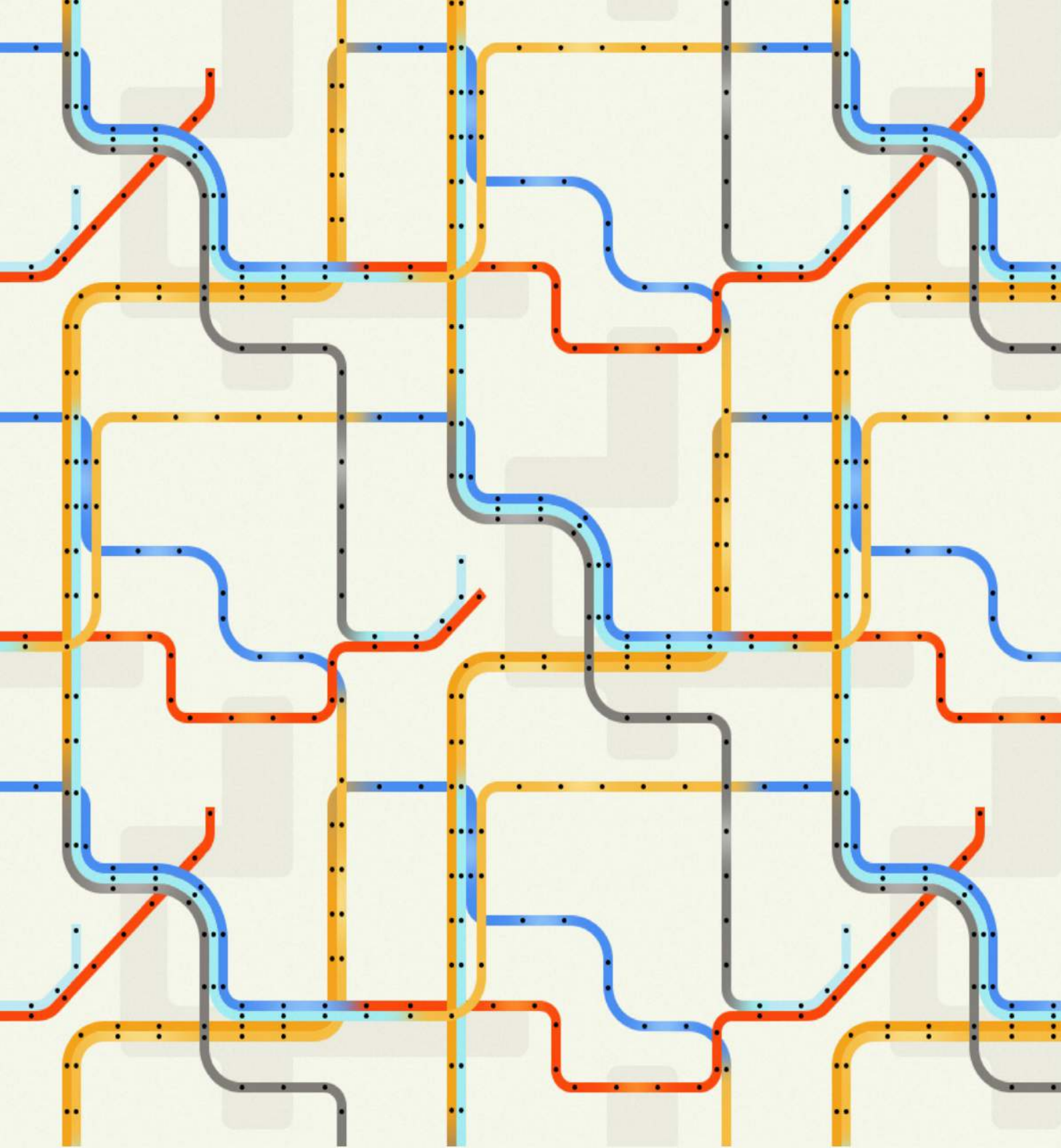




趣玩 CSS

Table of Contents

前言	0
01. 评分组件：兄弟选择器与镜像翻转的共用	1
02. Loading 动画：多动画的混用	2
03. Loading 动画再进化：换用 SVG 实现	3
04. SVG 做图像处理 (mask 篇)	4
05. SVG 做图像处理 (filter 篇)	5
06. 用 Sass 整理一份自己的样式助手（上）	6
07. 用 Sass 整理一份自己的样式助手（下）	7

