

## CSS3 box-shadow兼容loading效果兼IE10+ CSS Hack介绍

这篇文章发布于 2013年05月29日, 星期三, 22:31, 归类于 [CSS相关](#)。阅读 78418 次, 今日 8 次 [15 条评论](#)

by zhangxinxu from <http://www.zhangxinxu.com>

本文地址: <http://www.zhangxinxu.com/wordpress/?p=3357>

### 一、称为引言的部分

周一的时候, 见到条转发较多的微博, 是个页面链接, 页面中有4个充分发挥CSS3 `box-shadow` 潜力实现精湛的效果。您可以[点击这里前往查看](#)。

有4个demo4种效果哦!

比方说第三个的彩色转啊转的loading效果:



是不是帅得掉渣啊! 😎

然而这种效果的实现就像是在大街上要猴, 围观人挺多, 喝彩声也不少, 但是, 看完了, 大家各回各家、各找各妈了!

好比上面那个炫炫的酷酷的转转加载效果, 全部CSS实现的啊, 好赞的来! 然而, 仅少部分浏览器支持, 实际项目中压根用不了, 至少在PC上如此, 纯属华而不实的东西。于是, 看完了, 过瘾了, 页面关掉后, 就直接goodbye了, 估计这辈子不会再过来了。

难道就没有简单实用的效果吗, 最好可以向下兼容的, 然后实际项目也能实用的?

有! 本文就将展示如何完美向下兼容实现loading效果的, 顺便分享下我自己想出来的一个给IE10+浏览器做Hack的方法。

### 二、效果实现的主演们

对于IE10+浏览器, FireFox, Chrome, Opera浏览器(如无特殊说明, 均值最近版本)使用CSS3 `box-shadow` + `animation` 实现loading效果, 而IE6-IE9浏览器还是老方法, 使用gif loading效果的图片, 如下这个(32X32):



我是根据IE10下CSS生成效果基础上制作的(Photoshop制作, 如何制作可参考[“gif动画图片制作”](#)这篇前作), PSD源文件可[右键这里](#)下载。

简洁的HTML, 不违和的CSS, 成就我们在新技术上的尝试。

### 三、CSS3实现之box-shadow绘制

下图是我们想要的效果（FireFox浏览器中放大后的截图）：



8个点，有3个大小有规律变大，这样旋转的时候就有方向感和真实性。

8个点实际对应于 `box-shadow` 的8个投影，图片大小标准 `32px*32px`，考虑到点的八卦布局以及大小限制，因此，我们的容器本身只有 `3px*3px`，于是，有如下的实现代码：

```
.loading {
  width: 3px; height:3px;
  border-radius: 100%;
  box-shadow: 0 -10px 0 1px #333, /* 圆角 */
              10px 0px #333,      /* 上, 1px 扩展 */
              0 10px #333,        /* 右 */
              -10px 0 #333,       /* 左 */
              -7px -7px 0 .5px #333, /* 左上, 0.5px扩展 */
              7px -7px 0 1.5px #333, /* 右上, 1.5px扩展 */
              7px 7px #333,        /* 右下 */
              -7px 7px #333;       /* 左下 */
}
```

如果您的浏览器IE9+或Chrome一伙，同时不是在RSS中阅读本文，就会看到上面代码的效果，就是下面这个静止状态的loading圈圈效果：



CSS3 `box-shadow` 有四个值，分别表示“右偏移、下偏移、模糊大小、扩展大小”。因此，那3个稍大的圈圈就是利用了第4个参数——“扩展”来实现的。分别扩展 `0.5px`，`1px` 以及 `1.5px`。

