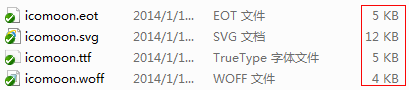
**6 IconFont**

**使用图标字体的优势和劣势**

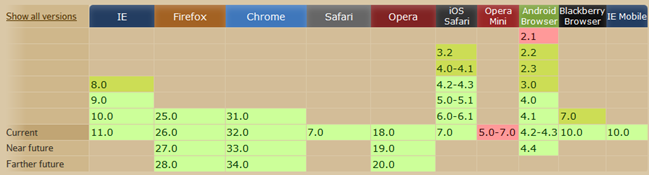
字体图标除了图像清晰度之外，比位图还有哪些优势呢?  
1、**轻量性**：一个图标字体比一系列的图像（特别是在Retina屏中使用双倍图像）要小。一旦图标字体加载了，图标就会马上渲染出来，不需要下载一个图像。可以减少HTTP请求，还可以配合HTML5离线存储做性能优化。



2、**灵活性**：图标字体可以用过font-size属性设置其任何大小，还可以加各种文字效果，包括颜色、Hover状态、透明度、阴影和翻转等效果。可以在任何背景下显示。使用位图的话，必须得为每个不同大小和不同效果的图像输出一个不同文件。



3、**兼容性**：网页字体支持所有现代浏览器，包括IE低版本。详细兼容性可以点击[这里](http://caniuse.com/#feat=fontface)。



除了以上优势之外，当然也有劣势  
1、图标字体只能被渲染成单色或者CSS3的渐变色，由于此限制使得它不能广泛使用。  
2、使用版权上有限制，有好多字体是收费的。当然也有很多免费开源的精美字体图标供下载使用。  
3、创作自已的字体图标很费时间，重构人员后期维护的成本偏高。

**如何获取图标字体及使用**

要想获取图标字体，不外乎两种途径，其一找到付费网站购买，其二就是到免费网站下载，提供免费下载网站很多，神飞曾经发表过一篇博客[icon font大搜罗](http://www.qianduan.net/icon-font-large-collecting.html)，上面罗列好多的免费网站的地址，大家有兴趣可以去下载。如何使用呢？一般来说，有3种方法：  
**1、把字符直接写在HTML文件里；**  
这个方法是简单比较直观，见如下代码，用一个<span>元素去包含一个字符“!”(或&#x0021;)，然后给这个<span>添加一个类。这个字母在选定的字体中被映射到一个特定的图标。

<a href=”javascript:;”><span cass=”icon”>!</span>赞</a><a href=”javascript:;”><span cass=”icon”>&#x0021;</span>赞</a>

为了显示效果，还需要编写样式类.Icon来决定此字符以哪种字体来显示，如下：

.icon {font-family: ‘ your-incofont -name ‘;}

**2、使用css来生成内容；**  
它不直接在HTML文件里添加字符，而是用CSS来生成字符内容。代码如下

<a href=”javascript:;” class=”icon praise”>赞</a>

可以看出，添加了一个类名“praise”。神奇的事就发生在CSS中，跟上面一样，第一步先定义好字体，然后使用:before伪元素来产生字符图标，其中“before”表示字符出现在左边，“after”则出现在右边。

.icon {font-family:’ your-incofont-name’ ;}.praise:before {content: “\f00a”; }

**3、用data-icon属性**  
还有一种跟上面相似方法是使用HTML5的“data-”属性。如：创建一个data-icon属性。  
aria-hidden=”true”是为了防止被阅读器直接把字符读取出来，不是对所有的平台都奏效。

<a href=”javascript:;”><span aria-hidden=”true” data-icon=”!”></span>赞</a>

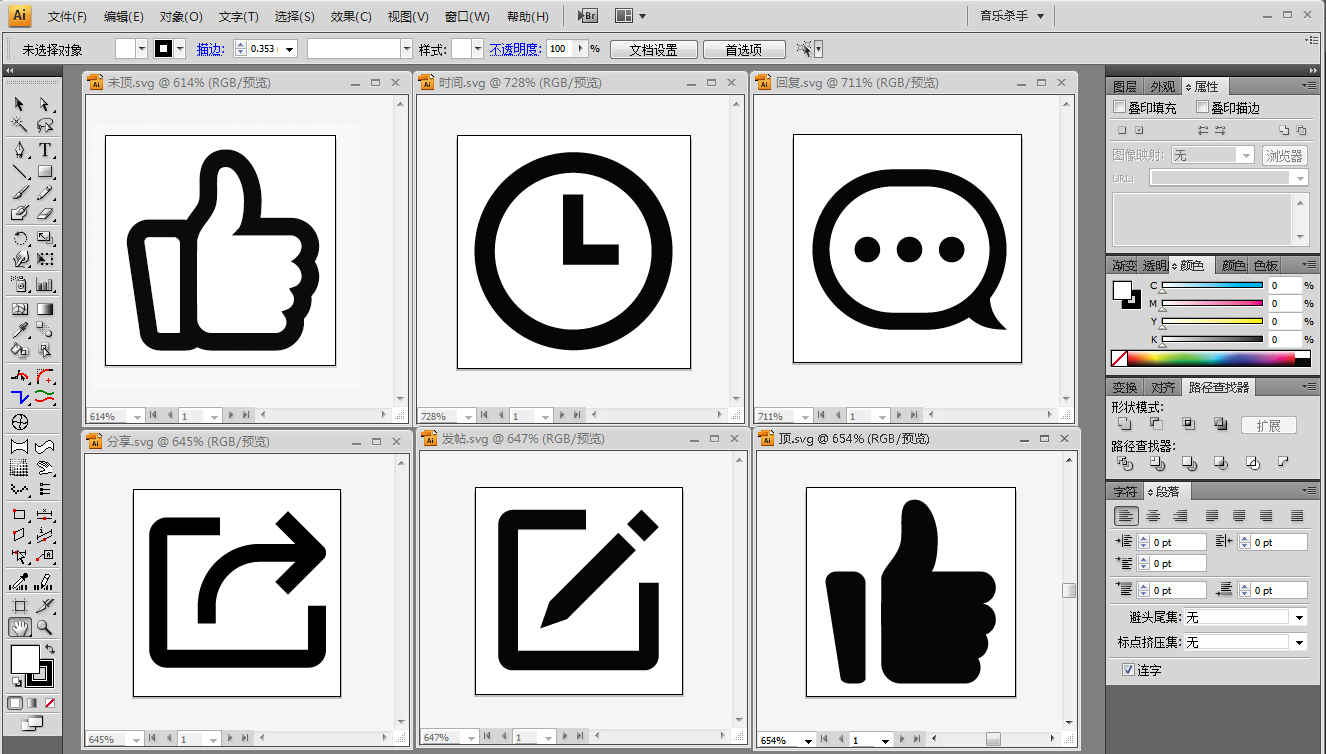
结合一些搭配使用的CSS属性，可以写成如下代码

[data-icon]:before {font-family: ‘ your-incofont-name’; content: attr(data-icon);speak:none;}

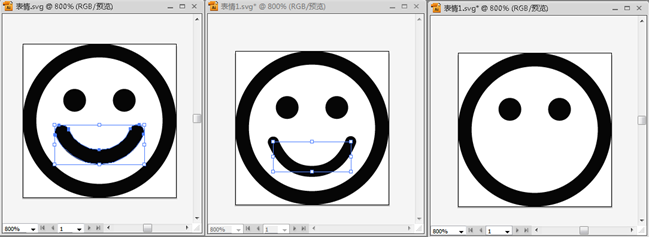
**制作自已图标字体**

以上是免费图标字体使用的几种方法，下面看看如何制作属于自己的图标字体。

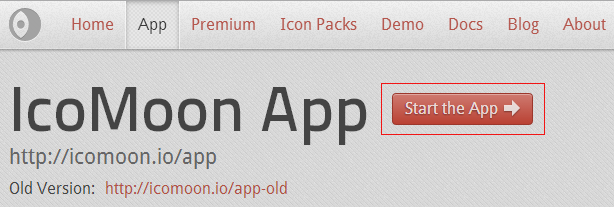
1、首先需要有创建矢量图标的软件，并且能够输出SVG格式，比如“Illustrator”或者“Inkscape”。也可以用Photoshop的路径工具画出所需要的图标，然后导出路径到Illustrator里去填色。 下图为Adobe Illustrator软件。



