

Demo.vue

<template>

    <!--界面布局显示内容的地方-->

    <div class="demo">

        当文件组件

    </div>

</template>

<style>

/\*当前界面的样式\*/

.demo{

    font-size: 24px;

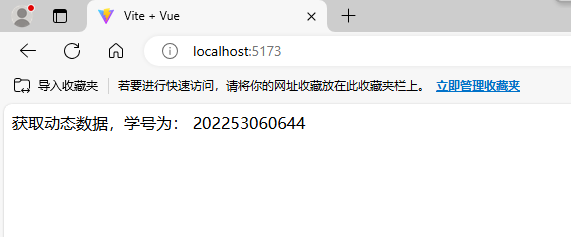
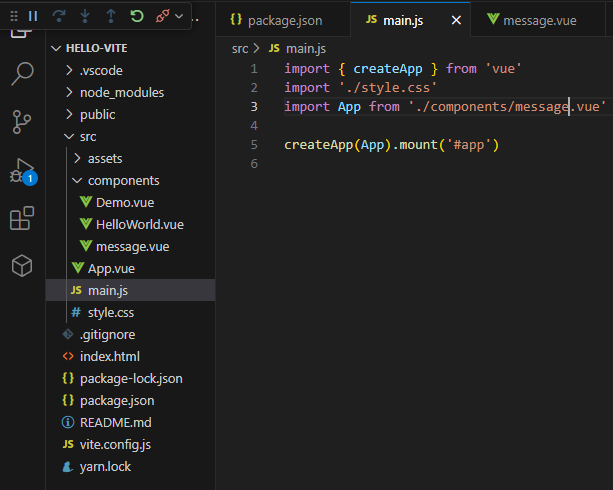
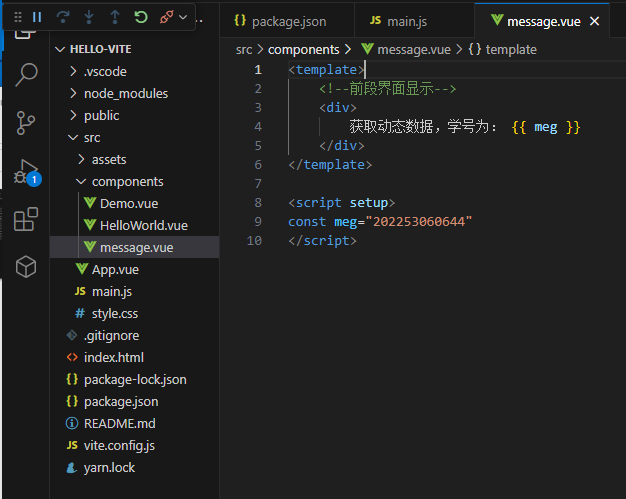
}

</style>

<script setup>

//当前界面中行为和事件

</script>



Message.vue

<template>

    <!--前段界面显示-->

    <div>

        获取动态数据，学号为： {{ meg }}

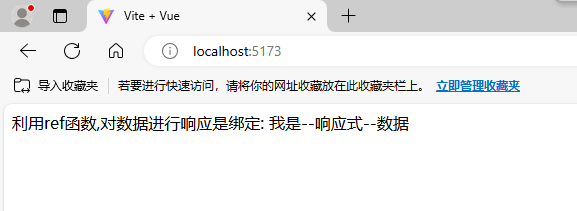
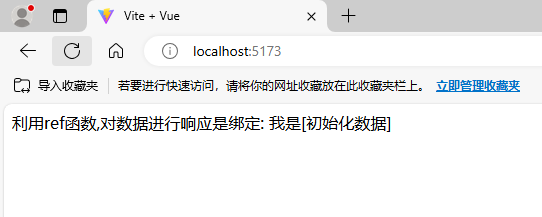
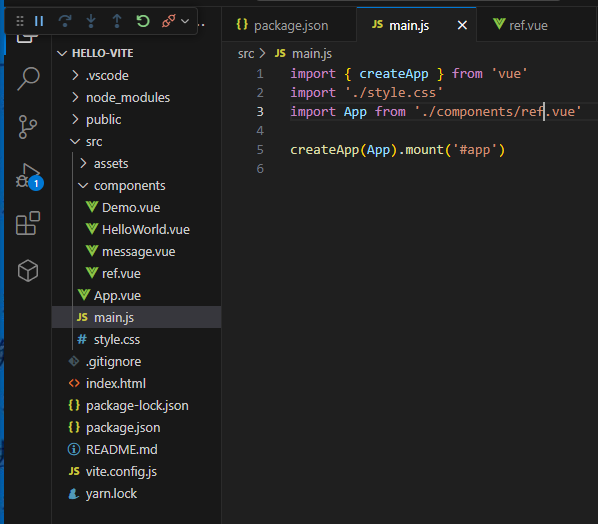
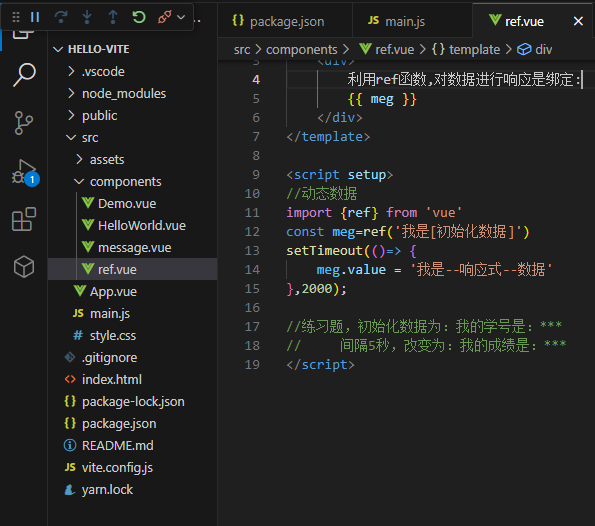
    </div>

