

真面目なアニメーション



html5j 2013, Tokyo
Brian Birtles, Mozilla Japan

課題

- **真面目なアニメーション？**

にゃんこのGIF以外、何かある？

- **難しいアニメーション**

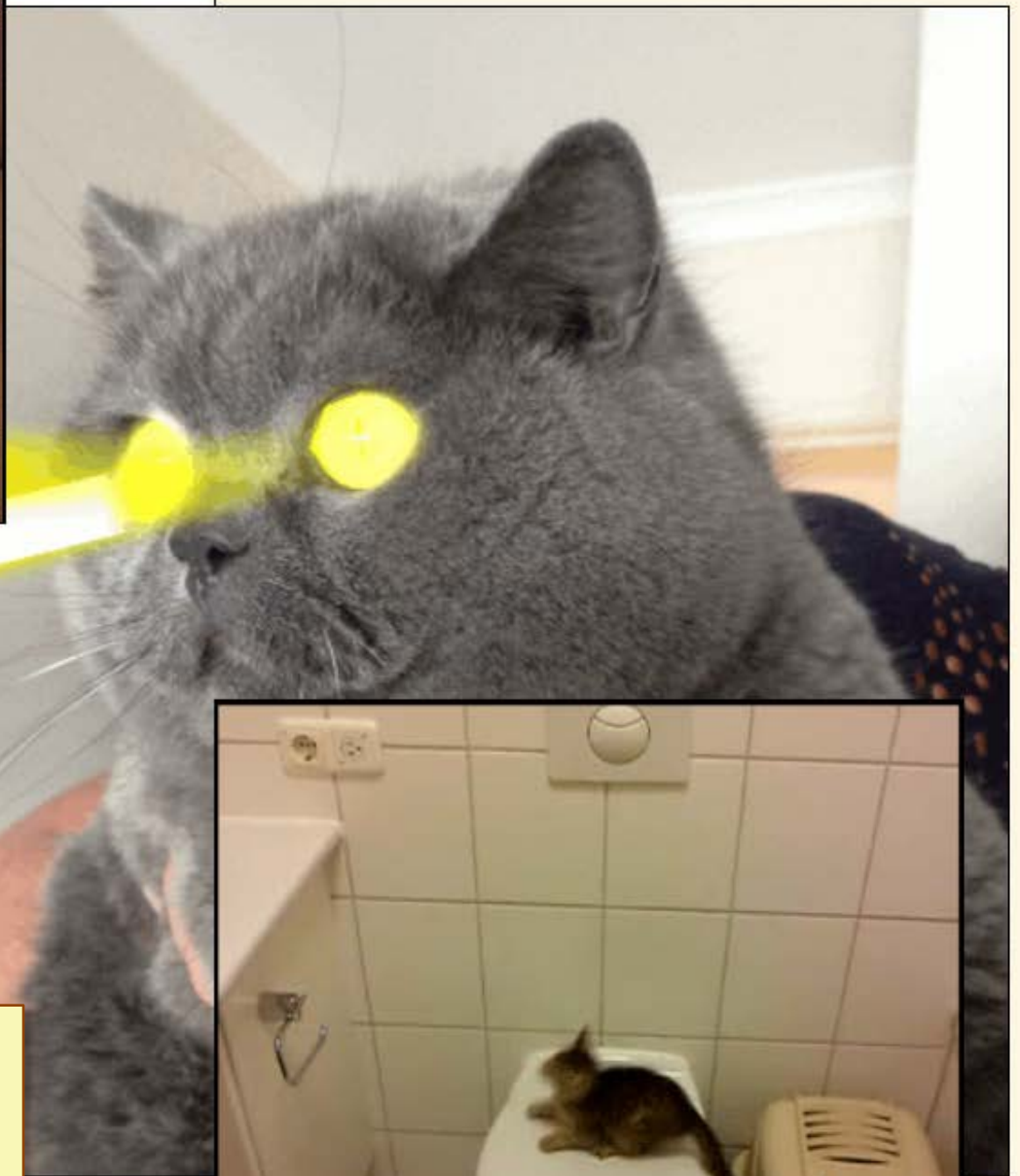
現在のウェブのまずさ

- **夢を叶えるアニメーション**

Web Animations

- **SVGの復讐のアニメーション**

Animation Elements



アニメーションと言えば、だいたいこの
ようなものを連想するでしょう。

またはこの人…



しかし、実はニャンコのGIF以外の使い方もあります。



OSXのGenie効果では
アニメーションによって最小化の
結果が分かりやすく伝えられて、
元に戻したいとき、どうすればいい
かすぐ推測できます。



Firefox をご利用いただき、
ありがとうございます

生まれ変わった
Android 版 Firefox

詳しくはこちら



Firefox のソフトウェア更新機能をご存じですか？

すでに Firefox をお使いなら、改めて完全なインストール手順

「ヘルプ」メニューから「Firefox について」という項目を
適用」ボタンをクリックしてください。

上記の更新機能を利用せずにダウンロードを行う場合は [ここ](#)

Firefoxのインターフェースにもアニメーションが同様に使われています。読み込んだファイルを開くときどこにあるかはすぐ分かります。

<http://www.flickr.com/photos/twitteroffice/5885172082/in/photostream/>

14:46 JST



14:46 JST: A powerful 9.0 ear

だいたいどんな変化でも
アニメーションによってより
分かりやすく伝えることができます。

長い物語もアニメーションによって
誰でも分かるように伝えられます。

そこは
何がしの院という
すてられたお邸――

<http://www.youtube.com/watch?v=Yg5BZARVDA8>

基本的にアニメーションは時間を生かして、情報を伝えることです。

Part 2

難しいアニメーション

現在のウェブのまずさ

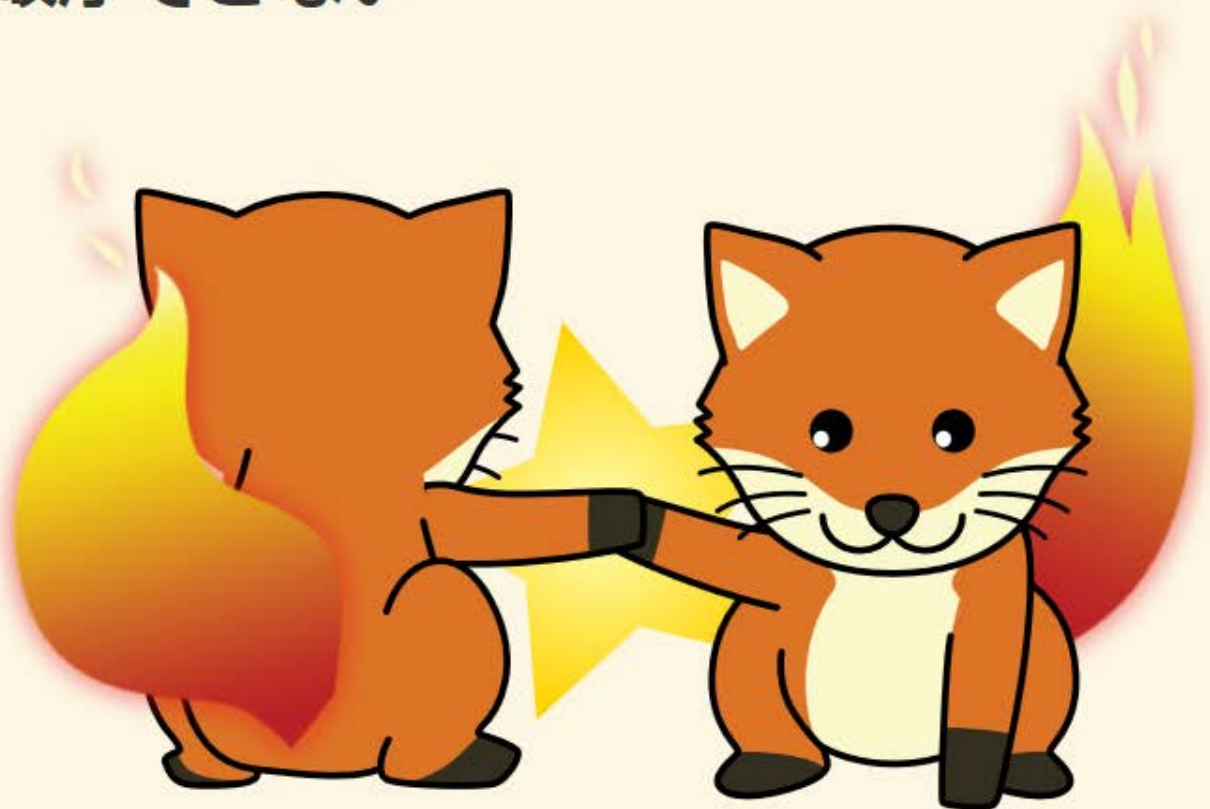


CSS Animations and Transitions

```
h3 {  
  animation-name: heading;  
  animation-duration: 2s;  
  animation-iteration-count: infinite;  
}  
  
@keyframes heading {  
  50% {  
    color: green;  
    font-size: 80%;  
  }  
}
```

CSS Animations and Transitions

☹ 同期、順序できない



CSS Animations and Transitions

- ☹ 同期、順序できない
- ☹ 再生位置の調整、一時停止、逆再生などできない

CSS Animations and Transitions

もし両方のクラスを同じ要素に適用すれば…

- ☹ 同期、順序できない
- ☹ 再生位置の調整、一時停止、逆再生などできない
- ☹ 同時に同じプロパティをアニメートできない

class="kaiten"



class="kakudai"



