

面试题库_fzq

HTML+CSS

Javascript1.0

JavaScript1.1

网络

Vue

HTML+CSS

1. 对WEB标准以及W3C的理解与认识

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

2. xhtml和html有什么区别

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

XHTML 元素必须被正确地嵌套。

XHTML 元素必须被关闭。

XHTML 区分大小写.标签名必须用小写字母。

XHTML 属性值要用双引号

XHTML 用 id 属性代替 name 属性

XHTML 特殊字符的处理

XHTML 文档必须拥有根元素。

3. 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？

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XMI声明可触发，解析方式更改为IE5.5 拥有IE5.5的bug

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

(2)、严格模式的排版和 JS 运作模式是 以该浏览器支持的最高标准运行。

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

(4)、DOCTYPE不存在或格式不正确会导致文档以混杂模式呈现。

4. 行内元素有哪些?块级元素有哪些?CSS的盒模型?

块级元素：div p h1 h2 h3 h4 form ul

行内元素：a b br i span input select input为什么可以设置宽高

Css盒模型:内容, border ,margin, padding

5. CSS引入的方式有哪些? link和@import的区别是?

页面级css; 行间样式; 外部css文件

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

(1) link属于XHTML标签, 而@import是CSS提供的;

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

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

(4) link方式的样式的权重 高于@import的权重.

6. CSS选择器有哪些?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?

所有元素可继承: visibility和cursor

内联元素可继承: letter-spacing、word-spacing、white-space、line-height、color、font、font-family、font-size、font-style、font-variant、font-weight、text-decoration、text-transform、direction。

终端块状元素可继承: text-indent和text-align。

列表元素可继承: list-style、list-style-type、list-style-position、list-style-image。

important优先级高

8. css的基本语句构成是?

选择器{属性1:值1;属性2:值2;.....}

9. 主流浏览器内核

10. 写出几种IE6 BUG的解决方法

1.双边距BUG float引起的 使用display: inline

2.3像素问题 使用float和注释引起的 使用display:inline -3px

3.超链接hover 点击后失效 使用正确的书写顺序 link visited hover active

4.le z-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:0.08 line-height:1px)

11. 标签上title与alt属性的区别是什么?

Alt 当图片不显示是 用文字代表。

Title 为该属性提供信息, 鼠标移入显示

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

a:alt(alt text):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 tool tip 显示)

title(tool tip):该属性为设置该属性的元素提供建议性的信息。

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

12. 描述css reset的作用和用途。

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

13. 解释css sprites, 如何使用。

Css 精灵 把一堆小的图片整合到一张大的图片上, 减轻服务器对图片的请求数量

定义: 一种网页图片应用处理方式

14. 浏览器标准模式和怪异模式之间的区别是什么?

盒子模型 渲染模式的不同

突出的不同是对 CSS IE盒模型缺陷的处理

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

区别是:

在严格模式中: width是内容宽度, 元素真正的宽度 = margin-left + border-left-width + padding-left + width + padding-right + border-right-width + margin-right;

在怪异模式中: width则是元素的实际宽度, 内容宽度 = width - (padding-left + padding-right + border-left-width + border-right-width)

15. 你如何对网站的文件和资源进行优化?期待的解决方案包括:

文件合并(减少http的请求)

文件最小化/文件压缩

使用CDN托管

缓存的使用

16. 什么是语义化的HTML?

1.定义: 直观的认识标签

2.好处: 对于搜索引擎的抓取有好处

17. 清除浮动的几种方式, 各自的优缺点

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after伪元素清除浮动(用于非IE浏览器)

```
1 ul{
2   border: 1px solid red;
3   zoom: 1;
4 }
5 li{
6   width: 100px;
7   height: 50px;
8   float: left;
9 }
```

```
10 ul:after{
11   content: ".";
12   display: block;
13   visibility: hidden;
14   width: 0;
15   height: 0;
16   clear: both;
17 }
```

