**日常问题积累**

1. **z-index起作用的条件**

是元素的position是static之外（absolute、relative、fixed）才行

  position: relative;z-index: 9999;

**2、indexOf函数用法**

var dep = document.getElementById('change\_dep');  
 var list = document.getElementById('catalog\_list');  
 function change\_dep(){  
 var depStr = dep.innerHTML;  
 if(depStr.indexOf('展开全部')>0){  
 dep.innerHTML = '&nbsp;收起目录↑';  
 dep.className = 'btm\_develop02';  
 }  
 else{  
 dep.innerHTML = '&nbsp;展开全部↓';  
 l ist.className = 'catalog\_detail';  
 }  
 }

1. **css中的zoom的作用**

1、检查页面的标签是否闭合

不要小看这条，也许折腾了你两天都没有解决的 CSS BUG 问题，却仅仅源于这里。毕竟页面的模板一般都是由开发来嵌套的，而他们很容易犯此类问题。

快捷提示：可以用 Dreamweaver 打开文件检查，一般没有闭合的标签，会黄色背景高亮。

2、样式排除法

有些复杂的页面也许加载了 N 个外链 CSS 文件，那么逐个删除 CSS 文件，找到 BUG 触发的具体 CSS 文件，缩小锁定的范围。

对于刚才锁定的问题 CSS 样式文件，逐行删除具体的样式定义，定位到具体的触发样式定义，甚至是具体的触发样式属性。

3、模块确认法

有时候我们也可以从页面的 HTML 元素出发。删除页面中不同的 HTML 模块，寻找到触发问题的 HTML 模块。

4、检查是否清除浮动

其实有不少的 CSS BUG 问题是因为没有清除浮动造成的。养成良好的清除浮动的习惯是必要的，推荐使用 无额外 HTML 标签的清除浮动的方法（尽量避免使用 overflow:hidden;zoom:1 的类似方法来清除浮动，会有太多的限制性）。

5、检查 IE 下是否触发 haslayout

很多的 IE 下复杂 CSS BUG 都与 IE 特有的 haslayout 息息相关。熟悉和理解 haslayout 对于处理复杂的 CSS BUG 会事半功倍。推荐阅读 old9 翻译的 《On having layout》（如果无法翻越穿越伟大的 GFW，可阅读 蓝色上的转帖 ）

快捷提示：如果触发了 haslayout，IE 的调试工具 IE Developer Toolbar 中的属性中将会显示 haslayout 值为 -1。

6、边框背景调试法

故名思议就是给元素设置显眼的边框或者背景（一般黑色或红色），进行调试。此方法是最常用的调试 CSS BUG 的方法之一，对于复杂 BUG 依旧适用。经济实惠还环保^^

最后想强调一点的是，养成良好的书写习惯，减少额外标签，尽量语义，符合标准，其实可以为我们减少很多额外的复杂 CSS BUG，更多的时候其实是我们自己给自己制造了麻烦。希望你远离 BUG ，生活越来越美好。

**4、兼容问题01**

1、IE8图片产生黑色边框  img去除边框：img{border:none;}

2、IE8圆角不支持：behavior:url("htc/ie-css3.htc");

3、IE8placeholder不支持：引入placeholder.js插件  可以在行内调placeholder样式<input type="text" class="input" placeholder="中国服饰发展史" style="color: #ccc;line-height: 38px;">

    应该在行内给placeholder加样式

4、ie背景图片不显示，在no-repeat前加空格；

**5、鼠标变小手：{cursor：pointer；}**

cursor其他取值

Auto ：标准光标

Default ：标准箭头

pointer, hand ：手形光标

Wait ：等待光标

Text ：I形光标

vertical-text ：水平I形光标

no-drop ：不可拖动光标

not-allowed ：无效光标

Help ：帮助光标

all-scroll ：三角方向标

Move ：移动标

Crosshair ：十字标

e-resize

n-resize

nw-resize

w-resize

s-resize

se-resize

sw-resize

**6、文字中间横线**

<span style="text-decoration:line-through;">删除线效果</span>

<strike>今天天气真好！</strike> 今天天气真好！(横线直接穿越文字，删除线。)

<b>今天天气真好！</b> **今天天气真好！**  
 <i>今天天气真好！</i> 今天天气真好！  
 <u>今天天气真好！</u> 今天天气真好！  
 <tt>今天天气真好！</tt> 今天天气真好！  
 <sup>今天天气真好！</sup> 今天天气真好！(上标样式：如某个数的次方)  
 <sub>今天天气真好！</sub> 今天天气真好！(下标样式：如某个化学表达式H20)  
 <s>今天天气真好！</s> 今天天气真好！(横线直接穿越文字，删除线。)  
 <strike>今天天气真好！</strike> 今天天气真好！(横线直接穿越文字，删除线。)

