

# SVG 資料第 5 回目 (その 2)

## 初期化で要素を作成と 自前のアニメーション

メディア専門ユニット I(SVG)

2016/5/23

- ▶ 円が直線上を回転しながら移動するとき、円上の一点が描く軌跡
- ▶ 円の半径を  $r$ 、回転した角を  $\theta$  とするとき、点の位置は次の式で与えられる

$$x = r(\theta - \sin \theta)$$

$$y = r(1 - \cos \theta)$$

- ▶ ここでは  $\theta$  の値を小さく変化させてその間の直線として<path>要素として描く

## 数学関数

Math オブジェクトのプロパティまたはメソッドとして記述

名称	種類	説明
E	定数	自然対数の底 (2.71828182...)
PI	定数	円周率 ( $\pi = 3.141592...$ )
abs(x)	関数	$x$ の絶対値
ceil(x)	関数	$x$ 以上の整数で最小な値を返す。
cos(x)	関数	余弦関数 $\cos x$
exp(x)	関数	指数関数 $e^x$
floor(x)	関数	$x$ の値を超えない整数
log(x)	関数	自然対数 $\log x$
max([x1, x2, ..., xN])	関数	与えられた引数の最大値
min([x1, x2, ..., xN])	関数	与えられた引数の最小値
pow(x, y)	関数	指数関数 $x^y$
random()	関数	0 と 1 の間の擬似乱数を返す
round(x)	関数	$x$ の値を四捨五入する
sin(x)	関数	正弦関数 $\sin x$
sqrt(x)	関数	平方根を求める。 $\sqrt{x}$
tan(x)	関数	正接関数 $\tan x$

第 5 回目 (その 2)

メディア専門ユニット I(SVG)

初期化で要素を作成

自前でアニメーションを作成

# サイクロイドコード (1)

第 5 回目 (その 2)

メディア専門ユニット I(SVG)

初期化で要素を作成

自前でアニメーションを作成

```
1<?xml version="1.0" encoding="UTF-8" ?>
2<svg xmlns="http://www.w3.org/2000/svg"
3    height="100%" width="100%">
4<title>サイクロイド</title>
5<script type="text/ecmascript">
6//
7let R = 50;
8window.onload = function() {
9    let Angle, rad, pX, pY, d = "M";
10    for( Angle=0; Angle&lt;= 360; Angle++) {
11        rad = Angle/180*Math.PI;
12        pX = R*(rad-Math.sin(rad));
13        pY = R*(-1+Math.cos(rad));
14        d += `${pX},${pY} `;
15    }
16    document.getElementById("cycliod").setAttribute("d",d);
17}
18//]]&gt;
19&lt;/script&gt;</pre></div>
```

## サイクロイド-コード (1)-解説

第 5 回目 (その 2)

メディア専門ユニット I(SVG)

初期化で要素を作成

自前でアニメーションを作成

- ▶ 7 行目で円の大きさを設定
- ▶ 変数  $d$  はサイクロイドの形状を示す<path>要素の属性  $d$  に設定するための値を保持する。"M"で初期化
- ▶ 角度を 0 から 360 に設定してそれに対する位置を計算 (10 行目から 15 行目)
  - ▶ 11 行目で角度をラジアンに変換。Math.PI は円周率を表す。
  - ▶ 12 行目で  $x$  座標の位置を、13 行目で  $y$  座標の位置をそれぞれ計算
  - ▶ 計算した結果を変数  $d$  に追加 (14 行目)
- ▶ 16 行目でサイクロイドを表示する<path>要素に属性  $d$  を設定

## サイクロイドコード (2)

第 5 回目 (その 2)

メディア専門ユニット I (SVG)

初期化で要素を作成

自前でアニメーションを作成

```
20<g transform="translate(100,150)">
21  <line x1="-50" y1="0" x2="400" y2="0" stroke-width="2" stroke="black"/>
22  <path id="cycliod" fill="none" stroke="black" stroke-width="3"/>
23</g>
24</svg>
```

表示する SVG の要素は円が転がる水平線 (21 行目) とサイクロイド (22 行目) だけ

# やってみよう

第 5 回目 (その 2)

メディア専門ユニット I(SVG)

初期化で要素を作成

自前でアニメーションを作成

- ▶ 正  $n$  角形を描く
- ▶ 正 7 角形の頂点を一つ置きに結んだ図形を描く

# サイクロイドのアニメーション (1)

第 5 回目 (その 2)

