项目开发周期，是否个人完成，团队开发负责哪些

希望研发团队是个什么样的团队

除了前端工具外别的常用的工具，比如git

项目管理工具和区间管理的工具

之前做到的项目遇到的困难是什么

1. dom和bom具体是什么？

BOM是浏览器对象模型，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。

DOM是文档对象模型，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。

由于DOM的操作对象是文档（Document），所以dom和浏览器没有直接关系。

1. 原生js的模块化开发：https://www.xuebuyuan.com/3242406.html

有哪几种方式把代码模块化管理起来？

https://segmentfault.com/a/1190000000733959

比如es6的import和export

目前，通行的Javascript模块规范共有两种：CommonJS和AMD。

AMD是"Asynchronous Module Definition"的缩写，即"异步模块定义"。它采用异步方式加载模块，模块的加载不影响它后面语句的运行。所有依赖这个模块的语句，都定义在一个回调函数中，等到加载完成之后，这个回调函数才会运行。

目前，主要有两个Javascript库实现了AMD规范：require.js和curl.js。

RequireJS是一个非常小巧的JavaScript模块载入框架，是AMD规范最好的实现者之一。它可以和其他的框架协同工作，使用RequireJS必将使您的前端代码质量得以提升。

1. js的原型链概念， 原型链之间的继承？

例如：一个school类，一个preson类，创建第三个类要有school和preson类的方法

原型： 每创建一个函数，函数上都有一个属性为 prototype，它的值是一个对象。 这个对象的作用在于当使用函数创建实例的时候，那么这些实例都会共享原型上的属性和方法。

原型链： 在 JavaScript 中，每个对象都有一个指向它的原型（prototype）对象的内部链接（proto）。这个原型对象又有自己的原型，直到某个对象的原型为 null 为止（也就是不再有原型指向）。这种一级一级的链结构就称为原型链（prototype chain）。 当查找一个对象的属性时，JavaScript 会向上遍历原型链，直到找到给定名称的属性为止;到查找到达原型链的顶部（Object.prototype），仍然没有找到指定的属性，就会返回 undefined。

\_\_proto\_\_是一个指针，它指的是构造它的对象的对象的prototype

<https://www.cnblogs.com/alichengyin/p/4852616.html>

https://blog.csdn.net/zhuanyemanong/article/details/80325387

1. call和apply常用的方法，区别有什么？

常用方法改变this指向

区别：

对于第一个参数意义都一样，但对第二个参数：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，而call方法的参数传递方法如下 func.call(func1,var1,var2,var3)

对应的apply写法为：func.apply(func1,[var1,var2,var3])；

同时使用apply的好处是可以直接将当前函数的arguments对象作为apply的第二个参数传入。

1. bind方法？

和call、apply一样的用来改变this指向

相似处：

1、都是用来改变函数的this对象的指向的。

2、第一个参数都是this要指向的对象。

3、都可以利用后续参数传参。

call() 和apply()的第一个参数相同，就是指定的对象。这个对象就是该函数的执行上下文。

call()和apply()的区别就在于，两者之间的参数。

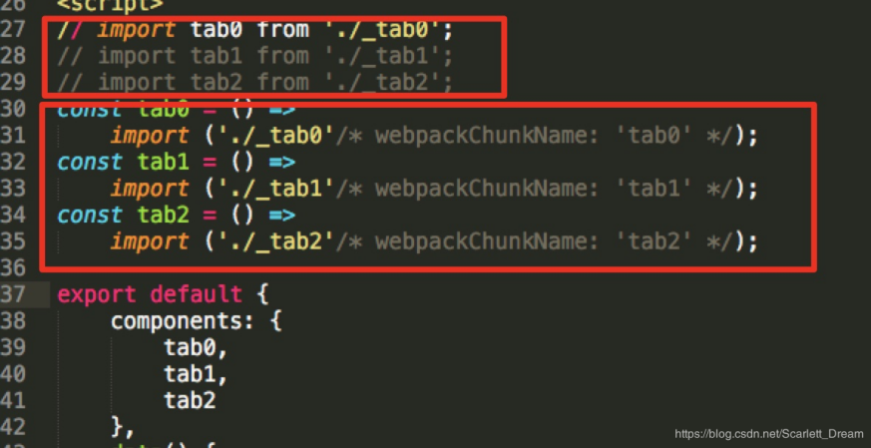
call()在第一个参数之后的 后续所有参数就是传入该函数的值。apply() 只有两个参数，第一个是对象，第二个是数组，这个数组就是该函数的参数。

