

# CSS3

第1天课堂笔记（本课程共3天）

班级：北京前端训练营1期

讲师：邵山欢

日期：2016年5月13日

邵山欢老师

微博：@邵山欢

QQ : 179427026

E-mail : shaoshanhuan@163.com

爱前端官网 ：[www.iqianduan.cn](http://www.iqianduan.cn)

# 目录

[CSS3 1](#_Toc5018)

[目录 2](#_Toc20441)

[一、CSS3概述 3](#_Toc16755)

[1.1 历史 3](#_Toc23571)

[1.2 兼容性 3](#_Toc22169)

[二、选择器 5](#_Toc32200)

[2.1 关系选择符 5](#_Toc8742)

[2.2 属性选择器 7](#_Toc3260)

[2.3 儿子序选择器 9](#_Toc5529)

[三、伪类 11](#_Toc29205)

[3.1 :hover能够给所有元素使用了 11](#_Toc17565)

[3.2 :focus得到焦点时 11](#_Toc28034)

[3.3 :not反着选 11](#_Toc18441)

[3.4 :only-child 唯一的儿子 11](#_Toc21697)

[3.5 :empty 空标签 11](#_Toc17890)

[3.6 :checked 选中的 11](#_Toc10262)

[3.7 :disabled和:enabled 12](#_Toc11875)

[四、伪元素 13](#_Toc27568)

[4.1 ::before 、 ::after 13](#_Toc32348)

[4.2 ::selection 被选中的文字样式 14](#_Toc29973)

[4.3 ::first-letter 、::first-line 第一个文字和第一行 14](#_Toc6495)

[五、圆角边框 15](#_Toc5386)

[六、box-shadow和text-shadow 17](#_Toc8864)

[七、background系列 18](#_Toc2445)

[7.1 渐变 18](#_Toc12648)

[7.2 backgrond-origin和background-clip属性 19](#_Toc16610)

[7.3 background-size属性 20](#_Toc26250)

[7.4 多背景 22](#_Toc32190)

[八、过渡 23](#_Toc21068)

[8.1 过渡的基本语法 23](#_Toc11797)

[8.2 什么属性能过渡 23](#_Toc22157)

[8.3 什么时候触发过渡 23](#_Toc4381)

# 一、CSS3概述

## 1.1 历史

CSS3 是 层叠样式表(Cascading Style Sheets) 语言的最新进展，目的在于扩展 CSS2.1。 它为我们带来了许多期待已久的新特性， 例如圆角，阴影，gradients(渐变)，transitions(过渡) 或 animations(动画) ， 当然还有新的布局如 multi-columns ， flexible box 或 grid layouts。

CSS3是一个没有准确边界的一个概念，很难说某一个属性或者某一个选择器是不是CSS3的内容，还是CSS2.1内容。因为都是不同时期被浏览器厂商广泛采用，倒逼W3C把这个属性、选择器定为标准的。

CSS3为什么能发展这么快，百花齐放，源于CSS从创立之初就有一个规定，就是遇见不认识的选择器、属性，静默不报错，不影响后面能够识别的语句：

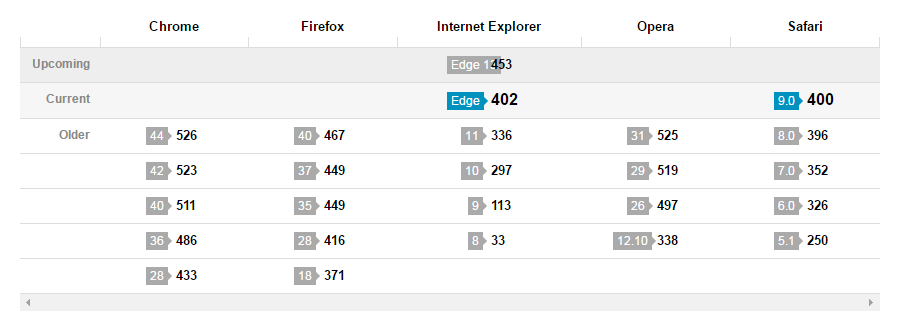
|  |
| --- |
| 1. <style type="text/css"> 2. shaoshanhuan{ → 这个选择器是错误的，但是不影响后面的正确选择器 3. width:4540px; 4. height: 4540px; 5. background-color: red; 6. } 7. div{ 8. adsffdg:435435px; → 这个属性是错的，但是不影响后面的正确属性 9. width: 200px; 10. dfgfdg:3243px; → 这个属性是错的，但是不影响后面的正确属性 11. height: 200px; 12. background-color: yellow; 13. } 14. </style> |

