



# 信息工程系

# 教

# 案

课程名称： 移动应用开发

教 师： 黄梅佳

总 学 时： 72

理论学时： 36

实训学时： 36

上课班级： 移动应用开发 241

授课学期： 2025-2026 第二学期

# 第一章 初识uni-app

---

## 一、课程基本信息

1. **课程名称**: 移动应用开发
2. **授课教师**:
3. **授课班级**:
4. **授课时间**: 共4课时, 每课时45分钟

## 二、教学目标

### (一) 知识目标

1. 清晰阐述uni-app的发展历程, 包括重要时间节点和关键事件。
2. 准确描述uni-app项目开发工具(如HBuilderX)的特点及优势。
3. 熟练掌握uni-app项目搭建的完整流程, 理解项目模板的选择依据。
4. 深入理解uni-app项目目录结构中各文件和文件夹的功能与作用。
5. 掌握uni-app项目运行和打包的多种方式及操作要点。

### (二) 能力目标

1. 能够熟练运用HBuilderX进行项目创建、编辑和管理。
2. 独立完成uni-app项目的搭建、运行和打包, 解决过程中出现的常见问题。
3. 能够根据项目需求, 合理规划和使用的uni-app项目目录结构。
4. 培养学生在移动应用开发过程中的创新思维 and 实践能力。

### (三) 素质目标

1. **树立文化自信与创新精神**: 通过介绍uni-app作为国产跨端开发框架的发展成就, 让学生了解中国在软件开发领域的技术实力, 激发学生的自豪感和文化自信, 鼓励学生在开发中勇于创新, 为推动国产技术发展贡献力量。
2. **培养职业道德与团队协作精神**: 在项目开发实践中, 强调遵循软件编码规范, 培养学生严谨、负责的职业道德。通过小组协作解决问题, 培养学生的团队协作精神和沟通能力。
3. **增强社会责任感与使命感**: 引导学生思考移动应用开发对社会生活的影响, 鼓励学生开发出有益于社会的应用程序, 增强学生的社会责任感和使命感。

## 三、教学重难点

1. **教学重点**：uni-app项目开发工具的使用；项目搭建、运行和打包的操作流程；项目目录结构的理解与应用。
2. **教学难点**：理解uni-app跨平台开发的原理；项目打包过程中复杂参数的配置；解决项目开发过程中的技术难题。

## 四、教学方法

1. **讲授法**：讲解uni-app的理论知识，如发展历程、概念原理等，构建系统知识框架。
2. **演示法**：在HBuilderX等工具中演示项目创建、运行、打包等操作，让学生直观了解操作步骤和效果。
3. **实践法**：安排学生进行实际项目开发操作，在实践中巩固知识，提高动手能力，教师现场指导。
4. **讨论法**：组织学生讨论开发过程中的问题、技术趋势等，促进思想交流和经验分享。

## 五、教学过程

### 第一课时：uni-app发展历程与开发工具（45分钟）

1. **课程导入（5分钟）**：展示当下热门且具有创新性的国产移动应用，如字节跳动旗下的抖音，探讨其多平台适配的技术需求。引出uni-app这一国产跨端开发框架，激发学生对国产技术的兴趣和探索欲望。  
\*\*【思政融入】\*\*引导学生思考：在移动互联网时代，掌握核心技术的重要性。介绍我国在移动操作系统领域曾面临的"卡脖子"困境，以及国产企业自主创新、突破技术封锁的奋斗历程，激发学生的科技报国意识。
2. **知识讲解（20分钟）**
  - **uni-app发展历程**：以时间轴为线索，详细讲述Dcloud从2012年开始在小程序技术研发上的探索，到2015年推出“流应用”，再到推动微信小程序业务发展，最终积累大量开发者的历程。强调uni-app的出现是中国软件开发团队不断创新和努力的成果，鼓励学生在学习和未来工作中发扬创新精神。  
\*\*【思政融入】\*\*将uni-app的发展历程置于中国软件产业崛起的大背景下，介绍“国产替代”战略的意义——从依赖国外开发框架到拥有自主知识产权的跨端开发平台，体现了中国科技工作者自力更生、艰苦奋斗的精神。引用习近平总书记关于“关键核心技术是要不来、买不来、讨不来的”重要论述，激励学生树立科技自立自强的信念。
  - **uni-app开发工具**：介绍前端主流开发工具，对比它们在性能、功能、适用场景等方面的差异。重点讲解HBuilderX，阐述其采用C++架构带来的高性能优势，绿色发行包轻巧便于安装和部署的特点。展示HBuilderX对vue的优化效果，如代码智能提示、快速生成代码片段等功能。介绍其支持多种插件（java、nodejs、vscode插件）的特性，以及新建文件默认类型为markdown的便捷性，让学生了解其在uni-app开发中的优势。
3. **课堂互动（10分钟）**：组织学生分组讨论国产技术在软件开发领域的发展前景，以及uni-app的出现对国内移动应用开发行业的影响。每组推选代表发言，分享讨论成果，教师进行点评和补充，引导学生深入思考国产技术发展的意义。
4. **课堂小结（5分钟）**：回顾uni-app发展历程中的关键节点和重大事件，总结HBuilderX开发工具的核心优势。鼓励学生关注国产技术的发展，积极参与技术创新。
5. **作业布置（5分钟）**：要求学生课后查阅资料，了解至少一种其他国产跨端开发框架，并对比其与uni-app在技术特点、应用场景方面的异同。同时，思考国产跨端开发框架的发展对提升国家科技竞争力的作

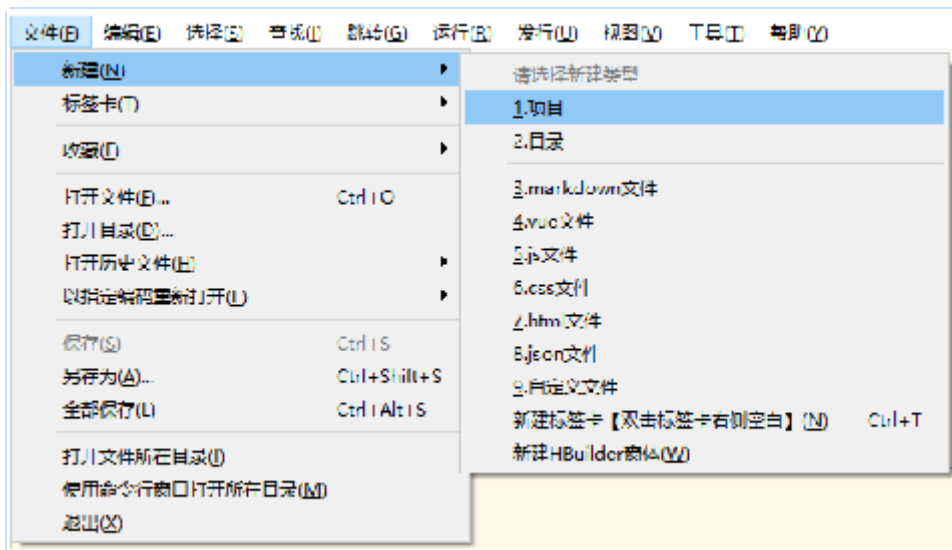
用。

## 第二课时：uni-app项目新建与目录结构（45分钟）

1. **课程导入（5分钟）**：回顾上节课HBuilderX开发工具的特点，提问学生如何利用该工具开启uni-app项目开发，引出本节课内容。

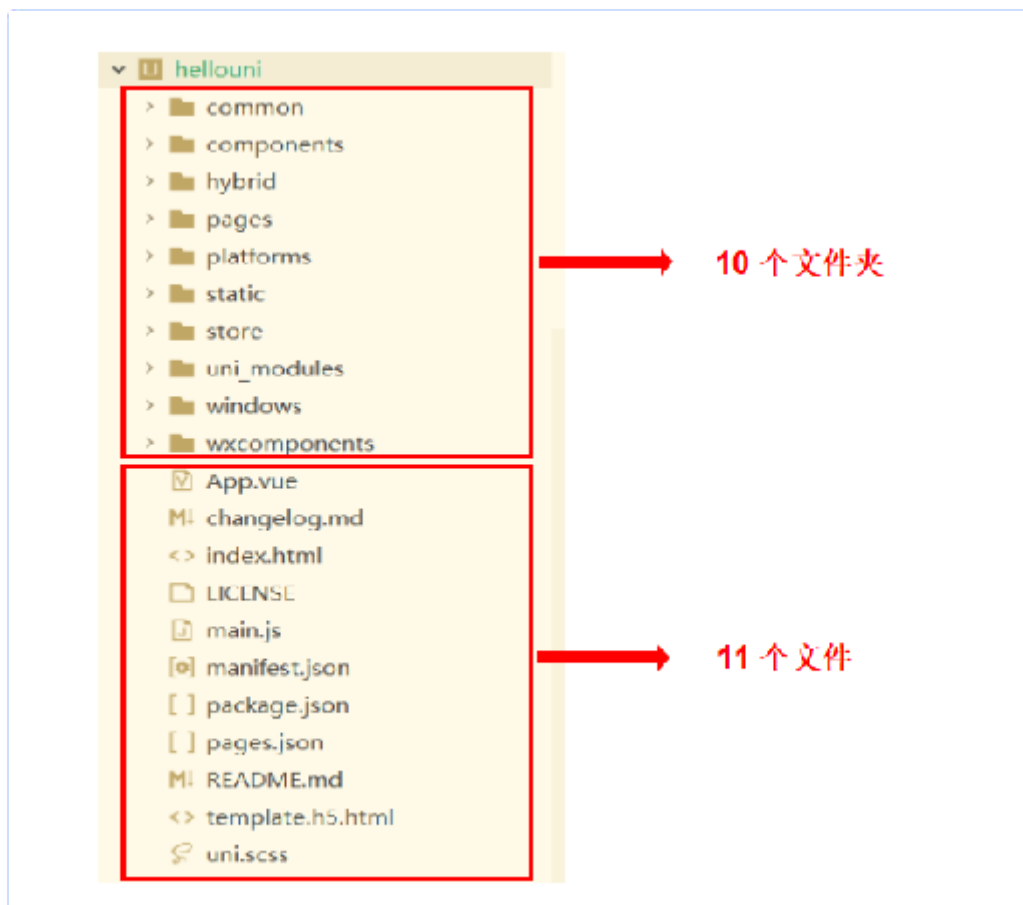
### 2. 知识讲解与演示（25分钟）

- **新建项目**：打开HBuilderX，详细演示新建uni-app项目的步骤。依次点击【文件】-【新建】-【项目】，在弹出的对话框中选择【uni-app】类型，解释项目名称命名规则和注意事项，如避免使用特殊字符。展示选择项目存放地址的方法，强调合理规划项目存储路径的重要性。介绍不同uni-app项目模板的特点和适用场景，如hello uni-app模板适合初学者快速了解项目结构，让学生根据需求选择合适模板后点击创建。



- **项目目录结构**：以新建的项目为例，深入讲解项目目录结构。详细介绍每个文件夹和文件的功能：common文件夹用于存放公共的js和css代码，实现代码复用；components文件夹存放可复用的组件，如按钮、表单等；pages文件夹存放所有页面文件，是项目页面展示的核心；static文件夹用于存放静态资源，如图片、音频等；platforms文件夹存放各平台专用页面，用于实现平台特定功能；App.vue用于配置App全局样式和监听应用生命周期；main.js是项目初始化入口文件，负责项目的启动和基础配置；manifest.json用于配置应用名称、appid、logo、版本等打包信息；pages.json用于配置页面路由、导航条、选项卡等页面类信息；uni.scss用于整体控制应用风格。结合实际项目开发场景，讲解各部分如何协同工作，如页面如何调用组件、如何引入公共样式等。在讲解过程中，强调遵循代码规范和良好的目录结构规划对项目可维护性的重要性，培养学生严谨的编程习惯。\*\*【思政融入】\*\*以“没有规矩不成方圆”的古训引入，讲解项目目录结构规范与国家法律法规的内在一致性——正如社会需要法治保障有序运行，软件项目也需要规范约束才能健康发展。引导学生理解“工匠精神”的内涵：在代码编写中追求精益求精、一丝不苟，每一个文

件的命名、每一个目录的规划都体现着程序员的职业操守和责任担当。



- 学生实践 (10分钟)**：学生在HBuilderX中按照演示步骤创建自己的uni-app项目，选择不同的项目模板进行尝试。教师巡视指导，帮助学生解决创建过程中遇到的问题，如路径选择错误、模板加载失败等。
- 课堂小结 (5分钟)**：总结项目新建的关键步骤和项目目录结构的核心要点，强调各部分在项目开发中的重要性。鼓励学生在实践中不断优化项目目录结构，提高项目开发效率。
- 作业布置 (5分钟)**：要求学生课后梳理自己创建项目的目录结构，绘制目录结构思维导图，并标注各部分功能。同时，尝试在项目中创建一个新的自定义组件，并将其放置在合适的目录位置，思考组件复用对项目开发的价值。

### 第三课时：uni-app项目运行 (45分钟)

- 课程导入 (5分钟)**：回顾项目新建和目录结构知识，提问学生如何让创建好的项目展示在设备上，引出项目运行的内容。
- 知识讲解与演示 (20分钟)**
  - 运行项目**：在HBuilderX中，依次演示在浏览器、真机、微信开发者工具中运行uni-app项目的操作。讲解在浏览器中运行的便捷性，可快速查看项目页面布局和基本交互效果，选择【运行】-【运行到浏览器】，并选择目标浏览器（如Chrome、Firefox等）即可。对于真机运行，详细介绍连接真机的准备工作，如开启手机开发者模式、安装手机驱动等，然后选择【运行】-【运行到手机或模拟器】，选择已连接的真机设备。重点讲解在微信开发者工具中运行的步骤，强调需要先搭建微信小程序环境。
  - 微信小程序环境搭建**：引导学生在“微信公众平台”上申请小程序账号，详细讲解申请流程，包括填写注册信息、选择账号类型、进行身份验证等。指导学生下载安装微信开发者工具，展示安装过

程中的注意事项，如选择合适的安装路径、安装过程中的提示信息处理等。打开微信开发者工具，演示如何进行项目配置，如设置项目目录、添加小程序AppID等。讲解微信开发者工具的界面功能，如模拟器、编辑器、调试器等模块的使用方法，让学生了解如何在工具中查看项目运行效果和调试代码。在讲解过程中，培养学生耐心细致的工作态度，以及遇到问题主动查阅文档解决问题的能力。**\*\*【思政融入】\*\***介绍微信小程序生态的发展对推动中国数字经济建设的贡献——小程序降低了创业门槛，助力小微企业数字化转型，体现了“科技惠民”的理念。引导学生认识到，作为开发者，搭建开发环境时的每一个细节都关乎用户体验，正如习近平总书记强调的“人民对美好生活的向往就是我们的奋斗目标”，我们开发的应用也应以服务人民、便利生活为宗旨。

- 3. 学生实践（15分钟）**：学生将自己的项目分别在浏览器、真机（若条件允许）、微信开发者工具中运行。教师在学生实践过程中进行现场指导，帮助学生解决运行过程中遇到的问题，如端口占用、网络连接异常、微信开发者工具配置错误等。鼓励学生在遇到问题时积极思考，尝试自行解决，培养学生独立解决问题的能力。
- 4. 课堂小结（5分钟）**：总结项目运行的不同方式及各自的操作要点，强调微信小程序环境搭建的关键步骤和注意事项。鼓励学生在课后多尝试在不同环境下运行项目，熟悉项目运行的各种情况。
- 5. 作业布置（5分钟）**：让学生课后尝试在不同的浏览器和真机设备上运行项目，记录遇到的问题及解决方法。同时，思考如何优化项目在不同设备上的运行性能，例如通过优化代码或调整资源加载方式来提高运行速度。