注意一定是封闭的路径，不能是单路径描边，如果是单路径的话生成字体图标后会显示不出来。两个以上的图形要合并、图形尽减少节点使用。如下图，中间的图(嘴巴部分)为单路径描边（不封闭）,导入图标生成器后会如右图所示，嘴巴部分显示不出来。



2、图标制作完成后，选择 “文件”菜单中“保存”，把文件保存成“SVG”格式。使用默认的SVG设置。  
3、把SVG格式的图标导入到[icomoon](http://icomoon.io/app/)、[fontello](http://fontello.com/)、[iconfont](http://www.iconfont.cn/)字体生成器中，去生成所要的图标字体，以下介绍一下几种字体生成器的用法：  
**（1）图标字体生成器**[**http://icomoon.io/app/**](http://icomoon.io/app/)  
IcoMoon！一个可以通过个性化设置来创建自定义图标（字体）的生成器！IcoMoon是一项免费的服务，通过使用不同设置使我们能够创建自定义的Icon图或Icon字体。除了自定义Icon外，IcoMoon也有自己免费的海量图标集，都非常赞。  
打开地址，点击Start the App按钮。



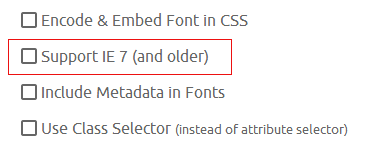
点击Import Icons按钮导入SVG图标，导入后对图标进行相关的操作，如选中、删除、移动、编辑等。



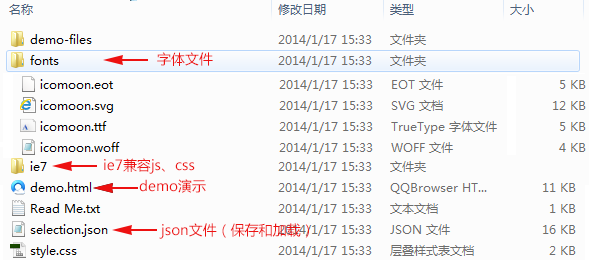
编辑完成后，就可以进行下载了，它提供两种下载方式：图片版和字体版！图片版是经过CSS Sprites技术处理的PNG格式，字体版有多种格式供我们选择（EOT,SVG,WOFF,TTF）。



如果要兼容IE7以下浏览器，请在设置里勾选Support IE7(and older)选项，会生成一个单独JS、CSS文件。



下载ZIP包后，解压后会得到如下图的文件。将fonts文件夹复制到你的网站，为项目添加字体。  
从style.css文件中复制CSS样式，并粘贴到你网站的CSS文件中，也可以单独存成一个样式文件。



复制完成后，在CSS文件中找到@font-face，将URL路径修改成你本地的相对路径。



字体和路径都设置完成后，在HTML页面只需调用相对应的class就可以了。如果想兼容IE7浏览器，需引用IE7目录的js。如：

调用class：<span class=”icon icon-add”></span>  
调用js：<script src=”ie7/ie7.js”></script>

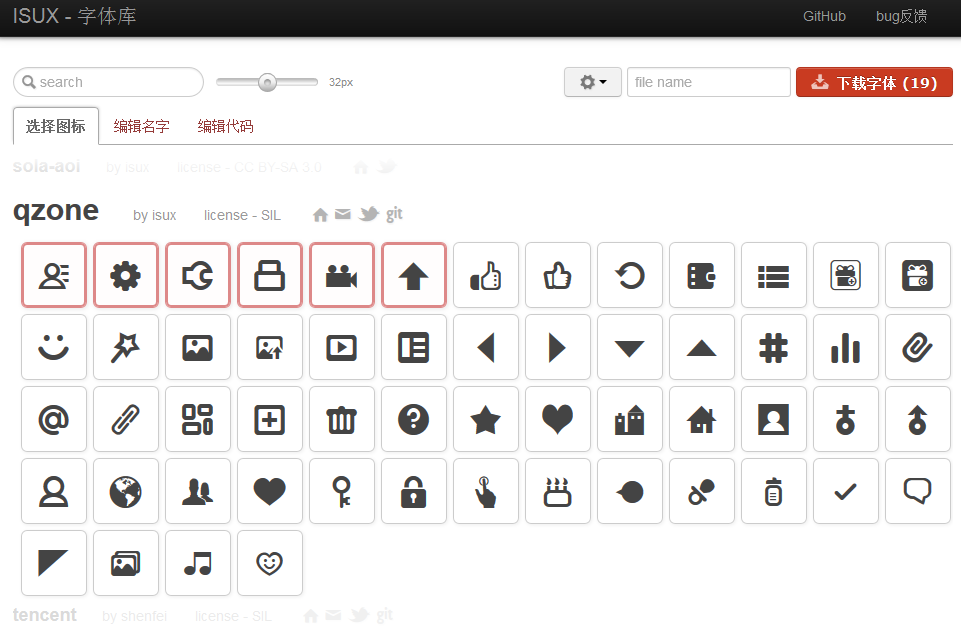
**（2）图标字体生成器**[**http://fontello.com/**](http://fontello.com/)

它的生成方式与上面的类同，这里不做介绍，很容易上手。



**（3）ISUX矢量字体图标库**[**http://font.isux.us/**](http://font.isux.us/)

ISUX矢量字体库目前提供了丰富的字体图标，可以供下载使用，暂时还不支持自定义图标的导入，期待不久的将来它的功能会越来越完善。大家有什么好的建议和想法可以联系他们。



通过以上几种生成器的生成图标字体，加上用CSS对其大小、颜色、透明度、阴影、Transition各种变换等控制，不仅可以缩放自如，制作出各种特殊效果，而且还很容易维护，可以通过多种不同的途径对它们进行操作。相信大家已经体验到他的强大之处，满足日常工作需要应该已经足够了。虽然有这么多优点，但图标字体并不是完美的，也存在些缺点。如：只能被渲染成单色的问题、屏幕阅读器(虽然有解决方法，但并不完善)的问题、性能问题等等， 等待着我们去发现和解决。相信未来会有更好的解决方案，比如：SVG（可伸缩矢量图形），未来可能取代位图的图形技术等。  
立足现在，放眼未来。最后对目前的图标字体生成器工具做一下展望吧！  
■ 能够支持导入更多的自定义格式，如EPS、AI等格式。  
■ 能够引入项目管理的机制， 在同一帐号可以同时管理多个项目图标。  
■ 提供更多免费、丰富的图标字体供下载使用。

**FAQ**

**1、跨域问题：**  
（1）通过配置自己的服务器。  
**# For Apache**  
<FilesMatch “.(eot|ttf|otf|woff)”>  
Header set Access-Control-Allow-Origin “\*”  
</FilesMatch>  
**# For nginx**  
location ~\* \.(eot|ttf|woff)$ {  
add\_header Access-Control-Allow-Origin \*;  
}  
（2）放在同一个域下。  
（3）使用base64置入CSS中(Icomoon在导出图标时，设置里勾选Encode & Embed Font in CSS选项)。  
**2、字体图标出现锯齿的问题：**  
-webkit-font-smoothing: antialiased;  
-moz-osx-font-smoothing: grayscale;  
**3、@font-face与性能：**  
关于@font-face的性能问题，可以参考以下文章。  
原文地址：<http://www.stevesouders.com/blog/2009/10/13/font-face-and-performance/>  
翻译地址：<http://www.cnblogs.com/demix/archive/2009/11/28/1612715.html>  
文章内的建议和总结:  
（1）只在你确定你非常需要 @font-face的时候才使用它;  
（2）将你的@font-face定义在所有的script标签前;  
（3）如果你有许多字体文件，考虑将它们分散到几个域名下;  
（4）不要包含没有使用的 @font-face声明——IE将不分它使用与否，通通加载;  
（5）Gzip字体文件，同时给它们一个未来的过期头部声明;  
（6）考虑字体文件的后加载，起码对于IE。

