CSS 函数的 8 个妙用

原创 2016-09-15 伯乐专栏/邓玉莹 前端大全

（点击上方公众号，可快速关注）

> 原文：Anselm Urban
> 译文：伯乐在线专栏作者 - 古鲁伊
> 链接：http://web.jobbole.com/87821/

CSS 比许多 web 程序员认为的更加强大。这种样式表语言正变得越来越强大，给浏览器带来了原本要用 JavaScript 实现的功能。本文将介绍无需 JavaScript 的 CSS 函数的 8 个妙用。

1. 纯 CSS 的 tooltips

很多网站仍在用 JavaScript 创建 tooltips，但实际上，用 CSS 实现更加简单。最简单的方法是在 HTML 代码的 data 属性中提供 tooltip 文本，比如：data-tooltip="..."。这样就可以在 CSS 中添加以下代码来在 attr() 函数中显示 tooltip 文本了：

```
.tooltip::after {
  content: attr(data-tooltip);
}
```

很好懂，对吧？当然要想设计 tooltips 还需要更多的代码，但是不必担心，有一款很棒的纯 CSS 库就是为此而生的，叫做 Hint.css 。

2. 使用自定义data 属性和 attr() 函数

我们已经在 tooltips 中用过 attr() 了，但是还有一些情况也可以用 attr()。结合 data 属性，可以通过 title 和 description 仅用一行 HTML 代码创建缩略图：

```
<a class="caption" href="#" data-title="Vulture" data-description="...">
  <img src="img.jpg" alt="Illustration"/>
</a>
```

接下来就可以用 attr() 函数来显示 title 和 description 了：

```
.caption::after {
  content: attr(data-title);
    ...
}
```

