小程序

[第一章：Vue全家桶之Vue基础 7](#_Toc193281267)

[1.1 Vue概述 7](#_Toc1482980205)

[1.2 Vue基本使用 7](#_Toc753098353)

[1.2.1 传统开发模式对比 7](#_Toc55403453)

[1.2.2 vue.js之HelloWorld基本步骤 8](#_Toc1305415420)

[1.2.3 Vue.js之HelloWorld细节分析 8](#_Toc1424026188)

[1.3 Vue模板语法 9](#_Toc2050379548)

[1.3.1 模板语法概述 9](#_Toc58979827)

[1.3.1.1 如何理解前端渲染 9](#_Toc1283991122)

[1.3.1.2 前端渲染方式 9](#_Toc2123102398)

[1.3.1.3 原生js拼接字符串 9](#_Toc393724634)

[1.3.1.4 使用前端模板引擎 10](#_Toc932807231)

[1.3.1.5 模板语法概览 11](#_Toc1060508317)

[1.3.2 指令 11](#_Toc1996497366)

[1.3.2.1 什么是指令？ 11](#_Toc699245987)

[1.3.2.2 v-cloak指令用法 11](#_Toc1196787125)

[1.3.2.3 数据绑定指令 13](#_Toc1069372073)

[1.3.2.4 数据响应式 14](#_Toc645789168)

[1.3.3 双向数据绑定指令 15](#_Toc396194638)

[1.3.3.1 什么是双向数据绑定 15](#_Toc1643975166)

[1.3.3.2 双向数据绑定分析 15](#_Toc766012660)

[1.3.3.3 MVVM设计思想 16](#_Toc210312855)

[1.3.4 事件绑定 17](#_Toc2117554670)

[1.3.4.1 Vue如何处理事件 17](#_Toc1642340606)

[1.3.4.2 事件函数的调用方式 17](#_Toc1211250151)

[1.3.4.3 事件函数参数传递 19](#_Toc1483797944)

[1.3.4.4 事件修饰符 20](#_Toc1611935844)

[1.3.4.5 按键修饰符 22](#_Toc1299523203)

[1.3.4.6 自定义按键修饰符 24](#_Toc1177782831)

[1.3.5 案例：简单计算器 25](#_Toc1639266218)

[1.3.6 属性绑定 27](#_Toc1079618563)

[1.3.6.1 Vue如何动态处理属性 27](#_Toc1059854838)

[1.3.6.2 v-model的底层实现原理分析 28](#_Toc1750894048)

[1.3.7 样式绑定 29](#_Toc307849895)

[1.3.7.1 class样式处理 29](#_Toc745079642)

[1.3.7.2 style样式处理 35](#_Toc576397437)

[1.3.8 分支循环结构 37](#_Toc212992042)

[1.3.8.1 分支结构 37](#_Toc2049493992)

[1.3.8.2 v-if与v-show的区别 37](#_Toc207825664)

[1.3.8.3 循环结构 38](#_Toc1117524826)

[1.4 基础案例TAB选项卡 41](#_Toc347773920)

[1.5 Vue常用特性 46](#_Toc1733269953)

[1.5.1 常用特性概览 46](#_Toc452428516)

[1.5.2 表单操作 46](#_Toc1873958032)

[1.5.2.1 基于Vue的表单操作 46](#_Toc617476922)

[1.5.2.2 表单域修饰符 51](#_Toc1293645750)

[1.5.3 自定义指令 52](#_Toc1179678022)

[1.5.3.1 为何需要自定义指令？ 52](#_Toc1279486650)

[1.5.3.2自定义指令的语法规则（获取元素焦点） 53](#_Toc1578369139)

[1.5.3.3 带参数的自定义指令（改变元素背景色） 54](#_Toc1932111429)

[1.5.3.4 局部指令 56](#_Toc896560916)

[1.5.4 计算属性 57](#_Toc1754047860)

[1.5.4.1 为何要计算属性？ 57](#_Toc1774360651)

[1.5.4.2 计算属性的用法 58](#_Toc1721539115)

[1.5.4.3 计算属性与方法的区别？ 59](#_Toc860729774)

