

## 第一部分 HTML&CSS整理答案

12

1. 什么是HTML5? 12
2. HTML5中什么是不同的新的表单元素类型? 12
3. HTML5的页面结构同HTML4或者更前的HTML有什么区别? 12
4. 哪些浏览器支持HTML5? 12
5. 为什么HTML5里面我们不需要DTD (Document Type Definition文档类型定义)? 13
6. HTML5的离线储存? 13
7. 浏览器是怎么对HTML5的离线储存资源进行管理和加载的呢? 13
8. 请描述一下 cookies, sessionStorage 和 localStorage 的区别? 13
9. 页面可见性 (Page Visibility) API 可以有哪些用途? 13
10. HTML5的form如何关闭自动完成功能? 13
11. 一个div, 要求实现当内容过少时, div的最低高度为200px, 当内容较多时, div的高度被内容撑开。13
12. 实现一个布局, 分左中右三栏。左栏固定宽为200px, 右栏固定150px, 中栏随屏幕宽自动适应。写出html和css。 13
13. a标签的四个伪类是什么? 如何排序? 为什么? 14
14. 如何实现浮动元素居中 14
15. 已知一个div内有一个img, 两者的高度均不知道, 但图片的高度一定小于div的高度。用css实现图片在div内的垂直居中。 14
16. HTML静态页面出现中文乱码如何解决? 15
17. 下列标签既是行内属性标签又可以设宽高的标签是? (D) 15
18. 用一两句话说一下你对“盒模型”这个概念的理解, 和它都涉及到哪些css属性。 15
19. 外边距、内边距、边框有几种书写形式, 列举说明。 15
20. 有上下两个div, 上一个div 设置margin-bottom: 10px; 下一个div设置margin-top: 5px; 那么两个div最后的间距是多少? 15
21. 写出下列CSS命令的最简写法{margin-left:20px; margin-bottom:50px; margin-right:20px; margin-top:20px;} 15
22. 我们用hack调兼容时, 用到如下命令 div{height: 300px; \*height: 400px; \_height: 500px;} , 在ie6里div显示高度是多少, 火狐里显示高度是多少? 15
23. 三层嵌套用在什么地方最合适, 有什么优点。 15
24. 背景图合并用在什么地方最适合, 有什么优点。 15
25. 页面如何在浏览器里达到居中, 并且左右自适应? 15
26. HTML5版本类型声明怎么写。有什么用? 15
27. XHTML1.0版本你知道么, 跟html5版本有什么区别 15
28. 书写ol, ul, table的嵌套规范 16
29. 前端页面有哪三层构成, 分别是什么? 作用是什么? 16
30. img的alt和title有什么区别? 16
31. HTML代码的书写规范有哪些? 16
32. 浏览器的调试工具有哪些? 都有什么功能? 16

33.table的合并边框属性是什么? 跨行是什么? 跨列是什么?	16
34.CSS是什么? 有什么用处。	16
35.你知道less,sass这些东西么 (小米) ;	16
36.解释W3C	16
37.页面重构	16
38.div+css与table布局的有何区别?	17
39.后台编码格式不是UTF-8怎么办?	17
40.一个CSS文件如果过大的话, 加载会很慢, 占用过大带宽, 如果解决?	17
41.input标签存在的兼容问题?	17
42.input中disabled与readonly有何区别?	17
43.input属性有哪些?	17
44.position属性值, 如只写了absolute, 是相对谁定的位?	17
45.CSS选择器中div.ps是什么意思	18
46.使用display:inline-block在IE6中不能正常显示, 如何解决?	18
47.png图片有几种格式	18
48.display的属性值有哪些?	18
49. 标签的隐藏 (display:none和visibility:hidden) 的区别	18
50. 在页面内居中 (水平和垂直方向) *3	18
51.如何让链接访问过后的hover消失	19
52.Ie6中为什么不能设置1px高的div	19
53.div中内容没有撑开高度的原因, 怎么解决	19
54.双倍边距bug	19
55.如何让div水平排列	19
56.定义id名和class名有什么区别	19
57.CSS有几种引入方式? link和@import有什么区别	19
58.在IE中给div设display:inline-block;它还会占一行不会?	20
59.如何制作渐变效果 (不使用CSS3)	20
60.上下两个div分别设置了margin-bottom和margin-top, 两个div之间的距离是多少?	20
61.除了ul、ol还用什么写列表	20
62.如果不写头部声明会有什么问题	20
63.如何优化你的页面	20
64.两栏自适应布局, 右侧div宽高不定	22
65.响应式布局是什么怎么用?	22
66.IE6 png的问题。	23
67.Hack是什么怎么用?	29
68.Border 虚边	29

69.li在ie6与ie8下的高度问题	29
70.Css常见兼容性问题，如何解决？	29
71.Css常见选择器有哪些？ 优先级？	29
72.你如何理解绝对定位和相对定位的？ 都用在什么地方？ 有什么优点和缺点？	29
73.左侧样式固定，右侧文本宽度自适应如何布局	29
看第五十六题面试题.	29
74.下面的布局如何实现.如何不用浮动还能使用什么布局方式	29
75.HTML静态页面出现中文乱码如何解决？	30
76.下列哪个选择器优先级是最高的？	30
77.写CSS命令，设置字体为宋体，12px大小，颜色为#eeeeee,加粗。	30
78.常用布局属性有哪些？ 有什么特点？	30
79.填空题	31
80.web网页中常见的图片格式有哪些？ 分别有什么特点？	31
81.在HTML中，SEO常见的白帽优化技巧有哪些？	31
82.块属性标签与行属性标签的区别？ 哪些标签是块属性的，哪些是行的？	31
83.IE6和IE7有什么不同？	32
84. 有三个兄弟div,设制样式div{float:left}如何使第二个div不浮动（不能用CSS3选择器）	32
85. 写出两个div并排显示的css样式，要求左侧的div宽度200px； 字体加粗； 行高30px； 与右侧div的外边距10px； 右侧的div宽度自适应，边框1px灰色；	32
86. css样式引用和js引入的位置问题（为什么会放在头部或者底部）。	33
87. 下列成立的是（ABCD）	33
88. 在下面的XHTML标签中，正确地标记折行的是（A）	33
89. 下面不支持overflow-y属性的浏览器是（A）	33
90. 读下面一段代码，相邻div的垂直外边距是多少？	33
91. 读下面一段代码，div#a和div#b哪个叠放层次更高，a覆盖b还是b覆盖a？	34
92. 标签语义化和CSS命名语义化有什么好处？	34
93. 某种效果，有两种实现方案都可以实现，方案一：<img>标签；方案二：背景图，两个方案如何取舍？ 衡量的标准是什么？	34
94. 应用DIV+CSS写出一个固定宽度的三列布局，能够使内容根据屏幕大小自适应居中。	35
95. CSS布局中，拥有布局(haslayout)有什么作用？ 怎样触发元素拥有布局？ 块级格式化范围(Block Formatting Context简称BFC)有什么作用？ 怎样触发元素BFC特性？	36
96. 你做的页面在哪些浏览器测试过？ 这些浏览器的内核是什么？ 经常遇到的浏览器的兼容性有哪些？ 怎么会出现？ 解决方法是什么？ （至少说3个）	36
97. 什么是CSS浮动问题？ 如何清除浮动？ 写出所有你知道的方法	36
98. 触发haslayout的方法	36
99. ie浏览器Hack的写法	37
100. 清除浮动的方法	37

101. 用css分别实现某个DIV元素上下居中和左右居中。	37
102. 表单中有一些checkbox该如何提交值, name一样吗	38
103. 你能描述一下渐进增强和优雅降级之间的不同吗?	38
104. 你如何对网站的文件和资源进行优化?	38
105. 如果你接到了一个使用Tab来缩进代码的项目, 但是你喜欢空格, 你会怎么做?	38
106. 你都使用那些工作来测试代码的性能?	38
107. 'data-'属性的作用是什么?	39
108. 描述css reset的作用和用途。	39
109. 解释css sprites以及如何使用。	39
110. 你使用CSS预处理器吗? 如果有请简单介绍一下	39
111. 谈谈你对WEB标准以及W3C的理解与认识。	39
112. 下面哪几项是HTML4.01中不建议使用的 (AC)	39
113. 你能描述一下你制作一个网页的工作流程吗?	40
114. 你能描述一下渐进增强和优雅降级之间的不同吗?	40
115. 请解释一下什么是语义化的HTML。	40
116. 你如何对网站的文件和资源进行优化?	40
117. 为什么利用多个域名来存储网站资源会更有效?	40
118. 请说出三种减低页面加载时间的方法	41
119. 什么是FOUC? 你如何来避免FOUC?	41
120. 文档类型的作用是什么? 你知道多少种文档类型声明方式?	41
121. 你常使用的HTML标签有哪些?	42
122. 说下你对标签语义化的理解。	42
123. 你知道的html 行级元素和块级元素标签分别有哪些?	42
124. 常用浏览器的内核有哪些?	42
125. 为什么要清除浮动, 有哪些方式?	43
126. SGML (标准通用标记语言) 和HTML (超文本标记语言), XML (可扩展标记语言) 和HTML的之间有什么关系?	43
127. Doctype作用? 严格模式与混杂模式如何区分? 它们有何意义?	44
128. 行内元素有哪些? 块级元素有哪些? 空(void)元素有那些?	44
129. 介绍一下CSS的盒子模型?	44
130. link和@import的区别是?	44
131. 如何居中div? 如何居中一个浮动元素?	45
132. 浏览器的内核分别是什么?	45
133. HTML语义化的理解?	45
134. iframe有那些缺点?	45
135. Label的作用是什么? 是怎么用的? (加 for)	45
136. 列出display的值, 说明他们的作用。position的值, relative和absolute定位原点是?	45

137.用纯 CSS 创建一个三角形	46
138.一个满屏”品”字布局 如何设计?	46
139.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?	47
140.经常遇到的浏览器的兼容性有哪些? 原因, 解决方法是什么, 常用hack的技巧?	47
141.为什么要初始化CSS样式。(面试官主要考的是对标签的默认样式了解)	47
142.absolute 的 containing block(容器块)计算方式跟正常流有什么不同?	47
143.对BFC规范的理解?	48
144.css定义的权重? (考的是css书写优先级)	48
145.解释下浮动和它的工作原理? 清除浮动的技巧	48
146.CSS优化、提高性能的方法有哪些?	49
147.浏览器是怎样解析CSS选择器的?	49
148.在网页中的应该使用奇数还是偶数的字体? 为什么呢?	49
149.margin和padding分别适合什么场景使用?	49
150.如何修改chrome记住密码后自动填充表单的黄色背景?	50
151.设置元素浮动后, 该元素的display值是多少?(自动变成display:block)	50
152.怎么让Chrome支持小于12px 的文字?	50
153.让页面里的字体变清晰, 变细用CSS怎么做?	50
154.font-style属性 可以让它赋值为“oblique” oblique是什么意思?	50
155.position:fixed;在android下无效怎么处理?	51
156.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)	51
157.display:inline-block 什么时候会显示间隙? (携程)	51
158.浅谈一下怎么处理浏览器的兼容性问题。	51
159.浏览器标准模式和怪异模式之间的区别是什么?	51
160.css hack知道哪些?	52
161.你对css盒模型的理解。	52
162.data-属性的作用是什么?	53
163.你能解释一下CSS的盒子模型么?	53
164.CSS3有哪些新特性 (包含哪些模块)?	53
165.请解释一下CSS3的Flexbox (弹性盒布局模型),已经适用场景?	54
166.用过媒体查询, 针对移动端的布局吗?	54
167.使用 CSS 预处理器吗? 喜欢那个?	55
168.元素竖向的百分比设定是相对于容器的宽度吗?	55
169.全屏滚动的原理是什么? 用到了CSS的那些属性?	55
170.什么是响应式设计? 为什么要做响应式设计? 响应式设计的基本原理是什么?	56
171.::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。本质上并不支持伪元素的双冒号(::)写法, 而是忽略掉了其中的一个冒号, 仍以单冒号来解析, 所以等同变相支持了E::before	56

## 第二部分：JS面试题

56

1. 获取一堆name名相同的div，点击弹出显示第几个div 56
2. abcdefgsdds 获取里面字面最多的字母（答案有很多） 57
3. abcdefghi 获取ghi怎么实现 57
4. 'asdjhklzxc' 截取'asd' 57
5. 翻转字符串 57
6. 删除数组第一个元素 57
7. [1,2,1,3,2,3,5,6,5] 去除重复元素 57
8. JS中有几种数据类型分别是哪几种？ 58
9. 说一下拖拽的原理？怎么样实现多个DIV在拖拽时有相撞让它们改变属性与属性值和他们的位置？（部分程度不太好的同学看到拖拽会懵，项目期有这个情况） 58
10. 用JS创建一个a标签，在body中显示，点击在新窗口打开 58
11. 鼠标滚轮动画 58
12. 定位右下角广告 58
13. 鼠标经过A A变大 其他的位置不动 59
14. JS中避免命名冲突的三个方法（有些同学或许会想不到这些概念） 59
15. JS在HTML中的引入方式有哪些？又有什么优缺点？ 59
16. 写出5条标识符的命名规范（） 59
17. 从JS的角度，写出8条以上优化代码的方法？ 59
18. 按优先级，对这几种运算符进行排序：！ && \* = > 59
19. 在循环中，break和continue语句的功能有何不同 59
20. 解释事件冒泡、事件捕获的区别 59
21. 说说什么叫做事件委托 59
22. 列举不少于6条的IE与FF脚本兼容性问题，需要写出命令（也许同学们缺乏总结，看着简单，实际很难答全） 59
23. 构造函数有何优缺点？ 60
24. 用自己的话描述一下“闭包” 60
25. 用自己的话描述一下对“异步”和“同步”的理解 60
26. 写倒计时 距离2018年还有 多少天多少小时多少分钟多少秒 60
27. alert怎样换行 61
28. 左侧内容可上下滚动，右侧是滚动条，如何实现布局 61
29. 向上滚动动画如何实现 61
30. form表单提交 61
31. javaScript与ECMAScript的区别？ 61
32. ==与===的区别？ 61
33. 什么是DOM？ 61
34. Undefined与null的区别？ 61

35. 有两个对象，var o = {num:1};var p = {num:2};要求书写一段代码，使得o+p===3为true。	61
36. 如果给你这么一个页面：顶部有logo什么的，下来是导航条，中上部有图片切换，右边是登录注册，左边有简单的信息展示，中下部是一些新闻，底部是一般的那些联系我们呀什么的传统页尾。这样一个布局加上所有的js效果和新闻链接包括所有的兼容，你大概能多久完成（新鸿儒）；	62
37. 讲解一下this（作用域）	62
38. 构造函数	62
39. 原型，原型链的理解	62
40. 说说前端优化从哪方面着手，具体说明有哪些？	62
41. 代码说明call和apply方法的区别（咱们这方面讲解的少，这样的题有变式，需要举例讲解一下）	62
42. function f(x){	62
43. var a=new Array(3);（实际上是闭包的用法）	62
44. var fn = function(i){	63
45. 对于js语言，下列说法中（C）是正确的？	63
46. 用两种方法，将第上题中的第一个p标签里的内容的颜色设置为红色	63
47. 如何测算数组的长度？	63
48. 滚动类效果中，会涉及到的定位属性有哪些，同时写出每个命令的含义。	63
49. 在放大镜效果中，获取鼠标位置的方法，同时说明鼠标位置是针对谁获取的？	63
50. 事件种类有哪些？分别举出不少于4个的例子？	63
51. 访问到tween算法中的Bounce的easeInOut方法？	63
52. Date对象中getDay()和getDate()分别是什么含义？取值范围是多少？	64
53. 设置当前时间为2004年12月12日的代码书写方法？（分写合写两种方式）	64
54. 写出不少于5种Math对象的方法，分别说明含义以及语法	64
55. 使用任意一种方法随机生成一个从100-300的整数。	64
56. tab切换效果 代码填空	64
57. 用JS截取标签中的abc	65
58. 用js如何获取一个上传文件的扩展名	65
59. 用js实现类、继承（继承构造函数内部的属性方法及原型的属性和方法）、私有变量	65
60. 写一个trim函数，兼容IE firefox chrome（正则）（怕部分学生想不到正则）	66
61. 列举事件绑定的方法（原生+JQ addEventListener与attachEvent的区别）addEventListener现代浏览器都支持，ie从ie9开始支持，可以绑定多个事件	66
62. addEventListener和通过on这种方式绑定的事件（例：ele.onclick = function(){})有什么区别	66
63. ajax请求时，如何解析json数据	66
64. 写一个获取非行间样式的函数	67
65. 闭包是什么，有什么特性，对页面有什么影响	67
66. 解释jsonp的原理，以及为什么不是真正的ajax（jsonp有些同学可能不知道）	67
67. 如何使用原生Javascript 代码深度克隆一个对象（注意区分对象类型）//对象的深复制	67
68. 截取字符串abcdefg的efg	68

69. 判断一个字符串中出现次数最多的字符，统计这个次数	68
70. IE与FF脚本兼容性问题	68
71. 规避javascript多人开发函数重名问题	69
72. javascript面向对象中继承实现	69
73. outerHTML和innerHTML区别	69
74. 编写一个方法 求一个字符串的字节长度（假设一个中文占两个字节）	70
75. 写出3个使用this的典型应用	70
76. 如何深度克隆	70
77. JavaScript中如何检测一个变量是一个String类型？请写出函数实现	71
78. 闭包	71
79. 请解释什么是Javascript的模块模式，并举出实用实例。	71
80. 你如何组织自己的代码？是使用模块模式，还是使用经典继承的方法	71
81. 你如何优化自己的代码？	71
82. 你能解释一下JavaScript中的继承是如何工作的吗？	71
83. 请说明navigator 常用属性？	71
84. 浏览器调试快捷键是什么？	72
85. 数组怎么去重？	72
86. 数组排序。	72
87. 常用变量声明规则？	73
88. call和apply的区别（动手验证）	73
89. 什么是闭包，在js中的作用是什么？	73
90. 请描述一下cookies、sessionStorage与localStorage的区别。（百度搜索）	73
91. JavaScript的数据类型都有什么？	74
92. 本地存储和事务存储之间的区别是什么？	74
93. 什么是事务存储？我们如何创建一个事务存储？	74
94. 本地存储的生命周期是什么？	74
95. HTML5中的本地存储概念是什么？	74
96. 我们如何从本地存储中添加和移除数据？	74
97. 如何实现浏览器内多个标签页之间的通信？(阿里)	75
98. 介绍JavaScript的基本数据类型。	75
99. 说几条写JavaScript的基本规范？	75
100. JavaScript有几种类型值？（堆：原始值和 栈：引用值），你能画一下他们的内存图吗？	76
101. Javascript如何实现继承？	76
102. 如何创建一个对象？（画出此对象的内存图）	77
103. 谈谈This对象的理解。	78
104. eval是做什么的？	78



105. null, undefined的区别?	78
106. ["1", "2", "3"].map(parseInt) 答案是多少?	79
107. 事件、IE与火狐的事件机制有什么区别? 如何阻止冒泡?	79
108. 什么是闭包 (closure) , 为什么要用它?	79
109. "use strict";是什么意思? 使用它的有什么好处或坏处?	80
110. new操作符具体干了什么呢?	80
111. Javascript中, 有一个函数, 执行时对象查找时, 永远不会去查找原型, 这个函数是?	80
112. 对JSON的了解?	80
113. 解释一下Math常用哪些方法?	81
114. js延迟加载的方式有哪些?	81
115. 知道ECMAScript 6里怎么写class么?	81
116. AMD (Modules/Asynchronous-Definition) 、CMD (Common Module Definition) 规范区别?	81
117. call() 和 .apply() 的区别?	82
118. JavaScript中的作用域与变量声明提升?	82
119. 如何编写高性能的Javascript?	82
120. 那些操作会造成内存泄漏?	82
121. 知道什么是webkit么? 知道怎么用浏览器的各种工具来调试和debug代码么?	83
122. 如何测试前端代码么? 知道BDD, TDD, Unit Test么? 知道怎么测试你的前端工程么(mocha, sinon, jasmine, qUnit..)?	83
123. 前端templating(Mustache, underscore, handlebars)是干嘛的, 怎么用?	84
124. 简述一下 Handlebars 的基本用法?	84
125. ie各版本和chrome可以并行下载多少个资源	84
126. 谈一谈你对 ECMAScript6 的了解?	84
127. 说说最近最流行的一些东西吧? 常去哪些网站?	85
128. 前端安全问题?	85
129. 如何判断当前脚本运行在浏览器还是node环境中? (阿里)	86
130. 把 Script 标签 放在页面的最底部的body封闭之前 和封闭之后有什么区别? 浏览器会如何解析它们?	86
131. 移动端的点击事件的有延迟, 时间是多久, 为什么会有? 怎么解决这个延时?	86
132. Node.js的适用场景?	86

## 第三部分 jQuery 框架 86

1. jQuery绑定事件的方法书写至少两种?	86
2. jQuery对象和Dom对象相互转换	86
3. Js用过的插件	86
4. jQuery如何获取元素?	86
5. JQuery链式写法原理	87
6. 用jQuery如何获取标签的id	87

7. 用jQuery如何获取标签<div></div>	87
8. 用jQuery如何获取标签内容	87
9. 用jQuery如何更换背景色	87
10. jQuery中，如何阻止事件冒泡和浏览器默认行为？	87
11. \$(document).ready与window.onload有何区别？	87
12. 列举jQuery的优势	87
13. jQuery中 :odd :gt 这两种选择器的含义以及用法举例	87
14. jQuery中，使用nth-child(3n+2)所选中的是以下哪些	87
15. 使用jQuery中的链式书写方法书写如下代码	88
16. jQuery中对动画的操作命令有哪些？举出不少于8种，并说明含义	88
17. jQuery中绑定事件的方法是什么，尝试书写例子：为id名为con的div绑定一个事件，事件为弹出一个数字0。获取标签时不要使用原生javascript方法。	88
18. jQuery中获取相对文档的坐标的方法是什么？	88
19. jQuery中对于节点的操作有哪些（举出不少于8种）？写出含义以及用法。	88
20. jQuery动画效果的实现	88
21. jQuery中的两个合成事件是_____和_____	89
22. 用jQuery获取标签“<div id='con'>div内部的信息</div>”当中的内容的代码为_____？	89
23. 举出不少于6种的jQuery中对动画的操作命令，并说明其含义：_____、_____、_____、_____、_____、_____	89
24. 编写一个jQuery的扩展方法（插件），让调用的对象设置宽度为200像素	89
25. 你使用过哪些Javascript库？	89
26. jQuery从哪个版本开始不兼容IE低版本，你项目中用的哪个版本，为什么用这个版本，其它版本的jQuery与你用的版本有什么区别？	90
27. 我们如何使用Canvas来画一条简单的线？手写出代码（注意：不是让你敲键盘，而是手写）	90
28. 针对 jQuery性能的优化方法？	90
29. jQuery与jQuery UI有啥区别？	91
30. jQuery和Zepto的区别？各自的使用场景？	91
31. Zepto的点透问题如何解决？	91

## 第四部分 AJAX 91

1. 如何实现局部内容和后台交互时的变化	91
2. 什么是AJAX，AJAX原理	91
3. 使用异步加载获取JS数据至少两种方法？	92
4. 什么是JSON，JSON的格式	92
5. 除了AJAX还有什么模板可以与后台交互	92
6. AJAX中的跨域问题：什么是跨域？如何解决跨域问题？	92
7. AJAX的流程是什么？	92
8. AJAX如何调用JSON数据	92

9. AJAX能够处理哪些格式的文件	92
10. AJAX实现表单验证用户注册流程	93
11. JSON数据的解析方法	93
12. 如何转化成JSON字符串	93
13. 说明异步请求的get和post方法的区别	93
14. 列举AJAX的优势	95
15. AJAX中创建请求的兼容性处理	95
16. 对于request.status的http状态码 用于表示网页服务器HTTP响应状态的3位数字代码。1,2,3,4,5开头的状态码 分别代表什么（提示：404页面表示禁止访问等）	95
17. 下列对get和post的区别描述错误的是（ ）	95
18. 如何将js对象转化成JSON字符串（ ）	96
19. AJAX流程 代码填空	96
20. AJAX调用JSON数据 代码填空	96
21. AJAX的意义，用法，优点和原理	97
22. Jsonp的意义，用法，优点和原理	97
23. 请尽可能详尽的解释AJAX的工作原理。	98
24. 你对Ajax的理解。	99
25. get与post两种方式的优缺点。	99
26. 一次完整的HTTP事务是怎样一个过程？	99
27. CORS是什么？对于跨域请求，如何将附带凭证（HTTP Cookie及HTTP认证信息）的请求发送至服务器？ （提示：后续真实工作中可能会用到）	99
28. 什么是Ajax和JSON，它们的优缺点	100
29. 浅谈一下如何避免用户多次点击造成的多次请求。	100
30. ajax是什么？	101
31. 同步和异步的区别？	101
32. 如何解决跨域问题？	101
33. 异步加载的方式有哪些？	101

# 第一部分 HTML&CSS整理答案

## 1. 什么是HTML5?

答：HTML5是最新的HTML标准。

注意：讲述HTML5推出的设计目的，以及现在市场的使用情况，浏览器支持情况等。。。。

设计目的

HTML5的设计目的是为了在移动设备上支持多媒体。新的语法特征被引进以支持这一点，如video、audio和canvas 标记。HTML5还引进了新的功能，可以真正改变用户与文档的交互方式，包括：

