平安面试题汇总

1、VUE的生命周期

<https://segmentfault.com/a/1190000008771768>

beforecreate : 可以在这加个loading事件，在加载实例时触发

created : 初始化完成时的事件写在这里，如在这结束loading事件，异步请求也适宜在这里调用

mounted : 挂载元素，获取到DOM节点

updated : 如果对数据统一处理，在这里写上相应函数

beforeDestroy : 可以做一个确认停止事件的确认框

nextTick : 更新数据后立即操作dom

2、邮箱的正则

/^([a-zA-Z0-9-\_])[+@([a-zA-Z0-9-\_])+(.[a-zA-Z0-9-\_])+/](mailto:+@([a-zA-Z0-9-_])+(.[a-zA-Z0-9-_])+/)

3、在VUE中改变一个标题的颜色是在那个生命周期的部分。

Update里面

4、在VUE中不同页面之间如何传递数据，组件通信还有哪几种方式实现。

子组件与父组件

路由

localStorage和sessionStorage

vuex 状态管理

5、觉得VUE有什么优势。

适合中小型项目开发，小85K，大量第三方插件，关键API，语法简单，易懂，可快速学习并加入开发

6、单页面和多页面开发的区别

单页面一般通过#hash的变化达到组件的变化而不引起页面的刷新

多页面是整个页面的跳转，静态资源会重新加载，切换慢，不能做转场动画

7、用js获取url里面的参数值：答案script里面

运用window.location

8、为什么浏览器会有缓存

Web缓存存在于服务器和客户端之间。Web缓存密切注视着服务器-客户端之间的通信，监控请求，并且把请求输出的内容（例如html页面、 图片和文件）另存一份；然后，如果下一个请求是相同的URL，则直接使用保存的副本，而不是再次请求源服务器。

优点：减少网络延迟，加速页面打开速度，缓解服务器压力，减少宽带流量的使用

缺点：缓存问题给开发者带来的困扰，缓存没有清理机制，js无法操作硬盘

15.你是如何解决1px 问题的

-webkit-linear-gradient( top ,transparent , #000) ;

-webkit-transform : scaleY ( 0.5) ;

18.vuex中的数据刷新过后,数据会丢失?如果不存在了,是怎么回事?如何解决这个问题

Js运行在内存中的，代码运行的所有变量，函数也是保存在内存中的。页面刷新的时候会重新加载脚本，变量重新赋值。可以用浏览器自带的缓存API

19.Axios请求的特点

axios.get('url', { params: { } }).then().catch();

axios.post ('url', { }).then().catch();

21.谈谈对MVVM的理解

http://blog.csdn.net/dlg1992/article/details/54138701

MVVM分为3个模块，Model；viewModel;View

View 和ViewModel是双向绑定的，相互影响，处理视图展示逻辑，ViewModel的职能：网络请求、网络请求后数据的处理以及数据模型的转换、数据库操作、页面跳转逻辑、以及相关的数据处理等。

ViewModel和Model是单向绑定的

22.vue实现双向数据绑定的原理

https://www.cnblogs.com/libin-1/p/6893712.html

Vue实现双向绑定是通过劫持数据结合发布者-订阅者实现的，

劫持数据：通过Object.definePropety实现的，他有set和get两个函数

View更新data，只需要触发监听事件就好了，如input标签监听input事件就可以了

23.vue项目中如何解决跨域

用webpack proxy: {

// 请求到 '/device' 下 的请求都会被代理到 target： http://debug.xxx.com 中

'/device/\*': {

target: 'http://debug.xxx.com',

secure: false, // 接受 运行在 https 上的服务

changeOrigin: true

}

}

后台更改Header

header('Access-Control-Allow-Origin:\*');//允许所有来源访问

header('Access-Control-Allow-Method:POST,GET');//允许访问的方式

Jsonp方法如jQuery，设置dataType

他的原理很简单，通过动态 插件script，他的src不用受同源策略规范

var script = document.createElement("script");

script.src = "https://xx.com /search?q=javascript& callback=handleResponse";

document.body.insertBefore(script, document.body.firstChild);

在页面中，返回的JSON作为参数传入回调函数中，我们通过回调函数来来操作数据。

function handleResponse(response){

Console.log(response)

}

24.谈谈http状态码

25.原生js 如何获取兄弟元素

26.谈谈你是如何实现登录15分钟内如果用户没有操作,就退出登录

27.谈谈vue 的生命周期,以及钩子函数

28.Vue-router 中的组件及方法

29.父组件如何传值给子组件,子传父?

