



# 信息工程系

# 教

# 案

课程名称: 微信小程序开发

教 师: 黄梅佳

总 学 时 : 54

理论学时: 0

实训学时: 54

上课班级: 计算机应用技术 241

授课学期: 2025-2026 第二学期

# 第 1 章 微信小程序入门

课题名称	第 1 章 微信小程序入门	计划课时	3 课时
教学引入	<p>微信小程序于 2017 年 1 月 9 日正式上线，凭借其开发成本低、微信用户数量庞大等优势，得到了许多用户的认可，同时还为许多商家提供了商机。为了满足人们的日常需求，微信小程序的开发技术也在不断更新。为了让读者对微信小程序有一个整体的认识，本章将对微信小程序的入门知识进行详细讲解。</p>		
教学目标	<ul style="list-style-type: none"> <li>● 使学生了解微信小程序，能够说出微信小程序的概念、特点、发展前景和宿主环境</li> <li>● 使学生掌握微信小程序开发账号的注册方法，能够独立完成微信小程序开发账号的注册</li> <li>● 使学生掌握获取微信小程序 AppID 的方法，能够从微信小程序管理后台获取 AppID</li> <li>● 使学生掌握微信开发者工具的安装方法，能够独立完成微信开发者工具的安装</li> <li>● 使学生掌握微信小程序项目的创建方法，能够使用微信开发者工具创建项目</li> <li>● 使学生熟悉微信小程序的项目结构，能够解释每个文件的作用</li> <li>● 使学生熟悉微信小程序的页面组成，能够解释 WXML、WXSS、JS 和 JSON 文件的作用</li> <li>● 使学生熟悉微信小程序的通信模型，能够解释微信小程序中渲染层、逻辑层及第三方服务器的通信方式</li> <li>● 使学生熟悉微信开发者工具的主界面，能够说出工具栏中常用快捷按钮的功能</li> <li>● 使学生掌握微信小程序的项目设置，能够根据需要对微信小程序进行设置</li> <li>● 使学生了解微信小程序开发常用快捷键，能够列举 4 类常用快捷键</li> <li>● 使学生了解项目成员，能够说出项目成员的组织结构、分工和权限</li> <li>● 使学生掌握添加项目成员和体验成员的方法，能够在微信小程序管理后台中添加项目成员和体验成员</li> <li>● 使学生熟悉微信小程序的版本，能够说明微信小程序的 4 种版本</li> <li>● 使学生熟悉微信小程序发布上线的流程，能够归纳出微信小程序发布上线的步骤</li> </ul>		
教学重点	<ul style="list-style-type: none"> <li>● 获取微信小程序 AppID</li> <li>● 创建微信小程序项目</li> </ul>		
教学难点	无		

教学方式	课堂教学以 PPT 讲授为主，并结合多媒体进行教学
思政元素	<p>科技伦理与绿色发展意识：对比微信小程序与其他应用时，重点强调小程序无需安装、节省手机存储空间的优势，引导学生认识到技术开发不仅要追求功能实现，更要注重资源节约与环境保护，将绿色发展理念融入技术选择与产品设计中，培养学生的社会责任感与可持续发展思维。</p> <p>职业规范与团队协作意识：讲解项目成员组织结构、分工与权限时，结合实际开发场景，强调不同角色（如产品组、开发组、测试组）在项目中的职责与协作流程，引导学生理解团队协作的重要性，树立规范的职业分工观念，培养学生的团队协作能力与责任担当。</p> <p>法治与合规意识：在讲解微信小程序注册、AppID 获取及版本发布流程时，介绍微信公众平台的相关规则与审核标准，引导学生认识到技术开发需遵守平台规范与行业法规，培养学生的法治思维与合规开发意识，避免因违规操作导致项目风险。</p>
教学过程	<p style="text-align: center;"><b>第一课时</b></p> <p style="text-align: center;"><b>（什么是微信小程序、微信小程序的特点、微信小程序的发展前景、微信小程序的宿主环境）</b></p> <p>一、通过直接引入的方式导入新课</p> <p>目前，微信的普及程度已经很高了，人们通过微信可以聊天、工作、支付等。微信小程序是一种运行在微信中的程序，因其独特优势受到了人们的广泛关注。本节课将对微信小程序进行详细讲解。</p> <p>对比微信小程序与其他应用时：引导学生思考微信小程序在便捷性、环保性等方面的优势。如小程序无需安装节省手机存储空间，减少资源浪费，体现绿色发展理念。让学生明白在技术开发中，不仅要追求功能实现，还要注重资源节约和环境保护，培养学生的环保意识和可持续发展观念。</p> <p>二、新课讲解</p> <p><b>知识点 1-什么是微信小程序</b></p> <p>教师通过 PPT 的方式讲解什么是微信小程序。</p> <p>(1) 微信小程序的概念。</p> <p>(2) 微信小程序的发展状况。</p> <p>(3) 打开微信小程序的途径。</p> <ul style="list-style-type: none"> <li>● 搜索关键词。</li> <li>● 扫码。</li> <li>● 群分享。</li> <li>● 好友分享。</li> </ul> <p><b>知识点 2-微信小程序的特点</b></p> <p>教师通过 PPT 的方式讲解微信小程序的特点以及微信小程序和原生应用、HTML5 应用的对比。</p> <p>(1) 微信小程序的特点。</p> <ul style="list-style-type: none"> <li>● 无须安装、触手可及。</li> <li>● 用完即走、无须卸载。</li> <li>● 名称唯一。</li> <li>● 入口丰富。</li> <li>● 传播能力强。</li> </ul>

- (2) 微信小程序与原生应用的对比。
- (3) 微信小程序与 HTML5 应用的对比。

#### 知识点 3-微信小程序的发展前景

教师通过 PPT 的方式讲解微信小程序的发展前景。

- (1) 生态体系。
- (2) 开放能力。
- (3) 用户黏性。

#### 知识点 4-微信小程序的宿主环境

教师通过 PPT 的方式讲解微信小程序的宿主环境。

- (1) 宿主环境的概念。
- (2) iOS 系统和 Android 系统提供了两种不同的宿主环境。
- (3) 微信小程序的宿主环境。
- (4) 微信小程序的宿主环境为微信小程序提供的内容。
  - 组件。
  - API。

#### 三、归纳总结

教师回顾本节课所讲的知识，并通过测试题的方式引导学生解答问题并给予指导。

#### 四、布置作业

教师通过学习通或学习通或高校教辅平台（<http://tch.ityx.com>）布置本节课作业以及下节课的预习作业。

### 第二课时

#### （注册微信小程序开发账号、获取微信小程序 AppID、安装微信开发者工具、创建微信小程序项目）

#### 一、复习巩固

教师通过上节课作业的完成情况，对学生吸收不好的知识点进行再次巩固讲解。

#### 二、通过直接引入的方式导入新课

为了帮助开发者简单高效地实现微信小程序的开发，微信团队提供了一套微信开发者工具。该工具集成了公众号网页调试和小程序调试两种开发模式。在正式开发微信小程序前，需要先进行开发前的准备工作。本节将对微信小程序开发前的准备工作进行详细讲解。

#### 三、新课讲解

##### 知识点 1-注册微信小程序开发账号

教师通过 PPT 结合实际操作的方式讲解注册微信小程序开发账号的步骤。

- (1) 使用浏览器访问微信公众平台的官方网站。
- (2) 单击“立即注册”链接，进入开发账号的注册页面，选择注册的账号类型。
- (3) 单击“小程序”链接，跳转到小程序注册页面。
- (4) 根据提示填入自己的邮箱、密码、确认密码等，完成账号信息的填写。
- (5) 单击“注册”按钮，进入提示邮箱激活页面。
- (6) 单击“登录邮箱”按钮，登录步骤（4）中填写的邮箱，查看收到的账号激活链接。

(7) 单击激活链接，将跳转到用户信息登记页面，根据页面提示进行用户信息登记。

(8) 填写完用户相关信息后，单击“继续”按钮，进入微信小程序管理后台。

### 知识点 2-获取微信小程序 AppID

教师通过 PPT 结合实际操作的方式讲解什么是 AppID 以及获取微信小程序 AppID 的方法。

(1) AppID 的概念。

(2) 获取微信小程序 AppID 的方法。

- 登录微信小程序管理后台。
- 在左侧边栏中选择“开发管理”。
- 选择“开发设置”，即可查看 AppID。

### 知识点 3-安装微信开发者工具

教师通过 PPT 结合实际操作的方式讲解微信开发者工具的安装方法。

(1) 在微信小程序管理后台的左侧边栏中选择“开发工具”，然后选择“开发工具”，找到微信开发者工具的下载页面。

(2) 单击“下载”按钮，跳转到微信开发者工具的下载链接页面。

(3) 单击稳定版的“Windows 64”链接下载该版本的微信开发者工具安装包。

(4) 双击微信开发者工具的安装包，进入微信开发者工具的安装向导，根据安装向导进行操作，完成微信开发者工具的安装。

### 知识点 4-创建微信小程序项目

教师通过 PPT 结合实际操作的方式讲解创建微信小程序项目的步骤。

(1) 首次打开微信开发者工具时出现一个登录页。在登录界面中使用微信扫码登录微信开发者工具。

(2) 使用微信扫码登录成功后进入微信开发者工具的项目选择界面。

(3) 单击“+”进入微信小程序项目的创建页面，将项目信息填写完成后，单击“确定”按钮创建微信小程序项目。

## 四、归纳总结

教师回顾本节课所讲的知识，并通过测试题的方式引导学生解答问题并给予指导。

## 五、布置作业

教师通过学习通或学习通或高校教辅平台 (<http://tch.ityx.com>) 布置本节课作业以及下节课的预习作业。

## 第三课时

(微信小程序的项目结构、微信小程序的页面组成、微信小程序的通信模型、认识微信开发者工具、微信小程序的项目设置、微信小程序开发常用快捷键)

### 一、复习巩固

教师通过上节课作业的完成情况，对学生吸收不好的知识点进行再次巩固讲解。

### 二、通过直接引入的方式导入新课

在正式进入微信小程序开发前，大家需要对微信小程序的项目结构、页面

组成以及和通信模型以及微信开发者工具的使用有一个基本的认识。本节将对微信小程序开发基础和微信开发者工具的使用进行详细讲解。

以优秀小程序项目为例，展示开发者如何精心设计项目结构和页面，注重细节，追求完美。引导学生在学习和实践中养成严谨认真的学习态度和工作作风，对待代码编写和项目开发要一丝不苟，培养学生的工匠精神。

### 三、新课讲解

#### 知识点 1-微信小程序的项目结构

教师通过 PPT 的方式讲解微信小程序的项目结构以及微信客户端启动微信小程序的流程。

##### (1) 微信小程序的项目结构。

- pages。
- .eslintrc.js。
- app.js。
- app.json。
- app.wxss。
- project.config.json。
- project.private.config.json。
- sitemap.json。

##### (2) 微信客户端启动微信小程序的流程。

- 首先把整个微信小程序的代码包下载到本地。
- 然后解析 app.json 全局配置文件，通过该文件解析出微信小程序的所有页面路径。
- 接着执行 app.js 入口文件，调用 App()函数创建微信小程序的实例。
- 最后渲染微信小程序的首页。

#### 知识点 2-微信小程序的页面组成

教师通过 PPT 的方式讲解微信小程序的页面组成部分、编写语言和微信客户端加载微信小程序页面的顺序。

##### (1) 微信小程序的页面组成部分（以 index 页面为例）。

- index.js。
- index.json。
- index.wxml。
- index.wxss。

##### (2) 编写微信小程序页面用到的 4 种语言。

- JS。
- JSON。
- WXML。
- WXSS。

##### (3) 微信客户端加载微信小程序页面的顺序。

- 首先读取并解析页面中 JSON 文件的配置。
- 然后加载页面的 WXML 文件、WXSS 文件和 JS 文件，实现页面渲染。

#### 知识点 3-微信小程序的通信模型

教师通过 PPT 的方式讲解微信小程序的通信模型。

	<p>(1) 实现微信小程序通信的组成部分。</p> <ul style="list-style-type: none"> <li>● 渲染层。</li> <li>● 逻辑层。</li> <li>● 第三方服务器。</li> </ul> <p>(2) 通信模型的组成部分。</p> <ul style="list-style-type: none"> <li>● 渲染层与逻辑层之间的通信。</li> <li>● 逻辑层与第三方服务器之间的通信。</li> </ul> <p><b>知识点 4-认识微信开发者工具</b> 教师通过 PPT 结合实际操作的方式介绍微信开发者工具主界面的组成部分。</p> <p>(1) 菜单栏。</p> <p>(2) 工具栏。</p> <p>(3) 模拟器。</p> <p>(4) 编辑器。</p> <p>(5) 调试器。</p> <p><b>知识点 5-微信小程序的项目设置</b> 教师通过 PPT 的方式讲解微信小程序的项目设置。</p> <p>(1) 基本信息的设置。</p> <p>(2) 性能分析。</p> <p>(3) 本地设置。</p> <p>(4) 项目配置。</p> <p><b>知识点 6-微信小程序开发常用快捷键</b> 教师通过 PPT 的方式讲解微信小程序开发常用快捷键。</p> <p>(1) 项目和文件相关的快捷键。</p> <p>(2) 编辑相关的快捷键。</p> <p>(3) 工具相关的快捷键。</p> <p>(4) 界面相关的快捷键。</p> <p>四、归纳总结 教师回顾本节课所讲的知识，并通过测试题的方式引导学生解答问题并给予指导。</p> <p>五、布置作业 教师通过学习通或学习通或高校教辅平台（<a href="http://tch.itymb.com">http://tch.itymb.com</a>）布置本节课作业以及下节课的预习作业。</p>
教学后记	<p>本次课程围绕微信小程序入门知识展开,通过理论讲解与实操演示结合的方式,帮助学生掌握了开发账号注册、开发者工具安装、项目创建等基础操作。从课堂反馈来看,学生对微信小程序的概念、特点及项目结构理解较为清晰,但在获取 AppID 及创建项目时,部分学生因对微信公众平台后台操作不熟悉出现卡顿。后续教学中,需增加平台后台操作的分步演示,同时提供操作指南文档供学生课后参考。</p>

## 第 2 章 微信小程序页面制作

课题名称	第 2 章 微信小程序页面制作	计划课时	8 课时
教学引入	<p>若想开发一个微信小程序，首先应学习如何进行页面制作，也就是学习如何搭建微信小程序的页面结构并实现美观的页面样式效果。为了使初学者快速掌握微信小程序的页面制作，本章将从微信小程序常用组件、页面样式等方面进行详细讲解。</p>		
教学目标	<ul style="list-style-type: none"> <li>● 使学生了解 WXML 的概念，能够说出 WXML 的特点、WXML 与 HTML 的区别</li> <li>● 使学生了解 WXSS 的概念，能够说出 WXSS 的特点、WXSS 与 CSS 的区别</li> <li>● 使学生了解组件的概念，能够说出组件的特点及常用的组件</li> <li>● 使学生掌握页面路径的配置方法，能够运用该方法进行页面管理</li> <li>● 使学生掌握 view 组件的使用方法，能够灵活运用 view 组件实现页面的布局效果</li> <li>● 使学生掌握 image 组件的使用方法，能够灵活运用 image 组件完成图片插入操作</li> <li>● 使学生掌握 rpx 单位的使用方法，能够灵活运用 rpx 单位解决屏幕适配的问题</li> <li>● 使学生掌握页面样式的导入方法，能够灵活运用该方法导入公共样式</li> <li>● 使学生掌握 swiper 和 swiper-item 组件的使用方法，能够灵活运用 swiper 和 swiper-item 组件完成轮播图的制作</li> <li>● 使学生掌握 text 组件的使用方法，能够灵活运用 text 组件定义行内文本</li> <li>● 使学生掌握 Flex 布局的使用方法，能够使用 Flex 布局的相关属性完成页面布局</li> <li>● 使学生掌握导航栏的配置方法，能够完成导航栏标题颜色、背景颜色等页面效果的设置</li> <li>● 使学生掌握标签栏的配置方法，能够完成页面标签栏的配置</li> <li>● 使学生掌握 vw、vh 单位的使用方法，能够灵活运用 vw、vh 单位设置宽度和高度</li> <li>● 使学生掌握 video 组件的使用方法，能够灵活运用 video 组件实现页面中视频的处理</li> <li>● 使学生掌握表单组件的使用方法，能够灵活运用表单组件完成表单页面的制作</li> </ul>		
教学重点	<ul style="list-style-type: none"> <li>● 页面路径配置</li> <li>● view 组件</li> <li>● image 组件</li> <li>● 【案例 2-1】个人信息-实现“个人信息”微信小程序页面结构</li> </ul>		

	<ul style="list-style-type: none"> <li>● swiper 和 swiper-item 组件</li> <li>● 表单组件</li> </ul>
教学难点	<ul style="list-style-type: none"> <li>● image 组件</li> </ul>
教学方式	课堂教学以 PPT 讲授为主，并结合多媒体进行教学
思政元素	<p>工匠精神与精益求精意识：以“个人信息”“本地生活”“婚礼邀请函”等案例开发为载体，强调页面结构搭建、样式设计的细节把控，如 Flex 布局的精准应用、rpx 单位的屏幕适配、表单组件的规范使用等，引导学生树立精益求精的工匠精神，培养学生对技术细节的专注力与严谨的开发态度。</p> <p>文化传承与审美素养：在“婚礼邀请函”案例开发中，引导学生结合传统婚礼文化元素（如“喜”字、传统配色等）设计页面样式，将传统文化与现代技术融合，既提升学生的页面审美设计能力，又增强学生对中华优秀传统文化的认同感与传承意识。</p> <p>用户至上与服务意识：讲解导航栏、标签栏配置及表单组件使用时，强调从用户体验角度出发设计页面交互，如导航栏标题清晰明了、表单填写便捷高效等，引导学生树立“用户至上”的服务理念，培养学生以用户需求为导向的产品开发思维。</p>
教学过程	<p style="text-align: center;"><b>第一课时</b></p> <p style="text-align: center;"><b>(WXML 简介、WXSS 简介、常用组件、页面路径配置、view 组件)</b></p> <p>一、复习巩固</p> <p>教师通过上节课作业的完成情况，对学生吸收不好的知识点进行再次巩固讲解。</p> <p>二、通过情景引入的方式导入新课</p> <p>大学毕业后，许多大学生都将迈出校园去求职，为了让招聘人员快速地了解自己，可以做一个“个人信息”微信小程序，展示自己的个人信息。本节课将详细讲解“个人信息”微信小程序用到的相关知识点。</p> <p>三、新课讲解</p> <p><b>知识点 1-WXML 简介</b></p> <p>教师通过 PPT 的方式讲解 WXML 简介。</p> <p>(1) WXML 作用。</p> <p>(2) WXML 与 HTML 的区别。</p> <p><b>知识点 2-WXSS 简介</b></p> <p>教师通过 PPT 的方式讲解 WXSS。</p> <p>(1) WXSS 作用。</p> <p>(2) WXSS 与 CSS 的区别。</p> <p><b>知识点 3-常用组件</b></p> <p>教师通过 PPT 的方式讲解常用组件。</p> <p>(1) 组件的概念。</p> <p>(2) 常用组件。</p> <ul style="list-style-type: none"> <li>● view 组件。</li> <li>● text 组件。</li> </ul>

- button 组件。
- image 组件。
- form 组件。
- video 组件。
- checkbox 组件。
- radio 组件。
- input 组件。
- audio 组件。

#### 知识点 4-页面路径配置

教师通过 PPT 结合实际操作的方式讲解页面路径配置。

- (1) 页面路径配置的方法。
- (2) 通过示例代码演示如何配置页面路径。
- (3) pages 数组中每个元素的含义。
- (4) 创建一个新页面的方法。

#### 知识点 5-view 组件

教师通过 PPT 结合实际操作的方式讲解 view 组件。

- (1) view 组件的作用。
- (2) 定义 view 组件的示例代码。
- (3) 列举 view 组件的常用属性。
- (4) 通过示例代码演示以 hover-class、hover-start-time 和 hover-stay-time 属性的使用。

#### 四、归纳总结

教师回顾本节课所讲的内容，并通过测试题的方式引导学生解答问题并给予指导。

#### 五、布置作业

教师通过学习通或学习通或高校教辅平台（<http://tch.ityx.com>）布置本节课作业以及下节课的预习作业。

### 第二课时

**（image 组件、rpx 单位、样式导入、【案例 2-1】个人信息-准备工作、【案例 2-1】个人信息-实现“个人信息”微信小程序的页面结构、【案例 2-1】个人信息-实现“个人信息”微信小程序的页面样式）**

#### 一、复习巩固

教师通过上节课作业的完成情况，对学生吸收不好的知识点进行再次巩固讲解。

#### 二、通过直接引入的方式导入新课

在上节课中学习了“个人信息”微信小程序用到的相关知识点，包括 WXML 简介、WXSS 简介、常用组件、页面路径配置、view 组件的相关内容，在本节课中将继续详细讲解剩余的知识点及“个人信息”微信小程序的开发。

#### 三、新课讲解

##### 知识点 1-image 组件

教师通过 PPT 结合实际操作的方式讲解 image 组件。

- (1) image 组件作用。
- (2) 定义 image 组件的示例代码。

- (3) 列举 image 组件的常用属性。
- (4) 列举 image 组件 mode 属性常用的 mode 合法值。
- (5) 通过示例代码演示 aspectFit 缩放模式和 top 裁剪模式的使用。

#### 知识点 2-rpx 尺寸单位

教师通过 PPT 的方式讲解 rpx 尺寸单位。

- (1) rpx 单位的作用。
- (2) rpx 单位的设计思想。
- (3) 不同屏幕宽度的 rpx 与 px 的换算。

#### 知识点 3-样式导入

教师通过 PPT 结合实际操作的方式讲解样式导入。

- (1) 样式导入的作用。
- (2) 样式导入的语法格式。
- (3) 通过示例代码演示如何创建公共样式文件并进行公共样式文件导入。

#### 知识点 4-【案例 2-1】个人信息-准备工作

教师通过 PPT 结合实际操作的方式讲解开发“个人信息”微信小程序的准备工作。

- (1) 创建项目。
- (2) 复制素材。

#### 知识点 5-【案例 2-1】个人信息-实现“个人信息”微信小程序的页面结构

教师通过 PPT 结合实际操作的方式讲解“个人信息”微信小程序页面结构的实现。

- (1) 编写“个人信息”微信小程序的页面结构。

#### 知识点 6-【案例 2-2】个人信息-实现“个人信息”微信小程序的页面样式

教师通过 PPT 结合实际操作的方式讲解“个人信息”微信小程序页面样式的实现。

- (1) 编写头像区域的页面样式。
- (2) 编写详细信息区域的页面样式。

#### 四、归纳总结

教师回顾本节课所讲的知识，并通过测试题的方式引导学生解答问题并给予指导。

#### 五、布置作业

教师通过学习通或学习通或高校教辅平台（<http://tch.ityxb.com>）布置本节课作业以及下节课的预习作业。

#### 第三课时（上机练习）

上机练习主要针对本章中需要重点掌握的知识点，以及在程序中容易出错的内容进行练习，通过上机练习可以考察同学对知识点的掌握情况，对代码的熟练程度。

上机：（考察知识点为：WXML 简介、WXSS 简介、常用组件、页面路径配置、view 组件、image 组件、rpx 单位、样式导入、【案例 2-1】个人信息-准备工作、【案例 2-1】个人信息-实现“个人信息”微信小程序的页面结构、【案例 2-1】个人信息-实现“个人信息”微信小程序的页面样式）

形式：单独完成

题目：完成【案例 2-1】个人信息的开发

**具体要求:**

- (1) 创建项目和复制素材。
- (2) 实现“个人信息”微信小程序的页面结构。
- (3) 实现“个人信息”微信小程序的页面样式。

**第四课时**

**(swiper 和 swiper-item 组件、text 组件、Flex 布局、**

**【案例 2-2】本地生活-准备工作、【案例 2-2】本地生活-实现“本地生活”微信小程序的页面结构、【案例 2-2】本地生活-实现“本地生活”微信小程序的页面样式)**

**一、复习巩固**

教师通过上节课作业的完成情况，对学生吸收不好的知识点进行再次巩固讲解。

**二、通过直接引入的方式导入新课**

“本地生活”微信小程序是一个介绍本地美食、装修、工作等信息的微信小程序，微信小程序的首页包含轮播图区域和九宫格区域。本节课将详细讲解“本地生活”微信小程序用到的相关知识点及开发。