#### 第四课时：uni-app项目打包（45分钟）

- 1. 课程导入（5分钟）**：回顾项目运行知识，提问学生如何将开发好的项目变成可供用户安装和使用的应用程序，引出项目打包的内容。
- 2. 知识讲解与演示（20分钟）**
  - 打包原生App（云端）**：在HBuilderX中，打开需要打包的uni-app项目，选择【发行】-【原生App-云打包】。详细讲解打包选项的配置，如应用名称、版本号的设置规则，选择打包平台（Android、iOS）及对应的设置选项，如Android包名的命名规范、证书选择（自有证书、云端证书、公共测试证书等）及使用方法。介绍渠道包制作、原生混淆、广告联盟集成等高级选项的作用和配置方法，如原生混淆可保护代码安全，广告联盟集成可实现应用变现。展示打包过程中的

日志信息，让学生了解打包进度和可能出现的问题。讲解打包成功后的提示信息，包括安装包的



下载链接、存储位置等。

- **打包原生App (本地)**：介绍本地打包的操作入口，选择【发行】-【原生App-本地打包】。讲解本地打包与云端打包的区别，如本地打包需要自行配置更多的环境和依赖，适用于对安全性和定制性要求较高的项目。展示本地打包的相关指南和文档链接，指导学生如何获取和配置本地打包所需的环境和工具，如Android Studio、Xcode等。
  - **发布为H5**：演示在manifest.json可视化界面进行配置，讲解与H5发布相关的参数设置，如页面标题、图标、启动页等。选择【发行】-【网站-PC Web或手机H5 (仅适用于uni-app)】，展示生成的H5项目文件结构和部署方法。介绍发布为H5后在不同设备和浏览器上的兼容性测试方法，如使用浏览器开发者工具进行调试。讲解发布为H5与小程序发布流程的差异，如无需经过平台审核，但需要自行处理服务器部署和域名备案等问题。在讲解过程中，引导学生思考如何根据项目的特点和目标用户选择合适的打包方式，培养学生的项目规划能力。**\*\*【思政融入】\*\***在讲解打包发布时，引入网络安全和数据保护意识——应用发布意味着面向公众服务，开发者有责任确保用户数据安全，遵守《中华人民共和国网络安全法》《个人信息保护法》等法律法规。强调“技术向善”理念：技术能力越大，社会责任越重，引导学生树立正确的网络安全观和信息安全意识，做遵纪守法的开发者。
3. **学生实践 (15分钟)**：学生对自己的项目进行打包操作，可选择云端或本地打包中的一种进行尝试，并完成发布为H5的操作。教师在学生实践过程中进行指导，帮助学生解决打包过程中遇到的问题，如证书配置错误、打包参数不匹配、H5发布后页面显示异常等。鼓励学生在实践中探索不同的打包配置，了解其对项目的影响。
  4. **课堂小结 (5分钟)**：总结项目打包的不同方式及其适用场景，强调打包过程中关键参数的配置要点和注意事项。鼓励学生在课后继续练习项目打包，提高打包操作的熟练程度。
  5. **作业布置 (5分钟)**：要求学生课后完成项目实战和拓展实训项目要求的内容。在项目实战中，完成uni-app默认模板项目创建，添加静态资源图片，修改title文字，并进行项目打包和发布；在拓展实训项目

中，搜索了解微信、支付宝等各类平台小程序开发，以及IOS、安卓等平台APP开发的开发平台和技术，制作一个简单的技术介绍页面，重点突出国产技术的优势和特点。同时，思考如何将uni-app开发技术与其他平台开发技术结合，开发出更具创新性和竞争力的应用。

## 六、教学资源

1. 配备安装HBuilderX、微信开发者工具、相关手机驱动等软件的计算机机房。
2. 精心制作的教学PPT，包含详细的文字说明、操作截图和案例展示。
3. 准备相关操作视频，用于辅助学生理解复杂的操作流程，如微信小程序环境搭建、项目打包等。
4. 提供uni-app官方文档、技术论坛链接等学习资源，方便学生查阅和深入学习。

## 七、教学反思

教学过程中，学生在实践环节对项目搭建和运行的操作掌握较好，但在项目打包的复杂参数配置上存在困难。后续教学需强化这部分内容，增加实际案例演示，让学生更好理解参数含义和作用。在思政融入方面，学生对国产技术发展关注度有所提升，但讨论深度有待加强，后续可以设置更具引导性的话题，促进学生深入思考技术发展与社会责任的联系。

---

# 第二章 uni-app基础

---

## 一、课程基本信息

1. **课程名称：**移动应用开发
2. **授课教师：**
3. **授课班级：**
4. **授课时间：**共6课时，每课时45分钟

## 二、教学目标

### 1. 知识目标

- 深入理解pages.json文件各配置项（如globalStyle、pages、tabBar、subPackages）的详细功能、相互关系及应用场景。
- 熟练掌握资源引用（js文件、css文件、静态资源）的多种方式及路径规则。
- 全面掌握uni-app支持的页面样式选择器，理解其优先级规则和使用技巧。
- 透彻理解尺寸单位rpx的计算逻辑和在不同设计稿下的适配原理。
- 精准把握各基础组件（view、text、navigator、image等）的属性、事件及数据绑定机制。
- 深入领会flex布局的概念、容器和项目属性的作用及相互配合方式。

### 2. 能力目标

- 能够根据项目需求，独立且精准地配置pages.json文件，实现多页面应用的搭建、底部导航栏设计及分包加载优化。
- 能够准确无误地进行各类资源的引用和管理，确保项目资源的正确加载与使用。
- 能够运用页面样式选择器和尺寸单位rpx，根据设计稿实现页面元素的精确布局和样式调整。
- 能够灵活运用基础组件，结合属性绑定和事件绑定，开发出交互性强、功能完善的页面。
- 能够熟练运用flex布局，根据复杂的页面布局需求，合理设置容器和项目属性，实现高度定制化的布局效果。
- 能够综合运用所学知识，完成模拟影院购票系统和志愿者服务小程序“志愿者活动页”的开发任务，具备一定的项目开发能力。

### 3. 素质目标

- **培养创新与探索精神**：在讲解uni-app各种特性和开发技巧时，鼓励学生尝试新的配置、布局和交互设计，培养创新思维。例如，在flex布局教学中，引导学生探索不同属性组合实现独特布局效果，激发学生对技术创新的追求。
- **强化职业素养与责任意识**：强调代码规范在团队协作开发中的重要性，培养学生严谨、负责的职业素养。如在资源引用和组件使用教学中，要求学生遵循命名规范和代码结构规范，为今后进入职场做好准备。
- **树立文化自信与社会担当**：通过展示国产uni-app框架在移动应用开发领域的优势，增强学生对国产技术的信心和认同感。引导学生思考如何利用所学技术开发出对社会有价值的的应用，如在模拟项目开发中，鼓励学生关注社会公益需求，培养社会担当意识。

## 三、教学重难点

### 1. 教学重点

- pages.json文件中tabBar和subPackages的配置与应用。
- 基础组件的属性设置、事件绑定和数据渲染。
- flex布局容器属性和项目属性的灵活运用，实现复杂页面布局。

### 2. 教学难点

- 理解subPackages分包加载原理及preloadRule预载配置的优化作用。
- 掌握属性绑定和事件绑定机制，处理复杂交互逻辑。
- 在flex布局中，灵活运用各属性实现高度定制化的页面布局效果，尤其是处理多行布局和项目对齐问题。

## 四、教学方法

1. **讲授法**：系统讲解理论知识，如pages.json配置项、组件属性、flex布局概念等，让学生构建完整的知识体系。

2. **演示法**：在HBuilderX中实时演示代码编写、页面布局设计、组件使用等操作，让学生直观了解操作流程和效果。
3. **实践法**：安排学生进行实际项目开发操作，在实践中巩固知识，提高动手能力，教师进行现场指导。
4. **讨论法**：组织学生讨论开发过程中的问题、布局实现思路等，促进学生之间的思想交流和经验分享。

## 五、教学过程

### 第一课时：pages.json文件配置（45分钟）

1. **课程导入（5分钟）**：回顾uni-app项目搭建和运行的基本知识，提问学生如何进一步优化项目的页面展示和导航功能，引出pages.json文件的重要配置作用。展示几个风格各异且用户体验良好的uni-app应用界面，让学生观察页面布局和导航特点，激发学生学习兴趣。

#### 2. 知识讲解（20分钟）

- o **globalStyle和pages解析**：新建“默认模板”项目，打开pages.json文件。详细解读globalStyle中各属性（如navigationBarTextStyle、navigationBarBackgroundColor等）对应用全局样式的影响，通过修改属性值实时展示页面导航栏、状态栏、背景色等的变化。讲解pages数组中path和style属性的作用，强调path指定页面路径的准确性，style配置页面窗口表现的灵活性，如设置页面标题、是否开启下拉刷新等。结合实际项目案例，说明合理设置这些属性对提升用户体验的重要性，培养学生关注用户需求的意识。**\*\*【思政融入】\*\***以“以人民为中心”的发展思想类比，讲解全局配置与页面个性化配置的关系——正如国家在统一方针政策下允许地方因地制宜，globalStyle提供全局统一风格，而pages中的style允许各页面灵活调整，体现了“统而不死、活而不乱”的治理智慧。

### globalStyle全局设置应用

**1** → 状态栏

**2** → 导航条

**3** → 标题

**4** → 窗口背景色

属性	类型	默认值	描述
navigationBarBackgroundColor	HexColor	#F7F7F7	导航栏背景颜色
navigationBarTextStyle	String	white	导航栏标题颜色及状态栏颜色，仅支持black/white
navigationBarTitleText	String		导航栏标题文字内容
backgroundColor	HexColor	#ffffff	下拉显示出来的窗口背景色
backgroundTextStyle	String	dark	下拉loading的样式，仅支持dark/light
enablePullDownRefresh	Boolean	false	是否开启下拉刷新
navigationStyle	String	default	值为default/custom, custom即取消默认原生导航栏。

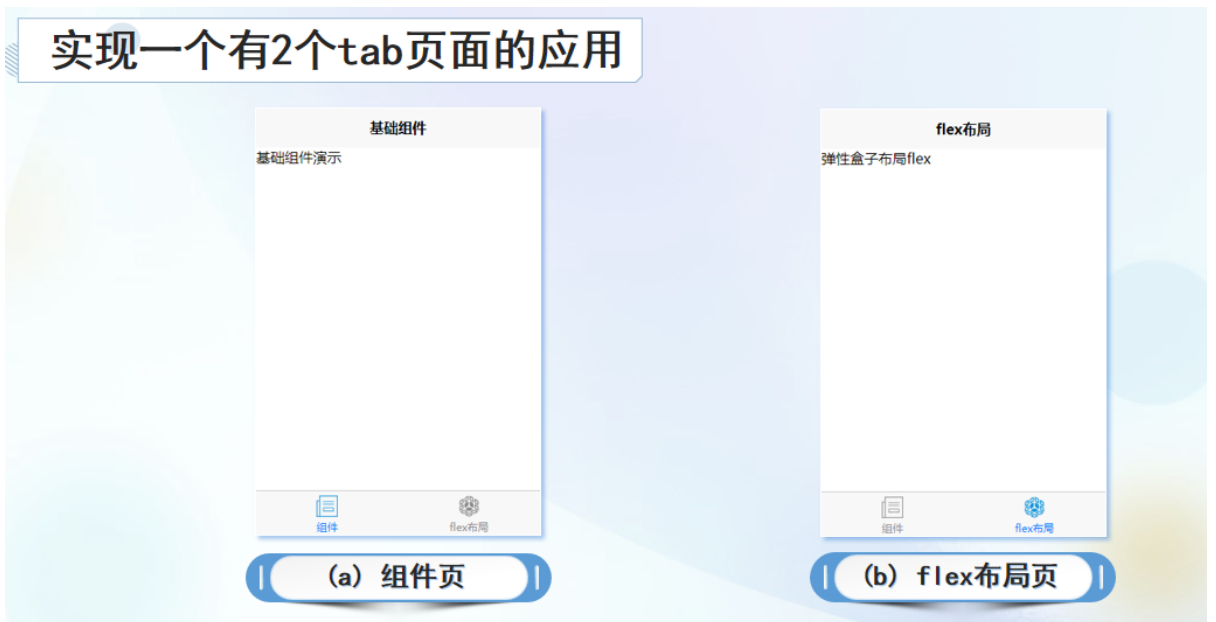
GlobalStyles属性

属性	类型	描述	属性
path	String	配置页面路径	path
style	Object	配置页面窗口表现，参考表2-1中globalStyle	style

pages属性

- o **tabBar配置与实践**：介绍tabBar在多页面应用导航中的重要性，讲解其各属性（color、selectedColor、backgroundColor等）的含义和设置方法。以实现一个有2个tab页面的应用为例，详细演示从准备tabBar图标（从iconfont网站搜索、下载并处理）、新建页面（在HBuilderX中选择合适模板创建vue文件和同名目录，并在pages.json中注册）到配置tabBar（设置页面路径、按钮文字、图标路径等）的全过程，展示最终页面效果，让学生直观感受tabBar的配置和应用。引导学生思考不同类型应用（如社交、电商、工具类）tabBar设计的差异，培养学生的创新设计思维。

## 实现一个有2个tab页面的应用



```
{
  "pages": [ //pages数组中第一项表示应用启动页，参考：https://uniapp.dcloud.io/collocation/pages
    {
      "path": "pages/compony/compony",
      "style": {
        "navigationBarTitleText": "基础组件",
        "enablePullDownRefresh": false
      }
    },
    {
      "path": "pages/index/index",
      "style": {
        "navigationBarTitleText": "uni-app"
      }
    },
    {
      "path": "pages/flex/flex",
      "style": {
        "navigationBarTitleText": "flex布局",
        "enablePullDownRefresh": false
      }
    }
  ],
  "tabBar": {
    "backgroundColor": "#F8F8F8",
    "color": "#8F8F94",
    "list": [
      {
        "text": "组件",
        "pagePath": "pages/compony/compony",
        "iconPath": "static/unComponet.png",
        "selectedIconPath": "static/componet.png"
      },
      {
        "text": "flex布局",
        "pagePath": "pages/flex/flex",
        "iconPath": "static/unflex.png",
        "selectedIconPath": "static/flex.png"
      }
    ]
  },
  "globalStyle": {
    "navigationBarTextStyle": "black",
    "navigationBarTitleText": "uni-app",
    "navigationBarBackgroundColor": "#F8F8F8",
    "backgroundColor": "#F8F8F8"
  }
}
```

- 课堂互动 (10分钟)**：组织学生分组讨论，如何根据特定应用场景（如校园服务应用）设计具有吸引力和便捷性的tabBar。每组推选代表分享讨论结果，教师进行点评和补充，引导学生从用户使用习惯、操作便捷性等方面优化设计。
- 课堂小结 (5分钟)**：总结pages.json中globalStyle、pages和tabBar的关键配置要点和作用，强调配置过程中的注意事项，如路径格式、属性取值范围等。
- 作业布置 (5分钟)**：要求学生课后创建一个有3个tab页面的uni-app项目，自定义tabBar样式和页面内容，思考如何从用户体验角度优化tabBar的交互效果，例如点击tab时的动画效果等。

### 第二课时：资源引用与页面样式 (45分钟)

- 课程导入 (5分钟)**：回顾上节课pages.json的配置内容，提问学生如何在项目中引入外部资源来丰富页面，引出资源引用和页面样式的内容。展示一些因资源引用不当或样式设置不合理导致用户体验差的页