30.页面间如何传参?

12、自己写过哪些插件？如何实现的。

恒大面试题汇总

1. 你觉得自己做的最满意的一个项目是什么？

2. 你在这个项目中遇到过什么困难，怎么解决的？

3. 这个项目有多少人？人员配置是怎样的？你负责什么模块？

4. webpack打包后是什么样的，看过文件吗？

5. 使用过sourceMap吗？

6. 使用过devtool吗？

7. 你的项目，整体的架构是什么样的？(项目文件目录)

8. webpack.config.js是怎么配置的

9. vue路由传值有几种方法？

10. 说说ES6的新特性

11. js创建对象有几种方式

12. 为什么会有跨域问题？解决跨域的方式有几种？

13. （接上）同源策略是什么？

14. 解释一下vuex

15. 做过vue的IE9以上的兼容吗？

16. v-if和v-show的区别

17. vue实现数据双向绑定的底层原理及步骤

18. 解释一下Axios？组件在那里请求数据？

19. 了解混合开发吗？

20. 说一下离线存储技术

21. 了解后台吗？做过前后端分离的项目吗？你是怎么处理数据的

22. 使用过哪些UI框架，说说你的使用心得

23. 你对加班怎么看？

24. 你还有什么要问的？

顺丰面试

1.     说一下对vuex的理解

2.     vue生命周期

3.     http状态码,知道哪些.分别怎么解决

4.     以前用过什么前端框架,为什么使用这个?

5.     Js底层原理 ECMA DOM BOM

6.     Css隔行变色

7.     说一下Computed

8.     有没有用过angular

9.     created与mounted之间有什么区别

10.  Axios请求的特点

11.  原生js 如何获取兄弟元素

组件通信还有哪几种方式实现。

平安科技面试题：

1 前端开发中，怎么根据设计图做适配，适应各种手机大小，跟UI怎么沟通方案？

2 开发项目中，怎样搭建环境，用什么打包工具？

3 假如用webpack搭建VUE项目，打包以后，怎么将CSS从JS文件中抽离出来？

4 写react项目中，使用redux怎么处理异步请求的，有哪些缺陷？

5 在 写react项目中，使用过哪些UI框架？你觉得有什么优越性？对你有什么帮助？

6 使用webpack打包后，测试处理过程中，具体怎么控制包大小，有什么方案？

7 谈一下CSS3的一些特性和好处，对你写页面有什么启发？

8 对react native,ant,ant desiger熟悉吗？有什么优点？

9 混合开发中，h5与原生怎么交互的？有什么方案？

10 开发过程中，怎么与后台交互？怎么约定接口？约定参数？111

航交所面试题

1.http协议

2. w3c标准是什么(结构行为表现,bom是行为和dom是表现,结构是ECMA Javascript)

3.用过rem插件吗

4.了解swiper,用过吗?

5.Vue和jq的区别

6.P标签内能嵌套div 吗?

7.H5的新属性

8.Css的新属性

9.自适应布局有几种方式

10.如何优化网站性能

11.用什么管理版本

12.了解过什么前端前沿技术

13.Float

14.class命名有没有自己的一套规则

15.怎样调试设备

16.网页测试(不太理解,后面他说查看页面元素,我说F12,)

17.混合开发 h5如何与ios进行交互

华为面试题

1.用一个div能写出几个圆？

2.浮动的优点及缺点是什么？

3.ajax的优点缺点是什么？

4.生命周期钩子

5.vue中父子组件，兄弟组件如何传值？

6.说下你写的最好的项目，用到了哪些技术

7.在开发过程中，你印象中遇到的最难的问题是什么？

8.水平垂直居中的方法有哪些

9.你对promise是怎么理解的？

10.你对es6了解多少？

11.闭包有了解吗，在项目中哪里有用到？举例

12.你在看源生代码时，最大的收获是什么？

**平安产险面试**

1.flex布局都有哪些属性？

2.var与let,const有什么区别？

3.一个url,按下回车键，他的实现原理是什么？

4.es6新特性有哪些？

5.HTML5新特性？

6.说下你对promise的理解，在哪里有用到？

7.vue中computed与methods的区别？

8.vue组件如何传值？

9.vue生命周期函数有哪些？

10.说下你对vuex的理解？

11.页面用keep-alive缓存后，如果要销毁这个组件，怎么操作？

12.如何优化网站性能?

