing graphics
- Save images generated by canvas
- Everything in canvas is a pixel
- No Mouse Interaction - By Default (No mouse API)
- High Animation
- JS Centric
- More “Book keeping”



SVG

What is it good for?

● SVG - High Level

- Resolution independent so it scales
- Easier to target other elements
- Import/Export
- Easy UIs
- Interactive
- Medium Animation
- Tree of Objects



Understanding HTML5 Canvas

Canvas Tag

Created using the new HTML5 tag

```
<canvas width="800" height="600">  
</canvas>
```

Height & Width can be set here.
Can be programmatically altered later.

Canvas elements start out invisible
(Transparent black background)



Canvas Properties & Methods

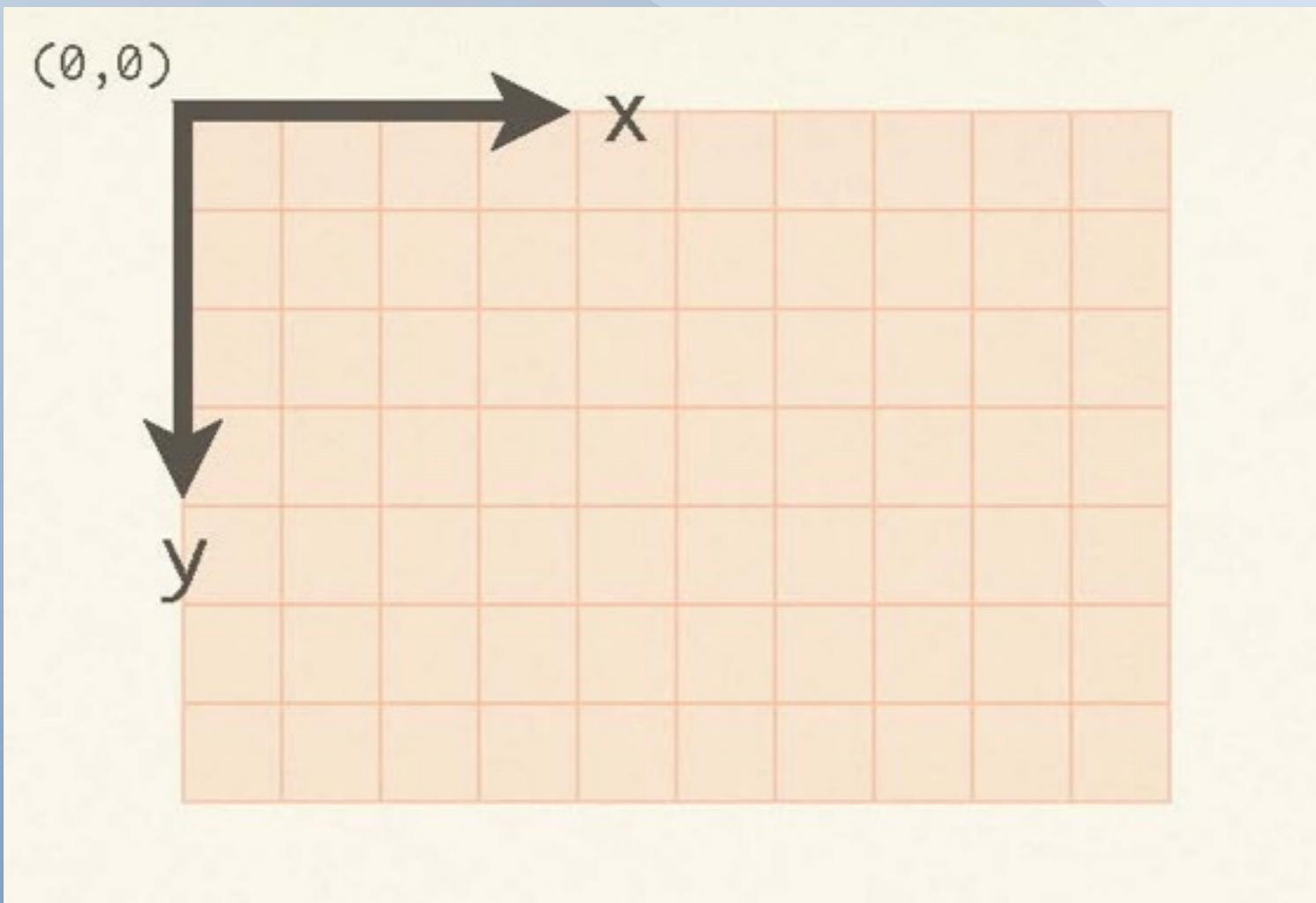
Attribute/Function	Description
width	The width of the canvas in pixels (defaults to 300)
height	The height of the canvas in pixels (defaults to 150)
toDataURL(type)	Convert the contents of the canvas to a static image. The type parameter indicates the image type (image/png)
getContext(ctxID)	Get the drawing context for the canvas