面案例，如图片加载失败、样式混乱影响信息阅读等，引发学生对正确资源引用和样式设置的重视。

## 2. 知识讲解与演示 (20分钟)

- **资源引用**：在项目中创建js和css文件，演示引用js文件（如import add from '@/common/add.js;）、css文件（如@import "../common/uni.css;）的不同方式，详细解释绝对路径和相对路径的区别及适用场景。展示在内引入静态资源（如）的方法，强调资源路径的准确性对资源加载的影响，通过故意设置错误路径，展示资源无法加载的效果，加深学生印象。结合实际项目，讲解资源管理的重要性，如合理组织资源目录结构，提高项目的可维护性。
  - **页面样式**：介绍uni-app支持的选择器（#id、class、element等），通过在页面中添加不同选择器的样式规则，演示如何使用选择器来选择和设置页面元素的样式。讲解选择器优先级的计算规则，通过实例展示不同优先级选择器对同一元素样式的影响，如!important声明、内联样式、ID选择器、类选择器、标签选择器等的优先级顺序，让学生理解如何避免样式冲突和实现预期的样式效果。强调遵循样式规范和代码复用原则，提高开发效率。**\*\*【思政融入】\*\***以选择器优先级类比社会规则体系——不同层级的法律法规具有不同的效力等级，正如!important声明具有最高优先级，宪法具有最高法律效力。引导学生理解“尊法守法”的重要性，在代码中遵循规范，在社会中遵守法律，做知法守法的好公民。
3. **学生实践 (10分钟)**：学生在自己的项目中尝试进行资源引用和页面样式设置，如引入自定义的js函数实现页面功能，引用css文件美化页面布局，使用不同选择器设置元素样式。教师巡视指导，帮助学生解决遇到的问题，如资源路径错误、样式不生效等，及时纠正学生的错误操作。
  4. **课堂小结 (5分钟)**：总结资源引用的方法和页面样式选择器的使用要点，强调选择器优先级的重要性，提醒学生在实际开发中注意样式的兼容性和可维护性。
  5. **作业布置 (5分钟)**：让学生课后优化上节课创建的3个tab页面项目的资源引用和页面样式，如添加自定义图标、优化页面颜色搭配等，要求记录在优化过程中遇到的问题及解决方法。思考如何运用样式设计提升页面的视觉吸引力和信息传达效果。

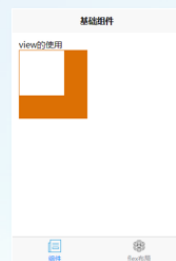
## 第三课时：尺寸单位与基础组件 - view和text (45分钟)

1. **课程导入 (5分钟)**：展示不同设计稿宽度下的页面设计图，提问学生如何确保页面元素在不同设备上显示一致，引出尺寸单位rpx的概念。展示一些因尺寸单位设置不当导致页面布局错乱的案例，让学生直观感受正确设置尺寸单位的重要性。
- ## 2. 知识讲解 (20分钟)
- **尺寸单位rpx**：详细讲解rpx的计算原理，即“750 \* 元素在设计稿中的宽度/设计稿基准宽度”。通过多个实际例子（如设计稿宽度为750px、640px、375px时元素宽度的计算），演示如何根据设计稿计算元素在uni-app中的rpx值，强调rpx在适配不同屏幕尺寸设备方面的优势。通过对比rpx与其他尺寸单位（如px）在不同设备上的显示效果，让学生更清晰地理解rpx的作用。引导学生思考如何根据目标用户群体的设备使用情况，合理选择尺寸单位，培养学生的用户导向思维。**\*\*【思政融入】\*\***以rpx的跨设备适配原理类比“共同富裕”理念——rpx的设计目标是让同一套代码在不同设备上都能呈现良好的显示效果，正如国家推动共同富裕是为了让不同地区、不同群体的人民都能共享发展成果。引导学生思考技术如何服务于“公平可及”的社会目标，做有温度的开发者。
  - **基础组件 - view和text**：介绍view组件作为容器组件的作用，类似于HTML中的div标签，讲解其常用属性（hover-class、hover-stop-propagation等）的功能和使用场景，如设置点击态效果、阻止点击事件冒泡等。通过在页面中添加view组件并设置不同属性，演示其在页面布局中的应用，

如创建不同的布局结构。讲解text组件用于包裹文本内容的功能，类似于Html中的span标签，介绍其属性（selectable、space、decode等）的作用，如实现文本可选、显示连续空格、解码特殊字符等。通过在页面中添加text组件并设置不同属性，展示文本内容的显示效果变化，如设置space属性显示不同类型的空格，设置decode属性解码特殊字符。结合实际应用场景，如新闻阅读类应用，说明如何利用这些组件提升用户阅读体验。

view是容器组件，类似于HTML中的div标签，用于包裹各种元素内容，是页面布局最常用的标签。

属性	类型	默认值	说明
hover-class	String	none	指定按下去的样式类。当 hover-class="none" 时，没有点击态效果
hover-stop-propagation	Boolean	false	指定是否阻止本节点的祖先节点出现点击态
hover-start-time	Number	50	按住后多久出现点击态，单位毫秒
hover-stay-time	Number	400	手指松开后点击态保留时间，单位毫秒



view组件属性

组件页面演示效果图

text组件用于包裹文本内容，类似于Html中的span标签。

属性	类型	默认值	说明
selectable	Boolean	false	文本是否可选（微信小程序user-select）
space	String		显示连续空格
decode	Boolean	false	是否解码

值	说明
ensp	中文字符空格一半大小
emsp	中文字符空格大小
nbsp	根据字体设置的空格大小

text组件属性

space属性的取值

- 课堂互动（10分钟）**：组织学生讨论在实际项目中，如何根据设计稿更高效地使用rpx进行页面布局，以及view和text组件在不同场景下的应用技巧。鼓励学生分享自己的想法和疑问，教师进行解答和指导，引导学生思考如何利用这些组件优化页面的用户体验。
- 课堂小结（5分钟）**：总结尺寸单位rpx的计算方法和view、text组件的关键属性及用法，强调在实际开发中要根据具体需求合理选择和设置这些属性。
- 作业布置（5分钟）**：要求学生课后使用view和text组件，结合rpx尺寸单位，设计一个简单的文本展示页面，如文章详情页，思考如何通过组件属性和尺寸单位优化页面的可读性和美观性。同时，思考如何利用view和text组件的特性，开发出更具交互性的文本展示效果，如点击文本展开更多内容。

#### 第四课时：基础组件 - navigator和image（45分钟）

- 课程导入（5分钟）**：回顾上节课view和text组件的使用方法，提问学生如何实现页面之间的跳转和图片展示功能，引出navigator和image组件。展示一些应用中页面跳转和图片展示的效果，激发学生的学习兴趣。
- 知识讲解与演示（20分钟）**

- **navigator组件**：讲解navigator组件常用属性（url、open-type、Hover-class等）的作用，重点介绍open-type属性不同取值（navigate、redirect、switchTab等）对应的跳转方式及应用场景，如保留当前页面跳转、关闭当前页面跳转、跳转到tabBar页面等。通过在页面中添加不同跳转方式的navigator组件，演示页面跳转效果，让学生理解各种跳转方式的区别，如在一个多页面应用中，使用不同open-type属性实现从详情页返回列表页、跳转到新的功能页面等操作。引导学生思考如何根据应用的业务逻辑和用户操作流程，合理选择跳转方式，提升用户体验。

属性	类型	默认值	说明
url	String		应用内的跳转链接，值为相对或绝对路径
open-type	String	navigate	跳转方式
Hover-class	Boolean	Navigate-hover	指定点击时的样式类，为none时，没有点击效果

#### navigator组件常用属性

- **image组件**：介绍image组件的常用属性（src、mode等），讲解src属性用于指定图片资源地址，mode属性用于设置图片裁剪、缩放的模式（scaleToFill、aspectFit等）。展示不同mode属性值下图片的显示效果，如拉伸、按比例缩放等，通过对比不同模式下同一图片的显示差异，让学生更直观地理解各模式的特点。演示如何根据设计需求选择合适的mode值来展示图片，如在展示商品图片时选择合适的模式保证图片不失真且能完整展示商品信息。强调图片资源的优化对页面加载速度和用户体验的影响，如选择合适的图片格式和压缩图片大小。

属性	类型	默认值	说明
src	String		图片资源地址
mode	String	'scaleToFill'	图片裁剪、缩放的模式

#### image组件常用属性

值	说明
scaleToFill	不保持纵横比缩放图片，使图片的宽度完全拉伸至填满image元素
aspectFit	保持纵横比缩放图片，使长边完全显示出来
aspectFill	保持纵横比缩放图片，使短边完全显示出来，长边方向将会发生截取
widthFix	宽度不变，高度自动变化，保持原图宽高比不变
heightFix	高度不变，宽度自动变化，保持原图宽高比不变

#### mode属性的取值

- 学生实践（10分钟）**：学生在自己的项目中添加navigator和image组件，实现页面跳转和图片展示功能，如在文章详情页添加返回按钮（使用navigator组件）和文章配图（使用image组件）。教师巡视指导，帮助学生解决组件使用过程中遇到的问题，如跳转路径错误、图片显示异常等，及时给予技术支持和建议。
- 课堂小结（5分钟）**：总结navigator和image组件的关键属性和使用要点，强调根据实际需求选择合适的跳转方式和图片显示模式，提醒学生注意图片资源的路径和大小优化，以提高页面加载速度。

5. **作业布置 (5分钟)**：让学生课后完善文章详情页的功能，如添加图片点击放大查看功能（可结合事件绑定实现），思考如何进一步优化页面跳转和图片展示的用户体验。同时，尝试使用不同的图片模式展示同一图片，观察对页面布局和加载速度的影响，并记录结果。

## 第五课时：flex布局 (45分钟)

1. **课程导入 (5分钟)**：展示一些复杂的页面布局效果，如九宫格导航、瀑布流布局等，提问学生如何实现这些布局，引出flex布局的概念。展示使用flex布局 and 传统布局方式实现同一页面布局的对比效果，让学生直观感受flex布局的优势。

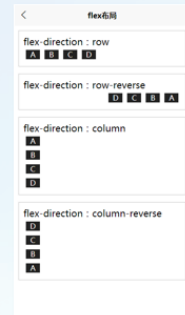
### 2. 知识讲解与演示 (20分钟)

- flex布局基本概念**：介绍flex布局（弹性盒模型）的起源和作用，讲解其在创建响应式网页布局方面的优势。通过示意图展示flex布局的容器和项目结构，让学生理解flex布局的基本原理，如容器如何分配空间给项目，项目如何在容器内排列等。强调flex布局在提升页面兼容性和适应性方面的作用，培养学生的技术前瞻性思维。**【思政融入】**以flex布局的"弹性"特性类比中国文化的"包容性"——flex布局能够根据容器大小自动调整项目排列，正如中华文明兼容并蓄、和而不同，在多元中求统一、在包容中促和谐。引导学生理解"和谐共生"的哲学思想，在技术开发中追求兼容并包，在人际交往中践行包容互鉴。

#### 容器属性：

#### 1 flex-direction属性

取值	描述
row	弹性盒子元素按横轴方向顺序排列（默认值）
row-reverse	弹性盒子元素按横轴方向逆序排列
column	弹性盒子元素按纵轴方向顺序排列
column-reverse	弹性盒子元素按纵轴方向逆序排列



flex-direction属性

flexdirection页面演示效果图

#### 2 flex-wrap属性

取值	描述
nowrap	容器为单行，该情况下flex子项可能会溢出容器。该值是默认属性值，不换行
wrap	容器为多行，flex子项溢出的部分会被放置到新行（换行），第一行显示在上方
wrap-reverse	反转wrap排列（换行），第一行显示在下方

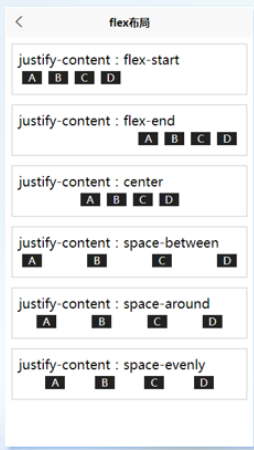


flex-wrap属性

flexwrap页面演示效果图

详细讲解flex-direction属性 (row、row-reverse、column等取值) 对项目排列方向的影响, 通过实例演示不同取值下项目的排列效果, 如在一个简单的列表布局中, 使用不同的flex-direction属性实现水平排列和垂直排列。讲解flex-wrap属性 (nowrap、wrap、wrap-reverse) 控制项目换行的方式, 展示项目在不同换行设置下的布局变化, 如当项目数量超出容器宽度时, 设置不同的flex-wrap属性观察项目的换行效果。

取值	描述
flex-start	弹性盒子元素将向行起始位置对齐 (默认值)
flex-end	弹性盒子元素将向行结束位置对齐
center	弹性盒子元素将向行中间位置对齐
space-between	弹性盒子元素会平均分布在行里, 第一个元素的边界与行的起始位置边界对齐, 最后一个元素的边界与行结束位置的边界对齐
space-around	弹性盒子元素会平均分布在行里, 两端保留子元素与子元素之间间距大小的一半
space-around	弹性盒子元素会平均分布在行里, 两端保留子元素与子元素之间间距大小的一半



**| justify-content属性 |**      **| justify-content页面演示效果图 |**

介绍justify-content属性 (flex-start、flex-end、center等取值) 用于设置项目在主轴上的对齐方式, 以及align-items属性 (flex-start、flex-end、center等取值) 用于设置项目在侧轴上的对齐方式, 通过实际案例演示这些属性的效果, 如在一个卡片式布局中, 使用justify-content和align-items属性实现卡片在容器内的居中对齐。

取值	描述
flex-start	弹性盒子元素向侧轴的起始位置对齐
flex-end	弹性盒子元素向侧轴的起始位置对齐
center	弹性盒子元素向侧轴的起始位置对齐
baseline	项目与行的基线对齐, 在未单独设置基线时, 等同于flex-start
stretch	默认值。如果项目未设置高度或设为auto, 将占满整个容器的高度



**| align-items属性 |**      **| align-items页面演示效果图 |**

讲解flex-flow属性作为flex-direction和flex-wrap的综合形式的用法, 以及align-content属性在多行布局中设置项目在侧轴方向对齐方式的作用, 通过复杂的多行布局案例展示align-content属性的效果。引导学生思考如何根据页面设计需求, 灵活组合这些属性实现独特的布局效果。

- 课堂互动 (10分钟)** : 组织学生分组讨论, 如何使用flex布局实现一个水平居中且垂直居中的元素, 以及如何实现一个自适应的多列布局。每组进行实践操作, 尝试实现讨论的布局效果, 教师巡视并参与讨论, 给予指导和建议, 引导学生思考不同属性之间的相互影响和配合方式。
- 课堂小结 (5分钟)** : 总结flex布局的基本概念和容器属性的关键点, 强调根据布局需求合理选择和组合这些属性, 提醒学生在实际应用中注意不同属性的兼容性和默认值。
- 作业布置 (5分钟)** : 要求学生课后使用flex布局重新设计文章详情页的布局, 使其在不同设备上具有更好的响应式效果, 思考如何利用flex布局实现更多复杂的页面布局需求。同时, 探索如何通过flex布局实现一些特殊的布局效果, 如不等宽的多列布局, 并记录实现过程和遇到的问题。