**vue生命周期：描述实例从创建到销毁之间发生的各种状态**

**1.beforeCreate 创建前执行的钩子函数**

**2.created 实例创建完成**

**3.beforeMount 挂在前**

**4.**[**mounted**](https://cn.vuejs.org/v2/api/#mounted) **挂载后 操作dom 数据请求 将实例挂载在el上**

**5.beforeUpdate**

**6.updated**

**7.beforeDestroy**

**8.destroyed**

VUE/react项目开发中遇到的问题。

1：关于定位的问题，

写web页面的时候，页面布局头部he底部导航栏一开始用的都是固定定位fixed，后面在调试过程在发现在IOS上运行会楚翔抖动的问题，后面把定位改成相对定位后问题解决。

2：首页进入详情页，一开始是采用父子组件的形式，但是这样出现一个问题，就是详细页要把底部导航栏遮盖住，但是由于定位是相对首页的，所以不能遮盖，后面采用了父子页面的写法，在首页组件外添加一个DIV标签，把子页面定位的父级页面相对于App。

3;关于事件监听的问题，在一个页面监听事件的时候，再跳转到另外一个页面，会出现报错，原因是因为监听的组件销毁的时候，监听还在，但是组件没有了，所以报错，所以在生命周期行数组件销毁前要移除监听。

4：首页弹出框，点击关闭的时候会触发下面的列表展示下a标签的点击事件，后面查了相关资料，发现事件的触发顺序是touchstart -> touchmove -> touchend，而移动端存在点击事件300ms延迟，当点击关闭弹出框的时候，弹出框小时，把事件派发给了a标签。所以触发了a标签的跳转。解决的方法有多个，但是后面我查了VUE的官方文档，有vue-tap插件可以解决。

// main.js import vueTap from 'v-tap' // 引入插件 Vue.use(vueTap) // 全局注册

v-tap="{methods:showReceiveModel}" 。

5：引入mint-ui插件需要引入css样式，官方文档中并没有说明，发现引入插件后css动画不显示，后面在main.js中引入了css样式。

6：关于引入MINT-ui中，官方文档中有两种引入方法，一种是全局引入，一种是局部引入，一开始用的全局引入，后面发现占的内存有点大，而已用到的不是很多，后面参考官方文档，使用局部引入，但是发现配置发生了错误，后面是在.babelrc里面配置的。

7：在处理后台请求数据 的时候，后台返回的图片格式不同而且附加有样式，在渲染的时候出现有的图片不能显示。后面需要自己写一个方法对照片格式进行处理。处理方式是判断照片是以什么格式结尾的，然后用if判断。

let hz = path.endsWith('png')?'.png':'.jpeg';

例如：

//fuss10.elemecdn.com/6/4a/ff8686dedbe5e433f134d47310820png.png?imageMogr/format/webp/thumbnail/!130x130r/gravity/Center/crop/130x130/

8：在处理后台返回的时间是一个时间戳，需要自己转化为显示年月日，也是需要自己写一个方法进行转化。

9：vue ios端白屏问题

vue 手机端项目在进入主页后 在进入子页面，直接按返回出现空白情况，然后轻触一下，空白区域就消失了。安卓手机上不会， ios 会出现这种情况

这是safari的ios render blank. 拿到数据之后 this.$nextTick(() => { window.scrollTo(0, 1) window.scrollTo(0, 0) })

10：比较难的就是自己处理一些后台返回的数据，之前就遇到过后台返回一个城市分类的问题。需要按字母首字母来排序。一开始我是直接for循环里面for循环，但是发现这样写的话需要遍历很多次，会给手机端的运行造成很大压力，

