

## 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)

2017/5/30

# HTML のソースコード (1)

第 6 回目 (その 3)

メディア専門ユニット I(SVG)

SVG と HTML の  
間でデータ交換

やってみよう

```
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="make-svg-elm.js"></script>
7<script type="text/ecmascript" src="SSClickPos.js"></script>
8<link rel="stylesheet" type="text/css" href="HTML.css">
9<title>HTML と SVG の間でデータを交換させる</title>
10</head>
```

- ▶ 6 行目で外部 JavaScript ファイルを読み込み
- ▶ 7 行目で外部 CSS ファイルを読み込むための  
<link>要素  
CSS ファイルの呼び出しが<style>要素でないことに  
注意

## HTML のソースコード (2)-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 と HTML の  
間でデータ交換

やってみよう

```
11 <body>
12 <h1 class="display">クリック位置を HTML 内に表示、HTML から移動させる</h1>
13 <div class="Cell">
14   <svg height="410" width="410" id="canvas">
15     <g transform="translate(5,5)">
16       <g id="field">
17         <rect x="0" y="0" width="400" height="400" fill="lightgray"/>
18         <circle id="Circle" cx="200" cy="50" r="20" fill="red"/>
19         <text class="textStyle" x="50" y="50"> X</text>
20         <text class="textStyle" id="X" x="150" y="50"></text>
21         <text class="textStyle" x="50" y="100"> Y</text>
22         <text class="textStyle" id="Y" x="150" y="100"></text>
23       </g>
24     <path fill="blue" d="M-5,-5 405,-5 405,405 -5,405z M0,0 0,400 400,400 400,0
25     </g>
26   </svg>
27 </div>
```

- ▶ 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 (17 行目) のが以前と異なる
- ▶ クリック範囲をわかりやすくするため、外枠を設定 (24 行目)

## ソースコード (3)–テキストボックス等の部分

第 6 回目 (その 3)

メディア専門ユニット I (SVG)

SVG と HTML の  
間でデータ交換

やってみよう

```
28 <div class="Cell" >
29   <div><label for="XP">x=</label>
30     <input type="text" id="XP" size="3" /></div>
31   <div><label for="YP">y=</label>
32     <input type="text" id="YP" size="3" /></div>
33   <div><label for="SelectColor">色</label>
34     <select id="SelectColor"></select></div>
35   <input id="SetColor" type="button" value="設定"></input>
36 </div>
37 </body>
38 </html>
```

- ▶ 円の中心を指定するテキストボックス (29 目から 32 行目)
- ▶ 色を選択するためのプルダウンメニュー (34 行目)
- ▶ 設定を実行するためのボタン (35 行目)

# JavaScript ソースコード (1)–プルダウンメニューの作成

第 6 回目 (その 3)

メディア専門ユニット I (SVG)

SVG と HTML の  
間でデータ交換

やってみよう

```
1 let Circle, X, Y, XP, YP, oT, oL, B;
2 window.onload = function() {
3   let Colors = {"red": "赤", "yellow": "黄色", "green": "緑",
4                 "blue": "青", "gray": "灰色", "black": "黒"};
5   let SelectColor = document.getElementById("SelectColor");
6   for( let Color in Colors) {
7     let opt = MKHTMLLElm(SelectColor, "option", {value: Color}, {});
8     opt.appendChild(document.createTextNode(Colors[Color]));
9   }
```

HTML ファイルの 34 行目にある<select>要素に現れる  
<option>要素を作成

- ▶ 6 行目で<select>要素を得ている
- ▶ それぞれの色に対して 8 行目で<option>要素を作成し、属性 value を配列のキーに設定
- ▶ 8 行目でテキストノードを作成し、<option>要素の子要素に設定

## ソースコード (2)-初期化

第 6 回目 (その 3)

メディア専門ユニット I(SVG)

SVG と HTML の  
間でデータ交換

やってみよう

```
10 XP = document.getElementById("XP");
11 YP = document.getElementById("YP");
12 Circle = document.getElementById("Circle");
13 X = document.getElementById("X");
14 Y = document.getElementById("Y");
15 XP.value = Circle.getAttribute("cx");
16 YP.value = Circle.getAttribute("cy");
```

