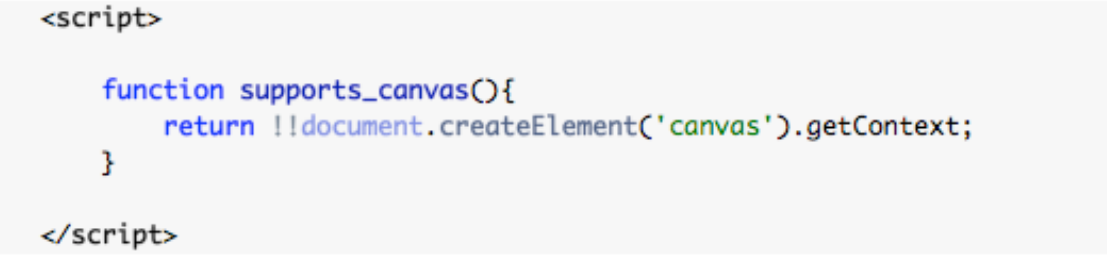
Working with the HTML5 Canvas API

HTML5 defines the < `canvas` > element as *a resolution-dependent bitmap canvas which can be used for rendering graphics, game graphics, or other visual images on the fly*.

Developing with the `canvas`:

1. Check for the `canvas` support in your browser. You can use `Modernizr` or else create a test canvas element. Get output using the javascript function below to test whether the canvas works in firefox, IE, and chrome.

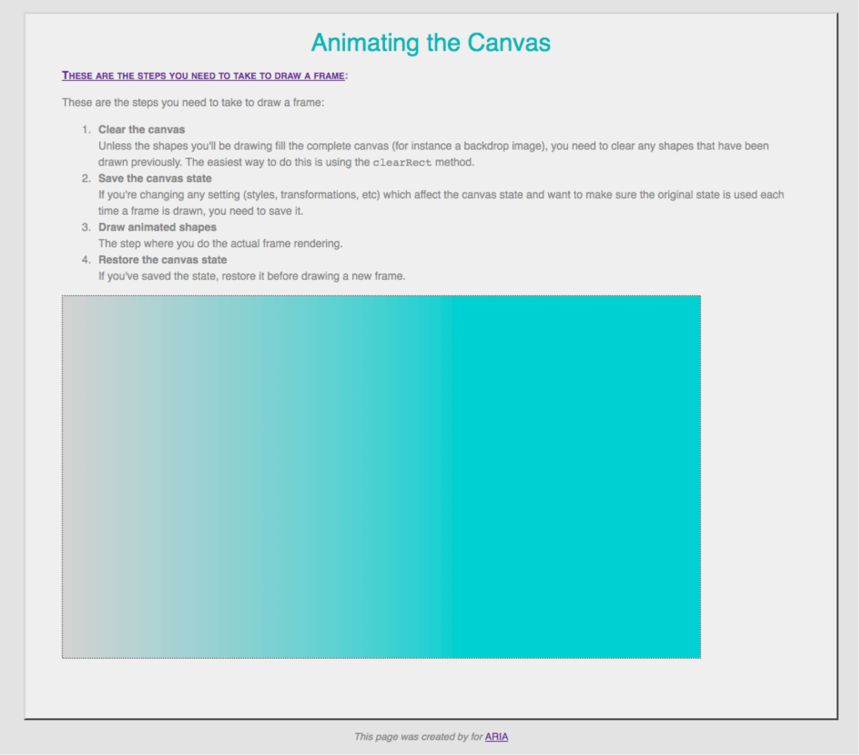


```
<script>

    function supports_canvas(){
        return !!document.createElement('canvas').getContext;
    }

</script>
```

2. Once you have tested whether or not your browser supports the canvas you should draw a **gradient box to** fill the entire canvas. Create a JavaScript function to draw this gradient. You can download canvas_starter.html from moodle to get you started – this includes a blank canvas and a JavaScript shim for requesting new frames.



