# 组件注册

组件名我们一般都遵循w3c的命名规则，（字母全小写，必须包含-并用用-隔开）。

两种命名方式，kebab-case和KebabCase

1. 全局注册

Vue.component('my-component-name', {  
   // ... 选项 ...  
 })

1. 局部注册

var ComponentA = { /\* ... \*/ }

1. 模块系统（ Babel 和 webpack 使用 ES2015 模块）

import ComponentA from './ComponentA.vue'  
  
 export default {  
 components: {  
 ComponentA  
 },  
 // ...  
 }

使用webpack (或在内部使用了 webpack 的 [**Vue CLI 3+**](https://github.com/vuejs/vue-cli))，那么就可以使用 require.context 只全局注册这些非常通用的基础组件。这里有一份可以让你在应用入口文件 (比如 src/main.js) 中全局导入基础组件的示例代码：



# 2.组件复用

组件是可复用的vue实例。

组件复用过程中要注意data 的使用。

1. #data必须是一个函数，

2.#组件的根元素只能有一个，

3.# 在组件上使用v-model



为了使组件的v-model的工作正常，value必须绑定在一个名叫value的prop上，

在其input事件触发时，将新的值通过自定义的input事件抛出。



1. prop使用

传入数字，字符串，数组的时候，用一个变量代替传入，不然js会把它解析为表达式。

Prop的单向数据流，，所有的prop都使得其父子之间形成了一个单向下行绑定：父组件的更新会向下流动到子组件里，但是反过来不行。不应该在子组件内部修改prop的值。（对象和数组是通过引用传入的，在子组件可以进行改动。）

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*传值改动处理\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

子组件如果想要获取值然后在子组件改变显示的话



\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Prop验证（便于用到自定义组件的人找到错误）：



# 3.自定义组件插槽

# 4.动态组件

#----------------------------------------------------------------------------#

**解析dom模板注意事项**

有些html元素，比如ul,ol,table,select ，html对于这些元素里面可以出现什么有很严格的限制， li，tr，option这些元素也只能在特定的地方出现。

需要注意的是**如果我们从以下来源使用模板的话，这条限制是*不存在*的**：

* 字符串 (例如：template: '...')
* [**单文件组件 (.vue)**](https://cn.vuejs.org/v2/guide/single-file-components.html)
* [**<script type="text/x-template">**](https://cn.vuejs.org/v2/guide/components-edge-cases.html#X-Templates)