

第3章 マウスの座標の取得

Web応用

第9回 マウスの座標位置の取得

第3章

マウス座標の取得

第3章 学習目標

マウスの座標を取得する方法について学びましょう。

addEventListener()

マウスが移動したときにプログラムが動作するという機能を制作します。

1. ファイルの準備

ファイルを準備しましょう。ファイル名は「sample9-3.html」です。

■ ソースコード

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8">
5     <title>サンプル9-3</title>
6     <style>
7     </style>
8   </head>
9   <body>
10
11     <script>
12
13     </script>
14   </body>
15 </html>
```

2. canvas要素

canvas要素(idは「canvas1」)と描画のコードを記述します。

■ ソースコード

```
9    <body>
10   <p><canvas id="canvas1" width="300" height="300"></canvas></p>
11   <script>
12     var canvas1 = document.getElementById("canvas1");
13     var ctx1 = canvas1.getContext("2d");
14     ctx1.fillStyle="blue";
15     ctx1.fillRect(50,100,200,100);
16   </script>
17 </body>
```

3. マウス座標の取得

それでは、マウスの座標を取得しましょう。

canvas要素に「イベントリスナ」を登録します。イベントリスナは、「なにかしらイベントが発生した時の処理」を指定します。

■ ソースコード

```
11   <script>
12     var canvas1 = document.getElementById("canvas1");
13     var ctx1 = canvas1.getContext("2d");
14
15     canvas1.addEventListener("mousemove",function(){
16       ctx1.fillStyle="blue";
17       ctx1.fillRect(50,100,200,100);
18     });
19   </script>
```

canvas要素上にマウスが入って移動したら青い長方形が表示されます。

解説：

- ・ イベントリスナは、「指定する要素.addEventListener(イベント,処理の内容);」という記述です。
- ・ サンプルでは、canvas要素上をマウスが移動(mousemove)した時に、function(){ }内の処理を実行します。
- ・ function(){…処理…}は、「無名関数」と言い、関数名をつけることなく直ちにプログラムを実行したい時に用います。

参考：

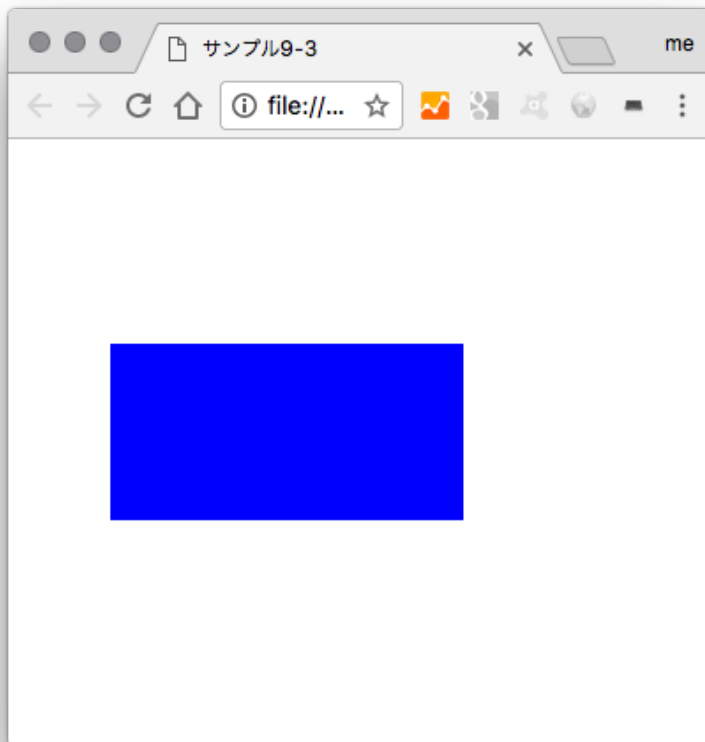
次のようにイベントリスナで関数名「rectDraw」を指定して、別に関数を定義「function rectDraw(){〜}」することも可能です。

■ ソースコード

```

11 <script>
12   var canvas1 = document.getElementById("canvas1");
13   var ctx1 = canvas1.getContext("2d");
14
15   canvas1.addEventListener("mousemove",rectDraw);
16
17   function rectDraw(){
18     ctx1.fillStyle="blue";
19     ctx1.fillRect(50,100,200,100);
20   }
21 </script>

```



e.clientX、e.clientY

では、いよいよマウスの座標を取得しましょう。

1. 仮引数「e」

先程の無名関数`function(){}()`の中に、仮引数「e」を記述しましょう。

■ ソースコード

```

15 canvas1.addEventListener("mousemove",function(e){
16     ctx1.fillStyle="blue";
17     ctx1.fillRect(50,100,200,100);
18 });

```

解説：

canvas要素の上をマウスが移動した時の位置情報などはイベントリスナ内の関数の第1仮引数に全て収められる仕様になっています。ですのでサンプルでは「e」が受け皿となって全ての情報が入っています。

2. マウスの座標

マウスの座標は、「e」の中に入っており、「e.clientX」「e.clientY」で取得できます。

■ ソースコード

```

15 canvas1.addEventListener("mousemove",function(e){
16     e.clientX;
17     e.clientY;
18     ctx1.fillStyle="blue";
19     ctx1.fillRect(50,100,200,100);
20 });

```

マウスの座標の位置に描画

描画の座標の変数を「x」「y」と宣言し、マウスの座標を代入して、矩形を描画させます。

1. 矩形の開始点の座標の変更

fillRect()内のx座標、y座標の値を、「x-2、y-2」に変更しましょう。

■ ソースコード

