

Vue传值--三种常用传值

[callback](#) 发布于 2018-07-23

总结

先写个总结，方便自己复习。嘿嘿。

- 父组件传值给子组件，首先父组件发送的形式是用bind(用缩写:)绑定值到子组件身上。然后子组件用属性props接收。
- 子组件传值父组件，首先是子组件发送的形式是this.\$emit(事件名,需要发送的值)，**事件名：父组件on监听的方法**，父组件第一件事就是监听子组件发送过来的事件名，@事件名=获取值的方法。**获取值的方法（值）***，就这样拿到了儿子发过来的值
- 兄弟传值，两个都要引入公共js->bus。首先是发送者，this.\$emit(名字，值)。然后是接受者,this.\$on("传过来的那个名字", 函数（值）)，这里要注意的是接受者的这个监听事件应该是放在钩子mounted事件内。同时注意this指向问题。（可用封装方法）

父组件向子组件进行传值：

父组件：

```
<template>
<div>
  父组件：
```

```

    <input type="text" v-model="name">
    <br>
    <br>
    <!-- 引入子组件 -->
    <child :inputName="name"></child>
  </div>
</template>
<script>
  import child from './child'
  export default {
    components: {
      child
    },
    data () {
      return {
        name: ''
      }
    }
  }
</script>

```

子组件:

```

<template>
  <div>
    子组件:
    <span>{{inputName}}</span>
  </div>
</template>
<script>
  export default {
    // 接受父组件的值
    props: {

```

```
        inputName: String,
        required: true
    }
}
</script>
```

子组件向父组件传值

子组件：

```
<template>
  <div>
    子组件：
    <span>{{childValue}}</span>
    <!-- 定义一个子组件传值的方法 -->
    <input type="button" value="点击触发"
  </div>
</template>
<script>
  export default {
    data () {
      return {
        childValue: '我是子组件的数据'
      }
    },
    methods: {
      childClick () {
        // childByValue是在父组件on监听的方法
        // 第二个参数this.childValue是需要传
        this.$emit('childByValue', this.c
      }
    }
  }
}
```

```
    }  
  }  
</script>
```

####父组件:

```
<template>  
<div>  
  父组件:  
  <span>{{name}}</span>  
  <br>  
  <br>  
  <!-- 引入子组件 定义一个on的方法监听子组件的状态-->  
  <child v-on:childByValue="childByValue"></ch:  
</div>  
</template>  
<script>  
  import child from './child'  
  export default {  
    components: {  
      child  
    },  
    data () {  
      return {  
        name: ''  
      }  
    },  
    methods: {  
      childByValue: function (childValue) {  
        // childValue就是子组件传过来的值  
        this.name = childValue  
      }  
    }  
  }  
</script>
```

```
}  
</script>
```

非父子组件进行传值。

(非父子组件之间传值，需要定义个公共的公共实例文件 bus.js，作为中间仓库来传值，不然路由组件之间达不到传值的效果。)

公共bus.js

```
//bus.js  
import Vue from 'vue'  
export default new Vue()
```

组件A:

```
<template>  
  <div>  
    A组件:  
    <span>{{elementValue}}</span>  
    <input type="button" value="点击触发" @click="  
  </div>  
</template>  
<script>  
  // 引入公共的bus，来做为中间传达的工具  
  import Bus from './bus.js'  
  export default {  
    data () {
```

```
    return {
      elementValue: 4
    }
  },
  methods: {
    elementByValue: function () {
      Bus.$emit('val', this.elementValue)
    }
  }
}
</script>
```

组件B:

```
<template>
  <div>
    B组件:
    <input type="button" value="点击触发" @cli
    <span>{{name}}</span>
  </div>
</template>
<script>
  import Bus from './bus.js'
  export default {
    data () {
      return {
        name: 0
      }
    },
    mounted: function () {
```

```
var vm = this
// 用$on事件来接收参数
Bus.$on('val', (data) => {
  console.log(data)
  vm.name = data
})
},
methods: {
  getData: function () {
    this.name++
  }
}
}
</script>
```