

# SVG <foreignObject>简介与截图等应用

这篇文章发布于 2017年08月4日, 星期五, 23:59, 归类于 [SVG相关](#)。阅读 24928 次, 今日 22 次 [21 条评论](#)

by zhangxinxu from <http://www.zhangxinxu.com/wordpress/?p=6326>

本文可全文转载, 但需得到原作者书面许可, 同时保留原作者和出处, 摘要引流则随意。

## 一、从SVG文本换行说起

SVG和CSS几乎可以看成是同一个年代出来的东西, 但是在Web界面展示这块, CSS长期统领江山, SVG偃旗息鼓, 直到这些年, SVG开始乘势而上。

时势造英雄。当年web网页都是以图文展示为主, 所以门户, 博客这些网站兴起。而SVG擅长的是图形展示, 对于文本呈现, 只能嘿嘿一下, 跟CSS相比那可弱了几条街。举个最简单的效果, 文本换行。在CSS下, 当我们一段文字很长的时候, 是会自动换到下一行的, 但是在SVG中, 基本上就是要手动控制换行, 对于新闻、博客展示而言简直就是噩梦。于是, CSS火了, SVG萎了。

但, 时代是发展的, 现代网络带宽有大腿粗, 电脑性能比第一次还快, 屏幕密度比姚明还高, 丰富的富媒体展现和复杂的图形表现需求越来越高, 随着flash的日渐衰微, 显然, 规范标准化的SVG就顺势崛起。

然而需求是千变万化的, 当我们使用SVG实现一些视觉表现精湛的效果的时候, 免不了会有增加一段描述文字的需求。呵呵, 这个时候SVG怕是要傻眼了, SVG的 `<text>` 元素应付多行文字实在不擅长, 如果SVG中的文字可以如同CSS中的表现一样就好了?

嘿, 还真有, 那就是SVG的 `<foreignObject>` 元素。

## 二、SVG forginObject元素简介

我们平常使用SVG使用的是SVG的命名空间:

```
<svg xmlns="http://www.w3.org/2000/svg"></svg>
```

`xmlns` 全称是“XML Namespaces”, 指“XML命名空间”。XML是一个比较大的统称, 是一种用于标记电子文件使其具有结构性的标记语言。我们平时写的HTML代码是XML, [RSS订阅文档](#)也是种XML, SVG也属于XML。而这些XML类型的渲染规则是有差异的, 例如HTML中裸露的 `<circle>` 元素是不会表现为圆, 而是一个普通的自定义元素。RSS中的XML浏览器会自动进行文章排版渲染等。

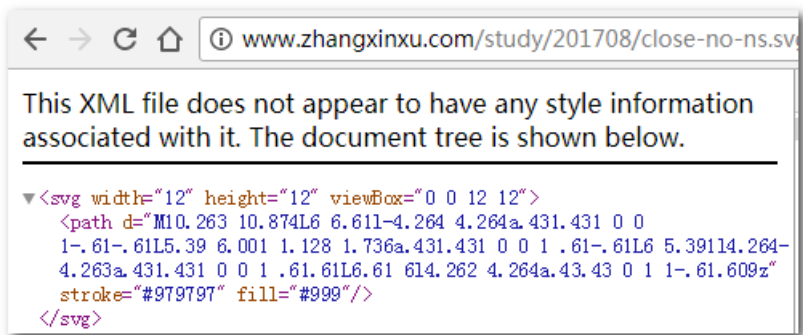
所以我们可以看到, 指定命名空间可以让浏览器精准解析, 举个例子, 如下XML代码:

```
<svg width="12" height="12" viewBox="0 0 12 12"><path d="M10.263 10.874L6 6.611-4.264 4.264a.431.431 0 0 1-.61-.61L5.39 6.001 1.128 1.736a.431.431 0 0 1 .61-.61L6 5.391L4.264-4.263a.431.431 0 0 1 .61.61L6.61 6L4.262 4.264a.43.43 0 1 1-.61.609z" stroke="#979797" fill="#99"/></svg>
```

