****1、常用那几种浏览器测试？有哪些内核(Layout Engine)?****

　　答：

        (1) 浏览器：IE，Chrome，FireFox，Safari，Opera。

　　   (2) 内核：Trident，Gecko，Presto，Webkit。

****2、 说下行内元素和块级元素的区别？行内块元素的兼容性使用？（IE8 以下）****

答：

　　(1) 行内元素：会在水平方向排列，不能包含块级元素，设置width无效，height无效(可以设置line-height)，margin上下无效，padding上下无效。

　　块级元素：各占据一行，垂直方向排列。从新行开始结束接着一个断行。

　　(2) 兼容性：display:inline-block;\*display:inline;\*zoom:1;

1. **谈谈浮动和清除浮动，**清除浮动有哪些方式？比较好的方式是哪一种？****

浮动的框可以向左或向右移动，直到他的外边缘碰到包含框或另一个浮动框的边框为止。由于浮动框不在文档的普通流中，所以文档的普通流的块框表现得就像浮动框不存在一样。浮动的块框会漂浮在文档普通流的块框上。

　 答：

        (一)

　　       （1）父级div定义height。

　　       （2）结尾处加空div标签clear:both。

　　       （3）父级div定义伪类:after和zoom。

　　       （4）父级div定义overflow:hidden。

　　       （5）父级div定义overflow:auto。

　　       （6）父级div也浮动，需要定义宽度。

　　       （7）父级div定义display:table。

　　       （8）结尾处加br标签clear:both。

　　 (二) 比较好的是第3种方式，好多网站都这么用。

****4、box-sizing常用的属性有哪些？分别有什么作用？****

答：

　　(1)box-sizing: content-box|border-box|inherit;

　　(2)content-box:宽度和高度分别应用到元素的内容框。在宽度和高度之外绘制元素的内边距和边框(元素默认效果)。

border-box:元素指定的任何内边距和边框都将在已设定的宽度和高度内进行绘制。通过从已设定的宽度和高度分别减去边框和内边距才能得到内容的宽度和高度。

****5、Doctype作用？标准模式与兼容模式各有什么区别?****

　　答：

(1) 告知浏览器的解析器用什么文档标准解析这个文档。DOCTYPE不存在或格式不正确会导致文档以兼容模式呈现。

(2) 标准模式的排版和JS运作模式都是以该浏览器支持的最高标准运行。在兼容模式中，页面以宽松的向后兼容的方式显示,模拟老式浏览器的行为以防止站点无法工作。

****6、HTML5 为什么只需要写<!DOCTYPE html> ？****

答：

　　HTML5不基于 SGML，因此不需要对DTD进行引用，但是需要doctype来规范浏览器的行为（让浏览器按照它们应该的方式来运行）。

而HTML4.01基于SGML,所以需要对DTD进行引用，才能告知浏览器文档所使用的文档类型。

****7、 页面导入样式时，使用link和@import有什么区别？****

　　答：

      （1）link属于XHTML标签，除了加载CSS外，还能用于定义RSS, 定义rel连接属性等作用；而@import是CSS提供的，只能用于加载CSS;

　　（2）页面被加载的时，link会同时被加载，而@import引用的CSS会等到页面被加载完再加载;

　　（3）import是CSS2.1 提出的，只在IE5以上才能被识别，而link是XHTML标签，无兼容问题。

　引入方式3种：行内添加定义style属性值，页面头部内内嵌调用和外链调用，

区别：

1.link是xhtml标签，除了加载css外，还可以定义RSS等其他事务，@import只能加载CSS

2.link引用CSS时候，页面载入的时候同时加载，@import需要页面网页完全载入后加载

3.link是XHTML标签，没有兼容问题，@import是在CSS2.1提出的，低版本的浏览器不支持。

4.link支持使用javascript控制DOM去改变样式，但是@import不支持。

****8、介绍一下你对浏览器内核的理解？****

　　答：

主要分成两部分：渲染引擎(layout engineer或Rendering Engine)和JS引擎。

　　渲染引擎：负责取得网页的内容（HTML、XML、图像等等）、整理讯息（例如加入CSS等），以及计算网页的显示方式，然后会输出至显示器或打印机。浏览器的内核的不同对于网页的语法解释会有不同，所以渲染的效果也不相同。所有网页浏览器、电子邮件客户端以及其它需要编辑、显示网络内容的应用程序都需要内核。

　　JS引擎则：解析和执行javascript来实现网页的动态效果。

　　最开始渲染引擎和JS引擎并没有区分的很明确，后来JS引擎越来越独立，内核就倾向于只指渲染引擎。

****9、html5有哪些新特性？如何处理HTML5新标签的浏览器兼容问题？如何区分 HTML 和 HTML5？****

答：

    (一)

　　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。

　　(1)绘画 canvas;

　　(2)用于媒介回放的 video 和 audio 元素;

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

　　(4)sessionStorage 的数据在浏览器关闭后自动删除;

　　(5)语意化更好的内容元素，比如 article、footer、header、nav、section;

　　(6)表单控件，calendar、date、time、email、url、search;

(7)新的技术webworker, websocket, Geolocation;

　(二)

　　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。

当然也可以直接使用成熟的框架、比如html5shim，

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，

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

****10、简述一下你对HTML语义化的理解？****

答：

用正确的标签做正确的事情。

　　html语义化让页面的内容结构化，结构更清晰，便于对浏览器、搜索引擎解析;

　　即使在没有样式CSS情况下也以一种文档格式显示，并且是容易阅读的;

　　搜索引擎的爬虫也依赖于HTML标记来确定上下文和各个关键字的权重，利于SEO;

　　使阅读源代码的人对网站更容易将网站分块，便于阅读维护理解。

#### **11、position的值， relative和absolute分别是相对于谁进行定**位的？

absolute :生成绝对定位的元素， 相对于最近一级的 定位不是 static 的父元素来进行定位。

fixed （老IE不支持）生成绝对定位的元素，通常相对于浏览器窗口或 frame 进行定位。

relative 生成相对定位的元素，相对于其在普通流中的位置进行定位。

static 默认值。没有定位，元素出现在正常的流中

sticky 生成粘性定位的元素，容器的位置根据正常文档流计算得出

****12、 什么叫优雅降级和渐进增强？****

答：

　　优雅降级：Web站点在所有新式浏览器中都能正常工作，如果用户使用的是老式浏览器，则代码会检查以确认它们是否能正常工作。由于IE独特的盒模型布局问题，针对不同版本的IE的hack实践过优雅降级了,为那些无法支持功能的浏览器增加候选方案，使之在旧式浏览器上以某种形式降级体验却不至于完全失效。

　　渐进增强：从被所有浏览器支持的基本功能开始，逐步地添加那些只有新式浏览器才支持的功能,向页面增加无害于基础浏览器的额外样式和功能的。当浏览器支持时，它们会自动地呈现出来并发挥作用。

**13、display:none和visibility:hidden的区别？**

display:none 隐藏对应的元素，在文档布局中不再给它分配空间，它各边的元素会合拢，就当他从来不存在。

visibility:hidden 隐藏对应的元素，但是在文档布局中仍保留原来的空间。

CSS中link 和@import的区别是？

(1) link属于HTML标签，而@import是CSS提供的;

(2) 页面被加载的时，link会同时被加载，而@import被引用的CSS会等到引用它的CSS文件被加载完再加载;

(3) import只在IE5以上才能识别，而link是HTML标签，无兼容问题;

(4) link方式的样式的权重 高于@import的权重.

**14、position:absolute和float属性的异同**

共同点：对内联元素设置float和absolute属性，可以让元素脱离文档流，并且可以设置其宽高。

不同点：float仍会占据位置，absolute会覆盖文档流中的其他元素。

**15、介绍一下box-sizing属性？**

box-sizing属性主要用来控制元素的盒模型的解析模式。默认值是content-box。

content-box：让元素维持W3C的标准盒模型。元素的宽度/高度由border + padding + content的宽度/高度决定，设置width/height属性指的是content部分的宽/高

border-box：让元素维持IE传统盒模型（IE6以下版本和IE6~7的怪异模式）。设置width/height属性指的是border + padding + content

标准浏览器下，按照W3C规范对盒模型解析，一旦修改了元素的边框或内距，就会影响元素的盒子尺寸，就不得不重新计算元素的盒子尺寸，从而影响整个页面的布局。

**16、CSS 选择符有哪些？哪些属性可以继承？优先级[算法](http://lib.csdn.net/base/datastructure" \o "算法与数据结构知识库" \t "http://blog.csdn.net/kebi007/article/details/_blank)如何计算？ [CSS3](http://lib.csdn.net/base/css3" \o "CSS3知识库" \t "http://blog.csdn.net/kebi007/article/details/_blank)新增伪类有那些？**

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）

**17、CSS3有哪些新特性？**

CSS3实现圆角（border-radius），阴影（box-shadow），

对文字加特效（text-shadow、），线性渐变（gradient），旋转（transform）

transform:rotate(9deg) scale(0.85,0.90) translate(0px,-30px) skew(-9deg,0deg);//旋转,缩放,定位,倾斜

增加了更多的CSS选择器 多背景 rgba

在CSS3中唯一引入的伪元素是::selection.

