CSS兼容IE6 IE7的方法

**CSS hack简介:**CSS hack由于不同厂商的[浏览器](http://baike.baidu.com/view/7718.htm)，比如Internet Explorer,Safari,[Mozilla Firefox](http://baike.baidu.com/view/193667.htm),Chrome等，或者是同一厂商的浏览器的不同版本，如IE6和IE7，对CSS的解析认识不完全一样，因此会导致生成的页面效果不一样，得不到我们所需要的页面效果。 这个时候我们就需要针对不同的浏览器去写不同的CSS，让它能够同时兼容不同的浏览器，能在不同的浏览器中也能得到我们想要的页面效果。简单的说，CSS hack的目的就是使你的CSS代码兼容不同的浏览器。当然，我们也可以反过来利用CSS hack为不同版本的浏览器定制编写不同的CSS效果。

**CSS hack原理**：由于不同的浏览器对CSS的支持及解析结果不一样，还由于CSS中的优先级的关系。我们就可以根据这个来针对不同的[浏览器](http://baike.baidu.com/view/7718.htm)来写不同的CSS。

CSS Hack大致有3种表现形式，CSS类内部Hack、选择器Hack以及HTML头部引用(if IE)Hack。

CSS Hack主要针对类内部[Hack](http://baike.baidu.com/view/63922.htm)：比如 IE6能识别下划线"\_"和星号" \* "，IE7能识别星号" \* "，但不能识别下划线"\_"，而firefox两个都不能认识。等等

选择器Hack：比如 IE6能识别\*html .class{}，[IE7](http://baike.baidu.com/view/392749.htm)能识别\*+html .class{}或者\*:first-child+html .class{}。等等

HTML头部引用(if IE)Hack：针对所有IE：<!--[if IE]><!--您的代码--><![endif]-->，针对IE6及以下版本：<!--[if lt IE 7]><!--您的[代码](http://baike.baidu.com/view/41.htm)--><![endif]-->，这类Hack不仅对CSS生效，对写在判断语句里面的所有代码都会生效。书写顺序，一般是将识别能力强的浏览器的CSS写在后面。

## 常用的CSS hack方式:

## （1）方式一 条件注释法

**只在IE下生效:**<!--[if IE]>这段文字只在IE浏览器显示<![endif]-->

**只在IE6下生效**<!--[if IE 6]>这段文字只在IE6浏览器显示<![endif]-->

**只在IE6以上版本生效**

<!--[if gte IE 6]>这段文字只在IE6以上(包括)版本IE浏览器显示<![endif]-->

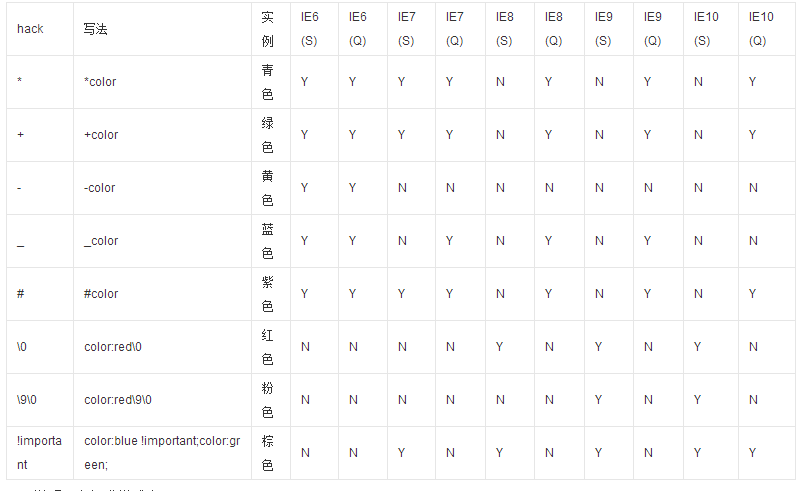
**只在IE8上不生效** <!--[if ! IE 8]> 这段文字在非IE8浏览器显示 <![endif]-->

