

# SVG 資料第 4 回目 (その 1) 簡単なイベント処理と開発者ツールの使い方

メディア専門ユニット I(SVG)

2016/5/16

- ▶ JavaScript の国際的規約は ECMAScript と呼ばれる言語仕様
- ▶ 2015 年に ECMAScript 第 6 版が、2016 年 6 月に第 7 版が発表されている。
- ▶ この授業では新しい規格を積極的に採用する。参考図書としてはオライリーから出ている「初めての JavaScript 第 3 版」を薦める。
- ▶ JavaScript のプログラムは<script>要素内に記述
  - ▶ 属性 type に text/ecmascript を指定
  - ▶ <script>要素内に直接プログラムを記述するためには CDATA セクションと呼ばれる範囲に記述
  - ▶ CDATA セクションの始まりは<![CDATA[、終了は]]>
  - ▶ この記述は JavaScript として文法上正しくないので //をつけて、JavaScript からみてコメントになるようにする。

# JavaScript の特徴

第 4 回目 (その 1)

メディア専門ユニット I(SVG)

JavaScript について

イベント

イベント処理

開発者ツール

- ▶ JavaScript では変数はどのような型のデータも保持できる。
- ▶ 変数は宣言しなくても利用可能
- ▶ 宣言する場合は `let` キーワードを用いる (通常は宣言をする)
- ▶ 文の最後には `;` を置く
- ▶ 文の記述方法は C 言語とほとんど同じ
- ▶ 関数は `function` キーワードで始める。
- ▶ そのあとに関数名と `()` 内に仮引数の宣言をする
- ▶ 関数本体は `{ }` 内に記述
- ▶ 値を戻す場合には `return` 文の後に戻り値を指定

# イベントとは

第 4 回目 (その 1)

メディア専門ユニット I(SVG)

JavaScript について

イベント

イベント処理

開発者ツール

- ▶ 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 1)

メディア専門ユニット I(SVG)

JavaScript について

イベント

イベント処理

開発者ツール

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 1)

メディア専門ユニット I(SVG)

JavaScript について

イベント

イベント処理

開発者ツール

- ▶ 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と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 1)

メディア専門ユニット I(SVG)

JavaScript について

イベント

イベント処理

開発者ツール

```
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 145 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 1)

メディア専門ユニット I(SVG)

JavaScript について

イベント

イベント処理

開発者ツール

- ▶ 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

各種ブラウザで表現は異なるが機能はほぼ同じ

- ▶ 開くためのショートカットキーは「F12」か「Ctrl+Shift+I」
- ▶ DOM ツリーの表示と修正など (Elements)
- ▶ インターラクティブな操作やエラー表示、プログラムからの出力 (Console)
- ▶ JavaScript などのプログラムソースの表示、ブレイクポイントの設定 (Source)
- ▶ ...

順番に見ていく

- ▶ 「Elements」
  - ▶ DOM ツリーを見る
  - ▶ 要素の属性を変えることができる
- ▶ 「Console」 タブ
  - ▶ プログラムの実行中にメッセージが出せる
  - ▶ その場で JavaScript のプログラムが実行できる
  - ▶ オブジェクトや変数の値の確認ができる
- ▶ 「Source」 タブ
  - ▶ JavaScript のソースコードにブレイクポイントを設定できる
  - ▶ 1 行ごとに実行させることができる

## 確認しましょう

自分の使用しているブラウザで次のことを確認する(ノートにメモすること)

- ▶ 「開発者ツール」に相当する機能の名称
- ▶ 「開発者ツール」を開くショートカットキー
- ▶ どこかの Web ページを開いて DOM ツリーのノードを展開し、DOM ツリー上でマウスカーソルを動かしたときの現象
- ▶ SVG ファイルを開いて要素のプロパティを修正する
- ▶ コンソールの入力がどこでできるか確認する。
- ▶ コンソールで `2+3;` と入力して結果が表示されることを確認
- ▶ コンソールで次のように入力したときの結果を確かめる(結果の展開が可能なはずなので確認すること)  
`document.getElementsByTagName("circle");`

ブラウザの機能、いかがでしたか