**7、input type="radio"**

**8、**

-webkit-backface-visibility:hidden 这个CSS3可以在CSS3动画时解决闪烁的问题

**9、<u>下划线</u>    <del>删除线</del>**

**10、 a:link{}没访问过的链接样式**

**a:visited{}访问过的链接样式**

**a:hover{}鼠标进入链接区域的链接样式**

**a:active{}点击链接时的链接样式**

**11、前端开发学习什么？** 学习HTML、CSS和DIV+CSS技术用来制作Web页面；

学习JavaScript、DOM、BOM等用建立开发基础；

学习photoshop和Axure等软件应用，完成页面UI设计；

也要认识一下NodeJS和PHP加上数据库等这样的后端语言，方便前后端开发配合；

学习HTML5、CSS3、响应式页面布局、微网站制作等开发移动互联网的应用；

像Ajax、JQuery、jQueryMobile、Bootstrap、AngularJS等更是前端开发必学的高级技术；

HTML5地理位置应用、离线应用、webworker多线程实现、websocket、跨平台开发技术和webAPP开发等，这些企业级应用技术也是现在开发的主流；现在前端开发最热门、要人最多，就是使用canvas开发网页游戏动画，以及会用Cocos2d-js制作游戏等开发。

**12、可以用embed标签代替video标签嵌入视频，并设置视频的宽高**

**13、background:url(../1.files/map.png) no-repeat left center;**

**背景图片地址路径应该这么写，要不然不显示**

**14、出现滚动条：样式里加**overflow:auto;

1. **z-index：9999；**

**调层不好使，应加：position：absolute；**

**即：z-index：999；position：absolute；**

**16、text-align: right;右对齐，left center**

**17、居中图片方法：**

第一种方法：

设置父元素内文字居中即可让图片居中。  
 element{ text-align:center; }  
 第二种方法：  
 设置图片为块级元素，设置左右margin为auto即可让图片居中。  
 img{ display:block;margin-left:auto;margin-right:auto; }

**18、**

* text() - 设置或返回所选元素的文本内容
* html() - 设置或返回所选元素的内容（包括 HTML 标记）
* val() - 设置或返回表单字段的值

**19、**

* append() - 在被选元素的结尾插入内容
* prepend() - 在被选元素的开头插入内容
* after() - 在被选元素之后插入内容
* before() - 在被选元素之前插入内容

**20、**

* remove() - 删除被选元素（及其子元素）
* empty() - 从被选元素中删除子元素

**21、**

## jQuery width() 和 height() 方法

width() 方法设置或返回元素的宽度（不包括内边距、边框或外边距）。

height() 方法设置或返回元素的高度（不包括内边距、边框或外边距）。

## jQuery innerWidth() 和 innerHeight() 方法

innerWidth() 方法返回元素的宽度（包括内边距）。

innerHeight() 方法返回元素的高度（包括内边距）。

## jQuery outerWidth() 和 outerHeight() 方法

outerWidth() 方法返回元素的宽度（包括内边距和边框）。

outerHeight() 方法返回元素的高度（包括内边距和边框）。

**22、viewport（视口）：web移动端**

       <！--只有移动端识别-->

       <!--页面承载在viewport中-->

       <!--viewport有默认宽度-->

        <!--

            设置宽度：content="width:320px;"

            宽度适配：content="width=device-width"

            设置默认缩放比：content="initial-scale=1.0"

            是否允许自行缩放：user-scalable  0  1  no  yes

            允许最大：maximum-scale

            允许最小：minimum-scale

        -->

        <meta name="viewport" content="width=device-width,initial-scale=1.0,user-scalable=0 " />

1物理像素1分辨率

1px4物理像素

base.css:

reset.css

\*,::before,::after{

    /\*选择所有标签\*/

    margin：0；padding：0；

    /\*清除移动端默认的点击高亮效果\*/

    -webkit-tap-highlight-color:transparent;

    /\*设置所有的都是以边框开始计算宽度  百分比\*/

    -webkit-box-sizing:border-box;

    box-sizing:border-box;

}

body{

    font-size:14px;

    font-family:"MicroSoft YaHei",sans-serif;/\*设备默认字体/

    color:#333;

}

a{

    color:#333;

}

a:hover{

    text-decoration:none;

}

input{

    border:none;

    outline:none;

    /\*清除移动端默认的表单样式\*/

    -webkit-appearance:none;

}、

ul,li{

    list-style:none;

