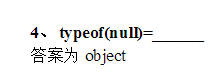
# web前端开发师笔试题

对ul li的样式设成无，应该是用什么属性\_\_\_\_\_\_\_\_。（6分）list-styl-type：none；

6、在新窗口打开链接的方法是\_\_\_\_\_\_\_\_。（4分）target=\_blank

7、Color:#666666;可缩写为\_\_\_\_\_\_\_\_。（2分）color：#666



**5**

**JS中给全部都是数字元素的数组排序的原生方法是\_\_\_sort\_\_\_\_\_\_\_**

**，其中使用的是**

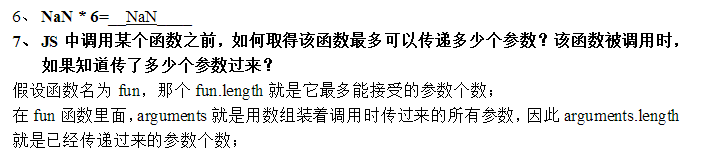
**冒泡排序方法。**

例如：

array.sort(function(a,b){return a-b;})

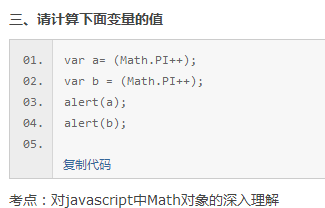
详细说明访问网站：

<http://www.admin5.com/article/20120718/445724.shtml>





颜色表达也可以缩写。#F50



输出全是3.141592653589793，没有改变，说明系统对象的属性是只读的。

<script>

var a = (++Math.PI);

alert(a); //1

alert(Math.PI); //2

var b = (Math.PI++);

alert(b); //3

alert(Math.PI); //4

var c = Math.PI = (++Math.PI);

alert(c); //5

alert(Math.PI); //6

var d = Math.PI = (Math.PI++);

alert(d); //7

alert(Math.PI); //8

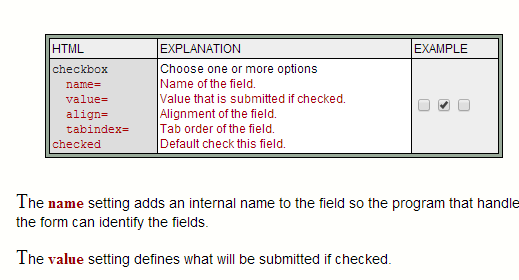
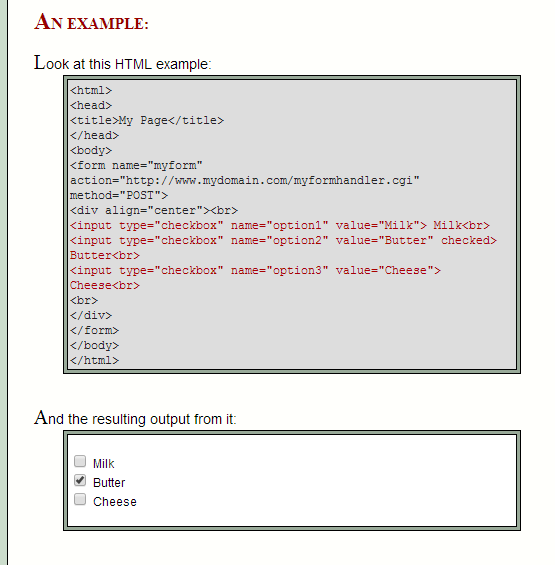
var e = Math.PI = (Math.PI + 1);

