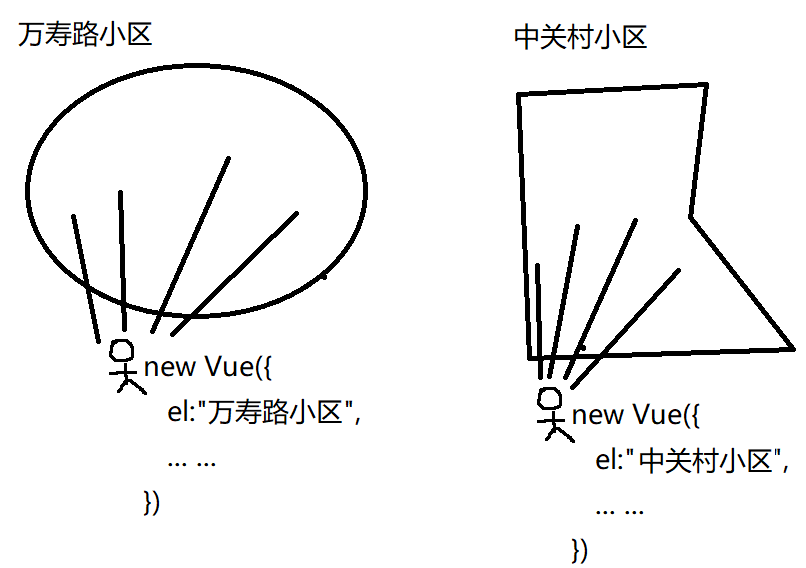
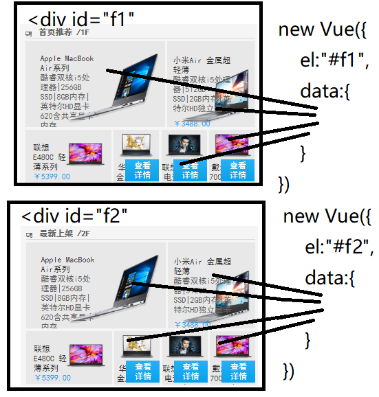
1. el:"选择器"，如果页面中多个选择器需要监控，那么new Vue()如何找到不同选择器下的元素进行更新？

答: 和平时快递小哥一样: 每个专门的区域，都由专门的一个快递小哥负责维护。



在网页中也是一样: 每个独立的功能区域，都需要一个专门的new Vue()对象监控:



所以，今后都是将页面分成多个功能区域，然后再新建多个new Vue()对象。每个new Vue()对象负责监控自己负责的区域中数据的变化，并显示数据。

2. new Vue的V必须大写吗？

答: new后跟的是一种类型名，意为创建某个类型的一个具体对象。比如new Student(), new Array()。一种类型的名称，通常都是大写字母开头的。比如: class Student{}, 比如class Array{}等。

3. Error compiling template错误:

答：凡是见到这个错误，都是HTML中绑定语法写错了！也可能是HTML写的不标准。

要看错误原因，直接跳过中间大片代码，直接看代码结尾的问题说明。说明包含三部分: 1. 错误的原因， 2. 修改意见， 3. 修改的样例。

4. vs code中，明明语法没写错，还冒红/绿提示错误？

答：vs code不同版本对vue的语法支持的不一定好，可能出现误判。所以，以最后能否运行为标准。不要过分依赖编辑器的报错提示。而应该以实际运行的结果是否报错为准。但是。。。十有八九是自己写错了。还是先检查一下

5. 现在应该如何复习以前的知识？

答: 现在学习阶段以及今后工作中，最忌讳的就是大篇幅的回看原来旧的知识。因为大部分可能你都有印象，知识不确定而已。大篇幅回看旧的知识会浪费大量宝贵的时间和精力。——看着真的会睡着。←\_←

