面试

JS变量可以是局部变量也可以是全局变量，私有变量可以用到闭包。在web页面中全局变量属于 window 对象，局部变量只能用于定义它函数内部。变量声明时如果不使用 var 关键字，那么它就是一个全局变量，即便它在函数内定义。

|  |  |
| --- | --- |
| 1.盒子模型： | 模型有两中，S3C和ie；   1. W3C盒子模型包括：margin、border、padding、content ，元素的width=content的宽度； 2. IE盒子模型与W3C的盒子模型唯一区别就是元素的宽度，元素的width=content+padding+border； |
| 2.promise： | Promise 是异步编程的一种解决方案，比传统的解决方案——回调函数和事件——更合理和更强大。它由社区最早提出和实现，ES6 将其写进了语言标准，统一了用法，原生提供了Promise对象。 |
| 3.数组排序： |  |
| 4.闭包： | 闭包是就是函数中的函数，里面的函数可以访问外面函数的变量，外面的变量的是这个内部函数的一部分。闭包是一种保护私有变量的机制，在函数执行时形成私有的作用域，保护里面的私有变量不受外界干扰。直观的说就是形成一个不销毁的栈环境。  闭包的作用：  1.使用闭包可以访问函数中的变量；2.可以使变量长期保存再内存中，生命周期比较长。  闭包不能滥用，否则会导致内存泄露，影响网页的性能。闭包使用完了后，要立即释放资源，将引用变量指向null。 |
| 5.异步处理： |  |
| 6.跨域请求： |  |
| 7.JSON格式： |  |
| 8.iframe缺点 | 1. iframe会阻塞主页面的Onload事件；2.  搜索引擎的检索程序无法解读这种页面，不利于SEO;3. 会影响页面的并行加载。解决办法：使用js动态给iframe的src加载页面内容 |
| 9.元素定位 | Position为元素定位属性 1. **absolute**绝对定位，相对位置为父元素为非static的第一个父元素进行定位。2. **fixed**固定定位（老IE6不支持），相对于浏览器窗口进行定位。3.relative相对定位，相对于默认布局进行定位。4. **Static，**默认值。没有定位，元素出现在正常的流中（忽略 top, bottom, left, right z-index 声明）。加分项：所有的定位如果left、top、right、bottom属性都为默认值，则为默认定位；absolute定位会脱离文档，浮动起来，多个层重叠可以使用z-index属性改变层叠顺序；solute定位忽略padding，相对位置为相对定位容器的左上角内边框。 |
| 10.样式导入方式 | Link/import：link除了引用样式文件，还可以引用图片等资源文件，而import只引用样式文件。 |
| 11.::before 和:befor区别 | 1.都可以用来表示伪类对象写法是等效的；2.::是css3:是css2，3.伪类对象要配合content属性一起使用，伪类对象不会出现在DOM中，所以不能通过js来操作，仅仅是在 CSS 渲染层加入，伪类对象的特效通常要使用:hover伪类样式来激活 |
| 12.CSS样式优先级 | 1. 优先级顺序为：**!important>style>权重值，2.**如果权重相同，则最后定义的样式的起作用，应该避免这种情况出现；标签的权重为1， class的权重为10 ，id的权重为100。 |
| 13.如何居中一个元素(正常，绝对，定位)？ | 1. 行级元素水平居中对齐（父元素设置 text-align:center）；2.  块级元素水平居中对齐（margin: 0 auto）3. 浮动元素水平居中；4. 让绝对定位的元素水平居中对齐；水平居中的主要属性有：1. text-alin:center; 2. margin:0 auto 3. position:relative|absolute; left:50%; |
| 14.CSS样式-display有那些作用？ | block(块级元素)，<div>、<p> <ul> <ol> <form> ……；  inline（内联元素），<span> <a> <img> <input> <select> <label> ……  list-item（项目列表），<li>  none(不显示) head>（头部元素都是） <title> <br> <thead> <tbody> <tfoot> |
| 15.为什么要初始化CSS样式？ | 因为浏览器的兼容的问题，不同浏览器有些标签的默认值是不同的，如果没有CSS初始化往往会出现浏览器之间的页面显示差异 |
| 16.一个满屏的吕字布局如何实现？ | 1.元素水平居中对齐，使用margin对齐(推荐)，使用left：50%，使用text-align；2。元素对相对窗口定位，使用filxed（推荐），使用absolute定位，使用html和body的width和height填?这个窗口；3.元素左右定位，使用float左右浮动，使用绝对定位进行左右定位（推荐） |
