

2014. 6. 23

医療者のための情報技術入門

第9回 プログラムがはたらくしくみを学ぶ (2)

日紫喜 光良

概要

1. はじめにー具体例からー

-----ここから-----

2. Javascript のプログラミング入門

-----次はどうする-----

3. 足りないものは借りてくるーJavascript のライブラリ

4. 仕事は人にやらせるーサーバーとブラウザの役割分担

今日の主題 : Javascript を使ってブラウザの上で動くプログラムを作る。

2. Javascript のプログラミング入門

Javascript は HP の構成要素を操作するために都合よくできている。HP の構成要素とは、HP に表示されるあらゆるモノ (オブジェクト, object) のことをいう。HP は、HTML (Hypertext Markup Language) という書式で書かれた文書なので、HP に表示されるものはすべて HTML 文書 (document) 上のオブジェクトである。そこで、HP の構造を Document Object Model (DOM) という。

つまり、Javascript できることとは、DOM の要素 (element) を操作することである。主な操作には、(a)生成、(b)取得、(c)消去、(d)イベント発生時の動作の命令、がある。

次のサンプルプログラム、音声ファイル、そして適当な小さな画像ファイル2つを同じフォルダに置き (音声と画像ファイルはプログラム中の名前に合わせる)、サンプルプログラムをブラウザ (Firefox か Chrome) で開き、ブラウザ上にカーソルを置くと、カーソルに画像が追従する。ある領域上では画像が変わり、音が出る。

```
1 | <!DOCTYPE html>
2 | <html>
3 | <META LANG=JA>
4 | <head>
5 | <script>
6 | var pos;
7 | window.onload = function() {
```

```
8 //document.body.div_sound.hidden = true;
9 var div = document.getElementById("div_sound");
10 div.hidden = true;
11 var cursor = document.createElement("img");
12 document.body.appendChild(cursor);
13 cursor.id = "my_cursor";
14 cursor.src = "./cursor.png";
15 cursor.style.position = 'absolute';
16 function move(cursor, pos) {
17     var x = pos.x;
18     var y = pos.y;
19     cursor.style.left = x + "px";
20     cursor.style.top = y + "px";
21 }
22 function setColor(cursor, pos) {
23     var x = pos.x;
24     var y = pos.y;
25     if ((100 <x)&&(x < 200)&&(100 < y)&&(y < 200)) {
26         cursor.src = "./cursor-2.png";
27         bark();
28     }
29     else if ((300 <x)&&(x < 400)&&(100 < y)&&(y < 200)) {
30         cursor.src = "./cursor-2.png";
31         bark();
32     }
33     else {
34         cursor.src = "./cursor.png";
35         stopBark();
36     }
37 }
38 function bark() {
39     document.getElementById("barksound").play();
40 }
41 function stopBark() {
42     var sound = document.getElementById("barksound");
43     sound.pause();
```

```

44     }
45     document.addEventListener("mousemove",function(ev) {
46         pos = {x: ev.clientX, y: ev.clientY};
47         //console.log(pos);
48         var cursor = document.getElementById("my_cursor");
49         //console.log(cursor);
50         move(cursor,pos);
51         setColor(cursor,pos);
52     });
53 }
54
55 </script>
56 </head>
57 <body>
58 <div id=div_sound>
59 <audio id="barksound" preload="auto">
60     <source src="./dog_barking1.mp3" type="audio/mp3">
61 </audio>
62 </div>
63 </body>
64
65 </html>

```

プログラム1 ここ掘れワンワン

インターネットでアクセス可能にした。

<http://hishikilab.sakura.ne.jp/project2014/sample01/index.html>

このプログラムの構造は、大きくみると次のようになっている。

```

1 <!DOCTYPE html>
2 <html>
3 <META LANG=JA>
4 <head>
5 <script>
  ...

