普及：浏览器的兼容性问题，往往是个别浏览器（没错，就是那个与众不同的浏览器）对于一些标准的定义不一致导致的。俗话说：没有IE就没有伤害。

贴士：内容都是自己总结的，不免会出现错误或者bug，欢迎更正和补充，本帖也会不断更新。

**Normalize.css**

不同浏览器的默认样式存在差异，可以使用 Normalize.css 抹平这些差异。当然，你也可以定制属于自己业务的 reset.css

|  |
| --- |
| <link href="<https://cdn.bootcss.com/normalize/7.0.0/normalize.min.css>" rel="stylesheet"> |

简单粗暴法

|  |
| --- |
| \* { margin: 0; padding: 0; } |

**html5shiv.js**

解决 ie9 以下浏览器对 html5 新增标签不识别的问题。

|  |
| --- |
| <!--[if lt IE 9]>    <script type="text/javascript" src="<https://cdn.bootcss.com/html5shiv/3.7.3/html5shiv.min.js>"></script>  <![endif]--> |

**respond.js**

解决 ie9 以下浏览器不支持 CSS3 Media Query 的问题。

|  |
| --- |
| <script src="<https://cdn.bootcss.com/respond.js/1.4.2/respond.min.js>"></script> |

**picturefill.js**

解决 IE 9 10 11 等浏览器不支持 <picture> 标签的问题

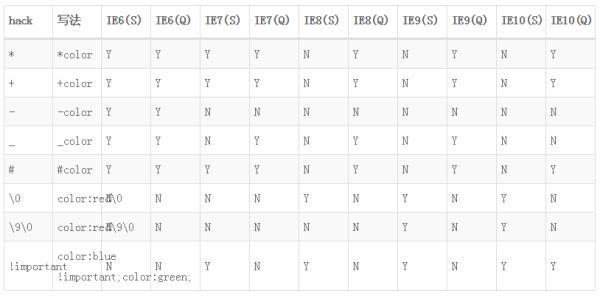
|  |
| --- |
| <script src="<https://cdn.bootcss.com/picturefill/3.0.3/picturefill.min.js>"></script> |

**IE 条件注释**

IE 的条件注释仅仅针对IE浏览器，对其他浏览器无效



**IE 属性过滤器（较为常用的hack方法）**

针对不同的 IE 浏览器，可以使用不同的字符来对特定的版本的 IE 浏览器进行样式控制  
  


**浏览器 CSS 兼容前缀**

|  |
| --- |
| -o-transform:rotate(7deg); // Opera    -ms-transform:rotate(7deg); // IE    -moz-transform:rotate(7deg); // Firefox    -webkit-transform:rotate(7deg); // Chrome    transform:rotate(7deg); // 统一标识语句 |

**a 标签的几种 CSS 状态的顺序**

很多新人在写 a 标签的样式，会疑惑为什么写的样式没有效果，或者点击超链接后，hover、active 样式没有效果，其实只是写的样式被覆盖了。

正确的a标签顺序应该是：==love hate==

* link:平常的状态
* visited:被访问过之后
* hover:鼠标放到链接上的时候
* active:链接被按下的时候

**完美解决 Placeholder**

|  |
| --- |
| <input type="text" value="Name \*" onfocus="this.value = '';" onblur="if (this.value == '') {this.value = 'Name \*';}"> |

**清除浮动 最佳实践**

|  |
| --- |
| .fl { float: left; }  .fr { float: right; }  .clearfix:after { display: block; clear: both; content: ""; visibility: hidden; height: 0; }  .clearfix { zoom: 1; } |

**BFC 解决边距重叠问题**

当相邻元素都设置了 margin 边距时，margin 将取最大值，舍弃小值。为了不让边距重叠，可以给子元素加一个父元素，并设置该父元素为 BFC：overflow: hidden;

|  |
| --- |
| <div class="box" id="box">    <p>Lorem ipsum dolor sit.</p>      <div style="overflow: hidden;">      <p>Lorem ipsum dolor sit.</p>    </div>      <p>Lorem ipsum dolor sit.</p>  </div> |

**IE6 双倍边距的问题**

设置 ie6 中设置浮动，同时又设置 margin，会出现双倍边距的问题

|  |
| --- |
| display: inline; |

**解决 IE9 以下浏览器不能使用 opacity**

|  |
| --- |
| opacity: 0.5;  filter: alpha(opacity = 50);  filter: progid:DXImageTransform.Microsoft.Alpha(style = 0, opacity = 50); |