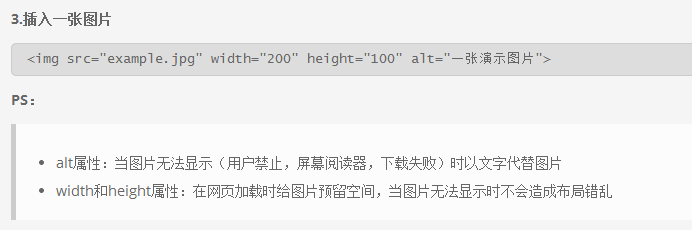
alert(e); //9

alert(Math.PI); //10

</script>

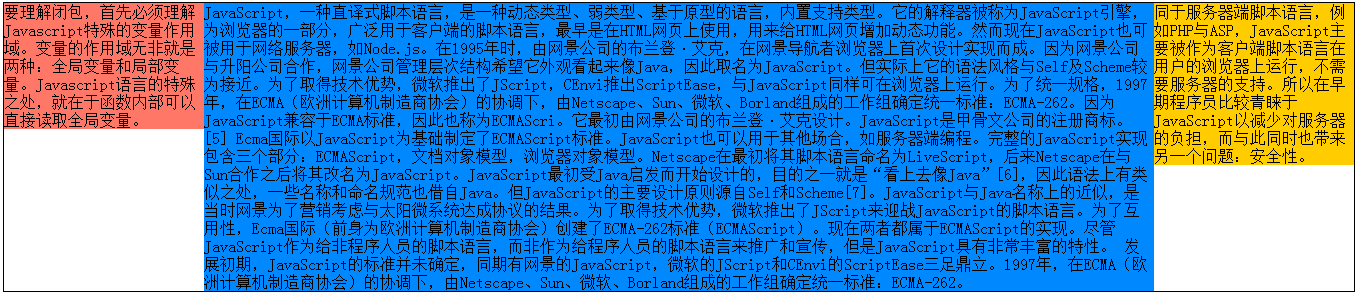




## CSS部分

# ****1.三个div并排，左右div为固宽200px，中间自适应****

[](http://www.atoctopus.com/images/post/interview/css01.png)CSS第一题两种方法实现：浮动和定位

|  |  |  |
| --- | --- | --- |
| 1  2  3  4  5 | <div id="box">  <div class="left">...</div>  <div class="right">...</div>  <div class="center">...</div>  </div> | |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44 | | /\* 方法一：浮动 \*/ #box {  border: 1px solid #000;  overflow: hidden;  }  .left {  float: left;  width: 200px;  background: red;  }  .right {  float: right;  width: 200px;  background: yellow;  }  .center {  margin: 0 200px;  background: blue;  } /\* 方法二：定位 \*/ #box {  position: relative;  border: 1px solid #000;  overflow: hidden;  }  .left {  position: absolute;  top: 0;  left: 0;  width: 200px;  background: red;  }  .right {  position: absolute;  top: 0;  right: 0;  width: 200px;  background: yellow;  }  .center {  margin: 0 200px;  background: blue;  } |

在线预览：[方法一](http://codepen.io/anon/pen/geixm/) [方法二](http://codepen.io/anon/pen/tAleE/)



**5.**清除浮动的方法

一般需要使用到浮动清除有两种情况：闭合浮动和清除浮动

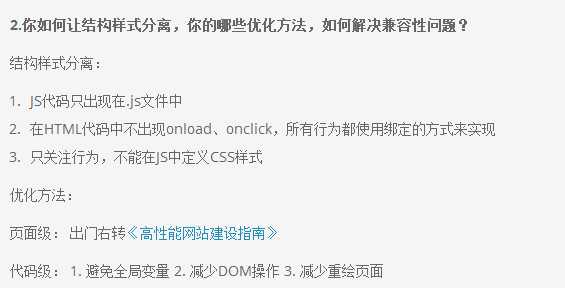
为什么要闭合浮动？

因为子元素浮动使自身脱离普通流，造成父素高度坍陷。

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46 | /\* 方法一：clearfix \*//\* 经典版 \*/ .clearfix:after {  content: ".";  display: block;  visibility: hidden;  height: 0;  clear: both;  }  .clearfix {  \*zoom: 1;  } /\* 精简版 \*/ .clearfix:after {  content: "\200B";  display: block;  height: 0;  clear: both;  }  .clearfix {  \*zoom: 1;  } /\* 真精简版 \*/ .clearfix:before, .clearfix:after {  content: "";  display: table;  }  .clearfix:after {  clear: both;  overflow: hidden;  }  .clearfix {  \*zoom: 1;  } /\* 方法二：overflow \*/ overflow: hidden | auto; /\* 方法三：额外标签 或 空标签 \*/ <div style="clear: both"></div> 或 <br clear="all"> /\* 方法四：父元素浮动 \*/ 父元素添加float: left | right; /\* 方法五：父元素添加display: inline-block \*/ 父元素添加display: table | inline-block; |