Understanding HTML5 Canvas

Coordinate System

Canvas uses the standard screen-based coordinate system



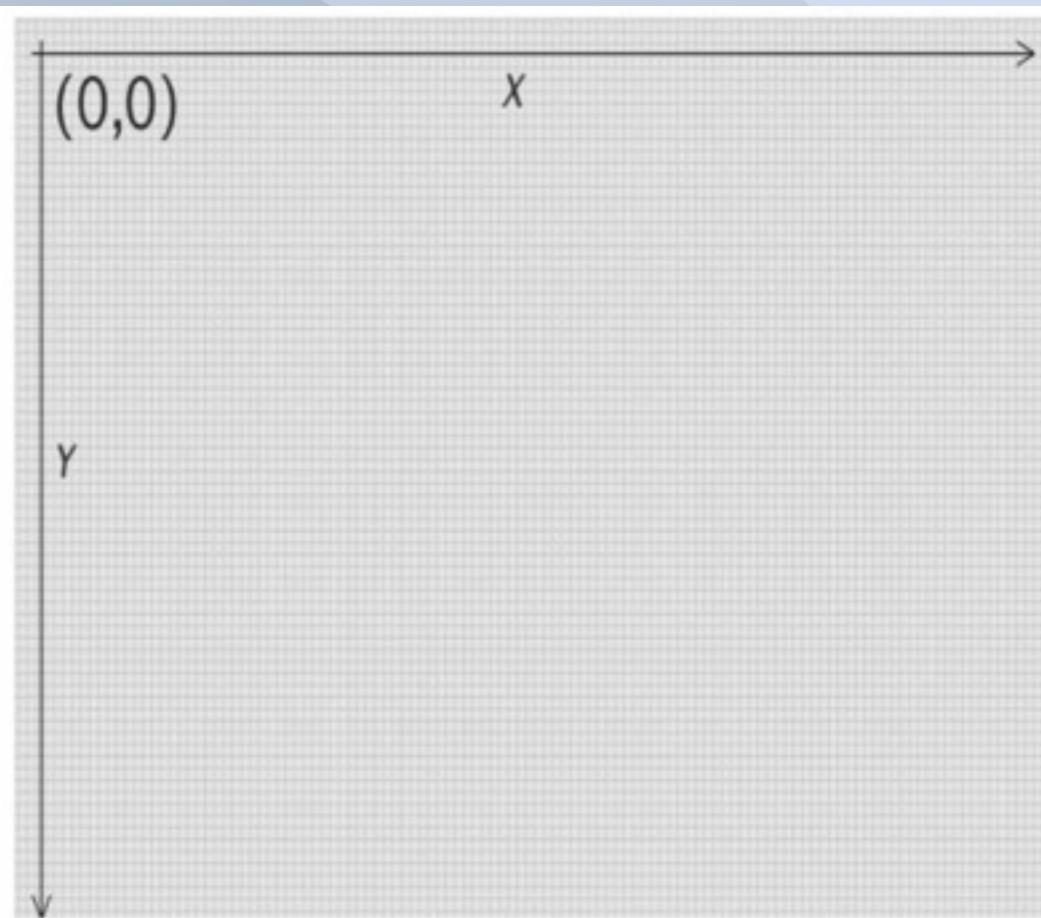


FULL SAIL
UNIVERSITY

Understanding HTML5 Canvas

Getting Started

```
<canvas id="myCanvas" width="350"  
height="640" > </canvas>
```



Web Interaction & Animation
Web Design & Development Bachelor of Science Degree

Web Interaction & Animation



FULL SAIL
UNIVERSITY

What if it's not supported?

Fallback

```
<canvas width="800" height="600">  
    Fallback Content  
</canvas>
```



FULL SAIL
UNIVERSITY

Fall Back

```
<canvas id="myCanvas" width="350"  
height="640" >  
  
    
  
</canvas>
```