```

```

55 | </script>
56 | </head>
57 | <body>
58 | <div id=div_sound>
59 | <audio id="barksound" preload="auto">
60 |   <source src="./dog_barking1.mp3" type="audio/mp3">
61 | </audio>
62 | </div>
63 | </body>
64 |
65 | </html>

```

このようにして、HTML のタグで、要素間の階層構造を表現している。

すなわち、次の記述によって、body 要素は div_sound という名前の div 要素を下位に（子として）持ち、その div 要素は、barksound という名前のサウンドプレーヤー要素 audio を下位構造（子要素）としてもっている。

```

57 | <body>
58 | <div id=div_sound>
59 | <audio id="barksound" preload="auto">
60 |   <source src="./dog_barking1.mp3" type="audio/mp3">
61 | </audio>
62 | </div>
63 | </body>

```

60 行目は、音源の mp3 ファイル dog_barking1.mp3 をこの HTML ファイルと同じディレクトリに置いてあることを表している。

音源は、小森平氏の次のページ（サイト）から取得した。

人間・動物・足音（無料効果音）

<http://taira-komori.jp.org/animals01.html>

動作の記述は 6～54 行目に書いてある。

この部分の構造は、大きく見ると、

```

7 | window.onload = function () {
  | ...
53 | }

```

となっている。

windowは、ウインドウオブジェクト、つまりページのことをいう。

.onloadとは、イベントハンドラ event handler の一種である。

イベントハンドラとは、オブジェクトの特定のイベントが発生したときに、オブジェクトに与えられるプロパティ property である。そしてそのプロパティが指し示すものは、一連のステートメント statement (命令文) をひとまとめにした、function () {...}; という関数である。

イベントハンドラ window.onload が指す命令の中身だが、まず、

```
9 | var div = document.getElementById("div_sound");
10 | div.hidden = true;
11 | var cursor = document.createElement("img");
12 | document.body.appendChild(cursor);
13 | cursor.id = "my_cursor";
14 | cursor.src = "./cursor.png";
15 | cursor.style.position = 'absolute';
```

のように、div_sound という名前のオブジェクトを変数 div として操作できるようにして (9行目)、これを画面に表示されなくする (hidden プロパティを true にする) (10行目)。次に、画像オブジェクトを生成し変数 cursor として操作できるようにして (11行目)、ページに所属させる (12行目)。このオブジェクトの名前を my_cursor に、また、用いる画像へのパスを ./cursor.png にする。(13, 14行目)。この画像オブジェクトの座標は、ページの左上隅を起点とした値とする (15行目)。

ここに示した、getElementById, createElement, appendChild は、要素を扱うための代表的なメソッド (ある種 (クラス) のオブジェクトを操作できる手段) である

document.getElementById("オブジェクト名称")	その名前の要素を返す
document.createElement("オブジェクトの種類")	その種類の要素を作る
親要素.appendChild(オブジェクト)	親要素の子にする

イベントハンドラ window.onload が指す命令の中身にはさらに次のようなものが含まれる。まず、move, setColor, bark, stopBark の、4種類の関数が定義される。

```
16 | function move(cursor, pos) {
    |     ここで関数 move が定義される。関数 move は変数 cursor が指す画像オブ
    |     ジェクトに、オブジェクトを指す変数 pos から取得できる x, y 座標を設定
```

```

    する。
21  }
22  function setColor(cursor,pos){
23      var x = pos.x;
24      var y = pos.y;
25      if ((100 <x)&&(x < 200)&&(100 < y)&&(y < 200)){
26          cursor.src = "./cursor-2.png";
27          bark();
28      }
29      else if ((300 <x)&&(x < 400)&&(100 < y)&&(y < 200)){
30          cursor.src = "./cursor-2.png";
31          bark();
32      }
33      else {
34          cursor.src = "./cursor.png";
35          stopBark();
36      }
37  }
38  function bark(){
39      document.getElementById("barksound").play();
40  }
41  function stopBark(){
42      var sound = document.getElementById("barksound");
43      sound.pause();
44  }