# px、em和百分比%之间的区别

* px相对于特定设备的屏幕分辨率
* em相对于容器元素（或者父元素）的字体大小。例如浏览器默认字体16px，那么1em等于16px，14px等于0.875em
* 百分比字体将在容器元素（或者父元素）字体大小的基础上发生变化。例如：设为一个值为120%，将会使当前元素中的字体在其上层元素字体大小的基础上增加20%





# 常见兼容性问题？

* \* png24位的图片在iE6浏览器上出现背景，解决方案是做成PNG8.
* \* 浏览器默认的margin和padding不同。解决方案是加一个全局的\*{margin:0;padding:0;}来统一。
* \* IE6双边距bug: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。

浮动ie产生的双倍距离 #box{ float:left; width:10px; margin:0 0 0 100px;}

这种情况之下IE会产生20px的距离，解决方案是在float的标签样式控制中加入 ——\_display:inline;将其转化为行内属性。(\_这个符号只有ie6会识别)

渐进识别的方式，从总体中逐渐排除局部。

首先，巧妙的使用“\9”这一标记，将IE游览器从所有情况中分离出来。

接着，再次使用“+”将IE8和IE7、IE6分离开来，这样IE8已经独立识别。

* css

.bb{

background-color:#f1ee18;/\*所有识别\*/

.background-color:#00deff\9; /\*IE6、7、8识别\*/

+background-color:#a200ff;/\*IE6、7识别\*/

\_background-color:#1e0bd1;/\*IE6识别\*/

}

* \* IE下,可以使用获取常规属性的方法来获取自定义属性,

也可以使用getAttribute()获取自定义属性;

Firefox下,只能使用getAttribute()获取自定义属性.

解决方法:统一通过getAttribute()获取自定义属性.

* \* IE下,even对象有x,y属性,但是没有pageX,pageY属性;

Firefox下,event对象有pageX,pageY属性,但是没有x,y属性.

* \* 解决方法：（条件注释）缺点是在IE浏览器下可能会增加额外的HTTP请求数。
* \* Chrome 中文界面下默认会将小于 12px 的文本强制按照 12px 显示,

可通过加入 CSS 属性 -webkit-text-size-adjust: none; 解决.

* 超链接访问过后hover样式就不出现了 被点击访问过的超链接样式不在具有hover和active了解决方法是改变CSS属性的排列顺序:

L-V-H-A : a:link {} a:visited {} a:hover {} a:active {}

# html5有哪些新特性、移除了那些元素？

# 如何处理HTML5新标签的浏览器兼容问题？如何区分 HTML 和 HTML5？

\* HTML5 现在已经不是 SGML 的子集，主要是关于图像，位置，存储，多任务等功能的增加。

### \* 绘画 canvas

用于媒介回放的 video 和 audio 元素

本地离线存储 localStorage长期存储数据，浏览器关闭后数据不丢失；

### sessionStorage 的数据在浏览器关闭后自动删除

### 语意化更好的内容元素，比如 article、footer、header、nav、section

### 表单控件，calendar、date、time、email、url、search

新的技术webworker, websockt, Geolocation

\* 移除的元素

纯表现的元素：basefont，big，center，font, s，strike，tt，u；

对可用性产生负面影响的元素：frame，frameset，noframes；

## 支持HTML5新标签：

### \* IE8/IE7/IE6支持通过document.createElement方法产生的标签，

### 可以利用这一特性让这些浏览器支持HTML5新标签，

浏览器支持新标签后，还需要添加标签默认的样式：

\* 当然最好的方式是直接使用成熟的框架、使用最多的是html5shim框架

<!--[if lt IE 9]>

<script> src="http://html5shim.googlecode.com/svn/trunk/html5.js"</script>

<![endif]-->

如何区分： DOCTYPE声明\新增的结构元素\功能元素

# 请描述一下 cookies，sessionStorage 和 localStorage 的区别？

* cookie在浏览器和服务器间来回传递。 sessionStorage和localStorage不会
* sessionStorage和localStorage的存储空间更大；
* sessionStorage和localStorage有更多丰富易用的接口；

sessionStorage和localStorage各自独立的存储空间；

# 介绍一下CSS的盒子模型？

* （1）有两种， IE 盒子模型、标准 W3C 盒子模型；IE的content部分包含了 border 和 pading;

（2）盒模型： 内容(content)、填充(padding)、边界(margin)、 边框(border).

