

3种纯CSS实现中间镂空的12色彩虹渐变圆环方法

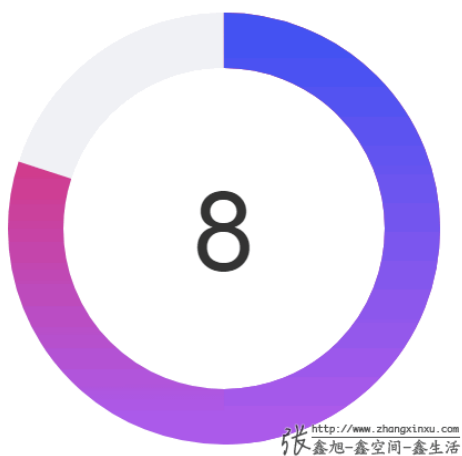
这篇文章发布于 2017年11月7日, 星期二, 00:17, 归类于 [CSS相关](#)。阅读 38071 次, 今日 28 次 [11 条评论](#)

by zhangxinxu from <http://www.zhangxinxu.com/wordpress/?p=6516>

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实现多彩圆环倒计时效果说起

上周我[做了个demo](#), 使用SVG和[stroke-dasharray](#)和[stroke-dashoffset](#)特性实现了一个彩条圆环倒计时效果, 大概长下面这样子:



原理是使用两个渐变半圆无缝叠加在一起。

使用SVG实现的优点是兼容性非常好, IE9浏览器也是支持的。

不足在于学习成本比较高, 相关的SVG代码要想完整的手写出来, 还是需要不少积累的。

于是我就琢磨有没有更简单的方法实现类似的多彩圆环渐变效果, 最好纯CSS就能搞定。

绞尽脑汁想出了下面三种实现方法, 为了尽可能验证方法的可行性, 我把圆环的渐变颜色从3色变成了12色。

下面依次介绍这三种方法。

二、借助CSS3 conic-gradient锥形渐变实现12色渐变圆环

环形带上的渐变效果, 本质上就是一种锥形渐变, 因此, 我们直接使用, CSS3 `conic-gradient` 锥形渐变实现应该是最简单最直接的方法。

如下HTML和CSS:

```
<div class="circle"></div>
```

```
.circle {  
  width: 300px; height: 300px;  
  background: conic-gradient(#9ED110, #50B517, #179067, #476EAF, #9f49ac, #CC42A2, #FF3BA7, #FF5800, #FF8100, #FEAC00, #FFCC00, #EDE604);  
}
```

```
border-radius: 50%;  
}
```

此时的效果如下图所示：



但是我们需要的效果不是圆饼，而是一个圆环，也就是中间区域需要镂空。

有一种比较巧妙的做法，就是搞一个跟背景色一样半径小一点的圆形饼覆盖在中间，这样视觉看上去就是一个圆环，这也是一个不错的方法。但是如果我们的背景是一个复杂的图案，此时，这种方法恐怕就不能适用。

你怎么需要找到一种中间区域是真实镂空的方法，怎么处理呢？

我们可以借助CSS3遮罩 `mask` 属性来实现，CSS3 `mask` 遮罩除了支持PNG图片，SVG图形，其还支持CSS3渐变背景图。于是，我们只需要使用CSS3径向渐变 `radial-gradient` 语法生成一个中间透明，边缘实色的圆环即可。

于是有（这里省略了 `mask` 属性的 `webkit` 私有前缀）：

```
.circle {  
    width: 300px; height: 300px;  
    background: conic-gradient(#9ED110, #50B517, #179067, #476EAF, #9f49ac, #CC42A2, #FF3BA7, #FF5800, #FF8100, #FEAC00, #FFCC00, #EDE604);  
    border-radius: 50%;  
    mask: radial-gradient(transparent 110px, #000 110px);  
}
```

于是我们的12色圆环渐变效果就实现了：



再配合JS设置 `clip-path` 剪裁我们的图形，就可以实现12色的彩虹圆环倒计时效果了。

眼见为实，您可以狠狠地点击这里：[借助CSS3 mask遮罩和conic-gradient实现的多彩圆环demo](#)

优点和不足

这个方法看上去很简单，很完美，但是却有个致命的缺陷，那就是兼容性问题！

CSS3锥形渐变 `conic-gradient` 目前仅Chrome浏览器才支持，并且Chrome浏览器还是需要开启实验功能选项才能支持！



因此当下此方法基本上没有在实际项目中应用的可能性，需要看看有没有什么其他解决方案？

三、借助CSS3 `linear-gradient`线性渐变近似实现12色渐变圆环

虽然锥形渐变浏览器的兼容性不怎么样，但是线性渐变它兼容性很好啊，是不是可以李代桃僵呢？

和一开始SVG圆环原理类似：

把我们的圆环分成左半部分和右半部分，左半部分是一套线性渐变，右半部分是另外一套线性渐变，然后最底部的线性渐变颜色是一样的，理论上在视觉上看起来就能够形成一个完美的环。