}

.f\_left{

    float:left

}

.f\_right{

    float:right

}

.clearfix::before,.clearfix::after{

    content:"";

    hight:0;

    line-hight:0;

    display:block;

    visibility:hidden;

    clear:both;

}

设置最大宽度和最小宽度；达到效果最佳

/\*主体盒子\*/

.jd\_layout{

    width:100%;

    max-width:640px;/\*不让放大显示效果好\*/

    min-width:300px;/\*为了更好的布局\*/

    margin:0 auto;

}

**23、去除文本基线：**

1、 img>display:block

2、父容器：font-size：0；

3、img>vertical-align:middle

**24、**

/\*原始尺寸显示图标会有毛边情况，显示不清晰失真\*/  
 /\*解决图标在移动端模糊 缩放 压缩图标\*/

/\*浮动定位以window最外层容器计算\*/

/\*opticaty:透明度影响效果；rgba不影响\*/

伪类转块：1、display：block； 2、float：left； 3、position：absolute；

&yen：人民币符号；

E： first-of-type 选择E同类型的同级的，第一个元素

text-decoration: line-through; /\*删除线\*/

**25、**img图片有左边距：img{display：block}

**26、**在JavaScript脚本中，如果程序员在对某个变量赋值之前未声明，赋值操作将自动声明该变量

var mood = “happy”，age = 33；

尽量使用驼峰命名法；

单双引号均可；

使用反斜线对字符进行转义 var mood = ‘don\' t ask';

var married = "true"; 字符串

var married = true; 布尔值

并非所有的元素都包含着属性，但所有的属性都被元素包含。

**27、**background-position：top right；

background：color image repeat attachment position 顺序

text-decoration：overline（上边）；

line-through（中间）；underline（下边）；none；

text-indent：50px；（文本缩进）

word-spacing：30px；字间距

font-style：normal；italic；obique；

链接可以用不同的样式：

a:link {color:#FF0000;}    /\* unvisited link \*/

a:visited {color:#00FF00;} /\* visited link \*/

a:hover {color:#FF00FF;}   /\* mouse over link \*/

a:active {color:#0000FF;}  /\* selected link \*/

列表可以有不同的样式：

ul.a {list-style-type:circle;}

ul.b {list-style-type:square;}

ol.c {list-style-type:upper-roman;}

ol.d {list-style-type:lower-alpha;}

也可以是：list-style-image:url('sqpurple.gif');

表格边框折叠重合为一个：

table{border-collapse:collapse;}

分别设置四个边的边框线：

p{

    border-top-style:dotted;

     border-right-style:solid;

    border-bottom-style:dotted;

     border-left-style:solid;

}

Fixed定位在ＩＥ７和ＩＥ８下需要描述！DOCTYPE才能支持；

absolute定位的元素的位置相对于最近的已定位父元素，如果元素没有已定位的父元素，那么相对于<html>

在IE8中使用margin：auto属性无法正常工作，需要声明！DOCTYPE；

margin-left：auto；margin-right：auto；也可居中，但width：100%；对齐是没有效果的；

before用法：h1:before {content:url(smiley.gif);} 在h1前面插入图片

<del>删除线</del> <ins>插入部分，下划线</ins>

**28、button按钮居中：{display:block;margin:0 auto;}**

高度auto、浮动float，高度不能撑开，应内层外层元素均浮动float；

**29、字符串断行需要使用反斜杠（\）：var x = "hello \world";**

因为在字符串中直接使用回车是会报错的；

**30、null用于对象，undefined用于变量属性和方法；**

对象只有被定义才有可能是null，否则为undefined；

**31、e = window.event || e;**

window.onload = function(e){

//alert(window.event.type) ie时调用，非ie注释掉否则报错

alert(e.type);

**}**

**32、display属性：**

none 此元素不会被显示。   
 block 此元素将显示为块级元素，此元素前后会带有换行符。   
 inline 默认。此元素会被显示为内联元素，元素前后没有换行符。   
 inline-block 行内块元素。（CSS2.1 新增的值）   
 list-item 此元素会作为列表显示。   
 run-in 此元素会根据上下文作为块级元素或内联元素显示。   
 compact CSS 中有值 compact，不过由于缺乏广泛支持，已经从 CSS2.1 中删除。   
 marker CSS 中有值 marker，不过由于缺乏广泛支持，已经从 CSS2.1 中删除。   
 table 此元素会作为块级表格来显示（类似 <table>），表格前后带有换行符。   
 inline-table 此元素会作为内联表格来显示（类似 <table>），表格前后没有换行符。   
 table-row-group 此元素会作为一个或多个行的分组来显示（类似 <tbody>）。   
 table-header-group 此元素会作为一个或多个行的分组来显示（类似 <thead>）。   
 table-footer-group 此元素会作为一个或多个行的分组来显示（类似 <tfoot>）。   
 table-row 此元素会作为一个表格行显示（类似 <tr>）。   
 table-column-group 此元素会作为一个或多个列的分组来显示（类似 <colgroup>）。   
 table-column 此元素会作为一个单元格列显示（类似 <col>）   
 table-cell 此元素会作为一个表格单元格显示（类似 <td> 和 <th>）   
 table-caption 此元素会作为一个表格标题显示（类似 <caption>）   
 inherit 规定应该从父元素继承 display 属性的值。

