

## SVG レファレンス

SVG 1.0 の仕様

W3C Candidate Recommendation 02 Nov. ¥ 2000 より

	要素名	要素の内容
コンテナ機能など	svg title desc g defs symbol use tref	SVG ドキュメントの宣言 タイトルの記述 コメントの記述 表示要素のグルーピング 参照要素のためのコンテナ宣言 コンテナの中にある複数の表示要素のグルーピング化 defs の中にある表示用コンテナの要素参照 defs の中にあるテキスト用コンテナの要素参照
基本機能	rect circle ellipse line polyline polygon path text tspan textPath image a	四角形(角の丸いものも含む)の描画 円の描画 楕円の描画 線の描画 折れ線の描画 多角形の描画 汎用的な描画要素 テキストの表示 テキストの部分修飾 path で指定された線上へのテキスト表示 イメージの表示 リンクの設定
アニメーション機能	animate set animateMotion animateColor animateTransform	描画要素のアニメーション指定 アニメーションのための XML や CSS の属性値の設定変更 path に沿って描画要素の移動 描画要素の色変更 transform による描画要素の移動・変形

フィルタ機能は省略してあります。

## 注意点

SVG 中に日本語を埋め込む場合、そのファイルはユニコードで保存しなければならない。

## <svg> SVG ドキュメントの宣言

SVG ドキュメントはこの要素の中に記述します。この要素の宣言に先立ち、DOCTYPE の宣言も必要になります。また SVG ドキュメントの宣言に留まらず、表示エリアの定義を行います。この例では幅 12cm、高さ 7cm を SVG の表示エリアとします。

```
<?xml version="1.0"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 20001102//EN"
    "http://www.w3.org/TR/2000/CR-SVG-20001102/DTD/svg-20001102.dtd">
<svg width="12cm" height="7cm">
    . . . . .
</svg>
```

## <title>、<desc> タイトルやコメントの記述

HTML や XML でのメッセージの記述は、<!-- -->の中に行います。SVG も同様の記述でメッセージが記述できます。SVG では、これ以外に title と desc が用意されています。これらの要素は、属性として名前空間が指定できるため、外部の XML 概念を記述することができます。title や desc 要素で括られた内容に対して SVG は、何の作用もしません（表示処理の対象外）。

```
<desc xmlns:mydoc="http://example.org/mydoc">
    <mydoc:title>This is an example SVG file</mydoc:title>
    <mydoc:para>The global description uses markup from the
    <mydoc:emph>mydoc</mydoc:emph> namespace.</mydoc:para>
</desc>
```

## <g> 表示要素のグルーピング

この要素は、HTML での DIV 要素と同じ働きをします。この要素で括られた範囲にある描画要素に対して属性を与え、一つのグループとして扱えるようにします。次の例では、2つの rect 要素に対して、fill:red のスタイル属性を与えることができます。

```
<g id="group1" style="fill:red">
    <rect x="1cm" y="1cm" width="1cm" height="1cm" />
    <rect x="3cm" y="1cm" width="1cm" height="1cm" />
</g>
```

## <defs>、<use> 参照要素のためのコンテナ宣言とその参照

defs 要素はいくつかの場所で参照される描画要素をテンプレートとして定義し、use 要素によってそのテンプレートが参照されます。この例では、rect 要素の id が参照元、参照側のインターフェイスになります。この rect では、高さが定義され、参照された場合、表示座標属性が付加されて、四角形を表示されます。

```
<defs>
    <rect id="MyRect" width="60" height="10"/>
</defs>
<use x="20" y="10" xlink:href="#MyRect" />
```

### <symbol> 複数の表示要素のテンプレート宣言

複数の描画要素を一塊として扱えるよう、defs 要素の中でテンプレート宣言を行います。この例では、2つの赤丸を use 要素で指定する座標に表示します。

```
<defs>
  <symbol id="MyCircle" style="fill:red">
    <circle cx="10" cy="10" r="5"/>
    <circle cx="50" cy="10" r="5"/>
  </symbol>
</defs>
<use x="20" y="10" xlink:href="#MyCircle" />
```

