自我介绍：

面试官您好，我叫赵涛，来自河北张家口，很荣幸今天面试贵公司的前端岗位，我从事web前端已经3年，供职过两家公司，熟练掌握vue技术栈，对前端工程化与模块化开发有深度的了解，同时具有web性能优化的实践经验，最近做的一个是vue的后台管理项目

它主要是为了打通医院的本身的医院管理系统，线上线下医患的同步操作



1.盒模型有两种，W3C 和IE 盒子模型

1、W3C定义的盒模型包括margin、border、padding、content，元素的宽度width=content的宽度

2、IE盒模型与W3C盒模型的唯一区别就是元素的宽度，元素的width=border + padding + content，在css3中新增了box-sizing

2、实现一个不设置宽高的盒子水平垂直居中的效果的方法

方法一:

css+定位

让未定义宽高的图片上下左右距离都为0.然后给一个margin自适应

<style>

img{

position:absolute;

top:0;

bottom:0;

left:0;

right:0;

margin: auto ;

}

</style>

<body>

<img src="img/20181008095016\_59996.jpg" alt="">

</body>

方法二：

通过设置内部元素绝对定位，给一个translate属性，让其在x轴和y轴进行平移。

<style>

.ided{

position:absolute;

top: 50%;

left: 50%;

transform: translate(-50%, -50%);

/\* X负数向左平移 Y轴负数往上平移 \*/

background-color: burlywood;

width:200px;

height:200px;

}

</style>

</head>

<body>

<div class="ided"> </div>

</body>

 方法三：

通过display:table-cell指让标签元素以表格单元格的形式呈现，vertical-align 用来指定行内元素（inline）或表格单元格（table-cell）元素的垂直对齐方式

//1.滚动条到顶端的距离（滚动高度）

var scrollTop = document.documentElement.scrollTop || document.body.scrollTop;

//2.滚动条到左端的距离

var scrollLeft = document.documentElement.scrollLeft || document.body.scrollLeft;

//3. IE9以下byClassName

function byClassName(obj,className){

//判断是否支持byClassName

if(obj.getElementsByClassName){

//支持

return obj.getElementsByClassName(className);

}else{

//不支持

var eles = obj.getElementsByTagName('\*'); //获取所有的标签

var arr = []; //空数组，准备放置找到的对象

//遍历所有的标签

for(var i = 0,len = eles.length;i < len;i ++){

//找出与我指定class名相同的对象

if(eles[i].className === className){

arr.push(eles[i]); //存入数组

}

}

return arr; //返回

}

}

//4. 获取非行内样式兼容 IE:currentStyle 标准：getComputedStyle

function getStyle(obj,attr){

return window.getComputedStyle ? getComputedStyle(obj,true)[attr] : obj.currentStyle[attr];

}

//div.style.width = '';设置样式

//obj['属性']： 对象是变量时，必须用对象['属性']获取。

//5. 获取事件对象的兼容

evt = evt || window.event

//6. 获取鼠标编码值的兼容

function getButton(evt){

var e = evt || window.event;

if(evt){

return e.button;

}else if(window.event){

switch(e.button){

case 1 : return 0;

case 4 : return 1;

case 2 : return 2;

}

}

}

//7. 获取键盘按键编码值的兼容

var key = evt.keyCode || evt.charCode || evt.which;

//8. 阻止事件冒泡的兼容

e.stopPropagation ? e.stopPropagation() : e.cancelBubble = true;

//9. 阻止超链接的默认行为的兼容

evt.preventDefault ? evt.preventDefault() : evt.returnValue = false;

//10. 添加事件监听器的兼容

function addEventListener(obj,event,fn,boo){

if(obj.addEventListener){

obj.addEventListener(event,fn,boo);

}else if(obj.attachEvent){

obj.attachEvent('on' + event,fn);

}

}

//11. 移除事件监听器的兼容

function removeEventListener(obj,event,fn,boo){

if(obj.removeEventListener){

obj.removeEventListener(event,fn,boo);

}else if(obj.detachEvent){

obj.detachEvent('on' + event,fn);

}

}

//12. 获取事件源的兼容

var target = event.target || event.srcElement;

