**自我介绍**

中国（上海）自由贸易试验区创新西路778号17幢108室

您好 我叫刘红 我在上一家公司主要负责web前端开发以及移动端开发，工作职责是：跟产品经理确定需求，根据ui设计稿百分百还原页面的搭建，与后端开发人员保持良好的沟通，我个人喜欢与人沟通交流，因为我觉得有效的沟通可以提高项目的速度和质量，也有利于整个团队的管理！

我最近做的一个项目是关于教育类的一个后台管理系统，上家公司是属于外包形的，主要接一些刚初创公司的业务，接的这个项目，甲方主要是想投入最少的财力达到最优的效果，而且只做互联网型的，不做那种学习机构实体的那种，确实，现在在互联网上做这种教育课程是非常火也是最流行的趋势，然后我们前期就是深入了解客户的需求，查一些资料完成页面的骨架的搭建，先做的后台管理系统，这里面也分为了多个模块，一些公共的课程，人工智能的课程的展示， 用户在线留言等，还有一些线上的简历等多个模块，但这些模块都是通用的，只是某些功能要划分不一样的档位，比如说普通的课程就是一些分类：如公共课，基础课，进阶课程等等，总体来说甲方公司就想做一个集中的项目，在这个项目中遇到的难点就是对视频的处理，要像之前直接就上传视频了，但这里需要做一些优化 兼容 也查找了很多资料，最终把项目处理好了，

**html5 有哪些新特性、HTML5 新标签的浏览器兼容问题如何处理?**

答、html5 新增了以下的几大类元素：

表单控件：calendar、date、time、email、url、search。

控件元素：webworker, websockt, Geolocation。

移出的元素有下列这些： 显现层元素：basefont，big，center，font, s，strike，tt，u。

性能较差元素：frame，frameset，noframes。

HTML5 已形成了最终的标准，概括来讲，它主要是关于图像，位置，存储，多任务等功能的增加

新增的元素有绘画 canvas ，用于媒介回放的 video 和 audio 元素，本地离线存储 localStorage 长期存储数据，浏览器关闭后数据不丢失，而 sessionStorage 的数据在浏 览器关闭后自动删除。 新的技术：canvas,svg,webworker, websocket, Geolocation

**CSS3 新特性有哪些？**

1. 颜色：新增 RGBA，HSLA 模式

2. 文字阴影（text-shadow）边框阴影： box-shadow

3.圆角（border-radius）

4. 盒子模型：box-sizing

5.背景：background-size 设置背景图片的尺寸 background-origin 设置背景图片的原点 background-clip 设置背景图片的裁切区域，以”，”分隔可以设置多背景，用于 自适应布局

6.渐变：linear-gradient、radial-gradient

7. 过渡：transition，可实现动画

8. 自定义动画

9. 在 CSS3 中唯一引入的伪元素是 ：selection.

10. 媒体查询，多栏布局

11. border-image

12.2D 3D转换：transform：translate(x，y) rotate(x，y) skew(x，y) scale(x，y)

**浏览器移动设备兼容问题：**

**Css兼容**

1. 不同浏览器的标签默认的外补丁和内补丁不同，设置margin:0;padding:0;
2. 块属性标签float后，又有横行的margin情况下，在ie6显示margin比设置的大，在float的标签样式控制中加入 display:inline;将其转化为行内属性
3. 设置较小高度标签（一般小于10px），在ie6，ie7，遨游中高度超出自己设置高度，给超出高度的标签设置overflow:hidden
4. 图片默认有间距，使用float属性为img布局
5. Png24位图片在ie6上显示会出现背景，需要把图片改为png8



**ES6：**

1. **let 和 const 命令**
2. **模板字符串**
3. **字符串扩展了新的API：includes()，startsWith()，endsWith()**

****includes()：****返回布尔值，表示是否找到了参数字符串；

****startsWith()：****返回布尔值，表示参数字符串是否在查找字符串的头部；

****endsWith()：****返回布尔值，表示参数字符串是否在查找字符串的尾部。

1. **扩展运算符（扩展运算符是三个点(…)， 将一个数组转为用逗号分隔的参数序列）**
2. **箭头函数**
3. **解构赋值（解构表达式，数组解构，对象解构）**
4. **函数优化（可以设置参数默认值）**
5. **Map（接收一个函数，将原数组中的所有元素用这个函数处理后放入新数组返回。）和reduce（接收一个函数（必须）和一个初始值（可选），该函数接收两个参数）set（Set中只能保存不同元素，如果元素相同会被忽略）**
6. **Promise，我们可以通过Promise的构造函数来创建Promise对象，并在内部封装一个异步执行的结果。**
7. **Class（类）**
8. **Generator函数（一种异步编程解决方案）**
9. **For...of循环**
10. **修饰器(Decorator)**
11. **Modules模块会使用import from来导入**

**过渡.动画transition+ transform+ animation**

**1.** transform（变形）可以定义五个子属性：

移动translate  
缩放scale  
旋转rotate  
扭曲skew  
矩阵变形matrix元素

**2.** translate（移动）——transform的一个属性值，定义元素的移动效果

translate(x,y)两坐标同时移动，可以使用百分比  
translateX(x)单方向移动  
translate3d(x,y,z)定义 3D 移动

**3.** transition（过渡）——定义某种属性的过渡效果，是一个简写属性，可以定义四个子属性：

transition-property    规定设置过渡效果的 CSS 属性的名称。

transition-duration    规定完成过渡效果需要多少秒或毫秒。

transition-timing-function    规定速度效果的速度曲线。

transition-delay    定义过渡效果何时开始。

**4.** animation（动画) animation 定义某种keyframe的动画效果，属性是一个简写属性，用于设置六个动画子属性：

animation-name    规定需要绑定到选择器的 keyframe 名称

animation-duration    规定完成动画所花费的时间，以秒或毫秒计。

animation-timing-function    规定动画的速度曲线。

animation-delay    规定在动画开始之前的延迟。

animation-iteration-count    规定动画应该播放的次数。

animation-direction    规定是否应该轮流反向播放动画。

**跨域：**

️在发送请求的时候，只要协议、域名、端口 任意一个不相同 就是跨域，也称为不同源

常用的解决方案有两种

jsonp使用要与后端积极沟通和配合，后端写好jsonp接口文档代码，我们前端只需要在dataType上写上jsonp，这样就可以实现跨域

jsonp原理

这个和Ajax一点关系也没有！本质就是动态创建了一个script标签添加到页面的顶部，通过script标签的src属性发送请求，src里要设置接口的地址，发送的数据，最重要的是callback，当请求发送给服务器之后，服务器会返回一个函数的调用，内容返回到浏览器之后会被解析为js，调用定义好的函数，传入一个参数，请求成功之后会被自动移除！

优点

兼容性超级好！在更加古老的浏览器中都可以运行

缺点

不支持post请求，数据大的话搞不定，文件上传搞不定

CORS使用

CORS是HYML5中推出的新标准，低版本浏览器不支持ie前端使用起来就很简单，我们前端什么都不用做，只需要后端工程师设置允许跨域：在express中：响应数据之前，设置一个允许的header（response.header('Access-Control-Allow-Origin', '\*')）