# CSS 选择符有哪些？

* \* 1.id选择器（ # myid）
* 2.类选择器（.myclassname）
* 3.标签选择器（div, h1, p）
* 4.相邻选择器（h1 + p）
* 5.子选择器（ul < li）
* 6.后代选择器（li a）
* 7.通配符选择器（ \* ）
* 8.属性选择器（a[rel = "external"]）
* 9.伪类选择器（a: hover, li: nth - child）

# 哪些属性可以继承？

* \* 可继承的样式： font-size font-family color, UL LI DL DD DT;
* \* 不可继承的样式：border padding margin width height ;
* \* 优先级就近原则，同权重情况下样式定义最近者为准;
* \* 载入样式以最后载入的定位为准;

# 优先级算法如何计算？

优先级为:

!important > id > class > tag

important 比 内联优先级高

# CSS3新增伪类有那些？

p:first-of-type 选择属于其父元素的首个 <p> 元素的每个 <p> 元素。

p:last-of-type 选择属于其父元素的最后 <p> 元素的每个 <p> 元素。

p:only-of-type 选择属于其父元素唯一的 <p> 元素的每个 <p> 元素。

p:only-child 选择属于其父元素的唯一子元素的每个 <p> 元素。

p:nth-child(2) 选择属于其父元素的第二个子元素的每个 <p> 元素。

:enabled :disabled 控制表单控件的禁用状态。

:checked 单选框或复选框被选中。

# 如何居中div？

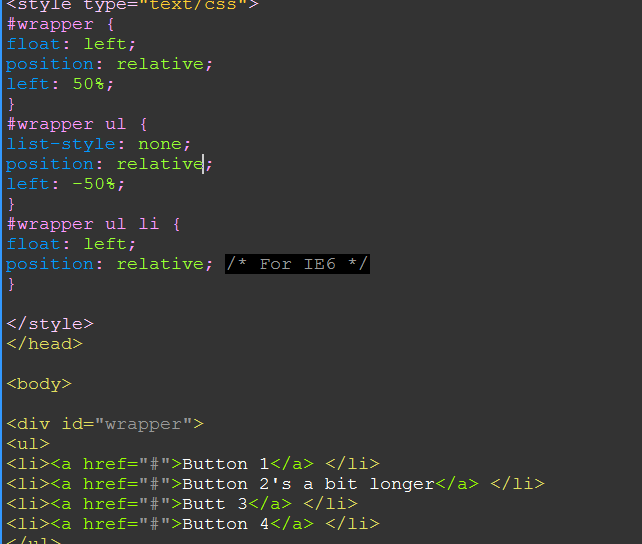
### 给div设置一个宽度，然后添加margin:0 auto属性

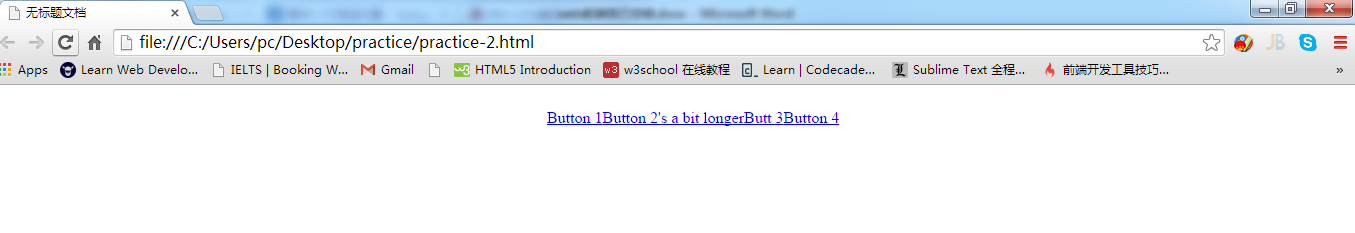
* + div{
  + width:200px;
  + margin:0 auto;
  + }

# 如何居中一个浮动元素？

* + 确定容器的宽高 宽500 高 300 的层
  + 设置层的外边距
  + .div {
  + Width:500px ; height:300px;//高度可以不设
  + Margin: -150px 0 0 -250px;
  + position:relative;相对定位
  + background-color:pink;//方便看效果
  + left:50%;
  + top:50%;

}





# 列出display的值，说明他们的作用。

1.

block 象块类型元素一样显示。

none 缺省值。象行内元素类型一样显示。

inline-block 象行内元素一样显示，但其内容象块类型元素一样显示。

list-item 象块类型元素一样显示，并添加样式列表标记。

# position的值， relative和absolute定位原点是？

2.