| 17.用纯CSS创建一个三角形原理？ | 采用的是均分原理，盒子都是一个矩形或正方形，从形状的中心，向4个角上下左右划分4个部。第一步 保证元素是块级元素，第二步 设置元素的边框，第三步 不需要显示的边框使用透明色。 |
| 18.CSS样式如何清除浮动？ | 1.使用clear:both；2.overflow属性；3.display属性；4.父级元素浮动； |
| 19.javascript中如何翻转字符串？ | 1.使用字符串函数；2.使用for循环；3.使用递归 |
| 20.如何向数组中插入或者删除元素？ | 1.使用splice方法，splice(start, deleteCount, value, ...)  2.常规方法； |
| 21.用js实现干位分割符？ | 1.使用正则表达式，正则表达式 \d{1,3}(?=(\d{3})+$)  表示前面有1~3个数字，后面的至少由一组3个数字结尾。?=表示正向引用，可以作为匹配的条件，但匹配到的内容不获取，并且作为下一次查询的开始。 $& 表示与正则表达式相匹配的内容，具体的使用可以查看字符串replace()方法；2.正常思维算法； |
| 22.创建，添加，移除，移动和查找节点？ | 1.创建元素createDocumentFragment();2 .createElement();3. createTextNode() ; 移除元素parentNode.removeChild(childNode);  替换元素parentNode.replaceChild(newChild,oldChild); 插入元素parentNode.insertBefore(newChild,refChild); 追加元素parentNode.appendChild(childNode); |
| 23.事件是什么？如何阻止事件冒泡？ | 事件是用于监听浏览器的操作行为，浏览器触发动作时被捕捉到而调用相应的函数。事件执行三个阶段① 事件捕获阶段 ② 处于目标阶段③ 事件冒泡阶段。捕获型事件是自上而下，而冒泡型事件是自下而上的，而我们程序员通常要做的就是第二阶段，完成事件的动作。而第一、三阶段由系统封装自动调用完成。冒泡阻止event.stopPropagation()IE浏览器event.cancelBubble = true; |
| 24.jquery中的bind,live,delegate,on的区别？ | Bind：把事件绑定到每一个匹配的元素上，主要特点：需要循环5次来绑定事件！  Live：把所有的事件都绑定到jquery对象$(document) ，主要特点：只执行一次，事件绑定在根节点对象上。  delegate：将事件绑定到指定的父元素上，和live类似但比较能活。主要特点：只执行一次，事件绑定在调用对象上。  On：一般不怎么用。 |
| 25.json的格式？ | Json是什么？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。是前后台数据交互最常见的一种数据格式。  Json语法：在建值对中：key:value.如{"age":"12", "name":"back","phone"};数据格式转换：1.JSON字符串转换为JSON对象: var obj =eval('('+ str +')'); var obj = JSON.parse(str); 2.JSON对象转换为JSON字符串：var last=JSON.stringify(obj); 3.数组转json字符串 var array=[1,2,3,4];  JSON.stringify($(array)); 4.json字符串转数组，使用jquery  $(JSON.parse('{"0":1,"1":2,"2":3,"length":3}')); |
| 26.原型是什么？原型链？ | 原型：在JavaScript中原型是一个prototype对象，用于表示类型之间的关系。原型链：JavaScript万物都是对象，对象和对象之间也有关系，并不是孤立存在的。对象之间的继承关系，在JavaScript中是通过prototype对象指向父类对象，直到指向Object对象为止，这样就形成了一个原型指向的链条，专业术语称之为原型链。  Student → Person → Object   ，学生继承人类，人类继承对象类 |
| 27.闭包是什么？为什么要用它？ | 闭包是就是函数中的函数，里面的函数可以访问外面函数的变量，外面的变量的是这个内部函数的一部分。 |