为什么想写这样一本小书？

作为一名前端工程师，工作中很大的乐趣在于想出各种不同的解决方案来实现设计师的精心创作。写下代码的时候，我们常常希望它能够适应于不同尺寸的设备、拥有良好的兼容性，还有很好的可维护性，希望事后不管是谁看起自己的代码来都觉得赏心悦目。

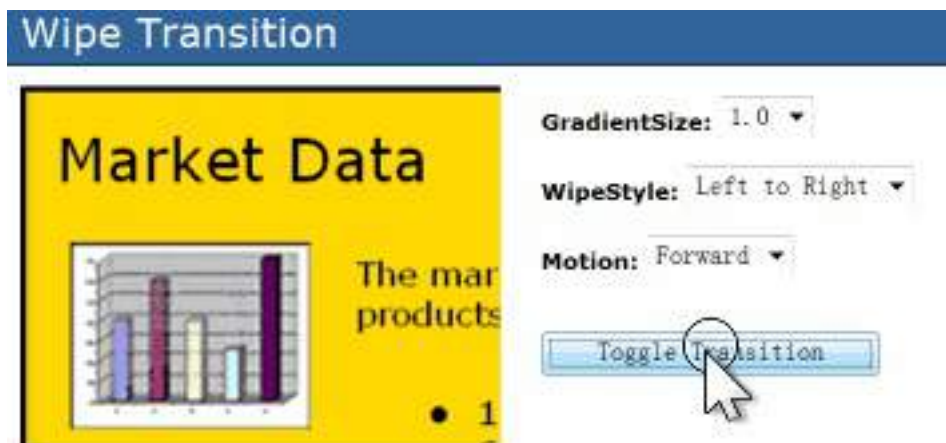
这本小书集中了一些你也许会觉得好玩的解决方案，希望你读完之后能够觉得，“唔，原来 CSS 也能这么玩”。我所预期的这本书的读者是，学习 CSS、SVG 不久的前端工程师、或是对它们感兴趣的开发者；如果你已是非常有经验的前端开发者，希望你看完之后，也愿意对这本小册子的内容指点一二。

每一小节的内容都是一篇独立的话题，你可以从任何一处翻起。我们将在小节的开始给出设计图，然后一步一步将它实现，同时对一些实现原理作详细描述。代码使用 SCSS 的部分基础功能来精简代码和提高可阅读性。

好了，我们开始吧。

SVG 做图像处理 (mask 篇)

如果读者小时候使用过 Windows XP + IE6/7/8 的系统，一定会发现系统里内置了许多有意思的图形变换特效（在 Powerpoint 2000 这些老古董里也可以找到）。



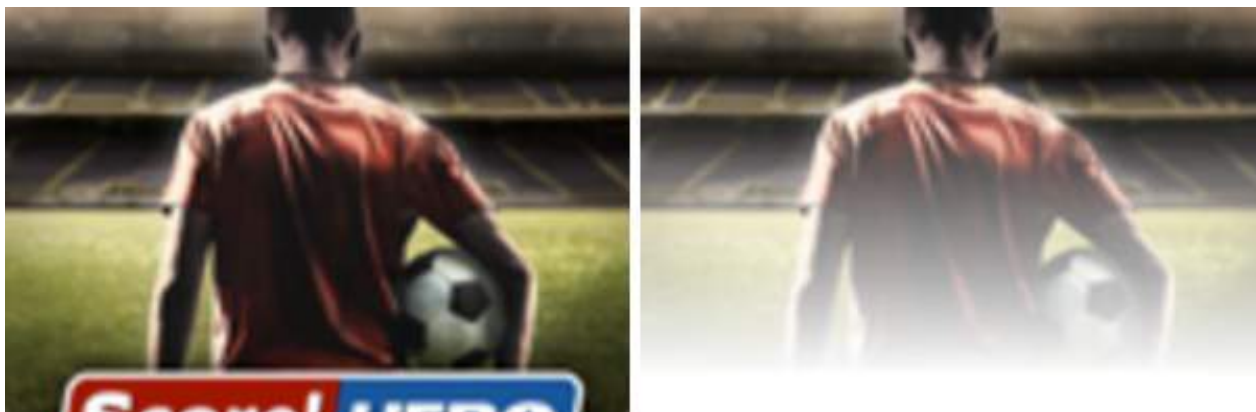
[https://msdn.microsoft.com/en-us/library/ms533001\(v=vs.85\).aspx.aspx](https://msdn.microsoft.com/en-us/library/ms533001(v=vs.85).aspx.aspx)