### <tref> テキストコンテナの宣言

上記では描画要素のテンプレート化しました。テキストも同様にテンプレート化できます。テンプレート化したテキストの参照は tref 要素によって行います。

```
<defs>
  <text id="Confidential">極秘資料</text>
</defs>
<text x="1cm" y="2cm" style="font-size:30pt; fill:red; font-family:'MS Gothic'">
  <tref xlink:href="#Confidential"/>
</text>
```

### <rect> 四角形の描画

四角形の描画は rect 要素によって行います。指定する属性は、開始座標、幅、高さ、それにスタイル情報です。この例では縁が幅4のスカイブルーで、中がグリーンの大きな四角形を描画します。rx、ryによって角を丸くすることもできます。

```
<rect x="5" y="5" width="690" height="490"
  style="fill:green; stroke:skyblue; stroke-width:4" />
```

### <circle> 円の描画

円の描画は circle 要素によって行います。指定する属性は円の中心座標、半径、それにスタイル情報です。この例では半径80の青丸を(300,100)を中心として描きます。

```
<circle cx="300" cy="100" r="80" style="fill:blue" />
```

### <ellipse> 楕円の描画

楕円の描画は ellipse 要素によって行います。指定する属性は、楕円の中心座標、x径、y径、それにスタイル情報です。この例ではx径80、y径50のピンクの楕円を(150,200)を中心として描きます。

```
<ellipse cx="150" cy="200" rx="80" ry="50" style="fill:pink" />
```

### <line> 線の描画

線の描画は line 要素によって行います。指定する属性は、開始座標、終了座標、それにスタイル属性です。この例

では(200,10)から(10,200)まで赤い線を描きます。

```
<line x1="200" y1="10" x2="10" y2="200" style="stroke:red" />
```

#### <polyline> 折れ線の描画

折れ線の描画は polyline 要素によって行います。指定する属性は、結ぶ点の座標列とスタイル情報です。この例では points で指定された座標列を線幅 3 の緑色線を描きます。

```
<polyline points="10,10 20,50 80,100 140,40 200,200" style="fill:none; stroke:green; stroke-width:3" />
```

#### <polygon> 多角形の描画

多角形の描画は polygon 要素によって行います。指定する属性は、多角形頂点の座標列とスタイル情報です。この例では points で指定された座標を多角形の頂点とし、線幅 3 で紺色の縁取りで、スカイブルーで塗られた多角形を描きます。

```
<polygon points="210,100 320,150 380,100 340,240 300,200" style="fill:skyblue; stroke:navy; stroke-width:3" />
```

#### <path> 汎用的な描画要素

この要素は汎用的な描画要素で、線の描画、多角形の描画、曲線（二次ベジエ、三次ベジエ、円弧）の描画ができます。

##### 線を描く

```
<path d="M 100 100 L 200 200 L 200 100"/>
<path d="M 100,100 L 200,200 L 200,100"/>
<path d="M 100 100 L 200 200 200 100"/>
<path d="M100 100L200 200 200 100"/>
```

上記のコードはいずれも (100,100)から(200,200)を通り、(200,100)まで線を描きます。

```
<path d="M 100 100 L 200 200 v -100"/>
```

このコードも同じように表示します。V は垂直方向、h は水平方向の移動を示します。

```
<path d="M 100 100 L 200 200 L 200 100 z"/>
```

このコードは(100,100)から(200,200)、(200,100)を通る閉領域を描きます。

##### 曲線を描く

##### 三次ベジエ曲線

```
<path d="M 100,200 C 100,100 250,100 250,200"/>
```

このコードは、(100,200)から(250,200)まで、(100,100)と(250,100)を制御点として、三次ベジエ曲線を描きます。

```
<path d="M 100,200 C 100,100 250,100 250,200 S 400,300 400,200"/>
```

