

## 一：hack篇

### 1.1：注释

<!--[if IE]>

<![endif]-->

<!--[if it IE 8]>

<![endif]-->

<!--[if ! IE 8]>

<![endif]-->

### 操作符含义

it：小于；

gt：大于；

lte：小于等于；

gte：不小于；

!：不等于；

### 1.2：关于 !important

申明最高优先级，非必要时刻，尽量不要使用它。

遨游1.6及更低版本、IE6及更低版本无法识别他。

### 1.3：属性过滤器（较常见的hack方法）

\_：只有IE6识别。 eg：\_color:red;

\*：只有IE6，IE7识别。 eg：\*color:red;

\9：IE8及以下浏览器可以识别。 eg：color:red\9;

hack	写法	IE6(S)	IE6(Q)	IE7(S)	IE7(Q)	IE8(S)	IE8(Q)	IE9(S)	IE9(Q)	IE10(S)	IE10(Q)
*	*color	Y	Y	Y	Y	N	Y	N	Y	N	Y
+	+color	Y	Y	Y	Y	N	Y	N	Y	N	Y
-	-color	Y	Y	N	N	N	N	N	N	N	N
_	_color	Y	Y	N	Y	N	Y	N	Y	N	N
#	#color	Y	Y	Y	Y	N	Y	N	Y	N	Y
\0	color:red\0	N	N	N	N	Y	N	Y	N	Y	N
\9\0	color:red\9\0	N	N	N	N	N	N	Y	N	Y	N
!important	color:blue !important;color:green;	N	N	Y	N	Y	N	Y	N	Y	Y

## 二：css兼容方案

### 2.1 IE6不支持透明咋办？

解决：1>使用8位的PNG图片。

2>为IE6准备一套特殊的图片。

### 2.2 透明度写法

- opacity: 0.8; //通用
- filter: alpha(opacity=80); //IE
- filter: progid:DXImageTransform.Microsoft.Alpha(style=0,opacity=80); //IE6的写法

### 2.3 IE6双边距

行内属性设置了块级属性（display: block;）后，会产生双倍边距。

解决方案是在添加一个 display: inline; 或者 display: table;

### 2.4 双倍margin

IE6情况下，浮动元素设置了margin会产生双倍margin。

只需要浮动元素设置为display: inline, 或者是IE6的hack: \_margin就可以了。

### 2.5 min-height的兼容写法

```
.divBox{  
  min-height:200px;  
  height:auto !important;  
  height:200px;  
  overflow:visible;  
}
```

### 2.6 盒模型差异

IE盒模型：margin 、 content（包含border、padding）

W3C盒模型：margin 、 border、 padding、 content

CSS3中的box-sizing的属性就是为了这两种模式，border-box（IE盒明星）和content-box（W3C）

### 2.7 img标签-图片存在边距

vertical-align: 可以清除这种边距。

### 2.8 IE6高度无法小于10px

- 添加overflow的属性
- 设置font-size的属性为高度的大小

### 2.9 双倍float

```



```

## 2.10 ul列表边距属性不统一

- IE下 ul的边距是使用margin
- firefox下ul的边距是使用padding

重置CSS基本样式， 统一使用margin或者padding

## 2.11 IE6下select元素显示bug

select元素在IE6下是以窗口的形式展现的，所以在你需要弹出一个提示框（modal）的时候，你会发现select在modal的上面，无论你设置多大z-index都无效。解决的方法也很简单，利用一个透明的iframe盖住select就可以了

```

<div class=""wrapper>
  <!--[if IE 6]>
    <iframe style="position:absolute;top:0;left:0;width:100%;height:100px;
    z-index:-1;filter:alpha(opacity=0);opacity=0;border-style:none;">
    </iframe>
  <![endif]-->
  // TODO;
</div>

```

## 三：js兼容方案篇

### 3.1 document.formName.item()

```

document.formName.item("itemName");
//IE支持
document.formName.elements["elementName"];
//均支持

```

### 3.2 获取属性

根据很多的人的反馈来看，无论是常规的获取自定义属性，或是jq的attr()在不同的环境下都是存在着些许兼容性问题，所以为了保险起见，最好是使用原生的getAttribute()来获取属性；

### 3.3 ID

在IE下，我们是可以通过document.idName来获取元素，但是Firefox是不允许的。

而且Firefox中我们使用与html对象ID相同的变量名，所以，获取元素最好使用

document.getElementById("idName") 代替 document.idName，避免不必要的bug