

# 《精通 Vue.js：Web 前端开发技术详解》

- 作者：孙卫琴 清华大学出版社出版
- 在线课程：[javathinker.ke.qq.com](http://javathinker.ke.qq.com)
- 技术支持网址：[www.javathinker.net](http://www.javathinker.net)
- QQ 学习群：915851077

## 目录

第 1 章	Vue 简介	10
1.1	MVVM 设计模式	12
1.2	Vue 框架的特点	13
1.3	第一个 Vue 范例	14
1.3.1	把模型数据绑定到视图	16
1.3.2	把视图上的输入数据与模型绑定	17
1.3.3	改变模型数据对视图的影响	18
1.4	Vue 组件的选项	19
1.4.1	data 选项	20
1.4.2	template 选项	20
1.4.3	methods 选项	22
1.5	Vue 组件实例的生命周期	24
1.6	Vue 的编译模板和渲染 DOM 的基本原理	29
1.6.1	编译模板	29
1.6.2	渲染 DOM	29
1.7	异步渲染 DOM	30
1.8	防抖动函数 <code>debounce()</code>	32
1.9	Vue 的开发和调试工具	35
1.9.1	NPM 软件包管理工具	35
1.9.2	vue-devtools 调试工具	36
1.10	小结	38
1.11	思考题	38
第 2 章	Vue 指令	41
2.1 节	内置 Vue 指令	41
2.1.1	v-bind 指令	41
2.1.2	v-model 指令	43
2.1.3	v-show 指令	45
2.1.4	v-if/v-else-if/v-else 指令	46
2.1.5	v-for 指令	49
2.1.6	v-on 指令	52
2.1.7	v-on 指令的事件修饰符	54

2.1.8	v-text 指令	57
2.1.9	v-html 指令	58
2.1.10	v-pre 指令	59
2.1.11	v-once 指令	59
2.1.12	v-cloak 指令	60
2.2	自定义 Vue 指令	61
2.2.1	注册自定义指令	61
2.2.2	自定义指令的钩子函数	62
2.2.3	自定义指令的动态参数和动态值	65
2.2.4	把对象字面量赋值给自定义指令	66
2.2.5	钩子函数简写	67
2.2.6	自定义指令范例: v-img 指令	67
2.2.7	自定义指令范例: v-drag 指令	68
2.2.8	自定义指令范例: v-clickoutside 指令	70
2.3	小结	73
2.4	思考题	74
第 3 章	计算属性和数据监听	76
3.1	计算属性	76
3.1.1	读写计算属性	77
3.1.2	比较计算属性和方法	79
3.1.3	用计算属性过滤数组	81
3.1.4	计算属性实用范例: 实现购物车	82
3.2	数据监听	85
3.2.1	用 Web Worker 执行数据监听中的异步操作	87
3.2.2	在 watch 选项中调用方法	90
3.2.3	比较同步操作和异步操作	91
3.2.4	深度监听	92
3.2.5	立即监听	93
3.2.6	比较计算属性和数据监听 watch 选项	93
3.3	Vue 的响应式系统的基本原理	95
3.4	小结	97
3.5	思考题	98
第 4 章	绑定表单	101
4.1	绑定文本域	101
4.2	绑定单选按钮	102
4.3	绑定复选框	103
4.4	下拉列表	106
4.5	把对象与表单绑定	108
4.6	小结	110
4.7	思考题	111
第 5 章	绑定 CSS 样式	112
5.1	绑定 class 属性	112
5.1.1	绑定对象类型的变量	114
5.1.2	绑定计算属性	115

