1. 动态地创建 React 元素时，若 React 元素内包含数量或顺序不确定的子元素时，我们需要提供 key 这个特殊的属性。
2. require参数不支持变量,在数据中写相对路径，再将绝对路径传过来。
3. this.state不会在this.setState调用之后立刻更新。直到下一次render函数调用时（或者下一次shouldComponentUpdate返回false时）this.state才会更新。
4. componentWillReceiveProps组件接收到新的props时调用，并将新参数作为参数nextProps使用。
5. 修改父组件中的state（isColorReversed）会导致无限刷新的问题，

方案一：进行判断如果本次index和上次一样则不修改，state中会增加preIndex及记录preIndex的逻辑

方案二：将数据放到父级（xuanmiao.js）中，但是大量的逻辑处理也会随之迁到父级，

最终解决方案：

使用shouldComponentUpdate的nexState

1. 我需要图片加载完成再加载组件，将组件细分并设置state。
2. key值不能用随机值
3. 定时器动画频繁的执行导致queueanim动画不能再次执行
   1. 定时器动画最好单独封装成组件,防止影响到其他组件
4. Parallax会导致在行间设置一系列属性导致鼠标事件无法使用，元素变为相对定位，width被改变等。需要再次在行间设置来修复。
5. fetch使用
   1. get请求的时候  
      fetch('MYURL').  
      then(res=>res.json()).  
      then(json=>dispatch(myActionName(json))).  
      catch(err=>dispatch(myErrActionName(err)));
   2. **POST请求的时候**fetch(‘URL’,  
      {  
      method:'POST',  
      headers: {   
      'Accept': 'application/json',  
      'Content-Type': 'application/json'   
      },  
      body:JSON.stringify({'data':json})  
      }  
      )  
      .then(res => res.json())  
      .then(json =>dispatch(receivePosts(json)))  
      .catch(error=>dispatch(requestExceptions(error)));  
      JSON.stringify这个很重要，否则我们的数据无法被识别为json会很痛苦。
6. 使用Masonry解决瀑布流响应式
7. 如何在redux中使用immutable
8. phpstorm 切换大小写, ctrl + shift + U