上記の曲線に続き、(250,200)から(400,200)まで、同じパターンで一つ目の制御点を想定し、もう一つ制御点(400,300)から生成する三次ベジェ曲線を描きます。

#### 二次ベジェ曲線

```
<path d="M200,300 Q400,50 600,300 T1000,300"/>
```

(200,300)から(400,50)を通り(600,300)まで二次ベジェ曲線を描きます。

この曲線と同じパターンで(1000,300)まで二次ベジェ曲線を描きます。

#### 円弧を描く

```
<path d="M 125, 75 A 100,50 0 0 0 225,125"/>
```

(125,75)から(225,125)まで、長径 100,50 の楕円で、小さい弧でマイナス方向の円弧を描きます。

0 0 0 はそれぞれ、回転角、large-arc-flag、sweep-flag である。large-arc-flag が 0 のとき小さい方の円弧、1 のとき大きい方の円弧が指定される。sweep-flag が 0 のときマイナス方向の円弧、1 のときプラス方向の円弧が指定される。

#### <text> テキストの表示

テキストの表示は text 要素によって行います。指定する属性は、表示位置、スタイル情報で、子供要素として、表示するテキストを記述します。この例では MS ゴシックのフォントサイズ 16 で“おはよう”を(10,10)の位置に表示します。

```
<text x="10" y="10" style="font-size:16; font-family:'MS Gothic'">おはよう</text>
```

#### <tspan> テキストへの部分修飾

テキストの一部を修飾するとき、tspan 要素を用います。指定する属性は表示位置を x,y で指定し、修飾したい属性をスタイル属性によって指定します。この例は“東京都新宿区河田町”の中で、“新宿区”を緑色し、他の文字列は黒色で表示します。スタイル情報は tspan 要素の中に留まり、それ以降は前段の要素のスタイル情報が継承されます。

```
<text>東京都  
    <tspan style="fill:green">新宿区</tspan>  
    河田町  
</text>
```

下のコードのように指定すると、テキストの 6 文字を x で指定する間隔で表示します。

```
<tspan x="10 20 30 40 50 60" y="50">東京都新宿区</tspan>
```

このコードは“新宿区”だけを 5 ドット上に書くような指定になっています。その指定は dx,dy 属性で行います。

```
<text>東京都  
    <tspan dy="-5" >新宿区</tspan><tspan dy="5"/>  
    河田町  
</text>
```

#### <textPath> テキストの曲線上への表示

テキストを指定した曲線上に乗せて表示するとき、textPath 要素によって行います。指定する属性は、defs 要素で指定した曲線情報を xlink:href で参照するだけです。SVG はユニコードをベースにしているため、記述する日本語もユニコードでなければ表示されません。また、フォントの種類は、Windows では MS ゴシック、Mac では osaka フォントに限られます。

```
<defs>
  <path id="MyPath" d="M 100 200 C 200 100 300 0 400 100
    C 500 200 600 300 700 200 C 800 100 900 100 900 100" />
</defs>
<text style="font-family:'MS Gothic'; font-size:42.3333; fill:blue">
  <textPath xlink:href="#MyPath">
    私の名前は大坂哲司です。
  </textPath>
</text>
```

#### <image> イメージの表示

イメージの表示は image 要素によって行います。指定する属性は、表示位置、表示領域の幅、高さ、それに xlink:href で参照ファイルを指定します。この例では表示位置(100,100)から表示領域 400,300 に、参照ファイルを表示します。

```
<image x="100" y="100" width="400" height="300"
  xlink:href="http://localhost/duke.gif" />
```

#### <a> リンクの埋め込み

リンクは<a>要素によって行います。参照するリンク先は xlink:href で指定します。<a>要素で括られたテキストや図形をクリックすると、指定のリンク先に飛んでいきます。この例では赤丸をクリックすると、W3C のサイトへ行きます。

```
<a xlink:href="http://www.w3.org">
  <circle cx="250" cy="150" r="10" style="fill:red"/>
</a>
```