**33、**pt：点。px：像素。

一英寸=72pt=96px。9pt=12px。

**34、**addLoadEvent：

解决onload事件缺陷，事件会在网页完全下载完后才会执行，包括网页中的图片或Flash等，如果网页中的图片比较大或有很多图，可能还没等网页完全下载完网友已经点击链接到其它网页去了，这样这个事件就没有执行了。

//老式的调用positionMessage函数：

// window.onload = positionMessage;

//addLoadEventd方法

调用positionMessage函数：

        function addLoadEvent(func){

            var oldonload = window.onload;

            if(typeof window.onload !='function'){

                window.onload = func;

            }else{

                window.onload = function(){

                    oldonload();

                    func();

                }

            }

        }

        addLoadEvent(positionMessage);

**35、**强制文本换行：word-wrap：break-word;

**36、**if(!document.getElementById) return false;

    if (!document.getElementById("message")) return false;

判断浏览器是否支持document.getElementById("message")这种方式获得对象。

基本所有浏览器都支持，写不写都一样。

**37、**使左右div高度想等方法：

代码如下:

<!DOCTYPE html>

<html lang="en">

<head>

     <meta charset="UTF-8">

     <title>new document</title>

     <style type="text/css">

         .wrap{border:1px solid #006699;width:500px;overflow:hidden;}

         .wrap .left{float:left;width:50%;background:green;padding-bottom:3000px;margin-bottom:-3000px;}

         .wrap .right{float:right;width:50%;background:blue;padding-bottom:3000px;margin-bottom:-3000px;}

     </style>

</head>

<body>

     <div class="wrap">

         <div class="left">

             <p>内容</br></p><p>内容</br></p>

         </div>

         <div class="right">

             <p>内容</br></p><p>内容</br></p><p>内容</br></p><p>内容</br></p>

         </div>

     </div>

</body>

</html>

想看原理把overflow:hidden;换成overflow:scroll;

JavaScript方法：

<script type="text/javascript">

     function changeHeight(){

         var oldHeight = document.getElementById("oldHeight");

         var newHeight = document.getElementsByClassName("newHeight");

         return newHeight[0].style.height = oldHeight.offsetHeight+"px";

     }

     window.onload = function() {

         changeHeight();

     }

</script>

**38、filter滤镜：**

blur（px）：给图像设置高斯模糊；

brightness（%）：使图片更亮或更暗；

contrast（%）：调整图像的对比度；

drop-shadow(h-shadow v-shadow blur spread color)：给图像设置阴影效果；

grayscale（%）：将图像转换为灰度图像；

hue-rotate（deg）：给图像应用色相旋转；

invert（%）：反转输入图像；

opacity（%）：转化图像的透明度；

saturate（%）：转换图像饱和度；

sepia（%）：将图像转换为深褐色；

**39、**background-origin：规定background-position属性相对于什么来定位

padding-box：背景图像相对于内边距框来定位；

border-box：背景图像相对于边框盒来定位；

content-box：背景图像相对于内容框来定位；

**40、**background另类用法：

background-image：url(),url();可以不止一个；

background-position：right bottom，left top；也可这么定位；属性也是和url对应的；

background: url(img\_flwr.gif) right bottom no-repeat,

url(paper.gif) left top repeat;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**41、**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*尽量少用浮动，用行级块\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**42、**文字打断处理（固定宽度内显示，超过显示...）：

$("#interrupt").each(function(){

     var p\_height = $(this).height();

     var $p = $("#interrupt", $(this)).eq(0);

     while ($p.outerHeight() > p\_height) {

         $p.text($p.text().replace(/(\s)\*([a-zA-Z0-9]+|\W)(\.\.\.)?$/, "..."));

     };

});

鼠标放上显示完整标题：title属性；

**43、**el表达式保留两位小数：

<fmt:formatNumber value="${85157874.005575/1024/1024}" pattern="#,###.##" />  
 记得要导入