**三、新课讲解**

**知识点 1-swiper 和 swiper-item 组件**

教师通过 PPT 结合实际操作的方式讲解 swiper 和 swiper-item 组件。

- (1) swiper 和 swiper-item 组件的作用。
- (2) swiper 和 swiper-item 组件的示例代码。
- (3) 列举 swiper 组件的常用属性。
- (4) 通过示例代码演示 swiper 和 swiper-item 组件的使用。

**知识点 2-text 组件**

教师通过 PPT 结合实际操作的方式讲解 text 组件。

- (1) text 组件的作用。
- (2) text 组件的示例代码。
- (3) 列举 text 组件的常用属性。
- (4) 通过示例代码演示 text 组件的使用。

**知识点 3-Flex 布局**

教师通过 PPT 结合实际操作的方式讲解 Flex 布局。

- (1) Flex 布局概念及作用。
- (2) Flex 布局的使用方法。
- (3) 列举 Flex 容器的常用属性。
- (4) 列举 Flex 项目的常用属性。
- (5) 通过示例代码演示 flex-direction、justify-content、align-items 和 flex-wrap 属性的使用。

**知识点 4-【案例 2-2】本地生活-准备工作**

教师通过 PPT 结合实际操作的方式讲解开发“本地生活”微信小程序的准备工作。

- (1) 创建项目。
- (2) 配置页面。
- (3) 复制素材。

**知识点 5-【案例 2-2】本地生活-实现“本地生活”微信小程序的页面结构**  
教师通过 PPT 结合实际操作的方式讲解“本地生活”微信小程序页面结构的实现。

- (1) 编写轮播图区域的页面结构。
- (2) 编写九宫格区域的页面结构。

**知识点 6-【案例 2-2】本地生活-实现“本地生活”微信小程序的页面样式**  
教师通过 PPT 结合实际操作的方式讲解“本地生活”微信小程序页面样式的实现。

- (1) 编写轮播图区域的页面样式。
- (2) 编写九宫格区域的页面样式。

#### 四、归纳总结

教师回顾本节课所讲的内容，并通过测试题的方式引导学生解答问题并给予指导。

#### 五、布置作业

教师通过学习通或高校教辅平台（<http://tch.ityxb.com>）布置本节课作业以及下节课的预习作业。

### 第五课时（上机练习）

上机练习主要针对本章中需要重点掌握的知识点，以及在程序中容易出错的内容进行练习，通过上机练习可以考察同学对知识点的掌握情况，对代码的熟练程度。

上机：（考察知识点为：**swiper** 和 **swiper-item** 组件、**text** 组件、**Flex** 布局、**【案例 2-2】本地生活-准备工作**、**【案例 2-2】本地生活-实现“本地生活”微信小程序的页面结构**、**【案例 2-2】本地生活-实现“本地生活”微信小程序的页面样式**）

形式：单独完成

题目：完成**【案例 2-2】本地生活的开发**

具体要求：

- (1) 创建项目、配置页面和复制素材。
- (2) 实现“本地生活”微信小程序的页面结构。
- (3) 实现“本地生活”微信小程序的页面样式。

### 第六课时

（**导航栏配置**、**标签栏配置**、**vw、vh 单位**、**video 组件**、**表单组件**）

#### 一、复习巩固

教师通过上节课作业的完成情况，对学生吸收不好的知识点进行再次巩固讲解。

#### 二、通过需求引入的方式导入新课

当一对新人即将举办婚礼时，通常会向他们的亲朋好友发送一些邀请函。相比于传统的纸质邀请函，通过微信小程序发送邀请函会让人眼前一亮，给人们不一样的便捷体验。本节课将详细讲解“婚礼邀请函”微信小程序用到的相关知识点。

#### 三、新课讲解

**知识点 1-导航栏配置**

教师通过 PPT 结合实际操作的方式讲解导航栏配置。

- (1) 导航栏配置的作用。
- (2) 列举导航栏的相关配置项。
- (3) 通过示例代码演示如何在页面配置文件和全局配置文件中对导航栏进行配置。

#### **知识点 2-标签栏配置**

教师通过 PPT 结合实际操作的方式讲解标签栏配置。

- (1) 配置标签栏的作用。
- (2) 配置标签栏的方法。
- (3) 列举 tabBar 配置项的属性。
- (4) 通过示例代码演示 tabBar 配置项的属性的使用。
- (5) list 属性的相关概念。
- (6) 标签按钮的相关属性。
- (7) 通过示例代码演示 tabBar 配置项中 list 属性的使用。

#### **知识点 3-vw、vh 单位**

教师通过 PPT 的方式讲解 vw、vh 单位。

- (1) vw、vh 单位作用。
- (2) vw、vh 单位的换算思想。

#### **知识点 4-video 组件**

教师通过 PPT 结合实际操作的方式讲解 video 组件。

- (1) video 组件的作用。
- (2) video 组件的示例代码。
- (3) 列举 video 组件的常用属性。
- (4) 通过示例代码演示 video 组件的使用。

#### **知识点 5-表单组件**

教师通过 PPT 结合实际操作的方式讲解表单组件。

- (1) 表单组件作用。
- (2) 常见的表单组件。
  - form 组件。
  - button 组件。
  - input 组件。
  - checkbox 和 checkbox-group 组件。
  - radio 和 radio-group 组件。
  - radio 组件。

#### **四、归纳总结**

教师回顾本节课所讲的内容，并通过测试题的方式引导学生解答问题并给予指导。

#### **五、布置作业**

教师通过学习通或高校教辅平台（<http://tch.ityxb.com>）布置本节课作业以及下节课的预习作业。

#### **第七、八课时**

（【案例 2-3】婚礼邀请函-准备工作、【案例 2-3】婚礼邀请函-项目初始化、【案例 2-3】婚礼邀请函-实现“邀请函”页面的结构、【案例 2-3】婚礼邀

邀请函-实现“邀请函”页面的样式、【案例 2-3】婚礼邀请函-实现“照片”页面的结构、【案例 2-3】婚礼邀请函-实现“照片”页面的样式、【案例 2-3】婚礼邀请函-实现“美好时光”页面的结构、【案例 2-3】婚礼邀请函-实现“美好时光”页面的样式、【案例 2-3】婚礼邀请函-实现“宾客信息”页面的结构、【案例 2-3】婚礼邀请函-实现“宾客信息”页面的样式)

#### 一、复习巩固

教师通过上节课作业的完成情况，对学生吸收不好的知识点进行再次巩固讲解。

#### 二、通过直接引入的方式导入新课

上节课对“婚礼邀请函”微信小程序中将要用到的知识点进行详细讲解，本节课中将详细讲解“婚礼邀请函”微信小程序的开发。

#### 三、新课讲解

##### 知识点 1-【案例 2-3】婚礼邀请函-准备工作

教师通过 PPT 结合实际操作的方式讲解开发“婚礼邀请函”微信小程序的准备工作。

- (1) 创建项目。
- (2) 配置页面。
- (3) 复制素材。
- (4) 安装 Node.js。
- (5) 启动服务器。

##### 知识点 2-【案例 2-3】婚礼邀请函-项目初始化

教师通过 PPT 结合实际操作的方式讲解“婚礼邀请函”微信小程序的项目初始化。

- (1) 标签栏的配置。
- (2) 各个导航栏标题的配置。
- (3) 全局样式文件中导航栏样式的配置。
- (4) 公共样式的编写。

##### 知识点 3-【案例 2-3】婚礼邀请函-实现“邀请函”页面的结构

教师通过 PPT 结合实际操作的方式讲解“婚礼邀请函”微信小程序“邀请函”页面结构的实现。

- (1) 实现“邀请函”页面的背景图片。
- (2) 编写内容区域的整体结构。
- (3) 编写新郎和新娘的姓名区域的结构。
- (4) 编写婚礼信息区域的结构。

##### 知识点 4-【案例 2-3】婚礼邀请函-实现“邀请函”页面的样式

教师通过 PPT 结合实际操作的方式讲解“婚礼邀请函”微信小程序“邀请函”页面样式的实现。

- (1) 编写背景图片的样式。
- (2) 编写内容区域中外层容器的样式。
- (3) 编写顶部图片区域的样式。
- (4) 编写标题区域的样式。
- (5) 编写合照区域的样式。
- (6) 编写新郎和新娘姓名区域的样式。
- (7) 编写新郎和新娘姓名区域中姓名的样式。

- (8) 编写新郎和新娘姓名区域中电话图片的样式。
- (9) 编写新郎和新娘姓名区域中“喜”字图片的样式。
- (10) 编写婚礼信息区域的样式。

#### **知识点 5-【案例 2-3】婚礼邀请函-实现“照片”页面的结构**

教师通过 PPT 结合实际操作的方式讲解“婚礼邀请函”微信小程序“照片”页面结构的实现。

- (1) 通过代码实现“照片”页面的结构。

#### **知识点 6-【案例 2-3】婚礼邀请函-实现“照片”页面的样式**

教师通过 PPT 结合实际操作的方式讲解“婚礼邀请函”微信小程序“照片”页面样式的实现。

- (1) 通过代码实现“照片”页面的样式。

#### **知识点 7-【案例 2-3】婚礼邀请函-实现“美好时光”页面的结构**

教师通过 PPT 结合实际操作的方式讲解“婚礼邀请函”微信小程序“美好时光”页面结构的实现。

- (1) 通过代码实现“美好时光”页面的结构。

#### **知识点 8-【案例 2-3】婚礼邀请函-实现“美好时光”页面的样式**

教师通过 PPT 结合实际操作的方式讲解“婚礼邀请函”微信小程序“美好时光”页面样式的实现。

- (1) 编写“美好时光”页面中的 3 个外层 view 组件的样式。
- (2) 编写标题和拍摄日期区域的样式。
- (3) 编写视频区域的样式。

#### **知识点 9-【案例 2-3】婚礼邀请函-实现“宾客信息”页面的结构**

教师通过 PPT 结合实际操作的方式讲解“婚礼邀请函”微信小程序“宾客信息”页面结构的实现。

- (1) 实现“宾客信息”页面的背景图片。
- (2) 编写内容区域的整体页面结构。
- (3) 编写姓名区域的结构。
- (4) 编写手机号码区域的结构。
- (5) 编写性别区域的结构。
- (6) 编写需要的点心区域的结构。

#### **知识点 10-【案例 2-3】婚礼邀请函-实现“宾客信息”页面的样式**

教师通过 PPT 结合实际操作的方式讲解“婚礼邀请函”微信小程序“宾客信息”页面样式的实现。

- (1) 编写背景图片的样式。
- (2) 编写内容区域中外层容器的样式。
- (3) 编写姓名和手机号码区域的样式。
- (4) 编写性别区域的样式。
- (5) 编写点心区域的样式。
- (6) 编写“提交”按钮的样式。

#### **四、归纳总结**

教师回顾本节课所讲的内容，并通过测试题的方式引导学生解答问题并给予指导。

#### **五、布置作业**

教师通过学习通或高校教辅平台（<http://tch.ityxb.com>）布置本节课作业以

	<p>及下节课的预习作业。</p> <p>上机练习主要针对上节课中需要重点掌握的知识点，结合“婚礼邀请函”微信小程序的“邀请函”“照片”“美好时光”“宾客信息”页面进行练习，通过上机练习可以考察同学对知识点的掌握情况，对代码的熟练程度。</p> <p><b>上机一：（考察知识点为：【案例 2-3】婚礼邀请函-准备工作、【案例 2-3】婚礼邀请函-项目初始化、【案例 2-3】婚礼邀请函-实现“邀请函”页面的结构、【案例 2-3】婚礼邀请函-实现“邀请函”页面的样式）</b></p> <p>形式：独立完成</p> <p>题目：“婚礼邀请函”微信小程序“邀请函”页面的开发。</p> <p>具体要求：</p> <ol style="list-style-type: none"> <li>(1) 创建项目、配置页面以及复制素材。</li> <li>(2) 项目初始化。</li> <li>(3) 实现“邀请函”页面的结构。</li> <li>(4) 实现“邀请函”页面的样式。</li> </ol> <p><b>上机二：（考察知识点为：【案例 2-3】婚礼邀请函-实现“照片”页面的结构、【案例 2-3】婚礼邀请函-实现“照片”页面的样式、【案例 2-3】婚礼邀请函-实现“美好时光”页面的结构、【案例 2-3】婚礼邀请函-实现“美好时光”页面的样式、【案例 2-3】婚礼邀请函-实现“宾客信息”页面的结构、【案例 2-3】婚礼邀请函-实现“宾客信息”页面的样式）</b></p> <p>形式：独立完成</p> <p>题目：“婚礼邀请函”微信小程序“照片”“美好时光”“宾客信息”页面的开发。</p> <p>具体要求：</p> <ol style="list-style-type: none"> <li>(1) 实现“照片”页面的结构。</li> <li>(2) 实现“照片”页面的样式。</li> <li>(3) 实现“美好时光”页面的结构。</li> <li>(4) 实现“美好时光”页面的样式。</li> <li>(5) 实现“宾客信息”页面的结构。</li> <li>(6) 实现“宾客信息”页面的样式。</li> </ol>
<p>教学后记</p>	<p>本章聚焦微信小程序页面制作，通过多个案例驱动教学，学生基本掌握了 WXML、WXSS 语法及常用组件的使用。在实操环节，“婚礼邀请函”案例因涉及多页面联动与样式复杂，部分学生在 Flex 布局与标签栏配置上存在困难，需在后续练习中增加针对性辅导。</p>

## 第3章 微信小程序页面交互

课题名称	第3章 微信小程序页面交互	计 划课 时	7 课时
教学引入	<p>通过第2章的学习，读者已可以在微信小程序项目中实现页面结构和样式效果，但是这些页面并不能进行交互。在实际的微信小程序项目中，用户是可以与微信小程序页面发生交互的，即可以通过触摸、长按等操作实现各种各样的功能。本章将对微信小程序页面交互的相关知识进行详细讲解。</p>		
教学目标	<ul style="list-style-type: none"> <li>● 使学生熟悉 Page()函数，能够归纳 Page()函数及其各个参数的作用</li> <li>● 使学生掌握数据绑定，能够运用数据绑定实现页面中数据的显示与修改</li> <li>● 使学生掌握事件绑定，能够在组件触发时执行对应的事件处理函数</li> <li>● 使学生熟悉事件对象，能够总结事件对象的属性及其作用</li> <li>● 使学生掌握 this 关键字的使用，能够运用 this 关键字访问当前页面中的数据或者函数</li> <li>● 使学生掌握 setData()方法的使用，能够完成数据的设置与更改</li> <li>● 使学生掌握条件渲染，能够运用条件渲染根据不同的判断结果显示不同的组件</li> <li>● 使学生掌握 &lt;block&gt; 标签，能够运用 &lt;block&gt; 标签同时显示或隐藏多个组件</li> <li>● 使学生熟悉 hidden 属性，能够区分其与 wx:if 控制属性的区别</li> <li>● 使学生掌握 data-* 自定义属性，能够完成 data-* 自定义数据的设置与获取</li> <li>● 使学生掌握模块的使用，能够完成模块的创建和引入</li> <li>● 使学生掌握列表渲染，能够运用列表渲染将数组中的数据渲染到页面中</li> <li>● 使学生掌握网络请求的实现，能够通过网络请求与服务器进行交互</li> <li>● 使学生掌握提示框的实现，能够在页面中显示消息提示框</li> <li>● 使学生掌握 WXS 的使用，能够运用 WXS 处理页面中的数据</li> <li>● 使学生掌握上拉触底的实现，能够运用上拉触底实现数据的动态加载</li> <li>● 使学生掌握下拉刷新的实现，能够运用下拉刷新实现数据的重新加载</li> <li>● 使学生掌握双向数据绑定，能够运用双向绑定实现数据的动态更改</li> </ul>		
教学重点	<ul style="list-style-type: none"> <li>● Page()函数</li> <li>● 事件绑定</li> <li>● setData()方法</li> <li>● 条件渲染</li> <li>● 列表渲染</li> <li>● 网络请求</li> <li>● WXS</li> <li>● 双向数据绑定</li> </ul>		
教学难点	<ul style="list-style-type: none"> <li>● 【案例 3-2】计算器-特殊情况处理</li> </ul>		

教学方式	课堂教学以 PPT 讲授为主，并结合多媒体进行教学
思政元素	<p>逻辑思维与问题解决能力：在“比较数字大小”“计算器”案例开发中，引导学生分析功能需求、梳理逻辑流程（如计算器特殊情况处理、数据绑定与事件触发逻辑），培养学生运用逻辑思维解决实际问题的能力，同时强调面对复杂问题时的耐心与细心，树立严谨的科学态度。</p> <p>诚信与数据安全意识：讲解网络请求与数据绑定功能时，介绍数据传输过程中的安全规范（如 HTTPS 协议使用、服务器域名信任配置），引导学生认识到数据安全性的重要性，培养学生在开发中保护用户数据、坚守诚信开发的职业操守。</p>
教学过程	<p style="text-align: center;"><b>第一课时</b></p> <p style="text-align: center;"><b>（Page()函数、数据绑定、事件绑定、事件对象、this 关键字）</b></p> <p>一、复习巩固</p> <p>教师通过上节课作业的完成情况，对学生吸收不好的知识点进行再次巩固讲解。</p> <p>二、通过直接引入的方式导入新课</p> <p>本案例将实现“比较数字大小”微信小程序，它的功能是当用户输入两个数字后，点击“比较”按钮可以自动比较这两个数字的大小。本节课将详细讲解“比较数字大小”微信小程序用到的相关知识点。</p> <p>三、新课讲解</p> <p><b>知识点 1-Page()函数</b></p> <p>教师通过 PPT 的方式讲解 Page()函数。</p> <ol style="list-style-type: none"> <li>(1) 什么是 Page()函数。</li> <li>(2) 通过示例代码演示 Page()函数的使用。</li> <li>(3) Page()函数的参数。 <ul style="list-style-type: none"> <li>● 页面的初始数据。</li> <li>● 页面生命周期回调函数</li> <li>● 页面事件处理函数</li> </ul> </li> </ol> <p><b>知识点 2-数据绑定</b></p> <p>教师通过 PPT 结合实际操作的方式讲解数据绑定。</p> <ol style="list-style-type: none"> <li>(1) 什么是数据绑定。</li> <li>(2) 通过示例代码演示数据绑定。</li> </ol> <p><b>知识点 3-事件绑定</b></p> <p>教师通过 PPT 结合实际操作的方式讲解事件绑定。</p> <ol style="list-style-type: none"> <li>(1) 什么是事件。</li> <li>(2) 常见的事件。</li> <li>(3) 事件的分类。</li> <li>(4) 为组件绑定事件的方法。</li> <li>(5) 通过示例代码演示如何为组件绑定事件。</li> <li>(6) 事件处理函数的简写形式。</li> </ol> <p><b>知识点 4-事件对象</b></p> <p>教师通过 PPT 结合实际操作的方式讲解事件对象。</p> <ol style="list-style-type: none"> <li>(1) 事件对象的作用。</li> <li>(2) 事件对象的属性。</li> </ol>

- (3) 通过示例代码演示事件对象的使用。
- (4) 事件对象属性中的 `target` 和 `currentTarget` 属性的区别。
- (5) 通过示例代码演示事件对象属性中的 `target` 和 `currentTarget` 属性的区别。

#### 知识点 5-this 关键字

教师通过 PPT 结合实际操作的方式讲解 `this` 关键字。

- (1) `this` 关键字的作用。
- (2) 通过示例代码演示 `this` 关键字的使用。

#### 四、归纳总结

教师回顾本节课所讲的知识,并通过测试题的方式引导学生解答问题并给予指导。

#### 五、布置作业

教师通过学习通或高校教辅平台 (<http://tch.ityxb.com>) 布置本节课作业以及下节课的预习作业。

### 第二课时

(`setData()`方法、条件渲染、`<block>`标签、`hidden` 属性、【案例 3-1】比较数字大小-准备工作、【案例 3-1】比较数字大小-实现“比较数字大小”微信小程序的页面结构、【案例 3-1】比较数字大小-获取并保存用户输入的数字、【案例 3-1】比较数字大小-判断数字大小并显示结果)

#### 一、复习巩固

教师通过上节课作业的完成情况,对学生吸收不好的知识点进行再次巩固讲解。

#### 二、通过直接引入的方式导入新课

在上节课学习了“比较数字大小”微信小程序用到的部分知识点,包括 `Page()`函数、数据绑定、事件绑定、事件对象、`this` 关键字。本节课中将详细讲解“比较数字大小”微信小程序的相关知识点和开发。

#### 三、新课讲解

##### 知识点 1-`setData()`方法

教师通过 PPT 的方式讲解 `setData()`方法。

- (1) `setData()`方法的作用。
- (2) `setData()`方法的基本语法。
- (3) `setData()`方法的参数。
  - `data`。
  - `callback`。
- (4) 通过示例代码演示 `setData()`方法的使用。

##### 知识点 2-条件渲染

教师通过 PPT 结合实际操作的方式讲解条件渲染。

- (1) 条件渲染的作用。
- (2) 实现条件渲染的 3 个控制属性。
  - `wx:if`。
  - `wx:elif`。
  - `wx:else`。

##### 知识点 3-`<block>`标签

教师通过 PPT 结合实际操作的方式讲解 `<block>`标签。

- (1) <block>标签的作用。
- (2) 通过示例代码演示<block>标签的使用。

#### 知识点 4-hidden 属性

教师通过 PPT 结合实际操作的方式讲解 hidden 属性以及 hidden 属性与 wx:if 控制属性的区别。

- (1) hidden 属性的作用。
- (2) hidden 属性的示例代码。
- (3) hidden 属性与 wx:if 控制属性的区别。

#### 知识点 5-【案例 3-1】比较数字大小-准备工作

教师通过 PPT 结合实际操作的方式讲解开发“比较数字大小”微信小程序的准备工作。

- (1) 创建项目。
- (2) 配置导航栏。
- (3) 复制素材。

#### 知识点 6-【案例 3-1】比较数字大小-实现“比较数字大小”微信小程序的页面结构

教师通过 PPT 结合实际操作的方式讲解“比较数字大小”微信小程序页面结构的实现。

- (1) 通过代码实现“比较数字大小”微信小程序的页面结构。
- (2) 运行程序，查看页面结构。

#### 知识点 7-【案例 3-1】比较数字大小-获取并保存用户输入的数字

教师通过 PPT 结合实际操作的方式讲解如何获取并保存用户输入的数字。

- (1) 通过代码实现对用户输入的数字的获取以及保存。
- (2) 运行程序，查看运行结果。

#### 知识点 8-【案例 3-1】比较数字大小-判断数字大小并显示结果

教师通过 PPT 结合实际操作的方式讲解如何判断数字大小并显示结果。

- (1) 通过代码实现数字大小的判断以及结果的显示。
- (2) 运行程序，查看运行结果。

#### 四、归纳总结

教师回顾本节课所讲的知识，并通过测试题的方式引导学生解答问题并予以指导。

#### 五、布置作业

教师通过学习通或高校教辅平台 (<http://tch.ityxb.com>) 布置本节课作业以及下节课的预习作业。

#### 第三课时（上机练习）

上机练习主要针对本章中需要重点掌握的知识点，以及在程序中容易出错的内容进行练习，通过上机练习可以考察同学对知识点的掌握情况，对代码的熟练程度。

上机：（考察知识点为：Page()函数、数据绑定、事件绑定、事件对象、this 关键字、setData()方法、条件渲染、【案例 3-1】比较数字大小-准备工作、【案例 3-1】比较数字大小-实现“比较数字大小”微信小程序的页面结构、【案例 3-1】比较数字大小-获取并保存用户输入的数字、【案例 3-1】比较数字大小-判断数字大小并显示结果）

形式：独立完成

题目：完成【案例 3-1】比较数字大小的开发。

具体要求：

- (1) 创建项目、配置导航栏和复制样式素材。
- (2) 实现“比较数字大小”微信小程序的页面结构。
- (3) 获取并保存用户输入的数字。
- (4) 判断数字大小并显示结果。