Animating the Canvas

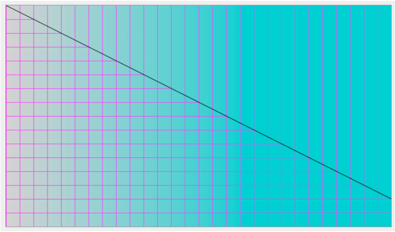
THESE ARE THE STEPS YOU NEED TO TAKE TO DRAW A FRAME:

These are the steps you need to take to draw a frame:

1. **Clear the canvas**
   Unless the shapes you'll be drawing fill the complete canvas (for instance a backdrop image), you need to clear any shapes that have been drawn previously. The easiest way to do this is using the `clearRect` method.
2. **Save the canvas state**
   If you're changing any setting (styles, transformations, etc) which affect the canvas state and want to make sure the original state is used each time a frame is drawn, you need to save it.
3. **Draw animated shapes**
   The step where you do the actual frame rendering.
4. **Restore the canvas state**
   If you've saved the state, restore it before drawing a new frame.

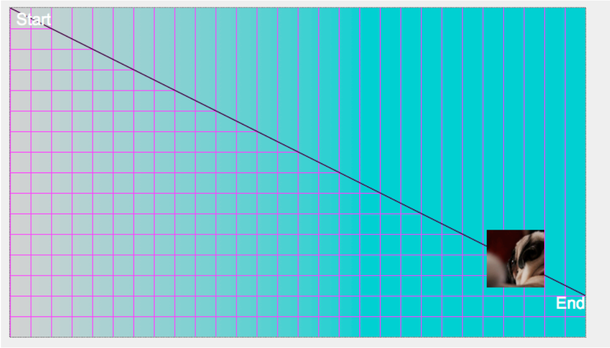
This page was created by for ARIA

3. Draw a **grid** on to top of the gradient. Change the color so that you can see the lines. Later in this tutorial we are going to animate an image from the top left corner down to the bottom right corner so the grid should reflect this.



4. Put in some text to describe the start and stop points on your canvas.

5. You can draw **images** into a canvas using a html img element. When the image is loaded it is drawn onto the canvas and then the source (src) specified to display the image. You can use any image that you want – scale it so that it is quite small.



6. You have all the elements for your animation above. A gradient box, a grid that is drawn with lines, some text that points to the start and end, and finally an image. The next step is to animate the canvas. These are the steps you need to take to draw a frame:

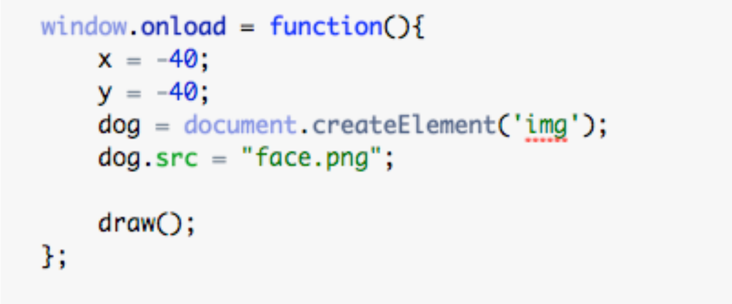
    a. **Clear the canvas**
    Unless the shapes you'll be drawing fill the complete canvas (for instance a backdrop image), you need to clear any shapes that have been drawn previously. The easiest way to do this is using the clearRect method.

    b. **Save the canvas state**
    If you're changing any setting (styles, transformations, etc) which affect the canvas state and want to make sure the original state is used each time a frame is drawn, you need to save it.

    c. **Draw animated shapes**
    The step where you do the actual frame rendering.

    d. **Restore the canvas state**
    If you've saved the state, restore it before drawing a new frame.

7. We are going to animate our image – move it along the path from the start to the end. The draw function will be used to clear and redraw the canvas. It can take a lot of time to redraw complex frames and this means that performance can vary.

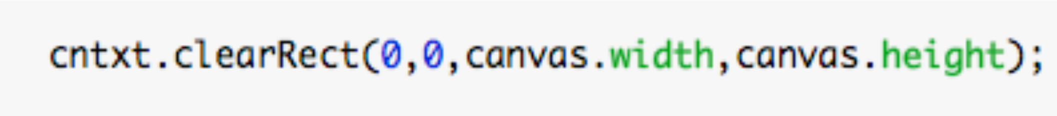


```
window.onload = function(){
    x = -40;
    y = -40;
    dog = document.createElement('img');
    dog.src = "face.png";

    draw();
};
```

In this function, the x and y values relate to the x and y co-ords of the image. The dog variable sets up our image. Finally, the animation draw function is called.