 <%@ taglib prefix="fmt" uri="http://java.sun.com/jsp/jstl/fmt"%>

也可不导入后者也是好使的！

如果是货币可用type="currency"

**44、**段落前空两个格：text-indent：2em

**45、**#interrupt {overflow: hidden;text-overflow: ellipsis;white-space: nowrap;}

**46、**jQuery点击切换颜色:

<script>

$(**function**(){

      $("li").each(**function**(){

         $(**this**).click(**function**(){

              $("li").addClass("default").removeClass("checked");

              $(**this**).addClass("checked").removeClass("default");

          })

     });

})

</script>

**47、**querySelector():方法返回文档中匹配指定 CSS 选择器的一个元素。

**注意：** querySelector() 方法仅仅返回匹配指定选择器的第一个元素。如果你需要返回所有的元素，请使用querySelectorAll() 方法替代。

**48、**clientX 事件属性返回当事件被触发时鼠标指针向对于浏览器页面（或客户区）的水平坐标。 客户区指的是当前窗口。语法: event.clientX

**49、**offsetLeft与style.left的区别

      offsetLeft 获取的是相对于父对象的左边距

      left 获取或设置相对于 具有定位属性(position定义为relative)的父对象的左边距，如果父div的position定义为relative,子div的position定义为absolute，那么子div的style.left的值是相对于父div的值，  
            这同offsetLeft是相同的，区别在于：  
            1. style.left 返回的是字符串，如28px，offsetLeft返回的是数值28，如果需要对取得的值进行计算，还用offsetLeft比较方便。  
            2. style.left是读写的，offsetLeft是只读的，所以要改变div的位置，只能修改style.left。  
            3. style.left的值需要事先定义，否则取到的值为空。而且必须要定义在html里，我做过试验，如果定义在css里，style.left的值仍然 为空，这就是我刚开始碰到的问题，总是取不到style.left的值。offsetLeft则仍然能够取到，无需事先定义div的位置。

**50、**

clientWidth:是对象看到的宽度（不含边线,即border）  
 scrollWidth:是对象实际内容的宽度（若无padding，那就是边框之间距离，如有padding，就是左padding和右padding之间距离）。  
 offsetWidth:是指对象自身的宽度，整型，单位像素（含边线，如滚动条的占用的宽，值会随着内容的输入而不断改变）。

scrollHeight: 获取对象的滚动高度。

scrollLeft:设置或获取位于对象左边界和窗口中目前可见内容的最左端之间的距离

scrollTop:设置或获取位于对象最顶端和窗口中可见内容的最顶端之间的距离

scrollWidth:获取对象的滚动宽度

offsetHeight:获取对象相对于版面或由父坐标 offsetParent 属性指定的父坐标的高度

offsetLeft:获取对象相对于版面或由 offsetParent 属性指定的父坐标的计算左侧位置

offsetTop:获取对象相对于版面或由 offsetTop 属性指定的父坐标的计算顶端位置

易混淆点：

clientX 设置或获取鼠标指针位置相对于当前窗口的 x 坐标，其中客户区域不包括窗口自身的控件和滚动条。   
 clientY 设置或获取鼠标指针位置相对于当前窗口的 y 坐标，其中客户区域不包括窗口自身的控件和滚动条。   
 offsetX 设置或获取鼠标指针位置相对于触发事件的对象的 x 坐标。   
 offsetY 设置或获取鼠标指针位置相对于触发事件的对象的 y 坐标。   
 screenX 设置或获取获取鼠标指针位置相对于用户屏幕的 x 坐标。   
 screenY 设置或获取鼠标指针位置相对于用户屏幕的 y 坐标。   
 x 设置或获取鼠标指针位置相对于父文档的 x 像素坐标(亦即相对于当前窗口)。   
 y 设置或获取鼠标指针位置相对于父文档的 y 像素坐标(亦即相对于当前窗口)。

document.documentElement.scrollTop 垂直方向滚动的值

event.clientX+document.documentElement.scrollTop 相对文档的水平座标+垂直方向滚动的量

以上主要指IE之中，FireFox差异如下：

IE6.0、FF1.06+：

clientWidth = width + padding  
 clientHeight = height + padding  
 offsetWidth = width + padding + border  
 offsetHeight = height + padding + border

IE5.0/5.5：

clientWidth = width - border

clientHeight = height - border

offsetWidth = width

offsetHeight = height(需要提一下：CSS中的margin属性，与clientWidth、offsetWidth、clientHeight、offsetHeight均无关)

网页可见区域宽： document.body.clientWidth;

网页可见区域高： document.body.clientHeight;

