

使用CSS offset-path让元素沿着不规则路径运动


这篇文章发布于 2017年03月5日, 星期日, 23:09, 归类于 [CSS相关](#)。阅读 26812 次, 今日 9 次 [9 条评论](#)

by zhangxinxu from <http://www.zhangxinxu.com/wordpress/?p=5989>

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、言左右而顾其他

要让一个元素按照不规则路径进行运动, 最好的办法就是使用“SVG SMIL animation”, 我之前有写过文章专门介绍, 参见[超级强大的SVG SMIL animation动画详解](#), 并且除了IE浏览器以外, 其他浏览器的支持情况都蛮不错的, 如下截图所示:

SVG SMIL animation  - REC

Method of using animation elements to animate SVG images

IE	Edge	Firefox	Chrome	Safari	Opera	iOS	Opera	Android	Chrome	Browser	QQ
						Safari	Mini	Browser	for	for	Browser
									Android	Android	
			49					4.4			
			55			9.3		4.4.4			
11	14	51	56	10	43	10.2	all	53	56	11	1.2
	15	52	57	10.1	44						
		53	58	TP	45						
		54	59								

张 <http://www.zhangxinxu.com>
鑫旭-鑫空间-鑫生活

“SVG SMIL animation”虽然强大, 但是由于是基于HTML属性生成的各类效果, 因为就容易存在复用的问题, 例如不同的位置的不同元素的不规则路径动画是一样的, 那么我在设置的时候, 要么冗余啰嗦, 要么交叉不利于维护, 有点早些年在HTML标签上写 `style` 风格的味道。

或许是这个原因, Chrome浏览器开始有了放弃“SVG SMIL animation”的迹象, 转而拥抱经过几十年成功验证的CSS来实现, `offset-path` 几乎可以看成是Chrome浏览器让元素沿着不规则路径运动的新宠儿。

但毕竟是新宠儿, 因此, 相比较“SVG SMIL animation”, 其兼容性还是差了两条街的。

CSS Motion Path - UNOFF

Allows elements to be animated along SVG paths or shapes

IE	Edge	Firefox	Chrome	Safari	Opera	iOS Safari	Opera Mini	Android Browser	Chrome for Android	Browser for Android	QQ Browser
			49					4.4			
			55			9.3		4.4.4			
11	14	51	56	10	43	10.2	all	53	56	11	1.2
	15	52	57	10.1	44						
		53	58	TP	45						
		54	59								

但是有一些内部的项目只需要兼容浏览器，因此，实际上 `offset-path` 也是有武之地的。

二、offset-path之前并不叫做offset-path

`offset-path` 属性一开始的时候并不叫做 `offset-path`，而叫做 `motion-path`，这是因为开始有规范对其进行明确定义了，如下截图示意：

W3C Editor's Draft	1	Introduction
	2	Module interactions
	3	Values
	4	Motion Paths
	4.1	Define a path: The <u>'offset-path'</u> property
	4.2	Position on the path: The <u>'offset-distance'</u> property
	4.2.1	Calculating the computed distance along a path
	4.3	Define the starting point of the path: The <u>'offset-position'</u> property
	4.4	Define an anchor point: The <u>'offset-anchor'</u> property
	4.5	Rotation at point: The <u>'offset-rotate'</u> property

其实不止 `motion-path`，还有其他 `motion-*` 属性也都开始变成 `offset-*` 开头的了，例如原来的 `motion-offset` 现在规范用法是 `offset-distance`。

根据一些资料的说法，之前语法开始于2015年9月，会在M58版本会移除，大约2017年4月，也就是说，等到下个月 `motion-path` 这些属性Chrome就会不念旧情，残忍抛弃。为了避免发生如此惨绝人寰的事情，所以从现在开始，我们都开始使用新的规范的属性名称。

不过本文的demo实例还是新老语法一起混用的，因为demo的主要目的是演示，不代表实际的应用，所以大家大可不必在意这个细节。

三、offset-path让元素沿着不规则路径运动

使用CSS属性让元素不规则运动要比使用HTML属性控制简单得多，比方说我们只需要下面几行CSS，就可以实现我们想要的效果了，例如：

```
.horse-run {
  offset-path: path("M10,80 q100,120 120,20 q140,-50 160,0");
  animation: move 3s linear infinite;
}
@keyframes move {
  100% { offset-distance: 100%;}
}
```

就可以实现一个马儿沿着不规则路径不停跑的效果。您可以狠狠地点击这里：[CSS offset-path实现的马儿不规则路径不停跑demo](#)



其中用到了两个CSS属性，一个是 `offset-path`，表示运动的路径，另外一个则是 `offset-distance`，我是运动的距离，可以是数值或者百分比单位，如果是 `100%` 则表示正好把所有的路都跑完了。