## 第六课时：案例实践与拓展（45分钟）

- 课程导入（5分钟）**：回顾前面所学的pages.json配置、组件使用、flex布局等知识，提问学生如何将这些知识综合运用到实际项目中，引出本节课的案例实践内容。展示模拟影院购票系统和志愿者服务小程序“志愿者活动页”的设计原型，激发学生的创作热情，同时引导学生思考这些项目对社会生活的积极影响，增强学生的社会责任感。**\*\*【思政融入】\*\***以“志愿者服务小程序”为载体，介绍中国志愿服务事业的发展成就——从2008年北京奥运会志愿服务到日常社区志愿服务，志愿者精神已成为社会主义核心价值观的生动体现。引导学生思考：作为开发者，我们不仅要写出好代码，更要用技术服务社会、回馈社会，将“奉献、友爱、互助、进步”的志愿精神融入开发实践。
- 案例实践（20分钟）**
  - 模拟影院购票系统**：详细讲解模拟影院购票系统的功能需求和设计思路，包括页面布局（如座位选择区、订单信息区）、交互逻辑（如点击座位选择、确认下单弹出提示框）。指导学生根据需求进行项目开发，在开发过程中引导学生运用所学知识，如使用flex布局进行页面布局，利用基础组件实现交互功能，通过js代码处理业务逻辑（如计算订单总价、弹出提示框）。教师实时巡视，帮助学生解决开发过程中遇到的问题，如布局错乱、交互功能异常等，及时给予技术支持和思路引导。鼓励学生在开发过程中创新，比如添加个性化的座位选择提示或优化订单确认流程，培养学生的创新能力。
  - 志愿者服务小程序“志愿者活动页”设计**：介绍志愿者服务小程序“志愿者活动页”的设计要求，包括搜索框、活动列表（包含活动标题、承办单位等信息）、报名按钮及页面跳转功能。指导学生进行页面设计和功能开发，鼓励学生创新设计，如优化活动列表的展示样式、添加更多的交互效果。教师巡视指导，帮助学生解决开发过程中遇到的问题，如数据绑定错误、页面跳转失败等，引导学生思考如何提升用户体验，培养学生关注社会需求、服务社会的意识。
- 课堂展示与讨论（10分钟）**：邀请部分学生展示自己开发的模拟影院购票系统和志愿者活动页，分享开发过程中的经验和遇到的问题及解决方法。组织其他学生进行点评和讨论，共同学习和进步，引导学生从他人的作品中汲取灵感，拓宽开发思路。在讨论过程中，强调团队协作和知识共享的重要性，培养学生的团队合作精神。
- 课堂小结（5分钟）**：总结本节课案例实践的重点内容和学生的表现，强调综合运用知识解决实际项目问题的重要性，鼓励学生在课后继续优化自己的作品，提高项目质量。同时，再次强调开发过程中融入社会责任和创新思维的意义。
- 作业布置（5分钟）**：要求学生课后进一步完善模拟影院购票系统和志愿者活动页的功能，如添加更多的页面交互效果、优化用户体验等。同时，鼓励学生自主探索uni-app的其他功能和应用场景，为后续学习做准备，如了解uni-app的插件使用、与后端接口的对接等。让学生思考如何将模拟项目拓展为具有实际应用价值的产品，以及如何运用所学知识为社会创造更多价值。

## 六、教学资源

- 安装HBuilderX开发工具的计算机机房，确保网络连接稳定。
- 精心制作的教学PPT，包含详细的代码示例、操作截图和案例演示。
- 提供相关的代码示例文件和素材资源，如模拟影院购票系统和志愿者服务小程序的设计图、图片素材、iconfont图标等，方便学生参考和使用。
- 准备uni-app官方文档和技术论坛链接，供学生查阅和深入学习。

## 七、教学反思

在教学过程中，学生在实践操作环节对基础组件和布局的应用有一定掌握，但在处理复杂交互逻辑和优化用户体验方面仍有不足。后续教学应增加复杂交互案例的讲解和练习，提升学生解决实际问题的能力。在思政融入方面，学生对项目的社会价值有一定思考，但创新思维的激发还需加强。后续可引入更多前沿技术案例，鼓励学生突破常规思维，同时深化思政教育与专业知识的融合，引导学生开发出更具社会价值和创新性的应用。

## 第三章 uni-app基础内容扩展

---

### 一、课程基本信息

1. **课程名称：**移动应用开发 - uni-app基础进阶
2. **授课教师：**
3. **授课班级：**
4. **授课时间：**共6课时，每课时45分钟

### 二、教学目标

#### 1. 知识目标

- 深入理解uni-app的应用、页面和组件生命周期函数，掌握其执行顺序和应用场景。
- 熟练掌握条件编译的语法和使用方法，能根据不同平台需求编写针对性代码。
- 掌握uni-ui、uview等扩展组件的安装、配置和使用方式，熟悉常用组件的属性和事件。
- 理解并掌握案例中新闻列表的实现原理和技术要点。

#### 2. 能力目标

- 能够在实际项目中，根据业务需求合理运用生命周期函数，实现数据加载、页面状态管理等功能。
- 能够运用条件编译，快速开发出适配不同平台的应用版本，提高开发效率。
- 熟练使用扩展组件进行页面布局和交互设计，提升页面的美观性和用户体验。
- 能够完成实训项目和拓展实训项目，将所学知识综合运用到实际开发中，具备独立开发小型uni-app应用的能力。

#### 3. 素质目标

- 在代码编写过程中，培养学生良好的软件编码规范素养，注重代码的可读性和可维护性。
- 鼓励学生在面对技术难题时，勇于探索、不畏困难，培养解决问题的能力 and 创新精神。
- 通过展示uni-app等国产技术在移动应用开发领域的成果，增强学生的民族自豪感和爱国情感。

### 三、教学重难点

## 1. 教学重点

- 应用、页面和组件生命周期函数的应用场景和使用方法。
- 条件编译的语法规则和在不同平台的应用。
- uni-ui扩展组件的安装、使用及常用组件属性和事件。
- 新闻列表案例的实现思路 and 关键代码。

## 2. 教学难点

- 理解组件生命周期函数之间的差异以及在复杂业务场景中的应用。
- 复杂条件编译逻辑的编写，确保不同平台代码的兼容性和正确性。
- 运用扩展组件实现复杂页面布局和交互效果，以及组件之间的协同工作。
- 新闻列表案例中数据的获取、传递和页面跳转逻辑的实现。

# 四、教学方法

1. **讲授法**：讲解理论知识，如生命周期函数、条件编译语法、扩展组件原理等，构建知识框架。
2. **演示法**：在HBuilderX中演示代码编写、组件使用、条件编译效果等，让学生直观了解操作流程和效果。
3. **实践法**：安排学生进行实际项目开发操作，在实践中巩固知识，提高动手能力，教师现场指导。
4. **讨论法**：组织学生讨论开发过程中的问题、实现思路等，促进学生之间的思想交流和经验分享。

# 五、教学过程

## 第一课时：uni-app生命周期（45分钟）

1. **课程导入（5分钟）**：回顾uni-app基础开发知识，展示一个包含页面切换、数据加载等功能的uni-app示例应用。引导学生思考应用在不同阶段（如启动、页面切换、数据更新）是如何进行管理和操作的，从而引出uni-app生命周期的概念，激发学生的学习兴趣。例如，展示一个简单的待办事项应用，在启动时加载本地存储的待办事项列表，在页面切换时更新页面标题和状态，在添加新事项时更新数据。提问学生，这些操作在代码层面是如何实现的，引导学生关注应用运行过程中的不同阶段和相应的处理逻辑。  
**\*\*【思政融入】\*\***以生命周期类比人生阶段——每个阶段都有其使命和责任，正如onLaunch是应用的"出生"，onShow是"登台亮相"，onHide是"退居幕后"，onUnload是"功成身退"。引导学生思考人生不同阶段的责任与担当，珍惜青春时光，在合适的阶段做合适的事，践行"不负韶华、不负时代"的人生信念。

## 2. 知识讲解（20分钟）

- **应用生命周期**：打开App.vue文件，详细讲解应用生命周期函数（onLaunch、onShow、onHide、onError等）的作用和触发时机。通过在函数中添加console.log语句，展示应用在不同状态下这些函数的执行情况，让学生直观感受应用生命周期的变化。比如，在onLaunch函数中添加console.log('App启动啦')，在onShow函数中添加console.log('App显示啦')，运行应用后，让学生观察控制台输出，理解应用启动和显示时函数的执行顺序。
- **页面生命周期**：创建一个新页面，详细讲解页面生命周期函数（onLoad、onShow、onReady、onHide、onUnload等）。重点介绍onLoad函数接收上个页面传递参数的功能，通过一个简单的页

面跳转传参示例，演示如何在onLoad中获取参数并进行相应操作。同时，讲解onPullDownRefresh和onReachBottom函数实现下拉刷新和上拉加载的原理，并通过实际代码演示其用法。例如，创建两个页面index.vue和detail.vue，在index.vue中通过uni.navigateTo跳转到detail.vue，并传递参数，在detail.vue的onLoad函数中获取参数并显示在页面上。然后，在index.vue中添加onPullDownRefresh和onReachBottom函数，实现下拉刷新数据和上拉加载更多数据的功能。

- 3. 课堂互动（10分钟）：**组织学生分组讨论，列举在实际应用中应用生命周期和页面生命周期函数的具体应用场景，如在onLaunch中进行用户登录状态检查、在onPullDownRefresh中更新数据等。每组推选代表分享讨论结果，教师进行点评和补充。可以引导学生从常见的应用类型（如社交、电商、资讯类应用）出发，思考不同场景下生命周期函数的应用，如社交应用在onLaunch时检查网络连接并自动登录，电商应用在onPullDownRefresh时获取最新商品信息等。
- 4. 课堂小结（5分钟）：**总结应用和页面生命周期函数的关键知识点，强调其在应用开发中的重要性。回顾每个函数的作用和触发时机，以及在实际项目中的应用场景。
- 5. 作业布置（5分钟）：**要求学生课后在自己的项目中添加应用和页面生命周期函数的相关代码，实现如记录应用启动次数、在页面加载时显示加载动画等功能。可以提供一些参考代码和思路，帮助学生完成作业，如记录应用启动次数可以在App.vue的onLaunch函数中使用uni.setStorageSync存储启动次数，在页面加载时显示加载动画可以在页面的onLoad函数中添加加载动画的代码。

## 第二课时：条件编译（45分钟）

- 1. 课程导入（5分钟）：**展示同一个uni-app应用在H5和微信小程序平台上的不同界面效果，提问学生如何实现这种针对不同平台的定制化开发，从而引出条件编译的概念。例如，展示一个应用在H5平台上使用HTML5的一些特性（如地理位置定位、本地存储）实现某些功能，在微信小程序平台上使用微信提供的接口（如微信支付、获取用户信息）实现相应功能，引导学生思考如何在代码层面实现这种平台差异化。**\*\*【思政融入】\*\***以条件编译类比"一国两制"伟大构想——在统一的国家框架下，允许不同地区根据实际情况采用不同的制度安排，正如条件编译在统一的代码库中为不同平台提供差异化实现。"一国两制"是中国智慧的体现，条件编译也是开发智慧的结晶，两者都体现了"实事求是、因地制宜"的思想方法。
- 2. 知识讲解（20分钟）**
  - **条件编译语法和平台标识：**详细讲解条件编译的语法（#ifdef、#ifndef）以及各个平台的标识（如APP-PLUS、H5、MP-WEIXIN等），通过在代码中添加不同平台的条件编译代码块，展示其生效条件和效果。比如，在标签内添加如下代码：

```
<!-- #ifdef H5 -->
<view>这是H5平台特有的内容</view>
<!-- #endif -->
<!-- #ifdef MP-WEIXIN -->
<view>这是微信小程序平台特有的内容</view>
<!-- #endif -->
```

运行应用后，分别在H5和微信小程序平台查看效果，让学生理解条件编译的作用。

- **在页面、样式和js中的应用**：分别在页面（如标签内）、样式（标签内）和js（标签内）中演示条件编译的使用方法，如根据不同平台显示不同的图片、设置不同的样式、执行不同的跳转逻辑等。在中，根据平台显示不同图片：

```
<template>
  <view>
    <!-- #ifdef H5 -->
    <image src="/static/h5-image.png"></image>
    <!-- #endif -->
    <!-- #ifdef MP-WEIXIN -->
    <image src="/static/weixin-image.png"></image>
    <!-- #endif -->
  </view>
</template>
```

在中，根据平台设置不同样式：

```
<style>
  /* #ifdef H5 */
  .text {
    color: red;
  }
  /* #endif */
  /* #ifdef MP-WEIXIN */
  .text {
    color: blue;
  }
  /* #endif */
</style>
```

在中，根据平台执行不同跳转逻辑：

```
<script>
  export default {
    methods: {
      goToPage() {
        // #ifdef H5
        uni.navigateTo({
          url: '/h5-page'
        });
        // #endif
        // #ifdef MP-WEIXIN
        uni.navigateTo({
          url: '/weixin-page'
        });
        // #endif
      }
    }
  }
}
```

```
}  
</script>
```

- 课堂互动 (10分钟)**：让学生思考并讨论在实际项目中还有哪些场景可以使用条件编译，如针对不同平台调用特定的API。鼓励学生分享自己的想法，教师进行总结和拓展。可以引导学生思考不同平台的独特功能，如App平台可以调用设备的摄像头、传感器等，小程序平台可以使用平台提供的支付、分享接口等，如何通过条件编译实现这些功能的调用。
- 课堂小结 (5分钟)**：总结条件编译的语法规则和应用要点，提醒学生注意不同平台代码的兼容性。强调条件编译在多平台开发中的重要性，以及在编写条件编译代码时需要注意的细节，如平台标识的正确使用、代码块的嵌套等。
- 作业布置 (5分钟)**：要求学生课后在项目中使用条件编译，实现H5平台显示一套广告组件，而小程序平台显示另一套广告组件的功能。可以提供一些广告组件的示例代码和思路，帮助学生完成作业，如在H5平台使用HTML5的广告嵌入代码，在小程序平台使用小程序广告组件。

### 第三课时：扩展组件uni-ui - 安装与基础使用 (45分钟)

- 课程导入 (5分钟)**：展示一些使用uni-ui扩展组件构建的精美页面，引发学生对提升页面开发效率和美观度的兴趣，从而引出uni-ui扩展组件的介绍。展示一些使用uni-ui构建的电商商品展示页面、社交应用个人中心页面等，让学生感受uni-ui组件的丰富性和实用性，激发学生的学习热情。

#### 2. 知识讲解与演示 (20分钟)

- uni-ui安装方式**：详细讲解uni-ui的安装方法，包括uni\_modules插件方式和npm方式，在HBuilderX中逐步演示操作步骤。强调安装过程中的注意事项，如HBuilderX版本要求、插件依赖等。以uni\_modules插件方式为例，演示如何打开HBuilderX的插件安装窗口，搜索uni-ui插件并进行安装；以npm方式为例，演示如何在项目根目录下打开命令行工具，执行npm install命令进行安装。
- 基本使用和辅助样式**：以uni-badge组件为例，详细演示如何在页面中使用uni-ui组件，包括在中添加组件代码、在中设置相关样式。介绍uni-scss辅助样式，如颜色类、边框半径类、间距类等，展示如何使用这些样式快速美化页面。在中添加uni-badge组件：

```
<template>  
  <view>  
    <uni-badge text="5" type="primary"></uni-badge>  
  </view>  
</template>
```

在中设置样式：

```
<style>  
  .uni-badge {  
    margin-right: 10px;  
  }  
</style>
```

然后介绍uni-scss的颜色类，如.uni-primary表示主色，.uni-success表示成功色等，让学生通过修改组件的颜色类来体验不同的颜色效果；介绍边框半径类，如.uni-radius-circle表示圆形边框，.uni-radius-t-0表示上面两个角为直角等，让学生尝试修改组件的边框半径；介绍间距类，如.uni-mt-2表示margin-top为8px，.uni-mx-2表示左右margin为8px等，让学生调整组件的间距。