我们今天的课程：

选择器、圆角、阴影、背景、过渡transition。

## 1.2 兼容性

CSS3和HTML5一起的兼容性，有一个网站专门做了打分：[www.html5test.com](http://www.html5test.com)



CSS3现在主要用于移动端，因为移动端没有IE浏览器，PC端真需要再等两年才能完全使用。

哆啦A梦CSS3测试：

http://www1.pconline.com.cn/pcedu/specialtopic/css3-doraemon/

|  |  |  |
| --- | --- | --- |
| chrome | firefox | IE8 |

# 二、选择器

CSS2.1中，选择器7种：

id选择器 #box

类选择器 .red

标签选择器 p

后代 div p

交集 div.red

并集 div,p

通配符 \*

## 2.1 关系选择符

> 儿子，亲儿子，不是后代，必须是儿子

+ next sibling，下一个兄弟

~ next all siblings ，下所有兄弟

不要加空格，因为CSS中的空格有后代的意思。

>儿子选择器

|  |
| --- |
| 1. .box>p{ 2. color:red; 3. } |

IE7开始兼容

+ 下一个兄弟，next sibling

|  |
| --- |
| 1. h3+p{ 2. color:green; 3. } |

会变色的元素：

|  |
| --- |
| 1. <h3></h3> 2. <p>●</p> 3. <p>●</p> 4. <p>●</p> 5. <h3></h3> 6. <p>●</p> 7. <p>●</p> |

IE7开始兼容

~ 后面所有兄弟，next all siblings

|  |
| --- |
| 1. h3~p{ 2. color:red; 3. } |

HTML结构中能够变红的：

|  |
| --- |
| 1. <p>●</p> 2. <h3></h3> 3. **<p>●</p>** 4. **<p>●</p>** 5. **<p>●</p>** 6. <h3></h3> 7. <h4></h4> 8. **<p>●</p>** 9. **<p>●</p>** 10. <div> 11. <p>●</p> 12. </div> |