\*absolute

生成绝对定位的元素，相对于 static 定位以外的第一个父元素进行定位。

\*fixed （老IE不支持）

生成绝对定位的元素，相对于浏览器窗口进行定位。

\*relative

生成相对定位的元素，相对于其正常位置进行定位。

\* static 默认值。没有定位，元素出现在正常的流中

\*（忽略 top, bottom, left, right z-index 声明）。

\* inherit 规定从父元素继承 position 属性的值。

# 为什么要初始化CSS样式。

- 因为浏览器的兼容问题，不同浏览器对有些标签的默认值是不同的，如果没对CSS初始化往往会出现浏览器之间的页面显示差异。

- 当然，初始化样式会对SEO有一定的影响，但鱼和熊掌不可兼得，但力求影响最小的情况下初始化。

\*最简单的初始化方法就是： \* {padding: 0; margin: 0;} （不建议）

淘宝的样式初始化：

body, h1, h2, h3, h4, h5, h6, hr, p, blockquote, dl, dt, dd, ul, ol, li, pre, form, fieldset, legend, button, input, textarea, th, td { margin:0; padding:0; }

body, button, input, select, textarea { font:12px/1.5tahoma, arial, \5b8b\4f53; }

h1, h2, h3, h4, h5, h6{ font-size:100%; }

address, cite, dfn, em, var { font-style:normal; }

code, kbd, pre, samp { font-family:couriernew, courier, monospace; }

small{ font-size:12px; }

ul, ol { list-style:none; }

a { text-decoration:none; }

a:hover { text-decoration:underline; }

sup { vertical-align:text-top; }

sub{ vertical-align:text-bottom; }

legend { color:#000; }

fieldset, img { border:0; }

button, input, select, textarea { font-size:100%; }

table { border-collapse:collapse; border-spacing:0; }

# css定义的权重

以下是权重的规则：标签的权重为1，class的权重为10，id的权重为100，以下例子是演示各种定义的权重值：

/\*权重为1\*/

div{

}

/\*权重为10\*/

.class1{

}

/\*权重为100\*/

#id1{

}

/\*权重为100+1=101\*/

#id1 div{

}

/\*权重为10+1=11\*/

.class1 div{

}

/\*权重为10+10+1=21\*/

.class1 .class2 div{

}

如果权重相同，则最后定义的样式会起作用，但是应该避免这种情况出现

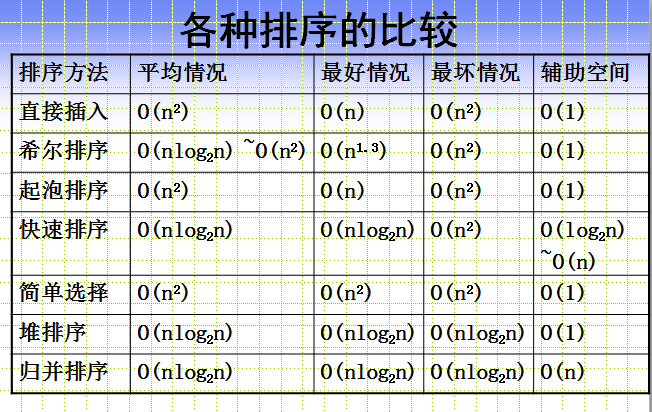
# 如果需要手动写动画，你认为最小时间间隔是多久，为什么？（阿里）

* 多数显示器默认频率是60Hz，即1秒刷新60次，所以理论上最小间隔为1/60＊1000ms ＝ 16.7ms

# display:inline-block 什么时候会显示间隙？(携程)

移除空格、使用margin负值、使用font-size:0、letter-spacing、word-spacing

# 常见排序算法的时间复杂度,空间复杂度

[](https://github.com/qiu-deqing/FE-interview/blob/master/img/sort-compare.png)