- 新的解析规则增强了灵活性
- 新属性
- 淘汰过时的或冗余的属性
- 一个HTML5文档到另一个文档间的拖放功能

## 2. HTML5中什么是不同的新的表单元素类型?

答：提示本地验证表单需要再外面添加form标签包起来，才能查看效果

color  
date  
datetime-local  
email  
time  
url  
range  
telephone  
number  
search

## 3. HTML5的页面结构同HTML4或者更前的HTML有什么区别?

答：一个典型的WEB页面包含头部，脚部，导航，中心区域，侧边栏。现在如果我们想在在HTML4的HTML区域中呈现这些内容，我们可能要使用DIV标签。

但是在HTML5中通过为这些区域创建元素名称使他们更加清晰，也使得你的HTML更加可读

以下是形成页面结构的HTML5元素的更多细节：（注意主要考的是标签语意化）

<header>：代表HTML的头部数据  
<footer>：页面的脚部区域  
<nav>：页面导航元素  
<article>：自包含的内容  
<section>：使用内部article去定义区域或者把分组内容放到区域里  
<aside>：代表页面的侧边栏内容

## 4. 哪些浏览器支持HTML5?

答：几乎所有的高版本浏览器Safari，Chrome，Firefox，Opera，IE8以上都支持HTML5

## 5. 为什么HTML5里面我们不需要DTD (Document Type Definition文档类型定义) ?

答: HTML5没有使用SGML或者XHTML, 他是一个全新的东西, 因此你不需要参考DTD, 对于HTML5, 你仅需放置下面的文档类型代码告诉浏览器识别这是HTML5文档即可

## 6. HTML5的离线储存?

答: localStorage 长期存储数据, 浏览器关闭后数据不丢失;  
sessionStorage 数据在浏览器关闭后自动删除。

## 7. 浏览器是怎么对HTML5的离线储存资源进行管理和加载的呢?

答: 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。

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

- (1) cookie在浏览器和服务端间来回传递, sessionStorage和localStorage不会。
- (2) sessionStorage和localStorage的存储空间更大。
- (3) sessionStorage和localStorage有更多丰富易用的接口。
- (4) sessionStorage和localStorage各自独立的存储空间。

## 9. 页面可见性 (Page Visibility) API 可以有哪些用途?

答: 在页面被切换到其他后台进程的时候, 自动暂停音乐或视频的播放。

## 10. HTML5的form如何关闭自动完成功能?

答: 给不想要提示的input是设置autocomplete=off即可。

## 11. 一个div, 要求实现当内容过少时, div的最低高度为200px, 当内容较多时, div的高度被内容撑开。

答: min-height:200px,兼容ie6的\_height:200px;

## 12. 实现一个布局, 分左中右三栏。左栏固定宽为200px, 右栏固定150px, 中栏随屏幕宽自动适应。写出html和css。

```
<div id=\"content\" class=\"clearfix\">
  <div id=\"left\">
  </div>
  <div id=\"middle\">
  </div>
```

```
<div id=\"right\">
</div>
</div>
```

```
#content{background:#ff6;}
#left,#right{width:200px;background:#fc0;float:left;}
#middle{float:left; width:100%;margin:0 -200px;}
#middle .inner{margin:0 200px;background:#ccc;}
#right{float:right;}
```

### 13.a标签的四个伪类是什么？如何排序？为什么？

爱恨分明原则: l v h a

注释：为了产生预期的效果，在 CSS 定义中，a:hover 必须位于 a:link 和 a:visited 之后！

注释：为了产生预期的效果，在 CSS 定义中，a:active 必须位于 a:hover 之后！！

注释：Pseudo-class（伪类）的名称对大小写不敏感。

注释：伪类可与 CSS 类配合使用：

```
a.red:visited {color: #FF0000;}
```

```
<a class="red" href="css_syntax.asp">CSS Syntax</a>
```

如果上面这个例子中的链接已访问过，那么它会显示为红色。

### 14.如何实现浮动元素居中

```
<div class="box">
<p>我是浮动的</p>
<p>我也是居中的</p>
</div>
```

```
.box{
float:left;
position:relative;
left:50%;
}
p{
float:left;
position:relative;
right:50%;
}
```

### 15.已知一个div内有一个img，两者的高度均不知道，但图片的高度一定小于div的高度。用css实现图片在div内的垂直居中。

参考：<http://www.jb51.net/css/76120.html>

16.HTML静态页面出现中文乱码如何解决？

答：<meta charset="UTF-8" />

17.下列标签既是行内属性标签又可以设置宽高的标签是？（D）

A. div B. span C. input D. img E. h1

18.用一两句话说一下你对“盒模型”这个概念的理解，和它都涉及到哪些css属性。

答：网页设计中常听的属性名：内容(content)、填充(padding)、边框(border)、边界(margin)，CSS盒子模式都具备这些属性。

这些属性我们可以用日常生活中的常见事物——盒子作一个比喻来理解，所以叫它盒子模式。

CSS盒子模型就是在网页设计中经常用到的CSS技术所使用的一种思维模型。

19.外边距、内边距、边框有几种书写形式,列举说明。

参考：<http://wangye.huseoblog.com/cssdivhezi.html>

20.有上下两个div，上一个div 设置margin-bottom: 10px；下一个div设置margin-top: 5px；那么两个div最后的间距是多少？

答：10px

21.写出下列CSS命令的最简写法{margin-left:20px; margin-bottom:50px; margin-right: 20px; margin-top:20px;}

答：margin: 20px 20px 50px;

22.我们用hack调兼容时，用到如下命令 div{height: 300px; \*height: 400px; \_height: 500px;}，在ie6里div显示高度是多少，火狐里显示高度是多少？

答：ie6:400px, 火狐：300px;

23.三层嵌套用在什么地方最合适，有什么优点。

答：有圆角边框不固定宽度同时兼容ie的时候

24.背景图合并用在什么地方最适合，有什么优点。

答：通常当遇到一个网站要加载大量icon(小图标)的时候，我们会把它合并成sprite(图片拼接)。目的是为了减少HTTP请求次数。这样做既能减少页面加载时间，又可以减轻服务器的负载

25.页面如何在浏览器里达到居中，并且左右自适应？

答：margin: 0 auto;

26.HTML5版本类型声明怎么写。有什么用？

答：<!DOCTYPE html>

27.XHTML1.0版本你知道么，跟html5版本有什么区别

答：XHTML 1.0 是 XML 风格的 HTML 4.01。

XHTML 1.1 主要是初步进行了模块化。

HTML5 是下一代 HTML，取代 HTML 4.01。

W3C 原本确实计划用 XHTML 系列替代 HTML 4.01，但 XHTML 系列实际上只活到了 1.1（1.1 和夭折的 1.2 已经体现出过分 XML 的迹象，而 W3C 的理想其实在疯狂的 XHTML 2 身上，当然，它没能诞生），还没脱离 HTML 4.01 的阴影就死了。

还没等 XHTML 兴起，它的地位就被 HTML5 取代了。

## 28.书写ol, ul, table的嵌套规范

答: <http://www.5icool.org/a/201308/a2081.html>

## 29.前端页面有哪三层构成, 分别是什么? 作用是什么?

答: 分成: 结构层、表示层、行为层。

结构层由 HTML 或 XHTML 之类的标记语言负责创建。标签, 也就是那些出现在尖括号里的单词, 对网页内容的语义含义做出了描述, 但这些标签不包含任何关于如何显示有关内容的信息。例如, P 标签表达了这样一种语义: “这是一个文本段。”

表示层由 CSS 负责创建。CSS 对“如何显示有关内容”的问题做出了回答。

行为层负责回答“内容应该如何对事件做出反应”这一问题。这是 Javascript 语言和 DOM 主宰的领域。

## 30.img的alt和title有什么区别?

答: alt 是图片加载失败时, 显示在网页上的替代文字; title 是鼠标放上面时显示的文字

## 31.HTML代码的书写规范有哪些?

答: <https://segmentfault.com/a/1190000003229217>

## 32.浏览器的调试工具有哪些? 都有什么功能?

答: <http://www.cnblogs.com/Excellent/p/5368812.html>

## 33.table的合并边框属性是什么? 跨行是什么? 跨列是什么?

答: 合并边框: cellpadding, 跨行: rowspan, 跨列: colspan

## 34.CSS是什么? 有什么用处。

答: 层叠样式表是一种用来表现HTML (标准通用标记语言的一个应用) 或XML (标准通用标记语言的一个子集) 等文件样式的计算机语言。

CSS 目前最新版本为 CSS3, 是能够真正做到网页表现与内容分离的一种样式设计语言。相对于传统 HTML 的表现而言, CSS 能够对网页中的对象的位置排版进行像素级的精确控制, 支持几乎所有的字体字号样式, 拥有对网页对象和模型样式编辑的能力, 并能够进行初步交互设计, 是目前基于文本展示最优秀的表现设计语言。CSS 能够根据不同使用者的理解能力, 简化或者优化写法, 针对各类人群, 有较强的易读性。

## 35.你知道less,sass这些东西么 (小米);

答: CSS 预处理器技术已经非常的成熟, 而且也涌现出了越来越多的 CSS 的预处理器框架。最为普遍的三款 CSS 预处理器框架, 分别是 Sass、Less CSS、Stylus。CSS 预处理器是一种语言用来为 CSS 增加一些编程的特性, 无需考虑浏览器的兼容性问题, 例如你可以在 CSS 中使用变量、简单的程序逻辑、函数等等在编程语言中的一些基本技巧, 可以让你的 CSS 更见简洁, 适应性更强, 代码更直观等诸多好处。

## 36.解释W3C

答: 万维网联盟创建于1994年, 是Web技术领域最具权威和影响力的国际中立性技术标准机构。到目前为止, W3C 已发布了200多项影响深远的Web技术标准及实施指南,

如广为业界采用的超文本标记语言 (标准通用标记语言下的一个应用)、可扩展标记语言 (标准通用标记语言下的一个子集) 以及帮助残障人士有效获得Web内容的信息无障碍指南 (WCAG) 等,

有效促进了Web技术的互相兼容, 对互联网技术的发展和作用起到了基础性和根本性的支撑作用。

## 37.页面重构

答: 网站重构不是一种技术, 不是css+div, 更不是标准, 网站重构是一种思想, 是一种理念。

引用WebReBuild.ORG 的话: 当前国内的同行普遍认为: 所谓的网站重构就是“DIV+CSS”, 想法固然极度局限。但也不是另一部分的人认为是“XHTML+CSS”, 因为“XHTML+CSS”只是页面重构。真正的网站重构



理应包含结构、行为、表现三层次的分离以及优化，行内分工优化，以及以技术与数据、人文为主导的交互优化等。

重构网站先重构人，重构你的理念，不要光光追求技术，追求还原设计稿，追求浏览器的兼容性，重要的是基础和理念。当你真正了解什么是网站重构的时候网站重构也就真正开始了。

### 38.div+css与table布局的有何区别？

答: div+css:布局简洁明了，使用方便，相对于表格来说更容易搜索的到！优化程度高点。不易出现错误或者不可控！样式繁多，易控易修改。

Table: 表格布局不建议使用，这个算是快要被淘汰的一种布局方式。布局不容易调整和规划。容易失控。大部分都不能使用太多样式。

### 39.后台编码格式不是UTF-8怎么办？

答: 统一编码格式

### 40.一个CSS文件如果过大的话，加载会很慢，占用过大带宽，如果解决？

- 答: 1.去除空格和换行,压缩css代码;  
2.尽量使用简写, 缩减代码;  
3.将css文件分成多个文件

### 41.input标签存在的兼容问题？

答: 当input标签在type为text时，在Firefox和Safari中的默认高度为22像素（包括上下边框）宽度为146像素（包括左右边框），而在IE中的默认高度为24像素，而宽度却和Firefox和Safari是一致的，也是146像素。

当 input标签在type为submit时，在Firefox中的高度为23像素（包括阴影），宽度为75像素。在Safari中高度为21像素，宽度为73像素，在IE中高度为25像素、宽度为73像素。

### 42.input中disabled与readonly有何区别？

答: Readonly只针对input(text / password)和textarea有效，而disabled对于所有的表单元素都有效，包括select, radio, checkbox, button等。

表单元素在使用了disabled后，当我们将表单以POST或GET的方式提交的话，这个元素的值不会被传递出去，而 readonly会将该值传递出去(这种情况出现在我们将某个表单中的textarea元素设置为disabled或readonly，但是submit button却是可以使用的)。

### 43.input属性有哪些？

值	描述
button	定义可点击按钮（多数情况下，用于通过 JavaScript 启动脚本）。
checkbox	定义复选框。
file	定义输入字段和 "浏览"按钮，供文件上传。
hidden	定义隐藏的输入字段。
image	定义图像形式的提交按钮。
password	定义密码字段。该字段中的字符被掩码。
radio	定义单选按钮。
reset	定义重置按钮。重置按钮会清除表单中的所有数据。
submit	定义提交按钮。提交按钮会把表单数据发送到服务器。
text	定义单行的输入字段，用户可在其中输入文本。默认宽度为 20 个字符。

### 44.position属性值，如只写了absolute，是相对谁定的位？

答:离它最近的已经定位的父元素

## 45.CSS选择器中div.ps是什么意思

答: 类名是ps的div

## 46.使用display:inline-block在IE6中不能正常显示，如何解决？

答: **方法1**: 直接让块元素设置为内联对象呈递（设置属性 display:inline），然后触发块元素的 layout（如：zoom:1 等）。兼容各浏览器的代码如下：`div {display:inline-block;*display:inline; *zoom:1;...}`

**方法2**: 先使用 display:inline-block 属性触发块元素，然后再定义 display:inline，让块元素呈递为内联对象（两个display 要先后放在两个 CSS 样式声明中才有效果，这是 IE 的一个经典 bug，如果先定义了 display:inline-block，然后再将 display 设回 inline 或 block，layout 不会消失）。代码如下（…为省略的其他属性内容）：`div {display:inline-block;...}div {*display:inline;}`

## 47.png图片有几种格式

答: png有3种不同深度的格式: png8、png24、png32

## 48.display的属性值有哪些？

值	描述
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。

## 49. 标签的隐藏（display:none和visibility:hidden）的区别

答: visibility:hidden隐藏，但在浏览时保留位置；CSS display:none视为不存在，且不加载！

## 50. 在页面内居中（水平和垂直方向）\*3

答: 垂直居中:水平方向给父级设置text-align;使用左右margin值为auto

竖直方向:设置line-height值为元素高度;设置元素绝对定位,top:0;bottom:0 margin:auto, 0;

## 51.如何让链接访问过后的hover消失

正常情况下: 爱恨原则: l —> v —> h —> a

为了达到上述效果,改变顺序即可: l —> h —> v —> a

## 52.Ie6中为什么不能设置1px高的div

答: 在用DIV构建网页的时候, 有时候需要的高度很小, 这时候就可能会出现这个问题, 因为, IE6下DIV有个默认的高度, 大约10-12px。当你试图定义一个高度小于这个默认值的 div 的时候, IE 会固执的认为这个层的高度不应该小于字体的行高。

解决办法:

第一: 定义该DIV字体大小。

例如:

```
<div style="height:1px;font-size:0;"></div>
```

第二:

直接限制自动调整

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

## 53.div中内容没有撑开高度的原因, 怎么解决

父亲div没有设置高度

孩子div设置高度了,

此时,父亲的高度是靠孩子撑起来的

但是当孩子浮动后,

父亲的高度也就没了

此时需要给孩子增加一个兄弟div,并且clear:both,问题完美解决

## 54.双倍边距bug

连接地址:[http://www.360doc.com/content/14/1224/20/21166337\\_435502508.shtml](http://www.360doc.com/content/14/1224/20/21166337_435502508.shtml)

在产生双边距bug的元素内,增加一行属性:display:inline;

第二种办法:采用cssHack;margin-left:20px;\_margin-left:20px;

## 55.如何让div水平排列

1:浮动

2:定位

## 56.定义id名和class名有什么区别

从概念上来说:

id是先找到结构/内容, 再给它定义样式; class是先定义好一种样式, 再套给多个结构/内容。

从样式效果上来说:

id的优先级要比class高出一个层次

html中不管有几个id,

在css获取到的就是所有的

但是在js中通过document获取到的是第一个

## 57.CSS有几种引入方式? link和@import有什么区别

link是XHTML标签，除了加载CSS外，还可以定义RSS等其他事务；@import属于CSS范畴，只能加载CSS。

link引用CSS时，在页面载入时同时加载；@import需要页面网页完全载入以后加载。

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

## 58.在IE中给div设display:inline-block;它还会占一行不会?

答: ie不支持display属性

## 59.如何制作渐变效果（不使用CSS3）

这里的关键点是h1 { position: relative } 和h1 span { position: absolute }

```
h1 {  
  font: bold 330%/100% "Lucida Grande";  
  position: relative;  
  color: #464646;  
}  
h1 span {  
  background: url(gradient.png) repeat-x;  
  position: absolute;  
  display: block;  
  width: 100%;  
  height: 31px;  
}
```

是的，就这些，你已经搞定了

## 60.上下两个div分别设置了margin-bottom和margin-top，两个div之间的距离是多少?

会发生融合效果,距离就是数值比较大的那个

## 61.除了ul、ol还用什么写列表

dl dt dd

## 62.如果不写头部声明会有什么问题

通常html DOCTYPE声明是必须的，而且使用div+css更是必不可少，如果缺少或错误document将会造成你的CSS失效或半失效，即因为css失效，网页布局变乱，有的css属性不能体现。

另外:由于万恶的IE（尤其指IE6和IE7），我们在页面重构时不免要对其进行各种bug修复及差异化处理。在标准浏览器[1]中可实现的效果在IE里却有各种离奇问题，例如IE6、IE7不能良好应对的inline-block和.clearfix问题。

所以就加入了IE版本区分代码，主要还是解决IE版本兼容性问题。

## 63.如何优化你的页面

一、提倡前端开发工程师在书写xhtml的时候做到结构语义化。

结构中主要包括了head和body两个部分，但是我们经常说的是结构语义化主要是body中的标签，但是我在这里还是简单的说一下head，head中其实包括了一些对于我们seo很有用的一些东西，比如title,Description,Keywords,这些东西在蜘蛛抓取的时候都是有帮助的，当然，还有其他的一些，我在此就不一一

说明了，比如设置缓存等一些其他的信息。

那么body中的话，包括的标签就很多了，我觉得作为一个合格的前端开发人员你应该去熟悉他们，比如div,span,h,ul,ol,dl,p等等这类的标签的使用。应该非常合理，还有就是注意h标签的断层，及h1标签的使用，这些都是非常重要的。

同时在我们的结构中不要出现style和onclick这样的内联的样式和事件。希望大家能够注意结构与表现、行为的分离。

## 二、css，js文件数量及大小的优化

那么关于css、js的优化的话，一般情况下建议css和js采用外联式。但是如果你的页面内容比较多，设计师把整个效果做得比较花的话，恐怕css就非常多了，那么这种情况下，你一定要把你的css规划好，尽量采用缩写，这样可以减少css文件的大小，那么对css做相应的规划也可以减少css的个数，减少http请求数，js同理。

## 三、背景图片数量及大小的优化

由于我们的背景图片数量比较多，这样的话，会给服务器带来影响，增加了http请求数，我们是否有一种好的解决办法呢？这个答案是肯定的，如果你是一个合格的前端开发，你应该清楚，在我们的css定义背景的时候，可以通过坐标来实现对背景进行定位的，既然如此，那么我们可以将这些背景合并起来，这样即可减少http请求数，同时，我们在背景整合的时候，也需要考虑图片质量，同时也需要考虑图片的大小。

你的背景图片保证不超过3个以上，你的css文件不超过2个，js文件不超过3个。而且良好的遵守web标准的一些规定，css放到head中，js文件放到</body>之前或者之后。

建议body中增加text-align:center

用 <link> 代替 @import

作为大型网站来说，首页使用内联式样式表，这样可以减少http请求数的同时，也可以防止裸奔。当然其他页面需要使用外联样式表，这样才可以方便维护。因为作为大型网站来说，他的首页访问量是非常的大，所以。。

把样式表置于顶部

把脚本置于页面底部

避免使用 CSS 表达式 (Expression)

使用外部 JavaScript 和 CSS

削减 JavaScript 和 CSS

用 <link> 代替 @import

避免使用滤镜

剔除重复脚本

减少DOM访问

开发智能事件处理程序

最好的方案就是按照 HTML 规范在文档 <head /> 内加载你的样式表。

对于拥有较大浏览量的首页来说，有一种技术可以平衡内置代码带来的 HTTP 请求减少与通过使用外部文件进行缓存带来的好处。其中一个就是在首页中内置 JavaScript 和 CSS，但是在页面下载完成后动态下载外部文件，在子页面中使用到这些文件时，它们已经缓存到浏览器了。

Cookie:

减小Cookie体积

对于页面内容使用无cookie域名

图片:

优化图像

优化CSS Sprite

不要在HTML中缩放图像

favicon.ico要小而且可缓存

优化前端网站:[http://blog.sina.com.cn/s/blog\\_630dca5b0101koe3.html](http://blog.sina.com.cn/s/blog_630dca5b0101koe3.html)

## 64.两栏自适应布局，右侧div宽高不定

```
<div class="main">
    <div class="content">右侧主体自适应区块</div>
</div>
<div class="sitebar">
    左侧定宽200px区块
</div>

.main,.sitebar{
    height: 300px;
    font: bolder 20px/300px "微软雅黑";
    color: white;
    text-align: center;
}
.main{
    width: 100%;
    float: left;
}
.main .content{
    margin-left: 200px;
    background: red;
}
.sitebar{
    width: 200px;
    float: left;
    margin-left: -100%;
    background-color: green;
}
```

## 65.响应式布局是什么怎么用?

用通俗的话来讲，所谓的响应式布局（Responsive Design），简单可以理解为自适应屏幕大小进行展示，并且获得比较好的体验。

其中，浏览器的兼容是很大的一个技术点，而布局设计是一个设计难点。如果将响应式布局简单理解为浏览器兼容又大合适，这样的技术在很多年前就已经有提出并有解决方案。以前的浏览器兼容，屏幕大小是相似的，操作习惯是相同的，所以兼容只要不用浏览器或者平台展示的样子一模一样就可以。

而响应式布局并不是说要展现得一样。比如说菜单，在PC可以是左右布局，直接展示。而在手机上，不应该是PC上的缩小版，而应该是菜单屏展示，类似手机原生应用，这就是所谓的不同媒介的适应性体验。

备记之

## 66.IE6 png的问题。

<http://www.cnblogs.com/yuzhongwusan/archive/2009/04/17/1438119.html>

<http://18sucai.com/article/338.htm>

### 方案1 - 纯CSS解决方案：

介绍：虽说是纯CSS解决方案，但是也使用了JavaScript来运算，只不过是脚本写到了CSS文件中，遗憾的是，此方案只支持img标签，对背景图片无效。

目录说明：

思路：

1、首先下载透明的图片文件

2、在需要设置透明的样式中加入下方代码，其中蓝色标注代码为刚才下载的透明图片，路径同样还是相对于HTML文件的位置（不相对于CSS文件！）：

```
img
{
    _azimuth:expression(this.pngSet?this.pngSet=true:(this.nodeName == "IMG" &&
this.src.toLowerCase().indexOf('.png')>-1?(this.runtimeStyle.backgroundImage = "none",this.runtimeStyle.filter =
"progid:DXImageTransform.Microsoft.AlphaImageLoader(src='" + this.src + "', sizingMethod='image')",this.src =
"images/blank.gif");(this.origBg = this.origBg?
this.origBg :this.currentStyle.backgroundImage.toString().replace('url(',').replace(')',''),this.runtimeStyle.filter =
"progid:DXImageTransform.Microsoft.AlphaImageLoader(src='" + this.origBg + "',
sizingMethod='crop')",this.runtimeStyle.backgroundImage = "none")),this.pngSet=true);
}
```

优点：

CSS代码看起来似乎很优雅，至少没有乱七八糟的文件了，基本没有外加的文件，效率还算不错。

缺点：

- 1、多引入了一个本不应该存在的blank.gif图片文件；
- 2、不支持背景图即Background；
- 3、当文件载入之前，会先暂时呈现灰底；
- 4、不支持Hover等伪类；

使用情况：

- 1、大部分透明的png存在于img标签中时可考虑；
- 2、如果有背景图的可以参考上面所说的支持背景图的两种方式；

### 方案2 - 滤镜解决方案：



介绍：滤镜从IE4.0被微软正式引入，所以我们可以使用滤镜解决IE6的PNG透明问题，滤镜不仅可以实现目前CSS3的一些旋转效果而且还可以引入图片。注意：此方法在部分版本的IE Test中无效，建议使用标准的IE6来进行测试！

目录说明：

思路：

1、书写正常的CSS代码，通过background导入图片，这样所有的浏览器均使用了此PNG图片；

```
background:url(..../images/W3CfunsLogo.png);
```

2、通过滤镜对引入图片，滤镜引入图片的时候是相对于HTML文件，而不是相对于CSS文件，语法如下：

```
filter:progid:DXImageTransform.Microsoft.AlphaImageLoader(src="images/W3CfunsLogo.png");
```

代码写到这里，我们放到IE6下测试后发现IE6还是没有透明，因为我们虽然设置了滤镜引入图片，但是background也同样加载了此图片，又因为background的图层比滤镜设置的高，所以才没有显示出来，如下图：



