

400 道求职常见面试 (笔试) 真题与解析

第 1 部分：HTML 基础篇

1. 面试题

Q: 面试官问

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？

A: 应聘者答

IE:trident 内核

Firefox: gecko 内核

Safari:webkit 内核

Opera:以前是 presto 内核，Opera 现已改用 GoogleChrome 的 Blink 内核

Chrome:Blink(基于 webkit, Google 与 OperaSoftware 共同开发)

2. 面试题

Q: 面试官问

每个 HTML 文件里开头都有个很重要的东西，Doctype，知道这是干什么的吗？

A: 应聘者答

<!DOCTYPE>声明位于文档中的最前面的位置，处于<html>标签之前。此标签可告知浏览器文档使用哪种 HTML 或 XHTML 规范。（重点：告诉浏览器按照何种规范解析页面）

3. 面试题

Q: 面试官问

Quirks 模式是什么？它和 Standards 模式有什么区别

A:应聘者答

从 IE6 开始，引入了 Standards 模式，标准模式中，浏览器尝试给符合标准的文档在规范上的正确处理达到在指定浏览器中的程度。

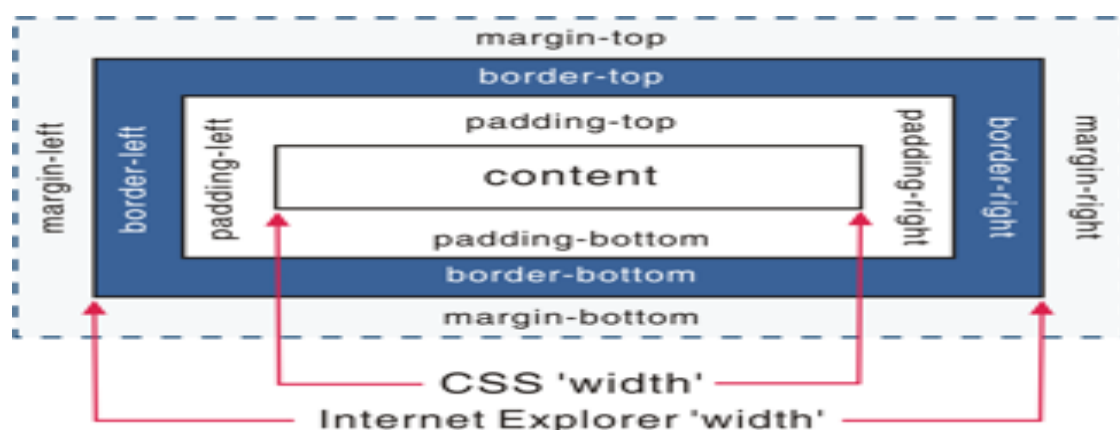
在 IE6 之前 CSS 还不够成熟，所以 IE5 等之前的浏览器对 CSS 的支持很差，IE6 将对 CSS 提供更好的支持，然而这时的问题就来了，因为有很多页面是基于旧的布局方式写的，而如果 IE6 支持 CSS 则将令这些页面显示不正常，如何在即保证不破坏现有页面，又提供新的渲染机制呢？

在写程序时我们也会经常遇到这样的问题，如何保证原来的接口不变，又提供更强大的功能，尤其是新功能不兼容旧功能时。遇到这种问题时的一个常见做法是增加参数和分支，即当某个参数为真时，我们就使用新功能，而如果这个参数不为真时，就使用旧功能，这样就能不破坏原有的程序，又提供新功能。IE6 也是类似这样做的，它将 DTD 当成了这个“参数”，因为以前的页面大家都不会去写 DTD，所以 IE6 就假定如果写了 DTD，就意味着这个页面将采用对 CSS 支持更好的布局，而如果没有，则采用兼容之前的布局方式。这就是 Quirks 模式（怪癖模式，诡异模式，怪异模式）。

区别：

总体会有布局、样式解析和脚本执行三个方面的区别。

盒模型：在 W3C 标准中，如果设置一个元素的宽度和高度，指的是元素内容的宽度和高度，而在 Quirks 模式下，IE 的宽度和高度还包含了 padding 和 border。



设置行内元素的高宽：在 Standards 模式下，给等行内元素设置 width 和 height 都不会生效，而在 quirks 模式下，则会生效。

设置百分比的高度：在 standards 模式下，一个元素的高度是由其包含的内容来决定的，如果父元素没有设置百分比的高度，子元素设置一个百分比的高度是无效的。用 margin:0 auto 设置水平居中：使用 margin:0 auto 在 standards 模式下可以使元素水平居中，但在 quirks 模式下却会失效。

（还有很多，答出什么不重要，关键是看他答出的这些是不是自己经验遇到的，还是说都是看文章看的，甚至完全不知道。）

4. 面试题

Q: 面试官问

div+css 的布局较 table 布局有什么优点？

A: 应聘者答

改版的时候更方便只要改 css 文件。

页面加载速度更快、结构化清晰、页面显示简洁。

表现与结构相分离。

易于优化（seo）搜索引擎更友好，排名更容易靠前。

5. 面试题

Q: 面试官问

a: img 的 alt 与 title 有何异同？ b: strong 与 em 的异同？

A: 应聘者答

a:

alt(alttext): 为不能显示图像、窗体或 applets 的用户代理（UA），alt 属性用来指定替换文字。替换文字的语言由 lang 属性指定。（在 IE 浏览器下会在没有 title 时把 alt 当成 tooltip 显示）

title(tooltip): 该属性为设置该属性的元素提供建议性的信息。

b:

strong: 粗体强调标签，强调，表示内容的重要性

em: 斜体强调标签，更强烈强调，表示内容的强调点

6. 面试题

Q: 面试官问

你能描述一下渐进增强和优雅降级之间的不同吗？

A: 应聘者答

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

优雅降级 gracefuldegradation: 一开始就构建完整的功能, 然后再针对低版本浏览器进行兼容。

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

“优雅降级”观点

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

在这种设计范例下, 旧版的浏览器被认为仅能提供“简陋却无妨(poor, but passable)”的浏览体验。你可以做一些小的调整来适应某个特定的浏览器。但由于它们并非我们所关注的焦点, 因此除了修复较大的错误之外, 其它的差异将被直接忽略。

“渐进增强”观点

“渐进增强”观点则认为应关注于内容本身。

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

那么问题了。现在产品经理看到 IE6,7,8 网页效果相对高版本现代浏览器少了很多圆角, 阴影(CSS3), 要求兼容(使用图片背景, 放弃 CSS3), 你会如何说服他?

7. 面试题

Q: 面试官问

为什么利用多个域名来存储网站资源会更有效?

A: 应聘者答

- CDN 缓存更方便
- 突破浏览器并发限制
- 节约 cookie 带宽
- 节约主域名的连接数, 优化页面响应速度
- 防止不必要的安全问题

8. 面试题

Q: 面试官问

请谈一下你对网页标准和标准制定机构重要性的理解。

A: 应聘者答

（无标准答案）网页标准和标准制定机构都是为了让 web 发展的更‘健康’，开发者遵循统一的标准，降低开发难度，开发成本，SEO 也会更好做，也不会因为滥用代码导致各种 BUG、安全问题，最终提高网站易用性。

9. 面试题

Q: 面试官问

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

A: 应聘者答

sessionStorage 用于本地存储一个会话（session）中的数据，这些数据只有在同一个会话中的页面才能访问并且当会话结束后数据也随之销毁。因此 sessionStorage 不是一种持久化的本地存储，仅仅是会话级别的存储。而 localStorage 用于持久化的本地存储，除非主动删除数据，否则数据是永远不会过期的。

webstorage 和 cookie 的区别

WebStorage 的概念和 cookie 相似，区别是它是为了更大容量存储设计的。Cookie 的大小是受限的，并且每次你请求一个新的页面的时候 Cookie 都会被发送过去，这样无形中浪费了带宽，另外 cookie 还需要指定作用域，不可以跨域调用。

除此之外，WebStorage 拥有 setItem, getItem, removeItem, clear 等方法，不像 cookie 需要前端开发者自己封装 setCookie, getCookie。但是 Cookie 也是不可或缺的：Cookie 的作用是与服务器进行交互，作为 HTTP 规范的一部分而存在，而 WebStorage 仅仅是为了在本地“存储”数据而生。

10. 面试题

Q: 面试官问

简述一下 src 与 href 的区别。

A:应聘者答

src 用于替换当前元素，href 用于在当前文档和引用资源之间确立联系。

src 是 source 的缩写，指向外部资源的位置，指向的内容将会嵌入到文档中当前标签所在位置；在请求 src 资源时会将其指向的资源下载并应用到文档内，例如 js 脚本，img 图片和 frame 等元素。

```
<scriptsrc="js.js"></script>
```

当浏览器解析到该元素时，会暂停其他资源的下载和处理，直到将该资源加载、编译、执行完毕，图片和框架等元素也如此，类似于将所指向资源嵌入当前标签内。这也是为什么将 js 脚本放在底部而不是头部。

href 是 HypertextReference 的缩写，指向网络资源所在位置，建立和当前元素（锚点）或当前文档（链接）之间的链接，如果我们在文档中添加

```
<linkhref="common.css"rel="stylesheet"/>
```

那么浏览器会识别该文档为 css 文件，就会并行下载资源并且不会停止对当前文档的处理。这也是为什么建议使用 link 方式来加载 css，而不是使用 @import 方式。

11. 面试题

Q:面试官问

知道的网页制作会用到的图片格式有哪些？

A:应聘者答

png-8, png-24, jpeg, gif, svg。

但是上面的那些都不是面试官想要的最后答案。面试官希望听到是 Webp,Apng。（是否有关新技术，新鲜事物）

科普一下 Webp: WebP 格式，谷歌（google）开发的一种旨在加快图片加载速度的图片格式。图片压缩体积大约只有 JPEG 的 2/3，并能节省大量的服务器带宽资源和数据空间。FacebookEbay 等知名网站已经开始测试并使用 WebP 格式。

在质量相同的情况下，WebP 格式图像的体积要比 JPEG 格式图像小 40%。

Apng: 全称是“AnimatedPortableNetworkGraphics”，是 PNG 的位图动画扩展，可以实现 png 格式的动态图片效果。04 年诞生，但一直得不到各大浏览器厂商的支持，直到日前得到 iOSsafari8 的支持，有望代替 GIF 成为下一代动态图标准。

12. 面试题

Q: 面试官问

知道什么是微格式吗？谈谈理解。在前端构建中应该考虑微格式吗？

A: 应聘者答

微格式（Microformats）是一种让机器可读的语义化 XHTML 词汇的集合，是结构化数据的开放标准。是为特殊应用而制定的特殊格式。

优点：将智能数据添加到网页上，让网站内容在搜索引擎结果界面可以显示额外的提示。（应用范例：豆瓣，有兴趣自行 google）

13. 面试题

Q: 面试官问

在 css/js 代码上线之后开发人员经常会优化性能，从用户刷新网页开始，一次 js 请求一般情况下有哪些地方会有缓存处理？

A: 应聘者答

dns 缓存，cdn 缓存，浏览器缓存，服务器缓存。

14. 面试题

Q: 面试官问

一个页面上有大量的图片（大型电商网站），加载很慢，你有哪些方法优化这些图片的加载，给用户更好的体验。

A: 应聘者答

图片懒加载，在页面上的未可视区域可以添加一个滚动条事件，判断图片位置与浏览器顶端的距离与页面的距离，如果前者小于后者，优先加载。

如果为幻灯片、相册等，可以使用图片预加载技术，将当前展示图片的前一张和后一张优先下载。

如果图片为 css 图片，可以使用 CSSsprite, SVGsprite, Iconfont、Base64 等技术。

如果图片过大，可以使用特殊编码的图片，加载时会先加载一张压缩的特别厉害的缩略图，以提高用户体验。

如果图片展示区域小于图片的真实大小，则因在服务器端根据业务需要先行进行图片压缩，图片压缩后大小与展示一致。

15. 面试题

Q:面试官问

你如何理解 HTML 结构的语义化？

A:应聘者答

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 样式在起作用，所以去掉或样式丢失的时候能让页面呈现清晰的结构不是语义化的 HTML 结构的优点，但是浏览器都有默认样式，默认样式的目的也是为了更好的表达 html 的语义，可以说浏览器的默认样式和语义化的 HTML 结构是不可分割的。

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。

例如,如果你使用的含语义的标记,屏幕阅读器就会“逐个拼出”你的单词,而不是试着去对它完整发音。

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）

使用语义标记可以确保这些设备以一种有意义的方式来渲染网页.理想情况下,观看设备的任务是符合设备本身的条件来渲染网页。

语义标记为设备提供了所需的相关信息,就省去了你自己去考虑所有可能的显示情况（包括现有的或者将来新的设备）。例如,一部手机可以选择使一段标记了标题的文字以粗体显示.而掌上电脑可能会以比较大的字体来显示.无论哪种方式一旦你对文本标记为标题,您就可以确信读取设备将根据其自身的条件来合适地显示页面。

搜索引擎的爬虫也依赖于标记来确定上下文和各个关键字的权重

过去你可能还没有考虑搜索引擎的爬虫也是网站的“访客”,但现在它们他们实际上是极其宝贵的用户.没有他们的话,搜索引擎将无法索引你的网站,然后一般用户将很难过来访问。

你的页面是否对爬虫容易理解非常重要,因为爬虫很大程度上会忽略用于表现的标记,而只注重语义标记。

因此,如果页面文件的标题被标记,而不是,那么这个页面在搜索结果的位置可能会比较靠后.除了提升易用性外,语义标记有利于正确使用 CSS 和 JavaScript,因为其本身提供了许多“钩钩”来应用页面的样式与行为。

SEO 主要还是靠你网站的内容和外部链接的。

便于团队开发和维护

W3C 给我们定了一个很好的标准，在团队中大家都遵循这个标准，可以减少很多差异化的东西，方便开发和维护，提高开发效率，甚至实现模块化开发。

16. 面试题

Q: 面试官问

谈谈以前端角度出发做好 SEO 需要考虑什么？

A: 应聘者答

了解搜索引擎如何抓取网页和如何索引网页

你需要知道一些搜索引擎的基本工作原理，各个搜索引擎之间的区别，搜索机器人（SErobot 或叫 webcrawler）如何进行工作，搜索引擎如何对搜索结果进行排序等等。

Meta 标签优化

主要包括主题（Title），网站描述（Description），和关键词（Keywords）。还有一些其它的隐藏文字比如 Author（作者），Category（目录），Language（编码语种）等。

如何选取关键词并在网页中放置关键词

搜索就得用关键词。关键词分析和选择是 SEO 最重要的工作之一。首先要给网站确定主关键词（一般在 5 个上下），然后针对这些关键词进行优化，包括关键词密度（Density），相关度（Relavancy），突出性（Prominency）等等。

了解主要的搜索引擎

虽然搜索引擎有很多，但是对网站流量起决定作用的就那么几个。比如英文的主要有 Google, Yahoo, Bing 等；中文的有百度，搜狗，有道等。不同的搜索引擎对页面的抓取和索引、排序的规则都不一样。还要了解各搜索门户和搜索引擎之间的关系，比如 AOL 网页搜索用的是 Google 的搜索技术，MSN 用的是 Bing 的技术。

主要的互联网目录

OpenDirectory 自身不是搜索引擎，而是一个大型的网站目录，他和搜索引擎的主要区别是网站内容的收集方式不同。目录是人工编辑的，主要收录网站主页；搜索引擎是自动收集的，除了主页外还抓取大量的内容页面。

按点击付费的搜索引擎

搜索引擎也需要生存，随着互联网商务的越来越成熟，收费的搜索引擎也开始大行其道。最典型的有 Overture 和百度，当然也包括 Google 的广告项目 GoogleAdwords。越来越多的人通过搜索引擎的点击广告来定位商业网站，这里面也大有优化和排名的学问，你得学会用最少的广告投入获得最多的点击。

搜索引擎登录

网站做完了以后，别躺在那里等着客人从天而降。要让别人找到你，最简单的办法就是将网站提交（submit）到搜索引擎。如果你的网站是商业网站，主要的搜索引擎和目录都会要求你付费来获得收录（比如 Yahoo 要 299 美元），但是好消息是（至少到目前为止）最大的搜索引擎 Google 目前还是免费，而且它主宰着 60% 以上的搜索市场。

链接交换和链接广泛度（LinkPopularity）

网页内容都是以超文本（Hypertext）的方式来互相链接的，网站之间也是如此。除了搜索引擎以外，人们也每天通过不同网站之间的链接来 Surfing（“冲浪”）。其它网站到你的网站的链接越多，你也会获得更多的访问量。更重要的是，你的网站的外部链接数越多，会被搜索引擎认为它的重要性越大，从而给你更高的排名。

第 2 部分：CSS 样式表篇

17. 面试题

Q: 面试官问

有哪项方式可以对一个 DOM 设置它的 CSS 样式？

A: 应聘者答

外部样式表，引入一个外部 css 文件

内部样式表，将 css 代码放在<head>标签内部

内联样式，将 css 样式直接定义在 HTML 元素内部

18. 面试题

Q: 面试官问

CSS 都有哪些选择器？

A: 应聘者答

派生选择器（用 HTML 标签申明）

id 选择器（用 DOM 的 ID 申明）

类选择器（用一个样式类名申明）

属性选择器（用 DOM 的属性申明，属于 CSS2，IE6 不支持，不常用，不知道就算了）

除了前 3 种基本选择器，还有一些扩展选择器，包括

后代选择器（利用空格间隔，比如 div.a{}

群组选择器（利用逗号间隔，比如 p,div,#a{}

那么问题来了，CSS 选择器的优先级是怎样定义的？

基本原则：

一般而言，选择器越特殊，它的优先级越高。也就是选择器指向的越准确，它的优先级就越高。

复杂的计算方法：

用 1 表示派生选择器的优先级

用 10 表示类选择器的优先级

用 100 标示 ID 选择器的优先级

div.test1 spanvar 优先级 1+10+10+1
span#xxx.songsli 优先级 1+100+10+1
#xxxli 优先级 100+1

那么问题来了，看下列代码，<p>标签内的文字是什么颜色的？。

```
<style>
classA{color:blue;}
classB{color:red;}
</style>
<body>
<pclass='classBclassA'>123</p>
</body>
```

red。与样式定义在文件中的先后顺序有关，即是后面的覆盖前面的，与在<pclass='classBclassA'>中的先后关系无关。

19. 面试题

Q:面试官问

CSS 中可以通过哪些属性定义，使得一个 DOM 元素不显示在浏览器可视范围内？

A:应聘者答

最基本的：

设置 display 属性为 none，或者设置 visibility 属性为 hidden

技巧性：

设置宽高为 0，设置透明度为 0，设置 z-index 位置在-1000

20. 面试题

Q:面试官问

超链接访问过后 hover 样式就不出现的问题是什么？如何解决？

A:应聘者答

被点击访问过的超链接样式不在具有 hover 和 active 了,解决方法是改变 CSS 属性的排列顺序:L-V-H-A（link,visited,hover,active）

21. 面试题

Q: 面试官问

什么是 CssHack? ie6,7,8 的 hack 分别是什么?

A: 应聘者答

针对不同的浏览器写不同的 CSScode 的过程, 就是 CSShack。

示例如下:

```
1#test{
2width:300px;
3height:300px;
4
5background-color:blue;/*firefox*/
6background-color:red\9;/*allie*/
7background-color:yellow\0;/*ie8*/
8+background-color:pink;/*ie7*/
9_background-color:orange;/*ie6*/}
10:root#test{background-color:purple\9;/*ie9*/
11@mediaalland(min-width:0px){#test{background-color:black\0;}}/*opera*/
12@mediascreenand(-webkit-min-device-pixel-ratio:0){#test{background-
color:gray;}}/*chromeandsafari*/
```

22. 面试题

Q: 面试官问

请用 Css 写一个简单的幻灯片效果页面

A: 应聘者答

知道是要用 css3。使用 animation 动画实现一个简单的幻灯片效果。

```
1/**HTML**/
2div.ani
3
4/**css**/
5.ani{
6width:480px;
7height:320px;
8margin:50pxauto;
9overflow:hidden;
```

```

10box-shadow:005pxrgba(0,0,0,1);
11background-size:cover;
12background-position:center;
13-webkit-animation-name:"loops";
14-webkit-animation-duration:20s;
15-webkit-animation-iteration-count:infinite;
16}
17@-webkit-keyframes"loops"{
180% {
19background:url(http://d.hiphotos.baidu.com/image/w%3D400/sign=c01e6adca964034f
0fcdc3069fc27980/e824b899a9014c08e5e38ca4087b02087af4f4d3.jpg)no-repeat;
20}
2125% {
22background:url(http://b.hiphotos.baidu.com/image/w%3D400/sign=edee1572e9f81a4c
2632edc9e72b6029/30adcbef76094b364d72bceba1cc7cd98c109dd0.jpg)no-repeat;
23}
2450% {
25background:url(http://b.hiphotos.baidu.com/image/w%3D400/sign=937dace2552c11df
ded1be2353266255/d8f9d72a6059252d258e7605369b033b5bb5b912.jpg)no-repeat;
26}
2775% {
28background:url(http://g.hiphotos.baidu.com/image/w%3D400/sign=7d37500b8544ebf8
6d71653fe9f9d736/0df431adcbef76095d61f0972cdda3cc7cd99e4b.jpg)no-repeat;
29}
30100% {
31background:url(http://c.hiphotos.baidu.com/image/w%3D400/sign=cfb239ceb0fb43161
a1f7b7a10a54642/3b87e950352ac65ce2e73f76f9f2b21192138ad1.jpg)no-repeat;
32}
33}

```

23. 面试题

Q: 面试官问

行内元素和块级元素的具体区别是什么？行内元素的 padding 和 margin 可设置吗？

A: 应聘者答

块级元素(block)特性:

总是独占一行，表现为另起一行开始，而且其后的元素也必须另起一行显示;

宽度(width)、高度(height)、内边距(padding)和外边距(margin)都可控制;

内联元素(inline)特性:

和相邻的内联元素在同一行;

宽度(width)、高度(height)、内边距的 top/bottom(padding-top/padding-bottom)和外边距的 top/bottom(margin-top/margin-bottom)都不可改变(也就是 padding 和 margin 的 left 和 right 是可以设置的),就是里面文字或图片的大小。

那么问题来了,浏览器还有默认的天生 inline-block 元素(拥有内在尺寸,可设置高宽,但不会自动换行),有哪些?

A:应聘者答

<input>、、<button>、<textarea>、<label>。

24. 面试题

Q:面试官问

什么是外边距重叠?重叠的结果是什么?

A:应聘者答

外边距重叠就是 margin-collapse。

在 CSS 当中,相邻的两个盒子(可能是兄弟关系也可能是祖先关系)的外边距可以结合成一个单独的外边距。这种合并外边距的方式被称为折叠,并且因而所结合成的外边距称为折叠外边距。

折叠结果遵循下列计算规则:

两个相邻的外边距都是正数时,折叠结果是它们两者之间较大的值。

两个相邻的外边距都是负数时,折叠结果是两者绝对值的较大值。

两个外边距一正一负时,折叠结果是两者的相加的和。

25. 面试题

Q:面试官问

rgba()和 opacity 的透明效果有什么不同?

A:应聘者答

rgba()和 opacity 都能实现透明效果,但最大的不同是 opacity 作用于元素,以及元素内的所有内容的透明度,

而 `rgba()` 只作用于元素的颜色或其背景色。(设置 `rgba` 透明的元素的子元素不会继承透明效果!)

26. 面试题

Q: 面试官问

css 中可以让文字在垂直和水平方向上重叠的两个属性是什么?

A: 应聘者答

垂直方向: `line-height`

水平方向: `letter-spacing`

那么问题来了, 关于 `letter-spacing` 的妙用知道有哪些么?

答案: 可以用于消除 `inline-block` 元素间的换行符空格间隙问题。

27. 面试题

Q: 面试官问

如何垂直居中一个浮动元素?

A: 应聘者答

```
1//方法一: 已知元素的高宽
2
3#div1{
4background-color:#6699FF;
5width:200px;
6height:200px;
7
8position:absolute;//父元素需要相对定位
9top:50%;
10left:50%;
11margin-top:-100px;//二分之一的 height, width
12margin-left:-100px;
13}
14
15//方法二:未知元素的高宽
16
```



```
17#div1{
18width:200px;
19height:200px;
20background-color:#6699FF;
21
22margin:auto;
23position:absolute;//父元素需要相对定位
24left:0;
25top:0;
26right:0;
27bottom:0;
28}
那么问题来了，如何垂直居中一个<img>?（用更简便的方法。）
1#container//<img>的容器设置如下
2{
3display:table-cell;
4text-align:center;
5vertical-align:middle;
6}
```

28. 面试题

Q: 面试官问

px 和 em 的区别。

A: 应聘者答

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

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

29. 面试题

Q: 面试官问

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

A:应聘者答

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

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

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

30. 面试题

Q:面试官问

Sass、LESS 是什么？大家为什么要使用他们？

A:应聘者答

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

例如 Less 是一种动态样式语言.将 CSS 赋予了动态语言的特性，如变量，继承，运算，函数.LESS 既可以在客户端上运行(支持 IE6+,Webkit,Firefox)，也可一在服务端运行(借助 Node.js)。

为什么要使用它们？

结构清晰，便于扩展。

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

可以轻松实现多重继承。

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

31. 面试题

Q:面试官问

`display:none` 与 `visibility:hidden` 的区别是什么？

A:应聘者答

`display`:隐藏对应的元素但不挤占该元素原来的空间。

`visibility`:隐藏对应的元素并且挤占该元素原来的空间。

即是，使用 `CSSdisplay:none` 属性后，HTML 元素（对象）的宽度、高度等各种属性值都将“丢失”；而使用 `visibility:hidden` 属性后，HTML 元素（对象）仅仅是在视觉上看不见（完全透明），而它所占据的空间位置仍然存在。

32. 面试题

Q:面试官问

知道 `css` 有个 `content` 属性吗？有什么作用？有什么应用？

A:应聘者答

知道。`css` 的 `content` 属性专门应用在 `before/after` 伪元素上，用于来插入生成内容。最常见的应用是利用伪类清除浮动。

1//一种常见利用伪类清除浮动的代码

2.clearfix:after{

3content:".";//这里利用到了 `content` 属性

4display:block;

5height:0;

6visibility:hidden;

7clear:both;}

8

9.clearfix{

10*zoom:1;

11}

`after` 伪元素通过 `content` 在元素的后面生成了内容为一个点的块级元素，再利用 `clear:both` 清除浮动。

那么问题继续还有，知道 `css` 计数器（序列数字字符自动递增）吗？如何通过 `csscontent` 属性实现 `css` 计数器？

答案：`css` 计数器是通过设置 `counter-reset`、`counter-increment` 两个属性、及 `counter()/counters()` 一个方法配合 `after/before` 伪类实现。

第 3 部分：HTML+CSS 篇

33. 面试题

Q:面试官问

对 WEB 标准以及 W3C 的理解与认识

A:应聘者答

标签闭合、标签小写、不乱嵌套、提高搜索机器人搜索几率、使用外链 css 和 js 脚本、结构行为表现的分离、文件下载与页面速度更快、内容能被更多的用户所访问、内容能被更广泛的设备所访问、更少的代码和组件，容易维护、改版方便，不需要变动页面内容、提供打印版本而不需要复制内容、提高网站易用性。

34. 面试题

Q:面试官问

xhtml 和 html 有什么区别

A:应聘者答

HTML 是一种基本的 WEB 网页设计语言，XHTML 是一个基于 XML 的置标语言最主要的不同：

XHTML 元素必须被正确地嵌套。

XHTML 元素必须被关闭。

标签名必须用小写字母。

XHTML 文档必须拥有根元素。

35. 面试题

Q:面试官问

Doctype?严格模式与混杂模式-如何触发这两种模式，区分它们有何意义？

A:应聘者答

用于声明文档使用那种规范（html/Xhtml）一般为严格过度基于框架的 html 文档加入 XML 声明可触发，解析方式更改为 IE5.5 拥有 IE5.5 的 bug

36. 面试题

Q:面试官问

行内元素有哪些？块级元素有哪些？CSS 的盒模型？

A:应聘者答

块级元素: div p h1 h2 h3 h4 form ul

行内元素: abbr span input select

Css 盒模型: 内容, border, margin, padding

37. 面试题

Q:面试官问

CSS 引入的方式有哪些? link 和 @import 的区别是?

A:应聘者答

内联内嵌外链导入

区别: 同时加载

前者无兼容性, 后者 CSS2.1 以下浏览器不支持

Link 支持使用 javascript 改变样式, 后者不可

38. 面试题

Q:面试官问

CSS 选择符有哪些? 哪些属性可以继承? 优先级算法如何计算? 内联和 important 哪个优先级高?

A:应聘者答

标签选择符 类选择符 id 选择符

继承不如指定 Id > class > 标签选择

后者优先级高

39. 面试题

Q:面试官问

前端页面有哪三层构成，分别是什么？作用是什么？

A:应聘者答

结构层 Html 表示层 CSS 行为层 js

40. 面试题

Q:面试官问

css 的基本语句构成是？

A:应聘者答

选择器{属性 1:值 1;属性 2:值 2;.....}

41. 面试题

Q:面试官问

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？

A:应聘者答

Ie(Ie 内核)火狐 (Gecko) 谷歌 (webkit) opear(Presto)

42. 面试题

Q:面试官问

写出几种 IE6BUG 的解决方法

A:应聘者答

- 1.双边距 BUGfloat 引起的使用 display
- 2.3 像素问题使用 float 引起的使用 display:inline-3px
- 3.超链接 hover 点击后失效使用正确的书写顺序 linkvisitedhoveractive
- 4.1ez-index 问题给父级添加 position:relative
- 5.Png 透明使用 js 代码改
- 6.Min-height 最小高度! Important 解决'
- 7.select 在 ie6 下遮盖使用 iframe 嵌套
- 8.为什么没有办法定义 1px 左右的宽度容器 (IE6 默认的行高造成的, 使用 over:hidden,zoom:0.08line-height:1px)

43. 面试题

Q:面试官问

标签上 title 与 alt 属性的区别是什么?

A:应聘者答

Alt 当图片不显示是用文字代表。
Title 为该属性提供信息。

44. 面试题

Q:面试官问

描述 cssreset 的作用和用途。

A:应聘者答

Reset 重置浏览器的 css 默认属性浏览器的品种不同, 样式不同, 然后重置, 让他们统一。

45. 面试题

Q:面试官问

解释 csssprites，如何使用。

A:应聘者答

Css 精灵把一堆小的图片整合到一张大的图片上，减轻服务器对图片的请求数量

46. 面试题

Q:面试官问

浏览器标准模式和怪异模式之间的区别是什么？

A:应聘者答

盒子模型渲染模式的不同

使用 `window.top.document.compatMode` 可显示为什么模式

47. 面试题

Q:面试官问

你如何对网站的文件和资源进行优化？期待的解决方案包括：

A:应聘者答

文件合并

文件最小化/文件压缩

使用 CDN 托管

缓存的使用

48. 面试题

Q:面试官问

什么是语义化的 HTML?

A:应聘者答

直观的认识标签对于搜索引擎的抓取有好处

49. 面试题

Q:面试官问

清除浮动的几种方式，各自的优缺点

A:应聘者答

- 1.使用空标签清除浮动 `clear:both` (理论上能清楚任何标签,,, 增加无意义的标签)
- 2.使用 `overflow:auto` (空标签元素清除浮动而不得不增加无意义代码的弊端,,使用 `zoom:1` 用于兼容 IE)
- 3.是用 `afert` 伪元素清除浮动(用于非 IE 浏览器)

第 4 部分：初级 JavaScript 篇

50. 面试题

Q:面试官问

JavaScript 的数据类型都有什么?

A:应聘者答

基本数据类型: `String,boolean,Number,Undefined,Null`

引用数据类型: `Object(Array,Date,RegExp,Function)`

那么问题来了, 如何判断某变量是否为数组数据类型?