CORS原理

浏览器能够识别这个header，请求发给服务器之后，服务器返回的响应头中有一个允许的标记，浏览器就会认为服务器允许跨域访问，就没有了跨域的错误

优点

get和post都支持

前端什么事都不需要做

缺点

兼容性比jsonp稍差一些……

**this的指向**

️普通函数的this指向全局对象window

️构造函数的this指向实例对象（new会改变this的指向）

️作为对象的方法被调用，谁调用this就指向谁

️es6箭头函数中，this指向创建者，并非调用者

️通过call apply bind 可以改变this的指向

**移动web适配问题**

**采用百分比布局的方式**，把盒子的宽度设置成百分比来根据屏幕的宽度来进行伸缩，不受固定像素的控制，内容向两侧填充，这样的布局可以适配移动端不同分辨率设备

**响应式开发**

Flex布局

通过css3中的媒介查询，使用rem作为单位，由于响应式开发有些繁琐，一般会采用第三方框架bootstrap来实现

**作用域**

**全局作用域：**

最外层函数定义的变量拥有全局作用域，函数的内部，都是可以访问的

**局部作用域：**

局部作用域只有函数内部的子函数可以访问，函数外部是访问不到的

非严格模式下不声明直接使用变量，变量会成为全局变量

**与解析：**

就是把声明的变量提升 把声明的函数提升，然后依次往下执行

**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。

**ajax 是什么?ajax 的交互模型?同步和异步的区别?如何解决跨域问题?**

通过异步模式，提升了用户体验

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

Ajax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。

**\*\* 简述一下Sass、Less，且说明区别？\*\***

他们是动态的样式语言，是CSS预处理器,CSS上的一种抽象层。他们是一种特殊的语法/语言而编译成CSS。

变量符不一样，less是@，而Sass是$;

Sass支持条件语句，可以使用if{}else{},for{}循环等等。而Less不支持;

Sass是基于Ruby的，是在服务端处理的，而Less是需要引入less.js来处理Less代码输出Css到浏览器

**null就是空的意思，undefined就是未定义**



当不考虑这两者的字符类型的时候，null==undefined是为true的

-但是如果考虑字符类型的时候，null的字符类型是Object，而undefined的字符类型是undefined

-如果你用typeof undefined==typeof null，那么返回结果则为false

-总结就是当不考虑字符类型时发现undefined等于null 当考虑字符类型时undefined不等于空

**前端缓存类型：**

**缓存优势：**

提高访问速度

节省带宽

减轻服务器压力

增强用户体验

**缓存类型：**

**服务端缓存：**数据库缓存：将查询的数据放到内存；CDN缓存(静态资源缓存)​​​​；代理服务器缓存。

**浏览器缓存：**浏览器都实现了HTTP缓存；cookie，web storage

**html5缓存：**可配置需要缓存的资源；网络无连接应用仍可用

**浏览器缓存的类型：**强缓存和协商缓存。

http header 命中强缓存，浏览器直接获取缓存资源；命中协商缓存，服务器返回304，浏览器获取本地缓存资源

强缓存是不需要发送HTTP请求到服务器，直接从本地磁盘获取缓存过的资源

协商缓存需要由服务器来确定客户端缓存资源是否可用

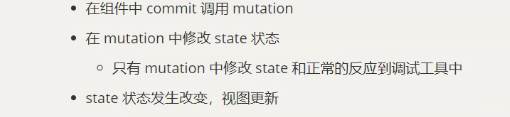
**Vue动态路由传参**

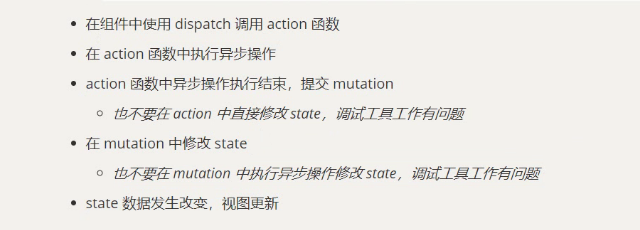
<router-link v-bind:to="/foo">登幽州台歌</router-link>

1. routers:[
2. *//动态路径参数 以冒号开头*
3. path:'/path/:id'
4. component:User
5. const User = {
6. *//路由组件中通过$route.params.id获取路由参数*
7. template:'<div>User{{$route.params.id}}</div>'

**vuex五个核心属性**

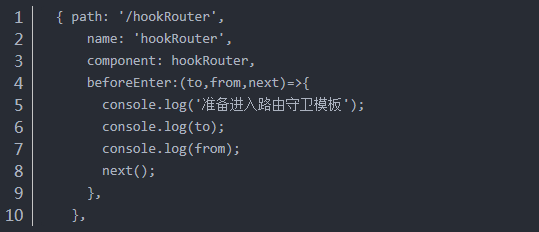
state => 基本数据  
getter => 相当与计算属性  
mutation => 提交更改数据的方法，同步！

  
action => 像一个装饰器，包裹mutation，使之可以异步。

  
module => 模块化Vuex

## vue路由的钩子函数（路由守卫）

首页可以控制导航跳转，beforeEach，afterEach等，一般用于页面title的修改。一些需要登录才能调整页面的重定向功能。



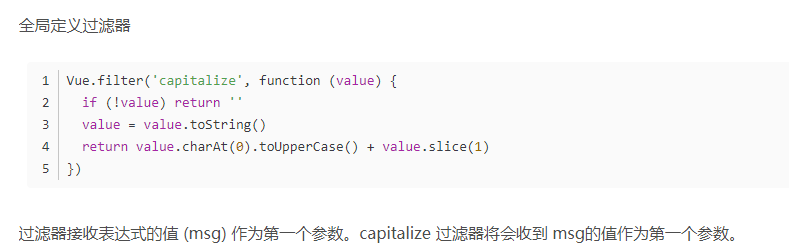
**beforeEach**主要有3个参数to，from，next：

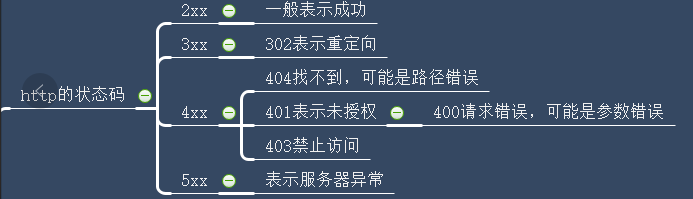
**to**：route即将进入的目标路由对象，

**from**：route当前导航正要离开的路由

**next**：function一定要调用该方法resolve这个钩子。执行效果依赖next方法的调用参数。可以控制网页的跳转。







**Bootstrap**

用来做响应式开发的一个框架,主要是媒体查询,根据页面分辨率的变化改变页面的布局.

**Webpack的配置**

创建 package.json 文件（保存安装包的依赖信息）

安装 webpack 到项目的开发依赖中

package.json中新增build:webpack

打包:npm run build

加载访问执行打包结果

(想自定义更多功能,项目根目录创建webpack.config.js写入配置信息)

打包npm run build

**如何实现单点登录**

以Cookie作为凭证媒介,登录父应用之后，应用返回一个加密的cookie，当用户访问子应用的时候，携带上这个cookie，授权应用解密cookie并进行校验，校验通过则登录当前用户

缺点:Cookie不安全,不能跨域实现免登

通过JSONP实现,可以解决跨域,用户在父应用中登录后,seccion匹配的cookie会存到客户端中,等用户需要在子应用中登录时,授权应用会访问父应用提供的jsonp接口,并在请求中携带父应用的cookie,验证登录状态,返回加密信息,子应用解析返回的加密信息验证用户,通过则登录成功.

