

web前端经典面试题（2019）汇总

Html5&CSS3

1、谈谈你对web标准的理解

[web标准的理解](#)

2、列举html中至少三个实体

&、 、<、>、®、© 详细请看w3schoolHTML实体符号参考手册 [w3schoolHTML实体符号参考手册](#)

3、 cellpadding与cellspacing有何区别？

cellpadding：代表单元格边框到内容之间的距离（留白） cellpadding： cellpadding属性用来指定表格各单元格之间的空隙。此属性的参数值是数字，表示单元格间隙所占的像素点数。

4、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（li a） 7.通配符选择器（*） 8.属性选择器（a[rel = "external"]） 9.伪类选择器（a: hover, li: nth - child） **可继承的样式：** font-size font-family color, UL LI DL DD DT; **不可继承的样式：** border padding margin width height **优先级** 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 **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。

5、 display与visibility有何异同？

display可以有很多值， visibility只有两个常用值： visible、hidden。 当display为none、 visibility为hidden时都会隐藏元素。但display会隐藏掉元素空间， visibility会保留元素空间。

6、怎么在网页中实现绝对定位？

absolute [绝对定位与相对定位](#)

7、 table-layout、 border-collapse有何用途？

①table-layout： 设置表格是否自动调整宽高 ②border-collapse： 表格与单元格及单元格间的边框是否融合在一起。

8、简述盒模型

9、链接标记target属性的top、parent、_blank、main、left、top各有何用处？

[链接标记target与Dom, Bom](#)

10、你对浏览器兼容怎么看？通常你都做哪些处理？

[浏览器的兼容性](#)

11、get和post的区别？

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：①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）。②发送包含未知字符的用户输入时，POST比GET更稳定也更可靠。

12、xhtml和html有什么区别

①HTML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 ②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。

13、行内元素有哪些?块级元素有哪些?空(void)元素有那些？

行内元素和块级元素的区别是什么？行内块元素的兼容性使用？（IE8以下）

块级元素：div p h1 h2 h3 h4 form ul ol dl dt dd **行内元素**：a b br i span input select image strong（强调的语气）**常见空元素**：
/<hr>//<input>/<link>/<meta> **不常见空元素**：

<area>/<base>/<col>/<command>/<embed>/<keygen>/<param>/<source>/<track>/<wbr> **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。**兼容性**：display:inline-block;*display:inline;*zoom:1;

14、CSS引入的方式有哪些? link和@import的区别是？

CSS引入的方式包括内联 内嵌 外链 导入 link和@import的区别是：①link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；③import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；④后者优先级更高

15、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?介绍一下你对浏览器内核的理解

ie(ie内核) 火狐 (Gecko) 谷歌 (webkit) opera(blink) **对内核的理解**：主要分成两部分：渲染引擎(layout engineer或Rendering Engine)和JS引擎。①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。②JS引擎：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。

16、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。适用于以下情况：①静态图片，不随用户信息的变化而变化 ②小图片，图片容量比较小 ③加载量比较大

17、清除浮动的几种方式，各自的优缺点

(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。比较好的是第3种方式，好多网站都这么用。

18、Doctype作用？标准模式与兼容模式各有什么区别？

①<!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。②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。**注意点：**HTML5只需要写<!DOCTYPEHTML>不需要对DTD进行引用，因为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DTD进行引用，才能告知浏览器文档所使用的文档类型。

19、iframe有哪些缺点？

①iframe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；②iframe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，这样可以避开以上两个问题。

20、如何实现浏览器内多个标签页之间的通信？

①通过WebSocket、SharedWorker来实现；②也可以调用localStorage、cookies等本地存储方式。localStorage另一个浏览上下文（另一个标签页）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。

21、如何在页面上实现一个圆形的可点击区域？

①map+area或者svg ②border-radius ③纯js实现 需要求一个点不在圆上简单算法、获取鼠标坐标等等

22、CSS3有哪些新特性？

① CSS3实现圆角 (border-radius:8px)，阴影 (box-shadow:10px)；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 [CSS3新增属性](#)

23、img的alt和title的异同？

alt 是图片加载失败时，显示在网页上的替代文字；**title** 是鼠标放上面时显示的文字，title是对图片的描述与进一步说明；

这些都是表面上的区别，alt是img必要的属性，而title不是。

对于网站seo优化来说，title与alt还有最重要的一点：搜索引擎对图片意思的判断，主要靠alt属性。所以在图片alt属性中以简要文字说明，同时包含关键词，也是页面优化的一部分。条件允许的话，可以在title属性里，进一步对图片说明。

24、简述一下src与href的区别

href 是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，用于超链接。

src是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。

25、html5新元素？

HTML5 中一些有趣的新特性：①用于绘画的 canvas 元素；②用于媒介回放的 video 和 audio 元素；③对本地离线存储的更好的支持；④新的特殊内容元素，比如 article、footer、header、nav、section；⑤新的表单控件，比如 calendar、date、time、email、url、search"

以上是w3c上的原话，简单说就是更符合标准，提供更多功能支持；更加规范，可读性更强，性能有提升，实现功能更加简单方便，就像手机一样，高版本的功能多一些。更详细请看[html5新元素](#)

26、CSS层叠是什么？介绍一下

CSS就是层叠式样式表（Cascading Style Sheets）的简称,CSS中层叠的意思就是在HTML文档树结构中的子标记能够继承所有父标记定义的样式，还可以多次定义自己的样式，全部样式按照从外到内、由先到后的顺序叠加起来，如果不发生冲突，则全部样式都有效，重复定义冲突时按照内层优先、后定义优先的原则进行覆盖，即内层子元素覆盖父元素样式、后定义的覆盖先定义的样式。

27、CSS实现垂直和水平居中

这是一道经典的问题，实现方法也有很多种，以下是其中一种实现：具体方案说明：①使用绝对定位left:50%与margin-left取宽度值一半的复数形式设置水平居中。②使用绝对定位top:50%与margin-top取高度值一半的复数形式设置垂直居中 HTML结构：

```
1 <div class="box1">
2     <div class="content"></div>
3 </div>
```

CSS代码：

```
1 .box1 {
2     position: relative;
3     width: 600px;
4     height: 600px;
5     background-color: red;
6 }
7 .content{
8     background-color:pink;
9     width:200px;
```

```
10     height:200px;
11     position: absolute;
12     //父元素相对定位,使子元素相对父元素定位
13     top: 50%;
14     left: 50%;
15     margin-top:-100px ;
16     //二分之一的height, width
17     margin-left: -100px;
18 }
```

28、什么是CSS Hack?

针对不同的浏览器写不同的CSS,就是 CSS Hack。详情请看以下链接 [CSS hack大全&详解 \(什么是CSS hack\)](#)

29、你知道哪些CSS浏览器兼容性问题。

[CSS 多浏览器兼容性问题及解决方案](#)

30、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,是绝对单位,类似的还有pt (磅, 1pt=1/72英寸)、pc (皮卡, 1pc=12pt)、mm (毫米)、cm (厘米)、in (英寸) 绝对单位指定是多少就是多少,计算比较容易。

em的值不是固定的,它是相对单位,em是指当前默认字号大小(继承父元素默认字号)的倍数,可根据父元素字号的改变而自动调整。例如2em是当前字号的2倍,若父元素或默认字号为12pt,则2em就是24pt。类似的还有ex,ex是当前字号高度值(通常是字体尺寸一半)的倍数。

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

31、HTML5离线存储

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。

32、说说你对语义化的理解

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

33、描述一段语义的HTML代码

```
1 (HTML5中新增加的很多标签(如: <article>、<nav>、<header>和<footer>等)就是基于语义化设计原则)
2
3 < div id="header"> < h1>标题< /h1>
4     < h2>专注Web前端技术< /h2>
5 < /div>
```

34、如何居中div?

给div设置一个宽度,然后添加margin:0 auto属性 div{width:200px; margin:0 auto; }

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

① **block** 像块类型元素一样显示。**none** 缺省值。像行内元素类型一样显示。**inline-block** 像行内元素一样显示，但其内容像块类型元素一样显示。**list-item** 像块类型元素一样显示，并添加样式列表标记。② **absolute** absolute 生成绝对定位的元素，相对于 static 定位以外的第一个父元素进行定位，若父元素都没有定位则相对于html的根元素（浏览器窗口）定位。**fixed** fixed（老IE不支持）生成绝对定位的元素，相对于浏览器窗口进行定位。**relative** relative生成相对定位的元素，相对于其原来所在的文档流中的位置进行定位。**static** 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）

36、对BFC规范的理解

BFC，块级格式化上下文，一个创建了新的BFC的盒子是独立布局的，盒子里面的子元素的样式不会影响到外面的元素。在同一个BFC中的两个毗邻的块级盒在垂直方向（和布局方向有关系）的margin会发生折叠。

（W3C CSS 2.1 规范中的一个概念,它决定了元素如何对其内容进行定位,以及与其他元素的关系和相互作用。）

37、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。最简化的CSS Reset

```
1 *{
2   outline:0;
3   padding:0;
4   margin:0;
5   border:0;
6 }
```

38、CSS定义的权重

CSS有自己的优先级计算公式，而不仅仅是行间>内部>外部样式；ID>class>元素。理解选择器的特殊性很重要，特别是在修复bug的时候，但是要尽量避免使用。[CSS选择器的权重与优先规则](#)

39、CSS定位方式有哪些?position属性的值有哪些?他们之间的区别是什么？

在CSS中关于定位的内容是：position:relative | absolute | static | fixed • static 自动定位，自动定位就是元素在页面普通文档流中由HTML自动定位，普通文档里的元素也称为流动元素。不能通过z-index进行层次分级。

- relative 相对定位，相对定位不脱离文档流，参考其在原来文档流中的位置，通过 top,bottom,left,right 定位，并且可以通过z-index进行层次分级。
- absolute 绝对定位，绝对定位脱离文档流，依据最近的已经定位（绝对、相对或固定定位）的父元素，通过 top,bottom,left,right 定位。当父级 position 为 static 时，absolute元素将依据body根元素（浏览器窗口）进行定位，可以通过z-index进行层次分级。
- fixed 固定定位，固定定位与父元素无关（无论父元素是否定位），直接根据浏览器窗口定位，且不随滚动条拖动页面而滚动，可通过z-index进行层次分级。

40、CSS中margin和padding的区别

margin是外边距，属于元素之外，相邻元素的margin可以融合。padding是内边距，在元素之内，相邻元素的padding不可融合。

41、使用CSS预处理器吗？喜欢哪个？

[什么是css预处理器？现在阶段我们用不用,怎样用？](#)

42、Html5与html4相比，各有何优缺点？ 怎样处理html5新标签的兼容性问题？

html5余html4的异同请看以下的链接 [html5与html4的异同](#) 兼容性问题 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。

```
1 <!--[if lt IE 9]>
2 <script> src="http://html5shim.googlecode.com/svn/trunk/html5.js"</script>
3 <![endif]-->
```

43、title与h1的区别、b与strong的区别、i与em的区别？

title与h1的区别 定义：title是网站标题，h1是文章主题 作用：title概括网站信息，可以直接告诉搜索引擎和用户这个网站是关于什么主题和内容的，是显示在网页Tab栏里的；h1突出文章主题，面对用户，更突出其视觉效果，指向页面主体信息，是显示在网页中的。

b与strong的区别 定义：b(bold)是实体标签，用来给文字加粗，而strong是逻辑标签，作用是加强字符语气。区别：b标签只是加粗的样式，没有实际含义，常用来表达无强调或着重意味的粗体文字，比如文章摘要中的关键词、评测文章中的产品名称、文章的导言；而strong表示标签内字符重要，用以强调，其默认格式是加粗，但是可以通过CSS添加样式，使用别的样式强调。 **建议**：为了符合CSS3的规范，b应尽量少用而改用strong

i与em的区别 定义：i(italic)是实体标签，用来使字符倾斜，而em(emphasis)是逻辑标签，作用是强调文本内容 **区别**：i标签只是斜体的样式，没有实际含义，常用来表达无强调或着重意味的斜体，比如生物学名、术语、外来语；而em表示标签内字符重要，用以强调，其默认格式是斜体，但是可以通过CSS添加样式。 **建议**：为了符合CSS3的规范，i应尽量少用而改用em

