

《Vue 应用开发》课程标准

执笔人：李娟 审核人： 制定日期： 年 月

目录

一、前言	2
(一) 课程的性质	2
(二) 设计思路	2
二、课程目标	2
(一) 知识目标	3
(二) 技能目标	3
三、课程内容和要求	3
四、教学设计	5
五、实施建议	8
(一) 教材选用与编写	8
(二) 教学建议	8
(三) 教学基本条件	9
(四) 课程资源的开发与利用	9
六、教学评价	9

《Vue 应用开发》课程标准

课程代码： **建议课时数：80** **学分：5**

适用专业：软件技术专业

先导课程：网页设计基础（html+css+div）、Javascript 基础

后续课程：后端开发

开课单位：计算机系

一、前言

（一）课程的性质

《Vue 应用开发》是软件技术专业的一门重要的专业基础课程，在第四学期开设。课程主要任务是通过理论教学与实践教学环节，让学生不仅掌握 Vue.js 库的使用方法，还可以积累项目开发经验，从而满足实际开发的需求。

通过本课程的学习，从 Vue 框架基本操作开始作到掌握 Vue 框架的各种应用技巧，侧重实战技能，使用简单易懂的实际案例进行讲解和操作指导，让学生更易掌握知识点。

课程类型为：（实践）课

（二）设计思路

课程设计理念是以职业需求分析为起点，以工作任务为驱动，以动态网页制作流程来设计教学程序，精心筛选典型任务组织教学。

在课程内容的选取上，紧密围绕完成工作任务所需的知识与技能，充分考虑高职院校学生的认知规律，突出教学内容的实用性、趣味性。

在课程内容的进程安排上，以 Vue 框架基本操作、Vue 框架的各种应用技巧和综合案例实践三大基本环节组织教学内容。

在课程教学实施上，将大量引用主流网站的常用特效为项目案例。在课堂教学中注重教学情景的创设，通过案例来引导来组织教学活动，使教学要求贴近工作需求，体现高等职业教育的特征，突出教学过程的目的性。

该门课程的总学时为 80 课时，其中理论课 0 课时，上机实验课 64 课时。

二、课程目标

通过本课程教学，使学生掌握 Vue 框架技术的应用，Vue.js 是当下很火的一个 JavaScript MVVM 库，它是以数据驱动和组件化的思想构建的。Vue.js 提供了更加简洁、更易于理解的 API，能够在很大程度上降低 Web 前端开发的难度，因此深受广大 Web 前端开发人员的喜爱。培养学生规范编码和良好的程序设计风格；培养学生面向对象编程的思维和提高逻辑思维能力；培养学生发现问题、分析问题、解决问题的能力；培养良好的职业素质；培养团队协作和和谐的沟通能力；并为学生以后从事更专业化的软件开发工作或者学习后续课程打下良好的基础。

学生通过该门课程的学习，为今后从事 Web 前端开发工作奠定的一定的基础。

（一）知识目标

1. 了解 Vue.js 的基本概念。
2. 掌握 Vue 实例和模板语法。
3. 掌握计算属性、侦听器和过滤器。
4. 掌握内置指令
5. 掌握页面元素样式的绑定
6. 掌握事件处理
7. 掌握表单输入绑定（双向数据绑定）
8. 掌握组件技术
9. 掌握项目脚手架 vue-cli
10. 掌握前端路由技术
11. 掌握状态管理

（二）技能目标

1. 能够熟练地使用 Vue 进行浏览器端的脚本开发。
2. 能够掌握网站建设的总体设计思想、步骤与方法。
3. 能够应用所学的相关知识开发各种形式的网站。
4. 能够独立开发 web 应用，具有一定的职业岗位竞争能力。