缺点,这种方法也不安全,虽然能解决跨域,但是加密算法泄露的话攻击者可以在本地建立一个实现了登录接口的假冒父应用，通过绑定Host来把子应用发起的请求指向本地的假冒父应用，并作出回应

通过页面重定向的方式 通过父应用和子应用来回重定向中进行通讯,实现信息的安全传递,父应用提供一个GET方式的登录接口,如果用户没有登录则返回登录页面,输入账号密码进行登录,如果已经登录了,就生成加密Token,并且重定向到子应用提供的验证token的接口,通过解密和校验之后,子应用登录当前用户

这种方式相对于前俩种,解决了安全问题和跨域问题,但是没有前俩种方式方便,安全与方便本身就是矛盾

**Css选择器有哪些,那些属性可以继承**

通配选择器，标签选择器，类选择器，ID选择器，简单属性选择，具体属性选择，根据部分属性值选择，特定属性选择，从结构上来分还有后代选择器，子元素选择器，相邻兄弟选择器以及伪类。

不可继承的：display、margin、border、padding、background、height、min-height、max- height、width、min-width、max-width、overflow、position、left、right、top、 bottom、z-index、float、clear、table-layout、vertical-align、page-break-after、 page-bread-before和unicode-bidi

所有元素可继承：visibility和cursor

内联元素可继承：letter-spacing、word-spacing、white-space、line-height、color、font、 font-family、font-size、font-style、font-variant、font-weight、text- decoration、text-transform、direction

块状元素可继承：text-indent和text-align

列表元素可继承：list-style、list-style-type、list-style-position、list-style-image

表格元素可继承：border-collapse

**BOM和DOM**

DOM是Document ，简称文档对象模型。是用来获取或设置文档中标签的属性，例如获取或者设置input表单的value值。document.getElementById("").value;  这属于DOM

BOM是browser object model的缩写，简称浏览器对象模型。是用来获取或设置浏览器的属性、行为，例如：新建窗口、获取屏幕分辨率、浏览器版本号等。 比如 alert();弹出一个窗口,这属于BOM

**Let var const三者的区别**

**Var:**

**可以变量提升，没有块级作用域，变量可以重复声明**

**let**

let ：在块级作用域有效，在该块级作用域之外将不能被访问。

let ：不存在变量提升

let ：声明的变量不能重复声明，不然会报错。

es6强制开始了严格模式，变量未声明不能使用，否则就会直接报错。

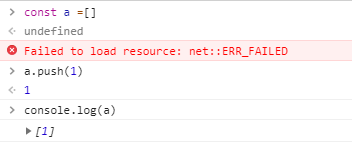
**const**

const ：声明常量，在程序运行中不能直接被修改，不然会报错。

const ：声明的时候一定要赋值，不然会报错。

const ：作用域和let一样。

复杂数据类型，比如数组 对象就可以修改



**element-ui中经常使用的form元素组件有那些**

el-form

el-form-item

el-input

el-select

el-option

el-col

el-switch

el-checkbox

el-radio

**经常用的http方法有那些,并说明使用场景**

超文本传输协议，是一种实现客户端和服务器之间通信的响应协议，它是用作客户端和服务器之间的请求

1、GET方法

GET方法用于使用给定的URI从给定服务器中检索信息，即从指定资源中请求数据。使用GET方法的请求应该只是检索数据，并且不应对数据产生其他影响。

在GET请求的URL中发送查询字符串（名称/值对），需要这样写：

说明：

GET请求是可以缓存的，我们可以从浏览器历史记录中查找到GET请求，还可以把它收藏到书签中；且GET请求有长度限制，仅用于请求数据（不修改）。

注：因GET请求的不安全性，在处理敏感数据时，绝不可以使用GET请求。

2、POST方法

POST方法用于将数据发送到服务器以创建或更新资源，它要求服务器确认请求中包含的内容作为由URI区分的Web资源的另一个下属。

POST请求永远不会被缓存，且对数据长度没有限制；我们无法从浏览器历史记录中查找到POST请求。

3、HEAD方法

HEAD方法与GET方法相同，但没有响应体，仅传输状态行和标题部分。这对于恢复相应头部编写的元数据非常有用，而无需传输整个内容。

4、PUT方法

PUT方法用于将数据发送到服务器以创建或更新资源，它可以用上传的内容替换目标资源中的所有当前内容。

它会将包含的元素放在所提供的URI下，如果URI指示的是当前资源，则会被改变。如果URI未指示当前资源，则服务器可以使用该URI创建资源。

5、DELETE方法

DELETE方法用来删除指定的资源，它会删除URI给出的目标资源的所有当前内容。

6、CONNECT方法

CONNECT方法用来建立到给定URI标识的服务器的隧道；它通过简单的TCP / IP隧道更改请求连接，通常实使用解码的HTTP代理来进行SSL编码的通信（HTTPS）。

7、OPTIONS方法

OPTIONS方法用来描述了目标资源的通信选项，会返回服务器支持预定义URL的HTTP策略。

8、TRACE方法

TRACE方法用于沿着目标资源的路径执行消息环回测试；它回应收到的请求，以便客户可以看到中间服务器进行了哪些（假设任何）进度或增量。

**什么是盒子模型**

把所有的网页元素都看成一个盒子，它具有：

content，padding，border，margin

四个属性，这就是盒子模型

盒子模型有两种形式：标准盒子模型，怪异盒子模型

 两种模式可以利用box-sizing属性进行自行选择：

　　标准模式：box-sizing:content-box;

　　怪异模式：box-sizing:border-box;

两种模式的区别：

标准模式会被设置的padding撑开，而怪异模式则相当于将盒子的大小固定好，再将内容装入盒子。盒子的大小并不会被padding所撑开

**如何实现本地存储**

cookie存储数据，当用户访问了某个网站（网页）的时候，我们就可以通过cookie来向访问者电脑上存储数据，或者某些网站为了辨别用户身份、进行session跟踪而储存在用户本地终端上的数据（通常经过加密）

localStorage(本地存储)持久化的本地存储，除非主动删除数据，否则数据是永远不会过期的。

存

lacalStorage.setItem('username','zxy');

取

localStorage.getItem('username');

删除

localStorage.remove('username');

**jQuery中怎样添加,移动,复制DOM节点**

1.查找节点

$("xxx")

2.创建节点

$(selector)  //选择器

$(element)  //Dom元素

$(html)       //html代码

3.插入节点

内部插入:append(content)  appendTo(content)   prepend(content)   prependTo(content)

外部插入:after(content)   insertAfter(content)   before(content)   insertBefore(content)

4.删除节点

$(selector).remove([expr])

5.替换节点

$("ul li:eq(1)").replaceWith($xxx)

6.复制节点

$(selector).clone([includeEvents])

**jQuery中实现AJAX请求的方法有哪些**

1.load方法：通过 AJAX 请求从服务器加载数据，并把返回的数据放置到指定的元素中

2.get方法：从指定的资源请求数据 可能返回缓存数据

3.post方法：向指定的资源提交数据 不会缓存数据连同请求一起发送数据

**Get和post**

Get是从服务器上请求数据,post是发送数据到服务器上,post一般适用于发送保密的或者比较大量的数据到服务器.get会把数据附带到网址后面,然后一起送到服务器,因此传输的数据量就会受到限制,但是效率比post方法好

Post方式的安全性比get方式的安全性好,传输数据包含机密信息我一般用post数据提交方式.