3. **课堂互动 (10分钟)**：学生在自己的项目中安装uni-ui并添加一个uni-badge组件，实现如显示未读消息数量的功能。教师巡视指导，帮助学生解决安装和使用过程中遇到的问题。可以引导学生思考如何根据实际需求设置uni-badge的属性，如修改颜色、大小、位置等，以及如何使用辅助样式调整组件的外观。

4. **课堂小结 (5分钟)**：总结uni-ui的安装步骤和基本使用方法，强调辅助样式的优势。回顾安装过程中的关键步骤和注意事项，以及uni-badge组件的基本使用方法和uni-scss辅助样式的特点。

5. **作业布置 (5分钟)**：要求学生课后使用uni-ui的其他组件（如uni-card、uni-list），搭建一个简单的信息展示页面，熟悉组件的使用。可以提供一些组件的示例代码和布局思路，帮助学生完成作业，如使用uni-card组件展示商品信息，使用uni-list组件展示商品列表。

#### 第四课时：扩展组件uni-ui - 深入使用 (45分钟)

1. **课程导入 (5分钟)**：回顾上节课uni-ui的安装和基本使用，展示一些使用uni-ui实现的复杂页面效果，如卡片式布局、列表交互等，引导学生思考如何实现这些效果，引出本节课对uni-ui深入使用的学习。展示一个使用uni-ui构建的新闻资讯页面，包含卡片式新闻展示、列表导航、下拉刷新等功能，让学生观察页面效果，激发学生的学习兴趣。

#### 2. 知识讲解与演示 (20分钟)

- **流式栅格系统和uni-card卡片**：详细讲解uni-ui的流式栅格系统，通过实际案例演示如何使用uni-row和uni-col组件进行页面布局，实现响应式设计。介绍uni-card卡片组件的各种属性（如title、sub-title、thumbnail等）和事件，展示如何创建不同样式的卡片，如双标题卡片、带封面图的卡片等。创建一个简单的栅格布局示例：

```
<template>
  <uni-row>
    <uni-col :span="12">
      <view>左边内容</view>
    </uni-col>
    <uni-col :span="12">
      <view>右边内容</view>
    </uni-col>
  </uni-row>
</template>
```

然后展示如何通过修改span属性实现不同的布局效果。接着创建一个双标题卡片示例：

```
<template>
  <uni-card title="标题" sub-title="副标题" extra="额外信息" :thumbnail="avatar">
    <text class="uni-body">卡片内容</text>
  </uni-card>
</template>
```

在中定义avatar变量：

```
<script>
  export default {
    data() {
      return {
        avatar: '/static/avatar.png'
      };
    }
  }
</script>
```

让学生了解如何设置卡片的各种属性。

- **uni-list列表组件**: 介绍uni-list列表组件的功能和应用场景, 讲解其常用属性 (如title、note、ellipsis等) 和事件 (click、switchChange等)。演示如何创建不同类型的列表, 如导航列表、设置列表等, 以及如何处理列表项的点击和切换事件。创建一个导航列表示例:

```
<template>
  <uni-list>
    <uni-list-item title="首页" showArrow></uni-list-item>
    <uni-list-item title="分类" showArrow></uni-list-item>
    <uni-list-item title="我的" showArrow></uni-list-item>
  </uni-list>
</template>
```

然后添加点击事件处理:

```
<template>
  <uni-list>
    <uni-list-item title="首页" showArrow @click="goToHome"></uni-list-item>
    <uni-list-item title="分类" showArrow @click="goToCategory"></uni-list-item>
    <uni-list-item title="我的" showArrow @click="goToMy"></uni-list-item>
  </uni-list>
</template>
<script>
  export default {
    methods: {
      goToHome() {
        uni.navigateTo({
          url: '/home'
        });
      },
      goToCategory() {
        uni.navigateTo({
          url: '/category'
        });
      },
      goToMy() {
```

```
        uni.navigateTo({
          url: '/my'
        });
      }
    }
  }
</script>
```

让学生了解如何处理列表项的点击事件。3. **课堂互动 (10分钟)**：学生在自己的项目中使用流式栅格系统、uni-card和uni-list组件，构建一个类似新闻资讯列表页面，包含卡片式新闻展示和列表导航功能。教师巡视指导，帮助学生解决组件使用过程中的问题。可以引导学生思考如何根据页面设计需求合理设置组件的属性，如调整卡片的样式、列表项的显示内容等，以及如何实现页面的交互效果，如点击列表项跳转到相应页面。4. **课堂小结 (5分钟)**：总结uni-ui组件的深入使用要点，强调组件属性和事件的灵活运用。回顾流式栅格系统、uni-card和uni-list组件的关键知识点，以及在实际项目中的应用场景和注意事项。5. **作业布置 (5分钟)**：要求学生课后优化自己构建的新闻资讯列表页面，添加更多的交互效果，如点击卡片跳转到详情页等。提供详细思路：在index.vue的uni-card组件上绑定@click事件，在点击事件函数中通过uni.navigateTo跳转到详情页，并传递新闻的唯一标识（如newsId）。在详情页info.vue的onLoad函数中接收该标识，根据标识从模拟数据或后端接口获取新闻详情数据进行展示。同时，鼓励学生探索uni-ui其他组件，比如使用uni-popup组件实现点击卡片时弹出新闻摘要弹窗等效果，提升页面的交互性。

## 第五课时：案例：新闻列表 (45分钟)

1. **课程导入 (5分钟)**：展示一个完整的新闻列表应用，包含列表页和详情页，引导学生分析其功能和页面结构，引出本节课对新闻列表案例的实现讲解。引导学生观察新闻列表页的布局，如标题、图片、摘要的展示方式，以及详情页中新闻内容的排版、相关信息的呈现等，激发学生对实现该案例的兴趣。

### 2. 知识讲解与演示 (20分钟)

- **案例实现思路**：详细讲解新闻列表案例的实现步骤，包括新建项目、导入uni-list组件、在index.vue页面展示新闻列表、新建info.vue页面显示新闻详情、数据获取和传递等。深入分析返回数据格式，展示如何解析数据并在页面中显示。以模拟数据为例，假设数据是一个包含多个新闻对象的数组，每个对象包含id、title、summary、cover等属性，讲解如何利用v-for指令遍历数据数组，将每个新闻对象的属性绑定到相应的组件上进行展示。同时强调数据获取的方式，实际开发中可通过网络请求从后端接口获取数据，这里先以本地模拟数据代替，让学生理解数据处理流程。
- **关键代码实现**：在HBuilderX中逐步演示关键代码的编写，如在index.vue中使用v-for指令渲染新闻列表、在info.vue的onLoad函数中获取传递的新闻项数据并显示。强调页面跳转和数据传递的逻辑实现。在index.vue中编写如下代码：

```
<template>
  <uni-list>
    <uni-list-item v-for="(news, index) in newsList" :key="index"
    @click="goToDetail(news.id)">
      <view class="news-item">
        <image :src="news.cover" mode="widthFix" class="news-cover" />
        <view class="news-content">
          <text class="news-title">{{ news.title }}</text>
          <text class="news-summary">{{ news.summary }}</text>
```

```

        </view>
    </view>
</uni-list-item>
</uni-list>
</template>
<script>
export default {
  data() {
    return {
      newsList: [
        { id: 1, title: '新闻标题1', summary: '新闻摘要1', cover:
'/static/news1.jpg' },
        { id: 2, title: '新闻标题2', summary: '新闻摘要2', cover:
'/static/news2.jpg' }
      ]
    };
  },
  methods: {
    goToDetail(id) {
      uni.navigateTo({
        url: `/info?id=${id}`
      });
    }
  }
};
</script>
<style>
.news-item {
  display: flex;
  margin-bottom: 15px;
}
.news-cover {
  width: 80px;
  height: 60px;
  margin-right: 10px;
}
.news-content {
  flex: 1;
}
.news-title {
  font-size: 16px;
  font-weight: bold;
  margin-bottom: 5px;
}
.news-summary {
  font-size: 14px;
  color: #999;
}
</style>

```

在info.vue中编写代码获取并显示新闻详情:

```

<template>
  <view class="info-page">
    <text class="info-title">{{ news.title }}</text>
    <image :src="news.cover" mode="widthFix" class="info-cover" />
    <text class="info-content">{{ news.content }}</text>
  </view>
</template>
<script>
export default {
  data() {
    return {
      news: {}
    };
  },
  onLoad(option) {
    const newsId = option.id;
    const newsList = [
      { id: 1, title: '新闻标题1', cover: '/static/news1.jpg', content: '新闻内容1' },
      { id: 2, title: '新闻标题2', cover: '/static/news2.jpg', content: '新闻内容2' }
    ];
    this.news = newsList.find(n => n.id === newsId);
  }
};
</script>
<style>
.info-page {
  padding: 15px;
}
.info-title {
  font-size: 20px;
  font-weight: bold;
  margin-bottom: 10px;
}
.info-cover {
  width: 100%;
  height: 200px;
  margin-bottom: 10px;
}
.info-content {
  font-size: 16px;
}
</style>

```

3. **课堂互动 (10分钟)**：学生按照讲解的步骤在自己的项目中实现新闻列表案例，教师巡视指导，帮助学生解决遇到的问题，如数据绑定错误、页面跳转失败等。组织学生互相交流遇到的问题和解决方法，鼓励学生自主探索解决方案，培养学生解决问题的能力。
4. **课堂小结 (5分钟)**：总结新闻列表案例的实现要点，强调数据处理和页面交互的关键技术。回顾从数据获取、列表渲染到页面跳转和详情展示的整个流程，以及在这个过程中涉及的uni-app知识和uni-ui组

件的应用。

5. **作业布置 (5分钟)**：要求学生课后对新闻列表案例进行优化，如添加下拉刷新和上拉加载功能，提升用户体验。提示学生可以利用之前学过的页面生命周期函数`onPullDownRefresh`和`onReachBottom`来实现，鼓励学生尝试从后端获取真实数据，替换模拟数据。

## 第六课时：实训与拓展项目 (45分钟)

1. **课程导入 (5分钟)**：回顾本章所学的生命周期函数、条件编译和扩展组件等知识，强调综合运用这些知识进行项目开发的重要性，引出本节课的实训和拓展项目。结合之前学习的知识点，引导学生思考如何将这些知识应用到实际项目中，提升项目的质量和用户体验。

### 2. 项目实践指导 (20分钟)

- **实训项目**：详细讲解小程序应用中个人中心页面的设计要求和实现思路，鼓励学生使用`uni-ui`或`uview`等扩展组件进行开发。指导学生分析页面布局和功能需求，如签到功能、积分显示、记录查看等，引导学生思考如何运用所学知识实现这些功能。例如，签到功能可以通过调用后端接口记录签到时间和积分变化，积分显示可以从本地存储或后端获取数据进行展示，记录查看可以使用`uni-list`组件展示历史记录。展示一些个人中心页面的设计参考图，帮助学生更好地理解布局和功能需求。同时，引导学生思考如何利用条件编译实现不同平台下个人中心页面的差异化展示，比如在App端和小程序端展示不同的第三方登录方式。
  - **拓展实训项目**：介绍“国家反诈中心”APP的功能和重要性，讲解利用本章知识实现其中页面的思路 and 关键技术点。鼓励学生在实现过程中注重页面的实用性和安全性，如使用合适的组件展示反诈信息、运用条件编译适配不同平台。**\*\*【思政融入】\*\***深入介绍“国家反诈中心”APP背后的故事——这是我国打击电信网络诈骗犯罪、保护人民群众财产安全的重要举措。引导学生认识到，技术可以成为守护人民利益的利器，作为开发者应当将个人技能与国家需要相结合，积极投身于维护社会公平正义的事业中。同时强调网络安全意识，提醒学生在开发中注重用户隐私保护和数据安全，做有社会责任感的程序员。例如，使用`uni-card`组件展示不同类型的反诈案例，利用条件编译根据不同平台的特点调整页面布局和交互方式。提醒学生关注数据的安全性，避免在页面中暴露敏感信息。还可以引导学生思考如何利用生命周期函数实现页面数据的实时更新，比如在页面显示时 (`onShow`) 获取最新的反诈资讯。
3. **课堂实践与交流 (10分钟)**：学生开始进行实训项目和拓展实训项目的开发，教师巡视指导，及时解答学生遇到的问题。组织学生进行小组交流，分享开发过程中的经验和遇到的问题，共同探讨解决方案。鼓励学生互相学习，借鉴他人的思路和方法，提高开发效率。
  4. **课堂总结 (5分钟)**：总结本节课学生的实践情况，强调在项目开发中要注重细节和用户体验，鼓励学生在课后继续完善自己的项目。对学生在课堂上的表现进行点评，肯定学生的努力和成果，同时指出存在的问题和改进方向。
  5. **作业布置 (5分钟)**：要求学生课后完成实训项目和拓展实训项目的开发，并对项目进行优化和测试，确保功能的完整性和稳定性。提醒学生在完成项目后，整理项目代码和文档，为后续的项目展示和总结做好准备。同时，鼓励学生思考如何将项目进一步拓展，比如为个人中心页面添加更多个性化功能，或者为“国家反诈中心”APP页面增加互动元素，提升用户参与度。

## 六、教学资源

1. **软件工具**：计算机机房需安装HBuilderX开发工具，保证其版本满足课程中扩展组件安装及使用要求，并配备相关手机驱动，确保真机调试顺畅。同时，学生需在设备上安装微信开发者工具等必要的小程序开

发环境，用于不同平台应用的测试与运行。

2. **教学课件**：精心制作PPT，详细涵盖uni-app生命周期函数、条件编译、扩展组件、新闻列表案例等知识点，以图文并茂的形式呈现理论知识、操作步骤和代码示例，辅助课堂教学。
3. **代码示例与素材**：提供丰富的代码示例文件，包括uni-app项目中各功能模块的完整代码，以及新闻列表案例、个人中心页面、“国家反诈中心”APP页面等相关素材资源，如图片、图标等，方便学生参考、模仿和实践操作。
4. **在线学习资源**：收集并向学生推荐uni-app官方文档、uni-ui插件文档、技术论坛链接等在线学习资源，便于学生在课堂内外查阅资料、深入学习，及时获取最新技术信息，解决学习过程中遇到的问题。

## 七、教学反思

1. **知识理解与应用情况**：从学生作业和项目实践来看，对于uni-app生命周期函数，多数学生能理解其基本概念和简单应用场景，但在复杂业务逻辑中，区分和灵活运用组件生命周期函数仍有困难。条件编译方面，学生在简单平台适配场景下掌握较好，但编写复杂条件逻辑时容易出错，反映出对不同平台差异的理解不够深入。后续教学应增加复杂案例分析，强化对组件生命周期函数的深度讲解，针对条件编译多设置复杂场景练习，加深学生对平台差异的认识。
2. **扩展组件使用问题**：在扩展组件uni-ui的学习中，学生安装和基础使用较为顺利，但在实现复杂页面布局和交互时问题较多。如使用流式栅格系统进行响应式布局，以及处理组件间协同工作时，常出现布局错乱、交互异常的情况。这表明学生对组件属性和事件的理解仅停留在表面，缺乏综合运用能力。后续应增加综合性案例教学，引导学生深入探究组件间的关联和相互影响，提升实际应用能力。
3. **项目实践能力评估**：通过新闻列表案例和实训项目，发现学生在数据获取、传递和页面跳转逻辑实现上存在不足，对前后端数据交互和页面通信机制理解不透彻。部分学生在实际项目开发中，代码规范性和可读性较差，缺乏良好的编程习惯。后续教学要加强数据交互相关知识的教学，注重代码规范讲解和示范，通过代码审查、优秀代码分享等方式，培养学生良好的编程习惯。
4. **教学方法改进方向**：当前教学方法虽结合多种方式，但在实践环节，部分学生自主探索能力未得到充分锻炼，对教师依赖较大。后续应调整教学策略，在实践任务布置时，设置更多开放性问题，鼓励学生自主查阅资料、尝试不同解决方案，培养独立思考和创新能力。同时，优化案例教学，给予学生更多自主操作和讨论时间，提高学生课堂参与度，促进学生之间的知识共享和经验交流。

