

# Vue.js 组件——动手实验

主讲人：和凌志

# 主要内容

---

- 全局组件注册
- 局部组件注册
- 父组件给子组件传值
- props 单向数据绑定

---

# 组件的注册与调用

## (1) 全局组件

```
<div id="container1">
  <my-component-1></my-component-1>
</div>

<div id="container2">
  <my-component-1></my-component-1>
</div>

<script>
  Vue.component('my-component-1', {
    template: '<div>hello world</div>'
  })
  new Vue({
    el: '#container1',
  })
</script>
```

## 全局组件（完整代码）

```
<div id="container1">
  <my-component-1></my-component-1>
</div>

<div id="container2">
  <my-component-1></my-component-1>
</div>

<script>
  Vue.component('my-component-1', {
    template: '<div>hello world</div>'
  })
  new Vue({
    el: '#container1',
  })
</script>
```

## (2) 局部组件

这是子组件A  
这是子组件B

```
<div id="container1">  
  <child-component-a></child-component-a>  
  <child-component-b></child-component-b>  
</div>
```

```
<script>  
  var ComponentA = {  
    template: '<div>这是子组件A</div>'  
  }  
  var ComponentB = {  
    template: '<div>这是子组件B</div>'  
  }  
  new Vue({  
    el: '#container1',  
    components: {  
      'child-component-a': ComponentA,  
      'child-component-b': ComponentB,  
    }  
  })
```

## 局部组件（完整代码）

```
<div id="container1">  
  <child-component-a></child-component-a>  
  <child-component-b></child-component-b>  
</div>
```

```
<script>  
  var ComponentA = {  
    template: '<div>这是子组件A</div>'  
  }  
  var ComponentB = {  
    template: '<div>这是子组件B</div>'  
  }  
  new Vue({  
    el: '#container1',  
    components: {  
      'child-component-a': ComponentA,  
      'child-component-b': ComponentB,  
    }  
  })  
</script>
```

---

## 组件的注册与调用



### ( 3 ) 学会使用 <template> 标签

这是 Template 片段

```
<div id="container1">
  <component1></component1>
</div>

<template id="myComponent">
  <div> 这是 Template 片段</div>
</template>

<script>
  new Vue({
    el: '#container1',
    components: {
      'component1': { template: '#myComponent' }
    }
  })
</script>
```

## 学会使用 <template> 标签

```
<div id="container1">
  <component1></component1>
</div>
<template id="myComponent">
  <div> 这是 Template 片段</div>
</template>
```

```
<script>
  new Vue({
    el: '#container1',
    components: {
      'component1': { template: '#myComponent' }
    }
  })
</script>
```

---

# 理解使用 props

## props 父组件给子组件传值

- 父组件的数据如何传给子组件呢？可以使用props把数据传给子组件。
- 注意：在子组件中定义props时，使用了camelCase命名法。由于HTML特性不区分大小写，camelCase的prop用于特性时，会转为短横线隔开
- 在props中定义的myName,在用作特性时需要转换为 my-name

## props 传值

名字： 张三

年龄： 20

```
<div id="container1">
  <component1 :my-name="name" :my-age="age"></component1>
</div>

<template id="myComponent">
  <div>
    <p> 名字: {{myName}} </p>
    <p> 年龄: {{myAge}} </p>
  </div>
</template>

<script>
  new Vue(
    {
      el: '#container1',
      data: {
        name: '张三',
        age: 20
      },
      components: {
        'component1': {
          template: '#myComponent',
          props: ['myName', 'my-age']
        }
      }
    }
  )
</script>
```

## props 父组件给子组件传值

```
<div id="container1">
  <component1
    :my-name="name"
    :my-age="age">
  </component1>
</div>
<template id="myComponent">
  <div>
    <p> 名字 : {{myName}} </p>
    <p> 年龄 : {{myAge}} </p>
  </div>
</template>
```

```
<script>
  new Vue(
    {
      el: '#container1',
      data: {
        name: '张三',
        age: 20
      },
      components: {
        'component1': {
          template: '#myComponent',
          props: ['myName', 'my-age']
        }
      }
    })
</script>
```

# 理解 props 的单向数据绑定

既然父组件使用props把数据传给了子组件，那么如果子组件修改了数据，对父组件是否有影响呢？

父组件数据		
name	张三	<input type="text" value="张三"/>
age	20	<input type="text" value="20"/>

子组件数据		
myName	张三	<input type="text" value="张三"/>
myAge	20	<input type="text" value="20"/>

代码参考：[vue-component-05.html](#)

# 课程预告

---

## ■ Element-UI 的引入