**解释下浮动和它的工作原理？清除浮动的技巧**

浮动元素脱离文档流，不占据空间。浮动元素碰到包含它的边框或者浮动元素的边框停留。

1.使用空标签清除浮动。

这种方法是在所有浮动标签后面添加一个空标签 定义css clear:both. 弊端就是增加了无意义标签。

**2.使用overflow。**

给包含浮动元素的父标签添加css属性 overflow:auto; zoom:1; zoom:1用于兼容IE6。

**3.使用after伪对象清除浮动。**

该方法只适用于非IE浏览器。具体写法可参照以下示例。使用中需注意以下几点。一、该方法中必须为需要清除浮动元素的伪对象中设置 height:0，否则该元素会比实际高出若干像素；

浮动元素引起的问题和解决办法？

**浮动元素引起的问题**：

（1）父元素的高度无法被撑开，影响与父元素同级的元素

（2）与浮动元素同级的非浮动元素会跟随其后

（3）若非第一个元素浮动，则该元素之前的元素也需要浮动，否则会影响页面显示的结构

**浏览器常见兼容性问题**

**第一类：块状元素float后，有添加了横向的margin，在IE6下比设置的值要大（属于双倍浮动的bug）**

解决方案：给float标签添加display：inline，将其转换为行内元素

**第二类：表单元素行高不一致**

解决方案：给表单元素添加float：left（左浮动）；或者是vertical-align：middle；（垂直对齐方式：居中）

**第三类：设置较小高度的容器（小于10px），在IE6下不识别小于10px的高度；**

解决方案：给容器添加overflow：hidden；

**第四类：当在a标签中嵌套img标签时，在某些浏览器中img会有蓝色边框；**

解决方案：给img添加border：0；或者是border：none；

**第五类：min-height在IE6下不兼容**

**第六类：图片默认有间隙**

解决方案：1）给img标签添加左浮动float：left；2）给img标签添加display：block；

**第七类：按钮默认大小不一**

解决方案：1）用a标签来模拟按钮，添加样式；2）如果按钮是一张背景图片，那么直接给按钮添加背景图；

**第八类：百分比的bug**

解决方案：父元素宽度为100%，子元素宽度各为50%，在IE6下各个元素宽度之和超过100%

解决方案：给右边浮动的子元素添加clear：right；

第九类：鼠标指针bug

描述：cursor：hand；只有ie浏览器识别，其他浏览器不识别

解决方案：cursor：pointer；IE6以上浏览器及其他内核浏览器都识别；

**第十类：透明度属性**

解决方案：针对IE浏览器：filter：alpha（opacity=value）；（取值范围1--100）

兼容其他浏览器：opacity：value；（取值范围0--1）

**第十一类：上下margin的重叠问题**

描述：给上边元素设置了margin-bottom，给下边元素设置了margin-top，浏览器只会识别较大值；

解决方案：margin-top和margin-bottom中选择一个，只设置其中一个值；

**关于hack**

我很少使用hacker的，可能是个人习惯吧，我不喜欢写的代码IE不兼容，然后用hack来解决。不过hacker还是非常好用的。使用hacker我可以把浏览器分为3类：IE6 ；IE7和遨游；其他（IE8 chrome ff safari opera等）

◆IE6认识的hacker 是下划线\_ 和星号 \*

◆IE7 遨游认识的hacker是星号 \*

**BFC是什么？**

怎么触发BFC

可以把BFC理解为一个封闭的大箱子，箱子内部的元素无论如何翻江倒海，都不会影响到外部。

浮动元素：float 除 none 以外的值。

绝对定位元素：position (absolute、fixed)。

display 为 inline-block、table-cells、flex。

overflow 除了 visible 以外的值 (hidden、auto、scroll)。

**88、 网页布局有哪几种，有什么区别**

静态、自适应、流式、响应式四种网页布局

静态布局：意思就是不管浏览器尺寸具体是多少，网页布局就按照当时写代码的布局来布置；

自适应布局：就是说你看到的页面，里面元素的位置会变化而大小不会变化；

流式布局：你看到的页面，元素的大小会变化而位置不会变化——这就导致如果屏幕太大或者太小都会导致元素无法正常显示。

响应式布局：每个屏幕分辨率下面会有一个布局样式，同时位置会变而且大小也会变。

**四种定位的区别**

static 是默认值

relative 相对定位 相对于自身原有位置进行偏移，仍处于标准文档流中

absolute 绝对定位 相对于最近的已定位的祖先元素, 有已定位(指position不是static的元素)祖先元素, 以最近的祖先元素为参考标准。如果无已定位祖先元素, 以body元素为偏移参照基准, 完全脱离了标准文档流。

