1. 父:

<ul>

<todo-item v-for="(task,i) of tasks" :task="task" :i="i" ... >

子:

props:["task","i"]

<li>{{i++}} - {{task}} x </li>

报错:

答: props在绑定时用法和data中是完全一样的，但是，{{}}中有一个限制，就是只能取值，不能改值，因为会形成死循环。因为i变化，触发响应系统，重新更新页面，页面更新重新执行i++，又修改了变量i，又重复触发响应系统，又更新页面...。i+1，不改变i的值，返回+1后的新值，所以合法。i++等于i=i+1，不但返回+1后的新值，还试图修改变量i的值，所以违反了{{}}的规定

2. Vue到底时什么？组件到底是什么？

答: Vue是一个类型: 2部分

1. 构造函数: function Vue({el, data, methods,...}){ ... }

整个项目中只有一个根组件对象监控和扫描整个页面：

new Vue({

el:"选择器",

data:{ ... },

...

}

除了根组件之外，还可以有页面组件、全局组件、子组件。

以上所有组件，虽然创建方式不同，但是本质上都是Vue的孩子！

组件:{

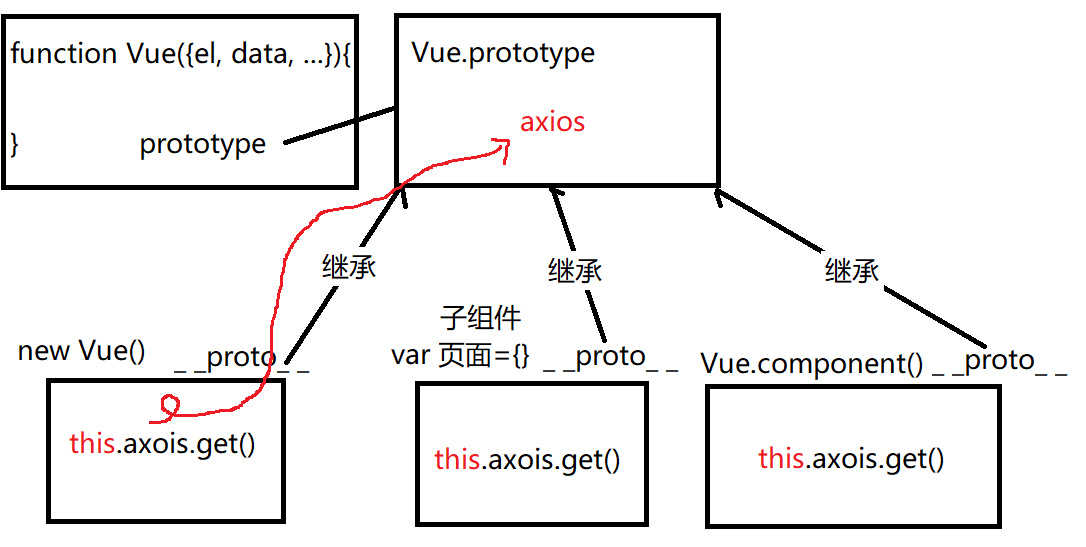
template:``,

data(){ return { ... } },

... ...

}

2. 原型对象: Vue.prototype 负责保存所有组件中都可以使用的对象或函数。

3. 什么是页面组件，什么是子组件，什么是全局组件？

答: 全局组件指多个页面共用的功能

子组件: 必须包含在父组件内的组件

所谓的页面组件，其实也是子组件:

var 页面={ ... }

只不过页面组件代表的是一个较大的完整的页面功能，用途不一样而已。

4. cli: command line interface

命令行工具

5. router.js中

{path:"/\*", component: { template:`<h2>404:Not Found</h2>`}}

报错: 不支持临时编译模板

答:

脚手架代码中的router.js中的某一条路由地址对象里的component属性后，不能接一个临时的{template:`html片段`}，因为脚手架代码不支持临时编译模板！所有模板必须都提前放在正式的.vue文件中。

6. 将脚手架项目从教室拷贝回家，axios不能用了

答: axios是和项目紧密相关的模块，必须以本地安装，直接安装在项目中: npm i -save axios，结果，在项目内node\_modules中多出axios文件夹。即使拷贝走项目，也不会丢axios。

如果采用全局安装axios； npm i -g axios，结果，axios是安装在电脑的系统盘中的node\_modules。没有在项目中，所有拷贝走项目，没有axios。

7. 如果prop:["i"],那么，data中: 能不能return { i: 0 }

答: 语法不报错！但是一定会有冲突！所以，强烈不建议这样使用。

8. <todo-item>中传过来的task和i两个变量值，是不是todoList里v-for每执行一次，就传一次？

答：是的：

todoList中:

<todo-item v-for="(t,i) of tasks" :key="i" :task="t" :i="i">

执行时:

v-for每循环一次就：

1. 创建一个<todo-item>并替换为<li>{{i+1}} - {{task}}</li>

2. 同时，为这个<li>创建一个子组件对象负责自动监听这个<li>中所有值的变化：{data:{...}, props:["task", "i"], ... }

3. 同时，将本次循环时取出的t和i的值，赋值给子组件中的task和i变量。比如t=吃饭， i=0， 然后，:task="t" :i="i"后，子组件的task=t=吃饭，子组件的i=父组件的i=0

4. 最后，将<li>{{i+1}} - {{task}}</li>，重新渲染为:

<li>{{0+1}} - {{吃饭}}</li>

结果: <li>1 - 吃饭</li>

每循环一次，以上1~4步都重复一次，因为每次循环获得的t和i变量的值不同，传给每个子组件副本对象中的task和i的值也就不同。最后每个<li>里渲染出的值也就各不相同。

9. 是不是如果父组件传来的是一个对象，props后也要写[]？

答: 是的。因为props后的["属性名1","属性名2",...]是声明多个属性变量的意思。[]中的属性变量，用于接收父组件传来的多个值。因为父组件可能给子组件多个值，所以子组件才需要用多个值来接。所以这里才主要用数组语法来同时声明多个变量。但是，props后的[]和每个属性变量接住什么类型的值毫无关系。仅仅是一种同时声明多个变量的特殊语法而已。

10. todoList里<todo-item v-for="(t,i) of tasks" :key="i" :i="i">为什么有了:key="i"，还要再定义一个:i="i"。而且，在子组件todoItem内部，用props:["key"]，拿不到下标。

答: v-for的:key属性，虽然是在开始标签中写的，但是，其实是被程序吸收进子组件对象内部，唯一标识这个子组件用的。非常关键。根本就不会让我们随意使用。所以，这个:key，不能当做普通属性使用。

11. this.$router.push()要用this 调用？

答: new VueRouter()是创建再 Vue的原型对象中的。不管是new Vue()，还是Vue.component()或子组件，其实都是Vue类型的孩子。在任何组件中，要使用Vue原型对象中的$router对象，必须用this.