Web Interaction & Animation

Web Design & Development Bachelor of Science Degree

**Web Interaction
& Animation**



FULL SAIL
UNIVERSITY

Canvas Basics

Basic Canvas Example

Web Interaction & Animation
Web Design & Development Bachelor of Science Degree

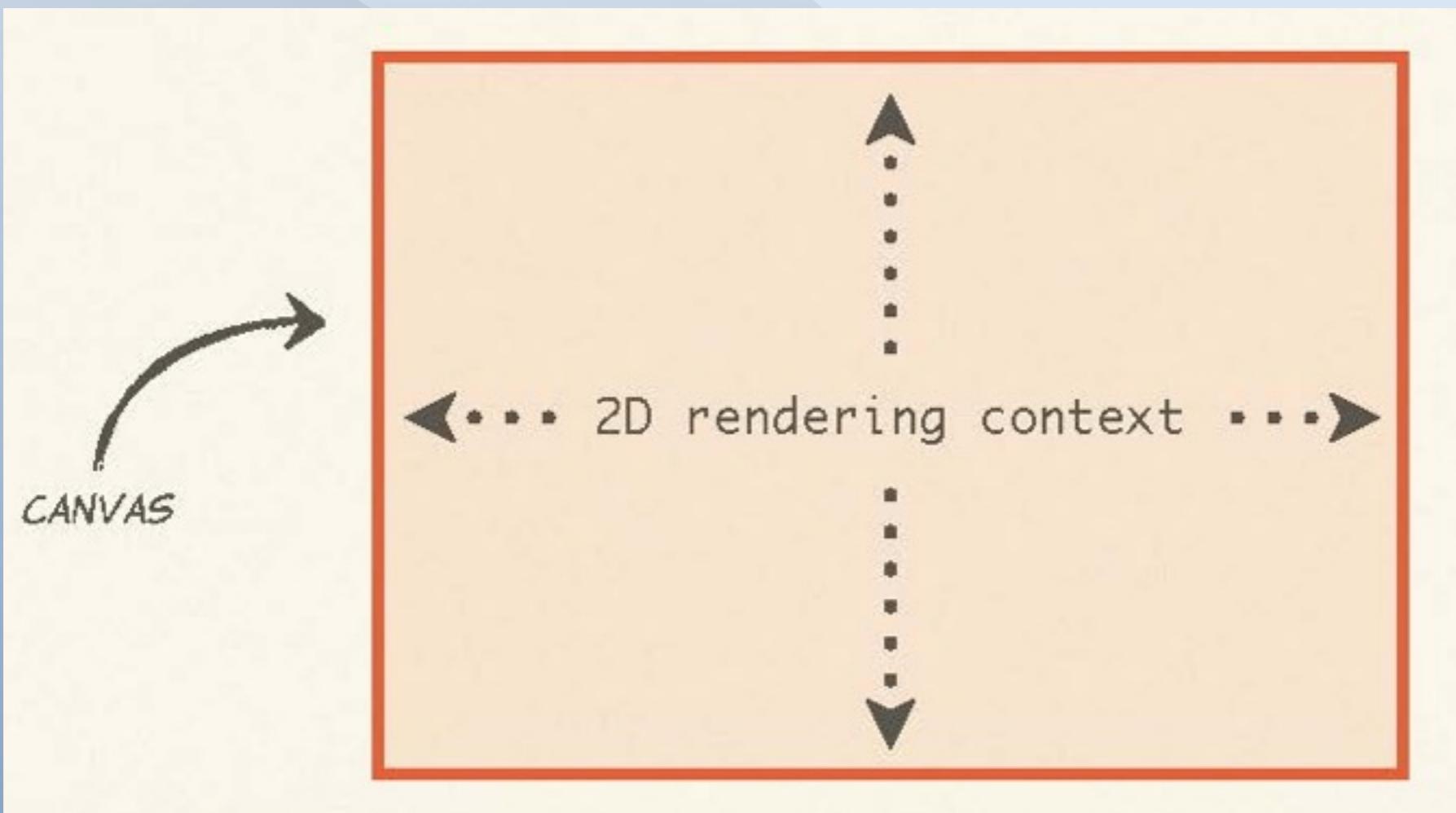
Web Interaction
& Animation



Understanding HTML5 Canvas

2D Context

Everything is drawn onto the 2D rendering context (ctx)





Understanding HTML5 Canvas

2D Context

The Canvas API includes a 2D context allowing you to draw shapes, render text, and display images onto the defined area of browser screen.

- The 2D context is a display API used to render the Canvas graphics
- The JavaScript applied to this API allows for keyboard and mouse inputs, timer intervals, events, objects, classes, sounds... etc



Canvas Properties & Methods

Canvas Tag

- `getContext()` method retrieves the drawing context for the canvas
 - Context contains info about the canvas and provides all of the drawing methods for that particular context
 - More on this in a little bit
- `toDataURL()` method converts the contents of the image to a `data://URL` of the given image type. `image/png` support is mandatory, but others can be supported.
Such as `image/jpg` or `image/gif`



Drawing On The Canvas

Requires a series of steps

Retrieve a reference to the canvas element

```
var cElem = document.getElementById("canvas");
```

Get the drawing context from the element using getContext()

```
var ctx = cElem.getContext("2d");
```



Drawing On The Canvas

Code

```
<canvas id= "canvas1" width = "400" height="300">
    Fallback Content
</canvas>

function drawOnCanvas () {

    var cElem = document.getElementById("canvas");
    var ctx = cElem.getContext ("2d");
    if (ctx !=null) {
        //go ahead and draw
    }
}
```



Set Your Styles

Styles...not just for fashion

Utilizing Style functions:

◆ **fillStyle**

- Style to use when filling - Default is black

◆ **strokeStyle**

- Style to use on strokes - Default is black

◆ **lineWidth**

- Width of the pen when drawing lines

`fillStyle()` and `strokeStyle()` define the style of shapes to be drawn on the canvas. Can use CSS Color, Gradient, or Pattern.



FULL SAIL
UNIVERSITY

Set Your Styles

Styles...not just for fashion

Two step process to draw with colors and styles

1. Set the fill & stroke styles and the line width
2. Call a drawing operation that creates a shape



Step 1 - Set the styles

Styles

```
ctx.fillStyle = 'rgb(255, 0, 0)';  
ctx.strokeStyle = 'rgba(0, 255, 0, 0.5);
```

Use **rgba** for alpha transparency

Instead of RGB, you can also use:

- String containing CSS colors
- CanvasGradient - will cover later
- CanvasPattern - will cover later



Step 2 - Draw a Simple Object

Start with the basics

Rectangles - the only primitive shape supported by canvas

- ◆ FillRect(posX, posY, width, height);
 - Draws a rectangle using the current fill style
- ◆ StrokeRect(posX, posY, width, height);
 - Draws a rectangle outline using current stroke style
- ◆ ClearRect(posX, posY, width, height);
 - Clears all the pixels within the given rectangle

