

基于 CSS3 的下一代 Web 应用开发, 第 1 部分: 发展历史及新特性

[秦 剑](#), 软件工程师, IBM

[陈 序明](#), 资深软件工程师, IBM

简介: CSS2.1发布至今已有12年的历史。在计算机产业飞速发展的12年间, 软硬件以及互联网已有日新月异的变化。CSS2.1已经无法再满足快速提供高性能、高用户体验的Web应用的需求。CSS3的产生大大简化了编程模型, 它不是仅对已有功能的扩展和延伸, 而更多的是对Web UI设计理念的和方法的革新。相信未来CSS3配合HTML5标准, 将极大的引起一场Web应用的变革, 甚至是整个Internet产业的变革。本文是《基于CSS3的下一代Web应用开发》系列文章的第一篇, 将介绍CSS的发展历史, 个版本的功能特点差异和CSS3引入的新特性, 并给出一些应用实例。

本文的标签: [ajax](#), [css](#), [dhtml \(dynamic html\)](#), [javascript](#), [web](#), [用户界面](#)

[标记本文!](#)

发布日期: 2011 年 1 月 04 日

级别: 中级

CSS 的发展历史

1982 年 Tim Berners-Lee 为使世界各地的物理学家能够方便的进行合作研究以及信息共享, 创造了 HTML (HyperText Markup Language) 超文本置标语言。1990 年他发明了世界上第一个浏览器 WorldWideWeb。在 1991 年 3 月, 他把这发明介绍给了给他在 CERN 工作的朋友, 当时网页浏览器被其世界各地的成员用来浏览 CERN 庞大的电话簿。1993 年 NCSA 推出了 Mosaic 浏览器并迅速爆红, 成为世界上第一个广泛应用的浏览器, 推动着互联网迅猛发展。在随后的 5 年里 Netscape 和 MicroSoft 两个软件巨头掀起了一场互联网浏览器大战。这场战争最后以 MicroSoft 的 Internet Explorer 全胜告终, 但它极大的推动了互联网的发展, 把网络带到了千千万万一般 PC 用户面前。从 1993 年互联网工程工作小组 (IETF) 工作草案发布, 到 1999 年 W3C HTML4.01 标准的发布, HTML 共经历过 5 个版本。如今的 HTML 不仅成为 Web 上最主要的文档格式, 而且在个人及商业应用中都发挥着它的作用。尽管他还有不足, 但是它将成为应用最广泛的格式化文档。

就在 HTML 迅猛发展的 90 年代, CSS (Cascading Style Sheet) 也以各种形式应运而生。不同的浏览器结合各自 HTML 语法结构实现了很多不同样式语言支持, 读者可以使用这些样式语言来调节网页的显示方式。最初的 HTML 版本只含有很少的显示属性, 读者可以自己来决定网页应该怎样被显示。但随着 HTML 的成长, 为了满足设计师的要求, HTML 增加了很多显示功能, 随着这些功能的增加, 外来定义样式的语言越来越没有意义了。

1994 年 Håkon Wium Lie 为 HTML 样式提出了 CSS 的最初建议。Bert Bos 当时正在设计一个叫做 Argo 的浏览器, 他们决定一起合作设计 CSS, 于是形成了 CSS 的最初版本。1994 年 Håkon Wium Lie 在芝加哥的一次会议上第一次正式提出了 CSS 的建议, 1995 年他与 Bert Bos 一起再次展示这个建议。当时 W3C 刚刚建立, W3C 对 CSS 的发展很感兴趣, 它为此专门组织了一次讨论会。1996 年 12 月 W3C 终于推出了 CSS 规范的第一版本。这一规范立即引起了各方的积极响应, 随即 MicroSoft 公司和 Netscape 公司纷纷表示自己的浏览器能够支持 CSS1.0, 从此 CSS 技术的发展几乎一马平川。1998 年 W3C 发布了 CSS 的第二个版本, 这也是至今流行最广目前主流浏览器都采用的标准。然而 CSS2.1 的发布至今已有 12 年历史。然而在这 12 年中, 计算机软件、硬件、互联网已经有日新月异的发展。用户对视觉用户体验提出了更高的要求, 开发人员对如何快速提供高性能、高用户体验的 Web 应用也提出更高的要求。2005 年 12 月 W3C 开始 CSS3 标准的制定, 到目前为止该标准还没有最终定稿。

[回页首](#)

CSS1.0 版本

在 1997 年 W3C 颁布 HTML4 标准的同时也发布了有关样式表的第一个标准 CSS1.0。CSS1.0 较为全面的规定了文档的显示样式，其大致可分为选择器、样式属性、伪类 / 对象几个大的部分。选择器大致分为派生选择器、ID 选择器和类选择器几种，用来定义你希望应用样式的 HTML 元素或者标签。样式属性主要包括 Font 字体、Text 文本、Background 背景、Position 定位、Dimensions 尺寸、Layout 布局、Margins 外边框、Border 边框、Padding 内边框、List 列表、Table 表格、Scrollbar 滚动条等等，用来定义你希望改变的样式。每个属性都有一个值，属性和值被冒号分开并由花括号包围，这样就组成了一个完整的样式声明。通常情况下样式的应用都需要指定你希望应用样式的 HTML 节点，然而很多时候我们希望动态的指定节点，也就是说节点的样式需要动态变化，这就需要伪类来完成。在 CSS1.0 中主要定义了针对锚对象 a 的 link、hover、active、visited 和针对节点的 first-letter、first-child、first-line 几个伪类属性。

CSS1.0 是对 HTML 语法的一次重大革新。以前的 HTML 版本中，各种样式功能的实现是通过标记元素实现的，比如要在一段文字中把一部分文字变成红色，HTML3.2 中应该是这样的：`<p>HelloWorld</p>`。而在 CSS 把这些标记属性化了，利用 CSS 上面的例子变成：`<p style="color:red">HelloWorld</p>`。CSS 的引入使得文档的可读性被大大加强、文档的结构设计也更加灵活、HTML 的编写者可以自己决定文档的显示结构和样式、同时文档的结构也相应的被简化了。之前由于各个 [浏览器](#) 厂商为了在竞争中取得更多的市场份额，在浏览器实现中都加入了很多各自相互不兼容的样式标记，这使得同一个 HTML 文档很难在多个浏览器中兼容。CSS 的出现从一定程度上解决了 HTML 跨浏览器通用性的问题。另外，随着后来 DHTML 的出现，开发人员可以使用一些 script 语言来动态修改 HTML 节点的样式。这极大的增强了 CSS 的表现力，凸显出了 CSS 的真正的魅力。

[回页首](#)

CSS2.0/2.1 版本

1998 年 W3C 发布了 CSS 的第二个版本，目前市面上主流的浏览器都已经支持了 CSS2.1 中 99.9% 的功能。CSS2 规范是基于 CSS1 设计的，其包含了 CSS1 所有的功能，并扩充和改进了很多更加强大的属性。

选择器

CSS2 提供了更多强大的选择器来定位 HTML 节点或者标签。"*"用来匹配任何标签，例如 `* {color:red}`。大于号">"用于指定父子节点关系，例如 `AAA > CCC > DDD {color:red}` 匹配父节点分别为 CCC 和 AAA 的节点 DDD 节点。空格用来匹配在某一层有某个父节点的节点，例如 `AAA CCC DDD {color:blue}` 匹配包含父节点 CCC 的父节点 AAA 的节点 DDD。加号"+"用于表示在同一个级别节点之间的关系，例如 `AAA + BBB {color:red}` 表示，有一个兄弟节点 AAA 的节点 BBB。另外，还可以选择包含特殊属性值的节点，例如：`BBB[text="xyz"] {color:blue}` 表示包含 text 属性值为"xyz"的 BBB 标签。