方法一.判断其是否具有“数组性质”,如 `slice()`方法。可自己给该变量定义 `slice` 方法,故有时会失效

方法二.`objinstanceofArray` 在某些 IE 版本中不正确

方法三.方法一二皆有漏洞，在 ECMAScript5 中定义了新方法 `Array.isArray()`，保证其兼容性，最好的方法如下：

```
1if(typeofArray.isArray=== "undefined")
2{
3Array.isArray=function(arg){
4returnObject.prototype.toString.call(arg)=== "[objectArray]"
5};
6}
```

51. 面试题

Q: 面试官问

已知 ID 的 Input 输入框，希望获取这个输入框的输入值，怎么做？（不使用第三方框架）

A: 应聘者答

```
1document.getElementById("ID").value
```

52. 面试题

Q: 面试官问

希望获取到页面中所有的 checkbox 怎么做？（不使用第三方框架）

A: 应聘者答

```
1var domList=document.getElementsByTagName('input')
2var checkBoxList=[];
3var len=domList.length;    //缓存到局部变量
4while(len--){    //使用 while 的效率会比 for 循环更高
5    if(domList[len].type=='checkbox'){
6        checkBoxList.push(domList[len]);
7    }
8}
```

53. 面试题

Q:面试官问

设置一个已知 ID 的 DIV 的 html 内容为 xxxx，字体颜色设置为黑色(不使用第三方框架)

A:应聘者答

```
1var dom=document.getElementById("ID");  
2dom.innerHTML="xxxx"  
3dom.style.color="#000"
```

54. 面试题

Q:面试官问

当一个 DOM 节点被点击时候，我们希望能够执行一个函数，应该怎么做？

A:应聘者答

直接在 DOM 里绑定事件：<div onclick="test()"></div>

在 JS 里通过 onclick 绑定：xxx.onclick=test

通过事件添加进行绑定：addEventListener(xxx,'click',test)

那么问题来了，Javascript 的事件流模型都有什么？

“事件冒泡”：事件开始由最具体的元素接受，然后逐级向上传播

“事件捕捉”：事件由最不具体的节点先接收，然后逐级向下，一直到最具体的

“DOM 事件流”：三个阶段：事件捕捉，目标阶段，事件冒泡

55. 面试题

Q:面试官问

什么是 Ajax 和 JSON，它们的优缺点。

A:应聘者答

Ajax 是异步 JavaScript 和 XML，用于在 Web 页面中实现异步数据交互。

优点：

可以使得页面不重载全部内容的情况下加载局部内容，降低数据传输量

避免用户不断刷新或者跳转页面，提高用户体验

缺点：

对搜索引擎不友好（

要实现 ajax 下的前后退功能成本较大

可能造成请求数的增加

跨域问题限制

JSON 是一种轻量级的数据交换格式，ECMA 的一个子集

优点：轻量级、易于人的阅读和编写，便于机器（JavaScript）解析，支持复合数据类型（数组、对象、字符串、数字）

56. 面试题

Q: 面试官问

看下列代码输出为何？解释原因。

A: 应聘者答

```
1 var a;  
2 alert(typeof a); // undefined  
3 alert(b); // 报错
```

解释：Undefined 是一个只有一个值的数据类型，这个值就是“undefined”，在使用 var 声明变量但并未对其赋值进行初始化时，这个变量的值就是 undefined。而 b 由于未声明将报错。注意未声明的变量和声明了未赋值的是不一样的。

57. 面试题

Q: 面试官问

9. 看下列代码，输出什么？解释原因。

A: 应聘者答

```
1 var a = null;  
2 alert(typeof a); // object
```

解释：null 是一个只有一个值的数据类型，这个值就是 null。表示一个空指针对象，所以用 typeof 检测会返回“object”。

58. 面试题

Q:面试官问

看下列代码,输出什么? 解释原因。

A:应聘者答

```
1var undefined;  
2undefined==null;//true  
31==true;//true  
42==true;//false  
50==false;//true  
60=="");//true  
7NaN==NaN;//false  
8[]==false;//true  
9[]==![];//true
```

undefined 与 null 相等, 但不恒等 (===)

一个是 number 一个是 string 时, 会尝试将 string 转换为 number

尝试将 boolean 转换为 number, 0 或 1

尝试将 Object 转换成 number 或 string, 取决于另外一个对比量的类型

所以, 对于 0、空字符串的判断, 建议使用“===”。“===”会先判断两边的值类型, 类型不匹配时为 false。

那么问题来了, 看下面的代码, 输出什么, foo 的类型为什么?

```
1var foo="11"+2-"1";  
2console.log(foo);  
3console.log(typeof foo);
```

执行完后 foo 的值为 111, foo 的类型为 Number。

```
1var foo="11"+2+"1";//体会加一个字符串'1'和减去一个字符串'1'的不同  
2console.log(foo);  
3console.log(typeof foo);
```

执行完后 foo 的值为 113, foo 的类型为 String。

59. 面试题

Q:面试官问

看代码给答案。

A:应聘者答

```
1var a=new Object();
```

```
2a.value=1;
3b=a;
4b.value=2;
5alert(a.value);
答案：2（考察引用数据类型细节）
```

60. 面试题

Q: 面试官问

```
varnumberArray=[3,6,2,4,1,5];（考察基础 API）
```

A: 应聘者答

```
1)实现对该数组的倒排，输出[5,1,4,2,6,3]
2)实现对该数组的降序排列，输出[6,5,4,3,2,1]
1varnumberArray=[3,6,2,4,1,5];
2
3numberArray.reverse();//5,1,4,2,6,3
4
5numberArray.sort(function(a,b){//6,5,4,3,2,1
6returnb-a;
7})
```

61. 面试题

Q: 面试官问

输出今天的日期，以 YYYY-MM-DD 的方式，比如今天是 2014 年 9 月 26 日，则输出 2014-09-26

A: 应聘者答

```
1vard=newDate();
2//获取年，getFullYear()返回 4 位的数字
3varyear=d.getFullYear();
4//获取月，月份比较特殊，0 是 1 月，11 是 12 月
5varmonth=d.getMonth()+1;
6//变成两位
7month=month<10?'0'+month:month;
```

```

8//获取日
9varday=d.getDate();
10day=day<10?'0'+day:day;
11alert(year+'-'+month+'-'+day);

```

62. 面试题

Q: 面试官问

将字符串”<tr><td>{\$id}</td><td>{\$name}</td></tr>”中的{\$id}替换成 10，{\$name}替换成 Tony（使用正则表达式）

A: 应聘者答

```

”<tr><td>{$id}</td><td>{$id}_{$name}</td></tr>”.replace(/\{$id\}/g,'10').replace(/\{$name\}/g,'Tony');

```

63. 面试题

Q: 面试官问

16.为了保证页面输出安全，我们经常需要对一些特殊的字符进行转义，请写一个函数 escapeHtml，将<,>,&,”进行转义

A: 应聘者答

```

1functionescapeHtml(str){
2returnstr.replace(/[<>”&]/g,function(match){
3switch(match){
4case“<”:
5return“&lt;”;
6case“>”:
7return“&gt;”;
8case“&”:
9return“&amp;”;
10case“\””:
11return“&quot;”;
12}
13});
14}

```

64. 面试题

Q: 面试官问

`foo=foo||bar`, 这行代码是什么意思? 为什么要这样写?

A: 应聘者答

`if(!foo)foo=bar;` //如果 `foo` 存在, 值不变, 否则把 `bar` 的值赋给 `foo`。

短路表达式: 作为“&&”和“||”操作符的操作数表达式, 这些表达式在进行求值时, 只要最终的结果已经可以确定是真或假, 求值过程便告终止, 这称之为短路求值。

65. 面试题

Q: 面试官问

看下列代码, 将会输出什么?(变量声明提升)

A: 应聘者答

```
1varfoo=1;
2function(){
3console.log(foo);
4varfoo=2;
5console.log(foo);
6}
```

答案: 输出 `undefined` 和 `2`。上面代码相当于:

```
1varfoo=1;
2function(){
3varfoo;
4console.log(foo);//undefined
5foo=2;
6console.log(foo);//2;
7}
```

函数声明与变量声明会被 JavaScript 引擎隐式地提升到当前作用域的顶部, 但是只提升名称不会提升赋值部分。

66. 面试题

Q: 面试官问

用 js 实现随机选取 10-100 之间的 10 个数字，存入一个数组，并排序。

A: 应聘者答

```
1varArray=[];
2functiongetRandom(istart,iend){
3varChoice=istart-iend+1;
4returnMath.floor(Math.random()*iChoice+istart;
5}
6for(vari=0;i<10;i++){
7iArray.push(getRandom(10,100));
8}
9iArray.sort();
```

67. 面试题

Q: 面试官问

把两个数组合并，并删除第二个元素。

A: 应聘者答

```
1vararray1=['a','b','c'];
2varbArray=['d','e','f'];
3varcArray=array1.concat(bArray);
4cArray.splice(1,1);
```

68. 面试题

Q: 面试官问

怎样添加、移除、移动、复制、创建和查找节点（原生 JS，实在基础，没细写每一步）

A:应聘者答

1) 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

2) 添加、移除、替换、插入

`appendChild()`//添加

`removeChild()`//移除

`replaceChild()`//替换

`insertBefore()`//插入

3) 查找

`getElementsByTagName()`//通过标签名称

`getElementsByName()`//通过元素的 Name 属性的值

`getElementById()`//通过元素 Id, 唯一性

69. 面试题

Q: 面试官问

有这样一个 URL: `http://item.taobao.com/item.htm?a=1&b=2&c=&d=xxx&e`, 请写一段 JS 程序提取 URL 中的各个 GET 参数(参数名和参数个数不确定), 将其按 key-value 形式返回到一个 json 结构中, 如 `{a:'1',b:'2',c:'',d:'xxx',e:undefined}`。

A:应聘者答

```
1functionserilizeUrl(url){
2varresult={ };
3url=url.split("?")[1];
4varmap=url.split("&");
5for(vari=0,len=map.length;i<len;i++){
6result[map[i].split("=")[0]]=map[i].split("=")[1];
7}
8returnresult;
9}
```

70. 面试题

Q: 面试官问

正则表达式构造函数 `var reg=new RegExp("xxx")` 与正则表达字面量 `var reg=//` 有什么不同？匹配邮箱的正则表达式？

A: 应聘者答

当使用 `RegExp()` 构造函数的时候，不仅需要转义引号（即“”表示”），并且还需要双反斜杠（即\\表示一个\）。使用正则表达字面量的效率更高。

邮箱的正则匹配：

```
1var regMail=/^([a-zA-Z0-9_-])+@([a-zA-Z0-9_-])+((.[a-zA-Z0-9_-]{2,3}){1,2})$/;
```

71. 面试题

Q: 面试官问

看下面代码，给出输出结果。

A: 应聘者答

```
1for(var i=1;i<=3;i++){
2setTimeout(function(){
3console.log(i);
4},0);
5};
```

答案：444。

原因：Javascript 事件处理器在线程空闲之前不会运行。那么问题来了，如何让上述代码输出 123？

```
1for(var i=1;i<=3;i++){
2setTimeout((function(a){//改成立即执行函数
3console.log(a);
4})(i),0);
5};
6
7//输出
82
93
```

72. 面试题

Q: 面试官问

写一个 function，清除字符串前后的空格。（兼容所有浏览器）

A: 应聘者答

使用自带接口 trim()，考虑兼容性：

```
1if(!String.prototype.trim){
2String.prototype.trim=function(){
3returnthis.replace(/^\s+/, "").replace(/\s+$/, "");
4}
5}
6
7//testthefunction
8varstr="\t\nteststring".trim();
9alert(str=="teststring");//alerts"true"
```

73. 面试题

Q: 面试官问

Javascript 中 callee 和 caller 的作用？

A: 应聘者答

caller 是返回一个对函数的引用，该函数调用了当前函数；

callee 是返回正在被执行的 function 函数，也就是所指定的 function 对象的正文。

那么问题来了？如果一对兔子每月生一对兔子；一对新生兔，从第二个月起就开始生兔子；假定每对兔子都是一雌一雄，试问一对兔子，第 n 个月能繁殖成多少对兔子？

（使用 callee 完成）

```
1varresult=[];
2functionfn(n){//典型的斐波那契数列
3if(n==1){
4return1;
5}elseif(n==2){
6return1;
7}else{
8if(result[n]){
9returnresult[n];
10}else{
```

```

11//argument.callee()表示 fn()
12result[n]=arguments.callee(n-1)+arguments.callee(n-2);
13returnresult[n];
14}
15}
16}

```

第 5 部分：中级 JavaScript 篇

74. 面试题

Q: 面试官问

实现一个函数 clone，可以对 JavaScript 中的 5 种主要的数据类型（包括 Number、String、Object、Array、Boolean）进行值复制

A: 应聘者答

考察点 1：对于基本数据类型和引用数据类型在内存中存放的是值还是指针这一区别是否清楚

考察点 2：是否知道如何判断一个变量是什么类型的

考察点 3：递归算法的设计

1//方法一：

```

2Object.prototype.clone=function(){
3varo=this.constructor===Array?[]:{};
4for(vareinthis){
5o[e]=typeofthis[e]=== "object"?this[e].clone():this[e];
6}
7returno;
8}
9

```

10//方法二：

11/**

12*克隆一个对象

13*@paramObj

14*@returns

15*/

```

16functionclone(Obj){

```

```

17varbuf;

```

```

18if(ObjinstanceofArray){

```

```

19buf=[];//创建一个空的数组
20vari=Obj.length;
21while(i--){
22buf[i]=clone(Obj[i]);
23}
24returnbuf;
25}elseif(ObjinstanceofObject){
26buf={};//创建一个空对象
27for(varkinObj){//为这个对象添加新的属性
28buf[k]=clone(Obj[k]);
29}
30returnbuf;
31}else{//普通变量直接赋值
32returnObj;
33}
34}

```

75. 面试题

Q: 面试官问

如何消除一个数组里面重复的元素？

A: 应聘者答

```

1vararr=[1,2,3,3,4,4,5,5,6,1,9,3,25,4];
2functiondeRepeat(){
3varnewArr=[];
4varobj={};
5varindex=0;
6varl=arr.length;
7for(vari=0;i<l;i++){
8if(obj[arr[i]]==undefined)
9{
10obj[arr[i]]=1;
11newArr[index++]=arr[i];
12}
13elseif(obj[arr[i]]==1)
14continue;
15}
16returnnewArr;
17

```

```

18}
19varnewArr2=deRepeat(arr);
20alert(newArr2);//输出 1,2,3,4,5,6,9,25

```

76. 面试题

Q: 面试官问

小贤是一条可爱的小狗(Dog)，它的叫声很好听(wow)，每次看到主人的时候就会乖乖叫一声(yelp)。从这段描述可以得到以下对象：

A: 应聘者答

```

1functionDog(){
2this.wow=function(){
3alert('Wow');
4}
5this.yelp=function(){
6this.wow();
7}
8}

```

小芒和小贤一样，原来也是一条可爱的小狗，可是突然有一天疯了(MadDog)，一看到人就会每隔半秒叫一声(wow)地不停叫唤(yelp)。请根据描述，按示例的形式用代码来实。(继承，原型，setInterval)

答案：

```

1functionMadDog(){
2this.yelp=function(){
3varself=this;
4setInterval(function(){
5self.wow();
6},500);
7}
8}
9MadDog.prototype=newDog();
10
11//fortest
12vardog=newDog();
13dog.yelp();
14varmadDog=newMadDog();
15madDog.yelp();

```

77. 面试题

Q: 面试官问

下面这个 ul，如何点击每一列的时候 alert 其 index?（闭包）

A: 应聘者答

```
1<ulid="test">
2<li>这是第一条</li>
3<li>这是第二条</li>
4<li>这是第三条</li>
5</ul>
答案：
1//方法一：
2varlis=document.getElementById('2223').getElementsByTagName('li');
3for(vari=0;i<3;i++)
4{
5lis[i].index=i;
6lis[i].onclick=function(){
7alert(this.index);
8};
9}
10
11//方法二：
12varlis=document.getElementById('2223').getElementsByTagName('li');
13for(vari=0;i<3;i++)
14{
15lis[i].index=i;
16lis[i].onclick=(function(a){
17returnfunction(){
18alert(a);
19}
20})(i);
21}
```

78. 面试题

Q: 面试官问

编写一个 JavaScript 函数，输入指定类型的选择器(仅需支持 id, class, tagName 三种简单 CSS 选择器，无需兼容组合选择器)可以返回匹配的 DOM 节点，需考虑浏览器兼容性和性能。

A: 应聘者答

```
/** *@param selector{String} 传入的 CSS 选择器。 *@return{Array} */
```

答案：（过长，点击打开）

79. 面试题

Q: 面试官问

请评价以下代码并给出改进意见。

A: 应聘者答

```
1 if(window.addEventListener){
2   var addListener=function(el,type,listener,useCapture){
3     el.addEventListener(type,listener,useCapture);
4   };
5 }
6 else if(document.all){
7   addListener=function(el,type,listener){
8     el.attachEvent("on"+type,function(){
9       listener.apply(el);
10    });
11 }
12 }
```

评价：

不应该在 if 和 else 语句中声明 addListener 函数，应该先声明；

不需要使用 window.addEventListener 或 document.all 来进行检测浏览器，应该使用能力检测；

由于 attachEvent 在 IE 中有 this 指向问题，所以调用它时需要处理一下改进如下：

```
1 function addEvent(elem,type,handler){
2   if(elem.addEventListener){
3     elem.addEventListener(type,handler,false);
```

```

4   }elseif(elem.attachEvent){
5       elem['temp'+type+handler]=handler;
6       elem[type+handler]=function(){
7           elem['temp'+type+handler].apply(elem);
8       };
9       elem.attachEvent('on'+type,elem[type+handler]);
10  }else{
11      elem['on'+type]=handler;
12  }
13}

```

80. 面试题

Q: 面试官问

给 String 对象添加一个方法，传入一个 string 类型的参数，然后将 string 的每个字符间价格空格返回，例如：

A: 应聘者答

```

addSpace("helloworld");//->'helloworld'
1String.prototype.spacify=function(){
2returnthis.split("").join(" ");
3};

```

接着上述答题，那么问题来了

1) 直接在对象的原型上添加方法是否安全？尤其是在 Object 对象上。(这个我没能答出？希望知道的说一下。)

2) 函数声明与函数表达式的区别？

答案：在 Javascript 中，解析器在向执行环境中加载数据时，对函数声明和函数表达式并非是一视同仁的，解析器会率先读取函数声明，并使其在执行任何代码之前可用（可以访问），至于函数表达式，则必须等到解析器执行到它所在的代码行，才会真正被解析执行。（函数声明提升）

81. 面试题

Q: 面试官问

定义一个 log 方法，让它可以代理 console.log 的方法。

A:应聘者答

可行的方法一：

```
1functionlog(msg) {  
2console.log(msg);  
3}  
4  
5log("helloworld!");//helloworld!  
6
```

如果要传入多个参数呢？显然上面的方法不能满足要求，所以更好的方法是：

```
1functionlog(){  
2console.log.apply(console,arguments);  
3};
```

那么问题来了，`apply` 和 `call` 方法的异同？

答案：

对于 `apply` 和 `call` 两者在作用上是相同的，即是调用一个对象的一个方法，以另一个对象替换当前对象。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。

但两者在参数上有区别的。对于第一个参数意义都一样，但对第二个参数：`apply` 传入的是一个参数数组，也就是将多个参数组合成为一个数组传入，而 `call` 则作为 `call` 的参数传入（从第二个参数开始）。如 `func.call(func1,var1,var2,var3)` 对应的 `apply` 写法为：`func.apply(func1,[var1,var2,var3])`。

82. 面试题

Q:面试官问

在 Javascript 中什么是伪数组？如何将伪数组转化为标准数组？

答案：

伪数组（类数组）：无法直接调用数组方法或期望 `length` 属性有什么特殊的行为，但仍可以对真正数组遍历方法来遍历它们。典型的是函数的 `argument` 参数，还有像调用 `getElementsByTagName`, `document.childNodes` 之类的，它们都返回 `NodeList` 对象都属于伪数组。可以使用 `Array.prototype.slice.call(fakeArray)` 将数组转化为真正的 `Array` 对象。

假设接第八题题干，我们要给每个 `log` 方法添加一个“(app)”前缀，比如 `'helloworld!'->'(app)helloworld!'`。方法如下：

```
1functionlog(){  
2varargs=Array.prototype.slice.call(arguments);//为了使用unshift数组方法,将argument  
转化为真正的数组  
3args.unshift('(app)');  
4  
5console.log.apply(console,args);  
6};
```

83. 面试题

Q: 面试官问

对作用域上下文和 `this` 的理解，看下列代码：

A: 应聘者答

```
1var User={
2  count:1,
3
4  getCount:function(){
5    return this.count;
6  }
7};
8
9console.log(User.getCount());//what?
10
```

```
11var func=User.getCount;
```

```
12console.log(func());//what?
```

问两处 `console` 输出什么？为什么？

答案是 1 和 `undefined`。

`func` 是在 `window` 的上下文中被执行的，所以会访问不到 `count` 属性。

那么问题来了，如何确保 `User` 总是能访问到 `func` 的上下文，即正确返回 1。

答案：正确的方法是使用 `Function.prototype.bind`。兼容各个浏览器完整代码如下：

```
1Function.prototype.bind=Function.prototype.bind||function(context){
2  var self=this;
3
4  return function(){
5    return self.apply(context,arguments);
6  };
7}
8
9var func=User.getCount.bind(User);
10console.log(func());
```

84. 面试题

Q: 面试官问

原生 JS 的 `window.onload` 与 JQuery 的 `$(document).ready(function(){})` 有什么不同? 如何用原生 JS 实现 Jq 的 `ready` 方法?

A: 应聘者答

`window.onload()` 方法是必须等到页面内包括图片的所有元素加载完毕后才能执行。

`$(document).ready()` 是 DOM 结构绘制完毕后就执行, 不必等到加载完毕。

```
1/*
2*传递函数给 whenReady()
3*当文档解析完毕且为操作准备就绪时, 函数作为 document 的方法调用
4*/
5varwhenReady=(function(){//这个函数返回 whenReady()函数
6varfuncs=[];//当获得事件时, 要运行的函数
7varready=false;//当触发事件处理程序时,切换为 true
8
9//当文档就绪时,调用事件处理程序
10functionhandler(e){
11if(ready)return;//确保事件处理程序只完整运行一次
12
13//如果发生 onreadystatechange 事件, 但其状态不是 complete 的话,那么文档尚未准备好
14if(e.type==='onreadystatechange'&&document.readyState!=='complete'){
15return;
16}
17
18//运行所有注册函数
19//注意每次都要计算 funcs.length
20//以防这些函数的调用可能会导致注册更多的函数
21for(vari=0;i<funcs.length;i++){
22funcs[i].call(document);
23}
24//事件处理函数完整执行,切换 ready 状态,并移除所有函数
25ready=true;
26funcs=null;
27}
28//为接收到的任何事件注册处理程序
29if(document.addEventListener){
30document.addEventListener('DOMContentLoaded',handler,false);
31document.addEventListener('readystatechange',handler,false);//IE9+
32window.addEventListener('load',handler,false);
33}elseif(document.attachEvent){
```

```

34document.attachEvent('onreadystatechange',handler);
35window.attachEvent('onload',handler);
36}
37//返回 whenReady()函数
38returnfunctionwhenReady(fn){
39if(ready){fn.call(document);}
40else{funcs.push(fn);}
41}
42})();

```

如果上述代码十分难懂，下面这个简化版：

```

1functionready(fn){
2if(document.addEventListener){//标准浏览器
3document.addEventListener('DOMContentLoaded',function(){
4//注销事件,避免反复触发
5document.removeEventListener('DOMContentLoaded',arguments.callee,false);
6fn();//执行函数
7},false);
8}elseif(document.attachEvent){//IE
9document.attachEvent('onreadystatechange',function(){
10if(document.readyState=='complete'){
11document.detachEvent('onreadystatechange',arguments.callee);
12fn();//函数执行
13}
14});
15}
16};

```

85. 面试题

Q: 面试官问

（设计题）想实现一个对页面某个节点的拖曳？如何做？（使用原生 JS）回答出概念即可，下面是几个要点

A: 应聘者答

给需要拖拽的节点绑定 mousedown,mousemove,mouseup 事件
 mousedown 事件触发后，开始拖拽
 mousemove 时，需要通过 event.clientX 和 clientY 获取拖拽位置，并实时更新位置
 mouseup 时，拖拽结束需要注意浏览器边界的情况

86. 面试题

Q: 面试官问

实现如下导航

请实现下面功能（只实现tips组件部分）

1. 用户第一次进来时显示，同一天访问该页面不显示tip提示
2. 用户点击“我知道了”此后访问该页面不再显示tip提醒。



A: 应聘者答

```
1functionsetcookie(name,value,days){//给 cookie 增加一个时间变量
2    varexp=newDate();
3    exp.setTime(exp.getTime()+days*24*60*60*1000);//设置过期时间为 days 天
4    document.cookie=name+"="+escape(value)+";expires="+exp.toGMTString();
5}
6functiongetCookie(name){
7    varresult="";
8    varmyCookie=""+"document.cookie+"";";
9    varsearchName="+name+"=";
10    varstartOfCookie=myCookie.indexOf(searchName);
11    varendOfCookie;
12    if(satrOfCookie!=-1){
13        startOfcookie+=searchName.length;
14        endOfCookie=myCookie.indexOf(";",startOfCookie);
15        result=(myCookie.substring(startOfCookie,endOfCookie));
16    }
17    returnresult;
18}
19(function(){
20    varoTips=document.getElementById('tips');//假设 tips 的 id 为 tips
21    varpage={
22        check:function(){//检查 tips 的 cookie 是否存在并且允许显示
23            vartips=getCookie('tips');
24            if(!tips||tips=='show')returntrue;//tips 的 cookie 不存在
25            if(tips=="never_show_again")returnfalse;
26        },
27        hideTip:function(bNever){
28            if(bNever)setcookie('tips','never_show_again',365);
29            oTips.style.display="none";//隐藏
```

```

30     },
31     showTip:function(){
32         oTips.style.display="inline";//显示，假设 tips 为行级元素
33     },
34     init:function(){
35         var _this=this;
36         if(this.check()){
37             _this.showTip();
38             setcookie('tips','show',1);
39         }
40         oTips.onclick=function(){
41             _this.hideTip(true);
42         };
43     }
44 };
45 page.init();
46 })());

```

87. 面试题

Q: 面试官问

说出以下函数的作用是？空白区域应该填写什么？

A: 应聘者答

```

1//define
2(function(window){
3functionfn(str){
4this.str=str;
5}
6
7fn.prototype.format=function(){
8vararg=_____;
9returnthis.str.replace(_____,function(a,b){
10returnarg[b]||"";
11});
12}
13window.fn=fn;
14})(window);
15
16//use

```



```

17(function(){
18var=newfn('<p><ahref="{0}">{1}</a><span>{2}</span></p>');
19console.log(t.format('http://www.alibaba.com','Alibaba','Welcome'));
20})();

```

答案：该函数的作用是使用 format 函数将函数的参数替换掉{0}这样的内容，返回一个格式化后的结果：

第一个空是：arguments

第二个空是：`\{(\d+)\}`/ig

88. 面试题

Q: 面试官问

用面向对象的 Javascript 来介绍一下自己。（没答案哦亲，自己试试吧）

A: 应聘者答

对象或者 Json 都是不错的选择哦

第 6 部分：高级 Javascript 篇

89. 面试题

Q: 面试官问

javascript 的 typeof 返回哪些数据类型

A: 应聘者答

Object number function boolean under find

90. 面试题

Q: 面试官问

例举 3 种强制类型转换和 2 种隐式类型转换？

A:应聘者答

强制 (parseInt,parseFloat,number)
隐式 (==,===)

91. 面试题

Q:面试官问

split()join()的区别

A:应聘者答

前者是切割成数组的形式，后者是将数组转换成字符串

92. 面试题

Q:面试官问

数组方法 pop()push()unshift()shift()

A:应聘者答

Push()尾部添加 pop()尾部删除
Unshift()头部添加 shift()头部删除

93. 面试题

Q:面试官问

事件绑定和普通事件有什么区别

94. 面试题

Q:面试官问

IE 和 DOM 事件流的区别

A:应聘者答

- 1.执行顺序不一样、
- 2.参数不一样
- 3.事件加不加 on
- 4.this 指向问题

95. 面试题

Q:面试官问

IE 和标准下有哪些兼容性的写法

A:应聘者答

```
Varev=ev||window.event  
document=documentElement.clientWidth||document.body.clientWidth  
Vartarget=ev.srcElement||ev.target
```

96. 面试题

Q:面试官问

ajax 请求的时候 get 和 post 方式的区别

A:应聘者答

一个在 url 后面一个放在虚拟载体里面
有大小限制
安全问题
应用不同一个是论坛等只需要请求的，一个是类似修改密码的

97. 面试题

Q:面试官问

call 和 apply 的区别

A:应聘者答

```
Object.call(this,obj1,obj2,obj3)
```

```
Object.apply(this,arguments)
```

98. 面试题

Q:面试官问

ajax 请求时，如何解释 json 数据

A:应聘者答

使用 evalparse 鉴于安全性考虑使用 parse 更靠谱

99. 面试题

Q:面试官问

写一个获取非行间样式的函数

A:应聘者答

```
functiongetStyle(obj,attr,value)
{
  if(!value)
  {
    if(obj.currentStyle)
    {
      returnobj.currentStyle(attr)
    }
    else
    {
      obj.getComputedStyle(attr,false)
    }
  }
}
```

```
}  
}  
else  
{  
  obj.style[attr]=value  
}  
}
```

100. 面试题

Q: 面试官问

闭包是什么，有什么特性，对页面有什么影响

A: 应聘者答

闭包就是能够读取其他函数内部变量的函数。

<http://blog.csdn.net/gaoshanwudi/article/details/7355794> 此链接可查看（问这个问题的不是一个公司）

101. 面试题

Q: 面试官问

添加删除替换插入到某个接点的方法

A: 应聘者答

```
obj.appendChild()  
obj.insertBefore  
obj.replaceChild  
obj.removeChild
```

102. 面试题

Q: 面试官问

解释 jsonp 的原理，以及为什么不是真正的 ajax

A:应聘者答

动态创建 script 标签，回调函数
Ajax 是页面无刷新请求数据操作

103. 面试题

Q:面试官问

javascript 的本地对象， 内置对象和宿主对象

A:应聘者答

本地对象为 array,obj,regexp 等可以 new 实例化
内置对象为 globalMath 等不可以实例化的
宿主为浏览器自带的 document,window 等

104. 面试题

Q:面试官问

documentload 和 documentready 的区别

A:应聘者答

Document.onload 是在结构和样式加载完才执行 js
Document.ready 原生种没有这个方法，jquery 中有\$.ready(function)

105. 面试题

Q:面试官问

”==”和“===”的不同

A:应聘者答

前者会自动转换类型
后者不会

106. 面试题

Q:面试官问

javascript 的同源策略

A:应聘者答

一段脚本只能读取来自于同一样源的窗口和文档的属性，这里的同一样源指的是主机名、协议和端口号的组合

107. 面试题

Q:面试官问

编写一个数组去重的方法

A:应聘者答

```
functionoSort(arr)
{
  varresult={ };
  varnewArr=[];
  for(vari=0;i<arr.length;i++)
  {
    if(!result[arr])
    {
      newArr.push(arr)
      result[arr]=1
    }
  }
  returnnewArr
}
```

108. 面试题

Q:面试官问

HTML 中定义表格的宽度用 80px 与 80%的区别是什么？

A:应聘者答

px 表示像素，%表示占整个页面宽度的百分比。

109. 面试题

Q:面试官问

简述 DIV 元素和 SPAN 元素的区别。

A:应聘者答

DIV 默认情况下是分行显示，SPAN 在同行显示。

110. 面试题

Q:面试官问

3、在 HTML 页面布局中，position 的值有哪几种？默认值又是什么？

A:应聘者答

- 1) position 的五种值分别是 static,relative,absolute,fixed,inherit
- 2) position 的缺省值是 static

111. 面试题

Q:面试官问

怎样使一个宽为 200px 和高为 200px 的层垂直居于浏览器中?写出 CSS 样式代码。

A:应聘者答

这里使用百分比和绝对定位，与外补丁负值的方法，负值的大小为其自身宽度高度一半。

112. 面试题

Q:面试官问

CSS 规范中，.（句点）后面跟一个名称代表什么含义？#（井号）后面跟一个名称代表什么含义？如果要对一个元素设置 CSS 样式（内嵌样式），应将 CSS 样式写在它的什么属性内？

A:应聘者答

- 1）.（句号）后面跟一个名称表示文档中所有 class 属性值包含这个名称的应用其样式，
- 2）#（井号）后面跟个名称表示文档中 ID 为此名称的元素应用其样式。
- 3）CSS 样式写在 style 属性内。

113. 面试题

Q:面试官问

6、CSS 具体含义是什么？在网页制作中为什么要使用 CSS 技术？

A:应聘者答

CSS(CascadingStyleSheets)也叫层叠样式表，它是用来控制网页中一个或多个元素的外观。在网页制作中，使用样式表可以减少很多具有相同外观元素的重复设置工作，从而给网页的设计和维护带来了很大的方便;具体应用是：改变样式表中元素显示样式的定义，则在使用该样式表的所有网页中相应元素的显示样式都会自动发生改变。

114. 面试题

Q:面试官问

CSS+DIV 开发 Web 页面的优势有哪些？

A:应聘者答

- 1) CSS+DIV，这个网页设计模式中，DIV 承担了网页的内容，CSS 承担了网页的样式。这样就使网页的内容和样式的分离开来。有利于页面的维护升级。
- 2) 有助于提高搜索引擎亲和力（快速找到需要的数据，而不是像在 TABLE 中一层的查找）
- 3) 有助于页面的重构(换皮肤如 blog，直接套用另外一套样式就可以实现，而不用改动网页脚本。)

115. 面试题

Q:面试官问

引用 CSS 样式，有哪几种方法？

A:应聘者答

- 1) 导入一个已经存在的 CSS 文件；
- 2) 直接在 HARD 里声明一个 CSS 代码段；
- 3) 直接在网页元素里声明使用 CSS 样式：style="color:red;"

116. 面试题

Q:面试官问

JavaScript 中 setTimeout 与 setInterval 两个函数的区别？

A:应聘者答

setTimeout 只执行一次，setInterval 没有次数限制。

117. 面试题

Q:面试官问

编写 JavaScript 脚本生成 1-6 之间的整数？

A:应聘者答

```
var NowFrame;  
NowFrame=Math.random()*6+1//随机生成一个 1-6 之间的小数  
NowFrame=parseInt(NowFrame)//把 1-6 之间的小数转化为整数
```

118. 面试题

Q:面试官问

设置什么属性可以让一个 DIV 隐藏和显示？值分别是什么？

A:应聘者答

- 1) 设置 display 属性的值为 none 时，该层隐藏。
- 2) 设置 display 属性的值为 block 时，该层显示。

119. 面试题

Q:面试官问

在 JavaScript 脚本中，isNaN 的作用是什么？

A:应聘者答

isNaN 的作用是判断值是否为数字

120. 面试题

Q:面试官问

JavaScript 中获取某个元素有哪几种方式？

A:应聘者答

`document.getElementById()`;此方法用于获得名为 ID 值的元素

`document.myform.xxx`;此方法按照层次结构来获取

`document.getElementsByName()`;此方法用于获得所有的名字相同的元素

121. 面试题

Q:面试官问

Ajax 的优缺点都有什么？

A:应聘者答

Ajax 的优点：

- 1) 页面无刷新，用户体验非常好。
- 2) 使用异步方式与服务器通信，具有更加迅速的响应能力。
- 3) 可以把一些服务器负担的工作转到客户端，利用客户端闲置的能力来处理，减轻服务器负担，节约空间和宽带租用成本。并且减轻服务器的负担，ajax 的原则是“按需取数据”，可以最大程度的减少冗余请求和响应对服务器造成的负担。
- 4) 基于标准化并被广泛支持的技术，不需要下载插件或者小程序。

Ajax 的缺点：

- 1) Ajax 不支持浏览器 back 按钮。
- 2) 安全问题，Ajax 暴露了与服务器交互的细节。
- 3) 对搜索引擎的支持比较弱。
- 4) 破坏了程序的异常机制。
- 5) 不容易调试。

122. 面试题

Q:面试官问

简述一下 Ajax 的工作原理。

A:应聘者答

Ajax 的核心是 JavaScript 对象 XMLHttpRequest。该对象在 Internet Explorer 5 中首次引入，它是一种支持异步请求的技术。简而言之，XMLHttpRequest 使您可以使用 JavaScript 向服务器提出请求并处理响应，而不阻塞用户。

123. 面试题

Q:面试官问

JavaScript 中的数据类型有几种？分别是什么？

A:应聘者答

JavaScript 中的数据类型有 7 种，分别是：

- 1) Number 数字类型
- 2) String 字符串类型
- 3) Boolean 布尔类型
- 4) Function 函数
- 5) Object 对象
- 6) Null
- 7) Undefined 没有定义类型

124. 面试题

Q:面试官问

在 JavaScript 脚本中，表达式 `2+5+"8"` 得到的结果是什么？并说明原因。

A:应聘者答

因为 2 和 5 是数字类型，“8”是字符串类型，所以相加得到的结果是 78。

第 7 部分：典型开发综合 1 篇

125. 面试题

Q:面试官问

`display:none` 和 `visibility:hidden` 的区别？

A:应聘者答

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

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

126. 面试题

Q:面试官问

CSS 控制文字，超出部分显示省略号

A:应聘者答

```
overflow:hidden;
text-overflow:ellipsis;
white-space:nowrap;
/*上面仅支持单行文本,多行呢?*/
```

```
display:-webkit-box;
-webkit-box-orient:vertical;
-webkit-line-clamp:3;
overflow:hidden;
```

使用了 WebKit 的 CSS 扩展属性，该方法适用于 WebKit 浏览器及移动端；

127. 面试题

Q:面试官问

CSS 实现三角形图标的原理

A:应聘者答

```
.caret{
display:inline-block;
width:0;
height:0;
margin-left:2px;
vertical-align:middle;
border-top:4px
```

```
solid;
border-right:4px
solid
transparent;
border-left:4px
solid
transparent;
}
```

首先，需要把元素的宽度、高度设为 0,然后设置边框样式。

要做倒立三角形、向右的、或者向左的三角形，只需要为三角形底部设置边框，两腰边框透明即可。

128. 面试题

Q:面试官问

渐进增强和优雅降级

A:应聘者答

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。

129. 面试题

Q:面试官问

CSS 中 link 和@import 的区别是？

A:应聘者答

- 1.link 属于 HTML 标签，而@import 是 CSS 提供的;
- 2.页面被加载的时，link 会同时被加载，而@import 被引用的 CSS 会等到引用它的 CSS 文件被加载完再加载;
- 3.import 只在 IE5 以上才能识别，而 link 是 HTML 标签，无兼容问题;
- 4.link 方式的样式的权重

高于@import 的权重.

