

## SVG 資料第 8 回目 (その 1)

# JSON

メディア専門ユニット I(SVG)

2016/6/13

JSON の情報は [www.json.org/json-ja.html](http://www.json.org/json-ja.html) で得られる。

- ▶ JSON は JavaScript Object Notation の略
- ▶ JavaScript のオブジェクトリテラルを定義する文字列の形式をまねたもの
  - ▶ 全体を { と } で囲む
  - ▶ 各項目はキーと値のペアで、間を, でつなぐ。
  - ▶ キーは文字列で、"で挟む
  - ▶ 値はオブジェクト、配列、数、文字列、true、true、false、null が可能

この定義から無条件に JavaScript のオブジェクトリテラルがそのまま JSON 形式として正しいとは限らない。

# JSON オブジェクトの利用法

第 8 回目 (その 1)

メディア専門ユニット I(SVG)

JSON

(配布資料 160 ページ)

```
1>A={x:10,y:20,add:function(){return(this.x+this.y);}}
2Object {x: 10, y: 20}
3>A.add()
430
5>B=JSON.stringify(A);
6 '{"x":10,"y":20}'
7>typeof B;
8 "string"
9>JSON.parse(B);
10▼ Object
11     x: 10
12     y: 20
13     ►__proto__: Object
```

- ▶ 1 行目でオブジェクトを定義。メソッドとして `add()` があり、2 つのメンバー `x` と `y` を加えた値を返す (3 行目)
- ▶ 5 行目でオブジェクトを JSON 形式に変換。この中にはメソッドの `add` が含まれていない。
- ▶ 7 行目では変換されたものが文字列であることを確認している。
- ▶ 9 行目では変換後の文字列をオブジェクトに変換している。

# Storage に配列データを保存 (1)

第 8 回目 (その 1)

メディア専門ユニット I(SVG)

JSON

```
1let Storage = window.localStorage;
2//let Storage = window.sessionStorage;
3window.onload = function(){
4  let Canvas = document.getElementById("Canvas");
5  let Cs = document.querySelectorAll('input[type="text"]');
6  let Paths = new Array(6).fill(1).map(function() {
7    return MKSVGElm(Canvas, "path", {"stroke-width": 6, "fill": "none"},{});
8  });
9  let Colors = Storage.C?JSON.parse(Storage.C):["red", "green"];
10 Colors.forEach(function(C,i){Cs[i].value = C;});
11 document.getElementById("SetColor").addEventListener("click", DrawFigs, true);
12 DrawFigs();
```

## Storage に配列データを保存 (2)

第 8 回目 (その 1)

メディア専門ユニット I(SVG)

JSON

```
13 function DrawFigs() {
14     let W1=8, W2=4;
15     Colors.forEach(function(C,i, Colors){return Colors[i] = Cs[i].value;});
16     Storage.C = JSON.stringify(Colors);
17     [[150, 30, W1, W2, Colors[0]], [144, 30, W1, W2, Colors[1]],
18      [80, 20, W1, W2, Colors[0]], [86.5, 20, W1, W2, Colors[1]],
19      [14, 20, 0, 0, Colors[0]], [10, 20, 0, 0, Colors[1]]].forEach(
20     function(Param, No) {
21         let [R, sR, W, W2, Color] = Param;
22         let d = "M" +
23             Array(720).fill(0).map(function(dummy,i) {
24                 let Ang= Math.PI*i/180/2;
25                 let R0=R+sR*(Math.cos(W*Ang)*Math.cos(W2*Ang));
26                 return `${R0*Math.cos(Ang)},${R0*Math.sin(Ang)}';
27             }).join(" ")+"z";
28         SetAttributes(Paths[No], {"d": d, "stroke": Color});
29     })
30 }
31 }
```