CSS Animations and Transitions

ひとつが勝ち。CSS Animation
で足し算できないから。

- ☹ 同期、順序できない
- ☹ 再生位置の調整、一時停止、逆再生などできない
- ☹ 同時に同じプロパティをアニメートできない

class="kaiten"



class="kakudai"



class="kaiten kakudai"



CSS Animations and Transitions

- ☹ 同期、順序できない
- ☹ 再生位置の調整、一時停止、
- ☹ 同時に同じプロパティをアニメートできない

前もって両方の効果が入っているアニメーションを作ってもタイミングが合わないし、全ての組み合わせを用意しなければならない。

class="kaiten"



class="kakudai"

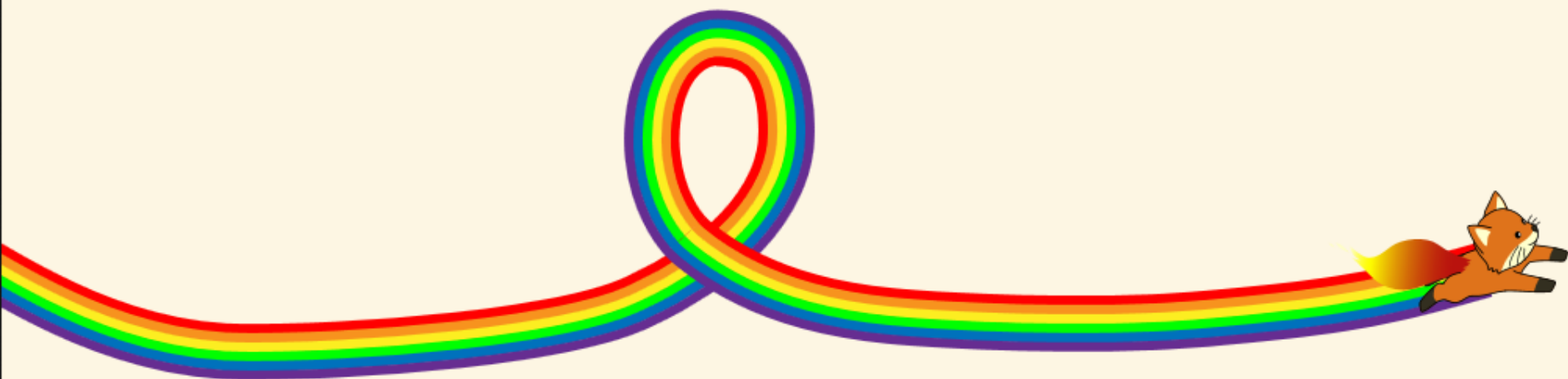


class="kaiten-kakudai"



CSS Animations and Transitions

- ☹ 同期、順序できない
- ☹ 再生位置の調整、一時停止、逆再生などできない
- ☹ 同時に同じプロパティをアニメートできない
- ☹ パスに沿って動かせない



CSS Animations and Transitions

- ☹ 同期、順序できない
- ☹ 再生位置の調整、一時停止、逆再生などできない
- ☹ 同時に同じプロパティをアニメートできない
- ☹ パスに沿って動かせない
- ☹ **CSSプロパティ以外**は動かせない
 - パス、スクロールバー、`canvas`、`WebGL`、`<video>`

CSS Animations and Transitions

- ☹ 同期、順序できない
- ☹ 再生位置の調整、一時停止、逆再生などできない
- ☹ 同時に同じプロパティをアニメートできない
- ☹ パスに沿って動かせない
- ☹ CSSプロパティ以外は動かせない
 - パス、スクロールバー、`canvas`、`WebGL`、`<video>`



このようにフォクすけの目をアニメートできない

CSS Animations and Transitions

- ☹ 同期、順序できない
- ☹ 再生位置の調整、一時停止、逆再生などできない
- ☹ 同時に同じプロパティをアニメートできない
- ☹ パスに沿って動かせない
- ☹ CSSプロパティ以外は動かせない
 - パス、スクロールバー、`canvas`、`WebGL`、`<video>`
- ☹ Javascriptでデバッグなどできない

SVG Animation

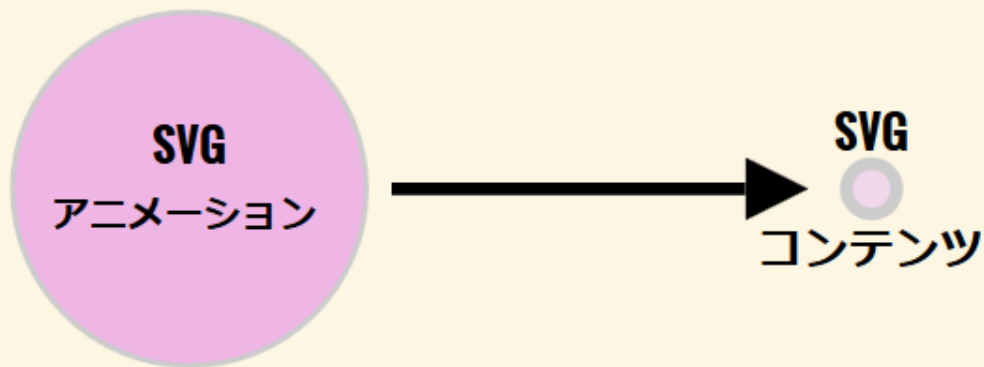
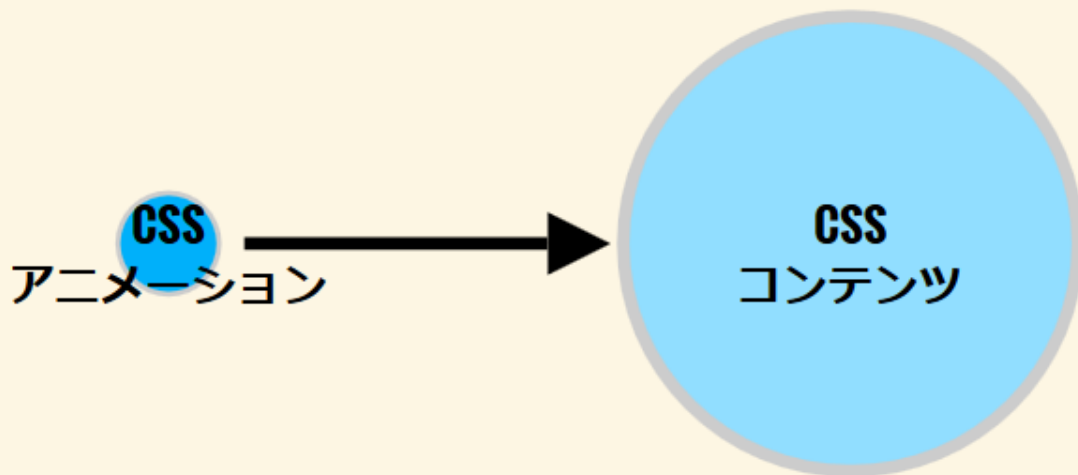
SVG Animationの機能に方が豊かだが...