130. 面试题

Q: 面试官问

position:absolute 和 float 属性的异同

A: 应聘者答

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

不同点: float 仍会占据位置, absolute 会覆盖文档流中的其他元素。

131. 面试题

Q: 面试官问

如何将页脚固定在页面底部?

A: 应聘者答

```
<div
  id="container">
  <div
    id="header">HeaderSection</div>
  <div
    id="page"
    class="clearfix">
    页面内容部分
  </div>
  <div
    id="footer">FooterSection</div>
</div>
```

<!--实现这页脚永远固定在页面的底部, 我们只需要四个 div, 其中 div#container 是一个容器, 在这个容器之中, 我们包含了 div#header (头部), div#page (页面主体部分, 我们可以在这个 div 中增加更多的 div 结构, 如上面的代码所示), div#footer (页脚部分)

-->

```
#container{
  min-height:100%;
```



```

height:auto
!important;
height:100%;/*IE6 不识别 min-height*/
position:relative;
}
#page{
width:960px;
margin:0
auto;
padding-bottom:60px;/*等于 footer 的高度*/
}
#footer{
position:absolute;
bottom:0;
width:100%;
height:60px;/*脚部的高度*/
background:#6cf;
clear:both;
}

```

132. 面试题

Q: 面试官问

如何实现元素水平居中、垂直居中、水平垂直居中？

A: 应聘者答

水平居中

水平居中：行内元素

如果被设置元素为文本、图片等行内元素时，水平居中是通过给父元素设置 text-align:center 来实现的

```

.parent{
text-align:center;
}

```

水平居中：定宽块状元素

满足定宽和块状两个条件的元素是可以通过设置“左右 margin”值为“auto”来实现居中的。

```

.item{
margin:3rem
auto;
}

```

水平居中：多个块状元素解决方案

将元素的 `display` 属性设置为 `inline-block`，并且把父元素的 `text-align` 属性设置为 `center` 即可：

```
.parent{
  text-align:center;
}
```

水平居中：多个块状元素解决方案(使用 `flexbox` 布局实现)

使用 `flexbox` 布局，只需要把待处理的块状元素的父元素添加属性 `display:flex` 及 `justify-content:center` 即可：

```
.parent{
  display:flex;
  justify-content:center;
}
```

水平居中：不定宽块状元素

a.改变块级元素的 `display` 为 `inline` 类型，然后使用

`text-align:center` 来实现居中效果

b.通过给父元素设置 `float`，然后给父元素设置 `position:relative` 和 `left:50%`，子元素设置 `position:relative` 和 `left:-50%`来实现水平居中

c.如果这个标签是 `table` 等表格元素可以使用左右 `margin` 居中

垂直居中

父元素高度确定的单行文本

垂直居中的方法是通过设置父元素的 `height` 和

`line-height`

高度一致来实现的

垂直居中的属性

`vertical-align`，但这个样式只有在父元素为

`td`

或

`th`

时，才会生效

/*`vertical-align` 和 `display:inline-block` 在一起有讲究*/

垂直居中：多行的行内元素解决方案

组合使用 `display:table-cell` 和 `vertical-align:middle` 属性来定义需要居中的元素的父容器元素生成效果，如下：

```
.parent{
  background:#222;
  width:300px;
  height:300px;
  /*以下属性垂直居中*/
  display:table-cell;
  vertical-align:middle;
}
```

垂直居中：已知高度的块状元素解决方案

```
.item{
```

```

top:50%;
margin-top:-50px;/*margin-top 值为自身高度的一半*/
position:absolute;
padding:0;
}
水平垂直居中
水平垂直居中：已知高度和宽度的元素解决方案
这是一种不常见的居中方法，可自适应,如下：
.item{
position:absolute;
margin:auto;
left:0;
top:0;
right:0;
bottom:0;
}
水平垂直居中：未知高度和宽度元素解决方案
.item{
position:absolute;
top:50%;
left:50%;
transform:translate(-50%,-50%);/*使用 css3 的 transform 来实现*/
}
水平垂直居中：使用 flex 布局实现
.parent{
display:flex;
justify-content:center;
align-items:center;
/*注意这里需要设置高度来查看垂直居中效果*/
background:#AAA;
height:300px;
}

```

133. 面试题

Q: 面试官问

谈谈 flex 语法

A: 应聘者答

容器属性

flex-direction
flex-wrap
flex-flow(flex-direction 和 flex-wrap 的简写)
justify-content
align-items
align-content
项目属性
order
flex-grow
flex-shrink
flex-basis
flex(flex-grow,flex-shrink 和 flex-basis 的简写)
align-self
详见：阮一峰老师的 Flex 布局教程：语法篇

134. 面试题

Q:面试官问

介绍一下 box-sizing 属性？

A:应聘者答

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

。

标准浏览器下，按照 W3C 规范对盒模型解析，一旦修改了元素的边框或内距，就会影响元素的盒子尺寸，

就不得不重新计算元素的盒子尺寸，从而影响整个页面的布局。

content-box

：让元素维持 W3C 的标准盒模型。元素的宽度/高度由 border+padding+
content

的宽度/高度决定，

设置 width/height 属性指的是

content

部分的宽/高

border-box

：让元素维持 IE 传统盒模型（IE6 以下版本和 IE6~7 的怪异模式）。

设置 width/height 属性指的是 border+padding+

content

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

content-box (默认)

布局所占宽度 **Width:**

$Width = width +$

$padding-left$

$+$

$padding-right$

$+$

$border-left$

$+$

$border-right$

布局所占高度 **Height:**

$Height = height +$

$padding-top$

$+$

$padding-bottom$

$+$

$border-top$

$+$

$border-bottom$

padding-box

布局所占宽度 **Width:**

$Width = width(包含 padding-left$

$+$

$padding-right) +$

$border-top$

$+$

$border-bottom$

布局所占高度 **Height:**

$Height = height(包含 padding-top$

$+$

$padding-bottom) +$

$border-top$

$+$

$border-bottom$

border-box

布局所占宽度 **Width:**

$Width = width(包含 padding-left$

$+$

$padding-right$

$+$

$border-left$

$+$

border-right)

布局所占高度 Height:

Height=height(包含 padding-top

+

padding-bottom

+

border-top

+

border-bottom)

z-index 的使用?

z-index

属性设置元素的堆叠顺序。拥有更高堆叠顺序的元素总是会处于堆叠顺序较低的元素的前面。

z-index

属性仅在节点的 position 属性为 relative,absolute 或者 fixed 时生效.

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

1.id 选择器 (#myid)

2.类选择器 (.myclassname)

3.标签选择器 (div,h1,p)

4.相邻选择器 (h1

+p)

5.子选择器 (ul

>li)

6.后代选择器 (lia)

7.通配符选择器 (

*

)

8.属性选择器 (a[rel

=

“external”])

9.伪类选择器 (a:hover,li:nth-child)

优先级为:

!important>id>class>tag

important 比

内联优先级高,但内联比 id 要高

135. 面试题

Q: 面试官问

CSS3 新增伪类举例:

A:应聘者答

p:first-of-type

选择属于其父元素的首个<p>元素的每个<p>元素。

p:last-of-type

选择属于其父元素的最后<p>元素的每个<p>元素。

p:only-of-type

选择属于其父元素唯一的<p>元素的每个<p>元素。

p:only-child

选择属于其父元素的唯一子元素的每个<p>元素。

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

:enabled

:disabled

控制表单控件的禁用状态。

:checked

单选框或复选框被选中。

136. 面试题

Q:面试官问

CSS3 有哪些新特性？

A:应聘者答

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

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

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

增加了更多的 CSS 选择器

多背景 rgba

在 CSS3 中唯一引入的伪元素是::selection.

媒体查询, 多栏布局

border-image

137. 面试题

Q: 面试官问

对 BFC 规范的理解？

A: 应聘者答

BFC，块级格式化上下文，一个创建了新的 BFC 的盒子是独立布局的，盒子里面的子元素的样式不会影响到外面的元素。

在同一个 BFC 中的两个毗邻的块级盒在垂直方向（和布局方向有关系）的 margin 会发生折叠。

W3CCSS2.1

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

138. 面试题

Q: 面试官问

上下 margin 重合问题

A: 应聘者答

IE 和 FF 都存在，相邻的两个 div 的 margin-left 和 margin-right 不会重合，但是 margin-top 和 margin-bottom 却会发生塌陷，

塌陷的最终尺寸是两个值中较大的一个。

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

当然，还可以将两个元素设置 display:inline-block;或者浮动

发生塌陷的不只是兄弟元素，父子关系也会发生，当父元素没有设定 padding-top 或 border-top 时（等于 0），

子元素的 margin-top 会和父元素的 margin-top 合并，与兄弟元素一样，margin 最终会取值较大的作用于父元素，子元素会紧贴父元素上边界。

padding 对于 block、inline 水平元素

对于 block 块级元素

padding 值暴走，一定会影响尺寸

对于 inline 水平元素

水平 padding 影响尺寸,垂直 padding 不影响尺寸,但是会影响背景色(占据空间)

139. 面试题

Q:面试官问

说说你对语义化的理解?

A:应聘者答

- 1.去掉或者丢失样式的时候能够让页面呈现出清晰的结构
 - 2.有利于 SEO: 和搜索引擎建立良好沟通,有助于爬虫抓取更多的有效信息:爬虫依赖于标签来确定上下文和各个关键字的权重;
 - 3.方便其他设备解析(如屏幕阅读器、盲人阅读器、移动设备)以意义的方式来渲染网页;
 - 4.便于团队开发和维护,语义化更具可读性,是下一步吧网页的重要动向,遵循 W3C 标准的团队都遵循这个标准,可以减少差异化。
- 前端开发规范手册(推荐)
详见:前端开发规范手册

140. 面试题

Q:面试官问

Doctype 作用?严格模式与混杂模式如何区分?它们有何意义?

A:应聘者答

- 1.<!DOCTYPE>声明位于文档中的最前面,处于<html>标签之前。告知浏览器以何种模式来渲染文档。
 - 2.严格模式的排版和 JS 运作模式是以该浏览器支持的最高标准运行。
 - 3.在混杂模式中,页面以宽松的向后兼容的方式显示。模拟老式浏览器的行为以防止站点无法工作。
 - 4.<!DOCTYPE>不存在或格式不正确会导致文档以混杂模式呈现。
- 你知道多少种 Doctype 文档类型?
该标签可声明三种 DTD 类型,分别表示严格版本、过渡版本以及基于框架的 HTML 文档。
- HTML4.01 规定了三种文档类型: Strict、Transitional 以及 Frameset。
XHTML1.0 规定了三种 XML 文档类型: Strict、Transitional 以及 Frameset。

Standards（标准）模式（也就是严格呈现模式）用于呈现遵循最新标准的网页，而 Quirks（包容）模式（也就是松散呈现模式或者兼容模式）用于呈现为传统浏览器而设计的网页。

141. 面试题

Q: 面试官问

HTML 与 XHTML——二者有什么区别

A: 应聘者答

区别:

- 1.所有的标记都必须要有个相应的结束标记
- 2.所有标签的元素和属性的名字都必须使用小写
- 3.所有的 XML 标记都必须合理嵌套
- 4.所有的属性必须用引号“”括起来
- 5.把所有<和&特殊符号用编码表示
- 6.给所有属性赋一个值
- 7.不要在注释内容中使“-”
- 8.图片必须有说明文字

142. 面试题

Q: 面试官问

常见兼容性问题?

A: 应聘者答

png24 位的图片在 ie6 浏览器上出现背景，解决方案是做成 PNG8.也可以引用一段脚本处理.

浏览器默认的 margin 和 padding 不同.解决方案是加一个全局的*{margin:0;padding:0;}来统一。

IE6 双边距 bug:块属性标签 float 后，又有横行的 margin 情况下，在 ie6 显示 margin 比设置的大。

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

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

0

0

100px;}

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

渐进识别的方式，从总体中逐渐排除局部。

首先，巧妙的使用 “\9” 这一标记，将 IE 浏览器从所有情况中分离出来。

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

```
.bb{
background-color:#f1ee18;/*所有识别*/
.background-color:#00deff\9;/*IE6、7、8 识别*/
+background-color:#a200ff;/*IE6、7 识别*/
_background-color:#1e0bd1;/*IE6 识别*/
}
```

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

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

可以使用 HTML5 推荐的写法：<!DOCTYPEhtml>

position 的值，relative 和 absolute 分别是相对于谁进行定位的？

absolute:生成绝对定位的元素，

相对于最近一级的

定位不是 static 的父元素来进行定位。

fixed（老 IE 不支持<IE11）生成绝对定位的元素，通常相对于浏览器窗口或 frame 进行定位。

relative 生成相对定位的元素，相对于其在普通流中的位置进行定位。

static 默认值。没有定位，元素出现在正常的流中

sticky 生成粘性定位的元素，容器的位置根据正常文档流计算得出

浮动元素引起的问题和解决办法？

143. 面试题

Q:面试官问

解释下浮动和它的工作原理？清除浮动的技巧

A:应聘者答

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 `cssclear: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；

144. 面试题

Q: 面试官问

浮动元素引起的问题：

A: 应聘者答

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

145. 面试题

Q: 面试官问

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

```
#parent:after{
  content: ".";
  height: 0;
  visibility: hidden;
  display: block;
  clear: both;
}
```

3. 浮动外部元素

4. 设置 `overflow` 为 `hidden` 或者 `auto`

DOM 操作——怎样添加、移除、移动、复制、创建和查找节点。
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()//并没有 insertAfter()
查找
getElementsByName()//通过标签名称
getElementsByName()//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

146. 面试题

Q: 面试官问

DOM 模型中节点类型有？

A: 应聘者答

Document 节点（document 节点是文档的根节点，每张网页都有自己的 document 节点）

Element 节点（Element 对象对应网页的 HTML 标签元素）

Text 节点（Text 节点代表 Element 节点和 Attribute 节点的文本内容）

DocumentFragment 节点（代表一个文档的片段，本身就是一个完整的 DOM 树形结构）

HTML5 有哪些新特性、移除了那些元素？如何处理 HTML5 新标签的浏览器兼容问题？如何区分 HTML 和 HTML5？

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

拖拽释放(Draganddrop)API

语义化更好的内容标签（header,nav,footer,aside,article,section）

音频、视频 API(audio,video)

画布(Canvas)API

地理(Geolocation)API

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

sessionStorage 的数据在浏览器关闭后自动删除

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

新的技术 webworker,websocket,Geolocation

移除的元素

纯表现的元素: basefont, big, center, font,s, strike, tt, u;
对可用性产生负面影响的元素: frame, frameset, noframes;

147. 面试题

Q:面试官问

什么是 FOUC（无样式内容闪烁）？你如何来避免 FOUC？

A:应聘者答

FOUC–FlashOfUnstyledContent 文档样式闪烁

```
<style  
type="text/css"  
media="all">@import  
  "../fouc.css";</style>
```

引用 CSS 文件的 @import 就是造成这个问题的罪魁祸首。IE 会先加载整个 HTML 文档的 DOM，然后再去导入外部的 CSS 文件，

因此，在页面 DOM 加载完成到 CSS 导入完成中间会有一段时间页面上的内容是没有样式的，这段时间的长短跟网速，电脑速度都有关系。

解决方法简单的出奇，只要在<head>之间加入一个<link>或者<script>元素就可以了。

WebWorker、WebSocket、WebStorage

WebWorker 主线程:

1.通过 worker=
new

Worker(url)加载一个 JS 文件来创建一个 worker，同时返回一个 worker 实例。

2.通过 worker.postMessage(data)方法来向 worker 发送数据。

3.绑定 worker.onmessage 方法来接收 worker 发送过来的数据。

4.可以使用

worker.terminate()来终止一个 worker 的执行。

WebSocket

WebSocket 是 Web 应用程序的传输协议，它提供了双向的，按序到达的数据流。

他是一个 HTML5 协议，WebSocket 的连接是持久的，他通过在客户端和服务端之间保持双工连接，

服务器的更新可以被及时推送给客户端，而不需要客户端以一定时间间隔去轮询。

WebStorage

WebStorage 是 HTML5 引入的一个非常重要的功能，可以在客户端本地存储数据。

类似 HTML4 的 cookie，但可实现功能要比 cookie 强大的多，cookie 大小被限制在 4KB，WebStorage 官方建议为每个网站 5MB。

WebStorage 又分为两种: sessionStorage、localStorage

浏览器本地存储

在较高版本的浏览器中，js 提供了 sessionStorage 和 globalStorage。

HTML5

中的 WebStorage

包括了两种存储方式: sessionStorage 和 localStorage。

sessionStorage 用于本地存储一个会话 (session) 中的数据, 这些数据只有在同一个会话中的页面才能访问并且当会话结束后数据也随之销毁。

因此 sessionStorage 不是一种持久化的本地存储, 仅仅是会话级别的存储。

而 localStorage 用于持久化的本地存储, 除非主动删除数据, 否则数据是永远不会过期的。

如何实现浏览器内多个标签页之间的通信?

调用 localStorage、cookies 等本地存储方式

148. 面试题

Q: 面试官问

webstorage 和 cookie 的区别

A: 应聘者答

- 1、WebStorage 的概念和 cookie 相似, 区别是它是为了更大容量存储设计的。
 - 2、Cookie 的大小是受限的, 并且每次你请求一个新的页面的时候 Cookie 都会被发送过去, 这样无形中浪费了带宽, 另外`cookie`还需要指定作用域, 不可以跨域调用。
 - 3、除此之外, WebStorage 拥有 setItem,getItem,removeItem,clear 等方法, 不像 cookie 需要前端开发者自己封装 setCookie, getCookie。
 - 4、但是 cookie 也是不可或缺的: cookie 的作用是与服务器进行交互, 作为 HTTP 规范的一部分而存在, 而 WebStorage 仅仅是为了在本地“存储”数据而生
- 浏览器的支持除了`IE 7`及以下不支持外, 其他标准浏览器都完全支持(ie 及 FF 需在 web 服务器里运行), 值得一提的是 IE 总是办好事,
- 例如 IE7、IE6 中的 userData 其实就是 javascript 本地存储的解决方案。通过简单的代码封装可以统一到所有的浏览器都支持 webstorage。
- localStorage 和 sessionStorage 都具有相同的操作方法, 例 setItem、getItem 和 removeItem 等

149. 面试题

Q: 面试官问

cookie 和 session 的区别:

A:应聘者答

- 1、cookie 数据存放在客户的浏览器上，session 数据放在服务器上。
- 2、cookie 不是很安全，别人可以分析存放在本地的 COOKIE 并进行 COOKIE 欺骗
考虑到安全应当使用 session。
- 3、session 会在一定时间内保存在服务器上。当访问增多，会比较占用你服务器的性能
考虑到减轻服务器性能方面，应当使用 COOKIE。
- 4、单个 cookie 保存的数据不能超过 4K，很多浏览器都限制一个站点最多保存 20 个 cookie。
- 5、所以个人建议：
将登陆信息等重要信息存放为 SESSION
其他信息如果需要保留，可以放在 COOKIE 中

150. 面试题

Q:面试官问

请你谈谈 Cookie 的弊端

A:应聘者答

cookie 虽然在持久保存客户端数据提供了方便，分担了服务器存储的负担，但还是有很多局限性的。

- 1.IE6 或更低版本最多 20 个 cookie
- 2.IE7 和之后的版本最后可以有 50 个 cookie。
- 3.Firefox 最多 50 个 cookie
- 4.chrome 和 Safari 没有做硬性限制

IE 和 Opera 会清理近期最少使用的 cookie，Firefox 会随机清理 cookie。

cookie 的最大大约为 4096 字节，为了兼容性，一般不能超过 4095 字节。

IE 提供了一种存储可以持久化用户数据，叫做 userdata，从 IE5.0 就开始支持。每个数据最多 128K，每个域名下最多 1M。这个持久化数据放在缓存中，如果缓存没有清理，那么会一直存在。

优点：极高的扩展性和可用性

- 1.通过良好的编程，控制保存在 cookie 中的 session 对象的大小。
- 2.通过加密和安全传输技术（SSL），减少 cookie 被破解的可能性。
- 3.只在 cookie 中存放不敏感数据，即使被盗也不会有重大损失。
- 4.控制 cookie 的生命期，使之不会永远有效。偷盗者很可能拿到一个过期的 cookie。

缺点：

- 1.`Cookie` 数量和长度的限制。每个 domain 最多只能有 20 条 cookie，每个 cookie 长度不能超过 4KB，否则会被截掉。

2.安全性问题。如果 cookie 被人拦截了，那人就可以取得所有的 session 信息。即使加密也与事无补，因为拦截者并不需要知道 cookie 的意义，他只要原样转发 cookie 就可以达到目的了。

3.有些状态不可能保存在客户端。例如，为了防止重复提交表单，我们需要在服务器端保存一个计数器。如果我们把这个计数器保存在客户端，那么它起不到任何作用。

151. 面试题

Q:面试官问

如何保证 Session 的安全？

A:应聘者答

验证用户的使用环境[浏览器和 IP 地址]。

分配给用户 SessionID 时，同时探明用户使用的浏览器和 IP 地址，作为验证依据，使非法用户不能进行 SessionID 欺骗。

正确处理 Session 变量。

当用户注销时，立即删除 SessionID。并设置好 Session 的生存周期，过期就自动删除。

JavaScript

152. 面试题

Q:面试官问

null 和 undefined 的区别？变量赋值、内存泄露、端事件差异区别

A:应聘者答

null 是一个表示“无”的对象，转为数值时为 0；undefined 是一个表示“无”的原始值，转为数值时为 NaN。

当声明的变量还未被初始化时，变量的默认值为 undefined。

null 用来表示尚未存在的对象，常用来表示函数企图返回一个不存在的对象。

undefined 表示“缺少值”，就是此处应该有一个值，但是还没有定义。典型用法是：

- 1.变量被声明了，但没有赋值时，就等于 undefined。
- 2.调用函数时，应该提供的参数没有提供，该参数等于 undefined。
- 3.对象没有赋值的属性，该属性的值为 undefined。
- 4.函数没有返回值时，默认返回 undefined。

null 表示“没有对象”，即该处不应该有值。典型用法是：

1.作为函数的参数，表示该函数的参数不是对象。

2.作为对象原型链的终点。

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);
```

153. 面试题

Q:面试官问

js 延迟加载的方式有哪些？

A:应聘者答

defer 和 async、动态创建 DOM 方式（创建 script，插入到 DOM 中，加载完毕后 callBack）、按需异步载入 js

154. 面试题

Q:面试官问

call()和 apply()的区别和作用？

A:应聘者答

作用：动态改变某个类的某个方法的运行环境（执行上下文）。

`apply()`函数有两个参数：第一个参数是上下文，第二个参数是参数组成的数组。如果上下文是 `null`，则使用全局对象代替。例如：

```
function.apply(this,[1,2,3])
```

`call()`的第一个参数是上下文，后续是实例传入的参数序列，例如：

```
function.call(this,1,2,3);
```

Scope 作用范围

```
(function(){
```

```
vara=b=5;
```

```
})();
```

```
console.log(b);//5
```

这里有两个变量声明，但 `a` 使用关键字 `var` 声明的。代表它是一个函数的局部变量。与此相反，`b` 变成了全局变量。

这个问题的另一个诀窍是，它没有使用严格模式（`'usestrict'`）。如果启用了严格模式，代码就会引发 `ReferenceError` 的错误：`B` 没有定义（`isnotdefined`）。

请记住，严格模式，则需要明确指定，才能实现全局变量声明。比如，你应该写：

```
(function(){
```

```
  'usestrict';
```

```
vara=
```

```
window.b=5;
```

```
})();
```

```
console.log(b);//
```

声明提升（Hoisting）

```
function
```

```
test(){
```

```
  console.log(a);
```

```
  console.log(foo());
```

```
  vara=1;
```

```
  function
```

```
  foo(){
```

```
    return2;}
}
```

```
}
```

```
test();
```

//结果是 `undefined` 和 2

//原因是，变量和函数的声明都被提前了（移到了函数的顶部），但变量不分配任何值。

//因此，在打印变量的时候，它在函数中存在（它被声明了），但它仍然是 `undefined` 哪些操作会造成内存泄漏？

内存泄漏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）

JavaScript 垃圾回收方法

标记清除 (markandsweep)

这是 JavaScript 最常见的垃圾回收方式，当变量进入执行环境的时候，比如函数中声明一个变量，

垃圾回收器将其标记为“进入环境”，当变量离开环境的时候（函数执行结束）将其标记为“离开环境”。

垃圾回收器会在运行的时候给存储在内存中的所有变量加上标记，然后去掉环境中的变量以及被环境中变量

所引用的变量（闭包），在这些完成之后仍存在标记的就是要删除的变量了

引用计数(referencecounting)

在低版本 IE 中经常会出现内存泄露，很多时候就是因为其采用引用计数方式进行垃圾回收。

引用计数的策略是跟踪记录每个值被使用的次数，当声明了一个

变量并将一个引用类型赋值

给该变量的时候这个值的引用次数就加 1，如果该变量的值变成了另外一个，则这个值得引用次数减 1，

当这个值的引用次数变为 0 的时

候，说明没有变量在使用，这个值没法被访问了，因此可以将其占用的空间回收，这样垃圾回收器会在运行的时候清理掉引用次数为 0 的值占用的空间。

在 IE 中虽然 JavaScript 对象通过标记清除的方式进行垃圾回收，但 BOM 与 DOM 对象却是通过引用计数回收垃圾的，也就是说只要涉及 BOM 及 DOM 就会出现循环引用问题。

JavaScript 中的 PC 端点击事件和移动端点击事件的差异性和原理：

在手持设备的浏览器上（此处主要指定 IOS 和 Android 系统上的 Webkit 内核的浏览器和嵌入在应用程序里面的 Webview），

由于两次连续的“轻触”是放大的操作（即使你两次轻触的是一个链接或一个有 click 的时间监听的元素），所以在第一次被

“轻触”后，浏览器需要先等一段时间，看看有没有所谓的“连续第二次轻触”。如果有，则进行“放大”操作。否则，才敢放心

地认为用户不是要放大，而是需要“click”，至此才敢触发 click 事件，导致“短按”（手指接触屏幕到离开屏幕的时间比较短）

的 click 时间通常大约会延迟 300ms。

在移动平台浏览器中事件的触发顺序，是 touchstart-touchmove-touchend-click

155. 面试题

Q: 面试官问

列举 IE 与其他浏览器不一样的特性？

A:应聘者答

IE 支持 `currentStyle`, Firefox 使用 `getComputedStyle`

IE 使用 `innerText`, Firefox 使用 `textContent`

滤镜方面: IE: `filter:alpha(opacity=num)`; Firefox: `-moz-opacity:num`

事件方面: IE: `attachEvent`; 火狐是 `addEventListener`

鼠标位置: IE 是 `event.clientX`; 火狐是 `event.pageX`

IE 使用 `event.srcElement`; Firefox 使用 `event.target`

IE 中消除 `list` 的原点仅需 `margin:0` 即可达到最终效果; Firefox 需要设置 `margin:0;padding:0` 以及 `list-style:none`

CSS 圆角: ie7 以下不支持圆角

说说你对 `Promise` 的理解

依照 `Promise` 的定义, `Promise` 有三种状态:

异步操作 “未完成” “(pending)”

异步操作 “已完成” “(resolved, 又称 fulfilled)”

异步操作 “失败” “(rejected)”

这三种的状态的变化途径只有两种。

异步操作从 “未完成” 到 “已完成”

异步操作从 “未完成” 到 “失败”。

另外, `fulfilled` 与 `rejected` 一起合称 `settled`。

`Promise` 对象用来进行延迟(deferred)和异步(asynchronous)计算。

`Promise` 的构造函数, 最基本的用法如下:

```
var promise =
new
Promise(function(resolve, reject){
  if(...){ // succeed
    resolve(result);
  } else { // fails
    reject(Error(errMessage));
  }
});
```

`Promise` 实例拥有 `then` 方法 (具有 `then` 方法的对象, 通常被称为 `thenable`)。它的使用方法如下:

```
promise.then(onFulfilled, onRejected)
```

//接收两个函数作为参数, 一个在`fulfilled`的时候被调用, 一个在`rejected`的时候被调用,

//接收参数就是`future`, `onFulfilled`对应`resolve`,`onRejected`对应`reject`。`

//`then` 方法可以链式使用

实现一个函数 `clone`, 可以对 JavaScript 中的 5 种主要的数据类型 (包括 `Number`、`String`、`Object`、`Array`、`Boolean`) 进行值复制

```
Object.prototype.clone=function(){
  var o=
  this.constructor===Array?[]:{};
```

```

for(var e in
this){
o[e]=typeof
this[e]=== "object"?
this[e].clone():
this[e];
}
return o;
}

```

JavaScript 中快速排序、去重

快速排序

//排序过程只需要三步:

//1.在数据集之中, 找一个基准点

//2.建立两个数组, 分别存储左边和右边的数组

//3.利用递归进行下次比较

```

function
quickSort(arr){
if(arr.length
<=1)return arr;//如果数组只有一个数, 就直接返回;
var index=Math.floor(arr.length/2);//找到中间数的索引值, 如果是浮点数, 则向下取整
var key=arr.splice(index,1)[0];//找到中间数的值
var left=[],right=[];
arr.forEach(function(v){
v<=key?
left.push(v):
right.push(v);//基准点的左边的数传到左边数组、右边的数传到右边数组
});
return
quickSort(left).concat([key],quickSort(right));//递归不断重复比较
}

```

去重、基本数组去重

```

Array.prototype.unique=function(){
var result=[];
this.forEach(function(v){
if(result.indexOf(v)<
0){
result.push(v);
}
});
return result;
}

```

去重、利用 hash 表去重, 这是一种空间换时间的方法

```

Array.prototype.unique=function(){
var result=[],hash={};

```

```

this.forEach(function(v){
  if(!hash[v]){
    hash[v]=true;
    result.push(v);
  }
});
return result;
}

```

上面的方法存在一个 bug，对于数组[1,2,'1','2',3]，去重结果为[1,2,3]，原因在于对象对属性索引时会进行强制类型转换，arr['1']和 arr[1]得到的都是 arr[1]的值，因此需做一些改变：

```

Array.prototype.unique=function(){
  var result=[],hash={ };
  this.forEach(function(v){
    var type=typeof(v); //获取元素类型
    hash[v]||(hash[v]=new
    Array());
    if(hash[v].indexOf(type)<
    0){
      hash[v].push(type); //存储类型
    }
    result.push(v);
  });
  return result;
}

```

说说严格模式的限制

严格模式主要有以下限制：

变量必须声明后再使用

函数的参数不能有同名属性，否则报错

不能使用 with 语句

不能对只读属性赋值，否则报错

不能使用前缀 0 表示八进制数，否则报错

不能删除不可删除的属性，否则报错

不能删除变量 deleteprop，会报错，只能删除属性 deleteglobal[prop]

eval 不会在它的外层作用域引入变量

eval 和 arguments 不能被重新赋值

arguments 不会自动反映函数参数的变化

不能使用 arguments.callee

不能使用 arguments.caller

禁止 this 指向全局对象

不能使用 fn.caller 和 fn.arguments 获取函数调用的堆栈

增加了保留字（比如 protected、static 和 interface）

设立“严格模式”的目的，主要有以下几个：

1.消除 Javascript 语法的一些不合理、不严谨之处，减少一些怪异行为；

2.消除代码运行的一些不安全之处，保证代码运行的安全；

3.提高编译器效率，增加运行速度；

4.为未来新版本的 Javascript 做好铺垫。

注：经过测试 IE6,7,8,9 均不支持严格模式。

如何删除一个 cookie

1.将时间设为当前时间往前一点。

