

● 工业和信息化人才培养工程系列丛书
● 1+X证书制度试点培训用书

Web 前端开发 **高级**

（上册）



工业和信息化部教育与考试中心 主编

 中国工信出版集团

 电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
<http://www.phei.com.cn>

工业和信息化人才培养工程系列丛书

1+X 证书制度试点培训用书

Web 前端开发（高级） （上册）

工业和信息化部教育与考试中心 主编

電子工業出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

面向职业院校和应用型本科院校开展 1+X 证书制度试点工作是落实《国家职业教育改革实施方案》的重要内容之一,为了便于 X 证书标准融入院校学历教育,工业和信息化部教育与考试中心组织编写了《Web 前端开发(高级)》教材。

本教材以《Web 前端开发职业技能等级标准》(高级)为编写依据,分上、下两册,包括前端高效开发框架技术与应用、移动 Web 设计与开发、性能优化与自动化技术三篇,分别对应《Web 前端开发职业技能等级标准》(高级)涉及的三门核心课程:“前端高效开发框架技术与应用”“移动 Web 设计与开发”“性能优化与自动化技术”。

本教材以模块化的结构组织各篇及其章节,以任务驱动的方式安排教材内容,选取移动 Web 典型应用作为教学案例。本教材可用于 1+X 证书制度试点工作中的 Web 前端开发职业技能等级证书教学和培训,也可以作为期望从事 Web 前端开发职业的应届毕业生和社会在职人员的自学参考用书。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有,侵权必究。

图书在版编目(CIP)数据

Web 前端开发:高级.上册 / 工业和信息化部教育与考试中心主编. —北京:电子工业出版社,2019.8
(工业和信息化人才培养工程系列丛书)

1+X 证书制度试点培训用书

ISBN 978-7-121-36799-1

I. ①W… II. ①工… III. ①网页制作工具—教材 IV. ①TP393.092.2

中国版本图书馆 CIP 数据核字(2019)第 113242 号

责任编辑:胡辛征 特约编辑:田学清

印 刷:三河市鑫金马印装有限公司

装 订:三河市鑫金马印装有限公司

出版发行:电子工业出版社

北京市海淀区万寿路 173 信箱

邮编:100036

开 本:787×1092 1/16 印张:11.5 字数:295 千字

版 次:2019 年 8 月第 1 版

印 次:2019 年 8 月第 1 次印刷

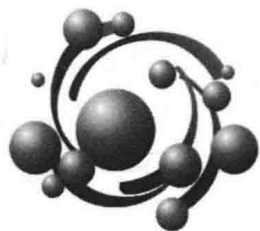
定 价:49.00 元

凡所购买电子工业出版社图书有缺损问题,请向购买书店调换。若书店售缺,请与本社发行部联系,联系及邮购电话:(010) 88254888, 88258888。

质量投诉请发邮件至 zlls@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

本书咨询联系方式:(010) 88254580, zuoya@phei.com.cn。

前 言



为积极响应《国家职业教育改革实施方案》，贯彻落实《关于深化产教融合的若干意见》《国家信息化发展战略纲要》的相关要求，应对新一轮科技革命和产业变革的挑战，促进人才培养供给侧和产业需求侧结构要素全方位融合，促进教育链、人才链与产业链、创新链有机衔接，推进人力资源供给侧结构性改革，深化产教融合、校企合作，健全多元化办学体制，完善职业教育和培训体系，着力培养高素质劳动者和技术技能人才。工业和信息化部教育与考试中心依据教育部《职业技能等级标准开发指南》中的相关要求，以客观反映现阶段行业的水平和对从业人员的要求为目标，在遵循有关技术规程的基础上，以专业活动为导向，以专业技能为核心，组织企业工程师、高职和本科院校的学术带头人共同开发了《Web 前端开发职业技能等级标准》。本教材以《Web 前端开发职业技能等级标准》中的职业素养和岗位技术技能为重点培养目标，以专业技能为模块，以工作任务为驱动进行组织编写，使读者对 Web 前端开发的技术体系有更系统、更清晰的认识。

