IE6及各个浏览器兼容问题汇总

**1.<!DOCTYPE HTML>文档类型的声明。**

**产生条件：IE6浏览器，当我们没有书写这个文档声明的时候，会触发IE6浏览器的怪异解析现象；**

**解决办法：书写文档声明**

**2.不同浏览器当中，很多的标签的默认样式不同，如默认的外部丁内补丁。**

**产生条件：不同浏览器；**

**解决办法：利用[CSS reset](http://t.cn/zQBD8ZH)文件进行样式的清除，然后再根据需要进行设置。**

**3.横向双倍外边距(块属性标签float后，又有横行的margin情况下，在IE6显示margin比设置的大)**

**产生条件：在IE6中块元素浮动后，会出现[横向双倍margin](http://t.cn/zQBkpaU)现象。**

**解决办法： 在float标签的样式控制中加入\_display：inline**

**4.默认行高**

**产生条件：IE6、IE7、遨游浏览器；设置的文字高度超出盒模型内容区域设置的高度时会影响布局。**

**解决办法：给超出高度的标签设置overflow：hidden；或者将文字的行高line-height设置为小于块的高度。**

**5.img外部的border**

**产生条件：img外部有a标签，即img标签有链接时**

**解决办法：设置img边框border:0;**

**6.图片默认有间距。**

**产生条件：img标签（每个img之后敲了回车）**

**解决办法：为img设置float的浮动布局方式。**

**备注:** 因为img标签是行内属性标签，所以只要不超出容器宽度，img标签都会排在一行里，但是部分浏览器的img标签之间会有个间距。去掉这个间距使用float是正道。(最佳解决方案:display:block)

**7.经典3像素bug**

**产生条件：IE6浏览器，浮动块元素与未浮动块元素处于同一行，有默认的3px间距。**

**解决办法：设置非浮动元素浮动。**

**8.默认行高**

**产生条件：清除浮动的时候，有些人会采取一种[清浮动的方法](http://t.cn/zQBsPBt)，使用一个空的div，然后为这个div设置{clear：both}。在大部分**

**浏览器当中，这样做是没有任何问题的，但是在IE6浏览器当中，div即使是空的，也会存在默认行高。**

**解决办法：设置其高度为0，并设置overflow:hidden。{height:0;/overflow:hidden;/clear:both;} line-height:0; font-size:0;**

**9.a标签hover不适用于所有标签**

**产生条件：IE6浏览器中hover只支持a标签的使用，不支持一切其它标签使用；**

**解决办法：合理用a标签嵌套其他行内标签或者用javascript模拟a的hover效果；**

**10.table标签当中border-color属性设置无效**

**产生条件：IE6中table设置属性border-color无效；**

**解决办法：运用CSS样式进行控制，而不是使用属性进行样式的处理。**

**11.png格式图片 ie6支持png8 其他浏览器支持png24**

**产生条件：IE6浏览器，不支持透明；**

**解决办法：使用javascript进行处理；或者使用gif、jpg图像替代掉png图片的使用。**

**12.透明rgba（只针对背景颜色和背景图片）与opacity**

**产生条件：IE6不支持此两种透明的设置方法；**

**解决办法：使用IE6当中的滤镜filter替代掉，如：**

**opacity:0.6;**

**filter:alpha(opacity=60)。**

**0-100**

**13.子选择器在IE6中不能使用**

**产生条件：IE6浏览器，使用E>F子选择器；**

**解决办法：采用其他选择器或者采用后代选择器进行控制，如：div p{margin:10px;} div p p{margin:0;}替代掉 div>p{margin:10px;}。**

**14.不支持最大最小宽高**

**产生条件：IE6浏览器，标签的最低高度/宽度设置（min/max-height）**

.min\_width{ min-width:300px; width:300px;\_width:300px;width:auto;}

**解决办法：为IE6单独设置hack，即\_height:最小高度值；\_width：最小宽度值（对于IE6，当实际宽高超出定义的宽高时，元素会自动调整宽高）。对于最大高度和最大宽度，没有必要设置兼容，当前对于开发者来说，只需要保证IE6下正常显示即可，无需在它身上花费太多功夫。**