以上就是CSS3 `box-shadow` 的绘制，相比一开始的炫炫的彩色效果，这个要通俗易懂的多，大家都能轻松把玩实现的效果，只要定好位置，一切都很简单。

## 四、CSS3实现之animation与无限旋转

一般开源的移动框架中，无限旋转的CSS都是使用 `spin` 作为类名以及动画关键字的，大家可以约定俗成的使用，无形中有利于前端大环境的建设。

CSS代码如下，目前在PC上，FireFox/IE10/Opera浏览器中animation动画都已经不需要私有前缀了（如果您不放心，可以自己加上），于是，如下CSS3代码：

```
.spin {
  animation: spin 1s steps(8) infinite;
}
@keyframes spin {
  from {transform: rotate(0deg);}
  to {transform: rotate(360deg);}
```

```
}
```

跟上面box-shadow生成的静态点点效果结合，如下，就有（如果你没有效果，请检查浏览器或[点击这里查看原文](#)）：



HTML代码为：

```
<div class="loading spin"></div>
```

## 五、高不成低不就的IE9浏览器

IE9浏览器支持CSS3 `box-shadow`，但是不支持CSS3 `animation`。于是，IE9浏览器下，虽有CSS生成的点点效果，但是却无法持续旋转。因此，考虑到兼容性，我们需要把IE9浏览器跟IE6-IE8归到一起去，sorry啦，谁让你不自动升级的😞！

于是，我们现在的处理就是：IE6~9使用图片（代码如下），IE10+以及其他现代浏览器使用CSS3生成，如何无缝拼接？

```
.loading {  
    width: 32px; height: 32px;  
    background: url(loading-css3.gif);  
}
```

## 六、IE10+浏览器的处理

平时处理类似渐进增强功能都是IE6-IE8是一伙，IE9+及其他浏览器是一伙，我都是使用 `:root ie9_oter{}` 实现，简单易懂，屡试不爽，但是，IE10+还是头一次遇到，怎么破？

网上有这么个Hack，说是针对IE10浏览器的：

```
@media screen and (-ms-high-contrast: active), (-ms-high-contrast: none) {  
    /* IE10的样式... */  
}
```

先不说适不适用这里的需求，单纯从代码量以及记忆难度上将，个人是畏惧了。敢问各位，上面的Hack选择器你可以一遍就记住吗？

因此，需要另寻它法，IE10较新，相关知识较少，那就自己动脑子想，一个小小兼容处理，有何难的！



思路

关键是要寻找IE9不支持，IE10支持的相关CSS样式或选择器，下面就是一些：

- `:valid/:invalid` IE9不支持，IE10以及其他浏览器支持
- `:required/:optional` IE9不支持，IE10以及其他浏览器支持
- `::before/::after` IE9不支持（IE10👇IE9支持😞），IE10以及其他浏览器支持
- `CSS Animations` IE9不支持，IE10以及其他浏览器支持😞
- `CSS Gradients` IE9不支持，IE10以及其他浏览器支持

等.....

本文兼容难点就是因为IE9不支持 `animation`，理论上，我们是可以借助 `animation` 本身实现IE9的向下兼容的，如何处理？在 `spin` 动画的 `from` 和 `to` 设置 `background:none`，以及放上 `box-shadow` 相关的CSS！类似：

```
@keyframes spin {
  from {transform: rotate(0deg); background:none; box-shaodw: ...}
  to {transform: rotate(360deg); background:none; box-shaodw: ...}
}
```

然而，这种处理有两大问题：

1. 破坏了 `spin` 的重用性；
2. 代码冗余；

因此，不考虑。

于是，我们可以从 `:valid/:invalid`，`:required/:optional`，`::before/::after` 以及 `CSS Gradients` 中寻找突破。

### 1. `:valid/:invalid`伪类实现

`:valid/:invalid` 伪类是针对表单元素的，合法还是不合法。相关tips如下：

- 适用于单复选框、单行多行文本框、下拉框等；
- `button` 类型按钮所有忽略；
- `submit` 类型按钮目前Firefox 21不支持，IE10,Opera,Chrome均支持；
- 表单元素如果设置了 `disabled` 或 `readonly`，伪类忽略；