**非IE浏览器生效** <!--[if !IE]> 这段文字只在非IE浏览器显示 <![endif]-->

## (2) 方式二 类内属性前缀法

属性前缀法是在CSS样式属性名前加上一些只有特定浏览器才能识别的hack前缀，以达到预期的页面展现效果。

IE浏览器各版本 CSS hack 对照表



说明：在标准模式中

.test{  
color:#000000;/\* FF,OP支持 \*/   
color:#0000FF\0; /\* IE8支持\*/   
[color:#000000;color:#00FF00;/\* SF,CH支持 \*/   
\*color:#FFFF00;/\* IE7支持 \*/  
\_color:#FF0000;/\* IE6支持 \*/   
}

“-″减号是IE6专有的hack

“\9″ IE6/IE7/IE8/IE9/IE10都生效

“\0″ IE8/IE9/IE10都生效，是IE8/9/10的hack

“\9\0″ 只对IE9/IE10生效，是IE9/10的hack

**（3）CSS hack方式三：**选择器前缀法 选择器前缀法是针对一些页面表现不一致或者需要特殊对待的浏览器，在CSS选择器前加上一些只有某些特定浏览器才能识别的前缀进行hack。

目前最常见的是

\*html \*前缀只对IE6生效\*+html \*+前缀只对IE7生效 @media screen\9{...}只对IE6/7生效 @media \0screen {body { background: red; }}只对IE8有效 @media \0screen\,screen\9{body { background: blue; }}只对IE6/7/8有效 @media screen\0 {body { background: green; }} 只对IE8/9/10有效 @media screen and (min-width:0\0) {body { background: gray; }} 只对IE9/10有效@media screen and (-ms-high-contrast: active), (-ms-high-contrast: none) {body { background: orange; }} 只对IE10有效等等

实际应用

编辑比如要分辨IE6和firefox两种浏览器，可以这样写：

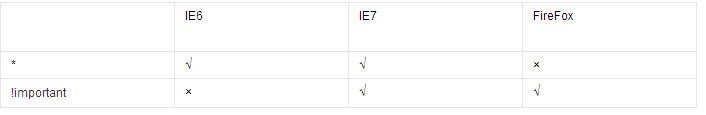


我在IE6中看到是红色的，在firefox中看到是绿色的。

解释一下：上面的css在firefox中，它是认识不了后面的那个带星号的东西是什么的，于是将它过滤掉，不予理睬，解析得到的结果是:div{background:green},于是理所当然这个div的背景是绿色的。在IE6中呢，它两个background都能识别出来，它解析得到的结果是:div{background:green;\*background:red;},于是根据优先级别，处在后面的red的优先级高，于是当然这个div的背景颜色就是红色的了。CSS hack:区分IE6，IE7，firefox



注：IE都能识别\*;标准浏览器(如FF)不能识别\*；IE6能识别\*；不能识别 !important;IE7能识别\*，能识别!important;FF不能识别\*，但能识别!important;



浏览器优先级别:FF<IE7<IE6,CSS hack书写顺序一般为FF IE7 IE6

以: " #demo {width:100px;} "为例; #demo {width:100px;} /\*被FIREFOX,IE6,IE7执行.\*/ \* html #demo {width:120px;} /\*会被IE6执行,之前的定义会被后来的覆盖,所以#demo的宽度在IE6就为120px; \*/ \*+html #demo {width:130px;} /\*会被IE7执行\*/

所以最后,#demo的宽度在三个浏览器的解释为: FIREFOX:100px; ie6:120px; ie7:130px;

IE8 最新css hack：

"\9"　例:"border:1px \9;".这里的"\9"可以区别所有IE和FireFox.（只针对IE9 Hack）

"\0"　IE8识别，IE6、IE7不能.

"\*"　IE6、IE7可以识别.IE8、FireFox不能.

"\_"　IE6可以识别"\_",IE7、IE8、FireFox不能.