**解决 IE6 不支持 fixed 绝对定位以及IE6下被绝对定位的元素在滚动的时候会闪动的问题**

|  |
| --- |
| /\* IE6 hack \*/  \*html, \*html body {    background-image: url(about:blank);    background-attachment: fixed;  }  \*html #menu {    position: absolute;    top: expression(((e=document.documentElement.scrollTop) ? e : document.body.scrollTop) + 100 + 'px');  } |

**IE6 背景闪烁的问题**

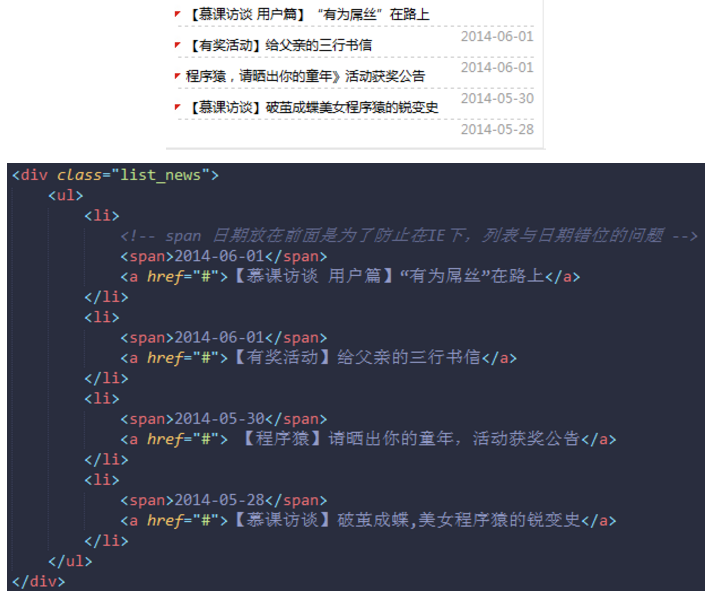
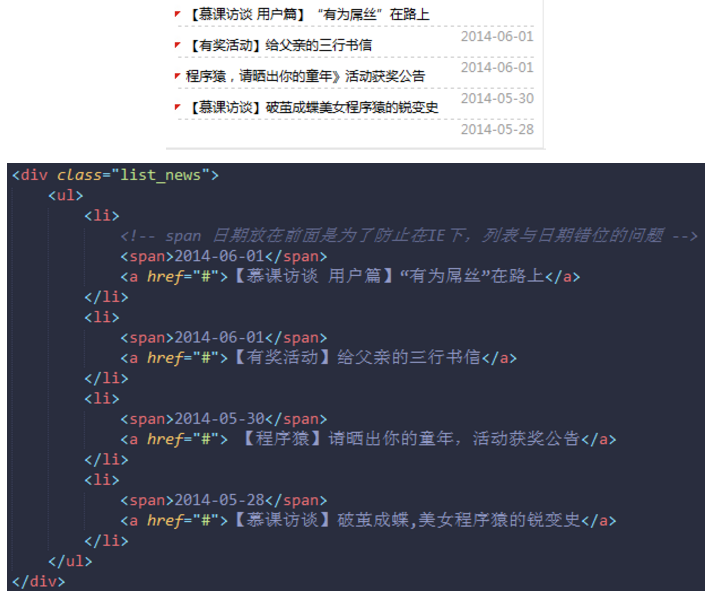
问题：链接、按钮用 CSSsprites 作为背景，在 ie6 下会有背景图闪烁的现象。原因是 IE6 没有将背景图缓存，每次触发 hover 的时候都会重新加载

解决：可以用 JavaScript 设置 ie6 缓存这些图片：

|  |
| --- |
| document.execCommand("BackgroundImageCache", false, true); |

**解决在 IE6 下，列表与日期错位的问题**

日期<span> 标签放在标题 <a> 标签之前即可

**解决 IE6 不支持 min-height 属性的问题**

|  |
| --- |
| min-height: 350px;  \_height: 350px; |

**让 IE7 IE8 支持 CSS3 background-size属性**

由于 background-size 是 CSS3 新增的属性，所以 IE 低版本自然就不支持了，但是老外写了一个 htc 文件，名叫 background-size polyfill，使用该文件能够让 IE7、IE8 支持 background-size 属性。其原理是创建一个 img 元素插入到容器中，并重新计算宽度、高度、left、top 等值，模拟 background-size 的效果。

|  |
| --- |
| html {    height: 100%;  }  body {    height: 100%;    margin: 0;    padding: 0;    background-image: url('img/37.png');    background-repeat: no-repeat;    background-size: cover;    -ms-behavior: url('css/backgroundsize.min.htc');    behavior: url('css/backgroundsize.min.htc');  } |

