

# 面试题(CSS部分答案)

## HTML和CSS部分

### 1.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，就当它从来不存在。

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。

### 2.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引用的CSS会等到页面被加载完再加载;
- (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.

### 3.position的absolute与fixed共同点与不同点

共同点：

- 1.改变行内元素的呈现方式，display被置为block；
- 2.让元素脱离普通流，不占据空间；3.默认会覆盖到非定位元素上

## 不同点：

- absolute的”根元素“是可以设置的，而fixed的”根元素“固定为浏览器窗口。当你滚动网页，fixed元素与浏览器窗口之间的距离是不变的。

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

## 5.CSS 选择符有哪些？哪些属性可以继承？优先级算法如何计算？ CSS3新增伪类有那些？

### 选择器

- 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:nthchild )

### 可继承的样式：

- fontsize fontfamily color, textindent;

### 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;**

**CSS3新增伪类举例：**

- p:firstoftype 选择属于其父元素的首个元素的每个元素。
- p:lastoftype 选择属于其父元素的最后元素的每个元素。
- p:onlyoftype 选择属于其父元素唯一的元素的每个元素。
- p:onlychild 选择属于其父元素的唯一子元素的每个元素。
- p:nthchild(2) 选择属于其父元素的第二个子元素的每个元素。
- :enabled :disabled 控制表单控件的禁用状态。
- :checked 单选框或复选框被选中。

## **6.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  
borderimage

## 7.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）

```
1.      淘宝的样式初始化:
2.      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; }
3.      body, button, input, select, textarea { font:12px/1.5tahoma, arial, \5b8b\4f53; }
4.      h1, h2, h3, h4, h5, h6{ fontsize:100%; }
5.      address, cite, dfn, em, var { fontstyle:normal; }
6.      code, kbd, pre, samp { fontfamily:courier new, courier, monospace; }
7.      small{ fontsize:12px; }
8.      ul, ol { liststyle:none; }
9.      a { textdecoration:none; }
10.     a:hover { textdecoration:underline; }
11.     sup { verticalalign:texttop; }
12.     sub{ verticalalign:textbottom; }
13.     legend { color:#000; }
14.     fieldset, img { border:0; }
15.     button, input, select, textarea { fontsize:100%; }
16.     table { bordercollapse:collapse; borderspacing:0; }
```

## 8.对BFC规范的理解？

A:

- 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。 )

**B:**

- 浮动和绝对定位元素，非块盒的块容器（诸如：inline-blocks, table-cells 和 table-captions），以及 overflow 值不为 visible（除非该值已经延伸到视口上）的块盒，都会为他们的内容创建新的块格式化上下文。
- 在一个块格式化上下文里，盒子从包含块顶部开始一个接一个的纵向排列。两个相邻兄弟盒子之间的垂直的间隙取决于 margin 定义。块级盒纵向相邻（水平书写格式下）的 margins 会在同一个块格式化上下文中折叠合并（取最大值）。
- 在一个块格式化上下文里，每个盒子的左外边界会触碰到包含块的左边界（如果是从右到左的书写格式，则为右边界），即使同时存在浮动元素也是如此（虽然盒子的行框集可能会由于浮动的存在而缩小），除非这个盒子创建了一个新的块格式化上下文（在这种情况下，盒子本身由于浮动可能会变得更窄）。

## 9.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。

## 10.HTML与XHTML——二者有什么区别

### 区别：

- 1.所有的标记都必须要有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“”
- 8.图片必须有说明文字

## 11.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marginleft或marginright，margin值会加倍。）

```
1.      #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浏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

```
1.
2.     .bb{
3.         backgroundColor:#f1ee18; /*所有识别*/
4.         .backgroundColor:#00deff\9; /*IE 6、7、8识别*/
5.         +backgroundColor:#a200ff; /*IE6、7识别*/
6.         _backgroundColor:#1e0bd1; /*IE6识别*/
7.     }
```

- 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,event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 webkittextsizeadjust: 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:  
LVHA : a:link {} a:visited {} a:hover {} a:active {}
- 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marginleft和marginright不会重合，但是margintop和marginbottom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margintop或者同时采用marginbottom。
- ie6对png图片格式支持不好(引用一段脚本处理)

## 12.浮动元素引起的问题和解决办法？

- **浮动元素引起的问题：**
  - (1) 父元素的高度无法被撑开，影响与父元素同级的元素
  - (2) 与浮动元素同级的非浮动元素会跟随其后
  - (3) 若非第一个元素浮动，则该元素之前的元素也需要浮动，否则会影响页面显示的结构
- **解决方法：** 使用CSS中的clear:both;属性来清除元素的浮动可解决2、3问题，对于
  - 1，添加如下样式，给父元素添加clearfix样式：<优点：兼容性很好  
弊端：代码过多，增加文件大小>