随着新一轮科技革命与信息技术革命的到来，推动了产业结构调整与经济转型升级发展新业态的出现。在战略性新兴产业爆发式发展的同时，对新时代产业人才的培养提出了新的要求与挑战。据中国互联网络信息中心统计，截至 2018 年 12 月，我国网民规模达 8.29 亿人，手机网民规模达 8.17 亿人，网站数量达 523 万个，手机 App（移动应用程序）在架数量达 449 万款。在“互联网+”战略的引导下，Web 前端开发人员已经成为网站开发、手机 App 开发和人工智能终端设备界面开发的主要力量。企业增加门户网站的推广，从 PC 端到移动端，再到新显示技术、智能机器人、自动驾驶、智能穿戴设备、语言翻译、自动导航等新兴领域，全部需要应用 Web 前端开发技术。在智能制造等战略及新兴产业的高速发展中，出现了极为明显的人才短缺与发展不均衡现象。目前，软件开发行业的企业对 Web 前端开发工程师的需求量极大，全国总缺口每年近百万人。

随着移动互联网技术的高速发展，网站在静态页面的基础上添加了各类桌面软件，网页不再只是承载单一的文字和图片，而是被要求具备炫酷的页面交互、跨终端的适配兼容功能，使用富媒体让网页的内容更加生动，从而让用户有更好的使用体验，这些都基于前端技术来实现，其中包括 HTML、CSS、HTML5、CSS3、AJAX、JavaScript、jQuery 等，使得无论是在开发难度上还是在开发方式上，都对前端开发人员提出了越来越高的要求。

本教材包括前端高效开发框架技术与应用、移动 Web 设计与开发、性能优化与自动化技术 3 个篇目 18 个章节。

第一篇前端高效开发框架技术与应用。主要讲述了渐进式框架 Vue 的应用、Express 服务器开发及组件化开发思想。其中包括第 1 章 Vue 基础，第 2 章 Vue 组件，第 3 章 Vue 工程化工具，第 4 章 Express 服务器开发，第 5 章 axios 网络交互，第 6 章 Vue 路由，第 7 章 Vuex 状态管理，第 8 章 Vue UI。

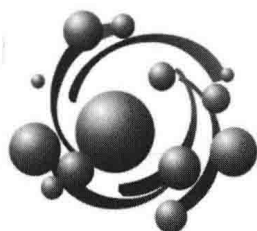
第二篇移动 Web 设计与开发。主要讲述了移动 Web 开发框架 jQuery Mobile 的引用，也包含多媒体、绘图、HTML5 新特性及 Less。其中包括第 9 章多媒体与绘图，第 10 章 HTML5 新特性，第 11 章 Less，第 12 章 jQuery Mobile。

第三篇性能优化与自动化技术。主要讲述前端代码优化、资源优化及模块化打包工具 webpack 的使用，以及 ES6 的基本语法。其中包括第 13 章 Web 前端开发概述，第 14 章 HTML 与 CSS 代码优化，第 15 章前端资源优化，第 16 章 JavaScript 代码优化，第 17 章 webpack 工具，第 18 章 ES6 基础。