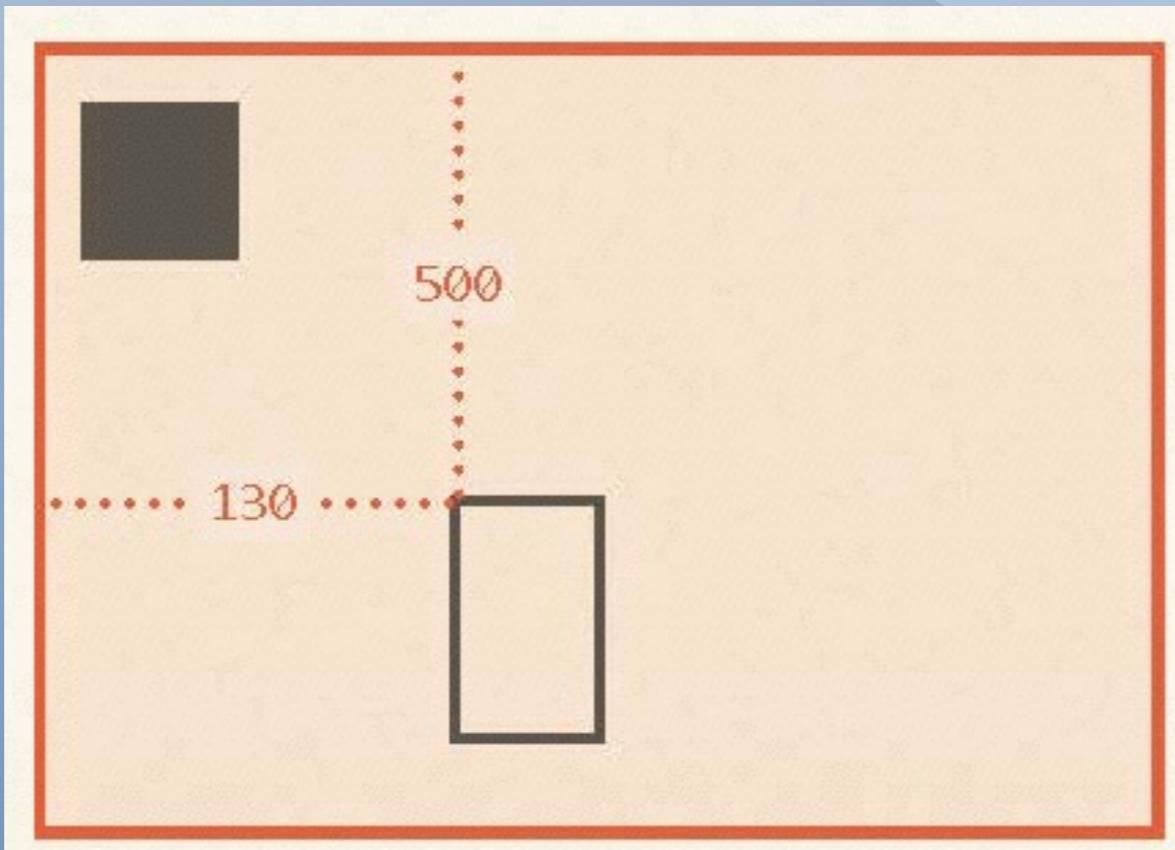


Understanding HTML5 Canvas

Code

```
ctx.fillStyle = 'rgb(65, 60, 50)';  
ctx.fillRect(25, 50, 100, 100);
```

```
ctx.strokeStyle = 'rgba(65, 60, 50, 1)';  
ctx.strokeRect(130, 500, 40, 70);
```





FULL SAIL
UNIVERSITY

Drawing Rectangles

Code

Example Time

Web Interaction & Animation


Web Interaction & Animation

Web Design & Development Bachelor of Science Degree



Lines

Methods

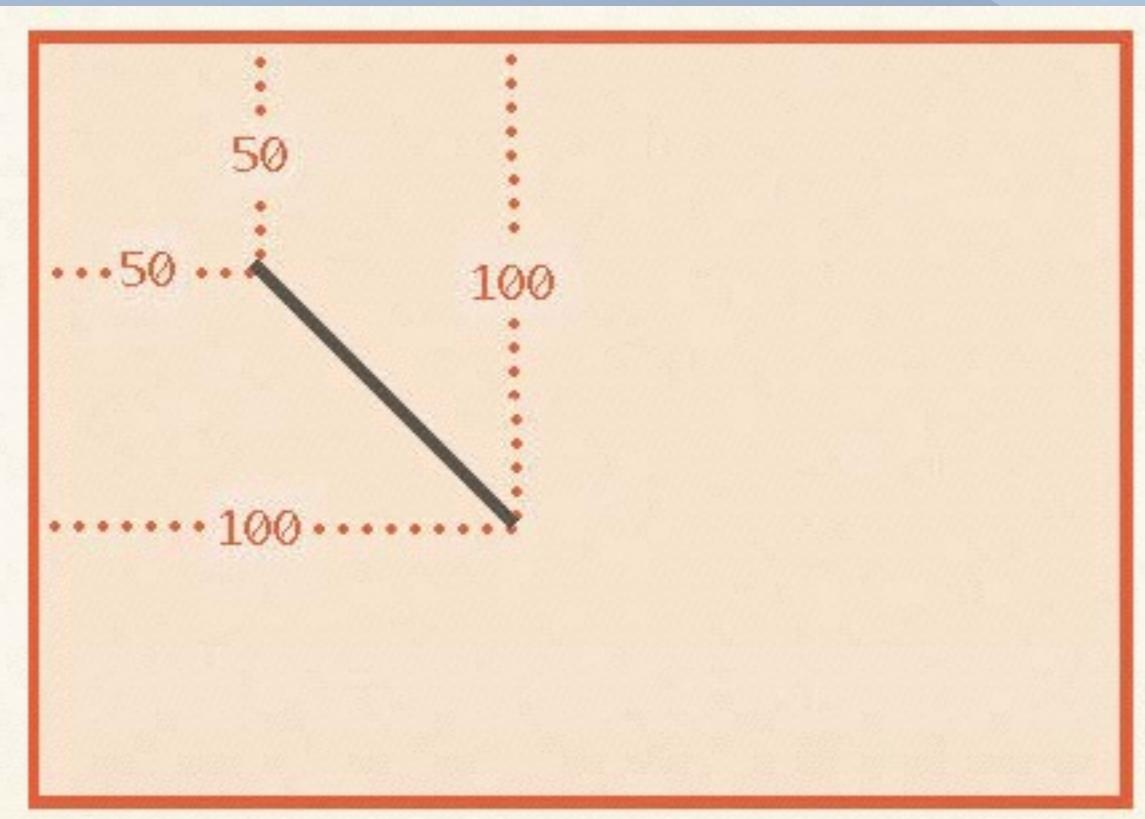
Attribute/Function	Description
<code>moveTo(x, y)</code>	Moves the pen to the given coordinates, does not draw
<code>lineTo(x, y)</code>	Draws a line from the current pen position to the new point
<code>lineWidth</code>	Determines the pixel width that lines will be drawn in
<code>lineCap</code>	How line endings are drawn: butt (default), round, square
<code>lineJoin</code>	How lines join together: round, bevel, miter (default)
<code>miterLimit</code>	The limit at which line joins are cut off and drawn as bevels (10)
<code>beginPath()</code>	Begins a new set of path-drawing operations
<code>stroke()</code>	Collects all of the current path commands and draws them