```
var date =
```

```
new
```

```
Date();
```

```
date.setDate(date.getDate()-
```

```
1); //真正的删除 setDate()方法用于设置一个月的某一天。
```

2.expires 的设置

```
document.cookie
```

```
=
```

```
    'user=' +
```

```
    encodeURIComponent('name') +
```

```
    ';expires='
```

```
    +
```

```
    new
```

```
Date(0)
```

```
document.write() 的用法
```

document.write()方法可以用在两个方面：页面载入过程中用实时脚本创建页面内容，以及用延时脚本创建本窗口或新窗口的内容。

```
document.write
```

只能重绘整个页面。innerHTML 可以重绘页面的一部分

编写一个方法

求一个字符串的字节长度

假设：一个英文字符占用一个字节，一个中文字符占用两个字节

```
function
```

```
GetBytes(str){
```

```
    var len =
```

```
    str.length;
```

```
    var bytes = len;
```

```
    for(var i = 0; i < len; i++){
```

```
        if(str.charCodeAt(i) >
```

```
            255) bytes++;
```

```
    }
```

```
    return bytes;
```

```
}
```

```
alert(GetBytes("你好,as"));
```

156. 面试题

Q: 面试官问

javascript 对象的几种创建方式

A: 应聘者答

1. 工厂模式
2. 构造函数模式
3. 原型模式
4. 混合构造函数和原型模式
5. 动态原型模式
6. 寄生构造函数模式
7. 稳妥构造函数模式

157. 面试题

Q: 面试官问

javascript 继承的 6 种方法

A: 应聘者答

1. 原型链继承
 2. 借用构造函数继承
 3. 组合继承(原型+借用构造)
 4. 原型式继承
 5. 寄生式继承
 6. 寄生组合式继承
- 详情: ECMAScript 面向对象的程序设计

158. 面试题

Q: 面试官问

用过哪些设计模式?

A:应聘者答

工厂模式：

主要好处就是可以消除对象间的耦合，通过使用工程方法而不是 `new` 关键字。将所有实例化的代码集中在一个位置防止代码重复。

工厂模式解决了重复实例化的问题

，但还有一个问题,那就是识别问题，因为根本无法搞清楚他们到底是哪个对象的实例。

```
function
createObject(name,age,profession){//集中实例化的函数 varobj=newObject();
obj.name=name;
obj.age=age;
obj.profession=profession;
obj.move=function(){returnthis.name+ 'at'+this.age+ 'engagedin'+this.profession;
};
returnobj;
}
vartest1=createObject('trigkit4',22,'programmer');//第一个实例
vartest2=createObject('mike',25,'engineer');//第二个实例
```

159. 面试题

Q:面试官问

构造函数模式

A:应聘者答

使用构造函数的方法

，即解决了重复实例化的问题

，又解决了对象识别的问题，该模式与工厂模式的不同之处在于：

- 1.构造函数方法没有显示的创建对象(`newObject()`);
- 2.直接将属性和方法赋值给 `this` 对象;
- 3.没有 `return` 语句。

160. 面试题

Q: 面试官问

js 继承方式及其优缺点

A: 应聘者答

原型链继承的缺点

一是字面量重写原型会中断关系，使用引用类型的原型，并且子类型还无法给超类型传递参数。

借用构造函数（类式继承）

借用构造函数虽然解决了刚才两种问题，但没有原型，则复用无从谈起。所以我们需要原型链+借用构造函数的模式，这种模式称为组合继承

组合式继承

组合式继承是比较常用的一种继承方法，其背后的思路是

使用原型链实现对原型属性和方法的继承，而通过借用构造函数来实现对实例属性的继承。这样，既通过在原型上定义方法实现了函数复用，又保证每个实例都有它自己的属性。

具体请看：JavaScript 继承方式详解

161. 面试题

Q: 面试官问

说说你对作用域链的理解

A: 应聘者答

javascript 语言特性中，有很多方面和我们接触的其他编程语言不太一样，比如说，javascript 语言实现继承机制的核心就是原型，而不是 Java 语言那样的类式继承。

Javascript 解析引擎在读取一个 Object 的属性的值时，会沿着原型链向上寻找，如果最终没有找到，则该属性值为 undefined 如果最终找到该属性的值，则返回结果。

与这个过程不同的是，当 javascript 解析引擎执行“给一个 Object 的某个属性赋值”的时候，如果当前 Object 存在该属性，则改写该属性的值，如果当前的 Object 本身并不存在该属性，则赋值该属性的值。

作用域链的作用是保证执行环境里有权访问的变量和函数是有序的，作用域链的变量只能向上访问，变量访问到 window 对象即被终止，作用域链向下访问变量是不被允许的。

162. 面试题

Q: 面试官问

说说你对闭包的理解

A: 应聘者答

使用闭包主要是为了设计私有的方法和变量。闭包的优点是可以避免全局变量的污染，缺点是闭包会常驻内存，

会增大内存使用量，使用不当很容易造成内存泄露。在 js 中，函数即闭包，只有函数才会产生作用域的概念

闭包有三个特性：

1. 函数嵌套函数
2. 函数内部可以引用外部的参数和变量
3. 参数和变量不会被垃圾回收机制回收

具体请看：详解 js 闭包

163. 面试题

Q: 面试官问

ajax 的缺点和在 IE 下的问题？阐述创建 ajax 的过程

A: 应聘者答

ajax 的缺点

- 1、ajax 不支持浏览器 back 按钮。
- 2、安全问题 AJAX 暴露了与服务器交互的细节。
- 3、对搜索引擎的支持比较弱。
- 4、破坏了程序的异常机制。
- 5、不容易调试。

164. 面试题

Q: 面试官问

IE 缓存问题

A:应聘者答

在 IE 浏览器下，如果请求的方法是 GET，并且请求的 URL 不变，那么这个请求的结果就会被缓存。

解决这个问题的办法可以通过实时改变请求的 URL，只要 URL 改变，就不会被缓存，可以通过在 URL

末尾添加上随机的时间戳参数(`'t'='+newDate().getTime()`)

或者：

```
open('GET','demo.php?rand='+Math.random(),'true');
```

165. 面试题

Q:面试官问

Ajax 请求的页面历史记录状态问题

A:应聘者答

可以通过锚点来记录状态，`'location.hash'`。让浏览器记录 Ajax 请求时页面状态的变化。

还可以通过`HTML5`的`history.pushState`，来实现浏览器地址栏的无刷新改变

166. 面试题

Q:面试官问

创建 ajax 的过程

A:应聘者答

(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.

```
varxmlHttp=newXMLHttpRequest();
xmlHttp.open('GET','demo.php','true');
xmlHttp.send()xmlHttp.onreadystatechange=function(){
if(xmlHttp.readyState===4&xmlHttp.status===200){
}
}
```

167. 面试题

Q:面试官问

异步加载和延迟加载

A:应聘者答

1.异步加载的方案:

动态插入 script 标签

2.通过 ajax 去获取 js 代码,然后通过 eval 执行

3.script 标签上添加 defer 或者 async 属性

4.创建并插入 iframe, 让它异步执行 js

5.延迟加载: 有些 js 代码并不是页面初始化的时候就立刻需要的, 而稍后的某些情况才需要的。

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

Flash、Ajax 各自的优缺点, 在使用中如何取舍?

Flash 适合处理多媒体、矢量图形、访问机器; 对 CSS、处理文本上不足, 不容易被搜索。

Ajax 对 CSS、文本支持很好, 支持搜索; 多媒体、矢量图形、机器访问不足。

共同点: 与服务器的无刷新传递消息、用户离线和在线状态、操作 DOM

168. 面试题

Q: 面试官问

请解释一下 JavaScript 的同源策略。为什么要有同源限制？

A: 应聘者答

概念: 同源策略是客户端脚本（尤其是 Javascript）的重要的安全度量标准。它最早出自 NetscapeNavigator2.0，其目的是防止某个文档或脚本从多个不同源装载这里的同源策略指的是：协议，域名，端口相同，同源策略是一种安全协议。指一段脚本只能读取来自同一来源的窗口和文档的属性。

169. 面试题

Q: 面试官问

为什么要有同源限制？

A: 应聘者答

举例说明：比如一个黑客程序，他利用 Iframe 把真正的银行登录页面嵌到他的页面上，当你使用真实的用户名，密码登录时，他的页面就可以通过 Javascript 读取到你的表单中 input 中的内容，这样用户名，密码就轻松到手了。

缺点：

现在网站的 JS 都会进行压缩，一些文件用了严格模式，而另一些没有。

这时这些本来是严格模式的文件，被 merge 后，这个串就到了文件的中间，不仅没有指示严格模式，反而在压缩后浪费了字节。

170. 面试题

Q: 面试官问

WEB 应用从服务器主动推送 Data 到客户端有那些方式？

A: 应聘者答

Javascript 数据推送

Commet: 基于 HTTP 长连接的服务器推送技术

基于 WebSocket 的推送方案

SSE (Server-SendEvent): 服务器推送数据新方式

你觉得 jQuery 或 zepto 源码有哪些写的好的地方

(答案仅供参考)

jquery 源码封装在一个匿名函数的自执行环境中, 有助于防止变量的全局污染,

然后通过传入 window 对象参数, 可以使 window 对象作为局部变量使用, 好处是当

jquery

中访问 window 对象的时候, 就不用将作用域链退回到顶层作用域了, 从而可以更快的访问 window 对象。

同样, 传入 undefined 参数, 可以缩短查找 undefined 时的作用域链。

```
(function(window,undefined){
```

```
//用一个函数域包起来, 就是所谓的沙箱
```

```
//在这里边 var 定义的变量, 属于这个函数域内的局部变量, 避免污染全局
```

```
//把当前沙箱需要的外部变量通过函数参数引入进来
```

```
//只要保证参数对内提供的接口的一致性, 你还可以随意替换传进来的这个参数
```

```
window.jQuery
```

```
=
```

```
window.$
```

```
=jQuery;
```

```
})(window);
```

jquery 将一些原型属性和方法封装在了

jquery.prototype 中, 为了缩短名称, 又赋值给了 jquery.fn, 这是很形象的写法。

有一些数组或对象的方法经常能使用到, jQuery 将其保存为局部变量以提高访问速度。

jquery 实现的链式调用可以节约代码, 所返回的都是同一个对象, 可以提高代码效率。

ES6 的了解新增模板字符串 (为 JavaScript 提供了简单的字符串插值功能)、箭头函数 (操作符左边为输入的参数, 而右边则是进行的操作以及返回的值 Inputs=>outputs。)、for-of (用来遍历数据—例如数组中的值。)

arguments 对象可被不定参数和默认参数完美代替。ES6 将 promise 对象纳入规范, 提供了原生的 Promise 对象。

增加了 let 和 const 命令, 用来声明变量。增加了块级作用域。let 命令实际上就增加了块级作用域。

ES6 规定, var 命令和 function 命令声明的全局变量, 属于全局对象的属性; let 命令、const 命令、class 命令声明的全局变量, 不属于全局对象的属性。还有就是引入`module`模块的概念。

171. 面试题

Q: 面试官问

XML 和 JSON 的区别?

A:应聘者答

(1).数据体积方面。

JSON 相对于 XML 来讲，数据的体积小，传递的速度更快些。

(2).数据交互方面。

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

(3).数据描述方面。

JSON 对数据的描述性比 XML 较差。

(4).传输速度方面。

JSON 的速度要远远快于 XML。

JSON 对象有两个方法：stringify()和 parse()。这两个方法分别用于把 JavaScript 对象系列化为 JSON 字符串和把 JSON 字符串解析为原生的 JavaScript 值。

网络

172. 面试题

Q:面试官问

说说网络分层里七层模型是哪七层

A:应聘者答

应用层：应用层、表示层、会话层（从上往下）（HTTP、FTP、SMTP、DNS）

传输层（TCP 和 UDP）

网络层（IP）

物理和数据链路层（以太网）

每一层的作用如下：

物理层：通过媒介传输比特,确定机械及电气规范（比特 Bit）

数据链路层：将比特组装成帧和点到点的传递（帧 Frame）

网络层：负责数据包从源到宿的传递和网际互连（包 PackeT）

传输层：提供端到端的可靠报文传递和错误恢复（段 Segment）

会话层：建立、管理和终止会话（会话协议数据单元 SPDU）

表示层：对数据进行翻译、加密和压缩（表示协议数据单元 PPDU）

应用层：允许访问 OSI 环境的手段（应用协议数据单元 APDU）

173. 面试题

Q: 面试官问

各种协议

A: 应聘者答

ICMP 协议:

(InternetControlMessageProtocol) 因特网控制报文协议

它是 TCP/IP 协议族的一个子协议, 用于在 IP 主机、路由器之间传递控制消息。

TFTP 协议:

(TrivialFileTransferProtocol) 简单文件传输协议

是 TCP/IP 协议族中的一个用来在客户机与服务器之间进行简单文件传输的协议, 提供不复杂、开销不大的文件传输服务。

HTTP 协议: (HyperTextTransferProtocol) 超文本传输协议

是一个属于应用层的面向对象的协议, 由于其简捷、快速的方式, 适用于分布式超媒体信息系统。

DHCP 协议:

(DynamicHostConfigurationProtocol) 动态主机配置协议

是一种让系统得以连接到网络上, 并获取所需要的配置参数手段。

HTTP 状态码

1XX 表示请求已经接受了, 正在处理

100Continue 继续, 一般在发送 post 请求时, 已发送了 httpheader 之后服务端将返回此信息, 表示确认, 之后发送具体参数信息

2XX 表示处理成功接受, 处理掉了

200OK 正常返回信息

201Created 请求成功并且服务器创建了新的资源

202Accepted 服务器已接受请求, 但尚未处理

3XX 重定向, 比如完成这个操作可能需要重定向, 或者更多操作

301MovedPermanently 请求的网页已永久移动到新位置。

302Found 临时性重定向。

303SeeOther 临时性重定向, 且总是使用 GET 请求新的 URI。

304NotModified 自从上次请求后, 请求的网页未修改过。

4XX 客户端错误

请求错误

比如说, 语法错误, 无法实现

400BadRequest 服务器无法理解请求的格式, 客户端不应当尝试再次使用相同的内容发起请求。

401Unauthorized 请求未授权。

403Forbidden 禁止访问。

404NotFound 找不到如何与 URI 相匹配的资源。

5XX 可能就是服务器端错误,

500InternalServerError 最常见的服务器端错误。

503ServiceUnavailable 服务器端暂时无法处理请求 (可能是过载或维护)。

GET 和 POST 的区别，何时使用 POST？

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

POST：一般用于修改服务器上的资源，对所发送的信息没有限制。

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

也就是说 Get 是通过地址栏来传值，而 Post 是通过提交表单来传值。

然而，在以下情况中，请使用 POST 请求：

无法使用缓存文件（更新服务器上的文件或数据库）

向服务器发送大量数据（POST 没有数据量限制）

发送包含未知字符的用户输入时，POST 比 GET 更稳定也更可靠

HTTPHeader 中，和缓存相关的字段有：

Cache-control

Expires

Age

Vary

Etag

HTTP 和 HTTPS

HTTP 协议通常承载于 TCP 协议之上，在 HTTP 和 TCP 之间添加一个安全协议层（SSL 或 TLS），这个时候，就成了我们常说的 HTTPS。

默认 HTTP 的端口号为 80，HTTPS 的端口号为 443。

174. 面试题

Q: 面试官问

为什么 HTTPS 安全

A: 应聘者答

因为网络请求需要中间有很多的服务器路由器的转发。中间的节点都可能篡改信息，而如果使用 HTTPS，

密钥在你和终点站才有。https 之所以比 http 安全，是因为他利用 ssl/tls 协议传输。

它包含证书，卸载，流量转发，负载均衡，页面适配，浏览器适配，refer 传递等。保障了传输过程的安全性

175. 面试题

Q:面试官问

说说 TCP 传输的三次握手四次挥手策略

A:应聘者答

为了准确无误地把数据送达目标处，TCP 协议采用了三次握手策略。用 TCP 协议把数据包送出去后，TCP 不会对传送

后的情况置之不理，它一定会向对方确认是否成功送达。握手过程中使用了 TCP 的标志：SYN 和 ACK。

发送端首先发送一个带 SYN 标志的数据包给对方。接收端收到后，回传一个带有 SYN/ACK 标志的数据包以示传达确认信息。

最后，发送端再回传一个带 ACK 标志的数据包，代表“握手”结束。

若在握手过程中某个阶段莫名中断，TCP 协议会再次以相同的顺序发送相同的数据包。

断开一个 TCP 连接则需要“四次握手”：

第一次挥手：主动关闭方发送一个 FIN，用来关闭主动方到被动关闭方的数据传送，也就是主动关闭方告诉被动关闭方：

我已经不会再给你发数据了(当然，在 fin 包之前发送出去的数据，如果没有收到对应的 ack 确认报文，主动关闭方依然会重发这些数据)，

但是，此时主动关闭方还可以接受数据。

第二次挥手：被动关闭方收到 FIN 包后，发送一个 ACK 给对方，确认序号为收到序号+1（与 SYN 相同，一个 FIN 占用一个序号）。

第三次挥手：被动关闭方发送一个 FIN，用来关闭被动关闭方到主动关闭方的数据传送，也就是告诉主动关闭方，我的数据也发送完了，不会再给你发数据了。

第四次挥手：主动关闭方收到 FIN 后，发送一个 ACK 给被动关闭方，确认序号为收到序号+1，至此，完成四次挥手。

176. 面试题

Q:面试官问

为什么要三次□？

A:应聘者答

在只有两次“握手”的情形下，假设 Client 想跟 Server 建立连接，但是却因为中途连接请求的数据报丢失了，

故 Client 端不得不重新发送一遍；这个时候 Server 端仅收到一个连接请求，因此可以正常的建立连接。

但是，有时候 Client 端重新发送请求不是因为数据报丢失了，而是有可能数据传输过程因为网络并发量很大在

某结点被阻塞了，这种情形下 Server 端将先后收到 2 次请求，并持续等待两个 Client 请求向他发送数据...问题就在这里，

Client 端实际上只有一次请求，而 Server 端却有 2 个响应，极端的情况可能由于 Client 端多次重新发送请求数据而导致

Server 端最后建立了 N 多个响应在等待，因而造成极大的资源浪费！所以，“三次握手”很有必要！

177. 面试题

Q: 面试官问

为什么要四次👉？

A: 应聘者答

试想一下，假如现在你是客户端你想断开跟 Server 的所有连接该怎么做？第一步，你自己先停止向 Server 端发送数据，并等待 Server 的回复。但事情还没有完，虽然你自身不往 Server 发送数据了，但是因为你们之前已经建立好平等的连接了，所以此时他也有主动权向你发送数据；故 Server 端还得终止主动向你发送数据，并等待你的确认。其实，说白了就是保证双方的一个合约的完整执行！

使用 TCP 的协议：FTP（文件传输协议）、Telnet（远程登录协议）、SMTP（简单邮件传输协议）、POP3（和 SMTP 相对，用于接收邮件）、HTTP 协议等。

178. 面试题

Q: 面试官问

TCP 和 UDP 的区别

A: 应聘者答

TCP（TransmissionControlProtocol，传输控制协议）是基于连接的协议，也就是说，在正式收发数据前，必须和对方建立可靠的连接。

一个 TCP 连接必须要经过三次“对话”才能建立起来

UDP（UserDataProtocol，用户数据报协议）是与 TCP 相对应的协议。它是面向非连接的协议，它不与对方建立连接，而是直接就把数据包发送过去！

UDP 适用于一次只传送少量数据、对可靠性要求不高的应用环境。

179. 面试题

Q:面试官问

一个页面从输入 URL 到页面加载显示完成，这个过程中都发生了什么？

A:应聘者答

分为 4 个步骤：

(1).当发送一个 URL 请求时，不管这个 URL 是 Web 页面的 URL 还是 Web 页面上每个资源的 URL，浏览器都会开启一个线程来

处理这个请求，同时在远程 DNS 服务器上启动一个 DNS 查询。这能使浏览器获得请求对应的 IP 地址。

(2).浏览器与远程 Web 服务器通过 TCP 三次握手协商来建立一个 TCP/IP 连接。该握手包括一个同步报文，一个同步 - 应答报文和一个应答报文，这三个报文在浏览器和服务器之间传递。该握手首先由客户端尝试建立起通信，而后服务器

应答并接受客户端的请求，最后由客户端发出该请求已经被接受的报文。

(3).一旦 TCP/IP 连接建立，浏览器会通过该连接向远程服务器发送 HTTP 的 GET 请求。远程服务器找到资源并使用 HTTP 响应返回该资源，值为 200 的 HTTP 响应状态表示一个正确的响应。

(4).此时，Web 服务器提供资源服务，客户端开始下载资源。

请求返回后，便进入了我们关注的前端模块

简单来说，浏览器会解析 HTML 生成 DOMTree，其次会根据 CSS 生成 CSSRuleTree，而 javascript 又可以根据 DOMAPI 操作 DOM

180. 面试题

Q:面试官问

什么是 Etag？

A:应聘者答

当发送一个服务器请求时，浏览器首先会进行缓存过期判断。浏览器根据缓存过期时间判断缓存文件是否过期。

情景一：若没有过期，则不向服务器发送请求，直接使用缓存中的结果，此时我们在浏览器控制台中可以看到 200OK(fromcache)，

此时的情况就是完全使用缓存，浏览器和服务器没有任何交互的。

情景二：若已过期，则向服务器发送请求，此时请求中会带上①中设置的文件修改时间，和 Etag 然后，进行资源更新判断。服务器根据浏览器传过来的文件修改时间，判断自浏览器上一次请求之后，文件是不是没有被修改过；

181. 面试题

Q: 面试官问

根据 Etag，判断文件内容自上一次请求之后，有没有发生变化

A: 应聘者答

情形一：若两种判断的结论都是文件没有被修改过，则服务器就不给浏览器发`index.html`的内容了，直接告诉它，文件没有被修改过，你用你那边的缓存吧——`304NotModified`，此时浏览器就会从本地缓存中获取`index.html`的内容。此时的情况叫协议缓存，浏览器和服务器之间有一次请求交互。

情形二：若修改时间和文件内容判断有任意一个没有通过，则服务器会受理此次请求，之后的操作同①①只有 get 请求会被缓存，post 请求不会 Expires 和 Cache-ControlExpires 要求客户端和服务端的时钟严格同步。HTTP1.1 引入 Cache-Control 来克服 Expires 头的限制。

如果 max-age 和 Expires 同时出现，则 max-age 有更高的优先级。

Cache-Control:no-cache,private,max-age=0

Etag:abcde

Expires:Thu,15Apr2014

20:00:00

GMT

Pragma:

private

Last-Modified:\$now//RFC1123format

Etag 应用:

Etag 由服务器端生成，客户端通过 If-Match 或者说 If-None-Match 这个条件判断请求来验证资源是否修改。

常见的是使用 If-None-Match。请求一个文件的流程可能如下：

====第一次请求====

1.客户端发起 HTTPGET 请求一个文件；

2.服务器处理请求，返回文件内容和一堆 Header，当然包括 Etag(例如“2e681a-6-5d044840”)(假设服务器支持 Etag 生成和已经开启了 Etag).状态码 200

====第二次请求====

客户端发起 HTTPGET 请求一个文件，注意这个时候客户端同时发送一个 If-None-Match 头，这个头的内容就是第一次请求时服务器返回的 Etag: 2e681a-6-5d0448402.服务器判断发送过来的 Etag 和计算出来的 Etag 匹配，因此 If-None-Match 为 False，不返回 200，返回 304，客户端继续使用本地缓存；流程很简单，问题是，如果服务器又设置了

Cache-Control:max-age 和 Expires 呢，怎么办答案是同时使用，也就是说在完全匹配 If-Modified-Since 和 If-None-Match 即检查完修改时间和 Etag 之后，服务器才能返回 304.(不要陷入到底使用谁的问题怪圈)

为什么使用 Etag 请求头？

Etag 主要为了解决 Last-Modified 无法解决的一些问题。

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。

JSONP: json+padding（内填充），顾名思义，就是把 JSON 填充到一个盒子里

```
<script>
function
createJs(sUrl){
var oScript=
document.createElement('script');
oScript.type
= 'text/javascript';
oScript.src=sUrl;
document.getElementsByTagName('head')[0].appendChild(oScript);
}
createJs('jsonp.js');
box({
  'name':
  'test'
});
function
box(json){
alert(json.name);
}
</script>
```

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。

通过修改 document.domain 来跨子域

将子域和主域的 document.domain 设为同一个主域.前提条件：这两个域名必须属于同一个基础域名！

而且所用的协议，端口都要一致，否则无法利用 document.domain 进行跨域主域相同的使用 document.domain

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。

编程能力

数据结构和算法

数据结构和算法贯穿于编程的每个角落，前端开发也不例外。虽然我们的工作集中在界面交互上，数据结构和算法的使用场景看似不多，

其实不知不觉我们在使用

HTML 文档本身就是一个棵树，DOM 定义了操作这棵树的标准接口，jQuery 在标准的 DOMAPI 的基础上提供了更好用的接口操作 DOM

JavaScript 的 Object 本身就是一个 hashmap

JSON 的数据的序列化和反序列化设计字符串处理的相关算法

前端模版引擎宋昌会利用一些词法分析算法去完成模版的解析和渲染

Facebook 的 React 在渲染 DOM 过程中就采用了 Diff 算法

数据图表类应用会涉及多种数据算法的应用

浏览器底层综合利用了多种数据结构和算法才将我们产出的 HTML，CSS 绘制成用户可见的页面

实现一个函数，输入一个字符串，检验是否为合法的 JSON 字符串。如果不合法，标出第一处不符合规则的字符的 index 和该字符。

判断一个 JSON 结构是否非法时很简单的，捕捉 `JSON.parse` 的异常即可。但要知道 index，需要有类似语法分析的流程

排序算法的时间开销主要可用算法执行中的数据比较次数和数据移动次数来衡量

时间开销

排序的时间开销可用算法执行中的数据比较次数与数据移动次数来衡量。

算法运行时间代价的大略估算一般都按平均情况进行估算。对于那些受对象排序码序列初始排列及对象个数影响较大的，需要按最好情况和最坏情况进行估算

空间开销

算法执行时所需的附加存储。

线性表的链式存储结构一定优于顺序存储结构吗？为什么？

顺序表的特点是逻辑上相邻的数据元素，物理存储位置也相邻，并且，顺序表的存储空间需要预先分配。

优点：

方法简单，各种高级语言中都有数组，容易实现。

不用为表示节点间的逻辑关系而增加额外的存储开销。

顺序表具有按元素序号随机访问的特点。

缺点：

在顺序表中做插入、删除操作时，平均移动表中的一半元素，因此对 n 较大的顺序表效率低。

需要预先分配足够大的存储空间，估计过大，可能会导致顺序表后部大量闲置；预先分配过小，又会造成溢出。

在链表中逻辑上相邻的数据元素，物理存储位置不一定相邻，它使用指针实现元素之间的逻辑关系。并且，链表的存储空间是动态分配的。

优点：

比顺序存储结构的存储密度小(链式存储结构中每个结点都由数据域与指针域两部分组成，相比顺序存储结构增加了存储空间)。

逻辑上相邻的节点物理上不必相邻。

插入、删除灵活(不必移动节点，只要改变节点中的指针)。

每个结点是由数据域和指针域组成。

缺点：

要占用额外的存储空间存储元素之间的关系，存储密度降低。存储密度是指一个节点中数据元素所占的存储单元和整个节点所占的存储单元之比。

链表不是一种随机存储结构，不能随机存取元素。

查找结点时链式存储要比顺序存储慢。

顺序表与链表的优缺点恰好相反，那么在实践应用中怎样选取存储结构呢？通常有以下几点考虑：

顺序表的存储空间是静态分配的，在程序执行之前必须明确规定它的存储规模，也就是说事先对“MaxSize”要有合适的设定，

设定过大会造成存储空间的浪费，过小造成溢出。因此，当对线性表的长度或存储规模难以估计时，不宜采用顺序表。

然而，链表的动态分配则可以克服这个缺点。链表不需要预留存储空间，也不需要知道表长如何变化，只要内存空间尚有空闲，

就可以再程序运行时随时地动态分配空间，不需要时还可以动态回收。

因此，当线性表的长度变化较大或者难以估计其存储规模时，宜采用动态链表作为存储结构。

当线性表的长度变化不大而且事先容易确定其大小时，为节省存储空间，则采用顺序表作为存储结构比较适宜。

基于运算的考虑（时间）

顺序存储是一种随机存取的结构，而链表则是一种顺序存取结构，因此它们对各种操作有完全不同的算法和时间复杂度。

例如，要查找线性表中的第 i 个元素，对于顺序表可以直接计算出 $a(i)$ 的地址，不用去查找，其时间复杂度为 $O(1)$ 。

而链表必须从链表头开始，依次向后查找，平均需要 $O(n)$ 的时间。所以，如果经常做的运算是按序号访问数据元素，显然顺表优于链表。

反之，在顺序表中做插入，删除时平均移动表中一半的元素，当数据元素的信息量较大而且表比较长时，这一点是不应忽视的；

在链表中作插入、删除，虽然要找插入位置，但操作是比较操作，从这个角度考虑显然后者优于前者。

基于环境的考虑（语言）

顺序表容易实现，任何高级语言中都有数组类型；链表的操作是基于指针的。相对来讲前者简单些，也用户考虑的一个因素。

总之，两种存储结构各有长短，选择哪一种由实际问题中的主要因素决定。通常“较稳定”的线性表，即主要操作是查找操作的线性表，

适于选择顺序存储；而频繁做插入删除运算的（即动态性比较强）的线性表适宜选择链式存储。

列出你所知道的排序算法名称。及其平均时间复杂度、空间复杂度

正则表达式

JavaScript 提供了 6 个方法用于正则表达式，其中两个是正则对戏那个的方法，4 个是字符串对象方法。

`RegExp.prototype.test()`

`RegExp.prototype.exec()`

`String.prototype.match()`

`String.prototype.search()`

`String.prototype.replace()`

`String.prototype.split()`

详见：RegExp 对象扩展技术 Node.js

node.js 相关 Node.js 之所以是一个出色的框架，基于下面几个原因：

JavaScript 端到端：可以用 JavaScript 同时编写服务器端和客户端脚本。

事件驱动的可扩展性：Node.js 应用时间模型在同一个线程上来处理 Web 请求。

可扩展性：Node.js 追随者多、社区活跃、安装新模块简单

快速执行：建立 Node.js，并在其中开发时超级容易的。

182. 面试题

Q: 面试官问

什么是错误优先的回调函数？

A: 应聘者答

错误优先的回调函数(Error-FirstCallback)用于同时返回错误和数据。

第一个参数返回错误，并且验证它是否出错；其他参数用于返回数据。

```
fs.readFile(filePath,function(err,data)
```

```
{
```

```
if(err)
```

```
{
```

```
//处理错误
```

```
return
```

```
console.log(err);
```

```
}
```

```
console.log(data);
```

```
});
```

如何避免回调地狱？

以下方式可以避免回调地狱:

模块化:将回调函数转换为独立的函数

使用流程控制库, 例如 `async`

使用 `Promise`

使用 `async/await`

183. 面试题

Q: 面试官问

什么是 `Stub`? 举例说明

A: 应聘者答

`Stub` 用于模拟模块的行为。测试时, `Stub` 可以为函数调用返回模拟的结果。比如说, 当我们写文件时, 实际上并不需要真正去写。

```
var fs =
  require('fs');
var writeFileStub =
  sinon.stub(fs, 'writeFile', function(path, data, cb)
  {
    return
    cb(null);
  });
expect(writeFileStub).to.be.called;
writeFileStub.restore();
```

184. 面试题

Q: 面试官问

如何保证依赖的安全性?

A: 应聘者答

编写 `Node.js` 应用时, 很可能依赖成百上千的模块。例如, 使用了 `Express` 的话, 会直接依赖 27 个模块。

因此, 手动检查所有依赖是不现实的。唯一的办法是对依赖进行自动化的安全检查, 有这些工具可供选择:

`npm outdated`

TracebyRisingStack

NSP

GreenKeeper

Snyk

请简要说明 `module.exports` 与 `exports` 的关系

`module` 和 `exports` 都是模块文件的上下文，更确切地讲，模块文件的代码真正被执行的是被包装过的，比如：

```
(function(exports,require,module,__filename,__dirname){  
  //包装头  
  console.log("helloworld!")  
  //原始文件内容  
})  
  //包装尾
```

`module` 和 `exports` 都是模块被执行时的参数。其中 `exports` 也是 `module` 的属性，默认情况下是一个空对象。

当 `require` 一个模块时，实际上得到的是该模块的 `exports` 属性。

在非 `repl` 模式下可以得到验证：