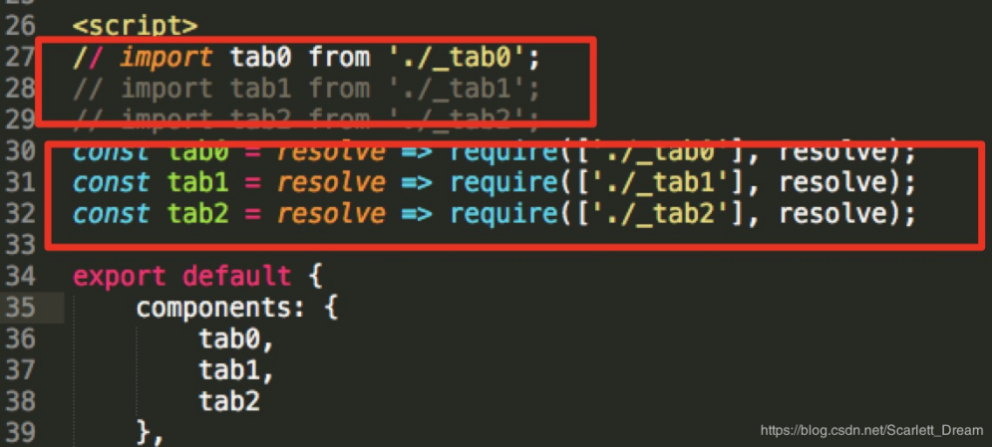
bind() 方法和前两者不同在于： bind()改过this后，不执行函数，会返回一个绑定新this的函数，而前两者是直接执行该函数。他的参数和call()相同。

<https://www.cnblogs.com/zt123123/p/7676633.html>

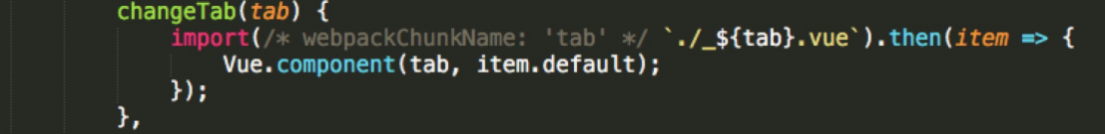
1. 什么情况下需要矫正this指向？
2. Vue代码包异步引入加载方法？

<https://blog.csdn.net/scarlett_dream/article/details/83756392>

1. 使用() => import()  
   
2. 使用resolve => require([’./\_account’], resolve)



1. 使用Webpack 的内置语句： import(\*)



1. require.ensure：

require.ensure([’./tab0.vue’], () => { resolve(require(’./tab0.vue’)) }, ‘tab0’)

1. Typescript：<https://www.tslang.cn/>
2. 移动端弹性布局

rem是CSS3新增的一个相对单位(root em,根em),使用rem为元素设定字体大小事，仍然是相对大小但相对的只是HTML根元素。

Px、rem、em区别：

IE无法调用那些使用px作为单位的字体大小，而em和rem可以缩放，rem相对的只是HTML根元素。这个单位可谓集相对大小和绝对大小的优点于一身，通过它既可以做到只修改根元素就成比例地调整所有字体大小，又可以避免字体大小逐层复合的连锁反应。目前，除了IE8及更早版本外，所有浏览器已支持rem。

弹性盒模型：

盒模型：1)，盒模型内元素分配器父元素空间

　　2），建立在块级和行级方向上（对块级元素和行内元素的布局），

弹性盒子：1），盒内子元素分配其父元素的可用空间（剩余空的空间）。

　　　2），建立在弹性流上（元素随可显示区域变化呈流式布局）。

优点：自适应，用在移动端开发非常合适，使用简单，哪怕对不定宽高元素依然好用。

10、样式决定整个页面事件禁用？

pointer-events: auto | none | visiblePainted | visibleFill | visibleStroke | visible | painted | fill | stroke | all | inherit

<http://www.mamicode.com/info-detail-2211783.html>

11、一个盒子宽度是可以有减法的，比如50% - 10px？

width: -webkit-calc(50% - 10px);

width: -moz-calc(50% - 10px);

width: calc(50% - 10px);

12、单行文本溢出隐藏显示省略号？

overflow: hidden;

text-overflow:ellipsis;

white-space: nowrap;

多行文本隐藏显示省略号？

overflow : hidden;

text-overflow: ellipsis;

display: -webkit-box;

-webkit-line-clamp: 2;

