# 第四章 VUE组件基础

1.组件的定义

2.全局组件

3.局部组件

4.DOM模板解析注意事项

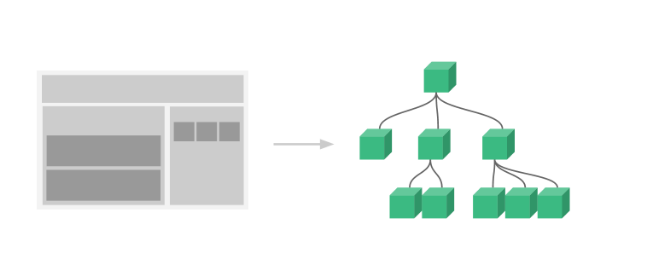
5.data必须是函数。

6.通过prop向子组件传递数据

7.单个根元素

1.组件的定义

组件(Component)是vue.js最强大的功能之一。组件可以扩展HTML元素，封装可重用代码。在较高层面上，组件是自定义元素，Vue.js的编译器为它添加特殊功能。在有些情况下，组件也可以表现为用is特性进行了扩展的原生HTML元素。所有Vue组件同时也都是Vue的实例，所以可接受相同的选项对象(除了一些根级特有的选项)并提供相同的生命周期钩子。



2.全局组件

我们可以通过以下的方式创建一个Vue实例



要注册一个全局组件，可以使用Vue.component（）。例如



组件在注册之后，便可以作为自定义元素<my-component></ my-component>在一个实例模板中使用。注意确保在初始化根实例之前注册组件:



3.局部组件

你不必把每个组件都注册到全局。你可以通过某个Vue实例/组件的实例选项components注册仅在其作用域可用的组件。



练习1：利用vue中的组件完成一行两列基本布局

4. DOM模板解析注意事项

当使用 DOM 作为模板时 (例如，使用 el 选项来把 Vue 实例挂载到一个已有内容的元素上)，你会受到 HTML 本身的一些限制，因为 Vue 只有在浏览器解析、规范化模板之后才能获取其内容。尤其要注意，像 <ul>、<ol>、<table>、<select> 这样的元素里允许包含的元素有限制，而另一些像 <option> 这样的元素只能出现在某些特定元素的内部。

在自定义组件中使用这些受限制的元素时会导致一些问题，例如：



当你查看DOM元素的时候，你会发现这个并不符合HTML5规范。所以，我们在这里必须借助Vue中的特殊属性，is属性



## 5.data必须是函数

我们在使用组件的时候，必须要注意到而且也要认识，组件中的data选项它不是一个对象，它是是一个函数，这点尤为重要。因为它需要每个实例可以维护一份被返回对象的独立拷贝。

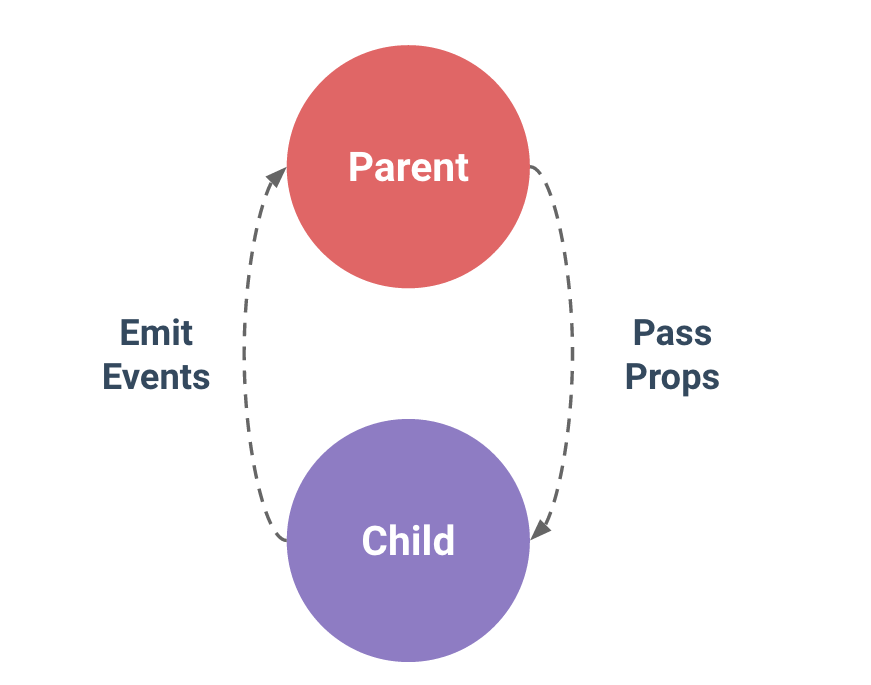


6. 通过prop向子组件传递数据

一).

组件设计初衷是要配合使用，最常见的就是父子组价的关系:组件A在它的模板中使用了组件B。他们之间必然需要相互通信：父组件可能要想子组件下发数据，子组件则可能要将它内部发生的事情告知父组件。然而，通过一个良好定义的接口来尽可能将父子组件解耦也是很重要的。这保证了每个组件的代码可以在相对隔离的环境中书写和理解，从而提高了其可维护性和复用性。

在 Vue 中，父子组件的关系可以总结为 prop 向下传递，事件向上传递。父组件通过 prop给子组件下发数据，子组件通过事件给父组件发送消息。看看它们是怎么工作的。



二).

prop 是父组件用来传递数据的一个自定义属性, 父组件的数据需要通过 props 把数据传给子组件，子组件需要显式地用 props 选项声明 "prop"。组件实例的作用域是孤立的。这意味着不能 (也不应该) 在子组件的模板内直接引用父组件的数据。父组件的数据需要通过 prop 才能下发到子组件中。



练习1：父组件向子组件传递值实现累加

练习2：父组件向子组件传递对象实现新闻列表

练习3：父组件向子组件传递值实现select下拉列表展示

练习4：父组件发起ajax请求实现新闻列表(jqueryAjax-axios)

7.单个根元素

当你构建一个组件的时候，你很有可能会出现 every component must have a single root element **(每个组件必须只有一个根元素)，**在编写组件的过程中，切记所有的元素都必须要有一个根元素包裹起来，一般选择使用div标签。