正确复习的办法很简单，就是打开那个阶段的思维导图，只看用法！实在需要看原理了，再找那个阶段对应的小视频看一下。

所以，思维导图最重要！

6. created()是什么？

答: created()就是Vue中的window.onload。

都是系统预先定义好的固定函数，在事件发生时自动调用。onload是整个页面加载后才执行。created()是Vue对象加载后就自动执行。onload不能乱改名，created()也不能乱改名。大家要学会类比和举一反三。

7. this的汇总:

1. 普通函数或匿名函数自调中的this->window

严格模式下: this->undefined

2. obj.fun() fun中的this->obj

3. new Fun() Fun中的this->正在创建的新对象

4. 原型对象中的函数里的this->将来调用当前函数的.前的当前类型的子对象。

5. btn.onclick=function(){... } this->btn

6. 回调函数:

arr.forEach(function(){ ... })

arr.map(function(){ ... })

setInterval(function(){ ... })

setTimeout(function(){ ... })

$.ajax({

...

success:function(){ ... }

}).then(function(){

...

})

this->window 所有回调函数，真正被调用时，前边是没有任何"对象."前缀

所以，通常如果希望回调函数中的this不指window，而是跟外部的this保持一致，都要改为箭头函数

7. jQuery中的回调函数:

$().each(function(){ ... })

$().animate({ ... }, ms, function(){ ... })

jquery中的多数回调函数,this->当前正在操作的dom元素

8. 不考虑之前已经总结的情况，vue中一切this都指向当前new Vue()对象。所以在vue js中访问任何变量都要加this.变量名。但是html中绑定变量名不用加this！

8. v-if为什么可以隐藏？没有写隐藏的方法呀？

答: 框架自动判断v-if，根据v-if的条件决定是显示元素还是隐藏元素。至于隐藏元素使用的什么具体语句，已经被框架封装在了内部，由框架无人值守的自动调用。这是人和框架之间基本的信任。我们无从考证，只能信任框架一定会帮我们完成这件事。不信任框架，就只能自己受累，每一句话自己写。打算用框架，就只能记住框架的要求和写法，然后相信框架一定会帮我们自动完成任务。

9. data:{ pm25:65 }的变量值是先写死的，后来才在created()中动态变化的。那么如果一开始不知道pm25的值，完全需要今后动态获取怎么办？

答: 无论是变量，还是对象的属性，其实都有一个良好的习惯，就是初始化时，都要给一个初始值。比如：

var obj=null; 或var obj={} 对象一般给null，或者{}；

var arr=[]; 数组一般给空数组[]；

var str="" 字符串，一般给""；

var num=0 或 var i=-1。数字，一般根据需要给-1或0作为起始值。

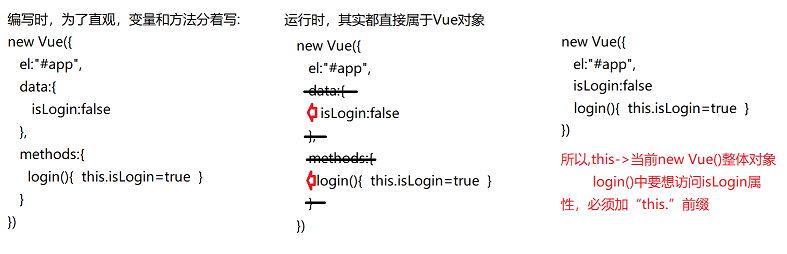
所以，将来变量pm25初始时不会空着，会根据需要，暂时定义为pm25:0，作为初始值。

9. var isLogin=this.isLogin;  
 无论isLogin=true，还是isLogin=false页面都不变？

答: 按值传递。this.isLogin中保存的是bool类型的值，是原始类型的值。原始类型的值，赋值给变量时，采用的是复制副本的方式。也就是拷贝this.isLogin中的值的副本给isLogin变量。在这一瞬间，内存里有两个值。一个给了变量isLogin，一个保存在了this.isLogin中。所以修改isLogin中的副本，原this.isLogin不变。自然就监控不到变化。页面不变。

10. methods中的login()中的this指谁？

答: vue中一切this，都指当前new Vue()对象整体:



11. v-if看起来也可以用来做轮播图?

