

# Canvas API

## ①canvas要素：描画の要素

```
25:    <canvas id="canvas" width="400" height="400"></canvas>
```

- ・ canvasは描画の要素

## ②canvas要素の取得と道具箱の生成

```
42:    const canvas = document.getElementById("canvas");  
43:    const c = canvas.getContext("2d");
```

- ・ 40行は要素の取得。41行は道具箱c(2次元のコンテキスト)の生成

## ③直線の描画

```
47:    c.translate(canvas.width / 2, canvas.height / 2);  
53:    c.save();  
55:    c.translate(0, -canvas.height / 2);  
56:    c.beginPath();                                //パスの開始  
57:    c.moveTo(line[i].x1, line[i].y1);             //開始点の設定  
58:    c.lineTo(line[i].x2, line[i].y2);             //終点の設定  
59:    c.strokeStyle = line[i].color;                //線色  
60:    c.lineWidth = line[i].width;                  //線の太さ  
61:    c.stroke();                                    //線描画
```

- ・ 47行はcanvasの原点を左上からcanvasの中心に移動している(中心を軸に回転するため)
- ・ 52行はひとつのパターンを描く際の原点をトップ中心に変更して座標がわかりやすくする
- ・ 53行以下、直線の描画

# 配列

## ①配列の生成

```
28:    let line = new Array();
```

- ・ 複数のデータが入る「配列(Array)」を作成

## ②配列に値を代入

```
31:    line[0] = { x1: 0, y1: 0, x2: 0, y2: 200, color: "white", width: 1  
};  
32:    //line[1] = { x1: 10, y1: 0, x2: 100, y2: 400, color: "lime", width:  
1 };  
33:    //line[2] = { x1: 20, y1: 0, x2: 150, y2: 400, color: "skyblue",  
width: 1 };  
34:    //line[3] = { x1: 30, y1: 0, x2: 200, y2: 400, color: "yellow",  
width: 1 };
```

- ・ 配列の0番地から開始点(x1,y1)終了点(x2,y2)色、先の太さを設定
- ・ 練習の都合上、1〜3番地はコメントアウトしているが、コメント記号(//)の消去・4番地以降の配列を追加しても構わない。

# 各種値の設定

## ①背景色

```
36:    let bgColor = "transparent";    //曼荼羅内背景色
44:    canvas.style.backgroundColor = bgColor;
```

- ・ canvas内の背景色をJavaScriptでCSS変更

## ②角丸

```
37:    let borderRadius = "50%";        //角丸
45:    canvas.style.borderRadius = borderRadius;
```

- ・ canvasの角をJavaScriptで変更（50%は半径分の角の丸み）

## ③一回転で繰返す数

```
38:    let divide = 1;                    //分割
48:    const repeat = setInterval(loop, duration*1000);
51:    function loop() {
52:        for (let i = 0; i < line.length; i++) {
62:        }
63:        c.rotate(2 * Math.PI / divide);
64:        i++;
65:        if (divide < i) { clearInterval(repeat); }
66:    }
```

- ・ setInterval：タイマー、loop関数を～ミリ秒で繰返し実行する
- ・ clearInterval：タイマーを終了させる

## ④周期：経過時間

```
39:    let duration = 0.5;                //周期(秒)
47:    const repeat = setInterval(loop, duration*1000);
```

- ・ ミリ秒を秒に変換(x1000)