**7 Filter**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 不同的滤镜Filters | | | | |
| **属性** | **参数** | **描述** | **例子** | **应用详解链接** |
| alpha | opacity  finishopacity  style  startx  starty  finishx  finishy | 允许您设置元素的透明度 | filter:alpha(  opacity=20,  finishopacity=100,  style=1,  startx=0,  starty=0,  finishx=140,  finishy=270) | [渐变文字](http://qy-0824.blog.163.com/blog/static/72507542201112494426606/)  [图片CSS滤镜效果—透明](http://qy-0824.blog.163.com/blog/static/72507542200941311227246/)  [图片雾化移动特效代码](http://qy-0824.blog.163.com/blog/static/72507542201106103839671/)  [Flash圆形雾化移动特效代码](http://qy-0824.blog.163.com/blog/static/725075422011022113920759/)  [图片CSS滤镜—波形加上半透明合成效果](http://qy-0824.blog.163.com/blog/static/725075422009413104755608/)  [让图片产生梦幻效果的代码](http://qy-0824.blog.163.com/blog/static/7250754220094701049815/) |
| blur | add  direction  strength | 使元素模糊 | filter:blur(  add=true,  direction=90,  strength=6); | [模糊文字](http://qy-0824.blog.163.com/blog/static/72507542201112642416378/)  [图片CSS滤镜—图片模糊效果](http://qy-0824.blog.163.com/blog/static/7250754220094140363260/) |
| chroma | color | 使指定的颜色透明 | filter:chroma  (color=#ff0000) | [透明文字](http://qy-0824.blog.163.com/blog/static/725075422011128113529462/) |
| fliph | none | 水平反转元素 | filter:fliph; | [反转文字](http://qy-0824.blog.163.com/blog/static/7250754220111264523661/)  [图片CSS滤镜—左右旋转效果](http://qy-0824.blog.163.com/blog/static/72507542200941310225455/) |
| flipv | none | 垂直反转元素 | filter:flipv; | [反转文字](http://qy-0824.blog.163.com/blog/static/7250754220111264523661/)  [图片CSS滤镜—倒影效果](http://qy-0824.blog.163.com/blog/static/7250754220095222239900/) |
| glow | color  strength | 使元素发光 | filter:glow  (color=#ff0000,  strength=5); | [发光文字](http://qy-0824.blog.163.com/blog/static/72507542201112325341957/)  [图片CSS滤镜—发光效果](http://qy-0824.blog.163.com/blog/static/725075422009413114312616/)  [粒状阴影框](http://qy-0824.blog.163.com/blog/static/72507542200951111549186/) |
| gray | none | 用黑白色来呈现元素 | filter:gray; | [图片虚线边框并黑白效果](http://qy-0824.blog.163.com/blog/static/725075422009414903457/)  [图片CSS滤镜效果—灰度](http://qy-0824.blog.163.com/blog/static/72507542200941384735902/) |
| invert | none | 用反转的颜色和亮度值来呈现元素 | filter:invert; | [图片CSS滤镜—色彩对换效果](http://qy-0824.blog.163.com/blog/static/725075422009414077902/) |
| mask | color | 呈现带有指定背景色和透明前景色的元素 | filter:mask  (color=#ff0000); | 编辑文字图片内容 |
| shadow | color  direction | 呈现带有阴影的元素 | filter:shadow  (color=#ff0000,  direction=90); | [投射阴影文字](http://qy-0824.blog.163.com/blog/static/72507542201112352713511/)  [图片CSS滤镜—阴影效果](http://qy-0824.blog.163.com/blog/static/72507542200941310739923/)  [立体阴影框](http://qy-0824.blog.163.com/blog/static/72507542200951111549186/) |
| dropshadow | color  offx  offy  positive | 呈现带有阴影的元素 | filter:dropshadow  (color=#ff0000,  offx=5,  offy=5,  positive=true); | [阴影文字](http://qy-0824.blog.163.com/blog/static/7250754220111249915873/) |
| wave | add  freq  lightstrength  phase  strength | 把元素呈现为波浪状 | filter:wave  (add=true,  freq=1,  lightstrength=3,  phase=0,  strength=5) | [水波纹文字](http://qy-0824.blog.163.com/blog/static/72507542201112645742932/)  [图片CSS滤镜—倒影效果](http://qy-0824.blog.163.com/blog/static/7250754220095222239900/)  [图片CSS滤镜—扭曲效果](http://qy-0824.blog.163.com/blog/static/725075422009521476397/)  [图片CSS滤镜—图片动感模糊效果](http://qy-0824.blog.163.com/blog/static/7250754220095212712411/)  [图片CSS滤镜—波形效果](http://qy-0824.blog.163.com/blog/static/72507542200941402452857/)  [图片CSS滤镜—波形加上半透明合成效果](http://qy-0824.blog.163.com/blog/static/725075422009413104755608/) |
| xray | none | 使用黑白色显示带有反转色和亮度值的元素 | filter:xray; | [图片CSS滤镜—Ｘ光效果](http://qy-0824.blog.163.com/blog/static/72507542200941311408252/) |

可视化滤镜属性只能用在HTML控件元素上，不能使用div的id，class定义。所谓的HTML空间元素就是它们在页面上定义了一个矩形空间，浏览器的窗口可以显示这些空间。下面列出了HTML合法的控件和它们的说明。

元素 说明   
  
BODY 网页文档的主体元素，所有的可见范围都在元素内   
BUTTON 表单域的按钮，可以有“发送(submit)”、“重置(reset)”等形式   
DIV 定义了网页上的一个区域，这个区域的高度、宽度或者绝对位置都是以知的   
IMG 图片元素，通过指定“src'属性来指定图片的来源   
INPUT 输入表单域   
MARQUEE 移动字幕效果   
SPAN 定义了网页上的一个区域，这个区域的高度、宽度或者绝对位置都是以知的   
TABLE 表格   
TD 表格数据单元格   
TEXTAREA 文本区域   
TFOOT 多行输入文本框   
TH 表格标题单元格   
THEAD 表格标题   
TR 表格行   
  
////////////////////////////////////////

IE4.0以上支持的滤镜属性表

滤镜效果 描述 ：

Alpha 设置透明度   
Blru 建立模糊效果   
Chroma 把指定的颜色设置为透明   
DropShadow 建立一种偏移的影象轮廓，即投射阴影   
FlipH 水平反转   
FlipV 垂直反转   
Glow 为对象的外边界增加光效   
Grayscale 降低图片的彩色度   
Invert 将色彩、饱和度以及亮度值完全反转建立底片效果   
Light 在一个对象上进行灯光投影   
Mask 为一个对象建立透明膜   
Shadow 建立一个对象的固体轮廓，即阴影效果   
Wave 在X轴和Y轴方向利用正弦波纹打乱图片   
Xray 只显示对象的轮廓

下面分别阐述：

1、Alpha 滤镜   
语法：{FILTER：ALPHA(opacity=opacity,finishopacity=finishopacity,style=style,startx=startx,   
starty=starty,finishx=finishx,finishy=finishy)}

'Alpha'属性是把一个目标元素与背景混合。设计者可以指定数值来控制混合的程度。这种“与背景混合”通俗地说

就是一个元素的透明度。通过指定坐标，可以指定点、线、面的透明度。他们的参数含义分别如下：

“opacity'代表透明度水准。默认的范围是从0 到 100，他们其实是百分比的形式。也就是说，0代表完全透明，

100代表完全不透明。”finishopacity'是一个可选参数，如果想要设置渐变的透明效果，就可以使用他们来指定

结束时的透明度。范围也是0 到 100。“style' 参数指定了透明区域的形状特征。其中0代表统一形状、1代表线形、2代表放射状

、3代表长方形。”STARTX“和”STARTY“代表渐变透明效果的开始X和Y坐标。”