44、实现不使用 border 画出1px高的线，在不同浏览器的标准模式与怪异模式下都能保持一致的效果。

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

45、请解释一下CSS3的Flexbox（弹性盒布局模型），以及适用场景

一个用于页面布局的全新CSS3功能，Flexbox可以把列表放在同一个方向（从上到下排列，从左到右），并让列表能延伸到占用可用的空间。常规布局是基于块和内联流方向，而Flex布局是基于flex-flow流可以很方便的用来做局中，能对不同屏幕大小自适应。在布局上有了比以前更加灵活的空间。 [Flexbox——快速布局神器](#)

46、用纯CSS创建一个三角形的原理是什么？

把上、左、右三条边隐藏掉（颜色设为 transparent）

```
1 demo {  
2   width: 0;  
3   height: 0;  
4   border-width: 20px;  
5   border-style: solid;  
6   border-color: transparent transparent red transparent;  
7 }
```

47、一个满屏品字布局如何设计

简单的方式：上面的div宽100%，下面的两个div分别宽50%，然后用float或者inline使其不换行即可

48、浏览器是怎样解析CSS选择器的？

样式系统从关键选择器开始匹配，然后从右左移查找规则选择器的祖先元素。浏览器从右到左进行查找的好处是为了尽早过滤掉一些无关的样式规则和元素。Firefox 称这种查找方式为 keyselector(关键字查询)，所谓的关键字就是样式规则中最后(最右边)的规则。只要选择器的子树一直在工作，样式系统就会持续左移，直到和规则匹配，或者是因为不匹配而放弃该规则。

49、::before 和 :after中双冒号和单冒号 有什么区别？解释一下这2个伪元素的作用。

单冒号(:)用于CSS3伪类，双冒号(::)用于CSS3伪元素。（伪元素由双冒号和伪元素名称组成）

双冒号是在当前规范中引入的，用于区分伪类和伪元素。不过浏览器需要同时支持旧的已经存在的伪元素写法，比如:first-line、:first-letter、:before、:after等，而新的在CSS3中引入的伪元素则不允许再支持旧的单冒号的写法。想让插入的内容出现在其它内容前，使用::before，否则，使用::after

在代码顺序上，::after生成的内容也比::before生成的内容靠后。如果按堆栈视角，::after生成的内容会在::before生成的内容之上

50、如果需要手动写动画，你认为最小时间间隔是多久，为什么？（阿里）

多数显示器默认频率是60Hz，即1秒刷新60次，所以理论上最小间隔为 $1/60 * 1000\text{ms} = 16.7\text{ms}$

51、li与li之间有看不见的空白间隔是什么原因引起的？有什么解决办法？

行框的排列会受到中间空白（回车\空格）等的影响，因为空格也属于字符,这些空白也会被应用样式，占据空间，所以会有间隔，把字符大小设为0，就没有空格了。

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

无论属于哪种，都要先找到其祖先元素中最近的 position 值不为 static 的元素，然后再判断：1、若此元素为 inline 元素，则 containing block 为能够包含这个元素生成的第一个和最后一个 inline box 的 padding box (除 margin, border 外的区域) 的最小矩形；2、否则,则由这个祖先元素的 padding box 构成。如果都找不到，则为 initial containing block。

补充：① static(默认的)/relative：简单说就是它的父元素的内容框（即去掉padding的部分）② absolute：向上找最近的定位为absolute/relative的元素③ fixed：它的containing block一律为根元素(html/body)，根元素也是initial containing block

53、CSS里的visibility属性有个collapse属性值是干嘛用的？在不同浏览器下以后有什么区别？

当在表格元素中使用时，此值可删除一行或一列，但是它不会影响表格的布局。被行或列占据的空间会留给其他内容使用。如果此值被用在其他的元素上，会呈现为“hidden”。

在谷歌浏览器里，使用collapse值和使用hidden值没有什么区别。

在火狐浏览器、Opera和IE11里，使用collapse值的效果就如它的字面意思：table的行会消失，它的下面一行会补充它的位置。

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

1、如果元素的display: none，那么position，float不起作用。2、如果元素拥有position: absolute或position: fixed属性，那么float不起作用。3、如果元素的float属性值不是none，元素会脱离文档流，根据float属性值来显示。4、设置了float、position: absolute、display: inline-block属性的元素，margin不会和垂直方向上的其他元素的margin发生外边距合并。5、设置了overflow: hidden属性的元素，不和它的子元素发生margin合并。[position跟display、margin、collapse、overflow、float这些特性相互叠加之后会怎么样？](#)

54、移动端的布局用过媒体查询吗？

通过媒体查询可以为不同大小和尺寸的媒体定义不同的css，适合相应的设备显示；即响应式布局

@media screen and (min-width: 400px) and (max-width: 700px) { ... } 上面这段CSS代码意思是：当页面小于700px大于400px的时候执行它下面括号中的CSS.这个应该没有太大疑问。

CSS优化，提高性能的方法有哪些？

- 1，首推的是合并css文件，如果页面加载10个css文件，每个文件1k，那么也要比只加载一个100k的css文件慢。
- 2，减少css嵌套，最好不要套三层以上。
- 3，不要在ID选择器前面进行嵌套，ID本来就是唯一的而且人家权值那么大，嵌套完全是浪费性能。
- 4，建立公共样式类，把相同样式提取出来作为公共类使用，比如我们常用的清除浮动等。
- 5，减少通配符*或者类似[hidden="true"]这类选择器的使用，挨个查找所有...这性能能好吗？当然重置样式这些必须的东西是不能少的。
- 7，巧妙运用css的继承机制，如果父节点定义了，子节点就无需定义。
- 8，拆分出公共css文件，对于比较大的项目我们可以将大部分页面的公共结构的样式提取出来放到单独css文件里，这样一次下载后就放到缓存里，当然这种做法会增加请求，具体做法应以实际情况而定。
- 9，不用css表达式，表达式只是让你的代码显得更加炫酷，但是他对性能的浪费可能是超乎你的想象的。
- 10，少用css rest，可能你会觉得重置样式是规范，但是其实其中有很多的操作是不必要不友好的，有需求有兴趣的朋友可以选择normalize.css
- 11，cssSprite，合成所有icon图片，用宽高加上background-position的背景图方式显现出我们要的icon图，这是一种十分实用的技巧，极大减少了http请求。

当然我们还需要一些善后工作，CSS压缩(这里提供一个在线压缩 YUI Compressor，当然你会用其他工具来压缩是十分好的)，GZIP压缩，Gzip是一种流行的文件压缩算法，详细做法可以谷歌或者百度。详细文章：[CSS提高性能的方法](#)

55、 页面可见性 (Page Visibility) API 可以有哪些用途？

当可见性发生了改变的时候，会触发 visibilityState事件，通过给事件注册一个监听函数，那么就可以进行一些操作：即能够在页面切换到不可见状态时暂停执行一些不必要的操作，以减少资源的浪费。

比如当前页面在播放动画，检测到用户切换了或者最小化了该窗口，则自动停止；有的程序会及时刷新访问界面（比如篮球比赛时的比分更新），若窗口关闭，他也会停止刷新；

56、 元素竖向的百分比设定是相对于容器的高度吗？

对于一些表示竖向距离的属性，例如padding-top,padding-bottom,margin-top,margin-bottom等，当按百分比设定它们时，依据的是父容器的宽度，而不是高度,请自行验证。

57、 全屏滚动的原理是什么？用到了CSS的哪些属性？

图片轮播原理，只不过图片宽高100%、超出隐藏、调整比例适应屏幕大小。

58、 什么是响应式设计？响应式设计的基本原理是什么？如何兼容低版本的IE？

响应式网站设计(Responsive Web design)的理念是：集中创建页面的图片排版大小，可以智能地根据用户行为以及使用的设备环境（系统平台、屏幕尺寸、屏幕定向等）进行相对应的布局，无论用户正在使用笔记本还是iPad，我们的页面都应该能够自动切换分辨率、图片尺寸及相关脚本功能等，以适应不同设备。

响应式Web设计不仅仅是关于屏幕分辨率自适应以及自动缩放图片等等，它更像是一种对于设计的全新思维模式，如我们需要兼容不同屏幕分辨率、清晰度以及屏幕定向方式竖屏(portrait)、横屏(landscape)，怎样才能做到让一种设计方案满足所有情况？

基本原理: 媒体查询 @media

兼容IE可以使用JS辅助一下来解决

59、 网页验证码是干嘛的，是为了解决什么安全问题？

区分用户是计算机还是人的公共全自动程序。可以防止：恶意破解密码、刷票、论坛灌水；有效防止黑客对某一个特定注册用户用特定程序暴力破解方式进行不断的登陆尝试；

60、 视差滚动效果，如何给每页做不同的动画？

[视差滚动效果原理解析与功能实现](#)

61、 如何修改chrome记住密码后自动填充表单的黄色背景？

可以通过input::-webkit-autofill来进行修改！

```
1 input:-webkit-autofill {  
2   background-color: #FAFFBD;  
3   background-image: none;  
4   color: #000;  
5 }
```

62、你对line-height是怎样理解的？

css中起高度作用的是height以及line-height了吧！line-height是行高，height就是高，通常height是对于某个框架或者图片来说，line-height用于文字，如果一个标签没有定义height属性(包括百分比高度)，那么其最终表现的高度一定是由line-height起作用，利用行高还可以实现单行或多行或图片垂直居中实现上的应用。[CSS行高的一些深入理解和应用](#)

63、设置元素浮动后，该元素的display值是多少？

无论行内元素还是块元素，被设置浮动之后，display属性值都变为block。

64、怎么让Chrome支持小于12px 的文字？

1、用图片：如果是内容固定不变情况下，使用将小于12px文字内容切出做图片，这样不影响兼容也不影响美观。
2、使用12px及12px以上字体大小：为了兼容各大主流浏览器，建议设计美工图时候设置大于或等于12px的字体大小，如果是接单的这个时候就需要给客户讲解小于12px浏览器不兼容等事宜。
3、继续使用小于12px字体大小样式设置：如果不考虑chrome可以不用考虑兼容，同时在设置小于12px对象设置-webkit-text-size-adjust:none，做到最大兼容考虑。
4、可以使用到css3里的一个属性：transform:scale() [CSS3 transform 属性介绍](#)

```
1 <style>  
2   p span{font-size:10px;-webkit-transform:scale(0.8);display:block;}  
3 </style>  
4 <p><span>测试10px</span></p>
```

65、让页面的里的字体变清晰，变细用CSS怎么做？

CSS3里面加入了一个“-webkit-font-smoothing”属性。这个属性可以使页面上的字体抗锯齿,使用后字体看起来会更清晰舒服。`webkit-font-smoothing:antialiased`

66、font-style属性可以给它赋值为oblique,oblique是什么意思？

可以理解为Oblique是让没有斜体属性的文字倾斜。italic和oblique都是向右倾斜的文字,但区别在于Italic是指斜体字，而Oblique是倾斜的文字，对于没有斜体的字体应该使用Oblique属性值来实现倾斜的文字效果。

67、position:fixed在安卓下无效怎么处理？

在head头中加入下面的代码 `<meta name="viewport" content="width=device-width, initial-scale=1.0, maximum-scale=1.0, minimum-scale=1.0, user-scalable=no"/>`

68、display:inline-block什么时候会显示间隙？（携程）

移除空格、使用margin负值、使用font-size:0、letter-spacing、word-spacing

69、有一个高度自适应的div，里面有两个div，一个的高度是100px，希望另一个填满剩下的高度。

box-sizing方案 和 absolute position方案 [有一个高度自适应的div，里面有两个div，一个高度100px，希望另一个填满剩下的高度。](#)

70、png/jpg/gif这些图片格式解释一下，分别什么时候用，有没有了解过webp?