| 28.ajax是什么？如何创建它？ | 思路最好先回答原生态Ajax使用的步骤，最后告诉面试官，我们在工作中是使用Jquery来实现Ajax请求和处理的。在回答的时候尽量详尽，包括方法的参数的个数，顺序和作用，以免被面试官抓住不放。不给面试官提问的机会，那你的回答就牛逼了！  Ajax：我理解Ajax 是一种异步请求数据的一种技术，对于改善用户的体验和程序的性能很有帮助。  使用：1.创建Ajax核心对象XMLHttpRequest；  2. 向服务器发送请求 xmlhttp.open(method,url,async);  send(string) ;  open 的参数要牢记，很多面试官爱问这样的细节, method：请求的类型；GET 或 POST, url：文件在服务器上的位置, async：true（异步）或 false（同步）;   1. 服务器响应处理:同部处理：直接在send()后面处理返回来的数据。异步处理：异步处理相对比较麻烦，要在请求状态改变事件中处理。 2. 请求状态：0: 请求未初始化，1: 服务器连接已建立，2: 请求已接收，3: 请求处理中，4: 请求已完成，且响应已就绪。 3. Xmlhttp.status:响应状态码，常见的有：200: "OK" 403   （禁止） 服务器拒绝请求。404   （未找到） 服务器找不到请求的网页。408  （请求超时） 服务器等候请求时发生超时。500   （服务器内部错误）  服务器遇到错误，无法完成请求。 |
| 29.js有几种类型的值？画内存图？ | 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）  不同点：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 |
| 30.对this的理解？ | 1. This的指向：this表示当前对象，this的指向是根据调用的上下文来决定的，默认指向window对象，指向window对象时可以省略不写。 2. 调用的上下文环境包括全局和局部 3. 全局环境就是在<script></script>里面，这里的this始终指向的是window对象； 4. 在全局作用域下直接调用函数，this指向window，对象函数调用，哪个对象调用就指向哪个对象，使用 new 实例化对象，在构造函数中的this指向实例化对象。使用call或apply改变this的指向 5. 用于区分全局变量和局部变量，需要使用this |
| 31.同步和异步有什么差别？ | 同步：种线性执行的方式，执行的流程不能跨越。一般用于流程性比较强的程序，我们做的用户登录功能也是同步处理的，必须用户通过用户名和密码验证后才能进入系统的操作。  异步：是一种并行处理的方式，不必等待一个程序执行完，可以执行其它的任务。在程序中异步处理的结果通常使用回调函数来处理结果。在JavaScript中实现异步的方式主要有Ajax和H5新增的Web Worker。  一个人边吃饭，边看手机，边说话，就是异步处理的方式。  同步处理就不一样了，说话后在吃饭，吃完饭后在看手机，必须等待上一件事完了，才执行后面的事情。 |
| 32.对模块化开发的理解？ | 模块化开发就是封装细节，提供使用接口，彼此之间互不影响，每个模块都是实现某一特定的功能。模块化开发的基础就是函数  **使用函数封装** （与其他模块发生变量名冲突，而且模块成员之间看不出直接关系），**使用对象封装（**封装在obj对象里。使用的时候，就是调用这个对象的属性**），立即执行函数写法** （（Immediately-Invoked FunctionExpression，IIFE），可以达到不暴露私有成员的目的。这个也是闭包处理的一种方式。），放大模式（如果一个模块很大，必须分成几个部分，或者一个模块需要继承另一个模块，这时就有必要采用"放大模式"（augmentation）。在原有的基础上扩展更多的方法。），**宽放大模式（**在浏览器环境中，模块的各个部分通常都是从网上获取的，有时无法知道哪个部分会先加载。如果采用上面的写法，第一个执行的部分有可能加载一个不存在空对象，这时就要采用"宽放大模式"。**），输入全局变量（**独立性是模块的重要特点，模块内部最好不与程序的其他部分直接交互。为了在模块内部调用全局变量，必须显式地将其他变量输入模块。**）** |
| 33.如何实现浏览器内多个标签页之间的通信？ | 1. **使用localStorage，**使用localStorage.setItem(key,value);添加内容，使用storage事件监听添加、修改、删除的动作；  2. **使用cookie+setInterval** |
| 34.call和apply的区别？ | **相同点**：两个方法产生的作用是完全一样的，都用来改变当前函数调用的对象。**不同点：**调用的参数不同。  Call的使用：call([thisObj[,arg1[, arg2[, [,.argN]]]]])，thisObj  可选项。将被用作当前对象的对象。arg1,arg2, , argN  可选项。将被传递方法参数序列。call 方法可以用来代替另一个对象调用一个方法。call 方法可将一个函数的对象上下文从初始的上下文改变为由 thisObj 指定的新对象。如果没有提供 thisObj 参数，那么 Global 对象被用作 thisObj。  Apply：apply与call的功能几乎一样，第一个参数意义都一样，只是第二个参数有点不同apply传入的是一个参数数组，也就是将多个参数组合成为一个数组传入，call从第二个参数开始，依次传值给调用函数的参数 |