FINISHX“和”FINISHY“代表渐变透明效果结束X和Y 的坐标。

[复制代码](javascript:void(0);)

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">  
<html xmlns="http://www.w3.org/1999/xhtml">  
<head>  
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />  
<title>css滤镜</title>  
<style>  
body{FILTER:ALPHA(opacity=50,finishopacity=80,style=2}  
</style>  
</head>  
<body>  
<div class="try"><font color="#CC33C2">施杨de编程世界</font></div>  
<div><img src="image/ch2.gif" /></div>  
</body>  
</html>

[复制代码](javascript:void(0);)

2、Blur 滤镜  
语法：对于HTML：{filter:blur(add=add,direction=direction,strength=strength)}   
对于Script语言： [oblurfilter=] object.filters.blur   
用手指在一幅尚未干透的油画上迅速划过时，画面就会变得模糊。”Blur'就是产生同样的模糊效果。

“ADD”参数是一个布尔判断“TRUE（默认）”或者“FALSE”。它指定图片是否被改变成印象派的模糊

效果。模糊效果是按顺时针的方向进行的，“DIRECTION”参数用来设置模糊的方向。其中0度代表垂直

向上，然后每45度为一个单位。它的默认值是向左的270度。“STRENGTH“值只能使用整数来指定，

她代表有多少像素的宽度将受到模糊影响。默认是5个。

代码如下：

[复制代码](javascript:void(0);)

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">  
<html xmlns="http://www.w3.org/1999/xhtml">  
<head>  
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />  
<title>css滤镜</title>  
<style>  
body{filter:blur(add=ture,direction=135,strength=200)}  
</style>  
</head>  
<body>  
<div class="try"><font color="#CC33C2">施杨de编程世界</font></div>  
<div><img src="image/ch2.gif" /></div>  
</body>  
</html>

[复制代码](javascript:void(0);)

效果：



3、FlipH, FlipV 滤镜   
语法：{filter:filph} ,{filter:filpv} 分别是水平反转和垂直反转  
  
4、Chroma 滤镜   
语法：{filter:chroma(color=color)}   
使用”Chroma'属性可以设置一个对象中指定的颜色为透明色，参数COLOR即要透明的颜色。  
代码如下：

[复制代码](javascript:void(0);)

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">  
<html xmlns="http://www.w3.org/1999/xhtml">  
<head>  
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />  
<title>css滤镜</title>  
<style>  
body{filter:chroma(color=#CC33C2)}  
</style>  
</head>  
<body>  
<div class="try"><font color="#CC33C2">施杨de编程世界</font></div>  
<div><img src="image/ch2.gif" /></div>  
</body>  
</html>

[复制代码](javascript:void(0);)

可以屏蔽颜色，大有用处。

5、DropShadow 滤镜   
语法：{filter:dropshadow(color=color,offx=ofx,offy=offy,positive=positive)}

“DropShaow'顾名思义就是添加对象的阴影效果。其工作原理是建立一个偏移量，加上较深。

'Color'代表投射阴影的颜色，'offx'和'offy'分别是X方向和Y方向阴影的饿偏移量。

'Positive'参数是一个布尔值，如果为“TRUE（非0）”，那么就为任何的非透明像素建立可见的投影。

如果为“FASLE（0）”，那么就为透明的像素部分建立透明效果 代码如下：

dropshadow(color=gray,offx=5,offy=5.positive=0)

效果：



6、Glow 滤镜   
语法：{filter:glow(color=color,strength)}

当对一个对象使用'glow'属性后，这个对象的边缘就会产生类似发光的效果。“COLOR”是指定

发光的颜色，“STRENGTH”则是强度的表现，可以从1到255之间的任何整数来指定这个力度。

filter:glow(color=red,strength=10) 后的效果   
  
filter:glow(color=#ffff00,strength=5) 后的效果

[复制代码](javascript:void(0);)

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">  
<html xmlns="http://www.w3.org/1999/xhtml">  
<head>  
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />  
<title>css滤镜</title>  
<style>  
body{filter:glow(color=red,strength=10)}  
.try{ margin-left:30px;}  
</style>  
</head>  
<body>  
<br /><br />  
<div class="try"><font color="#CC33C2">施杨de编程世界</font><br />  
<br />  
<div><img src="image/ch2.gif" /></div><br /><br /></div>  
</body>  
</html>

[复制代码](javascript:void(0);)

7、Gray ,Invert,Xray 滤镜  
语法：{filter:gray} ,{filter:invert},{filter:xray}   
  
Gray滤镜是把一张图片变成灰度图；（汶川大地震后很多网站色调都变灰色，就是用的这一点代码实现的）  
Invert滤镜是把对象的可视化属性全部翻转，包括色彩、饱和度、和亮度值；  
Xray滤镜是让对象反映出它的轮廓并把这些轮廓加亮，也就是所谓的“X”光片。

效果：



效果：



8、Light 滤镜   
语法：Filter{light}   
  
这个属性模拟光源的投射效果。一旦为对象定义了“LIGHT'滤镜属性，那么就可以调用它的“方法(Method)'来设置或者改变属性。“LIGHT'可用的方法有：   
  
·AddAmbient 加入包围的光源   
·AddCone 加入锥形光源   
·AddPoint 加入点光源   
·Changcolor 改变光的颜色   
·Changstrength 改变光源的强度   
·Clear 清除所有的光源   
·MoveLight 移动光源  
  
addAmbient (iRed,iGreen,iBlue,iStrength) : 为滤镜添加环境光。环境光是无方向的，并且均匀的洒在页面的表面。环境光有颜色和强度值，可以为对象田家更多的颜色。它通  
  
常和其他光一起使用。无返回值。参数见下表。  
iRed : 必选项。整数值(Integer)。 指定红色值。取值范围为 0 - 255 。   
iGreen : 必选项。整数值(Integer)。 指定绿色值。取值范围为 0 - 255 。   
iBlue : 必选项。整数值(Integer)。 指定蓝色值。取值范围为 0 - 255 。   
iStrength : 必选项。整数值(Integer)。 指定光强度。取值范围为 0 - 100 。  
  
changeColor (iLightNumber,iRed,iGreen,iBlue,fAbsolute) : 改变光的颜色。无返回值。参数见下表。  
iLightNumber : 必选项。整数值(Integer)。 指定光的标识符。   
iRed : 必选项。整数值(Integer)。指定红色值。取值范围为 0 - 255 。   
iGreen : 必选项。整数值(Integer)。指定绿色值。取值范围为 0 - 255 。   
iBlue : 必选项。整数值(Integer)。指定蓝色值。取值范围为 0 - 255 。   
fAbsolute : 必选项。布尔值(Boolean)。指定改变是替换当前设置的绝对值，还是加到当前设置的相对值。此参数不等于零表示采用绝对值。否则表示采用相对值。  
  
  
可以定义光源的虚拟位置，以及通过调整X轴和Y轴的数值来控制光源焦点的位置，还可以调整光源的形式（点光源或者锥形光源）指定光源是否模糊边界、光源的颜色、亮度等属  
  
性。如果动态的设置光源，可能回产生一些意想不到的效果。  
  
实现图片自动变色效果

9、Mask 滤镜   
语法：{filter:mask(color=color)}   
  
使用'MASK'属性可以为对象建立一个覆盖于表面的膜，其效果就象戴者有色眼镜看物体一样。

10、Shadow 滤镜  
语法：{filter:shadow(color=color,direction=direction)}   
  
利用“Shadow”属性可以在指定的方向建立物体的投影，COLOR是投影色，DIRECTION是设置投影的方向。其中0度代表垂直向上，然后每45度为一个单位。它的默认值是向左的270  
  