3、所以我们得出的结论就是当我们使用filter的时候，就要使background失效，因此我们可以使用CSSHack来解决此问题（如果您不知道IE6的CSSHack如何使用的话，请看[这里](#)！），只需要将IE6的background:none;即可，那么可以得出的代码如下：

```
_background:none; /*此代码只有IE6识别*/
```

又因为filter只在IE6下让其产生作用，IE6+版本的浏览器虽然也识别filter，但是png透明是没有灰底问题的，所以我们同样将filter也加上IE6 Hack即可。

4、最终我们可以得到如下代码：

```
#pics
{
background:url(..../images/W3CfunsLogo.png) no-repeat;
/*以下为IE6设置PNG透明代码*/
```



```
_background:none;
_filter:progid:DXImageTransform.Microsoft.AlphaImageLoader(src="images/W3CfunsLogo.png");
}
```

提示：如果需要使其支持链接的hover，那么需要在CSS中定义：cursor:pointer;使其呈现手型，否则将为默认的鼠标状态。

优点：

- 1、绿色无插件；
- 2、效率高，速度快；
- 3、网速慢的时候，不会出现先灰底再透明的情况，支持远程图片；
- 4、支持Hover等伪类，但是得使用两张图片，网速慢的情况下，会导致第二张图片暂时无法显示，因为还没有完全载入；

缺点：

- 1、不支持平铺，虽然filter有sizingMethod="scale", 拉伸缩放模式，但是图片会变形，如果单纯的颜色或简单的渐变色还能横向平铺；
- 2、不支持Img标签；
- 3、不支持CSS Sprite；

使用情况：

- 1、当没有img引入png时可考虑；
- 2、当没有CSS Sprite需求时可考虑；
- 3、当没有平铺需求时候可考虑；

### 方案3 - HTC插件解决方案：

介绍：从IE 5.5版本开始，Internet Explorer（IE）开始支持Web 行为的概念。这些行为是由后缀名为.htc的脚本文件描述的，它们定义了一套方法和属性，程序员几乎可以把这些方法和属性应用到HTML页面上的任何元素上去。

目录说明：

思路：

- 1、首先下载压缩文件 htc.zip (2.27 KB, 下载次数: 108)
- 2、复制并粘贴iepngfix.htc和blank.gif到您的网站文件夹中。
- 3、在需要使用的PNG标签上定义如下，相对于HTML文件的位置（不相对于CSS文件！）。例如，你可能看起来像这样：

```
<style type="text/css">
img,div{behavior:url(style/iepngfix.htc);}
</style>
```

- 5、如果您的网站使用的子文件夹，打开。HTC文件，大约在第16行更改blankImg变量，修改blank.gif路径像这样：同样路径相对于HTML文件的位置（不相对于CSS文件！）。

```
IEPNGFix.blankImg = "images/blank.gif";
```

- 6、复制并粘贴iepngfix.htc和blank.gif到您的网站文件夹中。

```
<script type="text/javascript" src="js/iepngfix_tilebg.js"></script>
```

- 7、由于此js只有使用IE6时才有用，所以为了让我们的页面更加高效的执行，我们可以将上方代码修改如下，只有IE6的时候才调用执行此JavaScript：

```
<!--[if IE 6]><script type="text/javascript" src="../js/iepngfix_tilebg.js"></script><![endif]-->
```

优点：

1、一次性配置好，只需要像平时一样引入png图片，也不需要考虑png相对于html路径的问题，当目录有所变化，只需要修改htc文件或css中htc文件路径即可。

- 2、支持平铺属性。
- 3、不支持Img标签；
- 4、不支持Hover等伪类；

缺点：

- 1、多引入了js、图片和htc，共三个文件；
- 2、不支持CSS Sprite；
- 3、当文件载入之前，会先暂时呈现灰底；

使用情况：

- 1、当没有img引入png时可考虑；
- 2、当没有CSS Sprite需求时可考虑；
- 3、PNG图片比较频繁修改时可考虑；

方案4 - jQuery解决方案：

介绍：jQuery为我们带来了很大的方便，jQuery没有让我们有太大的失望，img和png都同时得以支持，唯一美中不足的还是无法平铺，无法使用CSS Sprite。

目录说明：

思路：

- 1、首先下载此方案所用到的js文件和透明gif jQueryPngFix.zip (2.7 KB, 下载次数: 139)

2、找到js文件中找到blankgif: 'images/blank.gif'，将路径修改为相对于HTML文件的位置（不相对于CSS或js文件！）

3、由于此js只有使用IE6时才有用，所以为了让我们的页面更加高效的执行，我们可以将上方代码修改如下，只有IE6的时候才调用执行此JavaScript：

```
<!--[if IE 6]><script type="text/javascript" src="js/pngfix.js"></script><![endif]-->
```

优点：

- 1、CSS代码看起来很优雅，只需要引入js进行简单的配置一下就行了，效率还算不错；
- 2、支持背景图，支持img；

缺点：

- 1、额外加入了js文件和图片文件，增加http请求；
- 2、加载了一个庞大的jQuery类库；
- 3、多库共存的时候可能会出现问题；
- 4、不支持平铺；
- 5、不支持CSS Sprite；
- 6、当文件载入之前，会先暂时呈现灰底；
- 7、不支持Hover等伪类；

使用情况：

当您的项目中使用jQuery的时可以考虑；

jQuery解决方案 - DEMO入口

方案5 - 原生JavaScript解决方案：

介绍：利用了方案1的滤镜原理来实现，但由于此javascript没有读取css文件中的样式，所以此方案同样只支持img标签，对背景图片无效。

目录说明：

思路：

1、首先下载透明此方案所用到的js文件 [iepngfix.zip](#) (1.25 KB, 下载次数: 144)

2、由于此js只有使用IE6时才有用，所以为了让我们的页面更加高效的执行，我们可以将上方代码修改如下，只有IE6的时候才调用执行此JavaScript：

```
<!--[if IE 6]><script type="text/javascript" src="js/iepngfix.js"></script><![endif]-->
```

优点：

代码看起来似乎很优雅，基本没有外加的文件，效率还算不错。

缺点：

- 1、额外加入了js文件，增加http请求；
- 2、不支持背景图即Background；
- 3、当文件载入之前，会先暂时呈现灰底；
- 4、不支持Hover等伪类；

使用情况：

- 1、大部分透明的png存在于img标签中时可考虑；
- 2、如果有背景图的可以参考上面所说的支持背景图的两种方式；

#### 方案6 - EvPng解决方案：

介绍：此方案与第七种方案差不多，使用方法也如出一辙，效果也非常不错。

目录说明：

思路：

- 1、首先下载此方案所用到的文件， [EvPng.zip](#) (3.39 KB, 下载次数: 128)
- 2、参考第七种方案的使用方法。

优点：

- 1、CSS代码无需任何修改，按照平时的思路来写即可；
- 2、无需配置；
- 3、没有多余的gif图片；
- 4、支持img；
- 5、支持平铺；
- 6、支持CSS Sprite；
- 8、支持Hover等伪类；

缺点：

- 1、额外加入了js文件（文件4.93k，比DD\_belatedPNG的6.39k还小）和http请求，可以忽略不计；
- 2、当文件载入之前，会先暂时呈现灰底；
- 3、js文件过多的时候，可能会报错，导致js无法正常运行（这种情况极少出现，可以忽略不计）；
- 4、使用CSS Sprite技术的hover效果在部分情况下top可能会有1像素的偏差。

使用情况：

- 1、当前7种方法均不能解决问题的时候可考虑；
- 2、当DD\_belatedPNG效果不理想的时候可以考虑；

#### 方案7 - DD\_belatedPNG解决方案：

介绍：我们都知道在目前所用的png图片透明解决方案基本都是使用滤镜、xpression解决的、透明gif替代。但是这些方法都有一个缺点，就是不支持CSS中background-position与 background-repeat。而这次的js插件使用了微软的VML语言进行绘制且不需要引入其他文件，一个小小的js就可以完美解决png图片 bug就连img标签和hover伪类也可以很好的解决。

目录说明：

思路：

1、首先下载此方案所用到的文件， [DD\\_belatedPNG.zip \(3.76 KB, 下载次数: 254\)](#)

2、引入刚下载的js文件，同样由于此js只有使用IE6时才有用，所以为了让我们的页面更加高效的执行，我们可以将上方代码修改如下，只有IE6的时候才调用执行此JavaScript：

```
<!--[if IE 6]><script type="text/javascript" src="js/DD_belatedPNG.js"></script><![endif]-->
```

3、调用函数，设置参数如下：

```
DD_belatedPNG.fix("#pngImg,#pics,#picsRepeat");
```

其中传入的参数为所使用png图片的标签的ID、类样式和标签名称，同样也可以按照下方这样来写

```
DD_belatedPNG.fix("#content img");
```

此方法则表示#content下的所有img标签透明

如果为链接和链接的hover设置透明，那么您按照下方这么来写，在部分版本里面可以不用加入: hover直接写选择器即可，但是为了保险，建议咱们还是加上: hover：

```
DD_belatedPNG.fix("#links,#link: hover");
```

写到这里并且您使用过jQuery或者CSSQuery类库，那么您一定熟悉上面的这种选择方法，总之就是，在CSS中您是如何选择的元素，那么在这个js函数（方法）中传入什么，只不过多个选择的时候，使用逗号隔开即可。

KwooShung用此方法时的小技巧：如果页面中存在很多png，DD\_belatedPNG.fix()函数的参数岂不是很长？我们可以使用这种写法：

```
DD_belatedPNG.fix(".pngFix,.pngFix: hover");
```

如果使用上述的写法，我们的html中只需要在相对应的标签上加入class="pngFix"就行了，如果有多个类样式，按照平时的多个类样式的写法即可class="abc cbc pngFix"，

使用此方法的时候，我们每次都要加载两个js文件或者写两个<script>标签才行，这样不太好，http请求会增多，那么我们可以打开DD\_belatedPNG.js文件，在尾部加入如下代码即可：

```
window.onload = function()
{
    DD_belatedPNG.fix(".pngFix,.pngFix: hover");
}
```

这样我们只需要引入此[JS](#)，在需要透明的标签上加入class="pngFix"即可，简单 · 方便 · 快捷！

优点：

- 1、CSS代码无需任何修改，按照平时的思路来写即可；
- 2、无需配置；
- 3、没有多余的gif图片；
- 4、支持img；
- 5、支持平铺；
- 6、支持CSS Sprite；
- 8、支持Hover等伪类；

缺点：

- 1、额外加入了js文件（6.39k）和http请求，可以忽略不计；
- 2、当文件载入之前，会先暂时呈现灰底；
- 3、js文件过多的时候，可能会报错，导致js无法正常运行（这种情况极少出现，可以忽略不计）；

使用情况：

- 1、当前6种方法均不能解决问题的时候可考虑；

2、当png图片过多的时候可考虑，因为png图片太多，使用前面的几个方法，有的会导致CSS代码冗余过多，还不如引入此文件划算；

方案8 - PNG8格式的图片解决方案：

介绍：png8和gif都是8位的透明度，IE6与生俱来就支持png8的索引色透明度，但不支持png或8位以上的alpha透明度。而对于非动画的GIF建议你使用PNG8，因为体积会更小~

思路：一个最简单也最保险的方法让IE6支持PNG图片透明（小小的分享一下）

## 67.Hack是什么怎么用？

<http://www.kwstu.com/Admin/ViewArticle/201409011604277330>

## 68.Border 虚边

border: 1px dashed red;

## 69.li在ie6与ie8下的高度问题

在li样式上加上vertical-align:bottom;

## 70.Css常见兼容性问题，如何解决？

<http://www.zhufengpeixun.cn/CSS/2011-08-25/142.html>

## 71.Css常见选择器有哪些？ 优先级？

选择器:<http://www.runoob.com/cssref/css-selectors.html>

优先级:id > class > 层级选择器和标签选择器

## 72.你如何理解绝对定位和相对定位的？都用在什么地方？有什么优点和缺点？

绝对定位就是你的位置已经不属于你了，你只能漂浮在半空中。

相对定位就是你的位置你还占用的，人还在飘着

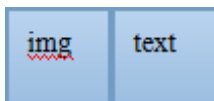
有意思的是这些都是相对父辈元素有position为相对或者绝对属性来定位的，都找不到的话就以body窗口来定位

因为绝对定位（和文档流没关系）如无申明，则其是对与body而言的，处理不好的话。如显示器尺寸变了，可能就会变形。

相对定位的元素属文档流，所以稳定的，相对定位是相对他该出现的位置，如无设top left之类，和普通div基本一样。

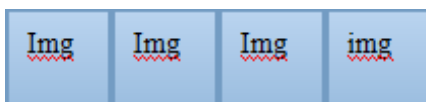
一般的绝对定位是这样用的，父元素要是相对定位的且须有布局，如有个高度，这样子元素用绝对定位，就可以相对它的父元素进行绝对定位，父元素若不这样做，那么子元素用绝对定位其实是相对body定位

## 73.左侧样式固定，右侧文本宽度自适应如何布局



看第五十六题面试题.

## 74.下面的布局如何实现.如何不用浮动还能使用什么布局方式



定位. 或则flex-box

## 75.HTML静态页面出现中文乱码如何解决?

答:<meta charset="UTF-8" />

## 76.下列哪个选择器优先级是最高的?

a. #a   b. .a .b .c .d .e .f .g .h .i .j .k .l .m .n .o   c. #a .b   d. div.b#a

答:d

## 77.写CSS命令，设置字体为宋体，12px大小，颜色为#eeeeee,加粗。

一、字体属性主要包括下面几个

font-family, font-style, font-variant, font-weight, font-size, font

font-family (字体族): "Arial"、"Times New Roman"、"宋体"、"黑体"等;

font-style (字体样式): normal (正常)、italic (斜体) 或 oblique (倾斜);

font-variant (字体变化): normal (正常) 或 small-caps (小体大写字母);

font-weight (字体浓淡): 是 normal (正常) 或 bold (加粗)。有些浏览器甚至支持采用100到900之间的数字 (以百为单位);

font-size (字体大小): 可通过多种不同单位 (比如像素或百分比等) 来设置, 如: 12xp, 12pt, 120%, 1em

如果用 font 属性的话, 就可以把几个样式进行简化, 减少书的情况; font 属性的值应按以下次序书写(各个属性之间用空格隔开):

顺序: font-style | font-variant | font-weight | font-size | line-height | font-family

二、font的简写实例

```
.font{
```

```
font-style:italic;
```

```
font-variant:small-caps;
```

```
font-weight:bold;
```

```
font-size:12px;
```

```
line-height:1.5em;
```

```
font-family:arial,verdana;
```

```
}
```

上面的样式简写为:

```
.font{font:italic small-caps bold 12px/1.5em arial,verdana;}
```

三、font的简写注意事项

1、简写时, font-size和line-height只能通过斜杠/组成一个值, 不能分开写。

2、顺序不能改变.这种简写方法只有在同时指定font-size和font-family属性时才起作用。而且, 如果你没有设定font-weight, font-style, 以及 font-varient , 他们会使用缺省值

## 78.常用布局属性有哪些? 有什么特点?

margin , padding , position , float

<http://www.codesec.net/view/404576.html>



## 79.填空题

一个div，css设置如下：{width:200px;margin:200px 20px;padding:50px 60px 70px;border:100px solid red;overflow:hidden;}在IE6怪癖解析下，这个div的实际宽是\_360px\_？在正常解析下，这个div的实际宽度是\_560px\_？

## 80.web网页中常见的图片格式有哪些？分别有什么特点？

常用的图片格式有JPG、GIF、PNG。

1、jpg: jpg全名是JPEG。JPEG图片以 24 位颜色存储单个位图。JPEG 是与平台无关的格式，支持最高级别的压缩，不过，这种压缩是有损耗的。渐进式 JPEG 文件支持交错。

2、gif:GIF分为静态GIF和动画GIF两种，扩展名为.gif，是一种压缩位图格式，支持透明背景图像，适用于多种操作系统，“体型”很小，网上很多小动画都是GIF格式。其实GIF是将多幅图像保存为一个图像文件，从而形成动画，最常见的就是通过一帧帧的动画串联起来的搞笑gif图，所以归根到底GIF仍然是图片文件格式。但GIF只能显示256色。和jpg格式一样，这是一种在网络上非常流行的图形文件格式。

3、png: PNG，图像文件存储格式，其设计目的是试图替代GIF和TIFF文件格式，同时增加一些GIF文件格式所不具备的特性。PNG的名称来源于“可移植网络图形格式(Portable Network Graphic Format, PNG)”，也有一个非官方解释“PNG's Not GIF”，是一种位图文件(bitmap file)存储格式，读作“ping”。PNG用来存储灰度图像时，灰度图像的深度可多到16位，存储彩色图像时，彩色图像的深度可多到48位，并且还可存储多到16位的α通道数据。PNG使用从LZ77派生的无损数据压缩算法，一般应用于JAVA程序、网页或S60程序中，原因是它压缩比高，生成文件体积小。

## 81.在HTML中，SEO常见的白帽优化技巧有哪些？

<http://seo.pingce.cc/seoweb/hmseo/20130626204.html>

## 82.块属性标签与行属性标签的区别？哪些标签是块属性的，哪些是行的？

在标准文档流里面，块级元素具有以下特点：

- ①总是在新行上开始，占据一整行；
- ②高度，行高以及外边距和内边距都可控制；
- ③宽如果不设置，默认是父级的宽度，与内容无关；
- ④它可以容纳内联元素和其他块元素。

行内元素的特点：

- ①和其他元素都在一行上；
- ②高，行高及外边距和内边距部分可改变；
- ③宽度只与内容有关；
- ④行内元素只能容纳文本或者其他行内元素。

不可以设置宽高，其宽度随着内容增加，高度随字体大小而改变，内联元素可以设置外边界，但是外边界不对上下起作用，只能对左右起作用，也可以设置内边界，但是内边界在ie6中不对上下起作用，只能对左右起作用

<http://www.cnblogs.com/imihiroblog/archive/2012/08/28/2660370.html>

### 83.IE6和IE7有什么不同?

答:

- I. 打开IE7, 第一感觉就是简洁干净了许多。主要是去除了工具菜单, 按钮图标变得更加圆滑。
- II. 只放一些常用的工具按钮。
- III. 新的收藏中心、RSS feeds和历史, 更加整洁、方便管理和使用。
- IV. 现在流行的任何一款浏览器都会有TAB (标签) 功能。IE当然也不能落伍, TAB可以方便的切换窗口, 而不像以前那样在任务栏上来回寻找窗口。
- V. 新建标签放在了网页标签旁边, 在节省空间上IE7做的淋漓尽致。
- VI. 快速标签, 可以在打开多个窗口时查看每个窗口的缩略图。这个功能不如Opera, Opera可以在鼠标放到TAB上时自动显示一个缩略图。
- VII. 搜索工具栏, 随着Google 百度的出现, 人们查东西的频率越来越高了, 整合到浏览器里更是方便。当然微软也不会为Google做广告, 默认的搜索引擎是MSN Search, 可以手动设置到Google.
- VIII. 时间就是金钱, 用户浏览网页只希望在第一时间看到所需的信息, 而不用浪费时间去检查大量的网站以及Blog的更新, 新闻聚合器做到了这一点。
- IX. IE7加入了自动检测到网站中的RSS Feeds功能。
- X. Feeds的收藏夹, 有趣的是里面还有个Internet Explorer Team Blog
- XI. 安全性和隐私得到了提高

### 84. 有三个兄弟div, 设制样式 {float:left}如何使第二个div不浮动 (不能用CSS3选择器)

```
<div></div>
<div style="float:none"></div>
<div></div>
```

### 85. 写出两个div并排显示的css样式, 要求左侧的div宽度200px; 字体加粗; 行高30px; 与右侧div的外边距10px; 右侧的div宽度自适应, 边框1px灰色;

答:

```
<!DOCTYPE html>
<html>
  <head>
    <style type="text/css">
      * {
        padding: 0;
        margin: 0;
      }
      .out {
        position: fixed;
        right: 0;
        top: 0;
        left: 0;
      }
      #left {
        width: 200px;
        height: 200px;
        font-weight: bold;
```



```

        line-height: 30px;
        background-color: red;
    }
    #right {
        margin-left: 10px;
        height: 198px;
        border: 1px solid gray;
        background: blue;
        position: absolute;
        left: 200px;
        top: 0;
        right: 0;
    }
</style>
</head>
<body>
    <div class="out">
        <div id="left"></div>
        <div id="right"></div>
    </div>
</body>
</html>

```

## 86. css样式引用和js引入的位置问题（为什么会放在头部或者底部）。

答: 在开发网页时，尽量将css放置在页面的header里，js不管是外部引入的或嵌入的js片段都应放在页面的尾部。

因为页面在加载时，css加载是可以并发请求的（同页面中的图片，ie6除外），而js加载时需要等待一个js文件加载完成后才加载其他资源，为了页面的快速呈现，放置在尾部效果较好。

同时由于嵌入的js片段会阻塞整个页面的呈现（外部引入js文件阻塞其他资源下载，也会阻塞该js引入位置以下的页面的内容呈现，所以要放置页面的最下面）

## 87. 下列成立的是 (ABCD)

A. null==undefined    B. NaN!=NaN    C. false==0    D. "5"==5    E. undefined==0

## 88. 在下面的XHTML标签中，正确地标记折行的是 (A)

A. <br />    B. <break>    C. <br>    D. </br>

## 89. 下面不支持overflow-y属性的浏览器是 (A)

A. IE7    B. Firefox3.1    C. Safari3.1    D. Opera 9.5

## 90. 读下面一段代码，相邻div的垂直外边距是多少？

```

<!--xhtml1.0-->
<html>
<head>
<style type="text/css">
div{height: 50px; margin: 50px; background-color: #999;}
</style>
</head>

```

```

<body>
  <div></div>
  <div></div>
  <div></div>
</body>
</html>

```

答: 50px

91. 读下面一段代码，div#a和div#b哪个叠放层次更高，a覆盖b还是b覆盖a?

```

<!--xhtml1.0-->
<html>
<head>
<style type="text/css">
body{margin: 0;padding: 100px;background-color: #000;}
div .elem{width: 100px;height: 120px;margin-top: -40px;
background-color:#FCE5A6;border: 1px solid #FFF;
position: absolute;
}
div .box{margin-top: -50px;padding: 10px 20px;
width: 100px;height: 100px;
background-color: #F97C01;border: 1px solid #FFF;
position: relative;
}
</style>
</head>
<body>
<div class="box" style="z-index: 2;">
<div class="elem" id="a" style="z-index: 3;">a</div>
  </div>
  <div class="box" style="margin-left: 60px; z-index: 1;" >
    <div class="elem" id="b" style="z-index: 4;">b</div>
  </div>
</body>
</html>

```

答: a覆盖b

标签语义化的好处:

92. 标签语义化和CSS命名语义化有什么好处?