本教材的编写与审校工作由严洁萍、陈慕菁完成，董旭依据《Web 前端开发职业技能等级标准》对全书做了内容统筹、章节结构设计和统稿。

由于编者水平有限，书中难免有不足之处，恳请读者不吝赐教并提出宝贵意见，相信读者的反馈将会为本教材再次修订提供良好的帮助。

目 录



第一篇 前端高效开发框架 技术与应用	
第 1 章 Vue 基础 2	
1.1 Vue 框架概述..... 3	
1.1.1 MVX 模式简介..... 3	
1.1.2 Vue 框架简介..... 4	
1.1.3 Vue.js 的安装与使用..... 6	
1.2 Vue 基础语法..... 8	
1.2.1 实例对象..... 8	
1.2.2 生命周期..... 10	
1.2.3 模板语法..... 14	
1.2.4 计算属性..... 19	
1.2.5 methods 方法..... 21	
1.3 Vue 渲染..... 21	
1.3.1 列表渲染..... 21	
1.3.2 条件渲染..... 28	
1.4 Vue 事件与表单..... 32	
1.4.1 事件处理..... 32	
1.4.2 事件对象..... 36	
1.4.3 事件委派..... 38	
1.4.4 表单处理..... 39	
1.5 Vue 基础应用..... 43	
1.6 本章小结..... 46	
第 2 章 Vue 组件 47	
2.1 Vue 组件概述..... 48	
2.1.1 组件简介..... 48	
2.1.2 组件对象剖析..... 50	
2.1.3 Vue 组件的 data 属性..... 52	
2.1.4 props 传递数据..... 55	
2.2 事件通信..... 59	
2.2.1 父子组件事件通信..... 59	
2.2.2 非父子组件事件通信..... 62	
2.2.3 插槽..... 65	
2.3 组件类型..... 73	
2.3.1 自定义组件 v-model 双向绑定..... 73	
2.3.2 动态组件..... 75	
2.3.3 递归组件..... 77	
2.4 Vue 组件应用..... 79	
2.4.1 分页组件..... 79	
2.4.2 购物车组件..... 80	
2.5 本章小结..... 83	
第 3 章 Vue 工程化工具 84	
3.1 搭建 Node.js 环境..... 85	
3.1.1 Node.js 简介..... 85	
3.1.2 Node.js 的下载与安装..... 85	
3.1.3 npm 的安装与使用..... 86	
3.1.4 yarn 的安装与使用..... 91	
3.2 Vue CLI 概述..... 92	
3.2.1 Vue CLI 简介..... 92	
3.2.2 Vue CLI 的使用..... 93	
3.3 Vue CLI 项目的配置..... 94	
3.4 JavaScript 模块..... 98	
3.4.1 CommonJS 模块..... 100	
3.4.2 ES6 模块..... 101	
3.5 vue-devtools 调试工具..... 104	

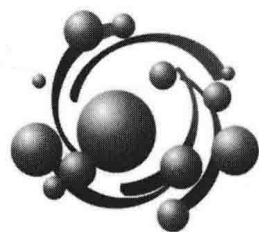
3.5.1	vue-devtools 的下载 与安装.....	104	6.2.3	路由参数	142
3.5.2	vue-devtools 的使用	104	6.2.4	在路由组件中使用 AJAX.....	143
3.6	本章小结	104	6.3	路由导航.....	144
第 4 章	Express 服务器开发	106	6.3.1	路由导航守卫	144
4.1	Express 概述.....	107	6.3.2	程式化路由导航.....	145
4.1.1	创建 Express 应用程序.....	107	6.3.3	全局路由导航守卫.....	146
4.1.2	Express 路由	109	6.4	嵌套路由和命名路由由视图	147
4.1.3	pug 视图模板.....	114	6.4.1	嵌套路由	147
4.2	请求与上传文件	116	6.4.2	命名路由视图	149
4.2.1	GET 请求和 POST 请求.....	116	6.5	Vue 路由应用	150
4.2.2	上传文件.....	118	6.5.1	用户登录与校验.....	150
4.3	Session 概述.....	121	6.5.2	注销功能及存在的问题.....	152
4.3.1	Session 简介.....	121	6.6	本章小结.....	154
4.3.2	Session 的使用.....	121	第 7 章	Vuex 状态管理	156
4.4	构建 RESTful API	122	7.1	Vuex 状态管理概述	157
4.5	Nodemon 概述.....	125	7.1.1	Vue 组件共享数据的方式	157
4.6	本章小结	126	7.1.2	Vuex 的安装与使用	158
第 5 章	axios 网络交互	128	7.2	Vuex.Store 对象的接口	159
5.1	axios 的基本使用.....	129	7.2.1	State	160
5.1.1	axios 简介	129	7.2.2	Getters.....	162
5.1.2	axios 的使用	129	7.2.3	Mutations	163
5.2	发送请求	130	7.2.4	Actions.....	165
5.2.1	发送 GET 请求	130	7.3	Vuex 多模块	167
5.2.2	发送 POST 请求	131	7.4	Vuex 插件	168
5.3	axios 跨域问题	132	7.5	重新组织 Vuex 目录结构	168
5.4	axios 网络交互应用.....	134	7.6	本章小结.....	169
5.5	本章小结	136	第 8 章	Vue UI	170
第 6 章	Vue 路由	138	8.1	Vue UI 概述	171
6.1	SPA 和 vue-router	139	8.1.1	Vue UI 简介	171
6.1.1	SPA.....	139	8.1.2	Element 的安装与使用	171
6.1.2	vue-router	139	8.2	Element UI 组件库.....	173
6.2	路由的使用	141	8.3	第三方扩展库 Express.....	174
6.2.1	定义路由.....	141	8.3.1	集成 Express 和 Vue.....	174
6.2.2	动态路由.....	142	8.3.2	配置 Express 和 Vue.....	174
			8.4	本章小结.....	177