度。   
  
filter:shadow(color=red,direction=225)   
filter:shadow(color=blue,direction=225)   
filter:shadow(color=gray,direction=225)

11、Wave 滤镜 语法：{filter:wave(add=add,freq=freq,lightstrength=strength,phase=phase,strength=strength)}   
  
'wave' 属性把对象按垂直的波形样式打乱。默认是“TRUE（非0）”，   
“ADD”表示是否要把对象按照波形样式打乱，   
“FREQ”是波纹的频率，也就是指定在对象上一共需要产生多少个完整的波纹，   
“LIGHTSTRENGTH”参数可以对于波纹增强光影的效果，范围0----100，   
“PHASE”参数用来设置正弦波的偏移量。   
“STRENGTH”代表振幅大小。



**8 appearance**

<http://www.cnblogs.com/nidilzhang/archive/2010/01/09/1642887.html>

# [使用X-UA-Compatible来设置IE浏览器兼容模式](http://www.cnblogs.com/nidilzhang/archive/2010/01/09/1642887.html)

文件兼容性用于定义让IE如何编译你的网页。此文件解释文件兼容性，如何指定你网站的文件兼容性模式以及如何判断一个网页该使用的文件模式。

**前言**

为了帮助确保你的网页在所有未来的IE版本都有一致的外观，IE8引入了文件兼容性。在IE6中引入一个增设的兼容性模式，文件兼容性使你能够在IE呈现你的网页时选择特定编译模式。  
新的IE为了确保网页在未来的版本中都有一支的外观，IE8引入了文件兼容性。当你引入一个增设的兼容性模式，  
此文章说明文件兼容性的必要性，列出现有版本IE能使用的文件兼容性模式并示范如何选择特定的兼容性模式。

**了解文件兼容性的必要性**

每个主要版本IE新增的功能都是为了让浏览器更容易使用、增加安全性及更支持业界标准。以这些作为IE的特色，其中一个风险就是旧版本网站无法正确的显示。

为了将这个风险降到最低，IE6允许网页开发人员选择IE编译和显示他们网页的方式。"Quirks mode"为预设，这会使页面以旧版本浏览器的视点显示，"Standards mode"(也称为"strict mode")特点是支持业界标准最为完善。然而要利用这个增强的支持功能，网页必须包含恰当的[**<!DOCTYPE>**](http://msdn.microsoft.com/en-us/library/ms533737%28VS.85%29.aspx)指令。

若一个网页没有包含**<!DOCTYPE>**指令，IE6会将它以quirks mode显示。若网页包含有效的**<!DOCTYPE>**指令但浏览器无法辨识，IE6会将它以IE6 standards mode显示。因为少数网站已经包含**<!DOCTYPE>**指令，兼容性模式的切换相当成功。这使网页开发人员能选择将他们的网页转移为standards mode的最佳时机。

随著时间经过，更多网站开始使用standards mode。它们也开始使用IE6的特性和功能来检测IE。举例来说，IE6不支持[universal selector](http://msdn.microsoft.com/en-us/library/aa770077%28VS.85%29.aspx)(即css之全局选择器 \* {})，一些网站便使用它来针对IE做特定的对应。

当 IE7增加了对全域选择器的支持，那些依赖IE6特点的网站便无法侦测出这个新版本的浏览器。因此那些针对IE的特定对应无法应用于IE7，造成这些网站便无法如他们预期的显示。由于**<!DOCTYPE>**只支持两种兼容性模式，受到影响的网站拥有者被迫更新他们的网站使其能支持IE7。

IE8 比之前的任何版本浏览器都更支持业界标准，因此针对旧版本浏览器设计的网页可能无法如预期般呈现。为了帮助减轻所有问题，IE8引入文件兼容性的概念，使你能选择你的网页设计要对应的特定IE版本。文件兼容性在IE8增加了一些新的模式，这些模式能告诉浏览器如何解析和编译一个网页。若你的网页无法在 ie8正确的显示，你可以更新你的网站使它支持最新的网页标准(优先选项)或在你的页面上新增一个[meta](http://msdn.microsoft.com/en-us/library/ms535853%28VS.85%29.aspx)元素用于告诉IE8如何依照旧版本浏览器编译你的页面。

这能让你选择将你的网站更新支持IE8新特点的时机。

**认识文件兼容性模式**

IE8支持几种文件兼容性模式，它们具有不同的特性并影响内容显示的方式。

•Emulate IE8 mode指示IE使用**<!DOCTYPE>**指令来决定如何编译内容。Standards mode指令会显示成IE8 Standards mode而quirks mode会显示成IE5 mode。不同于IE8 mode，Emulate IE8 mode重视**<!DOCTYPE>**指令。  
•Emulate IE7 mode指示IE使用**<!DOCTYPE>**指令来决定如何编译内容。Standards mode指令会显示成IE7 Standards mode而quirks mode会显示成IE5 mode。不同于IE7 mode，Emulate IE7 mode重视**<!DOCTYPE>**指令。**对于许多网页来说这是最推荐的兼容性模式。**  
•IE5 mode 编译内容如同IE7的quirks mode之显示状况，和IE5中显示的非常类似。  
•IE7 mode编译内容如同IE7的standards mode之显示状况，无论网页是否含有**<!DOCTYPE>**指令。  
•IE8 mode提供对业界标准的最高支持，包含 [W3C Cascading Style Sheets Level 2.1 Specification](http://www.w3.org/TR/CSS21/)和[W3C Selectors API](http://www.w3.org/tr/selectors-api)，并有限的支持 [W3C Cascading Style Sheets Level 3 Specification (Working Draft)](http://www.w3.org/TR/css3-roadmap/)。  
•Edge mode指示IE以目前可用的最高模式显示内容。当使用IE8时其等同于IE8 mode。若(假定)未来放出支持更高兼容性模式的IE，使用Edge mode的页面会使用该版本能支持的最高模式来显示内容。同样的那些页面在使用IE8浏览时仍会照常显示。

由于edge mode使用该IE版本所能支持的最高模式来显示所浏览的网页内容，建议仅使用于测试页及其他非商用页面。

**指定文件兼容性模式**

要为你的网页指定文件模式，需要在你的网页中使用**meta**元素放入X-UA-Compatible [http-equiv](http://msdn.microsoft.com/en-us/library/ms533876%28VS.85%29.aspx) 标头。以下是指定为Emulate IE7 mode 兼容性之范例。

<html>  
<head>  
<!-- Mimic Internet Explorer 7 -->  
<meta http-equiv="X-UA-Compatible" content="IE=EmulateIE7" />  
<title>My Web Page</title>  
</head>  
<body>  
<p>Content goes here.</p>  
</body>  
</html>

其内容随著指定的页面模式而更改，当要模拟IE7时，指定**IE=EmulateIE7**，指定**IE=5**, **IE=7**, 或**IE=8**来选择其中一种兼容性模式。你也可以指定**IE=edge**来指示IE8使用它支持的最高模式。

X-UA-compatible标头没有大小写之分。然而除了[title](http://msdn.microsoft.com/en-us/library/ms535910%28VS.85%29.aspx)元素及其他的**meta**元素之外，它必须出现在网页[header节](http://msdn.microsoft.com/en-us/library/ms535252%28VS.85%29.aspx)其它元素之前的位置，

**设定网站服务器以指定预设兼容性模式**

网站管理员可籍着为网站定义一个自订标头来为他们的网站预设一个特定的文件兼容性模式。这个特定的方法取决于你的网站服务器。举例来说，下列的web.config文件使Microsoft Internet Information Services (IIS)能定义一个自订标头以自动使用IE7 mode来编译所有网页。

<?xml version="1.0" encoding="utf-8"?>  
<configuration>  
<system.webServer>  
<httpProtocol>  
<customHeaders>  
<clear />  
<add name="X-UA-Compatible" value="IE=EmulateIE7" />  
</customHeaders>  
</httpProtocol>  
</system.webServer>  
</configuration>