PNG：这是三种中质量最好的一种，保存图像是静态图，可以保留32位色，也能保留透明与半透明区域，但是很多网站不支持PNG上传 **JPG**：这是目前使用最广泛的格式之一，因为其高质量的压缩率导致的图片大小减少，而且也支持32位色彩，因此被广泛使用，一般情况下，任何支持图片的地方都支持jpg，也为静态图，但缺点是不支持透明区域 **GIF**：这也是目前使用最广泛的格式之一，和jpg相比，有以下不同：颜色数只支持256色，支持透明区域，不支持半透明区域，可以保存为动态图

webp是谷歌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。Facebook Ebay等知名网站已经开始测试并使用WebP格式。但WebP是一种有损压缩。相较编码JPEG文件，编码同样质量的WebP文件需要占用更多的计算资源。

71、style标签写在body后与body前有什么区别？

写在head标签中利于浏览器逐步渲染（resources downloading->CSSOM+DOM->RenderTree(composite)->Layout->paint）写在body标签后由于浏览器以逐行方式对html文档进行解析，当解析到写在尾部的样式表（外联或写在style标签）会导致浏览器停止之前的渲染，等待加载且解析样式表完成之后重新渲染，在windows的IE下可能会出现FOUC现象（即样式失效导致的页面闪烁问题）

72、简述一下src与href的区别

href 是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，用于超链接。

src是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。

73、简述同步和异步的区别

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。

74、px和em的区别

相同点：px和em都是长度单位；

异同点：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。

75、浏览器的内核分别是什么？

IE: trident内核 Firefox: gecko内核 Safari: webkit内核 Opera: 以前是presto内核, Opera现已改用Google Chrome的Blink内核 Chrome: Blink(基于webkit, Google与Opera Software共同开发)

76、什么叫优雅降级和渐进增强？

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

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

区别:

- 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

77、sessionStorage、localStorage 和 cookie 之间的区别

不同点:

- (1)、存储内容是否发送到服务器端: 当设置了Cookie后, 数据会发送到服务器端, 造成一定的宽带浪费;

1 | web storage, 会将数据保存到本地, 不会造成宽带浪费;

- (2)、数据存储大小不同: Cookie数据不能超过4K, 适用于会话标识; web storage数据存储可以达到5M;

- (3)、数据存储的有效期限不同: cookie只在设置了Cookie过期时间之前一直有效, 即使关闭窗口或者浏览器;

1 | sessionStorage, 仅在关闭浏览器之前有效; localStorage, 数据存储永久有效;

- (4)、作用域不同: cookie和localStorage是在同源同窗口中都是共享的; sessionStorage不在不同的浏览器窗口中共享, 即使是同一个页面;

cookie和session的区别: 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、建议将登录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 6、session保存在服务器, 客户端不知道其中的信心; cookie保存在客户端, 服务器能够知道其中的信息 7、session中保存的是对象, cookie中保存的是字符串 8、session不能区分路径, 同一个用户在访问一个网站期间, 所有的session在任何一个地方都可以访问到, 而cookie中如果设置了路径参数, 那么同一个网站中不同路径下的cookie互相是访问不到的

*web Storage和Cookie的区别 * Web Storage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, 不可跨域调用。除此之外, web storage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。

78、Ajax的优缺点及工作原理？

AJAX = Asynchronous JavaScript and XML (异步的 JavaScript 和 XML)

优点:

- 1.减轻服务器的负担,按需取数据,最大程度的减少冗余请求
- 2.局部刷新页面,减少用户心理和实际的等待时间,带来更好的用户体验
- 3.基于xml标准化,并被广泛支持,不需安装插件等,进一步促进页面和数据的分离 缺点:

- 1.AJAX大量的使用了javascript和ajax引擎,这些取决于浏览器的支持.在编写的时候考虑对浏览器的兼容性.
- 2.AJAX只是局部刷新,所以页面的后退按钮是没有用的.
- 3.对流媒体还有移动设备的支持不是太好等

AJAX的工作原理:

- 1.创建ajax对象 (XMLHttpRequest/ActiveXObject(Microsoft.XMLHttp))
- 2.判断数据传输方式(GET/POST)
- 3.打开链接 open()
- 4.发送 send()
- 5.当ajax对象完成第四步 (onreadystatechange) 数据接收完成, 判断http响应状态 (status) 200-300之间或者 304 (缓存) 执行回调函数

79、开发及性能优化

80、规避javascript多人开发函数重名问题

- 命名空间
- 封闭空间
- js模块化mvc (数据层、表现层、控制层)
- seajs
- 变量转换成对象的属性
- 对象化

81、请说出三种减低页面加载时间的方法

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

82、你所了解到的Web攻击技术

(1) XSS (Cross-Site Scripting, 跨站脚本攻击): 指通过存在安全漏洞的Web网站注册用户的浏览器内运行非法的HTML标签或者JavaScript进行的一种攻击。(2) SQL注入攻击 (3) CSRF (Cross-Site Request Forgeries, 跨站点请求伪造): 指攻击者通过设置好的陷阱, 强制对已完成的认证用户进行非预期的个人信息或设定信息等某些状态更新。

83、web前端开发, 如何提高页面性能优化?

内容方面： 1.减少 HTTP 请求 (Make Fewer HTTP Requests)

2.减少 DOM 元素数量 (Reduce the Number of DOM Elements)

3.使得 Ajax 可缓存 (Make Ajax Cacheable) 针对CSS： 1.把 CSS 放到代码页上端 (Put Stylesheets at the Top)

2.从页面中剥离 JavaScript 与 CSS (Make JavaScript and CSS External)

3.精简 JavaScript 与 CSS (Minify JavaScript and CSS)

4.避免 CSS 表达式 (Avoid CSS Expressions)

针对JavaScript：

1. 脚本放到 HTML 代码页底部 (Put Scripts at the Bottom)

2. 从页面中剥离 JavaScript 与 CSS (Make JavaScript and CSS External)

3. 精简 JavaScript 与 CSS (Minify JavaScript and CSS)

4. 移除重复脚本 (Remove Duplicate Scripts) 面向图片(Image)： 1.优化图片

2 不要在 HTML 中使用缩放图片

3 使用恰当的图片格式

4 使用 CSS Sprites 技巧对图片优化

前端开发中，如何优化图像？图像格式的区别？

优化图像： 1、不用图片，尽量用css3代替。比如说要实现修饰效果，如半透明、边框、圆角、阴影、渐变等，在当前主流浏览器中都可以用CSS达成。

2、使用矢量图SVG替代位图。对于绝大多数图案、图标等，矢量图更小，且可缩放而无需生成多套图。现在主流浏览器都支持SVG了，所以可放心使用！

3、使用恰当的图片格式。我们常见的图片格式有JPEG、GIF、PNG。

基本上，内容图片多为照片之类的，适用于JPEG。

而修饰图片通常更适合用无损压缩的PNG。

GIF基本上除了GIF动画外不要使用。且动画的话，也更建议用video元素和视频格式，或用SVG动画取代。 4、按照HTTP协议设置合理的缓存。

5、使用字体图标webfont、CSS Sprites等。

6、用CSS或JavaScript实现预加载。

7、WebP图片格式能给前端带来的优化。WebP支持无损、有损压缩，动态、静态图片，压缩比率优于GIF、JPEG、JPEG2000、PG等格式，非常适合用于网络等图片传输。

84、浏览器是如何渲染页面

渲染的流程如下：

1.解析HTML文件，创建DOM树。

自上而下，遇到任何样式 (link、style) 与脚本 (script) 都会阻塞（外部样式不阻塞后续外部脚本的加载）。

2.解析CSS。优先级：浏览器默认设置<用户设置<外部样式<内联样式<HTML中的style样式；

3.将CSS与DOM合并，构建渲染树 (Render Tree)

4.布局 and 绘制, 重绘 (repaint) 和重排 (reflow) 的?

Javascript

1、JS如何使页面跳转?怎么引入一个外部JS文件?

①直接在head标签内写入js代码, 如下

```
1 | <Script Language="JavaScript">js 语句</Script>
```

②引入写好的js文件, 使用语句

```
1 | <script language="JavaScript" src="test.js"></script>
```

也是直接放入到head标签里头, 也有的是放在前面。

2、输入框的验证用什么事件?

change(fn)

3、undefined与null有何异同?

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。

undefined: (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:** (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。

它们都表示空, 转换为boolean后都为false, 但是null代表一个对象变量已经被初始化, 但未装入对象;

undefined表示未初始化变量

4、===与==有何异同?

相同点: 都是判定两个值是否相等 不同点: ==不会判断类型, 而===会判断类型

5、如何判断一个变量的值是否为数字? 以及有哪些手段判断变量值的数据类型?

全局函数isNaN可以判断一个变量的值是否为数字。可以使用运算符type、instanceof判断变量值的数据类型。

6、什么是Bom什么是Dom?你如何理解Dom?

[链接标记target与Dom, Bom](#)

7、Array的join、push、splice、slice各有何用途, splice与slice有何异同?

join: 使用指定间隔符连接所有元素为字符串 push: 在尾部添加元素并维护array实例的length splice与slice都是截取一部分元素。不同的在于: slice返回截取后的新实例, splice在原array实例上操作, 更详细的请见下文链接。[JS中数组对象详解](#)

8、如何阻止表单提交？

在onsubmit事件中返回false

9、如何动态操作表格？

可以像普通dom一样操作，但是因为表格的dom比较复杂，所以我通常是使用table的insertRow、deleteRow及tr对象的insetCell、deleteCell操作。

10、String.match与RegExp.exec有何区别？

match只会返回没有分组的全部匹配结果或者有分组的第一次匹配结果；而exec可以利用循环返回全部匹配结果。

11、为验证手机号写一个正则。

```
1 function checkSubmitMobil()
2 {
3   if ($("#phoneNum").val() == "") {
4     alert("手机号码不能为空!");
5     //$("#moileMsg").html("<font color='red'>手机号码不能为空! </font>");
6     $("#mobile").focus();
7     return false;
8   }
9   if (!$("#phoneNum").val().match(/^(\?:13\d|15\d|18\d)\d{5}(\d{3}|\*{3})$/))
10  {
11    alert("手机号码格式不正确!");
12    //$("#moileMsg").html("<font color='red'>手机号码格式不正确! 请重新输入! </font>");
13    $("#phoneNum").focus();
14    return false;
15  }
16  return true;
17 }
```

12、正则的i标记与g标记各有何用途？

i: 不区分大小写; g: 全局匹配。

13、为String添加trim()方法。

```
1 String.prototype.trim = function() {
2   return this.replace(/^ +| +$/g, "");
3 }
```

14、简述COOKIE。在JS中如何操作Cookie？

[简述cookie, 在JS中如何操作cookie](#)

15、谈谈javascript数组排序方法sort()的使用，重点介绍sort()参数的使用及其内部机制。

[JS数组排序方法sort\(\)的使用](#)

16、谈谈innerHTML outerHTML innerText之间的区别。

①innerHTML是w3c的html dom定义的方法，而后两者是IE独有的方法； ②innerHTML代表一个元素节点内由所有子节点，不包括当前节点组成的html代码； ③outerHTML代表一个元素节点内由所有子节点和当前节点组成的html代码； ④innerText代表一个元素节点内由所有子文本节点内容组成的文本；

17、在JavaScript中定时调用函数 foo() 如何写？

```
1 setTimeout(  
2     foo,  
3     1000    //这里设置延时数  
4 );
```

18、setTimeout与setInterval有何区别？

①setTimeout和setInterval的语法相同。它们都有两个参数，一个是将要执行的代码字符串，还有一个是以毫秒为单位的时间间隔，当过了那个时间段之后就将执行那段代码。 ②不过这两个函数还是有区别的，setInterval在执行完一次代码之后，经过了那个固定的时间间隔，它还会自动重复执行代码，而setTimeout只执行一次那段代码。