三、课程内容和要求

根据专业课程目标和涵盖的工作任务要求，确定课程内容和要求，说明学生应获得的知识、技能与态度。

序	工作任务	知识要求	技能要求
---	------	------	------

号			
第1章	Vue.js 简介	<ul style="list-style-type: none"> ● 1.1 Vue.js 概述 ● 1.2 Vue.js 的下载及使用 ● 1.3 实例化 Vue 对象、数据和方法 ● 1.4 MVVM 模式 	<ul style="list-style-type: none"> ● 掌握 Vue.js 的引入方法 ● 掌握实例化 Vue 对象、数据和方法 ● 掌握 Vue 中的 MVVM 模式
第2章	Vue 数据绑定	<ul style="list-style-type: none"> ● 2.1 Vue 模板语法 ● 2.2 响应式声明渲染机制 ● 2.3 Vue 计算属性 ● 2.4 侦听属性 	<ul style="list-style-type: none"> ● 掌握模板语法 ● 掌握 Vue 属性绑定 ● 掌握计算属性 ● 掌握侦听属性
第3章	● Vue 指令	<ul style="list-style-type: none"> ● 3.1 Vue 指令概述 ● 3.2 Vue 基本指令 ● 3.3 条件渲染 ● 3.4 列表渲染 	<ul style="list-style-type: none"> ● 掌握事件的绑定 ● 掌握事件的类型 ● 掌握事件对象
第4章	Vue 事件处理	<ul style="list-style-type: none"> ● 4.1 Vue 事件处理器 ● 4.2 修饰符 ● 4.3 综合案例 	<ul style="list-style-type: none"> ● 掌握事件处理器 ● 掌握修饰符
第5章	表单输入绑定	<ul style="list-style-type: none"> ● 5.1 双向绑定 ● 5.2 表单输入绑定基本用法 ● 5.3 值绑定 ● 5.4 修饰符 ● 5.5 案例实战 1 ● 5.6 案例实战 2 	<ul style="list-style-type: none"> ● 掌握双向绑定 ● 掌握表单输入绑定
第6章	Vue 组件	<ul style="list-style-type: none"> ● 6.1 组件的基本使用 ● 6.2 Vue 组件嵌套 ● 6.3 组件通信 ● 6.4 创建自己的组件 	<ul style="list-style-type: none"> ● 掌握组件的使用 ● 掌握组件的通信
第7章	自定义指令	<ul style="list-style-type: none"> ● 7.1 自定义指令概述 ● 7.2 钩子函数 ● 7.3 对象字面量 	<ul style="list-style-type: none"> ● 掌握自定义指令 ● 掌握钩子函数
第8章	过渡与动画	<ul style="list-style-type: none"> ● 8.1 transition 组件 ● 8.2 单元素/组件的过渡 ● 8.3 多个元素的过渡 ● 8.4 多个组件的过渡 ● 8.5 综合案例 	<ul style="list-style-type: none"> ● 掌握 transition 组件 ● 掌握元素/组件的过渡
第9章	渲染函数	<ul style="list-style-type: none"> ● 9.1 渲染函数概述 ● 9.2 createElement 方法 ● 9.3 使用 JavaScript 代替模板功能 	<ul style="list-style-type: none"> ● 掌握渲染函数
第	项目脚手	<ul style="list-style-type: none"> ● 10.1 脚手架的组件 	<ul style="list-style-type: none"> ● 掌握使用脚手架创建项目

第10章	vue-脚手架 cli	<ul style="list-style-type: none"> ● 10.2 脚手架环境搭建 ● 10.3 安装脚手架 ● 10.4 创建项目 	
第11章	Vue 路由 vue-router	<ul style="list-style-type: none"> ● 11.1 路由安装和使用 ● 11.2 传递参数及获取参数 ● 11.3 子路由 ● 11.4 命名视图和导航钩子 ● 11.5 元数据及路由匹配 	<ul style="list-style-type: none"> ● 掌握路由的应用 ● 掌握命名视图和导航钩子 ● 掌握元数据和路由匹配
第12章	状态管理	<ul style="list-style-type: none"> ● 12.1 Vuex 概述 ● 12.2 Vuex 的安装与使用 ● 12.3 在项目中使用时 Vuex 	<ul style="list-style-type: none"> ● 掌握 Vuex
第13章	综合案例篇	<ul style="list-style-type: none"> ● 综合案例 	<ul style="list-style-type: none"> ● 实现综合案例

四、教学设计

序号	工作任务	知识点	训练或工作项目	教学重点	教学情境与教学设计	建议学时
1	Vue.js 简介	1. 了解 Vue 是什么及 Vue 的特点 2. 了解 Vue 在前端开发中的优势 3. 掌握 Vue 的下载以及如何引入并应用 4. 掌握实例化 Vue 对象、数据和方法 5. 掌握数据挂载到 DOM 页面	任务 1: 下载和使用 Vue 任务 2: 实现将数据挂载到 DOM 任务 3: 实现 MVVM 模式效果	1. 实例化 Vue 对象、数据和方法 2. 数据挂载到 DOM 页面	1. 任务描述 2. 任务展示与实现 (1) 教师讲解本任务涉及的知识点 (2) 讲解和演示任务的实现 3. 学生动手操作 4. 任务小结	6
2	Vue 数据绑定	1. 模板语法 2. 响应式声明渲染机制 3. Vue 属性绑定 4. Vue 双向数据绑定 5. Vue 计算属性 Computed 6. 计算属性与	任务一: 实现双向数据绑定 任务二: 应用计算属性	1. Vue 属性绑定 2. Vue 双向数据绑定 3. Vue 计算属性 Computed	1. 任务描述 2. 任务展示与实现 (1) 教师讲解本任务涉及的知识点 (2) 讲解和演示任务的实现 3. 学生动手操作 4. 任务小结	10