Line Drawing

Code

```
ctx.strokeStyle = 'rgb(65, 60, 50)';  
ctx.beginPath();  
ctx.moveTo(50, 50);  
ctx.lineTo(100, 100);  
ctx.stroke();
```





FULL SAIL
UNIVERSITY

Lines

Line Code Examples

Let's take a look at an example

Web Interaction & Animation

Web Design & Development Bachelor of Science Degree

**Web Interaction
& Animation**



FULL SAIL
UNIVERSITY

Double-click to edit

Web Interaction & Animation
A decorative graphic element consisting of a series of small, red, circular dots arranged in a heart shape.

Web Interaction & Animation

Web Design & Development Bachelor of Science Degree



Canvas State

Save() and Restore()

- ◆ Drawing on the Canvas makes use of a stack of drawing “states”
- ◆ A state stores Canvas data of elements drawn
 - Transformations and clipping regions use data stored in states



Canvas State

Save & Restore

Save()

- Pushes the current state to the stack

Restore()

- Restores the last state saved from the stack to the Canvas

Note: current paths and current bitmaps are not part of saved states



Canvas State

What is it good for?

- **Your code may have set up a whole bunch of complex drawing settings that you don't want to have to manually keep track of**
- Let's look at an example



Complex Shapes & Paths

Path Drawing

Paths can be used to draw virtually any shape on the Canvas.

- ◆ Paths are simply lists of points for lines to be drawn in-between
- ◆ Each subpath is one or more points connected by straight or curved lines
- ◆ Rendering context always has one current path
- ◆ A new path should be created for each individual shape
- ◆ Can be open or closed. Closed ends at the starting points

Simple shapes are drawn without effecting the current path



Complex Shapes & Paths

Path Drawing Process

- ◆ Begin creating a path with `beginPath()` method
- ◆ Add paths using a drawing routine like: lines, arcs, curves
- ◆ Stroke the current path with `stroke()`
- ◆ Fill the current path with `fill()`
- ◆ Close the current path with `closePath()`



FULL SAIL
UNIVERSITY

Complex Shapes & Paths

Path Code Examples

Let's take a look at an example

Web Interaction & Animation
Web Design & Development Bachelor of Science Degree

**Web Interaction
& Animation**



Arcs

No Joans here

- ◆ Arcs are curves that are portions of a circle
- ◆ Circles are a full 360 degree arc

Path Operation	Description
<code>arc(x, y, r, sA, eA, aC)</code>	Adds an arc to the current path that starts at x,y and has a radius of r, with a starting angle of sA, and an ending angle of eA. The aC argument is true if the arc is anti-clockwise.
<code>arcTo(x1,y1,x2,y2,r)</code>	Adds an arc to the current path that starts at the current pen position, has the given control points, and a radius of r.
<code>closePath()</code>	Closes the current drawing path.

Note - Angles are in radians, not degrees!