媒体查询，多栏布局

border-image

CSS3中新增了一种盒模型计算方式：box-sizing。盒模型默认的值是content-box, 新增的值是padding-box和border-box，几种盒模型计算元素宽高的区别如下：

#### content-box（默认）

布局所占宽度Width：

Width = width + padding-left + padding-right + border-left + border-right

布局所占高度Height:

Height = height + padding-top + padding-bottom + border-top + border-bottom

#### padding-box

布局所占宽度Width：

Width = width(包含padding-left + padding-right) + border-top + border-bottom

布局所占高度Height:

Height = height(包含padding-top + padding-bottom) + border-top + border-bottom

#### 

#### border-box

布局所占宽度Width：

Width = width(包含padding-left + padding-right + border-left + border-right)

布局所占高度Height:

Height = height(包含padding-top + padding-bottom + border-top + border-bottom)

**18、对BFC规范的理解？**

BFC，块级格式化上下文，一个创建了新的BFC的盒子是独立布局的，盒子里面的子元素的样式不会影响到外面的元素。在同一个BFC中的两个毗邻的块级盒在垂直方向（和布局方向有关系）的margin会发生折叠。

（W3C CSS 2.1 规范中的一个概念，它决定了元素如何对其内容进行布局，以及与其他元素的关系和相互作用。

#### **19、说说你对语义化的理解？**

（1）去掉或者丢失样式的时候能够让页面呈现出清晰的结构

（2）有利于SEO：和搜索引擎建立良好沟通，有助于爬虫抓取更多的有效信息：爬虫依赖于标签来确定上下文和各个关键字的权重；

（3）方便其他设备解析（如屏幕阅读器、盲人阅读器、移动设备）以意义的方式来渲染网页；

（4）便于团队开发和维护，语义化更具可读性，是下一步吧网页的重要动向，遵循W3C标准的团队都遵循这个标准，可以减少差异化。

#### **20、Doctype作用? 严格模式与混杂模式如何区分？它们有何意义?**

（1）<!DOCTYPE> 声明位于文档中的最前面，处于 <html> 标签之前。告知浏览器以何种模式来渲染文档。

（2）严格模式的排版和 JS 运作模式是 以该浏览器支持的最高标准运行。

（3）在混杂模式中，页面以宽松的向后兼容的方式显示。模拟老式浏览器的行为以防止站点无法工作。

（4）DOCTYPE不存在或格式不正确会导致文档以混杂模式呈现。

#### **21、你知道多少种Doctype文档类型？**

该标签可声明三种 DTD 类型，分别表示严格版本、过渡版本以及基于框架的 HTML 文档。

HTML 4.01 规定了三种文档类型：Strict、Transitional 以及 Frameset。

XHTML 1.0 规定了三种 XML 文档类型：Strict、Transitional 以及 Frameset。

Standards （标准）模式（也就是严格呈现模式）用于呈现遵循最新标准的网页，而 Quirks

（包容）模式（也就是松散呈现模式或者兼容模式）用于呈现为传统浏览器而设计的网页。

#### **22、HTML与XHTML——二者有什么区别**

区别：

（1）所有的标记都必须要有一个相应的结束标记

（2）所有标签的元素和属性的名字都必须使用小写

（3）所有的XML标记都必须合理嵌套

（4）所有的属性必须用引号""括起来

（5）把所有<和&特殊符号用编码表示

（6）给所有属性赋一个值

（7）不要在注释内容中使“--”

1. 图片必须有说明文字

#### **23、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.也可以引用一段脚本处理.

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（IE6双边距问题：在IE6下，如果对元素设置了浮动，同时又设置了margin-left或margin-right，margin值会加倍。）

#box{ float:left; width:10px; margin:0 0 0 10px;}

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\*/

}

怪异模式问题：漏写DTD声明，Firefox仍然会按照标准模式来解析网页，但在IE中会触发

怪异模式。为避免怪异模式给我们带来不必要的麻烦，最好养成书写DTD声明的好习惯。现在

可以使用[html5](http://www.w3.org/TR/html5/single-page.html)推荐的写法：`<doctype html>`

上下margin重合问题

ie和ff都存在，相邻的两个div的margin-left和margin-right不会重合，但是margin-top和margin-bottom却会发生重合。

解决方法，养成良好的代码编写习惯，同时采用margin-top或者同时采用margin-bottom。

**24、ie各版本和chrome可以并行下载多少个资源**IE6 两个并发，iE7升级之后的6个并发，之后版本也是6个  
Firefox，chrome也是6个

#### **25、谈谈你对重构的理解**

网站重构：在不改变外部行为的前提下，简化结构、添加可读性，而在网站前端保持一致的行为。也就是说是在不改变UI的情况下，对网站进行优化，   
在扩展的同时保持一致的UI。

对于传统的网站来说重构通常是：

表格(table)布局改为DIV+CSS

使网站前端兼容于现代浏览器(针对于不合规范的CSS、如对IE6有效的)

对于移动平台的优化

针对于SEO进行优化

深层次的网站重构应该考虑的方面

减少代码间的耦合

让代码保持弹性

严格按规范编写代码

设计可扩展的API

代替旧有的框架、语言(如VB)

增强用户体验

通常来说对于速度的优化也包含在重构中

压缩JS、CSS、image等前端资源(通常是由服务器来解决)

程序的性能优化(如数据读写)

采用CDN来加速资源加载

对于JS DOM的优化

#### **26、什么样的前端代码是好的**

高复用低耦合，这样文件小，好维护，而且好扩展。

****27、对web标准以及w3c的理解和认识****

答:标签闭合，标签小写，不乱嵌套，提高搜索机器人搜索几率。使用外链css和js脚本，结构行为表现分离，内容能被更多广泛的设备所访问,更少的代码和组件，容易维护，改版方便，不需要变动页面内容。

****28、.Html和xhtml有什么区别?****

html是一种基本的web网页设计语言，xhtml是一个基于XML的置标语言。

最主要的不同:

XHTML元素必须正确的被嵌套，元素必须关闭，标签必须小写，必须有根元素。

****29、.严格模式和混杂模式的区分，以及如何触发这2种模式?****

严格模式就是浏览器根据web标准去解析页面，是一种要求严格的DTD，不允许使用任何表现层的语法，如

混杂模式是一种向后兼容的解析方法。

触发标准模式或者说严格模式很简单，就是Html前申明正确的DTD，出发混杂模式可以在html文档开始不声明DTD，或者在DOCTYPE前加入XML声明

****30、.行内元素有哪些？块级元素有哪些？ 空(void)元素有那些？****

行内元素：a、b、span、img、input、strong、select、label、em、button、textarea

块级元素：div、ul、li、dl、dt、dd、p、h1-h6、blockquote

空元素:br、meta、hr、link、input、img

****31、CSS选择符有哪些?优先级算法如何计算?内联和Important哪个优先级高****

回答：CSS选择符有类选择符，属性选择符，ID选择符，优先级算法是基于特殊性值进行计算的。分别如下：ID属性 0.1.0.0

类属性选择器，属性选择器，伪类 0.0.1.0 元素选择符，伪元素选择符 0.0.0.1

通配选择器对特殊性没有任何贡献值。

important优先级高

****32.前端页面有哪三层构成，分别是什么，作用是什么?****

结构层：html 表示层:css 行为层：js

**33、****GET和POST的区别，何时使用POST？**

GET：一般用于信息获取，使用URL传递参数，对所发送信息的数量也有限制，一般在2000个字符

POST：一般用于修改服务器上的资源，对所发送的信息没有限制。

GET方式需要使用Request.QueryString来取得变量的值，而POST方式通过Request.Form来获取变量的值，

也就是说Get是通过地址栏来传值，而Post是通过提交表单来传值。

然而，在以下情况中，请使用 POST 请求：

无法使用缓存文件（更新服务器上的文件或数据库）

向服务器发送大量数据（POST 没有数据量限制）

发送包含未知字符的用户输入时，POST 比 GET 更稳定也更可靠

1. **HTML5的离线储存怎么使用，工作原理能不能解释一下？**

在用户没有与因特网连接时，可以正常访问站点或应用，在用户与因特网连接时，更新用户机器上的缓存文件。  
原理：HTML5的离线存储是基于一个新建的.appcache文件的缓存机制(不是存储技术)，通过这个文件上的解析清单离线存储资源，这些资源就会像cookie一样被存储了下来。之后当网络在处于离线状态下时，浏览器会通过被离线存储的数据进行页面展示。

如何使用：  
（1）页面头部像下面一样加入一个manifest的属性；  
（2）在cache.manifest文件的编写离线存储的资源；  
CACHE MANIFEST  
#v0.11  
CACHE:  
js/app.js  
css/style.css  
NETWORK:  
resourse/logo.png  
FALLBACK:  
/ /offline.html  
（3）在离线状态时，操作window.applicationCache进行需求实现。

1. **浏览器是怎么对HTML5的离线储存资源进行管理和加载的**

在线的情况下，浏览器发现Html头部有manifest属性，它会请求manifest文件，如果是第一次访问APP，那么浏览器就会根据manifest文件的内容下载相应的资源并且进行离线存储。如果已经访问过APP并且资源已经离线存储了，那么浏览器就会使用离线的资源加载页面，然后浏览器会对比新的manifest文件与旧的manifest文件，如果文件没有发生改变，就不做任何操作，如果文件改变了，那么就会重新下载文件中的资源并进行离线存储。 离线的情况下，浏览器就直接使用离线存储的资源。

1. **xhtml和html有什么区别** HTML是一种基本的WEB网页设计语言，XHTML是一个基于XML的置标语言  
    最主要的不同：①XHTML 元素必须被正确地嵌套；②XHTML元素必须被关闭；③标签名必须用小写字母；④XHTML 文档必须拥有根元素。
2. **css的基本语句构成是?** 选择器{属性1:值1;属性2:值2;……}