-webkit-box-orient: vertical;

13、https://www.cnblogs.com/dongcanliang/p/7054176.html

匿名函数this指向>>>>window

在非箭头函数下， this 指向调用其所在函数的对象，而且是离谁近就是指向谁（此对于常规对象，原型链， getter & setter等都适用）；

构造函数下，this与被创建的新对象绑定；

DOM事件，this指向触发事件的元素；

内联事件分两种情况，bind绑定， call & apply 方法等

箭头函数this指向

由于箭头函数不绑定this， 它会捕获其所在（即定义的位置）上下文的this值， 作为自己的this值，

所以 call() / apply() / bind() 方法对于箭头函数来说只是传入参数，对它的 this 毫无影响。

考虑到 this 是词法层面上的，严格模式中与 this 相关的规则都将被忽略。（可以忽略是否在严格模式下的影响）

箭头函数会改变this指向，vue对this指向有过矫正，一直指向vue实例

14、vue计算属性和watch监听区别？

共同点：都是以Vue的依赖追踪机制为基础的，都是希望在依赖数据发生改变的时候，被依赖的数据根据预先定义好的函数，发生“自动”的变化。；

不同点：watch擅长处理的场景：一个数据影响多个数据；computed擅长处理的场景：一个数据受多个数据影响。虽然计算属性在大多数情况下更合适，但有时也需要一个自定义的侦听器，当需要在数据变化时执行异步或开销较大的操作时，通过侦听器最有用。

15、过滤器filters的作用？

过滤器是一个通过输入数据，能够及时对数据进行处理并返回一个数据结果的简单函数。

https://blog.csdn.net/tanga842428/article/details/77679853

计算属性的区别



后端给的时间戳用来改成不同的格式

Filters可以传参，不过第一个永远都是当前的数据，后面的是参数，不会因为传参的位置为改变

16、vue写组件，封装输入框组件

饿了么的输入框是封装过的，为什么可以用v-model取出输入框的value值，外面有一层div

父子组件的传参，v-model是什么

v-model就是一个事件加上去props， >>>>> bind：value+@input

17、http请求状态码？

200请求已成功，请求所希望的响应头或数据体将随此响应返回。

201请求已经被实现，而且有一个新的资源已经依据请求的需要而建立，且其 URI 已经随Location 头信息返回。假如需要的资源无法及时建立的话，应当返回 '202 Accepted'

202服务器已接受请求，但尚未处理。（正如它可能被拒绝一样，最终该请求可能会也可能不会被执行。在异步操作的场合下，没有比发送这个状态码更方便的做法了。返回202状态码的响应的目的是允许服务器接受其他过程的请求（例如某个每天只执行一次的基于批处理的操作），而不必让客户端一直保持与服务器的连接直到批处理操作全部完成。在接受请求处理并返回202状态码的响应应当在返回的实体中包含一些指示处理当前状态的信息，以及指向处理状态监视器或状态预测的指针，以便用户能够估计操作是否已经完成。）

203服务器已成功处理了请求，但返回的实体头部元信息不是在原始服务器上有效的确定集合，而是来自本地或者第三方的拷贝。（当前的信息可能是原始版本的子集或者超集。例如，包含资源的元数据可能导致原始服务器知道元信息的超级。使用此状态码不是必须的，而且只有在响应不使用此状态码便会返回200 OK的情况下才是合适的。）

302：请求的资源临时从不同的 URI 响应请求。由于这样的重定向是临时的，客户端应当继续向原有地址发送以后的请求。只有在 Cache-Control 或 Expires 中进行了指定的情况下，这个响应才是可缓存的。

303对应当前请求的响应可以在另一个 URI 上被找到，而且客户端应当采用 GET 的方式访问那个资源。这个方法的存在主要是为了允许由脚本激活的POST请求输出重定向到一个新的资源。这个新的 URI 不是原始资源的替代引用。同时，303响应禁止被缓存。当然，第二个请求（重定向）可能被缓存。 　　新的 URI 应当在响应的 Location 域中返回。除非这是一个 HEAD 请求，否则响应的实体中应当包含指向新的 URI 的超链接及简短说明。 　　注意：许多 HTTP/1.1 版以前的 浏览器不能正确理解303状态。如果需要考虑与这些浏览器之间的互动，302状态码应该可以胜任，因为大多数的浏览器处理302响应时的方式恰恰就是上述规范要求客户端处理303响应时应当做的。