</template>

<script setup>

const meg="202253060644"

</script>



Ref.vue

<template>

    <!--显示内容-->

    <div>

        利用ref函数,对数据进行响应是绑定:

        {{ meg }}

    </div>

</template>

<script setup>

//动态数据

import {ref} from 'vue'

const meg=ref('我是[初始化数据]')

setTimeout(()=> {

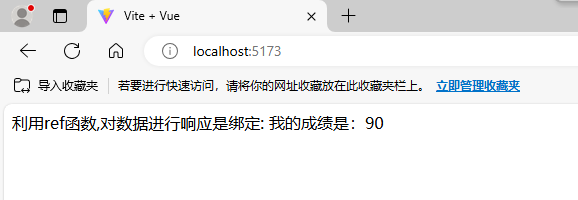
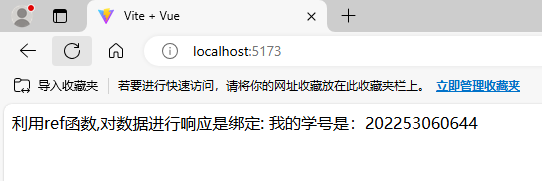
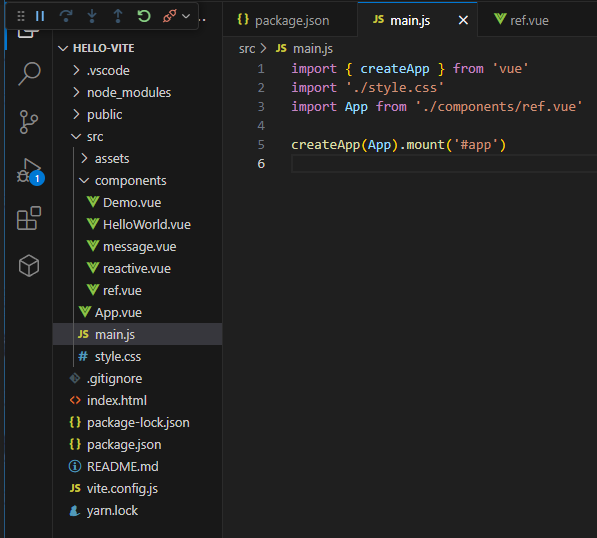
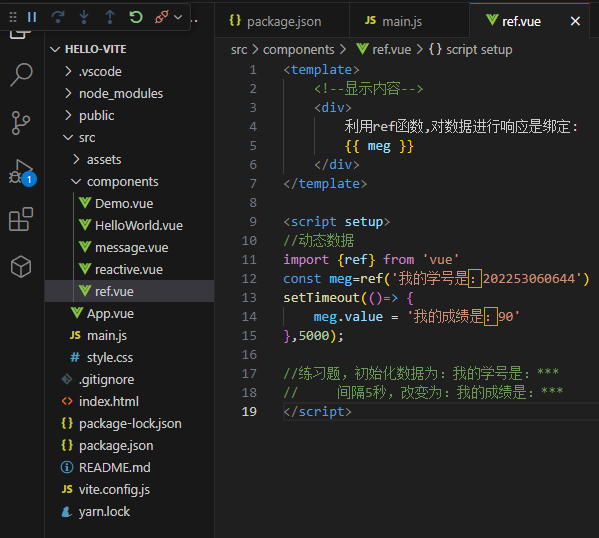
    meg.value = '我是--响应式--数据'

},2000);

//练习题，初始化数据为：我的学号是：\*\*\*

//     间隔5秒，改变为：我的成绩是：\*\*\*

</script>



Ref.vue

<template>

    <!--显示内容-->

    <div>

        利用ref函数,对数据进行响应是绑定:

        {{ meg }}

    </div>

</template>

<script setup>

//动态数据

import {ref} from 'vue'

const meg=ref('我的学号是：202253060644')