网页可见区域宽： document.body.offsetWidth (包括边线的宽);  
 网页可见区域高： document.body.offsetHeight (包括边线的宽);

网页正文全文宽： document.body.scrollWidth;  
 网页正文全文高： document.body.scrollHeight;

网页被卷去的高： document.body.scrollTop;  
 网页被卷去的左： document.body.scrollLeft;

网页正文部分上： window.screenTop;  
 网页正文部分左： window.screenLeft;

屏幕分辨率的高： window.screen.height;  
 屏幕分辨率的宽： window.screen.width;

屏幕可用工作区高度： window.screen.availHeight;  
 屏幕可用工作区宽度：window.screen.availWidth;

**51、**轮播  选项卡

**52、**居中问题:

      position:absolute;left:50%;margin-left:-(宽度的一半)  ;

**53、**点击 移入 移出js样例

    $(function(){

    $("#ms\_list>div").each(function(){

        $(this).click(function(){

            $("#ms\_list>div").addClass("m\_s\_l\_zi\_02").removeClass("m\_s\_l\_zi\_022");

            $(this).addClass("m\_s\_l\_zi\_022").removeClass("m\_s\_l\_zi\_02");

        }).mouseover(function(){

            $(this).addClass("m\_s\_l\_zi\_0222").removeClass("m\_s\_l\_zi\_02");

        }).mouseout(function(){

            $("#ms\_list>div").addClass("m\_s\_l\_zi\_02").removeClass("m\_s\_l\_zi\_0222");

        });

    });

});

**54、**

    /\*常规写法\*/

    .new\_main .content\_01 a:nth-child(3){color: #CC3300;behavior: url("htc/ie-css3.htc");}

    /\*IE写法\*/

    .new\_main .content\_01>a+a+a{color: #CC3300;behavior: url("htc/ie-css3.htc");}

**55、**对于一个div下多个<a>标签居中显示可以对父级div使用text-align:center;属性,使<a>标签均匀分布;

     可以用line-height:属性取消父级元素的高度节省代码;

**56、**iframe标签：

        iframe 元素会创建包含另外一个文档的内联框架（即行内框架）；

        iframe标签内的内容可以做为浏览器不支持iframe标签时显示；

**57、**ie8以下overflow：hidden失效；

一、给parent加：position:relative；

    二、给son加：position：relative；

**58、j**Query:slideDown();slideUp();

动画晃 动:$("#div").stop(true,false).slideDown/Up();

**59、**jQuery>attr:

        $(document).ready(function(){

            $("button").click(function(){

                $("#runoob").attr("href", function(i, origValue){

                    return origValue + "/jquery";

                });

            });

        });

        $("XXX").attr("属性名",fn(index,原属性值){});

        origValue:原属性值;

        attr()的回调函数两个参数:被选元素列表中当前元素的下标以及原始值;

**60、**font-size:的值也可以是100%等百分数;

**61、jQuery遍历查找:**

    $(document).ready(function(){ $("span").parent(); });

     $(document).ready(function(){ $("span").parents(); });

  $(document).ready(function(){ $("span").parents("ul"); });

    parentsUntil() 方法返回介于两个给定元素之间的所有祖先元素。

    下面的例子返回介于 <span> 与 <div> 元素之间的所有祖先元素：

    $(document).ready(function(){ $("span").parentsUntil("div"); });

    $(document).ready(function(){ $("div").children(); });

    $(document).ready(function(){ $("div").children("p.1"); });

    div后代的所有span元素:

    $(document).ready(function(){ $("div").find("span"); });

    div的所有后代:

    $(document).ready(function(){ $("div").find("\*"); });

    $(document).ready(function(){ $("h2").siblings(); });

    $(document).ready(function(){ $("h2").siblings("p"); });

    next():返回被选定元素的下一个同胞元素

    $(document).ready(function(){ $("h2").next(); });

    next():返回被选定元素的所有跟随的同胞元素

    $(document).ready(function(){ $("h2").nextAll(); });

    nextUntil() 方法返回介于两个给定参数之间的所有跟随的同胞元素。

    $(document).ready(function(){ $("h2").nextUntil("h6"); });

**62、**nginx代理打开html文件乱码：

    在nginx.conf文件里server里加default\_type 'text/html';charset utf-8;

**63、**文字两边对齐：text-align:justify;text-justify:inter-ideograph;

**64、**火狐浏览器中：左浮动  右浮动  要在左浮动div中加clear：left；

**65、**img之间有边距；img{display:block;}

**66、**解决子divmargin-top父div也跟着margin-top：

     给父div加：padding-top：1px；/float：left；/border:1px solied transparent;

