**什么是CSS hack**

由于不同厂商的流览器或某浏览器的不同版本（如IE6-IE11,Firefox/Safari/Opera/Chrome等），对CSS的支持、解析不一样，导致在不同浏览器的环境中呈现出不一致的页面展现效果。这时，我们为了获得统一的页面效果，就需要针对不同的浏览器或不同版本写特定的CSS样式，我们把这个针对不同的浏览器/不同版本写相应的CSS code的过程，叫做CSS hack!

**CSS hack的原理**

由于不同的浏览器和浏览器各版本对CSS的支持及解析结果不一样，以及CSS优先级对浏览器展现效果的影响，我们可以据此针对不同的浏览器情景来应用不同的CSS。

**CSS hack分类**

CSS Hack大致有3种表现形式，CSS属性前缀法、选择器前缀法以及IE条件注释法（即HTML头部引用if IE）Hack，实际项目中CSS Hack大部分是针对IE浏览器不同版本之间的表现差异而引入的。

* 属性前缀法(即类内部Hack)：例如 IE6能识别下划线"\_"和星号" \* "，IE7能识别星号" \* "，但不能识别下划线"\_"，IE6~IE10都认识"\9"，但firefox前述三个都不能认识。
* 选择器前缀法(即选择器Hack)：例如 IE6能识别\*html .class{}，IE7能识别\*+html .class{}或者\*:first-child+html .class{}。
* IE条件注释法(即HTML条件注释Hack)：针对所有IE(注：IE10+已经不再支持条件注释)： <!--[if IE]>IE浏览器显示的内容 <![endif]-->，针对IE6及以下版本： <!--[if lt IE 6]>只在IE6-显示的内容 <![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。

CSS hack书写顺序，一般是将适用范围广、被识别能力强的CSS定义在前面。

**CSS hack方式一：条件注释法**

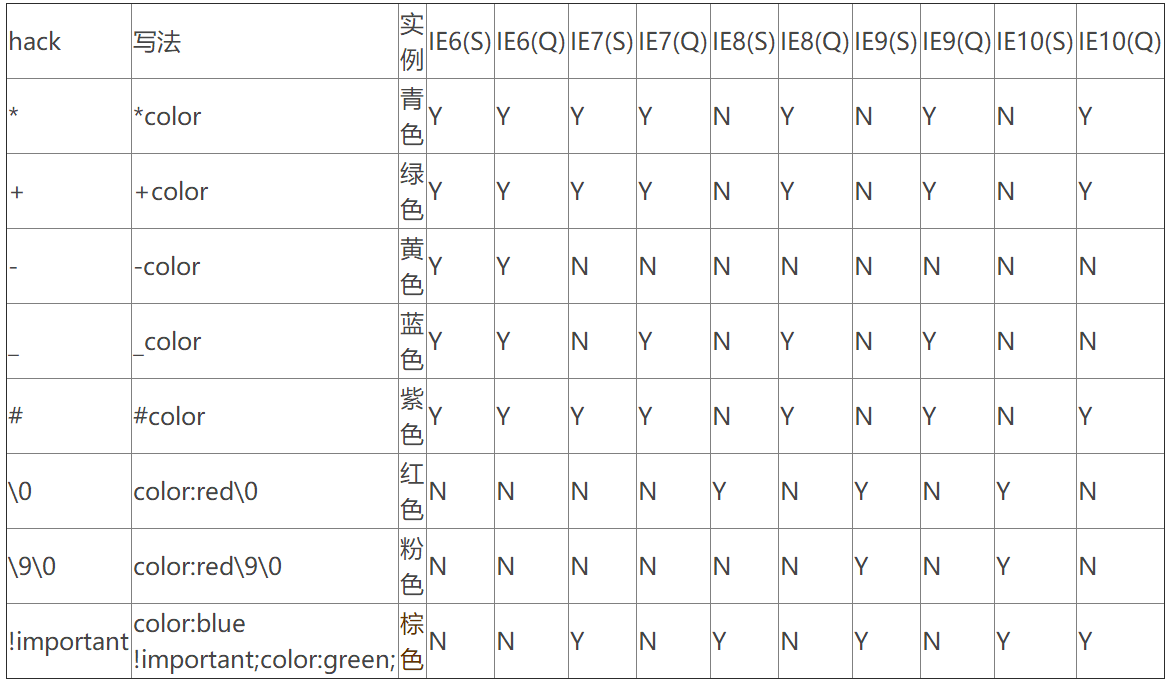
这种方式是IE浏览器专有的Hack方式，微软官方推荐使用的hack方式。举例如下