CSS
アニメーション

SVG
アニメーション

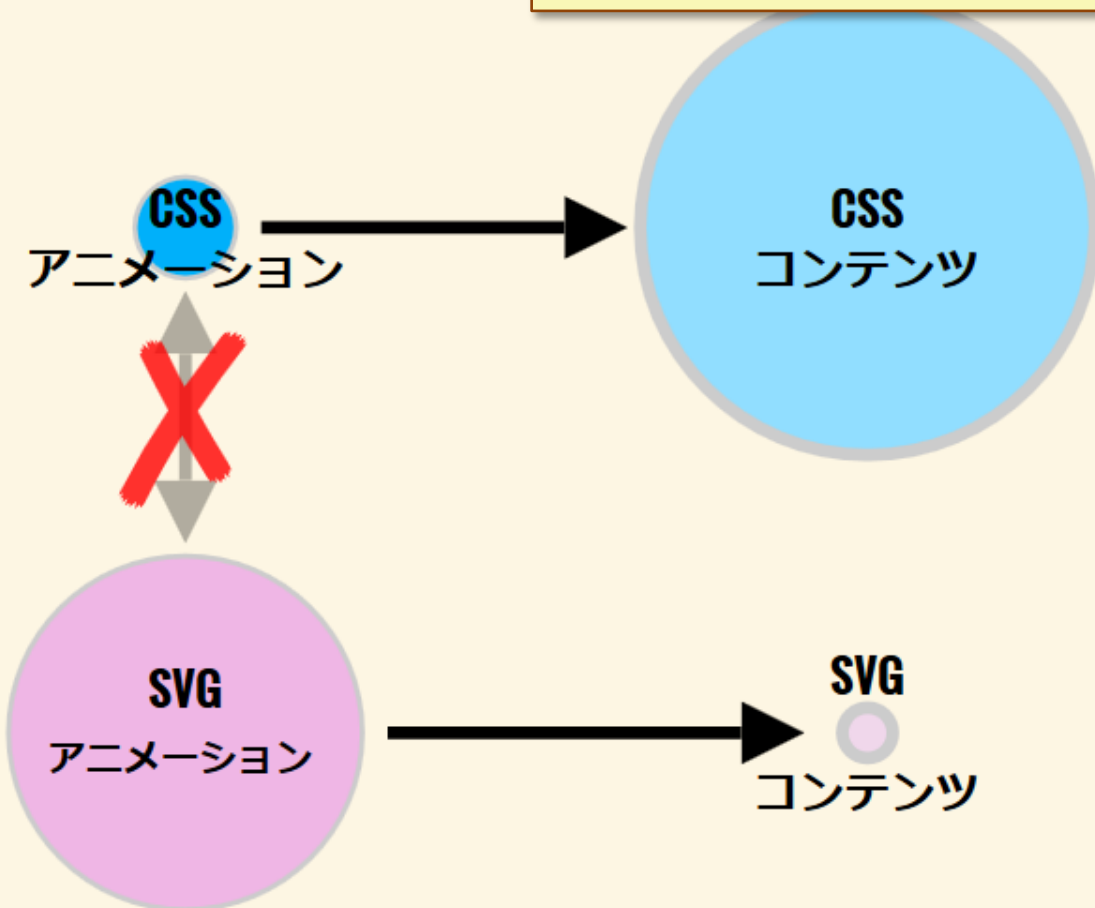
SVG Animation

CSSのコンテンツの方が多い。



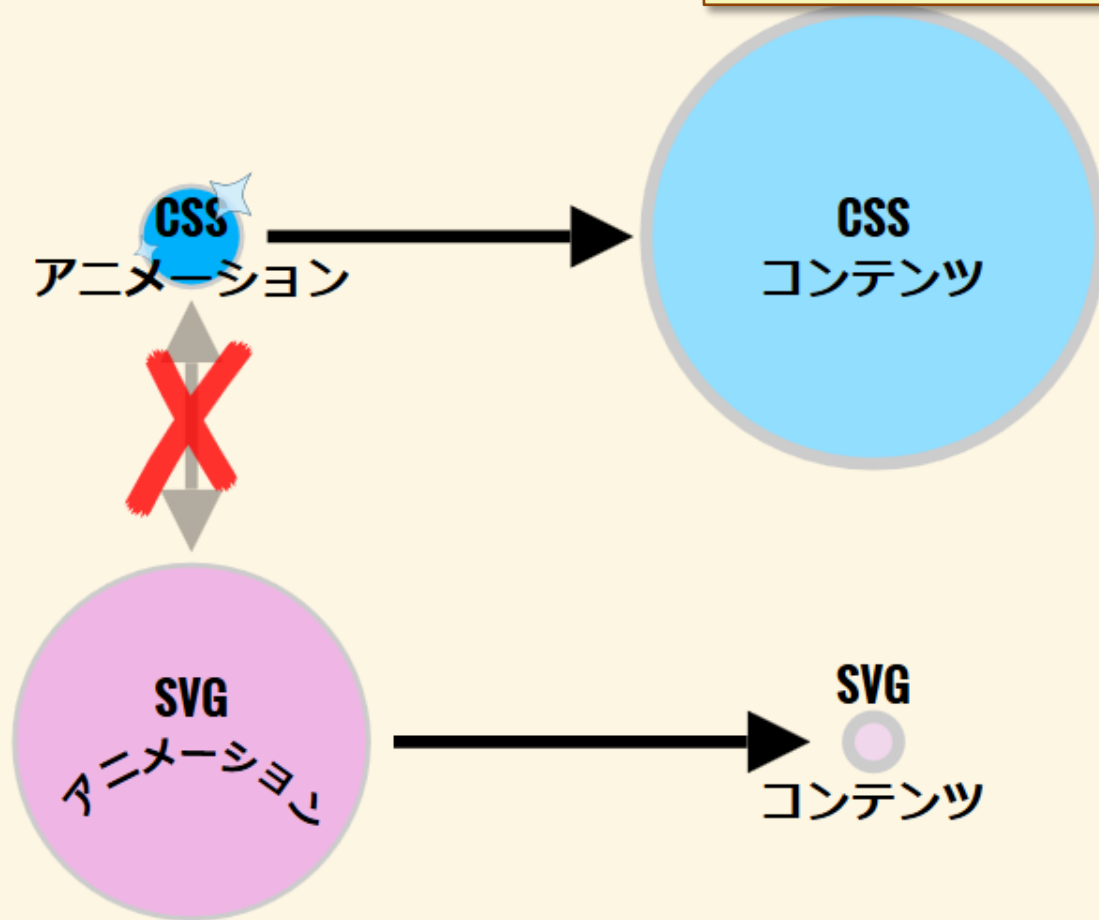
SVG Animation

そして多くのブラウザでは
CSS Animationの実装と
SVG Animationの実装が全く別のもの



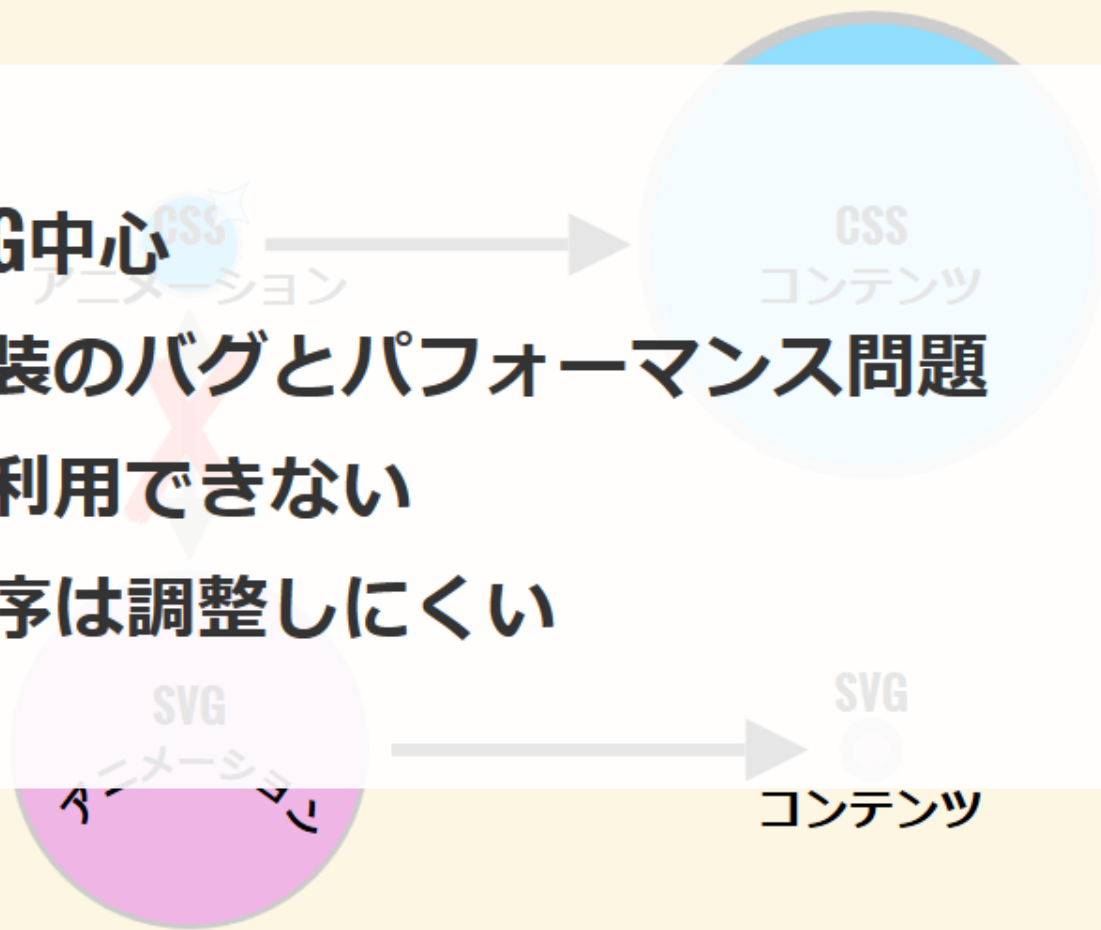
SVG Animation

その結果として、よく使われている
CSS Animationの方がちゃんと
実装されています。

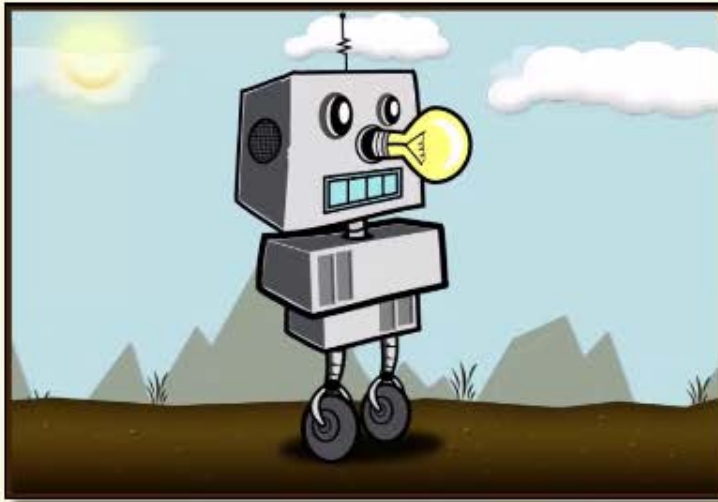


SVG Animation

- ☹️ SVG中心
- ☹️ 実装のバグとパフォーマンス問題
- ☹️ 再利用できない
- ☹️ 順序は調整しにくい



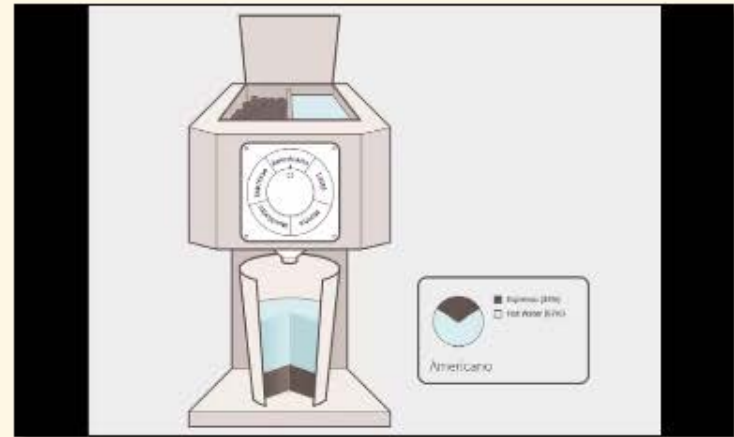
Javascript



jQuery • HTML <http://robot.anthonycalzadilla.com/>



WebGL <http://www.unrealengine.com/html5/>



Snap.svg • SVG <http://snapsvg.io/demos/#coffee>



Canvas <http://www.cuttherope.ie/>

Javascript

- ☹️ 主要スレッドで動 (CSS プロパティを変えるとき)
- ☹️ SVGアニメーションなどと足し算できません (加法的アニメーション)
- ☹️ Javascriptが動かないところ？
 - ``,
background-image: url("anim.svg") など
 - SVG-in-OpenType (絵文字など)
 - `<iframe sandbox>`

祭り



Part 3

夢を叶えるアニメーション

Web Animations



CSS



SVG

Webには二つより一つのアニメーション・モデルを作った方がよいじゃないかという声があつて...