[1.5.5 侦听器 61](#_Toc835465426)

[1.5.5.1 侦听器的应用场景 61](#_Toc1419330696)

[1.5.5.2 侦听器的用法 61](#_Toc442656796)

[1.5.5.3 侦听器案例 64](#_Toc849417164)

[1.5.6 过滤器 66](#_Toc1830473739)

[1.5.6.1 过滤器的作用是什么？ 66](#_Toc2068888098)

[1.5.6.2 全局过滤器 67](#_Toc1894534509)

[1.5.6.3 局部过滤器 69](#_Toc701458694)

[1.5.6.4 带参数的过滤器 69](#_Toc1878531675)

[1.5.7 生命周期 70](#_Toc177283531)

[1.5.7.1 主要阶段 70](#_Toc1044487128)

[1.5.7.2 Vue实例的产生过程 71](#_Toc1163829718)

[1.5.7.3 梦学谷给的解释 71](#_Toc1205013550)

[1.6综合案例 77](#_Toc1891943640)

[第二章：Vue全家桶之组件化开发 77](#_Toc106396351)

[2.1 组件化开发思想 77](#_Toc1497076953)

[2.1.1 现实中的组件化思想体现 77](#_Toc1453940819)

[2.1.2 编程中的组件化思想体现 77](#_Toc166925720)

[2.1.3 组件化规范：Web Components 78](#_Toc906933058)

[2.2 组件注册 78](#_Toc2132463047)

[2.2.1 全局组件注册语法 78](#_Toc951846146)

[2.2.2 组件注册注意事项 79](#_Toc1072489319)

[2.2.3 局部组件注册 82](#_Toc1497735162)

[2.3 Vue调试工具用法 84](#_Toc1779041247)

[2.4 组件间数据交互 84](#_Toc931421148)

[2.4.1 父组件向子组件传值 84](#_Toc1386931453)

[2.4.2 子组件向父组件传值 90](#_Toc1369426033)

[2.4.3 非父子组件间传值 94](#_Toc1361091732)

[2.5 组件插槽 98](#_Toc872931880)

[2.5.1 组件插槽的作用 98](#_Toc1905314503)

[2.5.2 组件插槽的基本用法 99](#_Toc1492191504)

[2.5.3 具名插槽的用法 100](#_Toc948578062)

[2.5.4 作用域插槽 103](#_Toc1980376353)

[2.6 基于组件的案例 105](#_Toc336320018)

[第三章：Vue全家桶之前后台交互 106](#_Toc353583622)

[3.1 前后端交互模式 106](#_Toc592683705)

[3.1.1 接口调用方式 106](#_Toc1205875149)

[3.1.2 URL 地址格式 106](#_Toc1340452504)

[3.2 Promise用法 107](#_Toc1881777698)

[3.2.1 异步调用 107](#_Toc1046100917)

[3.2.2 Promise概述 108](#_Toc369494030)

[3.2.3 Promise基本用法 108](#_Toc1710938733)

[3.2.4 基于Promise处理Ajax请求 110](#_Toc941252201)

[3.2.5 then参数中的函数返回值 113](#_Toc1261198405)

[3.2.6 Promise常用的API 115](#_Toc1297996945)

[3.2.6.1 实例方法 115](#_Toc1295768389)

[3.2.6.2 对象方法 117](#_Toc347649696)

[3.3 接口调用-fetch用法 119](#_Toc1792920832)

[3.3.1 fetch概述 119](#_Toc129888720)

[3.3.2 fetch的基本用法 120](#_Toc1196331688)

[3.3.3 fetch请求参数 121](#_Toc2004777002)

[3.3.4 fetch响应结果 125](#_Toc268651184)

[3.4 接口调用-axios用法 126](#_Toc1209823494)

[3.4.1 axios的基本特性 126](#_Toc1128293862)

[3.4.2 axios的基本用法 126](#_Toc954335624)

[3.4.3 axios的常用API 127](#_Toc2110956772)

[3.4.4 axios的参数传递 127](#_Toc273134917)

[3.4.5 axios的响应结果 131](#_Toc1405996380)