位置模型

最然 CSS1 中已经定义了一些关于位置 Positioning 的属性，CSS2 进一步增强了这部分功能，增加了 relative、absolute、fixed 几个值。relative 值：元素框相对父节点偏移某个距离，元素仍保持其未定位前的形状，它原本所占的空间仍保留。absolute 值：元素框从文档流完全删除，并相对于其包含块定位。包含块可能是文档中的另一个元素或者是初始包含块。元素原先在正常文档流中所占的空间会关闭，就好像元素原来不存在一样。元素定位后生成一个块级框，而不论原来它在正常流中生成何种类型的框。fixed 值：元素框的表现类似于 absolute 值，不过其包含块是视窗本身。在 CSS2 中还有另外一个非常有用的新属性 z-index，用来指定相互重叠的元素的叠放顺序，数值较大的元素会叠放在数值较小的元素前面。新的定位模型提供的这些属性使我们更加容易的建立列式及复杂布局，将布局的一部分与另一部分重叠，还可以完成多年来通常需要使用多个表格才能完成的任务。

布局、表格样式

display 属性用来规定元素应该生成的框的类型，但在 CSS1 中只有少数几个属性。CSS2 对该属性进行了

许多扩充, 你可以用该属性指定元素是否会显示以及如何显示, 也可以使用该属性配合位置和浮动进行页面的布局, 另外还可以将一个非表格的结构化文档显示为一个表格样式。这是一个非常有用而且重要的功能, 可惜例如表格的诸多样式在 IE 中都不被支持。

媒体类型

很多时候我们设计 CSS 的时候希望为特定的媒体设定特定样式, 例如用于屏幕和打印的两个媒介, 在屏幕显示时字体应比打印时稍大一些, 屏幕上无衬线字体要容易阅读一些, 而在纸面上有衬线的字体要容易阅读一些。因此我们有必要对不用媒体声明不同的样式。CSS2 引入了媒介类型, 用于对不同的媒介类型定义不同的样式。可用的媒介类型有: **aural** 用于语音和音频合成器, **braille** 用于盲人用点字法触觉回馈设备, **embossed** 用于分页的盲人用点字法打印, **handheld** 用于小的手持的设备, **print** 用于打印机, **projection** 用于方案展示如幻灯片, **screen** 用于电脑显示器。

伪类

在 CSS2 中不但增加了 `:focus`、`:first-child`、`:lang` 等几个新的伪类, 同时还扩充了伪类的使用范围。使得伪类不但可以和原来一样应用于 a 锚标签, 还可以应用到一个类和标签上, 例如 `:link:hover`、`myClass:hover`。

光标样式

CSS2 的另一大亮点就是增加了 `cursor` 属性, 这一属性指定了指定设备应该显示怎样的光标类型。光标的可选类型有: `auto` 基于上下文自动决定显示光标、`crosshair` 十字线、`default` 基于平台的缺省光标、`pointer`、指针光标、`move` 表示移动、`e-resize`、`ne-resize`、`nw-resize`、`n-resize`、`se-resize`、`sw-resize`、`s-resize`、`w-resize` 表示正在移动某个边、`text` 表示可以选择文本、`wait` 表示程序正忙、`help` 表示帮助。

CSS2 规范中还有一些属性和增强, 这里我们就不一一列出了。这些新功能极大的增强了 CSS 和 HTML 的表现能力, 使得 Web 应用拥有和本地应用程序一样的表现力, 甚至是更好的用户体验。虽然 CSS 规范有如此之多的优点, 但其推广和采用却一直很艰难。首先 CSS1 于 1996 年发布同年 Internet Explorer3 宣称部分支持 CSS1, 然而直到 3 年后大部分的浏览器才全部支持 CSS1。之后浏览器大多都宣称部分支持 CSS2, 然而直到 2009 年 3 月 Internet Explorer8 才全部支持 CSS2.1。其次浏览器的支持也是五花八门, 各种各样的 Bug 和诡异的行为, 都使得设计开发一个跨平台且表现一致的 web 应用变得非常困难。浏览器支持的现状和各种问题使得 W3C 不得不在 2007 年最终修订 CSS2 为 CSS2.1, 删除和修改一些属性和行为, 使其更接近当前浏览器支持和实现。

CSS2 可以说是 CSS 发展历史上的一个里程碑, 它对 Web 应用的发展有不可磨灭的作用。然而使得全部浏览器支持该标准却花费了数十年的时间。在这十几年间, 计算机软硬件和网络技术已经有了长足的发展, 用户对视觉体验提出了更高的要求, 开发人员对如何快速提供高性能、高用户体验的 web 应用也提出更高的要求。面对更高用户体验、更强交互性需求时, CSS2 显得有些力不从心。

CSS3 的新特性

早在 2001 年 5 月, W3C 就着手开始准备开发 CSS 第三版规范。CSS3 规范一个新的特点是规范被分为若干个相互独立的模块。一方面分成若干较小的模块较利于规范及时更新和发布, 及时调整模块的内容。另外一方面, 由于受支持设备和浏览器厂商的限制, 设备或者厂商可以有选择的支持一部分模块, 支持 CSS3 的一个子集。这样将有利于 CSS3 的推广。相信这对以前 CSS 支持混乱的局面将会有所改观。

下图是 W3C 关于 CSS3 规范新模块的一个开发进度表, 绿色背景表示将成为 CSS3 规范的核心模块。HTML Basic、CSS3 和 SVG 三列表示各个模块和他们之间的相关度, 没有涉及到的模块未来可能会被从规范中移除。黄色背景表示该模块未来很可能被从规范中移除。

图 1.CSS3 各模块开发进度表

Module	Editor(s)	HTML Basic	CSS3	SVG	WD1	WD2	LC	Ends	CR	PR	REC
•Syntax / grammar / etc.		X	X								
•Selectors	Tantek Çelik, Daniel Glazman, Ian Hickson	X	X		5 Oct 2000		26 Jan 2001	1 Mar 2001			
•Values, Units	Håkon Lie, Chris Lilley	X	X								
Value assignment / cascade / inheritance	Håkon Lie	X	X								
Box model / vertical	Bert Bos	X	X								
Positioning	Bert Bos		X								
•Color / gamma / color profiles	Tantek Çelik, Chris Lilley	X	X		22 Jun 1999	5 Mar 2001					
Background	Tim Boland	X	X								
Line box model	Eric Meyer	X	X								
•Text / bidi / vertical alignment	Chris Lilley, Michel Suignard	X	X		•17 May 2001						
•Ruby	Michel Suignard		X		16 Feb 2001						
Font properties	Chris Lilley, Michel Suignard	X	X								
WebFonts	Chris Lilley		X								
Generated content / markers	Håkon Lie		X								
Replaced content	Eric Meyer		X								
•Paged media	Steve Zilles		X		28 Sep 1999						
•User interface	Tantek Çelik		X		16 Sep 1999	16 Feb 2000					
Tables	Bert Bos	X	X								
•Columns	Håkon Lie		X		22 Jun 1999	18 Jan 2001					
•SVG	Chris Lilley		X	X							
Math	Angel Diaz										
•BECSS	Michel Suignard				4 Aug 1999						
ACSS											
•Media queries	Håkon W. Lie		X		•4 Apr 2001	•17 May 2001					
SML	Debbie Newman										
Test Suite	Eric Meyer		X								

CSS3 规范的全面推广和支持看起来还遥遥无期，但是目前主流浏览器都已迫不及待的开始支持 CSS3 部分特性了。虽然这部分特性还相对较少，但是这些以前很难使用 Javascript 和图片实现的效果依然令程序要兴奋不已。下面就简单介绍一下主流浏览器实现的一些 CSS3 新特性。

属性选择器

在 CSS3 中新添加了三个属性选择器 `[att^="value"]` `[att$="value"]` `[att*="value"]` 用来匹配属性中间包含某些特定的值。例如：

```
a[title$="IBM BTT"] {
  //add your property here
}
```

