**兼容性问题**

|  |
| --- |
| 所有浏览器：height:100px IE6专用：\_height:100px IE6专用：\*height:100px IE专用：\*+height:100px  IE7、FF专用：height:100px !important |
| !important 只针对IE6的兼容 |
| IE6、IE7 VS FF \*+html与\*html是IE特有的标签，FF不支持，\*+html又为IE7特有 |
| FF给div设置padding后会导致width和height增加，IE不会 |
| **居中问题：**  垂直居中：line-height设置为与当前div相同的高度，再通过vertical-align：middle 内容不换行  水平居中：margin：0 auto （不是万能的） |
| FF和IE对BOX理解的差异导致相差2px，float的div在IE下margin加倍等 |
| ul标签FF默认有list-style和padding，最好事先声明 |
| **IE 的bug**  1、IE6怪异解析之padding与border算入宽高  原因：未加文档声明造成非盒模型解析  解决方法：加入文档声明<!doctype html>  2、IE6在块元素、左右浮动、设定margin时造成margin双倍（双边距）  解决方法：display:inline  3、以下三种其实是同一种bug，其实也不算是个bug，举个例子：父标签高度20，子标签11，垂直居中，20-11=9，9要分给文字的上面与下面，怎么分？IE6就会与其它的不同，所以，尽量避免。  1）字体大小为奇数之边框高度少1px  解决方法：字体大小设置为偶数或line-height为偶数  2）line-height，文本垂直居中差1px  解决方法：padding-top代替line-height居中，或line-height加1或减1  3）与父标签的宽度的奇偶不同的居中造成1px的偏离  解决方法：如果父标签是奇数宽度，则子标签也用奇数宽度;如果是父标签偶数宽度，则子标签也用偶数宽度   4、内部盒模型超出父级时，父级被撑大  解决方法：父标签使用overflow:hidden   5、line-height默认行高bug  解决方法：line-height设值   6、行标签之间会有一小段空白  解决方法：float或结构并排(可读性差，不建议)   7、标签高度无法小于19px  解决方法：overflow: hidden;   8、左浮元素margin-bottom失效  解决方法：显示设置高度 or 父标签设置\_padding-bottom代替子标签的margin-bottom or 再放个标签让父标签浮动，子标签  margin- bottom，即(margin-bottom与float不同时作用于一个标签)   9、img于块元素中，底边多出空白  解决方法：父级设置overflow: hidden; 或 img { display: block; } 或 \_margin: -5px;   10、li之间会有间距  解决方法：float: left;   11、块元素中有文字及右浮动的行元素，行元素换行  解决方法：将行元素置于块元素内的文字前   12、position下的left，bottom错位  解决方法：为父级(relative层)设置宽高或添加\*zoom:1   13、子级中有设置position，则父级overflow失效  解决方法：为父级设置position:relative  以下是补充：上面要先看  1、终极方法：条件注释  <!--[if lte IE 6]> 这段文字仅显示在 IE6及IE6以下版本。 <![endif]-->  <!--[if gte IE 6]> 这段文字仅显示在 IE6及IE6以上版本。 <![endif]-->  <!--[if gt IE 6]> 这段文字仅显示在 IE6以上版本（不包含IE6）。 <![endif]-->  <!--[if IE 5.5]> 这段文字仅显示在 IE5.5。 <![endif]-->  <!--在 IE6及IE6以下版本中加载css-->  <!--[if lte IE 6]> <link type="text/css" rel="stylesheet" href="css/ie6.css" mce\_href="css/ie6.css" /><![endif]-->  缺点是在IE浏览器下可能会增加额外的HTTP请求数。   2、CSS选择器区分  IE6不支持子选择器；先针对IE6使用常规申明CSS选择器，然后再用子选择器针对IE7+及其他浏览器。  复制代码  代码如下:  /\* IE6 专用 \*/  .content {color:red;}  /\* 其他浏览器 \*/  div>p .content {color:blue;} -->  3、PNG半透明图片的问题  虽然可以通过JS等方式解决，但依然存在载入速度等问题，所以，这个在设计上能避免还是尽量避免为好。以达到网站最大优化。  4、IE6下的圆角  IE6不支持CSS3的圆角属性，性价比最高的解决方法就是用图片圆角来替代，或者放弃IE6的圆角。   5、IE6背景闪烁  如果你给链接、按钮用CSS sprites作为背景，你可能会发现在IE6下会有背景图闪烁的现象。造成这个的原因是由于IE6没有将背景图缓存，每次触发hover的时候都会重新加载，可以用JavaScript设置IE6缓存这些图片：  复制代码  代码如下:  document.execCommand("BackgroundImageCache",false,true);  6、最小高度  IE6 不支持min-height属性，但它却认为height就是最小高度。解决方法：使用ie6不支持但其余浏览器支持的属性!important。  