**How Does It Work?**

First, when the page loads, the code grabs the ‘canvas object ‘and attaches functions that will handle several JavaScript events for different mouse actions which are

* onMouseDown - is linked to the startDrawing() function. It sets the isDrawing variable, creates a new path, and then moves to the starting position to get ready to draw something:
* onMouseUp
* onMouseOut
* onMouseMove.

See below for detailed outline of mouse function:

var canvas;  
 var context;   
  
window.onload = function() {  
  
 // Get the canvas and the drawing context.   
 canvas = document.getElementById("drawingCanvas");  
 context = canvas.getContext("2d");   
  
// Attach the events that you need for drawing.   
canvas.onmousedown = startDrawing;   
canvas.onmouseup = stopDrawing;   
canvas.onmouseout = stopDrawing;   
canvas.onmousemove = draw; };

These events control the drawing process. At the same time, the page also stores the canvas in a global variable named **canvas**, and the drawing context in another global variable named **context**.

This is done so that the objects will now be easily available to the rest of the code to access

**Saving Function**

This is done by handing the user data URL over to an <img> element which provides a screenshot of the user’s input, then allowing the user to right-click the screenshot and save in their preferred destination ( see below for detailed outline)

Function saveCanvas () {  
// Find the <img> element.  
var imagecopy = document.getElementById(“savedImageCopy”);

//Show the canvas data in the image.  
imageCopy.src = canvas.toDataURL();

//unhide the <div> that holds the <img> , so the picture is now visible  
var imageContainer = documentGetElementById(“savedCopyContainer”);

imageContainer.style.display = “block”; }

**What Did I Use?**

* JavaScript with JQuery
* HTML
* CSS3