表示有 title 属性并以字符"IBM BTT"结尾的锚节点。

CSS3 中唯一新引入的连字符是通用的兄弟选择器（同级）。它针对一个元素的有同一个父级节点的所有兄弟级别元素。比如，给某个特定的 div 的同级的图片添加一个灰色的边框（div 和图片应该有同一个父级节点），在样式表中定义下面的样式就足够了：

```
div~img {
  //add your property here
}
```

目前这几个选择器除了 IE6 外的其他主流浏览器都支持。

RGBA 透明度

RGBA 让你可以不仅仅设定色彩，还能设定元素的透明度。例如：

```
background:rgba(254, 255, 200, 0.75);
```

图 2.RGBA 透明度演示



当设定一个 **RGBA** 色彩的时候, 参数依次设定为红、蓝、和绿色的颜色值, 可以是 0-255 或百分数。最后一个参数为透明值应该在 0 到 1 之间, 例如 0.5 代表 50% 的透明度。**RGBA** 和 **Opacity** 之间的不同点是前者只会应用到指定的元素上, 而后者会影响我们指定的元素及其子元素。

目前支持 **RBGA** 颜色的浏览器有 **WebKit** 核心系列浏览器、**Firefox 3+** 和 **Opera 9.5+**, **IE** 全系列浏览器都不支持。

多栏布局

这是新的 **CSS3** 选择器可以让你不用使用多个 **div** 标签就能实现多栏布局。浏览器解释这个属性并生成多栏, 让文本实现一个仿报纸的多栏结构。[tweetCC](#) 在其首页上将介绍文字显示为四栏, 这四栏并非浮动的 **div** 而是使用下面的 **CSS3** 多栏布局:

```
-moz-column-count : 4;
-moz-column-gap : 20px;
```

我们可以通过这个选择器定义三件事情: 栏数 (**column-count**)、栏宽 (**column-width**, 例子中没有用到) 和各栏之间的空白 / 间距 (**column-gap**)。如果 **column-count** 未设定, 浏览器会在允许的宽度内容纳尽可能多的栏目。为了在各栏之间添加一个数值的分隔, 我们可以使用 **column-rule** 属性, 其功能和 **border** 属性类似:

```
column-rule: 1px solid #000000;
```

图 3. 多栏布局演示



目前多栏布局目前被 **Webkit** 核心系列浏览器和 **Firefox 2+** 浏览器支持。

多背景图

CSS3 允许你使用多个属性比如 **background-image**、**background-repeat**、**background-size**、**background-position**、**background-origin** 和 **background-clip** 等在一个元素上添加多层背景图片。在一个元素上添加多背景的最简单的方法是使用简写代码, 你可以指定上面的所有属性到一条声明中, 只是最常用的还是 **image**, **position** 和 **repeat**:

```
background: url(example.jpg) top left no-repeat,
url(example2.jpg) bottom left no-repeat,
url(example3.jpg) center center repeat-y;
```

图 4. 多图片背景演示



目前支持形变的浏览器有 Webkit 系列浏览器、FireFox3.5+、Opear10.5+。

Word Wrap

`word-wrap` 属性用来防止太长的字符串溢出的。可以用两个属性值 `normal` 和 `break-word`。`normal` 值 (默认的) 只在允许的断点截断文字, 如连字符。如果使用了 `break-word`, 文字可以在任何需要的地方截断以匹配分配的空间并防止溢出。

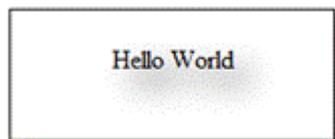
目前该属性被 IE8、Webkit 核心浏览器、Firefox3.5+、Opear10+ 支持。

块阴影与圆角阴影

尽管在 CSS2 中就已经存在, `box-shadow`、`text-shadow` 是两个未被广泛应用的 CSS 属性。它们将在 CSS3 中被广泛采用。这两个属性给设计师一个新的跨浏览器的工具来为设计添加一个维度以使文字和边框更加醒目、具有立体感。从此你不再需要 Photoshop 去处理很多复杂零碎的图片了。

```
box-shadow: 10px 10px 25px #ccc;
text-shadow: 10px 10px 25px #ccc;
```

图 5. 文字和盒阴影效果演示



前两个属性设置阴影的 X/Y 位移, 这里分别是 10px, 第 3 个属性定义阴影的虚化程度, 最后一个设置阴影的颜色。

目前这个属性被 webkit 核心浏览器、FireFox3.1+、Opera9.5+ 支持。

圆角

CSS3 新功能中最常用的一项就是圆角效果, `Border-radius` 无需背景图片就能给 HTML 元素添加圆角。不同于添加 Javascript 或多于的 HTML 标签, 仅仅需要添加一些 CSS 属性并从好的方面考虑。这个方案是清晰的和比较有效的, 而且可以让你免于花费几个小时来寻找精巧的浏览器方案和基于 Javascript 圆角。

```
border-radius: 6px 6px 6px 6px;
```

图 6. 圆角效果演示



目前 IE9、webkit 核心浏览器、FireFox3+ 都支持该属性。

边框图片

`border-image` 属性允许你在元素的边框上设定图片, 让你从通常的 `solid`、`dotted` 和其它边框样式中解放出来。该属性给设计师一个更好的工具, 用它可以方便的定义设计元素的边框样式, 比 `background-image` 属性 (对高级设计来说) 或枯燥的默认边框样式更好用。我们也可以明确的定义一个边框可以被如何缩放或平铺。

```
border: 5px solid #cccccc;
border-image: url (/images/border-image.png) 5 repeat;
```

图 7. 边框图片效果演示



目前支持的浏览器有 Webkit 核心浏览器、FireFox3.1+。

形变

在过去在 web 中想要达到 `rotate` 旋转、`scale` 伸缩、`skew` 倾斜这些效果, 我们通常需要依赖于图片或者 Flash。在 CSS3 通过 `transform` 实现一些形变的效果将变得非常简单。

```
transform: rotate(2deg) scale(0.90, 0.85) translate(0px, 0px) skew(0deg, 0deg);
transform-origin: 15% 25%;
```

图 8. 形变效果演示



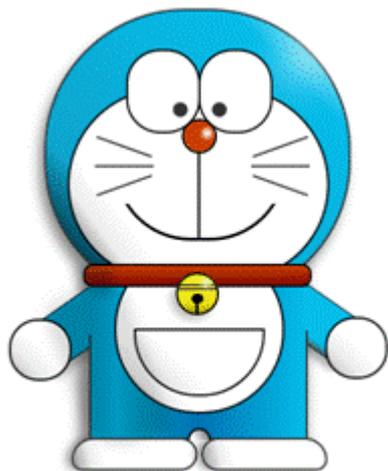
在变形属性中, 我们可以通过 `transform-origin` 指定变形的起始点 (比如对象的中心或者右上角等)。还可以指定变形的类型 (也称为变形函数), 并在函数后面的括号中指定的变形的程度。比如, `translate(10px, 20px)` 将会把相应的元素从原位置右移 10 个像素, 下移 20 个像素。其他支持的变形函数还包括: 旋转 (`rotate`)、伸缩 (`scale`)、倾斜 (`skew`)。

目前支持形变的浏览器有 Webkit 系列浏览器、FireFox3.5+、Opera10.5+, IE 全系列不支持。

[回页首](#)

小结

图 9.CSS 和 HTML 实现多啦 A 梦



使用 CSS 没有做不到的效果，只有你不敢想象的，前一阵我就惊讶的发现一个使用纯 CSS 和 HTML 实现的多啦 A 梦卡通效果。CSS3 为我们带来的惊喜不只是简单功能性增强，更多的是一种对 Web UI 设计理念和方法的变革。CSS3 结合 HTML5 使得 Web 应用可以提供和本地应用程序一样甚至更强大的功能，并且有更好的用户体验，同时也不需要额外安装软件，不必对软件版本升级兼容性等麻烦问题担忧。可以畅想，在未来的 PC 上我们已经不需要操作系统以及任何其他应用程序，开机只需要一个浏览器就可以解决所有问题。相信未来 CSS3 配合 HTML5 标准，将极大的引起一场 Web 应用的变革，甚至是整个 Internet 产业的变革。