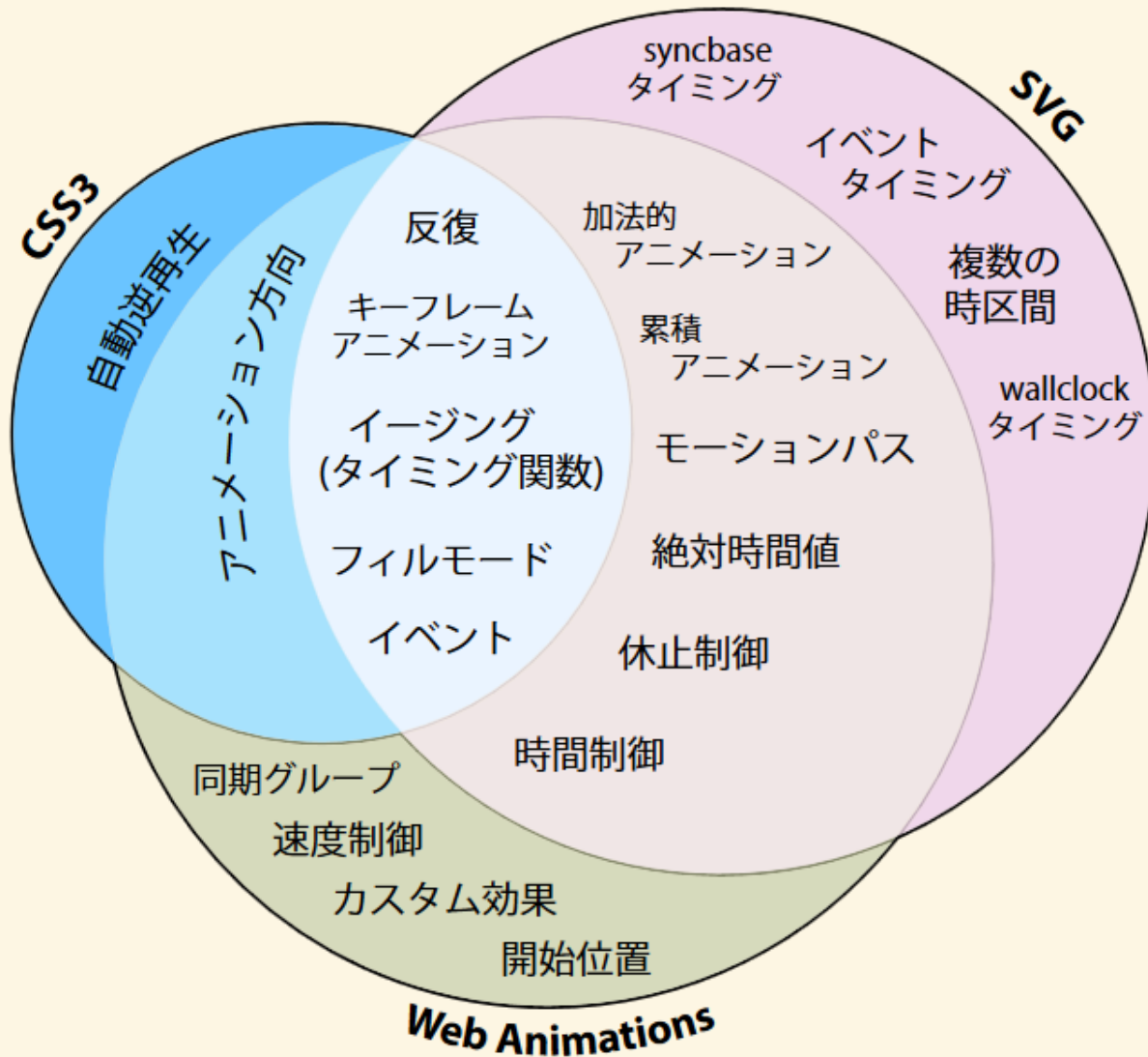


CSS

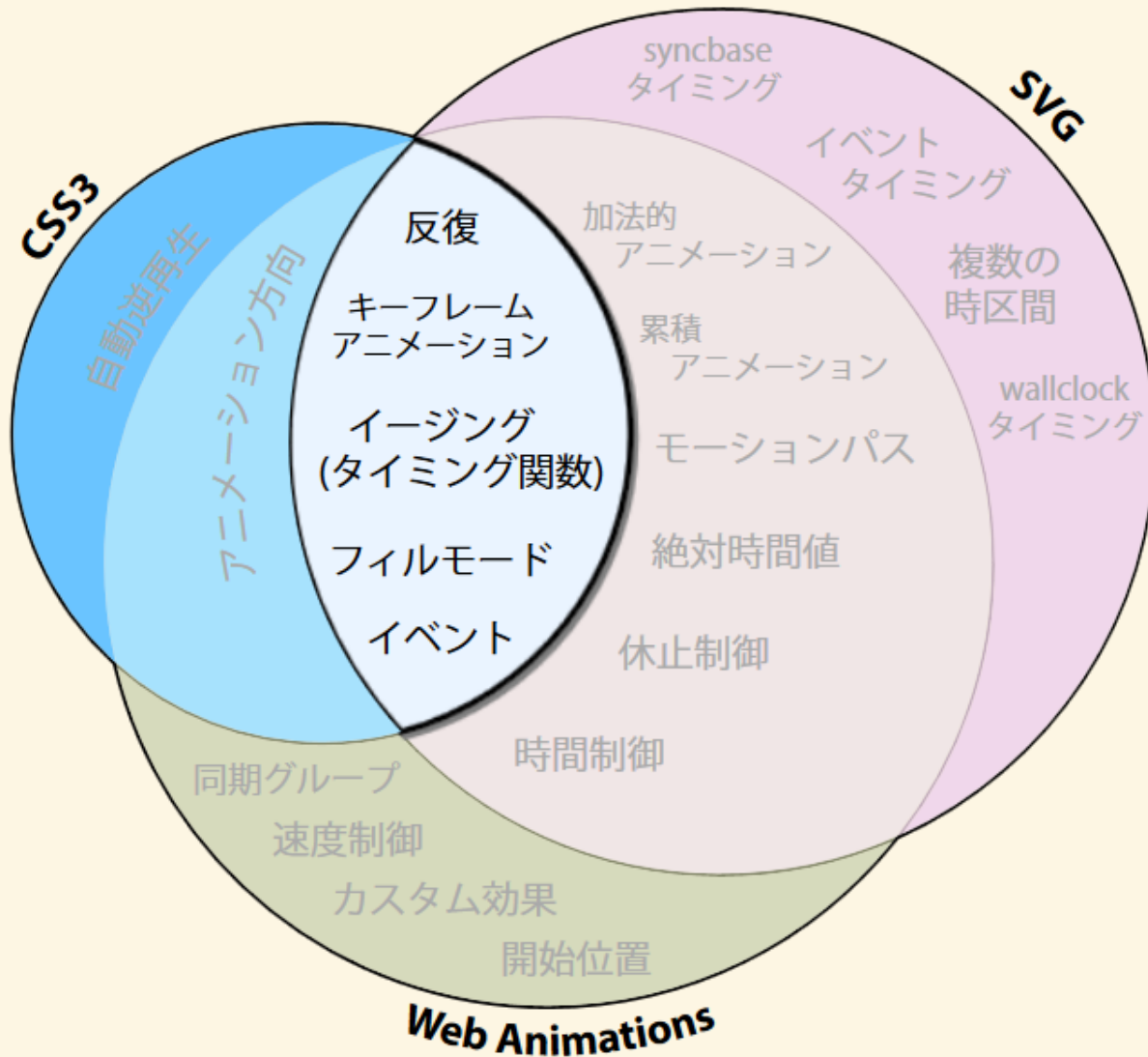
SVG

+a

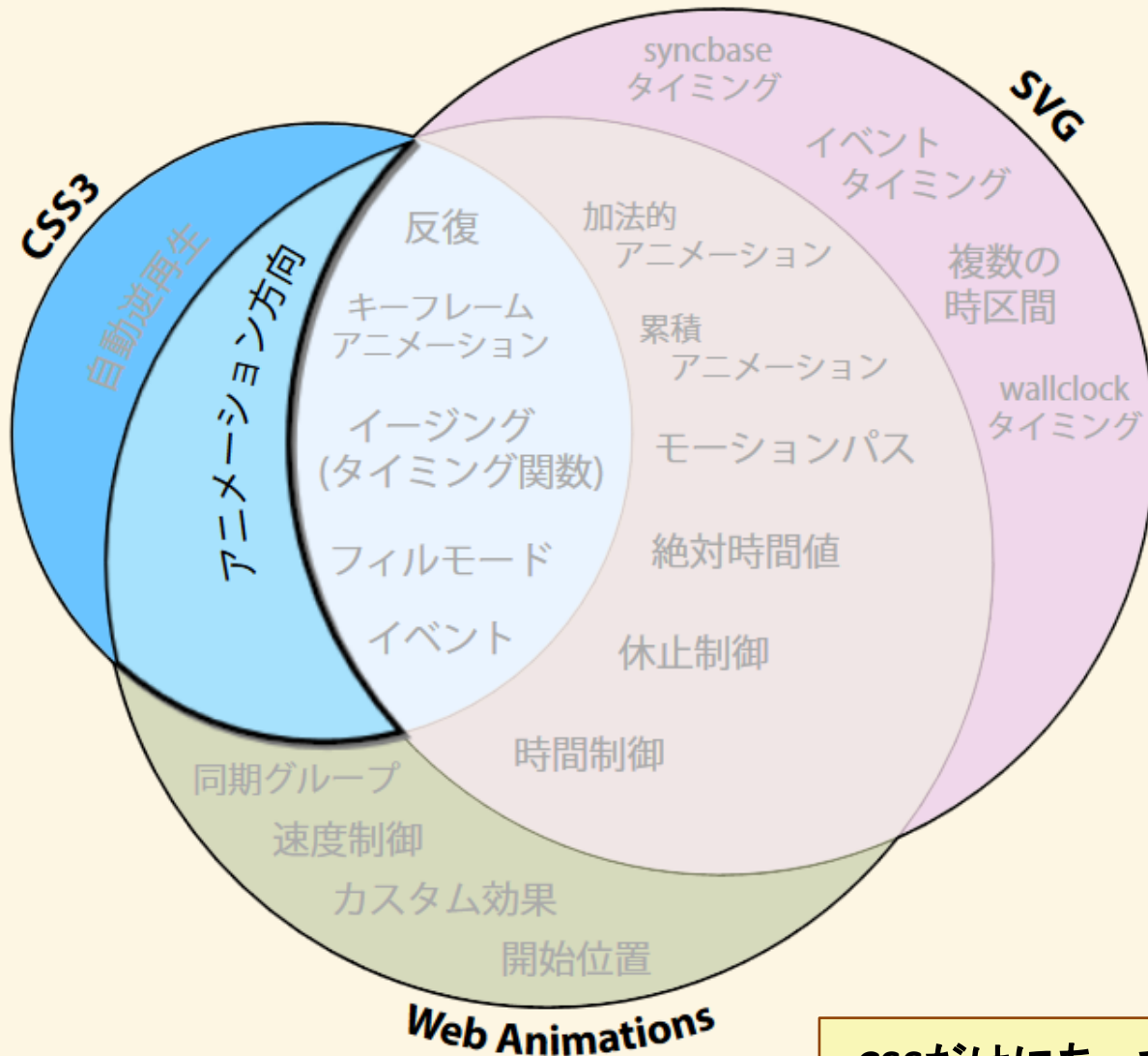
同時にWeb開発者が欠けている
