**Vue.js 模板语法**

Vue.js 使用了基于 HTML 的模版语法，允许开发者声明式地将 DOM 绑定至底层 Vue 实例的数据。

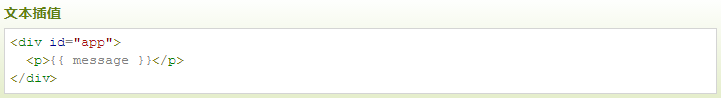
Vue.js 的核心是一个允许你采用简洁的模板语法来声明式的将数据渲染进 DOM 的系统。

结合响应系统，在应用状态改变时， Vue 能够智能地计算出重新渲染组件的最小代价并应用到 DOM 操作上。

**一、插值**

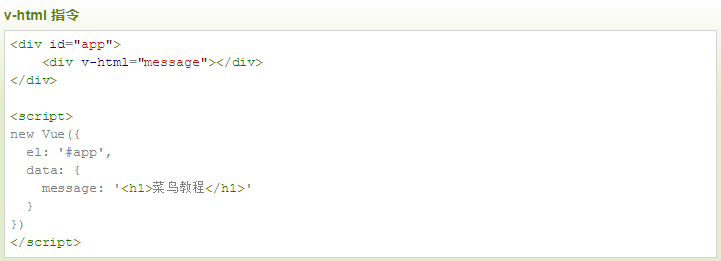
1、文本

数据绑定最常见的形式就是使用 {{...}}（双大括号）的文本插值：



2、Html

使用 v-html 指令用于输出 html 代码：



3、属性绑定

HTML 属性中的值应使用 v-bind 指令，可缩写为:class。

以下实例判断 class1 的值，如果为 true 使用 class1 类的样式，否则不使用该类：



4、JavaScript表达式

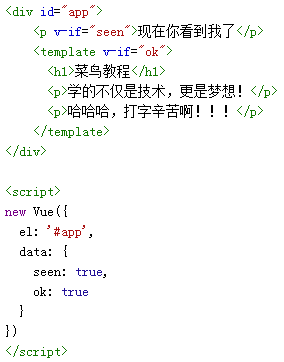
Vue.js 都提供了完全的 JavaScript 表达式支持。



**二、指令**

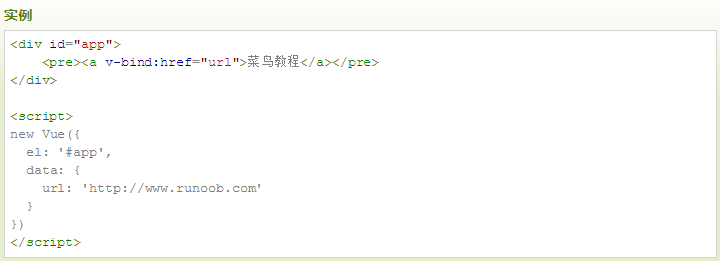
1、指令是带有 v- 前缀的特殊属性。

指令用于在表达式的值改变时，将某些行为应用到 DOM 上。如下例子：

这里， v-if 指令将根据表达式 seen 的值(true 或 false )来决定是否插入 p 元素。

2、参数

参数在指令后以冒号指明。例如， v-bind 指令被用来响应地更新 HTML 属性：



在这里 href 是参数，告知 v-bind 指令将该元素的 href 属性与表达式 url 的值绑定。

另一个例子是 v-on 指令，它用于监听 DOM 事件，可以缩写为@click：



在这里参数是监听的事件名。

3、修饰符

修饰符是以半角句号 . 指明的特殊后缀，用于指出一个指定应该以特殊方式绑定。例如，.prevent 修饰符告诉 v-on 指令对于触发的事件调用 event.preventDefault()：



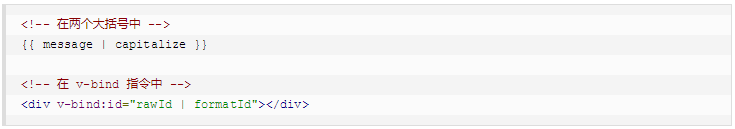
**三、用户输入**

1、在 input 输入框中我们可以使用 v-model 指令来实现双向数据绑定：



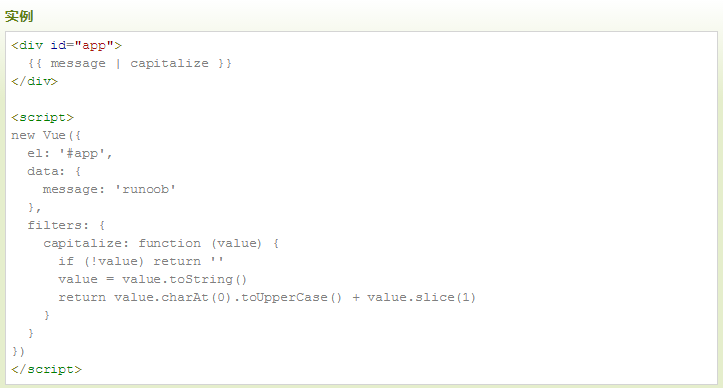
**四、过滤器**

1、Vue.js 允许你自定义过滤器，被用作一些常见的文本格式化。由"管道符"指示, 格式如下：



过滤器函数接受表达式的值作为第一个参数。

以下实例对输入的字符串第一个字母转为大写：



2、过滤器可以串联：



3、过滤器是 JavaScript 函数，因此可以接受参数：



注：以上三条中，message 是第一个参数，字符串 'arg1' 将传给过滤器作为第二个参数， arg2 表达式的值将被求值然后传给过滤器作为第三个参数。

**五、Vue实例**

1、构造器

每个 Vue.js 应用都是通过构造函数 Vue 创建一个 Vue 的根实例来启动的：



2、属性与方法

每个 Vue 实例都会代理其 data 对象里所有的属性：



除了 data 属性， Vue 实例暴露了一些有用的实例属性与方法。这些属性与方法都有前缀 $，以便与代理的 data 属性区分。例如：