# 第四章 uni-app组件

---

## 一、课程基本信息

1. **课程名称**：移动应用开发 - uni-app组件深入学习
2. **授课教师**：
3. **授课班级**：
4. **授课时间**：共12课时，每课时45分钟

## 二、教学目标

1. **知识目标**

- 全面深入理解uni-app各类常用组件（容器组件、基础组件、表单组件、媒体组件、地图组件等）的属性、事件和应用场景。
- 掌握组件之间的协同工作原理，以及如何根据不同的业务需求选择合适的组件。
- 理解案例中典型注册页、个人中心页以及复杂业务页面的设计思路和实现原理。

## 2. 能力目标

- 能够熟练且灵活地运用常用组件，结合组件属性和事件，独立完成复杂页面的搭建，如公司首页、仿学习强国首页等。
- 具备根据项目需求优化组件使用的能力，提升页面性能和用户体验。
- 能够在实际项目开发中，高效解决组件使用过程中出现的常见问题。

## 3. 素质目标

- 强化学生的软件编码规范意识，培养严谨、规范的编程习惯。
- 激发学生的创新精神和爱国主义情怀，鼓励学生开发具有创新性和社会价值的應用。
- 培养学生的团队协作精神和沟通能力，通过项目实践促进学生之间的交流与合作。

# 三、教学重难点

## 1. 教学重点

- 各常用组件（如scroll-view、swiper、button、picker、input等）的核心属性和事件的应用。
- 表单组件的使用及数据提交处理。
- 案例中页面布局和组件交互逻辑的实现。

## 2. 教学难点

- 复杂组件（如picker的不同模式、form组件的数据提交机制）的深入理解和灵活运用。
- 实现组件间复杂的交互效果和数据传递。
- 在实际项目中，根据具体需求选择合适的组件并进行优化配置。

# 四、教学方法

1. **讲授法**：讲解组件的概念、属性、事件等理论知识，构建系统的知识框架。
2. **演示法**：在HBuilderX中实时演示组件的使用方法、属性设置和事件绑定，让学生直观了解操作效果。
3. **实践法**：安排学生进行实际项目开发操作，在实践中巩固知识，提高动手能力，教师现场指导。
4. **讨论法**：组织学生讨论组件的应用场景、遇到的问题及解决方案，促进学生之间的思想交流和经验分享。

# 五、教学过程

## 第一课时：容器组件（45分钟）

1. **课程导入（5分钟）**：回顾uni-app的基础开发知识，通过展示新闻资讯APP（如今日头条）的界面，突出其滚动新闻列表和广告轮播图。向学生提问：“大家每天浏览新闻时，看到的这些可以滑动的新闻列表和自动切换的广告图，在开发中是如何实现的呢？”以此引发学生思考，进而引出容器组件（scroll-view、swiper）的概念，激发学生的学习兴趣。\*\*【思政融入】\*\*以“学习强国”APP为例，介绍其作为党员干部和群众学习习近平新时代中国特色社会主义思想的重要平台，日活用户超过千万，体现了信息技术在传播党的理论、服务人民学习方面的巨大力量。引导学生思考：我们学习的组件技术，最终可以服务于传播正能量、弘扬主旋律的应用开发。

### 2. 知识讲解（20分钟）

- o **scroll-view组件**：在HBuilderX中打开uniappch04项目，新建“scroll-view”页面。详细讲解scroll-view组件的常用属性，如scroll-x、scroll-y控制滚动方向时，通过修改这两个属性的值，展示页面内容在横向和纵向滚动的不同效果；对于upper-threshold、lower-threshold设置触发滚动事件的距离，在页面中添加相关事件处理函数，让学生观察当滚动到指定距离时事件的触发情况；演示scroll-top、scroll-left设置滚动条位置，实时修改这两个属性值，让学生看到滚动条位置的变化。结合电商商品展示的案例，在页面中模拟商品展示，添加多个商品元素，设置scroll-y="true"，让学生看到商品列表可以纵向滚动，同时展示如何在滚动到底部时，通过@scrolltolower事件加载更多商品数据（在控制台打印加载数据的提示信息）。

属性名	类型	默认值	说明
scroll-x	Boolean	false	允许横向滚动
scroll-y	Boolean	false	允许纵向滚动
upper-threshold	Number/String	50	距顶部/左边多远时（单位px），触发 scrolltoupper 事件
lower-threshold	Number/String	50	距底部/右边多远时（单位px），触发 scrolltolower 事件
scroll-top	Number/String		设置竖向滚动条位置
scroll-left	Number/String		设置横向滚动条位置
scroll-into-view	String		值应为某子元素id（id不能以数字开头）。设置哪个方向可滚动，则在哪个方向滚动到该元素
scroll-with-animation	Boolean	false	在设置滚动条位置时使用动画过渡
refresher-background	String	"#FFF"	设置自定义下拉刷新区域背景颜色
refresher-triggered	Boolean	false	设置当前下拉刷新状态，true 表示下拉刷新已经被触发，false 表示下拉刷新未被触发
enable-flex	Boolean	false	启用 flexbox 布局。开启后，当前节点声明了 display: flex 就会成为 flex container，并作用于其孩子节点。
scroll-anchoring	Boolean	false	开启 scroll anchoring 特性，即控制滚动位置不随内容变化而抖动，仅在 iOS 下生效，安卓下可参考 CSS overflow-anchor 属性。
@scrolltoupper	EventHandle		滚动到顶部/左边，会触发 scrolltoupper 事件
@scrolltolower	EventHandle		滚动到底部/右边，会触发 scrolltolower 事件
@scroll	EventHandle		滚动时触发，event.detail = {scrollLeft, scrollTop, scrollHeight, scrollWidth, deltaX, deltaY}

- o **swiper组件**：同样在项目中新建“swiper”页面。介绍swiper组件的常用属性，如indicator-dots控制指示点显示，设置indicator-dots="true"，让学生看到轮播图下方出现指示点；设置autoplay="true"实现自动切换，修改interval属性调整自动切换的时间间隔；演示circular="true"时图片的衔接滑动效果。以电商商品详情页为例，展示如何利用这些属性创建带指示点、自动播放

且循环播放的商品图片轮播图，提升用户体验。

属性名	类型	默认值	说明
indicator-dots	Boolean	false	是否显示面板指示点
indicator-color	Color	rgba(0, 0, 0, .3)	指示点颜色
indicator-active-color	Color	#000000	当前选中的指示点颜色
active-class	String		swiper-item 可见时的 class
changing-class	String		acceleration 设置为 true 时且处于滑动过程中，中间若干屏处于可见时的class
autoplay	Boolean	false	是否自动切换
current	Number	0	当前所在幻灯片的 index
interval	Number	5000	自动切换时间间隔
duration	Number	500	滑动动画时长
circular	Boolean	false	是否采用衔接滑动，即播放到末尾后重新回到开头
vertical	Boolean	false	滑动方向是否为纵向
display-multiple-items	Number	1	同时显示的滑块数量

- 3. 课堂互动 (10分钟)**：组织学生分组讨论，鼓励他们列举更多实际应用场景中使用scroll-view和swiper组件的例子。巡视各小组，参与讨论，引导学生从不同类型的APP出发，如旅游APP中用scroll-view展示景点介绍文字，用swiper展示景点图片；社交APP中用scroll-view展示聊天记录，用swiper展示用户相册等。每组推选代表分享讨论结果，教师进行点评，对学生的想法给予肯定和补充，拓展学生的思维。
- 4. 课堂小结 (5分钟)**：总结scroll-view和swiper组件的关键属性和使用要点，强调根据不同需求合理设置属性的重要性。回顾每个属性的作用，提醒学生在实际使用中注意属性的取值范围和兼容性问题，如某些低版本浏览器对特定属性的支持情况。
- 5. 作业布置 (5分钟)**：要求学生课后在uniappch04项目中，使用scroll-view和swiper组件创建一个页面，展示一些图片或文本内容，实现滚动和轮播效果。明确作业要求，如图片或文本内容需自行准备，合理设置组件属性，使页面具有良好的交互效果，并且要尝试在滚动和轮播过程中添加一些简单的交互反馈（如滚动时改变背景颜色、轮播切换时显示提示信息）。

## 第二课时：基础组件 (45分钟)

- 1. 课程导入 (5分钟)**：展示微信公众号文章页面和视频播放页面，让学生观察文章中的富文本排版（包含图片、链接、不同格式的文字）以及视频加载时的进度条。提问学生：“这些丰富多样的文章内容展示和加载进度条，在开发中是依靠什么组件实现的呢？”从而引出基础组件（rich-text、progress）的学习，让学生认识到这些组件在丰富页面内容和交互方面的重要性。
- 2. 知识讲解 (20分钟)**

- **rich-text组件**：在uniappch04项目中新建“rich-text”页面。讲解rich-text组件的属性，如nodes用于设置富文本内容，通过展示一段包含图片、链接和特殊字符的富文本内容，演示如何使用nodes属性进行设置，展示不同格式的富文本效果；设置selectable="true"，让学生尝试长按选中富文本内容，感受其可选中效果；设置preview="true"，点击图片时实现预览功能。同时，展示如何通过@itemclick事件拦截点击链接操作，在methods中定义处理函数，实现自定义的跳转逻辑（如跳转到指定页面或执行特定操作）。

属性名	类型	默认值	说明
space	string		显示连续空格
nodes	Array / String	[]	节点列表 / HTML String
selectable	Boolean	true	富文本是否可以长按选中，可用于复制，粘贴等场景
image-menu-prevent	Boolean	false	阻止长按图片时弹起默认菜单（将该属性设置为image-menu-prevent或image-menu-prevent="true"），只在初始化时有效，不能动态变更；若不想阻止弹起默认菜单，则不需要设置此属性
preview	Boolean		富文本中的图片是否可点击预览。在不设置的情况下，若 rich-text 未监听点击事件，则默认开启。未显示设置 preview 时会进行点击默认预览判断，建议显示设置 preview
@itemclick	EventHandle		拦截点击事件（只支持 a、img 标签），返回当前node信息 event.detail={node}

- **progress组件**：在项目中新建“progress”页面。介绍progress组件的属性，如percent设置进度百分比，在页面中添加一个按钮，点击按钮时通过修改percent的值，模拟文件上传进度增加的效果；设置show-info="true"，让学生看到进度条右侧显示百分比；展示修改activeColor、backgroundColor属性，改变已选择和未选择进度条的颜色。

属性名	类型	默认值	说明
percent	Number	无	百分比0~100
show-info	Boolean	false	在进度条右侧显示百分比
border-radius	Number/String	0	圆角大小
font-size	Number/String	16	右侧百分比字体大小
stroke-width	Number	6	进度条线的宽度，单位px
activeColor	Color	#09BB07	已选择的进度条的颜色
backgroundColor	Color	#EBEBEB	未选择的进度条的颜色
active	Boolean	false	进度条从左往右的动画
active-mode	String	backwards	backwards: 动画从头播; forwards: 动画从上次结束点接着播
duration	Number	30	进度增加1%所需毫秒数
@activeend	EventHandle		动画完成事件

- 课堂互动（10分钟）**：让学生思考并讨论在实际项目中，rich-text组件和progress组件还可以应用在哪些场景。鼓励学生积极分享自己的想法，如在线文档编辑中使用rich-text组件实现文字排版和格式设置；任务管理系统中用progress组件展示任务完成进度等。教师进行总结和拓展，引导学生思考如何根据不同场景合理设置组件属性。
- 课堂小结（5分钟）**：总结rich-text和progress组件的关键属性和应用场景，提醒学生注意属性的兼容性和使用细节，如nodes属性的格式要求、progress组件中percent值的范围等。
- 作业布置（5分钟）**：要求学生课后在项目中使用时使用rich-text组件展示一篇带有图片和链接的文章，使用progress组件模拟一个文件上传进度条。明确作业要求，如文章内容需自行准备，进度条要实现动态变化效果（可以通过按钮控制进度增加或减少），并且要在页面中添加适当的文字说明，解释组件的功能。

### 第三课时：表单组件（一）（45分钟）

- 课程导入（5分钟）**：展示电商APP的注册登录页面和调查问卷页面，引导学生观察页面中的用户名、密码、手机号等输入框，以及注册按钮、选择项等表单元素。提问学生：“在开发这些用于收集用户信息的页面时，我们需要用到哪些组件呢？”从而引出表单组件（button、picker、slider、input）的学习，让学生了解表单组件在实际应用中的重要性。**\*\*【思政融入】\*\***在讲解表单收集用户信息时，引入个人信息保护意识——介绍《中华人民共和国个人信息保护法》的颁布实施，强调开发者在收集用户信息时必须遵循“合法、正当、必要”原则，获取用户明确同意。引导学生树立“用户至上、隐私优先”的开发伦理，在技术实现中自觉保护用户合法权益。
- 知识讲解（20分钟）**
  - button组件**：在uniappch04项目中新建“button”页面。详细讲解button组件的属性，如size设置按钮大小时，展示设置为default和mini的不同效果；type决定按钮样式，分别展示primary、default、warn等不同类型按钮的颜色和外观；设置plain="true"，让按钮变为镂空样式；设置loading="true"，显示加载图标。同时，展示如何绑定按钮的点击事件，在methods中定义点击处理函数，如点击按钮后在控制台打印提示信息，让学生了解按钮点击事件的处理方法。
  - picker组件**：在项目中新建“picker”页面。介绍picker组件的多种模式及属性，如普通选择器、多列选择器、时间选择器、日期选择器、省区市选择器等。针对每种模式，详细讲解关键属性，如range设置选择范围，value表示当前选中值，@change事件监听值的变化。以选择城市为例，展示如何使用省区市选择器，设置range属性提供城市数据，监听@change事件获取用户选择的的城市信息，并在页面中显示出来；以选择日期为例，展示时间选择器和日期选择器的使用，设置start和end属性限制可选时间范围。
- 课堂互动（10分钟）**：组织学生分组讨论，在一个注册页面中，如何合理搭配button和picker组件，实现用户信息的选择和提交功能。巡视各小组，引导学生考虑用户体验，如选择器的默认值设置、按钮的位置和样式等。每组推选代表分享讨论结果，教师进行点评和指导，纠正不合理的设计，补充更优化的方案。
- 课堂小结（5分钟）**：总结button和picker组件的常用属性和应用场景，强调根据业务需求选择合适组件模式的重要性。回顾每个属性的作用和不同模式选择器的特点，让学生在实项目中能够准确选择和配置组件。
- 作业布置（5分钟）**：要求学生课后在项目中创建一个包含多种picker组件和button组件的页面，实现用户选择兴趣爱好（普通选择器）、出生日期（时间选择器）、所在地区（省区市选择器）等信息并提交的功能。明确作业要求，提交按钮点击后要在控制台输出用户选择的信息，并且页面布局要合理美观。