**38、写出几种IE6 BUG的解决方法**①双边距BUG float引起的 使用display  
②3像素问题 使用float引起的 使用dislpay:inline -3px  
③超链接hover 点击后失效 使用正确的书写顺序 link visited hover active  
④Ie z-index问题 给父级添加position:relative  
⑤Png 透明 使用js代码 改  
⑥Min-height 最小高度 ！Important 解决’  
⑦select 在ie6下遮盖 使用iframe嵌套  
⑧为什么没有办法定义1px左右的宽度容器（IE6默认的行高造成的，使用over:hidden,zoom:0.08 line-height:1px）  
  
**39、标签上title与alt属性的区别是什么?** Alt当图片不显示时用文字代表。Title为该属性提供信息。  
  
**40、描述css reset的作用和用途。** Reset重置浏览器的CSS默认属性浏览器的品种不同，样式不同，然后重置，让他们统一。  
  
**41、解释css sprites，如何使用。**  CSS精灵把一堆小的图片整合到一张大的图片上，减轻服务器对图片的请求数量。  
  
**42、浏览器标准模式和怪异模式之间的区别是什么?** 盒子模型，渲染模式的不同；

使用 window.top.document.compatMode 可显示为什么模式。  
  
**43、你如何对网站的文件和资源进行优化?期待的解决方案包括：** 文件合并、文件最小化/文件压缩、使用CDN托管，缓存的使用。  
  
**44、什么是语义化的HTML?** 直观的认识标签，对于搜索引擎的抓取有好处。

1. **iframe有那些缺点？**
2. 页面样式调试麻烦，出现多个滚动条；  
   （2）浏览器的后退按钮失效；  
   （3）过多会增加服务器的HTTP请求；  
   （4）小型的移动设备无法完全显示框架；  
   （5）产生多个页面，不易管理；  
   （6）不容易打印；  
   （7）代码复杂，无法被一些搜索引擎解读。
3. **HTML5的form如何关闭自动完成功能**

HTML的输入框可以拥有自动完成的功能，当你往输入框输入内容的时候，浏览器会从你以前的同名输入框的历史记录中查找出类似的内容并列在输入框下面，这样就不用全部输入进去了，直接选择列表中的项目就可以了。  
但有时候我们希望关闭输入框的自动完成功能，例如当用户输入内容的时候，我们希望使用AJAX技术从数据库搜索并列举而不是在用户的历史记录中搜索。  
关闭输入框的自动完成功能有3种方法：  
（1）在IE的Internet选项菜单里的内容--自动完成里面设置  
（2）设置Form的autocomplete为"on"或者"off"来开启或者关闭自动完成功能  
（3）设置输入框的autocomplete为"on"或者"off"来开启或者关闭该输入框的自动完成功能  
测试代码：（在每个form输入文字然后提交，然后再回来看看能否自动完成，注意要提交后才能有历史记录，才可能自动完成；提交后页面可能出错，不用管它，后退回去即可）

xml 代码

打开自动完成功能的Form<br>

<form name="form1" autocomplete="on">

打开自动完成功能的输入框

<input type="text" autocomplete="on" name="txtOff1"><br>

关闭自动完成功能的输入框

<input type="text" autocomplete="off" name="txtOn1"><br>

<input type="submit" value="提交"><br>

</form>

关闭自动完成功能的Form<br>

<form name="form1" autocomplete="off">

打开自动完成功能的输入框

<input type="text" autocomplete="on" name="txtOff1"><br>

关闭自动完成功能的输入框

<input type="text" autocomplete="off" name="txtOn1"><br>

<input type="submit" value="提交"><br>

</form>

1. **如何在页面上实现一个圆形的可点击区域？**

使用Dreamweaver制作热点

**49、实现不使用 border 画出1px高的线，在不同浏览器的**Quirksmode和CSSCompat模式下都能保持同一效果。

<div style="height:1px;overflow:hidden;background:black"></div>

**50、tite与h1的区别、b与strong的区别、i与em的区别？**

（1）b和strong的区别

盲人朋友使用阅读设备阅读网络时：<strong>会重读，<b>不会

两者虽然在网页中显示效果一样，但实际目的不同。  
<b>这个标签对应 bold，即文本加粗，其目的仅仅是为了加粗显示文本，是一种样式／风格需求；  
<strong>这个标签意思是加强字符的语气，表示该文本比较重要，提醒读者／终端注意。为了达到这个目的，浏览器等终端将其加粗显示；  
总结：<b>为了加粗而加粗，<strong>为了标明重点而加粗，也可以用其它方式来强调，比如下划线，比如字体加大，比如红色，等等，可以通过css来改变strong的具体表现。

（2）i和em的区别

同样，I是Italic(斜体)，而em是emphasize(强调)。

（3）title与h1的联系：

从网站角度看，title更重于网站信息。title可以直接告诉搜索引擎和用户这个网站是关于什么主题和内容的。

从文章角度看，h1则是用于概括文章主题。用户进入内容页，想看到的当然就是文章的内容，h1文章标题就是最重要的。文章标题最好只有一个，多个h1会导致搜索引擎不知道这个页面哪个标题内容最重要，导致淡化这个页面的标题和关键词，起不到突出主题的效果。

区别：

h1突出文章主题，面对用户，更突出其视觉效果，突出网站标题或关键字用title。一篇文章，一个页面最好只用一个h1，多个h1会稀释主题。一个网站可以有多个title,最好一个单页用一个title，以便突出网站页面主体信息，从seo看，title权重比h1高，适用性比h1广。标记了h1的文字页面给予的权重会比页面内其他权重高很多。一个好的网站是h1和title并存，既突出h1文章主题，又突出网站主题和关键字。达到双重优化网站的效果。

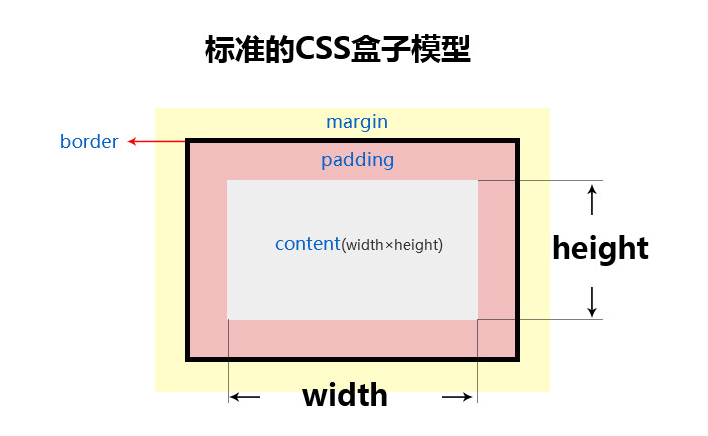
1. **介绍一下标准的CSS的盒子模型？与低版本IE的盒子模型有什么不同的？**

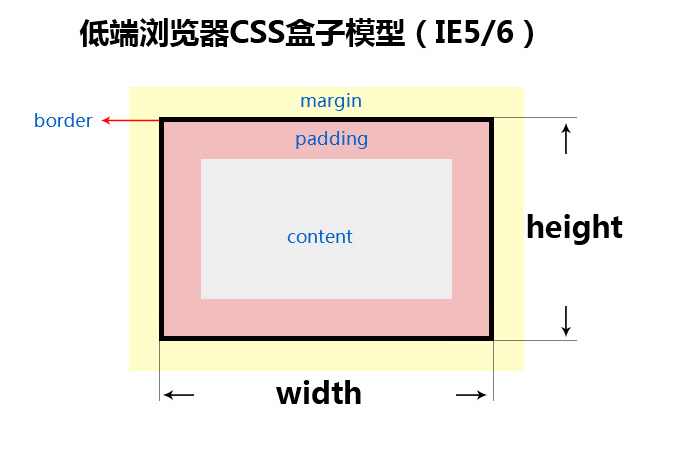
CSS盒子模型：由四个属性组成的外边距(margin)、内边距(padding)、边界(border)、内容区(width和height);

标准的CSS盒子模型和低端IE CSS盒子模型不同：宽高不一样

　　标准的css盒子模型宽高就是内容区宽高；

低端IE css盒子模型宽高 内边距﹢边界﹢内容区；





**52、CSS3新增伪类有那些？**

p:last-of-type         选择其父元素的最后的一个P元素

p:last-child            选择其父元素的最后子元素(一定是P才行)

p:first-of-type        选择其父元素的首个P元素

p:first-child           选择其父元素的首个p元素(一定是p才行)

p:only-child　　　　选择其父元素的只有一个元素(而且这个元素只能是p元素，不能有其他元素)

p:only-of-type　　　选择其父元素的只有一个p元素(不能有第二个P元素，其他元素可以有)

**选第N个**

p:nth-child(n)    　　选择其父元素的第N个 刚好是p的元素

p:nth-last-child(n)    ..............................................从最后一个子元素开始计数

p:nth-of-type(n)　  选择其父元素的n个元素

p:nth-last-of-type(n)   ........................从最后一个子元素开始计数

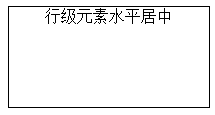
**53、如何居中div？如何居中一个浮动元素？如何让绝对定位的div居中？**

# （一）元素水平居中的方式

1）行级元素水平居中对齐（父元素设置 text-align:center）

[html] [view plain](http://blog.csdn.net/lxcao/article/details/52670724" \o "view plain) [copy](http://blog.csdn.net/lxcao/article/details/52670724" \o "copy)