19、你在js中用过array吗？如果用过，array中添加数据用什么方法？

在尾部添加使用push(); 在头部添加使用unshift(); 在任意位置添加使用splice(),但要注意把它的删除个数设置为0; array详细介绍请看下文链接 [JS中数组对象详解](#)

20、简述javascript的优缺点。

优点：简单易用，与java有类似的语法，可以使用任何文本编辑工具编写，只需要浏览器就可执行程序，并且事先不用编译，逐行执行，无需进行严格的变量声明，而且内置大量现成对象，编写少量程序可以完成目标； **缺点：**不适合开发大型应用程序；

21、Javascript有哪些内置对象？

只有Math和Global（在浏览器环境中，Global就是Window）

22、列举Javascript的本地对象。

Object、Function、Array、String、Boolean、Number、Date、RegExp、Error、EvalError、RangeError、ReferenceError、SyntaxError、TypeError、URIError

23、javascript的typeof返回哪些数据类型

object number function boolean undefind string

24、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 (== - ===)

25、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 ④this指向问题

26、事件绑定和普通事件有什么区别

①事件绑定就是针对dom元素的事件，绑定在dom元素上 ②普通事件即为非针对dom元素的事件

27、事件委托是什么

利用事件冒泡的原理，让自己的所触发的事件，由他的父元素代替执行！通俗的讲，事件就是onclick，onmouseover，onmouseout，等就是事件，委托呢，就是让别人来做，这个事件本来是加在某些元素上的，然而你却加到别人身上来做，完成这个事件。例子请看以下链接 [JS中的事件委托](#)

28、闭包是什么，有什么特性，对页面有什么影响

什么是闭包 “官方”的解释：所谓“闭包”，指的是一个拥有许多变量和绑定了这些变量的环境的表达式（通常是一个函数），因而这些变量也是该表达式的一部分。在 Scala、Scheme、Common Lisp、Smalltalk、Groovy、JavaScript、Ruby、Python、Go、Lua、objective c、swift 以及Java (Java8及以上) 等语言中都能找到对闭包不同程度的支持。通俗的讲就是函数a的内部函数b，被函数a外部的一个变量引用的时候，就创建了一个闭包。**闭包的特性**：①.封闭性：外界无法访问闭包内部的数据，如果在闭包内声明变量，外界是无法访问的，除非闭包主动向外提供访问接口；②.持久性：一般的函数，调用完毕之后，系统自动注销函数，而对于闭包来说，在外部函数被调用之后，闭包结构依然保存在；**对页面的影响** 使用闭包会占有内存资源，过多的使用闭包会导致内存溢出等。详细请看以下推荐链接 [深入理解JavaScript的闭包特性 如何给循环中的对象添加事件](#)

29、javascript的本地对象，内置对象和宿主对象

①本地对象为array obj regexp等可以new实例化 ②内置对象为global Math 等不可以实例化的 ③宿主为浏览器自带的document>window 等

30、编写一个数组去重的方法

思路：1.创建一个新的数组存放结果 2.创建一个空对象 3.for循环时，每次取出一个元素与对象进行对比，如果这个元素不重复，则把它存放到结果数组中，同时把这个元素的内容作为对象的一个属性，并赋值为1，存入到第2步建立的对象中。说明：至于如何对比，就是每次从原数组中取出一个元素，然后到对象中去访问这个属性，如果能访问到值，则说明重复。代码如下：

```

1 Array.prototype.unique3 = function(){
2     var res = [];
3     var json = {};
4     for(var i = 0; i < this.length; i++){
5         if(!json[this[i]]){
6             res.push(this[i]);
7             json[this[i]] = 1;
8         }
9     }
10    return res;
11 }
12 var arr = [112,112,34,'你好',112,112,34,'你好','str','str1'];
13 alert(arr.unique3());

```

31、this对象的理解

①this总是指向函数的直接调用者（而非间接调用者）；②如果有new关键字，this指向new出来的那个对象；③在事件中，this指向触发这个事件的对象，特殊的是，IE中的attachEvent中的this总是指向全局对象Window；

32、eval是做什么的？

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）。③由JSON字符串转换为JSON对象的时候可以用eval，`var obj = eval('(' + str + ')');`

33、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。

34、call() 和 apply() 的区别和作用？

①apply()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。如：`function.apply(this,[1,2,3])`；②call()的第一个参数是上下文，后续是实例传入的参数序列。如：`function.call(this,1,2,3)`；

35、请解释一下 JavaScript 的同源策略

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Netscape Navigator2.0，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样本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。

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

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；

37、JS的基本数据类型

number,string,boolean,object,undefined

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

```
1 function(obj)
2 {
3   return typeof(obj) == "string";
4 }
```

39、JavaScript的DOM是什么意思？

DOM是W3C的对象模型，DOM是中立于平台和语言的接口，它允许程序和脚本动态的访问和更新文档的内容结构和样式。

40、如何显示/隐藏一个DOM元素

更改元素的css style，设为display: none。此外还可以将visibility设为hidden，透明度设为0，或长、宽设为0。

41、JavaScript的节点是什么意思？

根据W3C的标准，HTML中文档的所有内容都是节点，整个文档是一个文档节点，每个html元素都是元素节点，Html元素中的文本是文本节点，每个html属性都是属性节点，注释是注释节点。

42、javascript对象的几种创建方式

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

43、javascript继承的6种方法

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 [JavaScript继承方式详解](#)

44、NaN 是什么鬼？typeof 的结果是？如果一个变量的值是 NaN，怎么确定？

NaN 是 'not a number' 的缩写，表示 "不是一个数字"，通常会在运算过程中产生：`console.log('abc' / 4);`
`console.log(4 * 'a');`

虽然它 "不是一个数字"，但是 NaN 的 typeof 结果却是 number：`console.log(typeof (4 * 'a'));` // number

NaN 和任何变量都不相等，包括 NaN 自己：`console.log(NaN === NaN);` // false

判断一个变量是不是 NaN 可以用 isNaN() 函数，但是这并不是一个完美的函数，有些时候用 `value !== value` 似乎更准确，幸运的是，ES6 已经有 `Number.isNaN()` 方法，将比 isNaN() 准确的多。

45、怎样添加、移除、移动、复制、创建和查找节点？

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

46、documentload和documentready的区别

页面加载完成有两种事件 ①load是当页面所有资源全部加载完成后（包括DOM文档树，css文件，js文件，图片资源等），执行的一个函数

问题：如果图片资源较多，加载时间较长，onload后等待执行的函数需要等待较长时间，所以一些效果可能受到影响

②\$(document).ready()是当DOM文档树加载完成后执行一个函数（不包含图片，css等）所以会比load较快执行。

在原生的js中不包括ready()这个方法，只有load方法就是onload事件

47、事件绑定的几种方法？

以button的Click事件为例：

```
1 <button id="btn">click me</button>
2 function clickBtn() {
3   alert('click!');
4 }
```

1、直接在元素上绑定回调函数 click me 2、JS获取DOM元素对象后，对onclick属性赋值，绑定事件：

document.getElementById('btn').onclick=clickBtn; 3、JS获取DOM对象后，调用对象的addEventListener函数绑定事件：document.getElementById('btn').addEventListener('click',clickBtn);

48、事件冒泡？

[JavaScript事件冒泡简介及应用](#)

49、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。 [js将伪数组转换为标准数组的多种方法](#)

50、常用JS框架都有什么？

[前端js框架汇总](#)

51、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。

52、 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 [JavaScript中数组对象详解](#)

53、 为什么要用IIFE

简单来说就是为了能模块化，创建私有变量等等，很多类库（比如 jQuery）都用了这样的写法。 [详解javascript立即执行函数表达式（IIFE）](#)

54、 严格模式下进行 Javascript 开发有啥好处？

参考阮一峰老师的Javascript 严格模式详解 [Javascript 严格模式详解](#)

55、 Node.js的适用场景

高并发、聊天、实时消息推送

56、 描述一下cookies

[web前端面试题第五道—简述Cookie，在JS中如何操作Cookie?](#)

57、 事件是？ IE与火狐的事件机制有什么区别？ 如何阻止冒泡？

①我们在网页中的某个操作（有的操作对应多个事件）。例如：当我们点击一个按钮就会产生一个事件。是可以被JavaScript 侦测到的行为。 ② 事件处理机制：IE是事件冒泡、火狐是 事件捕获； ③ ev.stopPropagation();

58、 如何判断一个对象是否属于某个类

使用instanceof if(a instanceof Person) { alert('yes'); }

59、 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

60、 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

61、 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

62、 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 \(Quicksort\) 的javascript实现](#) “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。

63、统计字符串“aaaabbbccccdddfgh”中字母个数或统计最多字母数。

```
1 varstr = "aaaabbbccccdddfgh";
2 varobj = {};
3 for(vari=0;istr.length;i++)
4 {
5     varv = str.charAt(i);
6     if(obj[v]&obj[v].value == v)
7     {
8         obj[v].count = ++ obj[v].count;    }
9     else{
10         obj[v] = {};
11         obj[v].count = 1;
12         obj[v].value = v;
13     }
14 }
15 for(key inobj)
16 {
17     document.write(obj[key].value +'='+obj[key].count+' ');
18 } // a=4 b=3 c=4 d=2 f=1 g=1 h=1 }
```

64、写一个function，清除字符串前后的空格。（兼容所有浏览器）

```
1 functiontrim(str){
2     if(str&typeof str === "string")
3     {
4         returnstr.replace(/(^s*)|(s*)$/g,"");
5     } //去除前后空白符
6 }
7 }
```

65、如何制作一个combo选项

combo选项就是可以手动输入值，也可以选择下拉列表值的选项。思路：①布局select和input，让input覆盖select，除了select的下拉图标，以方便select选择。②编写js，为select添加onchange事件，onchange时将input的value置成select选中的值。

这个网上有很多成品,大家可以自己试一下，下边给出一个。 [实现一个combo选项](#)

66、介绍js有哪些内置对象？

Object 是 JavaScript 中所有对象的父对象 数据封装类对象: Object、Array、Boolean、Number 和 String 其他对象: Function、Arguments、Math、Date、RegExp、Error