18、编译工具的快捷键，找项目目录下的文件快捷键

19、前端后台如何进行交互

20、display有多少个属性

21、inline-block的兼容支持

22、盒子模型有多少种

23、完美清除浮动

24、对于doctype的理解

问题1：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？（2分）

问题2：Quirks模式是什么？它和Standards模式有什么区别（3-4分）

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）。

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

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。）

25、HTML标签

问题1: div、p、ul、ol、label这几个标签是什么意思？（2分）

问题2: 你怎么理解语义化标签（3-3.5分）

26、CSS初级

问题1: 有哪项方式可以对一个DOM设置它的CSS样式？（2.5分）

27、CSS选择器

问题1: CSS都有哪项选择器？（2.5分）

问题2: CSS选择器的优先级是怎样定义的？（3-4分）

28、CSS属性

问题1: 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？（2-3.5分）

问题2: CSS中，position属性是什么含义，都有哪些值，都什么含义？（3分）

取值: static、relative、fixed、absolute

默认值: static

29. 如何居中一个浮动元素？

30. 方式1:设置容器的浮动方式为相对定位，然后确定容器的宽高 比如宽500 高 300 的层，然后设置层的外边距。

方式2: 需要position:absolute;绝对定位。而层的定位点，使用外补丁margin负值的方法。负值的大小为层自身宽度高度除以二。

31. 描述CSS hack技巧？

条件注释: 仅适用于IE

特定符号: 适用于能识别特定符号的浏览器

内核符号: 针对不同浏览器内核

32. html5\CSS3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？

33. 你怎么来实现页面设计图，你认为前端应该如何高质量完成工作？一个满屏 品 字布局 如何设计？

首先划分成头部、body、脚部；

实现效果图是最基本的工作，精确到2px；

与设计师，产品经理的沟通和项目的参与

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 HTML5。

34. 页面重构怎么操作？

编写 CSS、让页面结构更合理化，提升用户体验，实现良好的页面效果和提升性能。

35. 语义化的理解？

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，便于阅读维理解。

36. HTML5的离线储存?

localStorage 长期存储数据，浏览器关闭后数据不丢失；

sessionStorage 数据在浏览器关闭后自动删除。

37. 为什么要初始化CSS样式。

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

当然，初始化样式会对SEO有一定的影响，但鱼和熊掌不可兼得，但力求影响最小的情况下初始化。

38. 解释下浮动和它的工作原理？清除浮动的技巧

39. 用过媒体查询，针对移动端的布局吗？

40. 使用 CSS 预处理器吗？喜欢那个？

41. 经常遇到的CSS的兼容性有哪些？原因，解决方法是什么？

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

42. 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：

43. 自我评价一下HTML/CSS/JS的掌握情况

44. div+css 的布局较 table 布局有什么优点？

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。

45. 简述一下 src 与 href 的区别。

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。

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 Hypertext Reference 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

```
1 <link href="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。

46. 一个盒子垂直水平居中有哪几种方法

方法一：利用定位

```
1 .parent{
2   position: relative;
3 }
```

```
4 .child{
5   position:absolute;
6   top:50%;
7   left:50%;
8   margin-top:-50px;
9   margin-left:-50px;
10 }
```

方法二：利用margin:auto

```
1 .parent{
2   position:relative;
3 }
4 .child{
5   position:absolute;
6   margin:auto;
7   top:0;
8   left:0;
9   right:0;
10  bottom:0;
11 }
```

方法三：利用display:table-cell

```
1 .parent{
2   display:table-cell;
3   vertical-align:middle;
4   text-align:center;
5 }
6 .child{
7   display:inline-block;
8 }
```

方法四：利用display: flex;设置垂直水平都居中

```
1 .parent{
```

```
2 display:flex;
3 justify-content:center;
4 align-items:center;
5 }
```

方法五：计算父盒子与子盒子的空间距离(这跟方法一是一个道理)

```
1 .child{
2   margin-top:200px;
3   margin-left:200px;
4 }
```

方法六：利用transform

```
1 .parent{
2   position:relative;
3 }
4 .child{
5   position:absolute;
6   top:50%;
7   left:50%;
8   transform:translate(-50%,-50%);
9 }
```

方法七：利用calc计算

```
1 .parent{
2   position:relative;
3 }
4 .child{
5   position:absolute;
6   top:calc(200px);// (父元素高-子元素高) ÷ 2=200px
7   left:calc(200px);// (父元素宽-子元素宽) ÷ 2=200px
8 }
```


47. flex:1 是什么意思

<https://www.runoob.com/cssref/css3-pr-flex.html>

48. rem为什么可以实现自适应布局

https://blog.csdn.net/qq_42707446/article/details/93200711

Javascript1.0

1. javascript的typeof返回哪些数据类型

Object number function boolean undefined

2. 例举3种强制类型转换和2种隐式类型转换?

强制 (parseInt,parseFloat,number)

隐式 (== - ===)

3. split() join() 的区别

前者是切割成数组的形式，后者是将数组转换成字符串

4. 数组方法pop() push() unshift() shift()

Push()尾部添加 pop()尾部删除

Unshift()头部添加 shift()头部删除

5. 事件绑定和普通事件有什么区别

事件绑定就是针对dom元素的事件，绑定在dom元素上

普通事件即为非针对dom元素的事件

6. IE和DOM事件流的区别

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
- 7. IE和标准下有哪些兼容性的写法

```
1 var ev = ev || window.event;
2 document.documentElement.clientWidth || document.body.clientWidth
3 var target = ev.srcElement || ev.target
```

- 8. ajax请求的时候get 和post方式的区别
- 9. call和apply的区别
Object.call(this,obj1,obj2,obj3)
Object.apply(this,arguments)
- 10. ajax请求时, 如何解释json数据
使用eval parse 鉴于安全性考虑 使用parse更靠谱
- 11. b继承a的方法

```
1 function A(name){
2   this.name = name;
3   this.sayHello = function(){alert(this.name+" say Hello!");};
4 }
5 function B(name,id){
6   this.temp = A;
7   this.temp(name);      //相当于new A();
8   delete this.temp;
9   this.id = id;
10  this.checkId = function(ID){alert(this.id==ID)};
11 }
```

- 12. 写一个获取非行间样式的函数

