# VUE的基本使用day2

## vue过滤器

过滤器（filters）是vue为开发者提供的功能，常用于文本的格式化。过滤器可以用于两个地方：插值表达式 和 v-bind属性绑定

这就相当于调用过滤器函数，将原有（例如这里的message）的值作为函数的参数传递过去，最终显示的是过滤器函数处理完返回的结果

过滤器应该被添加在javascript表达式的尾部，由‘‘管道符’‘进行调节，示例代码如下：



第一个message就作为参数传递到capitalize过滤器函数中（函数的作用为将首字母转换为大写），最终会显示处理完成后的结果

注意：

1.过滤器是自己定义的，在Vue构造函数的参数中有filters节点（和methods定义函数的节点平级的节点），在filters中定义过滤器

2.过滤器本质是一个函数，但这个函数一定要有返回值，即一定要return

例：



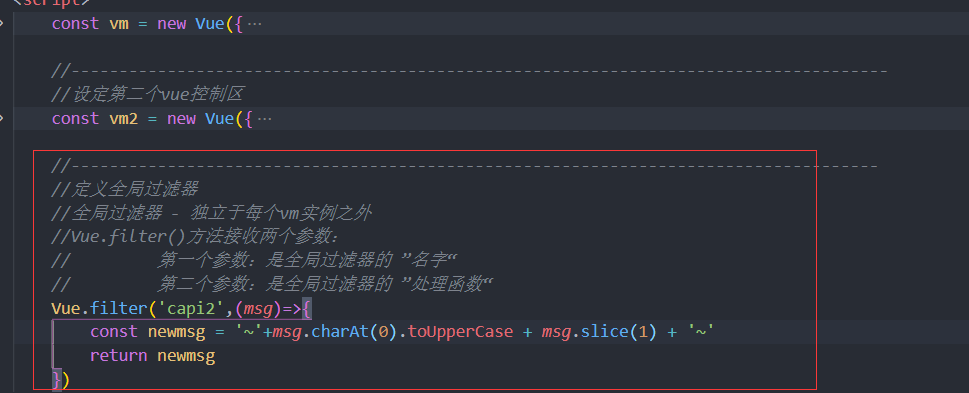
### 私有过滤器和全局过滤器

在filters节点下定义的过滤器，称为：“私有过滤器“ ，因为它只能在当前vm实例所控制的el区域内使用。

如果希望在多个vue实例之间共享过滤器，则就要设定定义全局过滤器

### 全局过滤器

可以按以下方式建立全局过滤器：



注意：如果全局过滤器和私有过滤器名字一致，此时就按照 “就近原则“ ，调用的是私有过滤器

### 连续调用多个过滤器

过滤器可以串联的进行调用，例如：



就是将前一个过滤器的结果作为参数传递到下一个过滤器，最终显示最后一个过滤器处理结果

### 过滤器传参

过滤器的本质是javascript函数，因此可以接收参数，格式如下：



### 过滤器的兼容性

过滤器仅在vue 2.x 和 1.x 中受支持，在vue 3.x的版本中剔除了过滤器相关功能。

在企业级项目开发中：

●如果使用的是2.x版本的vue，则依然可以使用过滤器相关的功能