综上所述，如果我们忽略FireFox浏览器，则可以借助 `submit` 类型按钮实现我们想要的效果，HTML为：

```
<input type="submit" class="loading spin" hidefocus="true" />
```

于是，配合结构如下的CSS我们就可以得到我们想要的效果了：

```
/* IE6-IE9, FireFox实现 */
.loading {
  width: 32px; height: 32px;
  background: url(loading-css3.gif);
  /* 按钮样式重置 */
  margin: 0; padding: 0; border: 0; outline: none;
  pointer-events: none;
}
/* IE10+, Chrome, Opera */
.loading:valid {
  background: none;
  width: 3px; height:3px;
  border-radius: 100%;
  box-shadow: ...
}
```

没有照顾到CSS支持，且用户比例也颇高的FireFox浏览器，显然不是最优方法。有人可能会提议使用单行 `input` 框或多行 `textarea` 域实现。理论上，这是可行的，但是，由于单行或多行文本框在 `focus` 的时候有光标，因此，在实现的时候要多些处理，HTML的和CSS部分都有，如下红色代码：

```
<input class="loading spin" onfocus="this.blur();" hidefocus="true">
```

```
.loading {  
    width: 32px; height: 32px;  
    background: url(loading-css3.gif);  
    /* 下面所有样式均为input框样式重置, 去光标 */  
    margin: 0; padding: 0; border: 0; outline: none;  
    line-height: 0; font-size: 0; text-indent: -1px;  
    cursor: default;  
    pointer-events: none;  
}
```

以上实现算是相当兼容的，您可以狠狠地点击[这里](#)：[input+:valid伪类兼容loading效果demo](#)。

由于下面有更佳的且非常类似的实现，因此，为了压轴，这里的效果截图就省略，换成一张搞笑图片，名为“韩国选手超霸气登场”，转换大脑，活跃气氛。😄



## 2. :required/:optional伪类实现

`:required/:optional` 伪类表示表单元素是选填呢还是必填，相关tips如下：

- 适用于单复选框、单行多行文本框、下拉框等；
- `input` 标签下，无论是 `button` 类型按钮还是 `submit` 类型按钮，Opera浏览器都忽略之；其他浏览器均可识别；
- `button` 标签下，无论是 `button` 类型按钮还是 `submit` 类型按钮，仅Chrome浏览器识别，其他浏览器均忽略
- 控件 `disabled` 后依然可识别；

相比 `:valid/:invalid` 伪类，虽然都有一种浏览器对按钮的支持有问题，但是，`:required/:optional` 伪类却没有 `disabled` 失效的问题。因此，我们就没有了所谓焦点获取、文本输入、光标闪烁等文本框问题都可以通过添加 `disabled` 避免。

因此，如下HTML与CSS代码就可以实现比较健壮的loading效果啦！

```
<input class="loading spin" disabled>
```

```
/* IE6-IE9实现 */  
.loading {  
    width: 32px; height: 32px;  
    background: url(loading-css3.gif);  
    /* input框样式重置 */  
    margin: 0; padding: 0; border: 0;  
}  
/* IE10+以及其他或骚或帅之流 */  
.loading:optional {  
    background: none;  
    width: 3px; height: 3px;
```

```
border-radius: 100%;
box-shadow: ...
}
```

您可以狠狠地点击这里: [\:optional伪类下渐进增强loading动画demo](#)

IE9以及IE10浏览器下的demo效果截图如下:



从上面截图, 我们似乎发现了一个问题——尺寸!

很好理解, IE6~IE9浏览器实际尺寸 32px\*32px, 而IE10+等浏览器实际尺寸 3px\*3px, 差别明显, 如何破?

