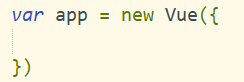
# Vue实例与数据绑定

## 实例与数据

1）使用Vue需要通过构造函数创建一个Vue实例：



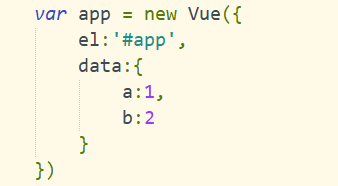
可以使用变量接收这个Vue实例。

2）在Vue实例中，有一个必不可少的参数：el：



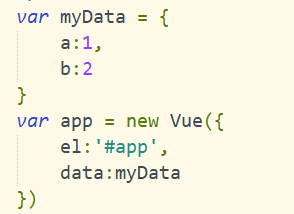
el需要指定一个页面中已存在的DOM元素来挂载Vue实例，它可以是HTMLElement，也可以是CSS选择器：

3）在Vue实例中，还有一个非常重要的参数：data



data参数是一个对象，用于数据绑定，建议将所有会用到的数据都预先在data内声明，这样可以避免将数据都散落在业务逻辑中，便于维护。

除了显式的在data中声明数据外，data参数也可以指向已有的变量，并且它们之间会默认建立双向绑定，当修改其中任意一个时，另一个也会一起变化：



## 生命周期

每个Vue实例在创建时，都会经历一系列的初始化过程，同时也会调用相应的生命周期钩子，我们可以利用这些生命钩子，在合适的时机执行我们的业务逻辑。常用的生命钩子有：

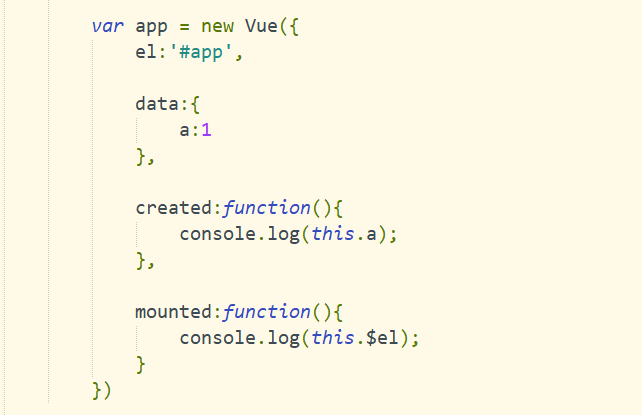
1. created：Vue实例创建完成后调用，此阶段完成数据的观测等，但尚未挂载，

$el还不可用，需要初始化处理一些数据时会比较有用，后面章节会讲解；

1. mounted：el挂载到实例上后调用，一般我们的第一个业务逻辑会在这里开始；
2. beforeDestroy：实例销毁之前调用，主要解绑一些使用addEventListener监听的

事件等。

这些钩子与el和data类似，都是作为参数写入Vue实例中的，它们是一个方法：



## 插值表达式

1. 使用双大括号“{{ }}”是最基本的文本插值方法，它会自动将我们双向绑定的数据实时显示出来。如：



上述代码中，{{info}} 会被替换为 Hello World。

2）通过插值表达式实时打印当前的时间：



1. 使用插值表达式无法输出HTML格式的内容，如果想要输出HTML格式的内容，需要使用v-html指令完成：

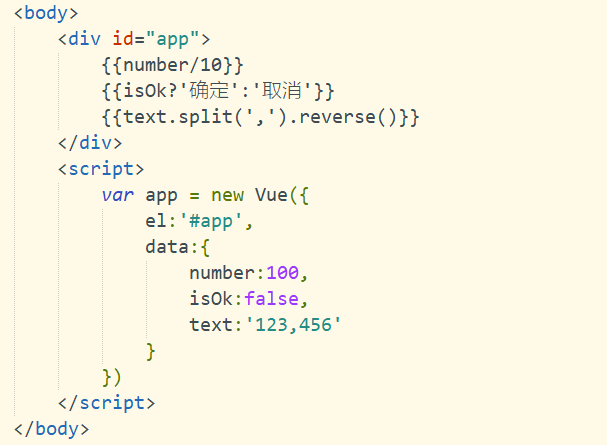


上述代码中，link的内容会被渲染为一个超链接放入<span>标签中，而使用插值表达式是无法完成这种功能的；但是，使用v-html有可能会导致XSS攻击，所有要在服务端对用户提交的内容进行处理，一般可将尖括号“<>”转义；