参考资料

学习

- 参考 W3C CSS 规范 Level 1: [CSS 规范 Level1](#)。
- 参考 W3C CSS 规范 Level 2/2.1: [CSS 规范 Level2/2.1](#)。
- 参考 W3C CSS 规范 Level 3: [CSS 规范 Level3](#)。
- [developerWorks Web development 专区](#): 通过专门关于 Web 技术的文章和教程，扩展您在网站开发方面的技能。
- [developerWorks Ajax 资源中心](#): 这是有关 Ajax 编程模型信息的一站式中心，包括很多文档、教程、论坛、blog、wiki 和新闻。任何 Ajax 的新信息都能在这里找到。
- [developerWorks Web 2.0 资源中心](#)，这是有关 Web 2.0 相关信息的一站式中心，包括大量 Web 2.0 技术文章、教程、下载和相关技术资源。您还可以通过 [Web 2.0 新手入门](#) 栏目，迅速了解 Web 2.0 的相关概念。
- 查看 [HTML5 专题](#)，了解更多和 HTML5 相关的知识和动向。

讨论

- 加入 [My developerWorks 中文社区](#)

基于 CSS3 的下一代 Web 应用开发, 第 2 部分: 实现和应用

[秦剑](#), 软件工程师, IBM

简介: CSS2.1 发布至今已有 12 年的历史。在计算机产业飞速发展的 12 年间, 软硬件以及互联网已有日新月异的变化。CSS2.1 已经无法再满足快速提供高性能、高用户体验的 Web 应用的需求。CSS3 的产生大大简化了编程模型, 它不是仅对已有功能的扩展和延伸, 而更多的是对 Web UI 设计理念的方法和革新。相信未来 CSS3 配合 HTML5 标准, 将极大的引起一场 Web 应用的变革, 甚至是整个 Internet 产业的变革。

[查看本系列更多内容](#)

本文的标签: [dhtml \(dynamic html\)](#), [xml](#), [技巧](#), [用户界面](#)

[标记本文!](#)

发布日期: 2011 年 5 月 03 日

级别: 中级

本文《[基于 CSS3 的下一代 Web 应用开发](#)》系列文章的第 2 部分, 第 1 部分我们将介绍了 CSS 的发展历史, 各个版本的功能特点差异和 CSS3 引入的新特性。第 2 部分, 我们针对这些 CSS3 新特性介绍其在不同浏览器中实现的差别, 并结合实例讲解如何利用这些特性进行开发。

前言

在该系列文章第 1 部分中我们介绍了 CSS 的发展历史以及各个版本的主要功能和差别。之后介绍了一些 CSS3 中引入的新特性和功能。这些新特性极大地增强了 Web 程序的表现能力, 同时简化了 Web UI 的编程模型。然而 CSS 一直存在的最大问题是众多浏览器对 CSS 标准支持不统一, 相同属性在不同浏览器中有不同的实现。因此了解和处理 CSS3 新特性在各个浏览器中的实现差别是十分重要的。在本文中, 我们将详细介绍这些 CSS3 新特性在不同浏览器中的实现支持情况, 并结合实例讲解如何利用这些特性进行开发一些绚丽的应用。

[回首页](#)

CSS3 新特性在浏览器中的不同实现

RGBA 颜色

RGBA 颜色是你可以使用 RGB 来控制兑现过的颜色并且可以调整对象的透明度, 无论是文本、背景还是边框均可使用该属性。该属性的语法在其支持的浏览器中相同, 如下:

清单 1. RGBA 颜色代码示例

```
background: rgba(254, 255, 200, 0.75);
```

例如上面代码所示, 前三个参数分别是 R、G、B 三原色, 范围是 0-255。第四个参数是背景透明度, 范围是 0-1, 如 0.5 代表透明度 50%。这个属性使我们在浏览器中也可以做到像 Win7 一样的半透明玻璃效果。

图 1.RGBA 颜色效果



另外 RGBA 和 Opacity 之间的不同点是前者只会应用到指定的元素上, 而后者会影响我们指定的元素及其子元素。目前支持 RGBA 颜色的浏览器有: Safari 4+、Chrome 1+、Firefox 3.0.5+、和 Opera 9.5+, IE 全系列浏览器都不支持该属性。

圆角

CSS3 新特性中除了 RGBA 颜色, 最常用的一项就是圆角效果。圆角属性无需使用背景图片就能给任意 HTML 元素添加圆角。使用 CSS3 圆角属性将大大减少你制作图片或者调试 JavaScript 的时间, 并且浏览器原生态的支持运行更快、更稳定、显示也更清晰。其在主要浏览器中的语法如下:

清单 2.CSS3 圆角代码示例

```
/*FireFox 语法 */
-moz-border-radius: 6px
                    6px
                    6px;
-moz-border-radius-topright: 6px;
-moz-border-radius-topleft: 6px;
-moz-border-radius-bottomright: 6px;
-moz-border-radius-bottomleft: 6px;

/*WebKit 核心浏览器语法 */
-webkit-border-radius: 6px
                       6px
                       6px;
-webkit-border-top-right-radius: 6px;
-webkit-border-top-left-radius: 6px;
-webkit-border-bottom-right-radius: 6px;
-webkit-border-bottom-left-radius: 6px;

/*CSS 标准语法 */
border-radius: 6px
               6px
               6px;
border-top-right-radius: 6px;
border-top-left-radius: 6px;
border-bottom-right-radius: 6px;
border-bottom-left-radius: 6px;
```

如上面代码所示, 四个角的效果可以通过一行代码 `border-radius: 6px 6px 6px 6px`来指定, 其中四个参数从左到右分别代表: 左上角、右上角、右下角、左下角。也可以类似 `border-top-right-radius: 6px;` 这样分别指定每一个角的效果。

图 2. 圆角效果



可以看出 Firefox 和 Webkit 核心的浏览器和 CSS 标准用法大同小异, 只是属性前缀有些差别。目前支持

圆角属性的浏览器有: Safari 3+、Chrome 1+、Firefox 1+, IE 全系列浏览器和 Opear 不支持该属性。

块阴影与文字阴影

块阴影和文字阴影是你可以不适用图片就能对 HTML 元素添加阴影, 增加显示的立体感增强设计的细节。块阴影使用 `box-shadow` 属性, 文字属性使用 `text-shadow` 属性, 其语法如下:

清单 3. 块阴影和文字阴影代码示例

```
/*FireFox 语法 */
-moz-box-shadow: 10px
                10px
                25px
                #CCC;
-moz-text-shadow: 10px
                 10px
                 25px
                 #CCC;

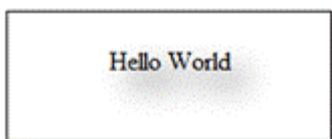
/*WebKit 核心浏览器语法 */
-webkit-box-shadow: 10px
                   10px
                   25px
                   #CCC;
-webkit-text-shadow: 10px
                    10px
                    25px
                    #CCC;

/*CSS 标准语法 */
box-shadow: 10px
           10px
           25px
           #CCC;

text-shadow: 10px
            10px
            25px
            #CCC;
```

如上面代码所示, 第一个参数表示阴影水平偏移量, 第二个表示阴影垂直偏移量, 第三个参数表示阴影扩散的半径, 第四个参数定义阴影的颜色。

图 3. 块阴影和文字阴影示例



另外我们还可以定义多个阴影中间用逗号隔开, 阴影的颜色可以使用 **RGBA** 透明色定义, 例如

清单 4. 多阴影代码示例

```
text-shadow: 0 0 10px rgba(0,255,0,0.5), -10px 5px 4px
            rgba(0,0,255,0.4), 15px -4px 3px rgba(255,0,0,0.7);
```