#### 第四课时

(data-\*自定义属性、模块、【案例 3-2】计算器-准备工作、  
【案例 3-2】计算器-实现“计算器”微信小程序的页面结构、【案例 3-2】计算器-实现“计算器”微信小程序的页面逻辑、【案例 3-2】计算器-特殊情况处理)

##### 一、复习巩固

教师通过上节课作业的完成情况，对学生吸收不好的知识点进行再次巩固讲解。

##### 二、通过情景引入的方式导入新课

在日常生活中，计算器是人们广泛使用的工具，可以帮助我们快速且方便地计算金额、成本、利润等。本节课将详细讲解“计算器”微信小程序的相关知识点和开发。

##### 三、新课讲解

###### 知识点 1- data-\*自定义属性

教师通过 PPT 结合实际操作的方式讲解 data-\*自定义属性。

- (1) 什么是 data-\*自定义属性。
- (2) data-\*自定义属性在事件处理函数中的获取方法。
  - target。
  - currentTarget 对象的 dataset 属性。
- (3) 通过示例代码演示 data-\*自定义属性如何实现参数的传递。

###### 知识点 2-模块

教师通过 PPT 的方式讲解模块。

- (1) 什么是模块。
- (2) 模块化开发语法。
  - 使用 module.exports 语法对外暴露接口。
  - 通过 require()语法引入模块。
- (3) 通过示例代码演示如何创建和引用模块。

###### 知识点 3-【案例 3-2】计算器-准备工作

教师通过 PPT 结合实际操作的方式讲解开发“计算器”微信小程序的准备工作。

- (1) 创建项目。
- (2) 配置导航栏。
- (3) 复制素材。

###### 知识点 4-【案例 3-2】计算器-实现“计算器”微信小程序的页面结构

教师通过 PPT 结合实际操作的方式讲解“比较数字大小”微信小程序页面结构的实现。

- (1) 编写页面整体结构。
- (2) 编写结果区域的结构。
- (3) 编写按钮区域第 1 行按钮的结构。
- (4) 编写按钮区域第 2 行按钮的结构。
- (5) 编写按钮区域第 3 行按钮的结构。
- (6) 编写按钮区域第 4 行按钮的结构。
- (7) 编写按钮区域第 5 行按钮的结构。

#### 知识点 5-【案例 3-2】计算器-实现“计算器”微信小程序的页面逻辑

教师通过 PPT 结合实际操作的方式讲解“比较数字大小”微信小程序页面逻辑的实现。

- (1) 引入 calc.js 文件，获得计算器对象。
- (2) 编写页面中所需的数据。
- (3) 编写数字按钮的事件处理函数 numBtn()。
- (4) 编写运算符按钮的事件处理函数 opBtn()。
- (5) 解决在点击数字按钮之后点击运算符按钮，再次输入的数字会拼接到第 1 个数字后面，而不是输入第 2 个数字的问题。
- (6) 编写“=”按钮的事件处理函数 execBtn()。
- (7) 编写“C”按钮（重置按钮）的事件处理函数 resetBtn()。
- (8) 编写“.”按钮（小数点按钮）的事件处理函数 dotBtn()。
- (9) 编写“DEL”按钮（删除按钮）的事件处理函数 delBtn()。
- (10) 编写“+/-”按钮（正负切换按钮）的事件处理函数 negBtn()。

#### 知识点 6-【案例 3-2】计算器-特殊情况处理

教师通过 PPT 结合实际操作的方式讲解特殊情况的处理。

- (1) 解决不能进行连等计算的问题。
- (2) 解决数字拼接的问题。
- (3) 解决没有输入第 2 个数字时不能按“=”按钮执行计算的问题。
- (4) 解决无法通过运算符按钮实现连续计算的问题。

#### 四、归纳总结

教师回顾本节课所讲的知识，并通过测试题的方式引导学生解答问题并给予指导。

#### 五、布置作业

教师通过学习通或高校教辅平台（<http://tch.ityxb.com>）布置本节课作业以及下节课的预习作业。

### 第五课时

#### （列表渲染、网络请求、提示框、WXS、上拉触底、下拉刷新）

#### 一、复习巩固

教师通过上节课作业的完成情况，对学生吸收不好的知识点进行再次巩固讲解。

#### 二、通过场景引入的方式导入新课

在日常生活中，人们线上进行线上点餐时，会注意到商家通过列表的形式来展示美食名称、美食图片以及美食商家的电话、地址和营业时间等信息，方便人们查找多种美食信息。本节课将详细讲解“美食列表”微信小程序用到的

相关知识点。

### 三、新课讲解

#### 知识点 1-列表渲染

教师通过 PPT 结合实际操作的方式讲解列表渲染。

- (1) 什么是列表渲染。
- (2) 列表渲染的控制属性的使用方法。
  - wx:for。
  - wx:key。
- (3) 通过示例代码演示列表渲染的使用。
  - 数组的列表渲染。
  - 数组中包含对象情况下的列表渲染。

#### 知识点 2-网络请求

教师通过 PPT 的方式讲解网络请求。

- (1) 什么是网络请求。
- (2) 微信小程序官方对服务器接口的请求的限制。
  - 只能请求 HTTPS 协议的服务器接口。
  - 必须登录微信小程序管理后台，将服务器接口的域名添加到信任列表中。
- (3) 发起网络请求的实现方法。
- (4) wx.request()方法的常见选项。
- (5) 取消请求任务的方法。
- (6) 通过示例代码演示如何调用 wx.request()方法发起一个 GET 方式的请求。

#### 知识点 3-提示框

教师通过 PPT 结合实际操作的方式讲解提示框。

- (1) 实现提示框的方法。
  - wx.showLoading()方法。
  - wx.showToast()方法。
- (2) wx.showLoading()方法的作用。
- (3) 列举 wx.showLoading()方法的常用选项。
- (4) 通过示例代码演示 wx.showLoading()方法的使用。
- (5) wx.showToast()方法的作用。
- (6) 列举 wx.showToast()方法的常用选项。
- (7) 通过示例代码演示 wx.showToast()方法的使用。

#### 知识点 4- WXS

教师通过 PPT 结合实际操作的方式讲解 WXS。

- (1) 什么是 WXS。
- (2) WXS 的应用场景。
- (3) WXS 的特点。
  - WXS 与 JavaScript 不同。
  - WXS 不能作为组件的事件回调
  - 具有隔离性。
  - 在 iOS 设备上效率高。
- (4) WXS 脚本的分类。

- 内嵌 WXS 脚本。
- 外联 WXS 脚本。

(5) 内嵌 WXS 脚本的使用方法。

(6) 外联 WXS 脚本的使用方法。

(7) 通过示例代码演示如何使用内嵌 WXS 脚本和外联 WXS 脚本。

#### 知识点 5-上拉触底

教师通过 PPT 结合实际操作的方式讲解上拉触底。

(1) 什么是上拉触底。

(2) 实现上拉触底的方法。

(3) onReachBottom()事件处理函数的示例代码。

(4) 修改上拉触底距离的方法。

(5) 通过示例代码演示如何配置上拉触底的距离。

#### 知识点 6-下拉刷新

教师通过 PPT 结合实际操作的方式讲解下拉刷新。

(1) 什么是下拉刷新。

(2) 启用下拉刷新有 2 种方式。

- 全局开启下拉刷新。

- 局部开启下拉刷新。

(3) 开始下拉刷新的方法。

(4) 停止下拉刷新的方法。

#### 四、归纳总结

教师回顾本节课所讲的知识,并通过测试题的方式引导学生解答问题并予以指导。

#### 五、布置作业

教师通过学习通或高校教辅平台 (<http://tch.ityxb.com>) 布置本节课作业以及下节课的预习作业。

#### 第六~七课时

(【案例 3-3】美食列表-准备工作、【案例 3-3】美食列表-获取初始数据、【案例 3-3】美食列表-实现页面渲染、【案例 3-3】美食列表-处理电话格式、【案例 3-3】美食列表-实现上拉触底、【案例 3-3】美食列表-实现下拉刷新)

#### 一、复习巩固

教师通过上节课作业的完成情况,对学生吸收不好的知识点进行再次巩固讲解。

#### 二、通过直接引入的方式导入新课

在上节课中学习了“美食列表”微信小程序用到的相关知识点,包括列表渲染、网络请求、提示框、WXS、上拉触底、下拉刷新相关内容,本节课中将详细讲解“美食列表”微信小程序的开发。

#### 三、新课讲解

##### 知识点 1-【案例 3-3】美食列表-准备工作

教师通过 PPT 结合实际操作的方式讲解开发“美食列表”微信小程序的准备工作。

(1) 创建项目。

(2) 配置页面。

- (3) 配置导航栏。
- (4) 复制样式素材。
- (5) 启动服务器

#### **知识点 2-【案例 3-3】美食列表-获取初始数据**

教师通过 PPT 结合实际操作的方式讲解获取初始数据的实现。

- (1) 通过代码定义页面所需的数据。
- (2) 运行程序，测试是否成功获取初始数据。

#### **知识点 3-【案例 3-3】美食列表-实现页面渲染**

教师通过 PPT 结合实际操作的方式讲解页面渲染的实现。

- (1) 通过代码实现“美食列表”微信小程序的页面渲染。
- (2) 运行程序，测试是否实现页面渲染。

#### **知识点 4-【案例 3-3】美食列表-处理电话格式**

教师通过 PPT 结合实际操作的方式讲解处理电话格式的实现。

- (1) 通过代码实现将电话号码进行处理之后显示在页面上。
- (2) 运行程序，测试是否成功处理电话格式。

#### **知识点 5-【案例 3-3】美食列表-实现上拉触底**

教师通过 PPT 结合实际操作的方式讲解上拉触底的实现。

- (1) 通过代码实现是用户在进行上拉操作时，当页面即将到达底部时加载下一页数据。
- (2) 运行程序，测试是否实现该功能。

#### **知识点 6-【案例 3-3】美食列表-实现下拉刷新**

教师通过 PPT 结合实际操作的方式讲解下拉刷新的实现。

- (1) 通过代码实现当用户进行下拉操作时，如果到达了顶部，再进行下拉，可以刷新页面。
- (2) 运行程序，测试是否实现该功能。

#### 四、归纳总结

教师回顾本节课所讲的知识，并通过测试题的方式引导学生解答问题并给予指导。

#### 五、布置作业

教师通过学习通或高校教辅平台（<http://tch.ityx.com>）布置本节课作业以及下节课的预习作业。

#### **（双向数据绑定、【案例 3-4】调查问卷-准备工作、【案例 3-4】调查问卷-获取初始数据、【案例 3-4】调查问卷-实现页面渲染）**

##### 一、复习巩固

教师通过上节课作业的完成情况，对学生吸收不好的知识点进行再次巩固讲解。

##### 二、通过情景引入的方式导入新课

假设有一位大学老师，想通过调查问卷来了解同学们的专业技能、对开设公开课的意见等信息，从而根据同学们的建议制订下一步的教学计划。在本节课中将详细讲解“调查问卷”微信小程序的开发。

##### 三、新课讲解

#### **知识点 1-双向数据绑定**

教师通过 PPT 的方式讲解双向数据绑定。

	<p>(1) 什么是单向数据绑定。  (2) 什么是双向数据绑定。  (3) 双向数据绑定的示例代码。</p> <p><b>知识点 2-【案例 3-4】调查问卷-准备工作</b>  教师通过 PPT 结合实际操作的方式讲解开发“调查问卷”微信小程序的准备工作。</p> <p>(1) 创建项目。  (2) 配置页面。  (3) 配置导航栏。  (4) 复制样式素材。  (5) 启动服务器。</p> <p><b>知识点 3-【案例 3-4】调查问卷-获取初始数据</b>  教师通过 PPT 结合实际操作的方式讲解获取初始数据的实现。</p> <p>(1) 通过代码定义页面所需的数据。  (2) 运行程序，测试是否成功获取初始数据。</p> <p><b>知识点 4-【案例 3-4】调查问卷-实现页面渲染</b>  教师通过 PPT 结合实际操作的方式讲解页面渲染的实现。</p> <p>(1) 通过代码实现“调查问卷”微信小程序的页面渲染。  (2) 运行程序，测试是否实现页面渲染。</p> <p>上机练习主要针对本章中需要重点掌握的知识点，以及在程序中容易出错的内容进行练习，通过上机练习可以考察同学对知识点的掌握情况，对代码的熟练程度。</p> <p>上机：（考察知识点为：双向数据绑定、【案例 3-4】调查问卷-准备工作、【案例 3-4】调查问卷-获取初始数据、【案例 3-4】调查问卷-实现页面渲染）  形式：单独完成  题目：完成【案例 3-4】调查问卷的开发。  具体要求：  (1) 创建项目、配置页面、配置导航栏、复制样式素材和启动服务器。  (2) 获取初始数据。  (3) 实现页面渲染。</p>
教学后记	<p>网络请求部分，因涉及服务器配置，个别学生出现请求失败问题，后续需简化服务器环境配置流程，降低实操门槛。</p>

## 第 4 章 微信小程序常用 API（上）

课题名称	第 4 章 微信小程序常用 API（上）	计划课时	6 课时
教学引入	<p>微信小程序为开发者提供了大量的 API。开发者通过 API 可以获得微信底层封装的高级特性，同时可以很方便地调用微信提供的各种能力，例如网络请求、获取用户信息、本地存储等能力。本章将讲解部分微信小程序中常用的部分 API。</p>		
教学目标	<ul style="list-style-type: none"> <li>● 使学生掌握 scroll-view 组件，能够运用 scroll-view 组件完成视图区域的横向滚动或者纵向滚动</li> <li>● 使学生掌握 slider 组件，能够运用 slider 组件完成滑动选择器的制作</li> <li>● 使学生掌握&lt;include&gt;标签，能够运用&lt;include&gt;标签引用其他文件中的代码</li> <li>● 使学生掌握背景音频 API，能够运用背景音频 API 实现音频后台播放、音频暂停等功能</li> <li>● 使学生掌握录音 API，能够运用录音 API 实现录音功能</li> <li>● 使学生掌握音频 API，能够运用音频 API 实现音频播放、暂停等功能</li> <li>● 使学生掌握选择媒体 API，能够运用 wx.chooseMedia()方法选择图片或视频</li> <li>● 使学生掌握图片预览 API，能够运用 wx.previewImage()方法预览图片</li> <li>● 使学生掌握文件上传 API，能够运用 wx.uploadFile()方法实现将本地资源上传到服务器中</li> <li>● 使学生掌握文件下载 API，能够运用 wx.downloadFile()方法实现资源文件的下载</li> <li>● 使学生掌握 canvas 组件，能够灵活运用 canvas 组件创建画布</li> <li>● 使学生掌握画布 API，能够运用画布 API 完成图形的绘制</li> </ul>		
教学重点	<ul style="list-style-type: none"> <li>● scroll-view 组件</li> <li>● slider 组件</li> <li>● 背景音频 API</li> <li>● 文件上传 API</li> <li>● 文件下载 API</li> <li>● 画布 API</li> </ul>		
教学难点	<ul style="list-style-type: none"> <li>● 【案例 4-2】录音机-实现播放录音功能</li> <li>● 【案例 4-4】模拟时钟-实现时钟走动效果</li> </ul>		
教学方式	<p>课堂教学以 PPT 讲授为主，并结合多媒体进行教学</p>		
思政元素	<p>责任担当与合规使用意识：讲解背景音频、录音、文件上传下载 API 时，强调 API 使用的合规性（如后台音频播放功能申请、文件上传的版权规范），引导学生认识到技术工具的合理使用边界，培养学生遵守平台规则、承担社会责任的职业素养。</p> <p>精益求精与细节把控：在“模拟时钟”案例开发中，讲解 canvas 组件绘制表盘与指针时，强调代码的精准性（如时钟走动效果的定时器控制、指针角度计算），引导学生关注技术细节，树立精益求精的工匠精神，培养学生对技术产品质量的责任感。</p> <p>环保与资源优化意识：介绍文件上传下载 API 时，讲解资源压缩与合理存储（如图片压缩上传、避免冗余文件存储），引导学生在开发中注重资源优化，减少</p>		

	服务器负担，将环保理念融入技术开发，培养学生的绿色开发意识。
教学过程	<p style="text-align: center;"><b>第一课时</b></p> <p style="text-align: center;">(scroll-view 组件、slider 组件、&lt;include&gt;标签、背景音频 API、【案例 4-1】音乐播放器-准备工作、【案例 4-1】音乐播放器-实现“音乐播放器”微信小程序的页面结构、【案例 4-1】音乐播放器-实现标签页切换)</p> <p>一、复习巩固</p> <p>教师通过上节课作业的完成情况，对学生吸收不好的知识点进行再次巩固讲解。</p> <p>二、通过情景引入的方式导入新课</p> <p>“音乐播放器”微信小程序可以让用户随时随地享受音乐，给用户带来了便捷的音乐体验，且支持后台播放，用户可以在听音乐的同时进行其他的操作。本节课将详细讲解“音乐播放器”微信小程序用到的相关知识点和开发。</p> <p>三、新课讲解</p> <p><b>知识点 1-scroll-view 组件</b></p> <p>教师通过 PPT 的方式讲解 scroll-view 组件。</p> <ol style="list-style-type: none"> <li>(1) scroll-view 组件的作用。</li> <li>(2) scroll-view 组件的语法格式。</li> <li>(3) 列举 scroll-view 组件的常用属性。</li> <li>(4) 通过示例代码演示 scroll-view 组件的使用。</li> <li>(5) e.detail 中所携带的数据。</li> </ol> <p><b>知识点 2-slider 组件</b></p> <p>教师通过 PPT 的方式讲解 slider 组件。</p> <ol style="list-style-type: none"> <li>(1) slider 组件的作用。</li> <li>(2) slider 组件的语法格式。</li> <li>(3) 列举 slider 组件的常用属性。</li> <li>(4) 通过示例代码演示 slider 组件的使用。</li> </ol> <p><b>知识点 3-&lt;include&gt;标签</b></p> <p>教师通过 PPT 的方式讲解&lt;include&gt;标签。</p> <ol style="list-style-type: none"> <li>(1) &lt;include&gt;标签的作用。</li> <li>(2) &lt;include&gt;标签的用途。</li> <li>(3) 通过示例代码演示&lt;include&gt;标签的使用。</li> </ol> <p><b>知识点 4-背景音频 API</b></p> <p>教师通过 PPT 结合实际操作的方式讲解背景音频 API。</p> <ol style="list-style-type: none"> <li>(1) 使用背景音频 API 前开启微信小程序后台音频播放功能。</li> <li>(2) 背景音频 API 的使用方法。</li> <li>(3) 获取 BackgroundAudioManager 实例的方法。</li> <li>(4) 列举 BackgroundAudioManager 实例常用的属性和方法。</li> <li>(5) 通过示例代码演示背景音频 API 的使用。</li> </ol> <p><b>知识点 5-【案例 4-1】音乐播放器-准备工作</b></p> <p>教师通过 PPT 结合实际操作的方式讲解开发“音乐播放器”微信小程序的准备工作。</p> <ol style="list-style-type: none"> <li>(1) 创建项目。</li> <li>(2) 配置页面</li> </ol>

- (3) 配置导航栏。
- (4) 复制素材。
- (5) 启动服务器

#### 知识点 6-【案例 4-1】音乐播放器-实现“音乐播放器”微信小程序的页面结构

教师通过 PPT 结合实际操作的方式讲解“音乐播放器”微信小程序页面结构的实现。

- (1) 通过代码实现“音乐播放器”微信小程序的页面结构。
- (2) 运行程序，查看“音乐播放器”微信小程序的页面结构。

#### 知识点 7-【案例 4-1】音乐播放器-实现标签页切换

教师通过 PPT 结合实际操作的方式讲解标签页切换功能的实现。

- (1) 通过代码实现通过点击标签栏区域的标签按钮切换到对应的标签页。
- (2) 运行程序，查看运行结果。

#### 四、归纳总结

教师回顾本节课所讲的知识，并通过测试题的方式引导学生解答问题并给予指导。

#### 五、布置作业

教师通过学习通或高校教辅平台（<http://tch.ityxb.com>）布置本节课作业以及下节课的预习作业。

### 第二课时

（【案例 4-1】音乐播放器-实现“音乐推荐”标签页、【案例 4-1】音乐播放器-实现“播放器”标签页、【案例 4-1】音乐播放器-实现播放器区域、【案例 4-1】音乐播放器-实现播放进度的控制、【案例 4-1】音乐播放器-实现“播放列表”标签页）

#### 一、复习巩固

教师通过上节课作业的完成情况，对学生吸收不好的知识点进行再次巩固讲解。

#### 二、通过直接引入的方式导入新课

在上节课中学习了“音乐播放器”微信小程序用到的相关知识点以及小程序的部分开发，包括准备工作、实现“音乐播放器”微信小程序的页面结构、实现标签页切换。本节课将继续详细讲解“音乐播放器”微信小程序的开发。

#### 三、新课讲解

##### 知识点 1-【案例 4-1】音乐播放器-实现“音乐推荐”标签页

教师通过 PPT 结合实际操作的方式讲解“音乐推荐”标签页的实现。

- (1) “音乐推荐”标签页的实现思路。
- (2) “音乐推荐”标签页的实现步骤。
- (3) 运行程序，查看运行结果。

##### 知识点 2-【案例 4-1】音乐播放器-实现“播放器”标签页

教师通过 PPT 结合实际操作的方式讲解“播放器”标签页的实现。

- (1) “播放器”标签页的实现思路。
- (2) “播放器”标签页的实现步骤。
- (3) 运行程序，查看运行结果。

##### 知识点 3-【案例 4-1】音乐播放器-实现播放器区域

教师通过 PPT 结合实际操作的方式讲解播放器区域的实现。

- (1) 通过代码实现播放器区域。
- (2) 运行程序，查看运行结果。

#### 知识点 4-【案例 4-1】音乐播放器-实现播放进度的控制

教师通过 PPT 结合实际操作的方式讲解播放器进度的控制的实现。

- (1) 通过代码实现播放进度的控制。
- (2) 运行程序，查看运行结果。

#### 知识点 5-【案例 4-1】音乐播放器-实现“播放列表”标签页

教师通过 PPT 结合实际操作的方式讲解“播放列表”标签页的实现。

- (1) “播放列表”标签页的实现思路。
- (2) “播放列表”标签页的实现步骤。
- (3) 运行程序，查看运行结果。

#### 四、归纳总结

教师回顾本节课所讲的知识，并通过测试题的方式引导学生解答问题并给予指导。

#### 五、布置作业

教师通过学习通或高校教辅平台（<http://tch.ityx.com>）布置本节课作业以及下节课的预习作业。

### 第三课时

（录音 API、音频 API、【案例 4-2】录音机-准备工作、【案例 4-2】录音机-初始化录音功能、【案例 4-2】录音机-实现“录音机”微信小程序的页面结构、【案例 4-2】录音机-实现录音功能、【案例 4-2】录音机-实现播放录音功能）

#### 一、复习巩固

教师通过上节课作业的完成情况，对学生吸收不好的知识点进行再次巩固讲解。

#### 二、通过需求引入的方式导入新课

录音机是生活中的常用工具，它可以在开会的时候记录说话人的声音，也可以在生活中留下一段美妙歌声。录音机可以记录声音和播放声音，因此成为新闻工作者工作的重要器材。本节课将详细讲解“录音机”微信小程序用到的相关知识点和开发。

#### 三、新课讲解

##### 知识点 1-录音 API

教师通过 PPT 的方式讲解录音 API。

- (1) 录音 API 的使用方法。
- (2) 获取 RecorderManager 实例的方法。
- (3) 列举 RecorderManager 实例的常用方法。
- (4) 通过示例代码演示录音 API 的使用。

##### 知识点 2-音频 API

教师通过 PPT 的方式讲解音频 API。

- (1) 音频 API 与背景音频 API 的区别。
- (2) 背景音频 API 的使用方法。

- (3) 获取 InnerAudioContext 实例的方法。
- (4) 列举 InnerAudioContext 实例特有的属性和方法。
- (5) 通过案例演示音频 API 的使用。