**IE6-7 line-height 失效的问题**

问题：在ie 中 img 与文字放一起时，line-height 不起作用

解决：都设置成 float

width:100%

width:100% 这个东西在 ie 里用很方便，会向上逐层搜索 width 值，忽视浮动层的影响.

Firefox 下搜索至浮动层结束，如此，只能给中间的所有浮动层加 width:100%才行，累啊。

opera 这点倒学乖了，跟了 ie

|  |
| --- |
| cursor:hand |

显示手型 cursor: hand，ie6/7/8、opera 都支持，但是safari 、 ff 不支持

|  |
| --- |
| cursor: pointer; |

**td 自动换行的问题**

问题：table 宽度固定，td 自动换行

解决：设置 Table 为 table-layout: fixed，td 为 word-wrap: break-word

**让层显示在 FLASH 之上**

想让层的内容显示在 flash 上，把 FLASH 设置透明即可

|  |
| --- |
| 1、<param name=" wmode " value="transparent">  2、<param name="wmode" value="opaque"> |

**键盘事件 keyCode 兼容性写法**

|  |
| --- |
| var inp = document.getElementById('inp')  var result = document.getElementById('result')    function getKeyCode(e) {    e = e ? e : (window.event ? window.event : "")    return e.keyCode ? e.keyCode : e.which  }    inp.onkeypress = function(e) {    result.innerHTML = getKeyCode(e)  } |

**求窗口大小的兼容写法**

|  |
| --- |
| // 浏览器窗口可视区域大小（不包括工具栏和滚动条等边线）  // 1600 \* 525  var client\_w = document.documentElement.clientWidth || document.body.clientWidth;  var client\_h = document.documentElement.clientHeight || document.body.clientHeight;    // 网页内容实际宽高（包括工具栏和滚动条等边线）  // 1600 \* 8  var scroll\_w = document.documentElement.scrollWidth || document.body.scrollWidth;  var scroll\_h = document.documentElement.scrollHeight || document.body.scrollHeight;    // 网页内容实际宽高 (不包括工具栏和滚动条等边线）  // 1600 \* 8  var offset\_w = document.documentElement.offsetWidth || document.body.offsetWidth;  var offset\_h = document.documentElement.offsetHeight || document.body.offsetHeight;    // 滚动的高度  var scroll\_Top = document.documentElement.scrollTop||document.body.scrollTop; |

**DOM 事件处理程序的兼容写法（能力检测）**

|  |
| --- |
| var eventshiv = {      // event兼容      getEvent: function(event) {          return event ? event : window.event;      },        // type兼容      getType: function(event) {          return event.type;      },        // target兼容      getTarget: function(event) {          return event.target ? event.target : event.srcelem;      },        // 添加事件句柄      addHandler: function(elem, type, listener) {          if (elem.addEventListener) {              elem.addEventListener(type, listener, false);          } else if (elem.attachEvent) {              elem.attachEvent('on' + type, listener);          } else {              // 在这里由于.与'on'字符串不能链接，只能用 []              elem['on' + type] = listener;          }      },        // 移除事件句柄      removeHandler: function(elem, type, listener) {          if (elem.removeEventListener) {              elem.removeEventListener(type, listener, false);          } else if (elem.detachEvent) {              elem.detachEvent('on' + type, listener);          } else {              elem['on' + type] = null;          }      },        // 添加事件代理      addAgent: function (elem, type, agent, listener) {          elem.addEventListener(type, function (e) {              if (e.target.matches(agent)) {                  listener.call(e.target, e); // this 指向 e.target              }          });      },        // 取消默认行为      preventDefault: function(event) {          if (event.preventDefault) {              event.preventDefault();          } else {              event.returnValue = false;          }      },        // 阻止事件冒泡      stopPropagation: function(event) {          if (event.stopPropagation) {              event.stopPropagation();          } else {              event.cancelBubble = true;          }      }  }; |

原文链接:<https://juejin.im/post/59a3f2fe6fb9a0249471cbb4>