

web前端开发题库-app开发功能,需要多少钱-学生个人网页制作html

产品名称	web前端开发题库-app开发功能,需要多少钱-学生个人网页制作html
公司名称	郑州龙之宇科技有限公司
价格	10000.00/套
规格参数	
公司地址	郑州市河南大学大学科技园东区12号楼602
联系电话	19137161875 13017688270

产品详情

web前端题库APP/小程序开发需要有哪些功能:

近期总结一些面试题 都是企业的面试题笔记题 感觉薪资10k下的都会出笔试题

特别高的薪资都是直接技术面试或者是 现场编程

总结很多人的面试题,后期会对于单个知识点再说笔记详细讲解。

更多*新的面试题--请关注微信公众号查看更多信息: 搜索公众号: 胖梅web前端

部分都是百度的答案,不是特全面的,可以自己找下 红色为常见面试题

===== 前端面试题: 1.

一个200*200的div在不同分辨率屏幕上下左右居中,用css实现 `div { ; ; ; top:50%; left:50%; ; ; :100%; }` 2.

写一个左中右布局占满屏幕,其中左右两块是固定宽度200

,中间自适应宽,要求先加载中间块,请写出结构及样式: 实现三列宽度自适应布局

我是左边

我是右边

我是中间

`html,body{ : 0px;: ****; } h3{ : ;20px 0 0;} #left,#{ : ;: #; ;top; } #left{left:0px;} #{: 0px;} #{:2px ;: #eee; ; } 3.`

阐述清楚浮动的几种方式(常见问题) 1.父级div定义

原理: 父级div手动定义,就解决了父级div无法自动获取到高度的问题。

优点: 简单、代码少、容易掌握

缺点: 只适合高度固定的布局,要给出**的高度,如果高度和父级div不一样时,会产生问题

2.父级div定义: 原理: 必须定义或`zoom:1`,同时不能定义,使用:时,浏览器会自动检查浮动区域的高度

优点: 简单、代码少、浏览器支持好 4. 结尾处加空div标签 :both

原理: 添加一个空div,利用css提高的:both清除浮动,让父级div能自动获取到高度

优点: 简单、代码少、浏览器支持好、不容易出现怪问题

缺点: 不少初学者不理解原理;如果页面浮动布局多,就要增加很多空div,让人感觉很不好 5. 解释css

,如何使用? CSS 其实就是把网页中一些背景图片整合到一张图片文件中,再利用CSS的“ ”, “

”, “ ”的组合进行背景定位,可以用数字能**的定位出背景图片的位置。 CSS

为一些大型的网站节约了带宽，让提高了用户的加载速度和用户体验，不需要加载更多的图片 6.

如何用原生js给一个按钮绑定两个事件？
var btn=(' btn ');//事件监听 绑定多个事件 var btn4 = .("btn4");
btn4(""); btn4(""); () { (" 1"); } () { (" 2"); } 7. 拖拽会用到哪些事件 · :拖拽开始时在被拖拽元素上触发此事件,监听器需要设置拖拽所需数据,从操作系统拖拽文件到浏览器时不触发此事件. ·

:拖拽鼠标进入元素时在该元素上触发,用于给拖放元素设置视觉反馈,如高亮 ·

:拖拽时鼠标在目标元素上移动时触发.监听器通过阻止浏览器默认行为设置元素为可拖放元素. ·

:拖拽时鼠标移出目标元素时在目标元素上触发.此时监听器可以取消掉前面设置的视觉效果. ·

drag:拖拽期间在被拖拽元素上连续触发 · drop:鼠标在拖放目标上释放时,在拖放目标上触发.此时监听器需要收集数据并且执行所需操作.如果是从操作系统拖放文件到浏览器,需要取消浏览器默认行为. ·

:鼠标在拖放目标上释放时,在拖拽元素上触发.将元素从浏览器拖放到操作系统时不会触发此事件. 8.