[3.4.6 axios的全局配置 132](#_Toc1818590719)

[3.4.7 axios拦截器 133](#_Toc2066950129)

[3.5 接口调用-async/await用法 135](#_Toc1535344231)

[3.5.1 async/await的基本用法 135](#_Toc366988065)

[3.5.2 async/await处理多个异步请求 136](#_Toc395374271)

[3.6 基于接口的案例 138](#_Toc740968879)

[第四章：Vue全家桶之前端路由 138](#_Toc206280400)

[4.1 路由的基本概念与原理 138](#_Toc916076542)

[4.1.1 路由 138](#_Toc1188176051)

[4.1.1.1 后端路由 138](#_Toc224455704)

[4.1.1.2 SPA(Single Page Application) 139](#_Toc1445732996)

[4.1.1.3 前端路由 139](#_Toc1804481614)

[4.1.1.4 实现简易前端路由 139](#_Toc1158423564)

[4.1.2 Vue Router 143](#_Toc538096446)

[4.2 vue-router的基本使用 143](#_Toc733330405)

[4.2.1 基本使用步骤 143](#_Toc675466702)

[4.2.2 路由重定向 147](#_Toc970302472)

[4.3 vue-router嵌套路由 148](#_Toc2030315233)

[4.4 vue-router动态路由匹配 153](#_Toc2140453848)

[4.4.1 动态匹配路由的基本用法 153](#_Toc2109252439)

[4.4.2 路由组件传递参数 156](#_Toc1693181244)

[4.5 vue-router命名路由 163](#_Toc991361511)

[4.6 vue-router编程式导航 165](#_Toc1634781951)

[4.6.1 页面导航的两种方式 165](#_Toc874470739)

[4.6.2 编程式导航基本用法 166](#_Toc1999113952)

[4.6.3 编程式导航参数规则 166](#_Toc1726533949)

[4.7 基于vue-router的案例 169](#_Toc1057042579)

[第五章：Vue全家桶-前端工程化 170](#_Toc1729897269)

[5.1 模块化相关规范 170](#_Toc1749786997)

[5.1.1 模块化概述 170](#_Toc1028996561)

[5.1.2 浏览器模块化规范 170](#_Toc659391436)

[5.1.3 服务端模块化规范 170](#_Toc1376246332)

[5.1.4 大一统的模块化规范-ES6模块化 171](#_Toc25740087)

[5.1.5 ES6模块化的基本语法 171](#_Toc969429162)

[5.2 webpack 175](#_Toc237495945)

[5.2.1 当前web开发面临的困境 175](#_Toc1569731489)

[5.2.2 webpack概述 176](#_Toc640532228)

[5.2.3 webpack的基本使用 176](#_Toc89633585)

[5.3 Vue单文件组件 177](#_Toc1085626548)

[5.3.1 传统组件的问题和解决方案 177](#_Toc1104327324)

[5.3.2 Vue单文件组件的基本用法 178](#_Toc1875657094)

[5.3.3 webpack中配置vue组件的加载器 178](#_Toc1256324545)

[5.3.4 在webpack项目中使用vue 179](#_Toc987410511)

[5.3.5 webpack打包发布 179](#_Toc1802318008)

[5.4 Vue脚手架 180](#_Toc1301919521)

[5.4.1 Vue脚手架的基本用法 180](#_Toc650510164)

[5.4.2 Vue 脚手架生成的项目结构分析 180](#_Toc285079471)

[5.4.3 Vue脚手架的自定义配置 181](#_Toc294652640)

[5.4.3.1 通过package.json配置项目（不推荐） 181](#_Toc129630498)

[5.4.3.2 通过单独的配置文件配置项目 181](#_Toc1151361828)

[5.5 Element-UI的基本使用 181](#_Toc2110583726)

[5.5.1 基于命令行方式手动安装 182](#_Toc445801736)

[5.5.2 基于图形化界面自动安装 182](#_Toc19332569)

[第六章：电商管理系统 182](#_Toc652456486)

[6.1 项目实战（上） 182](#_Toc784658620)

[6.1.1 项目概述 182](#_Toc60350113)