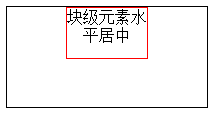
1. <div style="width: 200px; height: 100px;border: 1px solid;text-align:center;">
2. <span>行级元素垂直居中</span>
3. </div>



2)   块级元素水平居中对齐（margin: 0 auto）

[html] [view plain](http://blog.csdn.net/lxcao/article/details/52670724" \o "view plain) [copy](http://blog.csdn.net/lxcao/article/details/52670724" \o "copy)

1. <div style="width: 200px; height: 100px;border: 1px solid;text-align: center;">
2. <div style="border: 1px solid red;margin: 0 auto;height: 50px;width: 80px;"> 块级元素水平居中</div>
3. </div>



3）浮动元素水平居中

* 宽度不固定的浮动元素

html代码

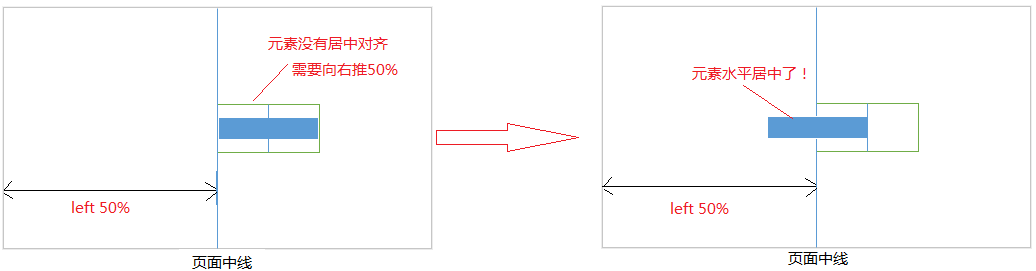
[html] [view plain](http://blog.csdn.net/lxcao/article/details/52670724" \o "view plain) [copy](http://blog.csdn.net/lxcao/article/details/52670724" \o "copy)

1. <div class="outerbox">
2. <div class="innerbox">我是浮动的</div>
3. </div>

CSS样式

[css] [view plain](http://blog.csdn.net/lxcao/article/details/52670724" \o "view plain) [copy](http://blog.csdn.net/lxcao/article/details/52670724" \o "copy)

1. .outerbox{
2. float:left;
3. position:relative;
4. left:50%;
5. }
6. .innerbox{
7. float:left;
8. position:relative;
9. right:50%;
10. }



* 宽度固定的互动元素

html代码

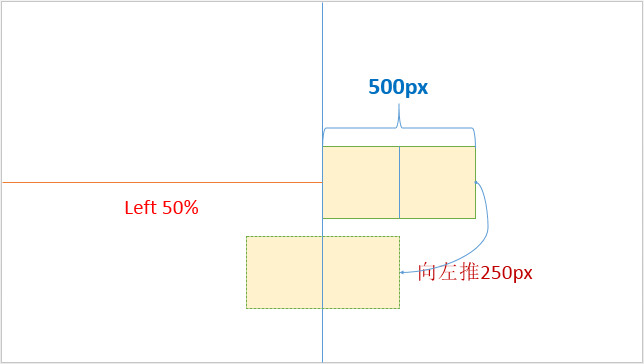
[html] [view plain](http://blog.csdn.net/lxcao/article/details/52670724" \o "view plain) [copy](http://blog.csdn.net/lxcao/article/details/52670724" \o "copy)

1. <div class="outerbox">
2. <div>我是浮动的</div>
3. </div>

css代码

[css] [view plain](http://blog.csdn.net/lxcao/article/details/52670724" \o "view plain) [copy](http://blog.csdn.net/lxcao/article/details/52670724" \o "copy)

1. .outerbox{
2. background-color:pink; /\*方便看效果 \*/
3. width:500px ;
4. height:300px; /\*高度可以不设\*/
5. margin: -150px 0 0 -250px; /\*使用marin向左移动250px，保证元素居中\*/
6. position:relative;   /\*相对定位\*/
7. left:50%;
8. top:50%;
9. }



    4）让绝对定位的元素水平居中对齐

[css] [view plain](http://blog.csdn.net/lxcao/article/details/52670724" \o "view plain) [copy](http://blog.csdn.net/lxcao/article/details/52670724" \o "copy)

1. .center{
2. position: absolute; /\*绝对定位\*/
3. width: 500px;
4. height:300px;
5. background: red;
6. margin: 0 auto; /\*水平居中\*/
7. left: 0; /\*此处不能省略，且为0\*/
8. right: 0; /\*此处不能省略，且为0\*/
9. }

**经验分享：**水平居中的主要属性有

1. text-alin:center;

2. margin:0 auto

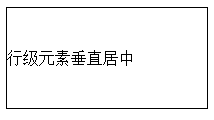
3. position:relative|absolute; left:50%;

# （二）元素垂直居中对齐

1）对**行级**元素垂直居中（heiht与line-height的值一样）

[css] [view plain](http://blog.csdn.net/lxcao/article/details/52670724" \o "view plain) [copy](http://blog.csdn.net/lxcao/article/details/52670724" \o "copy)

1. height:300px;
2. line-height:300px;



2）对**块级**元素垂直居中对齐

2.1 父元素高度固定的情况

1）父元素的height与line-height值相同

2）需要垂直居中的元素

 vertical-align:middle;// 垂直居中对齐

 display:inline|inline-block 块级元素转行级元素

HTML代码

[html] [view plain](http://blog.csdn.net/lxcao/article/details/52670724" \o "view plain) [copy](http://blog.csdn.net/lxcao/article/details/52670724" \o "copy)

1. <div class="center">
2. <div class="inner"></div>
3. </div>

CSS代码

[css] [view plain](http://blog.csdn.net/lxcao/article/details/52670724" \o "view plain) [copy](http://blog.csdn.net/lxcao/article/details/52670724" \o "copy)

1. .center{
2. width: 500px;
3. height:300px;
4. line-height: 300px;
5. border:1px solid;
6. }
7. .inner{
8. background: blue;
9. width: 300px;
10. height: 100px;
11. display: inline-block;
12. vertical-align: middle;
13. }

2.2 父元素高度不固定的情况

 父元素的padding-top和padding-bottom一样

**54、CSS3有哪些新特性？**

（1）选择器   
（2） RGBA和透明度   
（3）多栏布局   
（4）多背景图   
（5） Word Wrap   
（6） 文字阴影   
（7） @font-face属性   
（8）圆角(边框半径)   
（9） 边框图片   
（10） 盒阴影   
（11）盒子大小   
（12）媒体查询   
（13） 语音

1. **用纯CSS创建一个三角形的原理是什么？**

采用的是均分原理

 盒子都是一个矩形或正方形，从形状的中心，向4个角上下左右划分4个部

代码的实现

第一步 保证元素是块级元素

第二步 设置元素的边框

第三步 不需要显示的边框使用透明色

 示例代码

[css] [view plain](http://blog.csdn.net/lxcao/article/details/52689313" \o "view plain) [copy](http://blog.csdn.net/lxcao/article/details/52689313" \o "copy)

1. .square{
2. width:0;
3. height:0;
4. margin:0 auto;
5. border:6px solid transparent;
6. border-top: 6px solid red;
7. }

**56、一个满屏 品 字布局 如何设计?**

**（1）元素水平居中对齐**

1) 使用margin对齐(推荐)

2) 使用left：50%

3) 使用text-align：行内元素居中对齐，给父元素设置text-align： center   
**（2）元素对相对窗口定位**

1) 使用filxed（推荐）：总是根据浏览器的窗口来进行元素的定位

2) 使用absolute定位

3) 使用html和body的width和height填充这个窗口   
**（3）元素左右定位**

1) 使用float左右浮动

2) 使用绝对定位进行左右定位（推荐）   
**第一种方式：**

<!DOCTYPE html><html><head>

<meta charset="utf-8">

<title>满屏品字布局</title>

<style type="text/css">

\*{

margin: 0;

padding: 0;

}

html,body{

height: 100%;/\*此设置非常关键，因为默认的body，HTML高度为0，所以后面设置的div的高度无法用百分比显示\*/

}

.header{

height:50%; /\*此步结合html,body高度为100%，解决元素相对窗口的定位问题\*/

width: 50%;

background: #ccc;

margin:0 auto;

}

.main{

width: 100%;

height: 50%;

background: #ddd;

}

.main .left,.main .right{

float: left;/\*采用float方式，对元素进行左右定位\*/

width:50%;/\*此步解决元素相对窗口的定位问题\*/

height:100%;/\*此步解决元素相对窗口的定位问题\*/

background: yellow;

}

.main .right{

background: green;

}

</style></head><body><div class="header"></div><div class="main">

<div class="left"></div>

<div class="right"></div></div></body></html>

**57、li与li之间有看不见的空白间隔是什么原因引起的？有什么解决办法？**

有时，在写页面的时候，会需要将<li>这个块状元素横排显示，此时就需要将display属性设置为inline-block，此时问题出现了，在两个<li>元素之间会出现大约8px左右的空白间隙，

**引起这种空白间隔的原因：**

浏览器的默认行为是把inline元素间的空白字符（空格换行tab）渲染成一个空格，也就是我们上面的代码<li>换行后会产生换行字符，而它会变成一个空格，当然空格就占用一个字符的宽度。

**解决方案：**

**方法一：**既然是因为<li>换行导致的，那就可以将<li>代码全部写在一排，如下

* <div class="wrap"><h3>li标签空白测试</h3><ul><li class="part1"></li><li class="part2"></li><li class="part3"></li><li class="part4"></li></ul></div>