```
console.log(module.exports
```

```
===
```

```
exports);  
true
```

```
console.log(exports);  
{}
```

Node.js 在执行 `require(id)` 时是怎样找到一个模块的？

Node.js 中的 `require(id)` 执行分 3 种情况：引入内建模块、引入文件模块、引入一个包。

通过 `id` 可以分析得到该模块是一个内建模块、文件模块还是包

若是内建模块时，直接从内存中加载当 `id` 是相对路径或绝对路径时，模块被认为是一个文件模块，通过文件查找可以定位到文件的路径。

当 `id` 既不是内建模块也不是文件模块时，则被认为是一个包。这个包可能通过 `npm` 安装的第三方模块。

包的加载方式为，从当前路径下寻找 `node_modules` 目录中是否存在该包。如果没有，想上一级目录进行查找，知道根目录下的 `node_modules`。这个规则可以通过 `module.paths` 得到。

寻找到包后，UI 中啊到爆的描述文件 `package.json`，该文件的 `main` 字段表明了这个包的入口文件，

此时再按文件的方式找到对应文件即可。

（目前自动加载扩展名可省略类型 `.js`、`.json`、`.node`）`.node` 文件在不同的平台下内容不同，在 Windows 下其实是 `.dll` 文件，其他平台下是 `.so` 文件能否使用 `require('json')` 的方式加载大量的 JSON 文件？

Node.js 中推崇非阻塞 I/O，但是 `require` 一个模块时确实同步调用的，这会带来性能上的开销，但并不是每次 `require` 都很耗时因为在 `require` 成功之后会缓存起来，再次加载时直接从缓存读取，并没有额外开销。

当通过 `json` 的饭食家在文件时固然方便，但大量使用时会导致这些书被缓存，大量数据驻留在内存中，导致 GC 频繁和内存泄露。

185. 面试题

Q: 面试官问

Node.js 中的异步 I/O 是如何进行的？

A: 应聘者答

Node.js 的异步 I/O 通过时间循环的方式实现。其中异步 I/O 又分磁盘 I/O 和网络 I/O。

在磁盘 I/O 的调用中，当发起异步调用后，会将异步操作送进 libuv 提供的队列中，然后返回。

在磁盘 I/O 执行完成之后，会形成一个事件，事件循环的过程总发现该事件后，会将其消费。消费过程就是将得到的数据和传入的回调函数执行。

消费过程就是将得到的数据和传入的回调函数执行。

网络 I/O 与磁盘 I/O 的差异在于他不需要线程池来进行处理，而是在每次事件循环的过程中通过 IOCP/epoll/kqueue/eventports 来获取网络的 I/O 的事件队列。

186. 面试题

Q: 面试官问

Node 的子进程有什么原理？

A: 应聘者答

Node 的子进程模块为 `child_process`，类似操作系统的 API，但又有所不同。

子进程模块有 `exec`、`execFile`、`fork`、`spawn` 4 个异步方法和 `execSync`、`execFileSync`、`spawnSync` 3 个同步调用方法。

`child_process` 模块与操作系统底层 API 的区别在于它并不是用来 `spawn` 和 `fork` 当前进程的，通过这些 API 创建出来的子进程与父进程之间没有任何必然的关系。

当创建出来的子进程也是一个 node 进程时，可以在两个进程之间通信。Node 内建了 IPC 机制，暴露的 API 为 `child.send()` 和 `process.on('message')`。

其内部机制在不同的平台下有不同的实现，UNIX 下为 `DomainSocket`，Windows 为 `NamedPipe`。

187. 面试题

Q: 面试官问

进程 / 线程间的同步机制有？

A:应聘者答

话题：linux 下进程间的同步机制有哪些？

如何实现在父进程退出时子进程跟着退出？

请写一段代码，确保多个进程同时写入同一个文件成功 PHP 为例：

```
function writeData($filepath,$data)
{
    $fp=fopen($filepath,'a');
    do{
        usleep(100);//微秒
    }while(!flock($fp,LOCK_EX));//取得独占锁定
    $res=fwrite($fp,$data."\n");
    flock($fp,LOCK_UN);//释放锁定
    fclose($fp);
    return $res;
}
```

188. 面试题

Q:面试官问

Cluster 是什么原理？

A:应聘者答

Node 的 cluster 模块用于解决 Node 单进程无法充分利用操作系统多核 CPU 的问题。另外，通过 master-cluster 模式，

也可以使得 Node 应用的健壮性能得到保障。

Cluster 的底层是 child_process 模块，Node 进程间通信除了可以发送普通的消息外，还约定能够发送底层对象，包括 TCP、UDP 等。

当一个 TCP 链接从主进程发送到子进程后，子进程能够根据消息重建出 TCP 连接，这样 Cluster 可以决定 fork 出适合硬件资源的

子进程数。这些子进程能够通过 IPC 的机制分享对同一个端口的侦听，另外，通过对进程间通信管道的处理，可以自动完成进程退出重启等

编写一个简单的包，要求能够通过 npm 进行发布，发布成功，能在另一个项目中进行调用。

189. 面试题

Q: 面试官问

如何拼接多个 Buffer 为一个 Buffer?

A: 应聘者答

```
Buffer.concat(list[,totalLength]);  
list<Array>ListofBufferobjectstoconcat  
totalLength<Number>TotallengthoftheBuffersinthelistwhenconcatenated  
Return:<Buffer>
```

190. 面试题

Q: 面试官问

简述基于 jQuery 的组件开发和基于 React 的组件开发的异同

A: 应聘者答

jQuery 组件开发需要通过封装类似原生 DOM 的 API, 而 React 可以关注于抽象的组件封装, 组件包含了数据初始化、组件渲染、组件挂载等关键的钩子

jQuery 是通过手动定位来实现组件的改变的, 代码是命令式的。React 通过声明一个方法来表示如何根据数据渲染 HTML。代码是声明式的。

jQuery 的组件无法很好的解决嵌套问题。React 组件的全部 DOM 操作、生命周期都由框架一并控制, 已由框架解决组件的嵌套问题。

简述基于 React 的 FLUX 架构要解决的问题

简述 React 组件的生命周期

如果组件初始化时需要 Ajax 获取数据, 代码应该写在哪里?

前端工程

广义的前端工程是一个系统工程, 狭义的前端工程是前端开发流程的一部分。

前端工程包含许多方面: 有以下内容:

1. 代码规范: 保证团队所有成员以同样的规范开发代码
2. 分支管理: 不同工程师开发如那件的不同功能或组件, 按照统一的流程合并进主干
3. 模块管理: 一方面, 团队引用的模块应该是规范的; 另一方面, 必须保证这些模块可以正确得加入最终的编译好的软件包
4. 自动化测试: 为了保证合并进主干的代码达到质量标准, 不会让软件挂掉, 必须要有测试, 而且测试应自动化的, 可以回归。
5. 构建: 主干更新以后, 自动将代码编译为最终的目标格式, 并且准备好各种静态资源
6. 部署: 将构建好的代码部署到生产环境

很多静态文件不断修改，如何保证浏览器总是加载到修改后的版本？

一种方法是使用 HTTP 头信息，指定浏览器不得缓存（或者缓存时间较短）

Cache-Control:no-cache,no-store,must-revalidate

Pragma:no-cache

Expires:0

但是，这样浏览器每次加载都需要下载相关文件，会增加带宽消耗，也不利于性能。

常见的方法，为每个静态文件的 URL 附加一个版本号或哈希作为查询字符串。文件每次更新，就更新版本号。

```
<link
rel="stylesheet"
href="a.css?v=1.0.0"/>
```

这种方法对于很多静态文件相互依赖的页面会产生问题。因为 HTML 页面和静态资源文件往往不会同时部署上线。一旦新版的静态资源先上线，

同时此时又有用户去加载老版的 HTML 页面，就可能导致问题。

因此，更好解决方法是，每当静态资源文件有内容变化，就该用一个全新的文件名。

```
<link
rel="stylesheet"
href="a.v1.0.0.css"/>
```

这样就会使得 HTML 页面和静态资源的版本总是匹配的，再配上灰度发布，就能够实现平滑发布。

什么是代码的静态检查，以及什么工具可以检查代码的风格？

静态检查指在不运行代码的前提下，仅仅根据代码的静态解析发现错误的做法。也就是说，静态检查时编译时的检查，而不是运行时的检查。

常见的检查工具如下：

ESLint

JSLint

JSHint

JSCS

什么是测试金字塔？举例说明

测试金字塔反映了需要写的单元测试、集成测试以及端到端测试的比例：

测试 HTTP 接口时应该是这样的：

很多单元测试，分别测试各个模块(依赖需要 stub)

较少的集成测试，测试各个模块之间的交互(依赖不能 stub)

少量端到端测试，去调用真正地接口(依赖不能 stub)

自动化构建工具，谈谈你对 webpack 的看法

WebPack 是一个模块打包工具，你可以使用 WebPack 管理你的模块依赖，并编译输出模块们所需的静态文件。

它能够很好地管理、打包 Web 开发中所用到的 HTML、Javascript、CSS 以及各种静态文件（图片、字体等），

让开发过程更加高效。对于不同类型的资源，webpack 有对应的模块加载器。webpack 模块打包器会分析模块间的依赖关系，

最后

生成了优化且合并后的静态资源。

webpack 的两大特色：

1.codesplitting（可以自动完成）

2.loader 可以处理各种类型的静态文件，并且支持串联操作

webpack 是以 commonJS 的形式来书写脚本滴，但对 AMD/CMD 的支持也很全面，方便旧项目进行代码迁移。

webpack 具有 requireJs 和 browserify 的功能，但仍有很多自己的新特性：

1.对 CommonJS、AMD、ES6 的语法做了兼容

2.对 js、css、图片等资源文件都支持打包

3.串联式模块加载器以及插件机制，让其具有更好的灵活性和扩展性，例如提供对 CoffeeScript、ES6 的支持

4.有独立的配置文件 webpack.config.js

5.可以将代码切割成不同的 chunk，实现按需加载，降低了初始化时间

6.支持 SourceUrls 和 SourceMaps，易于调试

7.具有强大的 Plugin 接口，大多是内部插件，使用起来比较灵活

8.webpack 使用异步 IO 并具有多级缓存。这使得 webpack 很快且在增量编译上更加快数据可视化理论知识

计算机图形图像：图形学、图像处理

数学：高等数学、线性代数、数值计算

统计分析：数据处理、统计学

可视理论：数据可视化基本原理和方法

交互理论：认知心理学、工程实践

计算机编程：JavaScript 编程基础、Web 技能 Canvas、SVG、WebGL

百度的 echarts 相对来说使用 Canvas 是好的，Web 安全。Web 安全主要分为客户端脚本安全和服务器端应用安全。

客户端脚本安全主要包括浏览器安全、跨站脚本攻击(XSS)、跨站点请求伪造(CSRF)、点击劫持(ClickJacking)、HTML5 安全；

服务端安全主要包括

注入攻击：SQL 注入、XML 注入、代码注入、CRLF 注入等文件上传漏洞：如 FCKEditor 上传、绕过文件上传检查功能；

PHP 安全：一般会含漏洞、变量覆盖漏洞、代码执行漏洞；

认证与会话管理、访问控制、加密算法与随机数、Web 框架安全、应用层拒绝服务攻击、WebServer 配置安全等

浏览器端主要涉及：同源策略、沙箱等常见 web 安全及防护原理

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-sitescripting)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

首先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 htmltag 弄出来。这一个层面做好，至少可以堵住超过一半的 XSS 攻击。

首先，避免直接在 cookie 中泄露用户隐私，例如 email、密码等等。

其次，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。

如果网站不需要再浏览器端对 cookie 进行操作，可以在 Set-Cookie 末尾加上 HttpOnly 来防止 javascript 代码直接获取 cookie。

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：

登录受信任网站 A，并在本地生成 Cookie。

在不登出 A 的情况下，访问危险网站 B。

CSRF 的防御

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。

通过验证码的方法综合谈谈你对重构的理解

A:应聘者答

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

对于传统的网站来说重构通常是：

表格(table)布局改为 DIV+CSS

使网站前端兼容于现代浏览器(针对于不合规范的 CSS、如对 IE6 有效的)

对于移动平台的优化

针对于 SEO 进行优化

深层次的网站重构应该考虑的方面

减少代码间的耦合

让代码保持弹性

严格按规范编写代码

设计可扩展的 API

代替旧有的框架、语言(如 VB)

增强用户体验

通常来说对于速度的优化也包含在重构中

压缩 JS、CSS、image 等前端资源(通常是由服务器来解决)

程序的性能优化(如数据读写)

采用 CDN 来加速资源加载

对于 JSDOM 的优化

HTTP 服务器的文件缓存

你觉得前端工程的价值体现在哪

为简化用户使用提供技术支持（交互部分）

为多个浏览器兼容性提供支持

为提高用户浏览速度（浏览器性能）提供支持

为跨平台或者其他基于 webkit 或其他渲染引擎的应用提供支持

为展示数据提供支持（数据接口）

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。

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，

代码层面的优化

用 hash-table 来优化查找

少用全局变量

用 innerHTML 代替 DOM 操作，减少 DOM 操作次数，优化 javascript 性能

用 setTimeout 来避免页面失去响应

缓存 DOM 节点查找的结果

避免使用 CSSExpression

避免全局查询

避免使用 with(with 会创建自己的作用域，会增加作用域链长度)

多个变量声明合并

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

尽量避免写在 HTML 标签中写 Style 属性

移动端性能优化

尽量使用 css3 动画，开启硬件加速。

适当使用 touch 事件代替 click 事件。

避免使用 css3 渐变阴影效果。

可以用 transform:translateZ(0)来开启硬件加速。

不滥用 Float。Float 在渲染时计算量比较大，尽量减少使用

不滥用 Web 字体。Web 字体需要下载，解析，重绘当前页面，尽量减少使用。

合理使用 requestAnimationFrame 动画代替 setTimeout

CSS 中的属性(CSS3transitions、CSS3Dtransforms、Opacity、Canvas、WebGL、Video) 会触发 GPU 渲染，请合理使用。过渡使用会引发手机过耗电增加

PC 端的在移动端同样适用

对前端模块化的认识

AMD 是 RequireJS 在推广过程中对模块定义的规范化产出。

CMD 是 SeaJS 在推广过程中对模块定义的规范化产出。

AMD 是提前执行，CMD 是延迟执行。

AMD 推荐的风格通过返回一个对象做为模块对象，CommonJS 的风格通过对 module.exports 或 exports 的属性赋值来达到暴露模块对象的目的。

CMD 模块方式

```
define(function(require,exports,module){  
  //模块代码});
```

第 8 部分：典型开发综合 2 篇

191. 面试题

Q: 面试官问

call 和 apply 的区别

A: 应聘者答

call 方法:

语法: call([thisObj[,arg1[,arg2[,[,argN]]]])

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。

192. 面试题

Q: 面试官问

apply 方法:

A: 应聘者答

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。

实例：

```
function add(a,b){
  alert(a+b);
}
function sub(a,b){
  alert(a-b);
}
```

```
add.call(sub,3,1);
```

这个例子中的意思就是用 `add` 来替换 `sub`，`add.call(sub,3,1)===add(3,1)`，所以运行结果为：`alert(4)`；//注意：js 中的函数其实是对象，函数名是对 `Function` 对象的引用。

```
function Animal(){
  this.name="Animal";
  this.showName=function(){
    alert(this.name);
  }
}
function Cat(){
  this.name="Cat";
}
var animal=new Animal();
var cat=new Cat();
```

//通过 `call` 或 `apply` 方法，将原本属于 `Animal` 对象的 `showName()` 方法交给对象 `cat` 来使用了。

```
//输入结果为"Cat"
animal.showName.call(cat,"");
//animal.showName.apply(cat,[]);
```

`call` 的意思是把 `animal` 的方法放到 `cat` 上执行，原来 `cat` 是没有 `showName()` 方法，现在是把 `animal` 的 `showName()` 方法放到 `cat` 上来执行，所以 `this.name` 应该是 `Cat`

实现继承

```
function Animal(name){
  this.name=name;
  this.showName=function(){
    alert(this.name);
  }
}
function Cat(name){
  Animal.call(this,name);
}
var cat=new Cat("BlackCat");
cat.showName();
```

`Animal.call(this)`的意思就是使用 `Animal` 对象代替 `this` 对象,那么 `Cat` 中不就有 `Animal` 的所有属性和方法了吗, `Cat` 对象就能够直接调用 `Animal` 的方法以及属性了.

概括: `call` 和 `apply` 的作用基本相同,都可以改变一个函数的对象上下文,都可以实现对象的继承,唯一不同的地方在于调用方式, `apply` 需要接收一个数组或者参数列表作为参数

193. 面试题

Q: 面试官问

prototype 是什么

A: 应聘者答

每个函数都有一个 `prototype` 属性,这个属性是指向一个对象的引用,这个对象称为原型对象,原型对象包含函数实例共享的方法和属性,也就是说将函数用作构造函数调用(使用 `new` 操作符调用)的时候,新创建的对象会从原型对象上继承属性和方法。

数组有多少种属性方法

`push`,`pop`,`unshift`,`shift`,`concat`,`slice`,`reverse`,`sort`,`join`,`indexOf`,`splice`

194. 面试题

Q: 面试官问

JQ 中 `extent` 的用法

A: 应聘者答

`display` 有几种属性

`none`,`block`,`inline`,`inline-block`,`list-item`,`run-in`,`table`,`inline-table`,`table-row-group`,`table-header-group`,`table-footer-group`,`table-row`,`table-column-group`,`table-column`,`table-cell`,`table-caption`,`inherit`

195. 面试题

Q: 面试官问

css 显示隐藏有几种方法

A:应聘者答

`display:none,visibility:hidden`(所以它只会导致浏览器重绘而不会重排),`opacity:0`(只是透明度为 0 达到隐藏效果, 作用上并未隐藏)

196. 面试题

Q:面试官问

定义函数的方法有几种, 有什么不同

A:应聘者答

```
function func1(){}  
var func2=function(){};  
var func3=function func4(){};  
var func5=new Function();
```

第一种为最常见的定义方式—函数声明, 第二种为将一匿名函数赋值给一个变量, 第三种为将 `func4` 赋予变量 `func3`, 第四种为声明 `func5` 为一个对象

由于在代码执行之前, 解释器会先读取函数的声明, 所以声明前调用函数并不会产生语法错误, 这个特征叫做函数声明提升。

197. 面试题

Q:面试官问

HTML 的加载顺序

A:应聘者答

IE 下载的顺序是从上到下, 渲染的顺序也是从上到下, 下载和渲染是同时进行的。

在渲染到页面的某一部分时, 其上面的所有部分都已经下载完成 (并不是说所有相关联的元素都已经下载完)

如果遇到语义解释性的标签嵌入文件 (JS 脚本, CSS 样式), 那么此时 IE 的下载过程会启用单独连接进行下载。

并且在下载后进行解析, 解析过程中, 停止页面所有往下元素的下载都会阻塞加载

样式表在下载完成后, 将和以前下载的所有样式表一起进行解析, 解析完成后, 将对此前所有元素 (含以前已经渲染的) 重新进行渲染。

JS、CSS 中如有重定义, 后定义函数将覆盖前定义函数

198. 面试题

Q: 面试官问

this 是什么

A: 应聘者答

this 是 Javascript 语言的一个关键字。

它代表函数运行时，自动生成的一个内部对象，只能在函数内部使用。

随着函数使用场合的不同，this 的值会发生变化。但是有一个总的原则，那就是 this 指的是，调用函数的那个对象。

199. 面试题

Q: 面试官问

JQ 的 bind 和 on 有什么不同

A: 应聘者答

bind(), on(), live(), delegate() 都可以向匹配的元素添加一个或者多个事件处理器
bind() 适用于所有版本，live() 适用于 JQ1.8+，delegate 适用于 1.4.2+，on 适用于 1.7+
JQ1.7+ 以后官方推荐使用 on 来替代 bind();
bind 只能对已存在的元素进行绑定，on 可以对未来元素进行绑定
on 比 bind 多一个参数可以过滤指定元素

200. 面试题

Q: 面试官问

请写出下列代码的执行结果

A: 应聘者答

```
function add(a, b) {  
  alert(a + b);  
}
```

```

    }
    function sub(a,b){
        alert(a-b);
    }
    add.call(sub,3,1);
    var value="globalvar";
    function mFunc(){
        this.value="membervar";
    }
    function gFunc(){
        alert(this.value);
    }
    window.gFunc();
    gFunc.call(window);
    gFunc.call(new mFunc());
    输出: 4;
    输出: globalvar
    globalvar
    membervar

```

201. 面试题

Q: 面试官问

请解释 javascript 中 `a===b` 语句是什么意思

A: 应聘者答

“`===`”叫做严格运算符，“`==`”叫做相等运算符。

严格运算符的运算规则如下，

- (1) 不同类型值如果两个值的类型不同，直接返回 `false`。
- (2) 同一类的原始类型值同一类型的原始类型的值（数值、字符串、布尔值）比较时，值相同就返回 `true`，值不同就返回 `false`。
- (3) 同一类的复合类型值两个复合类型（对象、数组、函数）的数据比较时，不是比较它们的值是否相等，而是比较它们是否指向同一个对象。
- (4) `undefined` 和 `null` 与自身严格相等。`null===null//true`

202. 面试题

Q: 面试官问

请解释 IE 浏览器下 innerHTML 和 outerHTML

A: 应聘者答

innerHTML: 从对象的起始位置到终止位置的全部内容, 不包括 Html 标签。

outerHTML: 除了包含 innerHTML 的全部内容外, 还包含对象标签本身。

203. 面试题

Q: 面试官问

请解释 http 请求中以下的返回码

A: 应聘者答

200, 400, 403, 404, 500, 503

200 请求已成功, 请求所希望的响应头或数据体将随此响应返回。

400 语义有误, 当前请求无法被服务器理解或请求参数有误。

403 服务器已经理解请求, 但是拒绝执行它。

500 服务器遇到了一个未曾预料的状态, 导致了它无法完成对请求的处理。

503 由于临时的服务器维护或者过载, 服务器当前无法处理请求。

204. 面试题

Q: 面试官问

请写出下列 javascript 关键字说明和用法

A: 应聘者答

typeof, undefined, null, prototype, string, split, parentElement, getElementsByName, rowIndex, nextSibling, hasChildNodes

typeof 返回一个用来表示表达式的数据类型的字符串。
"number", "string", "boolean", "object", "function", 和 "undefined"

undefined 用于存放 JavaScript 的 undefined 值, 表示一个未声明的变量, 或已声明但没有赋值的变量, 或一个并不存在的对象属性。

null 表示没有对象(null 转为数值时为 0, 而 undefined 转为数值时为 NaN)

<http://www.ruanyifeng.com/blog/2014/03/undefined-vs-null.html>

prototype 每个函数都有一个属性, 这个属性是指向一个原型对象, 原型对象包含函数实例共享的方法和属性, 用作构造函数调用的时候, 新创建的对象会继承属性和方法。

string.split 用于把字符串分割为数组

parentElement 获取对象层次中的父对象

getElementsByName 可返回带有指定 name 的对象的集合

rowIndex 返回某一行在表格的行集合中的位置

nextSibling 返回指定节点之后紧跟的节点, 在相同的树层级中

hasChildNodes 检查元素是否有子节点

205. 面试题

Q: 面试官问

请解释 location.replace()和 location.href 的区别

A: 应聘者答

href 相当于打开一个新页面, replace 相当于替换当前页面

location.replace()不在浏览器中保存跳转前的网址, 因此按返回键将无效

206. 面试题

Q: 面试官问

请讲解 javascript 闭包的实现方法及作用

A: 应聘者答

闭包是有权访问另一个函数作用域的变量的函数

简单的说, Javascript 允许使用内部函数—即函数定义和函数表达式位于另一个函数的函数体内。而且, 这些内部函数可以访问它们所在的外部函数中声明的所有局部变量、参数和声明的其他内部函数。当其中一个这样的内部函数在包含它们的外部函数之外被调用时, 就会形成闭包。

闭包可以用在许多地方。它的最大用处有两个:一个是读取函数内部的变量, 另一个就是让这些变量的值始终保持在内存中。

示例:

```
functiona(){  
  varn=123456;
```

```

return function(){
  console.info(n);
}
}
var result = a();
for 循环中的闭包问题
function box(){
  var arr = [];
  for(var i = 0; i < 5; i++){
    arr[i] = function(){ return i; }
  }
  return arr;
}

```

在 for 循环里面的匿名函数执行 `return i` 语句的时候，由于匿名函数里面没有 `i` 这个变量，所以这个 `i` 他要从父级函数中寻找 `i`，而父级函数中的 `i` 在 for 循环中，当找到这个 `i` 的时候，是 for 循环完毕的 `i`，也就是 5，所以这个 `box` 得到的是一个数组 `[5,5,5,5,5]`;

解决方案-通过匿名函数自执行传参：

```

function box(){
  var arr = [];
  for(var i = 0; i < 5; i++){
    arr[i] = (function(num){ return num; })(i)
  }
  return arr;
}

```

207. 面试题

Q: 面试官问

请用一行代码 javascript 实现以下内容

A: 应聘者答

将字符串 `dsadsa214fsd` 变成 `dsadsa428fsd`

将字符串 `dsadsa214fsd` 变成 `dsadsa"2""1""4"fsd`

注：字符串中数字位置不确定

```
“dsadsa214fsd”.replace(/d/g,function(num){return num*2});
```

```
“dsadsa214fsd”.replace(/d/g,function(num){return “”+num+””});
```

208. 面试题

Q: 面试官问

清除浮动的方式有哪几种

A: 应聘者答

父级 div 定义 height, 解决父级 div 无法自动获取高度的问题

结尾处加空 div 标签, 并使用样式 clear:both

父级 div 定义伪类:after 和 zoom

父级 div 定义 overflow:hidden

父级 div 定义 overflow:auto

父级 div 也一起浮动

父级 div 定义 display:table

结尾处加 br 标签 clear:both

JS 中使用 typeof 返回的类型有哪些

string,number,function,object,boolean,undefined

209. 面试题

Q: 面试官问

JS 的事件委托原理是什么? 常用于什么场景下

A: 应聘者答

利用冒泡的原理, 把事件加到父级上, 触发执行效果

JS 的继承方式有几种?

对象冒充

call 方法

apply 方法

原型链方法

实例继承

枚举拷贝继承

示例: functionParent(test){

//基础对象

this.name=test;

this.say=function(){

console.log(this.name);

}

}

```

functionGet(test,test2){
//对象冒充
this.method=Parent;
this.method(test);
deletethis.method;
this.test2=test2;
this.say2=function(){
console.info(this.test2);
}
}
varparent=newParent('test');
varget=newGet('gettest','getpwd');
parent.say();
get.say();
get.say2();
functionGetCall(getcallnum){
//call 方法
Parent.call(this,getcallnum);
}
vargetcall=newGetCall('getcall');
getcall.say();
functionGetApply(getapplynum){
//apply 方法
Parent.apply(this,arguments);
}
vargetapply=newGetApply('getapply');
getapply.say();
Parent.prototype={
//原形链方法
num:123456,
sayNum:function(){
console.log(this.num);
}
}
functionChild(){ }
Child.prototype=newParent();
varc=newChild();
c.sayNum();
functionObjTo(objnum){
//原型链方法
varobj=newParent(objnum);
returnobj;
}
varo=newObjTo('objto');

```

```

o.say();
function copyTo(copytonum){
//枚举拷贝继承
var parent=new Parent(copytonum);
for(i in parent){
copyTo.prototype[i]=parent[i];
}
}
var c=new copyTo('copytonum');
c.say();

```

typeof、instanceof、Object.prototype.toString 这几种方式的作用和区别？

typeof: 在 JavaScript 里使用 typeof 来判断数据类型，只能区分基本类型，即“number”，“string”，“undefined”，“boolean”，“object”五种。

instanceof: 左操作数是一个对象，右操作数是标识对象的类，如果左侧的对象是右侧类的实例，则表达式返回 true，否则返回 false(可以区分 array 类)

Object.prototype.toString()返回[class],我们可以通过它获取对象的类型信息

210. 面试题

Q: 面试官问

JS 如何实现类似后台语言中的 private、public、protected、static 变量定义？

A: 应聘者答

```

静态属性，静态类: var staticClass={id:5,say:function(){console.info('abc')}}
公      有      属      性      ,      方      法      :
function PublicClass={this.name='abc',this.say:function(){console.info(this.name)}}
私      有      属      性      ,      方
法: function PrivateClass={var name='abc',var say=function(){console.info('abc')}}

```

211. 面试题

Q: 面试官问

如何解决移动端的 tap 事件点击穿透问题？

A: 应聘者答

起因: 移动端浏览器会在 touch 事件发生后 300ms 左右触发 click 方法

解决：不要混用 touch 和 click 方法

212. 面试题

Q:面试官问

将一个 32 位十进制数组转化成 2 进制后，翻转这个 2 进制数，再转换成十进制数

A:应聘者答

```
//res1
functionreverseB(num){
  varbits=num.toString(2);
  varlen=32-bits.length;
  bits=bits.split("").reverse().join("");
  while(len--){
    bits+="0";
  }
  returnparseInt(bits,2);
}
console.log(reverseB(43261596));//964176192
//res2
functionreverseBs(n){
  varm=0;
  for(vari=0;i<31;i++){
    m|=(n&1);
    n>>=1;
    m<<=1;
  }
  m|=(n&01);
  returnm;
}
console.log(reverseBs(43261596));//964176192
...
<!--more-->
```

213. 面试题

Q: 面试官问

罗马数组转换成十进制数字

A: 应聘者答

```
var romObj = {I:1,V:5,X:10,L:50,C:100,D:500,M:1000};
var romanToInt = function(str){
    var sum = 0;
    for(var i=0; i<str.length; i++){
        rom1 = romObj[str.substring(i,i+1)];
        rom2 = romObj[str.substring(i+1,i+2)];
        if(str.substring(i+1,i+2) !== "" && rom1 < rom2){
            sum += rom2 - rom1;
            i++;
        } else {
            sum += rom1;
        }
    }
    return sum;
};
console.log(romanToInt('II'));
// 2
```

214. 面试题

Q: 面试官问

总共有 n 阶楼梯，每次只能上一步或者两部，要到达楼上有多少种方式

A: 应聘者答

分析，当列举了当 n 为 1,2,3,4,5 时得出有 1,2,3,5,8 种方式
就是一个斐波拉契数列

```
var climbStairs = function(n){
    var a = 1,
        b = 2,
        sum = 1;
    if(n == 1) return a;
    if(n == 2) return b;
    n -= 2;
```

```

while(n--){
sum=a+b;
a=b;
b=sum;
}
returnsum;
};
console.log(climbStairs(3));
...

```

215. 面试题

Q: 面试官问

给数组增加一种方法，可以去除数组中的重复项并返回数组

A: 应聘者答

```

//fun1--hash
Array.prototype.delWeight=function()
{
varn=[],r=[];//n 为 hash 表, r 为临时数组
for(vari=0;i<this.length;i++)//遍历当前数组
{
if(!n[this[i]])//如果 hash 表中没有当前项
{
n[this[i]]=true;//存入 hash 表
r.push(this[i]);//把当前数组的当前项 push 到临时数组里面
}
}
returnr;
}
vararr=[1,1,1,1,1,2,2,3,3,4,5,6,5,6];
console.log(arr.delWeight());
//fun2--利用数组的 indexOf 方法
functiondelWeight(arr){
varresult=[];
for(vari=0;i<arr.length;i++){
if(result.indexOf(arr[i])===-1)result.push(arr[i]);
}
returnresult;
}

```

...

216. 面试题

Q: 面试官问

对不确定数量的数进行排序

A: 应聘者答

```
function mySort(){
  var tags = new Array(); // 使用数组作为参数存储容器
  for (var i = 0, len = arguments.length; i < len; i++) {
    tags.push(arguments[i]);
  }
  tags.sort(function(a, b) {
    return a - b;
  });
  return tags; // 返回已经排序的数组
}
var result = mySort(50, 11, 16, 32, 24, 99, 57, 100); // 传入参数个数不确定
console.info(result); // 显示结果
...
```

217. 面试题

Q: 面试官问

用短路求值的方式求前 n 项和

A: 应聘者答

```
function sumr(n) {
  var ans = n;
  ans && (ans += sumr(n - 1));
  return ans;
}
console.log(sumr(5));
...
```

218. 面试题

Q: 面试官问

正则表达式判断一个数是不是素数

A: 应聘者答

```
function is_prime(x){
  var r=[], s="";
  while(r.length<x){
    r.push("1");
  }
  s=r.join("");
  return !/^1?$|^(11+?)\1+$/.test(s);
}
console.log(is_prime(5));
console.log(is_prime(10));
console.log(is_prime(13));
console.log(is_prime(60));
console.log(is_prime(99));
...
```

219. 面试题

Q: 面试官问

辗转相除求最大公约数

A: 应聘者答

```
function ff(a,b){
  if(b>a) ff(b,a);
  return !(a%b)?b:ff(b,a%b);
}
console.log(ff(10,3));
...
```

220. 面试题

Q: 面试官问

Json 数据按照某一字段排序的方法

A: 应聘者答

```
varstuJson=[{ name:"daming",age:21,weight:66,sex:"boy"},
{ name:"lisa",age:19,weight:45,sex:"girl"},
{ name:"lili",age:20,weight:50,sex:"boy"}];
//按 age 升序
stuJson.sort(function(a,b){
returna.age-b.age;
});
console.log(stuJson);
...
```

221. 面试题

Q: 面试官问

求一个数组中的最大最小值

A: 应聘者答

```
vararr=[32,42,12,42,21,23,56,75,3,33,53,23,36];
varmin=Math.min.apply(null,arr);
console.log(min);
varaMax=Math.max.apply(null,arr);
console.log(aMax);
...
```

222. 面试题

Q: 面试官问

将一个匿名函数作为函数的一个参数

A:应聘者答

```
functionmap(f,a){
  varresult=[],//CreateanewArray
  i;
  for(i=0;i!=a.length;i++)
  result[i]=f(a[i]);
  returnresult;
}
//调用的时候实现这个引用参数函数
varre=map(function(x){returnx*x*x},{0,1,2,5,10});
console.log(re);//[0,1,8,125,1000].
...
```

223. 面试题

Q:面试官问

字符串中的大小写相互转换

A:应聘者答

```
functionupperToggleLower(str){
  vars="";
  for(vari=0,len=str.length;i<len;i++){
    if(str[i]<='z'&&str[i]>='a'||str[i]<='Z'&&str[i]>='A'){
      s+=String.fromCharCode(str.charCodeAt(i)^32);
    }else{
      s+=str[i];
    }
  }
  return;
}
varstr='abcdefFEDCBA';
console.log(upperToggleLower(str));//"ABCDEFfedcba"
...
```

224. 面试题

Q: 面试官问

js 快排

A: 应聘者答

