复习：

见思维导图

HTML标签、属性、事件名称的大小写：

HTML标签名：不分区大小写，推荐纯小写；官方标签只包含小写字母/数字，自定义标签只包含小写字母/数字+至少一个减号

<textarea></texterea>

<buy-count></buy-count>

HTML标签属性名：不分区大小写，推荐纯小写；属性中只包含小写字母/数字，不能有减号！

<a href="" title="" style=""></a>

<buy-count productcount=""></buy-count>

HTML标签事件名：不分区大小写，推荐纯小写；事件中只包含小写字母/数字，不能有减号！

<a onclick="" @mouseover=""></a>

<buy-count @userincrement=""></buy-count>

--------------------------------------------------------------------------------

ES6中对象的简写方法：

var eage = 30;

var ework = function(){}

var emp = {

age: uage,

work: ework

}

var age = 30;

var work = function(){}

var emp = {

age: age,

work: work

}

var age = 30;

var work = function(){}

var emp = {

age,

work

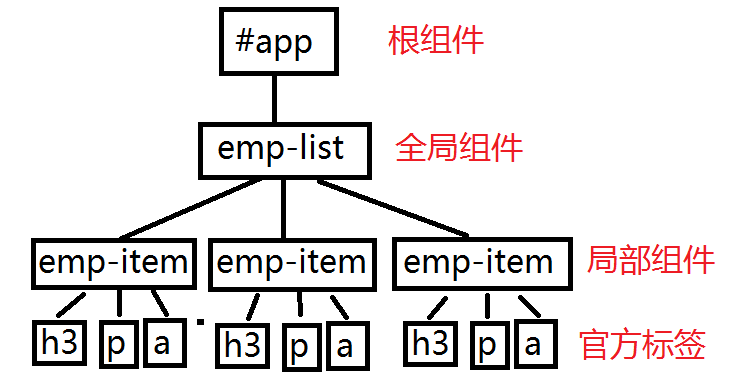
}

今日目标：

(1)组件间通信 —— 难点

(2)SPA和Vue-Router —— 重点

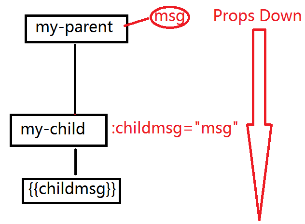
练习：根据下图写出全局组件和局部组件



1.Vue.js中组件直接传递数据 —— 重点&难点&面试题

**(1)父组件给子组件传递数据 day04 4-5-6**

"Props Down"：通过为子组件声明自定义属性，父组件中为该属性绑定模型数据，实现父=>子传数据效果。



**(2)子组件给父组件传递数据 day04 7-8**

"Event Up"：子组件发射自定义事件，并携带数据；父组件监听子组件可能抛出事件，并提供处理函数

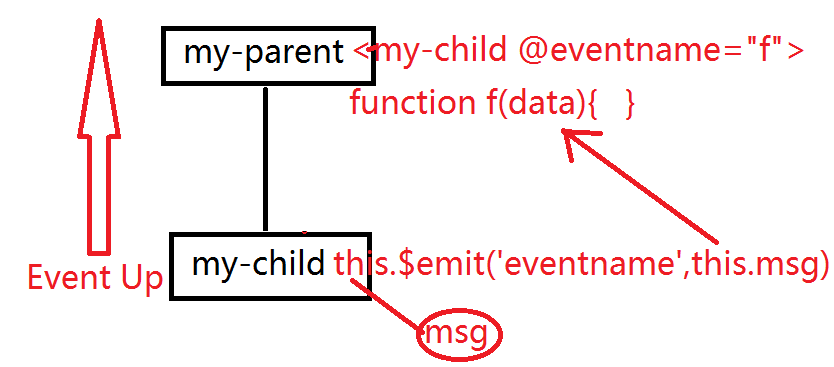
子组件：this.$emit('自定义事件名', **事件数据**);

父组件：<my-child @自定义事件名="func">

func(**事件数据**){

//接收到子组件发射的事件，并处理数据

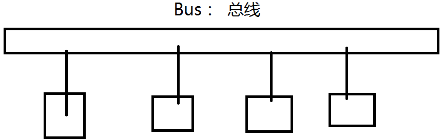
}



**(3)兄弟组件间传递数据 —— 实际应用场景比较少 day04 13**

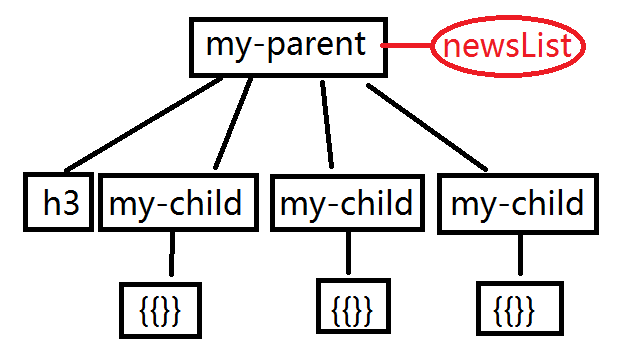
"Bus"：通信总线，可用于多个平级的兄弟间传递数据

var bus = new Vue();