```
1 function getStyle(obj, attr, value) {
2   if (!value) {
3     if (obj.currentStyle) {
4       return obj.currentStyle[attr]
5     }
6     else {
7       obj.getComputedStyle(attr, false)
```

```

8     }
9     }
10    else {
11        obj.style[attr] = value
12    }
13 }

```

13. 事件委托是什么

让利用事件冒泡的原理，让自己的所触发的事件，让他的父元素代替执行！

14. 闭包是什么，有什么特性，对页面有什么影响

闭包就是能够读取其他函数内部变量的函数。

15. 如何阻止事件冒泡和默认事件

cancelBubble return false

16. 添加 删除 替换 插入到某个接点的方法

obj.appendChild()

obj.insertBefore

obj.replaceChild

obj.removeChild

17. 解释jsonp的原理，以及为什么不是真正的ajax

动态创建script标签，回调函数

Ajax是页面无刷新请求数据操作

18. javascript的本地对象，内置对象和宿主对象

本地对象为array obj regexp等可以new实例化

内置对象为global Math 等不可以实例化的

宿主为浏览器自带的document>window 等

19. document load 和document ready的区别

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)

20 .javascript的同源策略

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

21. 编写一个数组去重的方法

```

1 var s=[1,2,3,1,5,2,12,2,1,,4,1,2];
2 for(var i=0,a=[],b=[],c=0;i<s.length;i++){
3     if(a[s[i]]){
4         c++
5     }else{
6         a[s[i]]=1;

```

```
7     b.push(s[i])
8 }
9 }
10 alert(b)
```

- 22、i++和++i区别
- 23、XMLHttpRequest是什么
- 24、正则表达式/d/w/s分别是什么意思
- 25、闭包和其作用
- 26、框架比较jquery prototype YUI
- 27、写下js里面的基础对象和基础数据类型都有什么
- 28、面向对象编程三大特性是什么
- 29、写一个冒泡排序
- 30、http常用的几种状态
- 31、网速性能优化的方法
- 32、截取字符串abcdefg的efg
- 33、判断一个字符串中出现次数最多的字符，统计次数
- 34、规避javascript多人开发函数重名问题
- 35、编写一个方法，求一个字符串的字节长度
- 36、编写一个方法，去掉一个数组的重复元素
- 37、写出3个使用this的典型应用
- 38、javascript中如何检测一个变量是一个string类型
- 39、javascript有哪几种数据类型
- 40、下面css标签在javascript中调用应如何拼写，border-left-color，moz-viewport
- 41、javascript中如何对一个对象进行深度clone
- 42、如何控制alert中的换行
- 43、请实现，鼠标点击页面中的任意标签，alert该标签的名称（注意兼容性）
- 45、请给出异步加载js方案
- 46、请设计一套方案，确保页面js加载完全
- 47、js中如何定义class，如何扩展prototype
- 48、如何添加HTML元素的事件，有几种方法？
- 49、document.write和innerHTML的区别
- 50、多浏览器检测通过什么
- 51、如何控制网页在网络传输过程中的数据量
- 52、flash、ajax各自的优缺点，在使用中如何取舍？
- 53、列出前端开发的优化方案

HTML部分

语义化HTML：好处在于可以使代码简洁清晰，支持不同设备，利于搜索引擎，便于团队开发；
减少DOM节点：加速页面渲染；

给图片加上正确的宽高值：这可以减少页面重绘，同时防止图片缩放；

防止src属性和link的href属性为空：当值为空时，浏览器很可能会把当前页面当成其属性值加载；

正确的闭合标签：如避免使用

CSS部分

避免使用 CSS Expressions(CSS表达式)：如background-color: expression((new Date()).getHours()%2 ? "#B8D4FF" : "#F08A00") ；

避免使用 CSS Filter (CSS滤镜) ；

使用CSS缩写，减少代码量；

通过CSSSprites把同类图片合成一张，减少图片请求；

减少查询层级：如.header .logo要好过.header .top .logo；

减少查询范围：如.header>li要好过.header li；

避免TAG标签与CLASS或ID并存：如a.top、button#submit；

删除重复的CSS；

Javascript部分

尽量少用全局变量；

使用事件代理绑定事件，如将事件绑定在body上进行代理；

避免频繁操作DOM节点；

不使用EVAL；

减少对象查找，如a.b.c.d这种查找方式非常耗性能，尽可能把它定义在变量里；

类型转换：把数字转换成字符串使用"" + 1，浮点数转换成整型使用Math.floor()或者Math.round()；

对字符串进行循环操作，譬如替换、查找，应使用正则表达式；

删除重复的JS；

服务器部分

尽量合并CSS、JS文件，或将其直接写在页面上，减少HTTP请求；

压缩CSS、JS文件，缩短文件传输时间；

避免404错误：特别要避免给404指定一个停摆页面，否则所有404错误都将会加载一次页面；

一般要求减少DNS查询次数，如同一个页面的请求资源尽量少的使用不同的主机名，这可以减少网站并行下载的数量，但很多网站为了加速下载资源其实是特意用了多个主机名，这里要做一个权衡；

使用CDN加速，使用户从离自己最近的服务器下载文件；

减少Cookie的大小，使用无cookie的域，客户端请求静态文件的时候，减少 Cookie 的反复传输对主域名的影响；

为文件头指定Expires，使内容具有缓存性；

使用gzip压缩内容；

(1) 减少http请求次数：CSS Sprites, 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CSS Expression (css表达式)又称Dynamic properties(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。
- (8) 避免在页面的主体布局中使用table, table要等其中的内容完全下载之后才会显示出来, 显示比div+css布局慢。

54、移动端和pc端的区别

55、浏览兼容

56、jsonp和其他跨域方法

57、h5的优缺点

58、数据类型

59、手写c3动画效果

60、手写原生js轮播图大致思路

61、angular五大特性

62、工作中用node.js做什么用

63、JavaScript初级

问题1: JavaScript是一门什么样的语言, 它有哪些特点? (2-4分)

问题2: JavaScript的数据类型都有什么? (2-3分)

问题3: 如何判断一个变量是什么类型的? (2-3分)

64、JavaScript DOM

问题1: 已知ID的Input输入框, 希望获取这个输入框的输入值, 怎么做? (1.5分) (不使用第三方框架)

问题2: 希望获取到页面中所有的checkbox怎么做? (3分) (不使用第三方框架)

问题3: 设置一个已知ID的DIV的html内容为xxxx, 字体颜色设置为黑色 (3分) (不使用第三方框架)

65、JavaScript事件

问题1: 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? (2-3分)

问题2: 如何获取用户在输入框里按下了回车键? (3分)

问题3: JavaScript的事件流模型都有什么? (3.5分)

66、Ajax

问题1: 什么是Ajax (2.5分)

问题2: 什么是JSON, 它有什么优点 (2.5分)

问题3: Ajax有什么优缺点 (2-4分)

67、基本数据类型细节

问题1

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

问题2

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

问题3