* 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]-->

**CSS hack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

* “-″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

demo如下

**[css]** [view plain](http://blog.csdn.net/freshlover/article/details/12132801) [copy](http://blog.csdn.net/freshlover/article/details/12132801)

* <script type="text/javascript">
* //alert(document.compatMode);
* </script>
* <style type="text/css">
* body:nth-of-type(1) .iehack{
* **color**: #F00;/\* 对Windows IE9/Firefox 7+/Opera 10+/所有Chrome/Safari的CSS hack ，选择器也适用几乎全部Mobile/Linux/Mac browser\*/
* }
* .demo1,.demo2,.demo3,.demo4{
* **width**:100px;
* **height**:100px;
* }
* .hack{
* /\*demo1 \*/
* /\*demo1 注意顺序，否则IE6/7下可能无法正确显示，导致结果显示为白色背景\*/
* **background-color**:red; /\* All browsers \*/
* **background-color**:blue !important;/\* All browsers but IE6 \*/
* \***background-color**:black; /\* IE6, IE7 \*/
* +**background-color**:yellow;/\* IE6, IE7\*/
* **background-color**:gray\9; /\* IE6, IE7, IE8, IE9, IE10 \*/
* **background-color**:purple\0; /\* IE8, IE9, IE10 \*/
* **background-color**:orange\9\0;/\*IE9, IE10\*/
* **\_background-color**:green; /\* Only works in IE6 \*/
* \*+**background-color**:pink; /\*  WARNING: Only works in IE7 ? Is it right? \*/
* }
* /\*可以通过javascript检测IE10，然后给IE10的<html>标签加上class=”ie10″ 这个类 \*/
* .ie10 #hack{
* **color**:red; /\* Only works in IE10 \*/
* }
* /\*demo2\*/
* .iehack{
* /\*该demo实例是用于区分标准模式下ie6~ie9和Firefox/Chrome的hack，注意顺序
* IE6显示为：绿色，
* IE7显示为：黑色，
* IE8显示为：红色，
* IE9显示为：蓝色，
* Firefox/Chrome显示为：橘色，
* （本例IE10效果同IE9,Opera最新版效果同IE8）
* \*/
* **background-color**:orange;  /\* all - for Firefox/Chrome \*/
* **background-color**:red\0;  /\* ie 8/9/10/Opera - for ie8/ie10/Opera \*/
* **background-color**:blue\9\0;  /\* ie 9/10 - for ie9/10 \*/
* \***background-color**:black;  /\* ie 6/7 - for ie7 \*/
* **\_background-color**:green;  /\* ie 6 - for ie6 \*/
* }
* /\*demo3
* 实例是用于区分标准模式下ie6~ie9和Firefox/Chrome的hack，注意顺序
* IE6显示为：红色，
* IE7显示为：蓝色，
* IE8显示为：绿色，
* IE9显示为：粉色，
* Firefox/Chrome显示为：橘色，
* （本例IE10效果同IE9，Opera最新版效果也同IE9为粉色）
* \*/
* .element {
* **background-color**:orange;    /\* all IE/FF/CH/OP\*/
* }
* .element {
* \***background-color**: blue;    /\* IE6+7, doesn't work in IE8/9 as IE7 \*/
* }
* .element {
* **\_background-color**: red;     /\* IE6 \*/
* }
* .element {
* **background-color**: green\0; /\* IE8+9+10  \*/
* }
* :root .element { **background-color**:pink\0; }  /\* IE9+10 \*/
* /\*demo4\*/
* /\*
* 该实例是用于区分标准模式下ie6~ie10和Opera/Firefox/Chrome的hack，本例特别要注意顺序
* IE6显示为：橘色，
* IE7显示为：粉色，
* IE8显示为：黄色，
* IE9显示为：紫色，
* IE10显示为：绿色，
* Firefox显示为：蓝色，
* Opera显示为：黑色，
* Safari/Chrome显示为：灰色，
* \*/
* .hacktest{
* **background-color**:blue;      /\* 都识别，此处针对firefox \*/
* **background-color**:red\9;      /\*all ie\*/
* **background-color**:yellow\0;    /\*for IE8/IE9/10 最新版opera也认识\*/
* +**background-color**:pink;        /\*for ie6/7\*/
* **\_background-color**:orange;       /\*for ie6\*/
* }
* @media screen and (**min-width**:0){
* .hacktest {**background-color**:black\0;}  /\*opera\*/
* }
* @media screen and (**min-width**:0) {
* .hacktest { **background-color**:purple\9; }/\*  for IE9/IE10  PS:国外有些习惯常写作\0，根本没考虑Opera也认识\0的实际 \*/
* }
* @media screen and (-ms-high-contrast: active), (-ms-high-contrast: none) {
* .hacktest { **background-color**:green; } /\* for IE10+ 此写法可以适配到高对比度和默认模式，故可覆盖所有ie10的模式 \*/
* }
* @media screen and (-webkit-min-device-pixel-ratio:0){ .hacktest {**background-color**:gray;} }  /\*for Chrome/Safari\*/
* /\* #963棕色 :root is for IE9/IE10, 优先级高于@media, 慎用！如果二者合用，必要时在@media样式加入 !important 才能区分IE9和IE10 \*/
* /\*
* :root .hacktest { background-color:#963\9; }
* \*/
* </style>

