

## SVG 資料第 8 回目 (その 1)

# WebStorage—localStorage

メディア専門ユニット I(SVG)

2016/6/13

# ブラウザでのデータの保存

第 8 回目 (その 1)

メディア専門ユニット I(SVG)

Web ページのデータの保存

localStorage の利用

やってみよう

- ▶ Web ページ上で作成したデータをそのままブラウザが動作しているコンピュータ上に保存することはセキュリティ上できない。
- ▶ ブラウザが管理する領域に保存することは可能
- ▶ 昔からあるのがクッキー
  - ▶ クライアントとサーバーの間で共有して常にクライアントからサーバーに提供
  - ▶ 小さなデータしか扱えない
  - ▶ サーバーとクライアントとのやりがあるたびに送られる
- ▶ HTML5 ではブラウザーのあるページ以降のページにだけ存在する `sessionStorage` とページが閉じられてもデータが存続できる `localStorage` が定義されている。

# バンジオ・ビンナの錯視図形

## 今回のサンプル

第 8 回目 (その 1)

メディア専門ユニット I(SVG)

Web ページのデータの保存

localStorage の利用

やってみよう

- ▶ 曲線で囲まれた部分は色が塗られていないが、その部分がうっすらと間の色で塗られているように見える
- ▶ 右側のテキストボックスに色を指定したのち、「設定」ボタンを押すと曲線の色がその色に設定される。

# バンジオ・ピンナの錯視図形

## HTML ファイル

第 8 回目 (その 1)

メディア専門ユニット I (SVG)

Web ページのデータの保存

localStorage の利用

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="pinna.js"></script>
8<link rel="stylesheet" type="text/css" href="pinna.css">
9<title>バンジオ・ピンナの錯視図形</title>
10</head>
11<body>
12  <h1 class="display">バンジオ・ピンナの錯視図形 (storage 不使用)</h1>
13  <div class="Cell">
14    <svg height="420" width="420">
15      <g id="Canvas" transform="translate(210,210)"/>
16    </svg>
17  </div>
18  <div class="Cell" >
19    <div><label for="color1">色 1</label>
20      <input type="text" id="color1" size="5"/></div>
21    <div><label for="color2">色 2</label>
22      <input type="text" id="color2" size="5"/></div>
23    <input id="SetColor" type="button" value="設定"></input>
24  </div>
25</body>
```

# バンジオ・ビンナの錯視図形

## HTML ファイル (解説)

第 8 回目 (その 1)

メディア専門ユニット I(SVG)

Web ページのデータの保存

localStorage の利用

やってみよう

前回の「SVG と HTML の間でデータを交換」と同じ構造

# バンジオ・ビンナの錯視図形

## CSS ファイル

第 8 回目 (その 1)

メディア専門ユニット I(SVG)

Web ページのデータの保存

localStorage の利用

やってみよう

```
1.display {  
2  font-size:25px;  
3}  
4.Cell {  
5  font-size:30px;  
6  display:inline-block;  
7  vertical-align:middle;  
8  padding-left:5px;  
9}  
10#color1, #color2 {  
11  font-size:25px;  
12  text-align: right;  
13}  
14#SetColor{  
15  font-size:25px;  
16  text-align:center;  
17}
```

前回の「SVG と HTML の間でデータを交換」と同じ

# バンジオ・ビンナの錯視図形

## JavaScript ファイル (1)

第 8 回目 (その 1)

メディア専門ユニット I(SVG)

Web ページのデータの保存

localStorage の利用

やってみよう

```
1let Canvas, C1, C2, Paths =[];
2window.onload = function(){
3  Canvas = document.getElementById("Canvas");
4  C1 = document.getElementById("color1");
5  C2 = document.getElementById("color2");
6  for(let i= 0; i<6;i++) {
7    Paths[i] = MKSVGElm(Canvas, "path", {"stroke-width": 6, "fill": "none"},{});
8  }
9  C1.value = "red";
10 C2.value = "green";
11 document.getElementById("SetColor").addEventListener("click", DrawFigs, true);
12 DrawFigs();
13}
```

# バンジオ・ビンナの錯視図形

## JavaScript ファイル (1)-解説

第 8 回目 (その 1)

メディア専門ユニット I(SVG)

Web ページのデータの保存

localStorage の利用

やってみよう

- ▶ 6 行目から 8 行目で 6 つの path 要素を作成し、配列に格納
- ▶ 9 行目と 10 行目で図形の色の初期値をテキストボックスの設定
- ▶ 11 行目で「設定」ボタンにクリックイベントの処理関数を設定
- ▶ 図形を描く関数を呼び出す (「設定」ボタンの処理関数と同じ)



# バンジオ・ビンナの錯視図形

## JavaScript ファイル (2)

第 8 回目 (その 1)

メディア専門ユニット I(SVG)

Web ページのデータの保存

localStorage の利用

やってみよう