四、其他offset-*运动相关属性值

除了 `offset-path` 和 `offset-distance` 这两个CSS属性，还有其他一些相关的属性，例如 `offset-rotation`（规范上显示的是 `offset-rotate`，浏览器是无效不识别的），表示运动的角度，默认是 `auto`，表示自动计算当前路径的切线方向，并朝着这个方向前进，所以，上面的马儿会有会自动爬坡的即视感。

但是如果我们定死了一个角度，例如设置：

```
offset-rotation: 30deg
```

则这个马儿一看就知道昨晚没睡好，得了落枕，头抬不起来只能保持一个姿势：



除了设定固定的角度值，我们还可以使用关键字属性值，例如：

```
offset-rotation: reverse;
```

则马儿立马上下颠倒跑起来，如下截图：



我们甚至可以把属性值组合起来，例如：

```
offset-rotation: auto 30deg;
```

表示在原来的切线方向上，再旋转30度，例如，之前值为 `auto` 的时候，马儿在平地上是朝前看的，现在则是看自己的蹄子美不美：



除了 `offset-rotation`，还有其他相关CSS属性，包括 `offset-position` 和 `offset-anchor`。

其中，`offset-anchor` 表示锚定的中心点，其属性值和 `transform-origin` 类似，可以是：

```
#item1 {
  offset-anchor: right top;
}
#item2 {
  offset-anchor: right bottom;
}
#item3 {
  offset-anchor: left bottom;
}
#item4 {
  offset-anchor: left top;
}
#item5 {
  offset-anchor: center;
}
...
```

等等，元素运动的时候，会让这个点和路径重合进行运动。。

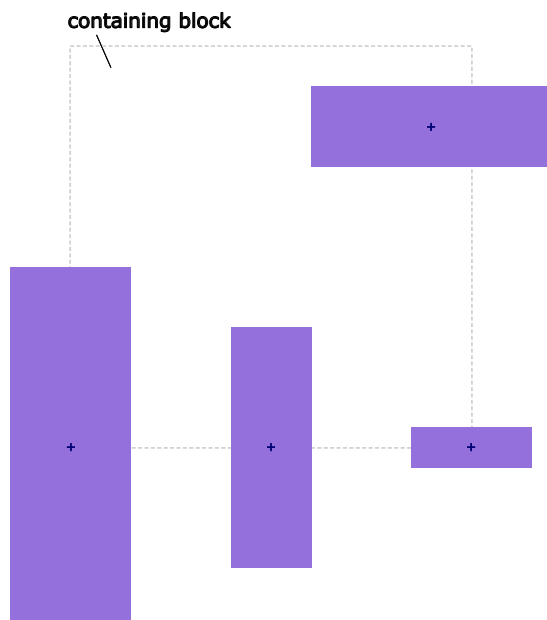
`offset-position` 指定路径的初始位置，行为类似于属性 `background-position` 。

根据我的测试，Chrome浏览器虽然认得 `offset-position` 和 `offset-anchor` 这两个属性，但是马儿并没有任何的变化，哪怕有一像素的位移或者旋转之类的。可能是我使用的方式不对，所以只能从规范的示例上挖掘这两个属性的表现：

已知四个元素 `offset-position` 值分别是：

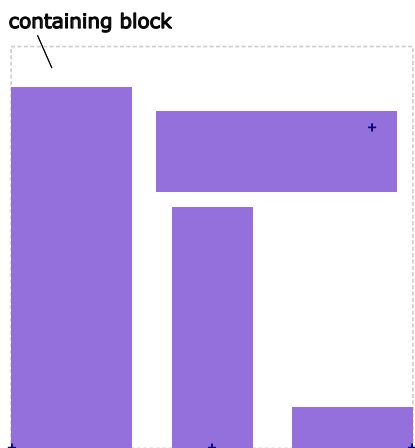
```
offset-position: 90% 20%;
offset-position: 100% 100%;
offset-position: 50% 100%;
offset-position: 0% 100%;
```

当 `offset-anchor` 值为 `center` 的时候，表现如下图（SVG图，IE9+）：



上图中的加号表示的就是 `offset-anchor` 确定的锚点位置。

当 `offset-anchor` 值为 `auto` 的时候，则表现就跟CSS2.1中 `background-position` 属性的百分比解析一模一样了：



图片示意的很清楚，还是很好理解的。如果按照上面两张规范图所示的表现的话，浏览器是应该有所表现才对，我猜测很有可能，浏览器还没有对其进行完全的解析，毕竟规范也才更新不久。

五、结束语

个人站点的新版首页上线啦！[点击这里](#)围观。

花了2周时间对进行了大刀阔斧的修改，增加了“好文推荐”和“我的微码”以及“工作机会”这几个模块，并对内页进行了无图片改造，去除几个以前学习目的开发的几个频道等，评论采用了社会化的评论，以前是自己开发的，需要注册才能评论，我自己都觉得麻烦。