再刷新页面看就没有空白了，就是这么神奇~

**方法二：**我们为了代码美观以及方便修改，很多时候我们不可能将<li>全部写在一排，那怎么办？既然是空格占一个字符的宽度，那我们索性就将<ul>内的字符尺寸直接设为0，将下面样式放入样式表，问题解决。

.wrap ul{font-size:0px;}

* 1

但随着而来的就是<ul>中的其他文字就不见了，因为其尺寸被设为0px了，我们只好将他们重新设定字符尺寸。

**方法三：**本来以为方法二能够完全解决问题，但经测试，将li父级标签字符设置为0在**Safari浏览器**依然出现间隔空白；既然设置字符大小为0不行，那咱就将间隔消除了，将下面代码替换方法二的代码，目前测试完美解决。同样随来而来的问题是li内的字符间隔也被设置了，我们需要将li内的字符间隔设为默认。

.wrap ul{letter-spacing: -5px;}1

之后记得设置li内字符间隔

.wrap ul li{letter-spacing: normal;}

**58、为什么要初始化CSS样式。**

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

**59、absolute的containing block计算方式跟正常流有什么不同？**

containing block  
一般来说，盒子本身就为其子孙建立了 containing block，用来计算内部盒子的位置、大小，而对内部的盒子，具体采用哪个 containing block 来计算，需要分情况来讨论：  
若此元素为 inline 元素，则 containing block 为能够包含这个元素生成的第一个和最后一个 inline box 的 padding box (除 margin, border 外的区域) 的最小矩形；  
否则则由这个祖先元素的 padding box 构成。  
根元素所在的 containing block 被称为 initial containing block，在我们常用的浏览器环境下，指的是原点与 canvas 重合，大小和 viewport 相同的矩形；  
对于 position 为 static 或 relative 的元素，其 containing block 为祖先元素中最近的 block container box 的 content box (除 margin, border, padding 外的区域)；  
对于 position:fixed 的元素，其 containing block 由 viewport 建立；  
对于 position:absolute 的元素，则是先找到其祖先元素中最近的 position 属性非 static 的元素，然后判断：  
如果都找不到，则为 initial containing block。

**60、CSS里的visibility属性有个collapse属性值是干嘛用的？在不同浏览器下以后什么区别？**

visibility有如下属性值：

| 属性值 | 属性值描述 |
| --- | --- |
| visible | 默认值。元素是可见的。 |
| hidden | 元素是不可见的，相当于display：hidden；，但此时仍占用页面空间 |
| collapse | 当在表格元素中使用时，此值可删除一行或一列，但是它不会影响表格的布局。被行或列占据的空间会留给其他内容使用。*如果此值被用在其他的元素上，会呈现为 “hidden”。* |
| inherit | 规定应该从父元素继承 visibility 属性的值。 |

## **visibility的第三种值collapse：**

* 对于一般的元素，它的表现跟display:hidden是一样的。
* 但例外的是，如果这个元素是table相关的元素，例如table行，table group，table列，table column group，它的表现却跟display: none一样，也就是说，它们占用的空间也会释放。

## **在不同浏览器下的区别：**

* 在谷歌浏览器里，使用collapse值和使用hidden值没有什么区别。
* 在火狐浏览器、Opera和IE11里，使用collapse值的效果就如它的字面意思：table的行会消失，它的下面一行会补充它的位置。

**61、position跟display、margin collapse、overflow、float这些特性相互叠加后会怎么样？**

**一、'display'、'position' 和 'float' 的相互关系**

（1）'display' 的值为 'none'

如果 'display' 的值为 'none'，那么 'position' 和 'float' 不起作用。在这种情况下，元素不产生框。因此浮动和定位无效。  
（2） 'position' 的值是 'absolute' 或 'fixed'  
否则，如果 'position' 的值是 'absolute' 或 'fixed'，框就是绝对定位的，'float' 计算后的值应该是 'none'，并且，'display' 会被按照上表设置。 框的位置将由 'top'，'right'，'bottom' 和 'left' 属性和该框的包含块确定。  
也就是说，当元素是绝对定位时，浮动失效，'display' 会被按规则重置。  
（3）'float' 的值不是 "none"。  
按照规则，SPAN 是行内元素，因此不能够设置其宽度和高度。但是浮动后，'display' 值按照转换对应表设置后，成为块级元素  
（4）元素是根元素  
如果元素是根元素，'display' 的值按照转换对应表设置。

（5） 否则，应用指定的 'display' 特性值。

**二、position跟display、overflow、float下的margin collapse。**

margin collapse我觉得这里的意思应该是Collapsing margins，即外边距折叠，指的是毗邻的两个或多个外边距 (margin) 会合并成一个外边距。  
其中所说的 margin 毗邻，可以归结为以下两点：  
•这两个或多个外边距没有被非空内容、padding、border 或 clear 分隔开。  
•这些 margin 都处于普通流中。  
（1）两个或多个毗邻的普通流中的块元素垂直方向上的 margin 会折叠。

（2）浮动元素、inline-block 元素、绝对定位元素的 margin 不会和垂直方向上其他元素的 margin 折叠.

（3）创建了块级格式化上下文的元素，不和它的子元素发生 margin 折叠

**62、对BFC规范(块级格式化上下文：block formatting context)的理解？**

一. BFC 是什么？

有了上面的基础后，可以正式介绍 BFC 了。从样式上看，具有 BFC 的元素与普通的容器没有什么区别，但是从功能上，具有 BFC 的元素可以看作是隔离了的独立容器，容器里面的元素不会在布局上影响到外面的元素，并且 BFC 具有普通容器没有的一些特性，例如可以包含浮动元素，上文中的第二类清除浮动的方法（如 overflow 方法）就是触发了浮动元素的父元素的 BFC ，使到它可以包含浮动元素，从而防止出现高度塌陷的问题。

简单来说，BFC 就是一种属性，这种属性会影响着元素的定位以及与其兄弟元素之间的相互作用。

二.如何触发 BFC

上面介绍了 BFC 的定义，那么如何触发 BFC 呢？

满足下面任一条件的元素，会触发为 BFC ：

浮动元素，float 除 none 以外的值

绝对定位元素，position（absolute，fixed）

display 为以下其中之一的值 inline-blocks，table-cells，table-captions

overflow 除了 visible 以外的值（hidden，auto，scroll）

但是，"display:table" 本身并不产生 BFC，而是由它产生匿名框，匿名框中包含 "display:table-cell" 的框会产 BFC。 总之，对于 "display:table" 的元素，产生 BFC 的是匿名框而不是 "display:table"。

在 CSS3 中，BFC 叫做 Flow Root，并增加了一些触发条件：

display 的 table-caption 值

position 的 fixed 值，其实 fixed 是 absolute 的一个子类，因此在 CSS2.1 中使用这个值也会触发 BFC ，只是在 CSS3 中更加明确了这一点。

值得注意的是，在前面 Kayo 已经说明过了，BFC 并不是元素，而是某些元素带有的一些属性，因此，是上面这些元素产生了 BFC ，而它们本身并不是 BFC ，这个概念需要区分清楚。

三. BFC 的特性

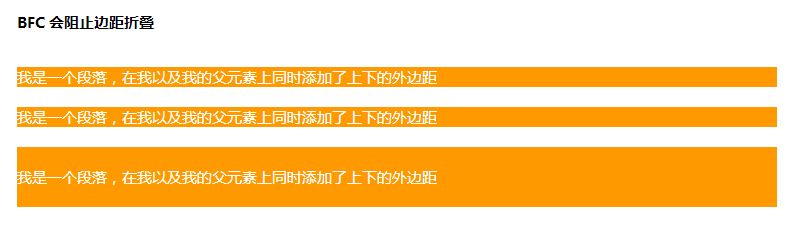
从整体上看，BFC 是隔离了的容器，这个具体可以表现为三个特性：

（1）BFC 会阻止外边距折叠

两个相连的 div 在垂直上的外边距会发生叠加，有些书籍会把这个情况列作 bug ，这里 Kayo 并不同意，这种折叠虽然会给不熟悉 CSS 布局的开发者带来一些不便，但实际上它具有完整且具体的折叠规则，并且在主流浏览器中都存在，因此 Kayo 更认为这应该是 CSS 的特性。当然，在实际开发中，或许我们有时会不需要这种折叠，这时可以利用 BFC 的其中一个特性——阻止外边距叠加。

在举例说明 BFC 如何阻止外边距折叠之前，首先说明一下外边距折叠的规则：仅当两个块级元素相邻并且在同一个块级格式化上下文时，它们垂直方向之间的外边距才会叠加。也就是说，即便两个块级元素相邻，但当它们不在同一个块级格式化上下文时它们的边距也不会折叠。因此，阻止外边距折叠只需产生新的 BFC 。

效果如图：

[](http://kayosite.com/wp-content/uploads/2012/10/bfc-1.png)

也可以看 [Demo](http://kayosite.com/demo/bfc.html" \l "bfc-1" \o "BFC 会阻止外边距折叠" \t "https://www.cnblogs.com/leejersey/p/_blank) 。

如上图的例子，三个 div 各包含一个 p 元素，三个 div 及其包含的 p 元素都有顶部和底部的外边距，但只有第三个 div 的边距没有与它的子元素 p 的外边距折叠。这是因为第三个 div 创建了新的 BFC ，由此可见：创建了 BFC 的元素，不和它的子元素发生外边距折叠。

（2）BFC 可以包含浮动的元素

这也正是使用 overflow: hidden 与 overflow: auto 方法闭合浮动的原理，使用 overflow: hidden 或 overflow: auto 触发浮动元素的父元素的 BFC 特性，从而可以包含浮动元素，闭合浮动。