答: v-if可以做轮播图，但是只能做瞬间显示隐藏的轮播图。因为v-if是采用添加删除元素的方式控制元素显示隐藏。所以，无法使用过度效果。所以，v-if几乎不能用于做轮播图

12. <ul>

<li></li>  
 <li></li>  
 <li></li>

</ul>  
 <div>  
 <div>  
 <div>  
 <div>

用上边li控制下边div的显示隐藏？

答: 标签页效果，三个div，三选一显示，思路:   
 用一个变量i控制三个div三选一显示。i变成几，就对应显示第几个div。  
 点击每个li时，通过事件处理函数传入当前li是第几个。然后，将li的序号修改到data中的i变量上。  
 代码:

data:{ i:1 //将来可能被改为1, 2, 3}  
 //三个div都监视着这个变量i，变成几，就显示第几个div。  
 methods:{ show(i){ this.i=i } }

//定义一个方法，让三个li共用，只不过，每次点击li时传入一个序号作为参数值。并将li的序号，改为要显示的div的序号。

HTML中:   
 <li @click="show(1)">  
 <li @click="show(2)">

<li @click="show(3)">

<div v-if="i==1">  
 <div v-else-if="i==2">  
 <div v-else>

结果：点击li，触发show，传入序号，修改data中的i为当前序号，触发通知，通知比较三个div的条件。哪个div的条件满足，就切换成显示哪个div。

其实: 事件绑定时，处理函数后的()，可加可不加。无参数传入时，就不加。必须传入参数时，才加。

13. 为什么v-if="条件"，不用()，而要用=？

答: 因为v-if内置上是HTML语言中的一种自定义属性。必须符合HTML的语法才行。HTML中，属性名和属性值之间只能用=。

14. create()为什么加()， methods:{为什么是:{？

答: created，是Vue自带的一个函数，可以执行一项任务。所以created时，要加()。

而methods是组织Vue中的函数的一种存储结构，不用来做事儿，仅仅用于划分一个专门的区域存储函数。所以不加()，而应该写成一个对象。

15. v-bind和v-if实现同样的功能，哪个效率高？

答: v-bind因为不改变DOM树，所以效率高。

v-if v-else-if v-else 在不断添加删除元素，频繁影响DOM树，所以效率略低。

16. 如果<li>上带class，能不能循环出来？

答: 可以。只要是v-for模板元素上带的，都可以循环出来。比如: <li class="tab tab-primary" v-for="(elem,i) of arr">

结果：<li class="tab tab-primary"></li>

<li class="tab tab-primary"></li>

<li class="tab tab-primary"></li>

... ...

17. v-for内可不可以嵌套v-if或v-show?

答: 可以

比如: <div id="app">

<ul>

<li v-for="(score,i) of scores" :key="i">

<span v-if="score>=60">及格</span>

<span v-else>不及格</span>

</li>

</ul>

</div>

<script>

new Vue({

el:"#app",

data:{ scores:[89,50,65,59] }

})

18. Error: listen EADDRINUSE :::3000

答: node app.js 启动了两次，第二次报3000端口被占用。只要关闭vscode，重新打开

19. Access to XMLHttpRequest at 'file:///C:/xampp/htdocs/dd/PROJECT/public/header.html'

答: 求求你们了！不要再在硬盘上双击打开网页了。必须用live server运行。求求你们了！

20. v-text="p2.title" 不加反引号

v-text="`用户名:${uname}`" 加反引号?

答: 其实v-text中本不需要加反引号``，直接可以写js变量或表达式

v-text="p2.title" 因为p2.title本身就可以直接使用，所以不加反引号``

但是，如果v-text中需要将变量和普通的字符串拼接显示时，就必须用js中的模板字符串语法拼接字符串

v-text="`用户名:${uname}`" 因为显示的结果是普通字符串"用户名:"和变量uname字符串拼接的结果，所以必须放在js规定的反引号中拼接好，再将结果给v-text

总结: HTML中，普通属性后的""中只能写字符串:

src="字符串" href=""

Vue中，:src="js语句或变量" :href=" js语句或变量" 其中的""中反而不能写普通的字符串。必须写一条正确的js表达式或变量

比如: :src="/images/logo.png" 错！

21. :src="{{p2.pic}}" 报错?