```
1. .clearfix:after{
2.     content: ".";
3.     display: block;
4.     height: 0;
5.     clear: both;
6.     visibility: hidden;
7. }
8. .clearfix{
9.     display: inline-block;
10. } /* for IE/Mac */
```

- 2.额外标签法，  
（缺点：不过这个办法会增加额外的标签使HTML结构看起来不够简洁。）
- 3.浮动外部元素
- 4.给父级元素设置 `overflow:hidden` 或者 `overflow:auto`

## 13.简介盒子模型，并且可以画出盒子模型

- CSS的盒子模型有两种：IE盒子模型、标准的W3C盒子模型模型  
盒模型：内容、内边距、外边距（一般不计入盒子实际宽度）、边框
- 标准的W3C盒子模型模型：content 部分不包含其他部分
- ie 盒子模型：content 部分包含了 border 和 padding

## 14.px和em的区别

- px和em都是长度单位，区别是，px的值是固定的，指定是多少就是多少，计算比较容易。em得值不是固定的，并且em会继承父级元素的字体大小。

浏览器的默认字体高都是16px。所以未经调整的浏览器都符合: 1em=16px。那么12px=0.75em, 10px=0.625em。

## 15.描述一个”reset”的CSS文件并如何使用它。知道normalize.css吗？你了解他们的不同之处？

- 重置样式非常多，凡是一个前端开发人员肯定有一个常用的重置CSS文件并知道如何使用它们。他们是盲目的在做还是知道为什么这么做呢？原因是不同的浏览器对一些元素有不同的默认样式，如果你不处理，在不同的浏览器下会存在必要的风险，或者更有戏剧性的性发生。

你可能会用Normalize来代替你的重置样式文件。它没有重置所有的样式风格，但仅提供了一套合理的默认样式值。既能让众多浏览器达到一致和合理，但又不扰乱其他的东西（如粗体的标题）。

在这一方面，无法做每一个复位重置。它也确实有些超过一个重置，它处理了你永远都不用考虑的怪癖，像HTML的audio元素不一致或line-height不一致。

## 16.Sass、LESS是什么？大家为什么要使用他们？

- 他们是CSS预处理器。他是CSS上的一种抽象层。他们是一种特殊的语法/语言编译成CSS。

例如Less是一种动态样式语言. 将CSS赋予了动态语言的特性，如变量，继承，运算，函数. LESS 既可以在客户端上运行 (支持IE 6+, Webkit, Firefox)，也可一在服务端运行 (借助Node.js)。

为什么要使用它们？

结构清晰，便于扩展。

可以方便地屏蔽浏览器私有语法差异。这个不用多说，封装对浏览器语法差异的重复处理，减少无意义的机械劳动。

可以轻松实现多重继承。

完全兼容 CSS 代码，可以方便地应用到老项目中。LESS 只是在 CSS 语法上做了扩展，所以老的 CSS 代码也可以与 LESS 代码一同编译。

## 17.对WEB标准以及W3C的理解与认识

- 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、结构行为表现的分离、文件下载与页面速度更快、内容能被更多的用户所访问、内容能被更广泛的设备所访问、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、提供打印版本而不需要复制内容、提高网站易用性。

## 18.Doctype作用？严格模式与混杂模式-如何触发这两种模式，区分它们有何意义？

- （1）、声明位于文档中的最前面，处于 标签之前。告知浏览器的解析器，用什么文档类型 规范来解析这个文档。
- （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。

## 19.如何居中div,如何居中一个浮动元素？

给div设置一个宽度，然后添加margin:0 auto属性

```
1.  div{
2.      width:200px;
3.      margin:0 auto;
4.  }
```

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