### 知识点 3-【案例 4-2】录音机-准备工作

教师通过 PPT 结合实际操作的方式讲解开发“录音机”微信小程序的准备工作。

- (1) 创建项目。
- (2) 配置导航栏。
- (3) 复制素材。

### 知识点 4-【案例 4-2】录音机-初始化录音功能

教师通过 PPT 结合实际操作的方式讲解初始化录音功能的实现。

- (1) 通过代码实现初始化录音功能。

### 知识点 5-【案例 4-2】录音机-实现“录音机”微信小程序的页面结构

教师通过 PPT 结合实际操作的方式讲解“录音机”微信小程序页面结构的实现。

- (1) 通过代码实现“录音机”微信小程序的页面结构。
- (2) 运行程序，查看运行结果。

### 知识点 6-【案例 4-2】录音机-实现录音功能

教师通过 PPT 结合实际操作的方式讲解录音功能的实现。

- (1) 通过代码实现录音功能。
- (2) 运行程序，查看运行结果。

### 知识点 7-【案例 4-2】录音机-实现播放录音功能

教师通过 PPT 结合实际操作的方式讲解播放录音功能的实现。

- (1) 通过代码实现播放录音功能。
- (2) 运行程序，查看运行结果。

## 四、归纳总结

教师回顾本节课所讲的知识，并通过测试题的方式引导学生解答问题并给予指导。

## 五、布置作业

教师通过学习通或高校教辅平台（<http://tch.ityxb.com>）布置本节课作业以及下节课的预习作业。

## 第四课时

（选择媒体 API、图片预览 API、文件上传 API、文件下载 API、【案例 4-3】头像上传下载-准备工作、【案例 4-3】头像上传下载-实现“头像上传下载”微信小程序的页面结构、【案例 4-3】头像上传下载-实现“头像上传下载”微信小程序的页面逻辑）

### 一、复习巩固

教师通过上节课作业的完成情况，对学生吸收不好的知识点进行再次巩固讲解。

### 二、通过情景引入的方式导入新课

头像上传下载是微信小程序开发中常见的一个功能，一般会出现在用户中心模块中，用于设置用户的头像。本节将详细讲解“头像上传下载”微信小程

序用到的相关知识点和开发。

### 三、 新课讲解

#### 知识点 1-选择媒体 API

教师通过 PPT 的方式讲解选择媒体 API。

- (1) 选择媒体 API 的作用。
- (2) 选择媒体 API 的实现方法。
- (3) 列举 wx.chooseMedia()方法的常用选项。
- (4) 通过示例代码演示选择媒体 API 的使用。

#### 知识点 2-图片预览 API

教师通过 PPT 的方式讲解图片预览 API。

- (1) 图片预览 API 的作用。
- (2) 图片预览 API 的实现方法。
- (3) 列举 wx.previewImage()方法的常用选项。
- (4) 通过示例代码演示图片预览 API 的使用。

#### 知识点 3-文件上传 API

教师通过 PPT 的方式讲解文件上传 API。

- (1) 文件上传 API 的作用。
- (2) 文件上传 API 的实现方法。
- (3) 列举 wx.uploadFile()方法的常用选项。
- (4) 通过示例代码演示文件上传 API 的使用。

#### 知识点 4-文件下载 API

教师通过 PPT 的方式讲解文件下载 API。

- (1) 文件下载 API 的作用。
- (2) 文件下载 API 的实现方法。
- (3) 列举 wx.downloadFile ()方法的常用选项。
- (4) 通过示例代码演示文件下载 API 的使用。

#### 知识点 5-【案例 4-3】头像上传下载-准备工作

教师通过 PPT 结合实际操作的方式讲解开发“头像上传下载”微信小程序的准备工作。

- (1) 创建项目。
- (2) 配置导航栏。
- (3) 复制素材。
- (4) 启动服务器。

#### 知识点 6-【案例 4-3】头像上传下载-实现“头像上传下载”微信小程序的页面结构

教师通过 PPT 结合实际操作的方式讲解“头像上传下载”页面结构的实现。

- (1) 通过代码实现“头像上传下载”微信小程序的页面结构。
- (2) 运行程序，查看运行结果。

#### 知识点 7-【案例 4-3】头像上传下载-实现“头像上传下载”微信小程序的页面逻辑

教师通过 PPT 结合实际操作的方式讲解“头像上传下载”微信小程序页面逻辑的实现。

- (1) 通过代码实现“头像上传下载”微信小程序的页面逻辑。
- (2) 运行程序，查看运行结果。

#### 四、归纳总结

教师回顾本节课所讲的知识,并通过测试题的方式引导学生解答问题并予以指导。

#### 五、布置作业

教师通过学习通或高校教辅平台 (<http://tch.itymb.com>) 布置本节课作业以及下节课的预习作业。

#### 第五课时（上机练习）

上机练习主要针对本章中需要重点掌握的知识点,以及在程序中容易出错的内容进行练习,通过上机练习可以考察同学对知识点的掌握情况,对代码的熟练程度。

**上机：（考察知识点为：选择媒体 API、图片预览 API、文件上传 API、文件下载 API、【案例 4-3】头像上传下载-准备工作、【案例 4-3】头像上传下载-实现“头像上传下载”微信小程序的页面结构、【案例 4-3】头像上传下载-实现“头像上传下载”微信小程序的页面逻辑）**

**形式：独立完成**

**题目：完成【案例 4-3】头像上传与下载的开发。**

**具体要求：**

- (1) 创建项目、配置导航栏、复制素材和启动服务器。
- (2) 实现“头像上传下载”微信小程序的页面结构。
- (3) 实现“头像上传下载”微信小程序的页面逻辑。

#### 第六课时

**（canvas 组件、画布 API、【案例 4-4】模拟时钟-准备工作、【案例 4-4】模拟时钟-初始化画布、【案例 4-4】模拟时钟-绘制表盘、【案例 4-4】模拟时钟-绘制指针、【案例 4-4】模拟时钟-实现时钟走动效果）**

#### 一、复习巩固

教师通过上节课作业的完成情况,对学生吸收不好的知识点进行再次巩固讲解。

#### 二、通过情景引入的方式导入新课

“模拟时钟”微信小程序是一个简约风格的动态时钟,该时钟时间与系统时间一致,且时针、分针、秒针会与系统时间同步更新,用户可以很方便地查看时间。本节课将详细讲解“模拟时钟”微信小程序用到的相关知识点和开发。

#### 三、新课讲解

##### 知识点 1-canvas 组件

教师通过 PPT 的方式讲解 canvas 组件。

- (1) canvas 组件的作用。
- (2) canvas 组件的语法格式。
- (3) 列举 canvas 组件的常用属性。
- (4) 通过示例代码演示 canvas 组件的使用。

##### 知识点 2-画布 API

教师通过 PPT 的方式讲解画布 API。

- (1) 画布 API 的作用。
- (2) 画布 API 的实现方法。

	<p>(3) 列举 <code>RenderingContext</code> 实例的常用属性和方法。</p> <p>(4) 通过绘制矩形和笑脸为例,演示使用 <code>canvas</code> 组件绘制图案的基本步骤。</p> <p><b>知识点 3-【案例 4-4】模拟时钟-准备工作</b></p> <p>教师通过 PPT 结合实际操作的方式讲解开发“模拟时钟”微信小程序的准备工作。</p> <p>(1) 创建项目。</p> <p>(2) 配置导航栏。</p> <p><b>知识点 4-【案例 4-4】模拟时钟-初始化画布</b></p> <p>教师通过 PPT 结合实际操作的方式讲解初始化画布的实现。</p> <p>(1) 通过代码实现初始化画布。</p> <p>(2) 运行程序,查看运行结果。</p> <p><b>知识点 5-【案例 4-4】模拟时钟-绘制表盘</b></p> <p>教师通过 PPT 结合实际操作的方式讲解绘制表盘的实现。</p> <p>(1) 通过代码实现表盘的绘制。</p> <p>(2) 运行程序,查看运行结果。</p> <p><b>知识点 6-【案例 4-4】模拟时钟-绘制指针</b></p> <p>教师通过 PPT 结合实际操作的方式讲解绘制指针的实现。</p> <p>(1) 通过代码实现指针的绘制。</p> <p>(2) 运行程序,查看运行结果。</p> <p><b>知识点 7-【案例 4-4】模拟时钟-实现时钟走动效果</b></p> <p>教师通过 PPT 结合实际操作的方式讲解时钟的走动效果的实现。</p> <p>(1) 通过代码实现时钟的走动效果。</p> <p>(2) 运行程序,查看运行结果。</p> <p>四、归纳总结</p> <p>教师回顾本节课所讲的知识,并通过测试题的方式引导学生解答问题并给予指导。</p> <p>五、布置作业</p> <p>教师通过学习通或高校教辅平台 (<a href="http://tch.ityxb.com">http://tch.ityxb.com</a>) 布置本节课作业以及下节课的预习作业。</p>
教学后记	<p>本章学习常用 API,学生通过“音乐播放器”“录音机”“模拟时钟”等案例掌握了 API 调用方法。在“模拟时钟”案例的时钟走动效果实现中,部分学生对定时器与画布重绘逻辑理解困难,需通过分步拆解代码、可视化演示帮助学生理解。</p>

## 第 5 章 微信小程序常用 API（下）

课题名称	第 5 章 微信小程序常用 API（下）	计划课时	9 课时
教学引入	第 4 章讲解了背景音频、录音、音频等 API 的使用，使用这些 API 即可实现相应的功能。本章将继续通过案例的方式讲解微信小程序中常用 API 的使用，主要包括如何使用动画 API 制作动画、如何通过登录 API 实现用户登录、如何调用微信开放能力实现用户头像和昵称的获取、如何通过地图 API 实现地图功能，以及如何通过 WebSocketAPI、SocketTask 实现在线聊天。		
教学目标	<ul style="list-style-type: none"> <li>● 使学生掌握动画 API，能够完成动画的制作</li> <li>● 使学生熟悉登录流程时序，能够归纳微信小程序的登录流程</li> <li>● 使学生掌握登录 API，能够运用 wx.login() 方法实现用户登录</li> <li>● 使学生掌握数据缓存 API，能够对数据进行存储、获取和移除等操作</li> <li>● 使学生掌握头像昵称填写功能，能够实现头像选择和昵称填写</li> <li>● 使学生掌握腾讯地图 SDK，能够实现腾讯地图 SDK 的接入和使用</li> <li>● 使学生掌握 map 组件，能够灵活运用 map 组件实现地图效果</li> <li>● 使学生掌握地图 API，能够实现地图中地理位置的获取等功能</li> <li>● 使学生掌握位置 API，能够实现获取当前地理位置的功能</li> <li>● 使学生掌握路由 API，能够利用路由 API 实现页面跳转</li> <li>● 使学生掌握 WebSocket API，能够成功创建 WebSocket 连接</li> <li>● 使学生掌握 SocketTask，能够使用 SocketTask 管理 WebSocket 连接</li> </ul>		
教学重点	<ul style="list-style-type: none"> <li>● 动画 API</li> <li>● 登录 API</li> <li>● 数据缓存 API</li> <li>● App()函数</li> <li>● 地图 API</li> <li>● 路由 API</li> <li>● WebSocket API</li> <li>● SocketTask</li> </ul>		
教学难点	<ul style="list-style-type: none"> <li>● WebSocket API</li> <li>● SocketTask</li> </ul>		
教学方式	课堂教学以 PPT 讲授为主，并结合多媒体进行教学		
思政元素	<p>身份认证与隐私保护意识：讲解登录 API 与用户头像昵称获取功能时，强调用户隐私保护规范（如获取用户信息需获得授权、数据缓存安全存储），引导学生尊重用户隐私，培养学生在开发中保护用户个人信息的法律意识与职业操守。</p> <p>地理信息与家国情怀：在“查看附近美食餐厅”案例中，使用腾讯地图 SDK 获取地理位置时，引导学生关注地理信息与生活服务的结合，同时介绍我国地理信息产业的发展成果，增强学生对国家科技发展的自豪感与家国情怀。</p>		

<p>教 学 过 程</p>	<p style="text-align: center;"><b>第一课时</b></p> <p style="text-align: center;">(动画 API、【案例 5-1】罗盘动画-准备工作、【案例 5-1】罗盘动画-实现“罗盘动画”微信小程序的页面结构、【案例 5-1】罗盘动画-实现“罗盘动画”微信小程序的页面逻辑)</p> <p>一、复习巩固</p> <p>教师通过上节课作业的完成情况，对学生吸收不好的知识点进行再次巩固讲解。</p> <p>二、通过情景引入的方式导入新课</p> <p>在日常开发中，常常会在页面中加入一些动画效果，例如旋转、缩放、移动等，通过这些动画效果可以提高用户体验。本节课以“罗盘动画”微信小程序为例，讲解微信小程序中动画效果的制作。</p> <p>三、新课讲解</p> <p><b>知识点 1-动画 API</b></p> <p>教师通过 PPT 结合实际操作的方式讲解动画 API。</p> <ol style="list-style-type: none"> <li>(1) 什么是动画 API。</li> <li>(2) 动画 API 的使用方法。</li> <li>(3) 列举 wx.createAnimation()方法的常用选项。</li> <li>(4) 列举 timingFunction 选项的合法值。</li> <li>(5) 列举 Animation 实例的常用方法。</li> <li>(6) 通过案例演示动画 API 的使用。</li> </ol> <p><b>知识点 2-【案例 5-1】罗盘动画-准备工作</b></p> <p>教师通过 PPT 结合实际操作的方式讲解开发“罗盘动画”微信小程序的准备工作。</p> <ol style="list-style-type: none"> <li>(1) 创建项目。</li> <li>(2) 配置页面。</li> <li>(3) 配置导航栏。</li> <li>(4) 复制素材。</li> </ol> <p><b>知识点 3-【案例 5-1】罗盘动画-实现“罗盘动画”微信小程序的页面结构</b></p> <p>教师通过 PPT 结合实际操作的方式讲解“罗盘动画”微信小程序页面结构的实现。</p> <ol style="list-style-type: none"> <li>(1) 编写图片区域的页面结构。</li> <li>(2) 编写按钮区域的页面结构。</li> </ol> <p><b>知识点 4-【案例 5-1】罗盘动画-实现“罗盘动画”微信小程序的页面逻辑</b></p> <p>教师通过 PPT 结合实际操作的方式讲解“罗盘动画”微信小程序的页面逻辑的实现步骤。</p> <ol style="list-style-type: none"> <li>(1) 在 pages/compass/compass.js 文件中编写页面所需要的数据。</li> <li>(2) 编写 onReady()函数，在页面初次渲染完成时，创建一个动画实例 animation。</li> <li>(3) 编写 rotate()函数，实现从原点随机旋转某一个角度的效果。</li> <li>(4) 编写 scale()函数，实现图片随机缩放的效果。</li> <li>(5) 编写 translate()函数，实现平移变换的效果。</li> <li>(6) 编写 skew()函数，实现使对象相对 x、y 轴坐标进行随机倾斜。</li> <li>(7) 编写 rotateAndScale()函数，实现同时进行旋转和缩放。</li> <li>(8) 编写 rotateThenScale()函数，实现旋转之后缩放的效果。</li> </ol>
----------------------------	---

- (9) 编写 all()函数，实现同时展示全部动画。
- (10) 编写 allOrder()函数，实现按顺序展示全部动画。
- (11) 编写 reset()函数，实现将动画回到初始状态。

#### 四、归纳总结

教师回顾本节课所讲的知识，并通过测试题的方式引导学生解答问题并给予指导。

#### 五、布置作业

教师通过学习通或高校教辅平台（<http://tch.ityx.com>）布置本节课作业以及下节课的预习作业。

### 第二课时（上机练习）

上机练习主要针对本章中需要重点掌握的知识点，以及在程序中容易出错的内容进行练习，通过上机练习可以考察同学对知识点的掌握情况，对代码的熟练程度。

**上机：（考察知识点为：动画 API、【案例 5-1】罗盘动画-准备工作、【案例 5-1】罗盘动画-实现“罗盘动画”微信小程序的页面结构、【案例 5-1】罗盘动画-实现“罗盘动画”微信小程序的页面逻辑）**

**形式：独立完成**

**题目：完成【案例 5-1】罗盘动画的开发。**

**具体要求：**

- (1) 创建项目、配置页面、配置导航栏和复制素材。
- (2) 实现“罗盘动画”微信小程序的页面结构。
- (3) 获取用户积分。
- (4) 实现“罗盘动画”微信小程序的页面逻辑。

### 第三、四课时

**（登录流程时序、登录 API、数据缓存 API、头像昵称填写、App()函数）**

#### 一、复习巩固

教师通过上节课作业的完成情况，对学生吸收不好的知识点进行再次巩固讲解。

#### 二、通过情景引入的方式导入新课

在日常生活中，需要用户登录的场景有很多，例如，当用户在手机中浏览文章想要收藏时、在线上购买商品时、进入软件查询个人信息时，只有用户登录自己的账号以后，才可以进一步使用这些功能，本节课将详细讲解“用户登录”微信小程序用到的相关知识点。

#### 三、新课讲解

##### 知识点 1-登录流程时序

教师通过 PPT 的方式讲解登录流程时序。

- (1) 什么是登录流程时序。
- (2) 用户登录流程的参与角色。
  - 小程序
  - 开发者服务器
  - 微信接口服务
- (3) 登录流程实现步骤。

- 微信小程序获取临时登录凭证 code;
- 微信小程序将 code 发送给开发者服务器;
- 开发者服务器通过微信接口服务校验登录凭证;
- 开发者服务器自定义登录态。

### 知识点 2-登录 API

教师通过 PPT 结合实际操作的方式讲解登录 API。

- (1) 登录 API 的使用方法。
- (2) 列举 wx.login()方法的常用选项。
- (3) 通过示例代码演示 wx.login()方法的使用。

### 知识点 3-数据缓存 API

教师通过 PPT 结合实际操作的方式讲解数据缓存 API。

- (1) 数据缓存 API 的作用。
- (2) 列举常见的数据异步缓存方法。
- (3) 列举常见的数据同步缓存方法。
- (4) 列举 wx.setStorage()方法的常用选项。
- (5) 列举 wx.getStorage()方法的常用选项。
- (6) 异步方法实现缓存数据的特点。
- (7) 同步方法实现缓存数据的特点。
- (8) 通过示例代码演示异步方法缓存数据的存储和获取。

### 知识点 4-头像昵称填写

教师通过 PPT 结合实际操作的方式讲解头像昵称填写的使用。

- (1) 为什么要获取头像昵称。
- (2) 头像选择的使用方法。
- (3) 昵称填写的使用方法。
- (4) 通过示例代码演示头像昵称填写功能的使用。

### 知识点 5-App()函数

教师通过 PPT 结合实际操作的方式讲解 App()函数。

- (1) 什么是 App()函数。
- (2) 如何理解应用生命周期函数。
- (3) 应用生命周期回调函数。
  - onLaunch()
  - onShow()
  - onError()
  - onHide()
  - onPageNotFound()
- (5) 通过示例代码演示 App 实例的获取。

#### 四、归纳总结

教师回顾本节课所讲的知识,并通过测试题的方式引导学生解答问题并给予指导。

#### 五、布置作业

教师通过学习通或高校教辅平台 (<http://tch.itymb.com>) 布置本节课作业以及下节课的预习作业。

#### 第五课时

(【案例 5-2】用户登录-准备工作、【案例 5-2】用户登录-实现用户登录、【案例 5-2】用户登录-检查用户是否已经登录、【案例 5-2】用户登录-获取用户的积分、【案例 5-2】用户登录-获取用户头像和昵称)

#### 一、复习巩固

教师通过上节课作业的完成情况，对学生吸收不好的知识点进行再次巩固讲解。

#### 二、通过直接引入的方式导入新课

在上节课中学习了“用户登录”微信小程序用到的相关知识点，包括登录流程时序、登录 API、数据缓存 API、头像昵称填写、App()函数相关内容，在本节课中将详细讲解“用户登录”微信小程序的开发。

#### 三、新课讲解

##### 知识点 1-【案例 5-2】用户登录-准备工作

教师通过 PPT 结合实际操作的方式讲解开发“用户登录”微信小程序的准备工作。

- (1) 创建项目。
- (2) 配置导航栏。
- (3) 复制素材。
- (4) 启动服务器

##### 知识点 2-【案例 5-2】用户登录-实现用户登录

教师通过 PPT 结合实际操作的方式讲解用户登录的实现。

- (1) 定义页面所需的数据。
- (2) 在微信小程序启动时调用 login()方法实现自动登录。
- (3) 在微信开发者工具的本地设置中勾选“不校验合法域名、web-view (业务域名)、TLS 版本以及 HTTPS 证书”复选框。

##### 知识点 3-【案例 5-2】用户登录-检查用户是否已经登录

教师通过 PPT 结合实际操作的方式讲解如何检查用户是否已经登录。

- (1) 检查用户是否已经登录的实现思路。
- (2) 检查用户是否已经登录的具体实现步骤。

##### 知识点 4-【案例 5-2】用户登录-获取用户的积分

教师通过 PPT 结合实际操作的方式讲解用户积分的获取。

- (1) 通过代码实现用户积分的获取。
- (2) 运行程序，查看运行结果。

##### 知识点 5-【案例 5-2】用户登录-获取用户头像和昵称

教师通过 PPT 结合实际操作的方式讲解用户头像和昵称的获取。

- (1) 通过代码实现用户头像和昵称的获取。
- (2) 运行程序，查看运行结果。

#### 四、归纳总结

教师回顾本节课所讲的知识，并通过测试题的方式引导学生解答问题并给予指导。

#### 五、布置作业

教师通过学习通或高校教辅平台 (<http://tch.ityxb.com>) 布置本节课作业以及下节课的预习作业。

## 第六课时

### (腾讯地图 SDK、map 组件、地图 API、位置 API、路由 API)

#### 一、复习巩固

教师通过上节课作业的完成情况，对学生吸收不好的知识点进行再次巩固讲解。

#### 二、通过情景引入的方式导入新课

信息化时代，人们仅通过手机即可了解到周边的娱乐、餐饮等信息，即使到了人生地不熟的地方，也可以通过手机地图快速了解周围的环境。本节课将详细讲解“查看附近美食餐厅”微信小程序用到的相关知识点。

#### 三、新课讲解

##### 知识点 1-腾讯地图 SDK

教师通过 PPT 结合实际操作的方式讲解腾讯地图 SDK。

- (1) 腾讯地图 SDK 的作用。
- (2) 腾讯地图 SDK 的 5 个使用步骤。
  - 申请开发者秘钥；
  - 下载腾讯地图 SDK；
  - 登录微信小程序管理后台添加合法域名；
  - 在微信小程序中声明权限；
  - 使用腾讯地图 SDK。
- (3) 通过示例代码演示腾讯地图 SDK 的使用。

##### 知识点 2-map 组件

教师通过 PPT 结合实际操作的方式讲解 map 组件。

- (1) map 组件的作用。

在微信小程序中，map 组件可以为用户提供地图功能。map 组件可以在页面中显示地图，并且支持移动、缩放、添加标记点等功能。

- (2) map 组件的定义。

map 组件通过<map></map>标签来定义。

- (3) 列举 map 组件的常用属性。
- (4) 列举 marker 对象的属性。
- (5) 通过示例代码演示 marker 标记点的定义。

##### 知识点 3-地图 API

教师通过 PPT 结合实际操作的方式讲解地图 API。

- (1) 地图 API 的作用。
- (2) 地图 API 的创建。
- (3) 列举 MapContext 实例的常用方法。
- (4) 列举 MapContext 实例的 getCenterLocation()方法的常用选项。
- (5) 列举 MapContext 实例的 moveToLocation()方法的常用选项。
- (6) 通过示例代码演示当前地图中心经纬度的获取。

##### 知识点 4-位置 API

教师通过 PPT 结合实际操作的方式讲解位置 API。

- (1) 位置 API 的作用。
- (2) 使用位置 API 的注意事项。
- (3) 列举 wx.getLocation()方法的常用选项。
- (4) wx.getLocation()方法的 success()回调函数的参数是一个对象，列

举该对象的属性。

(5) 通过示例代码演示 wx.getLocation()方法的使用。

#### 知识点 5-路由 API

教师通过 PPT 结合实际操作的方式讲解路由 API。

- (1) 路由 API 的作用。  
路由 API 用于实现页面跳转。
- (2) 列举常用的路由 API 方法。
- (3) 列举路由 API 方法的常用选项。
- (4) 通过示例代码演示路由 API 的使用。