请列举中的选择器： 9. 中的定时器有哪些？他们的区别及用法是什么？只执行一次 会一直重复执行

9.请描述一下 和区别 相同点：都存储在客户端 不同点：1.存储大小 · 数据大小不能超过4k. · 和

虽然也有存储大小的限制，但比大得多，可以达到5M或更大。 2.有效时间 ·

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； ·

数据在当前浏览器窗口关闭后自动删除。 · 设置的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 · 的数据会自动的传递到服务器，服务器端也可以写到客户端 ·

和不会自动把数据发给服务器，仅在本地保存。 10.计算一个数组arr所有元素的和 var

```
arr1=[1,2,3,4,5,6,7,8,9]; var sum1=0; for (var
```

```
i=0;i (业务逻辑处理) -> (数据持久化) ->View (将结果反馈给View)。
```

MVC使用非常广泛，比如中的SSH框架 三、MVVM (I) 如果说MVP是对MVC的进一步改进，那么MVVM则是思想的完全变革。它是将“数据模型数据双向绑定”的思想作为核心，因此在View和之间没有联系，通过进行交互，而且和之间的交互是双向的，因此视图的数据的变化会同时修改数据源，而数据源数据的变化也会立即反应view。 27.px和em的区别 px表示像素(计算机屏幕上的一个点：1px =

1/96in)，是**单位，不会因为其他元素的尺寸变化而变化； · ·

em表示相对于父元素的字体大小。em是相对单位

，没有一个固定的度量值，而是由其他元素尺寸来决定的相对值。 28.优雅降级和渐进增强 渐进增强 (

)：一开始就针对低版本浏览器进行构建页面，完成基本的功能，然后再针对**浏览器进行效果、交互、追加功能达到更好的体验。 优雅降级 (

)：一开始就构建站点的完整功能，然后针对浏览器测试和修复。比如一开始使用 CSS3

的特性构建了一个应用，然后逐步针对各大浏览器进行 hack 使其可以在低版本浏览器上正常浏览。其实渐进增强和优雅降级并非什么新概念，只是旧的概念换了一个新的说法。在传统软件开发中，经常会提到向上兼容和向下兼容的概念。渐进增强相当于向上兼容，而优雅降级相当于向下兼容 29.eval()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； eval("2+3");//执行加运算，并返回运算值。

eval("=10");//声明一个age变量 eval的作用域 () { 1. eval("var x=1");//等效于 var x=1; 2. .log(x); //输出1 3. } 4. a();

5. .log(x); //错误 x没有定 30. JS哪些操作会造成内存泄露 1) 意外的全局变量引起的内存泄露 leak(){

```
leak="xxx";//leak成为一个全局变量，不会被回收 } 2) 闭包引起的内存泄露
```

3) 3) 没有清理的DOM元素引用 4) 被遗忘的定时器或者回调 5) 子元素存在引起的内存泄露 31.

浏览器缓存有哪些，通常缓存有哪几种 一、http缓存 二、 缓存 32:响应式实现的原理

百分比布局+媒体查询 33.关于JS事件冒泡与JS事件代理(事件委托) 事件作为DOM操作重要的一环，需要大家好好理解和运用，今天特意看了一下事件冒泡和事件代理的相关资料，感触颇深，也深感自己的无知不知道多浪费了多少内存，废话不多说进入正题： 1.事件冒泡：通俗易懂的来讲，就是当一个子元素的事件被触发的时候(如事件)，该事件会从事件源(被点击的子元素)开始逐级向上传播，触发父级元素的点击事件。 2.事件委托 事件委托，首先按字面的意思就能看你出来，是将事件交由别人来执行，再联想到上面讲的事件冒泡，是不是想到了？对，就是将子元素的事件通过冒泡的形式交由父元素来执行。下面经过详细的例子来说明事件委托：有可能在开发的时候会遇到这种情况：如导航每一个栏目都要加一个事件，你可能会通过遍历来给每个栏目添加事件： 事件委托是怎 1. var ul = .(""); 2. ul. = () { 3.

```
var e = ||, 4. = e. || e;//表示在事件冒泡中触发事件的源元素，在IE中是 5. if(..() == "li"){
```

```
//判断只有li触发的才会输出内容 6. (.); 7. } 8. (e); //阻止继续冒泡 9. }; 10. () { 11. var li = .('li'); 12.
```

```
li="我是新孩子"; 13. ul.(li); 14. } 34. CSS样式覆盖规则
```