复制代码  代码如下:  #container {min-height:200px; height:auto !important; height:200px;}  7、最大高度  复制代码  代码如下:  //直接使用ID来改变元素的最大高度  var container = document.getElementById('container');  container.style.height = (container.scrollHeight > 199) ? "200px" : "auto";  //写成函数来运行  function setMaxHeight(elementId, height){  var container = document.getElementById(elementId);  container.style.height = (container.scrollHeight > (height - 1)) ? height + "px" : "auto";  }  //函数示例  setMaxHeight('container1', 200);  setMaxHeight('container2', 500);  8、100% 高度  在IE6下，如果要给元素定义100%高度，必须要明确定义它的父级元素的高度，如果你需要给元素定义满屏的高度，就得先给html和body定义height:100%;。   9、最小宽度  同max-height和max-width一样，IE6也不支持min-width。  复制代码  代码如下:  //直接使用ID来改变元素的最小宽度  var container = document.getElementById('container');  container.style.width = (container.clientWidth < width) ? "500px" : "auto";  //写成函数来运行  function setMinWidth(elementId, width){  var container = document.getElementById(elementId);  container.style.width = (container.clientWidth < width) ? width + "px" : "auto";  }  //函数示例  setMinWidth('container1', 200);  setMinWidth('container2', 500);  10、最大宽度  复制代码  代码如下:  //直接使用ID来改变元素的最大宽度  var container = document.getElementById(elementId);  container.style.width = (container.clientWidth > (width - 1)) ? width + "px" : "auto";  //写成函数来运行  function setMaxWidth(elementId, width){  var container = document.getElementById(elementId);  container.style.width = (container.clientWidth > (width - 1)) ? width + "px" : "auto";  }  //函数示例  setMaxWidth('container1', 200);  setMaxWidth('container2', 500);  11、双边距Bug  当元素浮动时，IE6会错误的把浮动方向的margin值双倍计算。个人觉得较好解决方法是避免float和margin同时使用。   12、清除浮动  如果你想用div(或其他容器)包裹一个浮动的元素，你会发现必须给div(容器)定义明确的height、width、overflow之中一个属性（除了auto值）才能将浮动元素严实地包裹。  复制代码  代码如下:  #container {border:1px solid #333; overflow:auto; height:100%;}  #floated1 {float:left; height:300px; width:200px; background:#00F;}  #floated2 {float:right; height:400px; width:200px; background:#F0F;}  更多：http://www.twinsenliang.net/skill/20090413.html  13、浮动层错位  当内容超出外包容器定义的宽度时，在IE6中容器会忽视定义的width值，宽度会错误地随内容宽度增长而增长。  浮动层错位问题在IE6下没有真正让人满意的解决方法，虽然可以使用overflow:hidden;或overflow:scroll;来修正， 但hidden容易导致其他一些问题，scroll会破坏设计；JavaScript也没法很好地解决这个问题。所以建议是一定要在布局上避免这个问题发 生，使用一个固定的布局或者控制好内容的宽度（给内层加width）。   14、躲猫猫bug  在IE6和IE7下，躲猫猫bug是一个非常恼人的问题。一个撑破了容器的浮动元素，如果在他之后有不浮动的内容，并且有一些定义了:hover的链接，当鼠标移到那些链接上时，在IE6下就会触发躲猫猫。  解决方法很简单：  1.在（那个未浮动的）内容之后添加一个<span style="clear: both;"> </span>  2.触发包含了这些链接的容器的hasLayout，一个简单的方法就是给其定义height:1%;   15、绝对定位元素的1像素间距bug  IE6下的这个错误是由于进位处理误差造成（IE7已修复），当绝对定位元素的父元素高或宽为奇数时，bottom和right会产生错误。