```
function quick_sort(array, l, r) {  
    if (l < r) {  
        var i = l, j = r, x = array[l];  
        while (i < j) {  
            while (i < j && array[j] >= x) {  
                j--;  
            }  
            if (i < j) {  
                array[i++] = array[j];  
            }  
            while (i < j && array[i] < x) {  
                i++;  
            }  
            if (i < j) {  
                array[j--] = array[i];  
            }  
        }  
        array[i] = x;  
        quick_sort(array, l, i - 1);  
        quick_sort(array, i + 1, r);  
    }  
    return array;  
}
```

225. 面试题

Q: 面试官问

给出 $2 * n + 1$ 个的数字，除其中一个数字之外其他每个数字均出现两次，找到这个数字

A:应聘者答

```
/*
 * @param A: Array of integers.
 * @return: The single number.
 */
int singleNumber(vector<int> &A) {
    if (!A.size()) return 0;
    int a = A[0];
    for (int i = 1; i < A.size(); i++)
    {
        a ^= A[i];
    }
    return a;
}
```

226. 面试题

Q: 面试官问

给出两个整数 a 和 b, 求他们的和, 但不能使用 + 等数学运算符。

A: 应聘者答

```
/*
 * @param a: The first integer
 * @param b: The second integer
 * @return: The sum of a and b
 */
int plus(int a, int b) {
    int n1;
    do {
        n1 = a ^ b;
        b = (a & b) << 1;
        a = n1;
    } while (b != 0);
    return n1;
}
```

227. 面试题

Q: 面试官问

计算在一个 32 位的整数的二进制表式中有多少个 1.

A: 应聘者答

```
/*
 * @param num: an integer
 * @return: an integer, the number of ones in num
 */
int countOnes(int num){
    int count=0;
    while(num!=0)
    {
        num=num&(num-1);
        count++;
    }
    return count;
}
```

228. 面试题

Q: 面试官问

找到单链表倒数第 n 个节点，保证链表中节点的最少数量为 n 。

A: 应聘者答

```
/**
 * Definition of ListNode
 * class ListNode{
 * public:
 * int val;
 * ListNode* next;
 * ListNode(int val){
 * this->val=val;
 * this->next=NULL;
 * }
 * }
 * @param head: The first node of linked list.
```

```

*@param n: An integer.
*@return: Nth to last node of a singly linked list.
*/
ListNode* nthToLast(ListNode* head, int n) {
    if (!head) return NULL;
    ListNode* h, *d;
    h = d = head;
    while (--n)
    {
        h = h->next;
    }
    while (h->next != NULL) {
        h = h->next;
        d = d->next;
    }
    return d;
}

```

229. 面试题

Q: 面试官问

写出一个高效的算法来搜索 $m \times n$ 矩阵中的值。

A: 应聘者答

```

/**
 * 这个矩阵具有以下特性：
 * 1. 每行中的整数从左到右是排序的。
 * 2. 每行的第一个数大于上一行的最后一个整数。
 * @param matrix, a list of list of integers
 * @param target, an integer
 * @return a boolean, indicate whether matrix contain target
 */
bool searchMatrix(vector<vector<int>>& matrix, int target) {
    if (!matrix.size()) return false;
    int i, j;
    for (i = 0; i < matrix.size(); i++)
    {
        if (target < matrix[i][0])
            break;
        if (target == matrix[i][0])

```

```

return true;
}
i--;
for(j=0;j<matrix[i].size();j++)
if(target==matrix[i][j])return true;
return false;
}

```

230. 面试题

Q: 面试官问

设计一种方法，将一个字符串中的所有空格替换成%20。

A: 应聘者答

```

/**
 * 你可以假设该字符串有足够的空间来加入新的字符，且你得到的是“真实的”字符长
度
 * @param string: An array of Char
 * @param length: The true length of the string
 * @return: The true length of the new string
 */
int replaceBlank(char string[], int length){
// Write your code here
int space_cnt=0;
int new_length=0;
char *p,*q;
for(int i=0;i<length;i++)
{
if(string[i]==' ')space_cnt++;
}
new_length=space_cnt*2+length;
p=string+(length-1);
q=string+(new_length-1);
while(p!=q)
{
if(*p=="")*q='0',q--,*q='2',q--,*q='%';
else*q=*p;
p--;
q--;
}
}

```

```
return new_length;
}
```

231. 面试题

Q: 面试官问

统计一个字符串中每个单词的个数

A: 应聘者答

```
/**
 * @param {string} str 输入的字符串
 * @return {Object} o 单词及对应数量
 */
function countWordNum(str) {
  if (!str) return;
  var wordsArr = str.split("");
  var o = {};
  for (var i = 0, len = wordsArr.length; i < len; i++) {
    if (wordsArr[i] in o) { // or if (o[wordsArr[i]]) { or if (o.hasOwnProperty(wordsArr[i])) {
      o[wordsArr[i]]++;
    }
    else {
      o[wordsArr[i]] = 1;
    }
  }
  return o;
}
```

232. 面试题

Q: 面试官问

给你六种面额 1、5、10、20、50、100 元的纸币，假设每种币值的数量都足够多，编写程序求组成 N 元（N 为 0~10000 的非负整数）的不同组合的个数

A: 应聘者答

```
public class changeDP {
```

```

staticintv[]={1,5,10,20,50,100};
publicstaticintchange(intn,inti){
    if(n<0)return0;
    if(n==0)return1;
    if(i<0)return0;
    returnchange(n,i-1)+change(n-v[i],i);
}
publicstaticvoidmain(String[]args){
    System.out.println(change(100,v.length-1));
}
}

```

233. 面试题

Q: 面试官问

给定非空整型数组 `arr` 和整数 `limit`，两次从 `arr` 中随机抽取元素（可能抽到同一个元素），获得整数 `x,y`，得到和 `s=x+y`。求所有不超过 `limit` 的 `s` 值中的最大数。

A: 应聘者答

```

functionlimitedMaxSum(arr,limit){
    varmax=0;
    for(vari=0,len=arr.length;i<len;i++){
        for(varj=0;j<len;j++){
            varsum=arr[i]+arr[j];
            if(sum<=limit&&sum>=max){
                max=sum;
            }
        }
    }
    returnmax;
}

```

234. 面试题

Q: 面试官问

`document.write` 和 `innerHTML` 的区别

A:应聘者答

`document.write` 只能重绘整个页面
`innerHTML` 可以重绘页面的一部分

235. 面试题

Q:面试官问

`.call()`和`.apply()`的区别和作用？

A:应聘者答

作用：动态改变某个类的某个方法的运行环境。

236. 面试题

Q:面试官问

哪些操作会造成内存泄漏？

A:应聘者答

内存泄漏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）

237. 面试题

Q:面试官问

如何判断当前脚本运行在浏览器还是 `node` 环境中？

A:应聘者答

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

238. 面试题

Q:面试官问

什么叫优雅降级和渐进增强？

A:应聘者答

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。

239. 面试题

Q:面试官问

WEB 应用从服务器主动推送 Data 到客户端有那些方式？

A:应聘者答

Javascript 数据推送

Commet：基于 HTTP 长连接的服务器推送技术

基于 WebSocket 的推送方案

SSE (Server-SendEvent)：服务器推送数据新方式

240. 面试题

Q:面试官问

对前端界面工程师这个职位是怎么样理解的？它的前景会怎么样？

A:应聘者答

前端是最贴近用户的程序员，比后端、数据库、产品经理、运营、安全都近。

- 1、实现界面交互
- 2、提升用户体验
- 3、有了 Node.js，前端可以实现服务端的一些事情

前端是最贴近用户的程序员，前端的能力就是能让产品从 90 分进化到 100 分，甚至更好，

参与项目，快速高质量完成实现效果图，精确到 1px；
与团队成员，UI 设计，产品经理的沟通；
做好的页面结构，页面重构和用户体验；
处理 hack，兼容、写出优美的代码格式；
针对服务器的优化、拥抱最新前端技术。

241. 面试题

Q:面试官问

你有哪些性能优化的方法？

A:应聘者答

(1) 减少 http 请求次数：CSSSprites,JS、CSS 源码压缩、图片大小控制合适；网页 Gzip，CDN 托管，data 缓存，图片服务器。

(2) 前端模板 JS+数据，减少由于 HTML 标签导致的带宽浪费，前端用变量保存 AJAX 请求结果，每次操作本地变量，不用请求，减少请求次数

(3) 用 innerHTML 代替 DOM 操作，减少 DOM 操作次数，优化 javascript 性能。

(4) 当需要设置的样式很多时设置 className 而不是直接操作 style。

(5) 少用全局变量、缓存 DOM 节点查找的结果。减少 IO 读取操作。

(6) 避免使用 CSSExpression (css 表达式)又称 Dynamicproperties(动态属性)。

(7) 图片预加载，将样式表放在顶部，将脚本放在底部加上时间戳。

242. 面试题

Q:面试官问

一个页面从输入 URL 到页面加载显示完成，这个过程中都发生了什么？

A:应聘者答

分为 4 个步骤：

(1)，当发送一个 URL 请求时，不管这个 URL 是 Web 页面的 URL 还是 Web 页面上每个资源的 URL，浏览器都会开启一个线程来处理这个请求，同时在远程 DNS 服务器上启动一个 DNS 查询。这能使浏览器获得请求对应的 IP 地址。

(2)，浏览器与远程 Web 服务器通过 TCP 三次握手协商来建立一个 TCP/IP 连接。该握手包括一个同步报文，一个同步-应答报文和一个应答报文，这三个报文在浏览器和服务器之间传递。该握手首先由客户端尝试建立起通信，而后服务器应答并接受客户端的请求，最后由客户端发出该请求已经被接受的报文。

(3)，一旦 TCP/IP 连接建立，浏览器会通过该连接向远程服务器发送 HTTP 的 GET 请求。远程服务器找到资源并使用 HTTP 响应返回该资源，值为 200 的 HTTP 响应状态表示一个正确的响应。

(4)，此时，Web 服务器提供资源服务，客户端开始下载资源。

请求返回后，便进入了我们关注的前端模块

简单来说，浏览器会解析 HTML 生成 DOMTree，其次会根据 CSS 生成 CSSRuleTree，而 javascript 又可以根据 DOMAPI 操作 DOM

243. 面试题

Q:面试官问

平时如何管理你的项目？

A:应聘者答

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

编写习惯必须一致（例如都是采用继承式的写法，单样式都写成一行）；

标注样式编写人，各模块都及时标注（标注关键样式调用的地方）；

页面进行标注（例如页面模块开始和结束）；

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

JS 分文件夹存放命名以该 JS 功能为准的英文翻译。

图片采用整合的 images.pngpng8 格式文件使用尽量整合在一起使用方便将来的管理

244. 面试题

Q:面试官问

说说最近最流行的一些东西吧？常去哪些网站？

A:应聘者答

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

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

245. 面试题

Q:面试官问

javascript 对象的几种创建方式

A:应聘者答

- 1, 工厂模式
- 2, 构造函数模式
- 3, 原型模式
- 4, 混合构造函数和原型模式
- 5, 动态原型模式
- 6, 寄生构造函数模式
- 7, 稳妥构造函数模式

246. 面试题

Q:面试官问

javascript 继承的 6 种方法

A:应聘者答

- 1, 原型链继承
- 2, 借用构造函数继承
- 3, 组合继承(原型+借用构造)
- 4, 原型式继承
- 5, 寄生式继承

6, 寄生组合式继承

247. 面试题

Q: 面试官问

ajax 过程

A: 应聘者答

- (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.

248. 面试题

Q: 面试官问

异步加载和延迟加载

A: 应聘者答

1. 异步加载的方案: 动态插入 script 标签
2. 通过 ajax 去获取 js 代码, 然后通过 eval 执行
3. script 标签上添加 defer 或者 async 属性
4. 创建并插入 iframe, 让它异步执行 js
5. 延迟加载: 有些 js 代码并不是页面初始化的时候就立刻需要的, 而稍后的某些情况才需要的。

249. 面试题

Q: 面试官问

前端安全问题?

A:应聘者答

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。

250. 面试题

Q:面试官问

XSS 原理及防范

A:应聘者答

Xss(cross-sitescripting)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。

251. 面试题

Q:面试官问

XSS 防范方法

A:应聘者答

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 htmltag 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

252. 面试题

Q:面试官问

XSS 与 CSRF 有什么区别吗？

A:应聘者答

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。

253. 面试题

Q:面试官问

CSRF 的防御

A:应聘者答

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

254. 面试题

Q: 面试官问

ie 各版本和 chrome 可以并行下载多少个资源

A: 应聘者答

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

255. 面试题

Q: 面试官问

javascript 里面的继承怎么实现，如何避免原型链上面的对象共享

A: 应聘者答

用构造函数和原型链的混合模式去实现继承，避免对象共享可以参考经典的 extend() 函数，很多前端框架都有封装的，就是用一个空函数当做中间变量

256. 面试题

Q: 面试官问

grunt，YUIcompressor 和 googleclosure 用来进行代码压缩的用法。

A: 应聘者答

YUICompressor 是一个用来压缩 JS 和 CSS 文件的工具，采用 Java 开发。

使用方法：

//压缩 JS

java-jaryuicompressor-2.4.2.jar--typejs--charsetutf-8-vsrc.js>packed.js

//压缩 CSS

java-jaryuicompressor-2.4.2.jar--typecss--charsetutf-8-vsrc.css>packed.css

257. 面试题

Q: 面试官问

Flash、Ajax 各自的优缺点，在使用中如何取舍？

A: 应聘者答

1、Flashjax 对比

Flash 适合处理多媒体、矢量图形、访问机器；对 CSS、处理文本上不足，不容易被搜索。

Ajax 对 CSS、文本支持很好，支持搜索；多媒体、矢量图形、机器访问不足。

共同点：与服务器的无刷新传递消息、用户离线和在线状态、操作 DOM

258. 面试题

Q: 面试官问

请解释一下 JavaScript 的同源策略。

A: 应聘者答

概念:同源策略是客户端脚本（尤其是 Javascript）的重要的安全度量标准。它最早出自 NetscapeNavigator2.0，其目的是防止某个文档或脚本从多个不同源装载。

这里的同源策略指的是：协议，域名，端口相同，同源策略是一种安全协议。

指一段脚本只能读取来自同一起来源的窗口和文档的属性。

259. 面试题

Q: 面试官问

为什么要有同源限制？

A:应聘者答

我们举例说明：比如一个黑客程序，他利用 Iframe 把真正的银行登录页面嵌到他的页面上，当你使用真实的用户名，密码登录时，他的页面就可以通过 Javascript 读取到你的表单中 input 中的内容，这样用户名，密码就轻松到手了。

260. 面试题

Q:面试官问

什么是“usestrict”;?使用它的好处和坏处分别是什么？

A:应聘者答

ECMAScript5 添加了第二种运行模式：“**严格模式**”（strictmode）。顾名思义，这种模式使得 Javascript 在更严格的条件下运行。

设立“**严格模式**”的目的，主要有以下几个：

消除 Javascript 语法的一些不合理、不严谨之处，减少一些怪异行为；

消除代码运行的一些不安全之处，保证代码运行的安全；

提高编译器效率，增加运行速度；

为未来新版本的 Javascript 做好铺垫。

注：经过测试 IE6,7,8,9 均不支持**严格模式**。

缺点：

现在网站的 JS 都会进行压缩，一些文件用了**严格模式**，而另一些没有。这时这些本来是**严格模式**的文件，被 merge 后，这个串就到了文件的中间，不仅没有指示**严格模式**，反而在压缩后浪费了字节。

261. 面试题

Q:面试官问

GET 和 POST 的区别，何时使用 POST？

A:应聘者答

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

POST：一般用于修改服务器上的资源，对所发送的信息没有限制。

A:应聘者答

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

也就是说 Get 是通过地址栏来传值，而 Post 是通过提交表单来传值。

然而，在以下情况中，请使用 POST 请求：

无法使用缓存文件（更新服务器上的文件或数据库）

向服务器发送大量数据（POST 没有数据量限制）

发送包含未知字符的用户输入时，POST 比 GET 更稳定也更可靠

262. 面试题

Q:面试官问

哪些地方会出现 css 阻塞，哪些地方会出现 js 阻塞？

A:应聘者答

s 的阻塞特性：所有浏览器在下载 JS 的时候，会阻止一切其他活动，比如其他资源的下载，内容的呈现等等。直到 JS 下载、解析、执行完毕后才开始继续并行下载其他资源并呈现内容。为了提高用户体验，新一代浏览器都支持并行下载 JS，但是 JS 下载仍然会阻塞其它资源的下载（例如.图片，css 文件等）。

由于浏览器为了防止出现 JS 修改 DOM 树，需要重新构建 DOM 树的情况，所以就会阻塞其他的下载和呈现。

嵌入 JS 会阻塞所有内容的呈现，而外部 JS 只会阻塞其后内容的显示，2 种方式都会阻塞其后资源的下载。也就是说外部样式不会阻塞外部脚本的加载，但会阻塞外部脚本的执行。

CSS 怎么会阻塞加载了？CSS 本来是可以并行下载的，在什么情况下会出现阻塞加载了(在测试观察中，IE6 下 CSS 都是阻塞加载)

当 CSS 后面跟着嵌入的 JS 的时候，该 CSS 就会出现阻塞后面资源下载的情况。而当把嵌入 JS 放到 CSS 前面，就不会出现阻塞的情况了。

根本原因：因为浏览器会维持 html 中 css 和 js 的顺序，样式表必须在嵌入的 JS 执行前先加载、解析完。而嵌入的 JS 会阻塞后面的资源加载，所以就会出现上面 CSS 阻塞下载的情况。

嵌入 JS 应该放在什么位置？

1、放在底部，虽然放在底部照样会阻塞所有呈现，但不会阻塞资源下载。

2、如果嵌入 JS 放在 head 中，请把嵌入 JS 放在 CSS 头部。

3、使用 defer（只支持 IE）

4、不要在嵌入的 JS 中调用运行时间较长的函数，如果一定要用，可以用`setTimeout`来调用

263. 面试题

Q: 面试官问

Javascript 无阻塞加载具体方式

A: 应聘者答

将脚本放在底部。<link>还是放在 head 中，用以保证在 js 加载前，能加载出正常显示的页面。<script>标签放在</body>前。

成组脚本：由于每个<script>标签下载时阻塞页面解析过程，所以限制页面的<script>总数也可以改善性能。适用于内联脚本和外部脚本。

非阻塞脚本：等页面完成加载后，再加载 js 代码。也就是，在 window.onload 事件发出后开始下载代码。

(1) defer 属性：支持 IE4 和 Firefox3.5 更高版本浏览器

(2) 动态脚本元素：文档对象模型 (DOM) 允许你使用 js 动态创建 HTML 的几乎全部文档内容。代码如下：

```
<script>
var script=document.createElement("script");
script.type="text/javascript";
script.src="file.js";
document.getElementsByTagName("head")[0].appendChild(script);
</script>
```

此技术的重点在于：无论在何处启动下载，文件下载和运行都不会阻塞其他页面处理过程。即使在 head 里（除了用于下载文件的 http 链接）。

JavaScript 原型，原型链有什么特点？

A: 应聘者答

原型对象也是普通的对象，是对象一个自带隐式的 proto 属性，原型也有可能有自己的原型，如果一个原型对象的原型不为 null 的话，我们就称之为原型链。

原型链是由一些用来继承和共享属性的对象组成的（有限的）对象链。

264. 面试题

Q: 面试官问

事件、IE 与火狐的事件机制有什么区别？如何阻止冒泡？

A:应聘者答

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。

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。；

ev.stopPropagation();注意旧 ie 的方法 ev.cancelBubble=true;

265. 面试题

Q:面试官问

ajax 是什么?ajax 的交互模型?同步和异步的区别?如何解决跨域问题?

A:应聘者答

通过异步模式，提升了用户体验

优化了浏览器和服务器之间的传输，减少不必要的往返，减少了带宽占用

Ajax 在客户端运行，承担了一部分本来由服务器承担的工作，减少了大用户量下的服务器负载。

Ajax 的最大的特点是什么。

Ajax 可以实现动态不刷新（局部刷新）

readyState 属性状态有 5 个可取值：0=未初始化，1=启动 2=发送，3=接收，4=完成
ajax 的缺点

- 1、ajax 不支持浏览器 back 按钮。
- 2、安全问题 AJAX 暴露了与服务器交互的细节。
- 3、对搜索引擎的支持比较弱。
- 4、破坏了程序的异常机制。
- 5、不容易调试。

跨域：jsonp、iframe、window.name、window.postMessage、服务器上设置代理页面

266. 面试题

Q:面试官问

js 对象的深度克隆

A:应聘者答

```
function clone(Obj){
    var buf;
    if(Obj instanceof Array){
        buf=[];//创建一个空的数组
        var i=Obj.length;
        while(i--){
            buf[i]=clone(Obj[i]);
        }
        return buf;
    }else if(Obj instanceof Object){
        buf={}; //创建一个空对象
        for(var k in Obj){ //为这个对象添加新的属性
            buf[k]=clone(Obj[k]);
        }
        return buf;
    }else{
        return Obj;
    }
}
```

267. 面试题

Q:面试官问

网站重构的理解？

A:应聘者答

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

对于传统的网站来说重构通常是：

表格(table)布局改为 DIV+CSS

使网站前端兼容于现代浏览器(针对于不合规范的 CSS、如对 IE6 有效的)

对于移动平台的优化

针对于 SEO 进行优化

深层次的网站重构应该考虑的方面

减少代码间的耦合

让代码保持弹性

严格按规范编写代码
设计可扩展的 API
代替旧有的框架、语言(如 VB)
增强用户体验
通常来说对于速度的优化也包含在重构中
压缩 JS、CSS、image 等前端资源(通常是由服务器来解决)
程序的性能优化(如数据读写)
采用 CDN 来加速资源加载
对于 JSDOM 的优化
HTTP 服务器的文件缓存

268. 面试题

Q: 面试官问

如何获取 UA?

A: 应聘者答

```
<script>
function whatBrowser(){
  document.Browser.Name.value=navigator.appName;
  document.Browser.Version.value=navigator.appVersion;
  document.Browser.Code.value=navigator.appCodeName;
  document.Browser.Agent.value=navigator.userAgent;
}
</script>
```

269. 面试题

Q: 面试官问

js 数组去重

A: 应聘者答

以下是数组去重的三种方法：

```
Array.prototype.unique1=function(){
  var n=[]; //一个新的临时数组
  for(var i=0;i<this.length;i++) //遍历当前数组
```

```

{
//如果当前数组的第 i 已经保存进了临时数组，那么跳过，
//否则把当前项 push 到临时数组里面
if(n.indexOf(this[i])!=-1)n.push(this[i]);
}
returnn;
}
Array.prototype.unique2=function()
{
varn={},r=[];//n 为 hash 表， r 为临时数组
for(vari=0;i<this.length;i++)//遍历当前数组
{
if(!n[this[i]])//如果 hash 表中没有当前项
{
n[this[i]]=true;//存入 hash 表
r.push(this[i]);//把当前数组的当前项 push 到临时数组里面
}
}
returnr;
}
Array.prototype.unique3=function()
{
varn=[this[0]];//结果数组
for(vari=1;i<this.length;i++)//从第二项开始遍历
{
//如果当前数组的第 i 项在当前数组中第一次出现的位置不是 i，
//那么表示第 i 项是重复的，忽略掉。否则存入结果数组
if(this.indexOf(this[i])!=i)n.push(this[i]);
}
returnn;
}

```

270. 面试题

Q: 面试官问

HTTP 状态码

A: 应聘者答

100Continue 继续，一般在发送 post 请求时，已发送了 httpheader 之后服务端将返回此信息，表示确认，之后发送具体参数信息

200OK 正常返回信息
201Created 请求成功并且服务器创建了新的资源
202Accepted 服务器已接受请求，但尚未处理
301MovedPermanently 请求的网页已永久移动到新位置。
302Found 临时性重定向。
303SeeOther 临时性重定向，且总是使用 GET 请求新的 URI。
304NotModified 自从上次请求后，请求的网页未修改过。
400BadRequest 服务器无法理解请求的格式，客户端不应当尝试再次使用相同的内容发起请求。
401Unauthorized 请求未授权。
403Forbidden 禁止访问。
404NotFound 找不到如何与 URI 相匹配的资源。
500InternalServerError 最常见的服务器端错误。
503ServiceUnavailable 服务器端暂时无法处理请求（可能是过载或维护）。

271. 面试题

Q: 面试官问

js 操作获取和设置 cookie

A: 应聘者答

```
//创建 cookie
function setCookie(name,value,expires,path,domain,secure){
    var cookieText=encodeURIComponent(name)+'='+encodeURIComponent(value);
    if(expires instanceof Date){
        cookieText+=';expires='+expires;
    }
    if(path){
        cookieText+=';path='+path;
    }
    if(domain){
        cookieText+=';domain='+domain;
    }
    if(secure){
        cookieText+=';secure';
    }
    document.cookie=cookieText;
}

//获取 cookie
function getCookie(name){
```



```

varcookieName=encodeURIComponent(name)+'=';
varcookieStart=document.cookie.indexOf(cookieName);
varcookieValue=null;
if(cookieStart>-1){
varcookieEnd=document.cookie.indexOf('; ',cookieStart);
if(cookieEnd===-1){
cookieEnd=document.cookie.length;
}
cookieValue=decodeURIComponent(document.cookie.substring(cookieStart+cookieNam
e.length,cookieEnd));
}
returncookieValue;
}
//删除 cookie
functionunsetCookie(name){
document.cookie=name+'=;expires="+newDate(0);
}

```

272. 面试题

Q: 面试官问

说说 TCP 传输的三次握手策略

A: 应聘者答

为了准确无误地把数据送达目标处，TCP 协议采用了三次握手策略。用 TCP 协议把数据包送出去后，TCP 不会对传送后的情况置之不理，它一定会向对方确认是否成功送达。握手过程中使用了 TCP 的标志：SYN 和 ACK。

发送端首先发送一个带 SYN 标志的数据包给对方。接收端收到后，回传一个带有 SYN/ACK 标志的数据包以示传达确认信息。最后，发送端再回传一个带 ACK 标志的数据包，代表"握手"结束

若在握手过程中某个阶段莫名中断，TCP 协议会再次以相同的顺序发送相同的数据包。

273. 面试题

Q: 面试官问

说说你对 Promise 的理解

A:应聘者答

依照 Promise/A+ 的定义，Promise 有四种状态：

pending: 初始状态, 非 fulfilled 或 rejected.

fulfilled: 成功的操作.

rejected: 失败的操作.

settled: Promise 已被 fulfilled 或 rejected, 且不是 pending

另外, fulfilled 与 rejected 一起合称 settled。

Promise 对象用来进行延迟(deferred)和异步(asynchronous)计算。

274. 面试题

Q: 面试官问

Promise 的构造函数

构造一个 Promise, 最基本的用法如下：

```
var promise = new Promise(function(resolve, reject){  
  if(...){ // succeed  
    resolve(result);  
  } else { // fails  
    reject(Error(errMessage));  
  }  
});
```

Promise 实例拥有 then 方法（具有 then 方法的对象，通常被称为 thenable）。它的使用方法如下：

```
promise.then(onFulfilled, onRejected)
```

接收两个函数作为参数，一个在 fulfilled 的时候被调用，一个在 rejected 的时候被调用，接收参数就是 future，onFulfilled 对应 resolve, onRejected 对应 reject。

275. 面试题

Q: 面试官问

Javascript 垃圾回收方法

A:应聘者答

标记清除 (markandsweep)

这是 JavaScript 最常见的垃圾回收方式，当变量进入执行环境的时候，比如函数中声明一个变量，垃圾回收器将其标记为“进入环境”，当变量离开环境的时候（函数执行结束）将其标记为“离开环境”。

垃圾回收器会在运行的时候给存储在内存中的所有变量加上标记，然后去掉环境中的变量以及被环境中变量所引用的变量（闭包），在这些完成之后仍存在标记的就是要删除的变量了

引用计数(reference counting)

在低版本 IE 中经常会出现内存泄露，很多时候就是因为其采用引用计数方式进行垃圾回收。引用计数的策略是跟踪记录每个值被使用的次数，当声明了一个变量并将一个引用类型赋值给该变量的时候这个值的引用次数就加 1，如果该变量的值变成了另外一个，则这个值得引用次数减 1，当这个值的引用次数变为 0 的时候，说明没有变量在使用，这个值没法被访问了，因此可以将其占用的空间回收，这样垃圾回收器会在运行的时候清理掉引用次数为 0 的值占用的空间。

在 IE 中虽然 JavaScript 对象通过标记清除的方式进行垃圾回收，但 BOM 与 DOM 对象却是通过引用计数回收垃圾的，也就是说只要涉及 BOM 及 DOM 就会出现循环引用问题。

谈谈性能优化问题

A:应聘者答

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。

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，
移动端性能优化

A:应聘者答

尽量使用 css3 动画，开启硬件加速。适当使用 touch 事件代替 click 事件。避免使用 css3 渐变阴影效果。

尽可能少的使用 box-shadow 与 gradients。box-shadow 与 gradients 往往都是页面的性能杀手

276. 面试题

Q:面试官问

什么是 Etag?

A:应聘者答

浏览器下载组件的时候，会将它们存储到浏览器缓存中。如果需要再次获取相同的组件，浏览器将检查组件的缓存时间，

假如已经过期，那么浏览器将发送一个条件 GET 请求到服务器，服务器判断缓存还有效，则发送一个 304 响应，

告诉浏览器可以重用缓存组件。

那么服务器是根据什么判断缓存是否还有效呢?答案有两种方式，一种是前面提到的 ETag，另一种是根据 Last-Modified

277. 面试题

Q:面试官问

Expires 和 Cache-Control

A:应聘者答

Expires 要求客户端和服务端的时钟严格同步。HTTP1.1 引入 Cache-Control 来克服 Expires 头的限制。如果 max-age 和 Expires 同时出现，则 max-age 有更高的优先级。

Cache-Control:no-cache,private,max-age=0

ETag:abcde

Expires:Thu,15Apr201420:00:00GMT

Pragma:private

Last-Modified:\$now//RFC1123format

278. 面试题

Q:面试官问

栈和队列的区别?

A:应聘者答

栈的插入和删除操作都是一端进行的，而队列的操作却是在两端进行的。

队列先进先出，栈先进后出。

栈只允许在表尾一端进行插入和删除，而队列只允许在表尾一端进行插入，在表头一端进行删除

279. 面试题

Q: 面试官问

栈和堆的区别？

A: 应聘者答

栈区 (stack) — 由编译器自动分配释放，存放函数的参数值，局部变量的值等。

堆区 (heap) — 一般由程序员分配释放，若程序员不释放，程序结束时可能由 OS 回收。

堆 (数据结构)：堆可以被看成是一棵树，如：堆排序；

栈 (数据结构)：一种先进后出的数据结构。

280. 面试题

Q: 面试官问

关于 Http2.0 你知道多少？

A: 应聘者答

HTTP/2 引入了“服务端推 (serverpush)”的概念，它允许服务端在客户端需要数据之前就主动地将数据发送到客户端缓存中，从而提高性能。

HTTP/2 提供更多的加密支持

HTTP/2 使用多路技术，允许多个消息在一个连接上同时交差。

它增加了头压缩 (headercompression)，因此即使非常小的请求，其请求和响应的 header 都只会占用很小比例的带宽。

281. 面试题

Q: 面试官问

请你谈谈 Cookie 的弊端

A:应聘者答

cookie 虽然在持久保存客户端数据提供了方便，分担了服务器存储的负担，但还是有很多局限性的。

第一：每个特定的域名下最多生成 20 个 cookie

1.IE6 或更低版本最多 20 个 cookie

2.IE7 和之后的版本最后可以有 50 个 cookie。

3.Firefox 最多 50 个 cookie

4.chrome 和 Safari 没有做硬性限制

IE 和 Opera 会清理近期最少使用的 cookie，Firefox 会随机清理 cookie。

cookie 的最大大约为 4096 字节，为了兼容性，一般不能超过 4095 字节。

IE 提供了一种存储可以持久化用户数据，叫做 uerData，从 IE5.0 就开始支持。每个数据最多 128K，每个域名下最多 1M。这个持久化数据放在缓存中，如果缓存没有清理，那么会一直存在。

优点：极高的扩展性和可用性

1.通过良好的编程，控制保存在 cookie 中的 session 对象的大小。

2.通过加密和安全传输技术 (SSL)，减少 cookie 被破解的可能性。

3.只在 cookie 中存放不敏感数据，即使被盗也不会有重大损失。

4.控制 cookie 的生命期，使之不会永远有效。偷盗者很可能拿到一个过期的 cookie。

缺点：

1.`Cookie`数量和长度的限制。每个 domain 最多只能有 20 条 cookie，每个 cookie 长度不能超过 4KB，否则会被截掉。

2.安全性问题。如果 cookie 被人拦截了，那人就可以取得所有的 session 信息。即使加密也与事无补，因为拦截者并不需要知道 cookie 的意义，他只要原样转发 cookie 就可以达到目的了。

3.有些状态不可能保存在客户端。例如，为了防止重复提交表单，我们需要在服务器端保存一个计数器。如果我们把这个计数器保存在客户端，那么它起不到任何作用。

282. 面试题

Q:面试官问

简单说一下浏览器本地存储是怎样的

A:应聘者答

在较高版本的浏览器中，js 提供了 sessionStorage 和 globalStorage。在 HTML5 中提供了 localStorage 来取代 globalStorage。

html5 中的 WebStorage 包括了两种存储方式：sessionStorage 和 localStorage。

sessionStorage 用于本地存储一个会话 (session) 中的数据, 这些数据只有在同一个会话中的页面才能访问并且当会话结束后数据也随之销毁。因此 sessionStorage 不是一种持久化的本地存储, 仅仅是会话级别的存储。

而 localStorage 用于持久化的本地存储, 除非主动删除数据, 否则数据是永远不会过期的。

283. 面试题

Q: 面试官问

webstorage 和 cookie 的区别

A: 应聘者答

WebStorage 的概念和 cookie 相似, 区别是它是为了更大容量存储设计的。Cookie 的大小是受限的, 并且每次你请求一个新的页面的时候 Cookie 都会被发送过去, 这样无形中浪费了带宽, 另外 cookie 还需要指定作用域, 不可以跨域调用。

除此之外, WebStorage 拥有 setItem, getItem, removeItem, clear 等方法, 不像 cookie 需要前端开发者自己封装 setCookie, getCookie。

但是 Cookie 也是不可或缺的: Cookie 的作用是与服务器进行交互, 作为 HTTP 规范的一部分而存在, 而 WebStorage 仅仅是为了在本地“存储”数据而生

浏览器的支持除了 IE 7 及以下不支持外, 其他标准浏览器都完全支持 (ie 及 FF 需在 web 服务器里运行), 值得一提的是 IE 总是办好事, 例如 IE7、IE6 中的 UserData 其实就是 javascript 本地存储的解决方案。通过简单的代码封装可以统一到所有的浏览器都支持 webstorage。

localStorage 和 sessionStorage 都具有相同的操作方法, 例如 setItem、getItem 和 removeItem 等

284. 面试题

Q: 面试官问

display:none 和 visibility:hidden 的区别?

A: 应聘者答

(1) link 属于 HTML 标签, 而 @import 是 CSS 提供的;

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

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

(4)link 方式的样式的权重高于@import 的权重.

285. 面试题

Q:面试官问

position 的 absolute 与 fixed 共同点与不同点

A:应聘者答

A：共同点：

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

B 不同点：

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

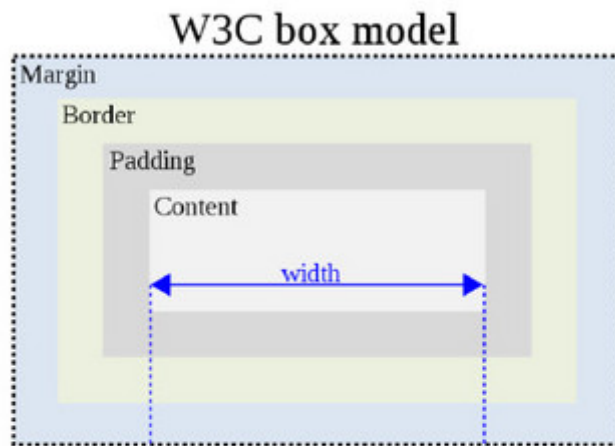
286. 面试题

Q:面试官问

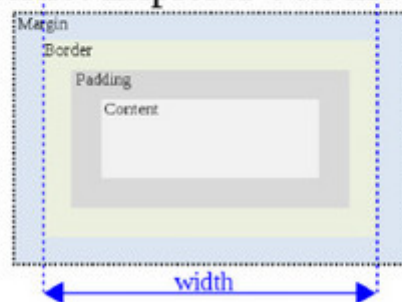
介绍一下 CSS 的盒子模型？

A:应聘者答

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



Internet Explorer box model



W3C和Internet Explorer的盒模型解释 `width` 属性的差异。

IE盒模型

287. 面试题

Q: 面试官问

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

A: 应聘者答

CSS 选择符：

- 1.id 选择器(#myid)
- 2.类选择器(.myclassname)

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

可继承的样式：

1. font-size
2. font-family
3. color
4. text-indent

不可继承的样式：

1. border
2. padding
3. margin
4. width
5. height

优先级算法：

1. 优先级就近原则，同权重情况下样式定义最近者为准；
2. 载入样式以最后载入的定位为准；
3. !important>id>class>tag
4. important 比内联优先级高，但内联比 id 要高

CSS3 新增伪类举例：

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

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

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

p:only-child 选择属于其父元素的唯一子元素的每个<p>元素。

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

:enabled:disabled 控制表单控件的禁用状态。

:checked 单选框或复选框被选中。

288. 面试题

Q: 面试官问

列出 display 的值，说明他们的作用。position 的值，relative 和 absolute 分别是相对于谁进行定位的？

A: 应聘者答

display 的值的的作用：

- 1.block 象块类型元素一样显示。
- 2.inline 缺省值。象行内元素类型一样显示。
- 3.inline-block 象行内元素一样显示，但其内容象块类型元素一样显示。
- 4.list-item 象块类型元素一样显示，并添加样式列表标记。

position 的值的定位区别：

- 1.absolute 生成绝对定位的元素，相对于 static 定位以外的第一个祖先元素进行定位。
- 2.fixed 生成绝对定位的元素，相对于浏览器窗口进行定位（老 IE 不支持）。
- 3.relative 生成相对定位的元素，相对于其在普通流中的位置进行定位。
- 4.static 默认值。没有定位，元素出现在正常的流中（忽略 top,bottom,left,right,z-index 声明）。
- 5.inherit 规定从父元素继承 position 属性的值。

289. 面试题

Q: 面试官问

CSS3 有哪些新特性？

A: 应聘者答

- 1.CSS3 实现圆角 (border-radius)，阴影 (box-shadow)，
- 2.对文字加特效 (text-shadow)，线性渐变 (gradient)，旋转 (transform)
- 3.transform: rotate(9deg) scale(0.85, 0.90) translate(0px, -30px) skew(-9deg, 0deg); // 旋转, 缩放, 定位, 倾斜
- 4.增加了更多的 CSS 选择器多背景 rgba
- 5.在 CSS3 中唯一引入的伪元素是 ::selection.
- 6.媒体查询，多栏布局
- 7.border-image

290. 面试题

Q: 面试官问

为什么要初始化 CSS 样式。

A: 应聘者答

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

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

*最简单的初始化方法就是（不建议）：

