

18. 组件的自定义事件

学习要点：

1. 自定义事件
2. `.sync` 修饰符

本节课我们来开始学习 **Vue** 的组件自定义事件，子传父的能力。

一. 自定义事件

1. 之前学习过 `v-on` 后面可以接 `:click` 等内置的事件，而组件可以自定义事件；
2. 而自定义事件，可以通过子组件给父组件传递信息的能力；

//下面代码为子组件 `html-a` 的自定义事件

```
template : `

{{name}}

`  
methods : {  
  //通过子组件的 click 事件来执行这个方法  
  //其中 this.$emit 是内置 API，可参考手册  
  //.$emit 表示触发当前实例事件，可附带参数  
  //参数一是自定义事件，不支持类似 props 驼峰写法  
  childClick(name) {  
    this.$emit('child-event', name);  
  }  
}
```

3. 由于事件名不存在大小写自动转换的问题，所以名称要完全匹配；

```
<!--这里 child-event 是子组件的自定义事件-->  
<!--而 parentFn 是父组件的方法，通过这个修改父组件内容-->  
<!--这里是不需要传参的，参数会自动传入，直接接收 即可-->  
<html-a v-on:child-event="parentFn"></html-a>
```

//这里是父组件的方法

```
methods : {  
  parentFn(name) {  
    console.log('父组件得到子组件的信息！' + name);  
    this.message = name;  
  }  
}
```

PS：可以测试使用多个子组件，让其中一个子组件更改父组件后，看其它子组件的变化；

二. .sync 修饰符

1. 还记得之前强行更改父组件通过 props 传给子组件的计数程序么？
2. 我们强行更改 props 值，控制台报错，且父组件的 data 并没有改变；
3. 通过将 props 的值保存子组件属性，再通过 .sync 修饰符更改父组件；

//这里是子组件部分

```
data() {  
  return {  
    childCount : this.count  
  }  
},  
template : `  
  <div>  
    <button v-on:click="childClick">{{childCount}}</button>  
  </div>  
`,  
methods : {  
  childClick() {  
    //this.count++  
    this.$emit('update:count', this.childCount++);  
  }  
},  
props : ['count']
```

//sync 修饰符修改父组件原始值

```
<html-a v-bind:count.sync="count"></html-a>
```

PS: 由于单向数据流的概念, 让维护清晰; 所以 vue2.0 时一度删除 .sync 修饰符; 虽然在 2.3.0+ 版本又引入回来, 使用的时候还是要格外小心, 避免混乱。