- 1.去掉样式或者样式丢失时页面结构依然清晰分明
- 2.移动设备能够更加完美的展示你的网页（移动设备对css的支持较弱）
- 3.阅读器会根据标签的语义自动解析，呈现更容易阅读的内容形式（无障碍阅读）
- 4.搜索引擎会根据标签的语义确定上下文和权重问题
- 5.便于后期的开发以及维护，团队合作效率提高

93. 某种效果，有两种实现方案都可以实现，方案一：<img>标签；方案二：背景图，两个方案如何取舍？衡量的标准是什么？

答: 如下场景使用img标签比较合适:

- 1、如果图像是等内容的一部分或图表或人(真正的人,而不是股票图人)，使用Img标签加上alt属性。
- 2、如果你想打印页面并且你想要的图像包括默认情况下使用IMG。

- 3、使用IMG(alt文本)图像有一个重要的语义时,比如一个警告图标。这将确保图像的意义可以很好的和user-agents沟通,包括屏幕阅读器。
- 4、如果你依赖于浏览器缩放图像比例并且可以呈现不错的效果时使用IMG。
- 5、如果配合 z - index 伸展背景图像来填补它的整个窗口时使用IMG。
- 6、使用img代替有背景图像可以显著提高性能的动画背景。
- 7、IMG会首先加载因为src在html文件本身中而在有背景图像源是样式表中引入的图像,加载样式表加载后,延迟加载的网页。

如下场景使用background-image属性比较合适:

- 1、如果图像不是内容的一部分时使用background-image。
- 2、当图像代替文本使用时使用background-image。
- 3、如果你想打印页面并且你不想要的图像包括默认情况下使用background-image。
- 4、如果需要缩短下载时间通过CSS sprites 时使用background-image。
- 5、如果你只需要展示图像的一部分通过CSS sprites, 时使用background-image。
- 6、如果你需要为不同的屏幕分辨率展示不同的图像使用 media查询时使用background-image。

## 94. 应用DIV+CSS写出一个固定宽度的三列布局, 能够使内容根据屏幕大小自适应居中。

答案: <!DOCTYPE html>

```
<html>
  <head>
    <style type="text/css">
      * {
        padding: 0;
        margin: 0;
      }
      .out {
        width: 75%;
        margin: 0 auto;
        overflow: hidden;
      }
      .out div {
        width: 33.33%;
        float: left;
      }
    </style>
  </head>
  <body>
    <div class="out">
      <div class="left"></div>
      <div class="middle"></div>
      <div class="right"></div>
    </div>
  </body>
</html>
```

95. CSS布局中，拥有布局(haslayout)有什么作用？怎样触发元素拥有布局？块级格式化范围(Block Formatting Context简称BFC)有什么作用？怎样触发元素BFC特性？

答: 参考[http://blog.sina.com.cn/s/blog\\_51048da701018o29.html](http://blog.sina.com.cn/s/blog_51048da701018o29.html)

96. 你做的页面在哪些浏览器测试过？这些浏览器的内核是什么？经常遇到的浏览器的兼容性有哪些？怎么会出现？解决方法是什么？（至少说3个）

答: IE内核浏览器：360，傲游，搜狗，世界之窗，腾讯TT

非IE内核浏览器：firefox opera safari chrome

1.就是ie6双倍边距的问题，在使用了float的情况下，不管是向左还是向右都会出现双倍，最简单的解决方法就是用display:inline;加到css里面去。

2.文字本身的大小不兼容。同样是font-size:14px的宋体文字，在不同浏览器下占的空间是不一样的，ie下实际占高16px，下留白3px，ff下实际占高17px，上留白1px，下留白3px，opera下就更不一样了。解决方案：给文字设定 line-height 。确保所有文字都有默认的 line-height 值。这点很重要，在高度上我们不能容忍1px 的差异。

3. ff下容器高度限定，即容器定义了height之后，容器边框的外形就确定了，不会被内容撑大，而ie下是会被内容撑大，高度限定失效。所以不要轻易给容器定义height。

4. 还讨论内容撑破容器问题，横向上的。如果float 容器未定义宽度，ff下内容会尽可能撑开容器宽度，ie下则会优先考虑内容折行。故，内容可能撑破的浮动容器需要定义width。

5. 浮动的清除，ff下不清除浮动是不行的。

6. mirrormargin bug，当外层元素内有float元素时，外层元素如定义margin-top:14px，将自动生成margin-bottom:14px。padding也会出现类似问题，都是ie6下的特产，该类bug 出现的情况较为复杂，远不只这一种出现条件，还没系统整理。解决方案：外层元素设定border 或 设定float。

7. 吞吃现象，限于篇幅，我就不展开了。还是ie6，上下两个div，上面的div设置背景，却发现下面没有设置背景的div 也有了背景，这就是吞吃现象。对应上面的背景吞吃现象，还有滚动下边框缺失的现象。解决方案：使用zoom:1。这个zoom好象是专门为解决ie6 bug而生的。

97. 什么是CSS浮动问题？如何清除浮动？写出所有你知道的方法

clear:both;

1.使用空标签清除浮动。这种方法是在所有浮动标签后面添加一个空标签定义css clear:both.弊端就是增加了无意义标签。

2.使用overflow。给包含浮动元素的父标签添加css属性overflow:auto;zoom:1;zoom:1用于兼容IE6。

3.使用after伪对象清除浮动。该方法只适用于非IE浏览器。具体写法可参照以下示例。使用中需注意以下几点。一、该方法中必须为需要清除浮动元素的伪对象中设置height:0，否则该元素会比实际高出若干像素；二、content属性是必须的，但其值可以为空，content属性的值设为"."，空亦是可以的。

4.浮动外部元素

此三种方法各有利弊，使用时应择优选择，比较之下第二种方法更为可取。

98. 触发haslayout的方法

答:大部分的 IE 显示错误，都可以通过激发元素的 haslayout 属性来修正。可以通过设置 css 尺寸属性(width/height)等来激发元素的 haslayout，使其“拥有布局”。可以通过设置 css 尺寸属性(width/height)等来激发元素的 haslayout，使其“拥有布局”。如下所示，通过设置以下 css 属性即可。

\* display: inline-block

\* height: (任何值除了auto) 通常用 \_height:1%;解决IE6的问题，height:1%不会改变实际高度

\* float: (left 或 right)

\* position: absolute

\* width: (任何值除了auto)

\* writing-mode: tb-rl

\* zoom: (除 normal 外任意值)

## 99. ie浏览器Hack的写法

- 1、IE6以及IE6以下版本浏览器  
\* html .demo {color: green;}
- 2、仅仅IE7浏览器  
\*:first-child+html .demo {color: green;}
- 3、除IE6之外的所有浏览器 (IE7-9, Firefox,Safari,Opera)  
html>body .demo {color: green;}
- 4、IE8-9,Firefox,Safari,Opear  
html>/\*\*/body .demo {color: green;}
- 5、IE9+  
:root .demo {color: red;}

## 100. 清除浮动的方法

clear:both;

1.使用空标签清除浮动。这种方法是在所有浮动标签后面添加一个空标签定义css clear:both.弊端就是增加了无意义标签。

2.使用overflow。给包含浮动元素的父标签添加css属性overflow:auto;zoom:1;zoom:1用于兼容IE6。

3.使用after伪对象清除浮动。该方法只适用于非IE浏览器。具体写法可参照以下示例。使用中需注意以下几点。一、该方法中必须为需要清除浮动元素的伪对象中设置height:0，否则该元素会比实际高出若干像素；二、content属性是必须的，但其值可以为空，content属性的值设为".",空亦是可以的。

4.浮动外部元素

此三种方法各有利弊，使用时应择优选择，比较之下第二种方法更为可取。

## 101. 用css分别实现某个DIV元素上下居中和左右居中。

方法一:

```
#one {  
    width: 100px;  
    height: 100px;  
    border: 4px solid red;  
    position: absolute;  
    left: 50%;  
    top: 50%;  
    transform: translate(-50%, -50%);  
}
```

方法二:

```
#one {  
    width: 100px;  
    height: 100px;  
    border: 4px solid red;  
    position: absolute;  
    left: 0;  
    right: 0;  
    bottom: 0;  
    top: 0;  
    margin: auto;  
}
```

方法三:

```
#one {
```

```
width: 100px;
height: 100px;
border: 4px solid red;
position: absolute;
left: 50%;
right: 50%;
margin-left: -50px;
margin-top: -50px;
}
```

## 102. 表单中有一些checkbox该如何提交值，name一样吗

答: name值可以写成一样的,但是要注意

比如这样写,

```
<form action="test.php" method="get">
<input type="checkbox" name="name1" value="1" />A
<input type="checkbox" name="name1" value="2" />B
  <input type="checkbox" name="name1" value="3" />C
  <input type="submit" value="OK"/>
</form>
```

如果你选择了多个选项,最终服务器只能接收到最后一个值  
需要这样写,

```
<form action="test.php" method="get">
  <input type="checkbox" name="name1[]" value="1" />A
  <input type="checkbox" name="name1[]" value="2" />B
  <input type="checkbox" name="name1[]" value="3" />C
  <input type="submit" value="OK"/>
</form>
```

如果你选择了多个选项,最终服务器接收到的是一个数组

## 103. 你能描述一下渐进增强和优雅降级之间的不同吗?

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。

## 104. 你如何对网站的文件和资源进行优化?

答: 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(多个域名来提供缓存)其他。

## 105. 如果你接到了一个使用Tab来缩进代码的项目,但是你喜欢空格,你会怎么做?

- 答:1、建议项目使用EditorConfig之类的规范  
2、为了保持风格的一致性,保持原有  
3、直接使用vim里的retab命令将所有的tab转换成空格

## 106. 你都使用那些工作来测试代码的性能?

例如JSPerf (<http://jsperf.com/>)

例如Dromaeo (<http://dromaeo.com/>)

## 107. 'data-' 属性的作用是什么?

data-\* 属性用于存储页面或应用程序的私有自定义数据。data-\* 属性赋予我们在所有 HTML 元素上嵌入自定义 data 属性的能力。存储的（自定义）数据能够被页面的 JavaScript 中利用，以创建更好的用户体验（不进行 Ajax 调用或服务器端数据库查询）。

data-\* 属性包括两部分：

属性名不应该包含任何大写字母，并且在前缀 "data-" 之后必须有至少一个字符

属性值可以是任意字符串

在HTML5中我们可以使用以 data- 为前缀来设置我们需要的自定义属性，来进行一些数据的存放。当然高级浏览器下可通过脚本进行定义和数据存取

```
<div data-author="david" data-time="2011-06-20" data-comment-num="10">...</div>
```

## 108. 描述css reset的作用和用途。

答: 在HTML标签在浏览器里有默认的样式, 不同浏览器的默认样式之间也会有差别, 往往会给我们带来麻烦, 影响开发效率。所以解决的方法就是一开始就将浏览器的默认样式全部去掉, 更准确说就是通过重新定义标签样式让他们统一. 这就是CSS reset。

## 109. 解释css sprites以及如何使用。

答: CSSSprites在国内很多人叫css精灵, 是一种网页图片应用处理方式。它允许你将一个页面涉及到的所有零星图片都包含到一张大图中去, 这样一来, 当访问该页面时, 载入的图片就不会像以前那样一幅一幅地慢慢显示出来了

把网页中一些背景图片整合到一张图片文件中, 再利用CSS的“background-image”, “background-repeat”, “background-position”的组合进行背景定位, background-position可以用数字精确的定位出背景图片的位置。

## 110. 你使用CSS预处理器吗? 如果有请简单介绍一下

答: CSS 预处理器是一种语言用来为 CSS 增加一些编程的特性, 无需考虑浏览器的兼容性问题, 例如你可以在 CSS 中使用变量、简单的程序逻辑、函数等等在编程语言中的一些基本技巧, 可以让你的 CSS 更见简洁, 适应性更强, 代码更直观等诸多好处。最为普遍的三款 CSS 预处理器框架, 分别是 Sass、Less CSS、Stylus。

## 111. 谈谈你对WEB标准以及W3C的理解与认识。

答: (1) web标准规范要求, 书写标签必须闭合、标签小写、不乱嵌套, 可提高搜索机器人对网页内容的搜索几率。--- SEO

(2) 建议使用外链css和js脚本, 从而达到结构与行为、结构与表现的分离, 提高页面的渲染速度, 能更快地显示页面的内容。

(3) 样式与标签的分离, 更合理的语义化标签, 使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, 从而降低维护成本、改版更方便

(4) 不需要变动页面内容, 便可提供打印版本而不需要复制内容, 提高网站易用性

遵循w3c制定的web标准, 能够使用户浏览者更方便的阅读, 使网页开发者之间更好的交流。

## 112. 下面哪几项是HTML4.01中不建议使用的 (AC)

A. `<font></font>`

B. `<em></em>`

C. `<u></u>`

D. `<i></i>`

### 113.你能描述一下你制作一个网页的工作流程吗?

第一步：项目需求评审（明确需求）

第二步：等待ui设计师提供psd（项目需求技术调研）

第三步：等到ui设计师设计稿通过以后，给前端制作静态html+js（自己在浏览器或者移动设置上浏览，查看是否有兼容问题）

以上为基本制作流程

以下几步为项目对接-一直到发布到线上的过程

第四步：前后台数据对接完成-测试没有问题以后，提交测试环境（任务给测试人员）

第五步：修改测试返回的问题（直到测试通过为准）

第六步：测试发送测试报告以后，提交预生产环境，先试着跑1到2周，观察情况（测试回归任务检验开发任务是否正常）- - - 如果不正常返回给响应的开发人员进行修复。

第七步：正式上线（一般在凌晨更新上线）- - 主要原因：减少更新项目对用户影响。

第八步：测试人员回归- - 前端同时参与回归- - 后端也参与回归（如果有问题就悲催了，解决bug，没有问题项目运行正常- - 回家睡觉）

### 114.你能描述一下渐进增强和优雅降级之间的不同吗?

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。

参考文档：<https://segmentfault.com/a/1190000000800711#articleHeader30>

### 115.请解释一下什么是语义化的HTML。

语义化的HTML使用每个html标签都特定的用途，例如p标签放大段文字，h1~h6常用于标题，strong加粗强掉…。。

### 116.你如何对网站的文件和资源进行优化?

文件合并

文件最小化/文件压缩

使用 CDN 托管

缓存的使用（多个域名来提供缓存）

### 117.为什么利用多个域名来存储网站资源会更有效?

答：使用多个域名的原因主要有以下方面：

1.将请求图片的url使用单独的域名,因为图片资源对网站来说是重要的一部，同时带宽要求高（主要做cdn加速）

2.静态资源单独的域名（如：css文件，js文件）



3.其中某个域名崩溃用户也能通过其他郁闷访问网站

## 118.请说出三种减低页面加载时间的方法

- 1、压缩css、js文件
- 2、合并js、css文件，减少http请求
- 3、外部js、css文件放在最底下
- 4、减少dom操作，尽可能用变量替代不必要的dom操作

## 119.什么是FOUC? 你如何来避免FOUC?

答：如果使用import方法对CSS进行导入,会导致某些页面在Windows 下的Internet Explorer出现一些奇怪的现象:以无样式显示页面内容的瞬间闪烁.这种现象称之为文档样式短暂失效(Flash of Unstyled Content),简称为FOUC.

原因大致为:

- 1, 使用import方法导入样式表。
- 2, 将样式表放在页面底部
- 3, 有几个样式表, 放在html结构的不同位置。其实原理很清楚: 当样式表晚于 结构性html 加载, 当加载到此样式表时, 页面将停止之前的渲染。此样式表被下载和解析后, 将重新渲染页面, 也就出现了短暂的花屏现象。解决方法: 使用LINK标签将样式表放在文档HEAD中更多

## 120.文档类型的作用是什么? 你知道多少种文档类型声明方式?

影响浏览器对html代码的编译渲染  
dtd三种方式

### 1过渡的(Transitional)

是要求非常宽松的DTD, 一般的情况下网页中会使用这个, 它允许你继续使用HTML4.01的标识(但是要符合xhtml 的写法)。

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd" >
```

### 2严格的(Strict)

要求严格的DTD, 对页面的标签使用非常严格, 页面里几乎不准使用 inline-style 的 CSS 样式。

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd" >
```

### 3框架的(Frameset)

专门针对框架页面设计使用的DTD, 以前Table 满街走的年代很流行这种DTD, 如果你的页面中包含有框架, 需要采用这种DTD。

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd" >
```

4适用于手机浏览器的DTD（已经不用了）

## 121.你常使用的HTML标签有哪些？

h1-h6 header footer nav article span p a div img ul li ol li dl dt dd br strong等

注意：不要只关注问题的字面意思，要深刻的去理解考官为什么要出这个题的目的

参考思路：

- 1.对标签的理解
- 2.对浏览器兼容问题的理解
- 3.对h5新增标签的理解，以及h5新增的属性有哪些？？？
- 4.对新技术的关注以及是否自己验证过

## 122.说下你对标签语义化的理解。

答：“语义化”是指在人为更少的干预下让机器更好的理解和收集信息网页信息；标签的语义化可以让人们更直观认识标签和属性的作用。特别是HTML5规范以后，新出现的许多标签如header、footer、article等，使html正在朝着更加健壮的语义化的HTML结构发展。

## 123.你知道的html 行级元素和块级元素标签分别有哪些？

块级元素会从新的一行出现，行内元素在一行并排显示；行级元素不能设置width与height.

块级元素有：h1~h6、ul、li、ol、dl、dt、dd、p、div、form、等

行内元素有：a、span strong 等

特殊元素具有行和块的属性：img、input、table

## 124.常用浏览器的内核有哪些？

答：一、Trident内核代表产品Internet Explorer，又称其为IE内核。Trident（又称为MSHTML），是微软开发的一种排版引擎。使用Trident渲染引擎的浏览器包括：IE、傲游、世界之窗浏览器、Avant、腾讯TT、Netscape 8、NetCaptor、Sleipnir、GOSURF、GreenBrowser和KKman等。

二、Gecko内核代表作品Mozilla FirefoxGecko是一套开放源代码的、以C++编写的网页排版引擎。Gecko是最流行的排版引擎之一，仅次于Trident。使用它的最著名浏览器有Firefox、Netscape6至9。

三、WebKit内核代表作品Safari、Chromewebkit 是一个开源项目，包含了来自KDE项目和苹果公司的一些组件，主要用于Mac OS系统，它的特点在于源码结构清晰、渲染速度极快。缺点是对网页代码的兼容性不高，导致一些编写不标准的网页无法正常显示。主要代表作品有Safari和Google的浏览器Chrome。

四、Presto内核代表作品OperaPresto是由Opera Software开发的浏览器排版引擎，供Opera 7.0及以上使用。它取代了旧版Opera 4至6版本使用的Elektra排版引擎，包括加入动态功能，例如网页或其部分可随着DOM及Script语法的事件而重新排版。

## 主流浏览器所使用的内核分类

Trident内核：IE,MaxThon,TT,The World,360,搜狗浏览器等

Gecko内核: Netscape6及以上版本, FF,MozillaSuite/SeaMonkey等

Presto内核: Opera7及以上

Webkit内核: Safari,Chrome等

典型的双核浏览器包括:

搜狗2.0: Trident内核和WebKit内核

傲游3.0Beta: Trident和WebKit内核

QQ浏览器5: Trident内核和WebKit内核

使用双核浏览器时, 可以自动/手动切换内核来浏览网页。

关于 不要再叫我IE11地址: <http://liulanmi.com/news/5546.html>

## 125.为什么要清除浮动, 有哪些方式?

这里其实是面试的陷阱 (归根结底需要多浏览器兼容性的问题了如指掌)

一个块级元素的高度如果没有设置height, 那么其高度就是由里面的子元素来撑开的, 如果子元素使用浮动, 脱离了标准的文档流, 那么父元素的高度会将其忽略, 如果不清除浮动, 父元素会出现高度不够, 那样如果设置border或者background都得不到正确的解析。

清除浮动的方法: 区分以下哪些是最好的方案 (1~5不可取)

- 1)在浮动元素下加子元素<div class="clear"></div>然后设置样式.clear{ height:0px;font-size:0;clear:both};
- 2)在浮动元素下面 (与浮动元素同级) 加<br clear="all"/>
- 3)给浮动元素的父级也加浮动(这种情况当父级margin:0 auto时不居中)
- 4)给父级加display:inline-block;(同方法2,不居中,只有IE6,7居中)
- 5)给浮动元素父级加overflow:auto;

6) 经典用法

```
.clear,.clear-all{clear:both;font-size:0;height:0;line-height:0;overflow:hidden}
.fix:after,.clearfix:after{clear:both;content: " ";display:block;font-size:0;height:0;visibility:hidden}
.fix,.clearfix{*zoom:1}
```

## 126.SGML (标准通用标记语言) 和HTML (超文本标记语言), XML (可扩展标记语言) 和HTML的之间有什么关系?

SGML (标准通用标记语言) 是一个标准, 告诉我们怎么去指定文档标记。他是只描述文档标记应该是怎么样的元语言, HTML是被用SGML描述的标记语言。

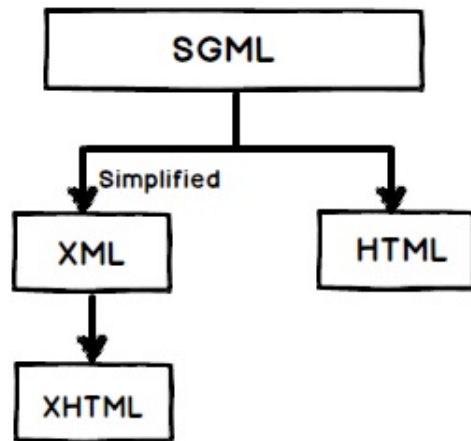
因此利用SGML创建了HTML参照和必须共同遵守的DTD, 你会经常在HTML页面的头部发现“DOCTYPE”属性, 用来定义用于解析目标DTD

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
```

现在解析SGML是一件痛苦的事情，所以创建了XML使事情更好。XML使用了SGML，例如：在SGML中你必须使用起始和结束标签，但是在XML你可以有自动关闭的结束标签。

XHTML创建于XML，他被使用在HTML4.0中。你可以参考下面代码片段中展示的XML DTD

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```



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

答：（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。

### 128.行内元素有哪些？块级元素有哪些？空(void)元素有那些？

答：（1）块级元素。默认display属性是block。div ul ol li dl dt dd h1 h2 h3 h4...p。

（2）行内元素。默认display属性是inline。a b span img input select strong。

（3）空元素。功能性元素，标签可以不闭合。<br> <hr> <img> <input> <link> <meta>

### 129.介绍一下CSS的盒子模型？

答：（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)。

### 130.link和@import的区别是？

答：（1）<link>属于XHTML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XHTML标签，无兼容问题;
- (4) <link>方式的样式的权重 高于@import的权重。

### 131.如何居中div? 如何居中一个浮动元素?

```
<div class="div1">div1</div>
```

```
<div class="div2">div2</div>
```

```
.div1{width:200px;height:200px;background:red;margin:0px auto;}
```

```
.div2{width:200px;height:200px;background:red;float:left;margin-left:-100px;position:absolute;left:50%;top:220px;}
```

### 132.浏览器的内核分别是什么?

答：IE浏览器的内核Trident、Mozilla的Gecko、Chrome的Blink（WebKit的分支）、Opera内核原为Presto，现为Blink。

### 133.HTML语义化的理解?

答：（1）用正确的标签做正确的事情！

- (2) html语义化就是让页面的内容结构化，便于对浏览器、搜索引擎解析；
- (3) 在没有样式CCS情况下也以一种文档格式显示，并且是容易阅读的。
- (4) 搜索引擎的爬虫依赖于标记来确定上下文和各个关键字的权重，利于 SEO。
- (5) 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。

### 134.iframe有那些缺点?

答：（1）iframe会阻塞主页面的Onload事件；

- (2) iframe和主页面共享连接池，而浏览器对相同域的连接有限制，所以会影响页面的并行加载。

解决方法：使用iframe之前需要考虑这两个缺点。如果需要使用iframe，最好是通过javascript动态给iframe添加src属性值，这样可以可以绕开以上两个问题。

### 135.Label的作用是什么? 是怎么用的?（加 for）

答：主要是和input配合使用，没有单独存在的意义

```
<label for="user">用户名</label>
```

```
<input type="text" name="username" value="" id="user" />
```

点击label的时候自动会聚焦input输入框

### 136.列出display的值，说明他们的作用。position的值， relative和absolute定位原点是?

答案：1.display:none\block\inline-block

display:none;使添加none属性元素隐藏

回答display: block; 先讲出哪些是行标签或块标签（都具备什么样的特性）

行元素（span strong a …）（或内联元素、行标签指的都是同一类型的元素，只是叫法不一样）

行元素的特性：能在一行显示排成一行

块元素(p div h1-h6)

块元素的特性：独自占一行只有内容放不下的时候，内容才会换行

display:block;将行元素具备了块元素的特性

display:inline-block;

将行元素——具备了块元素的特性的同时也具备行元素本身的特性，元素能设置宽高，当然特殊的标签本身可以设置宽高（例如：input img table）

块元素——具备了行元素的特性可以在一行显示

position:relative\absolute

默认没有设置left right top bottom 偏移值时，为浏览器的最左上角开始为原点

absolute:

- 1、父级元素没有relative时,当前元素的定位父级为document文档
- 2、父级有relative时，父级为当前元素的定位起点

## 137.用纯 CSS 创建一个三角形

```
.triangle{  
  
width: 0;height: 0;  
  
border-left: 60px solid transparent;  
  
border-right: 60px solid transparent;  
  
border-top: 120px solid red;  
  
}
```

## 138.一个满屏”品”字布局 如何设计?

```
<div class="header"></div>  
<div class="main">  
  <div class="left"></div>  
  <div class="right"></div>  
</div>  
  
.header{height:200px;background:#e33;width:100%;}  
  
.main{width:100%;height:200px;overflow:hidden;}  
  
.main .left,.main .right{width:50%;height:200px;float:left;background:#a23;}  
  
.main .right{background:#e11;}
```

### 139.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?

li使用display:inline-block以后浏览器会解析空白的字符。

解决办法: 使用float

### 140.经常遇到的浏览器的兼容性有哪些? 原因, 解决方法是什么, 常用hack的技巧?

(1)png24位的图片在IE6浏览器上出现背景, 解决方案是做成PNG8。

(2)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。

(3)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。

(4)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。

(5)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。

(6)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。

### 141.为什么要初始化CSS样式。(面试官主要考的是对标签的默认样式了解)

1.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初始化往会出现浏览器之间的页面显示差异。

2.初始化CSS样式主要是提高编码质量, 如果不初始化整个页面做完很糟糕, 重复的css样式很多。

去掉标签的默认样式如: margin, padding, 其它浏览器默认解析字体大小字体设置

```
html,body, dl, dd, h1, h2, h3, h4, h5, h6, p, form{margin:0;}
```

```
ul,ol,li{margin:0;padding:0}
```

详细参考地址: <http://www.zhangxinxu.com/wordpress/2010/04/css-reset%E7%9A%84%E9%87%8D%E6%96%B0%E5%AE%A1%E8%A7%86-%E9%81%BF%E5%85%8D%E6%A0%B7%E5%BC%8F%E9%87%8D%E7%BD%AE/>

### 142.absolute 的 containing block(容器块)计算方式跟正常流有什么不同?

(1)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;

(2)否则,则由这个祖先元素的 padding box 构成。

如果都找不到, 则为 initial containing block。

//

补充:

(1)static(默认的)/relative: 简单说就是它的父元素的内容框 (即去掉padding的部分)

(2)absolute: 向上找最近的定位为absolute/relative的元素

(3)fixed: 它的containing block一律为根元素(html/body), 根元素也是initial containing block



### 143.对BFC规范的理解?

答: 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定位,以及与其他元素的关系和相互作用。

参考地址: <https://segmentfault.com/a/1190000003827247>

### 144.css定义的权重? (考的是css书写优先级)

答: 权重的规则: 标签的权重为1, class的权重为10, id的权重为100, style 1000,

```
/*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{}
```