- ▶ 10, 11 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
- ▶ 12 行目では円の要素を得ている
- ▶ 13, 14 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
- ▶ 15, 16 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

## ソースコード (3)-初期化

メディア専門ユニッ  
ト (SVG)SVG と HTML の  
間でデータ交換

やってみよう

```

17 document.getElementById("field").addEventListener("click",click, false);
18 document.getElementById("SetColor").addEventListener("click",refresh, false);
19 B = document.getElementById("canvas").getBoundingClientRect();
20 oL = Math.floor(B.left)+5;
21 oT = Math.floor(B.top)+5;
22 refresh();
23}

```

- ▶ 17, 18 行目で SVG 上と「設定」ボタンのクリックイベントの処理関数を設定
- ▶ 19 行目では SVG 文書の、HTML 文書内での位置を示す BoundingClientRect オブジェクトを得ている。
- ▶ イベントオブジェクトで渡されるクリックした位置は HTML 文書内からの位置で、そのまま使うと円の位置がずれる。
- ▶ SVG は縁取りが 5px あるように作成しているので 20, 21 行目でその分も補正值に加えている
- ▶ 22 行目で画面の情報などを更新する関数 refresh() 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)

SVG と HTML の  
間でデータ交換

やってみよう

```
24function click(event) {
25  XP.value = event.clientX-oL;
26  YP.value = event.clientY-oT;
27  refresh();
28}
29function refresh() {
30  SetText(X,"cx", XP.value);
31  SetText(Y,"cy", YP.value);
32  SetAttributes(Circle,{fill:SelectColor.value, cx:XP.value, cy:YP.value});
33}
34function SetText(Element, attrib, Value) {
35  let txtNode = document.createTextNode(Value);
36  if(Element.firstChild) {
37    Element.replaceChild(txtNode, Element.firstChild);
38  } else {
39    Element.appendChild(txtNode);
40  }
41}
```



## ソースコード (4)-イベント処理関数 (解説)

第 6 回目 (その 3)

メディア専門ユニット I(SVG)

SVG と HTML の  
間でデータ交換

やってみよう

- ▶ 24 行目から 28 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 (25 行目と 26 行目) その後画面の書き直しを実行
- ▶ 29 行目から 33 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` の属性値と中心位置を変更
- ▶ 34 行目から 41 行目は SVG 内の座標位置の表示関数

# スタイルシート (その 1)

第 6 回目 (その 3)

メディア専門ユニット I(SVG)

SVG と HTML の  
間でデータ交換

やってみよう

```
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

## スタイルシート (その 2)

第 6 回目 (その 3)

メディア専門ユニット I (SVG)

SVG と HTML の  
間でデータ交換

やってみよう

```
8.Cell {  
9  font-size:30px;  
10 display:inline-block;  
11 vertical-align:middle;  
12 padding-left:5px;  
13}
```

属性 class が Cell の要素は左側の<svg>要素と右側の<input>要素 をそれぞれ内部に持つ<div>要素である。

- ▶ display は表示方法を指定。ここではひとつの文字のように取り扱うことを指定 (上記の 2 つの要素が横に並ぶ)
- ▶ vertical-align は垂直方向の位置指定。ここでは中央部を基準線に配置することを指定
- ▶ padding-left は要素の左側の空白を指定

## スタイルシート (その 3)

第 6 回目 (その 3)

メディア専門ユニット I(SVG)

SVG と HTML の  
間でデータ交換

やってみよう

```
14 #XP, #YP{
15   font-size:25px;
16   text-align: right;
17 }
18 #SetColor, #SelectColor {
19   font-size:25px;
20   text-align:center;
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レクトターに対して適用される
- ▶ 残りも同じ

- ▶ 色の選択の種類を増やす
- ▶ HTML 内の表題の部分のフォントの大きさ (`.display` の `font-size`) を変えてから再起動し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。
- ▶ 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が移動するようにする