		methods 的区别				
3	Vue 指令	<ol style="list-style-type: none"> vue 指令的定义 v-if 与 v-show 的区别 使用数组表达式绑定 class 属性 使用 v-model 绑定表单元素 vue 示例实现的任务展示 	<p>任务一：使用 v-model 缩写表单元素</p> <p>任务二：实现注册个人信息</p> <p>任务三：实现数据过滤</p>	<ol style="list-style-type: none"> Vue 内置指令 条件渲染 列表渲染 	<ol style="list-style-type: none"> 任务描述 任务展示与实现 <p>(1) 教师讲解本任务涉及的知识点</p> <p>(2) 讲解和演示任务的实现</p> <p>3. 学生动手操作</p> <p>4. 任务小结</p>	8
4	Vue 事件处理	<ol style="list-style-type: none"> Vue 事件处理器 方法事件处理器 内联事件处理器 事件修饰符 按键修饰符 修饰键 	<p>任务一：实现方法事件处理器</p> <p>任务二：实现事件处理综合案例</p>	<ol style="list-style-type: none"> Vue 事件处理 事件修饰符 按键修饰符 	<ol style="list-style-type: none"> 任务描述 任务展示与实现 <p>(1) 教师讲解本任务涉及的知识点</p> <p>(2) 讲解和演示任务的实现</p> <p>3. 学生动手操作</p> <p>4. 任务小结</p>	4
5	表单输入绑定	<ol style="list-style-type: none"> 双向绑定 基本用法 值绑定 修饰符 	<p>任务一：案例实战 1</p> <p>任务二：案例实战 2</p>	<ol style="list-style-type: none"> 双向绑定 值绑定 修饰符 	<ol style="list-style-type: none"> 任务描述 任务展示与实现 <p>(1) 教师讲解本任务涉及的知识点</p> <p>(2) 讲解和演示任务的实现</p> <p>3. 学生动手操作</p> <p>4. 任务小结</p>	6
6	Vue 组件	<ol style="list-style-type: none"> 组件的基本使用 Vue 组件嵌套 父组件向子组件通信 子组件向父组件通信 任意组件及平行组件通信 	<p>任务一：实现组件嵌套</p> <p>任务二：实现组件通信</p>	<ol style="list-style-type: none"> 组件使用 组件嵌套 组件通信 	<ol style="list-style-type: none"> 任务描述 任务展示与实现 <p>(1) 教师讲解本任务涉及的知识点</p> <p>(2) 讲解和演示任务的实现</p> <p>3. 学生动手操作</p> <p>4. 任务小结</p>	8
7	自定义指令	<ol style="list-style-type: none"> 自定义指令 注册全局 局部自定义指令 钩子函数 钩子函数参数 	<p>任务一：实现自定义指令</p> <p>任务二：钩子函数参数使用</p>	<ol style="list-style-type: none"> 自定义指令 钩子函数 	<ol style="list-style-type: none"> 任务描述 任务展示与实现 <p>(1) 教师讲解本任务涉及的知识点</p> <p>(2) 讲解和演示任务的实现</p>	4

		6. 指令参数 7. 对象字面量			3. 学生动手操作 4. 任务小结	
8	过渡与动画	1. 过渡与动画概述 2. Transition 组件 3. 过渡的命名 4. 单元素/组件的过渡 5. 实现过渡动画的三种方式 6. 多元素/组件的过渡	任务一：实现过渡效果 任务二：使用钩子函数实现动画	1. Transition 组件 2. 单元素/组件的过渡 3. 多元素/组件的过渡	1. 任务描述 2. 任务展示与实现 (1) 教师讲解本任务涉及的知识点 (2) 讲解和演示任务的实现 3. 学生动手操作 4. 任务小结	2
9	渲染函数	1. 什么是渲染函数 2. 使用渲染函数 3. createElement 方法 4. VNodes 必须唯一 5. 使用 JavaScript 代替模板功能	任务一：使用渲染函数 任务二：使用 createElement 方法	1. 使用渲染函数 2. createElement 方法	1. 任务描述 2. 任务展示与实现 (1) 教师讲解本任务涉及的知识点 (2) 讲解和演示任务的实现 3. 学生动手操作 4. 任务小结	4
10	项目脚手架 vue-cli	1. 脚手架的组件 2. 脚手架环境搭建 3. 安装脚手架 4. 创建项目	任务一：使用脚手架创建项目	1. 脚手架环境搭建 2. 安装脚手架 3. 创建项目	1. 任务描述 2. 任务展示与实现 (1) 教师讲解本任务涉及的知识点 (2) 讲解和演示任务的实现 3. 学生动手操作 4. 任务小结	2
11	Vue 路由 vue-router	1. 路由安装和基本配置 2. 传参及获取参数 3. 子路由 4. 命名视图和导航钩子 5. 元数据及路由匹配	任务一：使用 Vue 路由 任务二：创建子路由	1. 传参及获取参数 2. 子路由 3. 命名视图和导航钩子	1. 任务描述 2. 任务展示与实现 (1) 教师讲解本任务涉及的知识点 (2) 讲解和演示任务的实现 3. 学生动手操作 4. 任务小结	4
12	状态管理	1. 什么是 Vuex 2. 使用 Vuex 3. 在项目中使用的 Vuex	任务一：在项目中使用的 Vuex	1. 使用 Vuex 2. 在项目中使用的 Vuex	1. 任务描述 2. 任务展示与实现 (1) 教师讲解本任务涉及的知识点 (2) 讲解和演示任务	2