demo1是测试不同IE浏览器下hack 的显示效果

IE6显示为：粉色，

IE7显示为：粉色，

IE8显示为：蓝色，

IE9显示为：蓝色，

Firefox/Chrome/Opera显示为：蓝色，

若去掉其中的!important属性定义，则IE6/7仍然是粉色，IE8是紫色，IE9/10为橙色，Firefox/Chrome变为红色，Opera是紫色。是不是有些奇怪：除了IE6以外，其他所有的表现都符合我们的期待。那为何IE6表现的颜色不是\_background-color:green;的绿色而是\*+background-color:pink的粉色呢？其实是最后一句所谓的IE7私有hack惹的祸？不是说\*+是IE7的专有hack吗？？？错，你可能太粗心了！我们常说的IE7专有\*+hack的格式是\*+html selector，而不是上面的直接在属性上加\*+前缀。如果是为IE7定制特殊样式，应该这样使用：

* \*+html #ie7test { /\* IE7 only\*/
* color:green;
* }

经过测试，我发现属性前缀\*+background-color:pink;只有IE6和IE7认识。而\*+html selector只有IE7认识。所以我们在使用时候一定要特别注意。

demo2实例是用于区分标准模式下ie6~ie9和Firefox/Chrome的hack，注意顺序

IE6显示为：绿色，

IE7显示为：黑色，

IE8显示为：红色，

IE9显示为：蓝色，

Firefox/Chrome显示为：橘色，

（本例IE10效果同IE9,Opera最新版效果同IE8）