304如果客户端发送了一个带条件的 GET 请求且该请求已被允许，而文档的内容并没有改变，则服务器应当返回这个状态码。304 响应禁止包含消息体，因此始终以消息头后的第一个空行结尾。

401当前请求需要用户验证。该响应必须包含一个适用于被请求资源的 WWW-Authenticate 信息头用以询问用户信息。客户端可以重复提交一个包含恰当的 Authorization 头信息的请求。如果当前请求已经包含了 Authorization 证书，那么401响应代表着服务器验证已经拒绝了那些证书。如果401响应包含了与前一个响应相同的身份验证询问，且浏览器已经至少尝试了一次验证，那么浏览器应当向用户展示响应中包含的实体信息，因为这个实体信息中可能包含了相关诊断信息。

402该状态码是为了将来可能的需求而预留的。

403服务器已经理解请求，但是拒绝执行它。

404请求失败，请求所希望得到的资源未被在服务器上发现。

500服务器遇到了一个未曾预料的状况，导致了它无法完成对请求的处理。一般来说，这个问题都会在服务器端的源代码出现错误时出现。

18、RESTful api： <https://blog.csdn.net/wl_1013/article/details/81049691>

19、提交方式：put、delete：<https://www.cnblogs.com/foodoir/p/5911099.html>

20、app出现bug，手机上存在大量的缓存，自己这边修复好后，别人那边缓存还存在以为没有进行修复，如何去除缓存？

<http://www.cnblogs.com/xyyt/p/10271687.html>

<https://blog.csdn.net/li420520/article/details/83509127>

https://blog.csdn.net/sdta25196/article/details/85259557

21、平时浏览器（客户端）的信息，那些信息会带到后端，出现在后端的请求数据格式

Post请求，发送后台的数据，除了能接收到商务id之外还能接收到什么

请求头有什么，requestHead content-type返回内容的MIME类型，审查请求时json格式、表单格式请求的格式都是不同的，请求到的数据格式都是不同的

https://blog.csdn.net/xiaochengyihe/article/details/80910913

content-type： multipart/form-data application/json text/html

22、cookie，localStorage和sessionStorage区别？

sessionStorage和localStorage是HTML5有了本地数据存储方法，可以避免数据在浏览器和服务器间不必要地来回传递。

sessionStorage、localStorage、cookie都是在浏览器端存储的数据。

sessionStorage在浏览器窗口没有关闭时，即使刷新页面或进入同源另一个页面时，数据仍然存在。关闭窗口后，sessionStorage即被销毁。

cookies会发送到服务器端。其余两个不会。

cookie有大小限制，大小为每个域名50个。

cookie属性？

name字段 ：一个cookie的名称。

value字段 ：一个cookie的值。

domain字段 ：可以访问此cookie的域名

path字段：可以访问此cookie的页面路径。

Size字段 ：此cookie大小。

http字段 ：cookie的httponly属性。若此属性为true，则只有在http请求头中会带有此cookie的信息，而不能通过document.cookie来访问此cookie。

secure 字段 ：设置是否只能通过https来传递此条cookie

23、用户在浏览器上输入一个域名为什么会出来一个页面？

1.在浏览器中输入地址,如:www.baidu.com

2.向DNS服务器查询网站IP地址

3.DNS服务器返回网站IP地址(如:119.75.217.56)

4.浏览器得到IP地址后.浏览器会把用户输入的域名转化为HTTP服务请求

5.服务器接收到请求后,返回网页信息

6.客户端浏览器将这些信息组织成用户可以查看的网页形式

24、了解cdn？

https://blog.csdn.net/dc\_programming/article/details/80811333

cdn全称是内容分发网络。其目的是让用户能够更快速的得到请求的数据。简单来讲，cdn就是用来加速的，他能让用户就近访问数据，这样就更更快的获取到需要的数据。

25、浏览器优化性能？

1. 减少 HTTP请求数

2. 将外部脚本置底（将脚本内容在页面信息内容加载后再加载）

3. 异步执行 inline脚本(其实原理和上面是一样，保证脚本在页面内容后面加载。)

4. Lazy Load Javascript（只有在需要加载的时候加载，在一般情况下并不加载信息内容。）

5. 将 CSS放在 HEAD中

6. 异步请求 Callback（就是将一些行为样式提取出来，慢慢的加载信息的内容）

7. 减少不必要的 HTTP跳转

8. 避免重复的资源请求

9. 图片优化

10. 使用CDN

11. 开启GZIP

12. 样式表和JS文件的优化

13. 使用无cookie域名