**css省略号显示**

 /\* 显示一行，省略号 \*/

**white-space: nowrap;**

**text-overflow: ellipsis; //省略号**

**overflow: hidden;**

**word-break: break-all;**

/\* 显示两行，省略号 \*/

**text-overflow: -o-ellipsis-lastline;**

**overflow: hidden;**

**text-overflow: ellipsis;**

**display: -webkit-box;**

**-webkit-line-clamp: 2;**

**line-clamp: 2;**

**-webkit-box-orient: vertical;**

**介绍一下 ES6 中 Set, Map的区别？**

Map

在JS中的默认对象的表示方式为{}，即一组键值对，但是键必须是字符串。

为了使用Number或者其他数据类型作为键，ES6规范引入了新的数据类型Map。

Map是一组键值对的结构，具有极快的查找速度。初始化Map需要一个二维数组，或者直接初始化一个空Map。

Map 对象是键值对集合，和 JSON 对象类似，但是 key 不仅可以是字符串还可以是其他各种类型的值包括对象都可以成为Map的键

Set

Set也是一组key的集合，与Map类似。但是区别是Set不存储value，并且它的key不能重复。

创建一个Set，需要提供一个Array作为输入，或者直接创建一个空Set：

重复元素会在Set中自动被过滤

Set 对象类似于数组，且成员的值都是唯一的

**简述 aync await 的好处**

1. async/await最重要的好处是同步编程风格

2. async/await有本地浏览器支持。截至今天，所有主流浏览器 查看都完全支持异步功能。

3.async关键字。它声明 getBooksByAuthorWithAwait()函数返回值确保是一个 promise，以便调用者可以安全调用 getBooksByAuthorWithAwait().then(...)或 await getBooksByAuthorWithAwait()

**移动端点击事件 300ms 延迟如何去掉？原因是什么？**

300毫秒原因：

当用户第一次点击屏幕后，需要判断用户是否要进行双击操作，于是手机会等待300毫秒，

解决方法：FastClick.js

FastClick 是 FT Labs 专门为解决移动端浏览器 300 毫秒点击延迟问题所开发的一个轻量级的库。FastClick的实现原理是在检测到touchend事件的时候，会通过DOM自定义事件立即出发模拟一个click事件，并把浏览器在300ms之后的click事件阻止掉。

**区别webpack与gulp/grunt**

1). webpack是一种模块化打包工具，主要用于模块化方案，预编译模块的方案。  
2). Grunt/Gulp是非模块化打包工具, 是工具链，可以配合各种插件做js压缩，css压缩，less编译 替代手工实现自动化工作

**说说你对变量提升与函数提升的理解**

|  |
| --- |
| 变量提升: 在变量语句前就可以读取到变量, 值为undefined 函数提升: 在函数定义语句前就可以调用函数 原因: JS引擎在运行全局代码或执行函数前有预处理/解析 |

**区别localStorage和sessionStorage, session与cookie**

|  |
| --- |
| localStorage: 浏览器端持久化存储, 关闭浏览还存在, 最大5MB(基本没限制了) sessionStorage: 浏览器端内存存储, 关闭浏览器不存在 session: 服务器端创建, 服务器端保存, 依赖于cookie cookie: 服务器端创建, 浏览器端保存, 请求携带对应cookie, 长度和数量有限制(4kb) |

**vue的常用配置选项及其作用(5个)**

|  |
| --- |
| el: 最外层元素选择器  props: 声明接收哪些属性 data: 状态数据 computed: 计算属性 methods: 事件回调函数 watch: 监视属性变化  directives: 注册局部指令  filters: 注册局部过滤器  components: 配置组件 |

**区别2种路由实现：hash模式 和 history模式**

|  |
| --- |
| 1). hash模式：  在浏览器中符号“#”，#以及#后面的字符称之为hash，用window.location.hash读取；  注意：hash虽然在URL中，但不会提交给服务器端, 只在浏览器端使用 2). history模式：  路由路径中没有#, 利用HTML5的新特性: pushState()/replaceState()及popState事件监听  注意: 路径会提交给后台, 在生产环境下会出现404的问题  解决404问题: 后台服务器端配置404页面或通过自定义中间件指定404页面为index.html页面 |

**GET请求的2种请求参数**

|  |
| --- |
| query参数:   路由path: /register  请求path: /register?username=xxx&password=yyy   获取参数: req.query.username param参数:   路由path: /register/:username/:password  请求path: /register/xxx/123   获取参数: req.params.username |