数据查询时,用的是get方式,而在做数据添加修改或删除时建议用post

如果是向服务器传输数据(如帐号密码等)都是加密数据(post)，如果只是单单想要

从服务器获得数据或者传输的数据并不重要， 可以直接使用明文方式传输( get )

**如何理解vue中的key**

 key的作用是给予一个节点唯一的身份识别，有相同父元素的子元素必须有独特的 key 。这样它可以前后对比，算出哪些节点是要重复使用或者调整顺序。

**Vue中的watch怎样深度监听**

deep: true

deep的意思就是深入观察，监听器会一层层的往下遍历，给对象的所有属性都加上这个监听器，但是这样性能开销就会非常大了，任何修改obj里面任何一个属性都会触发这个监听器里的 handler。

优化，我们可以是使用字符串形式监听。

immediate: true,

// deep: true

**如何用DOM绑定事件**

1. 在html标签中直接绑定；

2. 在js中获取到相应的dom元素后绑定；

3. 在js中使用addEventListener（）实现绑定；

**Axios的配置**

请求拦截器和相应拦截器,就是看登录时候看看携带判断一下有没有tooken,有的话就跳转到下一页,没有就跳转到登录页

拦截器,截取请求或响应再被then或catch处理之前.....请求拦截器:请求发起之前做的事,request 相应拦截器:相应回来之后做的事

**移动端的适配**

根据不同屏幕动态写入font-size和viewport，以rem作为宽度单位

将屏幕分为固定的块数10

**vue-router的传参方式方式**

vue-router路由传参分为两种形式：params和query。

**1、params传参**

（1）传递参数，用$router，代码如下：

const routerParams = {

name: 'iResultManage',

params: {page: '1', code: '8989'}

};

this.$router.push(routerParams);

(2)接收参数，用$route，代码如下：

const params = this.$route.params

**2、query传参**

（1）传递参数，用$router，代码如下：

const routerParams = {

name: 'iResultManage',

query: {page: '1', code: '8989'}

};

this.$router.push(routerParams);

（2）接收参数，用$route，代码如下：

const params = this.$route.query

**注：用params传参，F5强制刷新参数会被清空，用query，由于参数适用路径传参的所以F5强制刷新也不会被清空**

Vue-cli版本 2.0,3.0,4.0

Vue版本 2.6.10

Webpack版本 4.41.2

**Vue首页加载慢 搜索引擎优化:**

不用客户端渲染,用服务端渲染,有利于seo.从客户端爬出来的内容是没有解析的字符串,从服务端爬到的内容是解析过的字符串,搜索引擎找的时可以跟着语义化标签去找你的网页,这时候就体现到了语义化标签的重要信.

**事件触发的几个阶段**

js事件的三个阶段分别为：捕获、目标、冒泡

1.捕获：事件由页面元素接收，逐级向下，到具体的元素

2.目标：具体的元素本身

3.冒泡：跟捕获相反，具体元素本身，逐级向上，到页面元素

IE5.5：div---body---document

IE6.0: div---body---html---document

Mozilla:div---body---html---document---window

事件捕获：当使用事件捕获时，父级元素先触发，子元素后触发

事件冒泡：当使用事件冒泡时，子级元素先触发，父元素后触发

W3C ： 任何事件发生时，先从顶层开始进行事件捕获，直到事件触发到达事件源，再从事件源向上进行事件捕获

标准浏览器：addEventListener("click","doSomething","true")方法,若第三参数为true则采用事件捕获，若为false，则采用事件冒泡

IE浏览器只支持事件冒泡，不支持事件捕获，所以它不支持addEventListener("click","doSomething","true")方法,所以ie浏览器使用ele.attachEvent("onclick",doSomething)

事件传播的阻止方法：

再W3C中，使用stopPropagation()方法

在IE下使用cancelBubble = true方法

阻止默认行为：

再W3c中，使用preventDefault（）方法

再IE下return false

**Vuex刷新之后的数据怎么解决空值的状态**

lacalStorage监听页面是否刷新,如果有刷新状态的话,先把lacalStorage中的值存一下,然后刷新了在取出来

**轮询 长轮询,短轮询**

一个账号登录一个用户,登录多个的话自动退出,通过检测token,就是用轮询的这种方式,让这个页面登录以后一直后端去发请求,从而建立一个之间的连接,如果新的账户登录带的是当前的token的话,那这个连接就会断掉,断掉之后我们就会让他跳转到登录页,从而在新的设备上进行一个登录,这样的话就可以实现仅在一台设备上登录

**Git命令**

git init 初始化（新建一个文件夹）将其目录切换至目录上，执行此命令，即可初始化该目录为git

git status 查看状态

git add  filename  将filename  添加到git的暂存区

git commit 提交（不过一般提交不上去，因为需要填写提交的内容，这就是git的严谨之处 一般用下面的提交命令）

git commit -m "message" 提交，提交的信息为message

vi filename   修改filename文件

cat filename 查看filename内

wq! 退出窗口

git log filename 查看日志

git log --pretty=online filename  查看filename日志，并且一行显示

git diff 查看对比两次文件内容具体修改了什么。

git diff HEAD -- <filename> （“--”前后有空格），可以查看工作区和版本库里面最新版本的区别。

git reset  --hard HEAD^    回退一步

git reset  --hard HEAD^^^     回退三步

git reflog filename 查看详细的操作

git reset  --hard 42e005    将版本回退到42e005时的版本

git reset --hard HEAD~3     回退3步

git chechout --版本回撤

git rm -f filename  删除filename

git branch 查看分支

git branch dev    新建dev分支

git checkout dev      切换分支

git  merge dev  将dev分支和当前分支合并

git branch -d dev    删除分支

 git checkout -b test   新建test分支并切换

git remote add origin "http:,......"   远程添加

git push -u origin master   将本地的项目推送到master分支（第一次）

git push   将本地的项目推送到master分支

git pull origin master    从远程分支master上下载

git clone  "http:,......"    克隆远程分支

ssh-keygen -t rsa -C "你git注册的邮箱"        生成ssh-key

以上是git的一些常用的命令：但是之前遇到了一个这样的问题  就是从远程拉取项目的时候，报 SSL certificate problem: self signed certificate的错误，最后查了一下说是在windows上出现的频率还是挺高的，我估计主要是git本身就是基于linux开发的，在windows上，容易缺失一些环境。参考了一些文章，解决了

1.创建临时环境变量：

windows上命令行输入：

set GIT\_SSL\_NO\_VERIFY=true git clone

Linux下：

env GIT\_SSL\_NO\_VERIFY=true git push

这里clon可以根据需要换成其他的git命令。

也可以把临时环境变量变为永久的，反正永远不验证ssl证书也没什么风险吧。。。

2.用git自带的配置命令：

git config --global http.sslVerify false

# 解决vue项目首页加载过慢的情况

1. app.js文件过大导致的。。。
2. 如何来处理  
   vue-route 懒加载  
   component： resolve => require([''],resolve)

在webpack打包的过程中，将多余文件去掉，如map文件

即在config/index.js中将productionSourceMap的值修改为false，就可以在编译时不生成.map文件了

在项目开发中，我们会用到很多第三方库，如果可以按需引入，我们可以只引入自己需要的组件，来减少所占空间

**1.css只在当前组件起作用**  
答：在style标签中写入**scoped**即可 例如：<style scoped></style>

**2.v-if 和 v-show 区别**  
答：v-if 是通过判断后面条件是true还是false来进行dom渲染，true就进行dom渲染，是false就dom不渲染，v-show，后面条件无论是true还是false都进行dom渲染，只是首次加载比较慢，后期比较快的原因是因为控制display block或none来显示或隐藏的

**3.$route和$router的区别**  
答：$route是“路由信息对象”，包括path，params，hash，query，fullPath，matched，name等路由信息参数。而$router是“路由实例”对象包括了路由的跳转方法，钩子函数等。

**4.vue.js的两个核心是什么？**  
答：数据驱动、组件系统

**5.vue几种常用的指令**  
答：v-for 、 v-if 、v-bind、v-on、v-show、v-else

**6.vue常用的修饰符？**  
答：.prevent: 提交事件不再重载页面；.stop: 阻止单击事件冒泡；.self: 当事件发生在该元素本身而不是子元素的时候会触发；.capture: 事件侦听，事件发生的时候会调用

**7.v-on 可以绑定多个方法吗？**  
答：可以 <p v-on="{click:dbClick,mousemove:MouseClick}"></p>

一个事件绑定多个函数：

<p @click="one(),two()">点击</p>

**8.vue中 key 值的作用？**  
答：当 Vue.js 用 v-for 正在更新已渲染过的元素列表时，它默认用“就地复用”策略。如果数据项的顺序被改变，Vue 将不会移动 DOM 元素来匹配数据项的顺序， 而是简单复用此处每个元素，并且确保它在特定索引下显示已被渲染过的每个元素。key的作用主要是为了高效的更新虚拟DOM。