1. 如果元素在固定容器中, 且水平垂直居中布局, 无需破, 效果一样;
2. 如果元素固定定位, absolute 化, 同时 .loading 中的 margin:0 改成 margin:-14.5px 0 0 -14.5px 即可位置兼容;
3. 自己看着办微调等;

难道没有尺寸也完全一样的方法吗? 有! 下面的伪元素就可以弥补。

### 3. ::before/::after伪元素登场

::before/::after 伪元素就是 :before/:after 的进阶版, IE8以及原生IE9支持后者, 却不认识前者。

上面提到了单标签实现尺寸不一的问题, 我们是不是可能这样处理, CSS3 box-shadow 生成的loading效果利用伪元素生成, 这样, 就可以保证外部容器宽度一致了! 但, ::before/::after 伪元素对于表单元素而言, 几乎就是克星, 因为表单元素都是大多都是单标签, 即使偶尔冒出个 textarea 也是酱油, button 由于兼容性问题, 这里实际上也无法大展拳脚。

怎么破?

循序渐进, 先使用最普通的 div 标签, 看下面的代码:

```
<div class="loading spin"></div>
```

😏 普通的 div 诶~

```
/* 所有浏览器实现 */
.loading {
    width: 32px; height: 32px;
    background: url(loading-css3.gif);
}
/* IE10+以及其他 */
.loading::after {
    content: '';
    width: 3px; height: 3px;
    margin: 14.5px 0 0 14.5px;
```

```
border-radius: 100%;
box-shadow: ... ;
position: absolute; /* 可缺省 */
}
```

上面代码实现有什么问题呢？——虽然尺寸一致了，但IE10+以及其他现代浏览器 `::after` 伪元素生成的转转转与背景图片转转转重复了，如下截图效果：



有demo，点击[这里](#)查看：[伪元素与原背景搞基demo](#)

于是，我们如果能够让IE10+浏览器无背景，IE10+浏览器使用 `::after` 伪元素就可以了，关键咋整呢？

#### 4. 元素Hack与兄弟选择器

CSS中有个兄弟选择器 `~`，在“[CSS radio/checkbox单复选框元素显隐技术](#)”一文中有展示，您可以观摩观摩。以及还有相邻兄弟选择器 `+`

同样的，既然，如 `:valid + input` 可以判断是IE10+浏览器，那我们也可以使用 `:valid + input + ~` 判断其兄弟元素也是IE10+浏览器，比方说下面的HTML代码：

```
<input class="hack">
<div class="loading spin"></div>
```

于是乎：

```
.hack:valid ~ .loading { /* IE10+浏览器 */ }
```

发现没有，颠覆性的：**CSS Hack**不是CSS代码，而是HTML元素！

于是，结合上面的 `::after` 或 `::before` 伪元素生成，我们就可以得到我们等尺寸且完美渐进增强的效果了！

如下CSS：

```
/* 隐藏Hack元素 */
.hack { position: absolute; visibility: hidden; }

.loading {
  width: 32px; height: 32px;
  background: url(loading-css3.gif);
}
/* IE10+/FireFox/Chrome/Opera */
.hack:valid ~ .loading {
  background: none;
}
.hack:valid ~ .loading::before {
  content: '';
  width: 3px; height: 3px;
  margin-left: 14.5px; margin-top: 14.5px; /* 居中显示 */
  border-radius: 100%;
  box-shadow: ...;
  position: absolute;
}
```

您可以狠狠地点击这里: [CSS3利用Hack元素实现兼容loading效果demo](#)

看看现在FireFox下的尺寸以及效果:



需要说明的

这里demo中, 起Hack作用的文本框和loading元素放在了一起, 我们在项目中实际使用的时候, 是不能这样子的。你想啊, 你的每一个loading效果旁边都跟着个隐藏的文本框元素, 岂不是很傻很天真。

实际使用的时候, 这个隐藏的Hack文本框显然是要放在很高的父级的, 由于无法与 `body` 平起平坐, 因此, 个人建议直接放在body标签之下 (下一节要具体展示), 然后, 类似:

```
.hack:valid ~ div .loading { /* div仅示意, 实际特定类等 */ }
```

