# 第五章 Vue组件进阶

1.动态props

2.props字面量语法与动态语法

3.单向数据流概念

4.props验证

5.组件中ref的用法

6.组件中native的用法

## 一.动态props

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：



我们也可以通过v-bind来向子组件传递一个对象，已知一个toDo对象:



## 2. prop字面量语法与动态语法

我们来阅读以下的代码片段，



这是一个很简单的组件定义，父组件通过属性传递向子组件传递了message，子组件通过props来接受。那么我们思考一下，父组件传递的message是什么基本类型呢？

定义prop时，要是不使用v-bind是字面量prop，传递了一个字符串而不是数值1，如果想传递一个真正的 JavaScript 数值，则需要使用 v-bind，从而让它的值被当作 JavaScript 表达式计算。





## 三.单向数据流

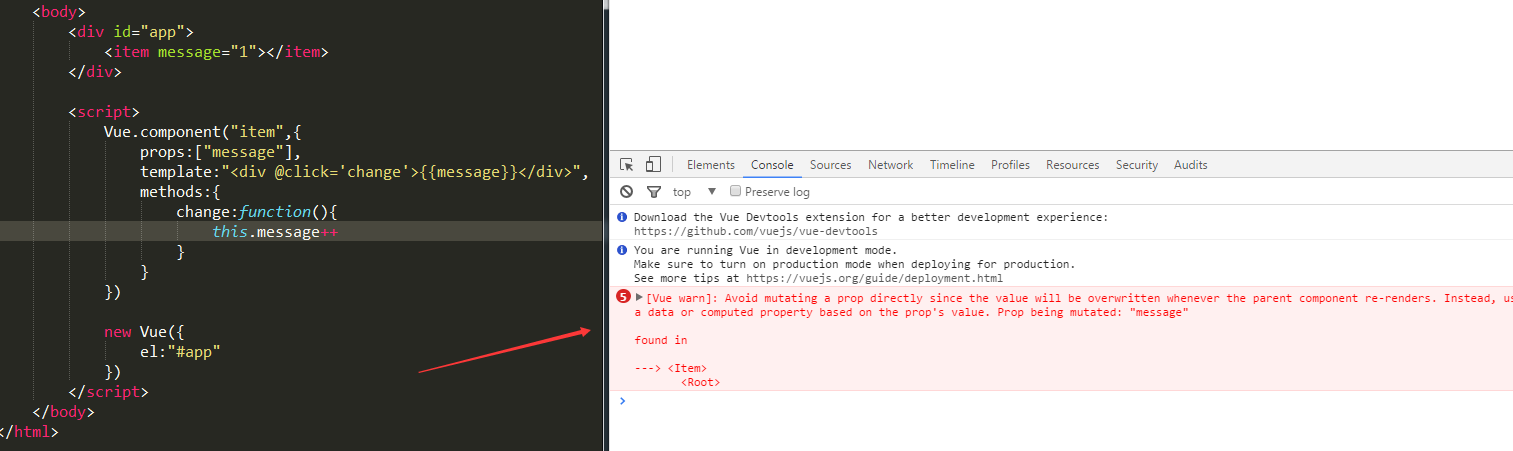
Prop 是单向绑定的：当父组件的属性变化时，将传导给子组件，但是反过来不会。这是为了防止子组件无意间修改了父组件的状态，来避免应用的数据流变得难以理解。

另外，每次父组件更新时，子组件的所有 prop 都会更新为最新值。这意味着你不应该在子组件内部改变 prop。如果你这么做了，Vue 会在控制台给出警告。

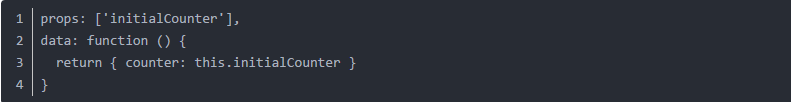
在两种情况下，我们很容易忍不住想去修改 prop 中数据：

1.Prop 作为初始值传入后，子组件想把它当作局部数据来用；

2.Prop 作为原始数据传入，由子组件处理成其它数据输出。



面对于这种情况，正确的应对方式是：定义一个局部变量，并用 prop 的值初始化它：

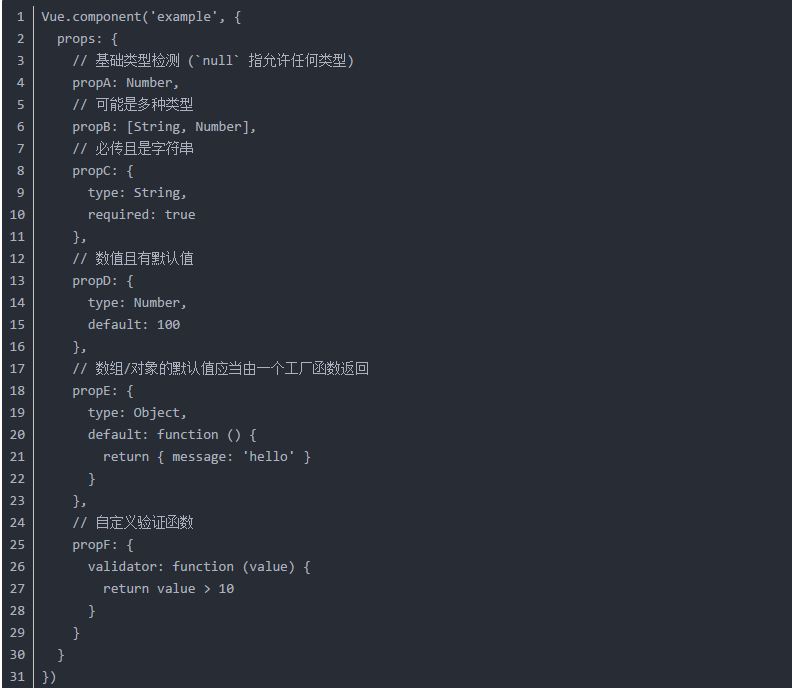




## 四.Props验证

我们可以为组件的 prop 指定验证规则。如果传入的数据不符合要求，Vue 会发出警告。这对于开发给他人使用的组件非常有用。

要指定验证规则，需要用对象的形式来定义 prop，而不能用字符串数组



## 5.组件中ref的用法

ref 被用来给元素或子组件注册引用信息。引用信息将会注册在父组件的 $refs对象上。如果在普通的 DOM 元素上使用，引用指向的就是 DOM 元素；如果用在子组件上，引用就指向组件实例：



## 6.组件中native的用法

你可能想在某个组件的根元素上监听一个原生事件。可以使用 v-on 的修饰符 .native

通俗点讲：就是在父组件中给子组件绑定一个原生的事件，就将子组件变成了普通的HTML标签，不加'. native'事件是无法触 发的。

可以理解为该修饰符的作用就是把一个vue组件转化为一个普通的HTML标签，并且该修饰符对普通HTML标签是没有任何作用的。