メディア専門ユニット I(SVG)

初期化で要素を作成

自前でアニメーションを作成

```
1<?xml version="1.0" encoding="UTF-8" ?>
2<svg xmlns="http://www.w3.org/2000/svg"
3    height="100%" width="100%">
4<title>サイクロイドを描く --- アニメーション版</title>
5<script type="text/ecmascript">
6//<![CDATA[
7let R = 50, Current = 1, Step=2, d ="M0,0 ";
8let T, Rot, C;
9window.onload = function() {
10  C =document.getElementById("cycliod");
11  T =document.getElementById("translate");
12  Rot =document.getElementById("rotate");
13  drawCurve();
14}
```

9 行目から 14 行目で初期化の関数が定義

- ▶ サイクロイドを表示する<path>要素と平行移動、回転のための<g>要素要素を変数に格納
- ▶ 13 行目で部分的に描く関数 drawCurve を呼び出し



## サイクロイドのアニメーション (2)

第 5 回目 (その 2)

メディア専門ユニット I(SVG)

初期化で要素を作成

自前でアニメーションを作成

```
15function drawCurve(){
16  let Next = Current + Step, rad;
17  if(Current<=360) {
18    for( ; Current< Next; Current++) {
19      rad = Current/180*Math.PI;
20      d += `${R*(rad-Math.sin(rad))},${R*(-1+Math.cos(rad))}`;
21    }
22    C.setAttribute("d",d);
23    T.setAttribute("transform",'translate(`${R*rad}`, -50)');
24    Rot.setAttribute("transform", 'rotate(`${Next}`)');
25    setTimeout(drawCurve,100);
26  }
27}
28//]]>
29</script>
```

# サイクロイドのアニメーション (2)

## コードの解説

### JavaScript

第 5 回目 (その 2)

メディア専門ユニット I(SVG)

初期化で要素を作成

自前でアニメーションを作成

15 行目から 29 行目で drawCurve を定義

- ▶ 表示済みの角度を保持する変数 Current が 1 回転を示していない (17 行目) ならば曲線の次の部分を追加
- ▶ 次の段階までの部分を計算して追加 (18 行目から 24 行目)
- ▶ 転がる円を回転 (26 行目) と平行移動 (25 行目) で移動
- ▶ setTimeout 関数は指定された関数 (第 1 引数) を経過時間 (第 2 引数、単位は ms) 後に実行するように設定。ここでは  $100\text{ms} = 0.1$  秒後に設定
- ▶ setTimeout 関数の 3 番目以降の引数は第 1 引数の関数の引数となる。

## サイクロイドのアニメーション (3)

第 5 回目 (その 2)

メディア専門ユニット I (SVG)

初期化で要素を作成

自前でアニメーションを作成

```
30<g transform="translate(100,150)">
31  <line x1="-50" y1="0" x2="400" y2="0" stroke-width="2" stroke="black"/>
32  <path id="cycliod" fill="none" stroke="black" stroke-width="3"/>
33  <g id="translate" transform="translate(0,-50)">
34    <g id="rotate">
35      <circle cx="0" cy="0" r="50" stroke-width="3" stroke="black" fill="none"/>
36      <line x1="0" y1="0" x2="0" y2="50" stroke-width="2" stroke="black" />
37      <circle cx="0" cy="50" r="5" fill="black"/>
38    </g>
39  </g>
40</g>
41</svg>
```

# サイクロイドのアニメーション (3)

## コードの解説

### 表示する図形

第 5 回目 (その 2)

メディア専門ユニット I (SVG)

初期化で要素を作成

自前でアニメーションを作成

- ▶ 円が転がる水平線 (33 行目) とサイクロイドを描くための<path>要素 (34 行目)
- ▶ 回転する円 (37 行目) と、円周上の点 (39 行目) とこの点と中心を結ぶ線 (38 行目) を囲む<g>要素が 2 つ
- ▶ 内側は回転、外側は平行移動

## やってみよう

第 5 回目 (その 2)

メディア専門ユニット I(SVG)

初期化で要素を作成

自前でアニメーションを作成

- ▶ ここでやっていることは 1 年の時の Proccessing で描くアニメーションと同じ
- ▶ したがって、Processing で行ったアニメーションを SVG に移植することが可能
- ▶ `setTimeout()` の次の実行時間を変えてアニメーションのスピードを変える