機能を指摘してくれた。



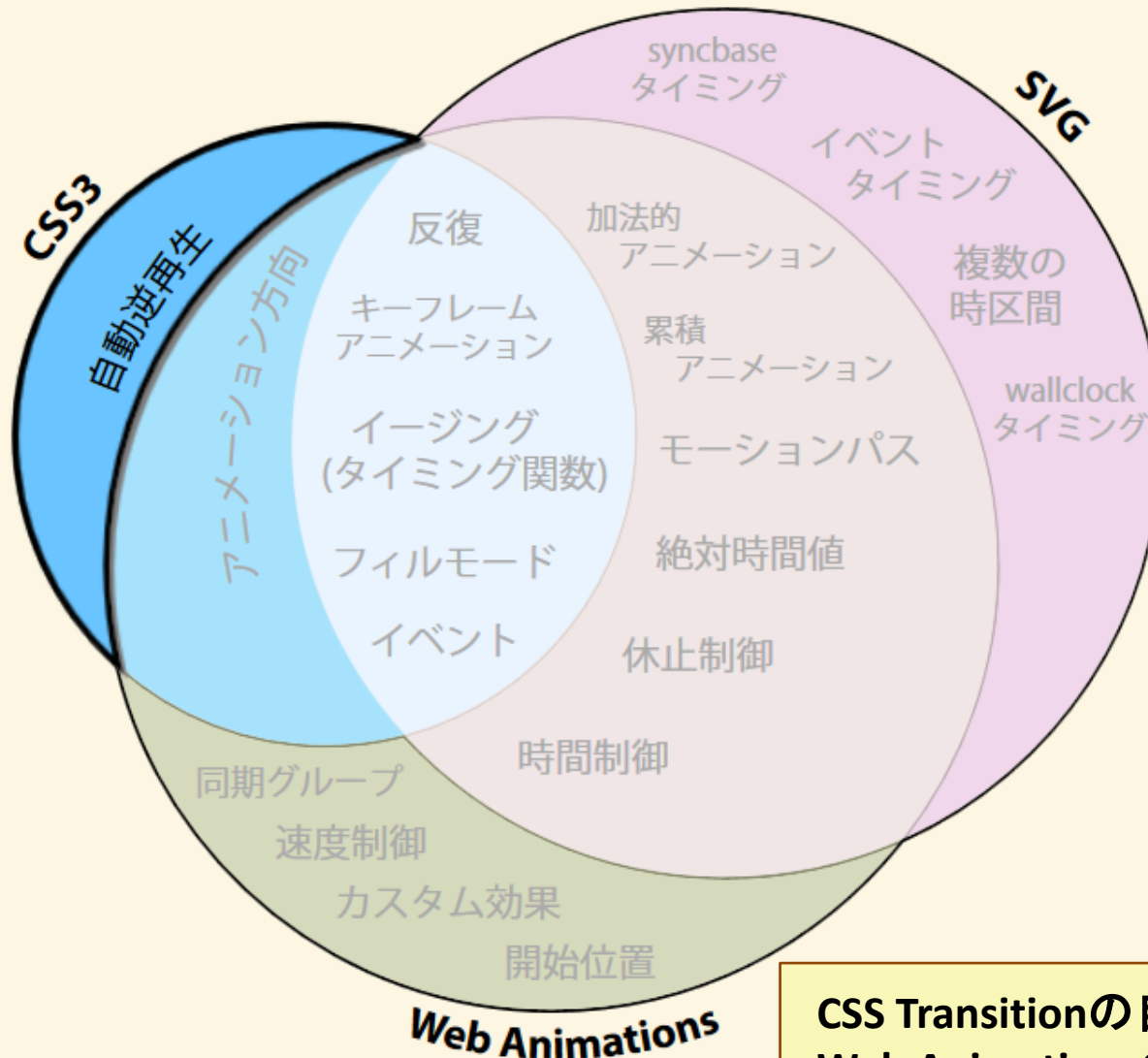
それでこうしました。



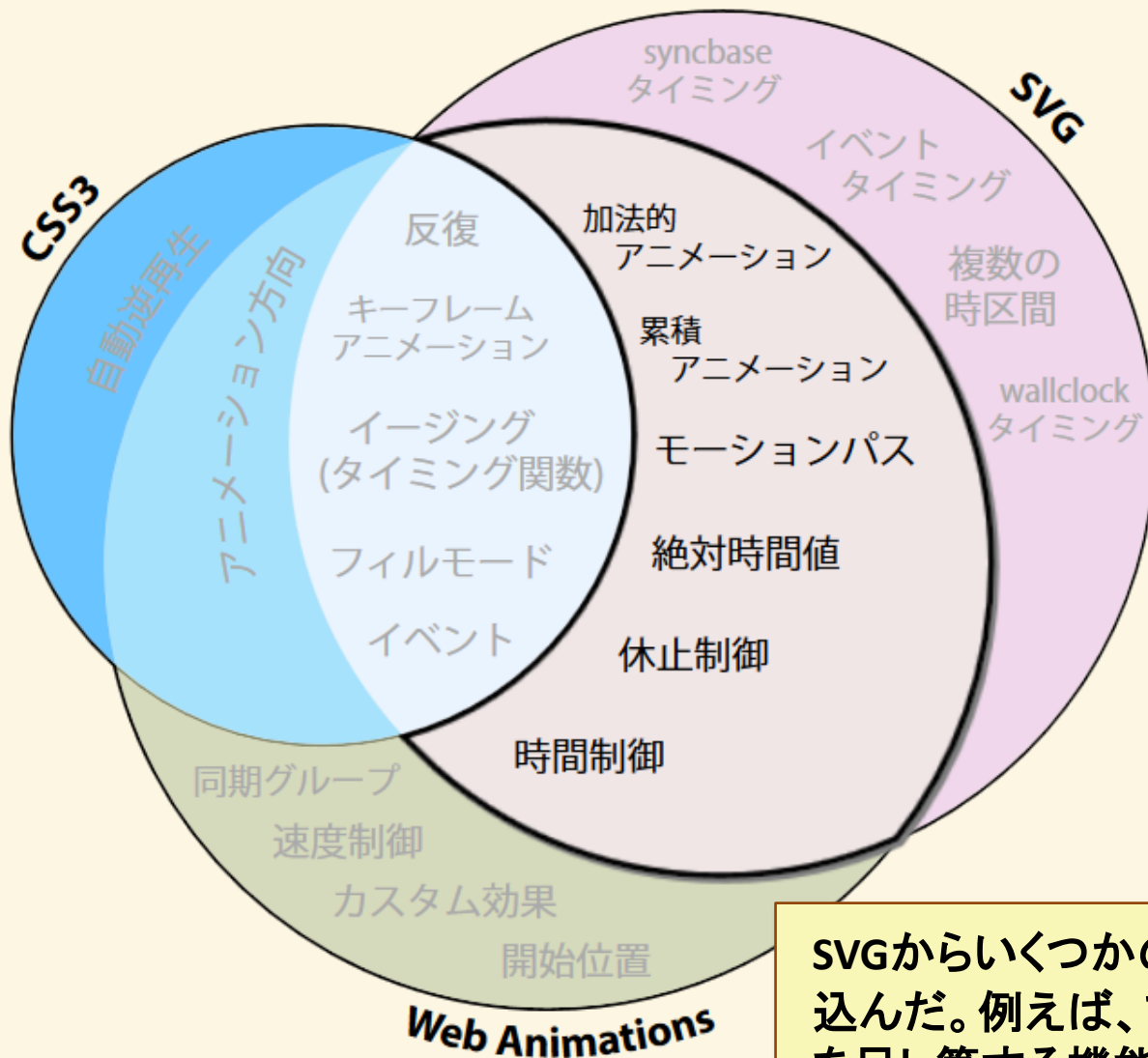
すでにSVGにもCSS
にもあった機能。



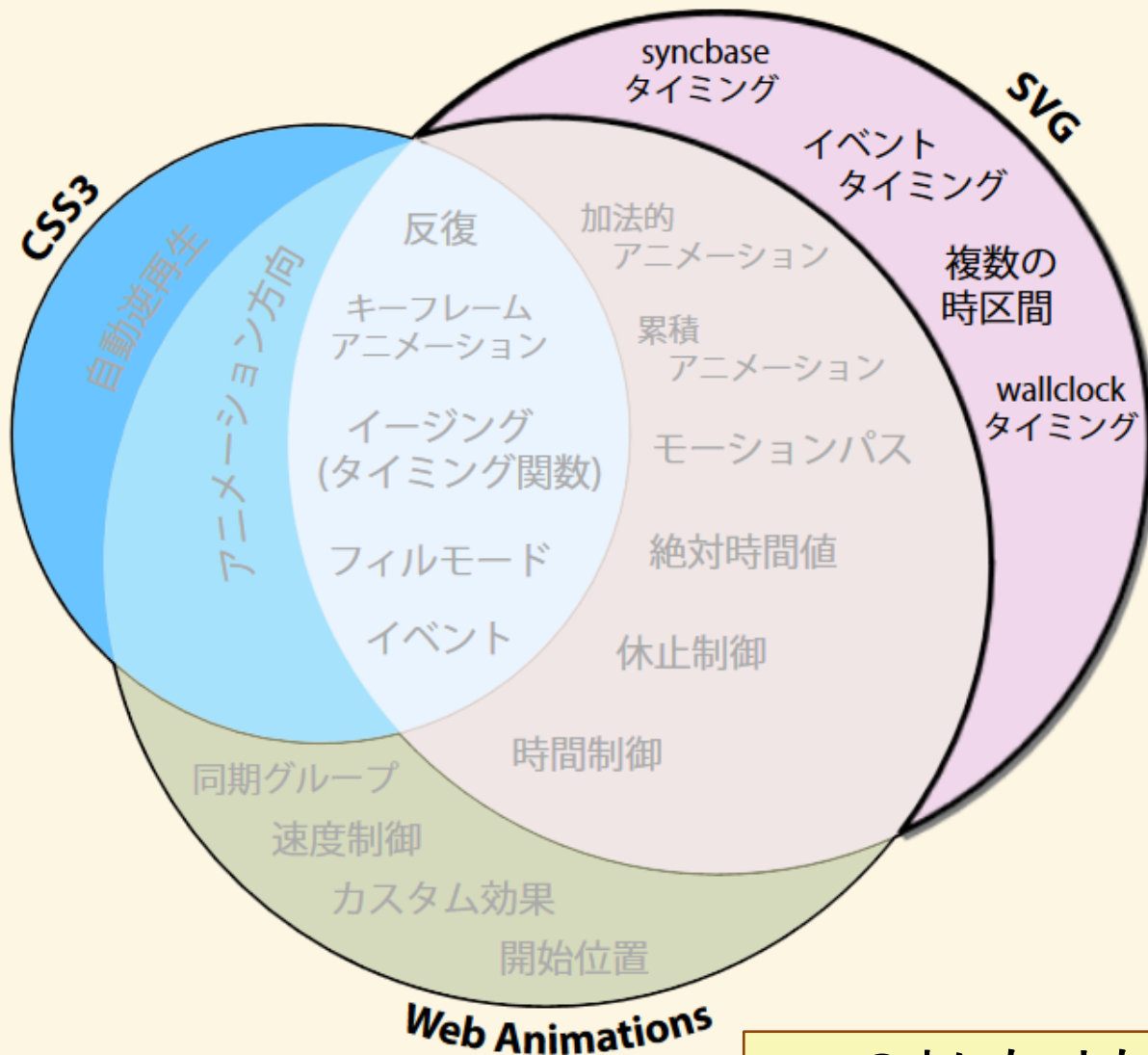