demo3实例也是用于区分标准模式下ie6~ie9和Firefox/Chrome的hack，注意顺序

IE6显示为：红色，

IE7显示为：蓝色，

IE8显示为：绿色，

IE9显示为：粉色，

Firefox/Chrome显示为：橘色，

（本例IE10效果同IE9，Opera最新版效果也同IE9为粉色）

demo4实例是用于区分标准模式下ie6~ie10和Opera/Firefox/Chrome的hack，本例特别要注意顺序

IE6显示为：橘色，

IE7显示为：粉色，

IE8显示为：黄色，

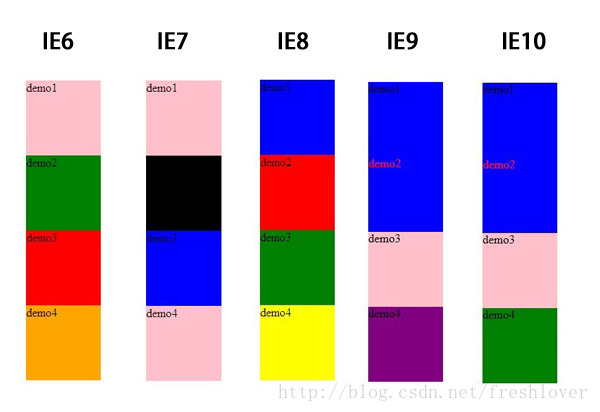
IE9显示为：紫色，

IE10显示为：绿色，

Firefox显示为：蓝色，

Opera显示为：黑色，

Safari/Chrome显示为：灰色，



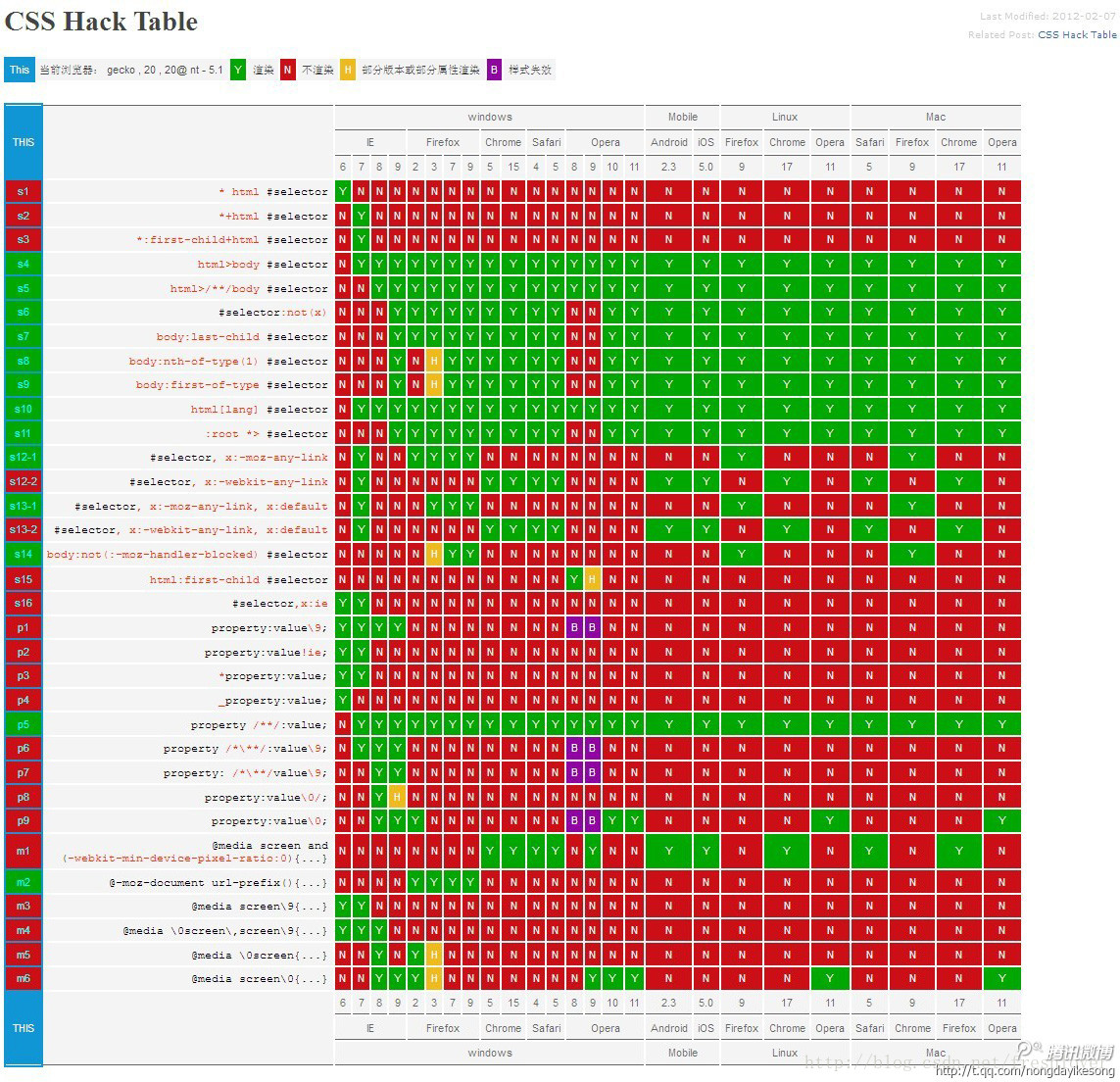
**CSS hack方式三：选择器前缀法**

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

结合CSS3的一些选择器，如html:first-child，body:nth-of-type(1)，衍生出更多的hack方式，具体的可以参考下表：



**CSS3选择器结合JavaScript的Hack**

我们用IE10进行举例：

由于IE10用户代理字符串（UserAgent）为：Mozilla/5.0 (compatible; MSIE 10.0; Windows NT 6.2; Trident/6.0)，所以我们可以使用javascript将此属性添加到文档标签中，再运用CSS3基本选择器匹配。