W3C 的原文是“'Auto' heights for block formatting context roots”，也就是 BFC 会根据子元素的情况自动适应高度，即使其子元素中包括浮动元素。

效果如图：

[](http://kayosite.com/wp-content/uploads/2012/10/bfc-2.png)

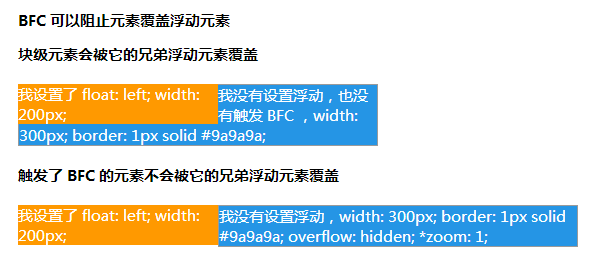
也可以看 [Demo](http://kayosite.com/demo/bfc.html" \l "bfc-2" \o "BFC 可以包含浮动的元素" \t "https://www.cnblogs.com/leejersey/p/_blank) 。

上面的例子中，有两个 div ，它们各包含一个设置了浮动的 p 元素，但第一个 div 出现了“高度塌陷”，这是因为内部的浮动元素脱离了普通流，因此该 div 相当于一个空标签，没有高度和宽度，即高度为 0 ，上下边框也重叠在一起。而第二个 div 使用 overflow: hidden 触发了 BFC ，可以包含浮动元素，因此能正确表现出高度，其边框位置也正常了。

（3）BFC 可以阻止元素被浮动元素覆盖

如上文所说，浮动元素的块级兄弟元素会无视浮动元素的位置，尽量占满一整行，这样就会被浮动元素覆盖，为该兄弟元素触发 BFC 后可以阻止这种情况的发生。

效果如图：

[](http://kayosite.com/wp-content/uploads/2012/10/bfc-31.png)

也可以看 [Demo](http://kayosite.com/demo/bfc.html" \l "bfc-3" \o "BFC 可以阻止元素被浮动元素覆盖" \t "https://www.cnblogs.com/leejersey/p/_blank) 。

如上图的例子，蓝色背景的 div 使用 overflow: hidden 触发了 BFC ，它并不会被它的兄弟浮动元素覆盖，而是处于它的旁边。值得注意的是，以上的情况仅仅是元素宽度之和没有超出父元素宽度的情况，假设浮动元素宽度和它的非浮动兄弟元素宽度都没有超过父元素宽度，但两个元素的宽度加起来超出了父元素宽度的时候，非浮动元素会下降到下一行，即处于浮动元素下方，效果如下图：

[](http://kayosite.com/wp-content/uploads/2012/10/bfc-4.png)

也可以看 [Demo](http://kayosite.com/demo/bfc.html" \l "bfc-4" \t "https://www.cnblogs.com/leejersey/p/_blank) 。

四. BFC 与 hasLayout

细心的童鞋会发现，在上面的例子中，除了使用 overflow: hidden 触发 BFC 外，还使用了一个 \*zomm: 1 的属性，这是 IEhack ，因为 IE6-7 并不支持 W3C 的 BFC ，而是使用私有属性 hasLayout 。从表现上来说，hasLayout 跟 BFC 很相似，只是 hasLayout 自身存在很多问题，导致了 IE6-7 中一系列的 bug 。触发 hasLayout 的条件与触发 BFC 有些相似，具体情况 Kayo 会另写文章介绍。这里 Kayo 推荐为元素设置 IE 特有的 CSS 属性 zoom: 1 触发 hasLayout ，zoom 用于设置或检索元素的缩放比例，值为“1”即使用元素的实际尺寸，使用 zoom: 1 既可以触发 hasLayout 又不会对元素造成其他影响，相对来说会更为方便。

这时我们需要注意一个问题：既然 hasLayout 有着跟 BFC 相似的功能，那么在实际开发中，就要为需要触发 BFC 的元素同时触发 hasLayout ，这样 BFC 和 hasLayout 具有的一些特殊性质可以在现代浏览器和 IE 中同时产生，避免一个元素在不同浏览器间的表现因为 BFC 或 hasLayout 出现差异。事实上，在实际开发中很多莫名其妙的问题其实都是因此而产生的。当然同样地，如果一个元素没有触发 BFC ，也要尽量保证它没有触发 hasLayout 。

1. **移动端的布局用过媒体查询吗？**

通过媒体查询可以为不同大小和尺寸的媒体定义不同的css，适合相应的设备显示；即响应式布局

@media screen and (min-width: 400px) and (max-width: 700px) { … }

@media handheld and (min-width: 20em), screen and (min-width: 20em) { … }

**64、使用 CSS 预处理器吗？喜欢那个？**

**一、什么是CSS预处理器**

　　CSS预处理器定义了一种新的语言，基本的思想是用一种专门的编程语言，开发者只需要使用这种语言进行编码工作，减少枯燥无味的CSS代码的编写过程的同时，它能让你的CSS具备更加简洁、适应性更强、可读性更加、层级关系更加明显、更易于代码的维护等诸多好处。

　　CSS预处理器种类繁多，本次就以Sass、Less、Stylus进行比较。

**二、语法**

　　在使用CSS预处理器之前最重要的是了解语法，我只写过stylus，就从网上找了下另外两种语法的格式，与大家对比分享。

　　首先Sass和Less都是用的是标准的CSS语法，因此你可以很方便的把已完成的CSS代码转为预处理器识别的代码，**Sass**默认使用 **.sass**扩展名，而**Less**默认使用**.Less**扩展名。

一下是两者的语法

/\* style.scss or style.less \*/

h1 {

color: #0982C1;

}

你注意到Sass同时也支持老语法，就是不包含花括号和分号的书写格式。

/\* style.sass \*/

h1

color: #0982c1

然而Stylus支持的语法就更多样性，它默认使用**.styl**的文件扩展名，下面是Stylus语法。

/\* style.styl \*/

h1 {

color: #0982C1;

}

/\* omit brackets \*/

h1

color: #0982C1;

/\* omit colons and semi-colons \*/

h1

color #0982C1

**三、变量**

　　你可以在CSS预处理中声明变量，并在整个样式单中使用，支持任何类型的变量，例如：颜色、数值（是否包含单位）、文本。然后你可以任意的调取和使用该变量。Sass的变量是必须$开始，然后变量名和值要冒号隔开，跟CSS属性书写格式一致。

$mainColor: #0982c1;

$siteWidth: 1024px;

$borderStyle: dotted;

body {

color: $publicColor;

border: 1px $borderStyle $mainColor;

max-width: $siteWidth;

}

而Less的变量名使用@符号开始：

@mainColor: #0982c1;

@siteWidth: 1024px;

@borderStyle: dotted;

body {

color: @publicColor;

border: 1px @borderStyle @mainColor;

max-width: @siteWidth;

}

　　Stylus对变量名没有任何限定，你可以是$开始，也可以是任意字符，而且与变量值之间可以用冒号、空格隔开，需要注意的是 Stylus (0.22.4) 将会编译 @ 开始的变量，但其对应的值并不会赋予该变量，换句话说，在 Stylus 的变量名不要用 @ 开头。

mainColor = #0982c1

siteWidth = 1024px

$borderStyle = dotted

body

color mainColor

border 1px $borderStyle mainColor

max-width siteWidth

上面三种不同的CSS写法，最终将会生成相同结果：

body {

color: #0982c1;

border: 1px dotted #0982c1;

max-width: 1024px;

}

　　最容易体现它的好处的是，假设你在CSS中使用同一种颜色数十次，如果你要修改显色，需要找到并修改十次相同的代码，而有了CSS预处理器，修改一个地方就够了。

**四、嵌套**

　　如果你需要在CSS中相同的parent引用多个元素，你需要一遍一遍的去写parent。例如：

section {

margin: 10px;

}

section nav {

height: 25px;

}

section nav a {

color: #0982C1;

}

section nav a:hover {

text-decoration: underline;

}

然而如果用CSS预处理器，就可以少些很多单词，而且父节点关系一目了然。

section {

margin: 10px;

nav {

height: 25px;

a {

color: #0982C1;

&amp;:hover {

text-decoration: underline;

}

}

}

}

stylus还可省略花括号，书写更加方便，根据个人喜好还可删除中间冒号。

section

margin: 10px;

nav

height: 25px;

a

color: #0982C1;

&amp;:hover

text-decoration: underline;

最终生成CSS结果是：

section {

margin: 10px;

}

section nav {

height: 25px;

}

section nav a {

color: #0982C1;

}

section nav a:hover {

text-decoration: underline;

}

**65、浏览器是怎样解析CSS选择器的？**

按照从上到下，从右到左的顺序解析。

.list a {color:blue;}

* 1

先解析到 a 标签，并将页面上所有 a 标签的字体颜色都按照 color:red 进行渲染（蓝色），再解析到 .list ，将页面上所有 .list 类目下的 a 标签的字体渲染成蓝色。是的，你没有看错，浏览器解析CSS从来就是这么苦逼。

**66、margin和padding分别适合什么场景使用？**

◆何时应当使用margin

需要在border外侧添加空白时。

空白处不需要背景（色）时。

上下相连的两个盒子之间的空白，需要相互抵消时。如15px+20px的margin，将得到20px的空白。

◆何时应当时用padding

需要在border内测添加空白时。

空白处需要背景（色）时。

上下相连的两个盒子之间的空白，希望等于两者之和时。如15px+20px的padding，将得到35px的空白。

◆浏览器兼容性问题

在IE5.x、IE6中，为float的盒子指定margin时，左侧margin可能会变成两倍的宽度。通过改用padding或指定盒子为display:inline可以解决。