**jQuery中，借用了这几个符号，> 、 + 、 ~ 都能够用于$()函数中，都是IE6也兼容的。**

|  |
| --- |
| 1. $("div>p").animate({"width":656},454); |

IE6也能够兼容

|  |
| --- |
| 1. $("div>p") |

等价于

|  |
| --- |
| 1. $("div").children("p") |

|  |
| --- |
| 1. $("div+p") |

等价于

|  |
| --- |
| 1. $("div").next("p") |

|  |
| --- |
| 1. $("div~p") |

等价于

|  |
| --- |
| 1. $("div").nextAll("p") |

## 2.2 属性选择器

哲学上讲，CSS2.1层面，只能通过id属性、class属性选择元素。CSS3中，可以通过任意属性选择元素了。

选择器img标签中所有src属性为images/2.png的：

|  |
| --- |
| 1. **img[src="images/2.png"]**{ 2. border:10px solid red; 3. } |

方括号就是属性选择器，可以把任何HTML标签的属性当做选择器。

|  |
| --- |
| 1. img[kaola="你"]{ 2. transform:rotate(45deg); 3. } |

IE7开始兼容



哲学上讲：

|  |
| --- |
| 1. #box |

等价于

|  |
| --- |
| 1. [id="box"] |

引申了几个选择器：

以val开头的，IE7兼容

|  |
| --- |
| 1. p[class**^=**"para"]{ 2. color:red; 3. } |

以val结尾的，IE7兼容

|  |
| --- |
| 1. p[class**$=**"para"]{ 2. color:red; 3. } |

任何一个部分出现了val，IE7兼容

|  |
| --- |
| 1. p[class**\*=**"para"]{ 2. color:red; 3. } |

以val开头，并且紧跟一个-，IE7兼容

|  |
| --- |
| 1. p[class**|=**"para"]{ 2. color:red; 3. } |

等价于

|  |
| --- |
| 1. p[class**^=**"para-"]{ 2. color:red; 3. } |

val是一个独立的用空格分开的个体，选择，IE7兼容

|  |
| --- |
| 1. div[class~="haha"]{ 2. background-color: red; 3. } |

选择的是标红的元素：

|  |
| --- |
| 1. <div class="box haha">√</div> 2. <div class="xixi haha">√</div> 3. <div class="hahahehe"></div> 4. <div class="haha-xixi"></div> 5. <div class="haha">√</div> |

哲学上讲：

|  |
| --- |
| 1. .red |

等价于

|  |
| --- |
| 1. [class**~=**”red”] |

**jQuery中，$()函数中，能够书写上面所有的选择器，并且IE6兼容。比如：**

|  |
| --- |
| 1. $("input[value~=哈]").fadeOut(); |

选择所有“哈”是用空格隔开的独立部分的元素。

## 2.3 **儿子**序选择器

都是用伪类的符号“:”

只要一个p是某个元素的第一个儿子，那么选择它，IE7兼容

|  |
| --- |
| 1. p:first-child |

所以，和jQuery中$(“p:first”)不一样， 因为CSS3中的:first-child不吹哨排队。

jQuery： 爸爸去哪儿的比喻，村长李锐要吹哨，吹出所有的p，然后大排队，第一个p变红。

CSS3： 不吹哨大排队，只要你是家里的老大，你就被选择。

只要一个p是某个元素的最后一个儿子，那么选择它，IE9兼容

|  |
| --- |
| 1. p:last-child |

只要一个p是某个元素的第1个儿子，那么选择它，IE9兼容

|  |
| --- |
| 1. p:nth-child(1) |

下标从1开始。不能是0，不能是负数。

只要一个p是某一个元素的第8、11、14、17……个儿子，那么选择它，IE9兼容

|  |
| --- |
| 1. p:nth-child(3n+8) |

从第8个开始，每3个选一个

同样的，jQuery也支持上面的选择器，IE6兼容。

只要一个p是某个元素的倒数第3个儿子，那么选择它，IE9兼容

|  |
| --- |
| 1. p:nth-last-child(3) |

只要一个p是某个元素的倒数第8、11、14……个儿子，那么选择它，IE9兼容

|  |
| --- |
| 1. p:nth-last-child(3n+8) |

只要一个h3是某一个而元素的h3儿子中的第2个，那么选择它，IE9兼容

|  |
| --- |
| 1. h3:nth-of-type(2) |

比如下面的html结构，能够选择上的是：

|  |
| --- |
| 1. <p></p> 2. <p></p> 3. <p></p> 4. <p></p> 5. <h3>h3</h3> 6. <div> 7. <h3>h3</h3> 8. **<h3>h3</h3>** 9. <h3>h3</h3> 10. <h3>h3</h3> 11. </div> 12. **<h3>h3</h3>** |

只要一个元素有.haha类并且是同种标签的第2个儿子，那么选择它，IE9兼容

|  |
| --- |
| 1. .haha:nth-of-type(2) |

倒数：

|  |
| --- |
| 1. :nth-last-of-type(3) |

总结一下，都暗含了是某个人的第几个儿子：

|  |
| --- |
| 1. :first-child() 只有这个IE7兼容，一下统统IE9兼容 2. :last-child() 3. :nth-child() 4. :nth-of-type() 5. :nth-last-child() 6. :nth-last-of-type() 7. :first-of-type() 8. :last-of-type() |