●如果项目已经升级到了3.x版本的vue，官方建议使用计算属性或方法替代剔除的过滤器功能

具体需要看官方文档说明

<https://v3.vuejs.org/guide/migration/filters.htmI#migration-strategy>

## vue侦听器

### 什么是watch侦听器

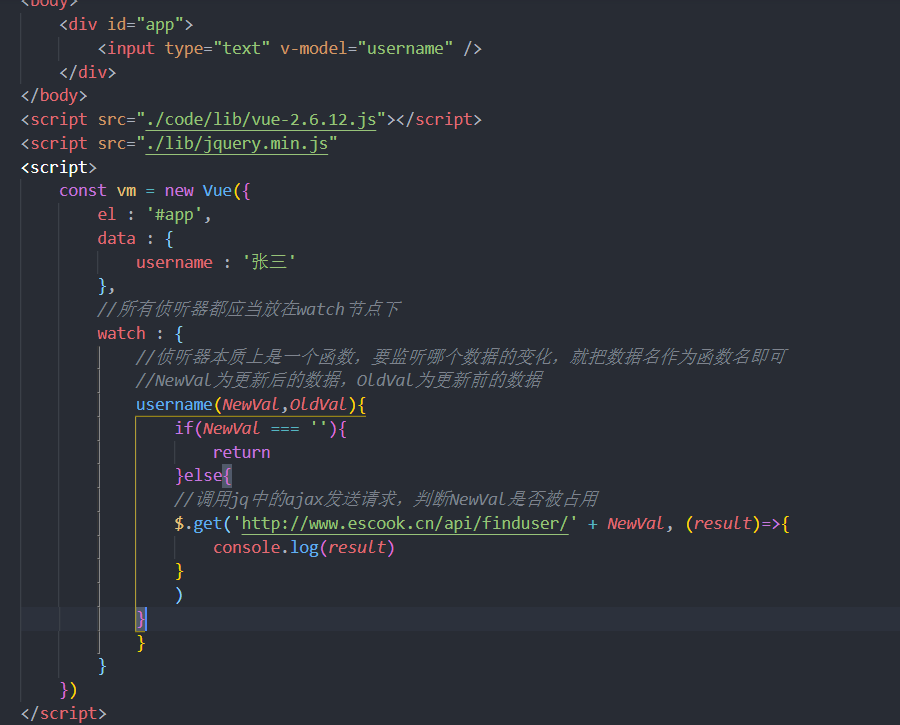
watch侦听器允许开发者监视数据的变化，从而针对数据的变化做特定的操作。

语法格式如下：



侦听器示例1 ——判断用户名是否被占用

在数据更新之后发送ajax请求向服务器发送判断



### 侦听器的格式

1. 方法格式的侦听器

○缺点1：无法在刚进入页面的时候自动触发

○缺点2：如果侦听的是一个对象，如果对象中的属性发生了变化，不会触发侦听器

1. 对象格式的侦听器

○好处：可以通过设置immediate 选项，让侦听器自动触发

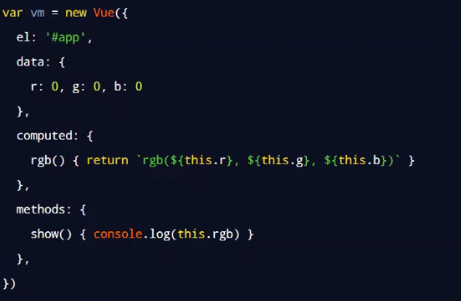
○好处2：可以通过设置deep 选项，让侦听器深度监听对象中的属性的变化

## 计算属性

### 什么是计算属性

计算属性指的是通过一系列运算之后，最终得到一个属性值。

这个动态计算出来的属性值定义在computed节点下，可以被模板结构或methods方法使用。示例代码如下：



这里定义了一个rbg的计算属性的方法，最终的返回值是rgb（x，x，x）

特点：

1.计算属性在定义时都是“方法格式”

2.计算属性在定义时是方法格式，但在定义完成后它就作为了vue实例的一个属性，值为它的返回值，要调用时直接和调用属性的方法一样即可

好处：

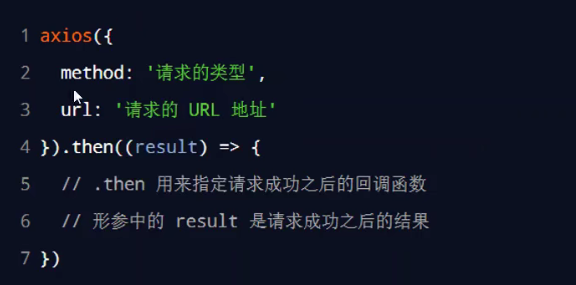
1. 实现了代码的复用
2. 只要计算属性中依赖的数据源发生了变化，则计算属性会自动重新求值

## axios

axios是一个专注于网络请求的库

### axios基本语法

axios的基本语法如下:

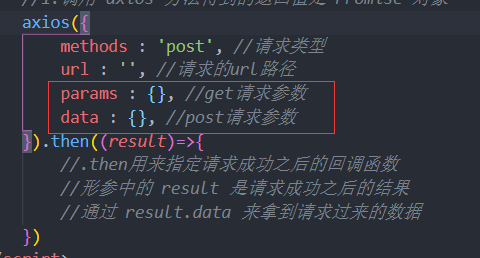


注意：在请求结果result对象中的data属性才是真正拿到的数据，所以要使用数据需要：result.data 来拿到

### axios的请求参数：

在axios中使用params和data两个节点来分别设置get请求参数和post请求参数

示例：



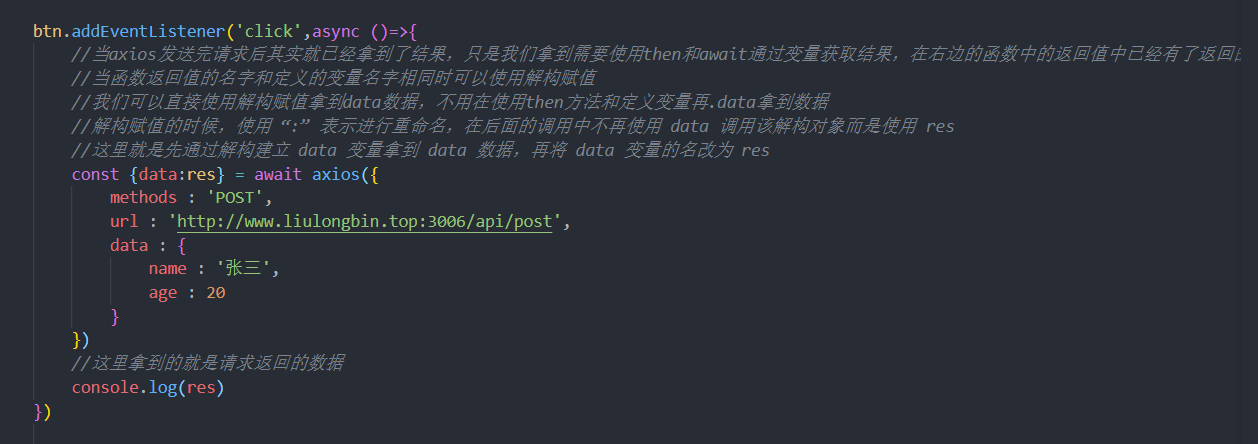
### axios的简化

在axios发送请求后可以简化获取结果的方式，示例：

1. 由于axios返回的是promise对象，所以可以使用 async/await



1. 通过解构赋值快速拿到正在需要的data数据



### axios直接发起GET和POST请求

1. GET请求

语法：

axios.get(‘url地址’, { params:{ ‘请求参数’ } } )

1. POST请求

语法：

axios.post(‘url地址’, { post请求体数据 } )

## vue-cli

### 什么是单页面应用程序

单页面应用程序简称（SPA），顾名思义，指的是一个web网站中只有唯一的一个HTML页面，所有的功能与交互都在这唯一的一个页面中完成

### 什么是vue-cli

vue-cli是Vue.js开发的标准工具。它简化了程序员基于webpack创建工程化的vue项目过程。

引用自vue-cli官网上的一句:

程序员可以专注在撰写应用上，而不必花好几天去纠结webpack配置的问题

中文官网：<https://cli.vuejs.org/zh/>

#### 安装和使用

##### 安装:

vue-cli 是npm上的一个全局包，使用npm install命令，即可的把它安装到自己的电脑上：

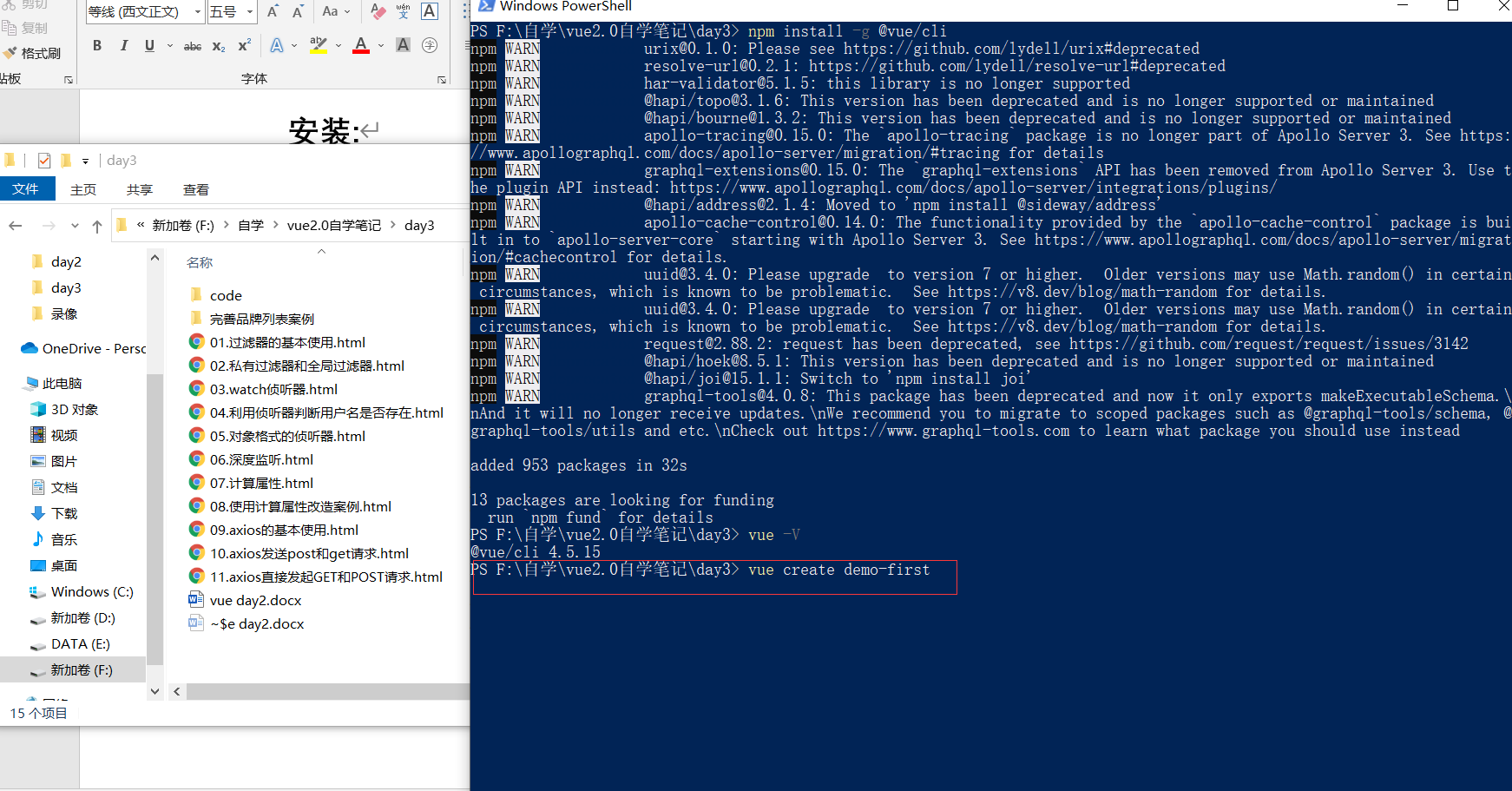
npm install -g @vue/cli

##### 使用：

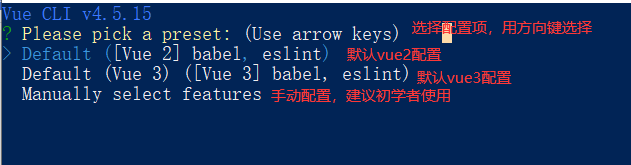
在当前项目位置运行shell，并执行以下命令：

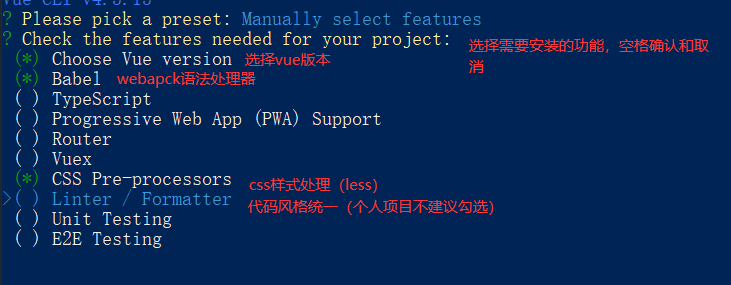
vue create 项目的名称

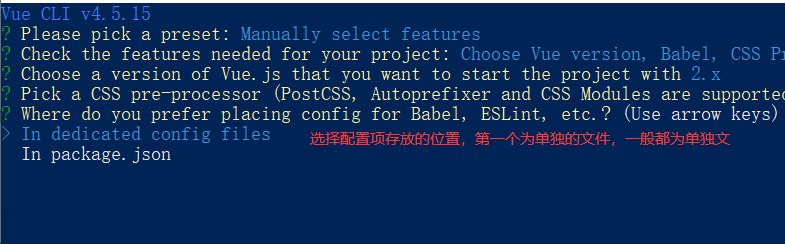
(项目名称建议都使用英文)

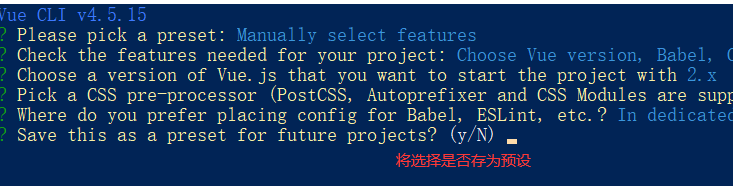


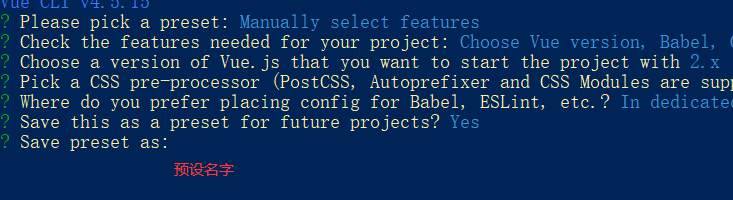
运行之后会进入一个窗口，这个窗口需要程序员进行选择后才能进行创建

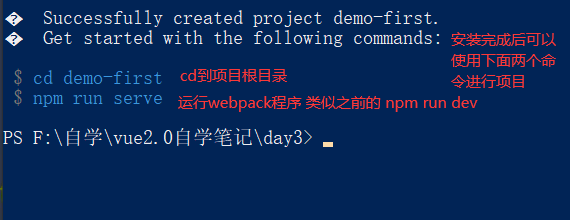












#### 了解创建的src目录构成

assets 文件夹：存放项目中用到的静态资源文件，例如：css样式表、图片资源

components 文件夹：程序员封装的、可复用的组件，都要存放到components目录下

