今天学习的内容

1:复习上周知识重点

2:今天的目标

2.1:组件

2.2:组件之间数据传递与通讯(重点&难点)

2.3:组件之间通信 (父组件-->(发送)数据->子组件)

步骤

(1)父组件在调用子组件 (传值)

<child :myValue="money"></child>

(2)子组件中获取父组件传来的值

Vue.component("child",{

props:["myValue"], //声明:父组件传递参数名

template:`<div>{{myValue}}</div>`

});

练习:创建父组件 my-login (表单) tips="用户名"

创建子组件 my-input (输入域) placeholder="tips"

练习:创建父组件 parent car="myCar" myCar="QQ"

创建子组件 child <input value="父元素数据qq">

2.4:组件之间通信 (子组件-->数据->父组件)

通过事件的方式来完成数据的传输

(1)在父组件中先定义一个方法，用于接收子组件通过事件发送

来的数据.

methods:{

getData:function(msg){

//参数msg:就是子组件通过事件传递过来的数据

}

}

(2)在父组件模板:绑定事件处理函数

<child @自定义事件名="方法名"></child>

示例:

<child @dataEvent="getData"></child>

(3)在子组件中触发事件,并且传递数据

this.**$emit**("触发事件名",传递数据);

示例:

this.$emit("dataEvent","交话费");

练习:14:26--14:36

创建父组件

ul>li 循环显示数组中数据

ul 条件如果isUserLogin为true添加false 删除

创建方法 getData 接收子组件传递的数据

修改isUserLogin {true/false}

创建子组件

创建方法setData 触发事件并且传递true/false给父组件

创建模块:触发点击事件执行setDat方法

2.5:组件之间通信 ($parent;$refs)

父组件要想获取子组件的(数据)

(1)在父组件调用子组件时候，指定属性ref

<child ref="子组件变量名"></child> //表示子组件对象

示例:

<chlid ref="mySon"></child>

(2)根据指定名称，找到子组件实例对象

this.**$refs**.mySon

子组件想获取父组件数据

(1) this.**$parent**

注意事项:需要获取数据保存在data属性，使用本方法

不在data中使用上面二个方法

练习:

父组件:模板显示列表(用户名称)

data ["tom","jerrry"]

子组件:数据当前用户名称

显示当前用户名

输入框 <input type="text" value="james">

<button>添加</button>

练习:16:35--16:45

父组件:模板显示列表(用户名称)<my-parent>

data ["tom","jerrry"]

<my-child userName="a"></my-child>

<my-child userName="b"></my-child>

子组件:数据当前用户名称 <my-child>

props:["userName"]

显示当前用户名

输入框 <input type="text" value="james">

<button>添加</button>

a:123

b:123

2.6:组件之间通信 (兄弟之间)

借助于一个公共vue实例对象，不同的组件可以通过该对象

完成事件绑定和触发.

var bus = new Vue();

bus.**$on**(); //绑定事件

bus.**$emit**(); //发送事件

熊大(发)想发消息给熊二(收)

接收方(熊二):事件绑定

bus.$on("cutomeEvent",function(msg){

//msg就是通过事件，传递来的数据

});

发送方(熊大):触发事件

bus.$emit("cutomeEvent ","....");