setTimeout(()=> {

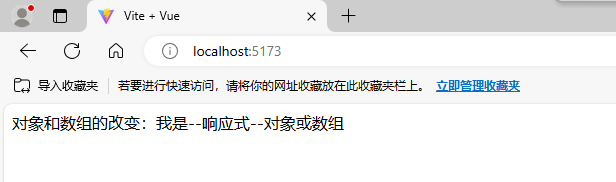
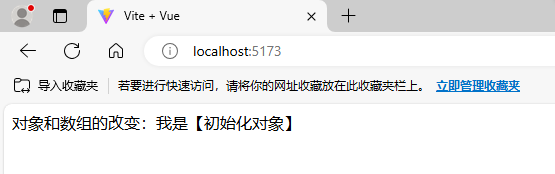
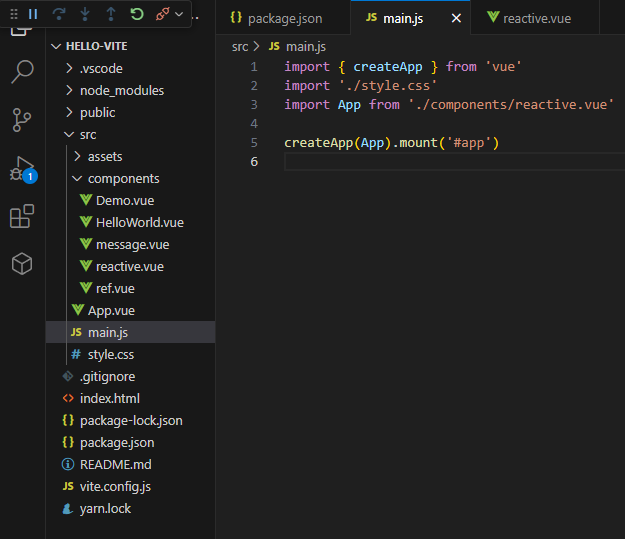
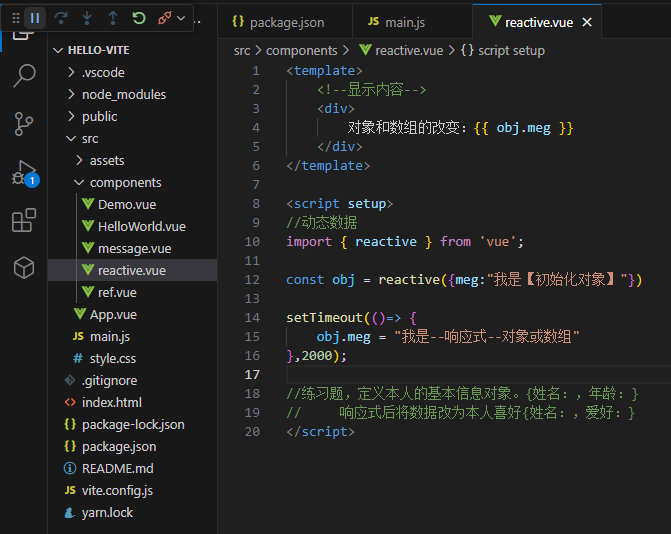
    meg.value = '我的成绩是：90'

},5000);

//练习题，初始化数据为：我的学号是：\*\*\*

//     间隔5秒，改变为：我的成绩是：\*\*\*

</script>



Reactive.vue

<template>

    <!--显示内容-->

    <div>

        对象和数组的改变：{{ obj.meg }}

    </div>

</template>

<script setup>

//动态数据

import { reactive } from 'vue';

const obj = reactive({meg:"我是【初始化对象】"})

setTimeout(()=> {

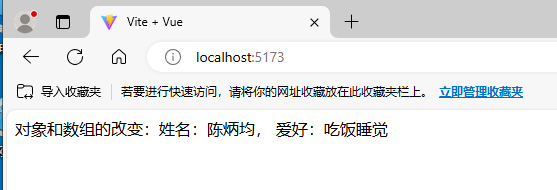
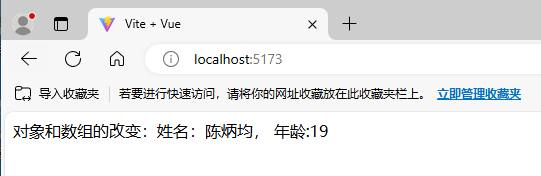
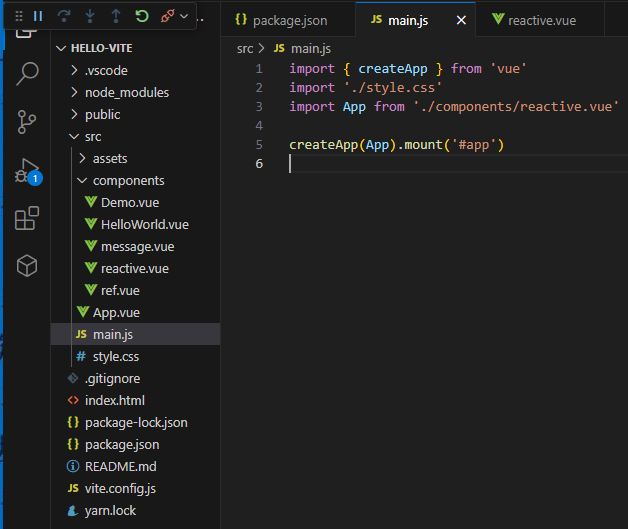
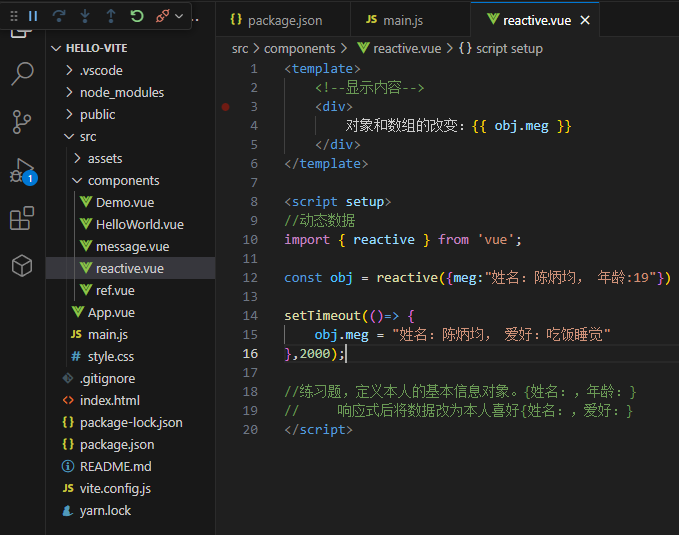
    obj.meg = "我是--响应式--对象或数组"