[6.1.1.1 电商项目基本业务概述 182](#_Toc692067807)

[6.1.1.2 电商后台管理系统的功能 183](#_Toc812200097)

[6.1.1.3 电商后台管理系统的开发模式（前后端分离） 183](#_Toc1240969947)

[6.1.1.4 电商后台管理系统的技术选型 184](#_Toc620719565)

[6.1.2 项目初始化 184](#_Toc2105655476)

[6.1.2.1 前端项目初始化步骤 184](#_Toc1368566219)

[6.1.2.2 后台项目的环境安装配置 185](#_Toc1942583363)

[6.1.3 登录/退出功能 185](#_Toc804696600)

[6.1.3.1 登录概述 185](#_Toc1831231041)

[6.1.3.2 登录--token原理分析 185](#_Toc1911960930)

[6.1.3.3 登录功能实现 186](#_Toc1529540449)

[6.1.3.4 退出 187](#_Toc1607071753)

[6.1.4 主页布局 187](#_Toc1153124352)

[6.1.4.1 整体布局 187](#_Toc1668553536)

[6.1.4.2 左侧菜单布局 188](#_Toc1537817026)

[6.1.4.3 通过接口获取菜单数据 188](#_Toc1125064337)

[6.1.4.4 动态渲染菜单数据并进行路由控制 189](#_Toc362800124)

[6.1.5 用户管理模块 192](#_Toc875610235)

[6.1.5.1 用户管理概述 192](#_Toc1823270401)

[6.1.5.2 用户管理-列表展示 193](#_Toc1261470564)

[第七章：Vue全家桶-Vuex 193](#_Toc1577205964)

[7.1 Vuex 概述 193](#_Toc1709982027)

[7.1.1 组件之间共享数据的方式 193](#_Toc2041763635)

[7.1.2 Vuex是什么 194](#_Toc1280218032)

[7.1.3 使用Vuex统一管理状态的好处 194](#_Toc985804531)

[7.1.4 什么样的数据适合存储到Vuex中 194](#_Toc580415912)

[7.2 Vuex基本使用 194](#_Toc1179508310)

[7.3 Vuex的核心概念 195](#_Toc574620713)

[7.3.1 核心概念概述 195](#_Toc416362832)

[7.3.2 State 195](#_Toc1308395498)

[7.3.3 Mutation 196](#_Toc2118073253)

[7.3.3.1触发 Mutation第一种方式this.$store.commit() 196](#_Toc1768230499)

[7.3.3.2 Mutation传递参数 198](#_Toc1771289507)

[7.3.3.3 触发mutation的第二种方式mapMutations 201](#_Toc1644429435)

[7.3.4 Action 204](#_Toc1958460802)

[7.3.4.1 Action基本用法 204](#_Toc1368841645)

[7.3.4.2 Action携带参数 207](#_Toc129217204)

[7.3.4.3 Action触发的第二种方式 211](#_Toc647580511)

[7.3.5 Getter 212](#_Toc438525381)