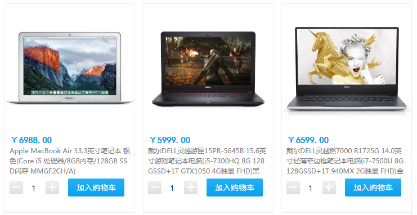


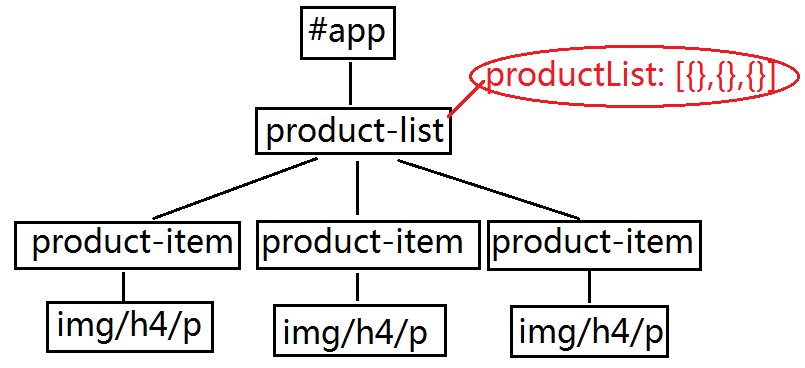
练习：在父组件内有模型数据：newsList: ['新闻标题1', ....]

根据此数据，渲染出多个子组件，其模板如下：

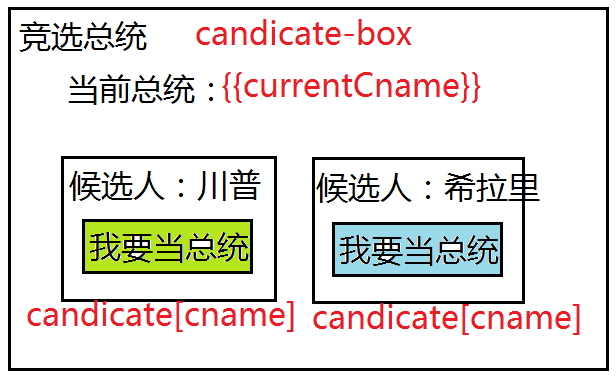


练习：使用父子组件实现商品列表：

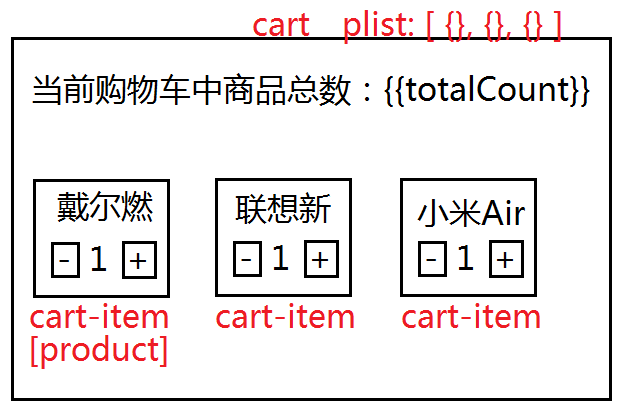




练习：竞选总统



练习：购物车计算器 **day04 11-12**



2.小知识：Vue.js中的计算属性

<p>{{totalCount}}</p>

var component = {

computed: {

totalCount: function(){

return 数据;

}

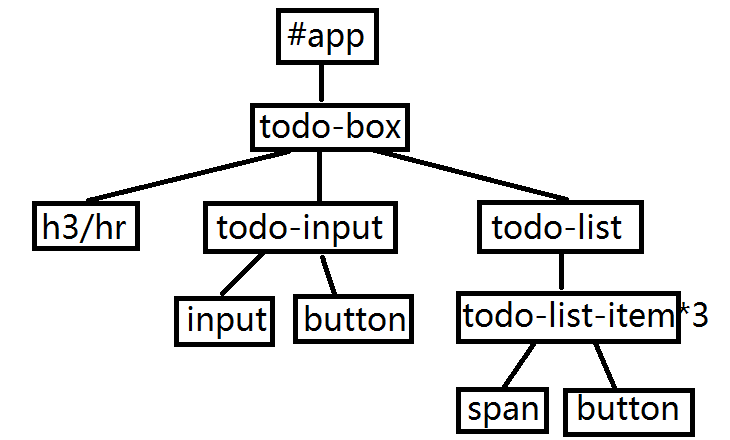
}

}

注意：计算属性的本质是函数，但比methods中的函数优越在：多次访问同一个计算属性仅会调用一次。

最终练习：TODO List

(1)分析组件结构



(2)显示已有列表项：

Props Down todo-box(msgList)=>todo-item

(3)添加新的列表项

Event Up input(userInput)=>todo-box

(4)删除已有列表项

Event Up todo-list-item=>todo-list=>todo-box

课后练习：

(1)完整实现Todo List

(2)完整实现“学子商城后台产品列表功能点”