},2000);

//练习题，定义本人的基本信息对象。{姓名：，年龄：}

//     响应式后将数据改为本人喜好{姓名：，爱好：}

</script>



Reactive.vue

<template>

    <!--显示内容-->

    <div>

        对象和数组的改变：{{ obj.meg }}

    </div>

</template>

<script setup>

//动态数据

import { reactive } from 'vue';

const obj = reactive({meg:"姓名：陈炳均， 年龄:19"})

setTimeout(()=> {

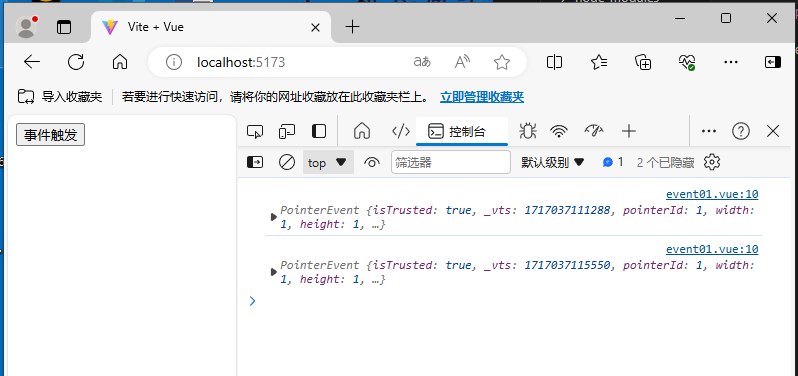
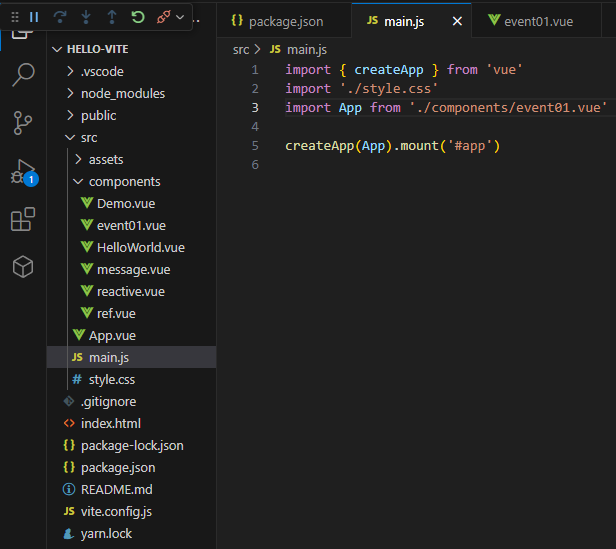
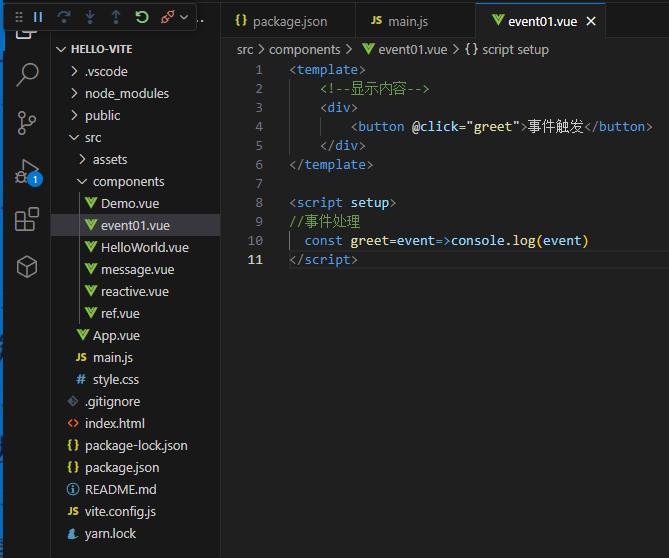
    obj.meg = "姓名：陈炳均， 爱好：吃饭睡觉"

},2000);