**9.什么是vue的计算属性？**  
答：在模板中放入太多的逻辑会让模板过重且难以维护，在需要对数据进行复杂处理，且可能多次使用的情况下，尽量采取计算属性的方式。好处：①使得数据处理结构清晰；②依赖于数据，数据更新，处理结果自动更新；③计算属性内部this指向vm实例；④在template调用时，直接写计算属性名即可；⑤常用的是getter方法，获取数据，也可以使用set方法改变数据；⑥相较于methods，不管依赖的数据变不变，methods都会重新计算，但是依赖数据不变的时候computed从缓存中获取，不会重新计算。

**10.vue等单页面应用及其优缺点**  
答：优点：Vue 的目标是通过尽可能简单的 API 实现响应的数据绑定和组合的视图组件，核心是一个响应的数据绑定系统。MVVM、数据驱动、组件化、轻量、简洁、高效、快速、模块友好。  
缺点：不支持低版本的浏览器，最低只支持到IE9；不利于SEO的优化（如果要支持SEO，建议通过服务端来进行渲染组件）；第一次加载首页耗时相对长一些；不可以使用浏览器的导航按钮需要自行实现前进、后退。

**Post和get区别**

**理论上说，GET是从服务器上请求数据，POST是发送数据到服务器**

从安全的角度来讲，get方法没有post方法安全

Get 方法通过 URL 请求来传递用户的数据，比如一个登陆页面，通过 Get 方式提交数据时，用户名和密码将出现在 URL上，就可以从历史记录获得该用户的帐号和密码，所以表单提交建议使用 Post 方法，

Post 是允许传输大量数据的方法，而 Get 方法会将所要传输的数据附在网址后面，然 后一起送达服务器，因此传送的数据量就会受到限制，但是执行效率却比 Post 方法好。 总结： 1、get 方式的安全性较 Post 方式要差些，包含机密信息的话，建议用 Post 数据提交 方式； 2、在做数据查询时，建议用 Get 方式；而在做数据添加、修改或删除时，建议用 Post 方式； 所以： 表达如果是向服务器传输数据(如帐号密码等)都是加密数据(post)，如果只是单单想要 从服务器获得数据或者传输的数据并不重要， 可以直接使用明文方式传输( get )

**Get 方式传输的数据量非常小，一般限制在 2 KB 左右**

**而 Post 方式传递的数据量相对较大**

# Vue是一套渐进式框架的理解

我理解的就是：如果想用vue就单纯的把vue引进来就可以用，用vuex就用vuex，，vue-router，vue-cli，所以就叫渐进式，不像是query要想使用就得全部引用，

渐进式代表的含义是：没有多做职责之外的事，vue.js只提供了vue-cli生态中最核心的组件系统和双向数据绑定，就好像 vuex、vue-router都属于围绕vue.js开发的库。