| 35.[“1”,”2”,”3”].map(parselnt)? | Map使用：array.map(callback[,thisArg]); 对数组的重新映射。将数组的各个元素依次传入到回调函数callback，回调函数返回的结果依次替换原数组对应的元素。回调函数callback 会被自动传入三个参数：数组元素，元素索引，原数组本身。parseInt(s,radix)解析一个字符串，并返回一个整数。**所以最终的结果是 [1,NaN,NaN]** |
| 36.eval是做什么的？ | 作用：把字符串参数解析成JS代码并运行，并返回执行的结果  作用域：在它所有的范围内容有效  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）。 |
| 37.js延迟加载的方式？ | 程序性能：  defer属性：<scriptsrc="file.js" defer></script>  async属性：<scriptsrc="file.js" async></script>  动态创建DOM方式  使用Jquery的getScript()方法 |
| 38.NEW操作符具体作用？ | 1.创建一个空对象，2.设置原型链，3. 让Func中的this指向obj，并执行Func的函数体，4. 判断Func的返回值类型如果是值类型，返回obj。如果是引用类型，就返回这个引用类型的对象。 |
| 39.CSS3新特性？ | 1.选择器：E:last-child 匹配父元素的最后一个子元素E。E:nth-child(n)匹配父元素的第n个子元素E。E:nth-last-child(n) CSS3 匹配父元素的倒数第n个子元素E。  2. @Font-face 特性  3. 圆角  4. 多列布局 （multi-column layout）  5. 阴影（Shadow）  6. CSS3 的渐变效果  7. css弹性盒子模型  8. CSS3制作特效：Transition 对象变换时的过渡效果，Transforms 2D转换效果，Animation动画特效 |
| 40.HTML5有哪些新特性？移除了哪些元素？ | 新特性： 绘画 canvas;/ 本地离线存储 localStorage/ sessionStorage的数据在浏览器关闭后自动删除;操作参考localStorage/  用于媒介回放的 video和 audio 元素;/ 语意化更好的内容元素，比如article、footer、header、nav、section/表单控件，calendar、date、time、email、url、search/新的技术webworker(专用线程)/ websocketsocket通信/ Geolocation 地理定位  移除元素  <basefont> 默认字体，不设置字体，以此渲染/ <font> 字体标签/   <center> 水平居中/ <u> 下划线/ <big> 大字体/ <strike> 中横线/ <tt> 文本等宽/ <frameset>/ <noframes>/  <frame> |
| 41.如何在页面上实现一个圆形可点击区域？ | 1.map+area；2.border-radius(H5);3.纯js. |
| 42.cookies,sessionstorage,localstorage的区别？ | **相同点：**都存储在客户端  不同点：1.存储大小cookie数据大小不能超过4k。sessionStorage和localStorage 虽然也有存储大小的限制，但比cookie大得多，可以达到5M或更大。2.有效时间localStorage存储持久数据，浏览器关闭后数据不丢失除非主动删除数据；sessionStorage数据在当前浏览器窗口关闭后自动删除。cookie设置的cookie过期时间之前一直有效，即使窗口或浏览器关闭3.数据与服务器之间的交互方式 cookie的数据会自动的传递到服务器，服务器端也可以写cookie到客户端 sessionStorage和localStorage不会自动把数据发给服务器，仅在本地保存。  设置Cookie：cookie的内容：采用 key=value;key=value……存储，参数名自定义；cookie的过期时间：使用参数expires；cookie的路径：使用参数path，"/"表示这个网站的页面，不推荐!容易产生冲突； |
| 43.什么是响应式开发？ | 同一个网站兼容不同的大小的设备。如PC端、移动端（平板、横屏、竖排）的显示风格。 需要用到的技术：1.Media Query（媒体查询）（用于查询设备是否符合某一特定条件，这些特定条件包括屏幕尺寸，是否可触摸，屏幕精度，横屏竖屏等信息。）；2.  使用em或rem做尺寸单位（用于文字大小的响应和弹性布局。）；3.  禁止页面缩放；4. 屏幕尺寸响应 |
| 44.页面渲染原理？ | 渲染引擎是干什么的？渲染引擎可以显示html、xml文档及图片，它也可以借助插件（一种浏览器扩展）显示其他类型数据，例如使用PDF阅读器插件可以显示PDF格式。  渲染引擎  渲染主教程：解析html以构建dom树 -> 构建render树 -> 布局render树 -> 绘制render树  第一步：渲染引擎开始解析html，根据标签构建DOM节点  第二步：根据css样式构建渲染树，包括元素的大小、颜色，隐藏的元素不会被构建到该树中。  第三步：根据css样式构建布局树，主要是确定元素要显示的位置。  第四步：根据前面的信息，绘制渲染树。 |