一定要记住吹不吹哨这个事儿：

jQuery中

|  |
| --- |
| 1. $("p").eq(3) |

不等价：

|  |
| --- |
| 1. $("p:nth-child(4)") |

因为eq大排队，而:nth-child不吹哨大排队。CSS3的选择器在jQuery中选线支持，并且IE6兼容。十年磨一剑！！方便了我们的jQuery编程。

比如我们想让第5个开始后面所有个人都选中：

|  |
| --- |
| 1. p:nth-child(4)~p |

# 三、伪类

CSS2.1中，只能给a标签增加伪类。a:link、a:hover、a:visited、a:active。

## 3.1 :hover能够给所有元素使用了

:hover这个选择器，在CSS3中，能够给任何元素使用了。

IE7兼容各种元素:hover

## 3.2 :focus得到焦点时

当一个表单元素有焦点，那么就能够被:focus伪类选中。

|  |
| --- |
| 1. input:focus{ 2. /\*动画\*/ 3. animation:haha 1s ease 0s infinite alternate; 4. } |

## 3.3 :not反着选

没有.haha类的p，IE9兼容

|  |
| --- |
| 1. p:not(.haha) |

## 3.4 :only-child 唯一的儿子

如果一个p是某一个元素的唯一的儿子，那么选择它。IE9兼容。

|  |
| --- |
| 1. p:only-child{ 2. color:red; 3. } |

## 3.5 :empty 空标签

当一个标签是完全首尾相接，没有任何的空格、tab、换行、文字，就是:empty

|  |
| --- |
| 1. <div></div> → 满足div:empty 2. <div> </div> 3. <div> 5. </div> 6. <div>我</div> |

## 3.6 :checked 选中的

单选按钮、复选框，如果被勾选，那么能被伪类:checked选中

|  |
| --- |
| 1. input[type=”radio”]:checked |

## 3.7 :disabled和:enabled

IE9兼容

无效和有效的input控件。

|  |
| --- |
| 1. <input type="text" **disabled**/> 2. <input type="text" /> 3. <input type="button" **disabled** value="adsfdsaf"/> 4. <input type="button" value="adsfdsaf"/> |

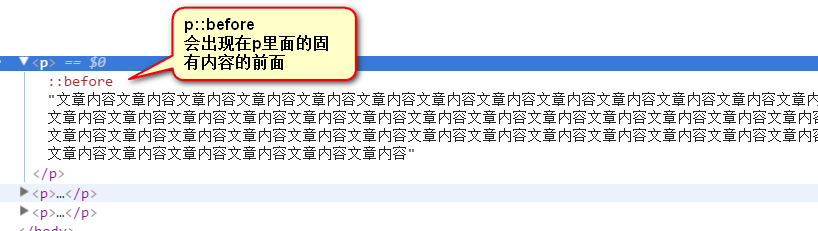
|  |
| --- |
| 1. input:disabled |

# 四、伪元素

## 4.1 ::before 、 ::after

两个冒号表示伪元素。

|  |
| --- |
| 1. p::before{ 2. **content:">>>>";** 3. } |

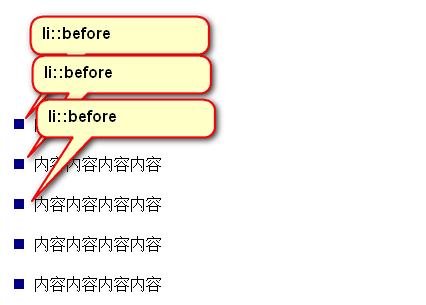


**content表示内容，所有的::before必须有content属性，否则语法错误。如果没有content，那么也要写content:””。**

**注意::before 、::after都是行内元素，必须转块或者脱标才能设置宽度高度。**

创建了假的元素，伪元素。

before的诞生，方便了很多：



after可以用来清除浮动。

|  |
| --- |
| 1. .cl::after{ 2. content:""; 3. display: block; 4. clear: both; 5. } |