8. The draw function needs to perform a number of roles in this case – it should first clear the canvas.

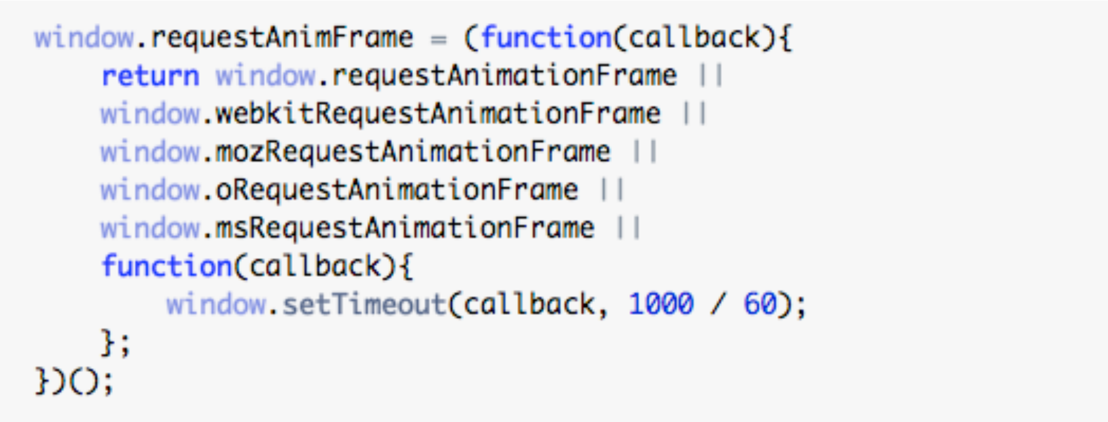


```
cntxt.clearRect(0,0,canvas.width,canvas.height);
```

9. You then need to redraw each of the component parts – this means drawing your gradient, lines and text first.

10. Next you must redraw the image. However, when you draw the dog you should change the x and y co-ordinates so that it travels down the line from start to finish.

11. When you initialize your canvas functions use the **`requestAnimFrame`** method to call your draw function. You can alternatively use, `setInterval` calls a function or evaluates an expression at specified intervals (in milliseconds) however this is not recommended. `setInterval` calls functions that are scheduled regardless of whether the browser tab they belong to is showing. **`requestAnimFrame`** is a JavaScript shim that deals with this issue by only calling functions when the tab is visible. The requestAnimFrame shim enables the browser to determine the optimal FPS for our animation, and for each animation frame we can update the elements on the canvas, clear the canvas, redraw the canvas, and then request another animation frame.



```
window.requestAnimFrame = (function(callback){
    return window.requestAnimationFrame ||
    window.webkitRequestAnimationFrame ||
    window.mozRequestAnimationFrame ||
    window.oRequestAnimationFrame ||
    window.msRequestAnimationFrame ||
    function(callback){
        window.setTimeout(callback, 1000 / 60);
    };
})();
```

12. Stop the image when it reaches the bottom of the page.

13. Accelerate the speed at which the image travels – the further it goes the faster it should get. To do this you will need to create a quadratic motion animation – check out http://www.html5canvastutorials.com/advanced/html5-canvas-quadratic-motion-animation/ for some tips.

Further References:

Mozilla Developer Network Canvas tutorial:
https://developer.mozilla.org/en/Canvas_tutorial

Let's Call it a Draw(ing survace) – Dive into HTML5:
http://diveintohtml5.info/canvas.html#further-reading

Canvas Demos – Applications, games, tools and tutorials for the HTML5 canvas element http://www.canvasdemos.com/

HTML5 Canvas Tutorials http://www.html5canvastutorials.com/

HTML5Rocks – Improving HTML5 Canvas Performance
http://www.html5rocks.com/en/tutorials/canvas/performance

HTML Canvas 2D Context http://dev.w3.org/html5/2dcontext/

Creating Vector Masks using the HTML5 Canvas
http://thinkvitamin.com/code/html5/create-vector-masks-using-the-html5-canvas/

Build your First Game with HTML5 http://net.tutsplus.com/tutorials/html-css-techniques/build-your-first-game-with-html5/