```

関数とは、

```
function 関数名 (引数) {命令};
```

の形で記述されるステートメントである。関数名は省略されることもある。

引数とは、関数に渡される変数のことである。つまり関数への入力である。複数の引数を、で区切って渡してもよい。また、引数がなくてもよい。その場合は括弧だけになる。

関数の出力を戻り値という。{...}内の命令の最後に

```
return オブジェクト;
```

の形で明示すれば、オブジェクトが戻り値になる。特に明示されなければ、boolean 値である true が戻り値になる。

関数の {...} の中が、処理の本体で、一つまたは複数のステートメントからなる。ステートメントは上から下へ順番に実行される。

条件によって実行するステートメントを分けたいときの方法の1つに、if ...else if ... else による制御がある。2か所の長方形の領域にカーソルが入ったときに、カーソルに追従する画像を取り替え、犬が吠える声を発する。それ以外の領域では音を止める。

```
22 | function setColor(cursor,pos){
23 |     var x = pos.x;
24 |     var y = pos.y;
25 |     if ((100 <x)&&(x < 200)&&(100 < y)&&(y < 200)){
26 |         cursor.src = "./cursor-2.png";
27 |         bark();
28 |     }
29 |     else if ((300 <x)&&(x < 400)&&(100 < y)&&(y < 200)){
30 |         cursor.src = "./cursor-2.png";
31 |         bark();
32 |     }
33 |     else {
34 |         cursor.src = "./cursor.png";
35 |         stopBark();
36 |     }
37 | }
```

bark と stopBark の定義は、別のところ（それぞれ 38~40, 41~44 行目）で行い、関数 setColor で呼び出している。

次に、ページ（ページの構成要素では最上位の document 要素）上で「マウス移動 (mousemove)」のイベントが生じた時に実行させるステートメントを定義する。

DOM 要素のイベント発生時に行うべき命令を定義するには、2つのアプローチがある。ひとつは、7行目で

```
windows.onload = function () {...};
```

としたように、イベントハンドラをオブジェクトに付けるやりかたである。

イベントハンドラの例を挙げる。

onload	読み込みが完了した時
onclick	マウスでクリックされたとき
onmousedown	マウスボタンが押されたとき
onmouseup	マウスボタンが離されたとき
onmouseover	マウスが要素に乗ったとき

参考：

イベントハンドラ (event handler) (Javascript プログラミング解説)

<http://so-zou.jp/web-app/tech/programming/javascript/event/handler/#no3>

もう1つのアプローチは、次のように、オブジェクトに対して、イベントリスナー event listener を定義することである。

```

45 | document.addEventListener("mousemove",function(ev){
46 |     pos = {x: ev.clientX, y: ev.clientY};
47 |     //console.log(pos);
48 |     var cursor = document.getElementById("my_cursor");
49 |     //console.log(cursor);
50 |     move(cursor,pos);
51 |     setColor(cursor,pos);
52 | });

```

イベントリスナーは、一般的には次のように定義される。

```

オブジェクト.addEventListener("イベントの種類", リスナー, false または true);

```

リスナーには、関数そのもの、または関数を指すもの（ふつうは関数の名前）が入る。関数にはイベントを表す引数（例えば ev とか event などの任意の名前でよい）を与えてもよい。関数の名前を入れた場合は、別に関数を定義することになり、この関数にイベントを表す引数を与える。false または true は省略可能で、デフォルト値は false である。

46 行目では、ページの左上隅を原点としたマウスの x 座標、y 座標からなるオブジェクトを pos という変数に収納している。ここで2点説明を加える。

まず、変数 pos の定義は既に6行目で終わっている。これは、変数 pos のスコープ scope（その名前が有効な範囲）が7~53行目全部に及ぶようにしたかったからである。var 宣言

を用いて定義された変数のスコープは、それよりも内側の () あるいは {} 内で、それよりも外側では有効ではない。

次に、`x, y` 座標のオブジェクトの構造は次のようになっている。

```
46 |     pos = {x: ev.clientX, y: ev.clientY};
```

一般的に、オブジェクトは、次のように定義される。

{属性 1: その値, 属性 2: その値, 属性 3: その値, ..., 属性 n: その値}

ただし、値はオブジェクトでも可能である。つまり、入れ子構造が可能である。

これを、Javascript Object Notation (JSON) という。

オブジェクトの属性の値を得るには、プログラムの 17, 18 行目のようにする。

```
16 |     function move(cursor, pos) {  
17 |         var x = pos.x;  
18 |         var y = pos.y;  
19 |         cursor.style.left = x + "px";  
20 |         cursor.style.top = y + "px";  
21 |     }
```

また、19, 20 行目では、カーソルオブジェクト `cursor` の位置を定義している。