					的实现 3.学生动手操作 4. 任务小结	
13	综合案例篇	综合案例实训	任务：实现综合案例实训	实现综合案例实训	1. 任务描述 2. 任务展示与实现 (1) 教师讲解本任务涉及的知识点 (2)讲解和演示任务的实现 3.学生动手操作 4. 任务小结	8

五、实施建议

（一）教材选用与编写

（1）在教材内容选取上，应凸显实践性、应用性和层次性的特征，强调与岗位业务相吻合，并易学、易懂、易接受。同时又要有前瞻性，应将本专业领域的发展趋势以及操作中的新知识、新观念、新技术和新方法及时地纳入其中。

（2）在教材内容组织上，本课程教材的编写应打破传统的学科型内容体系，构建以岗位职业能力以及职业标准为依据的课程内容体系，以本课程标准为依据进行编写。

（3）教材应注重教学内容的情景创设和案例链接，加强教学内容与专业培养目标的关系性，挖掘学生既有知识与讲解内容的关联性，激发学生的学习热情和积极性。

（4）教材编写应注重可操作性和拓展性。除了知识讲解外，教材应当配有可供学生练习的要素：例如课后习题、课堂小实验、实训指导等；教材应当根据教学内容设置立体资源，给出相关参考资料、网络视频等链接地址，方便师生拓展阅读学习。

（二）教学建议

（1）适时调整教学内容

在教学内容方面，教师在教学过程中应了解 WEB 前端的岗位需求和技能要求，适时调整教学内容以便适应行业要求。

（2）采用讲练结合的教学策略

该课程的实践性较强，在教学过程中，应以实例讲解和实践操作相结合，通过“理实一体”进行教学，培养学生思考问题，分析问题和解决问题的能力。通过课堂讲授、课堂练习、课堂小实验等，引导学生能够独立完成任务的制作。例如，在讲解各个内置对象的时候，教学过程中建议充分利用网络开展课堂小实验，选取贴近学生生活的实际案例，提高课堂趣味性与参与性。

(3) 采用信息化教学手段提高教学互动性

移动互联网时代下的学生学习习惯已经发生较大变化，传统教学模式已经很难吸引学生兴趣。在本课程教学中，建议在多媒体授课中，多引入案例等手段提高课堂互动性，增进教学效果。

(三) 教学基本条件

(1) 师资条件

本课程是计算机专业的基础学科，要求任课教师具备前端开发的经验，熟悉前端开发的相关技术，具有计算机科学与技术、软件工程、计算机应用技术等专业背景的教师较适合本门课程的教学。

(2) 学生条件

由于课程的交叉融合特性，要求学生在学习本课程之前有一定的基础知识，其先修课程至少应当包括《网页设计基础（html+css+div）》、《Javascript 基础》等，故此课程应当在软件专业的第四学期开设。

(3) 实训条件

在教学环境条件方面，为给学生提供充分的实践锻炼机会，应当建有基本的计算机应用实训室，主要需要常用的网页开发软件，包括 Hbuilder、sublime、VsCode 等软件和主流浏览器等。

(四) 课程资源的开发与利用

(1) 编写实训指导书和实验实训教学标准。

(2) 利用现代信息技术开发研制各种形式的多媒体教学课件，为学生搭建一个立体的学习平台，激发学生的学习兴趣和调动学生学习的主动性和积极性。

(3) 应充分利用互联网企业资源，进行校企合作，建立实习实训基地，满足学生实训和实习的需要，并在合作中关注学生职业能力的发展和教学内容的调整。

(4) 积极开发和利用网络课程资源。

(5) 校企合作编写校本特色教材。

六、教学评价

针对本课程操作性较强的特性，建议采用多元化的评价方式。

(1) 改革考核手段和方法，加强实践性教学的考核，可采用过程和结果考核相结合的考核方法。

(2) 由主讲教师结合考勤情况、学习态度、课业、平时测验、实验实训、现场操作等情况考核，综合评定学生成绩。

(3) 应注重对学生动手能力和在实践中分析问题、解决问题能力的考核，对在学习和应用上有创新的学生给予特别鼓励，综合评价学生的能力。

[附]执笔人：李娟 审核人：
制定（修订）日期： 年 月