IE6 hack

1) \_background-color:#CDCDCD;/\*ie6\*/

IE7 hack \*background-color:#dddd00;

/\* ie 7\*/IE8 hack background-color:red \0;

/\* ie 8/9\*/IE9 hack background-color:blue \9\0;火狐，傲游，浏览器通用 background-color:red!important;

注意写hack的顺序，其中：

background-color:red\0;IE8和IE9都支持；

background-color:blue\9\0; 仅IE9支持；

另外，background-color:eeeeee\9;的HACK支持IE6-IE8，

但是IE8不能识别“\*”和“\_”的CSS HACK。

可综合上述规律灵活应用。 IE9 和 IE8 以及其他版本的区别说明 background-color:blue; 各个浏览器都认识，这里给firefox用； background-color:red\9;\9所有的ie浏览器可识别； background-color:yellow\0; \0 是留给ie8的，最新版opera也认识，后面自有hack写了给opera认的，所以，\0我们就认为是给ie8留的； +background-color:pink; + ie7定了； \_background-color:orange; \_专门留给神奇的ie6； :root #test { background-color:purple\9; } :root是给ie9的，网上流传了个版本是 :root #test { background-　color:purple\0;}，这个，新版opera也认识，所以经笔者反复验证最终ie9特有的为:root 选择符 {属性\9;} @media all and (min-width:0px){ #test {background-color:black\0;} } 这个是老是跟ie抢着认\0的神奇的opera，必须加个\0,不然firefox，chrome，safari也都认识。。。 @media screen and (-webkit-min-device-pixel-ratio:0){ #test {background-color:gray;} }最后这个是浏览器新贵chrome和safari的。

选择符级Hack CSS内部选择符级Hack 语法 <hack> selector{ sRules } 说明

选择不同的浏览器及版本 尽可能减少对CSS Hack的使用。Hack有风险，使用需谨慎 通常如未作特别说明，本文档所有的代码和示例的默认运行环境都为标准模式。 一些CSS Hack由于浏览器存在交叉认识，所以需要通过层层覆盖的方式来实现对不同浏览器进行Hack的。 简单列举几个： \* html .test{color:#090;} /\* For IE6 and earlier \*/ \* + html .test{color:#ff0;} /\* For IE7 \*/ .test:lang(zh-cn){color:#f00;} /\* For IE8+ and not IE \*/ .test:nth-child(1){color:#0ff;} /\* For IE9+ and not IE \*/ 内部属性Hack CSS内部属性级Hack 语法：selector{<hack>?property:value<hack>?;} 取值： \_： 选择IE6及以下。连接线（中划线）（-）亦可使用，为了避免与某些带中划线的属性混淆，所以使用下划线（\_）更为合适。 \*：选择IE7及以下。诸如：（+）与（#）之类的均可使用，不过业界对（\*）的认知度更高。 \9：选择IE6+。 \0：选择IE8+和Opera。 [;property:value;]; 选择webkit核心浏览器（Chrome,Safari）。IE7及以下也能识别。中括号内外的3个分号必须保留，第一个分号前可以是任意规则或任意多个规则。 [;color:#f00;]; 与 [color:#f00;color:#f00;]; 与 [margin:0;padding:0;color:#f00;]; 是等价的。生效的始终是中括号内的最后一条规则，所以通常选用第一种写法最为简洁。

说明：一些CSS Hack由于浏览器存在交叉认识，所以需要通过层层覆盖的方式来实现对不同浏览器进行Hack的。如下面这个例子：如想同一段文字在IE6,7,8，chrome，safari，显示为不同颜色，可这样写[1]  ： .test{ color:#000; /\* 正常写法普遍支持 \*/ color:#00F\9; /\* 所有IE浏览器(ie6+)支持 \*/ /\*但是IE8不能识别“ \* ”和“ \_ ” \*/ [color:#000;color:#0F0; /\* SF,CH支持 \*/ color:#00F\0; /\* IE8支持\*/ \*color:#FF0; /\* IE7支持 \*/ \_color:#F00; /\* IE6支持 \*/ }

注意了：不管是什么方法，书写的顺序都是firefox的写在前面，IE7的写在中间，IE6的写在最后面。 补充：IE6能识别 \*，但不能识别 !important,IE7能识别 \*，也能识别!important;FF不能识别 \*，但能识别!important;下划线”\_“,IE6支持下划线，IE7和firefox均不支持下划线 。