示例：

使用Angular，必须接受以下东西：

1、必须使用它的模块机制。

2、必须使用它的依赖注入。

3、必须使用它的特殊形式定义组件（这一点每个视图框架都有，这是难以避免的）

所以Angular是带有比较强的排它性的，如果你的应用不是从头开始，而是要不断考虑是否跟其他东西集成，这些主张会带来一些困扰。

使用React，你必须理解：

1、函数式编程的理念。

2、需要知道它的副作用。

3、什么是纯函数。

4、如何隔离、避免副作用。

5、它的侵入性看似没有Angular那么强，主要因为它是属于软性侵入的。

Vue与React、Angular的不同是，但它是渐进的：

1、可以在原有的大系统的上面，把一两个组件改用它实现，就是当成jQuery来使用。

2、可以整个用它全家桶开发，当Angular来使用。

3、可以用它的视图，搭配你自己设计的整个下层使用。

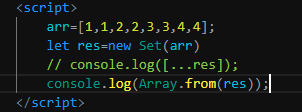
4、可以在底层数据逻辑的地方用OO和设计模式的那套理念。

5、可以函数式，它只是个轻量视图而已，只做了最核心的东西。

**Indexof数组去重**



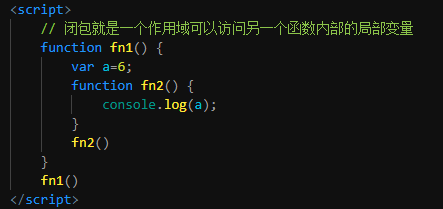
**Set方法去重**



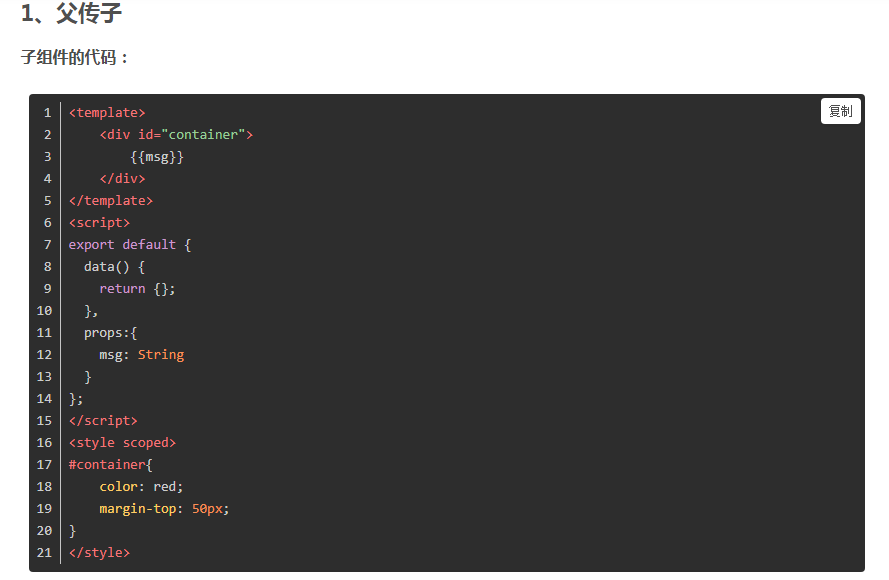
**闭包：**

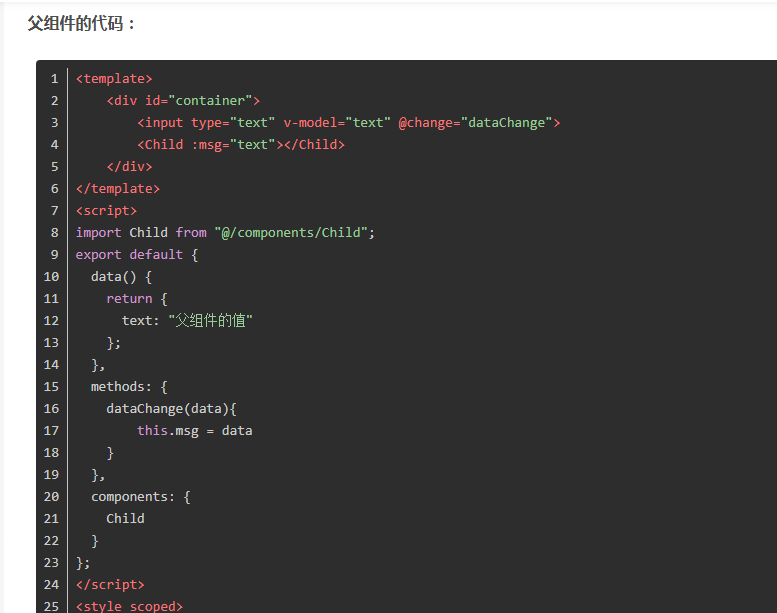
**优点：避免全局变量的污染**

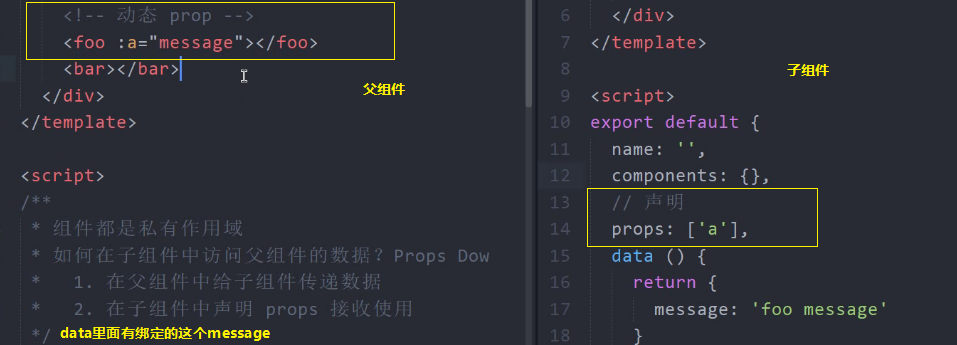
**缺点：会造成内存泄露 解决方案：在退出函数之前，将不使用的局部变量全部删除**



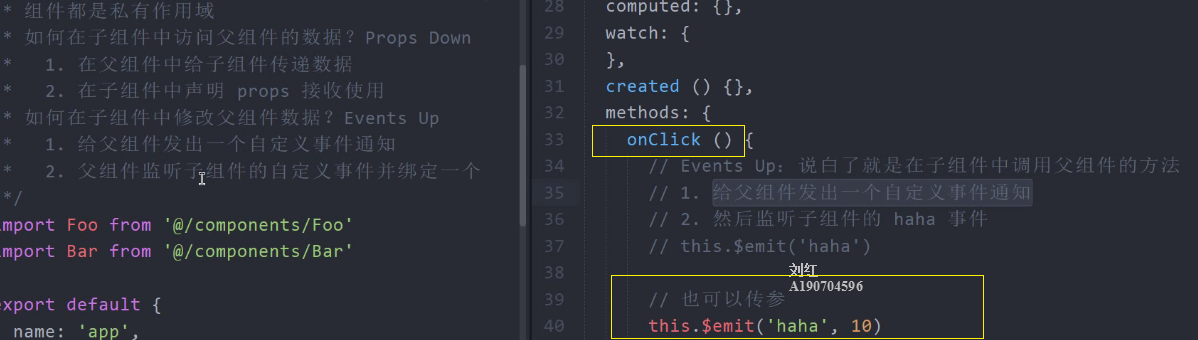
**组件传值**





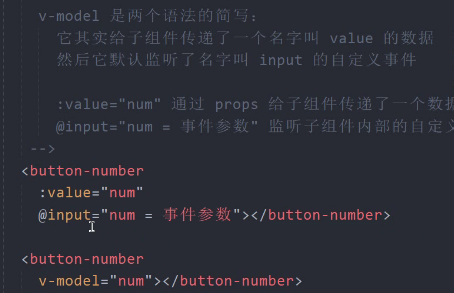


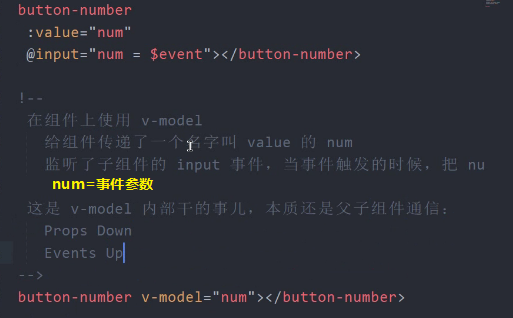
**父传子：1子组件通过props属性接收从父组件传过来的值。2传递的值一般都是动态的，所以父组件要动态的绑定传给子组件的数据；**



**父传子：1在子组件中通过this.$emit给父组件发出一个自定义事件的通知2父组件接收这个自定义事件，并绑定一个自己的处理函数，然后在自己的处理函数中就可以写逻辑，比如要是有参数传过来，就在处理函数中接收这个传过来的参数**

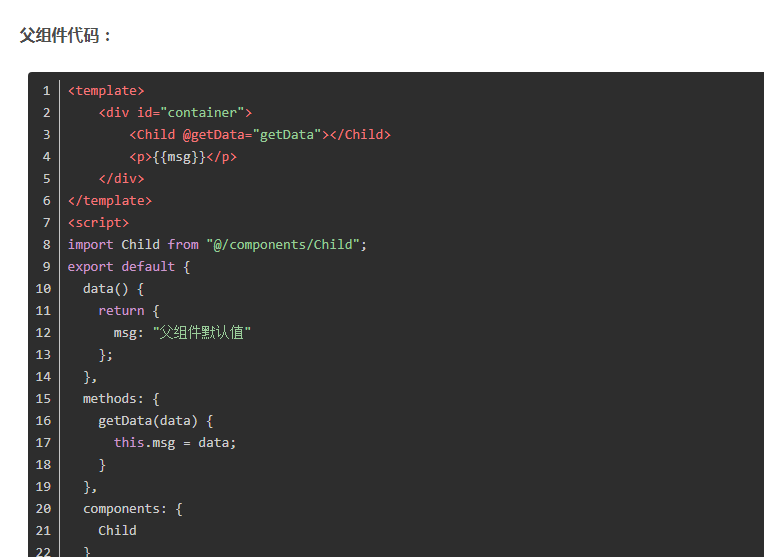
**v-model原理**

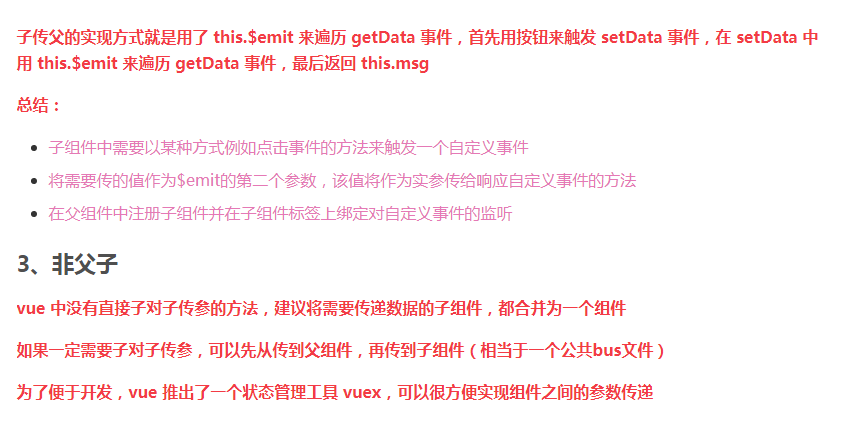












一.数组及数组的方法

1.数组的概念:

是值数据的集合,在JavaScript中数组中的数据类型可以是不同的.

2.数组的声明:

(1)var 数组的名称=new Array();

(2)var 数组的名称=[];

3.数组的初始化

(1)var 数组的名称=new Array(‘值1’,值2,‘值3’,…)//js 是一种弱类型的脚本语言索引可以存储不同的数据类型.