若你已于网站服务器指定了一个预设的文件兼容性模式，你可以在个别页面上指定不同的文件兼容性模式来盖过它。在网页中指定的模式优先权高于服务器中所指定的模式。

请查阅你的网站服务器关于指定自订标头的资讯，或看更多资料:

[Implementing the META Switch on Apache](http://msdn.microsoft.com/en-us/library/cc817573.aspx)[Implementing the META Switch on IIS](http://msdn.microsoft.com/en-us/library/cc817572.aspx)

**判定文件兼容性模式**

要判定网页使用IE8浏览时的文件兼容性模式，使用[document](http://msdn.microsoft.com/en-us/library/ms531073%28VS.85%29.aspx) object(文档对象)的[documentMode](http://msdn.microsoft.com/en-us/library/cc196988%28VS.85%29.aspx)功能。例如在IE8的网址列输入下列程式码会显示目前页面的文件模式。

javascript:alert(document.documentMode);

**documentMode**功能会回传一个数值对应目前页面的文件兼容性模式，举例来说，若网页指定为支持IE8模式，**documentMode**便会回传值"**8**"。

在IE6引入的[compatMode](http://msdn.microsoft.com/en-us/library/ms533687%28VS.85%29.aspx)功能不支持在IE8引入的**documentMode**功能。目前使用**compatMode**建立的应用程式还能在IE8中作用，但它们必须更新为使用documentMode。

若你希望使用JavaScript判定一个文件的兼容性模式，引入下面范例的这段程式码可支持旧版本的IE。

engine = null;  
if (window.navigator.appName == "Microsoft Internet Explorer")  
{  
// This is an IE browser. What mode is the engine in?  
if (document.documentMode) // IE8  
engine = document.documentMode;  
else // IE 5-7  
{  
engine = 5; // Assume quirks mode unless proven otherwise  
if (document.compatMode)  
{  
if (document.compatMode == "CSS1Compat")  
engine = 7; // standards mode  
}  
}  
// the engine variable now contains the document compatibility mode.  
}

**认识内容属性值**

内容属性值在接收到异于先前叙述的数值时是具有弹性的。这能使你对于IE如何显示你的网页更有操控性。举例来说，你可以设定内容属性值为IE=7.5。当你这样做的时候，IE尝试将这个值转换为[version vector](http://msdn.microsoft.com/en-us/library/ms537512%28VS.85%29.aspx)并选择最接近的结果。在这个例子中，IE会将其设定为IE7 mode。下面的范例显示该模式设定为其他值的状况。

<meta http-equiv="X-UA-Compatible" content="IE=4"> <!-- IE5 mode -->  
<meta http-equiv="X-UA-Compatible" content="IE=7.5"> <!-- IE7 mode -->  
<meta http-equiv="X-UA-Compatible" content="IE=100"> <!-- IE8 mode -->  
<meta http-equiv="X-UA-Compatible" content="IE=a"> <!-- IE5 mode -->   
  
<!-- This header mimics Internet Explorer 7 and uses  
<!DOCTYPE> to determine how to display the Web page -->  
<meta http-equiv="X-UA-Compatible" content="IE=EmulateIE7">

**注意:** 前面的范例显示单独的内容值。实际上IE只会执行网页中第一个X-UA-Compatible标头。

你也可以使用内容属性来指定复数的文件兼容性模式，这能帮助确保你的网页在未来的浏览器版本都能一致的显示。欲设定复数的文件模式，请设定内容属性以判别你想使用的模式。使用分号来分开各个模式。

如果一个特定版本的IE支持所要求的兼容性模式多于一种，将採用列于标头内容属性中最高的可用模式。你可以使用这个特性来排除特定的兼容性模式，虽然并不推荐这样做。举例来说，下列标头即会排除IE7 mode。

<meta http-equiv="X-UA-Compatible" content="IE=5; IE=8" />

**结论**

兼容性对于网页设计师来说是非常重要的顾虑。虽然最好是可以建立一个完全不需依赖任何网页浏览器特性或功能的网站，有时候这是不可能实现的。文件兼容性模式便能将网页限制在某个特定版本的IE中。

使用X-UA-Compatible标头来指定你的页面支持的IE版本。使用**document.documentMode**判定页面的兼容性模式。

选择支持某个特定版本的IE，你可以确保你的页面在未来的浏览器版本中也能有显示的一致性。

**9 user-select**

示例：<https://dev.windows.com/en-us/microsoft-edge/testdrive/demos/userselect/>

**非标准**  
该特性是非标准的，请尽量不要在生产环境中使用它！

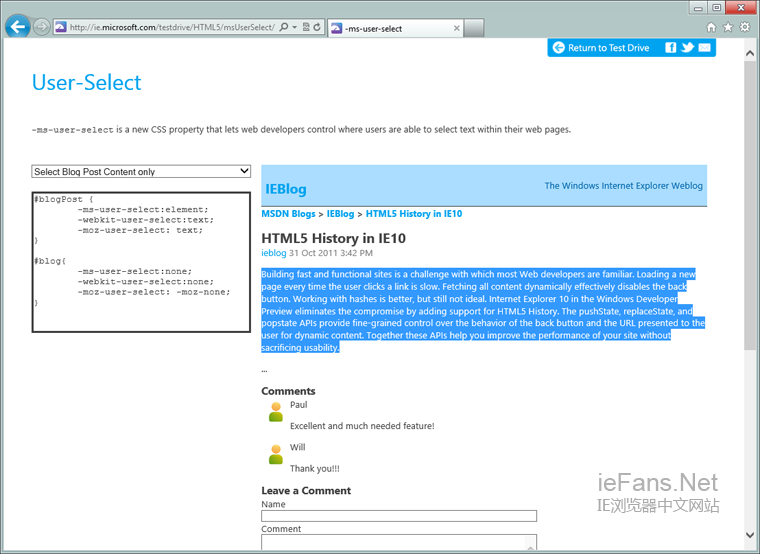
[IE10平台预览第四版](http://www.iefans.net/ie10-yulanban-pp4-xiazai/)中包含了对 CSS3 新属性 -ms-user-select 的支持，Web 开发人员可以利用这一新属性轻松精确的控制用户可以在网站上选择哪些文本。

以较为典型的新闻网站为例，大多数页面中会包含一条正文报道，有用户习惯在浏览新闻报道的时候边阅读边用鼠标选中内容，或者选中正文内容与他人进行分享。

与此同时，新闻网站页面上还会包含一些其他内容和链接，用户可能不需要选取这些内容。利用 -ms-user-select 属性，Web 开发人员可以指定对内容正文开启文本选择功能，而禁用选择其他内容。

<>

IE Test Drive 网站上提供了对此项功能的[演示示例](http://ie.microsoft.com/testdrive/HTML5/msUserSelect/)：

[](http://www.iefans.net/wp-content/uploads/2012/01/24ef_20120111-us-image1.png)

对整个页面设置 -ms-user-select:none，然后对包含博客正文的元素设置 -ms-user-select:element，这样就可以仅允许选择博客正文的内容。-ms-user-select:element 是 IE 首次引入的一个新属性，我们认为这一属性在许多情况下将非常实用。设置 -ms-user-select:element 表示用户可以选择该元素的文本，但选择范围受该元素的边界的约束。希望选择正文内容的用户可能不希望选择紧跟在正文后面的页脚元素。通过设置 -ms-user-select:element，用户可以轻松地选择正文内容，而不必担心鼠标拉动位置过大选取了多余的内容。

Formal syntax: none | text | all | element

(-prefix-)user-select: none;

(-prefix-)user-select: text;

(-prefix-)user-select: all;

(-prefix-)user-select: element;

-ms-user-select 可设置 4 个值：

* text – 可以选择文本
* element – 可以选择文本，但选择范围受元素边界的约束
* none – 不可以选择文本
* auto – 如果该元素包含可编辑的文本（如输入元素或可编辑内容的元素），则可以选择文本。否则，元素内容是否可选择由父节点的值决定。