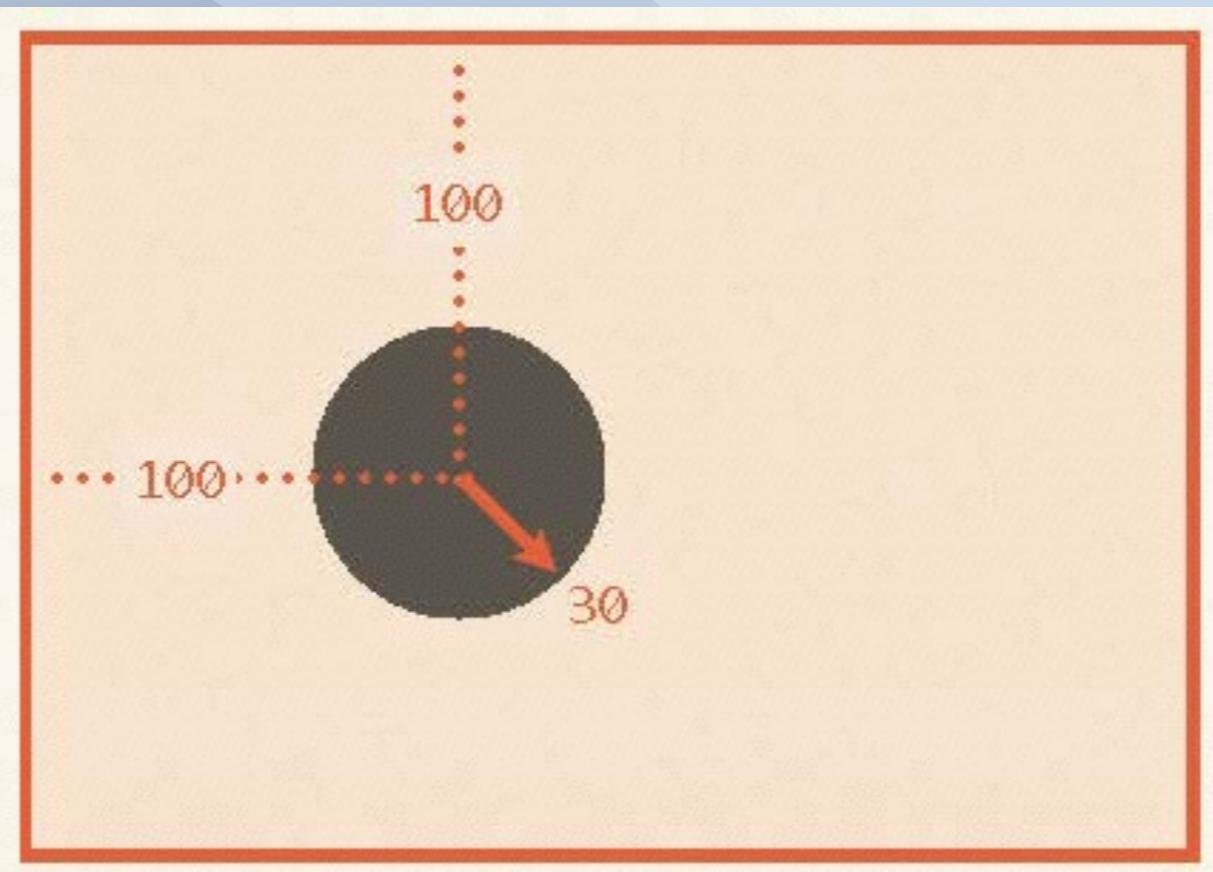
To Convert you can use:
 $(\text{Math.PI} / 180) * \text{degrees}$



Arc Path Drawing

Arc Code

```
ctx.fillStyle = 'rgb(65, 60, 50)';  
ctx.beginPath();  
ctx.arc(100,100,30,0, (Math.PI/180)*360,  
true);  
ctx.fill();
```



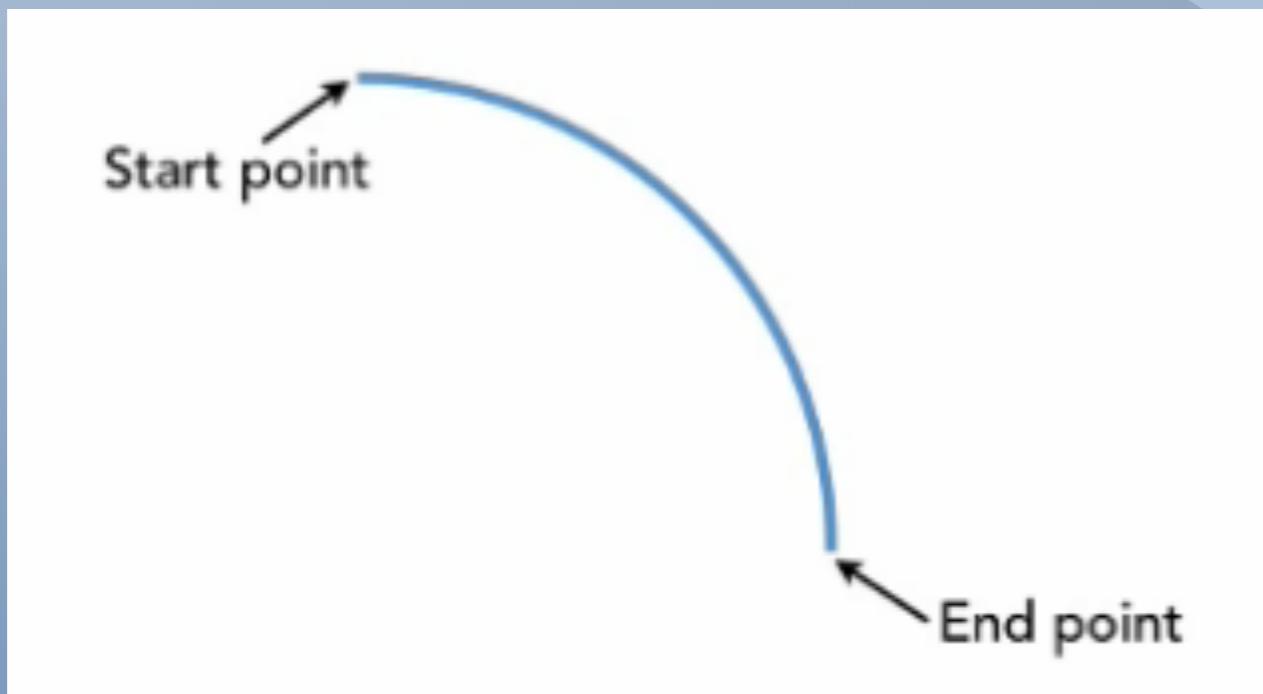


Arc Path Drawing

Arc Code

360 degree = 2 * Math.PI Radians

```
//Stroke a 90 degree arc, clockwise
ctx.beginPath();
ctx.arc(50,150,100,1.5*Math.PI,
       2*Math.PI,true);
ctx.stroke();
```