fixed 固定定位的元素会相对于视窗来定位,这意味着即便页面滚动，它还是会停留在相同的位置。一个固定定位元素不会保留它原本在页面应有的空隙。

**5、（1）请描述一下cookies、sessionStorage和localStorage区别？**

cookie是网站为了标示用户身份而储存在用户本地终端（Client Side）上的数据（通常经过加密）。

cookie数据始终在同源的http请求中携带（即使不需要），即会在浏览器和服务器间来回传递。

sessionStorage和localStorage不会自动把数据发给服务器，仅在本地保存

存储大小：

cookie数据大小不能超过4K。

sessionStorage和localStorage虽然也有存储大小的限制，但比cookie大得多，可以达到5M或更大。

有期时间：

localStorage：存储持久数据，浏览器关闭后数据不丢失除非主动删除数据；

sessionStorage：数据在当前浏览器窗口关闭后自动删除

cookie：设置的cookie过期时间之前一直有效，即使窗口或浏览器关闭

**4、html5有哪些新特性、移除了那些元素？如何处理HTML5新标签的浏览器兼容问题？**

答案解析：

新特性，新增元素：

1）内容元素：article、footer、header、nav、section

2）表单控件：calendar、date、time、email、url、search

3）控件元素：webworker，websockt，Geolocation

移除元素：

1）显现层元素：basefont，big，center，font，s，strike，tt，u

2）性能较差元素：frame，frameset，noframes

**24、css3有哪些新特性？**

CSS3 实现圆角（border-radius:8px;），阴影（box-shadow:10px）,对文字加特效（text-shadow）,线性渐变（gradient），旋转（transform）transform:rotate(9deg) scale(0.85,0.90) translate(0px,-30px) skew(-9deg,0deg);//旋转，缩放，定位，倾斜,增加了更多的 css 选择器 多背景 rgba

**ajax过程**

(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.

**7.如何解决跨域问题？**

由于浏览器同源策略的限制，访问非同源的数据需要进行跨域处理

实现跨域的几种方式:

1. JSONP

2. 后台设置访问源限制

3. 使用后台代理

**解释call、apply、bind的区别?**

call和apply都是调用一个函数，并指定this和参数，call和apply第一个参数都是指定的this的值，区别在于call第二个参数开始的参数是替换的参数，而apply的第二个参数是一个数组。

bind是由一个函数创建一个新函数，并绑定this和部分参数，参数形式和call类似

**16.new操作符具体干了什么?**

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

**什么是SPA？**

Single Page Application 单页应用程序

打开SPA项目的时候，通常只有一个主页面，然后根据点击不同的按钮，切换主页面中显示的内容。

通过点击不同的按钮(a标签)，来修改锚点的值，然后使用路由根据锚点，实现加载不同的页面

不需要进行网页跳转，即可切换页面

**9. 事件委托是什么**

答案: 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！

**10.描述事件委派(delegate)的原理和优点?**

原理: 在容器节点上绑定事件，利用冒泡，判断事件是否在匹配指定的选择器的节点上被触发

优点: 只用绑定一次，不用对每个目标做绑定，还有对动态插入的节点也生效，无需重新绑定

**10. 闭包是什么，有什么特性，对页面有什么影响**

答案：闭包就是能够读取其他函数内部变量的函数,使得函数不被GC回收，如果过多使用闭包，容易导致内存泄露

**11. 如何阻止事件冒泡**

答案：ie:阻止冒泡ev.cancelBubble = true;非IE ev.stopPropagation();

**12. 如何阻止默认事件**

答案：(1)return false；(2) ev.preventDefault();

**浅复制**--->就是将一个对象的内存地址的“”编号“”复制给另一个对象。**深复制**--->实现原理，先新建一个空对象，内存中新开辟一块地址，把被复制对象的所有可枚举的(注意可枚举的对象)属性方法一一复制过来，注意要用递归来复制子对象里面的所有属性和方法，直到子子.....属性为基本数据类型。总结，深复制理解两点，1,新开辟内存地址，2,递归来刨根复制。