```
var undefined;
undefined == null; // true
1 == true; // true
2 == true; // false
0 == false; // true
0 == ''; // true
NaN == NaN; // false
[] == false; // true
[] == ![]; // true
引用数据类型小题目1
```

```
1 var a = new Object();
2 a.value = 1;
3 b = a;
4 b.value = 2;
5 alert(a.value);
```

引用数据类型小题目2

```
1 var a = new Object();
2 a.value = 1;
3 b = a;
4 a.value = 2;
5 alert(b.value);
```

引用数据类型小题目3

```
1 function test(obj) {
2   obj.value = 1;
3   obj = new Object();
4   obj.value = 3;
5 }
6 var a = new Object();
```

```
7 a.value = 2;
8 test(a);
9 alert(a.value);
```

引用数据类型小题目4

已知数组var stringArray = ["This", "is", "Baidu", "Campus"]

Alert出"This is Baidu Campus"

```
1 alert(stringArray.join(""))
```

引用数据类型小题目5

var numberArray = [3,6,2,4,1,5];

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]

引用数据类型小题目5

var keyArray = [{id:3, name:"Simon"}, {id:2, name:"Andy"}, {id:4, name:"Candy"}]

实现对该数组按照id值升序排序，输出[{id:2, name:"Andy"}, {id:3, name:"Simon"}, {id:4, name:"Candy"}]

引用数据类型小题目6

输出今天的日期，以YYYY-MM-DD的方式，比如今天是2011年9月26日，则输出2011-09-26

引用数据类型小题目6

将字符串"{name}"中的{name}替换成Tony

引用数据类型小题目6

将字符串"Apple AppStore Webapp"中的app（不区分大小写）替换成为APP

引用数据类型小题目7

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

引用数据类型小题目8

实现一个函数objectClone，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

问题系列四：JavaScript作用域

作用域小题目1

```
1 var a = 1;
2 function test() {
3   alert(a);
4   var a = 2;
5   alert(a);
```