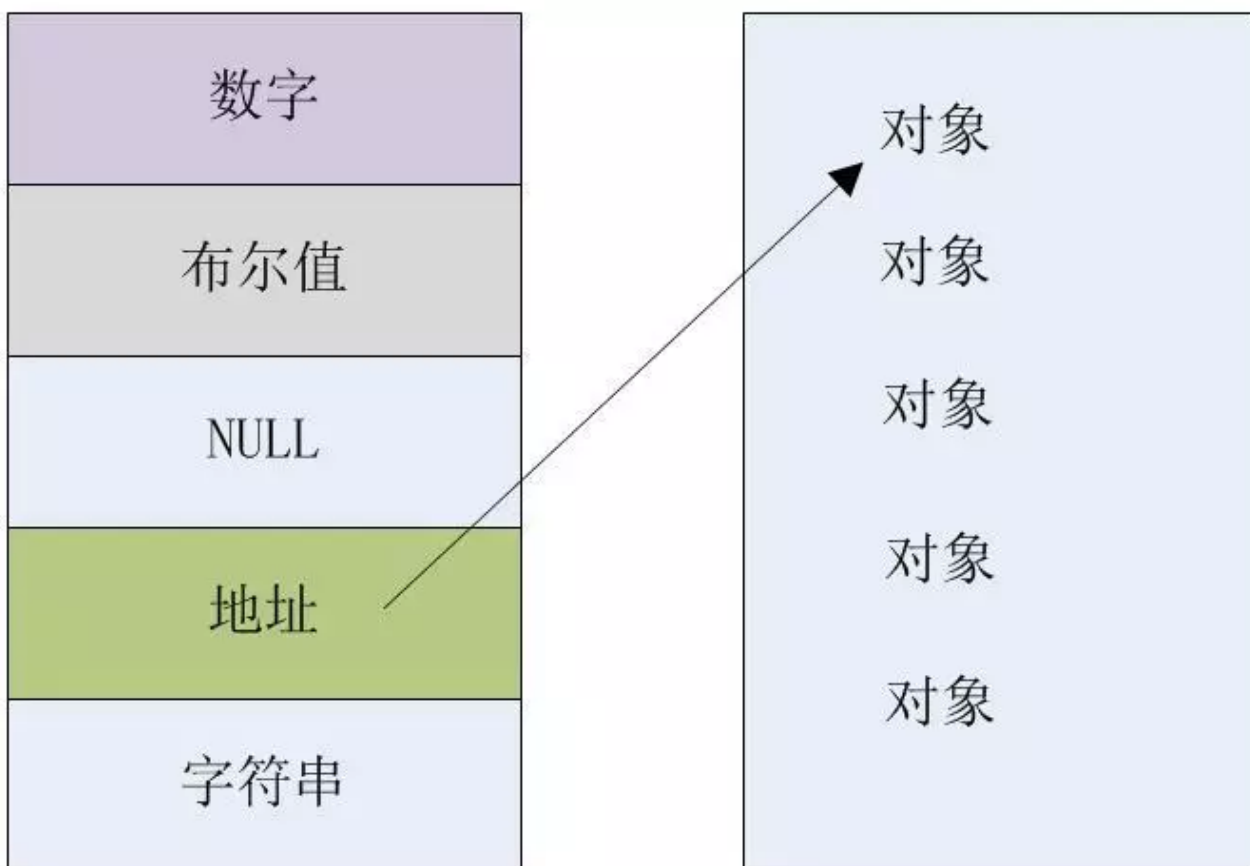
67、说出一些写Javascript的基本规范

1.不要在同一行声明多个变量。 2.请使用 ===/!==来比较true/false或者数值 3.使用对象字面量替代new Array这种形式 4.不要使用全局函数。 5.Switch语句必须带有default分支 6.函数不应该有时候有返回值，有时候没有返回值。 7.For循环必须使用大括号 8.If语句必须使用大括号 9.for-in循环中的变量 应该使用var关键字明确限定作用域，从而避免作用域污染。 10.命名规则中构造器函数首字母大写，如function Person(){ } 11.写注释。

68、 Javascript有几种类型的值？你能画一下他们的内存图吗？

数据类型分为两大类：堆和栈 栈：原始数据类型 (Undefined, Null, Boolean, Number、String) 堆：引用数据类型 (对象、数组和函数)

两种类型的区别是：存储位置不同；原始数据类型直接存储在栈(stack)中的简单数据段，占据空间小、大小固定，属于被频繁使用数据，所以放入栈中存储；引用数据类型存储在堆(heap)中的对象,占据空间大、大小不固定,如果存储在栈中，将会影响程序运行的性能； 引用数据类型在栈中存储了指针，该指针指向堆中该实体的起始地址。当解释器寻找引用值时，会首先检索其在栈中的地址，取得地址后从堆中获得实体。



69、 JavaScript创建对象的几种方式？

1.对象字面量的方式p = { } ； 2.用function来模拟无参的构造函数，再定义属性； 3.用function模拟构造函数，利用this； 4.利用工厂方式（内置对象Object）； 5.利用原型方式来创建； 6.混合方式来创建。

70、JavaScript作用链域

当代码在一个环境中执行时，会创建变量对象的一个作用域链。如果是个函数，则将其活动对象作为变量对象。活动对象在最开始只包含一个arguments对象。而下一个变量对象则来自下一个包含环境。如此一直延续到全局执行环境，这种组织形式即为作用域链。内部函数可访问外部变量，外部变量无法访问内部函数。注意：js没有块级作用域，若要形成块级作用域，可通过（function（）{ }）（）；立即执行的形式实现。[JavaScript 开发进阶：理解JavaScript作用域和作用域链](#)

71、["1","2","3"].map (parseInt) 答案是多少？

[1, NaN, NaN] parseInt() 函数可解析一个字符串，并返回一个整数。需要两个参数 (val, radix)，其中 radix 表示解析时用的基数。map 传了 3 个 (element, index, array)，对应的 radix 不合法导致解析失败。

[\["1","2","3"\].map \(parseInt\)](#)

72、算出字符串中出现次数最多的字符是什么，出现了多少次

```
1 var str = "adadfdfseffserfefsefseffftsdg";
2 var maxLength = 0;
3 var result = "";
4 while(str != ''){
5     oldStr = str;
6     str = str.replace(new RegExp(str.charAt(0),"g"),"");
7     if( oldStr.length-str.length > maxLength){
8         maxLength = oldStr.length-str.length;
9         result = getStr + "=" + maxLength;
10    }
11 }
12 alert(result);
```

73、编写一个方法 求一个字符串的字节长度

```
1 <script language="javascript" type="text/javascript">
2 new function(s) // s指最后括号中的参数
3 {
4     if(!arguments.length||!s)
5     //arguments指最下面括号中的参数集， .length指此集合的参数个数
6     //如果最下面括号中是("你好abc","哈哈"),
7     那么arguments={"你好abc","哈哈"}, .length=2
8     //!s 表示参数是否非空
9     return null;
10    if(""==s)
11    return 0;
12    var l=0;
13    for(var i=0;i<s.length;i++)
14    //中文的值都是大于255的，所以“你好”为4个字节，
15    //可以测试：alert(s.charCodeAt(i));
16    {
17        if(s.charCodeAt(i)>255) l+=2; else l++;
18    }
19    alert(l);
20 }
```

```
21 | ("hello你好, 我好,大家好! world!");
22 | </script>
```

74、JavaScript中有一个函数，执行对象查找的时候，永远不会去查找原型，这个函数是？

Object.hasOwnProperty(propertyName) 其中参数object是必选项，一个对象的实例。propertyName是必选项，一个属性名称的字符串值。hasOwnProperty 函数方法是返回一个布尔值，指出一个对象是否具有指定名称的属性。此方法无法检查该对象的原型链中是否具有该属性；该属性必须是对象本身的一个成员。如果 object 具有指定名称的属性，那么JavaScript中hasOwnProperty函数方法返回 true，反之则返回 false。

75、 页面编码和被请求资源的编码如果不一致怎么处理？

a.html 的编码是gbk或gb2312的。而引入的js编码为utf-8的，那就需要在引入的时候加上 `<script src="http://www.xxx.com/test.js" charset="utf-8"></script>` 同理，如果你的页面是utf-8的，引入的js是gbk的，那么就需要加上charset="gbk"

76、 模块化开发怎么做？

模块化就是将js文件按照功能分离，根据需求引入不同的文件中。源于服务器端。在ES6中提出用工具babel,webpack打包成浏览器识别的js来实现模块化开发，不同的框架和库的模块，有很多功能类似比如Node.js中的模块，Angular.js中的模块，React.js中的模块，只要接口Api可以对的上，那么也可以通用。

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

```
1 | typeof(obj)=== "string"
2 | typeof obj=== "string"
3 | obj.constructor===String
```

78、js去除字符串空格？

方法一：使用replace正则匹配的方法

- 去除所有空格：str=str.replace(/\s*/g, "");
- 去除两头空格: str = str.replace(/^\s|\s\$/g, "");
- 去除左空格： str = str.replace(/\s*/, "");
- 去除右空格： str = str.replace(/(\s*)/g, "");

方法二：使用str.trim()方法 str.trim()局限性：无法去除中间的空格

```
1 | var str=" jay mo";
2 | var str2=str.trim();
3 | console.log(str2);
```

同理，str.trimLeft(),str.trimRight()分别用于去除字符串左右空格。方法三：使用jquery，\$.trim(str)

79、 字符串操作函数

我这里只是列举了常用的字符串函数，具体使用方法，请[参考网址](#)。

- concat() 将两个或多个字符的文本组合起来，返回一个新的字符串。
- indexOf() 返回字符串中第一个子串出现的索引，如果没有匹配项，返回-1。
- charAt() 返回指定位置的字符
- lastIndexOf() 返回字符串中一个子串最后一处出现的索引，如果没有匹配项，返回 -1
- match 检查一个字符串是否匹配一个正则表达式。

```
1 var str="Hello world!"
2 document.write(str.match("world") + "<br />")
3 document.write(str.match("worl'd") + "<br />")
4 //输出
5 //world
6 //null
```

- substr() 函数 返回从string的startPos位置，长度为length的字符串

```
1 var str="Hello world!"
2 document.write(str.substr(3)); //lo world!
3
4 var str="Hello world!"
5 document.write(str.substr(3,7));
6 //lo worl
```

- substring() 返回字符串的一个子串。传入参数是起始位置和结束位置。
- slice() 提取字符串的一部分，并返回一个新字符串。
- replace() - 用来查找匹配一个正则表达式的字符串，然后使用新字符串代替匹配的字符串。
- search() - 执行一个正则表达式匹配查找。如果查找成功，返回字符串中匹配的索引值。否则返回 -1
- split() - 通过将字符串划分成子串，将一个字符串做成一个字符串数组
- toLowerCase() - 将整个字符串转成小写字母
- toUpperCase() - 将整个字符串转成大写字母

80、 、 移除、 移动、 复制、 创建和查找节点？

1) 创建新节点

```
1 createElement() //创建一个元素节点
2 createTextNode() //创建一个文本节点
3 element.appendChild(textNode)
4 appendChild() 方法是向节点添加最后一个子节点//例如：
5 document.getElementById("test").appendChild(node);
6
7 //jquery创建节点：
8 最常用的就是$("#test").append("<div></div>");
9 这样用append就可以创建一个div， 它的父类是id为test的元素
```

2) 添加、 移除、 替换、 插入


```
1 appendChild() //添加 可向节点的子节点列表的末尾添加新的子节点
2 removeChild() //移除
3 replaceChild() //替换
4 insertBefore() //插入
```

3) 查找

```
1 getElementsByTagName() //通过标签名称
2 getElementByName() //通过元素的Name属性的值
3 getElementById() //通过元素Id, 唯一性
```

81、3个使用this的典型应用

1)、在html元素事件属性中使用, 如:

```
1 <input type="button" onclick="showInfo(this);" value="点击一下"/>
```

(2)、构造函数

```
1 function Animal(name, color) {
2     this.name = name;
3     this.color = color;
4 }
```

(3)、input点击, 获取值

```
1 <input type="button" id="text" value="点击一下" />
2 <script type="text/javascript">
3     var btn = document.getElementById("text");
4     btn.onclick = function() {
5         alert(this.value);    //此处的this是按钮元素
6     }
7 </script>
```

(4)、apply()/call()求数组最值

```
1 var numbers = [5, 458, 120, -215];
2 var maxInNumbers = Math.max.apply(this, numbers);
3 console.log(maxInNumbers); // 458
4 var maxInNumbers = Math.max.call(this, 5, 458, 120, -215);
5 console.log(maxInNumbers); // 458
```

82、typeof与instanceof?