**important**资料扩展知识参考地址: [http://baike.baidu.com/link?](http://baike.baidu.com/link?url=0QBBKMjn_uHaipI5N18gg_J5eEKAUhFQeuKhjRScsW0JW56EZjbsuw-aOZOilum_hFon-47_G-PJ33Ey_UU03q)

[url=0QBBKMjn\\_uHaipI5N18gg\\_J5eEKAUhFQeuKhjRScsW0JW56EZjbsuw-aOZOilum\\_hFon-47\\_G-PJ33Ey\\_UU03q](http://www.w3cplus.com/css/the-important-css-declaration-how-and-when-to-use-it.html)

<http://www.w3cplus.com/css/the-important-css-declaration-how-and-when-to-use-it.html>

### 145.解释下浮动和它的工作原理? 清除浮动的技巧

答: 浮动元素脱离文档流, 不占据空间。浮动元素碰到包含它的边框或者浮动元素的边框停留。

1.使用空标签清除浮动。

这种方法是在所有浮动标签后面添加一个空标签 定义css clear:both. 弊端就是增加了无意义标签。

2.使用overflow。

给包含浮动元素的父标签添加css属性 overflow:auto; zoom:1; zoom:1用于兼容IE6。

3.使用after伪对象清除浮动。

该方法只适用于非IE浏览器。具体写法可参照以下示例。使用中需注意以下几点。一、该方法中必须为需要清除浮动元素的伪对象中设置 height:0, 否则该元素会比实际高出若干像素;

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 (内联元素) 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，对于问题1，添加如下样式，给父元素添加clearfix样式：

清除浮动的几种方法：

1，额外标签法，<div style="clear:both;"></div>（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。）

2，使用after伪类

## 146.CSS优化、提高性能的方法有哪些？

CSS 优化主要是四个方面：

- 1.加载性能
- 2.选择器性能
- 3.渲染性能
- 4.可维护性、健壮性

尽量减少HTTP请求个数——须权衡

一般就是把多个css用merge（后台）合并

把CSS放到顶部

避免使用CSS表达式

将CSS和JS放到外部文件中

精简CSS（主要是把空格和注释去掉，一般和第一点一起进行）

剔除重复的JS和CSS（一般大的工程很容易出现这样的问题）

高效的CSS书写规则（这点比较重要……简单讲，就是少些废话）

参考地址：<http://www.zhihu.com/question/19886806>

## 147.浏览器是怎样解析CSS选择器的？

答：浏览器解析CSS选择器的方式是从右到左

参考地址：<http://www.zhihu.com/question/20185756?rf=21310443>

## 148.在网页中的应该使用奇数还是偶数的字体？为什么呢？

答：都是使用偶数，如果在网站里面使用的奇数字体，文字在显示的时候不是很美观

## 149.margin和padding分别适合什么场景使用？

答：margin是设置元素外边距的属性，W3C官方解释为：围绕在元素边框的空白区域是外边距。设置外边距会在元素外创建额外的“空白”。设置外边距的最简单的方法就是使用margin属性，这个属性接受任何长度单位、百分数值甚至负值。

Padding是设置元素内边距的属性，W3C的官方解释为：元素的内边距在边框和内容区之间。控制该区域最简单的属性是 padding 属性。CSS padding 属性定义元素边框与元素内容之间的空白区域。

上下两个元素之间的间距使用margin合适。

padding的使用是针对当前设置元素的内边距

主要考的是对margin和padding的理解：在项目里面能用padding搞定的坚决不使用margin。

参考答案地址：[http://blog.sina.com.cn/s/blog\\_87f8e6160100xlt5.html](http://blog.sina.com.cn/s/blog_87f8e6160100xlt5.html)

## 150.如何修改chrome记住密码后自动填充表单的黄色背景？

通过审查元素可以看到这是由于chrome会默认给自动填充的input表单加上input:-webkit-autofill私有属性

参考地址：[工作中经常会碰到类似的问题](#)

<https://segmentfault.com/q/1010000000671971>

<http://www.poluoluo.com/jzxy/201510/430850.html>

扩展资料：

`-webkit-appearance: none;`

<http://www.css88.com/archives/5402>

## 151.设置元素浮动后，该元素的display值是多少?(自动变成display:block)

行元素设置浮动：具备了（display:block属性能设置宽高）

块元素设置浮动：具备了（display:inline属性能在一行显示）

## 152.怎么让Chrome支持小于12px 的文字？

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。

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;}`这个方式现在用的比较少了，需要清晰知道每个元素的默认样式来统一。

```
Html,body,p,h1,h2{margin:0}
```

## 153.让页面里的字体变清晰，变细用CSS怎么做？

`(-webkit-font-smoothing: antialiased;)`

## 154.font-style属性 可以让它赋值为“oblique” oblique是什么意思？

浏览器会显示一个倾斜的字体样式。（很少使用oblique）

在项目里面平常使用的italic比较多一些

### 155.position:fixed;在android下无效怎么处理?

使用js处理判断当前设备是否是android，如果是则添加监听当前页面滚动情况，设置position:absolute;不断的改变top值

### 156.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 $1/60 * 1000\text{ms} = 16.7\text{ms}$

### 157.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

### 158.浅谈一下怎么处理浏览器的兼容性问题。

答：合理化HTML结构，简洁化css属性：html结构合理规范了，很大程度上避免了兼容问题，css要尽量写的简单有效，特效用JavaScript来做，很多时候别用css伪类实现，针对比较特殊的用css hack 处理。

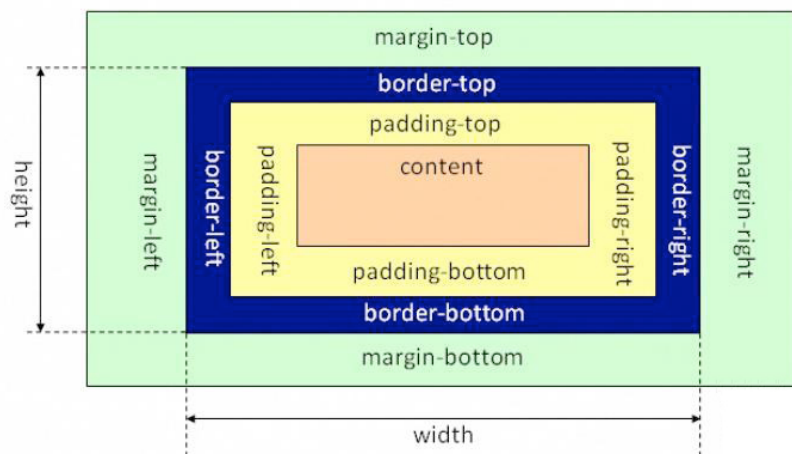
### 159.浏览器标准模式和怪异模式之间的区别是什么?

答：主要原因：doctype声明问题

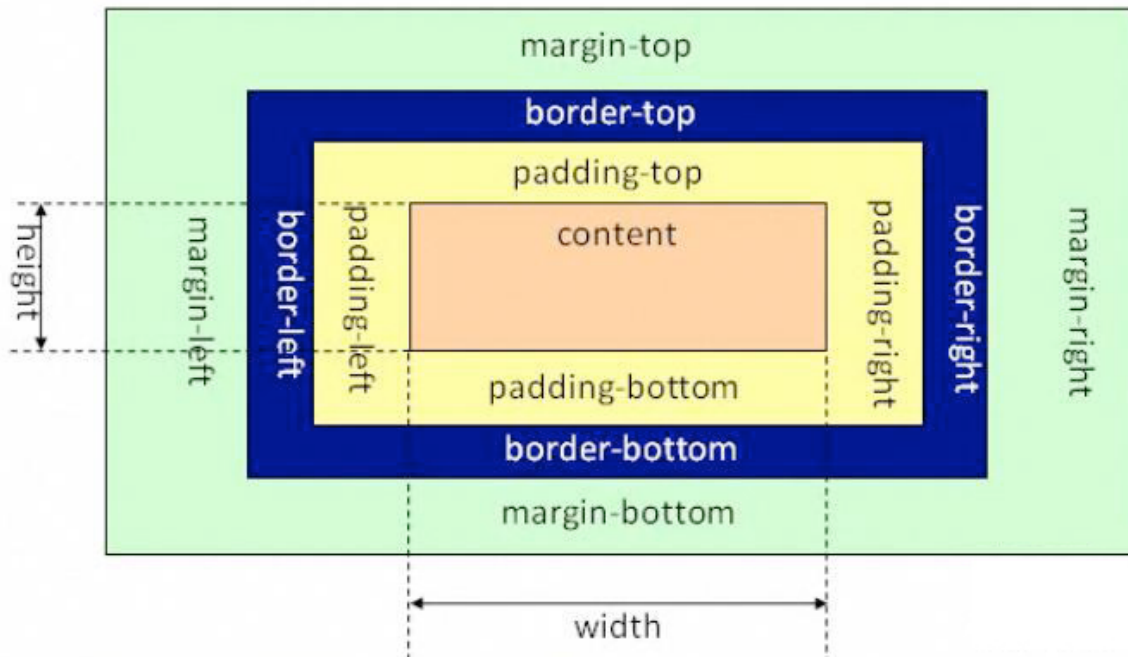
区别：盒模型解释不同

IE：IE盒子模型的width和height是包括了内容、内边距和边框长度的。IE6之前，以及IE6~8在怪异模式下时，就是采用这种盒子模型。

#### ■ IE盒子模型



## ■ 标准盒子模型



例如IE的怪异模式中，元素的width包含了padding和border，而标准模式中padding和border 并不属于宽度的一部分。

### 160.css hack知道哪些?

答：由于不同的浏览器，甚至同一浏览器的不同版本对CSS的解析认识不一样，导致生成的页面效果不一致，写出针对不同浏览器CSS code就称为CSS hack。

`_color:red;` 只有ie6生效 (IE6能识别下划线"\_"和星号" \* ")

`*color:blue;` IE7能识别星号" \* "

`*+html .class {}` IE7能识别

在标准模式中：

`"-"` 减号是IE6专有的hack

`"\9"` IE6/IE7/IE8/IE9/IE10都生效

`"\0"` IE8/IE9/IE10都生效，是IE8/9/10的hack

`"\9\0"` 只对IE9/IE10生效，是IE9/10的hack

### 161.你对css盒模型的理解。

答：盒模型有两种：标准盒模型（w3c标准）和IE盒模型。

标准盒模型中width和height指的是内容区域的宽度和高度，增加内边距、边框和外边距不会影响内容区域的尺寸，但是会增加元素框的总尺寸；

但在IE6浏览器的width不是内容的宽度，而是内容、内边距和边框的宽度总和。

## 162.data-属性的作用是什么？

答：data-为前端开发者提供自定义的属性

```
<div data-author="david" data-time="2011-06-20" data-comment-num="10">...</div>
```

## 163.你能解释一下CSS的盒子模型么？

CSS和模型是围绕在HTML元素周围的定义Border(边界)，padding(内边距)和margin(外边距)的矩形空间

Border(边界)：定义了元素包含的最大区域，我们能够使边界可见，不可见，定义高度和宽度等；

Padding(内边距)：定义了边界和内部元素的间距

Margin：定义了边界和任何相邻元素的间距

注意面试官想要的答案不止这些（需要回答什么是怪异盒模型盒标准盒模型）如果能回答就扯一下产生怪异模型盒的原因和结果。面试题1和2有答案

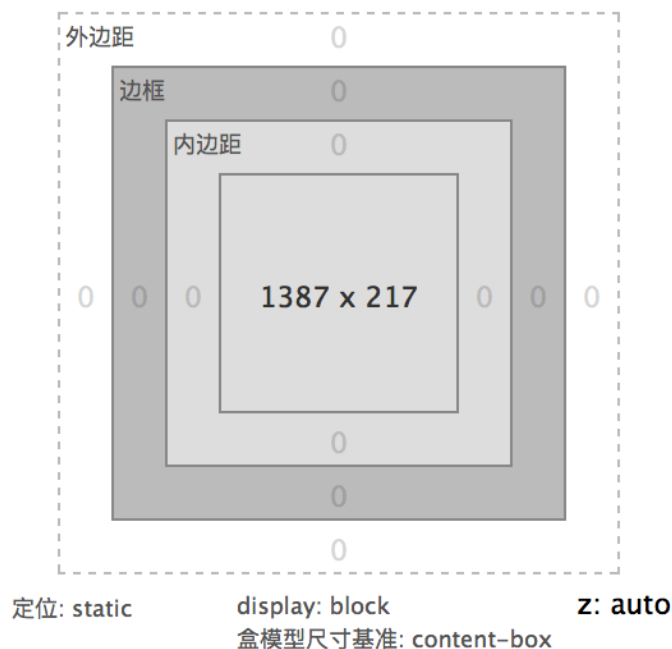
原因：主要原因是doctype文档声明问题

结果：

标准盒模型中width和height指的是内容区域的宽度和高度，增加内边距、边框和外边距不会影响内容区域的尺寸，但是会增加元素框的总尺寸；

但在IE6浏览器的width不是内容的宽度，而是内容、内边距和边框的宽度总和。

分别能画出盒模型的样式



## 164.CSS3有哪些新特性（包含哪些模块）？

圆角 (border-radius:8px) ,

阴影 (box-shadow:10px) ,

文字特效 (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

背景透明 rgba

更多选择器参考地址: <http://www.runoob.com/cssref/css-selectors.html>

其它模块: (Other modules)

media queries:感觉叫媒体选择比较合适,可以为网页中不同的对象设置不同的浏览设备。比如可以为某一块分别设置屏幕浏览样式和手机浏览样式,以前则只能设置整个网页。

multi-column layout:多列布局,让文字以多列显示,包括column-width、column-count、column-gap三个值

column-width:指定每列宽度

column-count:指定列数

column-gap:指定每列之间的间距

column-rule-color:控制列间的颜色

column-rule-style:控制列间的样式

column-rule-width:控制列间的宽度

column-space-distribution:平均分配列间距

## 165.请解释一下CSS3的Flexbox (弹性盒布局模型),已经适用场景?

△主要应用场景为: 移动端布局

flexbox的布局是一个用于页面布局的全新CSS3模块功能。它可以把列表放在同一个方向(从左到右或从上到下排列),并且让这些列表能延伸到占用可用的空间。较为复杂的布局可以通过嵌套一个伸缩容器(flex container)来辅助实现。

Flexbox可以简单快速的创建一个具有弹性功能的布局,当在一个小屏幕上显示的时候,Flexbox可以让元素在容器(伸缩容器)中进行自由扩展和收缩,从而容易调整整个布局。它的目的是使用常见的布局模式,比如说三列布局,可以非常简单的实现。

## 166.用过媒体查询,针对移动端的布局吗?

在移动端布局的时候,经常会使用到媒体查询针对不到显示比例的手机做样式处理。(注意了这里可以开启装逼模式:把现有的主流手机显示宽高比例以及型号倒背如流)

最终目的是一套代码在不通的移动设备上查看一样的效果

参考地址: <http://blog.csdn.net/iefreer/article/details/24212161>

公司现在用的统计平台:

<https://mtj.baidu.com/web/demo/terminal?appId=468475>

<http://tongji.baidu.com/data/mobile/brand>

之前google木有被屏蔽的时候都喜欢使用google的统计

## 167.使用 CSS 预处理器吗？喜欢那个？

答：常用的SASS、LESS（回答的时候要注意了，每个优缺点）

Sass (Syntactically Awesome Stylesheets)是一种动态样式语言，语法跟css一样(但多了些功能)，比css好写，而且更容易阅读。Sass语法类似与Haml，属于缩排语法（makeup），用意就是为了快速写Html和Css。

Less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)。

区别：

(1)Sass是基于Ruby的，是在服务端处理的，而Less是需要引入less.js来处理Less代码输出Css到浏览器，也可以在开发环节使用Less，然后编译成Css文件，直接放到项目中，也有Less.app、SimpleLess、CodeKit.app这样的工具，也有在线编译地址。

(2)变量符不一样，less是@，而Scss是\$，而且变量的作用域也不一样，后面会讲到。

(3)输出设置，Less没有输出设置，Sass提供4中输出选项：nested, compact, compressed 和 expanded。

(4)Sass支持条件语句，可以使用if{}else{} ,for{}循环等等。而Less不支持。

LESS咱们使用的HBuilder可以把less插件安装上就可以编写less了，注意需要建立一个css文件夹啊，必须把less文件放在css文件里面。

Less官网：<http://lesscss.cn/>

参考地址：<http://www.1024i.com/demo/less/>

Sass 官网：<http://www.sass-zh.com/>

参考地址：<http://caibaojian.com/css-processor.html>

## 168.元素竖向的百分比设定是相对于容器的宽度吗？

答：不是的，元素竖向的百分是相对于设定容器的高度的

## 169.全屏滚动的原理是什么？用到了CSS的那些属性？

答：1.transform

2.使用定位方式

简单的css3全屏滚动-左右方向

translateZ(0)可以触发GPU渲染，提高动画的性能。

1. transform:translateX的值（不考虑低版本浏览器的同时推荐使用该属性）

2. left的值；

经验分享：在项目里面要实现全屏滚动，可以做浏览器版本检查，高版本浏览器支持css3 transform的尽情的使用，不支持的使用定位方式

## 170.什么是响应式设计？为什么要做响应式设计？响应式设计的基本原理是什么？

答：页面的设计与开发应当根据用户行为以及设备环境(系统平台、屏幕尺寸、屏幕定向等)进行相应的响应和调整。具体的实践方式由多方面组成，包括弹性网格和布局、图片、CSS media query的使用等。无论用户正在使用笔记本还是iPad，我们的页面都应该能够自动切换分辨率、图片尺寸及相关脚本功能等，以适应不同设备；换句话说，页面应该有能力去自动响应用户的设备环境。

响应式网页设计就是一个网站能够兼容多个终端——而不是为每个终端做一个特定的版本。这样，我们就可以不必为不断到来的新设备做专门的版本设计和开发了。

响应式设计的基本原理是通过媒体查询检测不同的设备屏幕尺寸做处理。页面头部必须有meta声明viewport：

```
<meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=1, user-scalable=no">
```

## 171.::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。本质上并不支持伪元素的双冒号(::)写法，而是忽略掉了其中的一个冒号，仍以单冒号来解析，所以等同变相支持了E::before

答：在before元素前面，:after元素的后面插入内容（经典场景使用font-face小图标结合content: ""）

参考资料：

<http://www.css88.com/book/css/selectors/pseudo-element/before.htm>

<http://www.w3cplus.com/css3/learning-to-use-the-before-and-after-pseudo-elements-in-css.html>

## 第二部分：JS面试题

### 1. 获取一堆name名相同的div，点击弹出显示第几个div

（本题主要考察闭包的用法，是在面试中常被问的题目，不难但是比较容易上当的题目）

```
var boxes = document.getElementsByTagName("div");
for(var i = 0; i < boxes.length; i++) {
    boxes[i].index = i+1;
    boxes[i].onclick = function() {
        alert(this.index);
    }
}
```

答案二

```
var boxes = document.getElementsByTagName("div");
for(var i = 0; i < boxes.length; i++) {
    (function(i){
        boxes[i].onclick = function() {
            alert(i+1);
        }
    })(i);
}
```



## 2. abcdefgsdds 获取里面字面最多的字母（答案有很多）

```
var str = "abcdefgsdds";
var obj = {};
var arr = [];
var letter;
for(var i = 0,len = str.length;i<len;i++){
    letter = str[i];
    if(!obj[letter]){
        obj[letter] = 1;
    }else{
        obj[letter]++;
    }
}
var max_key,max_num=0;
for(key in obj){
    if(max_num < obj[key]){
        max_num = obj[key];
        max_key = key;
    }
}
document.write("字母: "+max_key+" 次数: "+max_num);
```

## 3. abcdefghi 获取ghi怎么实现

答: "abcdefghi".substring(6)

## 4. 'asdjhklzxc' 截取'asd'

答: "asdjhklzxc".substring(3,-1)

## 5. 翻转字符串

答: "asd".split("").reverse().join("")

## 6. 删除数组第一个元素

答: arr.shift()

## 7. [1,2,1,3,2,3,5,6,5] 去除重复元素

```
var arr = [1,2,1,3,2,3,5,6,5];
var newArr = [];
for(var i = 0; i < arr.length; i++) {
    if(newArr.indexOf(arr[i]) == -1) {
        newArr.push(arr[i]);
    }
}
```

```
}  
alert(newArr)
```

## 8. JS中有几种数据类型分别是哪几种?

答: Number,String,Boolean,Object,Null,Undefined

## 9. 说一下拖拽的原理? 怎么样实现多个DIV在拖拽时有相撞让它们改变属性与属性值和他们的的位置? (部分程度不太好的同学看到拖拽会懵, 项目期有这个情况)

mousedown,mouesrmove,mouseup事件

e.clientX,e.clientY 鼠标位置

碰撞检测

## 10. 用JS创建一个a标签, 在body中显示, 点击在新窗口打开

```
var alink = document.createElement("a");  
alink.innerHTML = "google";  
alink.href = "http://www.google.com";  
alink.target = "_blank";  
document.body.appendChild(alink);
```

## 11. 鼠标滚轮动画

```
function mousewheel(element,up,down) {  
    element.onmousewheel = fn;  
    if(window.addEventListener) {  
        element.addEventListener("DOMMouseScroll",fn,false)  
    }  
    function fn(e) {  
        var e = e || window.event;  
        console.log(e);  
        if(e.wheelDelta > 0 || e.detail < 0) {  
            up();  
        }else {  
            down();  
        }  
        e.cancelBubble = true;  
        e.stopPropagation();  
    }  
}  
mousewheel(document,function() {},function() {})//函数调用
```

## 12. 定位右下角广告

答: 使用css position:fixed

13. 鼠标经过A A变大 其他的位置不动

答: A绝对定位, 脱离文档流即可

14. JS中避免命名冲突的三个方法 (有些同学或许会想不到这些概念)

答: 命名空间, 闭包, 匿名函数

15. JS在HTML中的引入方式有哪些? 又有什么优缺点?

答: a外部引入 b内部嵌套 c行内代码 例: <button onclick="alert('hello')">按钮</button> 对比css引入

16. 写出5条标识符的命名规范 ()

答: 1.区分大小写;

2.第一个字符必须是一个字母、下划线 ( \_ ) 或一个美元符号 ( \$ )

3..不能含有空格

4.不能以关键字或保留字命名,不能以数字开头

17. 从JS的角度, 写出8条以上优化代码的方法?

答: 参考 [http://www.haorooms.com/post/web\\_xnyh\\_jscss](http://www.haorooms.com/post/web_xnyh_jscss)

18. 按优先级, 对这几种运算符进行排序: ! && \* = >

答: ! && \* > =

19. 在循环中, break和continue语句的功能有何不同

答: break 终止整个循环, continue终止当次循环

20. 解释事件冒泡、事件捕获的区别

答: 事件流解释即可

21. 说说什么叫做事件委托

答: 利用事件冒泡的原理, 子元素的事件会冒泡到父元素, 可以只给父元素添加事件, 通过事件目标判断元素。优点: 节省内存, 动态添加的子元素也包含事件

22. 列举不少于6条的IE与FF脚本兼容性问题, 需要写出命令 (也许同学们缺乏总结, 看着简单, 实际很难答全)

(1) window.event:

表示当前的事件对象, IE有这个对象, FF没有, FF通过给事件处理函数传递事件对象

(2) 获取事件源

IE用srcElement获取事件源, 而FF用target获取事件源

(3) 添加, 去除事件

IE: element.attachEvent("onclick", function) element.detachEvent("onclick", function)

FF: element.addEventListener("click", function, true) element.removeEventListener("click", function, true)

(4) 获取标签的自定义属性

IE: div1.value或div1["value"]

FF: 可用div1.getAttribute("value")

(5) document.getElementById()和document.all[name]

IE; document.getElementById()和document.all[name]均不能获取div元素

FF: 可以

(6) input.type的属性

IE: input.type只读

FF: input.type可读写

(7) innerText textContent outerHTML

IE: 支持innerText, outerHTML

FF: 支持textContent

(8) 是否可用id代替HTML元素

IE: 可以用id来代替HTML元素

FF: 不可以

## 23. 构造函数有何优缺点?

答: 构造函数可以创建多个对象,

缺点: 容易跟普通函数一样调用

## 24. 用自己的话描述一下“闭包”

答: 函数内部又定义了一个函数, 内部函数引用外部函数的变量, 就构成了闭包

## 25. 用自己的话描述一下对“异步”和“同步”的理解

同步: 按顺序执行, 同一时刻只能执行一个事件

异步: 不按顺序执行, 同一时刻可以执行多个任务

## 26. 写倒计时 距离2018年还有 多少天多少小时多少分钟多少秒

```
var time = document.getElementById("time");
function clock() {
    //设置目标日期
    var targetDate = new Date(2018,0,1);
    var currentDate = new Date();
    var remainTime = targetDate - currentDate;
    //天数
    remainDay = parseInt(remainTime/1000/60/60/24);
    //小时数
    remainHours = parseInt(remainTime/1000/60/60%24);
    //分钟
    remainMinutes = parseInt(remainTime/1000/60%60);
    //秒
```

```
remainSeconds = parseInt(remainTime/1000%60);
```

```
time.innerHTML = remainDay + "天"+remainHours+"小时"+remainMinutes+"分  
钟"+remainSeconds+"秒";  
}  
setInterval(clock,1000);
```

## 27. alert怎样换行

答：转义字符,换行符\n

```
alert("大家好\n大家好")
```

## 28. 左侧内容可上下滚动，右侧是滚动条，如何实现布局

答：固定高度，overflow-y:scroll

## 29. 向上滚动动画如何实现

```
document.body.scrollTop = "xxx";  
document.documentElement.scrollTop = "xxx";
```

## 30. form表单提交

答：默认提交，form.onSubmit = function(){}可在表单数据提交之前验证

## 31. JavaScript与ECMAScript的区别？

答：javascript是具体实现，ECMAScript是标准，大纲

## 32. ==与===的区别？

==等于，判断值是否相等，会有类型转换

===全等，判断类型跟值是否相等，类型不会转换

## 33. 什么是DOM？

答：文档对象模型

## 34. Undefined与null的区别？

答：Undefined无定义，null空

## 35. 有两个对象，var o = {num:1};var p = {num:2};要求书写一段代码，使得o+p===3为true。

```
Object.prototype.toString = function(){  
    return this.num;  
}
```

36. 如果给你这么一个页面：顶部有logo什么的，下来是导航条，中上部有图片切换，右边是登录注册，左边有简单的信息展示，中下部是一些新闻，底部是一般的那些联系我们呀什么的传统页尾。这样一个布局加上所有的js效果和新闻链接包括所有的兼容，你大概能多久完成（新鸿儒）；

37. 讲解一下this（作用域）

this的指向：普通函数内的this指向全局变量

构造函数内部this指向新创建出来的对象

对象方法内的this指向的是调用该方法的对象

call,apply,bind可以改变this的指向

38. 构造函数

答：new关键字后紧接的函数即为构造函数，约定首字母大写

39. 原型，原型链的理解

答：ECMA5之前没有类的概念，js的继承是基于原型链的

40. 说说前端优化从哪方面着手，具体说明有哪些？

html 语义化，符合标准，减少嵌套层数，加快浏览器DOM解析速度

css 合并，压缩 少写冗余css代码，使用精灵图 ...

js 减少DOM操作，使用变量缓存数据，少用全局变量，做动画的元素都绝对定位...

