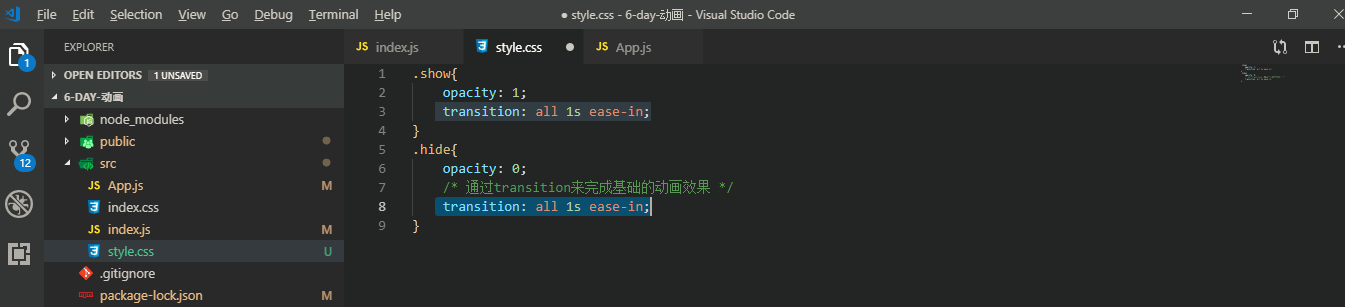
React动画的使用

1. 借助于css的动画属性完成基础的动画

transition: all 1s ease-in;



1. 使用@keyframes实现动画效果

.show{

opacity: 1;

transition: all 1s ease-in;

}

.hide{

/\* css3中使用动画 通过keyframes来设置 通过animation来引用 切记 如果animation中不设定forwards时

那么动画的过程会体现处理，但是最后的结果会是动画又回归到初始化状态，

因此当设置forwards时动画就不会回归到初始化状态 \*/

animation: hide-item 2s ease-in forwards;

}

@keyframes hide-item{

0% {

opacity: 1;

color: brown

}

50% {

opacity: 0.5;

color: aquamarine

}

100% {

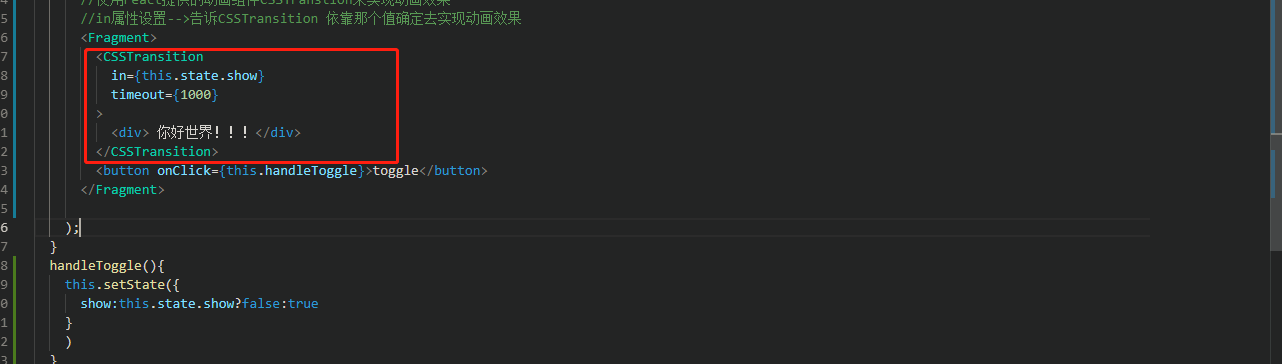
opacity: 0;

color: blue

}

}

1. 使用react-transition-group(react提供)来实现动画
2. 项目中使用npm install react-transition-group --save进行依赖包的安装
3. 使用<CSSTransition></CSSTransition>标签包裹想要加载动画的html代码模块。并设置相关的属性值。



in属性设置-->告诉CSSTransition 依靠那个值确定去判断是否实现动画效果

timeout属性设置-->动画持续的时间(单位是毫秒)其本质上是向包裹的html中自动添加样式

className属性设置-->react并没有将动画的特效做死，留有可供用户自定设置样式特效的空间。实现方法就是通过className=’fade’这样命名，同时在样式文件中重写样式属性即可。

