

## SVG 資料第 4 回目 (その 2) DOM の操作とイベント

メディア専門ユニット I(SVG)

2017/5/16

- ▶ プログラムの実行中に内部または外部から伝えられる情報
- ▶ イベントに対応して処理するプログラムをイベント駆動型と呼ぶ
- ▶ イベント駆動型ではイベントの発生順序が前もって決められないのでそれぞれの処理は独立する必要がある。

この演習でよく使うイベントはマウスに関するものとファイルのロードが完了したときのイベント

イベントの発生条件	イベントの属性名
ファイルのロード終了時	onload
ボタンがクリックされた	onclick
ボタンが押された	onmousedown
マウスカーソルが移動した	onmousemove
マウスボタンが離された	onmouseup
マウスカーソルが範囲に入った	onmouseover
マウスカーソルが範囲から出た	onmouseout

# クリックした円の塗りつぶしの色を表示

第 4 回目 (その 2)

メディア専門ユニット I(SVG)

イベント

イベント処理

イベント処理関数の登録

円の上をクリックすると円の色を表示する

# デモの解説

第 4 回目 (その 2)

メディア専門ユニット I(SVG)

イベント

イベント処理

イベント処理関数の登録

- ▶ 円の上をクリックするとクリックした円の塗りつぶしの色がメッセージボックスに表示
- ▶ メッセージボックスを消さないで次の操作ができない (モーダルな window)

# 円の上をクリックすると円の色を表示する ソースコード (配布資料 136 ページ)

## 要素にイベント処理関数を定義

第 4 回目 (その 2)

メディア専門ユニット I(SVG)

イベント

イベント処理

イベント処理関数の登録

```
1<?xml version="1.0" encoding="UTF-8" ?>
2<svg xmlns="http://www.w3.org/2000/svg"
3     xmlns:xlink="http://www.w3.org/1999/xlink"
4     height="100%" width="100%">
5  <title>クリックするとメッセージボックスが表示</title>
6  <script type="text/ecmascript">
7//    <![CDATA[
8  function click(event) {
9    alert("Circle " +event.target.getAttribute("fill")+" clicked.");
10 }
11 //    ]]>
12</script>
13 <circle cx="50" cy="50" r="20" fill="red"    onclick="click(evt)" />
14 <circle cx="100" cy="50" r="20" fill="blue"  onclick="click(evt)" />
15 <circle cx="150" cy="50" r="20" fill="green" onclick="click(evt)" />
16</svg>
```

## 円の上をクリックすると円の色を表示する ソースコード (配布資料 136 ページ) 解説

- ▶ 13 行目から 15 行目で定義されている<circle>要素の属性 onclick に対して関数 click(evt) が呼び出されるようにしている。
- ▶ 変数 evt は発生したイベントオブジェクト
- ▶ 呼び出される関数は 8 行目から 10 行目で定義
- ▶ マウスによるイベントはマウスイベントと呼ばれる (配布資料 134 ページ)。
- ▶ イベントの属性 target はイベントが発生したオブジェクト
- ▶ イベントが発生したオブジェクトは SVG の要素で、その属性の値は getAttribute(属性名) で求められる
- ▶ ここでは属性 fill の値を求めている
- ▶ JavaScript では+演算子はどちらかの被演算子が文字列のときは文字列の接続 (二つの文字列をつなげる)
- ▶ 関数 alert は与えられた引数をメッセージボックスに表示

# やってみよう

第 4 回目 (その 2)

メディア専門ユニット I(SVG)

イベント

イベント処理

イベント処理関数の登録

- ▶ 円の別の属性値を表示
- ▶ `alert` 内の初めの文字列 ("Circle") を `event.target.tagName` に変える
- ▶ 図形を変えていくつかの属性値を表示
- ▶ 異なる図形をいくつか置いて、同様のことをする



# 円の上をクリックすると円の色を表示する ソースコード (配布資料 138 ページ)

開始時にイベント処理関数を定義

第 4 回目 (その 2)

メディア専門ユニット I(SVG)

イベント

イベント処理

イベント処理関数の登録

```
1<?xml version="1.0" encoding="UTF-8" ?>
2<svg xmlns="http://www.w3.org/2000/svg"
3    xmlns:xlink="http://www.w3.org/1999/xlink"
4    height="100%" width="100%" >
5  <title>クリックするとメッセージボックスが表示</title>
6  <script type="text/ecmascript">
7//    <![CDATA[
8    window.onload = function() {
9      let Cs = document.getElementsByTagName("circle");
10     for( let i=0; i< Cs.length; i++) {
11       Cs[i].addEventListener("click",click, false);
12     }
13   }
14   function click(event) {
15     alert('Circle ${event.target.getAttribute("fill")} clicked. ');
16   }
17//    ]]></script>
18   <circle cx="50" cy="50" r="20" fill="red" />
19   <circle cx="100" cy="50" r="20" fill="blue"/>
20   <circle cx="150" cy="50" r="20" fill="green" />
21</svg>
```

