

## 課題 1 1 : マウストラッグで描画する

マウスの移動を読み取り、canvas 内に線を描く手法を学ぼう

### 【index.html】

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>マウスで描画</title>
    <link rel="stylesheet" href="css/main.css" type="text/css" media="all">
    <script src="js/canvas.js"></script>
  </head>
  <body>
    <h1>マウスで描画</h1>
    <form>
      文字 : <input type="text" size="4" id="char" value="翔">
      <input type="button" id="drawText" value="文字を描画">
    </form>
    <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>
```

### 【main.css】

```
canvas {
  border:1px solid black;
  position:absolute; /* Firefox */
}
```

### 【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;
    }
    // Canvas を塗りつぶす
    context.fillStyle = "black";
    context.fillRect(0, 0, canvasObj.width, canvasObj.height);
    // 描画
    canvasObj.oldX = 0;
    canvasObj.oldY = 0;
    canvasObj.drawFlag = false;
    canvasObj.addEventListener("mousemove", function(evt){
        if (!canvasObj.drawFlag){
            return;
        }
        var x = evt.offsetX || evt.layerX;
        var y = evt.offsetY || evt.layerY;
        context.strokeStyle = "rgba(255, 255, 255,1)";
        context.lineWidth = 10;
        context.lineCap = "round";
        context.beginPath();
        context.moveTo(canvasObj.oldX, canvasObj.oldY);
        context.lineTo(x, y);
        context.stroke();
        context.closePath();
        canvasObj.oldX = x;
        canvasObj.oldY = y;
    }, false);
    canvasObj.addEventListener("mousedown", function(evt){
        canvasObj.oldX = evt.offsetX || evt.layerX;
        canvasObj.oldY = evt.offsetY || evt.layerY;
    });

```

```

        canvasObj.drawFlag = true;
    }, false);
    canvasObj.addEventListener("mouseup", function(evt){
        canvasObj.drawFlag = false;
    }, false);
    // 文字を描画する
    document.getElementById("drawText").addEventListener("click", function(){
        context.fillStyle="black";
        context.fillRect(0, 0, canvasObj.width, canvasObj.height);
        context.fillStyle="gray";
        context.font = "normal bold 256pt 'MS P明朝";
        context.textAlign = "center";
        var text = document.getElementById("char").value;
        context.fillText(text, canvasObj.width/2, 320);
    }, true);
}, true);

```

大きな文字の表示は context.fillText で行なっている。drawText ボタンのクリックのイベントで char テキストボックスの中の文字を画面いっぱいに表示します。

マウスのダウンで、現在のマウスのポジションを oldX,oldY に保存します。普通は offsetX,offsetY で現在のマウスポジションが取れるのですが、困ったことに FireFox だけは layerX,layerY に値が返ってくるので、||演算子で offsetX を優先し、なければ layerX を見るようにします。

mousemove したら、クリックされている状態であればドラッグと判断します。それ以外はなにもしません。Mousemove イベントが発生するのはある程度マウスが移動した時ですから、その時に現在のポジションと oldX,oldY の間に線を引きます。そして、現在のポジションを新しい oldX,oldY にします。

マウスのアップで、描画フラグをオフにします。