如下HTML和CSS代码：

```
<div class="circle">
  <div class="circle-left"></div>
  <div class="circle-right"></div>
</div>
```

```
.circle {
  width: 300px; height: 300px;
  border-radius: 50%;
  mask: radial-gradient(transparent 110px, #000 110px);
  overflow: hidden;
}
.circle-left {
  width: 50%; height: 100%;
  background: linear-gradient(to bottom, #EDE604, #FFCC00, #FEAC00, #FF8100, #FF5800, #F3BA7, #CC42A2);
  float: left;
```

```

}
.circle-right {
  width: 50%; height: 100%;
  float: right;
  background: linear-gradient(to bottom, #9ED110, #50B517, #179067, #476EAF, #9f49ac, #C42A2);
}

```

然而最后的效果却是下面这样：



咦，奇怪，同样的实现原理，为什么一开始的SVG实现没有这样的现象呢？

实际上，SVG的实现底部的连接也是不完美的，但是由于我们上面的这个圆环渐变颜色比较少，同时颜色比较接近，因此不仔细看，根本看不出来，连接有问题的。

但是这里的12色渐变就不一样了，渐变颜色跨度明显区分大，因此很明显看出来接缝有问题，那有没有什么办法可以解决这个问题呢？

有！我想的办法是，就是在下面接缝不完美的地方打一个小补丁，就像是我国桥梁建造接缝处重新浇筑一样，于是HTML变成下面这样，增加一个 `.circle-bottom`：

```

<div class="circle">
  <div class="circle-left"></div>
  <div class="circle-right"></div>
  <div class="circle-bottom"></div>
</div>

```

对应的CSS如下：

```

.circle-bottom {
  position: absolute;
  height: 40px; width: 30px;
  bottom: 3px; left: 0; right: 0;
  margin: auto;
  /* 下面两个声明是补丁关键 */
  background: linear-gradient(to right, #DB3FA3, #C443A3);
  filter: blur(5px);
}

```

也就是接缝处搞一个小小的水平渐变，同时适当高斯模糊，淡化边界。然后肉眼所见的效果就变成下面这样，几乎看不出任何异常：



接下来就简单了，同样地，JS设置圆环 `clip-path` 动态剪裁，实现我们想要的倒计时效果。

眼见为实，您可以狠狠的点击这里：[linear-gradient线性渐变实现的多彩圆环demo](#)

优点和不足

此方法的优点在于兼容性足够（仅IE不支持），在移动端可以无障碍使用。

不足在于这个圆环的渐变并不是径向发散的，而是自上而下的，和真正意义上的圆环渐变还是有些差异的。

那有没有什么方法，既保证兼容性，同时渐变又是真正意义上的环形渐变呢？

四、借助CSS clip剪裁、transform旋转和模糊滤镜实现12色渐变圆环

此方法需要一定的理解成本，请容我慢慢道来。

1. 如何实现一个30°的扇形？

假设有如下HTML：

```
<div class="sector"></div>
```

则下面的CSS可生成一个30°的扇形：

```
.sector {  
  width: 300px; height: 300px;  
  position: absolute;  
  clip: rect(0 300px 300px 150px);  
  overflow: hidden;  
}  
.sector::after {  
  content: '';  
  width: 100%; height: 100%;  
  background: currentColor;  
  position: absolute;  
  clip: rect(0 150px 300px 0);  
  transform: rotate(30deg);  
  border-radius: 50%;  
}
```

实时效果如下：

原理如下：

1. 外面容器原本是个正方形，但通过 `clip` 剪裁只显示右半部分；
2. 子元素同样尺寸，但通过 `clip` 剪裁只显示左半部分。然后通过适当旋转露出我们想要的扇形区域；

原理示意图如下：



2. 如何使用12个30°扇形铺成一个圆

很简单，我们只要创新12个类似下面的HTML：

```
<div class="sector"></div>
```

然后每个扇形元素比前面一个多旋转30度就可以了，然后每一个扇形设置一个渐变颜色，CSS类似下面：

```
.sector:nth-of-type(1) {  
    transform: rotate(0);  
    color: #9ED110;  
}  
/* 2-11 略 */  
.sector:nth-of-type(12) {  
    transform: rotate(330deg);  
    color: #EDE604;  
}
```

我们就可以得到类似下图的效果：



3. 如何模糊扇形色块的边界？

这个我们可以借助CSS3的 `filter` 滤镜中的高斯模糊实现，假设12个扇形元素全部都在一个类名为 `.sector-group` 的元素中，则：

```
.sector-group {  
    filter: blur(15px);  
}
```