CSSだけにあってWeb Animationsに入れたのは direction です。



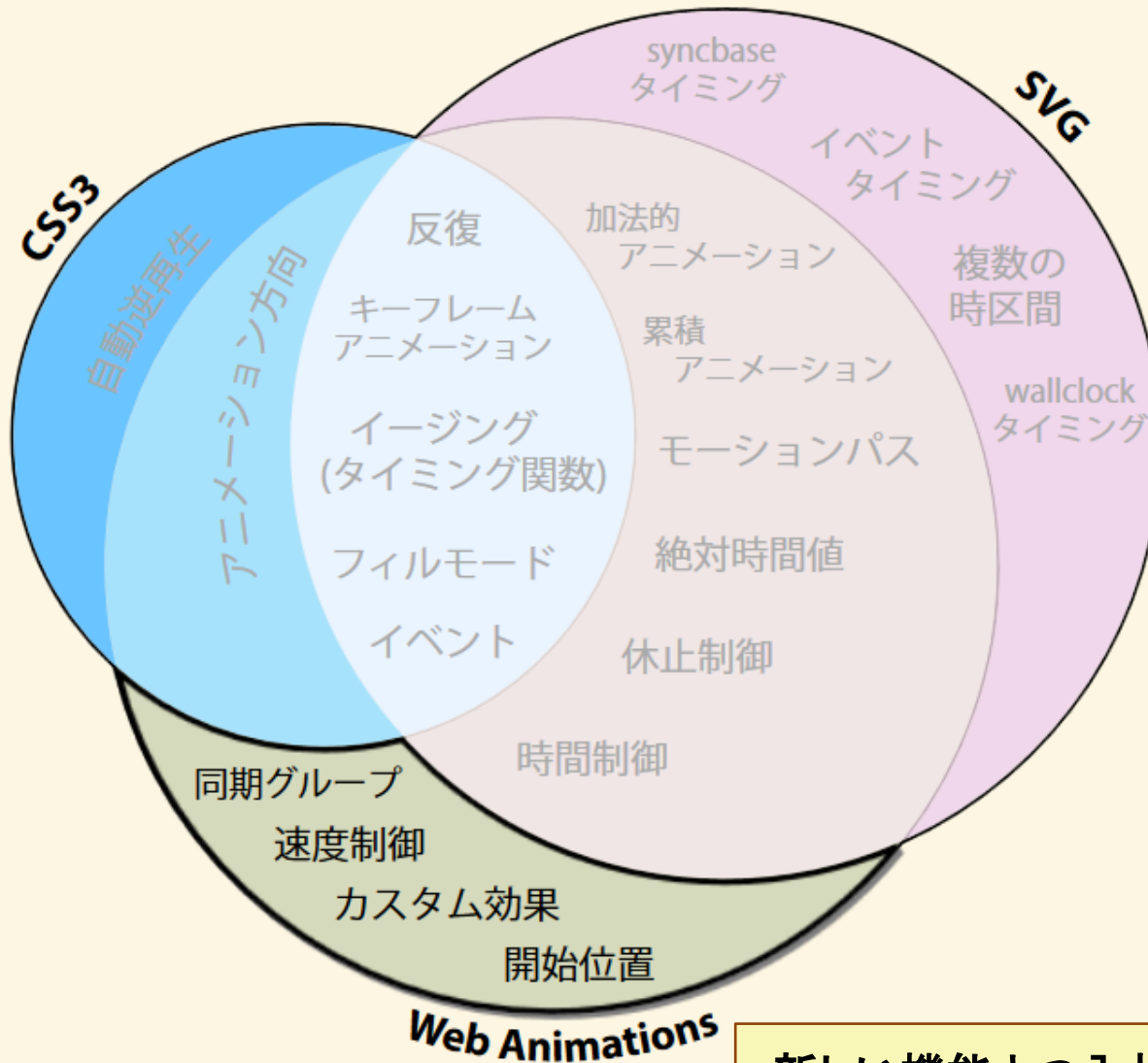
CSS Transitionの自動逆再生はWeb Animationsのモデルに入っていないけど他の機能ですぐ実現できる。