(2)var 数组的名称=[值1,值2,值3,…]

(3)var 数组的名称=new Array()

重点:通过遍历数组的方式来获取数组中的数据

4.操作数组数据的方式

(1)join():将数组成员通过一个分割符合并成字符串

<script type="text/javascript">

// (1)join():将数组成员通过一个分割符合并成字符串

var arr=[12,23,1,0,23,45];

var str=['L','O','V','E'];

// alert(str.join('-'));//输出结果为L-O-V-E

alert(arr.join('\*')); //输出结果为12\*23\*1\*0\*23\*45

(2)push()和pop(): 在数组的最后添加或是删除成员

(3)unshift()和shift(): 在数组的最前面添加或是删除成员

(4)reverse(): 表示为将数组的数据进行翻转

(5)indexOf()和lastIndexOf:这两个方法都是用来搜索整个数组中具有给定值的元素,返回找到的第一个元素的索引,如果没找到,则返回-1.

indexOf表示返回数组中元素第一次出现的索引值,

lastIndexOf则是反向搜索.

var num2=[1,2,13,1,2,2];

var index= num2.indexOf(2);//返回第一次出现2的索引

alert(index);//输出结果为1

例:第一个值表示要查询的数据;第二个值表示起始要查询的索引;查询不到返回-1

(6)splice():在数组中添加成员

例:

var num4=[1,2,3,4,5,6];

num4.splice(1,2,7,8,9);//第一个值1表示要开始添加成员的索引位置;第二个值2表示删除2两个成员(包含自身);第三个值7,8,9 表示 在此位置上添加 7,8,9

alert(index);//输出结果为1789456

(7)slice():从一个数组中选择(截取)元素

例:第一个值表示开始截取的位置索引;第二个值表示截取结束位置的索引,但是并不包含结束位置索引所对应的成员

(8)sort():返回排序后的数组.如果数组包含undefined,会被排序到数组的尾部.如果不带参数的用sort(),数组元素以字母表顺序排序.

(9)concat():创建并返回一个新的数组

(10).toString()和toLocaleString()

toLocaleString()方法是toString()的本地化版本.

(11).forEach()方法,从头至尾遍历数组,为每个元素调用指定的函数.

(12)map()方法:

和forEach()类似,调用数组的每个元素传递给指定函数,并返回一个数组,所以它和forEach()的区别在于,有一个返回值.不修改原数组,返回的数组长度和原数组相同.

(13)filter()方法，返回的是调用数组的一个子集。

(14)every()和some()

every()方法是只有数组中所以元素都满足某个条件才会返回true;some()方法是只要有满足条件的值,就返回true。

**Js中数组循环的方法**

forEach大多数情况都可以代替for循环

Some判断数组中是否包含指定条件的元素return一个布尔值,只要有一个满足条件,就停止执行,返回一个true,遍历结束都没有找到就返回false

every判断数组中的元素是否全部满足条件

filter 从数组中根据某个条件过滤数据得到一个新的数组,注意:过滤的结果还是一个数组

Find 根据条件查找某个元素,注意:会遍历找到第一个满足条件的元素,停止并立即返回

findIndex根据条件找查元素的索引

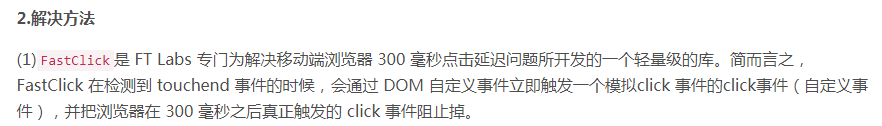
includes 判断数组是否包含指定元素 一般用于数组中都是普通数据类型,数字,字符串

移动端出现1px像素变粗的问题怎么解决





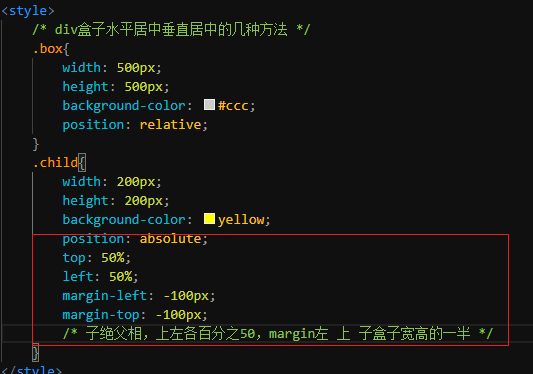


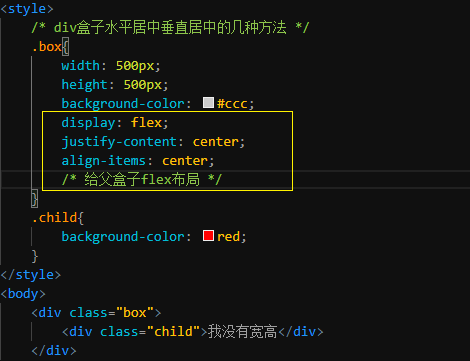




**让子盒子水平垂直居中的方法**

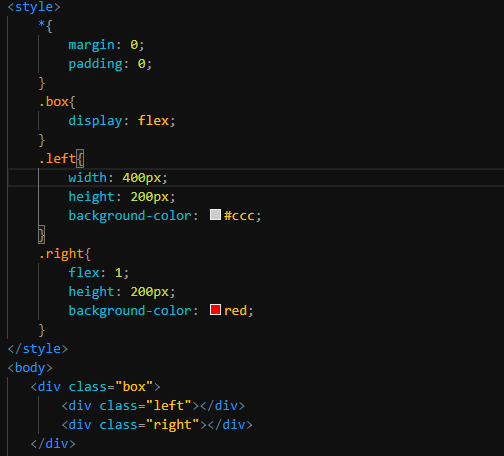








**左侧固定，右侧随意拉伸**



**Margin塌陷问题**

兄弟盒子：

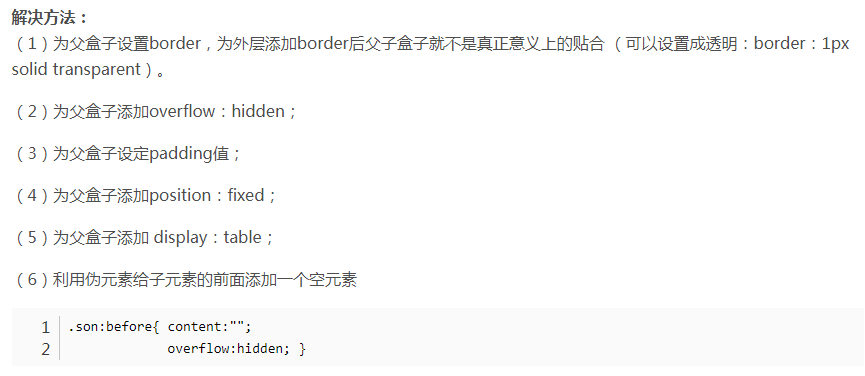
对box1我们为其设置margin-bottom：50px;  
对box2我们为其设置margin-top： 40px;

**两盒子之间的距离仅是50px**，也就是说两盒子之间的margin出现了重叠部分，故而我们可以得出：**垂直之间塌陷的原则是以两盒子最大的外边距为准。**

**父子盒子：**

当为子盒子添加margin-top：100px;时会发生如下情况

盒子会塌陷 子盒子跟父盒子距离顶部100px



**登录token过期问题**

我处理方式是，登录后生成一个token ,用户请求必须带着token，未过期，每次请求我都会去更新token的过期时间，生成一个新的Token返回给客户端，客户端需更新token的存储,之后客户端每次请求都带着最新token,只要用户在我设定的有效期内操作token就永不失效，若超过了我设定的过期时间未操作，就需要用户重新登录了

