

SVG 資料第 9 回目 (その 3)

サーバーで SVG ファイルを作成

メディア専門ユニット I(SVG)

2016/6/20

サーバーで SVG ファイルを作成 (1)–クライアント側

第 9 回目 (その 3)

メディア専門ユニット I (SVG)

クライアント側の
プログラム

サーバー側のプログラム

サーバー側のメインプログラム

サーバー側のユーティリティ関数

やってみよう

```
1<!DOCTYPE html>
2<html>
3  <head>
4    <meta charset="utf-8"/>
5    <style type="text/css">
6      .head {
7        font-size:20px;
8      }
9      .Cell {
10       text-align:center;
11     }
12   </style>
13   <title>正 n 角形を描く</title>
```

▶ クライアント側の初めの部分

サーバーで SVG ファイルを作成 (2)-クライアント側

第 9 回目 (その 3)

メディア専門ユニット I(SVG)

クライアント側の
プログラム

サーバー側のプロ
グラム

サーバー側のメインプロ
ラム

サーバー側のユーティティ関数

やってみよう

```
14 </head>
15 <body>
16   <form name="Test" method="PUT" action="./svg-polygon.php">
17     <div class="Cell head">正 n 角形を描く</div>
18     <div class="Cell">
19       <label for="N">n</label><input type="text" name="N" size="2"/>
20     </div>
21     <div class="Cell">
22       <input type="SUBMIT" value="実行"/>
23     </div>
24   </form>
25 </body>
26</html>
```

サーバーで SVG ファイルを作成 (2)-クライアント側解説

- ▶ 9 行目から 25 行目でデータ入力のフォームを定義
- ▶ 属性 `method` が `PUT` で属性 `action` が `./svg-polygon.php`
- ▶ 12 行目は属性 `class` として `Cell` と `head` の二つを指定
- ▶ 16 行目で辺の数を入れるテキストボックスを定義
 - ▶ 16 行目の `<label>` 要素の属性 `for` は属性 `name` がその値の要素に対応するものを意味する
 - ▶ その後に属性 `name` が `N` であるテキストボックスを置いている
- ▶ 21 行目に「実行」の文字のあるボタンを定義
- ▶ 属性 `type` が `submit` となっているので `<form>` 要素の属性 `action` で指定された URL に飛ぶ

第 9 回目 (その 3)

メディア専門ユニット I(SVG)

クライアント側のプログラム

サーバー側のプログラム

サーバー側のメインプログラム

サーバー側のユーティリティ関数

やってみよう

サーバーで SVG ファイルを作成 (3)-サーバー側

第 9 回目 (その 3)

メディア専門ユニット I(SVG)

クライアント側の
プログラム

サーバー側のプロ
グラム

サーバー側のメインプログラ
ム

サーバー側のユーティリティ関数

やってみよう

```
1<?php
2require_once('svg-func.php');
3function drawPath() {
4    $N = $_GET['N'];
5    $pathString = MaxSize . ',0';
6    for( $i = 1; $i < $N; $i++) {
7        $Angle = M_PI*(2.0*$i/$N);
8        $pathString .=
9            sprintf(' %.4f,%.4f', MaxSize*cos($Angle), MaxSize*sin($Angle));
10    }
11    outputTag('polygon',
12        array('points'=>$pathString, 'fill'=>'blue',
13            'stroke-width'=>4, 'stroke'=>'red'));
14}
15define("MaxSize",200);
16header("Content-type: image/svg+xml");
17printHeader('100%', '100%');
18outputTag('g', array('transform'=>'translate(210,210)'));
19drawPath();
20closeSVG();
21?>
```

サーバーで SVG ファイルを作成 (3)-サーバー側 (解説)

第 9 回目 (その 3)

メディア専門ユニット I(SVG)