答: 属性中不能用{{}}绑定，必须用:绑定。其实:是替换{{}}的作用。

22. 明明网页修改对了，但是刷新页面不更新?

答: 浏览器是有缓存机制的。浏览器怕反复请求服务器，所以会将网页的内容暂时缓存在本地硬盘中。每次重新请求时，先在本地硬盘中查找缓存使用。缓存中没有，或者过期，才被迫去服务器请求新资源。

解决: 右上角三个点按钮->历史记录->历史记录->左侧点清除浏览数据->清除数据

23. v-for反复创建元素时，元素的子元素是否会同时反复生成？

答: 会！

24. 用live server打开，端口号变为:127.0.0.1:5501或5502？

答：通常因为5500端口被占用，自动更换为一个新的端口继续运行网页。

如果想恢复5500，可关闭vscode，甚至重启电脑，再打开。

如果实在改不回5500，也可以去服务端app.js中将cors:后的地址和端口号改为一致的5501或5502

其实最偷懒的办法是，修改服务器端app.js中cors:\*

表示不限制地址，随便跨域访问。

25. 不是表单元素就不能用v-model双向绑定吗？

答: 不能！因为v-model双向绑定是为了通过修改页面的值或内容，来更改程序中的变量值。除表单元素外的其它元素，比如div, span, p元素，都只能看不能改。所以，用双向绑定没意义。

26. watch和created有什么差别？

答: created只在new Vue()对象创建完成后出发一次

watch中的函数，只要变量变化，就会出发。反复变化多次，就自动触发多次。

27. 关于两个下拉菜单级联选择省份，城市的做法:

答: <div id="app">

省份:<select v-model="i">

<option v-for="(p,index) of provs" :value="index" v-text="p"></option>

</select>

城市:<select v-model="j">

<option v-for="(p,index) of cities[i]" :value="index" v-text="p"></option>

</select>

</div>

<script>

new Vue({

el:"#app",

data:{

i:0,

j:0,

provs:["北京","河北","山东"],

cities:[

["东城","西城","海淀","朝阳"],

["石家庄","保定","唐山","廊坊"],

["济南","青岛","烟台","德州"]

]

},

watch:{

i(){ this.j=0 }

}

})

</script>

28. 多个页面跳转，要求不更新页面，只替换页面局部内容?

答: 1. 整个程序只有一个完整的home.html页面

1.1 页头，可以写死在home.html中: <header>...</header>

1.2 主体内容，只是一个<main></main>空元素

2. 其它所谓的页面，其实都只是一个HTML片段:

比如: 首页: index.html中只包含: <main>首页的内容</main>

3. 所有页面中的超链接，都不要用href，都应该改为单击事件:   
 <a href=”javascript:;”>首页</a>

$a.click(function(){

$.ajax({ //ajax请求html片段

[url:”index.html](url:)”，

type:”get”,

success:function(html){

$(html).replaceAll(“main”) //用ajax请求回来的html片段替换唯一主页home.html中的<main></main>元素

$(`<link rel=”stylesheet” href=”css/index.css”`).appendTo(“head”)

//好人做到家，动态添加link css

}

})

})

29. 即使写了[v-cloak]{ display:none }，也不隐藏？

答: 因为bootstrap中部分class中带有!important。如果想覆盖，必须: [v-cloak]{display:none !important}

30. 能不能监控内嵌在对象中的变量或属性？

答: data中的一切变量都可被监控，但是，如果内嵌的变量或者属性，监控时，函数名时一个完整的查找对象的字符串:

data:{

divStyle:{

top:"50px",

left:"50px"