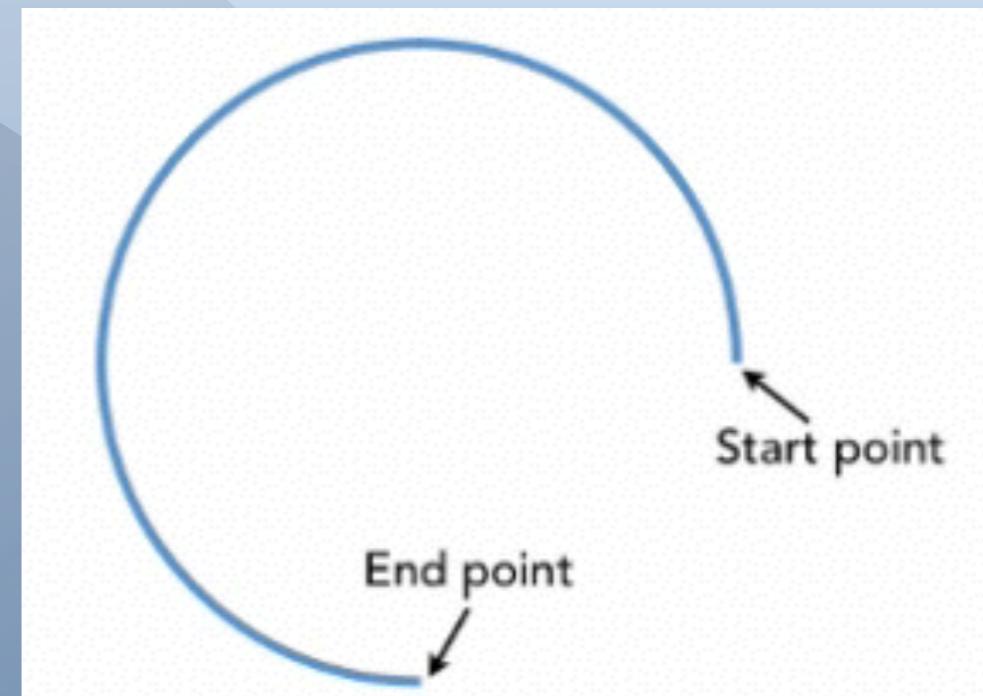


Arc Path Drawing

Arc Code

360 degree = 2 * Math.PI Radians

```
//Stroke a 270 degree arc, counter-clockwise
ctx.beginPath();
ctx.arc(300,150,100,0,1.5*Math.PI,
       false);
ctx.stroke();
```





FULL SAIL
UNIVERSITY

Simple Shapes & Paths

Arc Code Examples

Let's take a look at an example

Web Interaction & Animation

Web Design & Development Bachelor of Science Degree

**Web Interaction
& Animation**

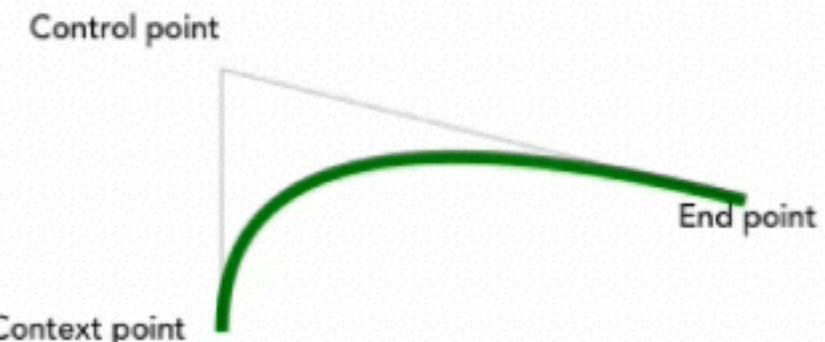


FULL SAIL
UNIVERSITY

Complex Curves

Quadratic Curves

Quadratic Curve



Quadratic curves are drawn using a start point, one control point and an end point

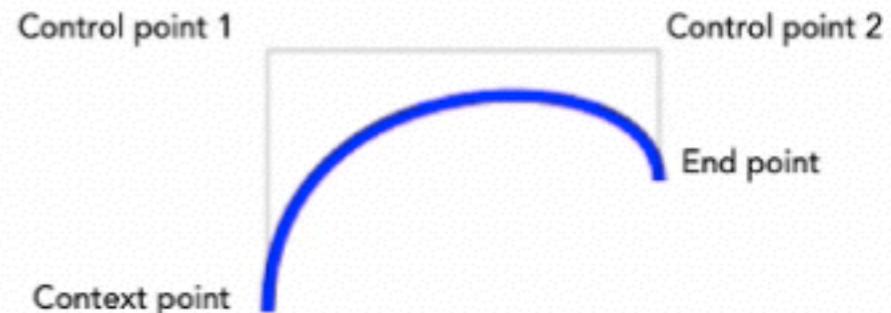
- ◆ `quadraticCurveTo(cx, cy, x, y)`
- ◆ Draw a quadratic curve starting at the current pen position using the given control point cx, cy and ending at the end point defined by x,y
- ◆ Context point - Where your pen is currently at. Use `moveTo()`



Complex Curves

Bezier Curves

Bézier Curve



Bezier curves are drawn from a context point to an end point using 2 control points to determine the curve