SVGからいくつかの機能を取り込んだ。例えば、アニメーションを足し算する機能(加法的アニメーション)、モーシヨンプラス、時間制御など。



SVGの中にもっとも複雑な機能を入れなかったがモデルに入っている機能ですぐ実現できる。



新しい機能よつ入れた。特に有力なのはタイミンググループとカスタム効果。



**Web
Animations
モデル**

根本的にWeb Animationsは抽象的なアニメーションモデルです。

.CSS { }

“CSS Animations Level 4”?

**Web
Animations
モデル**

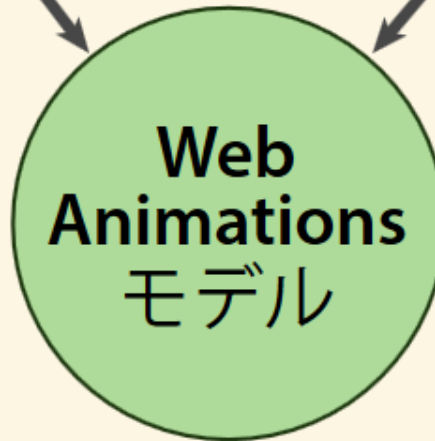
どうやってCSS構文をWeb Animations
のモデルによって実装するかとは別
のスペックで定義される。

.CSS { }

"CSS Animations Level 4"?

<svg>

"Animation elements"



同様にSVGの構文と仕組みは Animation Elementsというスペックで定義されている。

```
.CSS { }
```

"CSS Animations Level 4"?

```
<svg>
```

"Animation elements"

**Web Animations
モデル**

```
api();
```

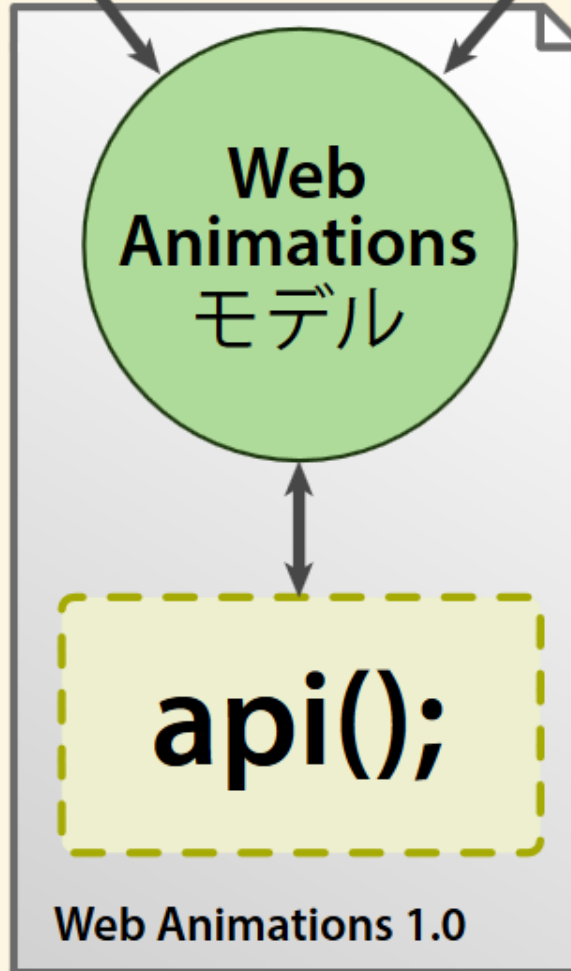
もう一つの界面としてモデルにはAPIもある。


```
.CSS { }
```

“CSS Animations Level 4”?

```
<svg>
```

“Animation elements”



現在モデルもAPIも同じスペックで定義されている。それは維持しやすいからです。

基本的な補間

```
Element.animate(<プロパティー>, <時間>)
```

基本的な補間

```
Element.animate(  
  { transform: 'rotate(360deg)' }, 1)
```



```
Element.animate(  
  { transform: 'rotate(360deg)' },  
  { duration: 1,  
    easing: 'ease-in-out',  
    direction: 'alternate',  
    iterations: Infinity } }
```



プレイヤー

- `cancel()`
- `finish()`
- `play()`
- `pause()`
- `reverse()`
- `currentTime`
- `playbackRate`
- `paused`

<http://brian.sol1.net/svg/2013/07/25/players-wanted-the-pause-and-seek-game/>



DVD Player designed by [Dan Hetteix](#) from The Noun Project

アニメーションなどの時間制御、一時停止、巻き戻しなどはプレイヤーという概念によって行う。

タイミンググループ

タイミンググループには種類二つある。名前はまた変わるかもしれないけど、とりあえずタイミンググループとタイミングチェーンと呼ばれる。

Timing Group



タイミンググループの子供は同時に動く。

Timing Chain



タイミンググループ

Timing Group



Timing Chain



タイミングチェーンのこともは順番に動く。

タイミンググループ

一緒に使える。

Timing Chain



Timing Group



前のデモの構成

Timing Chain

ドアが閉まる

Timing Group

ほこりが飛ぶ

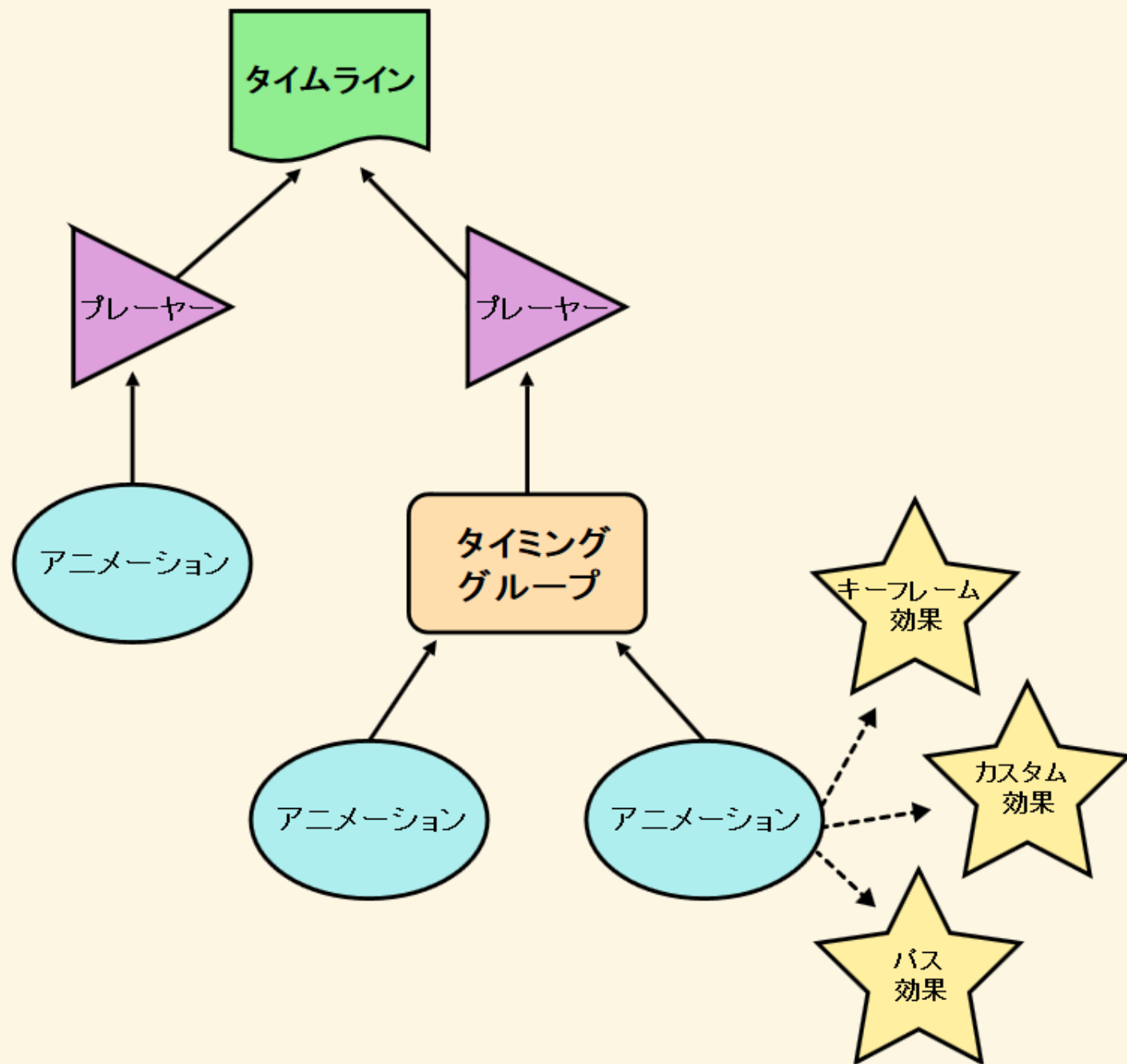
絵が傾く

カスタム効果

```
function sample (  
  timeFraction /* 区間割合 */,  
  iteration /* 反復目 */,  
  target /* 対象となる要素 */,  
  previousTimeFraction /* 前回の区間割合 */)   
{  
  ...  
}
```