classNames="fade" applies fade-enter, fade-enter-active, fade-enter-done, fade-exit, fade-exit-active, fade-exit-done, fade-appear, and fade-appear-active. Each individual classNames can also be specified independently like:

classNames={{

appear: 'my-appear',

appearActive: 'my-active-appear',

enter: 'my-enter',

enterActive: 'my-active-enter',

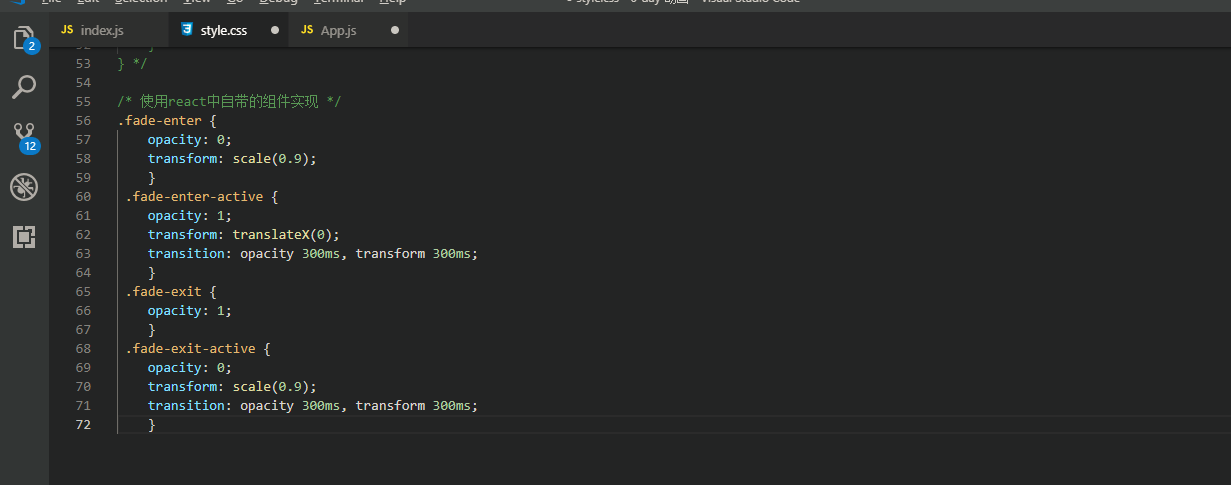
enterDone: 'my-done-enter',

exit: 'my-exit',

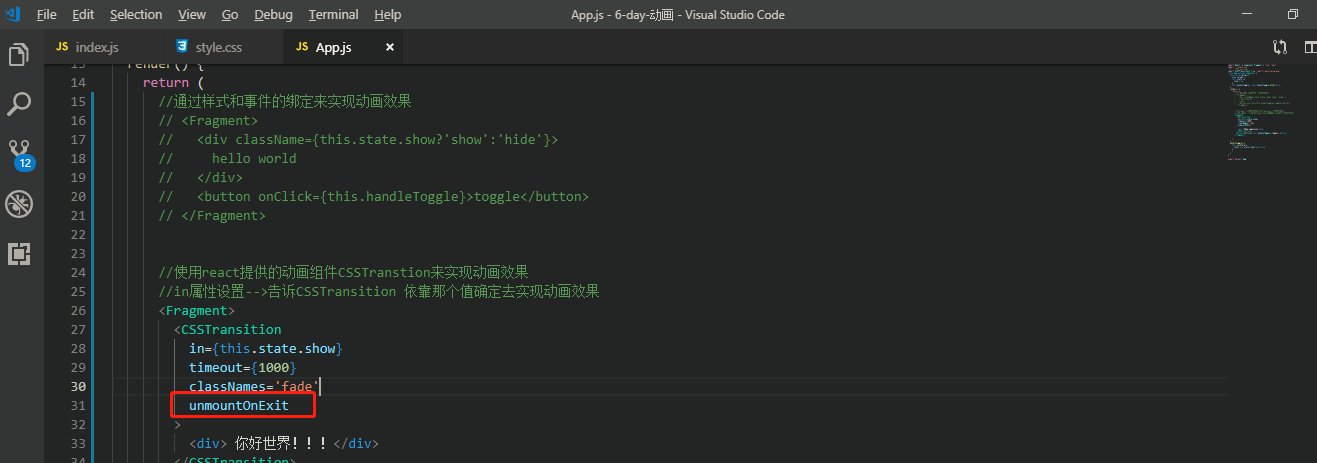
exitActive: 'my-active-exit',

