

SVG 資料第 6 回目 (その 3)

SVG と HTML の間でデータを交換する

メディア専門ユニット I(SVG)

2016/5/30

ソースコード (1)

第 6 回目 (その 3)

メディア専門ユニット I(SVG)

```
1<!DOCTYPE html>
2<html xmlns:svg="http://www.w3.org/2000/svg"
3      xmlns:xlink="http://www.w3.org/1999/xlink">
4<head>
5<meta charset="UTF-8"/>
6<script type="text/ecmascript" src="SSClickPos.js"></script>
7<link rel="stylesheet" type="text/css" href="HTML.css">
8<title>HTML と SVG の間でデータを交換させる</title>
9</head>
10<body>
11  <h1 class="display">クリック位置を HTML 内に表示、HTML から移動させる</h1>
12  <div class="Cell">
```

- ▶ 6 行目から JavaScript の部分が開始
- ▶ 10 行目から 11 行目で色の選択をするプルダウンメニューに現れる色の情報が格納されたオブジェクトを定義

ソースコード (2)–プルダウンメニューの作成

```
13 <svg height="410" width="410" id="canvas">
14   <g transform="translate(5,5)">
15     <g id="field">
16       <rect x="0" y="0" width="400" height="400" fill="lightgray"/>
17       <circle id="Circle" cx="200" cy="50" r="20" fill="red"/>
18       <text class="textStyle" x="50" y="50"> X</text>
19       <text class="textStyle" id="X" x="150" y="50"></text>
20       <text class="textStyle" x="50" y="100"> Y</text>
```

87 行目にある<select>要素に現れる<option>要素を作成

- ▶ 13 行目で<select>要素を得ている
- ▶ それぞれの色に対して 15 行目で<option>要素を作成
- ▶ 16 行目で属性 value を配列のキーに設定
- ▶ 17 行目でテキストノードを作成
- ▶ 18 行目でそれを<option>要素の子要素にする
- ▶ 19 行目で<option>要素を<select>要素の子要素にする

ソースコード (3)–初期化

第 6 回目 (その 3)

メディア専門ユニット I(SVG)

```
21         <text class="textStyle" id="Y" x="150" y="100"></text>
22     </g>
23     <path fill="blue" d="M-5,-5 405,-5 405,405 -5,405z M0,0 0,400 400,400 400,0
24 </g>
25 </svg>
26 </div>
27 <div class="Cell" >
28     <div><label for="XP">x</label>
29     <input type="text" id="XP" size="3" /></div>
30     <div><label for="YP">y</label>
31     <input type="text" id="YP" size="3" /></div>
32     <div><label for="SelectColor">色</label>
33     <select id="SelectColor"></select></div>
34     <input id="SetColor" type="button" value="設定"></input>
35 </div>
```

ソースコード (3)-初期化 (解説)

- ▶ 21, 22 行目で円の中心の座標が入る HTML 文書内のテキストボックスの要素を得ている
- ▶ 23 行目では円の要素を得ている
- ▶ 24, 25 行目では円の中心の座標が入る SVG 内のテキスト表示位置の要素を得ている
- ▶ 26, 27 行目では SVG 内の円の中心位置の座標を HTML 内のテキストボックスに設定
- ▶ 31 行目では SVG 文書の、HTML 文書内での上からと左からの位置を示す BoundingClientRect オブジェクトを得ている
- ▶ これはクリックした位置が HTML 文書内からの位置となり、イベントオブジェクトからの座標をそのまま使うと円の位置がずれるため
- ▶ SVG は縁取りが 5px あるようにつくっであるので 32, 33 行目でその分も補正值に加えている
- ▶ 34 行目では画面の情報などをアップデートするための関数 `refreah()` を呼んでいる。

ソースコード (4)–イベント処理関数

第 6 回目 (その 3)

メディア専門ユニット I(SVG)

```
36</body>
```

```
37</html>
```

- ▶ 36 行目から 40 行目では画面がクリックされたときの処理関数が定義
イベントが起きた座標から補正值を引いた値を HTML 内に設定し (37 行目と 38 行目) その後画面の書き直しを実行
- ▶ 41 行目から 45 行目で画面の情報をアップデートする関数を定義
SVG 内の座標表示と円の属性 `fill` の属性値を変更
- ▶ 47 行目から 55 行目は SVG 内の座標位置の表示関数 (以前と同じ)

37

- ▶ 58 行目で外部の CSS ファイルを読み込む
- ▶ 読み込むために<link>要素を用いる
- ▶ 59 行目は<title>要素を記述

スタイルシート (その 1)

```
1 .display {  
2   font-size:25px;  
3 }  
4 .textStyle {  
5   font-size:30px;  
6   text-anchor:end;  
7 }
```

- ▶ `.display` はクラス名が `display` の要素に適用
- ▶ ここでは表題の部分でフォントの大きさを指定
- ▶ `.textStyle` は SVG 内の座標の表示の部分に適用
- ▶ フォントの大きさと文字の位置 (右寄せ) を指定

```
8.Cell {
9  font-size:30px;
10 display:inline-block;
11 vertical-align:middle;
12 padding-left:5px;
13}
14#XP, #YP{
15  font-size:25px;
16  text-align: right;
17}
18#SetColor, #SelectColor {
19  font-size:25px;
20  text-align:center;
```

- ▶ 全体の要素を表の形で表示するための CSS が定義
- ▶ margin-left は左側に余白を設定
- ▶ table-row は表の行を指定
- ▶ table-cell は表の各項目を指定

21 }

- ▶ #XP は属性 id が XP の要素に対して適用
- ▶ ここでは数が入るので右寄せを指定
- ▶ , で並べるとそれぞれのセクターに対して適用される
- ▶ 残りも同じ

ソースコード (4)–SVG の部分

第 6 回目 (その 3)

メディア専門ユニット I(SVG)

ソースコード (4)–SVG の部分 (解説)

第 6 回目 (その 3)

メディア専門ユニット I(SVG)

- ▶ SVG の要素は HTML 内の表の一部として現れる
- ▶ SVG の大きさが設定されているのが甘えと異なる
- ▶ クリック範囲をわかりやすくするため、外枠を設定 (76 行目から 77 行目)

ソースコード (5)-テキストボックス等の部分

21

- ▶ 円の中心を指定するテキストボックス (82 目から 85 行目)
- ▶ 色を選択するためのプルダウンメニュー (87 行目)
- ▶ 設定を実行するためのボタン (88 行目)

- ▶ HTML 内の表題の部分のフォントの大きさ (.display の font-size) を変えてもクリックの位置と値が一致していることを確かめる
- ▶ 画面を表示した後、ブラウザの横幅を変えて表題部分の行数が増えるとクリックした位置と円の移動位置が一致しないことを確かめる
- ▶ 上記の不具合を直す
- ▶ 色の選択の種類を増やす
- ▶ SVG 内に図形を追加し、HTML 内に図形を選択するプルダウンメニューまたはラジオボタンを置き、クリックしたとき選択された図形だけが移動するようにする