**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

**7.JavaScript中常用的设计模式？**

1.单例模式

2.原型模式

3.构造函数

4.工厂模式

5.适配器模式

6.策略模式

**一、什么是MVVM？**

MVVM是Model-View-ViewModel的缩写。MVVM是一种设计思想。Model 层代表数据模型，也可以在Model中定义数据修改和操作的业务逻辑；View 代表UI组件，它负责将数据模型转化成UI 展现出来，ViewModel 是一个同步View 和Model的对象（桥梁）。

在MVVM架构下，View 和 Model 之间并没有直接的联系，而是通过ViewModel进行交互，Model 和 ViewModel 之间的交互是双向的， 因此View 数据的变化会同步到Model中，而Model 数据的变化也会立即反应到View 上。

ViewModel 通过双向数据绑定把 View 层和 Model 层连接了起来，而View 和 Model 之间的同步工作完全是自动的，无需人为干涉，因此开发者只需关注业务逻辑，不需要手动操作DOM, 不需要关注数据状态的同步问题，复杂的数据状态维护完全由 MVVM 来统一管理。

**Vue的双向数据绑定原理是什么？**

答：vue.js 是采用数据劫持结合发布者-订阅者模式的方式，通过Object.defineProperty()来劫持各个属性的setter，getter，在数据变动时发布消息给订阅者，触发相应的监听回调。

**父组件与子组件传值**

父组件通过标签上面定义传值 ：eg='data' 父组件中 data(){return {data:'egdata'}}

子组件通过props方法接受数据 props:['eg'] 在 props 中添加了元素之后，就不需要在 data 中再添加变量了

子组件向父组件传递数据.子组件通过$emit方法传递参数

**vuex中，有默认的五种基本的对象：**

* state：存储状态（变量）
* getters：对数据获取之前的再次编译，可以理解为state的计算属性。我们在组件中使用 $sotre.getters.fun()
* mutations：修改状态，并且是同步的。在组件中使用$store.commit('',params)。这个和我们组件中的自定义事件类似。
* actions：异步操作。在组件中使用是$store.dispath('')
* modules：store的子模块，为了开发大型项目，方便状态管理而使用的。这里我们就不解释了，用起来和上面的一样。

**有哪几种属性？**

有五种，分别是 State、 Getter、Mutation 、Action、 Module

**九、 v-show和v-if指令的共同点和不同点**

v-show指令是通过修改元素的display的CSS属性让其显示或者隐藏

v-if指令是直接销毁和重建DOM达到让元素显示和隐藏的效果

**十九、生命周期相关面试题**

总共分为8个阶段创建前/后，载入前/后，更新前/后，销毁前/后。

创建前/后： 在beforeCreate阶段，vue实例的挂载元素el和数据对象data都为undefined，还未初始化。在created阶段，vue实例的数据对象data有了，el还没有。

载入前/后：在beforeMount阶段，vue实例的$el和data都初始化了，但还是挂载之前为虚拟的dom节点，data.message还未替换。在mounted阶段，vue实例挂载完成，data.message成功渲染。

更新前/后：当data变化时，会触发beforeUpdate和updated方法。

销毁前/后：在执行destroy方法后，对data的改变不会再触发周期函数，说明此时vue实例已经解除了事件监听以及和dom的绑定，但是dom结构依然存在

**（3）、第一次页面加载会触发哪几个钩子**

答：第一次页面加载时会触发 beforeCreate, created, beforeMount, mounted 这几个钩子

（4）、DOM 渲染在哪个周期中就已经完成

答：DOM 渲染在 mounted 中就已经完成了。

**为什么使用key？**

当有相同标签名的元素切换时，需要通过 key 特性设置唯一的值来标记以让 Vue 区分它们，否则 Vue 为了效率只会替换相同标签内部的内容。

**6、Sass、LESS是什么？大家为什么要使用他们？**

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

例如Less是一种动态样式语言. 将CSS赋予了动态语言的特性，如变量，继承，运算， 函数. LESS 既可以在客户端上运行 (支持IE 6+, Webkit, Firefox)，也可一在服务端运行 (借助 Node.js)。

众所周知，浏览器的兼容性问题是开发人员经常会遇到的问题，它是由于不同浏览器的不同对代码的解析不同，而产生的在页面上显示的效果不同。那么，现在我把从工作当中发遇到的有关兼容性问题及解决方案做个简单总结，供各位参考：