整个页面只需要一个IE10+判断的HTML Hack元素, 你的loading效果可以自由使用。

## 5. CSS3 gradient的近乎完美处理

本段来自结尾, 😊

由于IE9及其以下浏览器不认识CSS3 背景渐变。因此, 对于IE10+我们需要重置掉的 `background: url(loading-css3.gif)` 就可以巧妙使用透明渐变背景实现。相关代码如下:

```
.loading {
  width: 32px; height: 32px;
  background: url(loading-css3.gif);
  /* IE10+ 去背景 */
  background:-webkit-gradient(linear, 0 0, 0 100%, from(rgba(0,0,0,0)), to(rgba(0,0,0,0)
));
  background:-moz-linear-gradient(top, rgba(0,0,0,0), rgba(0,0,0,0));
  background:-ms-linear-gradient(top, rgba(0,0,0,0), rgba(0,0,0,0));
  background:-o-linear-gradient(top, rgba(0,0,0,0), rgba(0,0,0,0));
  background:linear-gradient(top, rgba(0,0,0,0), rgba(0,0,0,0));
}
```

于是, 由于IE10+才认识 `::before/::after`, 于是, 再配合上面介绍的伪元素生成旋转动画, 页面单标签, 没有hack元素, 同时尺寸所有浏览器统一, 简直屌爆了!!

您可以狠狠地点击这里: [CSS3 background gradient渐变与兼容loading效果demo](#)



这里之所以说“近乎完美”, 是因为IE10浏览器向下自带IE9浏览器是认识 `::before/::after` 伪元素的, 因此, 会有重复背景, 不过, 这种情况我觉得可以忽略, 你觉得呢?

## 七、IE10+ CSS Hack介绍

如果你上面看得还算认真, 就应该知道我要讲的是怎么了。

如何写IE10+浏览器的hack呢? 如下处理:

1. body标签下放入hack元素, 如下截图所示:



2. 比方说, 我们的需要处理的元素全部都在 `#main` 的容器内, 于是, IE10+的CSS Hack就是:

```
.hack:valid ~ #main .ie10_other { }
```



或者是：

```
.hack:valid + #main .ie10_other { }
```

差别在于，前者 `#main` 只要在Hack元素之后，后者需要是相邻的兄弟，同时 `#main` 在Hack元素之后！

3. over!

千言万语不如一个demo来的实在，您可以狠狠地点击这里：[IE10+浏览器hack测试兼本文loading效果使用测试demo](#)

核心CSS为：

```
.hack:valid ~ #main .loading { /* 啪啪啪 */ }
```

页面效果如下（下图为IE10下）：



这是IE6下的：



有人可能要说了，我只需要IE10浏览器的Hack，而不是IE10+和其他浏览器的Hack。这很简单嘛：`\0`或`\9`不是IE10喜欢的零食嘛，在后面加上这个就好了，例如：

```
.hack:valid ~ #main .ie10 { color: green\0; }
```

补充，以上IE10 hack如果不满意，您可以试试这个：`_::-ms-reveal, .selector { }`

## 八、其他一些小tip

1. 伪类 `:read-only/:read-write` 也是IE9不支持，IE10+以及其他浏览器都支持。但是，目前，Firefox浏览器下，需要私有前缀才能使用，即 `:-moz-read-only/:-moz-read-write`。同时，需要注意的是，火狐的私有伪类必须与其他浏览器伪类分开写，否则无效，什么意思呢？看下面代码：

```
<input class="test1" readonly="readonly" />
<input class="test2" readonly="readonly" />
```

```
.test1:read-only{background-color:#eee;}
.test1:-moz-read-only{background-color:#eee;}
.test2:read-only, .test2:-moz-read-only{background-color:#eee;}
```

`.test1` 所在文本框在所有IE10+及其他浏览器中是有 `#eee` 背景色的，而 `.test2` 所在的文本框在任何浏览器下都没有效果。

正是由于这个不能逗号并列书写，因此，上面并没有把 `:read-only` 放在实现方法的队列中——虽然理论上很好用的。

您可以狠狠地点击这里：[:read-only伪类与逗号分隔符恩怨demo](#)