**67、**子div float父div overflow:hidden；

**68、**去除select标签默认样式：

select{appearance:none;-moz-appearance:none;-webkit-appearance:none;}

去除下拉框：

select::-ms-expand {display: none;}

**69、**图片居中：display:table-cell;text-align:center;vertical-align:middle;

**70、**修改placeholder字体的颜色：

input::-webkit-input-placeholder{color: #D0C7B2;}

input::-moz-input-placeholder{color: #D0C7B2;}

input::-ms-input-placeholder{color: #D0C7B2;}

**71、**解决火狐浏览器float：left；显示错乱问题在顶层加overflow：hidden；

**72、**$(function(){  
 //解决IE11文本框无法选中，强制获取焦点  
 $("input[type='text']").eq(0).focus();  
 $("input[type='text']").eq(0).blur();  
 })

**73、** <button>

  <a href=""></a>

</button>

在ie中点击无效，请换成：

<a href="">

<button></button>

</a>

**74、**图片有display：block；外层 text-align:center;失效；

**75、**解决display:table-cell;和max-width:100%;不兼容问题（在IE火狐中）;

html:

<div class="out">

<div class="mid">

  <img src="" alt="" >

</div>

</div>

css:

.out{display: table;table-layout: fixed;}

.mid{display: table-cell;vertical-align: middle;text-align:center;font-size: 0;}

img{width:auto;max-width:100%;height:auto;max-height:100%;}

**76、**

首先，我们需要知道的是：我们可以对表格table设置margin，而不能设置padding;对单元格td设置padding，而不能设置margin。所以说，我们不能对单元格td设置margin属性来调整单元格与单元格之间的距离，但是我们可以使用一下方法来达到同样的效果。  
1) 设置padding  
就像刚才所说的那样，我们可以对单元格td设置padding属性来调整单元格之间的间隔  
td{padding:5px 10px;}  
2) 使用标签  
我们可以在单元格td内添加标签，然后对标签设置margin属性  
p{margin:5px 10px;}    
3) 对表格table设置border-spacing(IE不支持)  
　　先让我们简单了解一下border-spacing属性。border-spacing:length;设置单元格td的边在横向和纵向上的间距。当您指定一个length值时，这个值将作用于横向和纵向上的间距;当您指定两个length值时，第一个将作用于横向间距，第二个值将作用于纵向间距。  
table{border-spacing:5px 10px;}

## 前端开发学习什么？

学习HTML、CSS和DIV+CSS技术用来制作Web页面；学

vaScript、DOM、BOM等用建立开发基础；学习photoshop和Axure等软件应用，完成页面UI设计；也要认识一下NodeJS和PHP加上数据库等这样的后端语言，方便前后端开发配合；学习HTML5、CSS3、响应式页面布局、微网站制作等开发移动互联网的应用；像Ajax、JQuery、jQuery   
Mobile、Bootstrap、AngularJS等更是前端开发必学的高级技术；HTML5地理位置应用、离线应用、webworker多线程实现、websocket、跨平台开发技术和webAPP开发等，这些企业级应用技术也是现在开发的主流；现在前端开发最热门、要人最多，就是使用canvas开发网页游戏动画，以及会用Cocos2d-js制作游戏等开发。

**77、**蒙层的样式代码

width: 100%;height:100%;background-color: rgb(0, 0, 0);opacity: 0.45;position: fixed;top: 0;left: 0;z-index:999999;

78、居中问题：

①、

display: table-cell;

vertical-align: middle;

text-align: center;

注：IE在上层添加

display: table;

table-layout: fixed;

②、

position: absolute;

 left: 50%;

 top: 50%;

width:200px;

height:100px;

margin-left:-100px;

margin-top:-50px;

③、

display: flex;