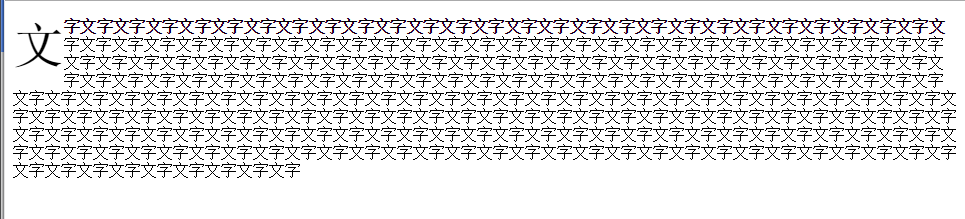
IE8不兼容::after，但是兼容:after。所以为了更大的兼容，写成:after即可。

## 4.2 ::selection 被选中的文字样式

|  |
| --- |
| 1. p::selection{ 2. background-color: orange; 3. color:white; 4. } |

IE9开始兼容

## 4.3 ::first-letter 、::first-line 第一个文字和第一行



IE9开始兼容

# 五、圆角边框

border-radius属性，值可以是一个px为单位的数值，也可以是%为单位的百分比数值。

IE9开始兼容

|  |
| --- |
| 1. div:nth-child(1){ 2. width: 200px; 3. height: 200px; 4. border: 1px solid #000; 5. border-radius: 20px; 6. } |

如果border-radius足够大，是宽度、高度的一半，就是正圆。

|  |
| --- |
| 1. div:nth-child(3){ 2. width: 600px; 3. height: 200px; 4. border: 1px solid #000; 5. **border-radius: 100px;** 6. } |

用px为单位的时候，四个角互不侵入：



拆开描述四个角落：

|  |
| --- |
| 1. border-radius: 20px 40px 60px 80px; |

分别是左上角、右上角、右下角、左下角

也可以是三个数值：

|  |
| --- |
| 1. border-radius: 20px 60px 80px; |

分别是：左上角、右上角和左下角、右下角

也可以通过小属性来层叠：

|  |
| --- |
| 1. border-top-left-radius: 60px; 2. border-top-right-radius: 60px; 3. border-bottom-right-radius: 100px; 4. border-bottom-left-radius:50px; |

用百分比来设置border-radius不一样，可以综合设置圆弧占了直边的百分之几。



|  |
| --- |
| 1. border-radius: 0% 100%; |



http://sc.chinaz.com/jiaobendemo.aspx?downloadid=2016111949321

<http://www.17sucai.com/pins/demoshow/11467>

# 六、box-shadow和text-shadow

标准的是5个参数，IE9开始兼容。

向右的偏移值、向下的偏移值、模糊半径、延展宽度、颜色。

|  |
| --- |
| 1. box-shadow: 2px 2px 2px 2px black; |

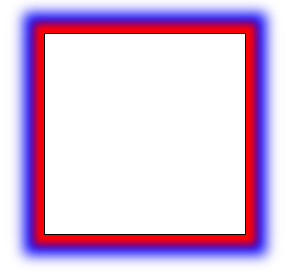
可以省略参数：

|  |
| --- |
| 1. box-shadow: 2px 2px 2px black; |

没有延展值。

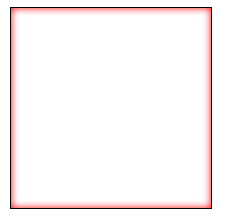
一个盒子可以有多个阴影，用逗号隔开：

|  |
| --- |
| 1. box-shadow: 0px 0px 12px 10px red , 0px 0px 20px 20px blue ; |



通过写inset，表示内阴影：

|  |
| --- |
| 1. box-shadow: inset 0px 0px 10px red; |



内阴影也可以有多个：

|  |
| --- |
| 1. box-shadow: inset 0px 0px 10px red, 2. inset 0px 0px 20px blue, 3. 0px 0px 10px green, 4. 0px 0px 20px orange; |