41. 代码说明call和apply方法的区别（咱们这方面讲解的少，这样的题有变式，需要举例讲解一下）

答：这两个都是函数的方法，可以改变this的指向，fn.call(obj,param1,param2,...)

call传入单个参数

fn.apply(obj,[param1,param2,...])

apply传入一个数组

42. function f(x){

    alert(x);(1)

        arguments[0]=null;

        alert(x);(2)

    }

    f(5);

    \_\_\_\_(2)\_\_\_\_

答案：5， null

43. var a=new Array(3);（实际上是闭包的用法）

a[0]=function(x){return x\*x;}

```
a[1]=20;  
a[2]=a[0](a[1]);
```

a[2]的结果\_\_\_400\_\_\_

44. `var fn = function(i){  
 return function(){  
 return i++;  
 }  
}(10);  
var a = fn();  
var b = fn();  
alert(a);(1)  
alert(b);(2)`

(1)\_10\_(2)\_11\_

45. 对于js语言，下列说法中（ C ）是正确的？

- A.一种标记语言规范的版本名
- B.前端编程中，新增特性的一种泛指和统称
- C.一种脚本语言
- D.可以实现跨平台开发的技术

46. 用两种方法，将第上题中的第一个p标签里的内容的颜色设置为红色

```
element.style.color = "red"  
element.className = "red".red{color:red}
```

47. 如何测算数组的长度？

答：array.length

48. 滚动类效果中，会涉及到的定位属性有哪些，同时写出每个命令的含义。

答：left,top,right,bottom,scrollLeft,scrollTop

49. 在放大镜效果中，获取鼠标位置的方法，同时说明鼠标位置是针对谁获取的？

答：e.clientX,e.clientY 相对于窗口坐标

50. 事件种类有哪些？ 分别举出不少于4个的例子？

答：鼠标事件，键盘事件，触屏事件

51. 访问到tween算法中的Bounce的easeInOut方法？

```
var startPos = 0;//确定起点位置  
var endPos = 500;//确定终点位置  
var everyPos = 20;//确定每步移动距离
```

```

var step = 0;//确定当前步数
var stepMax = 25;//确定总步数
var tween = {
  Bounce:{
    easeIn:xxxxx;
    easeOut:xxxxx;
    easeInOut:xxxx;
  }
}
tween.Bounce.easeInOut(step,startPos,(endPos-startPos),stepMax)

```

52. Date对象中getDay()和getDate()分别是什么含义？取值范围是多少？

getDay()获取星期几 0~6  
 getDate() 获取日期 1~31

53. 设置当前时间为2004年12月12日的代码书写方法？（分写合写两种方式）

```

new Date(2004,11,12)
var d = new Date(); d.setFullYear(2004),d.setMonth(11);d.setDate(12);

```

54. 写出不少于5种Math对象的方法，分别说明含义以及语法

Math.random()随机数0~1之间  
 Math.abs()绝对值  
 Math.floor() 向下取整  
 Math.ceil()向上取整  
 Math.min();Math.max().....

55. 使用任意一种方法随机生成一个从100-300的整数。

```

function rnd(min, max) {return parseInt(Math.random() * (max - min)) + min} //不包含300
rnd(100,300)

```

56. tab切换效果 代码填空

```

<style type="text/css">
    .show{display:block;}
    .select{background:#fcc;}
div {display:none}
</style>
<body>
    <h2>
        <span id='span1' class='select'>标题1</span>
        <span id='span2'>标题2</span>
        <span id='span3'>标题3</span>
    </h2>

```



```

<div id='con1' class='show'>内容1</div>
<div id='con2'>内容2</div>
<div id='con3'>内容3</div>
</body>
<script type="text/javascript">
var cons = document.getElementsByTagName('div');
var spans = document.getElementsByTagName('span')
for (var i=0; i< __spans.length__ ; i++) {
    __spans[i]__.onclick=function () {
        for (var j=0; j< __cons.length__ ; j++) {
            if ( __this__ == __spans[j]__ ) {
                cons[j].__className__ = __“show”__ ;
                spans[j].__className__ = __“select”__ ;
            }
            else{
                cons[j].__className__ = __“”__ ;
                spans[j].__className__ = __“”__ ;
            }
        }
    }
}
}
</script>

```

## 57. 用JS截取标签中的abc

```

<body>
<p id='con'>abcdef</p>
</body>
<script type="text/javascript">
var con = document.getElementById('con');
var string = con.__innerText__ ;
var result = string.__substring(0,3)__ ;
alert(result);
</script>

```

## 58. 用js如何获取一个上传文件的扩展名

```

<input type="file" id="file";
var file = document.getElementById("file")
alert(file.files[0].name)

```

## 59. 用js实现类、继承（继承构造函数内部的属性方法及原型的属性和方法）、私有变量

```

function Car() {

```

```

//构造函数
this.name = "benz"; //属性
var price = "1000000"; //私有变量，外部不能访问，只能通过下面的getPrice方法调用
this.getPrice = function() {
    return price;
}
}
Car.prototype.drive = function() {console.log("drive")} //给原型添加drive方法，创建出来的实例都有该方法
var c = new Car(); //创建实例对象

```

## 60. 写一个trim函数，兼容IE firefox chrome（正则）（怕部分学生想不到正则）

```

if(typeof String.prototype.trim !== "function") {
    String.prototype.trim = function() {
        return this.replace(/^\s* | \s*$ /g, "");
    }
}

```

直接调用即可，例：var str = " sdf ";str.trim();

如何提升网页加载速度

前端角度：合并压缩css,js减少http请求数以及大小，使用精灵图，减少dom操作...

## 61. 列举事件绑定的方法（原生+JQ addEventListener与attachEvent的区别）

addEventListener现代浏览器都支持，ie从ie9开始支持，可以绑定多个事件

attachEvent ie专有，可以绑定多个事件，但this指向全局对象

setInterval 与 setTimeout 的区别 (略)

split() join() 的区别

split()字符串方法，将字符串转换成数组

join()数组方法，将数组合并成字符串

## 62. addEventListener和通过on这种方式绑定的事件（例：ele.onclick = function(){}）

有什么区别

addEventListener可以绑定多个事件

on这种方式只能绑定一次，之后绑定的事件会覆盖前面绑定的，实质是对象属性重新赋值

DOM事件流

事件捕获，事件目标，事件冒泡

## 63. ajax请求时，如何解析json数据

三种方式 1.JSON.parse(string)

new Function("return " + string)()

eval("(" + string + ")")

## 64. 写一个获取非行间样式的函数

//获取非行间样式，兼容现代浏览器跟低版本ie

```
function getStyle(element,property) {  
    if(window.getComputedStyle) {  
        return window.getComputedStyle(element,null)[property];  
    }else {  
        return element.currentStyle[property];  
    }  
}
```

## 65. 闭包是什么，有什么特性，对页面有什么影响

答：闭包简单理解就是：函数内部有嵌套定义里一个函数，内部函数调用外部函数的局部变量的时候就构成闭包。能保存局部变量，函数执行完不销毁。消耗内存资源

## 66. 解释jsonp的原理，以及为什么不是真正的ajax（jsonp有些同学可能不知道）

原理：通过script标签的跨域功能，而非XMLHttpRequest

window load 和document ready的区别

document ready 实质是DOMContentLoaded事件

所有的节点渲染完毕，就执行

load事件要等到所有的资源加载完毕，如图片，视频，js脚本等

## 67. 如何使用原生 Javascript 代码深度克隆一个对象（注意区分对象类型）//对象的深复制

```
function deepCode(oldObj,newObj) {  
    var newObj = newObj || {};//如果存在newObj 就newObj,否则用空对象  
    for(var i in oldObj) {  
        if(oldObj.hasOwnProperty(i)) {  
            if(typeof oldObj[i] === "object") { //判断属性是否引用类型的  
                var c = (oldObj[i].constructor === Array)?[]:{};//如果判断值是否是数  
组  
                newObj[i] = deepCode(oldObj[i],c);  
                //递归  
            }else {  
                newObj[i] = oldObj[i];  
            }  
        }  
    }  
    return newObj;  
}
```

## 68. 截取字符串abcdefg的efg

答: alert('abcdefg'.substring(4));

## 69. 判断一个字符串中出现次数最多的字符，统计这个次数

```
var str = 'asdfsaaasasasasaa';
var json = {};

for (var i = 0; i < str.length; i++) {
    if(!json[str.charAt(i)]){
        json[str.charAt(i)] = 1;
    }else{
        json[str.charAt(i)]++;
    }
};

var iMax = 0;
var iIndex = "";
for(var i in json){
    if(json[i]>iMax){
        iMax = json[i];
        iIndex = i;
    }
}

alert("出现次数最多的是:"+iIndex+'出现'+iMax+'次');
```

参考答案: a出现9次

## 70. IE与FF脚本兼容性问题

obj.addEventListener(sEv, fn, false);

obj.attachEvent('on'+sEv,fn);

obj.getCurrentStyle[attr]

getComputedStyle(obj,false)[attr]

XMLHttpRequest

ActiveXObject("Mircorsoft.XMLHttp")

FF本地能设置读取cookie 其他不行

event ev

事件源

srcElement || target

toElement || relatedTarget

obj.setCapture();只有ie认

obj.releaseCapture();

## 71. 规避javascript多人开发函数重名问题

命名空间

封闭空间

js模块化mvc（数据层、表现层、控制层）

seajs（如果了解的呢，可以说）

变量转换成对象的属性

对象化

## 72. javascript面向对象中继承实现

```
function Person(name){
    this.name = name;
}

Person.prototype.showName = function(){
    alert(this.name);
}

function Worker(name, job){
    Person.apply(this,arguments)
    this.job = job;
}

for(var i in Person.prototype){
    Worker.prototype = Person.prototype;
}

new Worker('easy', 'coders').showName();
```

## 73. outerHTML和innerHTML区别

使用innerHTML的时候将找着元素的内容（不包含元素本身）

使用outerHTML的时候将找着元素的内容（包含元素本身）

注意:

W3C 只支持innerHTML. 其他都是微软的规定.(outerHTML,outerText,innerText只有微软的IE 好使, 其他浏览器不好用(firefox, mozilla等),必须用其他方法实现)

innerHTML & outerHTML: 获取/设置的内容是包含标签的

两者已经被HTML5纳入规范。

innerText & outerText: 获取/设置的内容是文本内容

innerHTML在获取的时候:

IE和Opera会将获取到的标签全部转换为大写形式。而苹果、谷歌、火狐会将内容按照原来的格式返回HTML，包括空格和缩进。（这也就意味着返回后的不同内容的DOM树结构截然不同）

outerHTML: 新建的DOM子树会替代掉“调用元素”

IE4+、Safari4+、Chrome、Opera8+支持, FF7-不支持

#### 74. 编写一个方法 求一个字符串的字节长度 (假设一个中文占两个字节)

```
var str = '22两是';  
alert(getStrlen(str))
```

```
function getStrlen(str){  
    var json = {len:0};  
    var re = /[u4e00-\u9fa5]/;  
    for (var i = 0; i < str.length; i++) {  
        if(re.test(str.charAt(i))){  
            json['len']++;  
        }  
    };  
    return json['len']+str.length;  
}
```

#### 75. 写出3个使用this的典型应用

答: 事件: 如onclick this->发生事件的对象

构造函数 this->new 出来的object

call/apply 改变this

#### 76. 如何深度克隆

```
var arr = [1,2,43];  
var json = {a:6,b:4,c:[1,2,3]};  
var str = 'sdfsdf';  
var json2 = clone(json);  
alert(json['c'])  
function clone(obj){  
    var oNew = new obj.constructor(obj.valueOf());  
    if(obj.constructor == Object){  
        for(var i in obj){  
            oNew[i] = obj[i];  
            if(typeof(oNew[i]) == 'object'){  
                clone(oNew[i]);  
            }  
        }  
    }  
    return oNew;  
}
```

77. JavaScript中如何检测一个变量是一个String类型？请写出函数实现

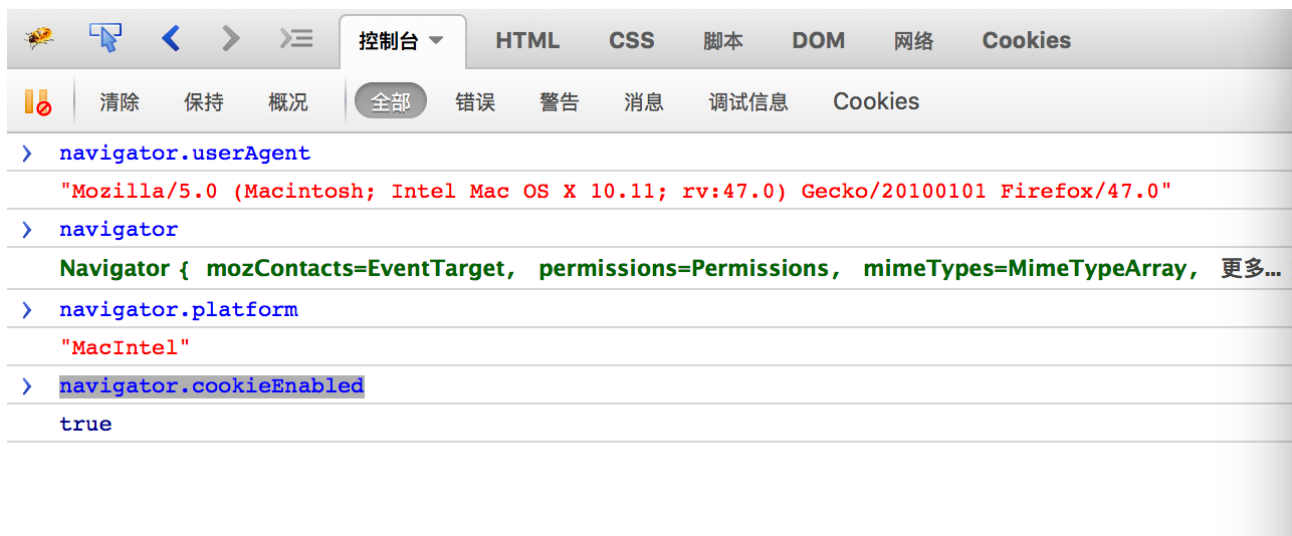
```
typeof(obj) == 'string'  
obj.constructor == String;
```

78. 闭包

答：子函数能被外部调用到，则该作用连上的所有变量都会被保存下来。

79. 请解释什么是Javascript的模块模式，并举出实用实例。

js模块化mvc（数据层、表现层、控制层）  
seajs



80. 你如何组织自己的代码？是使用模块模式，还是使用经典继承的方法

对内：模块模式  
对外：继承

81. 你如何优化自己的代码？

代码重用  
避免全局变量（命名空间，封闭空间，模块化mvc..）  
拆分函数避免函数过于臃肿  
加注释

82. 你能解释一下JavaScript中的继承是如何工作的吗？

子构造函数中执行父构造函数，并用call\apply改变this  
克隆父构造函数原型上的方法

83. 请说明navigator 常用属性？

navigator.userAgent 检查用户代理（设备信息、平台）

mac下的FF浏览器返回的信息：

```
"Mozilla/5.0 (Macintosh; Intel Mac OS X 10.11; rv:47.0) Gecko/20100101 Firefox/47.0"
```

检查cooke是否可用：

navigator.cookieEnabled 返回true为可用，false不可用

更多信息使用浏览器调试工具查看，进行总结

## 84. 浏览器调试快捷键是什么？

F12 或者 Ctrl+Shift+J两种方法。

ctrl+shift+m 调用FF浏览器的响应视图

ctrl+shift+i 调用chrome浏览器的调试工具

## 85. 数组怎么去重？

先将原数组进行排序；检查原数组中的第i个元素 与 结果数组中的最后一个元素是否相同，因为已经排序，所以重复元素会在相邻位置；如果不相同，则将该元素存入结果数组中。

利用对象的原型方法，代码为：

```
Array.prototype.unique2 = function(){
    this.sort(); //先排序
    var res = [this[0]];
    for(var i = 1; i < this.length; i++){
        if(this[i] !== res[res.length - 1]){
            res.push(this[i]);
        }
    }
    return res;
}
var arr = [1, 'a', 'a', 'b', 'd', 'e', 'e', 1, 0]
alert(arr.unique2());
//结果： 0,1,a,b,d,e
```

## 86. 数组排序。

数组排序方法还有冒泡排序：

注意：主要说清楚冒泡排序的执行顺序以及原理，编写代码其实是次要的

```
var arr = [3,7,9,1,5,10];
var temp = 0;
for (var i = 0; i < arr.length; i++){
    for (var j = 0; j < arr.length - i; j++){
        if (arr[j] > arr[j + 1]){
            temp = arr[j + 1];
            arr[j + 1] = arr[j];
```



```
        arr[j] = temp;
    }
}
}
alert(arr);
```

## 87. 常用变量声明规则?

匈牙利命名：开头字母用变量类型的缩写，其余部分用变量的英文或英文的缩写，要求单词第一个字母大写

小驼峰式：(little camel-case)：第一个单词首字母小写，后面其他单词首字母大写

大驼峰式：(big camel-case)：每个单词的第一个字母都大写；

## 88. call和apply的区别（动手验证）

call和apply，这两个方法基本上是一个意思，区别在于 call 的第二个参数可以是任意类型，而apply的第二个参数必须是数组，也可以是arguments。

## 89. 什么是闭包，在js中的作用是什么?

答：闭包就是讲函数内部生成的变量保存到内存中，进行下次调用；也可以说函数外不可以调用函数内部的变量；

当函数内部返回一个函数时，闭包搭建了方法内部与方法外部的桥梁，使得外部也可以任意的获取到方法内部的资源，但是闭包会造成变量在内存中持久占用，因此会有一定的性能问题，不要轻易使用，即便使用，也要在恰当的时机进行释放。

## 90. 请描述一下cookies、sessionStorage与localStorage的区别。(百度搜索)

答：共同点：都是保存在浏览器端，且同源的。

区别：cookie数据始终在同源的http请求中携带（即使不需要），即cookie在浏览器和服务器间来回传递。而sessionStorage和localStorage不会自动把数据发给服务器，仅在本地保存。cookie数据还有路径（path）的概念，可以限制cookie只属于某个路径下。存储大小限制也不同，cookie数据不能超过4k，同时因为每次http请求都会携带cookie，所以cookie只适合保存很小的数据，如会话标识。

sessionStorage和localStorage 虽然也有存储大小的限制，但比cookie大得多，可以达到5M或更大。数据有效期不同，sessionStorage：仅在当前浏览器窗口关闭前有效，自然也就不可能持久保持；localStorage：始终有效，窗口或浏览器关闭也一直保存，因此用作持久数据；cookie只在设置的cookie过期时间之前一直有效，即使窗口或浏览器关闭。作用域不同，sessionStorage不在不同的浏览器窗口中共享，即使是同一个页面；localStorage 在所有同源窗口中都是共享的；cookie也是在所有同源窗口中都是共享的。Web Storage 支持事件通知机制，可以将数据更新的通知发送给监听者。Web Storage 的 api 接口使用更方便。

注意：（这里面考虑到安全因素，必要突出安全性方面的信息，比如：敏感信息如用户信息—银行账户、其它登录账户或密码）

## 91. JavaScript的数据类型都有什么?

基本数据类型:

字符串 String 数字 Number 布尔 Boolean

复合数据类型:

数组 Array 对象 Object

特殊数据类型:

Null 空对象 Undefined 未定义

## 92. 本地存储和事务存储之间的区别是什么?

答: 本地存储数据持续永久, 但是会话在浏览器打开时有效知道浏览器关闭时会话变量重置。

## 93. 什么是事务存储? 我们如何创建一个事务存储?

会话存储和本地存储类似, 但是数据在会话中有效, 简而言之数据在你关闭浏览器的时候就被删除了。

为了创建一个会话存储你需要使用“sessionStorage.variableName.”在以下的代码我们创建了一个名为“clickcount”的变量;

如果你刷新浏览器则数目增加, 但是如果你关闭浏览器, “clickcount”变量又会从0开始。

实例:

```
if(sessionStorage.clickcount){
    sessionStorage.clickcount=Number(sessionStorage.clickcount)+1;
}else{
    sessionStorage.clickcount = 0;
}
```

## 94. 本地存储的生命周期是什么?

答: 本地存储没有生命周期, 它将保留知道用户从浏览器清除或者使用Javascript代码移除。

## 95. HTML5中的本地存储概念是什么?

本地存储是把网页上的某些数据按照键值对的方式存放在浏览器里面 一个域最多可以存5M数据 不区分离线还是在线 这些被存的数据都会一直在浏览器里面 除非手动清除数据。

(**小心有陷阱**) 面试官会问做过项目里面哪些地方用过?