**路由拦截器（导航守卫）**

控制页面访问权限，router.beforeEach(to,from,next)通过判断token，有就next（）方法，放行，没有就强制跳到登录页

**前置钩子/后置钩子应用场景是“进度条”（第三方包nprogress）**

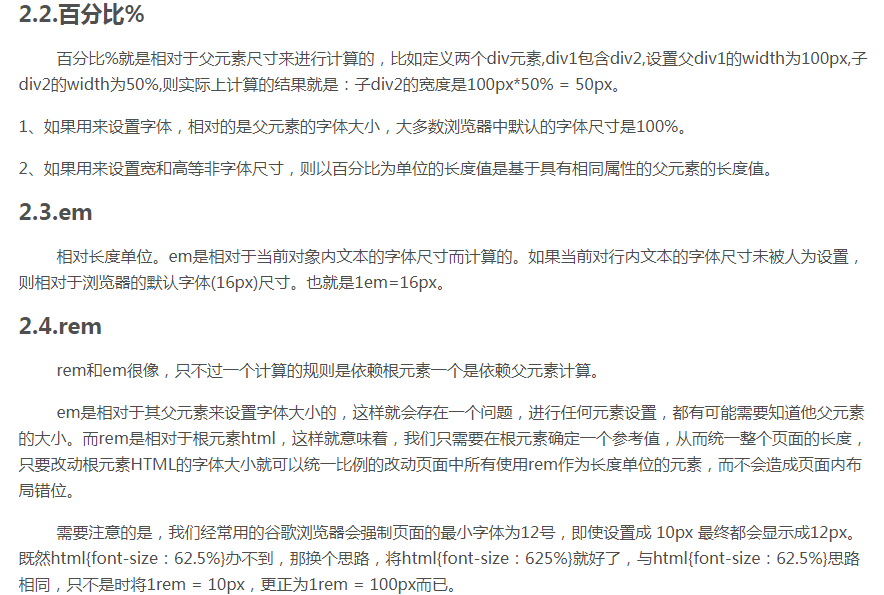
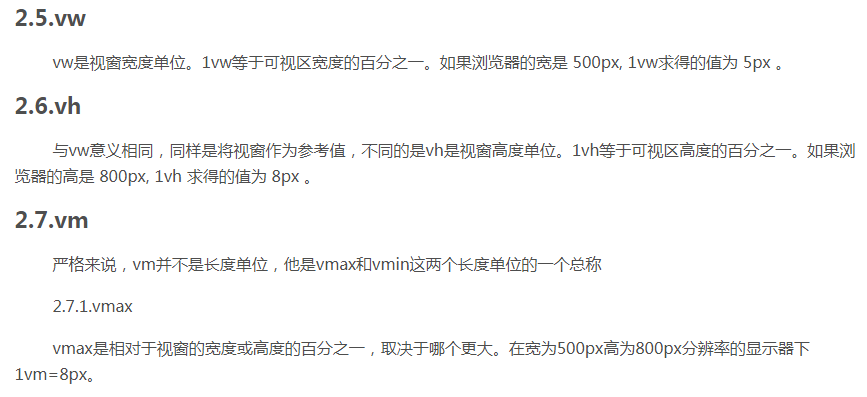
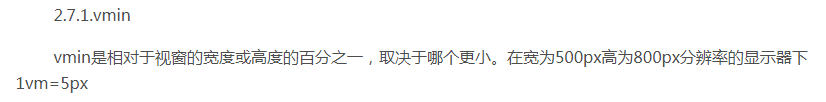
**如果在登录页并且是非登录状态访问非登录页面，这里要手动终止进度条，否则进度条会停止不前，Nprogress.done（）**

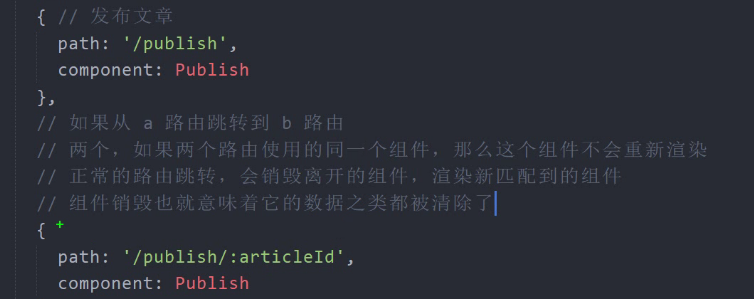
**组件表格默认只显示文本，如需显示图片等，需要自定义表格列**

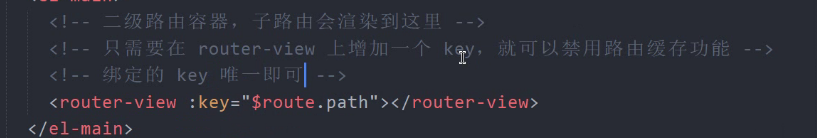


**Axios拦截器的作用，统一添加token**

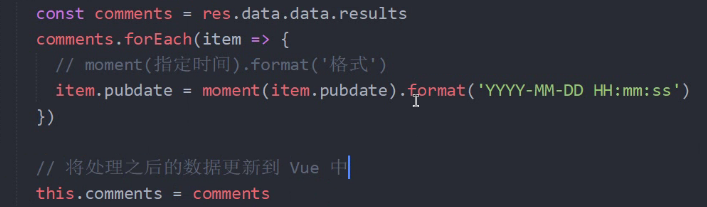


1. px是Pixel的缩写，表示像素，是绝对的长度单位，它是相对于显示器屏幕分辨率而言的
2. 
3. 
4. 
5. **禁用路由缓存**



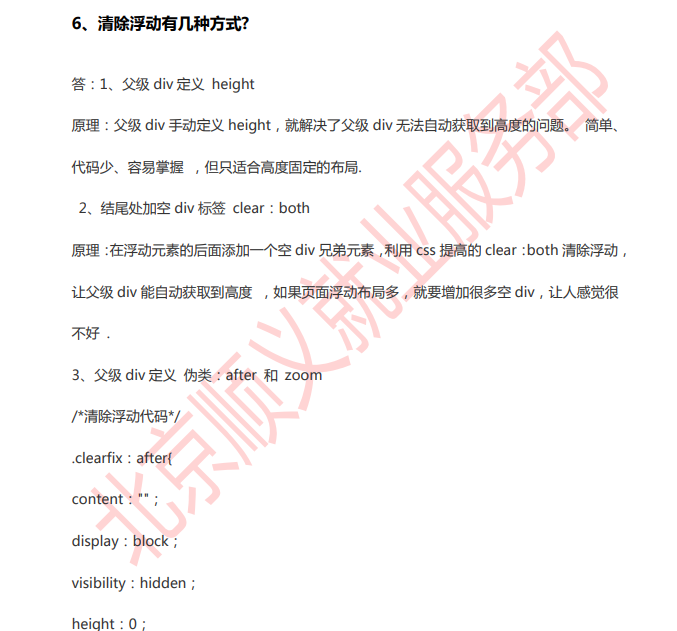


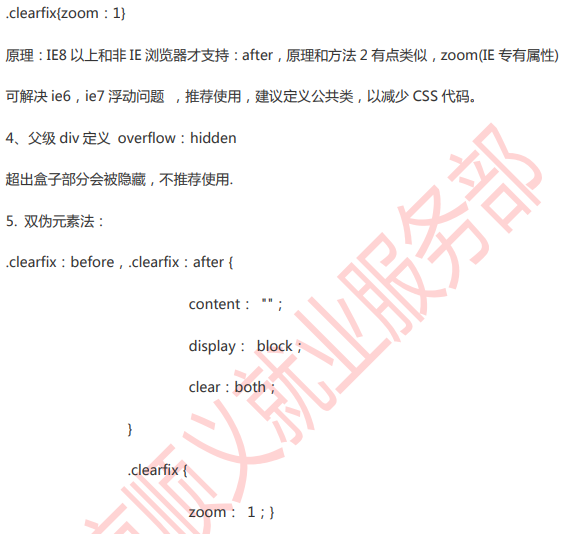
**更新时间格式用moment第三方包下载**



**全局过滤器**







# vue 路由跳转四种方式 (带参数)