### 第四课时：表单组件（二）（45分钟）

- 课程导入（5分钟）**：回顾上节课表单组件的内容，展示音乐APP的音量调节界面和社交APP的个人信息编辑页面，突出其中的滑动选择器和输入框。提问学生：“这些调节音量的滑动条和输入昵称的输入框，在开发中又是如何实现的呢？”引出本节课对slider和input组件的学习，让学生了解它们在用户交互中的作用。
- 知识讲解（20分钟）**
  - slider组件**：在uniappch04项目中新建“slider”页面。讲解slider组件的属性，如min、max设置最小值和最大值，通过修改这两个属性的值，展示滑动条可滑动范围的变化；step控制步长，设置不同的step值，让学生观察滑动时数值变化的间隔；展示修改activeColor、backgroundColor属性，

改变滑块左侧已选择部分和右侧背景条的颜色。同时，演示如何监听滑动事件获取当前值，在methods中定义事件处理函数，将当前值显示在页面上，模拟音量调节或亮度调节的效果。

- **input组件**：在项目中新建“input”页面。介绍input组件的常用属性，如value设置初始内容，type决定输入类型（文本、密码等），设置type="password"，让输入框显示为密码输入样式；placeholder显示占位符，设置placeholder="请输入内容"，让学生看到输入框为空时的提示信息；展示@input、@focus、@blur等事件监听输入框状态变化，在methods中定义相应的处理函数，如在@input事件中实时获取用户输入内容并进行简单的格式验证（如判断是否为数字），在@focus事件中改变输入框的边框颜色，在@blur事件中进行失去焦点后的验证提示。
3. **课堂互动（10分钟）**：让学生思考并讨论在实际项目中，slider和input组件可能会遇到的问题及解决方案，如输入框的输入限制（只能输入数字、限制长度等）、滑动选择器的精度控制等。鼓励学生分享自己的想法，教师进行总结和解答，展示如何通过设置maxlength属性限制输入框长度，通过调整step属性控制滑动选择器的精度。
  4. **课堂小结（5分钟）**：总结slider和input组件的关键属性和事件，强调在实际使用中要注意的细节，如输入验证和事件触发时机。回顾每个属性和事件的作用，让学生在项目中能够正确使用这两个组件。
  5. **作业布置（5分钟）**：要求学生课后在项目中创建一个包含slider和input组件的页面，实现一个可以调节数值并实时显示结果的功能，如通过slider调节字体大小，input输入数字并进行简单计算（如输入一个数字，乘以2后显示结果）。明确作业要求，页面要具有良好的交互效果，如添加适当的提示文字和按钮，方便用户操作。

## 第五课时：表单组件（三）（45分钟）

1. **课程导入（5分钟）**：展示设置页面中的单选、多选功能（如设置声音模式、选择通知类型）和注册页面的表单提交功能，引导学生观察其中的单选框、复选框、开关选择器和表单提交按钮。提问学生：“这些实现选择和表单提交功能的组件，在开发中是怎样的呢？”引出本节课对radio、checkbox、switch和form组件的学习，让学生了解它们在表单数据收集和管理中的作用。
2. **知识讲解（20分钟）**
  - **radio和checkbox组件**：在uniappch04项目中新建“radio&checkbox”页面。详细讲解radio和checkbox组件的属性，如value作为组件标识，checked判断是否选中，@change事件监听选中状态变化。创建一个包含单选框和复选框的页面，如选择性别（单选）和兴趣爱好（多选），演示如何在和中包裹单选框和复选框，通过监听@change事件获取选中的选项值，并在控制台输出。展示如何设置默认选中状态，通过设置checked="true"实现。
  - **switch组件**：在项目中新建“switch”页面。介绍switch组件的属性，如checked表示是否选中，type设置样式（switch或checkbox），@change事件监听状态变化。创建一个开关选择器，如是否开启通知，设置type="switch"，监听@change事件，当开关状态改变时，在控制台输出相应的提示信息，展示开关选择器的使用方法。
  - **form组件**：在项目中新建“form”页面。讲解form组件的属性，如report-submit用于返回formId发送模板消息，@submit事件提交表单数据，@reset事件重置表单。创建一个包含多个表单组件（如input、radio、checkbox、switch）的表单，将它们包裹在<form>标签内，设置report-submit="true"，通过监听@submit事件获取表单数据，并在控制台输出，展示如何使用@reset事件重置表单，使表单恢复到初始状态。

- 3. 课堂互动 (10分钟)**：组织学生分组讨论，在一个调查问卷页面中，如何合理使用radio、checkbox、switch和form组件来收集用户信息并提交表单。巡视各小组，引导学生思考如何优化表单布局，提高用户填写问卷的效率和体验，如合理分组、设置必填项提示等。每组推选代表分享讨论结果，教师进行点评和补充，提供更完善的设计思路。
- 4. 课堂小结 (5分钟)**：总结radio、checkbox、switch和form组件的常用属性和事件，强调组件之间的协同工作和表单数据处理的要点。回顾每个组件的关键属性和事件，以及它们在表单中的协同作用，让学生理解如何构建一个完整的表单功能。
- 5. 作业布置 (5分钟)**：要求学生课后在项目中创建一个调查问卷页面，包含各种表单组件，实现表单数据的收集和提交功能，并在控制台输出提交的数据。明确作业要求，调查问卷的问题类型要包含单选、多选、开关选择和输入框等，提交数据后要在控制台正确输出，并且页面要进行简单的美化，提高用户体验。

## 第六课时：媒体组件与地图组件 (45分钟)

- 1. 课程导入 (5分钟)**：展示相机APP、视频播放平台和地图导航APP的界面，分别突出拍照、视频播放和地图定位功能。提问学生：“这些我们日常生活中常用的拍照、看视频和导航功能，在开发中是借助什么组件实现的呢？”引出媒体组件 (camera、video) 和地图组件 (map) 的学习，让学生了解它们在丰富应用功能方面的重要性。

### 2. 知识讲解 (20分钟)

- **camera组件**：在uniappch04项目中新建页面。讲解camera组件的属性，如mode设置拍摄模式 (normal, scanCode)，演示在不同模式下的效果差异；device - position选择前置或后置摄像头，通过切换该属性值，让学生观察摄像头视角的变化；flash控制闪光灯，展示auto、on、off三种模式下闪光灯的不同状态。在页面中添加一个拍照按钮，通过代码实现点击按钮拍照，并将拍摄的照片显示在页面上，示例代码如下：

```
<template>
  <view>
    <camera device - position="back" flash="off" @error="handleError"
style="width: 100%; height: 300px;"></camera>
    <button type="primary" @click="takePhoto">拍照</button>
    <image v - if="photoPath" mode="widthFix" :src="photoPath"></image>
  </view>
</template>
<script>
export default {
  data() {
    return {
      photoPath: ''
    };
  },
  methods: {
    takePhoto() {
      const ctx = uni.createCameraContext();
      ctx.takePhoto({
        quality: 'high',
        success: (res) => {
          this.photoPath = res.tempImagePath;
        }
      });
    }
  }
}
```

```

        }
    });
},
handleError(e) {
    console.log('摄像头错误:', e.detail);
}
}
};
</script>

```

- **video组件**: 在项目中新建“video”页面。介绍video组件的属性，如src设置视频资源地址，可展示本地视频或网络视频的播放效果；autoplay决定是否自动播放，设置autoplay="true"时视频加载后自动播放；controls控制是否显示默认播放控件，通过切换该属性值，让学生观察播放控件的显示与隐藏；演示如何监听视频播放事件，如@play、@pause、@ended，在methods中定义相应的处理函数，在控制台输出视频播放状态的变化信息。
- **地图组件**: 在项目中新建“map”页面。讲解地图组件的使用方法和注意事项，强调在使用腾讯地图时需要在manifest.json文件中配置key。在页面中添加地图组件，设置地图的初始位置(latitude、longitude)和缩放级别(scale)，示例代码如下：

```

<template>
  <view>
    <map :latitude="latitude" :longitude="longitude" :scale="scale"
    style="width: 100%; height: 500px;"></map>
  </view>
</template>
<script>
export default {
  data() {
    return {
      latitude: 30.67,
      longitude: 104.06,
      scale: 15
    };
  }
};
</script>

```

- 课堂互动 (10分钟)**：让学生思考并讨论在实际项目中，媒体组件和地图组件可以如何与其他组件结合使用，如在电商APP中结合camera组件实现商品拍照上传，在旅游APP中结合地图组件展示景点位置等。鼓励学生分享自己的想法，教师进行总结和拓展，引导学生思考不同场景下组件结合的优势和实现方式。
- 课堂小结 (5分钟)**：总结camera、video和map组件的关键属性和应用场景，提醒学生注意地图组件的配置和各组件的兼容性问题。强调在实际开发中，要根据项目需求合理选择和配置这些组件。
- 作业布置 (5分钟)**：要求学生课后在项目创建一个页面，包含camera、video和map组件，实现拍照、视频播放和地图展示的功能。明确作业要求，拍照后能在页面展示照片缩略图，视频播放可控制暂

停和播放，地图能显示当前位置周边信息（可使用模拟数据）。

## 第七课时：案例一：典型注册页（45分钟）

1. **课程导入（5分钟）**：展示多个不同应用的注册页面，如微信、淘宝、抖音等，引导学生观察页面布局和包含的组件，提问学生：“这些注册页面都有哪些相似的地方？又有哪些不同？”引出本节课对典型注册页案例的学习，让学生了解如何将所学组件综合运用到实际页面开发中。

### 2. 知识讲解与演示（20分钟）

- **案例实现思路**：详细讲解典型注册页的制作步骤，包括在新工程的page.json中设置首页标题，在index.vue文件的中构建页面元素，如使用input组件创建用户名、手机号、密码等输入框，使用button组件实现注册按钮；在中编写样式美化页面。强调页面布局要符合用户操作习惯，提高用户体验。
- **关键代码实现**：在HBuilderX中逐步演示关键代码的编写过程。对于input组件，展示如何设置属性进行输入验证，如设置正则表达式验证用户名格式、手机号格式；对于button组件，展示如何绑定点击事件，在点击时检查所有输入框内容是否都已填写且符合要求，若符合则模拟发送注册请求（在控制台输出“注册成功”），否则提示用户填写完整或修改错误信息。示例代码如下：

```
<template>
  <view>
    <input type="text" v - model="username" placeholder="请输入用户名"
@blur="validateUsername">
    <input type="number" v - model="phone" placeholder="请输入手机号"
@blur="validatePhone">
    <input type="password" v - model="password" placeholder="请输入密码"
@blur="validatePassword">
    <button @click="register">注册</button>
  </view>
</template>
<script>
export default {
  data() {
    return {
      username: '',
      phone: '',
      password: ''
    };
  },
  methods: {
    validateUsername() {
      const reg = /^[a - zA - Z0 - 9_]{3,16}$/;
      if (!reg.test(this.username)) {
        uni.showToast({
          title: '用户名格式错误',
          icon: 'none'
        });
      }
    },
    validatePhone() {
      const reg = /^1[3 - 9]\d{9}$/;

```

```

        if (!reg.test(this.phone)) {
            uni.showToast({
                title: '手机号格式错误',
                icon: 'none'
            });
        }
    },
    validatePassword() {
        if (this.password.length < 6) {
            uni.showToast({
                title: '密码长度至少6位',
                icon: 'none'
            });
        }
    },
    register() {
        this.validateUsername();
        this.validatePhone();
        this.validatePassword();
        // 假设所有验证通过, 模拟注册请求
        uni.showToast({
            title: '注册成功',
            icon: 'success'
        });
    }
}
};
</script>

```

- 课堂互动 (10分钟)**：学生按照讲解的步骤在自己的项目中尝试实现典型注册页，教师巡视指导，帮助学生解决遇到的问题，如组件布局错乱、事件绑定无效等。组织学生互相交流遇到的问题和解决方法，鼓励学生自主探索解决方案。
- 课堂小结 (5分钟)**：总结典型注册页案例的实现要点，强调组件的合理使用和页面交互逻辑的实现。回顾注册页中各个组件的作用和关键代码，如输入框的验证逻辑、按钮的点击事件处理。
- 作业布置 (5分钟)**：要求学生课后优化典型注册页，添加更多的交互效果，如输入框的实时校验提示、密码强度检测等功能。提供一些实现思路，如使用正则表达式进行密码强度检测，在输入框失去焦点时触发检测并显示相应的强度提示（弱、中、强）。

## 第八课时：案例二：典型个人中心页 (45分钟)

- 课程导入 (5分钟)**：展示多个不同应用的个人中心页面，如支付宝、QQ、美团等，引导学生分析页面布局和功能模块，提问学生：“这些个人中心页面都有哪些功能？使用了哪些组件来实现这些功能？”引出本节课对典型个人中心页案例的学习，让学生进一步巩固所学组件的应用。
- 知识讲解与演示 (20分钟)**
  - 案例实现思路**：讲解典型个人中心页的制作步骤，包括新建工程并设置pages.json，在index.vue文件的中构建页面元素，如使用button组件实现功能按钮（我的订单、设置、收藏等），使用swiper

组件展示轮播内容（推荐活动、热门服务等）；在中编写样式美化页面。强调页面布局要简洁明了，方便用户操作。

- **关键代码实现**：在HBuilderX中逐步演示关键代码的编写过程。对于button组件，展示如何绑定点击事件实现页面跳转或功能操作；对于swiper组件，展示如何配置属性实现轮播效果，并添加指示点和自动切换功能。示例代码如下：

```
<template>
  <view>
    <button @click="goToOrders">我的订单</button>
    <button @click="goToSettings">设置</button>
    <swiper indicator - dots="true" autoplay="true" interval="3000">
      <swiper - item>
        <image src="/static/banner1.jpg" mode="widthFix"></image>
      </swiper - item>
      <swiper - item>
        <image src="/static/banner2.jpg" mode="widthFix"></image>
      </swiper - item>
    </swiper>
  </view>
</template>
<script>
export default {
  methods: {
    goToOrders() {
      uni.navigateTo({
        url: '/pages/orders/orders'
      });
    },
    goToSettings() {
      uni.navigateTo({
        url: '/pages/settings/settings'
      });
    }
  }
};
</script>
```

3. **课堂互动（10分钟）**：学生按照讲解的步骤在自己的项目中尝试实现典型个人中心页，教师巡视指导，帮助学生解决遇到的问题，如组件布局错乱、事件绑定无效等。组织学生互相交流遇到的问题和解决方法，鼓励学生自主探索解决方案。
4. **课堂小结（5分钟）**：总结典型个人中心页案例的实现要点，强调组件的合理使用和页面交互逻辑的实现。回顾个人中心页中各个组件的作用和关键代码，如按钮的点击事件处理、swiper组件的配置。
5. **作业布置（5分钟）**：要求学生课后优化典型个人中心页，添加更多的交互效果，如增加用户信息的编辑功能、展示个性化推荐内容等。提供一些实现思路，如使用input组件实现用户信息编辑，通过接口获取个性化推荐数据并展示。

第九课时：组件间交互与数据传递（45分钟）

1. **课程导入 (5分钟)**：回顾之前学习的各种组件，展示购物车页面中商品数量的增减 (input与button组件交互)、多页面间用户信息的传递等场景。提问学生：“在这些场景中，组件之间是如何进行交互和数据传递的呢？”引导学生思考，引出本节课的内容。

## 2. 知识讲解 (20分钟)

- **父子组件间的数据传递**：在uniappch04项目中创建父子组件示例。讲解父组件如何通过属性向子组件传递数据，在父组件的中，将数据作为属性传递给子组件，如：

```
<template>
  <view>
    <child - component :data - from - parent="parentData"></child - component>
  </view>
</template>
<script>
export default {
  data() {
    return {
      parentData: '这是来自父组件的数据'
    };
  }
};
</script>
```

在子组件中通过props接收数据，如：

```
<template>
  <view>
    <text>{{dataFromParent}}</text>
  </view>
</template>
<script>
export default {
  props: ['dataFromParent']
};
</script>
```