有小伙伴说，终于进来不全是广告了，😂

我老婆问我为什么不用黑色底，显得很酷，很有设计感。唉，现在风格已经成型，网站基色已经应用这么久，哪是随便改的了。说起来怪自己当年自由奔放的性格，网站不都是需要一个主色吗？当时我还是xp系统，我就用取色工具在我当时xp系统的工具栏上随便取了个色作为网站的主色，这个颜色就是 `#34538b`，早就记得滚瓜烂熟，然后用配色工具配了个色，结果现在的眼光看那几个颜色，咳咳……实际上，大家现在看到的上面导航栏的颜色已经不是当年配色表显示的颜色了，现在流行高饱和度大色块。算是顺应时势，在几年前我悄悄改了几个主色，提高了明度和饱和度，之前的颜色灰蒙蒙的，感觉有点脏，虽然现在也好不到哪里去，但比起之前的至少上了一层楼了。😂

欢迎反馈使用上的问题，我会慢慢优化，并用数据说话！

《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](#)。



猜你喜欢

- 深度掌握SVG路径path的贝塞尔曲线指令
- SVG+JS path等值变化实现CSS3兴叹的图形动画
- 纯CSS实现帅气的SVG路径描边动画效果
- 超级强大的SVG SMIL animation动画详解
- 再说CSS3 animation实现点点loading动画
- 如何使用Tween.js各类原生动画运动缓动算法
- 借助Web Animations API实现JS keyframes动画
- Canvas中颜色过渡动画效果的实现
- CSS3 Transitions, Transforms和Animation使用简介与应用展示
- web上渐进使用jQuery Mobile中animate相关CSS
- 从height:100%不支持聊聊CSS中的“死循环”

分享到:        1

标签: animation, motion-path, offset-anchor, offset-distance, offset-path, offset-position, offset-rotation, path, 动画

发表评论 (目前9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. 匿名说道:
2018年02月1日 15:25
我来试试这个评论功能
[回复](#)



2. 笑口常开说道:
2018年01月5日 23:11
offset-rotation现在不能用了, 回归了规范, offset-rotate,
另外offset-anchor,offset-position也不能用了
[回复](#)



3.

匿名说道：
2017年09月15日 11:39

张含韵蓝 #34538b

[回复](#)


4.

七月说道：
2017年07月20日 16:59

还是好多广告

[回复](#)


5.

先知先觉说道：
2017年04月18日 10:44

不知道运动路线怎么设呢？一点一点地调太慢了吧

[回复](#)


6.

caiyongmin说道：
2017年04月17日 13:41

不错不错，变得很好看了

[回复](#)


7.

后知后觉说道：
2017年03月27日 09:46

内容为王，别的那些都是浮云

[回复](#)


8.

bestRenekton说道：
2017年03月15日 17:22

有半个月没来 还以为进错了。。。。

[回复](#)


9.

刘东奇说道：
2017年03月9日 17:27

导航栏的颜色的确明亮了很多

[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](#) ⁽¹⁹⁰⁾
- » [未来必热：SVG Sprite技术介绍](#) ⁽¹¹⁹⁾
- » [粉丝群第1期CSS小测点评与答疑](#) ⁽¹¹⁵⁾
- » [HTML5终极备忘大全（图片版+文字版）](#) ⁽⁹³⁾
- » [让所有浏览器支持HTML5 video视频标签](#) ⁽⁸⁶⁾
- » [Selectivizr-让IE6~8支持CSS3伪类和属性选择器](#) ⁽⁸²⁾
- » [CSS3下的147个颜色名称及对应颜色值](#) ⁽⁷⁹⁾
- » [视区相关单位vw, vh..简介以及可实际应用场景](#) ⁽⁷⁶⁾
- » [写给自己看的display: flex布局教程](#) ⁽⁷⁶⁾
- » [小tips: 纯CSS实现打字动画效果](#) ⁽⁷⁶⁾



今年热议

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

猜你喜欢

- [深度掌握SVG路径path的贝塞尔曲线指令](#)
- [SVG+JS path等值变化实现CSS3兴叹的图形动画](#)
- [纯CSS实现帅气的SVG路径描边动画效果](#)
- [超级强大的SVG SMIL animation动画详解](#)
- [再说CSS3 animation实现点点点loading动画](#)
- [如何使用Tween.js各类原生动画运动缓动算法](#)
- [借助Web Animations API实现JS keyframes动画](#)
- [Canvas中颜色过渡动画效果的实现](#)
- [CSS3 Transitions, Transforms和Animation使用简介与应用展示](#)
- [web上渐进使用jQuery Mobile中animate相关CSS](#)
- [从height:100%不支持聊聊CSS中的“死循环”](#)