-ms-user-select 的默认值为 auto。

通过将 -ms-user-select 设置为 none，开发人员可以禁用文本选择功能。在IE浏览器中，如果将文本设置为 -ms-user-select:none，则用户将无法在该文本块中开始选择文本。不过，如果用户在页面的其他区域开始选择文本，则用户可以在页面的任何区域（包括 -ms-user-select 设置为 none 的区域）继续选择文本。在 Firefox 中，如果开发人员设置 –moz-user-select:none，则用户无法在该区域中开始选择文本，也无法在任何其他文本选择操作中包括该区域的文本。在 Webkit 中，设置 –webkit-user-select:none 后，看起来该文本似乎并未包含在选择范围内，但是，当您复制并粘贴内容时，您会看到实际上已包含了该内容。

user-select 最初是在 [CSS3 用户界面模块](http://www.w3.org/TR/2000/WD-css3-userint-20000216)中提出的，之后，该模块被[CSS3 基本用户界面模块](http://www.w3.org/TR/css3-ui/)所取代，但是后者并未定义该属性。[Mozilla](https://developer.mozilla.org/en/CSS/-moz-user-select) 和 [Webkit](http://developer.apple.com/library/safari/#documentation/AppleApplications/Reference/SafariCSSRef/Articles/StandardCSSProperties.html#//apple_ref/css/property/-webkit-user-select) 均支持带自定义前缀的属性版本。不过，如上所述，它们在属性实现方面略有差异。

## Examples

.unselectable {-moz-user-select: none;-webkit-user-select: none;-ms-user-select: none;}

<body><p class="unselectable">The user is not able to select this text in Firefox, Chrome, Safari and IE.</p></body>

## 浏览器兼容性

* **Desktop**

* Mobile

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Feature | Chrome | Firefox (Gecko) | Internet Explorer | Opera | Safari (WebKit) |
| Basic support | 1[-webkit](https://developer.mozilla.org/zh-CN/docs/Web/Guide/Prefixes) | (Yes)[-moz](https://developer.mozilla.org/zh-CN/docs/Web/Guide/Prefixes) | 10[-ms](https://developer.mozilla.org/zh-CN/docs/Web/Guide/Prefixes) | - | (Yes)[-webkit](https://developer.mozilla.org/zh-CN/docs/Web/Guide/Prefixes) |
| element | 未实现 | (Yes) | (Yes) |  | 未实现 |

**10 background-size**

<http://www.w3cplus.com/content/css3-background-size>

[**Background**](http://www.w3.org/TR/css3-background/)大家一定不会陌生，但是[ＣＳＳ3](http://www.w3.org/TR/CSS/#css3)开始，给这个[**Background**](http://www.w3.org/TR/css3-background/)属性增加了几个新的属性值 ：[**Background-size**](http://www.w3.org/TR/css3-background/#the-background-size),[**Background-clip**](http://www.w3.org/TR/css3-background/#the-background-clip)和[**Background-origin**](http://www.w3.org/TR/css3-background/#background-origin)。那么从今天开始分几节内容分别来介绍一下[**background-size**](http://www.w3.org/TR/css3-background/#the-background-size),[**background-clip**](http://www.w3.org/TR/css3-background/#the-background-clip),[background-origin](http://www.w3.org/TR/css3-background/#background-origin)等新属性。今天的主题是[**Background-size**](http://www.w3.org/TR/css3-background/#the-background-size)，在开始介绍[Backgrond-size](http://www.w3.org/TR/css3-background/#the-background-size)之前，大家一起来回忆一下[CSS2](http://www.w3.org/TR/CSS/#css2)中[**Background**](http://www.w3.org/TR/css3-background/)中带有的一些属性：

background: [<background-color>][,<background-image>][,<background-repeat>][,<background-attachment>][,<background-position>]

background我们可以把其属性值串写在一起，同时也可以一个一个分写出来，下面简简单重温一下各属性的取值情况：

**一、设置背景色：background-color**

background-color: transparent || <color>

background-color其主要用来设置元素的背景颜色，其默认值为transparent（不设置任何颜色情况下是透明色），<color>用来设置背景色彩，常用的颜色格式为：颜色名：如“red”；rgb色:如rgb(255, 0, 0)或rgb(100%, 0%, 0%)；hls值hsl(0, 100%, 50%)，二进制值 #FF0000。在支持CSS3的浏览器中还可以使用rgba值rgba(255,0,0,1)。

**二、设置背景图片:background-image**

background-image: none || <url>

background-image其主要用来设置元素的背景图片，默认值为none，<url>是指背景图片的地址，这个地址可以是相对地址，也可以是绝对地址。

**三、设置背景图片重复：禾王background-repeat**

background-repeat: repeat || repeat-x || repeat-y || no-repeat

background-repeat是用来设置background-image在元素中的铺放格式的，其默认为repeat，也就是背景图片沿元素的X轴和Ｙ轴同时平铺，另外几个值分别表示repeat-x：背景图片仅沿Ｘ轴水平平铺，repeat-y背景图片仅沿Ｙ轴平铺，no-repeat表示背景图片不做任何铺放格式设置。

**四、设置背景图像是否固定或者随着页面的其余部分滚动:background-attachment**

background-attachment: scroll || fixed

background-attachment主要是用来设置背景图像是否固定或者随着页面的其余部分滚动，，其默认值为scroll，表示背景图片会随滚动条一起滚动，而取值fixed时，背景图片固定不动。

**五、设置背景图片的位置background-position**

background-position: <percentage> || <length> || [left|center|right][,top|center|bottom]

background-position主要是用为设置背景图片的位置，其默认值为(0,0)||(0%,0%)||(left top),可以进行具体的百分数或数值设置，也可以使用left,center,top,right,top,bottom配合设置，而其中以下几种表示相同定位方式：

"top left","left top"和"0% 0%","0,0"代表元素的左上角；

"top","top center","center top"和"50% 0"表示在元素顶边居中位置；

"right top","top right"和"100% 0"代元素的是元素的右上角位置；

"left","left center","center left"和"0% 50%"表示在元素左边中间位置；

"center","center center"和"50% 50%"表示在元素中间位置；

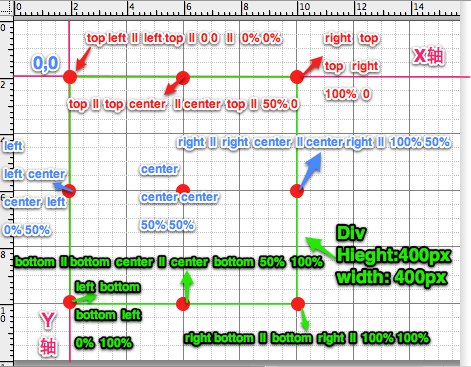
"right","right center","center,right"和"100% 50%"表示在元素右边中间位置；

"bottom left","left bottom"和"0% 100%"表示在元素的左下角；

"bottom","bottom center","center bottom"和"50% 100%"表示在元素的底下中间点位置；

"bottom right"，"right bottom"和“100％ 100％”表示在元素右下角位置

大家可以仔细对照下图：



上面简单的介绍了一下background的几个常用属性值的使用方法，那现在开始就进入到今天的主题：background-size。我们一起来学习background-size的属性，别的先不说了，我们也先从其语法开始入手：

background-size: auto || <length> || <percentage> || cover || contain

**取值说明：**

1、auto:此值为默认值，保持背景图片的原始高度和宽度；

2、<length>此值设置具体的值，可以改变背景图片的大小；

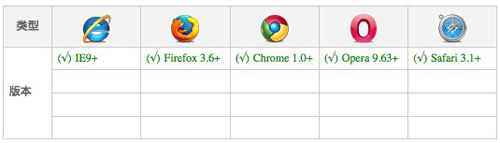
3、<percentage>此值为百分值，可以是0％〜100％之间任何值，但此值只能应用在块元素上，所设置百分值将使用背景图片大小根据所在元素的宽度的百分比来计算。