第一篇

前端高效开发框架技术与应用

第1章

Vue 基础



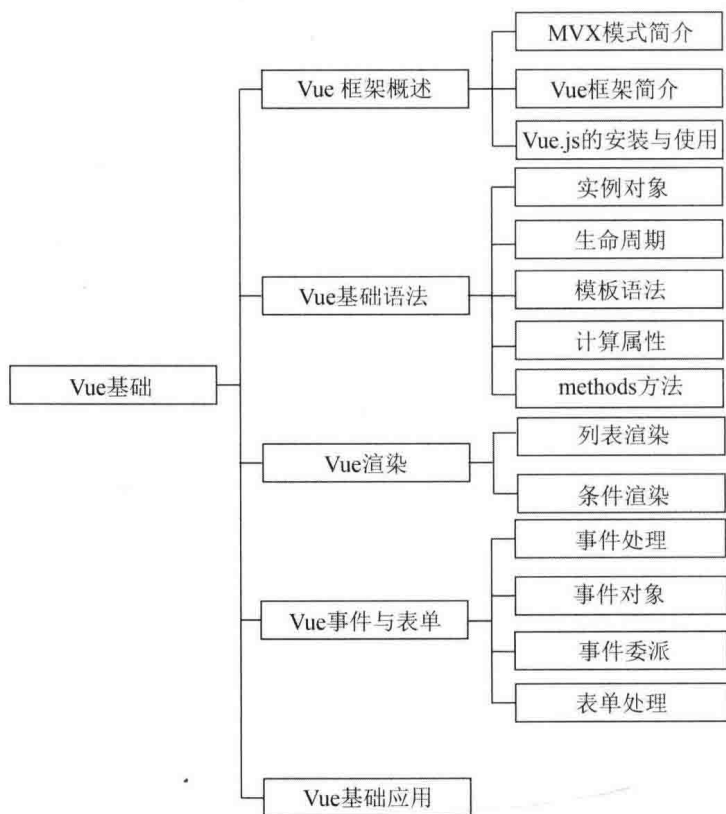
学习任务

【任务 1】了解 MVX 设计模式及 Vue 的框架和使用。

【任务 2】掌握 Vue 基础语法、Vue 多种渲染方式及 Vue 事件与表单。



学习路线



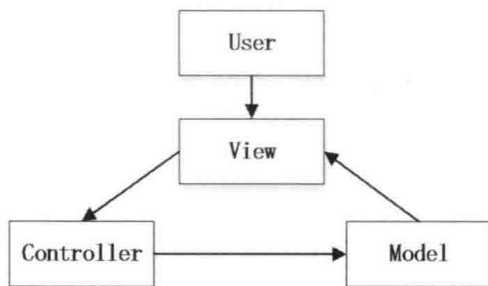
1.1 Vue 框架概述

1.1.1 MVX 模式简介

对于移动互联网公司的前端开发者来说，移动端对前端项目的要求越来越严格，前端项目已经不是简单地通过重新渲染来更新数据的频繁变化，后端的一些 MVC 模式也在往前端框架迁移。在介绍 Vue 框架之前，我们先来了解 MVX 模式。下面我们主要介绍 MVC 模式、MVVM 模式、两者之间的区别，以及 Web 前端开发经历的几个阶段。

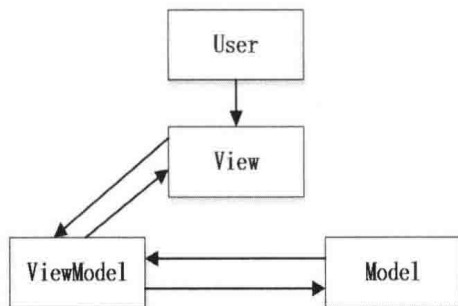
1. MVC 模式

MVC 模式是移动端应用广泛的软件架构之一，MVC 模式将应用程序划分为 3 部分：Model（模型）、View（视图）和 Controller（控制器）。MVC 模式的执行过程是将 View 层展示给用户，也就是通过 HTML 页面接收用户动作，将指令传递给 Controller。如单击一个按钮，就会将按钮触发的业务传递给 Controller；Controller 完成业务逻辑，要求 Model 改变状态。如果业务逻辑是单击按钮时显示列表数据，则 Controller 就处理业务逻辑，要求 Model 改变状态；Model 将新的数据发送给 View，则用户得到反馈。MVC 模式的执行过程如下图所示。