```
6 }  
7 test();
```

作用域小题目2

```
1 function factory() {  
2   var a = 1;  
3   var fn = function () {  
4     alert(a);  
5   }  
6   return fn;  
7 }  
8 function run(para) {  
9   var a = para;  
10  var func = factory ();  
11  func();  
12 }  
13 run(2);
```

68、Web前端性能

提高web页面传输性能都有哪些方法？（2-4分）

69、想实现一个对页面某个节点的拖拽，如何做？（3-4分）

回答出概念即可，下面是几个要点

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

问题系列二：如何在页面实现一个倒计时的功能？（3分）

回答出概念即可，下面是几个要点

利用setTimeout或者setinterval设置定时器，可以追问一下这二者的区别

然后在每个定时器触发时获取当前时间并输出到页面

问题系列三：数组去重（3分）

HTML文件中有一个多行文本输入框：，允许用户在其中输入自己的兴趣爱好，用户有可能用空格，回车，逗号（全半角都可能），顿号来分隔，写一个函数getUserHobbies()，功能是从输入框中拿到用户输入，经过处理后返回一个没有重复的用户兴趣的字符串数组：例如：

用户输入

70. IE6的双倍边距BUG指的是什么？怎么解决？

双边距：当块级元素有浮动样式的时候，给元素添加margin-left和margin-right样式，在ie6下就会出现双倍边距。

解决方案：给当前元素添加样式，使当前元素不为块，如：`display:inline;display:list-item`

71. 如果制作一个访问量很大的网站，对css，js和图片应该怎么处理？

方法1：资源文件按模块进行放置，有利于团队开发

方法2：图片尽量采取聚合技术

方法3：精简压缩css和js文件，减少文件大小

方法4：类库、媒体使用CDN加速，减轻服务器压力

72. 述ajax原理，什么是同步异步(主观题，答案不唯一)？

Ajax的工作原理：相当于在用户和服务器之间加了一个中间层，使用户操作与服务器响应异步化。这样把以前的一些服务器负担的工作转嫁到客户端，利于客户端闲置的处理能力来处理，减轻服务器和带宽的负担，从而达到节约ISP的空间及带宽租用成本的目的。

73. 平时有没有使用xml和json，在ajax交互中，哪一种更易于开发和维护，js中怎么序列化JSON字符串？

有，json相比xml可读性和可扩展性好、编码及解码难度较低、在数据交互中带宽占用少，并且在当下是最流行的数据交互格式。

序列化JSON字符串：`eval()` 或者 `JSON.parse()`

74. JS怎么创建一个类？

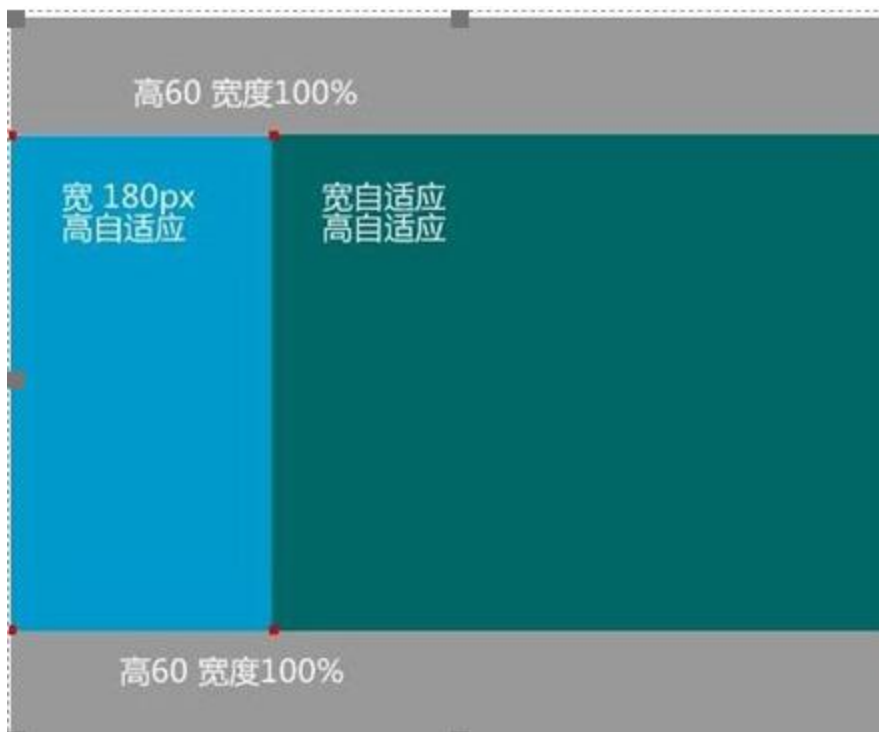
方式1：`var obj = new Object();`

方式2：`var obj = {};`

75. JS的typeof返回哪些数据类型？

string、number、object、boolean、function、undefined

76. 完成下面布局（兼容IE6-10以及FF、谷歌、苹果浏览器）？



```
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
5     <title></title>
6     <style>
7       body {
8         margin: 0;
9         padding: 0;
10      }
11
12      #body {
13        background-color: #0099CB;
14        width: 100%;
15        float: left;
16      }
17
18      #header {
19        height: 60px;
20        background-color: #999999;
21      }
22
23      #content {
24        background-color: #006766;
25        margin-left: 180px;
26      }
27
28      #sidebar {
29        width: 180px;
30        float: left;
31      }
32
33      #footer {
34        clear: both;
35        height: 60px;
36        background-color: #999999;
37      }
38    </style>
```

