

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>

1.

<template>

    <!-- 界面显示内容的地方 -->

    <div>

        我的学号：{{ meg }}

    </div>

</template>

<style>

    /\* 当前界面的样式 \*/

    .demo{

        font-size: 24px;

    }

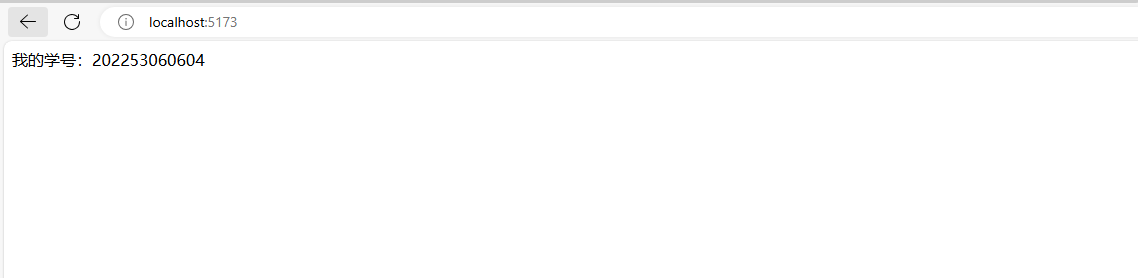
</style>

<script setup>

//当前界面中行为和事件

    const meg="202253060604"

</script>



2. <template>

<!-- 显示内容 -->

    <div>

        {{ meg }}

    </div>

</template>

<style>

</style>

<script setup>

    //动态数据

    import {ref} from 'vue'

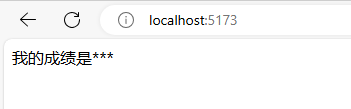
    const meg = ref('我的学号是：202253060604')

    setTimeout(()=>{

        meg.value = '我的成绩是\*\*\*'

    },5000);

</script>



3. <template>

    <div>

        姓名：苏彦霖，{{ obj.meg }}

    </div>

</template>

<style>

</style>

<script setup>

    import{reactive} from 'vue'

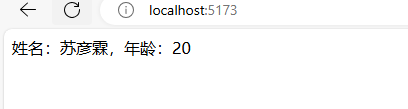
    const obj = reactive({meg:"年龄：20"})

    setTimeout(()=>{

        obj.meg = "爱好：编程"

    },5000)

</script>



4. <template>

    <div>

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

    </div>

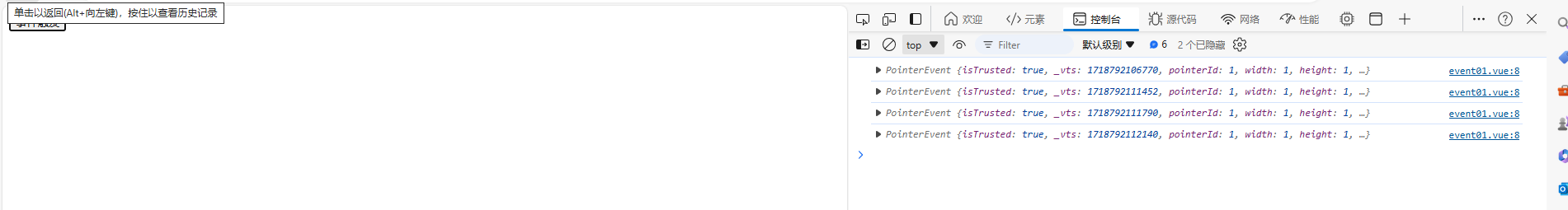
</template>

<script setup>

//时间处理

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

</script>



5. <template>

<div>

    <div>count的值为：{{ count+1 }}</div>

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

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

</div>

</template>

<script setup>

    import {ref} from 'vue'

    const count=ref(1)

    const addCount = event =>{

        count.value++

        if(count.value % 2 == 0){

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

        }else{

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

        }

    }

    const addCountN = (n,event) =>{

        count.value+=n

        if(count.value % 2 === 0){

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

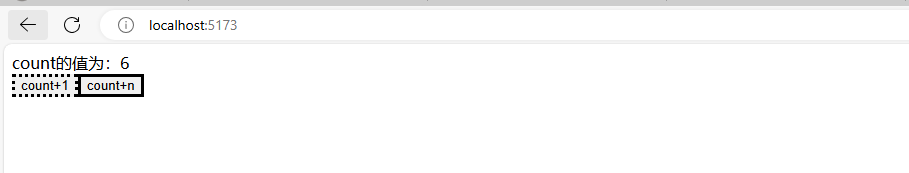
        }else{

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

        }

    }

</script>



6.

<template>

    <div>

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

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

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

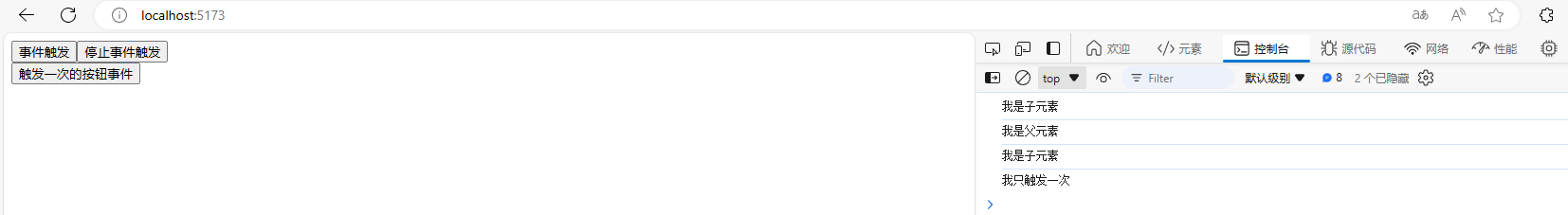
    </div>

</template>

<script setup>

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

</script>



7. <template>

    <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>

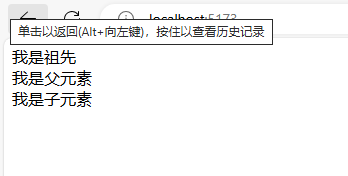
    </div>

</template>

<script setup>

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

</script>



8.

<template>

    <div>

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

    </div>

</template>

<script setup>

let submint=()=>console.log('我是enter建')

</script>