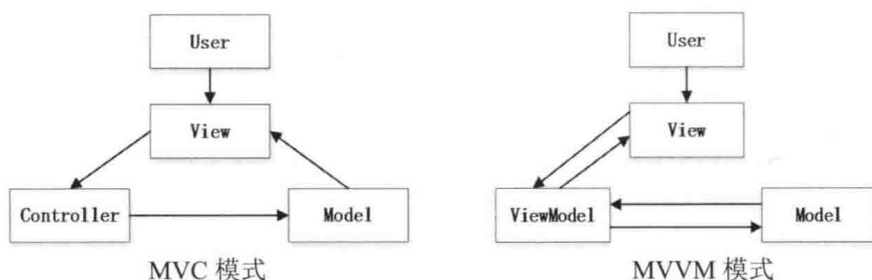
2. MVVM 模式

MVVM 模式是将 MVC 模式的 Controller 改成 ViewModel。View 的变化会自动更新 ViewModel，ViewModel 的变化也会自动同步到 View 层显示。View 层用来接收用户请求（DOM 事件、AJAX 等）；Model 层处理数据，不再与 View 层交互数据；ViewModel 监听 View 层请求状态的变化，同时刷新 View 层显示，ViewModel 和 Model 层之间进行数据双向绑定，Model 层监听 ViewModel 的变化。MVVM 模式的执行过程是 View 层接收到请求告诉 ViewModel，用户需要执行一些处理动作，当 ViewModel 发生变化时，告诉 View 层需要更新页面。所谓的数据双向绑定是 ViewModel 需要更新 Model 层的数据；反之，Model 层的数据改变，在 ViewModel 中的数据状态也要进行相应的改变。MVVM 模式的执行过程如下图所示。



3. MVC 模式和 MVVM 模式的区别

在 MVC 模式中,数据是单向通信的,按照 View→Controller→Model→View 方向循环。在 MVVM 模式中,数据可以双向通信,核心是 ViewModel 对象。这两种模式的数据通信方式如下图所示。



4. Web 前端开发阶段

- 源代码开发阶段: 直接使用 HTML、CSS 和 JavaScript 代码进行前端开发。

优点: 学习成本较低, 容易入手。

缺点: 代码结构混乱, 代码冗余, 浏览器兼容性不成熟, 不利于团队分工合作。

- 代码库开发阶段: 使用成熟的开源扩展库进行前端开发 (如在 JavaScript 中可以使用 jQuery、在 CSS 样式中可以使用 Bootstrap 响应式框架等)。

优点: 开发快速, 浏览器兼容性良好。

缺点: 视图层和数据层混合在一起, 不利于团队分工合作。

- 框架开发阶段: 采用前端 MVC 模式或 MVVM 模式开发。

优点: 代码分层, 利于团队合作, 便于后续代码维护。

缺点: 学习成本较高, 框架更新换代较快。

1.1.2 Vue 框架简介

Vue 是一套构建用户界面的渐进式框架, Vue 只关注视图层, 采用自底向上增量开发的设计, Vue 的目标是通过 API 实现数据绑定和组合视图组件。

1. Vue 框架的特性

目前，主流互联网公司及前端开发者喜欢 Vue 框架的原因是它具有自己的特性。

- 数据绑定：对于一些处于交互状态的前端 UI 界面，数据绑定非常简单、方便。
- 指令：拥有内置的简单指令（v-*），用户也可以自定义指令，通过对应表达式值的变化就可以修改对应的 DOM。
- 轻量级：Vue.js 的体积非常小且不依赖其他基础库。
- 插件化：Vue.js 的核心包不包括 Router、AJAX、表单验证等功能，但是可以方便加载对应的插件。

2. Vue 框架的内容

Vue 框架是 Web 开发的核心部分，人们熟练地掌握 Vue 框架的基础知识有助于后续的学习。目前在国内外的公司中，Vue 框架的使用非常广泛。人们在学习 Vue.js 之前需要学习一些网站开发的前导课程，如《Web 前端开发》《JavaScript 程序设计》《响应式设计》等，也需要掌握一些基本技能，如能够使用 HTML、CSS 和 JavaScript 进行静态网站的开发；熟练掌握 Web 开发工具的使用（Chrome 浏览器、Sublime/Vs Code 代码编辑器）。Vue 框架的主要内容如下表所示。