```
1      {padding:0;margin:0;}
```

淘宝的样式初始化：

```
1      body,h1,h2,h3,h4,h5,h6,hr,p,blockquote,dl,dt,dd,ul,ol,li,pre,form,fieldset,legend,button,input,textarea,t
2      body,button,input,select,textarea{ font:12px/1.5tahoma,arial,\5b8b\4f53;}
3      h1,h2,h3,h4,h5,h6{ font-size:100%;}
4      address,cite,dfn,em,var{ font-style:normal;}
5      code,kbd,pre,samp{ font-family:couriernew,courier,monospace;}
6      small{ font-size:12px;}
7      ul,ol{list-style:none;}
8      a{ text-decoration:none;}
9      a:hover{ text-decoration:underline;}
10     sup{ vertical-align:text-top;}
11     sub{ vertical-align:text-bottom;}
12     legend{ color:#000;}
13     fieldset,img{ border:0;}
14     button,input,select,textarea{ font-size:100%;}
15     table{ border-collapse:collapse;border-spacing:0;}
```

291. 面试题

Q: 面试官问

对 BFC 规范的理解？

A: 应聘者答

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

W3CCSS2.1 规范中的一个概念，它决定了元素如何对其内容进行布局，以及与其他元素的关系和相互作用。

纠错收藏

292. 面试题

Q: 面试官问

解释下 CSSsprites，以及你要如何在页面或网站中使用它。

A: 应聘者答

CSSSprites 其实就是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 可以用数字能精确的定位出背景图片的位置。这样可以减少很多图片请求的开销，因为请求耗时比较长；请求虽然可以并发，但是也有限制，一般浏览器都是 6 个。对于未来而言，就不需要这样做了，因为有了 http2。

293. 面试题

Q: 面试官问

说说你对语义化的理解？

A: 应聘者答

1. 去掉或者丢失样式的时候能够让页面呈现出清晰的结构
2. 有利于 SEO：和搜索引擎建立良好沟通，有助于爬虫抓取更多的有效信息：爬虫依赖于标签来确定上下文和各个关键字的权重；
3. 方便其他设备解析（如屏幕阅读器、盲人阅读器、移动设备）以意义的方式来渲染网页；
4. 便于团队开发和维护，语义化更具可读性，是下一步吧网页的重要动向，遵循 W3C 标准的团队都遵循这个标准，可以减少差异化。

294. 面试题

Q: 面试官问

Doctype 作用？严格模式与混杂模式如何区分？它们有何意义？

A: 应聘者答

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

- 2.严格模式的排版和 JS 运作模式是以该浏览器支持的最高标准运行。
- 3.在混杂模式中，页面以宽松的向后兼容的方式显示。模拟老式浏览器的行为以防止站点无法工作。
- 4.DOCTYPE 不存在或格式不正确会导致文档以混杂模式呈现。

295. 面试题

Q:面试官问

你知道多少种 Doctype 文档类型？

A:应聘者答

- 1.该标签可声明三种 DTD 类型, 分别表示严格版本、过渡版本以及基于框架的 HTML 文档。
- 2.HTML4.01 规定了三种文档类型：Strict、Transitional 以及 Frameset。
- 3.XHTML1.0 规定了三种 XML 文档类型：Strict、Transitional 以及 Frameset。
- 4.Standards（标准）模式（也就是严格呈现模式）用于呈现遵循最新标准的网页，而 Quirks（包容）模式（也就是松散呈现模式或者兼容模式）用于呈现为传统浏览器而设计的网页。

296. 面试题

Q:面试官问

HTML 与 XHTML——二者有什么区别

A:应聘者答

- 1.所有的标记都必须要有个相应的结束标记
 - 2.所有标签的元素和属性的名字都必须使用小写
 - 3.所有的 XML 标记都必须合理嵌套
 - 4.所有的属性必须用引号""括起来
 - 5.把所有<和&特殊符号用编码表示
 - 6.给所有属性赋一个值
 - 7.不要在注释内容中使用"--"
 - 8.图片必须有说明文字
- html 常见兼容性问题？

A:应聘者答

1.png24 位的图片在 ie6 浏览器上出现背景

解决方案：做成 PNG8，也可以引用一段脚本处理。

2.浏览器默认的 margin 和 padding 不同

解决方案：加一个全局的*{margin:0;padding:0;}来统一。

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

```
1      #box{float:left;width:10px;margin:00010px;}
```

这种情况之下 IE 会产生 20px 的距离

解决方案：在 float 的标签样式控制中加入_display:inline;将其转化为行内属性。（这个符号只有 ie6 会识别）

4.渐进识别的方式，从总体中逐渐排除局部。

首先，巧妙的使用“\9”这一标记，将 IE 浏览器从所有情况中分离出来。

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

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

5.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

6.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。

7.Chrome 中文界面下默认会将小于 12px 的文本强制按照 12px 显示

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

8.超链接访问过后 hover 样式就不出现了，被点击访问过的超链接样式不在具有 hover 和 active 了

解决方法：改变 CSS 属性的排列顺序 L-V-H-A

```
1      a:link{}
2      a:visited{}
3      a:hover{}
4      a:active{}
```

9.怪异模式问题：漏写 DTD 声明，Firefox 仍然会按照标准模式来解析网页，但在 IE 中会触发怪异模式。为避免怪异模式给我们带来不必要的麻烦，最好养成书写 DTD 声明的好习惯。现在可以使用[html5](http://www.w3.org/TR/html5/single-page.html)推荐的写法：
<!DOCTYPEhtml>

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

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

11.ie6 对 png 图片格式支持不好

解决方案：引用一段脚本处理

297. 面试题

Q:面试官问

解释下浮动和它的工作原理？清除浮动的技巧

A:应聘者答

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 `cssclear: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；

298. 面试题

Q:面试官问

浮动元素引起的问题和解决办法？

A:应聘者答

浮动元素引起的问题：

- 1.父元素的高度无法被撑开，影响与父元素同级的元素
- 2.与浮动元素同级的非浮动元素会跟随其后
- 3.若非第一个元素浮动，则该元素之前的元素也需要浮动，否则会影响页面显示的结构

解决方法：

使用 CSS 中的 `clear:both;` 属性来清除元素的浮动可解决 2、3 问题，对于问题 1，添加如下样式，给父元素添加 `clearfix` 样式：

```
1      .clearfix:after{content:".";display:block;height:0;clear:both;visibility:hidden;}
2      .clearfix{display:inline-block;}/*forIE/Mac*/
```

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` 伪元素

```
1      #parent:after{
2      content:".";
3      height:0;
4      visibility:hidden;
5      display:block;
6      clear:both;
7      }
```

3. 浮动外部元素

4. 设置 `overflow` 为 `hidden` 或者 `auto`

299. 面试题

Q: 面试官问

IE8 以下版本的浏览器中的盒模型有什么不同

A: 应聘者答

IE8 以下浏览器的盒模型中定义的元素的高包括内边距和边框

300. 面试题

Q: 面试官问

DOM 操作——怎样添加、移除、移动、复制、创建和查找节点。

A: 应聘者答

1. 创建新节点

```
1      createDocumentFragment()//创建一个 DOM 片段
```

- 2 createElement()//创建一个具体的元素
- 3 createTextNode()//创建一个文本节点

2.添加、移除、替换、插入

- 1 appendChild()
- 2 removeChild()
- 3 replaceChild()
- 4 insertBefore()//在已有的子节点前插入一个新的子节点

3.查找

- 1 getElementsByTagName()//通过标签名称
- 2 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
- 3 getElementById()//通过元素 Id，唯一性

301. 面试题

Q:面试官问

html5 有哪些新特性、移除了那些元素？如何处理 HTML5 新标签的浏览器兼容问题？
如何区分 HTML 和 HTML5？

A:应聘者答

新特性：

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

- 1.拖拽释放(Draganddrop)API
- 2.语义化更好的内容标签（header,nav,footer,aside,article,section）
- 3.音频、视频 API(audio,video)
- 4.画布(Canvas)API
- 5.地理(Geolocation)API
- 6.本地离线存储 localStorage 长期存储数据，浏览器关闭后数据不丢失；
- 7.sessionStorage 的数据在浏览器关闭后自动删除
- 8.表单控件，calendar、date、time、email、url、search
- 9.新的技术 webworker,websocket,Geolocation

移除的元素：

- 1.纯表现的元素：basefont，big，center，font,s，strike，tt，u；
 - 2.对可用性产生负面影响的元素：frame，frameset，noframes；
- 支持 HTML5 新标签：

1.IE8/IE7/IE6 支持通过 document.createElement 方法产生的标签，可以利用这一特性让这些浏览器支持 HTML5 新标签，浏览器支持新标签后，还需要添加标签默认的风格（当然最好的方式是直接使用成熟的框架、使用最多的是 html5shiv 框架）：

```
1      <!--[ifltIE9]>
2      <script>src="http://html5shiv.googlecode.com/svn/trunk/html5.js"</script>
3      <![endif]-->
```

如何区分：

DOCTYPE 声明新增的结构元素、功能元素

302. 面试题

Q: 面试官问

iframe 的优缺点？

A: 应聘者答

优点：

- 1.解决加载缓慢的第三方内容如图标和广告等的加载问题
- 2.Securitysandbox
- 3.并行加载脚本

缺点：

- 1.iframe 会阻塞主页面的 Onload 事件
- 2.即时内容为空，加载也需要时间
- 3.没有语意

303. 面试题

Q: 面试官问

如何实现浏览器内多个标签页之间的通信？

A: 应聘者答

调用 localStorage、cookies 等本地存储方式

304. 面试题

Q: 面试官问

webSocket 如何兼容低浏览器？

A: 应聘者答

AdobeFlashSocket、ActiveXHTMLFile(IE)、基于 multipart 编码发送 XHR、基于长轮询的 XHR

305. 面试题

Q: 面试官问

线程与进程的区别

A: 应聘者答

1. 一个程序至少有一个进程, 一个进程至少有一个线程
2. 线程的划分尺度小于进程, 使得多线程程序的并发性高
3. 另外, 进程在执行过程中拥有独立的内存单元, 而多个线程共享内存, 从而极大地提高了程序的运行效率
4. 线程在执行过程中与进程还是有区别的。每个独立的线程有一个程序运行的入口、顺序执行序列和程序的出口。但是线程不能够独立执行, 必须依存在应用程序中, 由应用程序提供多个线程执行控制
5. 从逻辑角度来看, 多线程的意义在于一个应用程序中, 有多个执行部分可以同时执行。但操作系统并没有将多个线程看做多个独立的应用, 来实现进程的调度和管理以及资源分配。这就是进程和线程的重要区别

306. 面试题

Q: 面试官问

你如何对网站的文件和资源进行优化？

A: 应聘者答

期待的解决方案包括：

1. 文件合并

- 2.文件最小化/文件压缩
- 3.使用 CDN 托管
- 4.缓存的使用（多个域名来提供缓存）
- 5.其他

307. 面试题

Q:面试官问

请说出三种减少页面加载时间的方法

A:应聘者答

- 1.优化图片
- 2.图像格式的选择（GIF：提供的颜色较少，可用在一些对颜色要求不高的地方）
- 3.优化 CSS（压缩合并 css，如 margin-top,margin-left...）
- 4.网址后加斜杠（如 www.campr.com/目录，会判断这个目录是什么文件类型，或者是目录。）
- 5.标明高度和宽度（如果浏览器没有找到这两个参数，它需要一边下载图片一边计算大小，如果图片很多，浏览器需要不断地调整页面。这不但影响速度，也影响浏览体验。当浏览器知道了高度和宽度参数后，即使图片暂时无法显示，页面上也会腾出图片的空位，然后继续加载后面的内容。从而加载时间快了，浏览体验也更好了）
- 6.减少 http 请求（合并文件，合并图片）

308. 面试题

Q:面试官问

你都使用哪些工具来测试代码的性能？

A:应聘者答

- 1.Profiler
- 2.JSPerf (<http://jsperf.com/nexttick-vs-setzerotimeout-vs-settimeout>)
- 3.Dromaeo

309. 面试题

Q: 面试官问

什么是 FOUC（无样式内容闪烁）？你如何来避免 FOUC？

A: 应聘者答

FOUC-FlashOfUnstyledContent 文档样式闪烁

```
<style type="text/css" media="all">@import "../fouc.css";</style>
```

而引用 CSS 文件的 @import 就是造成这个问题的罪魁祸首。IE 会先加载整个 HTML 文档的 DOM，然后再去导入外部的 CSS 文件，因此，在页面 DOM 加载完成到 CSS 导入完成中间会有一段时间页面上的内容是没有样式的，这段时间的长短跟网速，电脑速度都有关系。

解决方法简单的出奇，只要在 <head> 之间加入一个 <link> 或者 <script> 元素就可以了。

310. 面试题

Q: 面试官问

null 和 undefined 的区别？

A: 应聘者答

null 是一个表示“无”的对象，转为数值时为 0

undefined 是一个表示“无”的原始值，转为数值时为 NaN

当声明的变量还未被初始化时，变量的默认值为 undefined

null 用来表示尚未存在的对象，常用来表示函数企图返回一个不存在的对象

undefined 表示“缺少值”，就是此处应该有一个值，但是还没有定义。典型用法是：

1. 变量被声明了，但没有赋值时，就等于 undefined
2. 调用函数时，应该提供的参数没有提供，该参数等于 undefined
3. 对象没有赋值的属性，该属性的值为 undefined
4. 函数没有返回值时，默认返回 undefined

null 表示“没有对象”，即该处不应该有值。典型用法是：

1. 作为函数的参数，表示该函数的参数不是对象
2. 作为对象原型链的终点

311. 面试题

Q: 面试官问

new 操作符具体干了什么呢?

A: 应聘者答

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

```
1      var obj={};  
2      obj.__proto__=Base.prototype;  
3      Base.call(obj);
```

312. 面试题

Q: 面试官问

对 JSON 的了解?

A: 应聘者答

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。

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

313. 面试题

Q: 面试官问

js 延迟加载的方式有哪些?

A: 应聘者答

1. defer 和 async
2. 动态创建 DOM 方式 (创建 script, 插入到 DOM 中, 加载完毕后 callBack)

3.按需异步载入 js

314. 面试题

Q: 面试官问

如何解决跨域问题?

A: 应聘者答

- 1.jsonp (jsonp 的原理是动态插入 script 标签)
- 2.document.domain+iframe
- 3.window.name、window.postMessage
- 4.服务器上设置代理页面

315. 面试题

Q: 面试官问

document.write 和 innerHTML 的区别

A: 应聘者答

document.write 只能重绘整个页面
innerHTML 可以重绘页面的一部分

316. 面试题

Q: 面试官问

.call()和.apply()的作用?

A: 应聘者答

动态改变某个类的某个方法的运行环境

317. 面试题

Q: 面试官问

哪些操作会造成内存泄漏？

A: 应聘者答

内存泄漏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。

1. `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。
2. 闭包
3. 控制台日志
4. 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）

318. 面试题

Q: 面试官问

如何判断当前脚本运行在浏览器还是 `node` 环境中？

A: 应聘者答

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

319. 面试题

Q: 面试官问

什么叫优雅降级和渐进增强？

A: 应聘者答

1. 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。

2.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

320. 面试题

Q:面试官问

对 Node 的优点和缺点提出了自己的看法?

A:应聘者答

优点 :

1.因为 Node 是基于事件驱动和无阻塞的,所以非常适合处理并发请求,因此构建在 Node 上的代理服务器相比其他技术实现(如 Ruby)的服务器表现要好得多。

2.与 Node 代理服务器交互的客户端代码是由 javascript 语言编写的,因此客户端和服务端都用同一种语言编写,这是非常美妙的事情。

缺点 :

1.Node 是一个相对新的开源项目,所以不太稳定,它总是一直在变。

2.缺少足够多的第三方库支持。看起来,就像是 Ruby/Rails 当年的样子(第三方库现在已经很丰富了,所以这个缺点可以说不存在了)。

321. 面试题

Q:面试官问

对前端界面工程师这个职位是怎么样理解的?它的前景会怎么样?

A:应聘者答

前端是最贴近用户的程序员,比后端、数据库、产品经理、运营、安全都近。

1.实现界面交互

2.提升用户体验

3.有了 Node.js,前端可以实现服务端的一些事情

前景 :

1.前端是最贴近用户的程序员,前端的能力就是能让产品从 90 分进化到 100 分,甚至更好

2.参与项目,快速高质量完成实现效果图,精确到 1px ;

3.与团队成员,UI 设计,产品经理的沟通 ;

4.做好的页面结构,页面重构和用户体验 ;

- 5.处理 hack, 兼容、写出优美的代码格式 ;
- 6.针对服务器的优化、拥抱最新前端技术。

322. 面试题

Q:面试官问

你有哪些性能优化的方法?

A:应聘者答

- 1.减少 http 请求次数 :CSSSprites,JS、CSS 源码压缩、图片大小控制合适 ;网页 Gzip, CDN 托管, data 缓存, 图片服务器
- 2.前端模板 JS+数据, 减少由于 HTML 标签导致的带宽浪费, 前端用变量保存 AJAX 请求结果, 每次操作本地变量, 不用请求, 减少请求次数
- 3.用 innerHTML 代替 DOM 操作, 减少 DOM 操作次数, 优化 javascript 性能
- 4.当需要设置的样式很多时设置 className 而不是直接操作 style
- 5.少用全局变量、缓存 DOM 节点查找的结果。减少 IO 读取操作
- 6.避免使用 CSSExpression (css 表达式)又称 Dynamicproperties(动态属性)
- 7.图片预加载, 将样式表放在顶部, 将脚本放在底部, 加上时间戳

323. 面试题

Q:面试官问

http 状态码有那些? 分别代表是什么意思?

A:应聘者答

- 1.100-199 用于指定客户端应相应的某些动作
- 2.200-299 用于表示请求成功
- 3.300-399 用于已经移动的文件并且常被包含在定位头信息中指定新的地址信息
- 4.400-499 用于指出客户端的错误
- 400 : 语义有误, 当前请求无法被服务器理解
- 401 : 当前请求需要用户验证
- 403 : 服务器已经理解请求, 但是拒绝执行它
- 5.500-599 用于支持服务器错误
- 503 : 服务不可用

324. 面试题

Q:面试官问

一个页面从输入 URL 到页面加载显示完成，这个过程中都发生了什么？

A:应聘者答

分为 4 个步骤：

1.当发送一个 URL 请求时，不管这个 URL 是 Web 页面的 URL 还是 Web 页面上每个资源的 URL，浏览器都会开启一个线程来处理这个请求，同时在远程 DNS 服务器上启动一个 DNS 查询。这能使浏览器获得请求对应的 IP 地址。

2.浏览器与远程 Web 服务器通过 TCP 三次握手协商来建立一个 TCP/IP 连接。该握手包括一个同步报文，一个同步-应答报文和一个应答报文，这三个报文在浏览器和服务器之间传递。该握手首先由客户端尝试建立起通信，而后服务器应答并接受客户端的请求，最后由客户端发出该请求已经被接受的报文。

3.一旦 TCP/IP 连接建立，浏览器会通过该连接向远程服务器发送 HTTP 的 GET 请求。远程服务器找到资源并使用 HTTP 响应返回该资源，值为 200 的 HTTP 响应状态表示一个正确的响应。

4.此时，Web 服务器提供资源服务，客户端开始下载资源。

请求返回后，便进入了我们关注的前端模块

简单来说，浏览器会解析 HTML 生成 DOMTree，其次会根据 CSS 生成 CSSRuleTree，而 javascript 又可以根据 DOMAPI 操作 DOM

325. 面试题

Q:面试官问

平时如何管理你的项目？

A:应聘者答

- 1.先期团队必须确定好全局样式（globe.css），编码模式(utf-8)等
- 2.编写习惯必须一致（例如都是采用继承式的写法，单样式都写成一行）
- 3.标注样式编写人，各模块都及时标注（标注关键样式调用的地方）
- 4.页面进行标注（例如页面模块开始和结束）
- 5.CSS 跟 HTML 分文件夹并行存放，命名都得统一（例如 style.css）
- 6.JS 分文件夹存放命名以该 JS 功能为准的英文翻译

7.图片采用整合的 images.pngpng8 格式文件使用尽量整合在一起使用方便将来的管理

326. 面试题

Q:面试官问

说说最近最流行的一些东西吧？常去的哪些网站？

A:应聘者答

最流行的一些东西：

- 1.Node.js
- 2.Mongodb
- 3.npm
- 4.MVVM
- 5.MEAN
- 6.three.js
- 7.React

常去的网站：

- 1.牛客网
- 2.Github
- 3.CSDN

327. 面试题

Q:面试官问

javascript 对象的几种创建方式

A:应聘者答

- 1.工厂模式
- 2.构造函数模式
- 3.原型模式
- 4.混合构造函数和原型模式
- 5.动态原型模式
- 6.寄生构造函数模式
- 7.稳妥构造函数模式

328. 面试题

Q: 面试官问

javascript 继承的 6 种方法

A: 应聘者答

1. 原型链继承
2. 借用构造函数继承
3. 组合继承(原型+借用构造)
4. 原型式继承
5. 寄生式继承
6. 寄生组合式继承

329. 面试题

Q: 面试官问

ajax 的过程是怎样的

A: 应聘者答

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

330. 面试题

Q: 面试官问

异步加载和延迟加载

A:应聘者答

- 1.异步加载的方案：动态插入 script 标签
- 2.通过 ajax 去获取 js 代码，然后通过 eval 执行
- 3.script 标签上添加 defer 或者 async 属性
- 4.创建并插入 iframe，让它异步执行 js
- 5.延迟加载：有些 js 代码并不是页面初始化的时候就立刻需要的，而稍后的某些情况才需要的

331. 面试题

Q:面试官问

前端的安全问题？

A:应聘者答

- 1.XSS
- 2.sql 注入
- 3.CSRF：是跨站请求伪造，很明显根据刚刚的解释，他的核心也就是请求伪造，通过伪造身份提交 POST 和 GET 请求来进行跨域的攻击
完成 CSRF 需要两个步骤：
 - 1.登陆受信任的网站 A，在本地生成 COOKIE
 - 2.在不登出 A 的情况下，或者本地 COOKIE 没有过期的情况下，访问危险网站 B。

332. 面试题

Q:面试官问

ie 各版本和 chrome 可以并行下载多少个资源

A:应聘者答

- 1.IE62 个并发
- 2.iE7 升级之后的 6 个并发，之后版本也是 6 个
- 3.Firefox, chrome 也是 6 个

333. 面试题

Q: 面试官问

javascript 里面的继承怎么实现，如何避免原型链上面的对象共享

A: 应聘者答

用构造函数和原型链的混合模式去实现继承，避免对象共享可以参考经典的 extend() 函数，很多前端框架都有封装的，就是用一个空函数当做中间变量

334. 面试题

Q: 面试官问

grunt，YUIcompressor 和 googleclosure 用来进行代码压缩的用法。

A: 应聘者答

grunt：

UglifyJS 是基于 NodeJS 的 Javascript 语法解析/压缩/格式化工具

官网：<http://lisperator.net/uglifyjs/>或者 <https://github.com/mishoo/UglifyJS2>

安装：

```
1 $npminstalluglify-js-g
```

使用方法见官网 demo

YUIcompressor：

YUICompressor 是一个用来压缩 JS 和 CSS 文件的工具，采用 Java 开发。

使用方法：

```
1 //压缩 JS
```

```
2 java-jaryuicompressor-2.4.2.jar--typejs--charsetutf-8-vsrc.js>packed.js
```

```
3 //压缩 CSS
```

```
4 java-jaryuicompressor-2.4.2.jar--typecss--charsetutf-8-vsrc.css>packed.css
```

GoogleClosureCompiler：

官网：<https://developers.google.com/closure/compiler/>

使用方法：

1.在命令行下使用一个 google 编译好的 java 程序

2.使用 google 提供的在线服务

3.使用 google 提供的 RESTfulAPI

335. 面试题

Q: 面试官问

Flash、Ajax 各自的优缺点，在使用中如何取舍？

A: 应聘者答

Flash：

1. Flash 适合处理多媒体、矢量图形、访问机器
2. 对 CSS、处理文本上不足，不容易被搜索

Ajax：

1. Ajax 对 CSS、文本支持很好，支持搜索
2. 多媒体、矢量图形、机器访问不足

共同点：

1. 与服务器的无刷新传递消息
2. 可以检测用户离线和在线状态
2. 操作 DOM

336. 面试题

Q: 面试官问

请解释一下 JavaScript 的同源策略。

A: 应聘者答

概念：

同源策略是客户端脚本（尤其是 Javascript）的重要的安全度量标准。它最早出自 NetscapeNavigator2.0，其目的是防止某个文档或脚本从多个不同源装载。

这里的同源策略指的是：协议，域名，端口相同，同源策略是一种安全协议，指一段脚本只能读取来自同一天源的窗口和文档的属性。

为什么要有同源限制：

我们举例说明：比如一个黑客程序，他利用 Iframe 把真正的银行登录页面嵌到他的页面上，当你使用真实的用户名，密码登录时，他的页面就可以通过 Javascript 读取到你的表单中 input 中的内容，这样用户名，密码就轻松到手了。

337. 面试题

Q: 面试官问

为什么要有同源限制？

A: 应聘者答

我们举例说明：比如一个黑客程序，他利用 Iframe 把真正的银行登录页面嵌到他的页面上，当你使用真实的用户名，密码登录时，他的页面就可以通过 Javascript 读取到你的表单中 input 中的内容，这样用户名，密码就轻松到手了。

338. 面试题

Q: 面试官问

什么是 "usestrict";? 使用它的好处和坏处分别是什么？

A: 应聘者答

ECMAScript5 添加了第二种运行模式："严格模式" (strictmode)。顾名思义，这种模式使得 Javascript 在更严格的条件下运行。

设立"严格模式"的目的，主要有以下几个：

1. 消除 Javascript 语法的一些不合理、不严谨之处，减少一些怪异行为；
2. 消除代码运行的一些不安全之处，保证代码运行的安全；
3. 提高编译器效率，增加运行速度；
4. 为未来新版本的 Javascript 做好铺垫。

注：经过测试 IE6,7,8,9 均不支持严格模式。

缺点：

现在网站的 JS 都会进行压缩，一些文件用了严格模式，而另一些没有。这时这些本来是严格模式的文件，被 merge 后，这个串就到了文件的中间，不仅没有指示严格模式，反而在压缩后浪费了字节。

339. 面试题

Q: 面试官问

GET 和 POST 的区别，何时使用 POST？

A:应聘者答

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

POST：一般用于修改服务器上的资源，对所发送的信息没有限制

GET 方式需要使用 Request.QueryString 来取得变量的值

POST 方式通过 Request.Form 来获取变量的值

也就是说 Get 是通过地址栏来传值，而 Post 是通过提交表单来传值。

在以下情况中，请使用 POST 请求：

- 1.无法使用缓存文件（更新服务器上的文件或数据库）
- 2.向服务器发送大量数据（POST 没有数据量限制）
- 3.发送包含未知字符的用户输入时，POST 比 GET 更稳定也更可靠

340. 面试题

Q:面试官问

哪些地方会出现 css 阻塞，哪些地方会出现 js 阻塞？

A:应聘者答

js 的阻塞特性：所有浏览器在下载 JS 的时候，会阻止一切其他活动，比如其他资源的下载，内容的呈现等等。直到 JS 下载、解析、执行完毕后才开始继续并行下载其他资源并呈现内容。为了提高用户体验，新一代浏览器都支持并行下载 JS，但是 JS 下载仍然会阻塞其它资源的下载（例如.图片，css 文件等）。

由于浏览器为了防止出现 JS 修改 DOM 树，需要重新构建 DOM 树的情况，所以就会阻塞其他的下载和呈现。

嵌入 JS 会阻塞所有内容的呈现，而外部 JS 只会阻塞其后内容的显示，2 种方式都会阻塞其后资源的下载。也就是说外部样式不会阻塞外部脚本的加载，但会阻塞外部脚本的执行。

CSS 怎么会阻塞加载了？CSS 本来是可以并行下载的，在什么情况下会出现阻塞加载了(在测试观察中，IE6 下 CSS 都是阻塞加载)

当 CSS 后面跟着嵌入的 JS 的时候，该 CSS 就会出现阻塞后面资源下载的情况。而当把嵌入 JS 放到 CSS 前面，就不会出现阻塞的情况了。

根本原因：因为浏览器会维持 html 中 css 和 js 的顺序，样式表必须在嵌入的 JS 执行前先加载、解析完。而嵌入的 JS 会阻塞后面的资源加载，所以就会出现上面 CSS 阻塞下载的情况。

嵌入 JS 应该放在什么位置？

- 1.放在底部，虽然放在底部照样会阻塞所有呈现，但不会阻塞资源下载。
- 2.如果嵌入 JS 放在 head 中，请把嵌入 JS 放在 CSS 头部。
- 3.使用 defer（只支持 IE）

4.不要在嵌入的 JS 中调用运行时间较长的函数，如果一定要用，可以用 `setTimeout` 来调用

Javascript 无阻塞加载具体方式：

1.将脚本放在底部。`<link>`还是放在 `head` 中，用以保证在 js 加载前，能加载出正常显示的页面。`<script>`标签放在`</body>`前。

2.阻塞脚本：由于每个`<script>`标签下载时阻塞页面解析过程，所以限制页面的`<script>`总数也可以改善性能。适用于内联脚本和外部脚本。

3.非阻塞脚本：等页面完成加载后，再加载 js 代码。也就是，在 `window.onload` 事件发出后开始下载代码。

4.`defer` 属性：支持 IE4 和 Firefox 3.5 更高版本浏览器

5.动态脚本元素：文档对象模型（DOM）允许你使用 js 动态创建 HTML 的几乎全部文档内容。代码如下：