#### 四、归纳总结

教师回顾本节课所讲的知识，并通过测试题的方式引导学生解答问题并给予指导。

#### 五、布置作业

教师通过学习通或高校教辅平台 (<http://tch.itymb.com>) 布置本节课作业以及下节课的预习作业。

### 第七课时

(【案例 5-3】查看附近美食餐厅-准备工作、【案例 5-3】查看附近美食餐厅-获取初始数据、【案例 5-3】查看附近美食餐厅-在页面中显示地图、【案例 5-3】查看附近美食餐厅-实现跳转到优惠券页面、【案例 5-3】查看附近美食餐厅-实现查找附近美食餐厅功能)

#### 一、复习巩固

教师通过上节课作业的完成情况，对学生吸收不好的知识点进行再次巩固讲解。

#### 二、通过直接引入的方式导入新课

在上节课中学习了“查看附近美食餐厅”微信小程序用到的相关知识点，包括腾讯地图 SDK、map 组件、地图 API、位置 API、路由 API 相关内容，在本节课中将详细讲解“查看附近美食餐厅”微信小程序的开发。

#### 三、新课讲解

##### 知识点 1-【案例 5-3】查看附近美食餐厅-准备工作

教师通过 PPT 结合实际操作的方式讲解开发“查看附近美食餐厅”微信小程序的准备工作。

- (1) 创建项目。
- (2) 配置页面。
- (3) 配置导航栏。
- (4) 复制素材。

##### 知识点 2-【案例 5-3】查看附近美食餐厅-获取初始数据

教师通过 PPT 结合实际操作的方式讲解获取初始数据的实现。

- (1) 通过代码定义页面所需的数据。
- (2) 运行程序，测试是否实现该功能。

##### 知识点 3-【案例 5-3】查看附近美食餐厅-在页面中显示地图

教师通过 PPT 结合实际操作的方式讲解在页面中显示地图的实现。

- (1) 通过代码实现在地图界面上显示中心点的图片和附近美食餐厅的图片。

(2) 运行程序，测试是否实现该功能。

#### 知识点 4-【案例 5-3】查看附近美食餐厅-实现跳转到优惠券页面

教师通过 PPT 结合实际操作的方式讲解跳转到优惠券页面的实现。

(1) 通过代码实现点击优惠券，实现跳转到优惠券页面。

(2) 运行程序，测试是否实现该功能。

#### 知识点 5-【案例 5-3】查看附近美食餐厅-实现查找附近美食餐厅功能

教师通过 PPT 结合实际操作的方式讲解查找附近美食餐厅功能的实现。

(1) 通过代码实现根据当前位置（经度和纬度）搜索附近的餐厅，并将其通过标记点标记出来。

(2) 运行程序，测试是否实现该功能。

#### 四、归纳总结

教师回顾本节课所讲的知识，并通过测试题的方式引导学生解答问题并给予指导。

#### 五、布置作业

教师通过学习通或高校教辅平台（<http://tch.ityx.com>）布置本节课作业以及下节课的预习作业。

#### 第八课时（上机练习）

上机练习主要针对本章中需要重点掌握的知识点，以及在程序中容易出错的内容进行练习，通过上机练习可以考察同学对知识点的掌握情况，对代码的熟练程度。

上机：（考察知识点为：腾讯地图 SDK、map 组件、地图 API、位置 API、路由 API、【案例 5-3】查看附近美食餐厅-准备工作、【案例 5-3】查看附近美食餐厅-获取初始数据、【案例 5-3】查看附近美食餐厅-在页面中显示地图、【案例 5-3】查看附近美食餐厅-实现跳转到优惠券页面、【案例 5-3】查看附近美食餐厅-实现查找附近美食餐厅功能）

形式：单独完成

题目：完成【案例 5-3】查看附近美食餐厅的开发。

具体要求：

- (1) 创建项目、配置页面、配置导航栏和复制素材。
- (2) 获取初始数据。
- (3) 在页面中显示地图。
- (4) 实现跳转到优惠券页面。
- (5) 实现查找附近美食餐厅功能。

#### 第九课时

（WebSocket API、SocketTask、【案例 5-4】在线聊天-准备工作、【案例 5-4】在线聊天-实现 WebSocket 连接、【案例 5-4】在线聊天-实现发送消息功能、【案例 5-4】在线聊天-实现消息列表展示功能）

#### 一、复习巩固

教师通过上节课作业的完成情况，对学生吸收不好的知识点进行再次巩固讲解。

#### 二、通过情景引入的方式导入新课

在一些购物平台中，经常会通过人工客服或机器客服来为用户提供服务，

及时解答用户提出的问题，提高用户的使用体验。对于这类需求，就需要通过在微信小程序中开发在线聊天功能来实现。本节将对“在线聊天”微信小程序进行详细讲解。

### 三、新课讲解

#### 知识点 1- WebSocket API

教师通过 PPT 结合实际操作的方式讲解 WebSocket API。

- (1) WebSocket API 的作用。
- (2) WebSocket 的概念。
- (3) 列举 wx.connectSocket()方法的常用选项。
- (4) 通过示例代码演示 wx.connectSocket()方法的使用。

#### 知识点 2- SocketTask

教师通过 PPT 结合实际操作的方式讲解动画 API。

- (1) SocketTask 的作用。
- (2) 通过示例代码演示 SocketTask 实例的创建。
- (3) 列举 SocketTask 实例的常用方法。
- (4) 列举 SocketTask 实例的 send()方法的常用选项。
- (5) 列举 SocketTask 实例的 close()方法的常用选项。
- (6) 列举 SocketTask 实例的 onClose()方法的属性。
- (7) 列举 SocketTask 实例的 onOpen()方法的属性。
- (8) 通过示例代码演示 SocketTask 的使用。

#### 知识点 3-【案例 5-4】在线聊天-准备工作

教师通过 PPT 结合实际操作的方式讲解开发“在线聊天”微信小程序的准备工作。

- (1) 创建项目。
- (2) 配置导航栏。
- (3) 复制素材。
- (4) 启动服务器。

#### 知识点 4-【案例 5-4】在线聊天-实现 WebSocket 连接

教师通过 PPT 结合实际操作的方式讲解 WebSocket 连接的实现。

- (1) 通过代码实现“在线聊天”微信小程序的 WebSockt 连接。
- (2) 运行程序，查看 WebSocket 连接成功的控制台输出结果。

#### 知识点 5-【案例 5-4】在线聊天-实现发送消息功能

教师通过 PPT 结合实际操作的方式讲解发送消息的实现。

- (1) 通过代码实现 WebSocket 连接成功后，将消息发送给服务器的功能。
- (2) 运行程序，在微信小程序的输入框中输入“您好。服务器”，查看服务器的控制台输出的结果、微信小程序的控制台输出的消息。

#### 知识点 6-【案例 5-4】在线聊天-实现消息列表展示功能

教师通过 PPT 结合实际操作的方式讲解消息列表展示的实现。

- (1) 通过代码实现将服务器和微信小程序的消息列表展示在页面中效果。
- (2) 运行程序，页面中展示服务器和微信小程序发送的消息效果图。

### 四、归纳总结

教师回顾本节课所讲的知识，并通过测试题的方式引导学生解答问题并给

	予指导。 五、布置作业 教师通过学习通或高校教辅平台（ <a href="http://tch.ityxb.com">http://tch.ityxb.com</a> ）布置本节课作业以及下节课的预习作业。
教学后记	地图 SDK 有一定难度，但可结合大模型 API 给学生讲解数据 API 的使用

## 第 6 章 综合项目——“点餐”微信小程序

课题名称	第 6 章 综合项目——“点餐”微信小程序	计划课时	9 课时
教学引入	<p>在学习了微信小程序的基础知识后，接下来将开发一个综合项目——“点餐”微信小程序，让读者对所学知识进行综合运用。该微信小程序的页面包括商家首页、菜单列表页、订单确认页、订单详情页、订单列表页和消费记录页，具有用户登录、购物车、创建订单、查看订单、查询消费记录等功能。本章将对“点餐”微信小程序进行详细讲解。</p>		
教学目标	<ul style="list-style-type: none"> <li>● 使学生掌握封装网络请求的方法，能够通过封装网络请求简化项目中网络请求的代码</li> <li>● 使学生掌握用户登录的开发，能够独立实现用户登录</li> <li>● 使学生掌握商家首页的开发，能够独立完成商家首页的编写</li> <li>● 使学生掌握菜单列表页的开发，能够独立完成菜单列表页的编写</li> <li>● 使学生掌握购物车的开发，能够独立实现购物车</li> <li>● 使学生掌握订单确认页的开发，能够独立完成订单确认页的编写</li> <li>● 使学生掌握订单详情页的开发，能够独立完成订单详情页的编写</li> <li>● 使学生掌握订单列表页的开发，能够独立完成订单列表页的编写</li> <li>● 使学生掌握消费记录页的开发，能够独立完成消费记录页的编写</li> </ul>		
教学重点	<ul style="list-style-type: none"> <li>● 【任务 6-2】封装网络请求-封装网络请求函数</li> <li>● 【任务 6-3】用户登录-判断登录状态</li> <li>● 【任务 6-3】用户登录-执行登录操作</li> <li>● 【任务 6-6】购物车-实现添加商品到购物车</li> </ul>		
教学难点	<ul style="list-style-type: none"> <li>● 【任务 6-5】菜单列表页-实现点击左侧菜单项滚动右侧商品列表</li> <li>● 【任务 6-5】菜单列表页-实现滚动右侧商品列表激活左侧菜单项</li> <li>● 【任务 6-6】购物车-实现小球动画效果</li> </ul>		

<p>教学方式</p>	<p>课堂教学以 PPT 讲授为主，并结合多媒体进行教学</p>
<p>思政元素</p>	<p>商业伦理与诚信经营：讲解订单创建、消费记录功能时，强调数据准确性与订单信息透明（如订单金额计算精准、消费记录可追溯），引导学生树立诚信经营的商业伦理，培养学生在商业项目开发中坚守道德底线的职业素养。</p>
<p>教学过程</p>	<p style="text-align: center;"><b>第一课时</b></p> <p style="text-align: center;"><b>（【任务 6-1】项目开发准备-开发背景、【任务 6-1】项目开发准备-项目模块划分、【任务 6-1】项目开发准备-项目初始化）</b></p> <p>一、复习巩固 教师通过上节课作业的完成情况，对学生吸收不好的知识点进行再次巩固讲解。</p> <p>二、通过情景引入的方式导入新课 现如今，相比以服务员为中介完成点餐、送餐、买单的传统点餐方式，越来越多的餐厅开始使用微信小程序进行点餐。基于这一背景，开发一个“点餐”微信小程序。本节课将对“点餐”微信小程序的开发背景、模块划分和项目初始化进行讲解。</p> <p>三、新课讲解</p> <p><b>知识点 1-【任务 6-1】项目开发准备-开发背景</b> 教师通过 PPT 的方式讲解“点餐”微信小程序的开发背景。</p> <ol style="list-style-type: none"> <li>(1) “点餐”微信小程序的开发背景。</li> <li>(2) “点餐”微信小程序的优势。</li> </ol> <p><b>知识点 2-【任务 6-1】项目开发准备-项目模块划分</b> 教师通过 PPT 的方式讲解“点餐”微信小程序的模块划分。</p> <ol style="list-style-type: none"> <li>(1) 用户登录。</li> <li>(2) 商家首页。</li> <li>(3) 菜单列表页。</li> <li>(4) 购物车。</li> <li>(5) 订单确认页。</li> <li>(6) 订单详情页。</li> <li>(7) 订单列表页。</li> <li>(8) 消费记录页。</li> </ol> <p><b>知识点 3-【任务 6-1】项目开发准备-项目初始化</b> 教师通过 PPT 结合实际操作的方式讲解“点餐”微信小程序的初始化。</p> <ol style="list-style-type: none"> <li>(1) 创建项目。</li> <li>(2) 配置页面。</li> <li>(3) 配置导航栏。</li> <li>(4) 复制素材。</li> <li>(5) 配置标签栏。</li> </ol> <p>四、归纳总结 教师回顾本节课所讲的知识，并通过提问的方式引导学生解答问题并给予指导。</p> <p>五、布置作业</p>

教师通过学习通或高校教辅平台（<http://tch.itymb.com>）布置本节课作业以及下节课的预习作业。

## 第二课时

（【任务 6-2】封装网络请求-保存接口地址、【任务 6-2】封装网络请求-封装网络请求函数、【任务 6-2】封装网络请求-请求失败时的处理、【任务 6-2】封装网络请求-请求成功时的处理）

### 一、复习巩固

教师通过上节课作业的完成情况，对学生吸收不好的知识点进行再次巩固讲解。

### 二、通过需求引入的方式导入新课

在“点餐”微信小程序中，商家首页、菜单列表页、订单详情页等多个页面都需要发送网络请求，从服务器中获得数据。由于网络请求的代码使用较为频繁，并且部分请求参数及响应结果处理非常类似，为了提高代码的可重用性，对项目中的网络请求代码进行了封装。本节课将详细讲解网络请求的封装。

### 三、新课讲解

#### 知识点 1-【任务 6-2】封装网络请求-保存接口地址

教师通过 PPT 结合实际操作的方式讲解保存接口地址的实现。

##### (1) 为什么要保存接口地址。

在实际项目开发中，很多页面的请求地址 URL 的前半部分都是相同的，重复书写会导致代码冗余，而且如果请求地址更换了域名，修改也比较麻烦。将 URL 的公共部分提取出来，单独放置到配置文件中，方便后期修改。

##### (2) 如何保存接口地址。

在 `utils` 文件夹下新建 `config.js` 文件，在 `utils/config.js` 文件中编写 URL 的公共部分，完成接口地址的保存。

#### 知识点 2-【任务 6-2】封装网络请求-封装网络请求函数

教师通过 PPT 结合实际操作的方式讲解封装网络请求函数的实现。

##### (1) 分析封装网络请求的实现思路。

##### (2) 通过代码实现封装网络请求。

##### (3) 通过案例测试封装的网络请求函数是否可以正常使用。

#### 知识点 3-【任务 6-2】封装网络请求-请求失败时的处理

教师通过 PPT 结合实际操作的方式讲解数据请求失败时的处理。

##### (1) 分析数据请求失败后的实现思路。

##### (2) 编写数据请求失败的逻辑处理代码。

##### (3) 通过案例演示数据请求失败时的操作。

#### 知识点 4-【任务 6-2】封装网络请求-请求成功时的处理

教师通过 PPT 结合实际操作的方式讲解数据请求成功时的处理。

##### (1) 分析数据请求成功后的实现思路。

##### (2) 编写数据请求成功的逻辑处理代码。

##### (3) 通过案例演示数据请求成功时的操作。

### 四、归纳总结

教师回顾本节课所讲的知识，并通过提问的方式引导学生解答问题并给予指导。

### 五、布置作业

教师通过学习通或高校教辅平台 (<http://tch.ityxb.com>) 布置本节课作业以及下节课的预习作业。

### 第三课时

(【任务 6-3】用户登录-判断登录状态、【任务 6-3】用户登录-执行登录操作、【任务 6-3】用户登录-记住登录状态)

#### 一、复习巩固

教师通过上节课作业的完成情况, 对学生吸收不好的知识点进行再次巩固讲解。

#### 二、通过需求引入的方式导入新课

在“点餐”微信小程序中, 只有用户登录后, 才可以进行下单操作、获取之前的消费记录等。本节课将详细讲解用户登录功能的实现。

#### 三、新课讲解

##### 知识点 1-【任务 6-3】用户登录-判断登录状态

教师通过 PPT 结合实际操作的方式讲解如何判断用户登录状态。

- (1) 分析登录状态的实现思路。
- (2) 通过代码实现判断登录状态的逻辑处理。

##### 知识点 2-【任务 6-3】用户登录-执行登录操作

教师通过 PPT 结合实际操作的方式讲解如何执行登录操作。

- (1) 分析登录操作的实现思路。
- (2) 通过代码实现登录操作的逻辑处理。

##### 知识点 3-【任务 6-3】用户登录-记住登录状态

教师通过 PPT 结合实际操作的方式讲解如何记住用户登录状态。

- (1) 分析记住用户登录状态的实现思路。
- (2) 通过代码实现记住用户登录状态的逻辑处理。

#### 四、归纳总结

教师回顾本节课所讲的知识, 并通过提问的方式引导学生解答问题并给予指导。

#### 五、布置作业

教师通过学习通或高校教辅平台 (<http://tch.ityxb.com>) 布置本节课作业以及下节课的预习作业。

上机练习主要针对本章中需要重点掌握的知识点, 以及在程序中容易出错的内容进行练习, 通过上机练习可以考察同学对知识点的掌握情况, 对代码的熟练程度。

上机: (考察知识点为: 【任务 6-1】项目开发准备-项目初始化、【任务 6-2】封装网络请求-保存接口地址、【任务 6-2】封装网络请求-封装网络请求函数、【任务 6-2】封装网络请求-请求失败时的处理、【任务 6-2】封装网络请求-请求成功时的处理、【任务 6-3】用户登录-判断登录状态、【任务 6-3】用户登录-执行登录操作、【任务 6-3】用户登录-记住登录状态)

形式: 独立完成

题目: 完成【任务 6-1】项目开发准备、【任务 6-2】封装网络请求、【任务 6-3】用户登录的开发。

具体要求:

- (1) 项目初始化。
- (2) 保存接口地址。
- (3) 封装网络请求函数。
- (4) 请求失败时的处理。
- (5) 请求成功时的处理。
- (6) 判断登录状态。
- (7) 执行登录操作。
- (8) 记住登录状态。

#### 第四课时

（【任务 6-4】商家首页-加载商家首页数据、【任务 6-4】商家首页-实现商家首页的轮播图区域、【任务 6-4】商家首页-实现商家首页的中间区域、【任务 6-4】商家首页-实现商家首页的底部区域）

##### 一、复习巩固

教师通过上节课作业的完成情况，对学生吸收不好的知识点进行再次巩固讲解。

##### 二、通过需求引入的方式导入新课

“点餐”微信小程序的商家首页整体分为轮播图区域、中间区域和底部区域 3 个部分，点击“开启点餐之旅→”按钮可以进行点餐。本节课将详细讲解商家首页的开发。

##### 三、新课讲解

###### 知识点 1-【任务 6-4】商家首页-加载商家首页数据

教师通过 PPT 结合实际操作的方式讲解商家首页数据的获取。

- (1) 通过代码实现商家首页数据的获取。
- (2) 运行程序，查看首页数据是否获取成功。

###### 知识点 2-【任务 6-4】商家首页-实现商家首页的轮播图区域

教师通过 PPT 结合实际操作的方式讲解商家首页轮播图区域的实现。

- (1) 编写商家首页轮播图区域的页面结构。
- (2) 编写商家首页轮播图区域的页面样式。

###### 知识点 3-【任务 6-4】商家首页-实现商家首页的中间区域

教师通过 PPT 结合实际操作的方式讲解商家首页中间区域的实现。

- (1) 编写商家首页中间区域的页面结构。
- (2) 编写商家首页中间区域的页面样式。
- (3) 编写最新消息展示区域的页面样式。
- (4) 编写 start()方法，实现跳转到菜单列表。

###### 知识点 4-【任务 6-4】商家首页-实现商家首页的底部区域

教师通过 PPT 结合实际操作的方式讲解商家首页底部区域的实现。

- (1) 编写商家首页底部区域的页面结构。
- (2) 编写商家首页底部区域的页面样式。

##### 四、归纳总结

教师回顾本节课所讲的知识，并通过提问的方式引导学生解答问题并给予指导。

##### 五、布置作业

教师通过学习通或高校教辅平台（<http://tch.ityxb.com>）布置本节课作业以

及下节课的预习作业。

### 第五课时

（【任务 6-5】菜单列表页-加载菜单列表页数据、【任务 6-5】菜单列表页-实现折扣信息区域、【任务 6-5】菜单列表页-实现菜单列表区域、【任务 6-5】菜单列表页-实现点击左侧菜单项滚动右侧商品列表、【任务 6-5】菜单列表页-实现滚动右侧商品列表激活左侧菜单项）

#### 一、复习巩固

教师通过上节课作业的完成情况，对学生吸收不好的知识点进行再次巩固讲解。

#### 二、通过需求引入的方式导入新课

在商家首页点击“开启订餐之旅→”按钮后会跳转到菜单列表页。该页面分为折扣信息区域、菜单列表区域和购物车区域，其中菜单列表区域包括菜单栏区域和商品列表区域。由于购物车区域实现起来较为复杂，将在第七课时中进行讲解。本节课将详细讲解菜单列表页的开发。

#### 三、新课讲解

##### 知识点 1-【任务 6-5】菜单列表页-加载菜单列表页数据

教师通过 PPT 结合实际操作的方式讲解菜单列表页数据的获取。

- (1) 通过代码实现在页面加载完成时获取菜单列表页所需的数据。
- (2) 运行程序，查看菜单列表页数据是否获取成功。

##### 知识点 2-【任务 6-5】菜单列表页-实现折扣信息区域

教师通过 PPT 结合实际操作的方式讲解折扣信息区域的实现。

- (1) 编写折扣信息区域的页面结构。
- (2) 编写折扣信息区域的页面样式。

##### 知识点 3-【任务 6-5】菜单列表页-实现菜单列表区域

教师通过 PPT 结合实际操作的方式讲解菜单列表区域的实现。

- (1) 编写菜单列表区域的整体页面结构。
- (2) 编写菜单列表区域的整体页面样式。
- (3) 编写菜单栏区域的页面结构。
- (4) 编写菜单栏区域的页面样式。
- (5) 编写商品列表区域的页面结构。
- (6) 编写商品列表区域的页面样式。
- (7) 金额处理，在金额前面加上“¥”符号。
- (8) 修改商品列表区域中关于价格的代码，将价格经过处理之后再输出。

##### 知识点 4-【任务 6-5】菜单列表页-实现点击左侧菜单项滚动右侧商品列表

教师通过 PPT 结合实际操作的方式讲解点击左侧菜单项滚动右侧商品列表，切换到对应菜单分类下的商品列表的实现。

- (1) 分析点击左侧菜单项滚动右侧商品列表的实现思路。
- (2) 在 list.js 文件中添加两个属性（菜单中激活项的索引、用户按下的菜单项的索引）。
- (3) 修改菜单栏区域的页面结构。
- (4) 编写菜单项未选中和被选中时的样式。
- (5) 编写事件处理函数，实现左侧被点击的菜单项样式改变和右侧商品列表随对应的菜单项进行滚动展示。

(6) 使用 scroll-view 组件中的 scroll-into-view 属性实现右侧商品列表的滚动。

#### 知识点 5-【任务 6-5】菜单列表页-实现滚动右侧商品列表激活左侧菜单项

教师通过 PPT 结合实际操作的方式讲解滚动右侧商品列表激活左侧菜单项的实现。

(1) 分析滚动右侧商品列表激活左侧菜单项的实现思路。

(2) 通过代码实现滚动右侧商品列表激活左侧菜单项。

#### 四、归纳总结

教师回顾本节课所讲的知识,并通过提问的方式引导学生解答问题并给予指导。

#### 五、布置作业

教师通过学习通或高校教辅平台 (<http://tch.itymb.com>) 布置本节课作业以及下节课的预习作业。

### 第六课时（上机练习）

上机练习主要针对本章中需要重点掌握的知识点,以及在程序中容易出错的内容进行练习,通过上机练习可以考察同学对知识点的掌握情况,对代码的熟练程度。

上机: (考察知识点为:【任务 6-4】商家首页-加载商家首页数据、【任务 6-4】商家首页-实现商家首页的轮播图区域、【任务 6-4】商家首页-实现商家首页的中间区域、【任务 6-4】商家首页-实现商家首页的底部区域、【任务 6-5】菜单列表页-加载菜单列表页数据、【任务 6-5】菜单列表页-实现折扣信息区域、【任务 6-5】菜单列表页-实现菜单列表区域、【任务 6-5】菜单列表页-实现点击左侧菜单项滚动右侧商品列表、【任务 6-5】菜单列表页-实现滚动右侧商品列表激活左侧菜单项)

形式: 独立完成

题目: 完成【任务 6-4】商家首页、【任务 6-5】菜单列表页的开发。

具体要求:

- (1) 加载商家首页数据。
- (2) 实现商家首页的轮播图区域。
- (3) 实现商家首页的中间区域。
- (4) 实现商家首页的底部区域。
- (5) 加载菜单列表页数据。
- (6) 实现折扣信息区域。
- (7) 实现菜单列表区域。
- (8) 实现点击左侧菜单项滚动右侧商品列表。
- (9) 实现滚动右侧商品列表激活左侧菜单项。

### 第七、八课时

(【任务 6-6】购物车-实现底部购物车区域、【任务 6-6】购物车-实现添加商品到购物车、【任务 6-6】购物车-实现小球动画效果、【任务 6-6】购物车-实现满减优惠信息区域、【任务 6-6】购物车-实现购物车界面区域、【任务 6-6】购物车-实现增加和减少商品数量、【任务 6-6】购物车-实现清空购物车)

#### 一、复习巩固

教师通过上节课作业的完成情况，对学生吸收不好的知识点进行再次巩固讲解。

## 二、通过需求引入的方式导入新课

购物车区域位于商品列表页面底部，当购物车里商品数量为 0 时，购物车图标处于灰色不可点击的状态；当商品状态不为 0 时，购物车图标右上角显示商品数量，图标变为可点击状态，点击购物车图标可以展开里面的商品，此时可以对商品数量进行添加或者减少操作，并动态核算全部商品的价钱。本节课将详细讲解购物车的开发。

## 三、新课讲解


### 知识点 1-【任务 6-6】购物车-实现底部购物车区域

教师通过 PPT 结合实际操作的方式讲解底部购物车区域的实现。

- (1) 定义初始数据。
- (2) 在菜单列表区域的下方编写底部购物车区域的页面结构。
- (3) 编写底部购物车区域的页面样式。
- (4) 编写“选好了”按钮样式。
- (5) 编写购物车图标样式。
- (6) 编写购物车为空的样式。

### 知识点 2-【任务 6-6】购物车-实现添加商品到购物车

教师通过 PPT 结合实际操作的方式讲解添加商品到购物车的实现。

- (1) 定义用于保存购物车数据的属性 `cartList`。
- (2) 在事件处理函数中实现点击  按钮将商品添加到购物车。
- (3) 编写购物车中的商品购买数量样式。
- (4) 编写购物车中的商品价格的样式。

### 知识点 3-【任务 6-6】购物车-实现小球动画效果

教师通过 PPT 结合实际操作的方式讲解小球动画效果的实现。

- (1) 分析小球动画效果的实现思路。
- (2) 编写代码，引入购物车动画模块。
- (3) 在页面加载完成时获取页面上的节点信息。
- (4) 修改 `addToCart()` 函数，在添加商品到购物车时调用 `start()` 方法。
- (5) 编写小球的页面结构。
- (6) 编写小球的样式。

### 知识点 4-【任务 6-6】购物车-实现满减优惠信息区域

教师通过 PPT 结合实际操作的方式讲解满减优惠信息区域的实现。

- (1) 在底部购物车区域的上方编写满减优惠信息区域的页面结构。
- (2) 编写满减优惠信息区域的页面样式。

### 知识点 5-【任务 6-6】购物车-实现购物车界面区域

教师通过 PPT 结合实际操作的方式讲解购物车界面区域的实现。

- (1) 分析购物车界面区域的实现思路。
- (2) 定义 `showCart` 属性，控制购物车界面区域是否显示。
- (3) 在页面结构中，给购物车图标绑定 `tap` 事件。
- (4) 编写绑定 `tap` 事件的 `showCartList()` 函数，实现点击购物车图标后显示购物车界面区域。
- (5) 在满减优惠信息区域的上方编写购物车界面区域的布局。
- (6) 编写购物车界面区域的样式。

- (7) 编写购物车底部半透明遮罩样式，避免在操作购物车时误触。
- (8) 编写购物车头部区域的样式。
- (9) 编写“清空购物车”按钮的样式。
- (10) 编写购物车中的商品列表样式。
- (11) 编写购物车内的商品名称、价格、购买数量样式。
- (12) 编写减少、增加数量的图标按钮的样式。

#### 知识点 6-【任务 6-6】购物车-实现增加和减少商品数量

教师通过 PPT 结合实际操作的方式讲解增加和减少商品数量的实现。

- (1) 修改增加图标的页面结构。
- (2) 修改减少图标的页面结构。
- (3) 编写 `cartNumberAdd()`事件处理函数，实现购物车里商品数量的增加。
- (4) 编写 `cartNumberDec()`事件处理函数，实现减少购物车里商品数量的功能。

#### 知识点 7-【任务 6-6】购物车-实现清空购物车

教师通过 PPT 结合实际操作的方式讲解清空购物车的实现。

- (1) 在页面结构中，修改清空购物车区域的页面结构，添加 `tap` 事件，事件处理函数为 `cartClear()`。
- (2) 编写 `cartClear()`事件处理函数，实现清空购物车。

#### 四、归纳总结

教师回顾本节课所讲的知识，并通过提问的方式引导学生解答问题并给予指导。

#### 五、布置作业

教师通过学习通或高校教辅平台 (<http://tch.ityxb.com>) 布置本节课作业以及下节课的预习作业。

### 第九课时（上机练习）

上机练习主要针对本章中需要重点掌握的知识点，以及在程序中容易出错的内容进行练习，通过上机练习可以考察同学对知识点的掌握情况，对代码的熟练程度。

**上机：**（考察知识点为：**【任务 6-6】购物车-实现底部购物车区域、【任务 6-6】购物车-实现添加商品到购物车、【任务 6-6】购物车-实现小球动画效果、【任务 6-6】购物车-实现满减优惠信息区域、【任务 6-6】购物车-实现购物车界面区域、【任务 6-6】购物车-实现增加和减少商品数量、【任务 6-6】购物车-实现清空购物车**）

**形式：**独立完成

**题目：**完成**【任务 6-6】购物车**的开发。

**具体要求：**

- (1) 实现底部购物车区域。
- (2) 实现添加商品到购物车。
- (3) 实现小球动画效果。
- (4) 实现满减优惠信息区域。
- (5) 实现购物车界面区域。
- (6) 实现增加和减少商品数量。
- (7) 实现清空购物车。

教学后记	购物车中主要是数据的获取与呈现，实例目前

## 第 7 章 微信小程序开发进阶

课题名称	第 7 章 微信小程序开发进阶	计划课时	6 课时
教学引入	<p>在实际开发中，开发团队往往会在保证项目质量的前提下对项目的开发效率提出要求。为了提高开发效率，开发团队通常会将市面上的一些开源的框架和库引入项目中，并将项目中一些重复使用的功能代码封装成自定义组件。其中，框架可以为开发者提供一套全面的解决方案，可以解决在复杂场景下的开发问题；库是一个封装好的代码集合，提供给开发者使用，让开发者专注于实现功能而隐藏许多技术细节；自定义组件是开发者将页面中可被复用的代码自行封装成的组件。本章将讲解如何创建和使用自定义组件，并通过案例对 Vant Weapp 组件库、WeUI 组件库以及 uni-app 框架的使用进行演示，帮助读者对这些框架和库有更深入的了解。</p>		

<p>教学目标</p>	<ul style="list-style-type: none"> <li>● 使学生掌握自定义组件的创建方法，能够根据实际需要创建自定义组件</li> <li>● 使学生掌握自定义组件的使用方法，能够使用自定义组件</li> <li>● 使学生掌握使用自定义组件渲染标签栏的方法，能够使用自定义组件渲染标签栏</li> <li>● 使学生掌握 Vant Weapp 组件库的使用方法，能够使用 Vant Weapp 组件库快速搭建微信小程序的页面</li> <li>● 使学生掌握 WeUI 组件库的使用方法，能够使用 WeUI 组件库快速搭建微信小程序的页面</li> <li>● 使学生掌握 navigator 组件的使用方法，能够利用 navigator 组件实现页面跳转</li> <li>● 使学生了解 uni-app 框架的概念，能够说出使用 uni-app 框架开发项目的优势</li> <li>● 使学生掌握 HBuilder X 开发工具的设置，能够完成 HBuilder X 的基本设置和个性化设置</li> <li>● 使学生掌握 uni-app 项目的创建方法，能够完成 uni-app 项目的创建</li> <li>● 使学生熟悉 uni-app 项目的目录结构，能够解释各个文件和文件夹的作用</li> <li>● 使学生掌握 uni-app 项目运行至微信小程序的方法，能够将 uni-app 项目运行至微信小程序</li> <li>● 使学生掌握 uni-app 项目的全局配置文件，能够对导航栏、底部标签栏、页面的文件路径等进行配置</li> </ul>
<p>教学重点</p>	<ul style="list-style-type: none"> <li>● 创建自定义组件</li> <li>● 使用自定义组件</li> <li>● WeUI 组件库</li> <li>● navigator 组件</li> <li>● 将 uni-app 项目运行至微信小程序</li> </ul>
<p>教学难点</p>	<ul style="list-style-type: none"> <li>● navigator 组件</li> </ul>
<p>教学方式</p>	<p>课堂教学以 PPT 讲授为主，并结合多媒体进行教学</p>
<p>思政元素</p>	<p>讲解 Vant Weapp、WeUI 组件库及 uni-app 框架时，介绍开源社区的共享理念与协作精神，引导学生尊重开源协议、参与开源贡献，培养学生的共享意识与协作精神，同时强调开源开发中的诚信与版权意识。</p>
<p>教学过程</p>	<p style="text-align: center;"><b>第一、二课时</b></p> <p style="text-align: center;">（创建自定义组件、使用自定义组件、使用自定义组件渲染标签栏、Vant Weapp 组件库、【案例 7-1】自定义标签栏-准备工作、【案例 7-1】自定义标签栏-项目初始化、【案例 7-1】自定义标签栏-定义标签栏数据、【案例 7-1】自定义标签栏-实现页面布局、【案例 7-1】自定义标签栏-实现页面逻辑）</p> <p>一、复习巩固</p> <p>教师通过上节课作业的完成情况，对学生吸收不好的知识点进行再次巩固讲解。</p> <p>二、通过需求引入的方式导入新课</p> <p>通过前面的学习，我们知道微信小程序的底部标签栏可以在 app.json 全局配置文件中通过添加 tabBar 配置项的属性来实现。但是在实际开发中，底部标</p>

签栏会有不同的需求，例如需要设计更美观的样式或者需要添加更多的功能，此时就需要自定义标签栏。本节课将详细讲解“自定义标签栏”微信小程序案例的开发。

### 三、新课讲解

#### 知识点 1-创建自定义组件

教师通过 PPT 结合实际操作的方式讲解自定义组件的创建。

- (1) 为什么要创建自定义组件。
- (2) 自定义组件的文件组成。
- (3) 通过实际操作演示自定义组件的创建步骤。
- (4) 列举 data 属性和 properties 属性的区别。

#### 知识点 2-使用自定义组件

教师通过 PPT 结合实际操作的方式讲解自定义组件的使用。

- (1) 分析自定义组件局部引用和全局引用 2 种方式的适用场景。
- (2) 演示局部引用的使用。
- (3) 演示全局引用的使用。

#### 知识点 3-使用自定义组件渲染标签栏

教师通过 PPT 结合实际操作的方式讲解如何使用自定义组件渲染标签栏。

- (1) 使用自定义组件渲染标签栏的优势。
- (2) 分析自定义组件渲染标签栏的实现思路。
- (3) 通过示例代码演示使用自定义组件渲染标签栏的具体方法。

#### 知识点 4-Vant Weapp 组件库

教师通过 PPT 结合实际操作的方式讲解 Vant Weapp 组件库。

- (1) 什么是 Vant Weapp 组件库。
- (2) 演示 Vant Weapp 组件库在小程序中的使用步骤。
- (3) 通过示例代码演示如何在页面中使用 Vant Weapp 中的 button 组件。

#### 知识点 5-【案例 7-1】自定义标签栏-准备工作

教师通过 PPT 结合实际操作的方式讲解“自定义标签栏”微信小程序的准备工作。

- (1) 创建项目。
- (2) 安装 Vant Weapp 组件库。
- (3) 复制素材。

#### 知识点 6-【案例 7-1】自定义标签栏-项目初始化

教师通过 PPT 结合实际操作的方式讲解“自定义标签栏”微信小程序的初始化。

- (1) 创建自定义组件。
- (2) 编写项目的全局配置文件 app.json。
- (3) 编写“自定义标签栏”微信小程序 home、message、contact 页面的内容。

#### 知识点 7-【案例 7-1】自定义标签栏-定义标签栏数据

教师通过 PPT 结合实际操作的方式讲解标签栏数据的获取。

- (1) 通过代码实现标签栏数据的获取。
- (2) 运行程序，查看标签栏数据是否获取成功。

#### 知识点 8-【案例 7-1】自定义标签栏-实现页面布局

教师通过 PPT 结合实际操作的方式讲解“自定义标签栏”微信小程序页面

布局的实现。

- (1) 在 app.json 文件中引入所需的组件，配置对应的路径。
- (2) 使用 Vant Weapp 组件库中的 tabbar 标签栏组件实现自定义标签栏。

#### 知识点 9-【案例 7-1】自定义标签栏-实现页面逻辑

教师通过 PPT 结合实际操作的方式讲解“自定义标签栏”微信小程序页面逻辑的实现。

- (1) 在 Component()函数中编写 onChange()事件处理函数,实现点击标签项时跳转到对应的页面。
- (2) 在 home.js 文件中编写 onShow()函数,将该页面对应的标签项激活。
- (3) 在 message.js 文件中编写 onShow()函数,将该页面对应的标签项激活。
- (4) 在 contact.js 文件中编写 onShow()函数,将该页面对应的标签项激活

#### 四、归纳总结

教师回顾本节课所讲的知识,并通过测试题的方式引导学生解答问题并给予指导。

#### 五、布置作业

教师通过学习通或高校教辅平台 (<http://tch.itymb.com>) 布置本节课作业以及下节课的预习作业。

### 第三、四课时

(WeUI 组件库、navigator 组件、【案例 7-2】电影列表-准备工作、【案例 7-2】电影列表-实现标签栏、【案例 7-2】电影列表-实现“正在热映”标签页、【案例 7-2】电影列表-实现“搜索”标签页、【案例 7-2】电影列表-实现“分类”标签页)

#### 一、复习巩固

教师通过上节课作业的完成情况,对学生吸收不好的知识点进行再次巩固讲解。

#### 二、通过直接引入的方式导入新课

“电影列表”微信小程序用于展示正在热映的电影。为了提高开发效率,使用 WeUI 组件库来实现。本节课将详细讲解“电影列表”微信小程序的开发。

#### 三、新课讲解

##### 知识点 1-WeUI 组件库

教师通过 PPT 结合实际操作的方式讲解 WeUI 组件库。

- (1) 什么是 WeUI 组件库。
- (2) WeUI 样式样例效果。
- (3) 演示在微信小程序中引入 WeUI 组件库的两种方式。
- (4) 通过案例代码演示 WeUI 组件库中 dialog 弹窗组件的使用。

##### 知识点 2- navigator 组件

教师通过 PPT 结合实际操作的方式讲解 navigator 组件。

- (1) navigator 组件的作用。
- (2) 列举 navigator 组件的常用属性。
- (3) 列举 navigator 组件的 open-type 属性常用的合法值。
- (4) 通过案例演示 navigator 组件的使用。

##### 知识点 3-【案例 7-2】电影列表-准备工作

教师通过 PPT 结合实际操作的方式讲解“电影列表”小程序的准备工作。

- (1) 创建项目。
- (2) 配置页面。
- (3) 配置导航栏。
- (4) 复制素材。
- (5) 通过 useExtendedLib 扩展库的方式引入 WeUI 组件库，在全局配置文件 app.json 文件中添加配置。

#### 知识点 4-【案例 7-2】电影列表-实现标签栏

教师通过 PPT 结合实际操作的方式讲解标签栏的实现。

- (1) 编写实现标签栏所需的数据。
- (2) 编写页面结构。
- (3) 创建 tab1.wxml、tab2.wxml 和 tab3.wxml 文件，并为这些文件添加内容。
- (4) 编写 tabClick()函数，实现切换标签页的功能。

#### 知识点 5-【案例 7-2】电影列表-实现“正在热映”标签页

教师通过 PPT 结合实际操作的方式讲解“正在热映”标签页的实现。

- (1) 编写“正在热映”标签页所需的数据。
- (2) 编写页面结构。

#### 知识点 6-【案例 7-2】电影列表-实现“搜索”标签页

教师通过 PPT 结合实际操作的方式讲解“搜索”标签页的实现。

- (1) 分析“搜索”标签页的实现思路。
- (2) 编写“搜索”标签页所需的数据。
- (3) 编写搜索框的页面结构。
- (5) 编写事件处理函数。
- (6) 编写热搜词的页面结构。

#### 知识点 7-【案例 7-2】电影列表-实现“分类”标签页

教师通过 PPT 结合实际操作的方式讲解“分类”标签页的实现。

- (1) 分析“分类”标签页的实现思路。
- (2) 编写“分类”标签页所需的数据。
- (3) 编写页面结构。

#### 四、归纳总结

教师回顾本节课所讲的知识，并通过测试题的方式引导学生解答问题并给予指导。

#### 五、布置作业

教师通过学习通或高校教辅平台（<http://tch.ityxb.com>）布置本节课作业以及下节课的预习作业。

### 第五课时

（uni-app 框架概述、HBuilder X 开发工具、创建 uni-app 项目、uni-app 项目的目录结构、将 uni-app 项目运行至微信小程序、uni-app 项目的全局配置文件、【案例 7-3】待办事项-准备工作、【案例 7-3】待办事项-实现添加待办事项的功能、【案例 7-3】待办事项-实现列表区域）

#### 一、复习巩固

教师通过上节课作业的完成情况，对学生吸收不好的知识点进行再次巩固讲解。

## 二、通过场景引入的方式导入新课

在日常生活中，人们倾向于提前规划一段时间的生活和工作内容，这样实施起来非常有条理。“待办事项”微信小程序可以帮助用户把要做的事情一项一项列出来，并且整理出待办事项和已办事项，用户可以逐项完成。本节课将详细讲解“待办事项”微信小程序的开发。

## 三、新课讲解

### 知识点 1-uni-app 框架概述

教师通过 PPT 的方式讲解 uni-app 框架。

- (1) 什么是 uni-app 框架。
- (2) 使用 uni-app 框架开发项目的优势。

### 知识点 2-HBuilder X 开发工具

教师通过 PPT 结合实际操作的方式讲解 HBuilder X 开发工具。

- (1) HBuilder X 开发工具的优势。
- (2) 演示 HBuilder X 开发工具的下载和配置步骤。

### 知识点 3-创建 uni-app 项目

教师通过 PPT 结合实际操作的方式讲解如何创建一个 uni-app 项目。

- (1) 演示在 HBuilder X 中创建一个新的 uni-app 项目的步骤。
- (2) 创建成功后，查看新建 uni-app 项目的基本信息界面。

### 知识点 4-uni-app 项目的目录结构

教师通过 PPT 的方式讲解 uni-app 项目的目录结构。

- (1) 讲解 uni-app 项目常用的目录及其作用。
- (2) 讲解 uni-app 项目常用的文件及其作用。

### 知识点 5-将 uni-app 项目运行至微信小程序

教师通过 PPT 结合实际操作的方式讲解如何将 uni-app 项目运行至微信小程序。

- (1) 演示将 uni-app 项目运行至微信小程序的具体步骤。
- (2) 查看初次运行后的结果。

### 知识点 6- uni-app 项目的全局配置文件

教师通过 PPT 结合实际操作的方式讲解 uni-app 项目的全局配置文件。

- (1) 为什么要配置 uni-app 项目的全局配置文件。
- (2) 如何配置 uni-app 项目的全局配置文件。

### 知识点 7-【案例 7-3】待办事项-准备工作

教师通过 PPT 结合实际操作的方式讲解“待办事项”微信小程序的准备工作。

- (1) 创建项目。
- (2) 配置页面和导航栏。
- (3) 将“待办事项”项目运行到微信小程序。

### 知识点 8-【案例 7-3】待办事项-实现添加待办事项的功能

教师通过 PPT 结合实际操作的方式讲解添加待办事项功能的实现。

- (1) 分析添加待办事项功能的实现思路。
- (2) 定义页面所需的数据。
- (3) 编写页面的整体结构。
- (4) 编写页面的业务逻辑。
- (5) 编写输入区域的页面样式。

### 知识点 9-【案例 7-3】待办事项-实现列表区域

教师通过 PPT 结合实际操作的方式讲解列表区域的实现。

- (1) 分析列表区域的实现思路。
- (2) 编写待办事项的页面结构。
- (3) 编写 `doingHandle()` 事件处理函数，实现点击已办事项列表中的复选框时将事项状态设为 `finished`，将事项显示在已办事项列表中。
- (4) 在待办事项下方编写已办事项的页面结构。
- (5) 编写 `doneHandle()` 事件处理函数，实现点击已办事项列表中的复选框时将事项状态设为 `todo`，将事项显示在待办事项列表中。
- (6) 编写列表区域的页面样式。

#### 四、归纳总结

教师回顾本节课所讲的知识，并通过测试题的方式引导学生解答问题并给予指导。

#### 五、布置作业

教师通过学习通或高校教辅平台 (<http://tch.ityxb.com>) 布置本节课作业以及下节课的预习作业。

### 第六课时（上机练习）

上机练习主要针对本章中需要重点掌握的知识点，以及在程序中容易出错的内容进行练习，通过上机练习可以考察同学对知识点的掌握情况，对代码的熟练程度。

**上机：**（考察知识点为：创建自定义组件、使用自定义组件、使用自定义组件渲染标签栏、Vant Weapp 组件库、【案例 7-1】自定义标签栏-准备工作、【案例 7-1】自定义标签栏-项目初始化、【案例 7-1】自定义标签栏-定义标签栏数据、【案例 7-1】自定义标签栏-实现页面布局、【案例 7-1】自定义标签栏-实现页面逻辑）

**形式：**独立完成

**题目：**完成【案例 7-1】自定义标签栏的开发。

**具体要求：**

- (1) 创建项目、安装 Vant Weapp 组件库和复制素材。
- (2) 创建自定义组件、编写 `app.json` 全局配置文件、编写 `home`、`message`、`contact` 页面中的内容。
- (3) 定义标签栏数据。
- (4) 实现页面布局。
- (5) 实现页面逻辑。

**上机：**（考察知识点为：WeUI 组件库、navigator 组件、【案例 7-2】电影列表-准备工作、【案例 7-2】电影列表-实现标签栏、【案例 7-2】电影列表-实现“正在热映”标签页、【案例 7-2】电影列表-实现“搜索”标签页、【案例 7-2】电影列表-实现“分类”标签页）

**形式：**独立完成

**题目：**完成【案例 7-2】电影列表。

**具体要求：**

- (1) 创建项目、配置页面、配置导航栏、复制素材和引入 WeUI 组件库。
- (2) 实现标签栏。

	<p>(3) 实现“正在热映”标签页。</p> <p>(4) 实现“搜索”标签页。</p> <p>(5) 实现“分类”标签页。</p>
教学后记	

## 第 8 章 微信小程序云开发

课题名称	第 8 章 微信小程序云开发	计划课时	6 课时
教学引入	个人的小程序开发者可以选择小程序云开发；人员配备不齐，没有云服务器的中小企业，也可以选择小程序云开发；人员配备齐全、拥有自己的云服务器的企业，可以选择小程序传统开发，这样更方便自己来控制。		
教学目标	<p>(1) 了解什么是云开发及云开发和传统开发的区别。</p> <p>(2) 掌握云开发控制台的使用。</p> <p>(3) 掌握云开发的技术，包括云函数、文件存储、数据库的开发和使用。</p>		
教学重点	创建小程序云开发项目、云开发控制台的使用、云函数、云开发文件存储、云开发数据库		
教学难点	云开发控制台的使用、云函数		
教学方式	课堂教学以 PPT 讲授为主，并结合多媒体进行教学		
思政元素	<p>高效开发与社会责任：对比云开发与传统开发的效率优势，强调云开发降低中小企业开发成本的作用，引导学生认识到技术应服务于社会（如助力中小企业数字化转型），培养学生用技术服务社会的责任意识。</p>		