#### <animate> 描画要素のアニメーション指定

描画要素へのアニメーション指定は<animate>によって行います。指定する属性は次の通りです。attributeType はアニメーションさせる属性の種類を指定します。指定は "XML"、"CSS"、と "auto" です。"XML" は XML の属性を示し、"CSS" は CSS のスタイル属性を示し、"auto" は両者の自動判定を示します。attributeName はアニメーションさせる属性名を指定します。この属性に対して、具体的な変量は from、to で指定します。begin で開始時間、dur で表示時間、fill で表示終了後の状態を指定します。fill には表示状態を残す "freeze" と非表示にする "remove" があります。repeatCount で表示回数を指定します。無限に繰り返す場合、repeatCount="indefinite" とします。この例では、幅 300、高さ 100 の黄色い四角形に対して、5 秒間で幅を 300 から 800、高さを 100 から 300 と変化させる指定を行っています。

```
<rect x="300" y="100" width="300" height="100" style="fill:rgb(255,255,0)" >
  <animate attributeName="width" attributeType="XML"
```

```

        begin="0s" dur="5s" fill="freeze" from="300" to="800" />
    <animate attributeName="height" attributeType="XML"
        begin="0s" dur="5s" fill="freeze" from="100" to="300" />
</rect>

```

<set>                   アニメーションのためのパラメータ設定

描画要素の非数値属性やスタイル属性の変更は<set>によって行います。指定する属性は<animate>とほぼ同じです。この例では、半径 5 の緑色円を非表示でセットします。SVG ドキュメント表示から 6 秒後、この円を表示し、半径を 5 から 80 まで 5 秒間で表示します。

```

<circle cx="100" cy="150" r="5" style="fill:green; visibility:hidden">
    <set attributeName="visibility" attributeType="CSS" to="visible"
        begin="6s" dur="5s" fill="freeze" />
    <animate attributeName="r" attributeType="XML"
        begin="6s" dur="5s" fill="freeze" from="5" to="80"/>
</circle>

```

<animateMotion>           path に沿った描画要素の移動

描画要素を指定した path 上で移動させる場合<animateMotion>によって行います。指定する属性は<animate>の属性に、path の記述と描画要素に対する回転の指定が追加されます。この例では、テキストの初期位置を (1000,280) に置き、水平方向に -2000 までを 3 秒で移動させます。この path は単純ですが、複雑な曲線を指定すると、その上を巧みに移動します。

```

<text style="font-family:'MS Gothic'; font-size:60; fill:blue">
    私の名前は<tspan style="fill:green">大坂</tspan>です。
    <animateMotion dur="3" repeatCount="indefinite"
        path="M 1000 280 h-2000" rotate="0" />
</text>

```

<animateColor>           描画要素の色変更

指定し時間の中での色変は<animateColor>によって行います。指定する属性は<animate>と同じです。色の指定は value によって行うこともできます。value="yellow;orange;yellow;" と指定すると、黄色からオレンジ色から黄色と変化させることができます。

```

<text style="font-family:'MS Gothic'; font-size:60; fill:blue">
    私の名前は<tspan style="fill:green">大坂</tspan>です。
    <animateMotion dur="3" repeatCount="indefinite"
        path="M 1000 280 h-2000" rotate="0" />
    <animateColor attributeName="fill" attributeType="CSS"
        from="rgb(0,0,255)" to="rgb(255,0,0)"
        dur="3s" repeatCount="indefinite" />
</text>

```

<animateTransform> transform による描画要素の移動・変形

```
<animateTransform attributeName="transform" attributeType="XML"
  type="rotate" from="-30" to="0" begin="3s" dur="6s" fill="freeze" />
<animateTransform attributeName="transform" attributeType="XML"
  type="scale" from="1" to="3" additive="sum"
  begin="3s" dur="6s" fill="freeze" />
```