**67、抽离样式模块怎么写，说出思路，有无实践经验？**[阿里航旅的面试题]

 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： \* {padding: 0; margin: 0;} （强烈不建议）  
  
  
        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; }

**68、元素竖向的百分比设定是相对于容器的高度吗？**

（1）竖向百分比 ——之height

.con{  
width: 200px;  
height: 100px;  
background: gray;  
margin-left: 200px;  
}  
p{  
/\*margin-top: 20%;\*/  
background: cornflowerblue;  
height: 50%;  
}

<div class="con">  
<p>这是测试的内容！</p>  
</div>



可见对于height属性取值百分比，是现对于容器高度的

（2）竖向百分比——之 间距

对于margin-top、margin-bottom、padding-top、padding-bottom这些竖直方向的内外边距属性的百分比取值，

参考的其实是容器的宽度而不是高低。

p{  
margin-top: 20%;  
background: cornflowerblue;  
/\*height: 50%;\*/  
}



1. **什么是响应式设计？**

随着移动设备的流行使用，在移动设备端展示的互联网信息量越来越大，因此网页内容为了适应移动设备端的显示效果，从而出现了响应式布局设计理念。

    响应式布局设计的理念是：

    页面的布局方式应当根据用户所处的设备环境（系统平台，屏幕尺寸，屏幕方向）进行正确的响应布局调整，无论用户使用的是笔记本还是手机或者平板，我们的网站页面都能够自动切换分辨率，图片大小尺寸以及相关的脚本功能，响应式Web设计的目的就是为了：只需一个网站前台源码，却能兼容多个终端，而不是为了每个终端去单独设计网站前台。

1. ****视差滚动效果，如何给每页做不同的动画？（回到顶部，向下滑动要再次出现，和只出现一次分别怎么做？）****  
   答：视差滚动（Parallax Scrolling）就是这样的效果之一。这种技术通过在网页向下滚动的时候，控   
   制背景的移动速度比前景的移动速度慢来创建出令人惊叹的3D效果。  
   原理：（1）CSS3实现  
   优点：开发时间短、性能和开发效率比较好，缺点是不能兼容到低版本的浏览器  
   （2）jquery实现  
   通过控制不同层滚动速度，计算每一层的时间，控制滚动效果。  
   优点：能兼容到各个版本的，效果可控性好  
   缺点：开发起来对制作者要求高  
   （3）插件实现方式  
   例如：parallax-scrolling，兼容性十分好

**71、::before 和 :after中双冒号和单冒号 有什么区别？解释一下这2个伪元素的作用。**

单冒号(:)用于[CSS3](https://www.baidu.com/s?wd=CSS3&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1Y3nj79PhDYrHIBPADvnvFW0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EPjn1PW6vPW0zn1DkP1fLPWfY" \t "https://zhidao.baidu.com/question/_blank)伪类，双冒号(::)用于[CSS3](https://www.baidu.com/s?wd=CSS3&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1Y3nj79PhDYrHIBPADvnvFW0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EPjn1PW6vPW0zn1DkP1fLPWfY" \t "https://zhidao.baidu.com/question/_blank)伪元素。  
伪元素由双冒号和伪元素名称组成。双冒号是在[css3](https://www.baidu.com/s?wd=css3&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1Y3nj79PhDYrHIBPADvnvFW0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EPjn1PW6vPW0zn1DkP1fLPWfY" \t "https://zhidao.baidu.com/question/_blank)规范中引入的，用于区分伪类和伪元素。但是伪类兼容现存样式，浏览器需要同时支持旧的伪类，比如:first-line、:first-letter、:before、[:after](https://www.baidu.com/s?wd=:after&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1Y3nj79PhDYrHIBPADvnvFW0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EPjn1PW6vPW0zn1DkP1fLPWfY" \t "https://zhidao.baidu.com/question/_blank)等。  
对于CSS2之前已有的伪元素，比如:before，单冒号和双冒号的写法::before作用是一样的。

**72、如何修改chrome记住密码后自动填充表单的黄色背景 ？**

1. 情景一：input文本框是纯色背景的
2. input:-webkit-autofill {
3. -webkit-box-shadow: 0 0 0px 1000px white inset;
4. border: 1px solid #CCC!important;
5. }
6. 思路二： 关闭浏览器自带填充表单功能
7. 设置表单属性 autocomplete="off/on" 关闭自动填充表单，自己实现记住密码
9. <!-- 对整个表单设置 -->
10. <form autocomplete="off" method=".." action="..">
11. <!-- 或对单一元素设置 -->
12. <input type="text" name="textboxname" autocomplete="off">
13. **你对line-height是如何理解的？**

“行高”顾名思意指一行文字的高度。具体来说是指两行文字间基线之间的距离。基线实在英文字母中用到的一个概念，我们刚学英语的时使用的那个英语本子每行有四条线，其中底部第二条线就是基线，是a,c,z,x等字母的底边线。下图的红色线即为基线。



该属性会影响行框的布局。在应用到一个块级元素时，它定义了该元素中基线之间的最小距离而不是最大距离。

line-height 与 font-size 的计算值之差（在 CSS 中成为“行间距”）分为两半，分别加到一个文本行内容的顶部和底部。可以包含这些内容的最小框就是行框。

原始数字值指定了一个缩放因子，后代元素会继承这个缩放因子而不是计算值

1. **设置元素浮动后，该元素的display值是多少？**

无论行内元素还是块元素，被设置浮动之后，display属性值都变为block;

1. **怎么让Chrome支持小于12px 的文字**

Chrome 27之前的中文版桌面浏览器会默认设定页面的最小字号是12px，英文版则没有限制，主要是因为chrome认为汉字小于12px就会增加识别难度，尤其是中文常用的宋体和微软雅黑。而我们在实际项目中，对于数字/英文内容，其他字体的文本可能会有特殊的需求要求它们以更小的字号来显示，这个时候就需要取消浏览器的自动调整功能了。  
  
一般解决方案是禁止webkit浏览器配置调整网页的字体大小。如下CSS定义方式：  
[css] view plaincopy  
.classstyle{ -webkit-text-size-adjust:none; font-size:9px; }   
  
再讲一下text-size-adjust属性，该属性用来设定文字大小是否根据设备(浏览器)来自动调整显示大小，safari 3.0+，chrome 1.0+可以支持。属性值，可以为三种：  
percentage：字体显示的大小；  
auto：默认，字体大小会根据设备/浏览器来自动调整；  
none:字体大小不会自动调整  
  
据说该属性最初专门是为iPhone版safari设计的，用来自动调整普通网页在iPhone手机端字体的展现问题，不过，既然是webkit的私有属性，现在也经常用在webkit内核的桌面浏览器限制页面展示。实际上，这是webkit的一个bug。在最新版的Chrome已经修复。  
  
需要注意的是，不合理的使用-webkit-text-size-adjust:none会造成许多不好的影响。比如将其定义为全局属性的话，在Chrome中当用户放大缩小页面（PC上按住Ctrl滚动鼠标滚轮可缩放网页）的时候，文字却维持定义的大小而不放缩，给用户带来的不太友好的体验。所以我们在使用时，最好定义为局部有效，而不要图省事一句html{-webkit-text-size-adjust:none;}了事。  
  
由于没有 -o-text-size-adjust，这样的CSS 属性，在 Opera，我们就只能通过自己手动设置了：工具->首选项->高级->字体->最小字体大小（像素）。  
  
PC桌面版Chrome 27正式取消了-webkit-text-size-adjust属性的支持，实际上是修正了原有的bug。如果定义该属性在Chrome调试窗口会显示Unknown property name，所有字号最小为12px。但是，移动端chrome/safari目前依然支持-webkit-text-size-adjust属性。因为此属性的滥用会使得webkit内核的浏览器失去调节能力，对于有视觉障碍的浏览者非常不友好（尤其是移动终端），那么现在该如何实现原来的需求呢？  
我们可以尝试通过对文字区域局部应用以下样式解决：  
  
[css] view plaincopy  
.chrome\_adjust {   
 font-size: 9px;   
 -webkit-transform: scale(0.75);   
}   
  
12×0.75=9，对于其它浏览器来说，12px以下的字号都是可以识别的，这里仅需要对于Chrome浏览器进行缩放。可是如何分辨是Safari还是Chrome，目前尚没有有效的CSS hack。可以通过javascript来判断是否为Chrome。判断方法：var isChrome = !!window.chrome;当检测为Chrome的时候，将.chrome\_adjust这个类添加到对应的标签。  
  
但是，问题还没有解决。看到网页在三种浏览器的不同表现：  
1）、Chrome下由于启用了缩放，所以字符间距出现问题，影响了美观，这时候如果追求完美，可能你还会想到js判断为Chrome后再用CSS属性letter-spacing去修复；  
2）、Firefox不认识-webkit，所以表现正常，9px；  
  
3）、Opera 12.5+能够识别-webkit-前缀（Opera 12.15版本，内核暂未更换为webkit，但是已能够识别-webkit-前缀了，而且在检查元素时还抹掉了前缀），但又能够显示12px以下的字号，结果变成了9×0.75，影响了肉眼的识别，这时候，又得给opera添加-o-transform: scale(1);这个属性。  
  
[css] view plaincopy  
.chrome\_adjust {   
 font-size: 9px;   
 -webkit-transform: scale(0.75);   
 -o-transform: scale(1); //针对能识别-webkit的opera browser设置   
}

1. **让页面里的字体变清晰，变细用CSS怎么做？**

CSS3里面加入了一个“-webkit-font-smoothing”属性。这个属性可以使页面上的字体抗锯齿,使用后字体看起来会更清晰舒服。加上之后就顿时感觉页面小清晰了。

77、font-style属性可以让它赋值为“oblique” oblique是什么意思？

要搞清楚这个问题，首先要明白字体是怎么回事。一种字体有粗体、斜体、下划线、删除线等诸多属性。  
但是并不是所有字体都做了这些，一些不常用的字体，或许就只有个正常体，如果你用Italic，就没有效果了~这时候你就要用Oblique.  
可以理解成Italic是使用文字的斜体，Oblique是让没有斜体属性的文字倾斜！  
  
italic和oblique都是向右倾斜的文字, 但区别在于Italic是指斜体字，而Oblique是倾斜的文字，对于没有斜体的字体应该使用Oblique属性值来实现倾斜的文字效果.