main.js 是项目的入口文件。整个项目的运行，要先执行main.js

App.vue 是项目的根组件

#### vue项目的运行流程

在工程化的项目中，vue要做的事情非常简单：通过main.js把app.vue渲染到index.html的指定区域中



其中：

① app.vue用来编写待渲染的模板结构

② index.html中需要预留一个el区域

③ main.js把App.vue渲染到了index.html所预留的区域中

#### 组件的基本使用

自定义组件：Test.vue



将Test.vue渲染到HTML页面中：



#### VUE组件

1. 什么是组件化开发：

组件化开发是指：根据封装的思想，把页面上可重复使用的ui结构封装为组件，从而方便项目的开发和维护

1. vue中的组件化开发

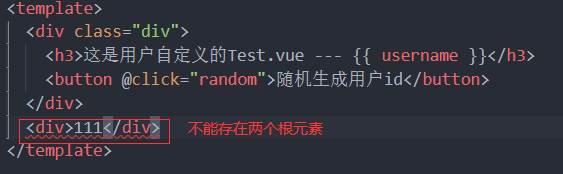
vue是一个支持组件化开发的前端框架

vue中规定：组件的后缀名为 .vue 。之前接触到的App.vue文件本质上就是一个Vue组件

1. vue组件的三个组成部分

每个.vue组件都是由3部分组成，分别是：

* template ->组件的模板结构（需复用的dom）(注意：在template中只能有一个根元素即只能在最外层存在一个大div，否则会报错)



* script ->组件的javascript行为(除了data要为函数，其他的还是和之前vue用法相同，例如在methods里定义事件处理函数等)
* style ->组件的样式