## 96. 我们如何从本地存储中添加和移除数据?

数据添加到本地存储采用键值对, 以下示例显示了城市数据”北京”添加了键”test”

```
localStorage.setItem("test","北京");
```

从本地存储中检索数据我们可以提供键名并使用”getItem”方法

```
localStorage.getItem("test");
```

如果你想存储json格式，你可以使用“JSON.stringify”方法，如下所示：

通过浏览器可以查看

```
var country = {};  
country.name = "中国";  
country.code = "I001";  
//localStorage.setItem("I001", country);  
//var country1 = localStorage.getItem("I001");  
localStorage.setItem("I001",JSON.stringify(country));  
localStorage.getItem("I001");
```

移除本地存储：localStorage.removeItem(指定键)

```
localStorage.removeItem("I001");
```

## 97. 如何实现浏览器内多个标签页之间的通信? (阿里)

调用localStorage、cookies等本地存储方式

## 98. 介绍JavaScript的基本数据类型。

基本数据类型5种：字符串（string）、数字（number）、布尔（Boolean）、空（Null）、未定义（Undefined）

复杂数据类型：数组、对象（跟本题没有多大关系，如果面试官换一种问法，就需要注意了，问JavaScript数据类型都有哪些呢？）

## 99. 说几条写JavaScript的基本规范？

- (1) 不要在同一行声明多个变量
- (2) 如果你不知道数组的长度，使用 push
- (3) 请使用 ===/!== 来比较 true/false 或者数值
- (4) 对字符串使用单引号 "(因为大多时候我们的字符串。特别html会出现)"
- (5) 使用对象字面量替代 new Array 这种形式
- (6) 绝对不要在一个非函数块里声明一个函数，把那个函数赋给一个变量。浏览器允许你这么做，但是它们解析不同
- (7) 不要使用全局函数
- (8) 总是使用 var 来声明变量，如果不这么做将导致产生全局变量，我们要避免污染全局命名空间
- (9) Switch 语句必须带有 default 分支
- (10) 使用 /\*...\*/ 进行多行注释，包括描述，指定类型以及参数值和返回值
- (11) 函数不应该有时候有返回值，有时候没有返回值
- (12) 语句结束一定要加分号
- (13) for 循环必须使用大括号
- (14) if 语句必须使用大括号
- (15) for-in 循环中的变量应该使用 var 关键字明确限定作用域，从而避免作用域污染

- (16) 避免单个字符名，让你的变量名有描述意义
- (17) 当命名对象、函数和实例时使用驼峰命名规则
- (18) 给对象原型分配方法，而不是用一个新的对象覆盖原型，覆盖原型会使继承出现问题
- (19) 当给事件附加数据时，传入一个哈希而不是原始值，这可以让后面的贡献者加入更多数据到事件数据里，而不用找出并更新那个事件的事件处理器

参考文档：<http://www.jb51.net/article/24579.htm>

## 100. JavaScript有几种类型值？（堆：原始值和 栈：引用值），你能画一下他们的内存图吗？

栈：原始数据类型（Undefined, Null, Boolean, Number, String）

堆：引用数据类型（对象、数组、函数）

两种类型的区别：存储位置不同

//

原始数据类型直接存储在栈(stack)中的简单数据段，占据空间小、大小固定，属于被频繁使用数据，所以放入栈中存储；

引用数据类型存储在堆(heap)中的对象,占据空间大、大小不固定,如果存储在栈中，将会影响程序运行的性能；引用数据类型在栈中存储了指针，该指针指向堆中该实体的起始地址。当解释器寻找引用值时，会首先检索其在栈中的地址，取得地址后从堆中获得实体。

## 101. Javascript如何实现继承？

(1)构造继承

(2)原型继承

(3)实例继承

(4)拷贝继承

//

原型prototype机制或apply和call方法去实现较简单，建议使用构造函数与原型混合方式。

```
function Parent() {  
    this.name = 'song';  
}  
  
function Child() {  
    this.age = 28;  
}  
  
Child.prototype = new Parent(); //通过原型,继承了Parent  
  
//  
var demo = new Child();  
alert(demo.age);  
alert(demo.name); //得到被继承的属性
```

## 102. 如何创建一个对象? (画出此对象的内存图)

javascript创建对象简单的说,无非就是使用内置对象或各种自定义对象,当然还可以用JSON;但写法有很多种,也能混合使用。

//

(1)对象字面量的方式

```
person={firstname:"Mark",lastname:"Yun",age:25,eyecolor:"black"};
```

(2)用function来模拟无参的构造函数

```
function Person() {}
```

var person = new Person(); //定义一个function, 如果使用new"实例化",该function可以看作是一个Class

```
person.name = "Xiaosong";
```

```
person.age = "23";
```

```
person.work = function() {
```

```
    alert("Hello " + person.name);
```

```
}
```

```
person.work();
```

(3)用function来模拟带参构造函数来实现 (用this关键字定义构造的上下文属性)

```
function Person(name,age,hobby) {
```

```
    this.name = name; //this作用域: 当前对象
```

```
    this.age = age;
```

```
    this.work = work;
```

```
    this.info = function() {
```

```
        alert("我叫" + this.name + ", 今年" + this.age + "岁, 是个" + this.work);
```

```
    }
```

```
}
```

```
var Xiaosong = new Person("WooKong",23,"程序猿"); //实例化、创建对象
```

```
Xiaosong.info(); //调用info()方法
```

(4)用工厂方式来创建 (内置对象)

```
var jsCreator = new Object();
```

```
jsCreator.name = "Brendan Eich"; //JavaScript的发明者
```

```
jsCreator.work = "JavaScript";
```

```
jsCreator.info = function() {
```

```
    alert("我是"+this.work+"的发明者"+this.name);
```

```
}
```

```
jsCreator.info();
```

(5)用原型方式来创建

```
function Standard() {}
```

```
Standard.prototype.name = "ECMAScript";
```

```
Standard.prototype.event = function() {
```

```
    alert(this.name+"是脚本语言标准规范");
```

```
}
```

```
var jiaoben = new Standard();
```

```
jiaoben.event();
```

(6)用混合方式来创建

```
function iPhone(name,event) {
    this.name = name;
    this.event = event;
}
iPhone.prototype.sell = function() {
    alert("我是"+this.name+"，我是iPhone5s的"+this.event+"~ haha!");
}
var SE = new iPhone("iPhone SE","官方翻新机");
SE.sell();
```

### 103. 谈谈This对象的理解。

this总是指向函数的直接调用者（而非间接调用者）；

如果有new关键字，this指向new出来的那个对象；

在事件中，this指向触发这个事件的对象，特殊的是，IE中的attachEvent中的this总是指向全局对象Window；

### 104. eval是做什么的？

它的功能是把对应的字符串解析成JS代码并运行；

应该避免使用eval，因为不安全，非常耗性能（2次，一次解析成js语句，一次执行）。

（注意：在项目里面写js代码的时候，禁止使用的，因为有安全因素）

### 105. null，undefined的区别？

null 表示一个对象被定义了，值为“空值”；

undefined 表示不存在这个值。

```
//
```

```
typeof undefined
```

```
//"undefined"
```

undefined :是一个表示"无"的原始值或者说表示"缺少值"，就是此处应该有一个值，但是还没有定义。

当尝试读取时会返回 undefined；

例如变量被声明了，但没有赋值时，就等于undefined。

```
//
```

```
typeof null
```

```
//"object"
```

null : 是一个对象(空对象, 没有任何属性和方法)；

例如作为函数的参数，表示该函数的参数不是对象；

```
//
```

注意：

在验证null时，一定要使用 === ，因为 == 无法分别 null 和 undefined

## 106. ["1", "2", "3"].map(parseInt) 答案是多少?

[1,NaN,NaN]

因为 parseInt 需要两个参数(val,radix)，其中 radix 表示解析时用的基数。

map 传了3个(element,index,array)，对应的 radix 不合法导致解析失败。

parseInt(转换数字,转换进制2、8、10、16)

一般般使用parseInt("010",10) //10    parseInt("010",8) //8

## 107. 事件、IE与火狐的事件机制有什么区别？ 如何阻止冒泡？

(1)我们在网页中的某个操作（有的操作对应多个事件）。例如：当我们点击一个按钮就会产生一个事件。是可以被 JavaScript 侦测到的行为。

(2)事件处理机制：IE是事件冒泡、Firefox同时支持两种事件模型，也就是：捕获型事件和冒泡型事件；

(3)ev.stopPropagation(); (旧ie的方法 ev.cancelBubble = true;)

## 108. 什么是闭包（closure），为什么要用它？

闭包是指有权访问另一个函数作用域中变量的函数，创建闭包的最常见的方式就是在一个函数内创建另一个函数，通过另一个函数访问这个函数的局部变量，利用闭包可以突破作用链域，将函数内部的变量和方法传递到外部。

//

闭包特性：

(1)函数内再嵌套函数

(2)内部函数可以引用外层的参数和变量

(3)参数和变量不会被垃圾回收机制回收

//li节点的onclick事件都能正确的弹出当前被点击的li索引

<ul>

<li> index = 0 </li>

<li> index = 1 </li>

<li> index = 2 </li>

<li> index = 3 </li>

</ul>

<script type="text/javascript">

var nodes = document.getElementsByTagName('li');

for(i = 0;i<nodes.length;i+=1) {

nodes[i].onclick = function() {

console.log(i+1); //不使用闭包的话，值每次都是4

}(4);

}

</script>

## 109. "use strict";是什么意思?使用它的有什么好处或坏处?

use strict是一种ECMAScript 5 添加的(严格)运行模式,这种模式使得 Javascript 在更严格的条件下运行,使JS编码更加规范化的模式,消除Javascript语法的一些不合理、不严谨之处,减少一些怪异行为。

默认支持的糟糕特性都会被禁用,比如不能用with,也不能在意外的情况下给全局变量赋值;

全局变量的显示声明,函数必须声明在顶层,不允许在非函数代码块内声明函数,arguments.callee也不允许使用;

消除代码运行的一些不安全之处,保证代码运行的安全,限制函数中的arguments修改,严格模式下的eval函数的行为和非严格模式的也不相同;

提高编译器效率,增加运行速度;

为未来新版本的Javascript标准化做铺垫。

## 110. new操作符具体干了什么呢?

(1)创建一个空对象,并且 this 变量引用该对象,同时还继承了该函数的原型。

(2)属性和方法被加入到 this 引用的对象中。

(3)新创建的对象由 this 所引用,并且最后隐式的返回 this 。

```
//  
var obj = {};  
obj.__proto__ = Base.prototype;  
Base.call(obj);
```

## 111. Javascript中,有一个函数,执行时对象查找时,永远不会去查找原型,这个函数是?

hasOwnProperty

JavaScript 中 hasOwnProperty 函数方法是返回一个布尔值,指出一个对象是否具有指定名称的属性。此方法无法检查该对象的原型链中是否具有该属性;该属性必须是对象本身的一个成员。

使用方法:

object.hasOwnProperty(propertyName)

其中参数object是必选项,一个对象的实例。

propertyName是必选项,一个属性名称的字符串值。

如果 object 具有指定名称的属性,那么JavaScript中hasOwnProperty函数方法返回 true,反之则返回 false。

## 112. 对JSON的了解?

JSON(JavaScript Object Notation)是一种轻量级的数据交换格式。

它是基于JavaScript的一个子集。数据格式简单,易于读写,占用带宽小。

如:

```
{"age": "12", "name": "back"}
```

(注意: 强调一下之前做的项目里面通过ajax请求, 后端返回数据常用的格式)



### 113. 解释一下Math常用哪些方法?

//x为传入参数

Math.abs(x); //返回数的绝对值。

Math.ceil(x); //对数进行上舍入或向上取整

Math.floor(x); //对数进行下舍入或向下取整

Math.max(x,y) //返回 x 和 y 中的最大值。

Math.min(x,y) //返回 x 和 y 中的最小值。

Math.random() // 返回 0 ~ 1 之间的随机数。

Math.round(x) //把数四舍五入为最接近的整数。

Math.PI //返回圆周率（约等于3.14159）。

### 114. js延迟加载的方式有哪些?

Defer(只支持 IE)和async、动态创建DOM方式（用得最多）、按需异步载入js

### 115. 知道ECMAScript 6里怎么写class么?

ES6的class可以看作只是一个语法糖，它的绝大部分功能，ES5都可以做到，新的class写法只是让对象原型的写法更加清晰、更像面向对象编程的语法而已。

//定义类

```
class Point {  
  constructor(x,y) { //构造方法  
    this.x = x; //this关键字代表实例对象  
    this.y = y;  
  }  
  toString() {  
    return '(' + this.x + ',' + this.y + ')';  
  }  
}
```

### 116. AMD (Modules/Asynchronous-Definition) 、CMD (Common Module Definition) 规范区别?

AMD 规范在这里: <https://github.com/amdjs/amdjs-api/wiki/AMD>

CMD 规范在这里: <https://github.com/seajs/seajs/issues/242>

## 117. call() 和 .apply() 的区别?

### 一、方法的定义

call方法:

语法: call(thisObj, Object)

定义: 调用一个对象的一个方法, 以另一个对象替换当前对象。

说明:

call 方法可以用来代替另一个对象调用一个方法。call 方法可将一个函数的对象上下文从初始的上下文改变为由 thisObj 指定的新对象。

如果没有提供 thisObj 参数, 那么 Global 对象被用作 thisObj。

apply方法:

语法: apply(thisObj, [argArray])

定义: 应用某一对象的一个方法, 用另一个对象替换当前对象。

说明:

如果 argArray 不是一个有效的数组或者不是 arguments 对象, 那么将导致一个 TypeError。

如果没有提供 argArray 和 thisObj 任何一个参数, 那么 Global 对象将被用作 thisObj, 并且无法被传递任何参数。

## 118. JavaScript中的作用域与变量声明提升?

参考地址: <http://blog.csdn.net/sunxing007/article/details/9034253>

<http://www.cnblogs.com/silentjesse/p/4024536.html>

## 119. 如何编写高性能的Javascript?

参考地址: <http://developer.51cto.com/art/200906/131335.htm>

## 120. 那些操作会造成内存泄漏?

内存泄漏是指任何对象在您不再拥有或需要它之后仍然存在。

垃圾回收器定期扫描对象, 并计算引用了每个对象的其他对象的数量, 如果一个对象的引用数量为0 (没有其他对象引用过该对象), 或对该对象的惟一引用是循环的, 那么该对象的内存即可回收。

//

setTimeout 的第一个参数使用字符串而非函数的话, 会引发内存泄漏。

闭包、控制台日志、循环 (在两个对象彼此引用且彼此保留时, 就会产生一个循环)

## 121. 知道什么是webkit么? 知道怎么用浏览器的各种工具来调试和debug代码么?

Chrome、safari浏览器的内核。

其中chrome的模板引擎使用的是Gecko

Gecko在互联网中是最流行的**排版引擎**之一，其流行程度仅次于**Trident**（用于Windows版的**Internet Explorer**4.0版本开始），其后尚有**WebKit**（用于**Safari**以及**Google Chrome**）以及**Presto**（用于Opera）。

可以通过js检查当前所使用浏览器的内核以及模版引擎

```
alert(navigator.userAgent);
```

## 122. 如何测试前端代码么? 知道BDD, TDD, Unit Test么? 知道怎么测试你的前端工程么(mocha, sinon, jasmine, qUnit..)?

如何测试前端代码么?

PC端浏览器：市场份额：<http://tongji.baidu.com/data/browser>

- 1.编写完前端静态html+css(注意不包括js)
- 2.加上js代码以及逻辑处理（再进行测试）

在各个浏览器里面做兼容测试：1）.ietest—ie6-ie11（ie模拟浏览器）、真正的windows版本自带的IE浏览器F12可以看见下图,可以切换不同的浏览器模式，后面跟着把文档模式也改成浏览器模式一样

2).FF(火狐浏览器)、chrome（谷歌浏览器）、opera（欧朋浏览器）、maxthon（遨游浏览器）、safari(苹果的浏览器)、猎豹安全浏览器

比较流氓的浏览器：（不知不觉电脑上就给安装上了）

360浏览器（可以切换浏览器模式）、搜狗高速浏览器、腾讯QQ浏览器、百度浏览器

还有其它用户量不大的浏览器就不用一一列举了。

<http://liulanqixiazai2016.lofter.com/>

移动端浏览器：移动份额

uc浏览器（知道为啥要第一个写吗？在移动端用户量最多）、腾讯qq浏览器、百度浏览器、360浏览器、猎豹浏览器、safari、chrome、opera、maxthon、ff、搜狗浏览器、海豚浏览器

2016年网页浏览器发展现状分析

<http://www.chinabgao.com/k/liulanqi/22495.html>

移动设备品牌占比

<http://tongji.baidu.com/data/mobile/brand>

知道BDD, TDD, Unit Test么?

TDD：测试驱动开发（Test-Driven Development）

BDD：行为驱动开发（Behavior Driven Development）

ATDD：验收测试驱动开发（Acceptance Test Driven Development）

参考地址: <http://www.testwo.com/article/414>

知道怎么测试你的前端工程么(mocha, sinon, jasmine, qUnit..)?

参考地址:

<http://www.tuicool.com/articles/yuMvQz>

sinon 参考地址: <http://www.maobashi.com/1366.html>

参考地址: <http://www.w3cmark.com/2016/446.html>

## 123. 前端templating(Mustache, underscore, handlebars)是干嘛的, 怎么用?

前端js模板引擎

Mustache:

参考地址: <http://www.open-open.com/lib/view/open1416792564461.html>

<http://www.cnblogs.com/iyangyuan/archive/2013/12/12/3471227.html>

underscore

[Underscore](#) 是一个 JavaScript 工具库, 它提供了一整套函数式编程的实用功能, 但是没有扩展任何 JavaScript 内置对象。他解决了这个问题: “如果我面对一个空白的 HTML 页面, 并希望立即开始工作, 我需要什么?” 他弥补了 [jQuery](#) 没有实现的功能, 同时又是 [Backbone](#) 必不可少的部分。

参考地址: <http://www.bootcss.com/p/underscore/>

简单做个了解就可以 (真实用在仔细学习)

## 124. 简述一下 Handlebars 的基本用法?

参考地址: [http://baike.baidu.com/link?url=OF4LwdRf8jbdYZt8w1Yig7Q7dSCRA2jvpfxRISOE91l6kQz1wZwmHnjb1CEJZ4Ov7IE5wHVPvQbBam8B-O0E\\_](http://baike.baidu.com/link?url=OF4LwdRf8jbdYZt8w1Yig7Q7dSCRA2jvpfxRISOE91l6kQz1wZwmHnjb1CEJZ4Ov7IE5wHVPvQbBam8B-O0E_)

<http://www.cnblogs.com/iyangyuan/archive/2013/12/12/3471227.html>

## 125. 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

## 126. 谈一谈你对 ECMAScript6 的了解?

ECMAScript 6 是JavaScript语言的下一代标准, 已经在2015年6月正式发布了。它的目标, 是使得JavaScript语言可以用来编写复杂的大型应用程序, 成为企业级开发语言。

标准的制定者有计划，以后每年发布一次标准，使用年份作为标准的版本。因为当前版本的ES6是在2015年发布的，所以又称ECMAScript 2015。也就是说，ES6就是ES2015

## 127. 说说最近最流行的一些东西吧？常去哪些网站？

Node.js、Mongodb、npm、MVVM、MEAN、three.js、React。

网站：w3cfuns、sf、hacknews、CSDN、慕课、博客园、InfoQ、w3cplus等

## 128. 前端安全问题？

sql注入原理

就是通过把SQL命令插入到Web表单递交或输入域名或页面请求的查询字符串，最终达到欺骗服务器执行恶意的SQL命令。

总的来说有以下几点：

- 1.永远不要信任用户的输入，要对用户的输入进行校验，可以通过正则表达式，或限制长度，对单引号和双"-"进行转换等。
- 2.永远不要使用动态拼装SQL，可以使用参数化的SQL或者直接使用存储过程进行数据查询存取。
- 3.永远不要使用管理员权限的数据库连接，为每个应用使用单独的权限有限的数据库连接。
- 4.不要把机密信息明文存放，请加密或者hash掉密码和敏感的信息。

XSS原理及防范

Xss(cross-site scripting)攻击指的是攻击者往Web页面里插入恶意html标签或者javascript代码。比如：攻击者在论坛中放一个 看似安全的链接，骗取用户点击后，窃取cookie中的用户私密信息；或者攻击者在论坛中加一个恶意表单，当用户提交表单的时候，却把信息传送到攻击者的服务器中，而不是用户原本以为的信任站点。

XSS防范方法

- 1.代码里对用户输入的地方和变量都需要仔细检查长度和对"<",">",";",'"'等字符做过滤；其次任何内容写到页面之前都必须加以encode，避免不小心把html tag 弄出来。这一个层面做好，至少可以堵住超过一半的XSS 攻击。
- 2.避免直接在cookie 中泄露用户隐私，例如email、密码等等。 3.通过使cookie 和系统ip 绑定来降低cookie 泄露后的危险。这样攻击者得到的cookie 没有实际价值，不可能拿来重放。

4.尽量采用POST 而非GET 提交表单

XSS与CSRF有什么区别吗？

XSS是获取信息，不需要提前知道其他用户页面的代码和数据包。CSRF是代替用户完成指定的动作，需要知道其他用户页面的代码和数据包。

要完成一次CSRF攻击，受害者必须依次完成两个步骤：

1.登录受信任网站A，并在本地生成Cookie。      2.在不登出A的情况下，访问危险网站B。  
CSRF的防御

1. 服务端的CSRF方式方法很多样，但总的思想都是一致的，就是在客户端页面增加伪随机数。
2. 使用验证码

### 129. 如何判断当前脚本运行在浏览器还是node环境中？（阿里）

通过判断Global对象是否为window，如果不为window，当前脚本没有运行在浏览器中

### 130. 把 Script 标签 放在页面的最底部的body封闭之前 和封闭之后有什么区别？浏览器会如何解析它们？

参考资料：

[https://segmentfault.com/a/1190000004292479?\\_ea=560235](https://segmentfault.com/a/1190000004292479?_ea=560235)

<http://www.zhihu.com/question/20027966>

### 131. 移动端的点击事件的有延迟，时间是多久，为什么会有？怎么解决这个延时？

参考资料：<http://www.mamicode.com/info-detail-666685.html>

### 132. Node.js的适用场景？

高并发、聊天、实时消息推送、服务器做前端资源压缩

## 第三部分 jQuery 框架

### 1. jQuery绑定事件的方法书写至少两种？

答：bind(),on(),click()

### 2. jQuery对象和Dom对象相互转换

答：jQuery对象转原生Dom对象 `$(“div”).get(0)`

原生Dom对象转jQuery对象 `$(ele)`

### 3. Js用过的插件

答：tween.js,swiper.js,touch.js,zepto.js

### 4. jQuery如何获取元素？

答：\$(selector)

## 5. JQuery链式写法原理

答: jquery里的方法调用完毕返回return this

## 6. 用jQuery如何获取标签的id

答: \$(element).attr("id")

## 7. 用jQuery如何获取标签<div></div>

答: \$("div")

## 8. 用jQuery如何获取标签内容

答: \$(element).text()/\$(element).html();

## 9. 用jQuery如何更换背景色

答: \$(element).css({backgroundColor:"red"})

## 10. jquery中，如何阻止事件冒泡和浏览器默认行为?

答: e.preventDefault();e.stopPropagation();

## 11. \$(document).ready与window.onload有何区别?

答: \$(document).ready 对于支持DOMContentLoaded事件的浏览器，会使用该事件，所有的节点渲染完毕，就执行,可以调用多次，不支持的就使用 onload事件

window.onload事件要等到所有的资源加载完毕，如图片，视频，js脚本等,只能绑定一次，后面绑定的覆盖前面所绑定的事件

## 12. 列举jQuery的优势

答: 兼容性处理，dom操作，获取元素简单，封装了ajax.....

## 13. jquery中 :odd :gt 这两种选择器的含义以及用法举例

:odd 匹配所有索引值为奇数的元素，从 0 开始计数 \$("div:odd")

:gt(index) 匹配所有大于给定索引值的元素 \$("div:gt(2)")

## 14. jquery中，使用nth-child(3n+2)所选中的是以下哪些

nth-child ( ) 匹配其父元素下的第N个子或奇偶元素



'eq(index)' 只匹配一个元素，而这个将为每一个父元素匹配子元素。:nth-child从1开始的，而:eq()是从0算起的

2, 5, 8

```
//传统写法
var div = document.getElementById('test');
div.style.color = 'red';
div.style.width = '20px';
```

15. 使用jQuery中的链式书写方法书写如下代码

答: `$("#test").css("color","red").width(20)`

16. jQuery中对动画的操作命令有哪些? 举出不少于8种, 并说明含义

答: `animate()`,`fadeIn()`,`fadeOut()`,`slideDown()`,`slideUp()`,`hide()`,`show()`,`toggle()`

17. jQuery中绑定事件的方法是什么, 尝试书写例子: 为id名为con的div绑定一个事件, 事件为弹出一个数字0。获取标签时不要使用原生javascript方法。

答: `$("#con").click(function){alert(0)}`

18. jQuery中获取相对文档的坐标的方法是什么?

答: `offset()`,获取匹配元素在当前视口的相对偏移。返回的对象包含两个整型属性: `top` 和 `left`, 以像素计。此方法只对可见元素有效。

`position()`获取匹配元素相对定位父级的偏移。没有定位父级即为文档坐标

返回的对象包含两个整型属性: `top` 和 `left`。为精确计算结果, 请在补白、边框和填充属性上使用像素单位。此方法只对可见元素有效。

19. jQuery中对于节点的操作有哪些 (举出不少于8种)? 写出含义以及用法。

文档处理
内部插入
<code>append(content fn)</code>
<code>appendTo(content)</code>
<code>prepend(content fn)</code>
<code>prependTo(content)</code>
外部插入
<code>after(content fn)</code>
<code>before(content fn)</code>
<code>insertAfter(content)</code>
<code>insertBefore(content)</code>
包裹
<code>wrap(html ele fn)</code>
<code>unwrap()</code>
<code>wrapAll(html ele)</code>
<code>wrapInner(html ele fn)</code>
替换
<code>replaceWith(content fn)</code>
<code>replaceAll(selector)</code>
删除
<code>empty()</code>
<code>remove([expr])</code>
<code>detach([expr])</code>
复制
<code>clone([Even[,deepEven]])</code>

20. jQuery动画效果的实现

答: `$ele().animate({top:100},500)`



21. jQuery中的两个合成事件是\_\_\_\_\_和\_\_\_\_\_

答: toggle、hover

22. 用jQuery获取标签“<div id='con'>div内部的信息</div>”当中的内容的代码为\_\_\_\_\_?

答: \$("#con").text()

23. 举出不少于6种的jQuery中对动画的操作命令，并说明其含义: \_\_\_\_\_、  
\_\_\_\_\_, \_\_\_\_\_、 \_\_\_\_\_、 \_\_\_\_\_、 \_\_\_\_\_

效果	a
基本	
show([s],[e],[fn])	
hide([s],[e],[fn])	
toggle([s],[e],[fn])	
滑动	
slideDown([s],[e],[fn])	
slideUp([s],[e],[fn])	
slideToggle([s],[e],[fn])	
淡入淡出	
fadeIn([s],[e],[fn])	
fadeOut([s],[e],[fn])	
fadeTo([s],o,[e],[fn])	
fadeToggle([s],[e],[fn])	

24. 编写一个jQuery的扩展方法（插件），让调用的对象设置宽度为200像素

```
jQuery.fn.extend({  
  setWidth:function() {  
    $(this).width(200);  
    return this;  
  }  
})
```

25. 你使用过哪些Javascript库?