相同点: JavaScript 中 typeof 和 instanceof 常用来判断一个变量是否为空, 或者是什么类型的。

typeof的定义和用法: 返回值是一个字符串, 用来说明变量的数据类型。

细节:

- typeof 一般只能返回如下几个结果：number,boolean,string,function,object,undefined。
- typeof 来获取一个变量是否存在，如 if(typeof a!="undefined"){alert("ok")}, 而不要去使用 if(a) 因为如果 a 不存在（未声明）则会出错。
- 对于 Array,Null 等特殊对象使用 typeof 一律返回 object，这正是 typeof 的局限性。

Instanceof定义和用法：instanceof 用于判断一个变量是否属于某个对象的实例。 实例演示：

a instanceof b?alert("true");alert("false"); //a是b的实例？ 真:假

```
1 var a = new Array();
2 alert(a instanceof Array); // true
3 alert(a instanceof Object) // true
```

如上，会返回 true，同时 alert(a instanceof Object) 也会返回 true;这是因为 Array 是 object 的子类。

```
1 function test(){};
2 var a = new test();
3 alert(a instanceof test) // true
```

83、理解闭包？

1、定义和用法：当一个函数的返回值是另外一个函数，而返回的那个函数如果调用了其父函数内部的其它变量，如果返回的这个函数在外部被执行，就产生了闭包。

2、表现形式：使函数外部能够调用函数内部定义的变量。 3、实例如下：

根据作用域链的规则，底层作用域没有声明的变量，会向上一级找，找到就返回，没找到就一直找，直到 window 的变量，没有就返回 undefined。这里明显 count 是函数内部的 flag2 的那个 count。

```
1 var count=10; //全局作用域 标记为flag1
2 function add(){
3     var count=0; //函数全局作用域 标记为flag2
4     return function(){
5         count+=1; //函数的内部作用域
6         alert(count);
7     }
8 }
9 var s = add()
10 s();//输出1
11 s();//输出2
```

4、变量的作用域

要理解闭包，首先必须理解javascript特殊的变量作用域。

变量的作用域分类：全局变量和局部变量。

特点：

- 1、函数内部可以读取函数外部的全局变量；在函数外部无法读取函数内的局部变量。
 - 2、函数内部声明变量的时候，一定要使用var命令。如果不用的话，你实际上声明了一个全局变量！
- 5、使用闭包的注意点

- 1) 滥用闭包，会造成内存泄漏：由于闭包会使得函数中的变量都被保存在内存中，内存消耗很大，所以不能滥用闭包，否则会造成网页的性能问题，在IE中可能导致内存泄露。解决方法是，在退出函数之前，将不使用的局部变量全部删除。
- 2) 会改变父函数内部变量的值。所以，如果你把父函数当作对象（object）使用，把闭包当作它的公用方法（Public Method），把内部变量当作它的私有属性（private value），这时一定要小心，不要随便改变父函数内部变量的值。

84、是跨域？跨域请求资源的方法有哪些？

1、什么是跨域？由于浏览器同源策略，凡是发送请求url的协议、域名、端口三者之间任意一与当前页面地址不同即为跨域。存在跨域的情况：

- 网络协议不同，如http协议访问https协议。
- 端口不同，如80端口访问8080端口。
- 域名不同，如qianduanblog.com访问baidu.com。
- 子域名不同，如abc.qianduanblog.com访问def.qianduanblog.com。
- 域名和域名对应ip,如www.a.com访问20.205.28.90。

2、跨域请求资源的方法：(1)、proxy代理 定义和用法：proxy代理用于将请求发送给后台服务器，通过服务器来发送请求，然后将请求的结果传递给前端。

实现方法：通过nginx代理；

注意点：1、如果你代理的是https协议的请求，那么你的proxy首先需要信任该证书（尤其是自定义证书）或者忽略证书检查，否则你的请求无法成功。

(2)、CORS 【Cross-Origin Resource Sharing】 定义和用法：是现代浏览器支持跨域资源请求的一种最常用的方式。

使用方法：一般需要后端人员在处理请求数据的时候，添加允许跨域的相关操作。如下：

```
1 es.writeHead(200, {
2   "Content-Type": "text/html; charset=UTF-8",
3   "Access-Control-Allow-Origin": 'http://localhost',
4   'Access-Control-Allow-Methods': 'GET, POST, OPTIONS',
5   'Access-Control-Allow-Headers': 'X-Requested-With, Content-Type'
6 });
```

(3)、jsonp

定义和用法：通过动态插入一个script标签。浏览器对script的资源引用没有同源限制，同时资源加载到页面后会立即执行（没有阻塞的情况下）。

特点：通过情况下，通过动态创建script来读取他域的动态资源，获取的数据一般为json格式。实例：

```
1 <script>
2     function testjsonp(data) {
3         console.log(data.name); // 获取返回的结果
4     }
5 </script>
6 <script>
7     var _script = document.createElement('script');
8     _script.type = "text/javascript";
9     _script.src = "http://localhost:8888/jsonp?callback=testjsonp";
10    document.head.appendChild(_script);
11 </script>
```

缺点：1、这种方式无法发送post请求（这里）2、另外要确定jsonp的请求是否失败并不容易，大多数框架的实现都是结合超时时间来判定。

85、垃圾回收机制方式及内存管理

回收机制方式 1、定义和用法：垃圾回收机制(GC:Garbage Collection),执行环境负责管理代码执行过程中使用的内存。

2、原理：垃圾收集器会定期（周期性）找出那些不在继续使用的变量，然后释放其内存。但是这个过程不是实时的，因为其开销比较大，所以垃圾回收器会按照固定的时间间隔周期性的执行。

3、实例如下：

```
1 function fn1() {
2     var obj = {name: 'hanzichi', age: 10};
3 }
4 function fn2() {
5     var obj = {name:'hanzichi', age: 10};
6     return obj;
7 }
8 var a = fn1();
9 var b = fn2();
```

fn1中定义的obj为局部变量，而当调用结束后，出了fn1的环境，那么该块内存会被js引擎中的垃圾回收器自动释放；在fn2被调用的过程中，返回的对象被全局变量b所指向，所以该块内存并不会被释放。

4、垃圾回收策略：标记清除(较为常用)和引用计数。 标记清除：定义和用法：当变量进入环境时，将变量标记"进入环境"，当变量离开环境时，标记为："离开环境"。某一个时刻，垃圾回收器会过滤掉环境中的变量，以及被环境变量引用的变量，剩下的就是被视为准备回收的变量。

到目前为止，IE、Firefox、Opera、Chrome、Safari的js实现使用的都是标记清除的垃圾回收策略或类似的策略，只不过垃圾收集的时间间隔互不相同。

引用计数：定义和用法：引用计数是跟踪记录每个值被引用的次数。基本原理：就是变量的引用次数，被引用一次则加1，当这个引用计数为0时，被视为准备回收的对象。

1、什么时候触发垃圾回收？

垃圾回收器周期性运行，如果分配的内存非常多，那么回收工作也会很艰巨，确定垃圾回收时间间隔就变成了一个值得思考的问题。

IE6的垃圾回收是根据内存分配量运行的，当环境中的变量，对象，字符串达到一定数量时触发垃圾回收。垃圾回收器一直处于工作状态，严重影响浏览器性能。

IE7中，垃圾回收器会根据内存分配量与程序占用内存的比例进行动态调整，开始回收工作。

2、合理的GC方案：(1)、遍历所有可访问的对象;(2)、回收已不可访问的对象。

3、GC缺陷：(1)、停止响应其他操作;

4、GC优化策略：(1)、分代回收 (Generation GC) ;(2)、增量GC

86、过程中遇到的内存泄露情况，如何解决的？

1、定义和用法：

内存泄露是指一块被分配的内存既不能使用，又不能回收，直到浏览器进程结束。C#和Java等语言采用了自动垃圾回收方法管理内存，几乎不会发生内存泄露。我们知道，浏览器中也是采用自动垃圾回收方法管理内存，但由于浏览器垃圾回收方法有bug，会产生内存泄露。

实例如下：

```
1 <div id="myDiv">
2     <input type="button" value="Click me" id="myBtn">
3 </div>
4 <script type="text/javascript">
5     var btn = document.getElementById("myBtn");
6     btn.onclick = function(){
7         document.getElementById("myDiv").innerHTML = "Processing...";
8     }
9 </script>
```

解决方法如下：

```
1 <div id="myDiv">
2     <input type="button" value="Click me" id="myBtn">
3 </div>
4 <script type="text/javascript">
5     var btn = document.getElementById("myBtn");
6     btn.onclick = function(){
7         btn.onclick = null;
8         document.getElementById("myDiv").innerHTML = "Processing...";
9     }
10 </script>
```

(2)、由于是函数内定义函数，并且内部函数--事件回调的引用外暴了，形成了闭包。闭包可以维持函数内局部变量，使其得不到释放。实例如下：

```
1 function bindEvent(){
2     var obj=document.createElement("xxx");
3     obj.onclick=function(){
4         //Even if it's a empty function
5     }
6 }
```

解决方法如下：

```
1 function bindEvent(){
2     var obj=document.createElement("xxx");
3     obj.onclick=function(){
4         //Even if it's a empty function
5     }
6     obj=null;
7 }
```

87、vascript面向对象中继承实现？

面向对象的基本特征有：封闭、继承、多态。在JavaScript中实现继承的方法：

1. 原型链 (prototype chaining)
2. call()/apply()
3. 混合方式(prototype和call()/apply()结合)
4. 对象冒充

继承的方法如下： 1、prototype原型链方式：

```
1 function teacher(name){
2     this.name = name;
3 }
4 teacher.prototype.sayName = function(){
5     console.log("name is "+this.name);
6 }
7 var teacher1 = new teacher("xiaoming");
8 teacher1.sayName();
9
10 function student(name){
11     this.name = name;
12 }
13 student.prototype = new teacher()
14 var student1 = new student("xiaolan");
15 student1.sayName();
16 // name is xiaoming
17 // name is xiaolan
```

2、call()/apply()方法

```
1 function teacher(name,age){
2     this.name = name;
3     this.age = age;
4     this.sayhi = function(){
5         alert('name:'+name+", age:"+age);
6     }
7 }
8 function student(){
9     var args = arguments;
10
11     teacher.call(this,args[0],args[1]);
```



```

12 // teacher.apply(this,arguments);
13 }
14 var teacher1 = new teacher('xiaoming',23);
15 teacher1.sayhi();
16
17 var student1 = new student('xiaolan',12);
18 student1.sayhi();
19
20 // alert: name:xiaoming, age:23
21 // alert: name:xiaolan, age:12

```

88、teacher构造函数（也是函数）在this对象下执行，那么 student就有了teacher的所有属性和方法

3、混合方法【prototype,call/apply】

```

1  function teacher(name,age){
2      this.name = name;
3      this.age = age;
4  }
5  teacher.prototype.sayName = function(){
6      console.log('name: '+this.name);
7  }
8  teacher.prototype.sayAge = function(){
9      console.log('age: '+this.age);
10 }
11
12 function student(){
13     var args = arguments;
14     teacher.call(this,args[0],args[1]);
15 }
16 student.prototype = new teacher();
17
18 var student1 = new student('xiaolin',23);
19 student1.sayName();
20 student1.sayAge();
21 // name:xiaolin
22 // age:23

```

4、对象冒充

```

1  function Person(name,age){
2      this.name = name;
3      this.age = age;
4      this.show = function(){
5          console.log(this.name+" ", "+this.age);
6      }
7  }
8
9  function Student(name,age){
10     this.student = Person; //将Person类的构造函数赋值给this.student

```

```

11 this.student(name,age); //js中实际上是通过对象冒充来实现继承的
12 delete this.student; //移除对Person的引用
13 }
14
15 var s = new Student("小明",17);
16 s.show();
17
18 var p = new Person("小花",18);
19 p.show();
20 // 小明, 17
21 // 小花, 18

```

89、vascript相关程序计算题

```

1 var str = 'asdfssaaasasasasaa';
2 var json = {};
3 for(var i=0;i<str.length;i++){
4     if(!json[str.charAt(i)]){
5         json[str.charAt(i)]=1;
6     }else{
7         json[str.charAt(i)]++;
8     }
9 };
10 var iMax=0;
11 var iIndex='';
12 for(var i in json){
13     if(json[i]>iMax){
14         iMax=json[i];
15         iIndex=i;
16     }
17 }
18 console.log('出现次数最多的是:' + iIndex + '出现' + iMax + '次');

```

90、vaScript 数组(Array)对象

这里只是做了相关的列举，具体的使用方法，[请参考网址](#)。

Array 对象属性

constructor 返回对创建此对象的数组函数的引用。

length 设置或返回数组中元素的数目。

prototype 使您有能力向对象添加属性和方法。Array 对象方法

concat() 连接两个或更多的数组，并返回结果。

join() 把数组的所有元素放入一个字符串。元素通过指定的分隔符进行分隔。

pop() 删除并返回数组的最后一个元素。

shift() 删除并返回数组的第一个元素

push() 向数组的末尾添加一个或更多元素，并返回新的长度。

unshift() 向数组的开头添加一个或更多元素，并返回新的长度。

reverse() 颠倒数组中元素的顺序。

slice() 从某个已有的数组返回选定的元素

sort() 对数组的元素进行排序

splice() 向/从数组中添加/删除项目，然后返回被删除的项目。

toSource() 返回该对象的源代码。

toString() 把数组转换为字符串，并返回结果。

toLocaleString() 把数组转换为本地数组，并返回结果。

valueOf() 返回数组对象的原始值

91、个方法 去掉一个数组的重复元素

方法一：

```
1 var arr = [0,2,3,4,4,0,2];
2 var obj = {};
3 var tmp = [];
4 for(var i = 0 ;i< arr.length;i++){
5     if( !obj[arr[i]] ){
6         obj[arr[i]] = 1;
7         tmp.push(arr[i]);
8     }
9 }
10 console.log(tmp);
```

结果如下： [0, 2, 3, 4] 方法二：

```
1 var arr = [2,3,4,4,5,2,3,6],
2     arr2 = [];
3 for(var i = 0;i< arr.length;i++){
4     if(arr2.indexOf(arr[i]) < 0){
5         arr2.push(arr[i]);
6     }
7 }
8 console.log(arr2);
```

结果为： [2, 3, 4, 5, 6]

方法三：

```
1 var arr = [2,3,4,4,5,2,3,6];
2 var arr2 = arr.filter(function(element,index,self){
3     return self.indexOf(element) === index;
4 });
5 console.log(arr2);
```

结果为: [2, 3, 4, 5, 6]

Ajax

1、什么是Ajax?