5.1.3	绑定数组	116
5.1.4	为 Vue 组件绑定 CSS 样式	117
5.2	绑定 style 属性	117
5.2.1	绑定对象类型的变量	118
5.2.2	绑定数组	119
5.2.3	与浏览器兼容	119
5.3	范例：变换表格奇偶行的样式	121
5.4	小结	123
5.5	思考题	124
第 6 章	CSS 过渡和动画	126
6.1	CSS 过渡	127
6.1.1	为<transition>组件设定名字	129
6.1.2	为<transition>组件显式指定过渡样式类型	131
6.1.3	使用钩子函数和 Velocity 函数库	132
6.1.4	设置初始过渡效果	135
6.1.5	切换过渡的 DOM 元素	136
6.1.6	过渡模式	138
6.1.7	切换过渡的组件	140
6.2	CSS 动画	141
6.2.1	使用第三方的 CSS 动画样式类型	144
6.2.2	使用钩子函数和 Velocity 函数库	144
6.3	过渡组合组件<transition-group>	146
6.4	动态控制过渡和动画	149
6.5	小结	152
6.6	思考题	153
第 7 章	Vue 组件开发基础	155
7.1	注册全局组件和局部组件	155
7.1.1	注册全局组件	155
7.1.2	注册局部组件	157
7.2	组件的命名规则	159
7.3	向组件传递属性	161
7.3.1	传递动态值	161
7.3.2	对象类型的属性	162
7.3.3	数组类型的属性	163
7.3.4	绑定静态数据	163
7.3.5	传递对象	165
7.3.6	属性的不可改变性	166
7.3.7	单向数据流	168
7.3.8	属性验证	170
7.4	non-prop 属性	173
7.4.1	单节点模板中根节点对 non-prop 属性的继承	174
7.4.2	在单节点模板中禁止 non-prop 属性的继承	175
7.4.3	多节点模板中节点与 non-prop 属性的绑定	176
7.5	组件树	176

7.6	监听子组件的事件	180
7.6.1	验证事件	181
7.6.2	通过 v-model 指令绑定属性	183
7.6.3	通过 v-model 指令绑定多个属性	184
7.6.4	v-model 指令的自定义修饰符	185
7.6.5	处理子组件中 DOM 元素的原生事件	188
7.7	综合范例：自定义组件<combobox>	189
7.8	小结	191
7.9	思考题	191
第 8 章	Vue 组件开发高级技术	193
8.1	插槽<slot>	193
8.1.1	<slot>组件的渲染作用域	195
8.1.2	<slot>组件的默认内容	195
8.1.3	为<slot>组件命名	196
8.1.4	<slot>组件的动态名字	198
8.1.5	<slot>组件的自定义属性	198
8.2	动态组件<component>	201
8.3	异步组件	203
8.3.1	异步组件的选项	204
8.3.2	局部异步组件	206
8.4	组件的生命周期	206
8.5	组件的混入块	208
8.5.1	合并规则	209
8.5.2	全局混入块	212
8.5.3	自定义合并策略	212
8.5.4	使用混入块的注意事项	213
8.6	组件之间的互相访问	214
8.6.1	访问根组件	214
8.6.2	访问父组件	215
8.6.3	访问子组件	215
8.6.4	依赖注入	217
8.7	组件的递归	220
8.8	定义组件模板的其他方式	222
8.9	<teleport>组件与 DOM 元素的通信	222
8.9.1	在<teleport>组件中包裹子组件	224
8.9.2	多个<teleport>组件与同一个 DOM 元素通信	225
8.10	小结	226
8.11	思考题	227
第 9 章	虚拟 DOM 和 render()函数	229
9.1	render()函数	229
9.2	真实 DOM	231
9.3	虚拟 DOM	232
9.4	h()函数的用法	233
9.4.1	虚拟 DOM 中虚拟节点的唯一性	234