| 45.跨域请求？ | 跨域请求：基于JavaScript的安全，JavaScript同源策略要求一个网站不能调用其它网站的js对象。构成跨域的条件就是一个页面的URL协议、域名、端口与另一个页面的URL只要有一个不同就构成了跨域请求。  1.jsonp的方式，原生的利用script标签 src 可以请求到任何资源, jquery 也提供了 jsonp的方式，jsonp的方式只支持get方式；2. 使用postMessage跨域 实现跨域  h5新提出来的一个api,支持ie8+,chrome,ff,  这种方式不太像是跨域请求数据的方式，而是跨页面传送数据；3.跨域资源共享 CORS；4. 使用代理去实现跨域请求，经常用于node,或者后台去发送请求； |
| 46.页面优化？ | 1.减少操作量；2.提前做加载操作；3.提升并行加载；4.js和css优化；5.ajax异步加载； |
| 47.浏览器的兼容性？ | 1.html部分：H5新标签在IE9以下的浏览器识别  <!--[if lt IE 9]> <script type="text/javascript" src="js/html5shiv.js"></script><![endif]-->；ul标签内外边距问题ul标签在IE6\IE7中，有个默认的外边距，但是在IE8以上及其他浏览器中有个默认的内边距。解决方法：统一设置ul的内外边距为0  2.CSS样式的兼容性：css的hack问题：主要针对IE的不同版本，不同的浏览器的写法不同；IE6下没有min-width的概念，其默认的width就是min-width  3.JS的兼容性：标准的事件绑定方法函数为addEventListener，但IE下是attachEvent；ajax的实现方式不同，这个我所理解的是获取XMLHttpRequest的不同，IE下是activeXObject |
| 48.web前端项目的结构是怎样的？文件有那些规范？ | js文件：命名为js，css文件：命名为css，图片文件：命名为images，数据文件：命名为data，文件存储规范：按项目模块分类存储，例如：用户信息管理，页面文件存放： shop/userinfo/userlist.html，js文件：js/userinfo/userlist.js，css样式：css/userinfo/userlist.css， |
| 49.css样式书写规范？ | 1.书写顺序：位置属性(position, top, right,z-index, display, float等)/ 大小(width, height, padding,margin)/ 文字系列(font, line-height,letter-spacing, color- text-align等)/ 背景(background, border等)/ 其他(animation, transition等)；  2.书写规范：使用CSS缩写属性CSS有些属性是可以缩写的，比如padding,margin,font等等，这样精简代码同时又能提高用户的阅读体验/去掉小数点前的“0”/简写命名/ 16进制颜色代码缩写有些颜色代码是可以缩写的，我们就尽量缩写吧，提高用户体验为主/长名称或词组可以使用中横线来为选择器命名，不建议使用“\_”下划线来命名CSS选择器/不要随意使用id id在JS是唯一的，不能多次使用，而使用class类选择器却可以重复使用，另外id的优先级优先与class，所以id应该按需使用，而不能滥用。/为选择器添加状态前缀  3.命名规范：  4.注释规范：/\* Header \*/内容区/\* End Header \*/  5.注意事项：一律小写/尽量用英文/尽量不缩写，除非一看就明白的单词。 |
| 50.js的编写规范？ | 简易的原型继承/使用闭包/勿再循环中创建函数/勿使用eval函数/this关键字/数组和对象的属性迭代/ |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
| 开发流程  1.正则表达式判断数据格式：  2.ajax的异步请求数据：  3.html/css进行布局：  4.css/jquery渲染  5.Ajax/json/jsp把需要加载的DOM元素组织，动态加载网页  6.用web的布局在移动设备上自适应，使用HTML5/CSS3新属性方法做布局以及页面特效  7.运用谷歌浏览器进行调试、抓包；  8.日常业务的更新以及维护 |  |
| 了解公司情况： | 1.公司产品介绍  2.盈利模式  3.发展方向 |
| 为什么要录用你？ | 1.能力  2.态度 |
| 对项目的描述 | 项目背景->功能介绍->自己做的部分->所用技术->遇到问题->解决办法； |
| 面试后交流： | 表达能力？问题没有答上？心态？总结自己的面试。 |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**Js :**

