PART I

a) First define the canvas element in the HTML portion, then use the canvas methods in JavaScript to draw and fill in individual parts of the element.

To do this we must assign a JavaScript variable to the HTML element, then assign another variable for the context of that variable using the .getContext method.

b) <canvas id="myCanvas" width="300" height="500" style="border: 2px solid black"></canvas>

c) This is so that we can easily identify which canvas we want to draw to in the JavaScript portion of the webpage

d) context.beginPath();

e) context.stroke();

f) strokeRect() will draw the border of the rectangle using whatever colour you assign it to using the strokeStyle method, and fillRect() will fill it with whatever colour you assigned using the fillStyle method.

g) lineTo(), quadraticCurveTo()

h) fillRect() will fill a rectangle, with given inputs for width and height, with a colour. fillText() will also fill a rectangle, but with words instead of a colour.

i) It is easier to use the arc() method because the arcTo() method is used more to create arcs between two lines. The arc() method takes 5 parameters with one optional counterclockwise parameter. The first two indicate the x/y coordinates of the centre of the circle, the third indicates the radius of the circle to be drawn. The fourth indicates the starting angle in radians, where 0 is at the 3 o'clock position, and the fifth argument is the ending angle. If we want to draw a full circle, we can use starting angle of 0 and ending angle of 2\*Math.PI (2PI).

j) It should flip the context horizontally by 100%.

k) context.transform()

l) It will take the original context and move the starting position to x+50, y+50.

m) When the image loads, run the function()

n) context.drawImage(img,x,y) will just grab the image file, and place it on the given x/y coordinates in the canvas element.

context.drawImage(img,x,y,width,height) will grab the image, place it on the inputted x/y coordinates in the canvas element, and also set its width and height.

context.drawImage(img,sx,sy,swidth,sheight,x,y,width,height) will take the image, clip it starting from sx/sy inside the image, with a size of swidth/sheight, and place it on the canvas element at x/y, with width/height for size.

o) A sprite sheet is an image file that contains lots of smaller graphics inside it. By using this method, we can save memory space while loading an animation, and also have easier access to all these different graphics by only having to open one image file.

p) context.stroke() will draw the shape, and context.fill() will fill the area that context encompasses with colour.

PART II

a) This code draws a blue upward facing arrow. (Why did we begin the triangle with lineTo(0,50) instead of moveTo(0,50)?)

b) This code draws a sort of diamond with a round top.

PART III

a) Change context.arc to the following: context.arc(xPos,yPos,radius,0,2\*Math.PI)