# <img>的title和alt有什么区别

1. title是[global attributes](http://www.w3.org/TR/html-markup/global-attributes.html#common.attrs.core)之一，用于为元素提供附加的advisory information。通常当鼠标滑动到元素上的时候显示。
2. alt是<img>的特有属性，是图片内容的等价描述，用于图片无法加载时显示、读屏器阅读图片。可提图片高可访问性，除了纯装饰图片外都必须设置有意义的值，搜索引擎会重点分析。

# 什么是web语义化,有什么好处

web语义化是指通过HTML标记表示页面包含的信息，包含了HTML标签的语义化和css命名的语义化。  
HTML标签的语义化是指：通过使用包含语义的标签（如h1-h6）恰当地表示文档结构  
css命名的语义化是指：为html标签添加有意义的class，id补充未表达的语义，如[Microformat](http://en.wikipedia.org/wiki/Microformats)通过添加符合规则的class描述信息  
为什么需要语义化：

* 去掉样式后页面呈现清晰的结构
* 盲人使用读屏器更好地阅读
* 搜索引擎更好地理解页面，有利于收录
* 便团队项目的可持续运作及维护

# css sprite是什么,有什么优缺点

概念：将多个小图片拼接到一个图片中。通过background-position和元素尺寸调节需要显示的背景图案。

优点：

1. 减少HTTP请求数，极大地提高页面加载速度
2. 增加图片信息重复度，提高压缩比，减少图片大小
3. 更换风格方便，只需在一张或几张图片上修改颜色或样式即可实现

缺点：

1. 图片合并麻烦
2. 维护麻烦，修改一个图片可能需要从新布局整个图片，样式

# display: block;和display: inline;的区别

block元素特点：

1.处于常规流中时，如果width没有设置，会自动填充满父容器  
2.可以应用margin/padding  
3.在没有设置高度的情况下会扩展高度以包含常规流中的子元素  
4.处于常规流中时布局时在前后元素位置之间（独占一个水平空间）  
5.忽略vertical-align

inline元素特点

1.水平方向上根据direction依次布局  
2.不会在元素前后进行换行  
3.受white-space控制  
4.margin/padding在竖直方向上无效，水平方向上有效  
5.width/height属性对非替换行内元素无效，宽度由元素内容决定  
6.非替换行内元素的行框高由line-height确定，替换行内元素的行框高由height,margin,padding,border决定  
6.浮动或绝对定位时会转换为block  
7.vertical-align属性生效

# PNG,GIF,JPG的区别及如何选

参考资料： [选择正确的图片格式](http://www.yuiblog.com/blog/2008/11/04/imageopt-2/)  
**GIF**:

1. 8位像素，256色
2. 无损压缩
3. 支持简单动画
4. 支持boolean透明
5. 适合简单动画

**JPEG**：

1. 颜色限于256
2. 有损压缩
3. 可控制压缩质量
4. 不支持透明
5. 适合照片

**PNG**：

1. 有PNG8和truecolor PNG
2. PNG8类似GIF颜色上限为256，文件小，支持alpha透明度，无动画
3. 适合图标、背景、按钮

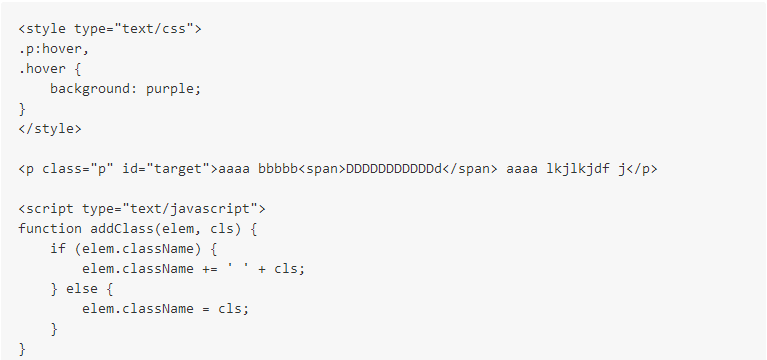
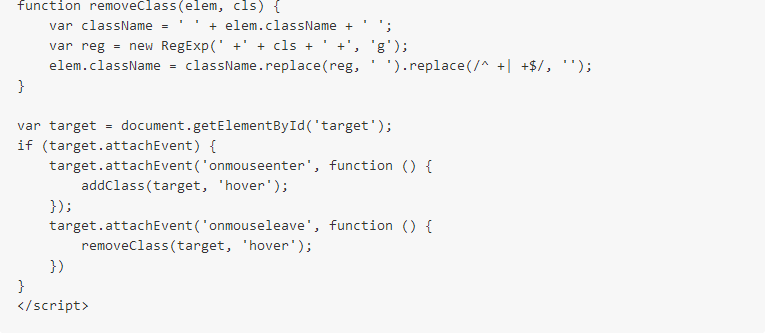
# CSS有哪些继承属性