2. 伪类中有个称之为 `:empty` 的伪类，内容为空起作用，蛮有用的。有个疑问？如果我们使用 `::be`

`fore::after` 进行内容生成，那 `:empty` 伪类还起作用吗？

如下测试HTML以及CSS：

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

.empty {
  width: 32px; height: 32px;
  background: url(loading-css3.gif);
}
.empty::before { content: ' '; display: block; }
.empty:empty {
  background-color: #eee;
}
```

结果如下：



表明，内容生成不会影响 `:empty` 伪类，换句话说就是，`::before/::after` 不会影响 `:empty` 伪类的作用。

您可以狠狠地点击这里：[CSS3 content内容生成与empty伪类demo](#)

## 九、称为结语的部分

基本上都是这周研究整理出来的东西，可能有遗漏，可能有更好的方法，例如，可以利用CSS3背景渐变打补丁，remove掉IE10+浏览器的背景图片，这个自己还没来得及研究。有个同事回家结婚去了，最近活有点多，没有再多时间了。

擦，还是没忍住，有花了些时间折腾了 `CSS Gradients` 下的处理，在结尾处唠叨不太好，还是放到上面去吧~~☺

如果您发现本文有表述不准确的地方，欢迎大力指正。如果您有什么其他的想法，欢迎交流。或评论，或邮件都可以！

感谢阅读，希望本文的内容对你我的工作和学习有所帮助。

《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](#)。



« [翻译：web制作、开发人员需知的Web缓存知识](#)

[伪元素表单控件默认样式重置与自定义大全](#) »

猜你喜欢

- [CSS之before, after伪元素特性表现两则](#)
- [CSS3 :default伪类选择器简介](#)

- CSS蛋疼应用之：数据上报和HTML验证
- 折腾：2颗星星+纯CSS实现星星评分交互效果
- 小tip: CSS动态实现文本框清除按钮的隐藏与显示
- 伪元素表单控件默认样式重置与自定义大全
- CSS :focus-within伪类选择器及纯CSS下拉等应用举例
- HTML5 Boolean类型属性(如required)值的JS获取
- 残忍：IE10 IE7-IE9 type=email的完全抛弃
- 如何在CSS中实现父选择器效果?
- CSS实现跨浏览器的box-shadow盒阴影效果(2)

分享到： 1

标签： :before, :empty, :optional, :valid, animation, box-shadow, css3, CSS伪类, hack, IE10, rotate, transform, 伪元素, 兄弟选择器, 渐变背景

## 发表评论（目前15条评论）

<input type="text"/>	名称 (必须)
<input type="text"/>	邮件地址(不会被公开) (必须)
<input type="text"/>	网站
<div></div>	
<div>提交评论</div>	

- 路过说道：  
2015年05月27日 14:25

valid 可用于 单复选框、下拉框？ 请指点一二

[回复](#)


- 小黑说道：  
2013年06月13日 14:20

支持的浏览器有限 不然真可以用于项目当中

[回复](#)


- s说道：  
2013年06月7日 20:52

我只知道大街耍猴是违法的。。。

[回复](#)


- Bug说道：  
2013年06月7日 14:54



赞！干货满满，文字风趣！

[回复](#)

5. 叶小钗说道：

2013年06月7日 08:36

哥，学习啦

[回复](#)



6. 靖鸣君说道：

2013年06月6日 13:58

JS是耗CPU？css3耗GPU？图片是消耗系统什么东西，也是GPU？我觉得从代价上来评估，用图片是可靠、更方便。

张大哥的东西还是很赞啊！！！！

[回复](#)

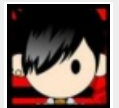


chenhz说道：

2013年06月7日 09:16

在中国CPU和GPU都不缺，最缺的是那可怜的带宽，你懂的。

[回复](#)



7. 哀音说道：

2013年05月30日 12:31