//练习题，定义本人的基本信息对象。{姓名：，年龄：}

//     响应式后将数据改为本人喜好{姓名：，爱好：}

</script>



Event01.vue

<template>

    <!--显示内容-->

    <div>

        <button @click="greet">事件触发</button>

    </div>

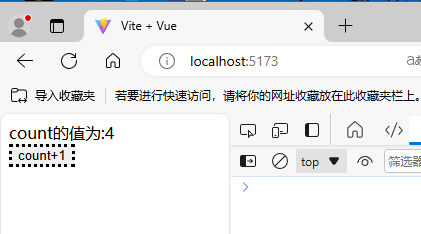
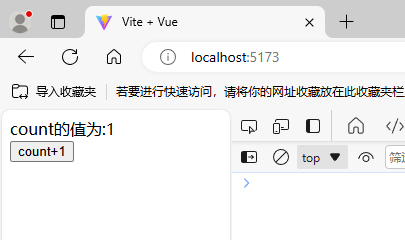
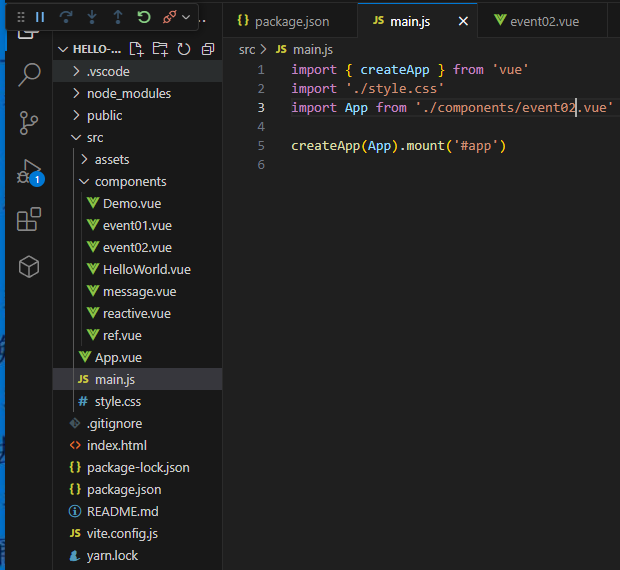
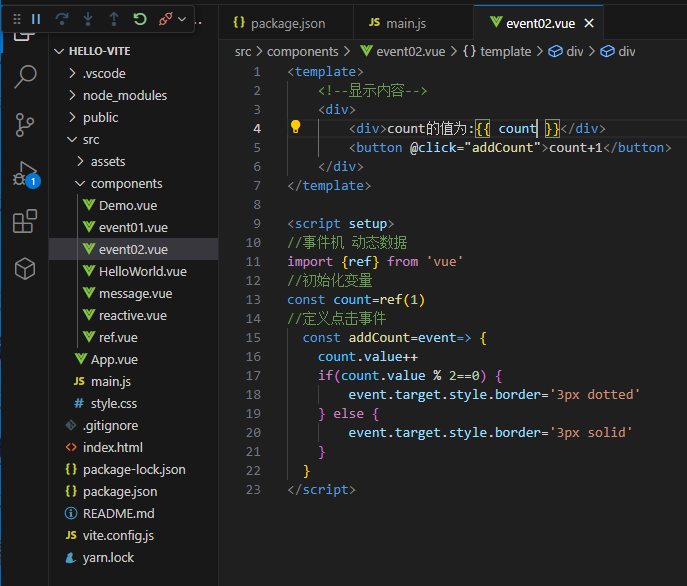
</template>

<script setup>

//事件处理

  const greet=event=>console.log(event)

</script>



Event02.vue

<template>

    <!--显示内容-->

    <div>

        <div>count的值为:{{ count }}</div>

        <button @click="addCount">count+1</button>

    </div>

</template>

<script setup>

//事件机 动态数据

import {ref} from 'vue'

//初始化变量

const count=ref(1)

//定义点击事件

  const addCount=event=> {

    count.value++

    if(count.value % 2==0) {

        event.target.style.border='3px dotted'