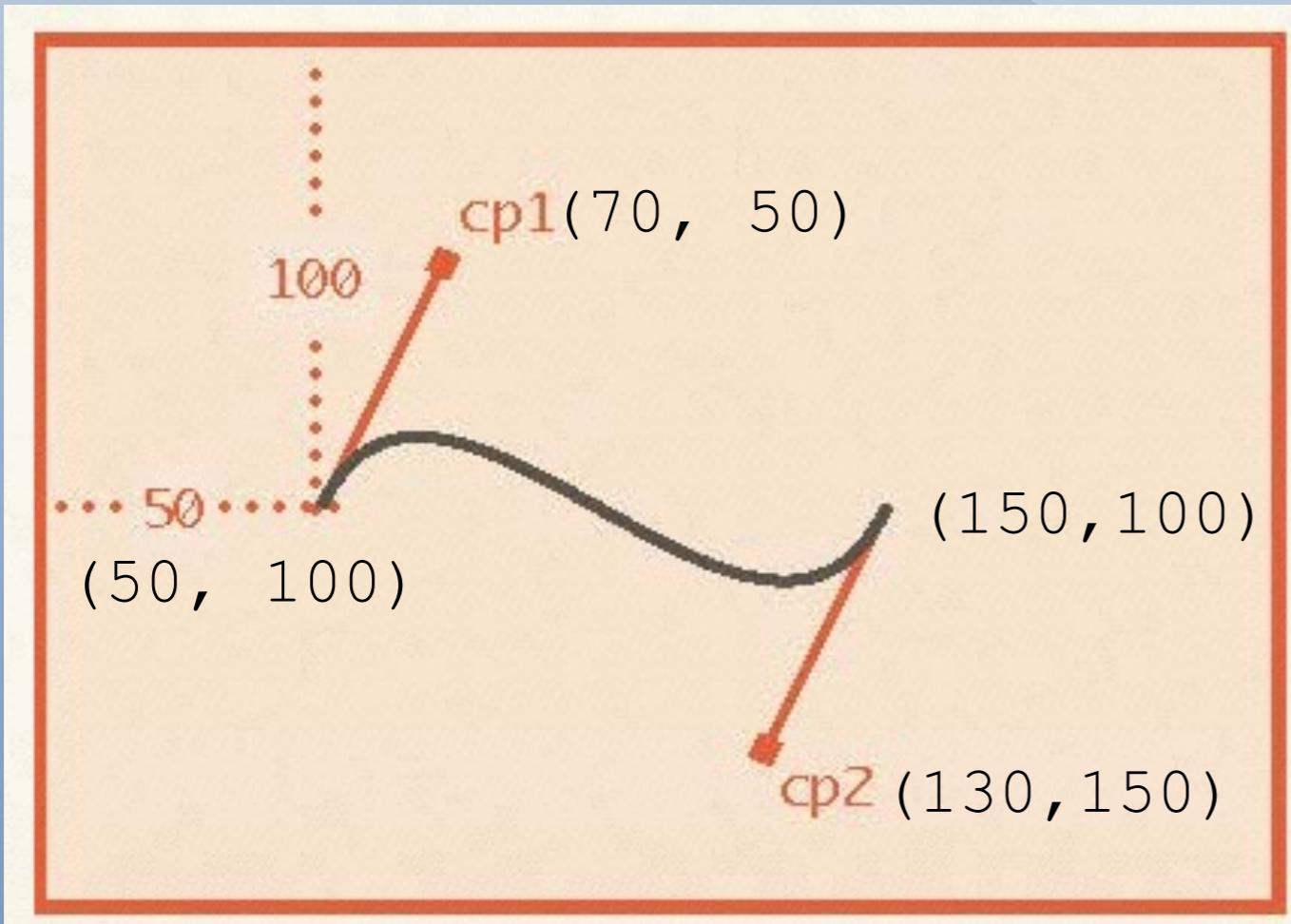
- ◆ 2 control points means we can make these curves even more complex
- ◆ `bezierCurveTo(cx, cy1, cx2, cy2, end1, end2)`
- ◆ Draw a Bezier curve starting at the current pen position using the two control points defined by `cx1, cy1` and `cx2, cy2` and ending at the point `end1, end2`



Complex Curves

Bezier Curves

```
ctx.strokeStyle = 'rgb(65, 60, 50)';  
ctx.beginPath();  
ctx.moveTo(50, 100);  
ctx.bezierCurveTo(70, 50, 130, 150, 150, 100);  
ctx.stroke();
```





FULL SAIL
UNIVERSITY

Complex Curves

Code For Curves

Example
Time

Web Interaction & Animation
Web Design & Development Bachelor of Science Degree

Web Interaction & Animation



Rendering Text

On the Canvas

Text:

- ◆ Text can be stroked or filled using the same `fillStyle` and `strokeStyle` as paths
- ◆ Text drawn on the canvas are paths...not actual text
- ◆ Not affected by any CSS Rules
- ◆ Not a replacement for regular document text
 - Because accessibility it won't work.



Rendering Text

Methods

Text Operation	Description
font	Font setting to use. Anything you would normally put into a font CSS rule: family, size, weight, variant, etc. (defaults to 10px sans-serif)
textAlign	"start" (default), "end", "left", "right", "center"
textBaseline	"top", "hanging", "middle", "alphabetic" (default), "ideographic", "bottom"
fillText(txt, x, y, [maxW])*	Render the text string at x, y no wider than maxW
strokeText(txt, x, y, [maxW])*	Render the text string at x, y no wider than maxW
measureText(text)	Returns the dimension metrics of the string using the current font settings

~ Ideographic and Hanging - used more in Chinese or Korean Languages

* [maxW] is optional



FULL SAIL
UNIVERSITY

Rendering Text

Examples

Example
Time

Web Interaction & Animation


Web Interaction & Animation

Web Design & Development Bachelor of Science Degree