パスアニメーション

```
new Animation (  
  element,  
  new PathAnimationEffect (  
    'M 100 200 ' + ... +  
    'C 800 100 900 100 900 100',  
    'auto-rotate'),  
  { duration: 5 }  
);
```



スペック

dev.w3.org/fxtf/web-animations

ポリフィル

git.io/webanim

ネイティブ実装現状



進んでいる



始まった



始まった



未発表

Part 4

SVGの復讐のアニメーション

Animation Elements



SVG Animationの紹介として、これはものすごくSVG Animationエディターのアプリ。以下のURLでアクセスできる。

<http://parapara-editor.mozlabs.jp/sandbox>

完成したキャラクターが交通のキャンバスで表示される。
それもSVG Animationでできている。

詳しくは以下のURLをご覧ください。

<http://parapara.mozlabs.jp>



キャラクターのSVGファイルの構成。

```
<g visibility="hidden">
  ...パスデータ...
  <set id="a" attributeName="visibility"
    to="visible" dur="0.3s"
    begin="0; c.end"/>
</g>
<g visibility="hidden">
  ...パスデータ...
  <set id="b" attributeName="visibility"
    to="visible" dur="0.3s" begin="a.end"/>
</g>
<g visibility="hidden">
  ...パスデータ...
  <set id="c" attributeName="visibility"
    to="visible" dur="0.3s" begin="b.end"/>
</g>
```

しかし、Animation Elementsでもっと簡単に書ける。

```
<g visibility="hidden" class="frame">  
  ...パスデータ...
```

```
</g>
```

```
<g visibility="hidden" class="frame">  
  ...パスデータ...
```

```
</g>
```

```
<g visibility="hidden" class="frame">  
  ...パスデータ...
```

```
</g>
```

```
<timingchain>
```

```
  <set select=".frame"
```

```
    attributeName="visibility" to="visible"
```

```
    dur="0.3s"/>
```

```
</timingchain>
```

selectという属性によってアニメーションの定義を再利用できる。

```
<g visibility="hidden" class="frame">
  ...パスデータ...
</g>
<g visibility="hidden" class="frame">
  ...パスデータ...
</g>
<g visibility="hidden" class="frame">
  ...パスデータ...
</g>
<timingchain>
  <set select=".frame"
    attributeName="visibility" to="visible"
    dur="0.3s" />
</timingchain>
```

タイミングチェーンによって順序が作りやすい。

```
<g visibility="hidden" class="frame">  
  ...パスデータ...
```

```
</g>
```

```
<g visibility="hidden" class="frame">  
  ...パスデータ...
```

```
</g>
```

```
<g visibility="hidden" class="frame">  
  ...パスデータ...
```

```
</g>
```

```
<timingchain>
```

```
  <set select=".frame"
```

```
    attributeName="visibility" to="visible"
```

```
    dur="0.3s" />
```

```
</timingchain>
```

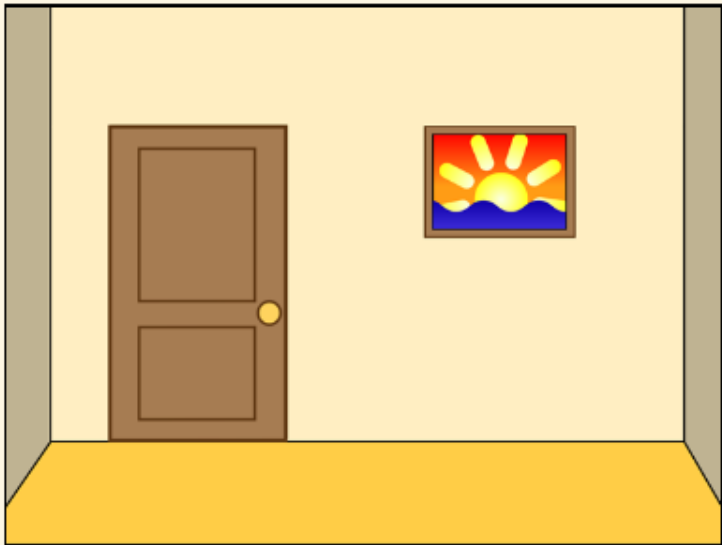
ドアが閉まるデモの構成。

```
<timingchain>
  <!-- ドアを閉める -->
  <animate href="door"
    attributeName="transform"
    to="rotate(0deg)" dur="2s"
    easing="cubic-bezier(0.9,0,1,1)"/>
  <timinggroup>
    <!-- 絵を傾ける -->
    <animate href="picture"
      attributeName="transform"
      to="rotate(-10deg)" dur="0.15s"/>
    <!-- ほこりを吹き乱される -->
    <animate href="smoke"
      attributeName="opacity"
      values="0; 0.8; 0" dur="0.2s"
      begin="0.4s"/>
  </timinggroup>
</timingchain>
```



要素構文はこんなような時間階層に適切。しかし、
<set>という要素によってCSSのクラスを適用すれば
CSS Animationも同時に使える。

タイミング情報を1か所またタイムシートと呼ばれる別のファイルに集めるパターンが多い。



画像コンテンツ



```
85 <!-- Picture -->
86 <g transform="translate(551.5 196.5)">
87 <g id="picture">
88 <g transform="translate(-551.5 -196.5)">
89 <!-- Frame -->
90 <rect x="468" y="135" width="167" height="123"
91 stroke="#008B13" stroke-width="2" fill="#A67C52"/>
92
93 <defs>
94 <clipPath id="pictureRegion">
95 <rect y="143" x="476" height="106" width="149"/>
96 </clipPath>
97 </defs>
98 <!-- Sky -->
99 <linearGradient id="sunsetSky" y2="142" gradientUnits="userSpaceOnUse"
100 x2="551.5" y1="248" x1="551.5">
101 <stop stop-color="#FF9C15" offset=".4439"/>
102 <stop stop-color="#F00" offset="1"/>
103 </linearGradient>
104 <rect clip-path="url(#pictureRegion)" height="106" width="149" stroke="#000"
105 stroke-miterlimit="10" y="142" x="477" stroke-width="2"
106 fill="url(#sunsetSky)"/>
107 <!-- Sun -->
108 <g clip-path="url(#pictureRegion)">
109 <g transform="translate(553 216)">
110 <!-- Splitting out the transforms on the sun for now since the shim
111 doesn't currently support additive animation for transforms of
112 different types. -->
113 <g id="taiyou">
114 <defs>
115 <radialGradient id="sunGrad" gradientUnits="userSpaceOnUse" cx="0" cy="0"
116 r="29.7">
117 <stop stop-color="#FFFEF3" offset="0"/>
118 <stop stop-color="#FFFF1B" offset="1"/>
119 </radialGradient>
120 </defs>
121 <circle r="29.7" fill="url(#sunGrad)"/>
122 <defs>
123 <radialGradient id="beamGrad" gradientUnits="userSpaceOnUse" cx="-21"
124 cy="0" r="40">
125 <stop stop-color="#FFFEF3" offset="0"/>
126 <stop stop-color="#FFFF1B" offset="1"/>
127 </radialGradient>
128 <rect x="-21" y="-8.5" width="42" height="17" rx="8.5"
129 fill="url(#beamGrad)" id="beam"/>
130 </defs>
131 <use xlink:href="#beam" transform="translate(56 0) rotate(8 -56 0)"/>
132 <use xlink:href="#beam" transform="translate(56 0) rotate(48 -56 0)"/>
133 <use xlink:href="#beam" transform="translate(56 0) rotate(88 -56 0)"/>
134 <use xlink:href="#beam" transform="translate(56 0) rotate(128 -56 0)"/>
135 <use xlink:href="#beam" transform="translate(56 0) rotate(168 -56 0)"/>
136 <use xlink:href="#beam" transform="translate(56 0) rotate(208 -56 0)"/>
137 <use xlink:href="#beam" transform="translate(56 0) rotate(248 -56 0)"/>
138 <use xlink:href="#beam" transform="translate(56 0) rotate(288 -56 0)"/>
139 </g>
140 </g>
141 </g>
142 <!-- Wave -->
143 <!-->
```

タイム・シート

ポリフィル

git.io/webanim

標準化のメーリングリスト

public-fx@w3.org

Brian Birtles

birtles@mozilla-japan.org

まじめな
アニメーション
つくってね！