exitDone: 'my-done-exit',}}





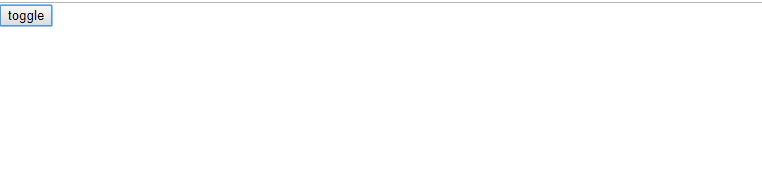
unmountOnExit属性设置-->移除demo 设置此后dome节点会在html中去除



效果图



效果图



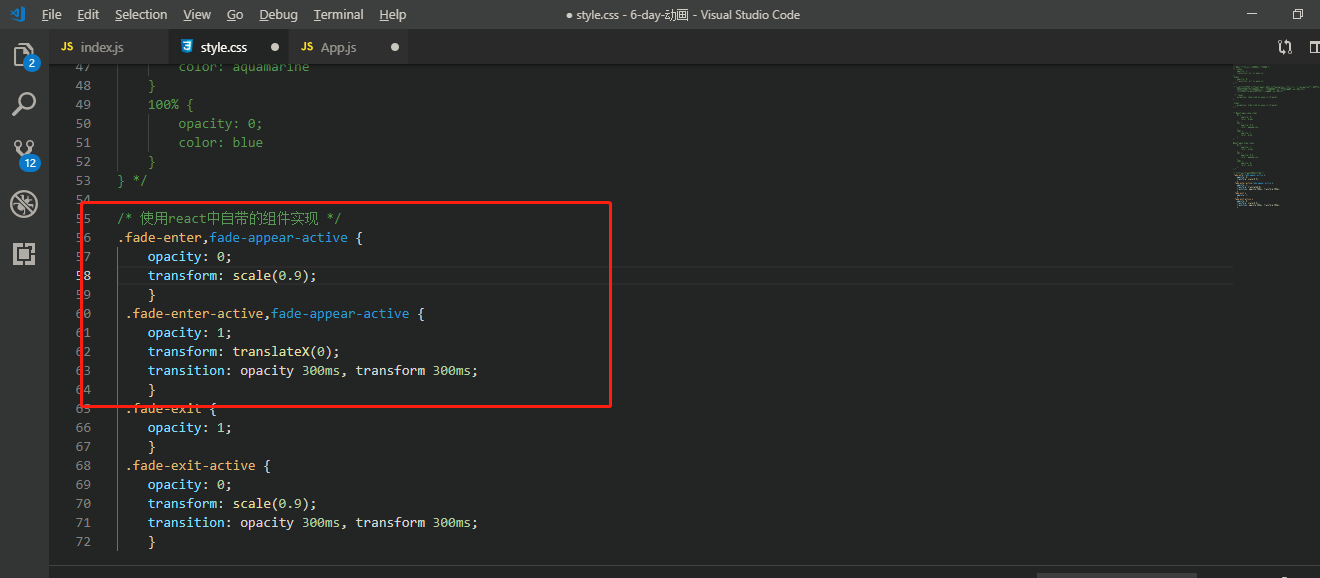
onEntered函数-->是一个生命周期函数，当入场动画执行完毕之后会去触发的函数。这个函数可以接受一个参数el，el指的就是包裹的标签元素。通过js直接实现样式操作

onEntered = {(el) =>{el.style.color='red'}}



appear属性设置-->当appear={true}设定为true时说明被样式组件包裹的区域第一次加载实现时就要有动画效果，同时在样式文件里添加fade-appear-active





1. 在一组标签中使用react动画
2. 项目中使用npm install react-transition-group --save进行依赖包的安装
3. 引入依赖之后，在每个标签上包裹二中描述的CSSTransition