text-shadow是文字阴影，只有四个参数，少了延展值，IE10开始兼容。

|  |
| --- |
| 1. text-shadow: 1px 1px 1px red; |



# 七、background系列

background系列非常的闹腾

## 7.1 渐变

再也不用切1px然后repeat-x做渐变了。现在可以用CSS3制作渐变，IE9开始兼容。

线型渐变，linear-gradient

|  |
| --- |
| 1. background-image: -webkit-linear-gradient(top,red,blue); |

首先先说一下，一定是background-image属性，而不是background-color。

linear-gradient表示线型渐变。三个参数，top表示从上到下，罗列颜色值。

私有前缀：浏览器厂商把一些还处于实验性质的css属性，都加上了自己的前缀。

chrome的是-webkit-

火狐是-moz-

IE是-ms-

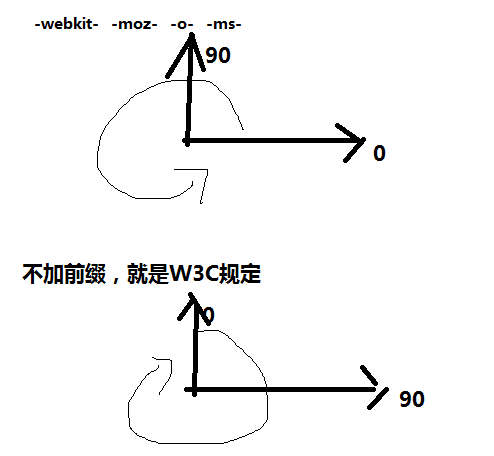
欧朋是-o-

所以一条属性要写很多个私有前缀的写法：

|  |
| --- |
| 1. background-image: **-webkit-**linear-gradient(top,red,blue); 2. background-image: **-moz-**linear-gradient(top,red,blue); 3. background-image: **-o-**linear-gradient(top,red,blue); 4. background-image: **-ms-**linear-gradient(top,red,blue); 5. background-image: linear-gradient(top,red,blue); |

第一个参数，可以写deg为单位的度数。deg就是英语degree的缩写，度数。一圈是360deg。

方向有分歧：



通常来说，写英语单词来描述渐变方向就行了，比如top、left、bottom、right。不要写deg。

可以罗列无数个颜色值，用逗号隔开。

|  |
| --- |
| 1. background-image: -webkit-linear-gradient(top,red,orange,yellow,green,blue,purple); |

可以用百分比描述这种颜色从何处开始：

|  |
| --- |
| 1. background-image: -webkit-linear-gradient(top,red,blue **30%**,green **70%**); |

径向渐变：

|  |
| --- |
| 1. background-image: -webkit-radial-gradient(100px 100px , blue , white); |

径向渐变还有一些参数，我们不多说了，自己看手册。

## 7.2 backgrond-origin和background-clip属性

背景起源属性：

|  |
| --- |
| 1. background-origin: content-box; |

|  |  |
| --- | --- |
| 原来的： | background-origin:content-box; |

content-box指的内容盒子，就是从内容盒子开始渲染背景。



background-clip表示背景裁切，如果值是content-box就表示padding区域没有背景，只有content区域有背景。

|  |
| --- |
| 1. background: url(images/0.jpg); 2. background-clip: content-box; 3. background-origin: content-box; |

上面的语句等价于：

|  |
| --- |
| 1. background: url(images/0.jpg) **content-box**; |

## 7.3 background-size属性

现在可以更改背景图片的尺寸了！啊哈哈哈哈。IE9开始兼容

先写宽度，然后是高度。

|  |
| --- |
| 1. background-size: 200px 200px; |

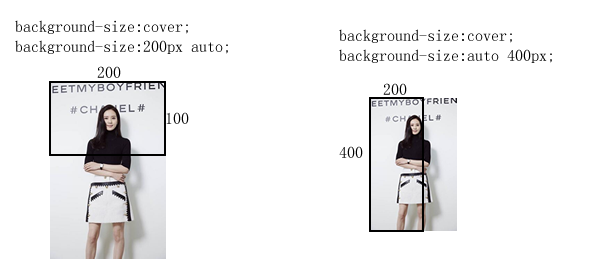


如果只想设置一个值，另外一个值自动按比例计算，那么就写auto：

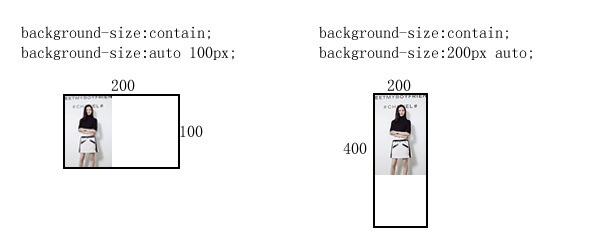
|  |
| --- |
| 1. background-size: 200px auto; |

|  |
| --- |
| 1. background-size: auto 200px; |

backgrond-size可以用cover来当做值，表示图片不变形，尽可能多的呈递图片。根据图片的宽高比不同，和盒子的宽高比不同，有两种区别。cover可能不能完整呈递图片：