唯一的解决办法就是给父元素定义明确的高宽值，但对于液态布局没有完美的解决方法。   16、3像素间距bug  在IE6中，当文本(或无浮动元素)跟在一个浮动的元素之后，文本和这个浮动元素之间会多出3像素的间隔。  给浮动层添加 display:inline 和 -3px 负值margin  给中间的内容层定义 margin-right 以纠正-3px   17、IE下z-index的bug  在IE浏览器中，定位元素的z-index层级是相对于各自的父级容器，所以会导致z-index出现错误的表现。解决方法是给其父级元素定义z-index，有些情况下还需要定义position:relative。   18、Overflow Bug  在IE6/7中，overflow无法正确的隐藏有相对定位position:relative;的子元素。解决方法就是给外包容器.wrap加上position:relative;。   19、横向列表宽度bug  如果你使用float:left;把<li>横向摆列，并且<li>内包含的<a>（或其他）触发了 hasLayout，在IE6下就会有错误的表现。解决方法很简单，只需要给<a>定义同样的float:left;即可。   20、列表阶梯bug  列表阶梯bug通常会在给<li>的子元素<a>使用float:left;时触发，我们本意是要做一个横向的列表(通常 是导航栏)，但IE却可能呈现出垂直的或者阶梯状。解决办法就是给<li>定义float:left;而非子元素<a>，或者 给<li>定义display:inline;也可以解决。   **21、垂直列表间隙bug**  当我们使用<li> 包含一个块级子元素时，IE6(IE7也有可能)会错误地给每条列表元素（<li>）之间添加空隙。  解决方法：把<a>flaot并且清除float来解决这个问题；另外一个办法就是触发<a>的hasLayout（如定 义高宽、使用zoom:1;）；也可以给<li> 定义display:inline;来解决此问题；另外还有一个极有趣的方法，给包含的文本末尾添加一个空格。   **22、IE6中的:hover**  在IE6中，除了(需要有href属性)才能触发:hover行为，这妨碍了我们实现许多鼠标触碰效果，但还是有一些法子是可以解决它的。最好是不要用:hover来实现重要的功能，仅仅只用它来强化效果。   **23、IE6调整窗口大小的 Bug**  当把body居中放置，改变IE浏览器大小的时候，任何在body里面的相对定位元素都会固定不动了。解决办法：给body定义position:relative;就行了。  **24、文本重复Bug**  在IE6中，一些隐藏的元素（如注释、display:none;的元素）被包含在一个浮动元素里，就有可能引发文本重复bug。解决办法：给浮动元素添加display:inline;。  **25、IE 不同版本之间的兼容：**[**通过Meta标签解决网页兼容性问题**](http://blog.csdn.net/zhuchunyan_aijia/article/details/51568128)  Meta标签在网页的HTML源代码中一个重要的代码“”（即通常所说的META标签）。META标签用来描述一个HTML网页文档的属性，例如作者、日期和时间、网页描述、关键词、页面刷新等。Meta标签还可以解决网页兼容性问题，由于IE已更新到11版本，但是我们有不少网站系统还是采用的IE8兼容模式，那么我们可以通过Meta标签在IE9+以上版本默认使用IE7或者IE8的标准进行宣染！  Meta标签默认使用IE7宣染  <meta http-equiv="X-UA-Compatible" content="IE=7">  360浏览器内核控制Meta标签  若页面需默认用ie兼容内核，增加标签：<meta name=”renderer” content=”ie-comp”> 若页面需默认用ie标准内核，增加标签：<meta name=”renderer” content=”ie-stand”>  当然，我们也可以让360浏览器优先使用极速核  若页面需默认用极速核，增加标签：<meta name="renderer" content="webkit">  对于一些比较旧版的网站，建议使用以下标签： <meta name="renderer" content="ie-comp"> <meta name="renderer" content="ie-stand"> <meta http-equiv="X-UA-Compatible" content="IE=7">  <meta http-equiv="X-UA-Compatible" content="IE=7">  #以上代码告诉IE浏览器，无论是否用DTD声明文档标准，IE8/9都会以IE7引擎来渲染页面。  <meta http-equiv="X-UA-Compatible" content="IE=8">  #以上代码告诉IE浏览器，IE8/9都会以IE8引擎来渲染页面。  <meta http-equiv="X-UA-Compatible" content="IE=edge">  #以上代码告诉IE浏览器，IE8/9及以后的版本都会以最高版本IE来渲染页面。  <meta http-equiv="X-UA-Compatible" content="IE=7,IE=9">  <meta http-equiv="X-UA-Compatible" content="IE=7,9">  <meta http-equiv="X-UA-Compatible" content="IE=Edge,chrome=1">  #以上代码IE=edge告诉IE使用最新的引擎渲染网页，chrome=1则可以激活Chrome Frame. |