```

39  </head>
40  <body>
41    <div id="header"></div>
42    <div id="body">
43      <div id="sidebar"></div>
44      <div id="content"></div>
45    </div>
46    <div id="footer"></div>
47  </body>
48 </html>

```

77. 闭包是什么？有什么特性？请简单书写一个简单事例？

必包：闭包是指可以包含自由（未绑定到特定对象）变量的代码块；这些变量不是在这个代码块内或者任何全局上下文中定义的，而是在定义代码块的环境中定义（局部变量）

特性：闭包是能够读取其他函数内部变量的函数，即在外面可以调用函数中的函数的变量，其实他就是将函数内外部连接起来的桥梁

78. 解释jsonp的原理，以及为什么不是真正的ajax（主观题）？

JSONP是一种非正式传输协议，该协议的一个要点就是允许用户传递一个callback参数给服务端，然后服务端返回数据时会将这个callback参数作为函数名来包裹住JSON数据，这样客户端就可以随意定制自己的函数来自动处理返回数据了

79. js延迟加载的方式有哪些？

defer和async、动态创建DOM方式（用得最多）、按需异步载入js

80. JavaScript原型，原型链？有什么特点？

81. eval是做什么的？

它的功能是把对应的字符串解析成JS代码并运行；

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）。

82. 写一个通用的事件侦听器函数

83. 99%的网站都需要被重构是那本书上写的？

网站重构：应用web标准进行设计（第2版）

84. 什么叫优雅降级和渐进增强？

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。

区别

a. 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

b. 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

c. 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

85. Node.js的适用场景？

高并发、聊天、实时消息推送

86. .WEB应用从服务器主动推送Data到客户端有那些方式

html5 websocket

WebSocket通过Flash

XHR长时间连接

XHR Multipart Streaming

不可见的Iframe

87. 谈谈This对象的理解？

this是js的一个关键字，随着函数使用场合不同，this的值会发生变化。

但是总有一个原则，那就是this指的是调用函数的那个对象。

this一般情况下：是全局对象Global。作为方法调用，那么this就是指这个对象

86. 事件、IE与火狐的事件机制有什么区别？如何阻止冒泡？

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、火狐是 事件捕获；

ev.stopPropagation();

87. 如何判断一个对象是否属于某个类？

使用instanceof

88. JSON 的了解？

JSON(JavaScript Object Notation) 是一种轻量级的数据交换格式。它是基于JavaScript的一个子集。

数据格式简单，易于读写，占用带宽小

{'age':'12', 'name':'back'}

89. 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？

同步是阻塞模式，异步是非阻塞模式。

同步就是指一个进程在执行某个请求的时候，若该请求需要一段时间才能返回信息，那么这个进程将会一直等待下去，直到收到返回信息才继续执行下去；

异步是指进程不需要一直等下去，而是继续执行下面的操作，不管其他进程的状态。当有消息返回时系统会通知进程进行处理，这样可以提高执行的效率。

90. 模块化怎么做？

91. 对Node的优点和缺点提出了自己的看法？

优点：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。此外，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：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，而且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。

92. 异步加载的方式？

(1) defer，只支持IE

(2) async：

(3) 创建script，插入到DOM中，加载完毕后callBack

document.write和 innerHTML的区别

document.write只能重绘整个页面

innerHTML可以重绘页面的一部分

93. 代码

```
1 (function (x) {  
2   delete x;  
3   alert(x);  
4 })(1 + 5);
```

函数参数无法delete删除，delete只能删除通过for in访问的属性。

当然，删除失败也不会报错，所以代码运行会弹出“1”。

94. JQuery与jQuery UI 有啥区别？

jQuery是一个js库，主要提供的功能是选择器，属性修改和事件绑定等等。

jQuery UI则是在jQuery的基础上，利用jQuery的扩展性，设计的插件。

提供了一些常用的界面元素，诸如对话框、拖动行为、改变大小行为等等

95. jquery 中如何将数组转化为json字符串，然后再转化回来？

jQuery中没有提供这个功能，所以你需要先编写两个jQuery的扩展：

```
1 $.fn.stringifyArray = function(array) {  
2   return JSON.stringify(array)  
3 }
```

```
1 $.fn.parseArray = function(array) {  
2   return JSON.parse(array)  
3 }
```

然后调用

```
1 $('').stringifyArray(array)
```

96. 102.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）

document.getElementById(“ID”).value

97. 希望获取到页面中所有的checkbox怎么做？（不使用第三方框架）

```

1 var domList = document.getElementsByTagName('input')
2 var checkBoxList = [];
3 var len = domList.length;    //缓存到局部变量
4 while (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 }

```

98. 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)

```

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

```

99. 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？

直接在DOM里绑定事件：

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

100. 什么是Ajax和JSON，它们的优缺点？

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）

JavaScript1.1

1. JavaScript是一门什么样的语言，它有哪些特点？

没有标准答案。

2. JavaScript的数据类型都有什么？

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

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

3. 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.obj instanceof Array 在某些IE版本中不正确

方法三.方法一二皆有漏洞，在ECMA Script5中定义了新方法Array.isArray()，保证其兼容性，最好的方法如下：

```
1 if (typeof Array.isArray === "undefined") {
2   Array.isArray = function (arg) {
3     return Object.prototype.toString.call(arg) === "[object Array]"
4   };
5 }
```

4. 为什么typeof(a)---undefined

Undefined是一个只有一个值的数据类型，这个值就是“undefined”

5. (typeof a); //object

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”。

6. 代码

```
1 var foo = "11"+2-"1";
2 console.log(foo); //111
3 console.log(typeof foo); //String
```

7. 代码

```
1 var a = new Object();
2 a.value = 1;
3 b = a;
4 b.value = 2;
5 alert(a.value); //2
```


8. 已知有字符串foo="get-element-by-id",写一个function将其转化成驼峰表示法"getElementById"。