答: jQuery Zepto (注意了: 面试官可能问你这两个库都什么区别)

jQuery是在Web上应用很广泛的JavaScript库，它提供了大量的工具和API函数，使用它的人相当普遍，使用的门槛也比较低。

Zepto最初是为移动端开发的库，是jQuery的轻量级替代品，因为它的API和jQuery相似，而文件更小，对任何项目都是个不错的选择，Zepto是不支持IE浏览器。

针对移动端程序，Zepto还有一些基本的触摸事件可以用来做触摸屏交互，如: tap事件——tap, singleTap, doubleTap, longTap;

Swipe事件——swipe, swipeLeft, swipeRight, swipeUp, swipeDown。

更多区别详情参考地址：<http://blog.csdn.net/kongjiea/article/details/42522305>

## 26. jQuery从哪个版本开始不兼容IE低版本，你项目中用的哪个版本，为什么用这个版本，其它版本的jQuery与你用的版本有什么区别？

注意了面试官不想单独问的是对版本的认识，出题的目的是考验咱们对jquery源码是否仔细读过或了解过

版本1.12将成为针对指定的微软Windows默认浏览器版本的最后一个官方支持,就是说版本1.13将只会支持IE8及其之后的版本，丢弃了IE6余IE7。

例如1.9版本丢弃了live()方法。其它可以去官网查看。

例如 \$.browser(), jQuery 1.9 还是支持 IE 6/7/8，但是 2.0 版本将不再支持这三个版本

参考地址：<http://www.alixixi.com/web/a/2013052788951.shtml>

## 27. 我们如何使用Canvas来画一条简单的线？手写出代码（注意：不是让你敲键盘，而是手写）

1.定义Canvas区域

2.获取访问canvas上下文区域

3.绘制图形

<canvas id="mycanvas" width="600" height="500">当前浏览器不支持canvas，请升级浏览器或者使用高版本的其它浏览器</canvas>

```
<script>
```

```
var cvs = document.getElementById("mycanvas");
```

```
// 获取上下文（绘制环境）
```

```
var ctx = cvs.getContext("2d");
```

```
// 注意如果画多条线的时候需要使用beginPath和closePath
```

```
// 开始绘制
```

```
ctx.beginPath();
```

```
// 设置起点
```

```
ctx.moveTo(10,10);
```

```
// 设置下一个点
```

```
ctx.lineTo(200,100);
```

```
// 关闭绘制
```

```
ctx.closePath();
```

```
// 设置填充
```

```
ctx.stroke();
```

```
</script>
```

## 28. 针对jQuery性能的优化方法？

基于Class的选择性的性能相对于Id选择器开销很大，因为需遍历所有DOM元素。

//

频繁操作的DOM，先缓存起来再操作。用Jquery的链式调用更好。

比如: `var str=$("#a").attr("href");`

`//`

`for (var i = size; i < arr.length; i++) {}`

for 循环每一次循环都查找了数组 (arr) 的.length 属性, 在开始循环的时候设置一个变量来存储这个数字, 可以让循环跑得更快:

`for (var i = size, length = arr.length; i < length; i++) {}`

## 29. jQuery与jQuery UI有啥区别?

jQuery`是一个js库, 主要提供的功能是选择器, 属性修改和事件绑定等等。

jQuery UI`则是在jQuery的基础上, 利用jQuery的扩展性, 设计的插件。提供了一些常用的界面元素, 诸如对话框、拖动行为、改变大小行为等等

## 30. jQuery和Zepto的区别? 各自的使用场景?

参考答案: <http://blog.csdn.net/kongjica/article/details/42522305>

jQuery偏向于pc端

Zepto偏向于移动端

## 31. Zepto的点透问题如何解决?

`e.preventDefault();` // 阻止“默认行为”

参考地址:

<http://www.bubuko.com/infodetail-649496.html>

<http://my.oschina.net/chainlong/blog/225702>

# 第四部分 AJAX

## 1. 如何实现局部内容和后台交互时的变化

答: 使用ajax

## 2. 什么是AJAX , AJAX原理

答: AJAX即“Asynchronous Javascript And XML”(异步JavaScript和XML) 通过在后台与服务器进行少量数据交换, AJAX 可以使网页实现异步更新。这意味着可以在不重新加载整个网页的情况下, 对网页的某部分进行更新。传统的网页(不使用 AJAX)如果需要更新内容, 必须重载整个网页页面。

原理: XMLHttpRequest

### 3. 使用异步加载获取JS数据至少两种方法?

答: post, get

### 4. 什么是JSON, JSON的格式

答: JSON(JavaScript Object Notation) 是一种轻量级的数据交换格式。JSON 语法是 JavaScript 对象表示语法的子集。

数据在键值对中

数据由逗号分隔

花括号保存对象

方括号保存数组

### 5. 除了AJAX还有什么模板可以与后台交互

答: jsonp

### 6. AJAX中的跨域问题: 什么是跨域? 如何解决跨域问题?

答: 域不一样的, 即为跨域, 包括 (协议, 域名, 端口号)

1.// 指定允许其他域名访问

```
header('Access-Control-Allow-Origin:');
```

2.使用jsonp

### 7. AJAX的流程是什么?

1.客户端产生js的事件

2.创建XMLHttpRequest对象

3.对XMLHttpRequest进行配置

4.通过AJAX引擎发送异步请求

5.服务器端接收请求并且处理请求, 返回html或者xml内容

6.XML调用一个callback()处理响应回来的内容

7.页面局部刷新

### 8. AJAX如何调用JSON数据

答: xml.responseText

### 9. AJAX能够处理哪些格式的文件

<https://developer.mozilla.org/zh-CN/docs/Web/API/XMLHttpRequest>

设置该值能够改变响应类型。就是告诉服务器你期望的响应格式。

Value	Data type of response property
"" (空字符串)	字符串(默认值)
"arraybuffer"	ArrayBuffer
"blob"	Blob
"document"	Document
"json"	JavaScript 对象，解析自服务器传递回来的JSON字符串。
"text"	字符串

XMLHttpRequest 2.0 新增了多种类型  
支持txt,json,js,图片，文档等等...

## 10. AJAX实现表单验证用户注册流程

用户触发ajax请求，后台接口返回json格式字符串

## 11. JSON数据的解析方法

- 1.JSON.parse(json)
- 2.new Function("return " + json) ();
- 3.eval("(" + json + ")")

## 12. 如何转化成JSON字符串

答: JSON.stringify(json)

## 13. 说明异步请求的get和post方法的区别

- 1.使用Get请求时,参数在URL中显示,而使用Post方式,则不会显示出来
- 2.使用Get请求发送数据量小,Post请求发送数据量大

get 请求:

	Cookies	Local storage
客户端/ 服务端	客户端和服务端都能访问数据。Cookie的数据通过每一个请求发送到服务端	只有本地浏览器端可访问数据，服务器不能访问本地存储直到故意通过POST或者GET的通道发送到服务器
大小	每个cookie有4095byte	每个域5MB
过期	Cookies有有效期，所以在过期之后cookie和cookie数据会被删除	没有过期数据，无论最后用户从浏览器删除或者使用JavaScript程序删除，我们都需要删除

```

function btn_get_click() {
    var xmlhttp = window.XMLHttpRequest ?
        new XMLHttpRequest() : new ActiveXObject("Microsoft.XMLHTTP");

    var username = document.getElementById("txt_username").value;
    var age = document.getElementById("txt_age").value;

    //添加参数,以求每次访问不同的url,以避免缓存问题
    xmlhttp.open("get", "Server.aspx?username=" + encodeURIComponent(username)
        + "&age=" + encodeURIComponent(age) + "&random=" + Math.random());

    xmlhttp.onreadystatechange = function () {
        if (xmlhttp.readyState == 4 && xmlhttp.status == 200) {
            document.getElementById("result").innerHTML = xmlhttp.responseText;
        }
    }

    //发送请求,参数为null
    xmlhttp.send(null);
}

```

post 请求:

```

function btn_post_click() {
    var xmlhttp = window.XMLHttpRequest ?
        new XMLHttpRequest() : new ActiveXObject("Microsoft.XMLHTTP");

    var username = document.getElementById("txt_username").value;
    var age = document.getElementById("txt_age").value;

    var data = "username=" + encodeURIComponent(username)
        + "&age=" + encodeURIComponent(age);

    //不用担心缓存问题
    xmlhttp.open("post", "Server.aspx", true);

    //必须设置,否则服务器端收不到参数
    xmlhttp.setRequestHeader("Content-Type", "application/x-www-form-urlencoded");

    xmlhttp.onreadystatechange = function () {
        if (xmlhttp.readyState == 4 && xmlhttp.status == 200) {
            document.getElementById("result").innerHTML = xmlhttp.responseText;
        }
    }

    //发送请求,要data数据
    xmlhttp.send(data);
}

```

## 14. 列举AJAX的优势

答：传统的Web应用交互由用户触发一个HTTP请求到服务器,服务器对其进行处理后再返回一个新的HTML页到客户端,每当服务器处理客户端提交的请求时,客户都只能空闲等待,并且哪怕只是一次很小的交互、只需从服务器端得到很简单的一个数据,都要返回一个完整的HTML页,而用户每次都要浪费时间和带宽去重新读取整个页面。这个做法浪费了许多带宽,由于每次应用的交互都需要向服务器发送请求,应用的响应时间就依赖于服务器的响应时间。这导致了用户界面的响应比本地应用慢得多。

与此不同, AJAX应用可以仅向服务器发送并取回必需的数据, 它使用SOAP或其它一些基于XML的Web Service接口, 并在客户端采用JavaScript处理来自服务器的响应。因为在服务器和浏览器之间交换的数据大量减少, 结果我们就能看到响应更快的应用。同时很多的处理工作可以在发出请求的客户端机器上完成, 所以Web服务器的处理时间也减少了。

## 15. AJAX中创建请求的兼容性处理

答：var xmlhttp = window.XMLHttpRequest ? new XMLHttpRequest() : new ActiveXObject("Microsoft.XMLHTTP");

## 16. 对于request.status的http状态码 用于表示网页服务器HTTP响应状态的3位数字代码。1,2,3,4,5开头的状态码分别代表什么（提示：404页面表示禁止访问等）

1xx（临时响应）表示临时响应并需要请求者继续执行操作的状态代码。

2xx（成功）表示成功处理了请求的状态代码。

3xx（重定向）表示要完成请求, 需要进一步操作。通常, 这些状态代码用来重定向。

4xx（请求错误） 这些状态代码表示请求可能出错, 妨碍了服务器的处理。

5xx（服务器错误） 这些状态代码表示服务器在尝试处理请求时发生内部错误。 这些错误可能是服务器本身的错误, 而不是请求出错。

## 17. 下列对get和post的区别描述错误的是（ ）

A、get比post更常用

B、get发送信息为明文发送, 安全性较差

C、get的性能只有post的1/3

D、get传输数据是通过URL进行的

答案：C

## 18. 如何将js对象转化成JSON字符串 ( )

A、JSON.parseInt()

B、JSON.parse()

C、JSON.string ()

D、JSON.stringify ()

答案：D

## 19. AJAX流程 代码填空

```
var xml = __ window.XMLHttpRequest ?
```

```
__ new XMLHttpRequest() : new ActiveXObject("Microsoft.XMLHTTP") __ ;
```

/\*创建

一个AJAX对象\*/

```
var url = 'a.html';
```

/\*设置要访问的url地址\*/

```
xml. __ open __ ( "get" , __ url __ , __ true __ ); /*确定即将连接的网址及方式,发起同步请求*/
```

```
xml. __ send(null) __ ;
```

/\*发送请求\*/

```
xml. __ onload __ = function () {
```

/\*当AJAX请求成功后执行这个函数\*/

```
con.innerHTML = __ JSON.parse(xml.responseText) __ ;
```

/\*对服务器返回的数据进行处理\*/

```
}
```

## 20. AJAX调用JSON数据 代码填空

json内容如下：

```
[
```

```
{
```

```
"pic": "spring.jpg",
```

```
"dis": "天街小雨润如酥，草色遥看近却无。最是一年春好处，绝胜烟柳满皇都。"
```

```
},
```

```
{
```

```
"pic": "summer.jpg",
```

```
"dis": "泉眼无声惜细流，树阴照水爱晴柔。小荷才露尖尖角，早有蜻蜓立上头。"
```

```
},
```

```
{
```

```
"pic": "fall.jpg",
```

```
"dis": "自古逢秋悲寂寥，我言秋日胜春朝。晴空一鹤排云上，便引诗情到碧霄。"
```

```
},
```

```
{
```

```
"pic": "winter.jpg",
```

```
"dis": "墙角数枝梅，凌寒独自开。遥知不是雪，为有暗香来。"
```

```
}
```

```
]
```

如何获取到"自古逢秋悲寂寥，我言秋日胜春朝。晴空一鹤排云上，便引诗情到碧霄。"这段文字？

...../\*前面代码省略\*/



```
xml.onload=function(){
    var obj = ____JSON.parse(xml.responseText)____
    var con = ____obj[2].dis____
}
```

## 21. AJAX的意义，用法，优点和原理

### 1.什么是AJAX?

AJAX全称为“Asynchronous JavaScript and XML”（异步JavaScript和XML），是一种创建交互式网页应用的网页开发技术。它使用：

使用XHTML+CSS来标准化呈现；

使用XML和XSLT进行数据交换及相关操作；

使用XMLHttpRequest对象与Web服务器进行异步数据通信；

使用JavaScript操作Document Object Model进行动态显示及交互；

使用JavaScript绑定和处理所有数据。

### 2.与传统的web应用比较

传统的Web应用交互由用户触发一个HTTP请求到服务器,服务器对其进行处理后再返回一个新的HTML页到客户端,每当服务器处理客户端提交的请求时,客户都只能空闲等待,并且哪怕只是一次很小的交互、只需从服务器端得到很简单的一个数据,都要返回一个完整的HTML页,而用户每次都要浪费时间和带宽去重新读取整个页面。这个做法浪费了许多带宽,由于每次应用的交互都需要向服务器发送请求,应用的响应时间就依赖于服务器的响应时间。这导致了用户界面的响应比本地应用慢得多。

与此不同, AJAX应用可以仅向服务器发送并取回必需的数据, 它使用SOAP或其它一些基于XML的Web Service接口, 并在客户端采用JavaScript处理来自服务器的响应。因为在服务器和浏览器之间交换的数据大量减少, 结果我们就能看到响应更快的应用。同时很多的处理工作可以在发出请求的客户端机器上完成, 所以Web服务器的处理时间也减少了。

### 3.AJAX的工作原理

Ajax的工作原理相当于在用户和服务器之间加了一个中间层(AJAX引擎),使用户操作与服务器响应异步化。并不是所有的用户请求都提交给服务器,像一些数据验证和数据处理等都交给Ajax引擎自己来做,只有确定需要从服务器读取新数据时再由Ajax引擎代为向服务器提交请求。

Ajax其核心有JavaScript、XMLHttpRequest、DOM对象组成,通过XmlHttpRequest对象来向服务器发异步请求,从服务器获得数据,然后用JavaScript来操作DOM而更新页面。这其中最关键的一步就是从服务器获得请求数据。

## 22. Jsonp的意义，用法，优点和原理

JSONP的优点是：它不像XMLHttpRequest对象实现的Ajax请求那样受到同源策略的限制；它的兼容性更好,在更加古老的浏览器中都可以运行,不需要XMLHttpRequest或ActiveX的支持；并且在请求完毕后可以通调用callback的方式回传结果。

JSONP的缺点则是：它只支持GET请求而不支持POST等其它类型的HTTP请求；它只支持跨域HTTP请求这种情况，不能解决不同域的两个页面之间如何进行JavaScript调用的问题。

### JSONP原理

JSONP的最基本的原理是：动态添加一个<script>标签，而script标签的src属性是没有跨域的限制的。这样说来，这种跨域方式其实与ajax XMLHttpRequest协议无关了。

这样其实"jQuery AJAX跨域问题"就成了个伪命题，jquery \$.ajax方法名有误导人之嫌。

如果设为dataType: 'jsonp', 这个\$.ajax方法就和ajax XMLHttpRequest没什么关系了，取而代之的则是JSONP协议。JSONP是一个非官方的协议，它允许在服务器端集成Script tags返回至客户端，通过javascript callback的形式实现跨域访问。

JSONP即JSON with Padding。由于同源策略的限制，XMLHttpRequest只允许请求当前源（域名、协议、端口）的资源。如果要进行跨域请求，我们可以通过使用html的script标记来进行跨域请求，并在响应中返回要执行的script代码，其中可以直接使用JSON传递javascript对象。这种跨域的通讯方式称为JSONP。

jsonpCallback 函数jsonp1236827957501(...): 是浏览器客户端注册的，获取跨域服务器上的json数据后，回调的函数

Jsonp的执行过程如下：

首先在客户端注册一个callback (如:'jsoncallback'), 然后把callback的名字(如:jsonp1236827957501)传给服务器。注意：服务端得到callback的数值后，要用jsonp1236827957501(.....)把将要输出的json内容包括起来，此时，服务器生成 json 数据才能被客户端正确接收。

然后以 javascript 语法的方式，生成一个function， function 名字就是传递上来的参数 'jsoncallback'的值 jsonp1236827957501 .

最后将 json 数据直接以入参的方式，放置到 function 中，这样就生成了一段 js 语法的文档，返回给客户端。

客户端浏览器，解析script标签，并执行返回的 javascript 文档，此时javascript文档数据，作为参数，传入到了客户端预先定义好的 callback 函数(如上例中jquery \$.ajax()方法封装的success: function (json))里。

可以说jsonp的方式原理上和<script src="http://跨域/...xx.js"></script>是一致的(qq空间就是大量采用这种方式来实现跨域数据交换的)。JSONP是一种脚本注入(Script Injection)行为，所以有一定的安全隐患。

## 23. 请尽可能详尽的解释AJAX的工作原理。

第一步：创建ajax对象（XMLHttpRequest/ActiveXObject(Microsoft.XMLHttp))

第二步：判断数据传输方式(GET/POST)

第三步：打开链接 open()

第四步：发送 send()

当ajax对象完成第四步（onreadystatechange）数据接收完成，判断http响应状态（status）200-300之间或者304（缓存）执行回调函数

**注意：**检测XMLHttpRequest对象的readyState属性，该属性表示请求 / 响应过程的当前活动阶段，属性值如下：

0：未初始化。尚未调用open()方法

1：启动。已经调用open()方法，但尚未调用send()方法

2：发送。已经调用send()方法，但尚未接收到响应

3：接收。已经接收到部分响应数据

4：完成。已经接收到全部响应数据，而且已经可以在客户端使用了（如果写原生的js ajax请求需要等到readyState==4的时候再做处理）其他的js库已经做好处理了，放心使用

## 24. 你对Ajax的理解。

- 1.AJAX 是一种用于创建更好更快以及交互性更强的 Web 应用程序的技术;
- 2.通过Ajax你的 JavaScript 可在不重载页面的情况与 Web 服务器交换数据;
- 3.AJAX 在浏览器与 Web 服务器之间使用异步数据传输 (HTTP 请求) , 这样就可使网页从服务器请求少量的信息, 而不是整个页面;
- 4.AJAX 可使因特网应用程序更小、更快, 更友好;
- 5.AJAX 是一种独立于 Web 服务器软件的浏览器技术。

## 25. get与post两种方式的优缺点。

get:

get是从服务器上获取数据, post是向服务器传送数据;

get传送的数据量较小, 不能大于2KB。post传送的数据量较大, 一般被默认为不受限制。但理论上, IIS4中最大量为80KB, IIS5中为100KB;

get安全性非常低, post安全性较高。但是执行效率却比Post方法好;

get是把参数数据队列加到提交表单的ACTION属性所指的URL中, 值和表单内各个字段一一对应, 在URL中可以看到;

在做数据查询时, 建议用Get方式;

post:

post是通过HTTP post机制, 将表单内各个字段与其内容放置在HTML HEADER内一起传送到ACTION属性所指的URL地址, 用户看不到这个过程;

所以: 包含机密信息的话, 建议用Post数据提交方式; 而在做数据添加、修改或删除时, 建议用Post方式。

## 26. 一次完整的HTTP事务是怎样一个过程?

- a.域名解析
- b.发起TCP的三次握手
- c.建立TCP连接后发起http请求
- d.服务器端响应http请求, 浏览器得到html码
- e.浏览器解析html代码, 并请求html代码中的资源
- f.浏览器对页面进行渲染并呈现给客户

更多详细参考地址: [http://www.360doc.com/content/14/0725/20/1073512\\_397054861.shtml](http://www.360doc.com/content/14/0725/20/1073512_397054861.shtml)

## 27. CORS是什么? 对于跨域请求, 如何将附带凭证 (HTTP Cookie及HTTP认证信息) 的请求发送至服务器? (提示: 后续真实工作中可能会用到)

答: 跨域请求一直是网页编程中的一个难题, 在过去, 绝大多数人都倾向于使用JSONP来解决这一问题。不过现在, 我们可以考虑一下W3C中一项新的特性——CORS (Cross-Origin Resource Sharing) 了。CORS是现代浏览器支持跨域资源请求的一种方式。

解决方法:

当你使用XMLHttpRequest发送请求时,浏览器发现该请求不合同源策略,会给该请求加一个请求头: Origin, 后台 (php或者其它接收数据方) 进行一系列处理,如果确定接受请求则在返回结果中加入一个响应头: Access-Control-Allow-Origin;浏览器判断该相应头中是否包含Origin的值,如果有则浏览器会处理响应,我们就可以拿到响应数据,如果不包含浏览器直接驳回,这时我们无法拿到响应数据。

更多参考地址: [www.cnblogs.com/dojo-lzz/p/4265637.html](http://www.cnblogs.com/dojo-lzz/p/4265637.html)

<http://www.cnblogs.com/rainman/archive/2011/02/20/1959325.html>

## 28. 什么是Ajax和JSON, 它们的优缺点

什么是Ajax? ? ?

术语Ajax用来描述一组技术,它使浏览器可以为用户提供更为自然的浏览体验。

Ajax它是“Asynchronous JavaScript + XML的简写”

定义Ajax:

Ajax不是一种技术。实际上,它由几种蓬勃发展的技术以新的强大方式组合而成。Ajax包含:

- \* 基于XHTML和CSS标准的表示;
- \* 使用Document Object Model进行动态显示和交互;
- \* 使用XMLHttpRequest与服务器进行异步通信;
- \* 使用JavaScript绑定一切。

Ajax的工作原理 (一定要掌握)

一共4步, 每一步代表什么意思以及什么作用? 面试题1里面有提到

对于JSON, 首先要明白JSON和XML一样也是一种简单文本格式。

JSON来自于javascript, 但是应用上面远远不止是javascript的json, 它相当于xml, 是一种比较流行的标准格式, 是数据的载体

javascript json 是用于javascript程序的, 它可以支持对象引用以及函数; ajax json只是最简单的字符串和数字, 它不可能完全支持javascript里面的数据类型。

相对于XML, JSON更加易读、更便于肉眼检查。在语法的层面上, JSON与其他格式的区别是在于分隔数据的字符

## 29. 浅谈一下如何避免用户多次点击造成的多次请求。

答: 我们在访问有的网站, 输入表单完成以后, 单击提交按钮进行提交以后, 提交按钮就会变为灰色, 用户不能再单击第二次, 直到重新加载页面或者跳转。这样, 可以一定程度上防止用户重复提交导致应用程序上逻辑错误。

还有很多其他的方式进行防止重复点击提交, 如

1> 定义标志位:

点击触发请求后, 标志位为false量; 请求 (或者包括请求后具体的业务流程处理) 后, 标志位为true量。通过标志位来判断用户点击是否具备应有的响应。

2> 卸载及重载绑定事件:

点击触发请求后，卸载点击事件；请求（或者包括请求后具体的业务流程处理）后，重新载入绑定事件。

### 3> 替换（移除）按钮DOM

点击触发请求后，将按钮DOM对象替换掉（或者将之移除），自然而然此时不在具备点击事件；请求（或者包括请求后具体的业务流程处理）后，给新的按钮DOM定义点击事件。

更多讲解参考：<http://www.cnblogs.com/jinguangguo/archive/2013/05/20/3086925.html>

## 30. ajax是什么?

ajax的全称：Asynchronous Javascript And XML。

异步传输+js+xml。

所谓异步，在这里简单地解释就是：向服务器发送请求的时候，我们不必等待结果，而是可以同时做其他的事情，等到有了结果它自己会根据设定进行后续操作，与此同时，页面是不会发生整页刷新的，提高了用户体验。

//

(1)创建XMLHttpRequest对象,也就是创建一个异步调用对象

(2)创建一个新的HTTP请求,并指定该HTTP请求的方法、URL及验证信息

(3)设置响应HTTP请求状态变化的函数

(4)发送HTTP请求

(5)获取异步调用返回的数据

(6)使用JavaScript和DOM实现局部刷新

## 31. 同步和异步的区别?

同步的概念应该是来自于操作系统中关于同步的概念:

不同进程为协同完成某项工作而在先后次序上调整(通过阻塞,唤醒等方式)。同步强调的是顺序性，谁先谁后；异步则不存在这种顺序性。

//

同步：浏览器访问服务器请求，用户看得到页面刷新，重新发请求,等请求完，页面刷新，新内容出现，用户看到新内容,进行下一步操作。

//

异步：浏览器访问服务器请求，用户正常操作，浏览器后端进行请求。等请求完，页面不刷新，新内容也会出现，用户看到新内容。

## 32. 如何解决跨域问题?

答：jsonp、iframe、window.name、window.postMessage、服务器上设置代理页面

## 33. 异步加载的方式有哪些?

(1)defer，只支持 IE

(2)async:

(3)创建 script，插入到 DOM 中，加载完毕后 callBack