HTML5 Canvas

Basic Setup

HTML Structure:

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>HTML5 Canvas Example</title>
</head>
<body>
   <canvas id="myCanvas" width="500" height="500" style="border:1px solid #000000;"></canvas>
   <script src="script.js"></script>
</body>
</html>
```

JavaScript for Drawing:

```
window.onload = function() {
   var canvas = document.getElementById('myCanvas');
   var context = canvas.getContext('2d');

   // Draw a rectangle
   context.fillStyle = "#FF0000";
   context.fillRect(0, 0, 150, 75);

   // Draw a circle
   context.beginPath();
   context.arc(200, 200, 40, 0, 2 * Math.PI);
   context.stroke();

   // Draw a line
   context.moveTo(250, 250);
   context.lineTo(350, 350);
   context.stroke();
};
```

Key Concepts

Getting the Canvas Context:

The canvas element provides a method getContext(), which can be used to get the rendering context and its drawing functions. For 2D rendering, you use canvas.getContext('2d').

Drawing Shapes:

Rectangles: fillRect(x, y, width, height) draws a filled rectangle.
Paths: You can create complex shapes by defining paths. Use methods like beginPath(), moveTo(x, y), lineTo(x, y), and stroke().
Arcs and Circles: arc(x, y, radius, startAngle, endAngle) is used for drawing circles and arcs.

Styling:

Fill Styles: fillStyle sets the color, gradient, or pattern used to fill the drawing.
Stroke Styles: strokeStyle sets the color, gradient, or pattern used for strokes.

Text:

You can draw text using fillText(text, x, y) and strokeText(text, x, y). The font can be set using context.font.

Advanced Topics

Images:

You can draw images using drawImage(image, dx, dy) for simple placement, or more complex forms with scaling and cropping.

Transforms:

Canvas supports transformations like translate(x, y), rotate(angle), and scale(x, y) to manipulate the drawing context.

Animation:

Animations can be created by repeatedly drawing on the canvas using requestAnimationFrame() for smooth rendering.

Example: Drawing an Image

HTML:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Canvas Image Example</title>
</head>
<body>
  <canvas id="myCanvas" width="500" height="500" style="border:1px solid
```

```
#000000;"></canvas>
  <script src="script.js"></script>
</body>
</html>
```

JavaScript:

```
window.onload = function() {
  var canvas = document.getElementById('myCanvas');
  var context = canvas.getContext('2d');
  var image = new Image();
  image.src = 'path/to/your/image.jpg';
  image.onload = function() {
    context.drawImage(image, 0, 0);
  };
};
```