虽然是SVG标签, 明摆着应该安装SVG渲染, 但是, 实际上, 由于没有指定 `xmlns`, 在浏览器中打开的时候, 直接以普通XML文档树渲染了。

可狠狠的点击此SVG文件亲自感受下: [close-no-ns.svg](#)

Chrome浏览器下：



Firefox浏览器下：



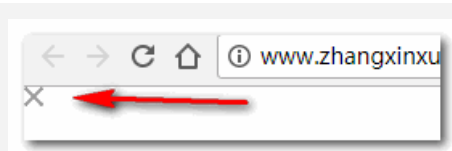
一旦我们加上了SVG专属命名空间，如下代码：

```
<svg width="12" height="12" viewBox="0 0 12 12" xmlns="http://www.w3.org/2000/svg"><path d="M10.263 10.874L6 6.61l-4.264 4.264a.431.431 0 0 1-.61-.61L5.39 6.001 1.128 1.736a.431.431 0 0 1 .61-.61L6 5.391l4.264-4.263a.431.431 0 0 1 .61.61L6.61 6l4.262 4.264a.43.43 0 1 1-.61.609z" stroke="#979797" fill="#999"/></svg>
```

则符合预期的SVG图形就渲染出来了。

可狠狠的点击此SVG文件亲自感受下：[close.svg](#)

结果如下截图：



这就是命名空间的作用。但这里有必要额外说明下：如果我们的SVG文件是直接内联在XHTML页面中，是可以不指定命名空间的！

好，回到 <foreignObject> 这里，<foreignObject> 元素的作用是在其中使用具有其它XML命名空间的XML元素，换句话说借助 <foreignObject> 标签，我们可以直接在SVG内部嵌入XHTML元素，举个简单的例子：

```
<svg xmlns="http://www.w3.org/2000/svg">
  <foreignObject width="120" height="50">
```

```
<body xmlns="http://www.w3.org/1999/xhtml">
  <p>文字。</p>
</body>
</foreignObject>
</svg>
```

可以看到 `<foreignObject>` 标签里面有一个设置了 `xmlns="http://www.w3.org/1999/xhtml"` 命名空间的 `<body>` 标签，此时 `<body>` 标签及其子标签都会按照XHTML标准渲染，实现了SVG和XHTML的混合使用。

这种混合特性有什么作用呢？作用很多，其中之一就是轻松实现SVG内的文本自动换行。

### 三、SVG `foreignObject`元素与文本自动换行

SVG要实现文本换行，往往需要手动阻断，类似下面的代码：

```
<svg xmlns="http://www.w3.org/2000/svg">
  <text font-size="12">
    <tspan x="0" y="10">一段需要word wrap</tspan>
    <tspan x="0" y="26">的文字。</tspan>
  </text>
</svg>
```

需要 2 个 `<tspan>` 元素，这一点都不工程。虽然Chrome浏览器可以对 `<text>` 标签进行 `white-space:normal` 的强制设置，但也只是Chrome浏览器可以。

但是如果使用 `<foreignObject>` 元素，则自动换行就是小菜：

```
<svg xmlns="http://www.w3.org/2000/svg">
  <foreignObject width="120" height="50">
    <body xmlns="http://www.w3.org/1999/xhtml">
      <p style="font-size:12px;margin:0;">一段需要word wrap的文字。</p>
    </body>
  </foreignObject>
</svg>
```

结果Chrome浏览器下效果为：



您可以狠狠地点击这里：[foreignObject下文本自动换行demo](#)

### 四、SVG `foreignObject`元素其它作用-图片生成

除了轻松实现文本换行，SVG `<foreignObject>` 元素还有其他更高级的应用，就是可以将页面上的DOM元素轻松变成图片，原理如下：

1. 获取对应DOM元素的 `outerHTML` 代码；
2. 放在 `<foreignObject>` 元素中；
3. 图片方式显示我们的SVG图形，例如：

```
<img width="300" height="150" src='data:image/svg+xml;charset=utf-8,<svg xmlns="http://www.w3.org/2000/svg"><foreignObject width="120" height="50"><body xmlns="http://www.w3.org/1999/xhtml"><p style="font-size:12px;margin:0;">一段需要word wrap的文字。</p></body></foreignObject></svg>'>
```

```
ody></foreignObject></svg>'>
```