目前支持块阴影和文字阴影的属性有: Safari 3+、Chrome 1+、Firefox 3.0.5+, IE 全系列浏览器和 Opear 不支持该属性。

渐变色

渐变色是 Web 设计中很常用的一项元素，它可以增强 Web 元素的立体感同时使单一颜色的页面看起来不是那么突兀。过去为了实现渐变色我们通常需要先制作一个渐变的图片，将它切割成很细的一小片，然后使用背景重复使整个 HTML 元素拥有渐变的背景色。这样做有两个弊端：为了使用图片背景很多时候使得本身简单的 HTML 结构变得复杂；另外受制于背景图片的长度或宽度，HTML 元素不能灵活的动态调整大小。CSS3 中 Webkit 和 Mozilla 对渐变都有强大的支持，但是不同之前的一些属性，这次两个厂商没有能采用一致的语法。

清单 5.Mozilla 渐变色语法

```
background: -moz-linear-gradient(
center top,/* 渐变开始的坐标 */
rgba(254, 216, 80, 0.75),/* 渐变开始颜色 */
rgba(230, 125, 30, 0.75) 50%,/* 中间渐变颜色 */
rgba(254, 235, 121, 0.75))/* 渐变结束颜色 */
)
repeat scroll 0
0 transparent;
```

如上面代码所示，Mozilla 采用 `-moz-linear-gradient` 标签表示线性渐变。第一个参数表示渐变开始的坐标，可以是坐标值，也可以是 `top`、`bottom`、`left`、`right`、`center` 等值。后面参数是渐变的颜色值，个数不限，之间用逗号隔开。每个颜色值可以是普通的十六进制颜色也可以是 RGBA 颜色值。每个颜色后面还可以跟一个百分比或者 0-1 之间的小数，表示该颜色在整个渐变色中的比例。

清单 6.Webkit 渐变色语法

```
background:-webkit-gradient(
linear,/* 渐变类型线性 */
left top,/* 渐变开始的坐标 */
left bottom,/* 渐变结束的坐标 */
from(rgba(254, 216, 80, 0.75)),/* 渐变开始颜色 */
to(rgba(254, 235, 121, 0.75)),/* 渐变结束颜色 */
color-stop(0.5,rgba(230, 125, 30, 0.75))/* 渐变的中间颜色 */
)
repeat scroll 0 0 transparent;
```

Webkit 浏览器使用 `-webkit-gradient` 属性表示渐变。第一个参数是渐变类型，通常是 `linear` 线性渐变。第二个参数是渐变开始的坐标，和 Mozilla 的第一个参数相同。第三个参数是渐变结束的坐标。第四个和第五个分别是渐变开始和结束的颜色，可以是十六进制颜色值，也可以是 RGBA 颜色值。最后的 `color-stop` 属性可以有无数个，在前五个参数的后面以逗号隔开，表示渐变中间的渐变颜色。在 `color-stop` 属性中，第一个参数是该渐变色占得比例，可以是 0-1 的一个小数或者是一个百分数；第二个参数是渐变的颜色值同样可以是十六进制的颜色值或者 RGBA 的颜色值。

图 4. 渐变色示例



从上面的效果图可以看出，线性渐变是一个很强大的功能。使我们使用很少的 CSS 代码就能做出以前需要使用很多图片才能完成的效果，同时该功能是浏览器原生态支持的，渲染更快效果更好。很可惜的是目前支持该属性的浏览器只有最新版的 Safari、Chrome、Firefox 浏览器支持，且语法差异较大。

形变

形变 Transform 是 CSS 继线性颜色渐变之后的又一枚重磅炸弹，通常使用 CSS 和 HTML 我们是不可能使 HTML 元素旋转或者倾斜一定角度的。为了使元素看起来更具有立体感，我们不得不把这种效果做成一个图片，这样就限制了很多动态的使用应用场景。Transform 属性的引入使我们以前通常要借助 SVG 等矢量绘图手段才能实现的功能，只需要一个简单的 CSS 属性就能实现。在 CSS3 中 Transform 属性主要包括 rotate 旋转、scale 缩放、translate 坐标平移、skew 坐标倾斜、matrix 矩阵变换。下面我们就来看看每个属性是如何使用的。

清单 7. 旋转代码示例

```

/*Webkit 核心浏览器 */
-webkit-transform: rotate(-90deg);

-webkit-transform: scale(2);
-webkit-transform: scale(2, 1);

-webkit-transform: translate(10px, 20px);

-webkit-transform: skew(30deg, -10deg);

-webkit-transform: matrix(1, -0.2, 0, 1, 0, 0);

/*Firefox 浏览器 */
-moz-transform: rotate(-90deg);

-moz-transform: scale(2);
-moz-transform: scale(2, 1);

-moz-transform: translate(10px, 20px);

-moz-transform: skew(30deg, -10deg);

-moz-transform: matrix(1, -0.2, 0, 1, 0, 0);

/*Opera 浏览器 */
-o-transform: rotate(-90deg);

-o-transform: scale(2);
-o-transform: scale(2, 1);

-o-transform: translate(10px, 20px);

-o-transform: skew(30deg, -10deg);

-o-transform: matrix(1, -0.2, 0, 1, 0, 0);

```

旋转属性代码非常简单，rotate 属性加上旋转角度参数，45deg 代表顺时针方向旋转 45 度。如果是逆时针旋转 45 度则是 -45deg。

类似于旋转，缩放属性是由 scale 关键字加上缩放参数实现的。当只有一个参数 2 时表示该 HTML 元素的 X 轴、Y 轴方向同时被放大 2 被，0.5 表示同时缩小一半。如果同时有 2 和 3 两个参数时，表示该 HTML 元素的 X 轴被放大 2 被 Y 轴方向被放大 3 被。

坐标平移属性顾名思义，就是将该 HTML 元素向 X、Y 轴方向平移若干像素，由 translate 属性实现。后面两个参数分别表示向 X 轴、Y 轴平移的量。

skew 属性也是一个很有用的 transform 功能，它可以将一个对象围绕着 X 和 Y 轴按照一定的角度倾斜。这和 rotate 的旋转不一样，rotate 只是旋转而不会让 HTML 元素的形状改变，而 skew 会让这个 HTML 元素的形状改变。skew 有两个参数，分别代表该 HTML 元素沿 X 和 Y 轴的倾斜度数。

Matrix, 你没有看错, 就是我们通常的矩阵变换。这个变换就是我们在解析几何中学到的坐标系变换。他有六个参数 (a, b, c, d, e, f), 是一个 3×3 的矩阵, 代表坐标变换的变换矩阵。使用它我们就可以很灵活的完成任意的坐标系变换。有兴趣的朋友话可以查看大学解析几何的课本, 或者 W3c 在 SVG 中关于 Matrix 变化的定义和描述。

目前支持这 5 中变换转属性的浏览器有 Safari 4+、Chrome 5+、Firefox 3.5+、Opera10.5+, IE 全系列浏览器不支持该属性。

[回页首](#)

应用举例

之前对 CSS3 的一些新特性如何使用以及在各个浏览器中的实现做了详细的介绍。那么如何使用这些属性来开发一些绚丽的应用呢? 下面我们就来看一个例子。

图 5.CSS 聊天框





Message Bubble 是 Web 应用上很常用的 UI 元素, 通常较为美观的 Message Bubble 都是通过图片加上一些 JavaScript 来实现的。上面几个是用纯 CSS3+HTML 编写的 Message Bubble 例子, 这样不但实现简单, 而且可以根据需要动态切换样式。下面就结合 Message Bubble 的小例子讲解如何应用我们之前介绍的 CSS3 的新特性。