9.4.2	h()函数的完整范例	235
9.4.3	创建组件的虚拟节点	237
9.5	用 render()函数实现模板的一些功能	238
9.5.1	实现 v-if 和 v-for 指令的流程控制功能	238
9.5.2	实现 v-model 指令的数据绑定功能	239
9.5.3	实现 v-on 指令的监听事件功能	240
9.5.4	实现事件修饰符和按键修饰符的功能	241
9.5.5	实现插槽功能	242
9.5.6	生成动态组件的节点	245
9.5.7	自定义指令	246
9.6	在 render()函数中使用 JSX 语法	247
9.7	综合范例：博客帖子列表	249
9.8	小结	251
9.9	思考题	251
第 10 章	Vue CLI 脚手架工具	253
10.1	Vue CLI 简介以及安装	253
10.2	创建 Vue 项目	254
10.2.1	vue create 命令的用法	254
10.2.2	删除预配置	258
10.2.3	vue ui 命令的用法	259
10.3	Vue 项目的结构	259
10.3.1	单文件组件	260
10.3.2	程序入口 main.js 文件	262
10.3.3	项目的 index.html 文件和 SPA 单页应用	262
10.3.4	运行项目	263
10.4	安装和配置 Visual Studio Code	263
10.4.1	安装 Vetur 和 ESLint 插件	265
10.4.2	在 VSCode 中打开 helloworld 项目	266
10.4.3	在 VSCode 中运行 helloworld 项目	267
10.5	创建单文件组件<Hello>	268
10.5.1	创建 Hello.vue 文件	268
10.5.2	修改 App.vue 文件	268
10.5.3	运行修改后的 helloworld 项目	269
10.6	创建正式产品	269
10.7	在 Tomcat 中发布正式产品	270
10.7.1	安装 Tomcat	271
10.7.2	把 helloworld 正式产品发布到 Tomcat 中	271
10.8	小结	271
10.9	思考题	272
第 11 章	Vue Router 路由管理器	273
11.1	简单的路由管理	273
11.2	路由管理器的基本用法	275
11.3	在 Vue 项目中使用路由管理器	277
11.3.1	创建 Home.vue 和 About.vue 组件文件	278

11.3.2	在组件中加入图片	279
11.3.3	在 index.js 中创建路由管理器实例	279
11.3.4	在 main.js 中使用路由管理器	280
11.3.5	在 App.vue 中加入<router-link>和<router-view>组件	280
11.3.6	运行 helloworld 项目	280
11.4	路由模式	281
11.5	动态链接	283
11.5.1	链接中包含路径参数	283
11.5.2	链接中包含查询参数	285
11.5.3	链接与通配符匹配	286
11.6	嵌套的路由	287
11.6.1	创建 Items 父组件的文件 Items.vue	288
11.6.2	创建 Item 子组件的文件 Item.vue	289
11.6.3	在 index.js 中设置父组件和子组件的路由	291
11.6.4	在根组件的模板中加入 Items 父组件的导航链接	292
11.7	命名路由	292
11.7.1	重定向	293
11.7.2	使用别名	294
11.8	命名视图	294
11.9	向路由的组件传递属性	297
11.9.1	向命名视图的组件传递属性	298
11.9.2	通过函数传递属性	298
11.10	程式化导航	299
11.11	导航守卫函数	302
11.11.1	全局导航守卫函数	305
11.11.2	验证用户是否登录	306
11.11.3	设置受保护资源	309
11.11.4	在单页面应用中设置目标路由的页面标题	310
11.11.5	特定路由的导航守卫函数	311
11.11.6	组件内的导航守卫函数	311
11.12	数据抓取	313
11.12.1	导航后抓取	313
11.12.2	导航前抓取	315
11.13	设置页面的滚动行为	317
11.13.1	scrollBehavior()函数的返回值	317
11.13.2	延迟滚动	318
11.14	延迟加载路由	318
11.14.1	把多个组件打包到同一个文件中	320
11.14.2	在路由的组件中嵌套异步组件	320
11.15	动态路由	321
11.16	小结	323
11.17	思考题	324
第 12 章	组合 (Composition) API	326
12.1	setup()函数的用法	326