规则一：由于继承而发生样式冲突时，*近祖先获胜。

规则二：继承的样式和直接指定的样式冲突时，直接指定的样式获胜

规则三：直接指定的样式发生冲突时，样式权值高者获胜。

样式的权值取决于样式的选择器，权值定义如下表。CSS选择器 权值 标签选择器 1 类选择器 10 ID选择器 100 内联样式 1000 伪元素(:等) 1 伪类(:link等) 10 可以看到，内联样式的权值>>ID选择器>>类选择器>>标签选择器，除此以外，后代选择器的权值为每项权值之和，比如"#nav.a"的权值为100 + 10 + 1 = 111。

规则四：样式权值相同时，后者获胜。规则五：!的样式属性不被覆盖。!可以看做是万不得已的时候，打破上述四个规则的“金手指”。如果你一定要采用某个样式属性，而不让它被覆盖的，可以在属性值后加上!，以规则四的例子为例，".a{red !;}"可以强行使链接显示红色。大多数情况下都可以通过其他方式来控制样式的覆盖，不能滥用!。

35. 介绍一下属性兼容问题 首先，属性在中存在兼容问题，所以需要使用做一下兼容。属性值 . : . : . : . 这是的默认属性值 . 是CSS2.1中规定的宽度高度的显示行为 .

在CSS中定义的宽度和高度就对应到元素的内容框 .

在CSS中定义的宽度和高度之外绘制元素的内边距和边框 .

在CSS中微元素设定的宽度和高度就决定了元素的边框盒 .

即为元素在设置内边距和边框是在已经设定好的宽度和高度之内进行绘制 .

CSS中设定的宽度和高度减去边框和内间距才能得到元素内容所占的实际宽度和高度 36.

css选择符有哪些？优先级算法如何计算？（常见） 37. 请简要描述重合问题，及解决方式 1.同向的重叠：1图片的与3图片的发生重叠，2图片的与3图片的发生重叠。这时候重叠之后的值由发生重叠两片的*大值决定；如果其中一个出现负值，则由*大的正边距减去**值*大的负边距，如果没有*大正边距，则由0减去**值*大的负边距。解决同向重叠的方法：（1）在*外层的div中加入:zoom:1 （2）在*外层加入:1px;属性 （3）在*外层加入:1px #;

2.异向重叠问题：1图片的与2图片的发生重叠，这时候重叠之后的值由发生重叠两图片的*大值的决定的。

解决异向重叠问题：

:left（只能解决IE6浏览器中的异向重叠问题，可以解决IE8以上、、下的同向重叠问题）

38：的值，\\分别相对于进行谁定位，有什么区别，什么时候用？ 39.解释下CSS

，以及你要如何在页面或网站中使用它。CSS 其实就是把网页中一些背景图片整合到一张图片文件中，再利用CSS的“ ”，“ ”，“ ”的组合进行背景定位，可以用数字能**的定位出背景图片的位置

40.什么是闭包，如何使用它，为什么要使用它？包就是能够读取其他函数内部变量的函数。由于在语言中，只有函数内部的子函数才能读取局部变量，因此可以把闭包简单理解成“定义在一个函数内部的函数”。所以，在本质上，闭包就是将函数内部和函数外部连接起来的一座桥梁。闭包可以用在许多地方。它的*大用处有两个，一个是前面提到的可以读取函数内部的变量，另一个就是让这些变量的值始终保持在内存中。使用闭包的注意点：· 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。· 闭包会在父函数外部，改变父函数内部变量的值。所以，如果你把父函数当作对象（）使用，把闭包当作它的公用方法（），把内部变量当作它的私有属性（），这时一定要小心，不要随便改变父函数内部变量的值。