クライアント側の
プログラム

サーバー側のプログラム

サーバー側のメインプログラム

サーバー側のユーティリティ関数

やってみよう

- ▶ 2 行目で外部ファイルを 1 度だけ読み込むための関数 `require_once` を用いて `svg-func.php` を読み込み
- ▶ 3 行目から 14 行目で正多角形の頂点を計算して文字列として戻す関数 `drawPath()` を定義
- ▶ 4 行目でフォームに入力された値を変数に格納。サーバーへのデータの渡し方が PUT だったのでスーパーグローバル変数 `$_GET` にある。

サーバーで SVG ファイルを作成 (3)-サーバー側

- ▶ 5 行目に現れるシンボル MaxSize は 15 行目の `define()` で定義されている定数
- ▶ 角度が 0° の点の位置で変数 `$pathString` を初期化 (5 行目)
- ▶ 6 から 10 行目で各頂点の座標を計算 (for で繰り返し)
- ▶ 7 行目で中央からの各頂点への角度をラジアンで計算。
`M_PI` は PHP における円周率を表す定数
- ▶ 9 行目で計算した頂点の座標を今までに求めた頂点の座標の後に PHP の文字列をつなぐ演算子 `.` でつなぐ
- ▶ `sprintf` の戻り値は C 言語の `printf` のように書式を与えて、データを文字列に変換したもの
- ▶ ここでは書式を指定しないと小数点以下の桁が多くなるのを防ぐために使用 (ここでは小数点以下が 4 桁)
- ▶ 11 行目から 13 行目の関数 `outputTag` は指定された要素と属性を持つ SVG の要素を文字列として出力
- ▶ 12 行目の `array` 関数は指定した要素を持つ配列を定義
- ▶ 連想配列 (JavaScript のオブジェクトリテラルのようなもの) で与える

サーバーで SVG ファイルを作成 (3)-サーバー側

第 9 回目 (その 3)

メディア専門ユニット I(SVG)

クライアント側の
プログラム

サーバー側のプロ
グラム

サーバー側のメインプログラ
ム

サーバー側のユーティリティ関数

やってみよう

- ▶ 15 行目で定数 MaxSize を定義
- ▶ 16 行目でこれから送るデータの形式 (MIME) を指定 (指定がないと text/html となる)
- ▶ ここでは SVG なので image/svg+xml を指定
- ▶ SVG ファイルの先頭部分を各関数 printHeader を呼び出す。引数は画像の幅と高さ (17 行目)
- ▶ <g>要素の出力 (18 行目)
- ▶ 正 n 角形の出力
- ▶ 閉じていない要素を閉じる関数の呼び出し (20 行目)

サーバーで SVG ファイルを作成 (4)–サーバー側 printHeader()

第 9 回目 (その 3)

メディア専門ユニット I(SVG)

クライアント側の
プログラム

サーバー側のプロ
グラム

サーバー側のメインプログラ
ム

サーバー側のユーティリティ関数

やってみよう

```
1<?php
2function printHeader($width, $height) {
3    print<<<_EOF_
4<?xml version="1.0" encoding="utf-8" ?>
5    <svg xmlns="http://www.w3.org/2000/svg"
6        xmlns:xlink="http://www.w3.org/1999/xlink"
7        height="$height" width="$width">
8    <title>正 n 角形の表示</title>
9_EOF_;
10    $GLOBALS['setTags'] = array("svg");
11}
```

- ▶ SVG ファイルのヘッダー部分を出力する関数
- ▶ 引数は SVG 画像の幅と高さ
- ▶ ヒアドキュメントとして記述
- ▶ 出力された要素をスタックとして保存するためのグローバル変数\$setTags を svg で初期化
- ▶ PHP では関数内の変数はすべてローカル。グローバル変数の参照にはスーパーグローバル\$GLOBALS 内に格納

サーバーで SVG ファイルを作成 (4) – サーバー側

outputSpaces() と outputTags()

第 9 回目 (その 3)

メディア専門ユニット I(SVG)