**说说你vue项目开发中碰到的问题, 原因及解决办法**

|  |
| --- |
| *1*). 创建Swiper/BScroll对象后, 轮播/滑动没有效果?  原因: 创建对象太早, 得在数据显示后创建  解决: watch + $nextTick() 或 callback + $nextTick() 2). 初始显示异常  情况1: Cannot read property 'xxx' of undefined"  原因: 初始值是空对象, 内部没有数据, 而模块中直接显示3层表达式  解决: 使用v-if指令    情况2: Cannot read property 'avatar' of null"  原因: 初始值是null, 而模块中直接显示2层表达式  解决: 初始值为{} 3). 问题: 更新状态数据, 对应的界面不变化  原因: 一般方法给一个已有绑定的对象中添加一个新的属性, 这个属性没有数据绑定  解决:   Vue.set(obj, 'xxx', value)才有数据绑定  this.$set(obj, 'xxx', value)才有数据绑定 4). 问题: 点击添加购物项, 会1次添加多个  原因: 创建了多个BScroll对象来管理同一个DOM元素  解决: 只创建一个BScroll对象   扩展: 单例对象:   创建前, 先判断是否已经存在, 只有不存在才创建  创建后, 保存创建的对象 |

**动态路由具体实现**

1. 创建vue实例的时候将vue-router挂载，但这个时候vue-router挂载一些登录或者不用权限的公用的页面。
2. 当用户登录后，获取用role，将role和路由表每个页面的需要的权限作比较，生成最终用户可访问的路由表。
3. 调用router.addRoutes(store.getters.addRouters)添加用户可访问的路由。
4. 使用vuex管理路由表，根据vuex中可访问的路由渲染侧边栏组件

**状态码**

1XX 信息性状态码

2XX 成功状态码

3XX 重定向状态码

4XX客户端错误状态码

5XX服务器错误状态码

我们常见的状态码有200--表示客户端请求成功，400--表示客户端请求有语法错误，403--表示服务器收到请求，但是没有权限，404--表示请求资源不存在，500--表示服务器端错误

**你做过哪些Vue的性能优化？**

1、首屏加载优化

2、路由懒加载

3、ssr

4、骨架屏、

5、图片懒加载

6、v-if,v-show，keep-alive

**说说你项目中使用的better-scroll库**

|  |
| --- |
| 1). 什么: 用来实现页面滑动相关效果的js库 2). 原理: 当内容的高度超过了容器的高度(固定)时, 才会形成滚动 3). 基本编码:  scroll = new BScroll(selector, {})  scroll.on(eventName, cb)  scroll.scrollTo()  scroll.scrollToElement()  scroll.refresh() 4). 注意:  创建BScroll对象的时机  创建BScroll对象的个数  BScroll禁用的原生dom事件, 自动定义事件分发 |

**35.$route和 $router的区别是什么？**

$router为VueRouter的实例，是⼀个全局路由对象，包含了路由跳转的⽅法、钩⼦函数等。

$route 是路由信息对象||跳转的路由对象，每⼀个路由都会有⼀个route对象，是⼀个局部对象，包含

path,params,hash,query,fullPath,matched,name

**46.computed、watch（⾃动监听、深度监听）、methods区别**

我们可以将同⼀函数定义为⼀个 method 或者⼀个计算属性。对于最终的结果，两种⽅式是相同的。

不同点：

computed：计算属性是基于它们的依赖进⾏缓存的，只有在它的相关依赖发⽣改变时才会᯿新求值。

methods：只要发⽣᯿新渲染， method 调⽤总会执⾏该函数。

watch监听对象需要深度监听，默认是浅监听

当⻚⾯中有某些数据依赖其他数据进⾏变动的时候，可以使⽤计算属性computed。

watch⽤于观察和监听⻚⾯上的vue实例，如果要在数据变化的同时进⾏异步操作或者是⽐较⼤的开销，那么watch为 最佳选择

**完整的说下从url解析到显示⻚⾯过程，结合项⽬中说**

1. ⾸先浏览器主进程接管，开了⼀个下载线程。

2. 然后进⾏HTTP请求（DNS查询、IP寻址等等），中间会有三次捂⼿，等待响应，开始下载响应报⽂。

3. 将下载完的内容转交给Renderer进程管理。

4. Renderer进程开始解析css rule tree和dom tree，这两个过程是并⾏的，所以⼀般我会把link标签放在⻚⾯

顶部。

5. 解析绘制过程中，当浏览器遇到link标签或者script、img等标签，浏览器会去下载这些内容，遇到时候缓存的使

⽤缓存，不适⽤缓存的᯿新下载资源。

6. css rule tree和dom tree⽣成完了之后，开始合成render tree，这个时候浏览器会进⾏layout，开始计算

每⼀个节点的位置，然后进⾏绘制。

1. 绘制结束后，关闭TCP

## 数组去重new Set

New set之后得到的是一个对象，可以进行遍历for...of

通过array.from(new set(arr))转换为数组