

# SVG 資料第 6 回目 (その 2)

## CSS セレクタと HTML 要素の追加

メディア専門ユニット I(SVG)

2016/5/30

- ▶ カスケーディングスタイルシート (CSS) は HTML 文書の要素の表示方法を指定
- ▶ CSS は JavaScript から制御可能

(配布資料付録 9 ページ) 参照

- ▶ 要素名で要素を選択するためには、要素名を指定する。
- ▶ 属性 id の属性値で要素を選択するためにはその属性値の前に # をつける。
- ▶ 属性 class の属性値で要素を選択するためにはその属性値の前に . をつける。

# CSS セレクタについて

第 6 回目 (その 2)

セレクタ	解説
*	任意
E	タイプが E
E[foo]	タイプが E で属性 "foo" を持つ
E[foo="bar"]	タイプが E で属性 "foo" の属性値が "bar"
E[foo~="bar"]	タイプが E で属性 "foo" の属性値が空白で区切られた一つが "bar"
E:link, E:visited	まだ訪れたことがない (:link) か訪れたことがある (visited) ハイパーリンクのアンカーである要素
E:active, E:hover, E:focus	ユーザーに操作されている状態中
E:enabled, E:disabled	使用可能 (:enable) か使用不可のユーザーインターフェイス
E:checked	チェックされているユーザーインターフェイス
E.warning	属性 class が "warning" である要素
E#myid	属性 id の属性値が "myid"
E F	要素 E の子孫である要素 F
E > F	要素 E の子である要素 F
E + F	要素 E の直後にある要素 F
E ~ F	要素 E の直前にある要素 F

メディア専門ユニット I(SVG)

CSS とは

HTML 要素をプログラムで作成する

やってみよう

## CSS セレクタを用いて要素を得る

第 6 回目 (その 2)

メディア専門ユニット I(SVG)

CSS とは

HTML 要素をプログラムで作成する

やってみよう

`querySelector()` と `querySelectorAll()` は引数に CSS セレクタを与えて要素のリストを得るメソッド  
例ラジオボタンでチェックしているものを探す

- ▶ ラジオボタンのグループは属性 `name`(属性名を `foo`) が共通なのでセレクタは  
`"input[name=\"foo\"] :checked"`
- ▶ このメソッドを使わないと、`getElenentsByName()` で該当の要素を選び、どれがチェックしているか調べる必要がある。

## やってみよう (1)

第 6 回目 (その 2)

メディア専門ユニット I(SVG)

CSS とは

HTML 要素をプログラムで作成する

やってみよう

前のビデオのサンプルに対して次のことを行おう。

- ▶ ラジオボタンのどこかをチェックした後で次のことを行う

```
document.querySelector("input[name=\"R1\"] : checked")
```

- ▶ `<select>`要素を`<select multiple>`にするとどうなるか

- ▶ このとき、「Cntrl+右クリック」をいくつかのところで行う
- ▶ コンソールで次のように入力した結果の違いを確認する。

```
document.querySelector("option:checked")
```

```
document.querySelectorAll("option:checked")
```

## やってみよう (2)

第 6 回目 (その 2)

メディア専門ユニット I(SVG)

CSS とは

HTML 要素をプログラムで作成する

やってみよう

次のメソッドを `querySelector()` または `querySelectorAll()` で書き直してみよう。

- ▶ `getElementsByTagName("foo")`
- ▶ `getElementById("foo")`
- ▶ `getElementsByClassName("foo")`
- ▶ `getElementsByName("foo")`

## <option>要素を追加 (1)

第 6 回目 (その 2)

メディア専門ユニット I(SVG)

CSS とは

HTML 要素をプログラムで作成する

やってみよう

```
1<!DOCTYPE html>
2<html>
3  <head>
4    <meta charset="UTF-8"/>
5    <title>HTML 要素の作成</title>
6    <script type="text/ecmascript" src="make-svg-elm.js"></script>
7    <script type="text/ecmascript">
8      //
9      window.onload = function(){
10        var F = document.getElementById("menu");
11        F.addEventListener("change",change,false);
12        F.addEventListener("click",click,false);
13        for(let i = 1; i&lt;=30; i++) {
14          let option = MKHTMLElm(F,"option",{ "value":i},{ });
15          option.appendChild(document.createTextNode(`${i}日`));
16        }
17      }</pre></div>
```

## <option>要素を追加 (1)–解説

第 6 回目 (その 2)

メディア専門ユニット I(SVG)

CSS とは

HTML 要素をプログラムで作成する

やってみよう

- ▶ 6 行目で前回作成した要素を作成する関数群のファイルを読み込む
- ▶ 10 行目で<select>要素を得ている。
- ▶ 11 行目と 12 行目でデバッグ用のイベントリスナーの登録
- ▶ 13 行目から 16 行目で<option>要素を<select>要素の子要素として追加
  - ▶ 14 行目で、<option>要素を作成し、同時にその属性値を設定
  - ▶ メニューに表示する文字列は<option>要素の子要素の `textContent` なので `createTextNode` で作成



## <option>要素を追加 (2)

第 6 回目 (その 2)

メディア専門ユニット I (SVG)

CSS とは

HTML 要素をプログラムで作成する

やってみよう

```
18 function change(E) {
19     console.log('change::target:${E.target.tagName}'+
20         ', E.currentTarget:${E.currentTarget}'+
21         ', value:${E.target.value}');
22 }
23 function click(E) {
24     console.log('click::target:${E.target.tagName}'+
25         ', E.currentTarget:${E.currentTarget}'+
26         ', value:${E.target.value}');
27 }
28 //]]>
29 </script>
30 </head>
31 <body>
32   <h3>プルダウンメニューの作成</h3>
33   <form>
34     <div>
35       <select id="menu"></select>
36     </div>
37   </form>
38 </body>
39 </html>
```

## <option>要素を追加 (2)–解説

第 6 回目 (その 2)

メディア専門ユニット I(SVG)

CSS とは

HTML 要素をプログラムで作成する

やってみよう

- ▶ 18 行目から 27 行目は前のビデオの例と同じ
- ▶ 35 行目に<option>要素が全くない<select>要素を作成

## やってみよう

第 6 回目 (その 2)

メディア専門ユニット I(SVG)

CSS とは

HTML 要素をプログラムで作成する

やってみよう

連番が設定できるメニューを作成する関数を作成せよ。それを用いて 3 つのメニューを並べて年、月、日が指定できるものを作成せよ。