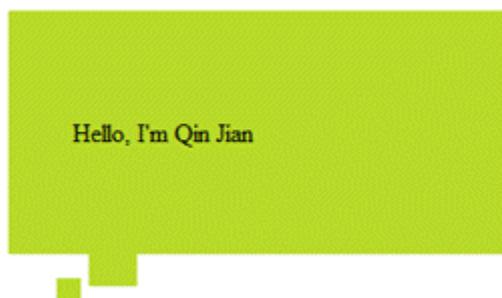
实现这类带不规则角的 Message Bubble 的关键是使用 :before、:after 两个伪类和 content 属性。通过 before、:after 两个伪类在 bubble 的 html 元素后面添加另外两个 dom 节点, 通过 content 属性将其内容设置为非空。之后通过调整节点大小和 border 的宽度来实现出不同的角样式。如下面代码所示:

清单 8.Message bubble 代码示例

```
<blockquote class="message-bubble">
<p>Hello, I'm Qin Jian</p>
</blockquote>
.message-bubble {
background:#B8DB29;
margin:80px;
padding:50px 40px;
position:relative;
width:230px;
}
.message-bubble:before {
background:#B8DB29;
bottom:-20px;
content:" ";
height:30px;
left:50px;
position:absolute;
width:30px;
z-index:-1;
}
.message-bubble:after {
content:" ";
background:#B8DB29;
height:15px;
width:15px;
position:absolute;
left:30px;
bottom:-30px;
z-index:-1;
}
```

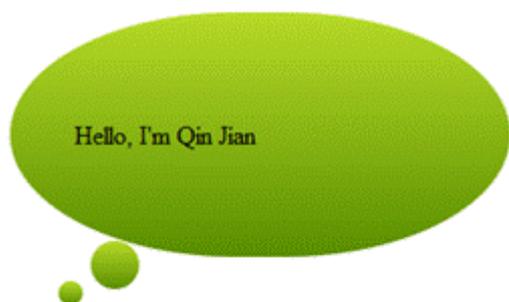
上面的代码是一个基本的 Message Bubble 的例子。首先为 Message Bubble 添加基本 HTML 结构, 加入你想要显示的文字。之后使用伪类 before 和 after 为通过设置 content 属性不为空为 Bubble 添加两个 dom 节点。通过设置这两个节点的大小、四个边的边框宽度、位置等属实现不同的 Bubble 角的效果。上面代码运行后将有下列的效果。

图 6.Message Bubble 基本效果



这样我们就得到了一个基本的 Message Bubble 的形状，接下来使用我们之前介绍的 CSS3 新特性，为 Bubble 加上一点圆角、背景渐变，效果如下：

图片 7. 基本效果加上渐变和圆角后的效果



这个 bubble 的效果是不是好了很多呢？之后我们再加上一点文字阴影、边框阴影、还有鼠标悬停在节点上时节点放大和旋转效果。

图片 7. 加上阴影、文字和旋转后的效果



这样一个效果很酷且具有很强交互性的 Message-Bubble 就做好了。重要的是我们没有使用任何图片，只需要更改 CSS 属性我们就可以变换出各种各样的样式。例如图 5 所示可以通过调整两个伪类生成节点的大小和边框宽度来设计出各种各样的角的类型。有兴趣的读者可以自己实验一下。以下是这个例子的全部 CSS 代码。

清单 9. 本例的全部 CSS 代码

```
.message-bubble {
background:#B8DB29;
margin:80px;
padding:50px 40px;
position:relative;
width:230px;

-moz-border-radius:220px 220px 220px 220px / 120px 120px 120px 120px;
```

```

-webkit-border-radius:220px 220px 220px 220px / 120px 120px 120px 120px;
background:-moz-linear-gradient(center top , #B8DB29, #5A8F00) repeat scroll \
0 0 transparent;
background:-webkit-gradient(
    linear,
    left top,
    left bottom,
    from(#B8DB29),
    to(#5A8F00)
)
repeat scroll 0 0 transparent;

text-align:center;
border:1px solid #CCC;
color:#FFFFFF;
-webkit-box-shadow:10px 10px 20px #000000;
-moz-box-shadow:10px 10px 20px #000000;
box-shadow:10px 10px 20px #000000;
}
.message-bubble p {
font-size:1.25em;
text-shadow:5px 5px 10px #000000;
}
.message-bubble:before {
background:#B8DB29;
bottom:-20px;
content:" ";
height:30px;
left:50px;
position:absolute;
width:30px;
z-index:-1;

-moz-border-radius:30px 30px 30px 30px;
-webkit-border-radius:30px 30px 30px 30px;
background:-moz-linear-gradient(center top , #B8DB29, #5A8F00) repeat scroll \
0 0 transparent;
background:-webkit-gradient(
    linear,
    left top,
    left bottom,
    from(#B8DB29),
    to(#5A8F00)
)
repeat scroll 0 0 transparent;

border:1px solid #CCC;
-webkit-box-shadow:10px 10px 20px #000000;
-moz-box-shadow:10px 10px 20px #000000;
box-shadow:10px 10px 20px #000000;
}
.message-bubble:after {
background:#B8DB29;
bottom:-30px;
content:" ";
height:15px;
left:30px;
position:absolute;
width:15px;
z-index:-1;

-moz-border-radius:15px 15px 15px 15px;
-webkit-border-radius:15px 15px 15px 15px;
background:-moz-linear-gradient(center top , #B8DB29, #5A8F00) repeat scroll \
0 0 transparent;
background:-webkit-gradient(
    linear,
    left top,
    left bottom,

```

```

    from(#B8DB29),
    to(#5A8F00)
  )
  repeat scroll 0 0 transparent;

border:1px solid #CCC;
-webkit-box-shadow:10px 10px 20px #000000;
-moz-box-shadow:10px 10px 20px #000000;
box-shadow:10px 10px 20px #000000;
}

.message-bubble:hover{
cursor:pointer;
-webkit-transform:scale(1.2) rotate(-5deg);
-moz-transform:scale(1.2) rotate(-5deg);

-webkit-transition-property:-webkit-transform;
-webkit-transition-duration:2s;
}

```

使用 CSS3 新特性的几点建议

CSS3 为我们带来的很多令人激动的新特性，使我们使用很简单的代码就能实现以前不能实现的效果和功能。然而由于这些新特性在不同浏览器中的支持不同，特别是 IE 浏览器的支持，使得我们很难做到兼容浏览器且视觉效果统一。以下作者根据对 CSS3 的使用经验给出几个使用上的建议。

- 1 CSS3 的使用不应当影像页面在各个浏览器中的正常显示。我们可以使用 CSS3 的一些属性来增强页面表现力和用户体验，但是这个效果提升不应当影响其他浏览器用户正常访问该页面。用户应当还能正常使用页面的功能，只是页面效果不是那么好罢了。
- 1 同一页面在不同浏览器中不必完全显示一致。功能较强的浏览器页面可以显示的更炫一些，而较弱的浏览器可以显示的不是那么酷，只要能完成基本的功能就行。大可不必为了在各个浏览器中得到同样的现实效果而大费周折。
- 1 在某些不支持 CSS3 的浏览器中，我们可以使用替代方法来实现这些效果，但是需要平衡实现的复杂度和实现的性能问题。

[回页首](#)

小结

CSS3 虽然还没有正式发布，但是各个浏览器厂商已经开始迫不及待的支持部分新特性了。本文在该系列文章第 1 部分的基础上，对 CSS3 的部分新特性进行了详细的介绍，并结合一个实例讲解如何使用这些新特性来开发一些应用。相信良好的艺术创意加上 CSS3 的变现能力，可以完成很多我们意想不到的效果和功能。然而 IE6/7/8 系列浏览器都不支持 CSS3 特性，迟迟未发布的 IE9 对 CSS3 支持也很有限，这使得很多开发人员因为 IE 的兼容性问题对 CSS3 望而却步。在该系列文章的最后一部分，将详细介绍如何在 IE 系列浏览器中使用替代方案来实现部分 CSS3 效果。