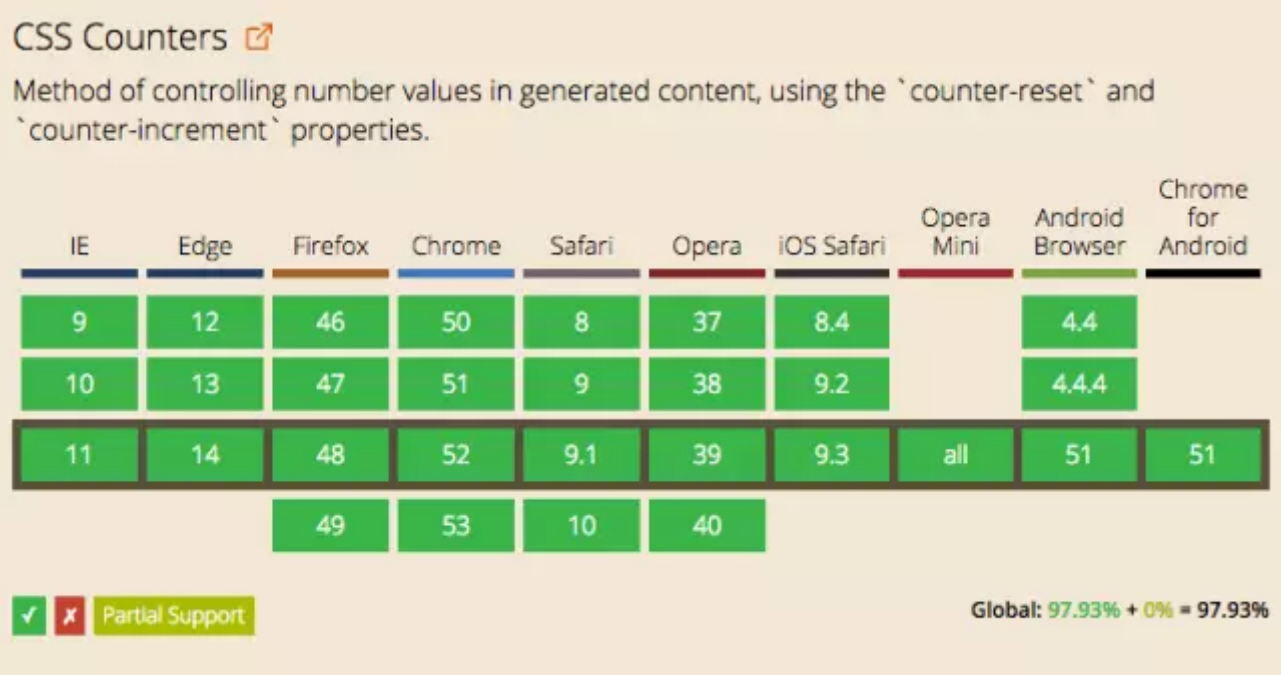
这里提供一个 hover 上去有字幕动画效果的缩略图例子：

http://codepen.io/SitePoint/pen/akAmPw

说明：CSS 生成的 content 是 不易获取 的。在此方面，这篇文章的 关于获取 CSS 生成 content 的内容（http://tink.uk/accessibility-support-for-css-generated-content/） 部分可以借鉴。

3. CSS 计数器

CSS 计数器可以做出神奇的效果。计数器不是最有名的特点，多数人可能认为它的支持性不够好，但实际上，所有浏览器都支持 CSS 计数器：



计数器用于分页或是在 gallery 下方展示项目数量都很棒，但是不应该用在有序列表（<ol>）上。也可以用 CSS 计数器计算已选项的数量，代码量会令你惊叹（并且没有 JavaScript）：

