

16. 组件 Prop 通信

学习要点：

1. 实际问题
2. 数据父传子

本节课我们来开始学习 **Vue3.x** 的组件中父组件给子组件传递数据的方法。

一. 实际问题

1. 根组件(父组件)的 **data()** 数据，子组件在目前的知识是无法获取的；
2. 我们如果想要子组件和父组件同时正确的获取数据，采用如下方法：

```
// 定义一个局部组件
const htmlA = {
  // 数据
  data() {
    return {
      message : '子组件-Vue3.x~'
    }
  },

  // 模板
  template : `
    <div>{{message}}</div>
  `
}

// 创建一个Vue 应用实例，这个本身也是一个组件，根组件，最顶层的
const app = Vue.createApp({
  data() {
    return {
      message : '父组件-Vue3.x~'
    }
  },

  // 创建一个局部组件，也是一个子组件
  components : {
    'html-a' : htmlA
  }
})

<div id="app">
  <html-a></html-a>
</div>
```

PS: 上面的例子中暴露的问题，就是子组件无法直接使用父组件的 `message` 属性；

PS: 所以只能在 HTML 端进行设计，但这样做，父组件的内容无法和子组件参与逻辑；

```
<div id="app">
  <div>{{message}}</div>
  <html-a></html-a>
</div>
```

二. 数据父传子

1. 根据上面的问题，我们最好是能够直接从子组件获取到父组件的数据内容；
2. 这时，Vue3.x 提供了一个 `props` 来获取父组件给子组件传递的值；

// 定义一个局部组件

```
const htmlA = {
  // 数据
  data() {
    return {
      message : '子组件-Vue3.x~'
    }
  },

  // props
  props : ['abc'],

  // 模板
  template : `
    <div>{{abc}}</div>
    <div>{{message}}</div>
  `
}
```

```
<div id="app">
  <html-a v-bind:abc="message"></html-a>
</div>
```

PS: `props` 里定义数组，表示可以定义多个；在 HTML 端使用 `v-bind:abc` 来获取父组件对应值；

PS: 这里采用了 `abc` 表示 `props` 的元素名称可以自定义，如果定义成 `parent-message`，则：

```
props : ['parentMessage'],
```

// 模板

```
template : `
  <div>{{parentMessage}}</div>
  <div>{{message}}</div>
`
```

```
<html-a v-bind:parent-message="message"></html-a>
```