数据类型、运算、对象、Function、继承、闭包、作用域、原型链、事件、RegExp、JSON、Ajax、DOM、BOM、内存泄漏、跨域、异步装载、模板引擎、前端MVC、路由、模块化、Canvas、ECMAScript 6、Nodejs

HTML&CSS

对Web标准的理解、浏览器内核差异、兼容性、hack、CSS基本功：布局、盒子模型、选择器优先级、HTML5、CSS3、Flexbox

其他

移动端、响应式、自动化构建、HTTP、离线存储、WEB安全、优化、重构、团队协作、可维护、易用性、SEO、UED、架构、职业生涯、快速学习能力

如何实现浏览器内多个标签页之间的通信?

WebSocket、SharedWorker；也可以调用localstorge、cookies等本地存储方式；

webSocket如何兼容低浏览器？(阿里)

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

前端开发面试题

|  |  |
| --- | --- |
| 05-20 | 1. 纯css创建一个三角形的原理：把上左右三条边边界透明度设置为无色； 2. Vue中组件的注册； |
| 05-29 | 1. 箭头函数：let f=v=>v;变量名=参数=>返回值（函数体）; 2. Set数据类型：类似于数组，成员是唯一的，添加值可以用链式来写；扩展运算符；set可以用来去重操作； 3. Map数据类型：类似于对象，键值对的方式； 4. 留言板实例： 5. Vue指令：-text/-html/-on/show/if/bind/for/model/ 6. For of：用来遍历数组；用键值对的形式； |
| 05-30 |  |
| 06-01 | 1. vue + vuex+ vue-resource + vue-router + webpack + es6 + less； 2. 如何解决跨域问题？JSONP原理是：动态插入script标签，通过script标签引入一个js文件，这个js文件载入成功后会执行我们在url参数中指定的函数，并且会把我们需要的json数据作为参数传入。 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）  继承方式？：构造/原型/实例/拷贝  创建对象的方式？ json/工厂格式（内置对象）/对象字面量/function方式  http状态码有哪些？1\*\*信息类 2\*\*响应成功 3\*\*重定向类 4\*\*客户端错误类 5\*\*服务端错误类  call apply的区别？ 相同之处：代替另一个对象调用一个方法 ，将一个函数的对象上下文从初始的上下文改变为由thisObj指定的新对象 传入的参数列表形式不同；  8. ajax？如何创建？  (1)创建XMLHttpRequest对象,也就是创建一个异步调用对象  (2)创建一个新的HTTP请求,并指定该HTTP请求的方法、URL及验证信息  (3)设置响应HTTP请求状态变化的函数  (4)发送HTTP请求  (5)获取异步调用返回的数据  (6)使用JavaScript和DOM实现局部刷新  9. 闭包是什么？为什么要用它？  指有权访问另一个函数作用域中变量的函数，创建闭包的最常见的方式就是在一个函数内创建另一个函数，通过另一个函数访问这个函数的局部变量,  利用闭包可以突破作用链域，将函数内部的变量和方法传递到外部。  闭包的特性？  1.函数内再嵌套函数  2.内部函数可以引用外层的参数和变量  3.参数和变量不会被垃圾回收机制回收  10. 原型，原型链 ？每个对象都会在其内部初始化一个属性，就是prototype(原型)，当我们访问一个对象的属性时，  如果这个对象内部不存在这个属性，那么他就会去prototype里找这个属性，这个prototype又会有自己的prototype，  栈：存放原始数据类型 堆：存放引用数据类型   1. 数组的使用：Es5：indexof(依据索引来查找元素)/slice（截取部分元素然后生成一个新的数组）/push（向数组末尾追加元素）/pop（把数组最后一个元素删掉）/sort（排序）/reverse（反转）/splice（删除添加元素）/concat（连接两个数组）/join（用指定的符号把数组连起来）/unshift（头部添加若干元素）/shift（删除第一个元素）/every/filter/reduce/ 2. Es6中数组的方法：数组创建 from（转换为数组）/of/ 转换为可迭代对象：set（数组过滤）/map 查找：find/findIndex/ 填充：fill/ copyWithin；遍历 entries/keys/values/包含：includes/ 扩展运算符； 3. 构造函数和原型：prototype的作用是共享方法、节约内存； 4. 对象原型：对象自身的一种属性，它指向原型对象prototype，创建的对象实例里面有\_\_ proto \_\_，它指向构造函数的prototype，因此可以使用prototype里面的方法。 5. 原型链：依据原型链一层一层往上查找，实例对象→→→构造函数原型对象→→→Object原型对象； 6. XML和json的区别： 7. Webpack？模块打包工具；特色：（1.code splitting/2.loader） 8. Tcp传输的三次握手四次挥手策略： 9. Tcp和udp的区别？ 10. 作用域链的理解：作用域链的作用是保证执行环境里有权访问的变量和函数是有序的，作用域链的变量只能向上访问，变量访问到window对象即被终止，作用域链向下访问变量是不被允许的。 11. 创建ajax过程？ 12. Web安全以及防护原理？ 13. http和https？ 14. 对前端模块化的认识？ 15. 垃圾回收方法？标记清除，引入计数； 16. 性能优化问题？代码层面：避免使用css表达式，避免使用高级选择器，通配选择器。 17. 什么是etag？ 18. Es6的了解？ 19. 对闭包的理解？使用闭包主要是为了设计私有的方法和变量。闭包的优点是可以避免全局变量的污染，缺点是闭包会常驻内存，会增大内存使用量，使用不当很容易造成内存泄露。在js中，函数即闭包，只有函数才会产生作用域的概念；三大特性：（1.函数嵌套函数 2.函数内部可以引用外部的参数和变量 3.参数和变量不会被垃圾回收机制回收）； 20. 请你谈谈Cookie的弊端？缺点：数量和长度的限制/存在安全风险/用户配置为禁用 优点：数据持久/简单/ 21. 浏览器本地存储？ 22. web storage和cookie的区别？ 23. cookie 和session 的区别？ 24. CSS中link 和@import的区别是？ 25. CSS3有哪些新特性？ |
| 06-02 | 1. 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；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） 2. 同源策略？:同源策略是客户端脚本（尤其是Javascript）的重要的安全度量标准，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。一段脚本只能读取来自同一来源的窗口和文档的属性； 3. Get/set？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是通过地址栏来传值，而Post是通过提交表单来传值，post更安全； 4. 如何阻止冒泡？ev.stopPropagation()`; 5. Ajax的缺点？ 6. 对重构的理解？ 7. Promise？ Promise 有四种状态：pending: 初始状态, 非fulfilled 或 rejected.fulfilled: 成功的操作.rejected: 失败的操作.settled: Promise已被fulfilled或rejected，且不是pending; Promise 对象用来进行延迟(deferred) 和异步(asynchronous ) 计算; var promise = new Promise(function(resolve, reject) { if (...) { // succeed resolve(result); } else { // fails reject(Error(errMessage)); then,all,race,catch |
|  |  |
|  |  |
|  |  |