模 板	主 要 内 容
Vue 基础知识	Vue 框架基础语法、Vue 开发工具
Vue 组件	Vue 组件、Vue 模块化开发
Vue 工程化工具	npm/yarn 和 webpack 开发工具、Vue CLI 开发工具
Express 服务器开发和 axios 网络请求	创建 Express 应用程序、axios 发送 AJAX 请求、Postman 插件
Vue 路由	单页面应用 SPA、vue-router 实现路由机制
Vuex 状态管理	本地存储
Vue UI 库	Vue 中的 UI 库

3. 主流的 MVVM 框架

目前，主流的 MVVM 框架有 Angular、React 和 Vue.js，下面分别介绍这 3 个主流框架的区别。

- **Angular**：谷歌推出的 MVVM 框架。

优点：功能强大，包含模板、数据双向绑定、路由、模块化、服务等功能，自带了丰富的 Angular 指令。AngularJS 由谷歌维护，社区非常活跃，能够很好地促进它的发展。AngularJS 和 Ionic 是移动端跨平台 App 开发的黄金搭档。

缺点：功能复杂。

- **React**：Facebook 推出的 JavaScript 库。

优点：学习简单，视图层使用 JSX 语法，增强视图层功能，以 JavaScript 为核心，其规范标准，跨平台移动 App 开发能力强大，可以快速开发移动端应用。

缺点：操作 DOM 能力较差，需要大量扩展的支持。

- Vue.js：尤雨溪于 2014 年开发的 MVVM 框架。

优点：拥有非常强大的 CLI 工具，方便应用快速部署和模块化开发，扩展模块丰富，扩展功能强大，与 Laravel 集成，Laravel 是服务器端的一个 PHP 框架，适用于服务器端 API 开发。

缺点：跨平台移动 App 开发能力薄弱。

4. 主流 MVVM 框架的应用场景

MVVM 三大框架主要应用场景如下表所示。

	Angular	React	Vue.js
发布时间	2009 年	2013 年	2014 年
维护者	谷歌	Facebook	尤雨溪
视图引擎	HTML	JSX	HTML
学习曲线	困难	较复杂	简单
大小	500KB	130KB	50KB
功能复杂度	完善	完善	简单
文档	英文	英文	中文
应用场景	大型复杂应用	大中型应用、移动跨平台开发	中小型轻量级应用

1.1.3 Vue.js 的安装与使用

1. Vue.js 的安装

- 在 Vue.js 的官网上直接下载 vue.min.js 并用<script>标签引入。

Vue.js 的下载地址为 <https://vuejs.org/js/vue.js>。

Vue 官方安装说明网址为 <https://cn.vuejs.org/v2/guide/installation.html>。

打开下载地址，按“Ctrl+A”组合键选中网页内容并进行复制，打开 VS Code 软件，新建 Vue.js 文件，然后粘贴复制的文件即可。这样我们就准备好了 Vue.js 文件，当使用时直接在应用程序中用<script>标签引入 Vue.js 文件即可。

- 利用 CDN 方式引入 Vue.js 文件。

CDN 是内容分发系统，它的主要作用是将网站的内容发布到最接近用户的网络“边缘”，使用户可以就近取得所需的内容，提高用户访问网站的响应速度。

Vue 推荐使用的网址为 <https://unpkg.com/vue>。

Staticfile CDN（国内）推荐使用的网址为 <https://cdn.staticfile.org/vue/2.2.2/vue.min.js>。

unpkg 推荐使用的网址为 <https://unpkg.com/vue/dist/vue.js>。

cdnjs 推荐使用的网址为 <https://cdnjs.cloudflare.com/ajax/libs/vue/2.1.8/vue.min.js>。

可以通过如下代码进行引用：

```
<script src="https://unpkg.com/vue"></script>
```

- 采用 npm 和 webpack 模块包形式引入 Vue.js 文件。

npm 是现代前端开发的主要工具，将复杂的 JavaScript 文件库组织成模块化管理方式，在使用时只需引入指定模块即可。

webpack 将模块化编码的程序进行打包生成，从而允许在不支持 ES6 的浏览器中访问。

2. 第一个 Vue 实例

根据上面的介绍，我们创建一个 Vue 实例来更好地理解程序，后续的程序可以在这个实例的基础上进行修改。Vue 实例主要包括 HTML 代码、CSS 代码和编写 JavaScript-Vue 代码。创建第一个 Vue 实例的主要步骤如下。