Ajax (Asynchronous JavaScript + XML) , 即异步JavaScript + XML的缩写, 主要用来页面异步刷新, 也是构建RIA的一种基础技术。

2、如何使用Ajax从服务器获取数据?

①创建XMLHttpRequest对象, 注意兼容IE6的情况 ②使用XMLHttpRequest对象的open方法, 其中有三个参数: a. 字符串, 代表html的请求:GET,POST。 b.要访问的服务器的URL。 c.Boolean值, true表示异步, false表示同步, 一般情况下是异步, 默认为true。 ③ajax的回调函数。xhr.readyState==4表示请求已经结束, 服务器响应完成。status表示http请求的状态, 200表示正常响应; 404表示资源找不到; 500表示服务器端错误。 ④发送ajax请求。如果没有数据, 可以不传或者传递null; 如果post请求传递数据: 首先设置xhr的请求头信息:

```
1 | xhr.setRequestHeader("Content-type","application/x-www-form-urlencoded");
```

再传递参数:

```
1 | xhr.send(name=liujianhong&password=123);
```

3、解释XMLHttpRequest是什么?

XMLHttpRequest是我们得以实现异步通讯的根本。最早在IE5 中以ActiveX组件实现; 最近, Mozilla 1.0和Safari 1.2中实现为本地对象。XMLHttpRequest虽然不是W3C标准, 但却得到了FireFox、Safari、Opera、Konqueror、IE等绝大多数浏览器的支持。

4、谈谈你对Ajax的理解。你在项目中如何使用Ajax? 手写一个简单的Ajax操作。

Ajax (Asynchronous JavaScript + XML) , 即异步JavaScript + XML的缩写, 主要用来页面异步刷新, 也是构建RIA的一种基础技术。因为它涉及浏览器兼容、跨域等问题, 在项目中一般会使用一些基础类库辅助实现, 如jQuery等。一个简单的Ajax操作如下。

```
1 | var xhr = new XMLHttpRequest();
2 | //在环境中需要做浏览器兼容, 这里省略了
3 | xhr.onreadystatechange = function() {
4 | //这里注册当xhr状态发生改变后调用事件
5 | if( xhr.readyState == 4 ) {
6 | //通常在读取状态为4的时候才能获取到部分数据
7 | 所以一般状态在4的时候才进行处理
8 | if(status==200) {
9 | //当正常请求到资源时的处理,
10 | 可以调用xhr.responseText或xhr.responseXml获取数据
11 | }
```

```

12         else {
13             //当请求资源失败时的处理
14         }
15     }
16 }
17 xhr.open( "GET", url);
18 //设置xhr的请求方式和url,这里使用的是GET方式,
19 //如果有参数,则连接在url后面
20 /*
21 如果是POST请求,还当设置请求的Content-Type
22 数据使用send作为参数发送
23 */
24 xhr.send();

```

5、谈谈你对JSON的理解。

JSON(JavaScript Object Notation) 是一种轻量级的数据交换格式。易于人阅读和编写。同时也易于机器解析和生成。它基于JavaScript Programming Language, Standard ECMA-262 3rd Edition - December 1999的一个子集。JSON采用完全独立于语言的文本格式,但是也使用了类似于C语言家族的习惯(包括C, C++, C#, Java, JavaScript, Perl, Python等)。这些特性使JSON成为理想的数据交换语言。所以它往往在AJAX中替代XML, 交换数据。

6、你的项目中有使用到跨域吗? 你在项目中是如何处理JS跨域问题的?

①有。②主要是使用其它网站提供的javascript api如QQ。使用script的src可以直接读取跨域资源。③当然跨域还有其它处理方式: 如代理服务器、改变domain、JSONP等。

7、你在项目中有使用到网页到服务器的即时通信吗? 说说你都采用什么手段处理以及你所知道的处理办法?

没有用到, 但我知道html的websockets、flash的socket、ajax长轮询等都可以实现。

8、你在AJAX中有遇到乱码吗? 如果遇到, 你是如何解决的?

①遇到过。②一般我首先统一页面和服务器编码, 对请求和响应的Content-Type设置正确编码; 对请求参数进行编码处理。

9、解释jsonp的原理, 以及为什么不是真正的ajax

- 1 Ajax与JSONP这两种技术看起来很像, 目的也一样, 都是请求一个url, 然后把服务器返回的数据进行处理, 因此jQuery等框架都把JSONP作为Ajax的一种形式。
- 2 实际上Ajax与JSONP有着本质上的不同。Ajax的核心是通过XMLHttpRequest获取数据, 而JSONP的核心则是动态添加<script>标签来调用服务器提供的js文件。
- 3 Ajax与JSONP的区别也不在于是否跨域, Ajax通过服务端代理也可以跨域, JSONP也可获取同源数据。

10、ajax 有那些优缺点?如何解决跨域问题?

优点: 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。③Ajax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。④Ajax可以实现动态不刷新(局部刷新) **缺点:** ①安全问题 AJAX暴露了与服务器交互的细节。②对搜索引擎的支持比较弱。③不容易调试。 **跨域问题:** jsonp、iframe、window.name、window.postMessage、服务器上设置代理页

面。

11、JavaScript原型，原型链？有什么特点？

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。

JQuery

1、你使用过jQuery吗？如果有，你为什么要使用jQuery呢？

①用过。②如果使用原生javascript开发的话，会面临很多问题，如浏览器兼容、Ajax数据解析、Dom、事件注册操作等都非常烦琐，而jQuery正好解决了这些问题。③当然jQuery还有非常有用的其它特性，如为dom对象绑定数据、动画、等。并且jQuery还非常容易扩展，在它的基础上开发非常灵活，也有众多的插件可用，如jQueryUI、easyUI等。

2、Javascript与jQuery有什么区别？

jquery 就对javascript的一个扩展，封装，就是让javascript更好用，更简单。

3、在jQuery中如何注册事件？

使用bind()方法注册事件，但通常我们使用与事件同名的方法注册更方便，如：click()、hover()等。

4、如何获取Html内容？如何获取文本内容？如何获取属性值？如何获取input值？如何创建新的节点？

- 1 可以使用html()获取html内容。
- 2 使用text()获取文本内容。
- 3 使用attr()可以获取属性值，使用css()可以获取样式属性值。
- 4 通过val()便可以获取input的值
- 5 把节点元素名加上<、>作为参数调用jQuery方法便可创建新节点,如：jQuery("<div>")。

5、如何向页面插入节点？

调用append方法，将新节点作为参数。

6、ajax、get、post、ajaxSetup、getJSON各有何用途？

①ajax：jQuery对ajax执行的核心方法。其它ajax方法都是使用该方法实现。②get：专门用于发送get请求的便捷方法。③post：专门用于发送post请求的便捷方法。④ajaxSetup：设置调用ajax方法时的默认值。⑤getJSON：专门用于向服务器请求json格式数据的便捷方法。

7、如何使用从服务器获取一个复杂数据（对象）？

①通常会把这个数据转换为通用的数据交换格式，如xml或json。由于xml解析比较麻烦，所以使用json比较多。②在jQuery中有专门的获取服务器json数据的方法，getJSON()，在回调中，jQuery会自动将json转换为javascript对象。

8、addClass、css有何用途？

①addClass：为元素设置class属性，如果该元素已经存在class属性，则在其值后添加空格及新的class值。②css：操作元素的style属性的方法。

9、如何获取一个元素的实际位置？

使用position()或offset()都可以。

10、bind()、unbind()、hover()有何用途？

bind()：注册特定事件。unbind()：删除特定事件。hover()：同时注册鼠标移入、移出事件。

11、你知道jQuery插件吗？你了解jQuery执行原理和插件机制吗？你都用过哪些jQuery插件？

①知道jQuery插件。②其原理是扩展jQuery本身及其核心函数的原型实现。可以调用其extend实现对它的扩展。③jQuery插件有很多，常见的有：jQueryUI、jQuery-Cookie、jQuery-Timer等。

12、ery 库中的 \$() 是什么？

()函数是jQuery()函数的别称。()函数用于将任何对象包裹成jQuery对象，接着你就被允许调用定义在jQuery对象上的多个不同方法。你可以将一个选择器字符串传入()函数，它会返回一个包含所有匹配的DOM元素数组的jQuery对象。

13、找到所有 HTML select 标签的选中项？

```
$("[name=selectname]:selected")
```

14、this) 和 this 关键字在 jQuery 中有何不同？

\$(this) 返回一个jQuery对象，你可以对它调用多个jQuery方法，比如用text()获取文本，用val()获取值等等。

而this代表当前元素，它是JavaScript关键词中的一个，表示上下文中的当前DOM元素。你不能对它调用jQuery方法，直到它被()函数包裹，例如(this)。

15、uery怎么移除标签onclick属性？

```
1  获得a标签的onclick属性: $("a").attr("onclick")
2
3  删除onclick属性: $("a").removeAttr("onclick");
4
5  设置onclick属性: $("a").attr("onclick","test();");
```

16、uery中addClass,removeClass,toggleClass的使用。

```
1 $(selector).addClass(class): 为每个匹配的元素添加指定的类名
2
3 $(selector).removeClass(class): 从所有匹配的元素中删除全部或者指定的类, 删除class中某个值;
4
5 $(selector).toggleClass(class): 如果存在 (不存在) 就删除 (添加) 一个类
6
7 $(selector).removeAttr(class); 删除class这个属性;
```

17、ery有几种选择器?

```
1 (1)、基本选择器: #id, class, element, *;
2
3 (2)、层次选择器: parent > child, prev + next , prev ~ siblings
4
5 > $("#one +div") 选取id为one的元素的下一个<div>同辈元素    等同于$("#one").next("div")
6 > $("#one~div") 选取id为one的元素的元素后面的所有<div>同辈元素    等同于
  $("#one").nextAll("div")
7
8 (3)、基本[过滤器] (https://link.jianshu.com?t=http%3A%2F%2Fzhidao.baidu.com%2Fsearch%3Fword%3D%25E8%25BF%2587%25E6%25BB%25A4%25E5%2599%25A8%26fr%3Dqb_search_exp%26ie%3Dutf8) 选择器: :first, :last, :not, :even, :odd, :eq, :gt, :lt
9
10 (4)、内容[过滤器] (https://link.jianshu.com?t=http%3A%2F%2Fzhidao.baidu.com%2Fsearch%3Fword%3D%25E8%25BF%2587%25E6%25BB%25A4%25E5%2599%25A8%26fr%3Dqb_search_exp%26ie%3Dutf8) 选择器: :contains, :empty, :has, :parent
11
12 (5)、可见性[过滤器] (https://link.jianshu.com?t=http%3A%2F%2Fzhidao.baidu.com%2Fsearch%3Fword%3D%25E8%25BF%2587%25E6%25BB%25A4%25E5%2599%25A8%26fr%3Dqb_search_exp%26ie%3Dutf8) 选择器: :hidden, :visible
13
14 (6)、属性过滤器选择器: [attribute], [attribute=value], [attribute!=value], [attribute^=value], [attribute$=value], [attribute*=value]
15
16 (7)、子元素过滤器选择器: :nth-child, :first-child, :last-child, :only-child
17
18 (8)、表单选择器: :input, :text, :password, :radio, :checkbox, :submit 等;
19
20 (9)、表单过滤器选择器: :enabled, :disabled, :checked, :selected
```

18、ery中的Delegate()函数有什么作用?

delegate()会在以下两个情况下使用到:

1、如果你有一个父元素, 需要给其下的子元素添加事件, 这时你可以使用delegate()了, 代码如下:

```
1 | $("ul").delegate("li", "click", function(){ $(this).hide(); });
```

2、当元素在当前页面中不可用时, 可以使用delegate()

19、document).ready()方法和window.onload有什么区别?