```
1 function combo(msg){
2   var arr=msg.split("-");
3   for(var i=1;i<arr.length;i++){
4     arr[i]=arr[i].charAt(0).toUpperCase()+arr[i].substr(1,arr[i].length-1);
5   }
6   msg=arr.join("");
7   return msg;
8 }
```

9. .var numberArray = [3,6,2,4,1,5]; (考察基础API)

实现对该数组的倒排, 输出[5,1,4,2,6,3]

实现对该数组的降序排列, 输出[6,5,4,3,2,1]

```
1 function combo(msg){
2   var arr=msg.split("-");
3   for(var i=1;i<arr.length;i++){
4     arr[i]=arr[i].charAt(0).toUpperCase()+arr[i].substr(1,arr[i].length-1);
5   }
6   msg=arr.join("");
7   return msg;
8 }
```

10. 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

```
1 var d = new Date();
2 // 获取年, getFullYear()返回4位的数字
3 var year = d.getFullYear();
4 // 获取月, 月份比较特殊, 0是1月, 11是12月
5 var month = d.getMonth() + 1;
6 // 变成两位
7 month = month < 10 ? '0' + month : month;
8 // 获取日
9 var day = d.getDate();
```

```
10 day = day < 10 ? '0' + day : day;
11 alert(year + '-' + month + '-' + day);
```

11. 将字符串“{name}”中的{name}替换成Tony （使用正则表达式）

```
1 "{id}_{$name}".replace(/{$id}/g, '10').replace(/{$name}/g, 'Tony')
;
```

12. 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

```
1 function escapeHtml(str) {
2   return str.replace(/[<>"&]/g, function(match) {
3     switch (match) {
4       case "<":
5         return "<";
6       case ">":
7         return ">";
8       case "&":
9         return "&";
10      case "\"":
11        return "\"";
12    }
13  });
14 }
```

13. foo = foo||bar ，这行代码是什么意思？为什么要这样写？

答案：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。

14. 代码

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

```
6 }
```

14. 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

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

15. 把两个数组合并，并删除第二个元素。

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

16. 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）

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

17. 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}。

```
1 function  serilizeUrl(url) {
2   var    result = {};
3   url    = url.split("?")[1];
4   var    map = url.split("&");
5   for(var  i = 0, len = map.length; i < len; i++) {
6     result[map[i].split("=")[0]] = map[i].split("=")[1];
7   }
8   return  result;
9 }
```

18. 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？

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：

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

19. 代码

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

Javascript事件处理器在线程空闲之前不会运行。追问，如何让上述代码输出1 2 3？

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

20. 写一个function，清除字符串前后的空格。（兼容所有浏览器）

使用自带接口trim()，考虑兼容性：

```
1 if (!String.prototype.trim) {
2   String.prototype.trim = function() {
3     return this.replace(/^\s+/, "").replace(/\s+$/, "");
4   }
5 }
6
7 // test the function
8 var str = " \t\n test string ".trim();
9 alert(str == "test string"); // alerts "true"
```

21. Javascript中callee和caller的作用？

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）

```
1 var result=[];
2 function fn(n){ //典型的斐波那契数列
3   if(n==1){
4     return 1;
5   }else if(n==2){
6     return 1;
7   }else{
8     if(result[n]){
9       return result[n];
10    }else{
11      //argument.callee()表示fn()
12      result[n]=arguments.callee(n-1)+arguments.callee(n-2);
13      return result[n];
14    }
15  }
16 }
```

22. px和em的区别

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

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

23. 对JSON 的了解？

JSON(JavaScript Object Notation) 是一种轻量级的数据交换格式。它是基于JavaScript的一个子集。数据格式简单，易于读写，占用带宽小{'age':'12', 'name':'back'}

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

hasOwnProperty

25. 函数的几种定义方法？

```
1 function a() {},  
2 var a = function() {}
```

26. 对象的定义方法？

```
1 a = new Object(), a = {}
```

27. 类的定义方法（prototype）（继承）

```
1 var a = function() {}  
2 a.prototype = {}  
3 new a();
```

28. 常用JS框架都有什么？是否使用过jQuery，以及jQuery的优点是什么？

29. JS用了多久，介绍一下自己做过的JS项目？

30. 开发调试工具和方法都有什么（编辑器、浏览器）

31. 类、函数、对象（代码表达）

32. 闭包（setTimeout）（产生两个首尾相连的计时器）（使用for循环产生10个计时器）||

33. JQuery Mobile 相关

34. HTML5/CSS3的掌握情况

35. 是否听说过和理解webapp？

36. 个人擅长的语言，优缺点都是什么？

37. 介绍一下曾经参与过的项目经验，合作开发、独立开发

38. 编程的重要知识？

39. 学习遇到的困难怎么解决

40. 数组去重