}

},

watch:{

"divStyle.top"(){

console.log(`top:${this.divStyle.top}`);

},

"divStyle.left"(){

console.log(`left:${this.divStyle.left}`);

}

}

31. 什么时候用methods，computed，created，watch

答: 所有事件处理函数和根据主要自定义的函数，都放在methods中

所有需要计算才能显示的属性，都放在computed中

希望vue对象加载后自动执行的操作，放在created中

希望实时监控一个变量的变化时，都放在watch中

32. 如果想为一个套在很多层标签内的子元素绑定内容或样式，怎么办？

答: 根本不要查找，哪里需要哪里绑定即可！页面中任何位置，都可使用各种绑定语法找到data中的变量。

33. vue中获得文本框<input>或<select>等元素的内容，还用不用input.value或者select.value?

答: 不用了！vue中只要想获得表单元素的值，都用v-model，请表单元素自动给咱们送到程序的变量中。不用咱们自己找，不用咱们自己调用DOM的属性value。

比如，想获得<input>文本框的值: <input v-model="变量"

想获得<select>选中项的值: <select v-model="变量"

运行时，只要表单元素的值发生变化，v-model就会自动将新值送到底层的变量中保存。

v-model的原理，其实是自动给<input>绑定了oninput事件，给<select>绑定了onchange事件。在事件中，提前定义好了要修改哪个变量。等事件发生时，就按照提前预定的计划，更新对应的变量。

34. 如何从ftp下载.js文件?

答: 右键单击文件链接，选择"链接另存为"

35. 同样是methods中的函数。为什么@click="add"就不加()，为什么{{total()}}，就要加()

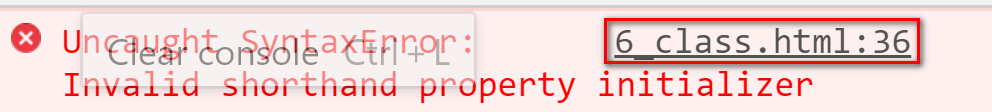
答: ()是调用函数的意思，立刻执行。@click="add"，是绑定事件处理函数，事件处理函数不是立刻执行。所以，不加()。而{{}}中调用total()，一定是要立刻执行并获得函数执行的结果，所以，必须加()。

所以，加不加()，取决于该函数是否需要立刻执行，还是等着今后再执行。

36. 英文不好，怎么定位错误？

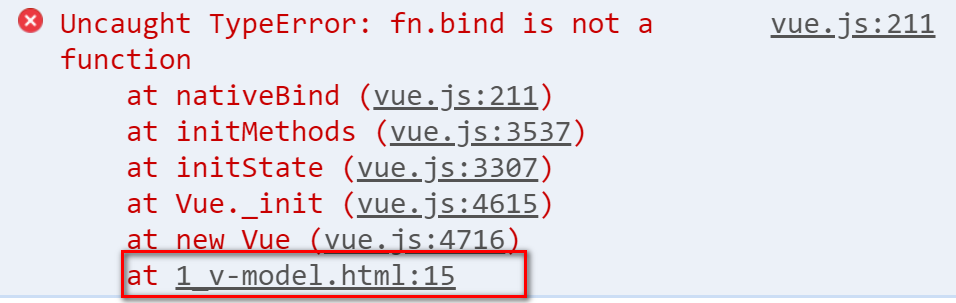
答: 只要出错，就找当前文件名！当前文件名后的数字，就是出错的行号！

比如：



说明第36行出错啦！

再比如:



说明第15行出错啦！

37. 将v-focus放在两个input上，只有一个input起作用，能获得点。难道指令只能放在一个元素上吗？

答: 任何指令没有使用元素的个数限制。

但是这个获得焦点的指令需求特殊。因为一个页面同一时刻只能有一个焦点。如果下一个元素获得了焦点，上一个元素自动就会失去焦点！所以，v-focus这个指令，只能让一个元素获得焦点！但是v-focus这个指令实际是可以用在多个元素上的。没有限制。