**Vue.js 组件**

组件可以扩展 HTML 元素，封装可重用的代码。

组件系统让我们可以用独立可复用的小组件来构建大型应用，几乎任意类型的应用的界面都可以抽象为一个组件树。

1. 全局和局部组件
2. 全局组件

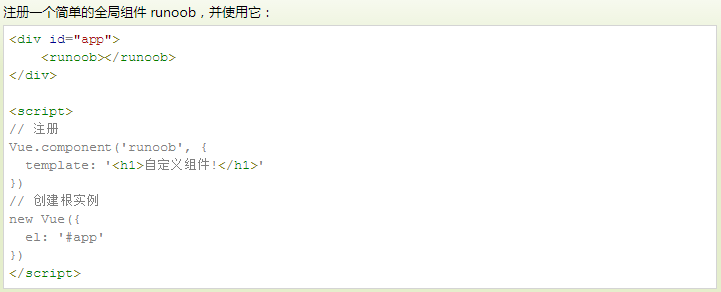
语法格式：



tagName 为组件名，options 为配置选项。注册后，我们可以使用以下方式来调用组件：



所有实例都能用全局组件：



1. 局部组件

我们也可以在实例选项中注册局部组件，这样组件只能在这个实例中使用：



1. **Prop**
2. 用props传递数据

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"：

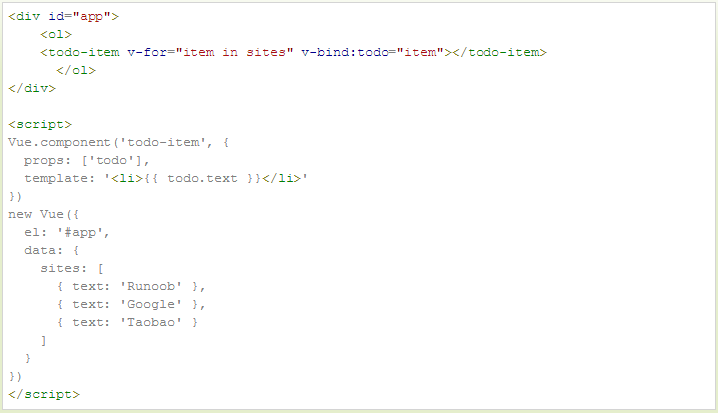


1. 动态Prop

类似于用 v-bind 绑定 HTML 特性到一个表达式，也可以**用 v-bind 动态绑定 props 的值到父组件的数据**中。每当父组件的数据变化时，该变化也会传导给子组件：



1. 将prop属性传到每个重复的组件中



注意：prop 是单向绑定的：当父组件的属性变化时，将传导给子组件，但是不会反过来。

1. Prop验证

组件可以为 props 指定验证要求。

prop 是一个对象而不是字符串数组时，它包含验证要求：



type 可以是下面原生构造器：

String Number Boolean Function Object Array

type 也可以是一个自定义构造器，使用 instanceof 检测。

1. 自定义事件

**父组件模板的内容在父组件作用域内编译；子组件模板的内容在子组件作用域内编译。**

用 v-on 绑定自定义事件, 每个 Vue 实例都实现了事件接口(Events interface)，即：

使用 $on(eventName) 监听事件 使用 $emit(eventName) 触发事件

**小结：（数据传递）父传子用props，子传父用自定义事件，然后父组件可以在使用子组件的地方用v-on来监听子组件触发的事件。**

以下实例中子组件已经和它外部完全解耦了。它所做的只是触发一个父组件关心的内部事件。



如果你想在某个组件的根元素上监听一个原生事件。可以使用 .native 修饰 v-on 。例如：