4、cover：此值是将图片放大，以适合铺满整个容器，这个主要运用在，当图片小于容器时，又无法使用background-repeat来实现时，我们就可以采用cover;将背景图片放大到适合容器的大小，但这种方法会使用背景图片失真；

5、contain:此值刚好与cover相反，其主要是将背景图片缩小，以适合铺满整个容器，这个主要运用在，当背景图片大于元素容器时，而又需要将背景图片全部显示出来，此时我们就可以使用contain将图片缩小到适合容器大小为止，这种方法同样会使用图片失真。

当background-size取值为<length>和<percentage>时可以设置两个值，也可以设置一个值，当只取一个值时，第二个值相当于auto，但这里的auto并不会使背景图片的高度保持自己原始高度，而会与第一个值相同。

**兼容的浏览器**



background-size和其他的一些CSS3属性一样，需要加上自己的别名，

*/\*Ｍozilla\*/*

-moz-background-size: auto || <length> || <percentage> || cover || contain

/\*Webkit\*/

-webkit-background-size: auto || <length> || <percentage> || cover || contain

/\*Presto\*/

-o-background-size: auto || <length> || <percentage> || cover || contain

/\*W3c标准\*/

background-size: auto || <length> || <percentage> || cover || contain

上面主要针对性介绍了background-size的理论知识，下面我们分别来看看其在各种值怕显示效果：

为了说明效效果，先给出一个ＤＥＭＯ结构和一张背景图：

**HTML Code:**

<divclass="demo"></div>

先初步为其加上一些效果：

.demo{

background:url("../images/background-image.jpg") no-repeat;width:200px;height:100px;line-height:100px;border:1px solid #ccc;font-size:14px;text-align: center;}

随便找张图片(50px\*50px)来当作背景图片使用：



**DEMO一、background-size:auto;**

我来看第一个DEMO,在前面的DEMO上加上和个class名为"backgroundSizeAuto",在这个Demo上我们应用上前面所示的背景，并把background-size取值为auto,

.backgroundSizeAuto{

-moz-background-size: auto;-webkit-background-size: auto;-o-background-size: auto;background-size: auto;}

**效果：**



效果中明显得知，这张背景图完全没有作出任何变化，前面也说过，auto值就是使用背景图保持原样，不做任何参数修改，这个效果就相当于没加background-size效果一样。

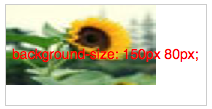
**Demo二、background-size:<length>:**

在这个DEMO中，给background-size一个具体的值，比如说这里取值为150px 80px;如：

.backgroundLength{

-moz-background-size:150px 80px;-webkit-background-size:150px 80px;-o-background-size:150px 80px;background-size:150px 80px;}

**效果：**



从效果图中告诉我们，此时背景图片由当时的50px\*50px变成了150px\*80px,此时背景图片也变形失真了。上面是取值为两个值时，第一个值重新变成了背景图片的长度值，而第二个值也相应变成了背景图片的高度值；但我们如果职取一个值时，那么此时的值将同时定义给背景图片的宽和高，相当于background-size:150px auto.另一个其只能了两个值，决不会有第三个值出现，如果有第三个值出现，你就用错了，将无任何效果显示。

**DEMO三、background-size:<percentage>:**

现在将ＤＥＭＯ的值设置成百分数，我这里取了一个80% 50%:

.backgroundPer{

-moz-background-size:80% 50%;-webkit-background-size:80% 50%;-o-background-size:80% 50%;background-size:80% 50%;}

**效果：**



同样效果告诉我们，图片会改变大小，但这里有一点需要特别注意，图片大小不是按背景图片大小的百分数来计算，而是装截背景图的元素百分比计算，就拿我们这个DEMO来说，div.demo的大小是200px的宽100px的高，此时的背景图片大小将根据div.demo的宽和高的百分比计算，这样一来，背景图片的宽就是200px\*80%等 于160px;而背景图片的高同样是100px\*50%等于50px；如果有内边距的话还需要加上padding的值一起计算。另外当其只取一个值时，那么宽度和高度将相同，相当于background-size: 80％ auto。

ＤＥＭＯ二和DEMO三分别是以固定值和百分比来改变background-image的尺寸，同进还可以两种取值一起使用，如：

.backgroundLengthPre{

-moz-background-size:150px 80%;-webkit-background-size:150px 80%;-o-background-size:150px 80%;background-size:150px 80%;}

**效果：**

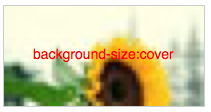


**DEMO四、background-size: cover;**

.backgroundCover{

-moz-background-size: cover;-webkit-background-size: cover;-o-background-size: cover;background-size: cover;}

**效果：**



效果图再次证明前面所讲的，background-size取值为cover时，背景图片自己会放大到适合容器的尺寸，在这里我需提示的是，在ＩＥ中有一个滤镜是类似于cover的功能，

filter:progid:DXImageTransform.Microsoft.AlphaImageLoader(src=’../images/background-image.jpg’, sizingMethod=’scale’);

-ms-filter: “progid:DXImageTransform.Microsoft.AlphaImageLoader(src=’../images/background-image.jpg’, sizingMethod=’scale’)”;

如果使用滤镜的话，background-size:cover；只有在IE6中不支持了。

**DEMO五、background-size:contain;**

ＤＥＭＯ三中的cover是把背景图片放大到适合元素容器的尺寸，这时的contain刚好是跟cover相反，是把背景图片缩小到适合元素容器的尺寸，如：

.backgroundSizeContain{

width:30px;height:30px;-moz-background-size: contain;-webkit-background-size: contain;-o-background-size: contain;background-size: contain;}

**效果：**

C:\Users\lenoo\AppData\Local\YNote\data\lisande-youxiang@163.com\f3e0ac2db36c4a038754227ddb54b881\size-contain.png

前面的DEMO中大家都知道，例子中使用的背景图片是50px\*50px;而现在将元素改成了30px\*30px；效果中可以看到背景图片此时的尺寸也相应的缩小适合了元素容器的尺寸。

从上面的几个DEMO效果可以看出，只有当background-size值为auto时，背景图片才不会变形失真，而其他值都会造成背景图片变形夫真，所以大家使用时需要仔细考虑好，以免给你的制作带来不良效果。

background-size的cover的使用可以让我们解决在不同分辨大小的浏览器下使用一张背景图片，他也常配合[W3C CSS3 Media Queries Module](http://www.w3.org/TR/css3-mediaqueries/)一起使用，将达到你意想不到的效果，也就是所说的，一张背景图在，不同分辨大小的浏览器下显示满屏，具体如何使用，大家先看一段下面的代码：

*/\*这段代码是当不支持CSS3 Media Queries浏览器按background-size:cover模式显示背景图片\*/*body{

background:#000url(myBackground\_1280x960.jpg) center center fixed no-repeat;-moz-background-size: cover;-webkit-background-size: cover;-o-background-size: cover;background-size: cover;}*/\*1024px X 768px \*/*@media only all and (max-width: 1024px) and (max-height: 768px) {

body{

-moz-background-size:1024px 768px;-webkit-background-size:1024px 768px;-o-background-size:1024px 768px;background-size:1024px 768px;}

}

*/\*640px X 480px\*/*@media only all and (max-width: 640px) and (max-height: 480px) {

body{

-moz-background-size:640px 480px;-webkit-background-size:640px 480px;-o-background-size:640px 480px;background-size:640px 480px;}

}

上例中后面两段代码分别实现了1024\*768和640\*480两种分辨背景图片显示满屏，加上最最前面那段代码是当，浏览器不支持[W3C CSS3 Media Queries Module](http://www.w3.org/TR/css3-mediaqueries/)时，背景图将按cover缩放形式显示。关于这方面更详细的应用大家可以到[alistapart.com](http://www.alistapart.com/articles/supersize-that-background-please/)和[Soak Ｂlog](http://www.soak.co.uk/blog/css3-scalable-background-images/)查阅。