```
14function DrawFigs() {
15  let W1=8, W2=4;
16  let Color1 = C1.value;
17  let Color2 = C2.value;
18  DrawFigure(150, 30, W1, W2, Color1, 0);
19  DrawFigure(144, 30, W1, W2, Color2, 1);
20  DrawFigure(80, 20, W1, W2, Color1, 2);
21  DrawFigure(86.5, 20, W1, W2, Color2, 3);
22  DrawFigure(14, 20, 0, 0, Color1, 4);
23  DrawFigure(10, 20, 0, 0, Color2, 5);
24}
25function DrawFigure(R, sR, W, W2, Color, No) {
26  let d = "M", i, Ang, R0;
27  for(i=0;i<720;i++) {
28    Ang= Math.PI*i/180/2;
29    R0=R+sR*(Math.cos(W*Ang)*Math.cos(W2*Ang));
30    d += `${R0*Math.cos(Ang)},${R0*Math.sin(Ang)} `;
31  }
32  SetAttributes(Paths[No], {"d": `${d}z`, "stroke": Color});
33}
```

# バンジオ・ビンナの錯視図形

## JavaScript ファイル (2) 解説

第 8 回目 (その 1)

メディア専門ユニット I(SVG)

Web ページのデータの保存

localStorage の利用

やってみよう

- ▶ DrawFigs() は図形全体を表示するための関数
  - ▶ 15 行目で波打つ曲線のパラメータを設定
  - ▶ 16 行目と 17 行目で曲線で使用する 2 種類の色をテキストボックスから読みだして変数に格納
  - ▶ 18 行目から 23 行目で個々の曲線を表示
- ▶ DrawFigure() は一つの曲線を描く関数
  - ▶  $0.5^\circ$  間隔で図形の設定するために角度を計算 (28 行目)
  - ▶ 曲線の原点からの距離を次の式で計算 (29 行目)

$$r = R + r_0 \cos(W\theta) * \cos(W_2\theta)$$

- ▶ これから  $\theta$  方向の  $x, y$  座標をもとめ、d に追加 (30 行目)

# バンジオ・ビンナの錯視図形

## JavaScript ファイル (Storage 使用)

第 8 回目 (その 1)

メディア専門ユニット I(SVG)

Web ページのデータの保存

localStorage の利用

やってみよう

```
1let Storage = window.localStorage;
2//let Storage = window.sessionStorage;
3let Canvas, C1, C2, Paths =[];
4window.onload = function(){
5  Canvas = document.getElementById("Canvas");
6  C1 = document.getElementById("color1");
7  C2 = document.getElementById("color2");
8  for(let i= 0; i<6;i++) {
9    Paths[i] = MKSVGElem(Canvas, "path", {"stroke-width": 6, "fill": "none"},{});
10 }
11 C1.value = Storage.C1?Storage.C1:"red";
12 C2.value = Storage.C2?Storage.C2:"green";
13 document.getElementById("SetColor").addEventListener("click", DrawFigs, true);
14 DrawFigs();
15}
16function DrawFigs() {
17  let W1=8, W2=4;
18  let Color1 = C1.value;
19  let Color2 = C2.value;
20  Storage.C1 = Color1;
21  Storage.C2 = Color2;
22  DrawFigure(150, 30, W1, W2, Color1, 0);
23  DrawFigure(144, 30, W1, W2, Color2, 1);
24  DrawFigure(80, 20, W1, W2, Color1, 2);
25  DrawFigure(86.5, 20, W1, W2, Color2, 3);
```

# バンジオ・ビンナの錯視図形

## JavaScript ファイル (Storage 使用) 解説

第 8 回目 (その 1)

メディア専門ユニット I(SVG)

Web ページのデータの保存

localStorage の利用

やってみよう

Storage を使用していないコードと異なる点は次の通り

- ▶ 11 行目と 12 行目で localStorage の値が設定されていればその値を、そうでなければデフォルト値を色のテキストボックスの値に設定
- ▶ 「設定」ボタンが押されたとき、テキストボックスの値を localStorage に保存
- ▶ 29 行目以下には前と同じ関数 DrawFigure が記述されている (表示が欠けている)。追加されている部分は 20 行目と 21 行目で、この関数が呼び出されたときに、Storage にテキストボックスの値を保存

次のことを確かめよ。

- ▶ ブラウザをいったん閉じてから再度ページを開くと、最後に設定された値がテキストボックスに表示
- ▶ localStorage の値を確認
- ▶ コンソールで `sessionStorage.test=1;` のように適当なキーで数を設定し、その後、値を読み出すとどうなるか確認する。設定する値として配列やオブジェクトでも確認すること (次のビデオを見る前に確かめておくことを強く勧める)。
- ▶ 新しいフォルダを作成し、そのフォルダに必要なファイルをコピーして、それを表示させたときにどうなるか
- ▶ 9 行目のコメントを外したらときに上と同じことを行って同じ結果になるか確認し、localStorage と sessionStorage の違いを理解する。