41.请解释的工作原理，以及它为什么不是真正的AJAX。（JSON with）是一个简单高效的跨域方式，HTML中的标签可以加载并执行其他域的，于是我们可以通过标记来动态加载其他域的资源。例如我要从域A的页面加载域B的数据，那么在域B的页面中我以的形式声明需要的数据，然后在中用标签把加载进来，那么中的脚本就会得以执行。在此基础上加入了回调函数，加载完之后会执行中定义的函数，所需要的数据会以参数的形式传递给该函数。易于实现，但是也会存在一些安全隐患，如果第三方的脚本随意地执行，那么它就可以篡改页面内容，截获敏感数据。但是在受信任的双方传递数据，是非常合适的选择。AJAX是不跨域的，而是一个是跨域的，还有就是二者接收参数形式不一样！

42.请解释一下的同源策略。在客户端编程语言中，如和，同源策略是一个很重要的安全理念，它在保证数据的安全性方面有着重要的意义。同源策略规定跨域之间的脚本是隔离的，一个域的脚本不能访问和操作另外一个域的绝大部分属性和方法。那么什么叫相同域，什么叫不同的域呢？当两个域具有相同的协议，相同的端口，相同的host，那么我们就可以认为它们是相同的域。同源策略还应该对一些特殊情况做处理，比如限制file协议下脚本的访问权限。本地的HTML文件在浏览器中是通过file协议打开的，如果脚本能通过file协议访问到硬盘上其它任意文件，就会出现安全隐患，目前IE8还有这样的隐患。

43.怎样添加、移除、移动、复制、创建和查找节点？ 1) 创建新节点 ent() //创建一个DOM片段 () //创建一个具体的元素 () //创建一个文本节点 2) 添加、移除、替换、插入 () //添加 () //移除 () //替换 () //插入 3) 查找 e() //通过标签名称 () //通过元素的Name属性的值 () //通过元素Id，唯一性

44.谈谈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:)、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：fn1(){ var obj = {name: "", age: 10}; } fn2(){ var obj = {name: "", age: 10}; obj; } var a = fn1(); 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、、、、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。

45. 的事件委托方法bind、live、、on之间有什么区别？(1)、bind【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],());

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：\$("#li a").bind("",(e){}); (2)、live【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.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，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；实例如下：\$.on("", "#li a", (e) {});

(3)、【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：(,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**时间进行处理。

(2)、更**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：

\$("#").("td","",){/*显示更多信息*/}); \$("").find("#info").("td","",){/*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,[],[data],fn) 实例如下：\$("#").on("", "td",){/*显示更多信息*/});参数的位置写法与不一样。

说明：on方法是当前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(), .(),.on()分别对应的相反事件为：.(),.die(), .(),.off() 46、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.

47、浏览器的内核分别是什么? IE: 内核：内核：内核：以前是内核，现已改用的内核：(基于，与

共同开发) 48、什么叫优雅降级和渐进增强？渐进增强：针对低版本浏览器进行构建页面，保证*基本的功能，然后再针对**浏览器进行效果、交互等改进和追加功能达到更好的用户体验。优雅降级：

一开始就构建完整的功能，然后再针对低版本浏览器进行兼容。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 49、、和

之间的区别 共同点：用于浏览器端存储的缓存数据 不同点：

(1)、存储内容是否发送到服务器端：当设置了后，数据会发送到服务器端，造成一定的宽带浪费；web,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：数据不能超过4K,适用于会话标识；web 数据存储可以达到5M;

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

,仅在关闭浏览器之前有效；数据存储**有效；(4)、作用域不同：和是在同源同窗口中都是共享的；不在不同的浏览器窗口中共享，即使是同一个页面； 50、浏览器是如何渲染页面的？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、）与脚本（）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