**15.input 聚焦框颜色与样式不同**

**产生条件：各个浏览器表现不同；**

**处理方法：使用outline:none，清除默认样式之后再统一设置。**

**16.子标签无法撑开父标签的高度**

**产生条件：父标签内部含一个或多个子标签，父标签没有设置浮动，而子标签发生浮动；**

**处理方法：方法1：在子标签最后添加清除浮动的设置<div style='height:0;clear:both'></div>; 方法2：为父标签添加{overflow:hidden;}的样式（不要用）； 方法3：为父标签设置固定高度。**

**17.li的间距问题**

**产生条件：IE6浏览器 li标签设置宽高，且li里面的元素发生了浮动；**

**处理方法：方法1：li不设置宽高； 方法2：li内部的标签不进行浮动。**

**18.行内元素布局混乱**

**产生条件：行内元素为包含框时，如果包含框包含的绝对定位元素以百分比为单位进行定位；**

**处理方法：在行内元素当中加入{zoom:1}，触发IE的hasLayout。**

**19.多显示一个字**

**产生条件：多个浮动元素中间夹杂HTML注释语句，浮动元素宽度设置为100%；则在下一行多显示一个上一行的最后一个字符；**

**处理方法：果断删掉注释！**

**20.img图片下部高度多余5px**

**产生条件：IE6浏览器；**

**处理方法：将图片转化为块级对象，即display:block。**

**21.** ie6.0横向margin加倍

产生因素：块属性、float、有横向margin。

解决方法：display：inline；

22. ie6.0下默认有行高

解决方法：overflow:hidden;或font-size:0;或line-height：xx px；

23. Ie6里面：如li设宽、高，并且li里面的标签浮动，那么li之间会有间距

解决方法：li不设宽、高或者li内的标签不浮动

24.li之间有间距

解决方法：li 设置vertical-align:middle;

25. ie6 不支持 fixed

解决办法:IE6设成绝对定位position:absolute

26. z-index不起作用的 bug

1）ie6下 首先讲讲第一种z-index无论设置多高都不起作用情况。这种情况发生的条件有三个：1、父标签position属性为relative；2、问题标签含有浮动(float)属性。

2）所有浏览器：它只认第一个爸爸

层级的高低不仅要看自己，还要看自己的老爸这个后台是否够硬。用术语具体描述为：

父标签position属性为relative或absolute时，子标签的absolute属性是相对于父标签而言的。而在IE6下，层级的表现有时候不是看子标签的z-index多高，而要看它们的父标签的z-index谁高谁低。

27.

**CSS hack**

CSS hack 地址

<http://www.duitang.com/static/csshack.html>

<http://blog.csdn.net/freshlover/article/details/12132801>

 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

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| hack | 写法 | 实例 | IE6(S) | IE6(Q) | IE7(S) | IE7(Q) | IE8(S) | IE8(Q) | IE9(S) | IE9(Q) | IE10(S) | IE10(Q) |
| \* | \*color | 青色 | Y | Y | Y | Y | N | Y | N | Y | N | Y |
| + | +color | 绿色 | Y | Y | Y | Y | N | Y | N | Y | N | Y |
| - | -color | 黄色 | Y | Y | N | N | N | N | N | N | N | N |
| \_ | \_color | 蓝色 | Y | Y | N | Y | N | Y | N | Y | N | N |
| # | #color | 紫色 | Y | Y | Y | Y | N | Y | N | Y | N | Y |
| \0 | color:red\0 | 红色 | N | N | N | N | Y | N | Y | N | Y | N |
| \9\0 | color:red\9\0 | 粉色 | N | N | N | N | N | N | Y | N | Y | N |
| !important | color:blue !important;color:green; | 棕色 | N | N | Y | N | Y | N | Y | N | Y | Y |

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

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{

.className{color:red;}

}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，

CSS hack利弊（面试的时候）

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。

360浏览器,搜狗 遨游浏览器 是引入了两个内核 一个是谷歌 一个是ie（系统本身的ie内核-->系统本身的ie最高是ie10 360 也是ie10）



极速模式 谷歌

兼容模式是ie

!important 比行内样式的权重还要大