# 円の上をクリックすると円の色を表示する ソースコード (配布資料 138 ページ) 解説

## 開始時にイベント処理関数の解説 (1)

- ▶ 18 行目から 20 行目にある<circle>要素には属性 onclick がない
- ▶ 8 行目から 13 行目に window.onload に関数を代入
- ▶ JavaScript では関数もオブジェクトなので変数に代入が可能
- ▶ load イベントはファイルの内容がすべて呼び出されたのちに発生
- ▶ これにより onload イベントが発生したときにここで定義した関数が呼び出される

## 円の上をクリックすると円の色を表示する ソースコード (配布資料 138 ページ) 解説

### 開始時にイベント処理関数の解説 (2)

- ▶ document オブジェクトは SVG 文書を指す。
- ▶ `getElementsByTagName` は指定されたオブジェクト (ここでは document) の子要素のうち引数で与えられた要素名を持つオブジェクトをすべて求めてリスト (配列のようなもの) として返す
- ▶ このリストの長さは `length` プロパティで得られる
- ▶ 10 行目から 12 行目でこのリストのそれぞれにイベント処理関数 (イベントリスナー) を登録
- ▶ イベント処理関数の登録は `addEventListener` メソッドを利用
- ▶ 引数は、イベント名 (文字列)、イベント処理関数 (関数名)、イベント処理するタイミング (論理値、この目的は後日説明) の 3 つ
- ▶ イベント処理関数は 14 行目から 16 行目にある `click`
- ▶ `alert` 内の文字列はテンプレートリテラルで表示 (配布資料 137 ページ)

# 属性値を変える

## クリックした円の位置を変更 (配布資料 139 ページ)

第 4 回目 (その 2)

メディア専門ユニット I(SVG)

イベント

イベント処理

イベント処理関数の登録

```
1<?xml version="1.0" encoding="UTF-8" ?>
2<svg xmlns="http://www.w3.org/2000/svg"
3     xmlns:xlink="http://www.w3.org/1999/xlink"
4     height="100%" width="100%">
5  <title>クリックするとその円が移動</title>
6  <script type="text/ecmascript">
7//    <![CDATA[
8  window.onload = function() {
9    let Cs = document.getElementsByTagName("circle");
10    for( let i=0; i< Cs.length; i++) {
11      Cs[i].addEventListener("click",click, false);
12    }
13  }
14  function click(T) {
15    let Y = 150-parseInt(T.target.getAttribute("cy"));
16    T.target.setAttribute("cy",Y);
17  }
18//    ]]></script>
19  <circle cx="50" cy="50" r="20" fill="red"/>
20  <circle cx="100" cy="50" r="20" fill="blue"/>
21  <circle cx="150" cy="50" r="20" fill="green"/>
22</svg>
```

## ソースコードの解説-属性値を変える

第 4 回目 (その 2)

メディア専門ユニット I(SVG)

イベント

イベント処理

イベント処理関数の登録

- ▶ 以前のコードと違う点はイベント処理関数の内容だけ
- ▶ 15 行目でクリックされた円の中心の  $y$  座標 (属性 `cy`) を `getAttribute` メソッドで得ている
- ▶ 得られた  $y$  座標の値は文字列なので文字列を数に直す `parseInt` を用いて数に変換し、その値を 150 から引いている
- ▶ 属性 `cy` の初期値はすべての円で 50 なので 1 回目にクリックされるとその値が  $150 - 50 = 100$  となる。
- ▶ 次にクリックされると  $150 - 100 = 50$  となり、初期値に戻る。
- ▶ この値を `setAttribute` でクリックされた円の属性 `cy` に設定している (16 行目)

## やってみよう

第 4 回目 (その 2)

メディア専門ユニット I(SVG)

イベント

イベント処理

イベント処理関数の登録

- ▶ 円に代わりに正方形をいくつか置いてそのうえでクリックしたときに移動する
- ▶ クリックした要素の色を変える
- ▶ 異なる種類の要素を置いてクリックしたときに移動する (tagName プロパティを使う)