## 第一至四课时

### 8.1 云开发和传统开发的对比

(1) 在效率上，云开发只需关心自身业务逻辑实现，聚焦于小程序开发，效率更高；而传统开发既需要关注小程序开发，又需要提供小程序业务相关接口，效率难以提升。

(2) 在成本上，云开发提供免费的配额，如需要提升配额，可按需付费；而传统开发需要自己购置服务器，自己运维部署。

(3) 在生态上，云开发与微信使用私有协议天然鉴权，原生集成微信 SDK；而传统开发不具备这样的优势，需要自己实现产品相关逻辑。

(4) 在运维上，云开发底层上由腾讯云提供专业支持；而传统开发需要自行维护系统，学习成本高，运维难度大。

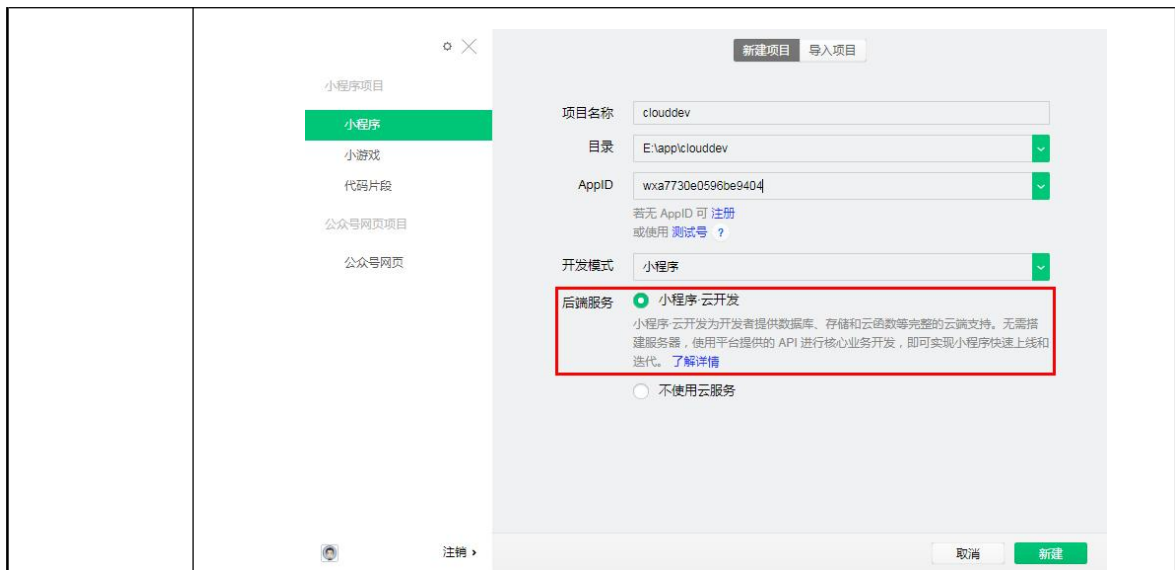
(5) 在开发部署速度上，云开发可以快速开发、快速发布；而传统开发需要小程序开发者和后端开发者进行前后端联调，然后才能发布上线。

个人的小程序开发者可以选择小程序云开发；人员配备不齐，没有云服务器的中小企业，也可以选择小程序云开发；人员配备齐全、拥有自己的云服务器的企业，可以选择小程序传统开发，这样更方便自己来控制。

### 8.2 创建小程序云开发项目

下面我们一起创建一个小程序云开发项目。

(1) 创建一个“clouddev”云开发项目，AppId 不能使用测试号，如图所示。



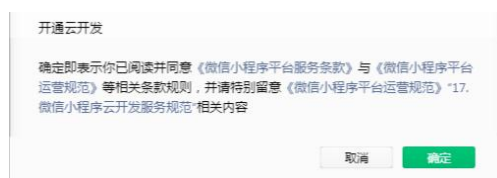
### 创建云开发项目

(2) 创建“clouddev”云开发项目后，在工具栏区域单击“云开发操作”按钮，进入云控制台界面，会看到云函数、数据库、文件存储、数据分析的介绍，如图所示。



### 开通云开发

单击“开通”按钮，会弹出开通云开发的相关协议，如图所示。单击“确定”按钮开通云开发。



### 云开发协议

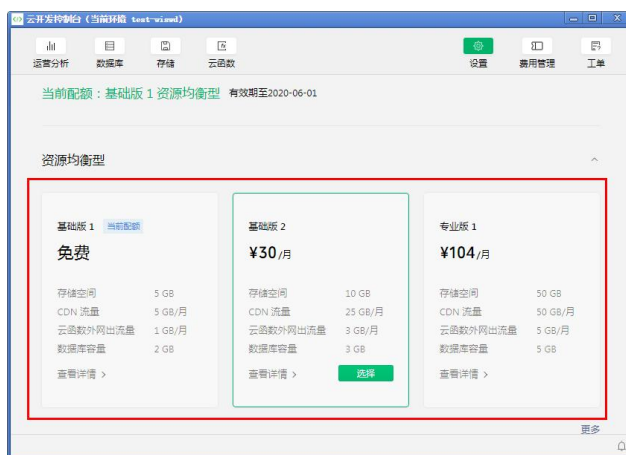
(3) 开通云开发之后，就可以开始创建环境了。首先填写环境名称。环境名称只能包含数字、小写字母和-；只能以小写字母开头，最多 20 字符。比如

可以创建一个测试环境“test”、线上环境“release”。然后填写环境 ID，环境 ID 只能包含数字、小写字母和-；只能以小写字母开头，最多 27 字符。环境 ID 是在使用云服务时需要用到的全局唯一标识符，一经创建便不可修改。下面我们添加一个测试环境“test”，环境 ID 填写为“test-wiswd”，如图所示。



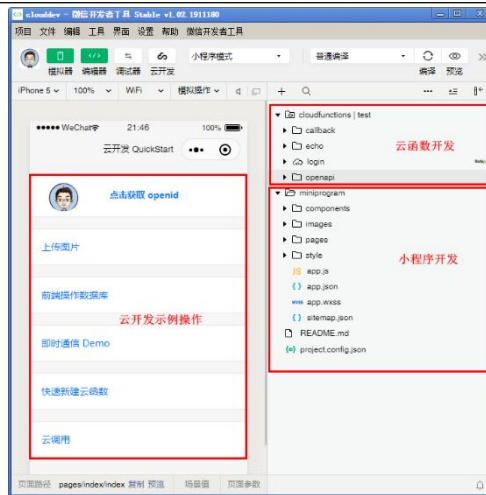
### 创建测试环境“test”

(4) 环境配额小程序默认提供给开发者的是免费版本的基础配额，对于用于开发和业务量不多的开发者，免费版本的基础配额就够使用。也可以付费升级配额，如图所示。



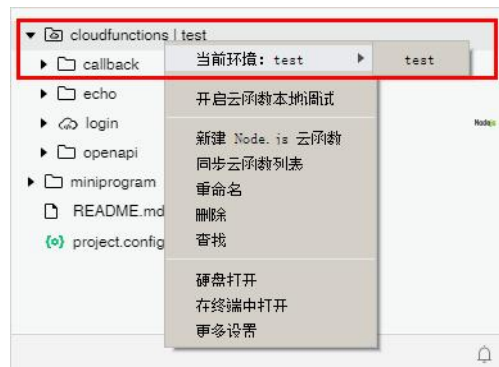
### 升级配额

(5) 云开发开通之后，回到开发界面，可以看到云开发示例操作、云函数开发、小程序开发区域，如图所示。



云开发界面

(6) 云开发可以配置多个环境，在 cloudfunctions 文件夹上的右键下拉选项列表里选择环境 test，如图所示。



选择环境

## 云开发控制台的使用

云开发控制台包含 7 个功能：运营分析、数据库、存储、云函数、设置、费用管理和工单。

### 8.3 运营分析

运营分析包含以下 3 方面内容。

(1) 资源使用：可以查看数据库容量、本日数据库读请求数、存储容量、本月 CDN 流量、本月云函数资源使用量。

(2) 用户访问：可以查看今日活跃用户、本周活跃用户、本月活跃用户，同时可以查看访问用户列表。当用户授权后，可以查看用户头像。

(3) 监控图表：可以查看数据库监控，包含数据库读取次数、数据库写入次数；可以查看存储监控，包含存储上传调用次数、存储下载调用次数、CDN 流量、CDN 回源流量。

## 8.4 数据库

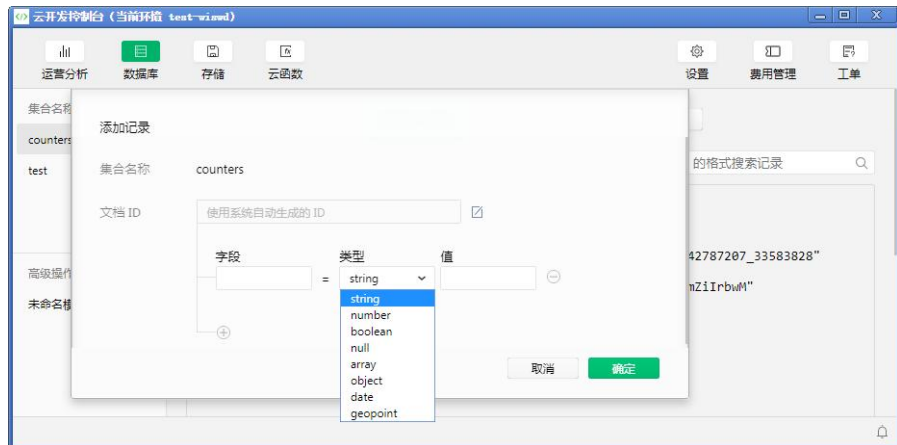
微信小程序云开发提供了一个 JSON 数据库，数据库中的每条记录都是一个 JSON 格式的对象。一个数据库可以有多个集合（相当于关系型数据中的表），一个集合可看作一个 JSON 数组，数组中的每个对象就是一条记录，记录的格式是 JSON 对象。JSON 数据库如图所示。



JSON 数据库

(1) 单击“集合名称”区域的+号可以创建新的集合。输入集合的名称。集合名称支持大小写英文字母、数字、-和\_。

(2) 添加集合记录。首先要选中需要添加记录的集合，然后单击“添加记录”按钮。文档 ID 可以自动生成，也可以自己填写。再输入需要添加的字段，比如学生的姓名，可以填写 name，字段类型有 string、number、boolean、null、array、object、date、geopoint。如图所示。



添加集合记录

(3) 数据库支持导入集合和导出集合，导出集合支持 JSON 格式和 CSV 格式。

(4) 索引管理可以用来提高数据查询效率。按业务查询的字段可以创建唯一索引和非唯一索引，同时可以指定升序、降序、地理位置，如图所示。



索引管理

(5) 数据库权限设置分为 4 种：所有用户可读，仅创建者可读写；仅创建者可读写；所有用户可读；所有用户不可读写。如图所示。



数据库权限设置

第 5-8 课时

## 8.5 存储

在云开发控制台的“存储”界面可以查看已上传的文件，可以进行创建文件夹、上传文件、删除文件等操作，如图所示。



文件存储管理

针对文件可以进行权限管理，分为 4 种：所有用户可读，仅创建者可读写；

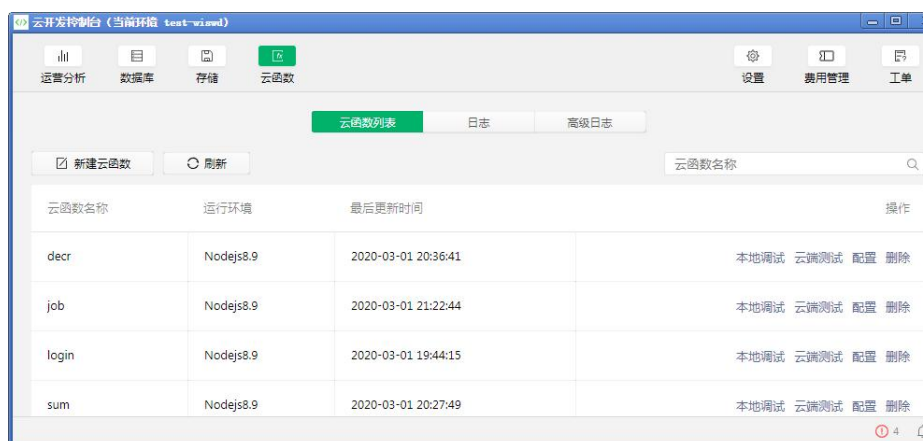
仅创建者可读写；所有用户可读；所有用户不可读写。如图所示。



文件存储权限管理

## 8.6 云函数

在云开发控制台的“云函数”界面中可以查看云函数列表，如图所示。小程序创建的云函数，可以上传部署到云端，然后在云函数列表里显示。同时可以在云开发控制台创建云函数，进行本地测试、云端测试、配置、删除等操作。

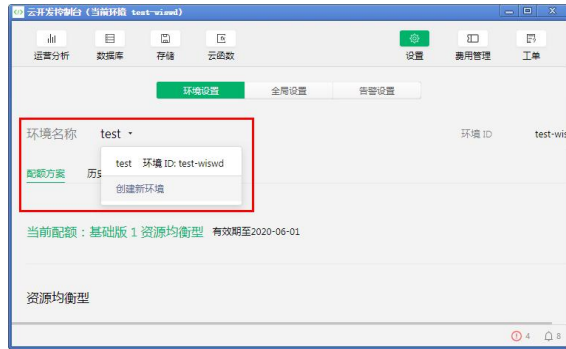


云函数列表

在日志里可以查看云函数调用情况，方便调试和使用。

## 8.7 设置

在“环境设置”里可以创建新环境、查看已有的环境；“全局设置”里可以设置消息推送；“告警设置”里可以加入告警群。如图所示。



设置



## 8.8 费用和工单

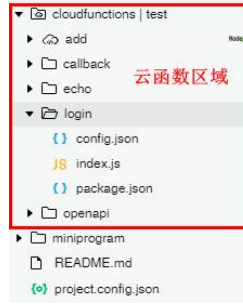
在“费用管理”中可以查看发票、申请开具发票及申领代金券。“工单”是用来处理云开发相关的资源故障和费用相关问题的。

## 云函数

云函数是部署在云端（服务器端）运行的函数。云函数的写法与一个在本地定义的 JavaScript 方法无异。云函数代码运行在云端 Node.js 中。开发者无须购买、搭建服务器，只需编写函数代码并部署到云端即可在小程序端调用。同时云函数之间也可互相调用。云函数的独特优势在于与微信登录鉴权的无缝整合，当小程序端调用云函数时，云函数的传入参数中会被注入小程序端用户的 openid，开发者无须校验 openid 的正确性，因为微信已经完成了这部分鉴权，开发者可以直接使用该 openid。

## 8.9 云函数介绍

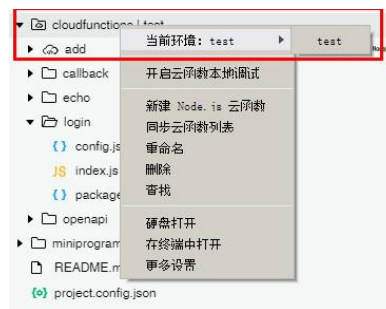
小程序使用云开发创建项目后，会生成云函数目录区域，可以看到 cloudfunctions 是云函数的根目录。在项目根目录 project.config.json 文件中配置了 cloudfunctionRoot 字段指向的文件目录，指向哪个文件目录，该文件目录就作为云函数的本地根目录。云函数的根目录的图标会变成云目录图标 。云函数根目录下的第一级目录名称（云函数目录）与云函数名字相同。如果对应的线上环境存在该云函数，则会用一个特殊的云图标  标明。如图所示。



### 云函数区域

在云函数根目录或者云函数目录上，通过鼠标右键菜单可以完成以下操作。

(1) 查看当前环境，切换环境。当选定环境之后，云函数的操作都是基于此环境进行。如图所示。



### 云函数环境

(2) 新建 Node.js 云函数。开发者工具在本地创建云函数目录和两个文件 (index.js、package.json)，同时在线上环境中创建出对应的云函数。

- 云函数目录：以云函数名字命名的目录，存放该云函数的所有代码。
- index.js 文件：云函数入口文件。云函数被调用时实际执行的入口函数是 index.js 文件中导出的 main 方法，默认会生成模板函数。代码如下。

// 云函数入口文件，wx-server-sdk 是微信提供的云函数操作数据库、存储及调用其他云函数用的库

```
const cloud = require('wx-server-sdk')
```

```
// 初始化云函数 sdk
```

```
cloud.init()
```

```
// 云函数入口函数，传入参数 event 对象和 context 对象
```

```
// event 是小程序端调用云函数传入的参数，及自动注入的小程序用户的 openid 和小程序的 appid
```

```
// context 对象包含了此处调用的调用信息和运行状态
```

```
exports.main = async (event, context) => {
```

```
  // 调用云函数库，获得 wxContext 对象，该对象包含 OPENID、APPID、UNIONID
```

```
  const wxContext = cloud.getWXContext()
```

```
    return {
      event,
      openid: wxContext.OPENID,
      appid: wxContext.APPID,
      unionid: wxContext.UNIONID,
    }
  }
}
```

• package.json 文件: npm 包定义文件, 其中默认定义了最新的 wx-server-sdk 依赖。在创建成功后, 工具会提示是否为该云函数立即安装本地依赖, 即 wx-server-sdk, 如是, 则工具会开启终端执行 npm install。

(3) 同步云函数列表。将线上环境中的云函数列表同步到本地。开发者工具会根据云函数的名字, 在本地创建出对应的云函数目录。

(4) 下载云函数。在一个云函数目录上单击鼠标右键可以在弹出的快捷菜单中选择下载该云函数, 云函数代码会被下载到指定目录。

(5) 上传并部署。在云函数目录上单击鼠标右键, 可以将云函数整体打包上传并部署到线上环境中。有两种方式: 云端安装依赖(不上传 node modules) 和所有文件(上传 node\_modules 文件夹)。

## 8.10 云函数的使用

下面我们来创建一个云函数, 用于两个数相加, 并且在小程序端进行调用。

(1) 在云函数根目录上单击鼠标右键, 在弹出的快捷菜单中选择创建一个新的 Node.js 云函数, 将该云函数命名为“add”。修改 add 云函数目录下的 index.js 文件。代码如下。

```
// 云函数入口文件
const cloud = require('wx-server-sdk')

cloud.init()

// 云函数入口函数
exports.main = async (event, context) => {
  return {
    sum: event.a + event.b
  }
}
```

(2) 在小程序端业务逻辑处理文件里, 调用云函数 add。需要使用 wx.cloud.callFunction 来调用云函数。示例代码如下。

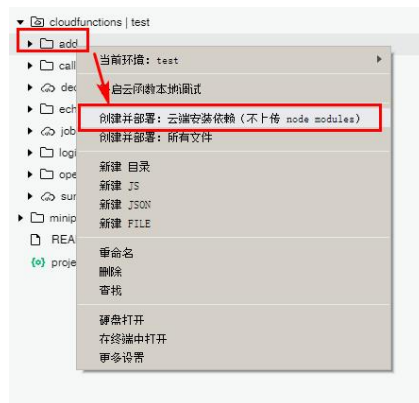
```
Page({
```

```

onLoad:function(){
  wx.cloud.callFunction({
    // 云函数名称
    name: 'add',
    // 传给云函数的参数
    data: {
      a: 1,
      b: 2,
    },
    success: function (res) {
      console.log(res)
      console.log(res.result.sum)
    },
    fail: function(res){
      console.log(res)
      console.error
    }
  })
}
})

```

(3) 调用云函数前，需要先将 add 云函数部署到云端。在 add 云函数目录上单击鼠标右键，选择对应的选项，如图所示，将云函数整体打包上传并部署到线上环境中。



创建并部署

(4) 云函数的调用支持 Promise 风格返回值形式。示例代码如下。

```

Page({
  onLoad: function () {
    wx.cloud.callFunction({

```

```

// 云函数名称
name: 'add',
// 传给云函数的参数
data: {
  a: 1,
  b: 2,
}
}).then(res => {
  console.log(res.result.sum)
}).catch(console.error)
}
})

```

### 8.11 Promise 风格

Promise 对象用于异步操作，它表示一个尚未完成且预计在未来完成的异步操作。Promise 有 3 个状态：pending（未完成），初始状态；fulfilled（成功），操作成功；rejected（失败），操作失败。当 Promise 状态发生改变，就会触发 then() 里的响应函数处理后续步骤。Promise 状态一经改变，不会再变。

Promise 可以用于在云函数中处理一些异步操作，在异步操作完成后再返回结果给调用方。

Promise 对象常用方法如下。

(1) then()。语法：Promise.prototype.then ( onFulfilled , onRejected) 。.then() 接收两个函数作为参数，分别代表 fulfilled（成功）和 rejected（失败）。.then() 返回一个新的 Promise 实例，所以它可以链式调用。如果返回新的 promise，那么下一级 .then() 会在新的 promise 状态改变之后执行；如果返回其他任何值，则会立即执行下一级 .then()。

(2) catch()。语法：Promise.prototype.catch ( onRejected) 。.catch() 用于指定发生错误时的回调函数，自动捕获内部异常，并交给 rejected 响应函数处理。示例代码如下。

```

promise.then(function(data) {
  console.log('success');
}).catch(function(error) {
  console.log('error', error);
});

/*****等同于*****/
promise.then(function(data) {
  console.log('success');

```

```
}).then(undefined, function(error) {  
    console.log('error', error);  
});
```

Promise 对象的错误会一直向后传递,直到被捕获,即错误总会被下一个 catch 所捕获。then 方法指定的回调函数若抛出错误,也会被下一个 catch 捕获。catch 中也能抛错,这样就需要后面的 catch 来捕获。

(3) .resolve()。语法: Promise.resolve (value)。.resolve() 的作用是将 Promise 对象的状态从“未完成”变为“成功”(即从 pending 变为 fulfilled)。在异步操作成功时调用,并将异步操作的结果作为参数传递出去。示例代码如下。

```
Promise.resolve('Success');  
  
/*****等同于*****/  
new Promise(function (resolve) {  
    resolve('Success');  
});
```

(4) .reject()。语法: Promise.reject (reason)。reject 的作用是将 Promise 对象的状态从“未完成”变为“失败”(即从 pending 变为 rejected)。在异步操作失败时调用,并将异步操作报出的错误作为参数传递出去。示例代码如下。

```
Promise.reject(new Error('error'));  
  
/*****等同于*****/  
new Promise(function (resolve, reject) {  
    reject(new Error('error'));  
});
```

Promise 对象可以用来返回异步结果。

(1) 在云函数根目录下单击鼠标右键,选择创建一个新的 Node.js 云函数,将该云函数命名为“sum”。修改 sum 云函数目录下的 index.js 文件。代码如下。

```
// 云函数入口文件  
const cloud = require('wx-server-sdk')  
  
cloud.init()  
  
// 云函数入口函数  
exports.main = async (event, context) => {  
    return new Promise((resolve, reject) => {
```

```
// 在 3 秒后返回结果给调用方（小程序/其他云函数）
setTimeout(() => {
  resolve(event.a + event.b)
}, 3000)
})
}
```

（2）在小程序端业务逻辑处理文件里调用云函数 sum，需要使用 wx.cloud.callFunction 来调用云函数。示例代码如下。

```
Page({
  onLoad:function(){
    wx.cloud.callFunction({
      // 云函数名称
      name: 'sum',
      // 传给云函数的参数
      data: {
        a: 1,
        b: 2,
      }
    }).then(res => {
      console.log(res.result.sum)
    }).catch(console.error)
  }
})
```

## 8.12 npm 的安装与使用

在云函数中可以引入第三方依赖库来进行更快的开发。由于云函数的运行环境是 Node.js，所以可以使用 npm 安装第三方依赖。npm 是 Node.js 的包管理工具（package manager）。

下面我们一起安装一下 npm，并且使用 npm 引入第三方依赖库。

（1）在 Node.js 官方网站下载相应版本的 Node 的镜像或者安装包，比如下载 Windows 64 位版本镜像 node-v12.16.1-x64.msi，然后一步一步安装。在 Windows 上安装时务必选择全部组件，包括勾选“Add to Path”。

（2）在 Windows 环境下，打开命令提示符，然后输入 node -v 命令，如果安装正常，可以看到 Node 的安装版本。Node 安装完成后，npm 也顺带安装完成，可以通过 npm -v 命令来查看版本。如图所示。