参考资料

学习

- 1 参考 W3C CSS 规范 Level 1: [CSS 规范 Level1](#)。
- 1 参考 W3C CSS 规范 Level 2/2.1: [CSS 规范 Level2/2.1](#)。
- 1 参考 W3C CSS 规范 Level 3: [CSS 规范 Level3](#)。

- | [“基于 CSS3 的下一代 Web 应用开发, 第 1 部分: 发展历史及新特性”](#): CSS3 大大简化了编程模型, 它不仅是对已有功能的扩展和延伸, 而更多的是对 Web UI 设计理念的和方法的革新。本文首先将介绍 CSS 的发展历史, 然后讲解了各个版本的功能特点和差异, 随后重点讲解了 CSS3 引入的新特性, 并给出一些应用实例。
- | [developerWorks Web development 专区](#): 通过专门关于 Web 技术的文章和教程, 扩展您在网站开发方面的技能。
- | [developerWorks Ajax 资源中心](#): 这是有关 Ajax 编程模型信息的一站式中心, 包括很多文档、教程、论坛、blog、wiki 和新闻。任何 Ajax 的新信息都能在这里找到。
- | [developerWorks Web 2.0 资源中心](#), 这是有关 Web 2.0 相关信息的一站式中心, 包括大量 Web 2.0 技术文章、教程、下载和相关技术资源。您还可以通过 [Web 2.0 新手入门](#) 栏目, 迅速了解 Web 2.0 的相关概念。
- | 查看 [HTML5 专题](#), 了解更多和 HTML5 相关的知识和动向。

基于 CSS3 的下一代 Web 应用开发, 第 3 部分: 在 IE 中的特殊处理

[秦剑](#), 软件工程师, IBM

简介: 本文是《基于 CSS3 的下一代 Web 应用开发》系列文章的第三部分, 第一部分我们将介绍了 CSS 的发展历史, 个版本的功能特点差异和 CSS3 引入的新特性。第二部分, 我们针对这些 CSS3 新特性介绍其在不同浏览器中实现的差别, 并结合实例讲解如何利用这些特性进行开发。第三部分, 由于 IE6/7/8 系列浏览器不支持 CSS3 的新特性, 使得很多开发人员对这 CSS3 新功能望而却步。本部分将详细介绍如何在 IE 中实现这些功能。

[查看本系列更多内容](#)

本文的标签: [css](#), [dhtml \(dynamic html\)](#), [javascript](#), [用户界面](#)

[标记本文!](#)

发布日期: 2011 年 5 月 05 日

级别: 中级

Opacity 透明度

透明度属性是 CSS 中非常常用和流行的一个属性, 目前已经广泛被主要浏览器支持, 可惜的是 IE6/7/8 均不支持该属性。虽然 IE 没有支持这个 CSS 属性, 但是我们可以通过使用 IE 特有的滤镜 filter 属性来实现相同效果。

清单 1. Opacity 代码示例

```
opacity: 0.4; /*Chrome、Safari、Firefox、Opera */
filter: progid:DXImageTransform.Microsoft.Alpha(opacity=40); /* IE6/IE7/8 */
-ms-filter: "progid:DXImageTransform.Microsoft.Alpha(opacity=40)"; /* IE8 */
```

如上面代码所示, 第一行是在其他浏览器中使用 Opacity 属性; 第二行是 IE6/7/8 中使用滤镜属性设置透明度; 第三行是新的 IE 滤镜属性, 它只在 IE8 中起作用在其他浏览器中会被忽略, 如果不是针对 IE8 设置, 使用第二行代码就可以了。参数的设置与 CSS 中 Opacity 属性也基本相同, 在 IE 滤镜中使用 0-100 表示透明度, 因此 opacity=40 相当于 Opacity 属性为 0.4, 亦即透明度为 40%。使用该滤镜属性有两个缺点: 由于使用了 Microsoft 特有的属性, 会使得你的 CSS 非验证的; 另外, IE 的滤镜属性会使得所有的 HTML 子节点都继承这个属性。

图 1. IE 浏览器中透明度演示



[回页首](#)

Border-Radius 圆角效果

圆角效果是 CSS3 中另一个非常常用和流行的效果。它使得程序员不再需要费力的拼装很多圆角图片或者

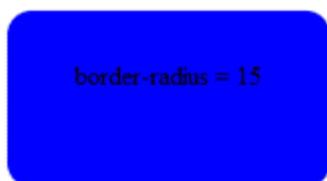
使用大量复杂的 JavaScript 来实现的相同的效果。在很大程度上圆角属性简化了 HTML 代码结构，同时优化了显示效果。然而 IE 全系列浏览器依然不支持这个属性。幸运的是 [Remiz Rahnas](#) 使用 VML 编写了一个 HTC 文件，为 IE 浏览器提供圆角效果的支持。

清单 2. 圆角代码示例

```
-moz-border-radius: 15px; /*Firefox*/
-webkit-border-radius: 15px; /*Safari、Chrome*/
border-radius: 15px; /*Opera 10.5+、IE 6+*/
behavior: url(ie-css3.htc); /* 调用脚本添加圆角效果 */
```

如上面代码所示，圆角属性的使用和 CSS 圆角属性一致，只是在后面多添加了一句 `behavior: url(ie-css3.htc)`。`behavior` 属性只被 IE 浏览器支持和识别，用来告诉 IE 在哪些设置了该样式 Class 的元素上调用脚本添加圆角效果。一句简洁的代码就使得 IE 支持圆角效果，而不需要你额外的维护任何代码。然而这种折中的方法也有一些缺陷：首先，在 Server 端需要引入一个 HTC 文件，经过 gzip 压缩后对 server 端和 client 端性能应该不会有太大的影响；其次，它会使得你的 CSS 验证不合法；另外，这个脚本在 IE8 上有一些问题，它会使得 `z-index` 值变成负数。因此使用时还需要小心。

图 2.IE 浏览器中圆角效果演示



[回页首](#)

Box Shadow 盒阴影

盒阴影是另一个 CSS3 中的很有用的属性，它使得程序员可以通过简单的添加一个属性，就创建一个立体效果带有阴影的元素。在 IE 全系列浏览器中依然在不支持这个属性，但 IE 为这个效果提供了滤镜属性；另外和上面圆角效果一样，我们可以通过 VML 脚本来实现这个效果。

清单 3.Box Shadow 的滤镜实现代码

```
-moz-box-shadow: 2px 2px 3px #969696; /* Firefox 3.5+ */
-webkit-box-shadow: 2px 2px 3px #969696; /* Safari and Chrome */
filter: progid:DXImageTransform.Microsoft.Shadow(color='#969696',
Direction=145, Strength=3);
```

清单 4.Box Shadow 的 VML 脚本实现

```
-moz-box-shadow: 2px 2px 3px #969696; /* Firefox */
-webkit-box-shadow: 2px 2px 3px #969696; /* Safari and Chrome */
box-shadow: 2px 2px 3px #969696; /* Opera 10.5+、IE6+*/
behavior: url(ie-css3.htc); /* 调用脚本添加阴影效果 */
```

如上面代码所示，上面两种方法均可实现这个效果。但是要注意的是，IE 滤镜效果语法与其他浏览器中 CSS3 属性语法不同。因此需要开发人员仔细调试使得在不同浏览器中看起来有相同的效果。

图 3.IE 浏览器中盒阴影效果演示


[回页首](#)

Text Shadow

文字阴影在不仅流行于打印效果中，同时也在 Web 设计中也非常流行。然而对于 Text Shadow 这个属性我们就没有那么好运气了，IE 没有提供相应的滤镜效果，也没有现成的 VML 脚本可以使用。在过去的 Web 开发中，我们通常使用图片来实现文字阴影效果。[Kilian Valkhof](#)为了解决 IE 下实现文字阴影这个难题，而编写了一个 [jQuery](#) 的插件。

清单 5.IE 浏览器中实现 Text Shadow 效果代码

```
text-shadow: #aaa 5px 5px 8px;

$(document).ready(function(){
  $(".text-shadow").textShadow();
});
```