图中的变换效果称作 Gradient Wipe，在 IE 6 ~ 8 中可以用

`filter:progid:DXImageTransform.Microsoft.Wipe(GradientSize=1.0, wipeStyle=0, motion='forward')` 直接声明；在现代的浏览器里，我们可以用 SVG 的 `mask` 来实现。

Gradient Mask

在讨论 Gradient Wipe 之前，我们先看看 SVG 的遮罩怎么用。



首先准备好一张图片：

```
<svg width="300" height="200">
  <image xlink:href="http://7d9o0k.com1.z0.glb.clouddn.com/s28
</svg>
```

`preserveAspectRatio` 的 `xMidYMid` 类似于 CSS 里的 `background-position: center` , `slice` 类似于 CSS 里的 `background-size: cover` 。

接下来定义一个 `svg mask`：

```
<svg id="svg-mask">
  <defs>
    <linearGradient id="gradient" x1="0" y1="0" x2="0" y2="1"
      <stop stop-color="white" offset="0.8"></stop>
      <stop stop-color="black" offset="0.95"></stop>
    </linearGradient>
    <mask id="mask" maskUnits="objectBoundingBox" maskContentUnits="objectBoundingBox">
      <rect x="0" y="0" width="1" height="1" fill="url(#gradient)"></rect>
    </mask>
  </defs>
</svg>
```

我们首先定义了一个 `linear gradient`, 渐变位置将从 80% 到 95% , 由于期望 0% ~ 80% 部分为可见, 设置为白色。遮罩与图像重合后, 将保留白色区域, 隐去黑色区域。接下来定义了一个 `id` 为 `mask` 的遮罩, `maskUnits` 与 `maskContentUnits` 设置为 `objectBoundingBox` 保证 `mask` 作用于整张图像。`<mask>` 中的内容为一个填充了 `#gradient` 的矩形, 这张遮罩在作用时将拉伸为同图像一样的尺寸。

最后, 我们将遮罩作用于图像, 只用简单添加一个 `mask` 属性：

```
<svg width="300" height="200">  
  <image xlink:href="http://7d9o0k.com1.z0.glb.clouddn.com/s28  
</svg>
```

点击查看这个例子: <http://codepen.io/simonday/full/PZLZmZ/>

Gradient Wipe

在遮罩的基础上，我们可以很轻松的实现文章开头的 wipe transition 效果了。



<http://codepen.io/simonday/full/aOmWEY/>

以 `x=0, y=0` 为起点放置两张相同的图片，对上一张作灰度处理（使用一个 `type="saturate" values="0"` 的 `feColorMatrix`，将在下篇中讲解），同时添加一个遮罩。遮罩在最初为全白色，因此一开始并不起到视觉上遮住什么东西的作用。

```
<svg width="0" height="0">
  <defs>
    <filter id="grayscale">
      <feColorMatrix type="saturate" values="0" />
    </filter>
    <linearGradient id="gradient", x1="0", y1="0", x2="100%"
      <stop stop-color="black", offset="0"></stop>
      <stop stop-color="white", offset=".1"></stop>
    </linearGradient>
    <mask id="mask" maskUnits="objectBoundingBox" maskContent
      <rect x="0" y="0" width="2" height="1" fill="url(#gr
    </mask>
  </defs>
</svg>
<svg width="1762" height="278">
  <image xlink:href="http://anno.qiniudn.com/color.jpg" width=
  <image xlink:href="http://anno.qiniudn.com/color.jpg" width=
</svg>
<div>
  <button class="button-animate">Animate!</button>
</div>
```

逐渐增加 `linearGradient` 的第二个 `stop` 的 `offset` 值，从左侧起黑色部分将逐渐增多，此时顶部的灰色图从左侧起开始消失，显示出底部的彩色部分。我们可以描述逐渐增加 `stop` 值的过程为：

```
var $ = function(sel) {  
    return document.querySelector(sel)  
}  
var gradientStop = $('#gradient stop:last-child')  
    , startOffset = 0  
function wipe() {  
    if (startOffset < 1) {  
        startOffset += 0.01  
        gradientStop.setAttribute('offset', startOffset)  
        requestAnimationFrame(wipe)  
    }  
}  
$('#.button-animate').addEventListener('click', function() {  
    startOffset = 0  
    wipe()  
})
```

总结

使用 `Canvas`，我们也可以通过 `getImageData` 获得彩色图像信息，再用 `putImageData` 修改其中的 `alpha` 通道来实现同样的效果。但 SVG 内置的滤镜和遮罩，使得我们很轻松直观地实现了 Gradient Wipe 的效果。