查看 npm 的版本

(3) 在小程序根目录 (miniprogram) 打开命令符窗口, 使用 npm 来安装一个第三方 vantui 库。执行如下命令。

**npm install vant-ui-weapp -S --production**

(4) 使用 npm 安装完 vantui 第三方库后, 在命令符窗口进行初始化操作。执行如下命令。

**npm init -y**

(5) 在“开发者工具”-“详情”-“本地设置”里, 勾选“使用 npm 模块”, 如图所示。



勾选“使用 npm 模块”

(6) 在“开发者工具”-“工具”下拉选项里单击“构建 npm”。构建完成后根目录 (miniprogram) 下会显示 miniprogram\_npm 文件夹, 同时可以看到 vantui 库 vant-weapp。

(7) 引入 vantui 库之后, 可以在小程序页面上使用这个 vantui 库。在 .JSON 文件中引用需要使用的组件如下。

```
{
  "usingComponents": {
    "van-button": "vant-weapp/button"
  }
}
```

在 .WXML 文件中直接使用组件如下。

```
<van-button type="primary">主要按钮</van-button>
```

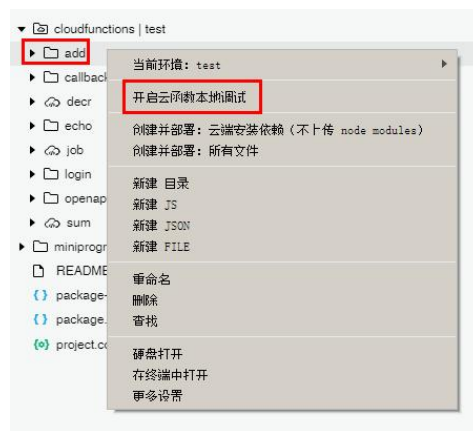
### 8.13 云函数本地调试

在本地安装 Node.js 环境之后，就可以进行云函数本地调试了。它是与线上一致的 Node.js 云函数运行环境，开发者可以在本地对云函数调试，提高开发、调试效率。

本地调试可以对云函数 Node.js 实例进行单步调试/断点调试。在模拟器中对小程序发起的交互点击等操作，如果触发了开启了本地调试的云函数，就会请求到本地实例而不是云端。

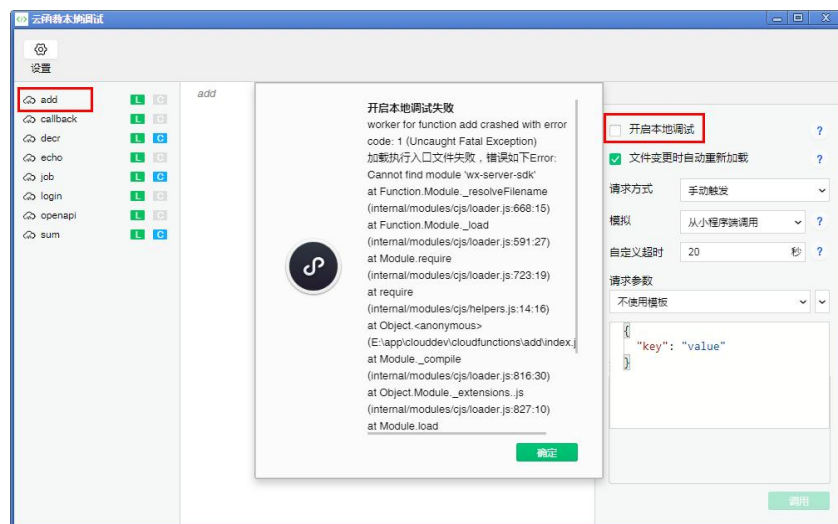
下面我们一起来使用云函数本地调试功能。

(1) 在云函数上单击鼠标右键，在弹出的快捷菜单里单击“开启云函数本地调试”命令，如图所示。



选择“开启云函数本地调试”命令

(2) 在“云函数本地调试”窗口里，单击“add”云函数，接着勾选“开启本地调试”复选框。如果没有安装 node modules，可以忽略并继续执行。有可能会碰到开启本地调试失败的错误，如图所示。



开启本地调试失败

解决办法：在“add”云函数目录下，打开命令符操作窗口，执行如下命令。

**npm install --save wx-server-sdk@latest**

在使用 npminstall 安装第三方依赖包时，会碰到类似如下网络错误。

**npminstall this is a problem related to network connectivity**

解决办法：将代理设置为空，然后重新设置。

**npm config get proxy**

**npm config get https-proxy**

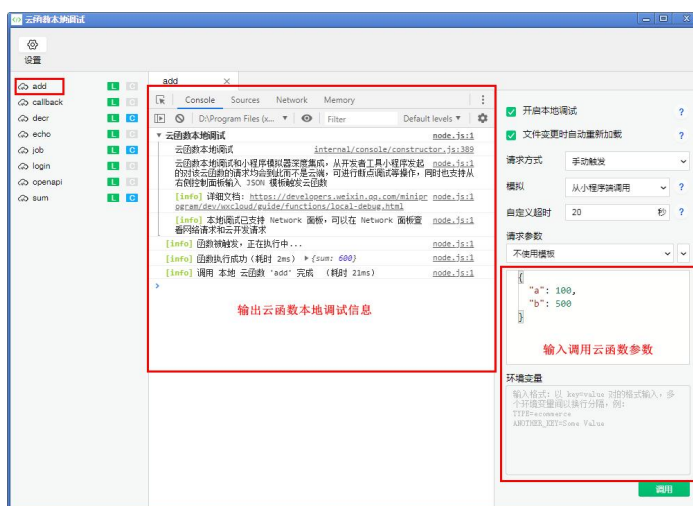
如果返回值不为 null，将其设置为 null：

**npm config set proxy null**

**npm config set https-proxy null**

**npm config set registry http://registry.cnpmjs.org/**

(3) 安装 npminstall --save wx-server-sdk 成功后，单击需要调试的云函数 add，接着勾选“开启本地调试”复选框，会看到输入云函数本地调试信息和输入调用云函数参数区域，如图所示。



“云函数本地调试”窗口

## 8.14 云函数定时器

云函数提供了定时器配置。如果云函数需要定时执行，需要在云函数目录下新建文件 config.json，格式如下。

```
{  
  // triggers 字段是触发器数组，目前仅支持一个触发器，即数组  
  // 只能填写一个，不可添加多个  
  "triggers": [  
    {  
      // name 为触发器的名字  
      "name": "myTrigger",  
      // type: 触发器类型，目前仅支持定时触发器 timer
```

```

        "type": "timer",
        // config 为触发器配置。在定时触发器下, config 格式为 cron
        表达式
        "config": "0 0 2 1 * * *"
    }
}
}

```

(1) 定时触发器名称 (name)：最大支持 60 个字符，支持 a~z、A~Z、0~9、-和\_。必须以字母开头，且一个函数下不支持同名的多个定时触发器。

(2) 定时触发器触发周期 (config)：指定的函数触发时间。用户需填写自定义标准的 cron 表达式来决定何时触发函数。cron 表达式有 7 个必需字段，按空格分隔，第 1 个字段（秒：0~59 的整数）、第 2 个字段（分钟：0~59 的整数）、第 3 个字段（小时：0~23 的整数）、第 4 个字段（日：1~31 的整数）、第 5 个字段（月：1~12 的整数）、第 6 个字段（星期：0~6 的整数）、第 7 个字段（年：1970~2099 的整数）。

(3) cron 表达式同时支持通配符配置。

- , (逗号)：代表取用逗号隔开的字符的并集。例如：在“小时”字段中 1,2,3 表示 1 点、2 点和 3 点。

- - (破折号)：包含指定范围的所有值。例如：在“日”字段中，1-15 包含指定月份的 1 号到 15 号。

- \* (星号)：表示所有值。在“小时”字段中，\*表示每个小时。

- / (正斜杠)：指定增量。在“分钟”字段中，输入 1/10 以指定从第 1 分钟开始，每隔 10 分钟重复。例如，第 11 分钟、第 21 分钟和第 31 分钟重复。

(4) 示例配置。

\* / 5 \* \* \* \* \* \* : 表示每 5s 触发一次。

0 0 2 1 \* \* \* : 表示在每月的 1 日的凌晨 2 点触发。

0 0 10, 14, 16 \* \* \* \* : 表示在每天上午 10 点、下午 2 点、4 点触发。

0 \* / 30 9-17 \* \* \* \* : 表示在每天上午 9 点到下午 5 点内每半小时触发。

## 云开发文件存储

云开发提供文件存储功能，提供了以下文件存储 API：  
 wx.cloud.uploadFile 文件上传 API、wx.cloud.downloadFile 文件下载 API、  
 wx.cloud.deleteFile 文件删除 API、wx.cloud.getTemp FileURL 获取文件临时路径 API。

### 8.15 uploadFile 文件上传 API

uploadFile 将本地资源上传至云存储空间，如果上传至同一路径则是覆盖写。参数 cloudPath，为云存储路径，必传参数；参数 filePath，为要上传文件资源的路径，必传参数；参数 config，为配置 env 使用的环境 ID，填写后忽

略 init 指定的环境，非必传参数。文件上传后可以在云开发控制台存储模块里查看上传的文件。

示例代码如下。

```
Page({
  onLoad:function(){
    // 选择图片
    wx.chooseImage({
      count: 1,
      sizeType: ['compressed'],
      sourceType: ['album', 'camera'],
      success: function (res) {
        const filePath = res.tempFilePaths[0]
        // 上传图片
        const cloudPath = 'my-image' +
filePath.match(/\.[^.]*/)[0]
        wx.cloud.uploadFile({
          cloudPath,
          filePath,
          success: res => {
            console.log('[上传文件] 成功: ', res)
            app.globalData.fileID = res.fileID
            app.globalData.cloudPath = cloudPath
            app.globalData.imagePath = filePath
          },
          fail: e => {
            console.error('[上传文件] 失败: ', e)
          }
        })
      },
      fail: e => {
        console.error(e)
      }
    })
  }
})
```

小程序端调用文件上传时使用 wx.cloud.uploadFile，云函数端调用文件

上传时使用 `cloud.uploadFile`。

### 8.16 downloadFile 文件下载 API

`downloadFile` 从云存储空间下载文件。参数 `fileID`，为云文件 ID，必传参数；参数 `config`，为配置 `env` 使用的环境 ID，填写后忽略 `init` 指定的环境，非必传参数。

示例代码如下。

```
Page({
  onLoad:function(){
    wx.cloud.downloadFile({
      fileID:
      'cloud://test-wiswd.7465-test-wiswd-1253222435/my-image.jpg',
      success: res => {
        console.log(res.tempFilePath)
      },
      fail: err => {
        console.log(err)
      }
    })
  }
})
```

小程序端调用文件下载时使用 `wx.cloud.downloadFile`；云函数端调用文件下载时使用 `cloud.downloadFile`。

### 8.17 deleteFile 文件删除 API

`deleteFile` 从云存储空间删除文件，一次最多删除 50 个。参数 `fileList`: `string[]` 表示云文件 ID 字符串数组。

示例代码如下。

```
Page({
  onLoad:function(){
    wx.cloud.deleteFile({
      fileList:
      ['cloud://test-wiswd.7465-test-wiswd-1253222435/my-image.jpg'],
      success: res => {
        console.log(res.fileList)
      },
    })
  }
})
```

```
        fail: err => {
            console.log(err)
        }
    })
}
```

小程序端调用文件删除时使用 `wx.cloud.deleteFile`；云函数端调用文件删除时使用 `cloud.deleteFile`。

### 8.18 getTempFileURL 获取文件临时路径 API

`getTempFileURL` 用来获取文件临时路径，用云文件 ID 换取真实链接。公有的文件获取的链接不会过期，私有的文件获取的链接有 10 分钟有效期。一次最多获取 50 个文件链接。参数 `fileList: string[]` 表示云文件 ID 字符串数组。示例代码如下。

```
Page({
  onLoad:function(){
    wx.cloud.getTempFileURL({
      fileList:
        ['cloud://test-wiswd.7465-test-wiswd-1253222435/my-image.jpg'],
      success: res => {
        console.log(res.fileList)
      },
      fail: err => {
        console.log(err)
      }
    })
  }
})
```

小程序端获取文件临时路径时使用 `wx.cloud.getTempFileURL`；云函数端获取文件临时路径时使用 `cloud.getTempFileURL`。

### 8.19 云开发数据库

云开发数据库用来存储数据，它可以实现数据库的增、删、改、查操作，满足小程序对数据的操作，方便用户的使用。

## 8.20 数据库介绍

JSON 数据库提供了数据库 database、集合 collection（相当于表 table）、记录 record / doc（相当于行 row）、字段 field（相当于表列 column）。

云开发数据库提供了数据类型：String（字符串）、Number（数字）、Object（对象）、Array（数组）、Bool（布尔值）、Date（时间，精确到毫秒）、Geo（多种地理位置类型）、Null（占位符，字段存在但是值为空）。

示例代码如下。

```
[
  {
    "_id": "Wzh76lk5_O_dt0vO",
    "title": "微信小程序图解案例教程",
    "name": "小刚",
    "characters": [
      "组件",
      "API",
      "案例"
    ],
    "publishInfo": {
      "year": 2020,
      "country": "中国"
    }
  },
  {
    "_id": "Wzia0lk5_O_dt0vR",
    "_openid": "ohl4L0Rnhq7vmmbT_DaNQa4cPaz0",
    "title": "微信小程序开发全案精讲",
    "author": "小刚",
    "characters": [
      "仿酷狗音乐小程序",
      "仿京东小程序",
      "仿猫眼电影小程序"
    ],
    "publishInfo": {
      "year": 2019,
      "country": "中国"
    }
  }
]
```

```
    }  
  }  
}
```

## 8.21 数据库初始化

使用数据库 API 进行数据操作前，需要先获取数据库的实例引用。小程序可以调用获取默认环境的数据库的引用。

```
const db = wx.cloud.database()
```

小程序也可以指定环境来创建数据库的引用，通过 env 字段指定要使用的环境。以下代码假设环境名为“test”，用作测试环境。

```
const db = wx.cloud.database({  
  env: 'test'  
})
```

## 8.22 数据新增操作

通过调用 add 方法，可以向集合 counters 中插入一条数据。集合 counters 需要在云开发控制台里创建。示例代码如下。

```
Page({  
  onAdd: function () {  
    const db = wx.cloud.database()  
    db.collection('counters').add({  
      data: {  
        count: 1  
      },  
      success: res => {  
        wx.showToast({  
          title: '新增记录成功',  
        })  
        console.log('[数据库] [新增记录] 成功')  
      },  
      fail: err => {  
        wx.showToast({  
          icon: 'none',  
          title: '新增记录失败'  
        })  
      }  
    })  
  }  
})
```

```
        console.error('[数据库] [新增记录] 失败: ', err)
    }
    })
}
})
```

### 8.23 数据查询操作

通过调用 `get` 方法，可以获取集合 `counters` 中的一条数据或者多条数据。示例代码如下。

```
Page({
  onQuery: function () {
    const db = wx.cloud.database()
    // 查询当前用户所有的 counters
    db.collection('counters').get({
      success: res => {
        console.log('[数据库] [查询记录] 成功: ', res)
      },
      fail: err => {
        wx.showToast({
          icon: 'none',
          title: '查询记录失败'
        })
        console.error('[数据库] [查询记录] 失败: ', err)
      }
    })
  }
})
```

查询数据时，可以通过调用 `where` 方法设定条件来进行查询。示例代码如下。

```
Page({
  onQuery: function () {
    const db = wx.cloud.database()
    // 查询当前用户所有的 counters
    db.collection('counters').where({
      name: '小刚'
    })
  }
})
```

```

    }).get({
      success: res => {
        console.log('[数据库] [查询记录] 成功: ', res)
      },
      fail: err => {
        console.error('[数据库] [查询记录] 失败: ', err)
      }
    })
  }
})

```

查询数据时，可通过调用 `orderBy` 方法进行排序。`orderBy` 方法接收一个必填字符串参数 `fieldName` 用于定义需要排序的字段，参数值只能取 `asc` 或 `desc`。可通过调用 `field` 方法指定返回结果中记录需返回的字段。通过调用 `skip` 方法指定查询返回结果时从指定序列后的结果开始返回。通过调用 `limit` 方法指定查询结果集数量上限。通过组合使用 `skip` 和 `limit` 方法来实现分页功能，`skip` 是指定那条数据开始查询，`limit` 是指定查询条数。示例代码如下。

```

Page({
  onQuery: function () {
    const db = wx.cloud.database()
    //分页查询用户数据
    db.collection('counters').field({
      description: true,
      done: true,
      progress: true,
    }).where({
      name: '小刚'
    }).orderBy('id', 'asc').skip(10).limit(20).get({
      success: res => {
        console.log('[数据库] [查询记录] 成功: ', res)
      },
      fail: err => {
        console.error('[数据库] [查询记录] 失败: ', err)
      }
    })
  }
})

```

## 8.24 数据修改操作

数据修改操作有如下两种方式。

(1) 通过调用 update 方法更新指定字段的数据值。示例代码如下。

```
Page({
  onUpdate: function () {
    const db = wx.cloud.database()
    db.collection('counters').doc('10').update({
      data: {
        count: 2
      },
      success: res => {
        console.log('[数据库] [更新记录] 成功: ', res)
      },
      fail: err => {
        icon: 'none',
        console.error('[数据库] [更新记录] 失败: ', err)
      }
    })
  }
})
```

(2) 通过调用 set 方法更新整条记录的数据值。示例代码如下。

```
Page({
  onUpdate: function () {
    const db = wx.cloud.database()
    db.collection('counters').doc('10').set({
      data: {
        description: "learn cloud database",
        due: new Date("2018-09-01"),
        tags: [
          "cloud",
          "database"
        ],
        style: {
          color: "skyblue"
        }
      }
    })
  }
})
```

```

    },
    location: new db.Geo.Point(113, 23),
    done: false
  }
})
}
})

```

调用 doc 方法是用来获取集合中指定记录的引用。doc 方法接收一个 id 参数，指定需引用的记录的 \_id，根据这个值找到这条记录，然后进行更新操作。

## 8.25 数据删除操作

通过数据删除操作删除一条数据记录的示例代码如下。

```

Page({
  onDelete: function () {
    const db = wx.cloud.database()
    db.collection('counters').doc('10').remove({
      success: function (res) {
        console.log(res.data)
      }
    })
  }
})

```

通过数据删除操作删除多条数据记录的示例代码如下。

```

Page({
  onDelete: function () {
    const db = wx.cloud.database()
    db.collection('counters').where({
      done: true
    }).remove({
      success: function (res) {
        console.log(res.data)
      }
    })
  }
})

```

## 8.26 数据库操作符

数据库操作符用来处理查询更新操作指令，比如 eq 为等于、gt 为大于等于。通过 db.command 获取实例。示例代码如下。

```
Page({
  onCommand: function () {
    const db = wx.cloud.database()
    const _ = db.command
    db.collection('counters').where({
      // gt 方法用于指定一个“大于”条件，此处 _.gt(30) 是一个
      // “大于 30”的条件
      progress: _.gt(30)
    }).get({
      success: function (res) {
        console.log(res.data)
      }
    })
  }
})
```

数据库操作符提供如下方法。

(1) Command.addToSet (value: any|Object)：数组更新操作符。原子操作。给定一个或多个元素，除非数组中已存在该元素，否则添加进数组。

(2) Command.all (values: any[])：数组查询操作符。用于数组字段的查询筛选条件，要求数组字段中包含给定数组的所有元素。

(3) Command.and (expressions: any[])：查询操作符。用于表示逻辑“与”的关系，表示需同时满足多个查询筛选条件

(4) Command.bit (object: Object)：更新操作符。对字段进行位运算，可以进行 and/or/xor 运算。

(5) Command.elemMatch (condition: Object|Command)：用于数组字段的查询筛选条件。要求数组中包含至少一个满足 elemMatch 给定的所有条件的元素。

(6) Command.eq (value: any)：查询筛选条件。表示字段等于某个值。eq 指令接收一个字面量 (literal)，可以是 number、boolean、string、object、array、Date。

(7) Command.exists (value: boolean)：判断字段是否存在。

(8) Command.expr (aggregateExpression: Expression)：查询操作符，用于在查询语句中使用聚合表达式。该方法接收一个参数，该参数必须为聚合表达式。

(9) Command.geoIntersects (options: Object)：找出给定的地理位置

图形相交的记录。

(10) Command.geoNear (options: Object)：按从近到远的顺序，找出字段值在给定点的附近的记录。

(11) Command.geoWithin (options: Object)：找出字段值在指定区域内的记录，无排序。指定的区域必须是多边形 (Polygon) 或多边形集合 (MultiPolygon)。

(12) Command.gt (value: any)：查询筛选操作符，表示需大于指定值。可以传入 Date 对象用于进行日期比较。

(13) Command.gte (value: any)：查询筛选操作符，表示需大于或等于指定值。可以传入 Date 对象用于进行日期比较。

(14) Command.in (value: any[])：查询筛选操作符，表示要求值在给定的数组内。

(15) Command.inc (value: number)：更新操作符。原子操作。用于指示字段自增。

(16) Command.lt (value: any)：查询筛选操作符。表示需小于指定值。可以传入 Date 对象用于进行日期比较。

(17) Command.lte (value: any)：查询筛选操作符，表示需小于或等于指定值。可以传入 Date 对象用于进行日期比较。

(18) Command.max (value: any)：更新操作符。给定一个值，只有该值大于字段当前值才进行更新。

(19) Command.min (value: any)：更新操作符。给定一个值，只有该值小于字段当前值才进行更新。

(20) Command.mod (divisor: number, remainder: number)：查询筛选操作符。给定除数 divisor 和余数 remainder，要求字段作为被除数时  $value \% divisor = remainder$ 。

(21) Command.mul (value: number)：更新操作符。原子操作。用于指示字段自乘某个值。

(22) Command.neq (value: any)：查询筛选条件，表示字段不等于某个值。eq 指令接收一个字面量 (literal)，可以是 number、boolean、string、object、array、Date。

(23) Command.nin (value: any[])：查询筛选操作符。表示要求值不在给定的数组内。

(24) Command.nor (expressions: any[])：查询操作符。用于表示逻辑“都不”的关系，表示需不满足指定的所有条件。如果记录中没有对应的字段，则默认满足条件。

(25) Command.not (command: Command)：查询操作符。用于表示逻辑“非”的关系，表示需不满足指定的条件。

(26) Command.or (expressions: any[])：查询操作符。用于表示逻辑“或”的关系，表示需同时满足多个查询筛选条件。“或”指令有两种用法，一是可以进行字段值的“或”操作；二是也可以进行跨字段的“或”操作。

(27) Command.pop()：数组更新操作符。对一个值为数组的字段，将数组尾部元素删除。

(28) Command.pull (value: any)：数组更新操作符。给定一个值或一个查询条件，将数组中所有匹配给定值或查询条件的元素都移除掉。

	<p>(29) <code>Command.pullAll (value: any)</code>: 数组更新操作符。给定一个值或一个查询条件, 将数组中所有匹配给定值的元素都移除掉。与 <code>pull</code> 的差别在于只能指定常量值、传入的是数组。</p> <p>(30) <code>Command.push (values: Object)</code>: 数组更新操作符。对一个值为数组的字段, 往数组添加一个或多个值。或字段原为空, 则创建该字段并设数组为传入值。</p> <p>(31) <code>Command.remove()</code>: 更新操作符, 用于表示删除某个字段。</p> <p>(32) <code>Command.rename (value: string)</code>: 更新操作符。字段重命名。如果需要对嵌套深层的字段进行重命名, 需要用点路径表示法。不能对嵌套在数组里的对象的字段进行重命名。</p> <p>(33) <code>Command.set (value: any)</code>: 更新操作符。用于设定字段等于指定值。</p> <p>(34) <code>Command.shift()</code>: 数组更新操作符。对一个值为数组的字段, 将数组头部元素删除。</p> <p>(35) <code>Command.size (value: string)</code>: 更新操作符。用于数组字段的查询筛选条件。要求数组长度为给定值。</p> <p>(36) <code>Command.unshift (values: any[])</code>: 数组更新操作符。对一个值为数组的字段, 往数组头部添加一个或多个值。若字段原为空, 则创建该字段并设数组为传入值。</p>
<p>教学后 记</p>	