```
1.  .div {
2.      Width:500px ;
3.      height:300px; //高度可以不设
4.      Margin: -150px 0 0 -250px;
5.      position:relative; 相对定位
6.      background-color:pink; //方便看效果
7.      left:50%;
8.      top:50%;
9.  }
```

## 20.css优先级算法如何计算？

- !important > id > class > 标签
- !important 比 内联优先级高
- 优先级就近原则，样式定义最近者为准;
- 以最后载入的样式为准;

## 21.Doctype作用？ 严格模式与混杂模式-如何触发这两种模式，区分它们有何意义？

- 1. 声明位于文档中的最前面，处于 标签之前。告知浏览器的

解析器，用什么文档类型 规范来解析这个文档。

- 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。

## 22.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

## 23.你能描述一下渐进增强和优雅降级之间的不同吗

- 渐进增强 progressive enhancement：针对低版本浏览器进行构建页面，保证最基本的功能，然后再针对高级浏览器进行效果、交互等改进和追加功能达到更好的用户体验。  
优雅降级 graceful degradation：一开始就构建完整的功能，然后再针对低版本浏览器进行兼容。

区别：优雅降级是从复杂的现状开始，并试图减少用户体验的供给，而渐进增强则是从一个非常基础的，能够起作用的版本开始，并不断扩充，以适应未来环境的需要。降级（功能衰减）意味着往回看；而渐进增强则意味着朝前看，同时保证其根基处于安全地带。

### “优雅降级”观点

“优雅降级”观点认为应该针对那些最高级、最完善的浏览器来设计网站。而将那些被认为“过时”或有功能缺失的浏览器下的测试工作安排在开发周期的最后阶段，并把测试对象限定为主流浏览器（如 IE、Mozilla 等）的前一个版本。

在这种设计范例下，旧版的浏览器被认为仅能提供“简陋

却无妨 (poor, but passable)” 的浏览体验。你可以做一些小的调整来适应某个特定的浏览器。但由于它们并非我们所关注的焦点，因此除了修复较大的错误之外，其它的差异将被直接忽略。

“渐进增强”观点

“渐进增强”观点则认为应关注于内容本身。

内容是我们建立网站的诱因。有的网站展示它，有的则收集它，有的寻求，有的操作，还有的网站甚至会包含以上的种种，但相同点是它们全都涉及到内容。这使得“渐进增强”成为一种更为合理的设计范例。这也是它立即被 Yahoo! 所采纳并用以构建其“分级式浏览器支持 (Graded Browser Support)”策略的原因所在。

## 24. 哪些css属性可以继承

letter-spacing、word-spacing、white-space、【line-height】、【color】、font、【font-family、font-size、font-style】、font-variant、【font-weight】、【text-decoration ( 文字的装饰 )】、text-transform、direction、text-indent ( 首行缩进 ) 和text-align ( 水平居中对齐 )、list-style ( 列表样式 )、( list-style-type、list-style-position、list-style ) -image

## 25. visibility 属性值有什么，是干嘛用的

- visible 默认值。元素是可见的。
- hidden 元素是不可见的。
- 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。

## 26.两列布局&三列布局

### 两列布局

```
1. <style>
2.     *{
3.         margin: 0;
4.         padding: 0;
5.     }
6.     .box1{
7.         width: 200px;
8.         height: 300px;
9.         background: aqua;
10.        float: left;
11.    }
12.    .box2{
13.        height: 400px;
14.        background: yellow;
15.        margin-left: 200px;
16.    }
17. </style>
18. </head>
19. <body>
20. <div class="box1"></div>
21. <div class="box2"></div>
22.
23.
24. </body>
```

### 三列布局



```
1. <style>
2.     .box1{
3.         width: 200px;
4.         height: 200px;
5.         background: red;
6.         float: left;
7.     }
8.     .box2{
9.         height: 300px;
10.        background: orange;
11.        margin: 0 200px;
12.    }
13.    .box3{
14.        width: 200px;
15.        height: 400px;
16.        background: lawngreen;
17.        float: right;
18.    }
19. </style>
20. </head>
21. <body>
22. <!--左右固定（宽度都为200px） 中间自适应-->
23. <div class="box1"></div>
24. <div class="box3"></div>
25. <div class="box2"></div>
26.
27. </body>
```

## 27.块级元素、行内元素、行内块级元素有哪些

- 1.块级元素: div、p、h1~h6、hr、ul、li、ol、dl、dt、dd、form、table、tr、td、thead、tbody、tfoot
- 2.行内元素: a、span、i、em、b、strong、u、del、s、sup、

sub、label

- **3.行内块级元素:** input、img、select

## display的相关属性

- none: 不显示，不占位置
- block: 显示，转为块级元素
- inline: 转为行内元素
- inline-block: 转为行内块级元素
- 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-column: 此元素会作为一个单元格列显示（类似 `<table>`）
- table-cell: 此元素会作为一个表格单元格显示（类似 `<table>` 和 `<tr>`）
- inherit: 规定应该从父元素继承 display 属性的值。