4. 上一步的图片本质还是SVG，我们可以借助 `canvas` `drawImage()` 方法将图片放在画布上，然后使用 `canvas.toDataURL()` 方法转换成 `png` 或 `jpg` 图片，核心代码：

```
var canvas = document.createElement('canvas');
var context = canvas.getContext('2d');
canvas.drawImage(img, 0, 0);
img.src = canvas.toDataURL('image/png');
```

上一篇文章“[canvas实现图片前端JS压缩](#)”有详细的canvas绘图和转图片示意，有兴趣可以参考下。

一旦我们可以把DOM元素转换成图片，我们就可以轻轻松松配合JS在前端实现网页截图功能。

想了想，还是做个demo示意下吧，您可以狠狠地点击这里：[借助forignObject实现DOM转图片并直接下载demo](#)

点击demo页面右侧小模块任意区域，都可以保存想要内容，例如，外面框框：

结果：

触发保存，[保存后的图片](#)（点击可预览）就是DOM页面的样子。

补充说明：理论上，借助 `<foreignObject>` 实现截图效果Firefox也是支持的，但是，上面例子为了简化，直接把页面上的 `<style>` 元素以及里面CSS代码一起内联到SVG的 `<foreignObject>` 中了，这种用法只有Chrome浏览器支持。因此，上面demo效果仅在Chrome及其内核浏览器下有效。Firefox浏览器要想支持，需要使用特殊方法将所有元素样式 `style` 属性内联在元素上。

## SVG forignObject兼容性

可以看出SVG的 `<foreignObject>` 的特性非常棒！但相比其他SVG元素兼容性要差一些，主要是IE浏览器，好在大头Chrome和Firefox都支持，Safari虽然支持 `<foreignObject>` 貌似由于安全限制，转换成图片功能受限。

## 五、结束语

SVG `<foreignObject>` 元素可以让DOM变图片，发挥我们的创造力，我们可以做的事情就非常非常多，举个例子，各种特效，因为DOM图片化，我们就能使用 `canvas` 读取图片完整的像素点信息，通过特点的算法，我们就可以做很多效果，例如常见的高斯模糊。高斯模糊浏览器天然支持，那我们可以实现浏览器默认没有的动感模糊或者镜像模糊；又或者我们可以把DOM元素变成粒子特效（`canvas`擅长的）；又或者实现类似OS X系统的DOCK中软件打开和收起的拉伸扭曲效果等等。

有技术，有创意，还不赶快行动，把你的网页炫起来！

《CSS世界》签名版独家发售，包邮，可指定寄语，点击显示购买码

（本篇完） // 想要打赏？点击[这里](#)。有话要说？点击[这里](#)。





« HTML5 file API加canvas实现图片前端JS压缩并上传

HTML5按钮元素新属性formaction,formenctype等简介 »

### 猜你喜欢

- 小tips:使用canvas在前端实现图片水印合成
- canvas文本绘制自动换行、字间距、竖排等实现
- 图片动态局部毛玻璃模糊效果的实现
- 瞎折腾：把JS,CSS任意文本文件加密成一张图片
- word-break:break-all和word-wrap:break-word的区别
- JS检测PNG图片是否有透明背景、抠图等相关处理
- 借助wbr标签实现连续英文字符的精准换行
- 有人考了我一道CSS题目
- JS前端创建html或json文件并浏览器导出下载
- 学习了，CSS中内联SVG图片有比Base64更好的形式
- HTML5 Canvas,WebGL,CSS Shaders,GLSL的暧昧关系

分享到： 0

标签：`canvas`, `download`, `drawImage`, `foreignObject`, `SVG`, `toDataURL`, `white-space`, `xmlns`

### 发表评论（目前21条评论）

名称 (必须)

邮件地址(不会被公开) (必须)