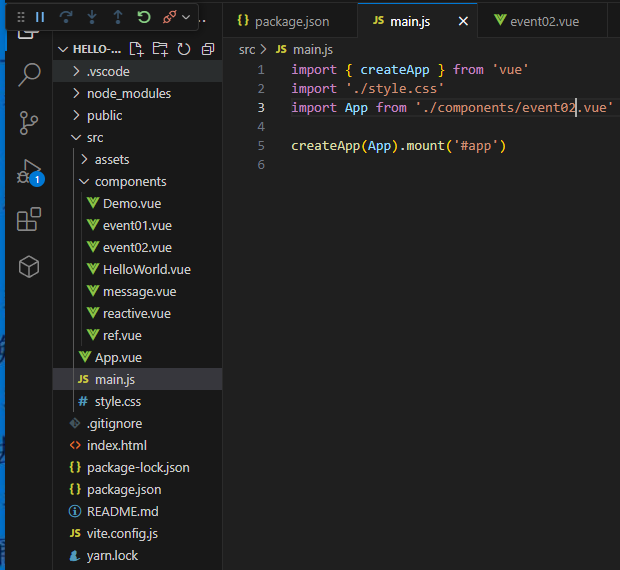
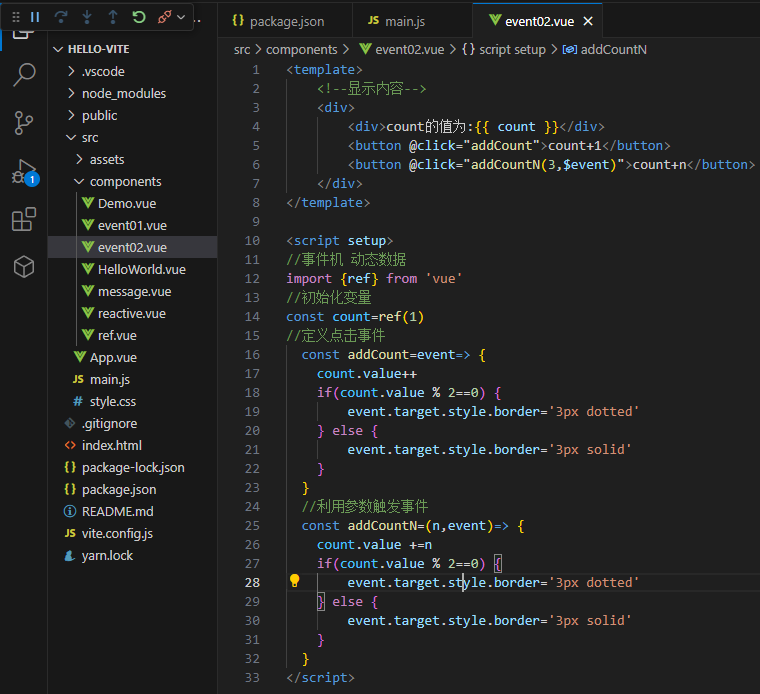
    } else {

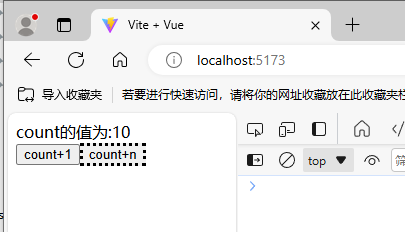
        event.target.style.border='3px solid'

    }

  }

</script>





Event02.vue

<template>

    <!--显示内容-->

    <div>

        <div>count的值为:{{ count }}</div>

        <button @click="addCount">count+1</button>

        <button @click="addCountN(3,$event)">count+n</button>

    </div>

</template>

<script setup>

//事件机 动态数据

import {ref} from 'vue'

//初始化变量

const count=ref(1)

//定义点击事件

  const addCount=event=> {

    count.value++

    if(count.value % 2==0) {

        event.target.style.border='3px dotted'

    } else {

        event.target.style.border='3px solid'

    }

  }

  //利用参数触发事件

  const addCountN=(n,event)=> {

    count.value +=n

    if(count.value % 2==0) {

        event.target.style.border='3px dotted'

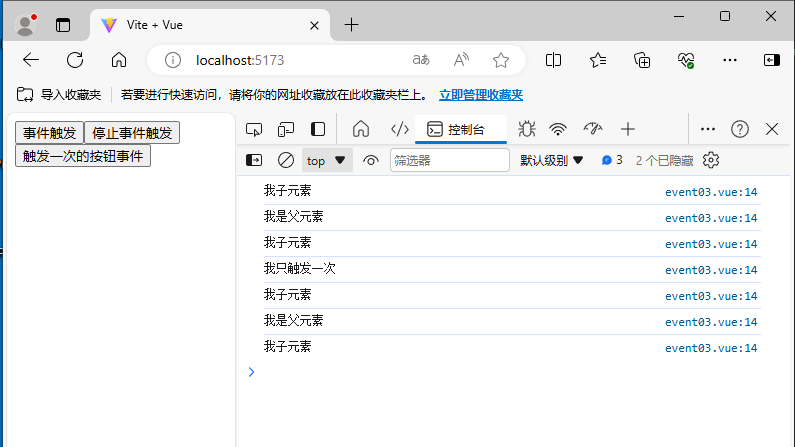
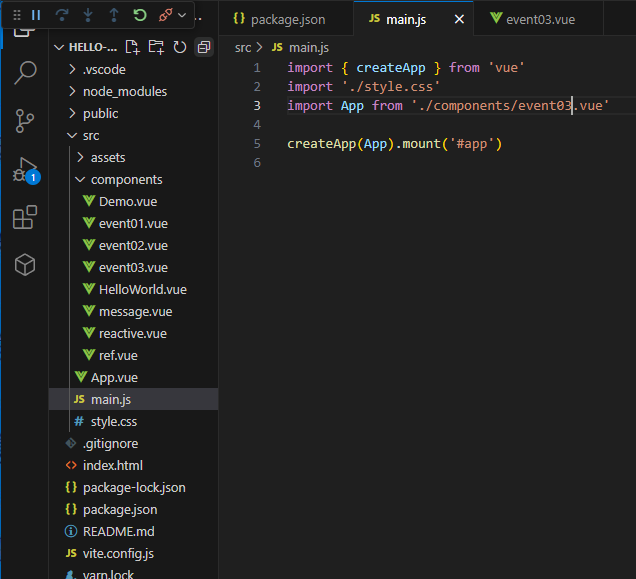
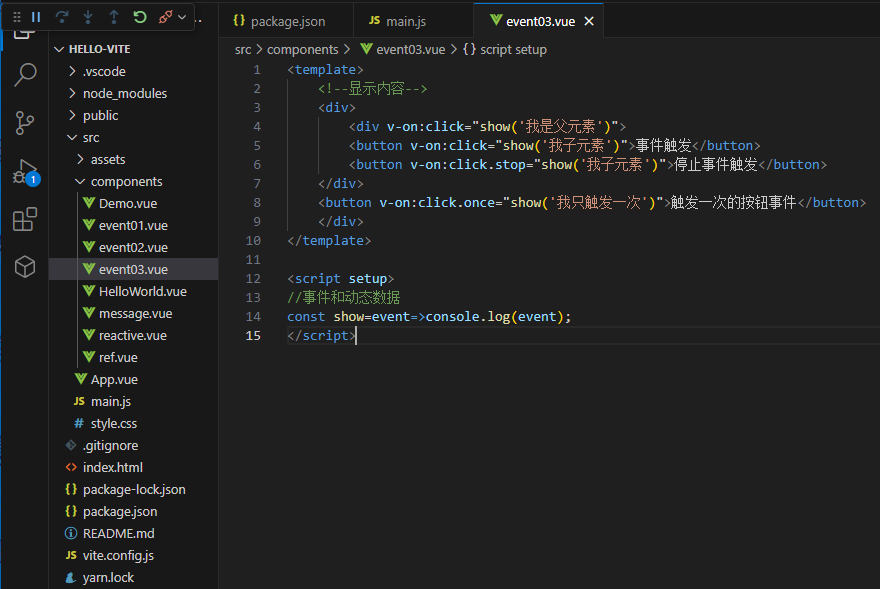
    } else {