http://codepen.io/lonekorean/pen/wKbzv

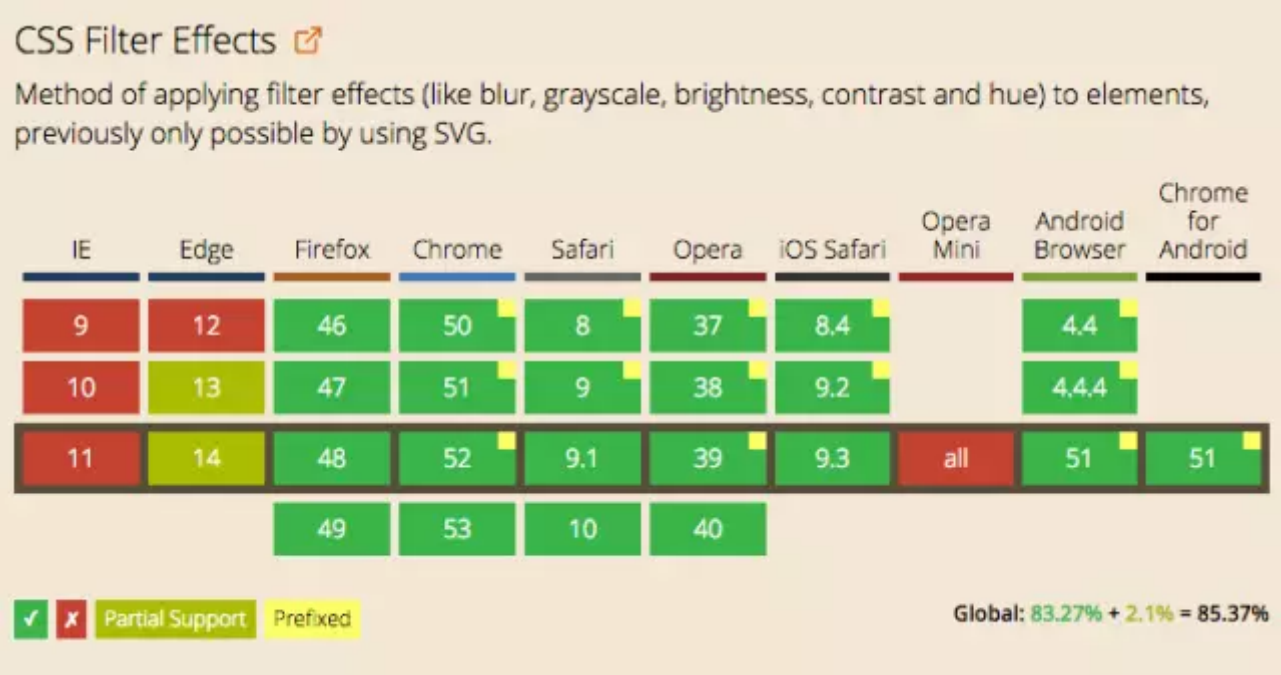
在可拖放排序的列表中用 CSS 计数器动态改变数量也很赞：

http://codepen.io/SitePoint/pen/bdeOKJ

像上个例子一样，记住——CSS 生成的 content 是不易获取的。

4. CSS filters实现毛玻璃效果

苹果的 iOS 7 给我们带来了“毛玻璃”效果——看起来像磨砂玻璃窗的、半透明、模糊的元素。在苹果的推行下，在越来越多的地方可以见到这种效果了。再现这种效果有点复杂——在有 CSS filters 之前，只能用 模糊的半透明图片 来实现毛玻璃效果。目前几乎所有的主流浏览器都支持 CSS filters，再现毛玻璃效果就简单多了 。



目前 backdrop filters 和 filter() 函数 只有 Safari 支持，但是将来我们就可以用它们实现相似的效果了。

5. HTML Elements做背景图

通常都是指定 JPEG 或 PNG 文件充当背景图或渐变。但是你知道 element() 函数可以用 做背景图吗？目前只有 Firefox 支持 element() 函数：

Element() 函数拥有无限可能，这里有个 MDN 上的 例子 。

6. calc() 实现智能栅格

Fluid grids（流体网格划分）很棒，但是有几个严重的问题。例如，无法实现顶部和底部的空隙和左右的空隙大小相同。此外，根据所使用的栅格系统不同，标记有些混乱。即使是 flexbox 也不是做好的解决方法。但是有了可在 CSS 中作为 value 使用的 calc() 函数，栅格会变得更棒。在 SitePoint 的此篇教程 中，George Martsoukos 列举了几个实例，比如间隔完美的网格画廊。通过 CSS 预编译器，比如 Sass，利用 calc() 实现一套栅格系统 非常简单，并且易于维护。calc() 的浏览器支持性近乎完美，极力推荐使用。

7. CSS calc() 对齐 position:fixed 元素

calc() 函数的另一个使用场景是对齐 fixed 定位的元素。例如，有一个左右均有空隙的 content wrapper，如果想要对 wrapper 内的一个 fixed 元素精准对齐——要算出给“right”或是“left”属性赋值多少是很困难的。用 calc() 可以结合 relative 和 absolute 的值来精准对齐元素：

```
.wrapper {
  max-width: 1060px;
  margin: 0 auto;
}
.floating-bubble {
  position: fixed;
  right: calc(50% - 530px); /* 50% - half your wrapper width */
}
```

这里提供一个例子：

http://codepen.io/SitePoint/pen/NAVRZQ

8. cubic-bezier() 动画

要想网站或 app 的 UI 更加引人注目的话，可以使用动画。但是标准的 easing 选项非常有限，比如“linear”或是“ease-in-out”。像是弹跳动画更是标准选项无法实现的。使用 cubic-bezier() 函数，就可以让元素按你想要的方式去动画。

使用 cubic-bezier() 有两种方式—— 理解背后的数学知识 然后自己创建，或者使用 cubic-bezier 生成器 。

说实话，我喜欢后一种方式。

结语

聪明使用 CSS 函数不仅可以解决像是建立更合理的栅格系统等难题，它也给了你更多机会去创造。随着浏览器的支持性越来越好，你真的应该重新审视一下你的 CSS 代码，考虑用 calc() 等函数进行优化了。

译者简介 （点击 → 加入专栏作者）

古鲁伊：立志做一名有格调的程序媛

打赏支持作者写出更多好文章，谢谢！