# 第一章：小程序起步

### 小程序简介

###### 小程序与普通网页开发的区别



###### 体验小程序

可使用手机微信(6.7.2 及以上版本)扫码下方小程序码，体验小程序：



### 1.2 第一个小程序

###### 1.2.1 注册小程序开发账号

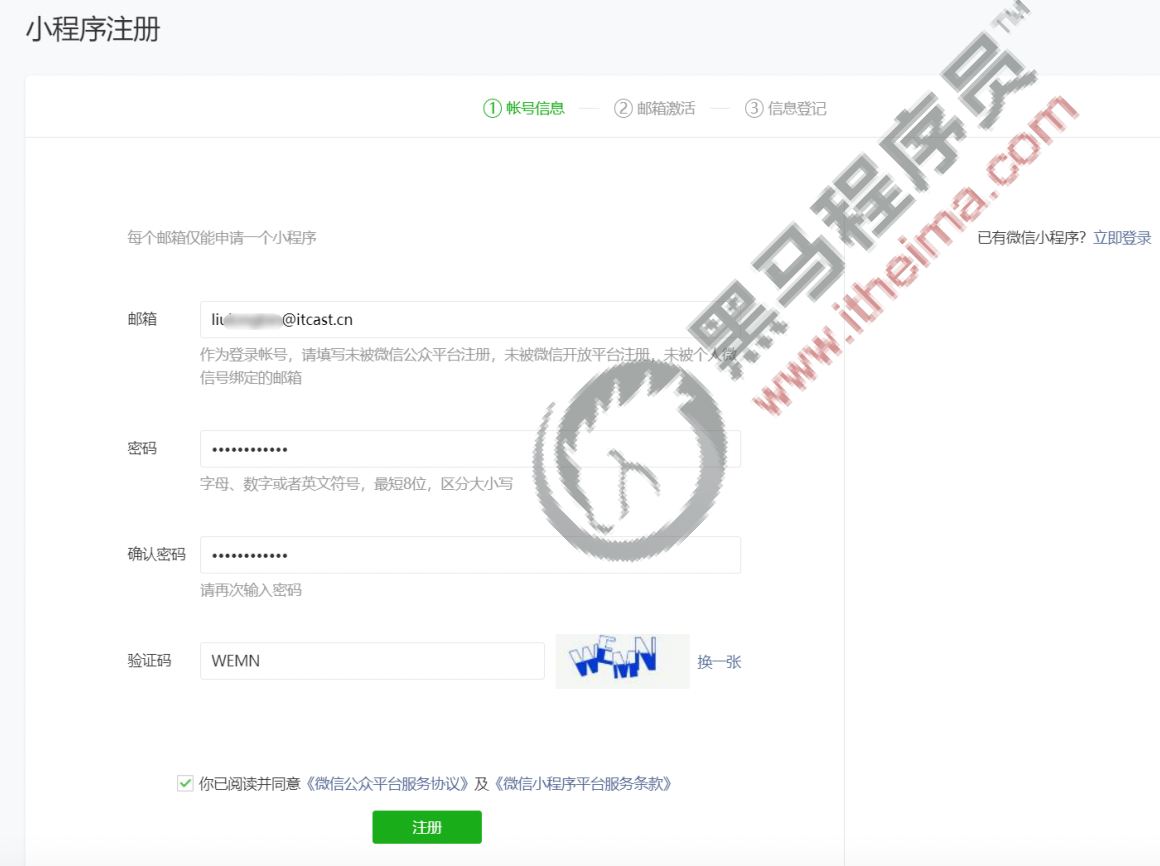
1. 点击注册按钮



1. 选择注册账号的类型



1. 填写账号信息



1. 提示邮箱激活



1. 点击链接激活账号



1. 选择主体类型