网站

提交评论

#### 1. Nick说道：

2018年10月19日 12:25

使用foreignObject换行时，我是通过js动态的向p标签下添加文本，结果添加的文本在页面上没有显示。我的添加代码是：

```
var svg = d3.select("svg");

svg.append("switch")
.append("foreignObject")
.attr("x", 20)
.attr("y", 0)
.attr("width", 150)
.attr("height", 200)
.attr("style", "outline: 1px solid blue;")
```



```
.append("body")
.append("p")
.attr("xmlns", "http://www.w3.org/1999/xhtml")
.text("Text goes here, Text goes here, Text goes here, Text goes here");
```

[回复](#)

2. 一条鱼说道:

2018年09月13日 15:42

把您的代码抓下来尝试了一下, 图片是本地的, 但是一直报这个错“Tainted canvases may not be exported.”, 搞不懂是怎么回事? ? ?

[回复](#)



3. 咸鱼康说道:

2018年08月25日 14:34

看了您的文章,手动操作后遇到了一个问题,就是svg转成img后字体的大小不对,比如我在html里设置了字体的大小为60px.转成svg后市正常的,但是由svg转成img后,img的字体大小就变成了默认的了.....总觉得和svg有问题,但找不到问题的所在,求大佬帮忙看看

[回复](#)



4. v8eejing说道:

2018年08月21日 14:16

看了你的文章实在很精彩, 在截取的时候遇到两个问题, 1、遇到视频标签出了点问题, 2、遇到滚动文本只有样式没有内容。..... 觉得和样式有点关系, 求助大神。

[回复](#)



张 鑫旭说道:

2018年08月22日 17:23

恕我愚钝, 没明白你所说的具体是个什么问题。

[回复](#)



5. domtoimage.download is not a function 说道:

2018年05月25日 11:17

已经安装了dom-to-image插件, 想问这是是什么原因引起的呢

[回复](#)



6. jsoncode说道:

2018年04月23日 00:31

根据这篇教程, 写了一个html2img的插件, 中间遇到一个问题, html中包含svg, 并且svg中包含xlink:href="#id", 就无法到处图片, 关键再这个井号, 去掉井号就可以, 但是去掉井号, 明显这句引用就失效了, 不知道大大有什么解决办法。我搜了一晚上, 都没找到, 现在已经凌晨了。

测试代码: html:

```
<svg viewBox="0 0 100 100">
<path d="M 0,50 a 50,50 0 1,1 0,1 z" id="circle"/>
<text>
<textPath xlink:href="#circle">test</textPath>
</text>
</svg>
```

[回复](#)



张鑫旭说道:

2018年04月23日 21:52

会不会是id重复或者其他什么原因啊。

[回复](#)



7. **clarence**说道:

2018年03月30日 11:33

尝试了一下

- 1.必须使用内联样式;
- 2.rem不生效;
- 3.元素的背景图片获取不到

[回复](#)

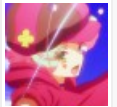


8. **Handle**说道:

2018年03月16日 11:47

请问foreignObject元素的真实宽高是否有办法获取到手

[回复](#)



9. **svg**上添加的**foreignObject**如何设置坐标，**foreignObject**的坐标我用x y设置之后，g元素的坐标就被干掉了说道:

2018年02月8日 15:54

请问下svg上添加的foreignObject如何设置坐标，foreignObject的坐标我用x y设置之后，g元素的坐标就被干掉了

[回复](#)



10. **小白**说道:

2017年12月8日 17:22

demo的方法css样式是不是会被清掉

[回复](#)

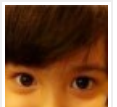


11. **老猪**说道:

2017年11月22日 20:28

foreignObject 这玩意不晓得能不能直接引入link css，  
整了半天样式就是出不来，换了style的方式果然是可以。。唉

[回复](#)



meow-z说道:

2018年07月9日 18:40

我也是，样式表没有了、、、只有内联样式可以存在，有解决方案么？~

[回复](#)



12. **bobalan**说道:

2017年11月22日 15:06

如何用foreignObject实现类似iOS那样的毛玻璃效果?

这个实验进行的怎么样啦

[回复](#)



13.

zhangya说道:

2017年11月15日 15:11

请问一下，”Firefox浏览器要想支持，需要使用特殊方法将所有元素样式style属性内联在元素上。”这里的特殊方法是指什么方法？还有‘内联在元素上’，这个元素是什么元素？

回复


14.