- (1) 引入 CDN 文件。
- (2) 创建 Vue 实例对象。
- (3) 在 HTML 文件中添加关联 Vue 实例对象。

在 Vue 实例对象中，data 属性表示用户需要为 Vue 实例对象传递一些数据。Vue 实例对象的参数需要放在双大括号之间 ({{...}})。

下面的实例，通过<script>标签引入官方推荐的 CDN 文件。首先创建一个 Vue 实例对象，Vue 实例对象通过 el 元素和 html 元素相关联，绑定 id 为 app 的 div 元素，也就是只有 id 为 app 的 div 元素才可以响应 Vue 中的数据。示例代码如下：

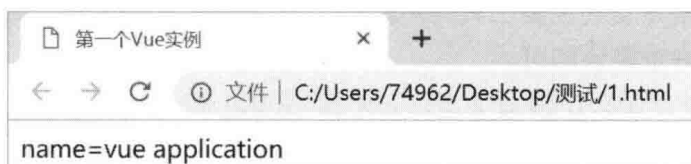
```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>第一个 Vue 实例</title>
  <script type="text/javascript"></script>
</head>
<body>
  <div id="app">
    name={{name}}
  </div>
  <script src="https://unpkg.com/vue"></script>
  <script>
    //创建一个 Vue 实例对象
    var vm = new Vue({
      el:'#app',//el 绑定的 id 为 app 的 div 元素
      data:{
        name:'vue application'
      }
    });
```

```

</script>
</body>
</html>

```

运行结果如下图所示。



3. Vue 框架开发工具

对于 Vue 框架的开发主要用到开发文档、浏览器及代码编写工具等，下面分别介绍 Vue 框架开发工具。

- 开发文档。

Vue 官方文档的下载地址为 <https://cn.vuejs.org/v2/guide>。

npm 官方文档的下载地址为 <https://www.kancloud.cn/shellway/npm-doc/199982>。

webpack 官方文档的下载地址为 <https://doc.webpack-china.org/concepts/>。

- 浏览器。

Chrome 浏览器。

vue-devtools 调试工具。

- 代码编写工具。

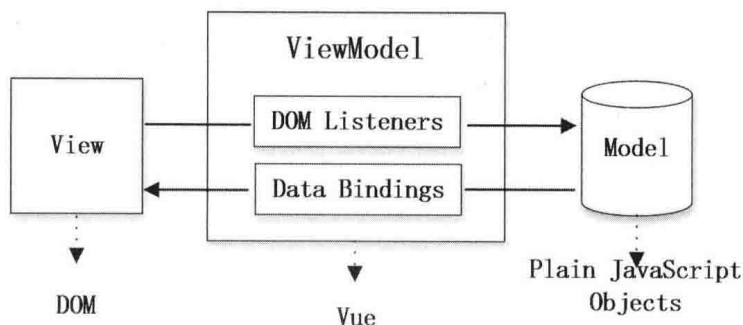
Atom、Sublime Text、Visual Studio Code 等。

1.2 Vue 基础语法

1.2.1 实例对象

Vue 实例对象是 Vue 框架的核心，人们需要掌握 Vue 实例对象的基本使用方法，同时也要深入理解 Vue 实例对象，包括数据的双向绑定机制、实例对象的访问方法与设置方法等。

什么是 Vue 实例对象？Vue 框架的核心是 Vue 实例对象，即 ViewModel 对象。Vue 实例对象连接 View（视图）层和 Model（模型）层，核心就是 ViewModel 对象，在 ViewModel 对象中关联 View 层和 Model 层，其中 View 层执行一个数据的双向绑定，当 View 层触发动作以后，告诉 ViewModel 对象的 DOM Listeners（事件监听机制），从而更新 Model 层中的数据；当 Model 层中的数据发生改变之后，交给数据双向绑定机制，双向绑定机制告诉 View 层要刷新数据。Vue 实例对象如下图所示。



在 Vue 框架中，使用 ViewModel 的方法非常简单，每个 Vue 应用程序都是通过创建一个新的 Vue 实例开始的。在创建 Vue 实例对象时，可以传入一系列实例选项，例如，data、props、methods、computed 等选项。创建 Vue 实例对象的语法格式为：

```
var vm = new Vue({ });
```