后面自己想了很久，找了网上的一些资料，发现很多都是插件，后面我找了一个js插件来研究源码，自己试着写，当时的思路是冒泡排序，但是还涉及到一个去重的问题，所以我一开始是先按字母把所以的城市都归类。先定义一个空字符串，循环返回的数据，if判断字符串中如果不存在这个首字母，就设定这个首字母是空数组，如果存在，就把i中的城市添加到这个数组里，后面讲字符串转成数组，进行冒泡排序。后面就实现了城市排序。

let obj = {};

cities.map(item=>{

//取出首字母

let letter = item.pinyin[0];

//按首字母归类

if(!obj[letter]){

obj[letter] = [];

}

obj[letter].push(item);

})

//将obj转arr

let arr = [];

for(let key in obj){

arr.push({

letter: key,

value: obj[key]

})

}

console.log(arr);

// 对arr按字母排序

for(let i = 0; i < arr.length; i++){

let min = i;

for(let j = i+1; j < arr.length; j++){

if(arr[i].letter>arr[j].letter){

// console.log(min.letter+'进来了'+val.letter);

let tmp;

tmp = arr[i];

arr[i] = arr[j];

arr[j] = tmp;

}

}

}

混合开发与后台合作开发

1：写完自己的web模块后，打包压缩，使用webpack的npm run build，然后把打包的代码放到一个文件夹里面，把所有页面使用ant压缩，然后给到原生使用我的本地包，然后放到app文件里边。

2：调用原生的方法，后台会提供方法，然后把这些方法封装在一个js文件里边，然后通过export default暴露出去，需要调用的地方通过import引入。

3：js文件里需要判断设备是安卓还是ios，通过window.navigator.userAgent判断设备类型，通过不同的设备调用不同的方法。

var UA = window.navigator.userAgent, isAndorid = /android/i.test(UA), isIphone = /iphone/i.test(UA), //isIphone = /(?:iPhone)/.test(UA), isPad = /ipad/i.test(UA), isDolphin = typeof dolphin !== 'undefined'; device = isAndorid ? 'android' : isIphone ? 'iphone' : isPad ? 'ipad':'unknow';

附加匹配主流浏览器设备：

用JavaScript代码来判断浏览器类型及设备访问该网页，支持手机上所有的浏览器。具体代码如下：

// 检测设备类型 var UA = window.navigator.userAgent, isQQBrowser = /(?:MQQBrowser | QQ)/.test(UA), isWindowsPhone = /(?:Windows Phone)/.test(ua), isSymbian = /(?:SymbianOS)/.test(ua) || isWindowsPhone, isAndroid = /(?:Android)/.test(ua), isFireFox = /(?:Firefox)/.test(ua), isChrome = /(?:Chrome|CriOS)/.test(ua), isIpad = /(?:iPad|PlayBook)/.test(ua), isTablet = /(?:iPad|PlayBook)/.test(ua)||(isFireFox && /(?:Tablet)/.test(ua)), isSafari = /(?:Safari)/.test(ua), isPhone = /(?:iPhone)/.test(ua) && !isTablet, isOpen= /(?:Opera Mini)/.test(ua), isUC = /(?:UCWEB|UCBrowser)/.test(ua), isPc = !isPhone && !isAndroid && !isSymbian;

**原生js与jQuery的区别**

不使用框架的话可以根据自己的业务写自己的代码，更有针对性，没有多余的代码；缺点是如果经验不足的话代码的兼容性和扩展能力较差，代码抽象的层次较低，要想做好的话需要经验和大量的时间和精力，最后也许会形成自己的框架。

使用框架的优势是代码兼容性较好，扩展能力好，抽象层次高，开发效率高，不需要太多经验也可以上手；缺点是业务简单时框架里的大部分代码是无用的，框架掩盖了一些问题的本质，对新手积累经验无益。

