

一、你做的页面在哪些浏览器测试过？这些浏览器的内核分别是什么？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？

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 好象是专门为了解决 ie6 bug 而生的。

8. 注释也能产生 bug~~~ “多出来的一只猪。” 这是前人总结这个 bug 使用的文案, ie6 的这个 bug 下, 大家会在页面看到猪字出现两遍, 重复的内容量因注释的多少而变。解决方案: 用 “<![if !IE]> picRotate start <![endif]->” 方法写注释。

9. 里加 float <div/>, 这是一个典型的, 棘手的兼容问题, 希望引起大家正视, 给 li 不同的属性会有不同的解释效果, ff 下的解释稍可理解, ie6 下的解释会让你摸不着头脑, 由于问题的复杂性, 将另起一文专门讨论该问题。在《ul 使用心得》一文里有相关成果, 却没给出问题解决的过程。

10. img 下的留白。解决方案: 给 img 设定 display:block。

11. 失去 line-height。<div style=" line-height:20px" >文字</div>, 很遗憾, 在 ie6 下单行文字 line-height 效果消失了。。。, 原因是 这个 inline-block 元素和 inline 元素写在一起了。解决方案: 让 img 和文字都 float 起来。

12. 链接的 hover 状态。a:hover img{width:300px} 我们想让鼠标 hover 时, 链接里包含的图片宽度变化, 可惜在 ie6 下无效, ie7、ff 下有效。

13. 非链接的 hover 状态。div:hover{} 这样的样式 ie6 是不认的, 在 ie7、ff 下才有效果。

14. ie 下 overflow:hidden 对其下的绝对层 position:absolute 或者相对层 position:relative 无效。解决方案: 给 overflow:hidden 加 position:relative

或者 position:absolute。另，ie6 支持 overflow-x 或者 overflow-y 的特性，ie7、ff 不支持。

15. ie6 下严重的 bug，float 元素如没定义宽度，内部如有 div 定义了 height 或 zoom:1，这个 div 就会占满一整行，即使你给了宽度。float 元素如果作为布局用或复杂的容器，都要给个宽度的。16. ie6 下的 bug，绝对定位的 div 下包含相对定位的 div，如果给内层相对定位的 div 高度 height 具体值，内层相对层将具有 100% 的 width 值，外层绝对层将被撑大。解决方案给内层相对层 float 属性。17. ie6 下的 bug，<head></head> 内有 <base target="_blank" /> 的情况下，position:relative 层下的 float 层内文字无法选中。

18. 终于来了个 ff 的缺点。width:100% 这个东西在 ie 里用很方便，会向上逐层搜索 width 值，忽视浮动层的影响，ff 下搜索至浮动层结束，如此，只能给中间的所有浮动层加 width:100% 才行，累啊。opera 这点倒学乖了跟了 ie

二、你如何理解 HTML 结构的语意化。

a) 去掉或样式丢失的时候能让页面呈现清晰的结构：

html 本身是没有表现的，我们看到例如 <h1> 是粗体，字体大小 2em，加粗； 是加粗的，不要认为这是 html 的表现，这些其实 html 默认的 css 样式

b) 屏幕阅读器（如果访客有视障）会完全根据你的标记来“读”你的网页。

- c) PDA、手机等设备可能无法像普通电脑的浏览器一样来渲染网页（通常是因为这些设备对 CSS 的支持较弱）。
- d) 便于团队开发和维护

三、null, undefined 的区别？

undefined 表示变量声明但未初始化的值，null 表示准备用来保存对象，还没有真正保存对象的值。从逻辑角度看，null 表示一个空对象

四、Node.js 的适用场景？

高并发、聊天、实时消息推送

五、创建对象的方式都有哪些？

- 1、直接量
- 2、用 new 关键字创建
- 3、构造函数

六、谈谈 this 对象

this 是 js 的一个关键字，随着函数适应场合不同，this 的值会发生变化。但是有一个总原则，this 是指调用函数的那个对象。

this 一般情况 是指全局对象 Global。作为方法调用时，这个 this 指的是调用方法的那个对象。

七、如何解决跨域问题？

JSONP:

原理是：动态插入 script 标签，通过 script 标签引入一个 js 文件，这个 js 文件载入成功后会执行我们在 url 参数中指定的函数，并且会把我们需要的 json 数据作为参数传入。

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，为了实现跨域请求，可以通过 script 标签实现跨域请求，然后在服务端输出 JSON 数据并执行回调函数，从而解决了跨域的数据请求。

优点是兼容性好，简单易用，支持浏览器与服务器双向通信。缺点是只支持 GET 请求。

CORS

服务器端对于 CORS 的支持，主要就是通过设置 Access-Control-Allow-Origin 来进行的。如果浏览器检测到相应的设置，就可以允许 Ajax 进行跨域的访问。

使用 window.name 来进行跨域

window 对象有个 name 属性，该属性有个特征：即在一个窗口(window)的生命周期内,窗口载入的所有的页面都是共享一个 window.name 的，每个页面对 window.name 都有读写的权限，window.name 是持久存在一个窗口载入过的所有页面中的

使用 HTML5 中新引进的 window.postMessage 方法来跨域传送数

据

还有 flash、在服务器上设置代理页面等跨域方式。个人认为 window.name 的方法既不复杂，也能兼容到几乎所有浏览器，这真是极好的一种跨域方法。

八、XML 和 JSON 的区别？

JSON 相对于 XML 来讲，数据的体积小，传递的速度更快些。

JSON 与 JavaScript 的交互更加方便，更容易解析处理，更好的数据交互。

JSON 对数据的描述性比 XML 较差。

JSON 的速度要远远快于 XML。

九、渐进增强和优雅降级

渐进增强：针对低版本浏览器进行构建页面，保证最基本的功能，然后再针对高级浏览器进行效果、交互等改进和追加功能达到更好的用户体验。

优雅降级：一开始就构建完整的功能，然后再针对低版本浏览器进行兼容。

十、谈谈性能优化问题

代码层面：避免使用 css 表达式，避免使用高级选择器，通配选择器。

减少 DOM 操作次数，优化 javascript 性能

缓存 DOM 节点查找的结果

避免使用 CSS Expression (CSS 表达式)

避免全局查询

避免图片和 iFrame 等的空 Src。空 Src 会重新加载当前页面，影响速度和效率

缓存利用:缓存 Ajax,使用 CDN,使用外部 js 和 css 文件以便缓存,添加 Expires 头,服务端配置 Etag,减少 DNS 查找等

请求数量:合并样式和脚本,使用 css 图片精灵,初始首屏之外的图片资源按需加载,静态资源延迟加载。

请求带宽:压缩文件,开启 GZIP

十一、清除浮动有哪些方式? 比较好的方式是哪一种?

- (1) 父级 div 定义 height。
- (2) 结尾处加空 div 标签 clear:both。
- (3) 父级 div 定义伪类:after 和 zoom。
- (4) 父级 div 定义 overflow:hidden。
- (5) 父级 div 定义 overflow:auto。
- (6) 父级 div 也浮动,需要定义宽度。
- (7) 父级 div 定义 display:table。
- (8) 结尾处加 br 标签 clear:both

十二、前端页面有哪三层结构,分别是什么? 作用是什么?

答案：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。负责标记页面内容，提供页面内容的基本框架；

表示层：

由 CSS 负责创建。负责为页面内容提供样式布局，渲染页面元素；

行为层：

由 Javascript 创建。负责对页面元素进行操作，DOM、BOM 以及数据交互。

DOM 怎样添加、移除、移动、复制、创建和查找节点

Javascript 代码：

// 创建新节点

createDocumentFragment() //创建一个 DOM 片段

createElement() //创建一个具体的元素

createTextNode() //创建一个文本节点

// 添加、移除、替换、插入

appendChild()

removeChild()

replaceChild()

insertBefore() //在已有的子节点前插入一个新的子节点

// 查找

getElementsByTagName() //通过标签名称

getElementsByTagName() //通过元素的 Name 属性的值(IE 容错能力较强, 会得到一个数组, 其中包括 id 等于 name 值的)

getElementById() //通过元素 Id, 唯一性

你知道 ng2 么 简单说一下

答案:

- 1.相比 Angular1.x, Angular2 的改动很大, 几乎算是一个全新的框架。
- 2.基于 TypeScript (可以使用 TypeScript 进行开发), 在大型项目团队协作时, 强语言类型更有利。
- 3.组件化, 提升开发和维护的效率。
- 4.还有 module 支持动态加载, new router, promise 的原生支持等等。
- 5.迎合未来标准, 吸纳其他框架的优点, 值得期待, 不过同时要学习的东西也更多了 (ES next、TS、Rx 等) 。

项目开发经历了哪几个阶段

需求分析及变更管理

项目模型及业务流程分析

系统分析及建模设计

界面设计及代码开发

系统测试，部署和文档编写

维护

浏览器和 WEB 服务器是用什么协议通讯的？

应用层使用的是 HTTP 协议，传输和路由使用的是 TCP/IP

ajax 的步骤与原理

1,创建一个 XMLHttpRequest 对象.

2,调用 open 方法 (准备发起请求).

open("提交方式" , " URL 地址" , " 是否同步(false/true)").

3,如果是提交方式是 post,就需要设置请求

头.SetrequestHeard("content-Type" ,application/X-www;1);

4,设置回调函数 XMLHttpRequest.onreadystatechange=XX;

5,在服务器端 response.setContentType("");

response.setHeard("");

6,通过判断 XMLHttpRequest.readystatc==4 是否会应.

XMLHttpRequest.status==200 判断状态码.

7,用 XMLHttpRequest.responseText 接受文本内容

或者用 XMLHttpRequest.XmlResponse 接受 Xml 内容.

5. 请分别写出 IE 与 W3C 的 CSS 的盒子模型

IE 的 content 部分包含了 border 和 padding

W3C 标准盒模型： 内容(content)、填充(padding)、边界
(margin)、 边框(border)

7. 分别列出至少四中“可继承样式”与“不可继承样式”

可继承样式：font-size, font-family, color, text-indent

不可继承样式：width, height, border, padding, margin

12. 请简述一下 js 事件的三个阶段

捕捉阶段 (capture phase)：从最上层元素，直到最下层（你点击的那个 target)元素。路过的所有节点都可以捕捉到该事件。

命中阶段 (target phase)：如果该事件成功到达了 target 元素，它会进行事件处理。

冒泡阶段 (bubbling phase)：事件从最下层向上传递，依次触发父元素的该事件处理函数。

4. 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，不安全，非常耗性能（2 次，一次解析成 js 语句，一次执

行)。

由 JSON 字符串转换为 JSON 对象的时候可以用 `eval, var obj = eval('(' + str + ')');`

如果现在页面上出现了卡顿现象，例如鼠标移动不流畅，页面滚动不流畅，你会如何进行 debug？

答：用 chrome 浏览器查看 network 和 timeline 来查看具体导致页面卡顿的原因。

HTTP 是无状态协议，那么网站应该如何记住用户的登录状态？尽量详细回答整个过程。

答：可以使用 session 来记录用户登录状态，session 是服务器保存数据，当用户登录时，服务器会生成一个 sessionid，这个 sessionid 下面就存储着用户的所有信息，这个时候也会传递这个 sessionid 给浏览器，这个 sessionid 存储于浏览器 cookie 中，之后用户请求的时候，就会把这个 sessionid 与服务器匹配，找到相对应的信息。

【如何进行自我介绍】

自我介绍这个问题，不用多说了，面试必定会问！如果想要在自我介绍的时候就能够打动面试官，吸引面试官对我们的兴趣，那么像我们这种接受过培训的程序员自我介绍当然不能和应届生或者其他非技术岗位一样。如何进行自我介绍比较好？有没有什么套路呢？当然有。

- 1、我叫 xxx,来自于吉林敦化，从网上看得到公司招聘的这个职位，觉得非常适合自己的发展。所以来这里争取下这份工作。
- 2、接下来可以讲解自己的核心竞争力（最强的技能、最深入研究的知识领域、个性中最积极的部分）和闪光点（做过的最成功的事，主要的成就）。核心就是要体现自己胜任这份工作（介绍时候要自然有底气，合情合理，面试前要充分准备好）。
- 3、先介绍到这里，面试官看看有什么问题，我可以再补充。

4、最后要说谢谢，任何 HR 都喜欢有礼貌的人！