JavaScript代码:

* var htmlObj = document.documentElement;
* htmlObj.setAttribute('data-useragent',navigator.userAgent);
* htmlObj.setAttribute('data-platform', navigator.platform );

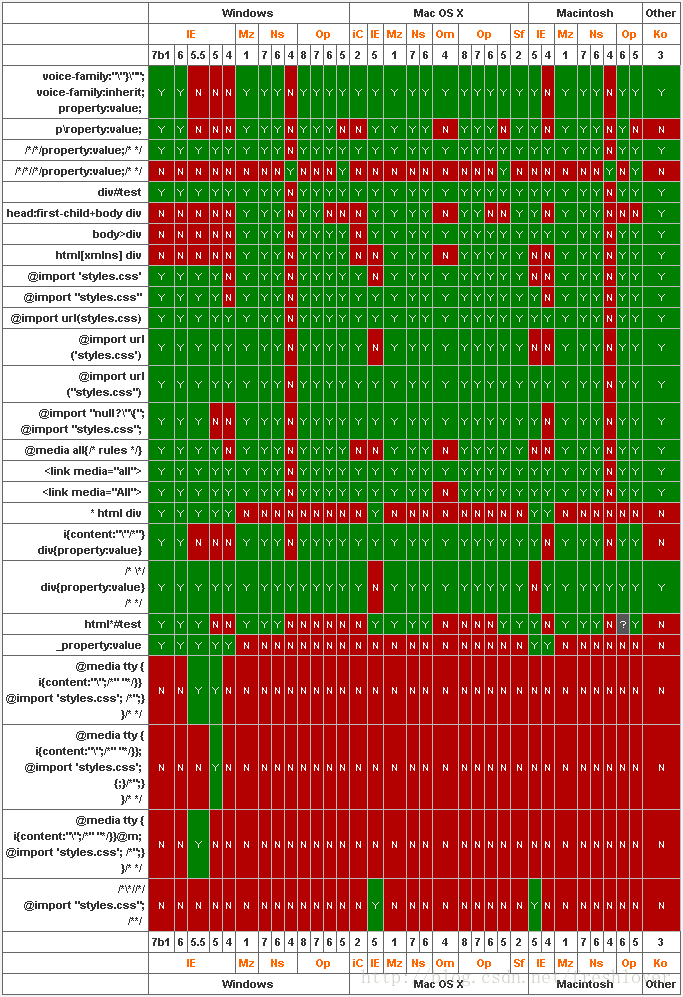
CSS3匹配代码：

* html[data-useragent\*='MSIE 10.0'] #id {
* color: #F00;
* }

**CSS hack利弊**

一般情况下，我们尽量避免使用CSS hack，但是有些情况为了顾及用户体验实现向下兼容，不得已才使用hack。比如由于IE8及以下版本不支持CSS3,而我们的项目页面使用了大量CSS3新属性在IE9/Firefox/Chrome下正常渲染，这种情况下如果不使用css3pie或htc或条件注释等方法时,可能就得让IE8-的专属hack出马了。使用hack虽然对页面表现的一致性有好处，但过多的滥用会造成html文档混乱不堪，增加管理和维护的负担。相信只要大家一起努力，少用、慎用hack，未来一定会促使浏览器厂商的标准越来越趋于统一，顺利过渡到标准浏览器的主流时代。抛弃那些陈旧的IE hack，必将减轻我们编码的复杂度，少做无用功。

最后补上一张引自国外某大牛总结的CSS hack表，这时一张6年前的旧知识汇总表了，放在这里仅供需要时候方便参考。



说明：本文测试环境为IE6~IE10，Chrome 29.0.1547.66 m，Firefox 20.0.1 ，Opera 12.02等。