1.你以前写项目时，遇到哪些比较难解决的问题？

2.你以前写移动端项目时，有没有遇到兼容性问题？你怎么解决的

3.word-wrap:break-word

4.fullpage=>github

5.swiper轮播图

6.zepto.js 类似移动端jq框架，不兼容pc端，多了手势相关事件，基于模块化，是轻量级的库，jq所有功能都在一个js中,解决了300ms延时问题

7.移动端如何解决300ms延时问题？

1.fastclick.js

2.zepto.js

3.touch.js

8.js动画和c3动画区别

1.cs动画运行速度要比js快

2.cs动画兼容性没有js动画好，一般用于移动端

9.h5中的api有哪些？

1 地理定位geolocation

2 本地存储localStorage（永久性）

3 本地存储sessionStorage（浏览器关闭则存储数据消失）

4 touch

10.animate.css 动画库 <https://daneden.github.io/animate.css/>

11.fullpage 移动端

12.原型继承 闭包

13.网站播放视频

1 video

2 flash

3 iframe

4.jq插件

14 h5新增新特性

1.新增新标签

2 新增api

15.vue介绍

1.可以开发单页面应用 （运行速度快 混合app 微信里面 后台管理系统）

2.基于模块化（组件化）的开发模式

3.绑定数据 采用虚拟dom的方式 (运行速度快、操作数据方便)

4.关注视图 简单

16.localStorage 和cookie的区别

1.存储大小： cookie 4k localStorage 5M左右

2.用途：localStorage 一般用在移动端(保存用户信息 、缓存数据 、 购物车 、收藏)

3.过期时间：cookie 有过期时间 localStorage没有过期时间 永久保存

4.服务器是否设置：cookies可以在服务器端语言设置（nodejs、java，php）， localStorage只 能在浏览器设置

17.如何学习一门新技术？

网上查找官方文档及相关视频

18.全选 反选

19.你的优势是什么

20.axios 请求数据 / vue.resouce

21.XML 和 JSON的区别

json:占用带宽小

22.你想对我们公司有什么想要了解的

1.目前你们公司有多少技术人员？

2.我入职后主要工作是什么？

3.前后台如何配合开发

23.状态码304含义

从缓存区获取数据成功

302含义

临时重定向

24.怎么清楚浏览器缓存

ctrl+f5强制刷新 或 在设置里面清除

25.移动端上拉分页加载更多

26.webpack

27.scoped

避免样式冲突，表示css样式只在当前页面有效

28.单文件组件的使用

1.cd 到要创建项目的目录

2.vue init webpack-simple vuedemo01

3.cd vuedemo01

4.运行 npm install安装依赖

5.运行npm run dev

6、编码

29.ref关键字

C#中关键字通过引用传递参数

30. event.srcElement

获取触发事件的元素

31.如何提取网站api接口？

32.应用fetch或axios 获取数据

看官方文档

33.h5什么时候规范的

2014年10月

34.微信小程序

2016年10月发布测试版

35.vue在2016年5月左右发布2.0

36js中eval函数的作用

37.原生js上啦分页加载更多

原理：浏览器高度+滚动条下拉的高度>最后一条数据距离顶部的高度，则加载更多

38.ready()用法 函数规定当 ready 事件发生时执行的代码。函数仅能用于当前文档，因此无需选择

器。

语法1 $(document).ready(*function*)

语法2 $().ready(*function*)

语法3 $(*function*)

39.document.write（）于innerHTML的区别

document.write（） 重新渲染新的页面，

例document.write('<p>hello document</p>')

innerHTML 渲染局部页面

40.什么是伪数组？

a.具有length属性，必须是number类型，且number不为0

b.按索引方式储存数据；必须是个对象

c.不具有数组的push,pop等方法；

41.模块化开发思想:指将页面根据内容的关联性分解成不同的且相互独立的模块进行开发， 每个模块之间没有必然的联系，互不影响。

优势:提高重用性，提高开发效率，降低维护成本，提升代码质量等方面

common.js模块化:把一些公共的功能单独抽离成一个js文件,作为一个模块,使用这个模块时必须在里面通过module.exports或exports将数据暴露,在外面通过require或import引入

42.axios使用方法及属性

先安装:npm install axios

引入:**import** axios from 'axios'

axios.get('/user?ID=12345')

  .then(function (response) {

    console.log(response);

  })

  .catch(function (error) {

    console.log(error);

  });