封装：

封装就是把抽象的数据和对数据进行的操作封装在一起，数据被保存在内部，程序的其他部分只有通过被授权的操作（成员方法）才能对数据进行操作。

继承：

继承可以解决代码复用问题，让我们编程更加靠近人类的思维，当多个类存在相同的属性（变量）和方法时，可以从这些类中抽象出父类（比如刚才的Student），在父类中定义这些相同的属性和方法，所有的子类不需要重新定义这些属性和方法，只需要通过extend语句来声明继承 父类

多态：

所谓多态，就是指一个引用（类型）在不同情况下的多种状态，你也可以这样理解：多态是指通过指向父类的指针，来调用在不同子类中实现的方法。

闭包：

内部函数总是可以访问其所在的外部函数中声明的参数和变量。

闭包找到的是同一地址中父级函数中对应变量最终的值

一个函数能访问外部函数的变量

2020-05-18

1. H5手机端（移动端）开发；
2. Vue实战经验；
3. 熟练掌握HTML5,CSS3,ES5,ES6,Vue,Webpack等Web前端开发技术；
4. 熟悉主流浏览器常见兼容性问题和解决办法；
5. 善于沟通合作，责任心强，能够承担一定的压力和挑战；
6. 熟悉网络协议（HTTP/SSL）

面试总结

1. 2020-06-04 今天去面试一家外包（交通银行）
2. Css布局（如何垂直居中水平居中，在屏幕的上下左右）
3. Js 闭包
4. Vue 挂载点时间
5. Webpack 打包时的必要参数