其中，data 属性是向 Vue 实例对象的响应式系统中添加双向绑定的数据，当 Vue 实例对象的 data 属性发生改变时，在视图中的数据也会相应发生改变；当视图中绑定的 data 数据改变时，在 Vue 实例对象 data 属性中的相应数据也会发生改变。在创建 Vue 实例对象之后，可以通过 vm.\$data 访问原始数据对象，Vue 实例对象代理了 data 所有的属性，因此访问 vm.a 等价于访问 vm.\$data.a。

下面的实例，通过 console.log(vm) 可以在控制台打印出 Vue 实例对象的所有属性，在浏览器中显示的是 data 中 name 的属性值'你好'，通过控制台输入 vm.name="1369" 可以改变浏览器的显示值，也可以通过 vm.\$data 回车显示原始数据对象。示例代码如下：

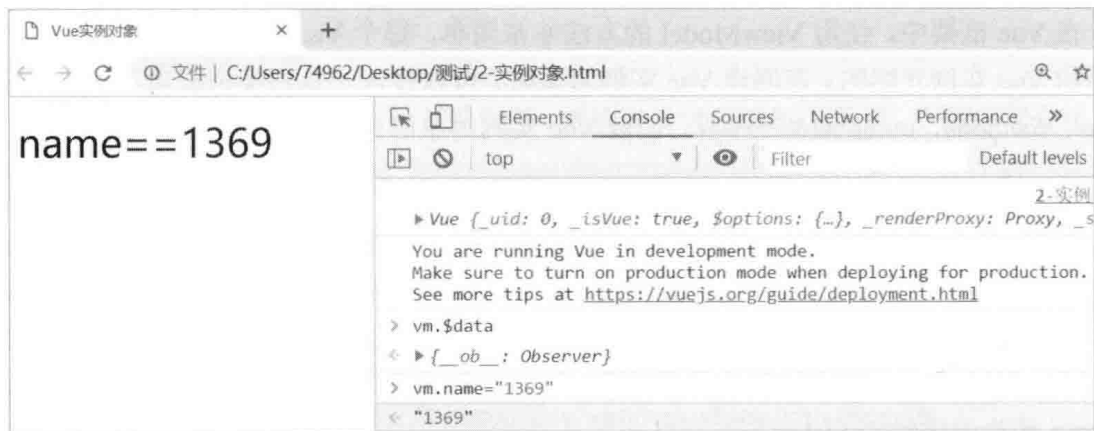
```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Vue 实例对象</title>
  <script type="text/javascript"></script>
</head>
<body>
  <div id="app">
    name=={{name}}
  </div>
  <script src="https://unpkg.com/vue"></script>
  <script>
    //创建一个 Vue 实例对象
    var vm = new Vue({
      //绑定的 DOM 元素
      el: '#app',
      //双向绑定的数据 VM 对象可以映射到 Model 中，Model 改变以后也会影响 VM 数据
      //VM 可以改变 View 中的数据，View 中的数据又会影响 VM 中的数据
      //双向绑定的数据都是键值对的形式
      data: { //通过 data 属性可以向双向绑定机制中传入数据
```

```

        name: '你好'
      }
    });
    console.log(vm); //包含一系列属性
  </script>
</body>
</html>

```

运行结果如下图所示。



1.2.2 生命周期

Vue 生命周期是 Vue 实例对象衍生出的一个机制，生命周期其实就是 Vue 实例对象在创建过程中所实现的回调函数，我们可以在回调函数中编写代码，从而实现一些特定的功能。

每个 Vue 实例对象在被创建之前都要经过一系列初始化过程，同时在这个过程中也会运行一些生命周期的回调函数，给予用户自定义启动过程的权利，具体的回调函数说明如下表所示。

回调函数	说明
beforeCreate()	Vue 实例对象创建之前
created()	Vue 实例对象创建之后
beforeMount()	Vue 实例对象和文档节点挂载之前
mounted()	Vue 实例对象和文档节点挂载之后
beforeUpdate()	View 视图更新之前
updated()	View 视图更新之后
beforeDestroy()	Vue 实例对象销毁之前
destroyed()	Vue 实例对象销毁之后

对于上文生命周期中的回调函数，我们要明白这些函数的使用方法、意义及使用场景。根据表格中的回调函数，大致可以分为 4 类，下面分别进行讲解。