效果如下图：



然而高斯模糊的边缘似乎有些问题，不是很清晰，怎么办？

4. 模糊的圆环边缘如何处理？

其实很简单，外部容器尺寸限制，里面模糊元素适当 `transform` 比例放大就可以了，例如：

```
.circle {  
    width: 300px; height: 300px;  
    border-radius: 50%;  
    position: relative;  
    overflow: hidden;  
}  
.sector-group {  
    filter: blur(15px);  
    transform: scale(1.2);  
}
```

此时效果为：



5. 使用mask遮罩是显示边缘圆环

CSS代码示意如下（实际开发 `webkit` 前缀不可省略）

```
.circle {
  width: 300px; height: 300px;
  border-radius: 50%;
  position: relative;
  overflow: hidden;
  mask: radial-gradient(transparent 110px, #000 110px);
}
.sector-group {
  filter: blur(15px);
  transform: scale(1.2);
}
```

最终，我们的12色彩虹渐变圆环效果即达成，如下Firefox浏览器下载图：



接下来就简单了，同样地，JS设置圆环 `clip-path` 动态剪裁，实现我们想要的倒计时效果。

眼见为实，您可以狠狠的点击这里：[借助CSS3 filter模糊和clip实现的多彩圆环demo](#)

优点和不足

优点在于真正意义上的锥形渐变，同时12种渐变颜色头首尾如贪吃蛇一样，完全融为了一体，更有视觉优势。

不足之处在于HTML和CSS代码稍微多了点。

四、结束语

严格来讲，本文一共介绍了四种彩色圆环渐变效果的实现，可以说各有优缺点，大家可以根据自己项目情况，选择合适的方法实现。

然而一个人的积累总是有限，而创意总是无限的，因此一定还有其他更好更妙更简单的实现，欢迎分享欢迎指教！

当然彩色渐变圆环效果不一定是用在倒计时上，例如下图这种loading效果（下图是图片实现的）实际上也可以完全通过CSS实现了，比较推荐的做法是使用左右半区的线性渐变加遮罩（在retina屏幕下效果真的赞），具体代码就不展示了，大家可以自己作为课后作业实践实践。



以上就是本文的全部内容，内容略多，感谢一直阅读到此处！

《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

- 小tip: 使用CSS将图片转换成模糊(毛玻璃)效果
- 小tips: 纯CSS实现蜡烛、火焰以及熄灭后烟雾效果
- CSS3+js实现多彩炫酷旋转圆环时钟效果
- 基于clip-path的任意元素的碎片拼接动效
- 10个demo示例学会CSS3 radial-gradient径向渐变
- 纯CSS实现任意格式图标变色的研究
- 客栈说书: CSS遮罩CSS3 mask/masks详细介绍
- 团购类网站倒计时的js实现
- -webkit-box-reflect属性简介及元素镜像倒影实现
- canvas getImageData与任意字符图形点、线动效实现
- 博闻强识: 了解CSS中的@ AT规则

分享到:         0

标签: blur, clip, clip-path, conic-gradient, css3, filter, linear-gradient, mask, radial-gradient, rotate, 倒计时

发表评论 (目前11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.

bill说道:

2018年09月15日 17:36

鑫哥, 请问锯齿边怎么处理呢?

回复



2.

懒人说道:

2017年11月15日 17:29

我想切个图。。。

回复



12345说道:

2017年11月17日 12:31

shia



回复

3. 美丝学院说道:
2017年11月15日 08:18

css真能玩出花来啊

回复



4. cshenger说道:
2017年11月10日 16:27
最后这种方法才是最浪的

回复



码农说道:
2017年11月28日 10:47

但是效果最完美

回复



5. 码农说道:
2017年11月10日 15:46
这渐变背景色可以
<http://lab.iamvdo.me/houdini/conic-gradient>

回复



6. 静默说道:
2017年11月10日 10:52
认认真看完第一个方法点进去一看什么都没, 结果下面才说不兼容。。。

回复



7. xshuai说道:
2017年11月9日 12:54

哥, 你拿了多少股票?

回复



8. 我是小况说道:
2017年11月9日 11:30
鑫哥, 我怎么感觉这里的运用和css揭秘中的很多方法有异曲同共之妙!!!!

回复



9. 郭子说道:
2017年11月9日 08:59
恭喜阅文上市~
壕老师的博客以后还能一如既往的更新吗? ?

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 ⁽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 scrollIntoView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

- 小tip: 使用CSS将图片转换成模糊(毛玻璃)效果
- 小tips: 纯CSS实现蜡烛、火焰以及熄灭后烟雾效果
- CSS3+js实现多彩炫酷旋转圆环时钟效果
- 基于clip-path的任意元素的碎片拼接动效
- 10个demo示例学会CSS3 radial-gradient径向渐变
- 纯CSS实现任意格式图标变色的研究
- 客栈说书: CSS遮罩CSS3 mask/masks详细介绍
- 团购类网站倒计时的js实现
- -webkit-box-reflect属性简介及元素镜像倒影实现
- canvas getImageData与任意字符图形点、线动效实现
- 博闻强识: 了解CSS中的@ AT规则