```
1      <script>
2      varscript=document.createElement("script");
3      script.type="text/javascript";
4      script.src="file.js";
5      document.getElementsByTagName("head")[0].appendChild(script);
6      </script>
```

此技术的重点在于：无论在何处启动下载，文件下载和运行都不会阻塞其他页面处理过程，即使在 `head` 里（除了用于下载文件的 `http` 链接）。

`eval` 是做什么的？

A:应聘者答

- 1.它的功能是把对应的字符串解析成 JS 代码并运行
- 2.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）

341. 面试题

Q:面试官问

写一个通用的事件侦听器函数

A:应聘者答

```
1      //event(事件)工具集，来源：github.com/markyun
2      markyun.Event={
3      //页面加载完成后
4      readyEvent:function(fn){
5      if(fn==null){
6      fn=document;
```

```

7      }
8      var oldonload=window.onload;
9      if(typeof window.onload!='function'){
10         window.onload=fn;
11     }else{
12         window.onload=function(){
13             oldonload();
14             fn();
15         };
16     }
17 },
18 //视能力分别使用 dom0||dom2||IE 方式来绑定事件
19 //参数：操作的元素,事件名称,事件处理程序
20 addEvent:function(element,type,handler){
21     if(element.addEventListener){
22         //事件类型、需要执行的函数、是否捕捉
23         element.addEventListener(type,handler,false);
24     }elseif(element.attachEvent){
25         element.attachEvent('on'+type,function(){
26             handler.call(element);
27         });
28     }else{
29         element['on'+type]=handler;
30     }
31 },
32 //移除事件
33 removeEvent:function(element,type,handler){
34     if(element.removeEventListener){
35         element.removeEventListener(type,handler,false);
36     }elseif(element.detachEvent){
37         element.detachEvent('on'+type,handler);
38     }else{
39         element['on'+type]=null;
40     }
41 },
42 //阻止事件(主要是事件冒泡，因为 IE 不支持事件捕获)
43 stopPropagation:function(ev){
44     if(ev.stopPropagation){
45         ev.stopPropagation();
46     }else{
47         ev.cancelBubble=true;
48     }
49 },
50 //取消事件的默认行为

```

```

51     preventDefault:function(event){
52     if(event.preventDefault){
53     event.preventDefault();
54     }else{
55     event.returnValue=false;
56     }
57     },
58     //获取事件目标
59     getTarget:function(event){
60     return event.target||event.srcElement;
61     },
62     //获取 event 对象的引用，取到事件的所有信息，确保随时能使用 event；
63     getEvent:function(e){
64     varev=e||window.event;
65     if(!ev){
66     varc=this.getEvent.caller;
67     while(c){
68     ev=c.arguments[0];
69     if(ev&&Event==ev.constructor){
70     break;
71     }
72     c=c.caller;
73     }
74     }
75     return ev;
76     }
77     };

```

342. 面试题

Q: 面试官问

Node.js 的适用场景

A: 应聘者答

- 1.高并发
- 2.聊天
- 3.实时消息推送

343. 面试题

Q: 面试官问

JavaScript 原型，原型链?有什么特点?

A: 应聘者答

- 1.原型对象也是普通的对象，是对象一个自带隐式的__proto__属性，原型也有可能有自己的原型，如果一个原型对象的原型不为 null 的话，我们就称之为原型链
- 2.原型链是由一些用来继承和共享属性的对象组成的（有限的）对象链

344. 面试题

Q: 面试官问

怎么重构页面？

A: 应聘者答

- 1.编写 CSS
- 2.让页面结构更合理化，提升用户体验
- 3.实现良好的页面效果和提升性能

345. 面试题

Q: 面试官问

WEB 应用从服务器主动推送 Data 到客户端有那些方式？

A: 应聘者答

- 1.html5websocket
- 2.WebSocket 通过 Flash
- 3.XHR 长时间连接
- 4.XHRMultipartStreaming
- 5.不可见的 Iframe

6.<script>标签的长时间连接(可跨域)

346. 面试题

Q:面试官问

事件、IE 与火狐的事件机制有什么区别？如何阻止冒泡？

A:应聘者答

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();

注意旧 ie 的方法：ev.cancelBubble=true;

347. 面试题

Q:面试官问

Ajax 是什么？Ajax 的交互模型？同步和异步的区别？如何解决跨域问题？

A:应聘者答

Ajax 是什么：

1.通过异步模式，提升了用户体验

2.优化了浏览器和服务器之间的传输，减少不必要的数据往返，减少了带宽占用

3.Ajax 在客户端运行，承担了一部分本来由服务器承担的工作，减少了大用户量下的服务器负载。

Ajax 的最大的特点：

1.Ajax 可以实现动态不刷新（局部刷新）

2.readyState 属性状态有 5 个可取值：0=未初始化，1=启动，2=发送，3=接收，4=完成

Ajax 同步和异步的区别：

1.同步：提交请求->等待服务器处理->处理完毕返回，这个期间客户端浏览器不能干任何事

2.异步：请求通过事件触发->服务器处理（这是浏览器仍然可以作其他事情）->处理完毕

ajax.open 方法中，第 3 个参数是设同步或者异步。

Ajax 的缺点：

- 1.Ajax 不支持浏览器 back 按钮
- 2.安全问题 Ajax 暴露了与服务器交互的细节
- 3.对搜索引擎的支持比较弱
- 4.破坏了程序的异常机制
- 5.不容易调试

解决跨域问题：

- 1.jsonp
- 2.iframe
- 3.window.name、window.postMessage
- 4.服务器上设置代理页面

348. 面试题

Q: 面试官问

js 对象的深度克隆代码实现

A: 应聘者答

```
1      function clone(Obj){
2          var buf;
3          if(Obj instanceof Array){
4              buf=[];//创建一个空的数组
5              var i=Obj.length;
6              while(i--){
7                  buf[i]=clone(Obj[i]);
8              }
9              return buf;
10         }else if(Obj instanceof Object){
11             buf={}; //创建一个空对象
12             for(var k in Obj){ //为这个对象添加新的属性
13                 buf[k]=clone(Obj[k]);
14             }
15             return buf;
16         }else{
17             return Obj;
18         }
19     }
```

349. 面试题

Q: 面试官问

对网站重构的理解

A: 应聘者答

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

对于传统的网站来说重构通常是：

1. 表格(table)布局改为 DIV+CSS
2. 使网站前端兼容于现代浏览器(针对于不合规范的 CSS、如对 IE6 有效的)
3. 对于移动平台的优化
4. 针对于 SEO 进行优化
5. 深层次的网站重构应该考虑的方面
6. 减少代码间的耦合
7. 让代码保持弹性
8. 严格按规范编写代码
9. 设计可扩展的 API
10. 代替旧有的框架、语言(如 VB)
11. 增强用户体验
12. 通常来说对于速度的优化也包含在重构中
13. 压缩 JS、CSS、image 等前端资源(通常是由服务器来解决)
14. 程序的性能优化(如数据读写)
15. 采用 CDN 来加速资源加载
16. 对于 JSDOM 的优化
17. HTTP 服务器的文件缓存

350. 面试题

Q: 面试官问

351. 如何获取 UA

A:应聘者答

```
1      <script>
2      functionwhatBrowser(){
3      document.Browser.Name.value=navigator.appName;
4      document.Browser.Version.value=navigator.appVersion;
5      document.Browser.Code.value=navigator.appCodeName;
6      document.Browser.Agent.value=navigator.userAgent;
7      }
8      </script>
```

352. 面试题

Q:面试官问

js 数组去重

A:应聘者答

答案请参考[这里](#)：用 JavaScript 脚本为 Array 对象添加一个去除重复项的方法

353. 面试题

Q:面试官问

HTTP 状态码

A:应聘者答

100Continue 继续，一般在发送 post 请求时，已发送了 httpheader 之后服务端将返回此信息，表示确认，之后发送具体参数信息

200OK 正常返回信息

201Created 请求成功并且服务器创建了新的资源

202Accepted 服务器已接受请求，但尚未处理

301MovedPermanently 请求的网页已永久移动到新位置

302Found 临时性重定向

303SeeOther 临时性重定向，且总是使用 GET 请求新的 URI

304NotModified 自从上次请求后，请求的网页未修改过

400BadRequest 服务器无法理解请求的格式, 客户端不应当尝试再次使用相同的内容发起请求

401Unauthorized 请求未授权

403Forbidden 禁止访问

404NotFound 找不到如何与 URI 相匹配的资源

500InternalServerError 最常见的服务器端错误

503ServiceUnavailable 服务器端暂时无法处理请求 (可能是过载或维护)

354. 面试题

Q: 面试官问

cache-control

A: 应聘者答

网页的缓存是由 HTTP 消息头中的“Cache-control”来控制的, 常见的取值有 private、no-cache、max-age、must-revalidate 等, 默认为 private。

Expires 头部字段提供一个日期和时间, 响应在该日期和时间后被认为失效。允许客户端在这个时间之前不去检查 (发请求), 等同 max-age 的效果。但是如果同时存在, 则被 Cache-Control 的 max-age 覆盖。

Expires="Expires": "HTTP-date

例如 :

Expires: Thu, 01 Dec 1994 16:00:00 GMT (必须是 GMT 格式)

如果把它设置为 -1, 则表示立即过期

Expires 和 max-age 都可以用来指定文档的过期时间, 但是二者有一些细微差别

1. Expires 在 HTTP/1.0 中已经定义, Cache-Control: max-age 在 HTTP/1.1 中才有定义, 为了向下兼容, 仅使用 max-age 不够

2. Expires 指定一个绝对的过期时间(GMT 格式), 这么做会导致至少 2 个问题 :

2.1 客户端和服务端时间不同步导致 Expires 的配置出现问题。

2.2 很容易在配置后忘记具体的过期时间, 导致过期来临出现浪涌现象

3. max-age 指定的是从文档被访问后的存活时间, 这个时间是个相对值(比如: 3600s), 相对的是文档第一次被请求时服务器记录的 Request_time(请求时间)

4. Expires 指定的时间可以是相对文件的最后访问时间(Atime)或者修改时间(MTime), 而 max-age 相对对的是文档的请求时间(Atime)

5. 如果值为 no-cache, 那么每次都会访问服务器。如果值为 max-age, 则在过期之前不会重复访问服务器。

355. 面试题

Q: 面试官问

js 操作获取和设置 cookie

A: 应聘者答

```
1      //创建 cookie
2      functionsetCookie(name,value,expires,path,domain,secure){
3      varcookieText=encodeURIComponent(name)+'='+encodeURIComponent(value);
4      if(expiresinstanceofDate){
5      cookieText+=';expires='+expires;
6      }
7      if(path){
8      cookieText+=';expires='+expires;
9      }
10     if(domain){
11     cookieText+=';domain='+domain;
12     }
13     if(secure){
14     cookieText+=';secure';
15     }
16     document.cookie=cookieText;
17     }
18     //获取 cookie
19     functiongetCookie(name){
20     varcookieName=encodeURIComponent(name)+'=';
21     varcookieStart=document.cookie.indexOf(cookieName);
22     varcookieValue=null;
23     if(cookieStart>-1){
24     varcookieEnd=document.cookie.indexOf(';',cookieStart);
25     if(cookieEnd===-1){
26     cookieEnd=document.cookie.length;
27     }
28     cookieValue=decodeURIComponent(document.cookie.substring(cookieStart+cookieName.length,co
29     )
30     returncookieValue;
31     }
32     //删除 cookie
33     functionunsetCookie(name){
34     document.cookie=name+"=";expires="+newDate(0);
35     }
```

第 9 部分：典型开发综合 4 篇

356. 面试题

Q:面试官问

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

A:应聘者答

cookie 是网站为了标示用户身份而储存在用户本地终端（ClientSide）上的数据（通常经过加密）。

cookie 数据始终在同源的 http 请求中携带（即使不需要），记会在浏览器和服务器间来回传递。

sessionStorage 和 localStorage 不会自动把数据发给服务器，仅在本地保存。

存储大小：

cookie 数据大小不能超过 4k。

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。

有期时间：

localStorage 存储持久数据，浏览器关闭后数据不丢失除非主动删除数据；

sessionStorage 数据在当前浏览器窗口关闭后自动删除。

cookie 设置的 cookie 过期时间之前一直有效，即使窗口或浏览器关闭

357. 面试题

Q:面试官问

2.iframe 有那些缺点？

A:应聘者答

iframe 会阻塞主页面的 Onload 事件；

搜索引擎的检索程序无法解读这种页面，不利于 SEO；

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。

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，这样可以绕开以上两个问题。

358. 面试题

Q: 面试官问

如何实现浏览器内多个标签页之间的通信？

A: 应聘者答

WebSocket、SharedWorker

也可以调用 localStorage、cookies 等本地存储方式。

localStorage 另一个浏览上下文里被添加、修改或删除时，它都会触发一个事件，我们通过监听事件，控制它的值来进行页面信息通信。

注意 quirks：Safari 在无痕模式下设置 localStorage 值时会抛出 QuotaExceededError 的异常。

359. 面试题

Q: 面试官问

如何在页面上实现一个圆形的可点击区域？

A: 应聘者答

- (1) map+area 或者 svg
- (2) border-radius
- (3) 纯 js 实现需要求一个点是否在圆上简单算法、获取鼠标坐标等等

360. 面试题

Q: 面试官问

介绍一下标准的 CSS 的盒子模型？低版本 IE 的盒子模型有什么不同的？

A:应聘者答

- (1) 有两种，IE 盒子模型、W3C 盒子模型。
- (2) 盒模型：内容(content)、填充(padding)、边界(margin)、边框(border)。
- (3) 区别：IE 的 content 部分把 border 和 padding 计算了进去。

361. 面试题

Q:面试官问

CSS 优先级算法如何计算？

A:应聘者答

优先级就近原则，同权重情况下样式定义最近者为准；
载入样式以最后载入的定位为准；
优先级为：
!important>id>class>tag
important 比内联优先级高(style)

362. 面试题

Q:面试官问

为什么要使用 CSSsprites

A:应聘者答

CSSSprites 其实就是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-position”的组合进行背景定位，这样可以减少很多图片请求的开销，
因为请求耗时比较长；请求虽然可以并发，但是如果请求太多会给服务器增加很大的压力。

363. 面试题

Q: 面试官问

`display:none` 和 `visibility:hidden` 的区别?

A: 应聘者答

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

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

364. 面试题

Q: 面试官问

position 的 `absolute` 与 `fixed` 区别

A: 应聘者答

`absolute` 浮动定位是相对于父级中设置 position 为 `relative` 或者 `absolute` 最近的父级元素 `fixed` 浮动定位是相对于浏览器视窗的

365. 面试题

Q: 面试官问

IE8 以下版本的浏览器中的盒模型有什么不同?

A: 应聘者答

IE8 以下浏览器的盒模型中定义的元素的高不包括内边距和边框

366. 面试题

Q: 面试官问

常用那几种浏览器测试? 有哪些内核(LayoutEngine)?

A:应聘者答

(Q1)浏览器：IE, Chrome, FireFox, Safari, Opera。
(Q2)内核：Trident, Gecko, Presto, Webkit。

367. 面试题

Q:面试官问

说下行内元素和块级元素的区别？行内块元素的兼容性使用？（IE8 以下）

A:应聘者答

(Q1)行内元素：会在水平方向排列，不能包含块级元素，设置 width 无效，height 无效(可以设置 line-height)，
margin 上下无效，padding 上下无效。
块级元素：各占据一行，垂直方向排列。从新行开始结束接着一个断行。
(Q2)兼容性：display:inline-block;*display:inline;*zoom:1;

368. 面试题

Q:面试官问

清除浮动有哪些方式？比较好的方式是哪一种？

A:应聘者答

(Q1)
(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。
(Q2)比较好的是第 3 种方式，好多网站都这么用。

369. 面试题

Q: 面试官问

box-sizing 常用的属性有哪些？分别有什么作用？

A: 应聘者答

(Q1) box-sizing: content-box | border-box | inherit;

(Q2) content-box: 宽度和高度分别应用到元素的内容框。

在宽度和高度之外绘制元素的内边距和边框(元素默认效果)。

border-box: 元素指定的任何内边距和边框都将在已设定的宽度和高度内进行绘制。

通过从已设定的宽度和高度分别减去边框和内边距才能得到内容的宽度和高度。

370. 面试题

Q: 面试官问

Doctype 作用？标准模式与兼容模式各有什么区别？

A: 应聘者答

(Q1) <!DOCTYPE> 告知浏览器的解析器用什么文档标准解析这个文档。

DOCTYPE 不存在或格式不正确会导致文档以兼容模式呈现。

(Q2) 标准模式的排版和 JS 运作模式都是以该浏览器支持的最高标准运行。

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

371. 面试题

Q: 面试官问

HTML5 为什么只需要写 <!DOCTYPE HTML>？

A:应聘者答

HTML5 不基于 SGML，因此不需要对 DTD 进行引用，但是需要 doctype 来规范浏览器的行为

(让浏览器按照它们应该的方式来运行)。而 HTML4.01 基于 SGML,所以需要对 DTD 进行引用，

才能告知浏览器文档所使用的文档类型。

372. 面试题

Q:面试官问

页面导入样式时，使用 link 和 @import 有什么区别？

A:应聘者答

(1) link 属于 XHTML 标签，除了加载 CSS 外，还能用于定义 RSS,定义 rel 连接属性等作用；

而 @import 是 CSS 提供的，只能用于加载 CSS；

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

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

373. 面试题

Q:面试官问

介绍一下你对浏览器内核的理解？

A:应聘者答

主要分成两部分：渲染引擎(layoutengineer 或 RenderingEngine)和 JS 引擎。

渲染引擎：负责取得网页的内容（HTML、XML、图像等等）、整理讯息（例如加入 CSS 等），

以及计算网页的显示方式，然后会输出至显示器或打印机。浏览器的内核的不同对于网页的语法解释会有不同，

所以渲染的效果也不相同。所有网页浏览器、电子邮件客户端以及其它需要编辑、显示网络内容的应用程序都需要内核。

JS 引擎则：解析和执行 javascript 来实现网页的动态效果。

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

374. 面试题

Q: 面试官问

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

A: 应聘者答

(Q1) HTML5 现在已经不是 SGML 的子集，主要是关于图像，位置，存储，多任务等功能的增加。

- (1) 绘画 canvas;
- (2) 用于媒介回放的 video 和 audio 元素;
- (3) 本地离线存储 localStorage 长期存储数据，浏览器关闭后数据不丢失;
- (4) sessionStorage 的数据在浏览器关闭后自动删除;
- (5) 语义化更好的内容元素，比如 article、footer、header、nav、section;
- (6) 表单控件，calendar、date、time、email、url、search;
- (7) 新的技术 webworker, websocket, Geolocation;

(Q2) IE8/IE7/IE6 支持通过 document.createElement 方法产生的标签，

可以利用这一特性让这些浏览器支持 HTML5 新标签，

浏览器支持新标签后，还需要添加标签默认的样式。

当然也可以直接使用成熟的框架、比如 html5shim;

```
<!--[ifltIE9]>
```

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

```
<![endif]-->
```

375. 面试题

Q: 面试官问

简述一下你对 HTML 语义化的理解？

A:应聘者答

用正确的标签做正确的事情。

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

即使在没有样式 CSS 情况下也以一种文档格式显示，并且是容易阅读的；

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

使阅读源代码的人对网站更容易将网站分块，便于阅读维护理解。

376. 面试题

Q:面试官问

XHTML 和 HTML 有什么区别

A:应聘者答

HTML 是一种基本的 WEB 网页设计语言，XHTML 是一个基于 XML 的置标语言
最主要的不同：

XHTML 元素必须被正确地嵌套。

XHTML 元素必须被关闭。

标签名必须用小写字母。

XHTML 文档必须拥有根元素。

377. 面试题

Q:面试官问

什么是语义化的 HTML？

A:应聘者答

直观的认识标签对于搜索引擎的抓取有好处，用正确的标签做正确的事情！

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；

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。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。

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。

378. 面试题

Q: 面试官问

常见的浏览器内核有哪些？

A: 应聘者答

Trident 内核：IE, MaxThon, TT, TheWorld, 360, 搜狗浏览器等。[又称 MSHTML] Gecko 内核：Netscape6 及以上版本, FF, MozillaSuite/SeaMonkey 等
Presto 内核：Opera7 及以上。[Opera 内核原为：Presto, 现为：Blink;] Webkit 内核：Safari, Chrome 等。
[Chrome 的：Blink (WebKit 的分支)]

379. 面试题

Q: 面试官问

HTML5 有哪些新特性、移除了哪些元素？如何处理 HTML5 新标签的浏览器兼容问题？如何区分 HTML 和 HTML5？

A: 应聘者答

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

绘画 canvas

用于媒介回放的 video 和 audio 元素

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

sessionStorage 的数据在浏览器关闭后自动删除

语义化更好的内容元素，比如 article、footer、header、nav、section

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

新的技术 webworker, websocket, Geolocation

移除的元素

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

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

支持 HTML5 新标签：

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

可以利用这一特性让这些浏览器支持 HTML5 新标签，

浏览器支持新标签后，还需要添加标签默认的风格：

380. 面试题

Q:面试官问

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

A:应聘者答

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

381. 面试题

Q:面试官问

如何实现浏览器内多个标签页之间的通信？

A:应聘者答

调用 localStorage、cookies 等本地存储方式

382. 面试题

Q:面试官问

HTML5 为什么只需要写!DOCTYPEHTML?

A:应聘者答

HTML5 不基于 SGML，因此不需要对 DTD 进行引用，但是需要 doctype 来规范浏览器的行为

（让浏览器按照它们应该的方式来运行）而 HTML4.01 基于 SGML,所以需要对 DTD 进行引用，

才能告知浏览器文档所使用的文档类型。

383. 面试题

Q: 面试官问

Doctype 作用？标准模式与兼容模式各有什么区别？

A: 应聘者答

!DOCTYPE 声明位于 HTML 文档中的第一行，处于 html 标签之前。告知浏览器的解析器用什么文档标准解析这个文档。

DOCTYPE 不存在或格式不正确会导致文档以兼容模式呈现。

标准模式的排版和 JS 运作模式都是以该浏览器支持的最高标准运行。在兼容模式中，页面以宽松的向后兼容的方式显示，

模拟老式浏览器的行为以防止站点无法工作。

384. 面试题

Q: 面试官问

Doctype?严格模式与混杂模式-如何触发这两种模式，区分它们有何意义？

A: 应聘者答

用于声明文档使用那种规范（html/Xhtml）一般为严格模式基于框架的 html 文档。加入 XML 声明可触发，解析方式更改为 IE5.5 拥有 IE5.5 的 Bug。

385. 面试题

Q: 面试官问

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

A: 应聘者答

cookie 在浏览器和服务器间来回传递。sessionStorage 和 localStorage 不会

sessionStorage 和 localStorage 的存储空间更大；
sessionStorage 和 localStorage 有更多丰富易用的接口；
sessionStorage 和 localStorage 各自独立的存储空间；

386. 面试题

Q: 面试官问

如何实现浏览器内多个标签页之间的通信？

A: 应聘者答

调用 localStorage、cookies 等本地存储方式

387. 面试题

Q: 面试官问

CSS 实现垂直水平居中

A: 应聘者答

一道经典的问题，实现方法有很多种，以下是其中一种实现：

HTML 结构：

```
<div class="wrapper">  
<div class="content"></div>  
</div>
```

CSS：

```
.wrapper{position:relative;}  
.content{  
background-color:#6699FF;  
width:200px;  
height:200px;  
position:absolute;//父元素需要相对定位  
top:50%;  
left:50%;  
margin-top:-100px;//二分之一的 height, width  
margin-left:-100px;  
}
```

388. 面试题

Q: 面试官问

display 有哪些值？说明他们的作用。

A: 应聘者答

block 块类型。默认宽度为父元素宽度，可设置宽高，换行显示。

none 缺省值。象行内元素类型一样显示。

inline 行内元素类型。默认宽度为内容宽度，不可设置宽高，同行显示。

inline-block 默认宽度为内容宽度，可以设置宽高，同行显示。

list-item 象块类型元素一样显示，并添加样式列表标记。

table 此元素会作为块级表格来显示。

inherit 规定应该从父元素继承 display 属性的值。

389. 面试题

Q: 面试官问

行内元素有哪些？块级元素有哪些？CSS 的盒模型？

A: 应聘者答

块级元素：div,p,h1,form,ul,li;

行内元素：span>,a,label,input,img,strong,em;

CSS 盒模型：内容，border,margin, padding

390. 面试题

Q: 面试官问

CSS 引入的方式有哪些？link 和 @import 的区别是？

A: 应聘者答

内联内嵌外链导入

区别：同时加载
前者无兼容性，后者 CSS2.1 以下浏览器不支持
Link 支持使用 javascript 改变样式，后者不可

391. 面试题

Q: 面试官问

CSS 选择符有哪些?哪些属性可以继承?优先级算法如何计算?内联和 important 哪个优先级高?

A: 应聘者答

标签选择符 类选择符 id 选择符
继承不如指定 Id>class>标签选择
后者优先级高

392. 面试题

Q: 面试官问

CSS 清除浮动的几种方法（至少两种）

A: 应聘者答

使用带 clear 属性的空元素
使用 CSS 的 overflow 属性；
使用 CSS 的:after 伪元素；
使用邻接元素处理；

393. 面试题

Q: 面试官问

CSS 居中（包括水平居中和垂直居中）

A:应聘者答

内联元素居中方案

水平居中设置：

1.行内元素

设置 `text-align:center` ；

2.Flex 布局

设置 `display:flex;justify-content:center`;(灵活运用,支持 Chrome, Firefox, IE9+)

垂直居中设置：

1.父元素高度确定的单行文本（内联元素）

设置 `height=line-height` ；

2.父元素高度确定的多行文本（内联元素）

a:插入 `table`（插入方法和水平居中一样），然后设置 `vertical-align:middle` ；

b:先设置 `display:table-cell` 再设置 `vertical-align:middle` ；

块级元素居中方案

水平居中设置：

1.定宽块状元素

设置左右 `margin` 值为 `auto` ；

2.不定宽块状元素

a:在元素外加入 `table` 标签（完整的，包括 `table`、`tbody`、`tr`、`td`），该元素写在 `td` 内，然后设置 `margin` 的值为 `auto` ；

b:给该元素设置 `display:inline-block` 方法 ；

c:父元素设置 `position:relative` 和 `left:50%`，子元素设置 `position:relative` 和 `left:50%` ；

垂直居中设置：

使用 `position:absolute (fixed)` ,设置 `left`、`top`、`margin-left`、`margin-top` 的属性；

利用 `position:fixed (absolute)` 属性，`margin:auto` 这个必须不要忘记了；

利用 `display:table-cell` 属性使内容垂直居中；

使用 `css3` 的新属性 `transform:translate(x,y)`属性；

使用 `before` 元素；

394. 面试题

Q:面试官问

在书写高效 CSS 时会有哪些问题需要考虑？

A:应聘者答

参照上题“描述下“reset”CSS 文件的作用和使用它的好处”的答案。

规范命名。尤其对于没有语义化的 `html` 标签，例如 `div`，所赋予的 `class` 值要特别注意。抽取可重用的部件，注意层叠样式表的“优先级”。

395. 面试题

Q: 面试官问

html5 为什么只需要写<!doctypehtml>?

A: 应聘者答

html5 不是基于 sgml (标准通用标记语言), 不需要对 dtd 文件进行引用, 但是需要 doctype 来规范浏览器的行为,

否则浏览器将开启怪异模式, 而 html4.01 基于 sgml, 需要引入 dtd, 才能告知浏览器文档使用的文档类型

396. 面试题

Q: 面试官问

行内元素有哪些? 块级元素有哪些? 空 (void) 元素有哪些?

A: 应聘者答

行内元素有: `abspaceinputselectstrong` (强调的语气)

块级元素有: `divulollidldtddh1h2h3h4...p`

知名的空元素:

`
<hr><input><link><meta>`

鲜为人知的是:

`<area><base><col><command><embed><keygen><param><source><track><wbr>`

397. 面试题

Q: 面试官问

页面导入样式时, 使用 `link` 和 `@import` 有什么区别?

A: 应聘者答

两者都是外部引用 CSS 的方式, 但是存在一定的区别:

区别 1: link 是 XHTML 标签, 除了加载 CSS 外, 还可以定义 RSS 等其他事务; @import 属于 CSS 范畴, 只能加载 CSS。

区别 2: 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。

区别 3: 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。

区别 4: link 支持使用 Javascript 控制 DOM 去改变样式; 而 @import 不支持。

补充: @import 最优写法

@import 的写法一般有下列几种:

@import 'style.css'//WindowsIE4/NS4,MacOSXIE5,MacintoshIE4/IE5/NS4 不识别

@import "style.css"//WindowsIE4/NS4,MacintoshIE4/NS4 不识别

@import url(style.css)//WindowsNS4,MacintoshNS4 不识别

@import url('style.css')//WindowsNS4,MacOSXIE5,MacintoshIE4/IE5/NS4 不识别

@import url("style.css")//WindowsNS4,MacintoshNS4 不识别

由上分析知道, @import url(style.css)和@import url("style.css")是最优的选择, 兼容的浏览器最多。

从字节优化的角度来看 @import url(style.css)最值得推荐。

398. 面试题

Q: 面试官问

常见的浏览器内核有哪些?

A: 应聘者答

使用 Trident 内核的浏览器: IE、Maxthon、TT、TheWorld 等;

使用 Gecko 内核的浏览器: Netcape6 及以上版本、FireFox、MozillaSuite/SeaMonkey;

使用 Presto 内核的浏览器: Opera7 及以上版本;

使用 Webkit 内核的浏览器: Safari、Chrome。

399. 面试题

Q: 面试官问

html5 有哪些新特性? 移除了哪些元素? 如何处理 HTML5 新标签的浏览器兼容性问题? 如何区分 html 和 html5?

A:应聘者答

新增的元素有绘画 canvas，用于媒介回放的 video 和 audio 元素，本地离线存储 localStorage 长期存储数据，浏览器关闭后数据不丢失，而 sessionStorage 的数据在浏览器关闭后自动删除，此外，还新增了以下的几大类元素。

内容元素，article、footer、header、nav、section。

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

控件元素，webworker、websocket、Geolocation。

移出的元素有下列这些：

显现层元素：basefont，big，center，font，s，strike，tt，u。

性能较差元素：frame，frameset，noframes。

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

处理兼容问题有两种方式：

1. IE8/IE7/IE6 支持通过 document. 方法产生的标签，利用这一特性让这些浏览器支持 HTML5 新标签。

2. 使用是 html5shim 框架

另外，DOCTYPE 声明的方式是区分 HTML 和 HTML5 标志的一个重要因素，此外，还可以根据新增的结构、功能元素来加以区分。

400. 面试题

Q:面试官问

iframe 有哪些优缺点？

A:应聘者答

优点：

1. 用来实现长连接，在 websocket 不可用的时候作为一种替代，最开始由 google 发明。Comet：基于 HTTP 长连接的“服务器推”技术

2. 跨域通信。JavaScript 跨域总结与解决办法，类似的还有浏览器多页面通信，比如音乐播放器，用户如果打开了多个 tab 页，应该只有一个在播放。

3. 历史记录管理，解决 ajax 化网站响应浏览器前进后退按钮的方案，在 html5 的 historyapi 不可用时作为一种替代。

4. 纯前端的 utf8 和 gbk 编码互转。比如在 utf8 页面需要生成一个 gbk 的 encodeURIComponent 字符串，可以通过页面加载一个 gbk 的 iframe，然后主页面与子页面通信的方式实现转换，这样就不用在页面上插入一个非常巨大的编码映射表文件了

缺点：

1、在网页中使用框架结构最大的弊病是搜索引擎的“蜘蛛”程序无法解读这种页面。当“蜘蛛”程序遇到由数个框架组成的网页时，它们只看到框架而无法找到链接，因此它们会以为该网站是个死站点，并且很快转身离去。对一个网站来说这无异于一场灾难。

如果你想销售产品，你需要客户;如想得到客户，你首先要让人们访问你的网站，而要做到这一点，你就非求助于搜索引擎不可。你花费了大量的时间、精力和金钱开设了一家网上商店，却又故意不让搜索引擎检索你，这就好象开家零售商店，却将窗户全部漆成黑色，而且还不挂任何招牌一样。

2、框架结构有时会让人感到迷惑，特别是在几个框架中都出现上下、左右滚动条的时候。这些滚动条除了会挤占已经非常有限的页面空间外，还会分散访问者的注意力。访问者遇到这种网站往往会立刻转身离开。他们会想，既然你的主页如此混乱，那么网站的其他部分也许更不值得浏览。

3、链接导航问题。使用框架结构时，你必须保证正确设置所有的导航链接，如不然，会给访问者带来很大的麻烦。比如被链接的页面出现在导航框架内，这种情况下访问者便被陷住了，因为此时他没有其他地方可去。

鸣谢及备注：部分内容摘自网络，其版权归原作者所有。