```

15 var x,y;
16 canvas1.addEventListener("mousemove",function(e){
17     x = e.clientX;
18     y = e.clientY;
19     ctx1.fillStyle="blue";
20     ctx1.fillRect(x-2,y-2,4,4);
21 });

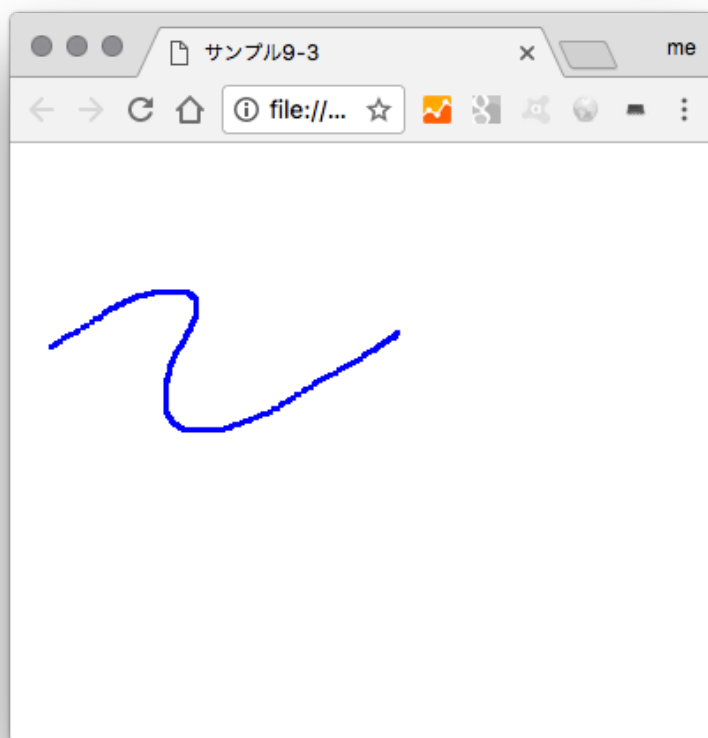
```

解説：

1. var x,y;で、変数xと変数yを一度に宣言します。
2. マウスの座標「e.clientX」「e.clientY」を変数x、変数yにそれぞれ代入します。
3. 矩形の横幅、高さも4pxと小さくし、矩形の開始点の座標は「x-2,y-2」に変更します。

2. 確認

ブラウザで確認しましょう。絵を描くことができるようになりました。



座標を補正

先ほど作ったアプリの動作をよく見ると、マウスポインタの先端と描画されている矩形の位置がずれています。これは、マウスの座標の原点とcanvasの原点の位置が一致していないからです。この章で補正しましょう。

1.補正

次のように補正します。

■ ソースコード

```
15  var x,y;
16  canvas1.addEventListener("mousemove",function(e){
17      var rect = e.target.getBoundingClientRect();
18      x = e.clientX-rect.left;
19      y = e.clientY-rect.top;
20      ctx1.fillStyle="blue";
21      ctx1.fillRect(x,y,200,100);
22  });
```

解説：

- 「rect」は、ここではcanvas要素の境界面の情報になります。
- rect.leftはコンテンツ表示内左端とcanvas要素の左の隙間の値、rect.topはコンテンツ表示内上端とcanvas要素の上の隙間の値になります。これらを引き算して補正します。

クリックしたときだけ動作させる

このままでは、マウスが移動したらそのまま描画されてしまいますので、マウスボタンを押しているときだけ描画するように改良し、ボタンを押しているときだけ描画できるように改造します。

1. マウスを押したときだけ描画する

変数「drawOK」を用意します。マウスボタンをクリックしたときのイベントリスナ、ボタンを離したときのイベントリスナも準備し、変数「drawOK」の値を変えることで描画を制御します。

■ ソースコード

```
15  var x,y;
16  var drawOK=false;
17  canvas1.addEventListener("mousemove",function(e){
18      var rect = e.target.getBoundingClientRect();
19      x = e.clientX-rect.left;
20      y = e.clientY-rect.top;
21      ctx1.fillStyle="blue";
22      if(drawOK){
23          ctx1.fillRect(x,y,200,100);
24      }
25  });
26
27  canvas1.addEventListener("mousedown",function(e){
28      drawOK = true;
29  });
30
31  canvas1.addEventListener("mouseup",function(e){
32      drawOK = false;
33  });
```

解説：

- 「drawOK」は、値に「true」「false」のみを持つ「Boolean型（論理型）」という型の変数です。
- 条件式では、「if(drawOK == true){〜}」などと記述しなくても、「if(drawOK){〜}」だけで条件を満たす（=true）ならば処理を行います。
- 「mousedown」はマウスのボタンを押し下げた瞬間、「mouseup」はマウスのボタンを押し上げた瞬間のイベントになります。

練習問題1

問題

【クイズ】 択一選択（即解答表示）

マウスの座標を取得するのに、用いるソースコードはどれか。なおイベントリスナでの仮引数は「e」とする。

- ☐ e.mouseX と e.mouseY
- ☐ e.targetX と e.targetY
- ☐ e.positionX と e.positionY
- ☐ e.clientX と e.clientY

練習問題1の解説

正解は、**e.clientX** と **e.clientY** です。

マウスの座標は、イベントリスナでの仮引数「e」としたとき、

x軸方向は「e.clientX」、y軸方向は「e.clientY」

になります。

第3章 まとめ

マウスの座標を取得する方法について学びました。

第3章 終わり

Web応用

第9回 マウスの座標位置の取得

第3章

マウス座標の取得

おわり

© Cyber University Inc.