

課題8：Canvas を使って図形を書こう

HTML5 には図形を描画する Canvas という領域がある。これを使うと java と同様に絵を描くことができる。これを使って絵を書いてみよう

【index.html】

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Canvas に基本図形を描画する</title>
    <link rel="stylesheet" href="css/main.css" type="text/css" media="all">
    <script src="js/canvas.js"></script>
  </head>
  <body>
    <h1>Canvas に基本図形を描画する</h1>
    <canvas id="myCanvas" width="640" height="360">
      Canvas が利用できるブラウザを使用してください。
    </canvas>
    <div id="status"></div>
  </body>
</html>
```

このように canvas 領域を作っておく。ここに javascript で画像を描画していく。

【canvas.js】

```
// ページの読み込みが完了したら処理をする
window.addEventListener("load", function(){
  var ele = document.getElementById("status");
  // Canvas が使えるか調べる
  if (!window.HTMLCanvasElement){
    ele.innerHTML = "Canvas が使用できません";
    return;
  }
  // Canvas の要素
  var canvasObj = document.getElementById("myCanvas");
  // 2D コンテキストの取得
  var context = canvasObj.getContext("2d");
```

```
if (!context){
    ele.innerHTML = "2D コンテキストが取得できません";
    return;
}
```

```
// 四角形を描画する
```

```
context.fillStyle = "black";
context.fillRect(0,0, canvasObj.width, canvasObj.height);
context.strokeStyle = "rgba(255, 0, 0, 0.5)";
context.lineWidth = 10;
context.strokeRect(10,250, 200, 100);
```

```
// 直線を描画する
```

```
context.strokeStyle = "hsla(120, 100%, 50%, 0.5)";
context.beginPath();
context.moveTo(10,20);
context.lineTo(250,350);
context.stroke();
```

```
// 円弧を描画する
```

```
context.beginPath();
context.strokeStyle = "orange";
context.arc(90, 100, 50, 0, Math.PI, true);
context.stroke();
```

```
// 3次ベジエ曲線を描画する
```

```
context.strokeStyle = "white";
context.lineCap = "round";
context.beginPath();
context.moveTo(470, 50);
context.bezierCurveTo(520, 120, 630, 100, 570, 260);
context.stroke();
```

```
// 三角形を描画する
```

```
context.strokeStyle = "cyan";
context.fillStyle = "blue"
context.beginPath();
context.moveTo(250, 30);
context.lineTo(300, 150);
context.lineTo(200, 150);
context.closePath();
context.fill();
```

```
context.stroke();
// クリッピング領域内に四角形を描画する
context.save();
context.beginPath();
context.arc(430, 250, 90, 0, Math.PI*2, false);
context.clip();
context.strokeStyle = "yellow";
context.lineWidth = 20;
context.strokeRect(310,200, 200, 100);
context.restore();
}, true);
```

【main.css】

```
canvas {
    border:1px solid black;
}
```

描画は3Dも可能です。ここでは2Dで行います。

四角のパラメータはすべて（X座標、Y座標、横幅、縦幅）となっています。

strokeRect()は四角形の枠だけ。fillRect()は塗りつぶされた四角形。

直線は、stroke()を使用します。Canvasでは開始座標、終了座標で描画するのではなく、パス（経路）を構築してそれを直線や曲線でつないでいくという手法を使います。

円弧は arc()を使用します。（中心 X 座標、中心 Y 座標、半径、開始角度、終了角度、反時計回りかどうかのフラグ）のパラメータで行います。角度は45度や135度のような度（degree）ではなく、ラジアンで指定します。ラジアンだと180度が π 、360度が 2π です。JavascriptではMath.PIで表します。回り方はtrueが反時計、falseが時計回りです。

曲線はベジエ曲線で描画します。bezierCurveTo()。ベジエは指定した座標を滑らかに近接する曲線である。

Save()とrestore()はcanvasの状態をスタックに保存、復帰させるもので、これをうまく使うと回転の処理を行うことができます。

描画できたら、数値を変化させて、どのように描画されるか試してみよう。