43.bootstrap

目前使用较广的是版本2和3，其中2的最新版本的是2.3.2，3的最新版本是3.3.7。

使用前先下载bootstrap文件,开发模式/生产模式

以前开发项目是用的比较多的是css,只要引入bootstrap的css文件,然后看文档使用

44.手机端click事件300ms延迟问题 问题由来:双击事件

假定这么一个场景。用户在iOS Safari 里边点击了一个链接。由于用户可以进行双击缩放或者双击滚动的操作，当用户一次点击屏幕之后，浏览器并不能立刻判断用户是确实要打开这个链接，还是想要进行双击操作。因此，iOS Safari 就等待 300 毫秒，以判断用户是否再次点击了屏幕。于是就出现了300ms的延迟事件。

  解决方法:1.fastclick.js

FastClick的使用方法非常简单，在 window load 事件之后，在body上调用FastClick.attach()即可。

window.addEventListener(function(){

    FastClick.attach( document.body );

},false );

2.zepto.js的tap(touchstart/touchend)事件

3给点击事件添加setTimeout(function(),320)

45.面向对象编程思想

面向过程：分析出解决问题所需步骤，用函数一一实现，使用时一一调用就行

面向对象：把构成问题的事物分解成各个对象，每个对象都有自己独立的熟悉和方法，对象可以将整个事物问题进行分工，不同对象做不同的事，这种编程思想更贴近实际生活，所以被计算机语言广泛应用。

类概念：类是对象的抽象，对象是类的具体，对象必须由类来创建，同一个类可以创造出任意多个具有相同属性和方法的对象。

面向对象基本特征：

封装：将属性和方法封装在一起

继承：子类继承父类的属性及方法

原型链继承，构造函数继承，组合继承

多态：不同类的对象对同一消息做出不同反应.我用不同牌子的手机, 收到电话的时候铃声是不同的.

46.fullpage.js

是一个基于jQuery插件,他能够很方便很轻松的构建全屏网站

使用方法:引入<link rel="stylesheet" href="css/jquery.fullPage.css">

<script src="js/jquery.fullPage.js"></script>

HTML:<div id="fullpage">

    <div class="section">第一屏</div>

    <div class="section">第二屏</div>

    <div class="section">

        <div class="slide">第三屏的第一屏</div>

        <div class="slide">第三屏的第二屏</div>

        <div class="slide">第三屏的第三屏</div>

        <div class="slide">第三屏的第四屏</div>

    </div>

    <div class="section">第四屏</div>

</div>

js逻辑:$(function(){

    $('#fullpage').fullpage({

配置相对性参数

});

});

47.正向代理与反向代理

正向代理 代理的对象是客户端,是一个位于客户端和原始服务器(origin server)之间的服务器，为了从原始服务器取得内容，客户端向代理发送一个请求并指定目标(原始服务器)，然后代理向原始服务器转交请求并将获得的内容返回给客户端。,即利用自己的后台服务去请求远程的数据,正向代理过程中隐藏了真实的客户端。

反向代理:代理的对象是服务端,设立一个专门接收和发送数据的服务器,而真正的服务器只需要和代理服务器进行交互,由代理服务器响应客户端,反向代理的实现需要有一个负载均衡设备来分发用户请求,将用户请求分发到空闲的服务器上,反向代理过程中隐藏了真实的服务端

48**AMD与CMD**

**AMD**即Asynchronous Module Definition，中文名是“异步模块定义”的意思.是 RequireJS 在推广过程中对模块定义的规范化的产出。都是模块加载器，倡导模块化开发理念

RequireJS解决了两个问题

（1）实现js文件的异步加载，避免网页失去响应； （2）管理模块之间的依赖性，便于代码的编写和维护。

49.W3C标准

英文名:World Wide Web Consortium 中文名:万维网联盟 创建于1994年

网页主要组成部分：结构（Structure）、表现（Presentation）和行为（Behavior）。

结构化标准语言:主要包括[XHTML](https://baike.baidu.com/item/XHTML)和[XML](https://baike.baidu.com/item/XML)