* 关于文字排版的属性如：
  + [font](https://developer.mozilla.org/en-US/docs/Web/CSS/font)
  + [word-break](https://developer.mozilla.org/en-US/docs/Web/CSS/word-break)
  + [letter-spacing](https://developer.mozilla.org/en-US/docs/Web/CSS/letter-spacing)
  + [text-align](https://developer.mozilla.org/en-US/docs/Web/CSS/text-align)
  + [text-rendering](https://developer.mozilla.org/en-US/docs/Web/CSS/text-rendering)
  + [word-spacing](https://developer.mozilla.org/en-US/docs/Web/CSS/word-spacing)
  + [white-space](https://developer.mozilla.org/en-US/docs/Web/CSS/white-space)
  + [text-indent](https://developer.mozilla.org/en-US/docs/Web/CSS/text-indent)
  + [text-transform](https://developer.mozilla.org/en-US/docs/Web/CSS/text-transform)
  + [text-shadow](https://developer.mozilla.org/en-US/docs/Web/CSS/text-shadow)
* [line-height](https://developer.mozilla.org/en-US/docs/Web/CSS/line-height)
* [color](https://developer.mozilla.org/en-US/docs/Web/CSS/color)
* [visibility](https://developer.mozilla.org/en-US/docs/Web/CSS/visibility)
* [cursor](https://developer.mozilla.org/en-US/docs/Web/CSS/cursor)

# IE6浏览器有哪些常见的bug,缺陷或者与标准不一致的地方,如何解决

* ol内li的序号全为1，不递增。解决方法：为li设置样式display: list-item;
* 未定位父元素overflow: auto;，包含position: relative;子元素，子元素高于父元素时会溢出。解决办法：1）子元素去掉position: relative;; 2）不能为子元素去掉定位时，父元素position: relative;



* IE6只支持a标签的:hover伪类，解决方法：使用js为元素监听mouseenter，mouseleave事件，添加类实现效果：
*  

# IE5-8不支持opacity，解决办法：

.opacity {

opacity: 0.4

filter: alpha(opacity=60); /\* for IE5-7 \*/

-ms-filter: "progid:DXImageTransform.Microsoft.Alpha(Opacity=60)"; /\* for IE 8\*/

}

* IE6在设置height小于font-size时高度值为font-size，解决办法：font-size: 0;
* IE6不支持PNG透明背景，解决办法: **IE6下使用gif图片**

## IE6-7不支持display: inline-block解决办法：设置inline并触发hasLayout

display: inline-block;

\*display: inline;

\*zoom: 1;

* IE6下浮动元素在浮动方向上与父元素边界接触元素的外边距会加倍。
* 解决办法：  
  1）使用padding控制间距。  
  2）浮动元素display: inline;这样解决问题且无任何副作用：css标准规定浮动元素display:inline会自动调整为block

## 通过为块级元素设置宽度和左右margin为auto时，IE6不能实现水平居中，解决方法：为父元素设置text-align: center;

# 容器包含若干浮动元素时如何清理(包含)浮动

1. 容器元素闭合标签前添加额外元素并设置clear: both
2. 父元素触发块级格式化上下文(见块级可视化上下文部分)
3. 设置容器元素伪元素进行清理[推荐的清理浮动方法](http://nicolasgallagher.com/micro-clearfix-hack/)