华陈说道:

2017年08月16日 16:27

生命不息，折腾不止

回复


15.

曾小乱说道:

2017年08月7日 23:48

怎么把rss订阅设置为摘要了？ 这样不友好。

回复


16.

马里奥说道:

2017年08月5日 08:43

svg绘制的二维图形可以按照以上方式截图保存么？

回复

张鑫旭说道:

2017年08月5日 11:42

自然可以！

回复




17.

jmz说道:

2017年08月5日 00:42

比第一次还快，嘿嘿嘿，是我想太多吗

回复



#### 最新文章

- » 常见的CSS图形绘制合集
- » 粉丝群第1期CSS小测点评与答疑
- » 分享三个纯CSS实现26个英文字母的案例
- » 小tips: 纯CSS实现打字动画效果
- » CSS/CSS3 box-decoration-break属性简介
- » CSS :placeholder-shown伪类实现Material Design占位符交互效果
- » 从天猫某活动视频不必要的3次请求说起
- » CSS vector-effect与SVG stroke描边缩放
- » CSS ::backdrop伪元素是干嘛用的？
- » 周知：CSS -webkit-伪元素选择器不再导致整行无效

#### 今日热门

- » 常见的CSS图形绘制合集 <sup>(190)</sup>
- » 未来必热：SVG Sprite技术介绍 <sup>(119)</sup>
- » 粉丝群第1期CSS小测点评与答疑 <sup>(115)</sup>
- » HTML5终极备忘大全（图片版+文字版） <sup>(93)</sup>
- » 让所有浏览器支持HTML5 video视频标签 <sup>(86)</sup>
- » Selectivizr-让IE6~8支持CSS3伪类和属性选择器 <sup>(82)</sup>
- » CSS3下的147个颜色名称及对应颜色值 <sup>(79)</sup>



- » [视区相关单位vw, vh..简介以及可实际应用场景](#) <sup>(76)</sup>
- » [写给自己看的display: flex布局教程](#) <sup>(76)</sup>
- » [小tips: 纯CSS实现打字动画效果](#) <sup>(76)</sup>



#### 今年热议

- » [《CSS世界》女主角诚寻靠谱一起奋斗之人](#) <sup>(76)</sup>
- » [不借助Echarts等图形框架原生JS快速实现折线图效果](#) <sup>(64)</sup>
- » [看, for..in和for..of在那里吵架!](#) <sup>(60)</sup>
- » [是时候好好安利下LuLu UI框架了!](#) <sup>(47)</sup>
- » [原来浏览器原生支持JS Base64编码解码](#) <sup>(35)</sup>
- » [妙法攻略: 渐变虚框及边框滚动动画的纯CSS实现](#) <sup>(33)</sup>
- » [炫酷H5中序列图片视频化播放的高性能实现](#) <sup>(31)</sup>
- » [CSS scroll-behavior和JS scrollIntoView让页面滚动平滑](#) <sup>(30)</sup>
- » [windows系统下批量删除OS X系统.DS\\_Store文件](#) <sup>(26)</sup>
- » [写给自己看的display: flex布局教程](#) <sup>(26)</sup>

#### 猜你喜欢

- [小tips:使用canvas在前端实现图片水印合成](#)
- [canvas文本绘制自动换行、字间距、竖排等实现](#)
- [图片动态局部毛玻璃模糊效果的实现](#)
- [瞎折腾: 把JS,CSS任意文本文件加密成一张图片](#)
- [word-break:break-all和word-wrap:break-word的区别](#)
- [JS检测PNG图片是否有透明背景、抠图等相关处理](#)
- [借助wbr标签实现连续英文字符的精准换行](#)
- [有人考了我一道CSS题目](#)
- [JS前端创建html或json文件并浏览器导出下载](#)
- [学习了, CSS中内联SVG图片有比Base64更好的形式](#)
- [HTML5 Canvas,WebGL,CSS Shaders,GLSL的暧昧关系](#)