1. 如果不想编译某个插值表达式或者是想显示{{}}标签，可以使用v-pre跳过这个元素和它子元素的编译



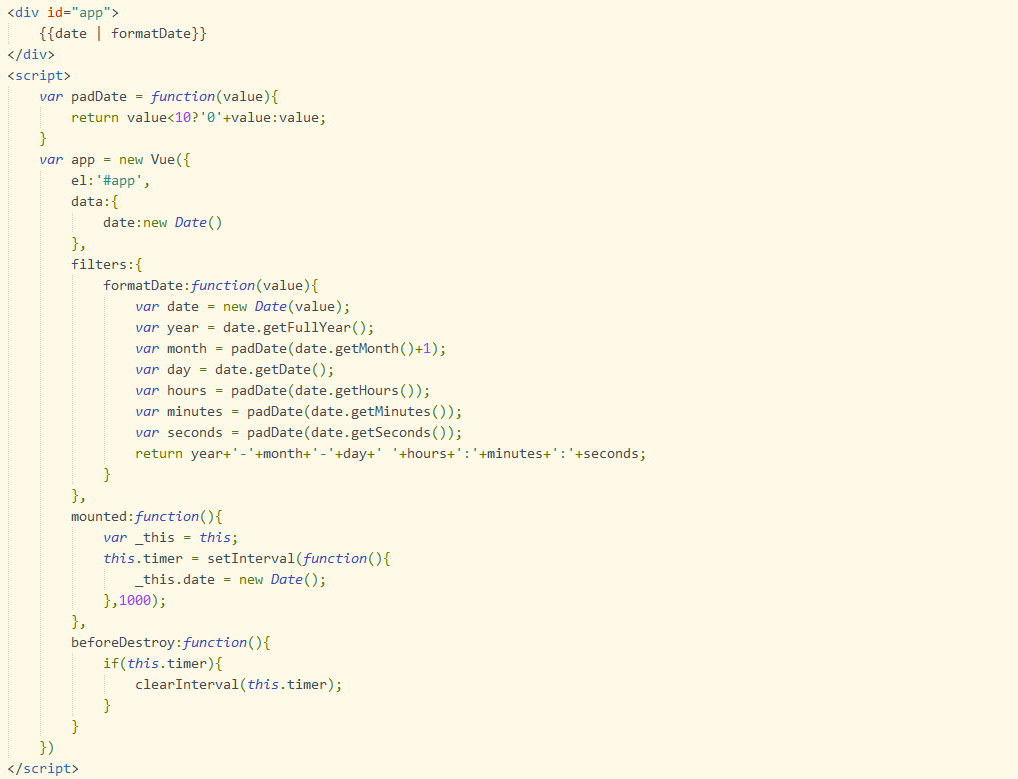
1. 在插值表达式中除了属性绑定值外，还可以使用JS表达式进行简单的运算、调用方法等：



## 过滤器

Vue.js支持在{{}}插值的尾部添加一个管道符“|”对数据进行过滤，经常用于格式化文本，比如格式化日期、字母大小写、货币千位使用逗号分隔等。过滤的规则是自定义的，通过给Vue实例添加filters参数设置。

例如在上一节中实时显示当前时间的示例，可以对时间进行格式化处理：



过滤器应当处理简单的文本转换，如果要实现更为复杂的数据变换，应该使用计算属性，下一章会介绍它的用法。

# 指令与事件

指令（Directives）是Vue.js模板中最常用的一项功能，它带有前缀v-，比如v-if、v-html、v-pre等，指定的主要职责就是当表达式的值改变时，相应地将某些行为或数据应用到DOM上。

Vue.js内置了很多指令，帮助我们快速完成常见的DOM操作，如循环渲染、显示隐藏等，在第 5章会详细介绍这些内置指令，但考虑到学习的层次性，这里先了解一下几个指令的用法。

## v-if

v-if指令用于逻辑判断。如：



当show为true时，<p>元素才会插入到DOM树中，为false则不会被插入；数据驱动DOM是Vue.js的核心理念，所以不到万不得已时不要主动操作DOM，只需要维护好数据，DOM的事Vue会帮你优雅的处理；

## v-bind

v-bind指令的基本用途是动态更新HTML元素上的属性，将数据绑定到DOM元素的

属性上，如id、class等。如：



这只是v-bind指令的基本用法，它在Vue.js中还有着很重要的作用，在第4章和第7章会详细介绍。

## v-on

1）v-on指令用于绑定事件监听器。如：



在button按钮上，使用v-on指令给该元素绑定了一个单击事件，在普通元素上，v-on指令可以监听原生的DOM事件，除了click外，还有dbclick、keyup等事件都可以监听；表达式可以是一个方法名，这些方法都写在Vue实例的methods参数内，并且是函数的形式，函数中的this指向当前Vue实例本身，因此可以直接使用this.xxx的形式来访问或修改数据。

2）v-on指令表达式除了可以绑定方法外，也可以直接是一个内联语句：



如果绑定的事件要处理的业务逻辑很复杂，建议还是使用在Vue实例中methods参数中声明方法，这样可读性更好，也好维护。

1. Vue.js将Vue实例中methods里的方法也代理了，所以也可以像访问Vue数据那样来调用方法：



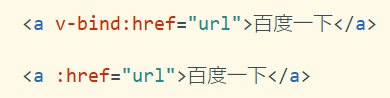
更多关于v-on事件的用法会在第7章中详细介绍。

# 语法糖

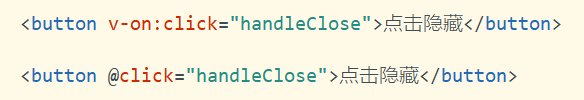
语法糖是指在不影响功能的情况下，添加某种方法实现同样的效果，从而方便程序开发。

Vue.js的v-bind和v-on指令都提供了语法糖，也可以说是缩写，比如v-bind，可以省略v-bind，直接写一个冒号“：”；v-on可以省略v-on，直接写一个“@”。如：

1. 使用冒号“：”代替v-bind：



1. 使用“@”代替v-on：



使用语法糖可以简化代码的书写，在后面的学习中所有示例的v-bind和v-on指令将默认使用语法糖的写法。