JavaScript

HTML5 Canvas

Coding Exercise Challenge

Question 1: Basic Canvas Setup

Q: How do you create a canvas element in HTML and access it in JavaScript for drawing?

HTML:

```
<canvas id="myCanvas" width="200" height="200"></canvas>
```

JavaScript:

```
const canvas = document.getElementById('myCanvas');
```

const ctx = canvas.getContext('2d');

Explanation:

First, a <canvas> element is defined in HTML with a specific id. In JavaScript, getElementById() is used to access this canvas. The getContext('2d') method of the canvas object is called to obtain the drawing context and its drawing functions.

Question 2: Drawing a Line

Q: How do you draw a simple line on the canvas?

ctx.beginPath();

ctx.moveTo(10, 10); // Starting point (x,y)

ctx.lineTo(150, 50); // End point (x,y)

ctx.stroke();

Explanation:

The beginPath() method starts a new path and moveTo(x, y) sets the starting point of the line. The lineTo(x, y) method defines the end point of the line. Finally, stroke() actually draws the line on the canvas.

Question 3: Setting Line Width

Q: How do you change the width of a line on the canvas?

ctx.lineWidth = 5;

Explanation:

The lineWidth property of the context ctx sets the width of lines drawn in the future. This value is set before drawing shapes like lines or circles.

Question 4: Drawing a Circle

Q: How do you draw a circle on the canvas?

```
ctx.beginPath();
ctx.arc(100, 75, 50, 0, 2 * Math.PI);
ctx.stroke();
```

Explanation:

The arc(x, y, radius, startAngle, endAngle) method creates a circle. The x and y parameters define the center of the circle, radius sets its size, and the angles define the start and end points of the arc in radians.

Question 5: Filling a Circle

Q: How do you fill a circle with color on the canvas?

```
ctx.fillStyle = 'red';
ctx.fill();
```

Explanation:

fillStyle sets the color, gradient, or pattern used to fill the drawing. fill() fills the current drawing (path) with the color specified in fillStyle.

Question 6: Creating Dashed Lines

Q: How do you create a dashed line on the canvas?

```
ctx.setLineDash([5, 15]);
ctx.stroke();
```

Explanation:

The setLineDash() method of the context sets the line dash pattern as an array of numbers, which specify the lengths of alternating dashes and gaps.

Question 7: Clearing the Canvas

Q: How do you clear the entire canvas?

ctx.clearRect(0, 0, canvas.width, canvas.height);

Explanation:

clearRect(x, y, width, height) clears the specified rectangular area and sets it to transparent black.

Question 8: Drawing a Semi-Circle

Q: How do you draw a semi-circle on the canvas?

```
ctx.beginPath();
ctx.arc(100, 75, 50, 0, Math.PI);
ctx.stroke();
```

Explanation:

The arc() method with the endAngle parameter set to Math.PI (180 degrees in radians) draws a semi-circle.

Question 9: Creating Multiple Circles in a Loop

Q: How can you draw multiple circles on the canvas using a loop?

```
for(let i = 0; i < 5; i++) {
   ctx.beginPath();
   ctx.arc(50 * (i + 1), 50, 40, 0, 2 * Math.PI);
```

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

Explanation:

Using a for loop, multiple circles are drawn. The arc() method's x coordinate is multiplied by the iterator (i + 1), placing each circle further along the x-axis.

Question 10: Responding to User Input

Q: How can you draw a line on the canvas in response to user mouse movements?

HTML:

```
<canvas id="myCanvas" width="200" height="200"></canvas>
```

JavaScript:

```
canvas.addEventListener('mousemove', function(event) {
    const x = event.offsetX;
    const y = event.offsetY;
    ctx.lineTo(x, y);
    ctx.stroke();
});
```

Explanation:

An event listener for mousemove is added to the canvas. The offsetX and offsetY properties of the event provide the mouse's position, which are used in lineTo() and stroke() to draw a line following the mouse movement.