/\*\*

\* 在标准浏览器下使用

\* 1 content内容为空格用于修复opera下文档中出现

\* contenteditable属性时在清理浮动元素上下的空白

\* 2 使用display使用table而不是block：可以防止容器和

\* 子元素top-margin折叠,这样能使清理效果与BFC，IE6/7

\* zoom: 1;一致

\*\*/

.clearfix:before,

.clearfix:after {

content: " "; /\* 1 \*/

display: table; /\* 2 \*/

}

.clearfix:after {

clear: both;

}

/\*\*

\* IE 6/7下使用

\* 通过触发hasLayout实现包含浮动

\*\*/

.clearfix {

\*zoom: 1;

}

# 如何创建块级格式化上下文(block formatting context),BFC有什么用

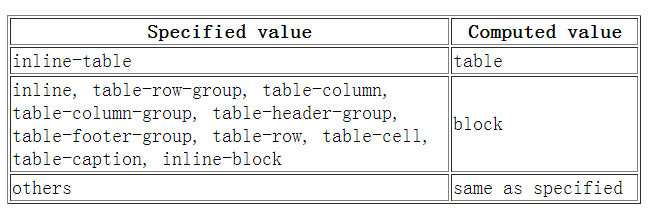
创建规则：

1. 根元素
2. 浮动元素（float不是none）
3. 绝对定位元素（position取值为absolute或fixed）
4. display取值为inline-block,table-cell, table-caption,flex, inline-flex之一的元素
5. overflow不是visible的元素

作用：

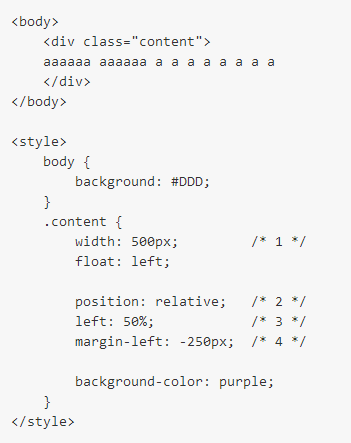
1. 可以包含浮动元素
2. 不被浮动元素覆盖
3. 阻止父子元素的margin折叠

# display,float,position的关系

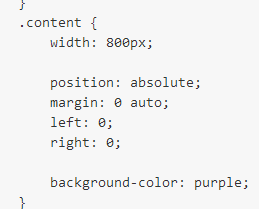
1. 如果display为none，那么position和float都不起作用，这种情况下元素不产生框
2. 否则，如果position值为absolute或者fixed，框就是绝对定位的，float的计算值为none，display根据下面的表格进行调整。
3. 否则，如果float不是none，框是浮动的，display根据下表进行调整
4. 否则，如果元素是根元素，display根据下表进行调整
5. 其他情况下display的值为指定值  
   总结起来：**绝对定位、浮动、根元素都需要调整display** [](https://github.com/qiu-deqing/FE-interview/blob/master/img/display-adjust.png)

# 如何水平居中一个元素

* **如果需要居中的元素为常规流中inline元素，为父元素设置text-align: center;即可实现**
* **如果需要居中的元素为常规流中block元素，1）为元素设置宽度，2）设置左右margin为auto。3）IE6下需在父元素上设置text-align: center;,再给子元素恢复需要的值**
* **如果需要居中的元素为浮动元素，1）为元素设置宽度，2）position: relative;，3）浮动方向偏移量（left或者right）设置为50%，4）浮动方向上的margin设置为元素宽度一半乘以-1**



* 如果需要居中的元素为**绝对定位元素**，1）为元素设置宽度，2）偏移量设置为50%，3）偏移方向外边距设置为元素宽度一半乘以-1
* 如果需要居中的元素为**绝对定位元素**，1）为元素设置宽度，2）设置左右偏移量都为0,3）设置左右外边距都为auto



# [8种CSS实现垂直居中水平居中的绝对定位居中技术](http://blog.csdn.net/freshlover/article/details/11579669)





# 父容器使用相对定位，子元素使用绝对定位后，这样子元素的位置不再相对于浏览器左上角，而是相对于父容器左上角

# .相对定位和绝对定位需要配合top、right、bottom、left使用来定位具体位置，这四个属性只有在该元素使用定位后才生效，其它情况下无效。另外这四个属性同时只能使用相邻的两个，不能即使用上又使用下，或即使用左，又使用右