```
1 var arr = [1,1,1,1,2,2,2,2,2,1,1,1,0,0];
2 Array.prototype.unique = function() {
3     var temp = {},
4     arr = [],
5     len = this.length;//也是优化，不用每次都this了
6     for(var i = 0; i < len; i++){
7         if(!temp[this[i]]) {//如果有0，!0==true，所以还是"abc"吧
8             temp[this[i]] = "abc";
9             arr.push(this[i]);
10        }
11    }
12    return arr;
13 }
```

41. 想实现一个对页面某个节点的拖曳？如何做？（使用原生JS）。

42. 在Javascript中什么是伪数组？如何将伪数组转化为标准数组？

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。

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

43. 手写数组快速排序

关于快排算法的详细说明，可以参考阮一峰老师的文章快速排序

“快速排序”的思想很简单，整个排序过程只需要三步：

(1) 在数据集之中，选择一个元素作为”基准”（pivot）。

(2) 所有小于”基准”的元素，都移到”基准”的左边；所有大于”基准”的元素，都移到”基准”的右边。

(3) 对”基准”左边和右边的两个子集，不断重复第一步和第二步，直到所有子集只剩下一个元素为止。

44. 统计字符串”aaaabbbbccccdddfgh”中字母个数或统计最多字母数。

```

1 var str = "aaaabbbccccdddfgh";
2 var obj = {};
3 for(var i=0; i<str.length; i++){
4     var v = str.charAt(i);
5     if(obj[v] & obj[v].value == v){
6         obj[v].count = ++ obj[v].count;
7     }else{
8         obj[v] = {};
9         obj[v].count = 1;
10        obj[v].value = v;
11    }
12 }
13 for(key in obj){
14     document.write(obj[key].value + '=' + obj[key].count + ' '); // a=4
15 }

```

45. 写一个function, 清除字符串前后的空格。(兼容所有浏览器)

```

1 function trim(str) {
2     if (str & typeof str === "string") {
3         return str.replace(/(^s)|($s)/g, ""); //去除前后空白符
4     }
5 }

```


网络

1. http状态码有那些？ 分别代表是什么意思？
 2. 平时如何管理你的项目， 如何设计突发大规模并发架构？
- 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 功能为准英文翻译；

图片采用整合的 images.png png8 格式文件使用 尽量整合在一起使用方便将来的管理

3. 那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）

4. 你说你热爱前端，那么应该WEB行业的发展很关注吧？说说最近最流行的一些东西吧？

Node.js、Mongodb、npm、MVVM、MEAN

5. 你了解我们公司吗？说说你的认识？

因为我想去阿里，所以我针对阿里的说

最羡慕就是在双十一购物节，350.19亿元，每分钟支付79万笔。海量数据，居然无一漏单、无一故障。太厉害了。

6. absolute的containing block计算方式跟正常流有什么不同？

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

8. 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？（流程说的越详细越好）

9. 除了前端以外还了解什么其它技术么？你最厉害的技能是什么？

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

11. 谈谈你认为怎样做能是项目做的更好？

12. 你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？

13. 移动端（比如：Android IOS）怎么做好用户体验？

14. 加班的看法：加班就像借钱，原则应当是-----救急不救穷

15. HTML5的离线储存有几种方式？

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

16. iframe有那些缺点？

iframe会阻塞主页面的Onload事件；

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，这样可以可以绕开以上两个问题。

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

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；

18. 为什么利用多个域名来存储网站资源会更有效？

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

19. 请谈一下你对网页标准和标准制定机构重要性的理解。

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。

20. 三次握手和四次挥手

21. 常见状态码

22. http的8种请求方式及区别

Vue

1. axios底层是怎么实现的

<https://blog.csdn.net/luchuanqi67/article/details/81329358>

<https://www.jianshu.com/p/8bc48f8fde75>

2. Vue的生命周期

created和mounted的区别

<https://cn.vuejs.org/v2/guide/instance.html>

3. Vue之间父子组件传值

父传子、子传父、兄弟组件之间的传值 <https://cn.vuejs.org/v2/guide/components-props.html>

<https://www.jianshu.com/p/af9cb05bfbafe> [https://www.cnblogs.com/chenyi-yi-p/11152391.html](https://www.cnblogs.com/chenyi-yi/p/11152391.html)

4. keep-alive的作用

<https://www.jianshu.com/p/9523bb439950>

5. Vue的双向绑定原理

<https://www.jianshu.com/p/78b31df97b70>

大场css会问实现一个啥，不再是那些死记应背的东西，一定要自己去灵活运用css知识，BFC、伪类、flex等，也会问懒加载和延迟加载、回流和重绘等，问js必问ES6 promise await this 继承 手撕节流和防抖、手撕二叉树的前中后（非递归）平衡二叉树的左右旋、基本的排序和查找、动态归划、贪心等 proxy也会问，深拷贝浅拷贝，垃圾回收以及它的算法，事件绑定、事件监听、事件委托，设计模式、手撕单例、工厂等，函数式编程、跨域、加密（Token、JWT）缓存cookie与session、xss和csrf攻击、https与http、了解SSL和TLS、了解SSR和SEO、前端测试怎么做、TCP与UDP DNS 问VUE必问webpack，git的常用指令、怎样实现版本回退、Linux常用指令、数据库了解吗？SQL基本操作、MongoDB、MySQL等