(1)、window.onload方法是在网页中所有的元素(包括元素的所有关联文件)完全加载到浏览器后才执行的。

(2)、\$(document).ready() 方法可以在DOM载入就绪时就对其进行操纵，并调用执行绑定的函数。

20、用jQuery禁用浏览器的前进后退按钮？

```
1 <script type="text/javascript" language="javascript">
2     $(document).ready(function() {
3         window.history.forward(1);
4         //OR window.history.forward(-1);
5     });
6 </script>
```

21、uery中.get()提交和.post()提交有区别吗？

相同点：都是异步请求的方式来获取服务端的数据；

异同点：

- 1、请求方式不同：.get()方法使用GET方法来进行异步请求的。.post()方法使用POST方法来进行异步请求的。
- 2、参数传递方式不同：get请求会将参数跟在URL后进行传递，而POST请求则是作为HTTP消息的实体内容发送给Web服务器的，这种传递是对用户不可见的。
- 3、数据传输大小不同：get方式传输的数据大小不能超过2KB 而POST要大的多
- 4、安全问题：GET方式请求的数据会被浏览器缓存起来，因此有安全问题。

22、一个简单的\$.ajax()的请求方式？

```
1 $.ajax({
2     url: 'http://www.baidu.com',
3     type: 'POST',
4     data: data,
5     cache: true,
6     headers: {},
7     beforeSend: function() {},
8     success: function() {},
9     error: function() {},
10    complete: function() {}
11 });
```

23、uery的事件委托方法bind、live、delegate、on之间有什么区别？

(1)、bind 【jQuery 1.3之前】 定义和用法：主要用于给选择到的元素上绑定特定事件类型的监听函数；

语法：bind(type,[data],function(eventObject));

特点：

- (1)、适用于页面元素静态绑定。只能给调用它的时候已经存在的元素绑定事件，不能给未来新增的元素绑定事件。
- (2)、当页面加载完的时候，你才可以进行bind()，所以可能产生效率问题。

实例如下: `$("#members li a").bind("click", function(e) {});`

(2)、live 【jQuery 1.3之后】

定义和用法: 主要用于给选择到的元素上绑定特定事件类型的监听函数;

语法: `live(type, [data], fn);`

特点:

(1)、live方法并没有将监听器绑定到自己(this)身上, 而是绑定到了this.context上了。

(2)、live正是利用了事件委托机制来完成事件的监听处理, 把节点的处理委托给了document, 新添加的元素不必再绑定一次监听器。

(3)、使用live () 方法但却只能放在直接选择的元素后面, 不能在层级比较深, 连缀的DOM遍历方法后面使用, 即 `("ul").live... 可以, 但("body").find("ul").live...不行;`

通过 live() 方法附加的事件处理程序适用于匹配选择器的当前及未来的元素 (比如由脚本创建的新元素) 。

(3)、delegate 【jQuery 1.4.2中引入】

定义和用法: 将监听事件绑定在就近的父级元素上

语法: `delegate(selector,type,[data],fn)`

特点:

(1)、选择就近的父级元素, 因为事件可以更快的冒泡上去, 能够在第一时间进行处理。

(2)、更精确的小范围使用事件代理, 性能优于.live()。可以用在动态添加的元素上。

实例如下:

`$("#info_table").delegate("td", "click", function() { / 显示更多信息 / });`

`$("table").find("#info").delegate("td", "click", function() { / 显示更多信息 / });`

(4)、on 【1.7版本整合了之前的三种方式的新事件绑定机制】

定义和用法: 将监听事件绑定到指定元素上。

语法: `on(type,[selector],[data],fn)`

实例如下: `$("#info_table").on("click", "td", function() { / 显示更多信息 / });` 参数的位置写法与delegate不一样。

说明: on方法是当前jQuery推荐使用的事件绑定方法, 附加只运行一次就删除函数的方法是one()。

总结: .bind(), .live(), .delegate(), .on()分别对应的相反事件为: .unbind(), .die(), .undelegate(), .off()

其它

1、HTTP状态码知道哪些?

100 Continue 继续, 一般在发送post请求时, 已发送了http header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 200 OK 正常返回信息 201 Created 请求成功并且服务器创建了新的资源 202 Accepted 服务器已接受请求, 但尚未处理 301 Moved Permanently 请求的网页已永久移动到新位置。 302 Found 临时性重定向。 303 See Other 临时性重定向, 且总是使用 GET 请求新的 URI。 304 Not Modified 自从上次请求后, 请求的网页未修改

过。400 Bad Request 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。401 Unauthorized 请求未授权。403 Forbidden 禁止访问。404 Not Found 找不到如何与 URI 相匹配的资源。500 Internal Server Error 最常见的服务器端错误。503 Service Unavailable 服务器端暂时无法处理请求（可能是过载或维护）。

2、你有哪些性能优化的方法

(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。

3、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。

4、哪些常见操作会造成内存泄漏？

①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。②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。③setTimeout 的第一个参数使用字符串而非函数的话，会引发内存泄漏。④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）

5、线程与进程的区别

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。

6、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？

(1) 查找浏览器缓存 (2) DNS解析、查找该域名对应的IP地址、重定向 (301)、发出第二个GET请求 (3) 进行HTTP协议会话 (4) 客户端发送报头(请求报头) (5) 服务器回馈报头(响应报头) (6) html文档开始下载 (7) 文档树建立, 根据标记请求所需指定MIME类型的文件 (8) 文件显示

7、以下自行发挥

①为什么换工作 ②你常用的开发工具是什么，为什么？ ③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？ ④加班的看法？ ⑤有没有女朋友

8、次完整的HTTP事务是怎样的一个过程？

a. 域名解析 b. 发起TCP的3次握手 c. 建立TCP连接后发起http请求 d. 服务器端响应http请求，浏览器得到html代码 e. 浏览器解析html代码，并请求html代码中的资源 f. 浏览器对页面进行渲染呈现给用户

9、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。

10、何判断当前脚本运行在浏览器还是node环境中？（阿里）

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

11、你所了解到的web攻击技术

①XSS（Cross-Site Scripting，跨站脚本攻击）：指通过存在安全漏洞的Web网站注册用户的浏览器内运行非法的HTML标签或者JavaScript进行的一种攻击。②SQL注入攻击 ③CSRF（Cross-Site Request Forgeries，跨站点请求伪造）：指攻击者通过设置好的陷阱，强制对已完成的认证用户进行非预期的个人信息或设定信息等某些状态更新。

12、Jquery是什么？

jQuery是一个快速、简洁的JavaScript框架，是继Prototype之后又一个优秀的JavaScript代码库（或JavaScript框架）。

jQuery设计的宗旨是“write Less，Do More”，即倡导写更少的代码，做更多的事情。它封装JavaScript常用的功能代码，提供一种简便的JavaScript设计模式，优化HTML文档操作、事件处理、动画设计和Ajax交互。

jQuery的核心特性可以总结为：①具有独特的链式语法和短小清晰的多功能接口；②具有高效灵活的css选择器，并且可对CSS选择器进行扩展；③拥有便捷的插件扩展机制和丰富的插件。

jQuery兼容各种主流浏览器，如IE 6.0+、FF 1.5+、Safari 2.0+,Opera 9.0+等。

13、你知道哪些关于ES6新增的东西

[关于ES6新增的东西](#)

14、如何控制网页在网络传输过程中的数据量？

最显著的方法是启用GZIP压缩。此外保持好的编码习惯，避免重复和css、JavaScript代码，多余的HTML标签和属性。

Flash、Ajax各自的优缺点，在使用中如何取舍？

Flash的缺点是需要客户端安装Flash插件，比较大，且更改了默认的HTML页面行为；但可以方便地实现很多特效及动画，且具有较高权限。

Ajax的缺点是编程较为复杂，需要服务器端的支持，能实现的效果只能是DOM API提供的，权限很低，较难跨域；但可以显著加快页面的载入速度和用户体验。

此外，二者都不能被搜索引擎索引（Google已支持Flash文本的索引），不利于SEO。

建议:重要和关键部分直接用HTML, 交互部分可以使用Ajax, 复杂的动画可采用Flash。

15、常使用的库有哪些?常用的前端开发工具?开发过什么应用或组件?

最常用的库: jquery-1.4.2.min.js 常用的前端开发工具: firebug、photoshop、editplus、取色器、色板、eclipse

16、说说YSlow

Yslow是雅虎开发的基于网页性能分析浏览器插件, 是基于Mozilla Firefox上firebug插件的一个插件。它的出现的主要目的就是检测我们的页面性能。

它让用户可以就近取得所需的内容, 解决网络拥挤的状况, 提高用户访问网站的响应速度。

17、是否了解flex布局?

参照阮一峰老师的flex布局教程 [Flex 布局教程: 语法篇](#) [Flex 布局教程: 实例篇](#)

18、是否了解webpack

[看完让你彻底搞懂Websocket原理](#)

19、webSocket如何兼容低浏览器?

Adobe Flash Socket 、 ActiveX HTMLFile (IE) 、 基于 multipart 编码发送 XHR 、 基于长轮询的XHR。

20、简述同步和异步的区别

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。

21、异步ajax的优缺点都有什么?

优点: • 相对于同步ajax: 不会造成UI卡死, 用户体验好。• 相对于刷新页面, 省流量 缺点: • 后退按钮无效; • 多个请求同时触发时, 由于回调时间不确定, 会造成混乱, 避免这种混乱需要复杂的判断机制。• 搜索引擎不友好 • 数据安全

22、如何提高网页运行性能?

[如何提高网页的运行性能](#)

23、对前端工程师这个职位你是怎么样理解的?

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

24、网站重构的理解?

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JS DOM的优化 HTTP服务器的文件缓存

25、AMD和CMD 规范的区别？

AMD规范是 RequireJS 在推广过程中对模块定义的规范化产出的，而CMD规范是Seajs 在推广过程中对模块定义的规范化产出的。这些规范的目的都是为了 JavaScript 的模块化开发，特别是在浏览器端的。主要区别总结为以下三条 1、CMD推崇依赖就近，而AMD推崇依赖前置 2、执行顺序上：CMD是延迟执行的，而AMD是提前执行的。 3、api设计角度上：AMD 的 API 默认是一个当多个用，CMD 的 API 严格区分，推崇职责单一。比如 AMD 里，require 分全局 require 和局部 require，都叫 require。CMD 里，没有全局 require，而是根据模块系统的完备性，提供 seajs.use 来实现模块系统的加载启动。CMD 里，每个 API 都简单纯粹。

26、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)

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DOM Tree，其次会根据CSS生成CSS Rule Tree，而javascript又可以根据DOM API操作DOM。

27、WEB应用从服务器主动推送Data到客户端有那些方式？

JavaScript数据推送 Comet：基于HTTP长连接的服务器推送技术 基于WebSocket的推送方案 SSE (Server-Send Event)：服务器推送数据新方式