1. 主体信息登记

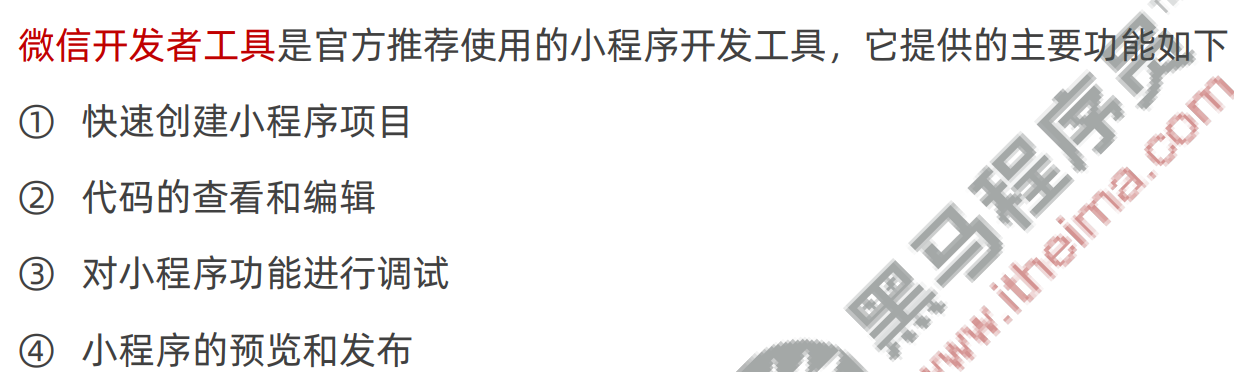


1. 获取小程序的AppID



###### 1.2.2 安装开发者工具

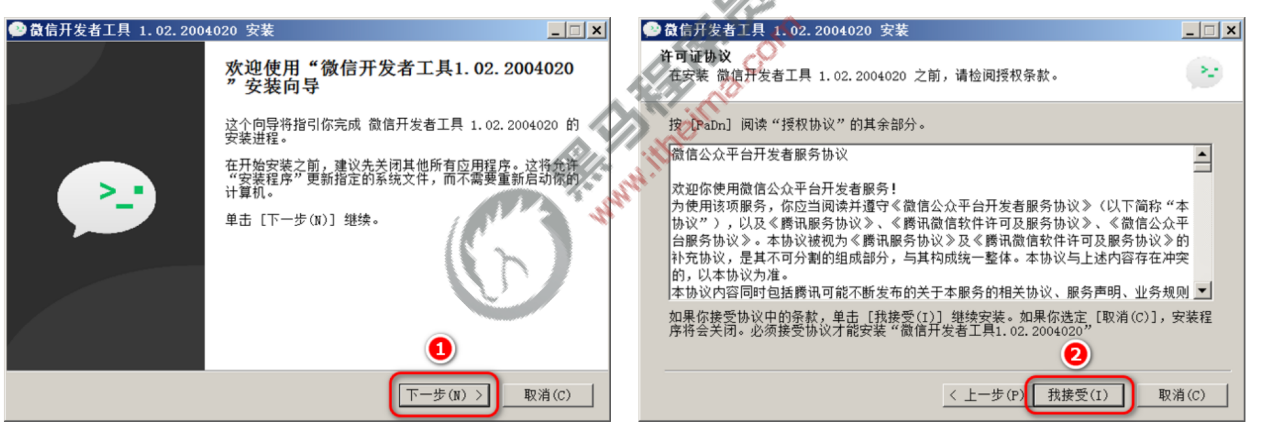
（1）了解微信开发者工具

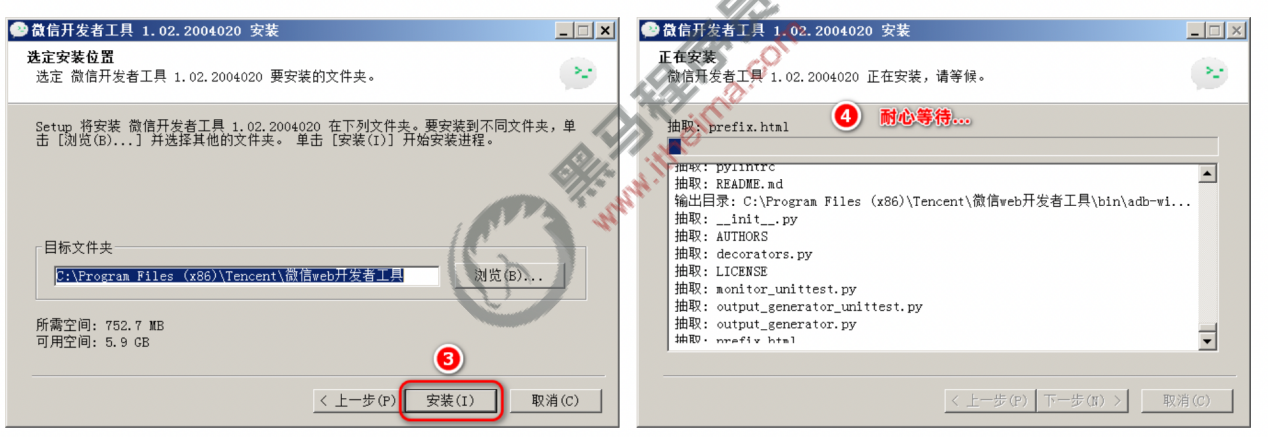


1. 下载



1. 安装







1. 扫码登录





1. 设置外观和代理



###### 1.2.3 创建小程序项目