表现标准语言:主要包括CSS

行为标准:主要包括对象模型（如W3C DOM）、ECMA[Script](https://baike.baidu.com/item/Script)等

50.浏览器兼容性处理

Html兼容性:对于有的浏览器不支持HTML5的一些标签我们可以使用[html5shiv.js](http://apps.bdimg.com/libs/html5shiv/3.7/html5shiv.min.js)来进行处理，html5shiv 内核是 IE Trident;FF 是 Gecko;opera 是 presto;safari and chrome 是 webkit

css兼容性:对于特定浏览器支持的样式我们有时要加-ms- -webkit- -moz- -o-进行兼容处理 ,注意的是IE10开始才支持animation ,IE9开始支持透明度opacity ,IE的透明效果,可以设置属性filter:alpha(opacity=0-100) ,ff透明加 -moz-opacity:0.10;

常见问题兼容:IE的浮动margin变成双倍问题 解决添加display:inline;

javaScript兼容:

获得事件:e=e||event

获得事件源:e.target||e.srcElement

阻止默认行为:e.preventDefault || e.returnValue=false

阻止事件冒泡:e.stopPropagation || e.cancelBubble

IE8-不支持event事件:e = e||window.event;

document.documentElement.scrollTop||document.body.scrollTop兼容性

51.懒加载

原理:懒加载的原理就是先在页面中把所有的图片统一使用一张占位图进行占位，把正真的路径存在元素的“data-url”（这个名字起个自己认识好记的就行）属性里，要用的时候就取出来，再设置；

优点:**页面加载速度快、可以减轻服务器的压力、节约了流量,用户体验好**

懒加载的主要目的是作为服务器前端的优化，减少请求数或延迟请求数。

**52预加载**

**提前加载图片，当用户需要查看时可直接从本地缓存中渲染**

预加载可以说是牺牲服务器前端性能，换取更好的用户体验，这样可以使用户的操作得到最快的反映。

方法一：用CSS和JavaScript实现预加载

方法二：仅使用JavaScript实现预加载

方法三：使用Ajax实现预加载

53懒加载与预加载的区别

两种技术的本质：两者的行为是相反的，一个是提前加载，一个是迟缓甚至不加载。懒加载对服务器前端有一定的缓解压力作用，预加载则会增加服务器前端压力。

54React生命周期钩子

组件创建的过程：

constructor() 创建组件：读取内部属性和方法

componentWillMount() 组件将要渲染

render() 渲染函数，通过这个函数，得到组件的dom结构

componentDidMount() 渲染完成

-------------------------------------

组件更新的过程：（state, props 发生变化都会执行更新）

componentWillReceiveProps() 组件将要接收新的外部属性

shouldComponentUpdate() 组件是否应该更新dom 需要return true/false

return true:才会执行下面三个方法

componentWillUpdate() 将要更新

render() 正在渲染

componentDidUpdate() 已经更新

组件销毁：

componentWillUnmount()

接收到了子组件的错误信息

componentDidCatch()

组件方法：

setState({}, ()=>{}) 修改state

forceUpdate() 强制执行更新的生命周期方法

组件内部属性：通过this访问

props

state

组件的类属性 通过类名访问

defaultProps

displayName

55 Webpack

*webpack* 是一个现代 JavaScript 应用程序的*静态模块打包器(module bundler)*。当 webpack 处理应用程序时，它会递归地构建一个*依赖关系图(dependency graph)*，其中包含应用程序需要的每个模块，然后将所有这些模块打包成一个或多个 *bundle*。

四个核心概念:

* + 入口(entry)
  + 输出(output)
  + loader
  + 插件(plugins

loader 可以将所有类型的文件转换为 webpack 能够处理的有效[模块](https://doc.webpack-china.org/concepts/modules)，然后你就可以利用 webpack 的打包能力，对它们进行处理。

56 Axios请求的特点

1简单易用，api接近于jquery，比原生的fetch之类的简单

2浏览器兼容性好，都能兼容IE7，使用fetch就得自己处理兼容

3通用性好，能在node和浏览器中使用，api一致

4稳定大牌，vue官网文档有推荐

57 Require.js作用

1.实现js文件异步加载,避免网页失去响应.

2.管理模块之间依赖性,便于代码的编写与维护.