        event.target.style.border='3px solid'

    }

  }

</script>



Event03.vue

<template>

    <!--显示内容-->

    <div>

        <div v-on:click="show('我是父元素')">

        <button v-on:click="show('我子元素')">事件触发</button>

        <button v-on:click.stop="show('我子元素')">停止事件触发</button>

    </div>

    <button v-on:click.once="show('我只触发一次')">触发一次的按钮事件</button>

    </div>

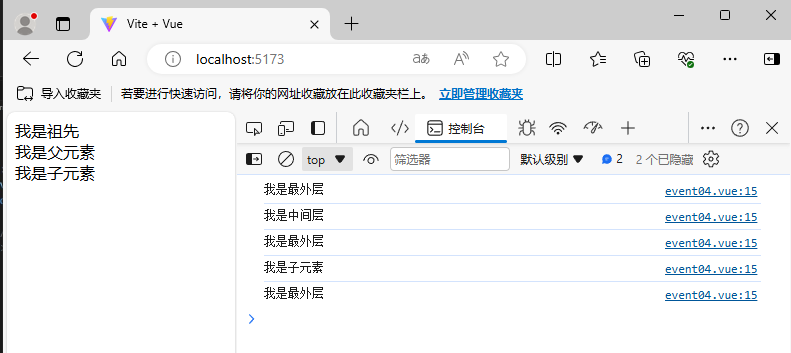
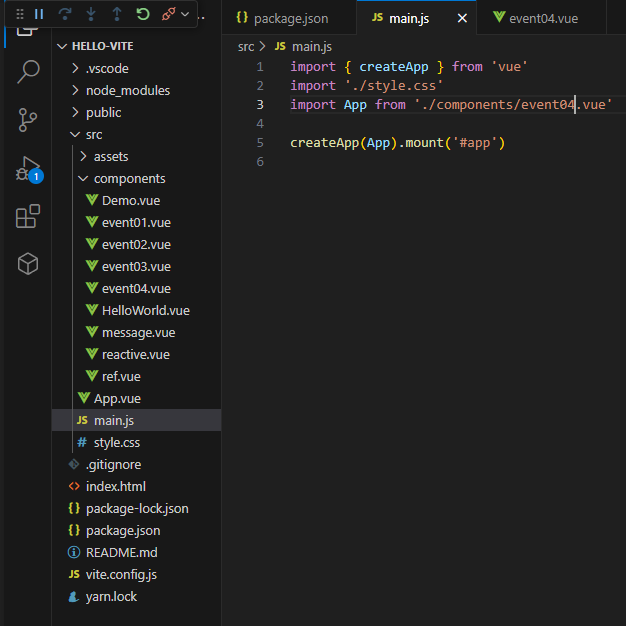
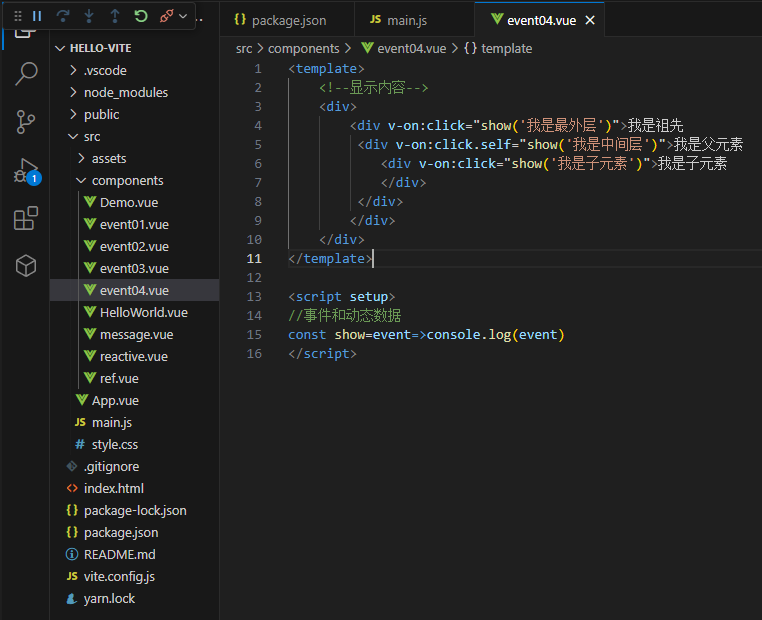
</template>