.hack:valid ~.loading::before

这种直接把::before的属性给.loading 覆盖width height 就行了吧，

[回复](#)



张 鑫旭说道：

2013年05月30日 19:33

@哀音 没错，眼神很犀利的嘛！主要是照顾IE10 IE9才这么写的哦~

[回复](#)



玖头说道：

2013年08月6日 11:12

大神，这个IE10 IE9是什么意思啊

[回复](#)



张 鑫旭说道：

2013年08月6日 11:57

@玖头 IE10浏览器下IE9兼容模式。



8. tgreen129说道：

2013年05月30日 09:07

不错，学习了

[回复](#)



9. web someone说道：

2013年05月30日 01:35

這個.....為毛每次插圖都是那個女的 哈哈哈



[回复](#)

10. **hdcn**说道:

2013年05月30日 00:09

前端的同学真心不容易啊，一个小小的Loading indicator就需要这么多功夫

[回复](#)



11. **exoticknight**说道:

2013年05月29日 23:09

渐进的讲解形式非常好，非常有帮助！

[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](#) <sup>(193)</sup>
- » [未来必热: SVG Sprite技术介绍](#) <sup>(120)</sup>
- » [粉丝群第1期CSS小测点评与答疑](#) <sup>(115)</sup>
- » [HTML5终极备忘大全 \(图片版+文字版\)](#) <sup>(93)</sup>
- » [让所有浏览器支持HTML5 video视频标签](#) <sup>(86)</sup>
- » [Selectivizr-让IE6~8支持CSS3伪类和属性选择器](#) <sup>(82)</sup>
- » [CSS3下的147个颜色名称及对应颜色值](#) <sup>(80)</sup>
- » [视区相关单位vw, vh..简介以及可实际应用场景](#) <sup>(77)</sup>
- » [写给自己看的display: flex布局教程](#) <sup>(76)</sup>
- » [小tips: 纯CSS实现打字动画效果](#) <sup>(76)</sup>

#### 今年热议

- » [《CSS世界》女主角诚寻靠谱一起奋斗之人](#) <sup>(76)</sup>
- » [不借助Echarts等图形框架原生JS快速实现折线图效果](#) <sup>(64)</sup>
- » [看, for..in和for..of在那里吵架!](#) <sup>(60)</sup>
- » [是时候好好安利下LuLu UI框架了!](#) <sup>(47)</sup>
- » [原来浏览器原生支持JS Base64编码解码](#) <sup>(35)</sup>
- » [妙法攻略: 渐变虚框及边框滚动动画的纯CSS实现](#) <sup>(33)</sup>
- » [炫酷H5中序列图片视频化播放的高性能实现](#) <sup>(31)</sup>

- » [CSS scroll-behavior和JS scrollIntoView让页面滚动平滑](#) <sup>(30)</sup>
- » [windows系统下批量删除OS X系统.DS\\_Store文件](#) <sup>(26)</sup>
- » [写给自己看的display: flex布局教程](#) <sup>(26)</sup>

## 猜你喜欢

- [CSS之before, after伪元素特性表现两则](#)
- [CSS3 :default伪类选择器简介](#)
- [CSS蛋疼应用之：数据上报和HTML验证](#)
- [折腾：2颗星+纯CSS实现星星评分交互效果](#)
- [小tip: CSS动态实现文本框清除按钮的隐藏与显示](#)
- [伪元素表单控件默认样式重置与自定义大全](#)
- [CSS :focus-within伪类选择器及纯CSS下拉等应用举例](#)
- [HTML5 Boolean类型属性\(如required\)值的JS获取](#)
- [残忍：IE10 !\[\]\(5ba1bc70d78f05c00988641e5e513c62\_img.jpg\) IE7-IE9 type=email的完全抛弃](#)
- [如何在CSS中实现父选择器效果?](#)
- [CSS实现跨浏览器的box-shadow盒阴影效果\(2\)](#)