如上面代码所示，下载了 jQuery 核心包和 Drop Shadow 插件后我们就可以在 IE 中使用文字阴影效果了。textShadow() 方法还可以填入一个 JavaScript 对象参数，如下表所示：

表 1.textShadow 属性参数表

属性名	类型	默认值	描述
left	整型	4	阴影距离
top	整型	4	阴影角度
blur	整型	2	阴影扩散度
opacity	0-1 小数	0.5	阴影透明度
color	颜色值	black	字体阴影颜色
swap	布尔	false	是否换行

图 4.Text Shadow 演示图


[回页首](#)

Gradients 渐变色

CSS3 中的渐变色为渐变背景色提供了很大的方便，我们不必为了渐变的背景色使用大量细小的图片，同时也不用为了适应浏览器分辨率做很多工作。尽管 IE 浏览器还是不支持该属性，我们依然可以通过 IE 是由的滤镜属性实现该效果。

清单 6.IE 浏览器中实现渐变色

```
background-image: -moz-linear-gradient(top, #81a8cb, #4477a1); /* Firefox 3.6 */
background-image: -webkit-gradient(linear, left bottom, left top,
color-stop(0, #4477a1), color-stop(1, #81a8cb)); /* Safari & Chrome */
filter: progid:DXImageTransform.Microsoft.gradient(GradientType=0,
startColorstr='#81a8cb', endColorstr='#4477a1'); /* IE6 & IE7 */
-ms-filter: "progid:DXImageTransform.Microsoft.gradient(GradientType=0,
startColorstr='#81a8cb', endColorstr='#4477a1')"; /* IE8 */
```

如上面代码所示，我们可以使用 IE 的私有的滤镜来实现该效果。其中 `GradientType` 可以有两个选项值 0 和 1，分别代表水平渐变和垂直渐变。`startColorstr` 和 `endColorstr` 分别代表渐变开始的颜色值和结束的颜色值。虽然参数和功能不如 CSS3 中的渐变多，但是对已一般的需求来说也算够用了。另外该滤镜属性在 IE6/7 和 IE8 中支持的语法不相同，因此我们需要写不同的代码去做 IE6/7 和 IE8 的兼容处理。

图 5.IE 浏览器渐变色演示



[回页首](#)

RGBA 颜色

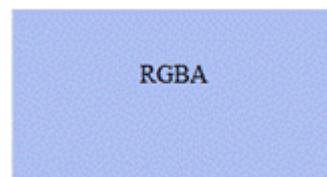
CSS3 中提供了另一种方式设置背景透明度，那就是 RGBA 颜色。RGBA 颜色使得开发人员在指定颜色时，不仅可以指定 R、G、B 三原色的值，同时还可以指定颜色的透明度。这样我们就就可以在浏览器中实现类似 Windows7 中一样透明的玻璃效果，这大大增强了 Web 程序的视觉感官体验。在 IE 全系列浏览器中依然不支持 RGBA 颜色，我们只能利用 IE 滤镜模拟实现这样的效果。

清单 7.IE 浏览器 RGBA 颜色实现代码

```
background: rgba(50, 95, 224, .4);
filter: progid:DXImageTransform.Microsoft.gradient(GradientType=0,
startColorstr='#886287a7', endColorstr='#886287a7');
```

如上面代码所示，利用 IE 中渐变色滤镜将渐变开始颜色和结束颜色设置相同，能模拟出和其他浏览器十分类似的 RGBA 颜色效果。另外为了在 IE 中获得较好的透明背景效果，使用 PNG 图片作为背景图片也是一个很不错的选择。不过这样做的缺点是你不得不处理大量的图片，同时为了适应图片背景不得不修改 HTML 的结构。

图 6.IE 浏览器中滤镜实现 RGBA 颜色效果



[回页首](#)

Rotation 旋转

在最新的 Firefox 和 Webkit 浏览器中都对 CSS3 形变和动画效果做了不同程度的支持。你可以旋转、拉伸、平移一个 HTML 元素来实现以前只能用图片实现的立体效果，也可以使用 CSS 属性来实现绚丽的淡入、淡出等动画效果。目前 IE 全系列浏览器还不支持该属性，但是很少有人知道使用 IE 的滤镜可以实现简单的 HTML 元素旋转的效果。

清单 8.IE 浏览器中实现旋转

```
-webkit-transform: rotate(90deg);
-moz-transform: rotate(90deg);
filter: progid:DXImageTransform.Microsoft.BasicImage(rotation=1);
```

如上面代码所示，最后一行代码在 IE 中实现旋转效果。其中的参数 rotation 的选项为 1、2、3 和四，分别代表旋转 90 度、180 度、270 度和 360 度。和 CSS3 中的旋转 360 度功能相比，IE 中的旋转滤镜功能就显得十分有限了，每次 90 度的旋转很难满足开发人员的需求。

另外 dojo 在 1.5 版本对 CSS3 中的 transform 属性开始有一定的支持，其中包括了 matrix、rotate、skew、scale、translate 等功能。如果有 CSS3 形变需求的话，可以参考使用 dojo 对这部分功能的扩展。

清单 9. 使用 dojo 设置 HTML 元素旋转

```
dojox.html.ext-dojo.style("transform", "rotate(10deg)");
```

图 7.IE 浏览器中滤镜实现 HTML 元素旋转



[回页首](#)

小结

CSS3 虽然还没有正式发布，但是各个浏览器厂商已经开始部分支持这些新特性了。这些新特性的支持给我带来了惊喜，使我们很容易实现我们意想不到的华丽的效果，简化了很多前端实现的代码。但是 CSS3 也给我们带来了很多困扰，各个浏览器厂商对 CSS3 标准支持不一致，尤其是特立独行的 IE 给开发人员带来更多的麻烦。为了实现 Web 程序的跨浏览器和显示的一致性，我们不得不花费很多时间在各个浏览器差异的调试上。本文针对部分 IE 不支持的 CSS3 效果以及在 IE 浏览器上的替代方案做了详细的介绍。希望能为广大开发者在未来的开发工作中提供一点启示。

参考资料

学习

- 1 参考 W3C CSS 规范 Level 1: [CSS 规范 Level1](#)。
- 1 参考 W3C CSS 规范 Level 2/2.1: [CSS 规范 Level2/2.1](#)。
- 1 参考 W3C CSS 规范 Level 3: [CSS 规范 Level3](#)。

- | [“基于 CSS3 的下一代 Web 应用开发, 第 1 部分: 发展历史及新特性”](#): CSS3 大大简化了编程模型, 它不仅是对已有功能的扩展和延伸, 而更多的是对 Web UI 设计理念的和方法的革新。本文首先将介绍 CSS 的发展历史, 然后讲解了各个版本的功能特点和差异, 随后重点讲解了 CSS3 引入的新特性, 并给出一些应用实例。
- | [“基于 CSS3 的下一代 Web 应用开发, 第 2 部分: 发展历史及新特性”](#): CSS3 本文将继续第 1 部分的内容, 为您详细介绍 CSS3 新增的特性和 UI 效果, 讲解其在不同浏览器中实现的差别, 并结合实例说明如何利用这些特性进行开发。
- | [developerWorks Web development 专区](#): 通过专门关于 Web 技术的文章和教程, 扩展您在网站开发方面的技能。
- | [developerWorks Ajax 资源中心](#): 这是有关 Ajax 编程模型信息的一站式中心, 包括很多文档、教程、论坛、blog、wiki 和新闻。任何 Ajax 的新信息都能在这里找到。
- | [developerWorks Web 2.0 资源中心](#), 这是有关 Web 2.0 相关信息的一站式中心, 包括大量 Web 2.0 技术文章、教程、下载和相关技术资源。您还可以通过 [Web 2.0 新手入门](#) 栏目, 迅速了解 Web 2.0 的相关概念。
- | 查看 [HTML5 专题](#), 了解更多和 HTML5 相关的知识和动向。