12.1.1	props 参数	331
12.1.2	context 参数	331
12.1.3	ref()函数	331
12.1.4	reactive()函数	332
12.1.5	toRefs()函数	333
12.1.6	readonly()函数	333
12.1.7	定义计算属性	334
12.1.8	注册组件的生命周期钩子函数	334
12.1.9	通过 watch()函数监听数据	336
12.1.10	通过 watchEffect()函数监听数据	337
12.1.11	获取模板中 DOM 元素的引用	338
12.1.12	依赖注入 (provide/inject)	339
12.2	分割 setup()函数	341
12.2.1	把 setup()函数分割到多个函数中	341
12.2.2	把 setup()函数分割到多个文件中	344
12.3	小结	345
12.4	思考题	345
第 13 章	通过 Axios 访问服务器	348
13.1	Axios 的基本用法	348
13.1.1	同域访问和跨域访问	349
13.1.2	获取响应结果	351
13.1.3	处理错误	352
13.2	在 Vue 项目中使用 Axios	353
13.2.1	异步请求	354
13.2.2	POST 请求方式	356
13.2.3	对象和查询字符串的转换	358
13.2.4	下载图片	359
13.2.5	上传文件	361
13.2.6	设置反向代理服务器	362
13.3	Axios API 的用法	363
13.4	请求配置	364
13.4.1	创建 axios 实例	367
13.4.2	设定默认的请求配置	368
13.4.3	请求配置的优先顺序	368
13.4.4	取消请求的令牌	369
13.5	并发请求	371
13.6	请求拦截器和响应拦截器	372
13.7	前端与后端的会话	375
13.7.1	通过 Cookie 跟踪会话	376
13.7.2	通过 token 令牌跟踪会话	376
13.8	前端与后端代码的整合	380
13.9	小结	382
13.10	思考题	382
第 14 章	通过 Vuex 进行状态管理	384

14.1	Vuex 的基本工作原理	384
14.2	Vuex 的基本用法	386
14.3	在 Vue 项目中使用 Vuex	388
14.3.1	strict 严格模式	389
14.3.2	通过计算属性访问状态	390
14.3.3	状态映射函数: <code>mapState()</code>	390
14.3.4	更新荷载 (payload)	392
14.3.5	更新映射函数: <code>mapMutations()</code>	393
14.3.6	把更新函数的名字设为常量	395
14.3.7	更新函数只能包含同步操作	396
14.4	仓库的 <code>getters</code> 选项	396
14.4.1	<code>getters</code> 映射函数: <code>mapGetters()</code>	398
14.4.2	为 <code>getters</code> 选项的属性设置参数	398
14.5	仓库的 <code>actions</code> 选项	399
14.5.1	传入更新荷载 (payload)	401
14.5.2	动作映射函数: <code>mapActions()</code>	402
14.6	异步动作	402
14.6.1	异步动作范例	403
14.6.2	使用 <code>async/await</code> 的范例	405
14.7	表单处理	407
14.7.1	在处理 <code>input</code> 事件的方法中提交更新函数	408
14.7.2	可读写的计算属性	409
14.8	仓库的模块化	409
14.8.1	模块的局部状态	411
14.8.2	访问根状态	412
14.8.3	命名空间	412
14.9	通过 <code>Composition API</code> 访问仓库	419
14.10	状态的持久化	420
14.11	小结	421
14.12	思考题	422
第 15 章	创建综合购物网站应用	424
15.1	前端组件的结构	424
15.2	前端开发技巧	427
15.2.1	状态管理	427
15.2.2	状态同步	430
15.2.3	运用 <code>Composition API</code> 提高代码可重用性	432
15.2.4	在组件中显示图片	433
15.2.5	路由管理	433
15.2.6	每个组件的页面标题	435
15.2.7	用户登录流程	435
15.2.8	受保护的资源	437
15.2.9	异步处理 <code>Axios</code> 的请求	438
15.2.10	单独运行前端项目	440
15.3	后端架构	441



15.3.1	实现业务数据	442
15.3.2	实现业务逻辑服务层	446
15.3.3	实现 DAO 层	449
15.3.4	实现控制器层	450
15.3.5	前端与后端的数据交换	453
15.4	发布和运行 netstore 应用	457
15.4.1	安装 SAMPLEDB 数据库	457
15.4.2	发布后端 netstore 项目	458
15.4.3	调试和运行前端 netstore 项目	458
15.4.4	创建并发布前端项目的正式产品	458
15.4.5	运行 netstore 应用	459
15.5	小结	463
附录 A	思考题答案	464