  justify-content: center;//水平

  align-items: center;//垂直

**79、前端页面优化：**

1.减少http请求（合并css、合并JavaScript、合并图片）；

2.使用浏览器缓存（css、JavaScript、logo、图片等静态的更新频率低的）；

3.启用压缩（css、JavaScript、html），压缩图片大小；

4.css放在页面最上边，JavaScript放在页面最下边（Lazy Load Javascript）；

5.减少cookie传输；

6.JavaScript代码优化；

7.CDN加速；

8.反向代理；

**80、**

about input[type=radio and type=checkbox] style

input[type=radio],input[type=checkbox] {  
 display: inline-block;  
 vertical-align: middle;  
 width: 16px;  
 height: 16px;  
 margin-left: 5px;  
 -webkit-appearance: none;  
 background-color: transparent;  
 border: 0;  
 outline: 0 !important;  
 line-height: 20px;  
 color: #d8d8d8;  
 }  
 input[type=checkbox] {  
 background: @btnPrimaryBackground;  
 overflow: hidden;  
 border-radius: 3px;  
 }  
 input[type=radio]:after {  
 content: "";  
 display:block;  
 width: 16px;  
 height: 16px;  
 border-radius: 50%;  
 text-align: center;  
 line-height: 12px;  
 font-size: 12px;  
 color: #fff;  
 border: 1px solid #ddd;  
 background-color: #fff;  
 box-sizing:border-box;  
 }  
 input[type=checkbox]:after {  
 content: "";  
 display:block;  
 width: 16px;  
 height: 16px;  
 text-align: center;  
 line-height: 12px;  
 font-size: 12px;  
 color: #fff;  
 border: 1px solid #ddd;  
 background-color: #fff;  
 box-sizing:border-box;  
 border-radius: 2px;  
 }  
 input[type=checkbox]:checked:after {  
 border: 1px solid #ddd;  
 background-color: @btnPrimaryBackground;  
 }  
 input[type=radio]:checked:after {  
 content: "L";  
 transform:matrix(-0.766044,-0.642788,-0.642788,0.766044,0,0);  
 -webkit-transform:matrix(-0.766044,-0.642788,-0.642788,0.766044,0,0);  
 border-color: @btnPrimaryBackground;  
 background-color: @btnPrimaryBackground;  
 font-weight: bold;  
 }  
 input[type=checkbox]:checked:after {  
 content: "L";  
 transform:matrix(-0.766044,-0.642788,-0.642788,0.766044,0,0);  
 -webkit-transform:matrix(-0.766044,-0.642788,-0.642788,0.766044,0,0);  
 border-color: @btnPrimaryBackground;  
 background-color: @btnPrimaryBackground;  
 font-weight: bold;  
 }

81、meta标签用法总结：

声明文档使用的字符编码：  
 <meta charset="utf-8" />  
 声明文档的兼容模式：  
 <meta http-equiv="X-UA-Compatible" content="IE=Edge,chrome=1"/>

//指示IE以目前可用的最高模式显示内容  
 定义对页面的描述：  
 <meta name="description" content="hello html5" />  
 定义页面的最新版本：  
 <meta name="revised" content="Frank, 2017/5/20" />  
 SEO 优化  
 页面描述  
 <meta name="description" content="不超过850个字符"/>  
 定义针对搜索引擎的关键词：  
 <meta name="keywords" content="HTML5COL学院, HTML5, CSS3, JavaScript" />

定义网页作者：

<meta name="author" content="HTML5COL" />  
 定义网页搜索引擎索引方式，使用英文逗号「,」分隔，常有如下几种取值： none,noindex,nofollow,all,index和follow：  
 <meta name="robots" content="index,follow" />  
 为移动设备添加 viewport  
 <meta name="viewport" content="width=device-width, initial-scale=1.0" />  
 参数说明  
 content参数 释义   
 width viewport 宽度(数值/device-width)   
 height viewport 高度(数值/device-height)   
 initial-scale 初始缩放比例   
 maximum-scale 最大缩放比例   
 minimum-scale 最小缩放比例   
 user-scalable 是否允许用户缩放(yes/no)   
 ISO 设备  
 添加到主屏后的标题（iOS 6开始）：  
 <meta name="apple-mobile-web-app-title" content="标题" />  
 设置状态栏的背景颜色：  
 <meta name="apple-mobile-web-app-title" content="yes" />  
 是否启用 WebApp 全屏模式：  
 <meta name="apple-mobile-web-app-capable" content="yes" />  
 设置状态栏的背景颜色：  
 <meta name="apple-mobile-web-app-status-bar-style" content="black-translucent" />  
 注：只有在 "apple-mobile-web-app-capable" content="yes" 时生效  
 content 参数：  
 content参数 释义   
 default 默认值，网页内容从状态栏底部开始   
 black 状态栏背景是黑色，网页内容从状态栏底部开始   
 black-translucent 状态栏背景是黑色半透明，网页内容充满整个屏幕，顶部会被状态栏遮挡   
 Windows 8  
 Windows 8 磁贴颜色：  
 <meta name="msapplication-TileColor" content="#000" />  
 Windows 8 磁贴图标：  
 <meta name="msapplication-TileImage" content="icon.png" />  
 其他  
 禁止数字识自动别为电话号码：  
 <meta name="format-detection" content="telephone=no" />  
 不让android识别邮箱：  
 <meta name="format-detection" content="email=no" />  
 每 8 秒刷新一次页面：  
 <meta http-equiv="refresh" content="8" />