值也可以是contain，表示图片不会变形，一定能呈递完整图片：



background-size用于精灵的时候，比较复杂，background-size是精灵图整体的尺寸。

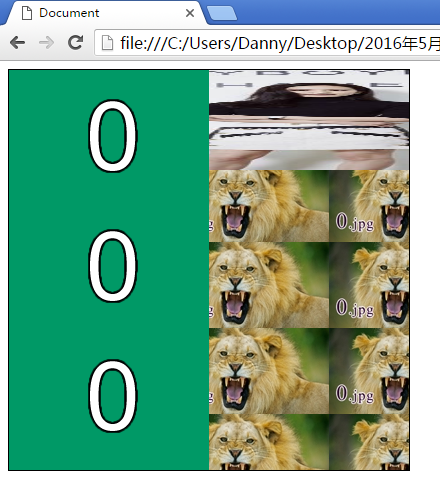
## 7.4 多背景

一个盒子可以同时携带多个背景了，用逗号隔开。IE9兼容。

但是，同一个盒子只能携带一个背景色，可以携带多个背景图。

|  |
| --- |
| 1. background: url(images/0.png) repeat-y, 2. url(images/shishi.jpg) no-repeat , 3. url(images/0.jpg); |

按照顺序，前面没有占满的空间，就可以给后面的图片使用。



background-position、background-size等等属性都能用逗号隔开数值了，表示对不同的图片来设置值。

|  |
| --- |
| 1. background-size: auto auto, 2. 600px 100px, 3. auto auto; |

# 八、过渡

transition属性，就是过渡。让某一个元素从CSS状态A变为CSS状态B的时候不是干蹦，而是动画。

制作动画，不需要使用setInterval了，可以使用CSS3的transition属性来制作动画。transition属性使用的是浏览器内核中的C++代码来实现的，效率要比setInterval高的多的多。动画的平滑程度、细腻程度，要比setInterval牛逼很多。

## 8.1 过渡的基本语法

下面就是一个完整的过渡语法：

|  |
| --- |
| 1. transition:all 1s ease 0s; |

第1个参数： 就是想让什么属性参与过渡，如果想让所有属性参与过渡，写all

第2个参数： 就是过渡动画的时间，s表示秒，一定要写s，不要不写单位，也不要些别的；

第3个参数： 缓冲描述词，ease表示不匀速，匀速就是linear。

第4个参数： 延迟时间，0s就是不延迟。

贝塞尔曲线可以从http://cubic-bezier.com/ 生成：

|  |
| --- |
| 1. transition:all 2s **cubic-bezier(.29,1.05,.78,.07)** 0s; |

## 8.2 什么属性能过渡

**基本所有属性都能过渡：**

background-color、background-position、background-image、border-radius、border、padding、font-size等等。

**不能过渡的：**

渐变色、float

尤其要注意的是，jQuery中的animate函数，不支持background-color、background-position。所以，如果想要制作背景颜色、背景定位的动画，必须使用CSS3。

## 8.3 什么时候触发过渡

**任何对元素的CSS改变都能触发过渡。**

不要认为只有:hover能够触发过渡。任何造成元素的改变，都会引发过渡。

比如元素的类名改变，能够触发过渡。

比如直接用JS来设置属性，也能够触发过渡。

transition就像护身符一样，任何人胆敢改我的css样式，一定是动画实现的。并且动画效率比setInterval还高。

用过渡来实现动画，早晚有一天要替代setInterval()原理。

明天很爽：过渡做点东西、变形、3D、滚滚屏、元素进出。