

Vue学习第4天

反馈

回顾

1. 数据的设置放在了定时器的外面

```
1  axios
2  .get(`http://wthrcdn.etouch.cn/weather_mini?city=${this.city}`)
3  .then(backData => {
4    // 这里赋值，loading动画还没结束，数据就显示，消失的动画来没来得及播放，重新加载
5    this.weatherList = backData.data.data.forecast
6    setTimeout(() => {
7      // 关闭loading
8      this.isLoading = false
9      // 成功
10     // console.log(backData)
11     // this.weatherList = backData.data.data.forecast
12   }, 1000)
13 })
```

2. axios 是什么？

1. 专注于网络请求的一个库
2. 和ajax的关系
 1. 浏览器中，基于Ajax的封装，和jQ的封装 大同小异
 2. axios.get().then():相比于jQ的从嵌套变为链式
 3. Promise

Vue动画钩子

[传送门](#)

1. Vue提供的让程序员可以在动画的各个时机，添加 自定义逻辑 的回调函数，也可以称之为(动画钩子)
2. 进入动画的 开始，中，结束，取消时执行
3. 离开动画的 开始，中，结束，取消时执行
4. jQ动画回调函数的高级版本
5. afterEnter:注册的位置是 `methods`
6. 把el：动画应用的dom元素传递给你

```

1 <transition
2   v-on:before-enter="beforeEnter"
3   v-on:enter="enter"
4   v-on:after-enter="afterEnter"
5   v-on:enter-cancelled="enterCancelled"
6
7   v-on:before-leave="beforeLeave"
8   v-on:leave="leave"
9   v-on:after-leave="afterLeave"
10  v-on:leave-cancelled="leaveCancelled"
11 >
12   <!-- ... -->
13 </transition>

```

Demo-播放器

实现步骤

1. 双击播放歌曲，获取封面，获取歌曲评论

1. 双击的事件中，添加下列逻辑即可

2. 封面 接口 <https://autumnfish.cn/song/detail?ids=347230>

1. v-bind:src="coverUrl"

3. 评论接口 <https://autumnfish.cn/comment/hot?id=862101001&type=0>

1. v-for commentList:[]

2. 歌曲播放，停止播放的动画效果切换

1.

```

<div class="player_con">
  
  
   == $0

```

2. 添加是否播放的标记：isPlaying

3. 上述的3个img 播放时，添加 playing 类名

1. audio的 play触发

4. 暂停播放时，移除 playing 即可

1. audio的pause触发

3. 双击的时候，获取歌曲的mv，如果有的话

1. 获取到mv之后开始播放

2. video v-bind:src=""

3. 获取mv的接口 <https://autumnfish.cn/mv/url?id=522362>

4. 传入的是mvid不要传错了

4. 设置左侧滚动 使用的是 iScroll

1. 保证结构满足，容器，内容容器，内容

2. 哪个钩子中实例化iScroll mounted

3. data中保存 iScroll 的实例

4. 获取dom元素

1. .song_wrapper 添加ref='wrapper'
2. 通过this.\$refs.wrapper
5. updated中 调用 iScroll的 refresh方法

注意点

1. 如果涉及到接口数据获取基本的逻辑都是
 1. 在某些时候 调用接口
 2. 数据回来之后操纵页面
2. 数据的获取一般都需要通过接口
3. 特定的时候基本都提供了对应的钩子函数给我们让我们添加自己的逻辑
 1. window.onload
 2. \$.aniamte(function({})
 3. updated
 4. onplay
 5. onpause
 6. ..
4. 不停的旋转，过渡还是动画实现？
 1. 动画
5. 操纵杆如何旋转的？
 1. transform:rotate
6. v的播放和歌曲的播放类似，重点是要找到接口以及传入对应的参数即可

iScroll

[传送门](#)

[中文文档传送门](#)

使用步骤

1. 准备一个3层嵌套的页面结构
 1. 标签可以随意，但是结构需要保证如下

```
1      <div id="wrapper">
2          <div>
3              <li>剁辣椒</li>
4              <li>剁辣椒</li>
5              <li>剁辣椒</li>
6              <li>剁辣椒</li>
7              <li>剁辣椒</li>
8              <li>剁辣椒</li>
9              <li>剁辣椒</li>
10             <li>剁辣椒</li>
11             <li>剁辣椒</li>
```

```
12     <li>剁辣椒</li>
13     <li>剁辣椒</li>
14     <li>剁辣椒</li>
15   </div>
16 </div>
```

2. 最外层容器的高度 要固定
3. 导入iScroll
4. 创建一个iScroll的对象即可

```
1 <script src="./lib/iscroll.js"></script>
2 <script>
3   // 获取dom
4   const wrapper = document.querySelector('#wrapper')
5   // 创建iScroll
6   const myScroll = new iScroll(wrapper)
7
8   // 也可以
9   const myScroll = new iScroll('#wrapper')
10 </script>
```

5. 内容更新了之后需要通过 `refresh()` 通知库，内容更新了，重新计算尺寸

```
1 // 通知iScroll dom更新了
2 myScroll.refresh();
```

Vue生命周期钩子 - mounted

1. 执行的时机：
 1. vue把内部的数据和页面的内容同步之后就会执行
 2. 挂载到页面上之后
2. 当我们碰到需要获取vue解析的页面内部的dom元素时，基本上都是写在这个钩子函数中
3. 执行一次
4. 数据改变之后，`updated`这个会多次执行

\$refs属性

[ref属性传送门](#)

[\\$refs属性传送门](#)

1. 给要获取的元素添加ref属性，名字随便起：有意义即可
2. 通过 `vue实例.$refs.属性名` 即可获取标记的元素
3. 这个才是vue推荐的元素获取方式
4. 如果ref重名后面的会吧前面的覆盖

Demo-播放器结合iscroll

1. ref可以帮助我们 获取元素
 1. 元素上加ref属性
 2. 获取用的是 this.\$refs
2. data中的数据一定要渲染到页面上吗？
3. 如果要获取vue管理的元素，最起码写在哪个钩子中？mounted

Vue组件基本使用

[传送门](#)

基本使用

1. 组件的注册 `vue.component('组件名',{参数})`
2. 结构用 `template`
 1. 直接写字符串：不好
 2. `template:"#模板id"`
 1. 抽取到模板中 `script type='html/text' id="tem"`
3. 页面中使用的时候，直接把 组件名当做标签名即可

方法和数据绑定

1. 方法的使用和之前没有区别
2. 数据的绑定data不在是对象，而是 函数
3. 数据的使用和之前一样没有区别

```
1    data: function() {  
2      return {  
3        message: '阿尼阿瑟哟,欧巴！'  
4      }  
5    },
```

组件的数据为什么是一个函数

画图解释

所有组件都引用同一份数据，任何一个组件修改了数据，所有引用数据的其他组件，都同步更新，互相干扰

用了函数之后

每次使用组件，都会执行这个函数，返回一个组件独有的数据对象，保证互不干扰

Vue路由基本使用

[传送门](#)

高级的tab栏

前端路由：url地址和组件的对应关系