<script setup>

//事件和动态数据

const show=event=>console.log(event);

</script>



Event04.vue

<template>

    <!--显示内容-->

    <div>

        <div v-on:click="show('我是最外层')">我是祖先

         <div v-on:click.self="show('我是中间层')">我是父元素

            <div v-on:click="show('我是子元素')">我是子元素

            </div>

         </div>

        </div>

    </div>

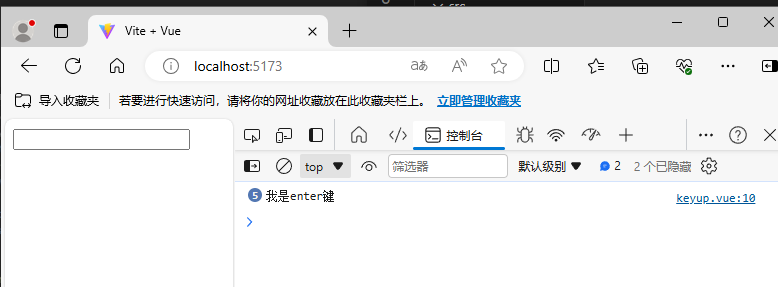
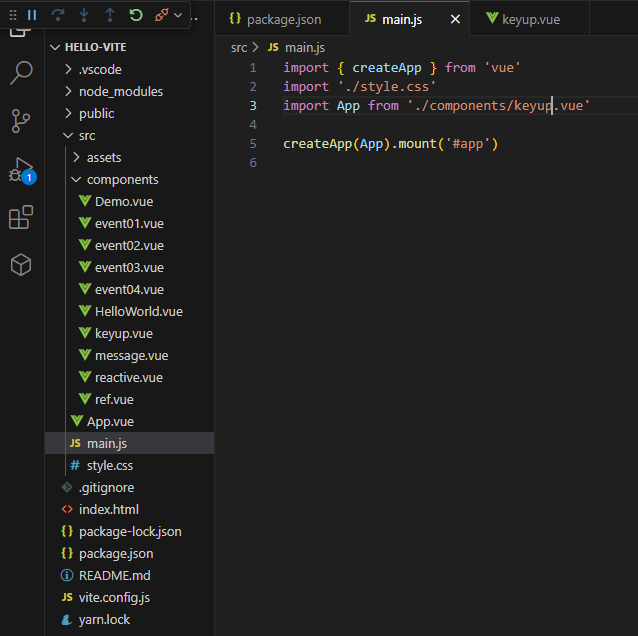
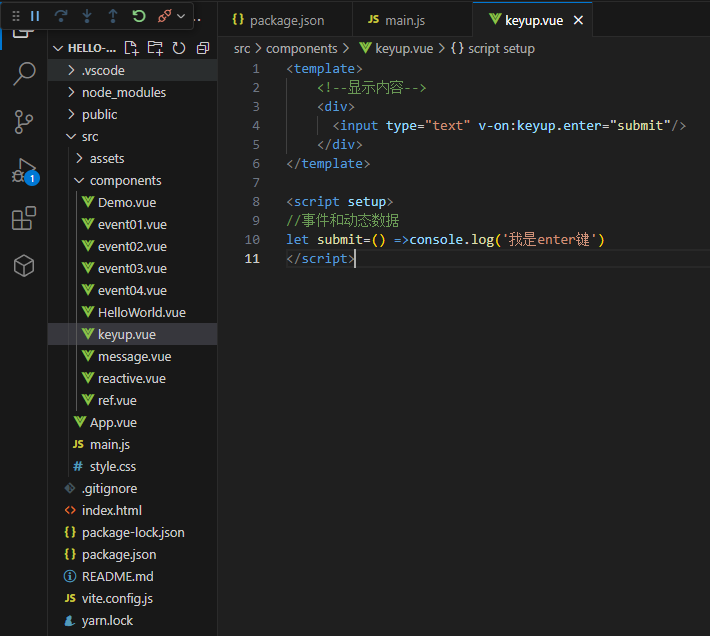
</template>

<script setup>

//事件和动态数据

const show=event=>console.log(event)

</script>



Keyup.vue

<template>

    <!--显示内容-->

    <div>

      <input type="text" v-on:keyup.enter="submit"/>

    </div>

</template>

<script setup>

//事件和动态数据

let submit=() =>console.log('我是enter键')

</script>