1. **position:fixed;在android下无效怎么处理？**

<meta name="viewport" content="width=device-width, initial-scale=1.0, maximum-scale=1.0, minimum-scale=1.0, user-scalable=no"/>

1. **如果需要手动写动画，你认为最小时间间隔是多久，为什么？**

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

1. **display:inline-block 什么时候会显示间隙？**

inline-block水平呈现的元素间，换行显示或空格分隔的情况下会有间距

**80、overflow: scroll时不能平滑滚动的问题怎么处理？**

（1）阻止所有能导致页面滚动的事件。//scroll不能阻止，只能阻止mousewheel，鼠标拽[滚动条](https://www.baidu.com/s?wd=%E6%BB%9A%E5%8A%A8%E6%9D%A1&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1YYPh7WP1bkm1f1rHw-PyN-0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EPWRsnHb3nW0dnHbznjnYnjfd" \t "https://zhidao.baidu.com/question/_blank)就悲剧了；   
（2）bodyoverflow:hidden//win下右侧[滚动条](https://www.baidu.com/s?wd=%E6%BB%9A%E5%8A%A8%E6%9D%A1&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1YYPh7WP1bkm1f1rHw-PyN-0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EPWRsnHb3nW0dnHbznjnYnjfd" \t "https://zhidao.baidu.com/question/_blank)会消失导致页面横移，移动端阻止不了；  
（3）把滚动部分单独放在一个div里，和弹出部分同级，body和window同高。//所有涉及offset/scrollTop的方法都要修改。fix并且width100%的元素(比如[微博](https://www.baidu.com/s?wd=%E5%BE%AE%E5%8D%9A&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1YYPh7WP1bkm1f1rHw-PyN-0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EPWRsnHb3nW0dnHbznjnYnjfd" \t "https://zhidao.baidu.com/question/_blank)顶栏)会压在内容区[滚动条](https://www.baidu.com/s?wd=%E6%BB%9A%E5%8A%A8%E6%9D%A1&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1YYPh7WP1bkm1f1rHw-PyN-0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EPWRsnHb3nW0dnHbznjnYnjfd" \t "https://zhidao.baidu.com/question/_blank)上；   
（4）弹出时算scrollTop，给内容区fix然后top移动到目前位置，同时body给一个overflow-y:scroll强撑出滚动条。

**81、有一个高度自适应的div，里面有两个div，一个高度100px，希望另一个填满剩下的高度。**

## box-sizing方案     和 absolute position方案

（1）外层box-sizing: border-box; 同时设置padding: 100px 0 0；

（2）内层100像素高的元素向上移动100像素，或使用absolute定位防止占据空间；

（3）另一个元素直接height: 100%;       代码如下

       <!DOCTYPE html>  
       <html lang="en">  
       <head>  
      <meta charset="UTF-8">  
      <title>Title</title>  
      <style type="text/css">  
  
html,  
body { height: 100%; padding: 0; margin: 0; }  
.outer { height: 100%; padding: 100px 0 0; box-sizing: border-box ; }  
.A { height: 100px; margin: -100px 0 0; background: #BBE8F2; }  
.B { height: 100%; background: #D9C666; }  
  
</style>  
  
  
</head>  
<body>  
<div class="outer">  
<div class="A"></div>  
<div class="B"></div>  
</div>  
  
</body>  
</html>

    第二种css  代码  如下 ：

html, body { height: 100%; padding: 0; margin: 0; }

.outer { height: 100%; padding: 100px 0 0; box-sizing: border-box ; position: relative; }

.A { height: 100px; background: #BBE8F2; position: absolute; top: 0 ; left: 0 ; width: 100%; }

.B { height: 100%; background: #D9C666; }

  第三种css  代码 如下：

html, body { height: 100%; padding: 0; margin: 0; }

.outer { height: 100%; position: relative; }

.A { height: 100px; background: #BBE8F2; }

.B { background: #D9C666; width: 100%; position: absolute; top: 100px ; left: 0 ; bottom: 0; }

**82、png、jpg、gif 这些图片格式解释一下，分别什么时候用。有没有了解过webp？**

三种都是图片格式，但它们保存的方式不同，所以区别很大，PNG：这是三种中质量最好的一种，保存图像是静态图，可以保留[32位色](https://www.baidu.com/s?wd=32%E4%BD%8D%E8%89%B2&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1YLrH7hmWb1rHI9uW6snARY0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EnW0dnHm3n1c4PH63PHc3n1DYPs" \t "https://zhidao.baidu.com/question/_blank)，也能保留透明与半透明区域，如果你是位图像设计者，要保存一幅自己设计的并且只有一个图层的图，建议你选择Png，但是很多网站不支持PNG上传 JPG：这是目前使用最广泛的格式之一，因为其高质量的压缩率导致的图片大小减少，而且也支持[32位色](https://www.baidu.com/s?wd=32%E4%BD%8D%E8%89%B2&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1YLrH7hmWb1rHI9uW6snARY0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EnW0dnHm3n1c4PH63PHc3n1DYPs" \t "https://zhidao.baidu.com/question/_blank)彩，因此被广泛使用，一般情况下，任何支持图片的地方都支持jpg，也为静态图，但确点是不支持透明区域 GIF：这也是目前使用最广泛的格式之一，和jpg相比，有以下不同：颜色数只支持256色，支持透明区域，不支持半透明区域，可以保存为动态图

**83、style标签写在body后与body前有什么区别？**

写在head标签中利于浏览器逐步渲染（resources downloading->CSSOM+DOM->RenderTree(composite)->Layout->paint）。  
写在body标签后由于浏览器以逐行方式对html文档进行解析，当解析到写在尾部的样式表（外联或写在style标签）会导致浏览器停止之前的渲染，等待加载且解析样式表完成之后重新渲染，在windows的IE下可能会出现FOUC现象（即样式失效导致的页面闪烁问题）

**84、页面重构怎么操作？**

网站重构：在不改变外部行为的前提下，简化结构、添加可读性，而在网站前端保持一致的行为。

也就是说是在不改变UI的情况下，对网站进行优化，在扩展的同时保持一致的UI。

对于传统的网站来说重构通常是：

表格(table)布局改为DIV+CSS

使网站前端兼容于现代浏览器(针对于不合规范的CSS、如对IE6有效的)

对于移动平台的优化

针对于SEO进行优化

深层次的网站重构应该考虑的方面

减少代码间的耦合

让代码保持弹性

严格按规范编写代码

设计可扩展的API

代替旧有的框架、语言(如VB)

增强用户体验

通常来说对于速度的优化也包含在重构中

压缩JS、CSS、image等前端资源(通常是由服务器来解决)

程序的性能优化(如数据读写)

采用CDN来加速资源加载

对于JS DOM的优化

HTTP服务器的文件缓存

**87、你用的得心应手用的熟练地编辑器&开发环境是什么样子？**

Sublime Text 3 + 相关插件编写前端代码

Google chrome 、Mozilla Firefox浏览器 +firebug 兼容测试和预览页面UI、动画效果和交互功能

Node.js+Gulp

git 用于版本控制和Code Review

**88、平时如何管理你的项目？**

先期团队必须确定好全局样式（globe.css），编码模式(utf-8) 等；

编写习惯必须一致（例如都是采用继承式的写法，单样式都写成一行）；

标注样式编写人，各模块都及时标注（标注关键样式调用的地方）；

页面进行标注（例如 页面 模块 开始和结束）；

CSS跟HTML 分文件夹并行存放，命名都得统一（例如style.css）；

JS 分文件夹存放 命名以该JS功能为准的英文翻译。

**89、移动端（Android IOS）怎么做好用户体验?**

清晰的视觉纵线、

信息的分组、极致的减法、

利用选择代替输入、

标签及文字的排布方式、

依靠明文确认密码、

合理的键盘利用

**90、Label的作用是什么？是怎么用的？（加 for 或 包裹）**label标签是用来定义表单控制间的关系，当用户选择该标签时，浏览器会自动将焦点转到和标签相关的表单控件上。

主要有以下两种用法用法：

一、

[html] [view plain](http://blog.csdn.net/mrchengzp/article/details/78336687" \o "view plain) [copy](http://blog.csdn.net/mrchengzp/article/details/78336687" \o "copy)

1. <label for = "Name">Number:</label>
2. <input type="text" name="Name" id="Name" />

二、

[html] [view plain](http://blog.csdn.net/mrchengzp/article/details/78336687" \o "view plain) [copy](http://blog.csdn.net/mrchengzp/article/details/78336687" \o "copy)

1. <label>Data:<input type="text" name="B" /><label>