クライアント側の
プログラム

サーバー側のプロ
グラム

サーバー側のメインプロ
ラム

サーバー側のユーティティ関数

やってみよう

```
12function outputSpaces($n) {
13  while( $n-- > 0 ) print '  ';
14}
15function outputTag($tagName, $attributes, $close=false){
16  outputSpaces(count($GLOBALS['setTags']));
17  print "<$tagName ";
18  foreach($attributes as $key => $value ) {
19    print $key . '="' . $value . '" ';
20  }
21  if($close) {
22    print '/';
23  } else {
24    array_push($GLOBALS['setTags'],$tagName);
25  }
26  print ">\n";
27}
```

サーバーで SVG ファイルを作成 (4)-サーバー側

- ▶ 関数 `outputSpaces()` は引数の 2 倍の空白を出力する (出力ファイルの要素を読みやすくするためのインデント)。
- ▶ 関数 `outputTags` は要素名、属性と属性値のペアからなる連想配列、属性をそこで閉じるかのフラグの 3 つを引数にとる
- ▶ 最後の引数は省略可で、デフォルト値は `false`(閉じる)
- ▶ 閉じていない要素名のリスト `$setTags` の長さ `count()` を利用してインデントのための空白を出力 (17 行目)
- ▶ 17 行目で開始要素を出力
- ▶ 18 行目から 20 行目で属性値と属性名を `foreach` を用いて出力
- ▶ 3 番目の仮引数が `true` であればそこで要素の終了を意味する `</` を出力 (23 行目)
- ▶ そうでなければ今出力した要素をグローバル変数 `$setTags` の最後の要素として付け加える (`array_push()` を利用)
- ▶ 最後に要素の終了を示す `>` を出力

サーバーで SVG ファイルを作成 (4)-サーバー側

第 9 回目 (その 3)

メディア専門ユニット I(SVG)

クライアント側のプログラム

サーバー側のプログラム

サーバー側のメインプログラム

サーバー側のユーティリティ関数

やってみよう

```
28function closeTag() {
29  outputSpaces(count($GLOBALS['setTags'])-1);
30  print '</' . array_pop($GLOBALS['setTags']) . ">\n";
31}
32function closeSVG() {
33  while (count($GLOBALS['setTags'])>0) {
34    closeTag();
35  }
36}
37?>
```

- ▶ 28 行目から 31 行目は閉じられていない要素の終了タグを出力する関数を定義
- ▶ 29 行目でインデントを一つ下げる分の空白を出力
- ▶ 30 行目では array_pop() を用いて最後の登録された閉じられていない要素の終了要素を出力
- ▶ 32 行目から 35 行目ではファイルの最後で閉じられていない要素をすべて出力する関数 closeSVG を定義
- ▶ グローバル変数 \$setTags の長さが 0 になるまで closeTags() を呼び出している。

サーバーで SVG ファイルを作成

第 9 回目 (その 3)

メディア専門ユニット I(SVG)

クライアント側の
プログラム

サーバー側のプロ
グラム

サーバー側のメインプロ
グラム

サーバー側のユーティリティ関数

やってみよう

- ▶ HTML リスト 8.1 と PHP リスト 8.2、8.3 を作成して正 n 角形の SVG ファイルを返すファイルができることを確認
- ▶ HTML のファイル名は `svg-polygon.php` とすること
- ▶ アドレスバーに
`http://localhost/svg-polygon.php?N=10` と打ち込んだ場合

やってみよう

第 9 回目 (その 3)

メディア専門ユニット I(SVG)

クライアント側の
プログラム

サーバー側のプロ
グラム

サーバー側のメインプロ
グラム

サーバー側のユーティリティ関数

やってみよう

- ▶ 正 n 角形の SVG ファイルを返すファイルができることを確認
- ▶ アドレスバーに
`http://localhost/svg-polygon.php?N=10` と打ち込んだ場合どうなるか
- ▶ 212 ページの header 関数を追加して、SVG ファイルが保存できることを確認

次のことを調べる。

- ▶ MIME とはなにか。