讲解子组件如何通过事件向父组件传递数据，在子组件中定义一个方法，通过\$emit触发事件并传递数据，如：

```
<template>
  <view>
    <button @click="sendDataToParent">向父组件传递数据</button>
  </view>
</template>
<script>
export default {
  methods: {
    sendDataToParent() {
```

```

        this.$emit('child - event', '这是来自子组件的数据');
    }
}
};
</script>

```

在父组件中监听子组件触发的事件，如：

```

<template>
  <view>
    <child - component @child - event="handleChildEvent"></child - component>
  </view>
</template>
<script>
export default {
  methods: {
    handleChildEvent(data) {
      console.log('接收到子组件的数据:', data);
    }
  }
};
</script>

```

- **非父子组件间的数据传递**：介绍使用事件总线（EventBus）实现非父子组件间的数据传递。在 main.js 中创建一个事件总线对象：

```

import Vue from 'vue';
const eventBus = new Vue();
Vue.prototype.$eventBus = eventBus;

```

在发送数据的组件中，通过 \$eventBus.\$emit 触发事件并传递数据：

```

<template>
  <view>
    <button @click="sendData">发送数据</button>
  </view>
</template>
<script>
export default {
  methods: {
    sendData() {
      this.$eventBus.$emit('non - parent - child - event', '非父子组件间传递的数据');
    }
  }
}

```

```
};  
</script>
```

在接收数据的组件中，通过\$eventBus.\$on监听事件并处理数据：

```
<template>  
  <view></view>  
</template>  
<script>  
export default {  
  mounted() {  
    this.$eventBus.$on('non - parent - child - event', (data) => {  
      console.log('接收到非父子组件的数据:', data);  
    });  
  },  
  beforeDestroy() {  
    this.$eventBus.$off('non - parent - child - event');  
  }  
};  
</script>
```

- 3. 课堂互动 (10分钟)**：组织学生分组讨论，在实际项目中还有哪些场景需要组件间的交互和数据传递，以及如何选择合适的方式进行实现。巡视各小组，参与讨论，引导学生思考不同场景下的最佳实践。每组推选代表分享讨论结果，教师进行点评和补充。
- 4. 课堂小结 (5分钟)**：总结组件间交互与数据传递的方式，包括父子组件间和非父子组件间的数据传递方法，强调在实际开发中要根据组件关系和业务需求选择合适的方式。
- 5. 作业布置 (5分钟)**：要求学生课后在项目中创建一个包含父子组件和非父子组件的数据传递和交互的页面，如父组件传递商品数据给子组件展示，子组件修改数据后通知父组件更新，同时实现两个非父子组件之间的数据传递。明确作业要求，通过控制台输出数据传递的过程和结果。

## 第十课时：组件的优化与性能提升 (45分钟)

- 1. 课程导入 (5分钟)**：展示一个页面加载缓慢、操作卡顿的uni-app应用，引导学生分析可能存在的问题，如组件使用不合理、数据加载过多等。提问学生：“如何优化这些问题，提升应用的性能和用户体验呢？”引出本节课对组件优化与性能提升的学习。
- 2. 知识讲解 (20分钟)**
  - 组件性能优化原则**：讲解组件性能优化的基本原则，如减少不必要的渲染、避免频繁的DOM操作、合理使用缓存等。解释这些原则的重要性，以及它们如何影响应用的性能。以频繁的DOM操作为例，过多的DOM操作会导致浏览器重新计算布局和样式，消耗大量资源，进而使应用卡顿，而减少不必要的渲染能有效降低这种性能损耗。
  - 具体优化方法**：针对不同类型的组件，介绍具体的优化方法。对于scroll-view组件，讲解如何设置合适的滚动区域和事件监听，避免频繁触发滚动事件导致性能下降，比如可以通过设置 `throttle` (节流) 或 `debounce` (防抖) 函数来限制滚动事件的触发频率；对于swiper组件，介绍如何优化图片加载，如使用图片懒加载技术，减少初始加载的图片数量，提高页面加载速度，可

借助uni - lazyload插件实现，只需在图片标签上添加相应指令即可；对于表单组件，讲解如何优化输入验证逻辑，避免在输入过程中进行过于复杂的验证，导致卡顿，例如可以将复杂的验证放在表单提交时进行，而在输入时仅进行简单的格式检查。以图片懒加载为例，展示如何使用uniapp的图片加载优化插件，在页面中添加图片时，设置相关属性实现图片懒加载，代码如下：

```
<template>
  <view>
    <image v-lazy="imageUrl" mode="widthFix"></image>
  </view>
</template>
<script>
export default {
  data() {
    return {
      imageUrl: 'your - image - url.jpg'
    };
  }
};
</script>
```

- 3. 课堂互动（10分钟）：**让学生思考并讨论在自己之前的项目中，哪些地方可能存在性能问题，以及可以采取哪些优化措施。鼓励学生分享自己的想法，教师进行总结和解答，提供更多的优化思路和方法。比如学生提到在注册页面中，输入框验证逻辑过于复杂导致卡顿，教师可引导学生讨论如何简化验证逻辑，或者采用异步验证的方式。
- 4. 课堂小结（5分钟）：**总结组件优化与性能提升的要点，强调在开发过程中要注重性能优化，提高用户体验。回顾优化原则和具体方法，让学生在真实项目中能够应用这些知识。
- 5. 作业布置（5分钟）：**要求学生课后对之前制作的注册页、个人中心页等页面进行性能优化，记录优化前后的加载时间和操作流畅度变化情况。建议学生使用浏览器的开发者工具来测量加载时间和性能指标，如Chrome浏览器的Performance面板。

## 第十一课时：综合项目实践（一）（45分钟）

- 1. 课程导入（5分钟）：**展示公司首页、仿学习强国首页等复杂页面的效果图，向学生介绍本次综合项目实践的任务，即使用所学的uni-app组件来实现这些页面。强调通过实践可以巩固所学知识，提升实际项目开发能力，激发学生的积极性。**【思政融入】**"学习强国"APP是新时代宣传思想工作的重要阵地，是推动马克思主义中国化时代化大众化的生动实践。在开发仿学习强国首页时，引导学生体会技术为思想传播服务的使命——一行行代码承载的是党的声音，一个个组件展示的是国家的发展成就。激励学生将专业技能与国家需求相结合，做新时代有理想、有本领、有担当的青年。
- 2. 项目分析与规划（20分钟）：**带领学生对项目进行详细分析，以公司首页为例，分析页面布局结构，划分出不同的功能模块，如公司简介、案例展示、新闻中心、服务范围等。引导学生思考每个模块可以使用哪些组件实现，如使用scroll-view展示新闻中心的滚动新闻列表，用swiper展示案例展示模块的图片轮播等。同时，讲解项目开发的流程和规范，包括代码的组织结构、命名规范等，强调团队协作的重要性，如在多人开发时，如何通过良好的代码规范避免冲突，提高开发效率。
- 3. 学生实践（10分钟）：**学生开始动手搭建项目框架，根据项目分析的结果，在HBuilderX中创建相应的页面和组件。教师巡视指导，帮助学生解决遇到的问题，如项目初始化错误、组件导入失败等。鼓励学生

自主思考，尝试解决问题，培养独立开发能力。

4. **课堂小结 (5分钟)**：总结本节课学生的实践情况，对遇到的共性问题进行集中解答。提醒学生在后续开发中注意代码规范和组件的合理使用，为下一课时的项目开发做好准备。
5. **作业布置 (5分钟)**：要求学生课后继续完善公司首页的基本布局搭建，完成各个功能模块的初步组件搭建和样式设置，确保每个组件的基本功能正常。

## 第十二课时：综合项目实践（二）（45分钟）

1. **课程导入 (5分钟)**：回顾上节课公司首页项目的进展情况，强调本节课要完成项目的细节优化和功能完善，使其具备完整的交互功能和良好的用户体验。
2. **功能完善与优化 (20分钟)**：学生继续进行项目开发，重点实现组件间的交互功能，如在公司首页中，实现点击新闻列表项跳转到新闻详情页，点击案例展示的图片弹出放大预览效果等。同时，优化页面样式，使其更加美观和符合企业风格。教师巡视指导，帮助学生解决交互功能实现过程中遇到的问题，如页面跳转传参错误、事件绑定无效等。引导学生思考如何提升用户体验，如添加加载动画、优化页面布局的合理性。
3. **项目展示与评价 (10分钟)**：邀请部分学生展示自己的项目成果，分享开发过程中的经验和遇到的问题及解决方法。组织其他学生进行评价，从页面效果、功能完整性、代码规范性等方面进行打分和提出建议。教师进行总结评价，肯定学生的优点，指出存在的不足，并提供改进方向。
4. **课堂小结 (5分钟)**：总结本次综合项目实践的重点内容和学生的整体表现，强调在实际项目开发中要注重细节和用户体验，鼓励学生在课后继续完善项目，将所学知识运用到更多的实际项目中。
5. **作业布置 (5分钟)**：要求学生课后根据课堂评价的反馈，对公司首页项目进行最后的优化和完善，提交完整的项目代码。同时，思考如何将本次项目开发的经验运用到仿学习强国首页的开发中，为后续的学习和实践做好准备。

## 六、教学资源

1. **软件工具**：安装HBuilderX开发工具的计算机机房，确保网络稳定，且安装微信开发者工具等相关调试软件，方便学生进行项目开发、调试与多平台预览。
2. **教学课件**：精心制作的PPT，包含组件属性、事件讲解，操作步骤截图，以及代码示例等，辅助教学。
3. **学习资料**：提供uni-app官方文档链接，方便学生查阅组件详细信息。准备丰富的代码示例和素材资源，如组件使用示例代码、项目图标、图片等，便于学生参考和实践。

## 七、教学反思

1. **教学内容方面**：部分学生对复杂组件的理解和应用存在困难，后续应增加实际案例，帮助学生更好地掌握组件在不同场景下的使用方法。
2. **教学方法方面**：实践环节中部分学生自主学习和解决问题能力不足，今后要注重培养学生自主探索能力，引导学生主动查阅资料解决问题。
3. **学习效果方面**：学生在组件综合应用和页面布局方面表现欠佳，后续需加强综合性项目训练，提升学生整体项目开发能力。

4. **思政融入方面**：思政元素融入不够自然，后续应更巧妙地将思政教育与专业知识结合，激发学生的爱国情怀和创新精神。

## 第五章 uni-app常用API (1)

---

### 一、课程基本信息

1. **课程名称**：移动应用开发 - uni-app常用API (1)
2. **授课教师**：
3. **授课班级**：
4. **授课时间**：共10课时，每课时45分钟

### 二、教学目标

#### 1. 知识目标

- 全面理解uni-app中各类常用API的功能、参数和使用场景，包括计时器、交互反馈、网络、数据缓存、路由等。
- 掌握各个API接口的具体使用方法，能够根据项目需求选择合适的API进行开发。
- 了解综合案例智云翻译项目的开发流程和技术要点。

#### 2. 能力目标

- 熟练运用各种API，结合Vuex插件管理数据，完成复杂功能模块的开发，如智云翻译项目中的文本翻译、历史记录管理等。
- 具备独立进行项目测试的能力，能够发现和解决API使用过程中出现的问题。
- 学会使用插件扩展项目功能，提升项目开发效率和质量。

#### 3. 素质目标

- 培养学生团队协作精神，在项目开发过程中能够与团队成员有效沟通和协作。
- 强化学生的软件编码规范素养，使其遵循良好的代码编写习惯。
- 培养学生遵循软件项目开发流程的职业素养，提升其职业竞争力。
- 鼓励学生探索新知、不畏困难，在面对技术难题时能够积极主动地寻求解决方案。

### 三、教学重难点

#### 1. 教学重点

- 常用API (如uni.request、uni.setStorage、uni.navigateTo等) 的使用方法和参数配置。
- 交互反馈API在提升用户体验方面的应用。

- 综合案例智云翻译项目的核心功能实现。

## 2. 教学难点

- 理解不同API在跨平台开发中的差异和兼容性问题。
- 掌握数据缓存的策略和数据管理，确保数据的准确性和安全性。
- 在智云翻译项目中，实现复杂的业务逻辑，如翻译功能的优化、历史记录的管理等。

# 四、教学方法

1. **讲授法**：讲解理论知识，如API的概念、分类、功能等，构建知识框架。
2. **演示法**：在HBuilderX中实时演示API的使用方法、代码编写和效果展示，让学生直观了解操作流程和效果。
3. **实践法**：安排学生进行实际项目开发操作，在实践中巩固知识，提高动手能力，教师现场指导。
4. **讨论法**：组织学生讨论API的应用场景、项目开发中的问题及解决方案，促进学生之间的思想交流和经验分享。

# 五、教学过程

## 第一课时：API概述（45分钟）

1. **课程导入（5分钟）**：回顾uni-app组件相关知识，展示一些功能丰富的uni-app应用，如社交APP的消息提示、电商APP的数据加载与缓存、地图导航APP的定位与地图展示等。引导学生思考这些功能是如何实现的，引出本节课对uni-app常用API的学习，激发学生的学习兴趣。**\*\*【思政融入】\*\***介绍我国在人工智能API领域的突破——如百度文心一言、阿里通义千问等国产大模型API的发布，标志着中国在AI基础设施领域实现了从跟跑到并跑的转变。引导学生认识到，API不仅是技术接口，更是国家科技实力的体现，鼓励学生关注国产API生态建设。
2. **知识讲解（20分钟）**
  - **API分类介绍**：详细讲解uni-app中API的分类，包括网络、基础、数据缓存、界面、媒体、设备、文件、绘画、第三方服务等模块。结合实际应用场景，举例说明每个模块中API的用途，如网络模块用于获取服务器数据，数据缓存模块用于存储用户信息等。
  - **Promise化API与非Promise化API**：讲解许多API进行了Promise化的特点，展示如何使用Promise的调用方式进行API调用，如uni.connectSocket的Promise化调用。同时，介绍几类没有进行Promise化的API，如同步方法（以sync结束）、以create开头和以manager结束的方法，让学生了解不同API的调用风格和适用场景。
3. **课堂互动（10分钟）**：组织学生分组讨论，列举生活中常见的应用场景，分析可能用到的uni-app API模块。每组推选代表分享讨论结果，教师进行点评和补充，引导学生加深对API分类的理解。
4. **课堂小结（5分钟）**：总结本节课的重点内容，包括API的分类、Promise化API和非Promise化API的特点，强调理解API分类对于后续学习和项目开发的重要性。
5. **作业布置（5分钟）**：要求学生课后查阅uni-app官方文档，了解更多关于各API模块的详细信息，记录下自己感兴趣的API及其功能。

## 第二课时：计时器（45分钟）

1. **课程导入（5分钟）**：展示一些包含倒计时、定时提醒功能的应用，如电商APP的限时抢购倒计时、学习类APP的定时提醒等。引导学生思考这些功能的实现原理，引出本节课对计时器API（setTimeout和setInterval）的学习。**\*\*【思政融入】\*\***以“时不我待、只争朝夕”的精神引入——计时器精确控制时间，正如新时代青年应珍惜时间、把握机遇。习近平总书记寄语青年“青春虚度无所成，白首衔悲亦何及”，引导学生树立时间意识，在学习和生活中做到今日事今日毕，以奋斗姿态书写青春华章。

### 2. 知识讲解（20分钟）

- **setTimeout和setInterval函数**：详细讲解setTimeout和setInterval的语法、参数含义和功能。通过在HBuilderX中编写简单示例，展示如何使用setTimeout设定一个定时器，在指定时间后执行回调函数；如何使用setInterval按照指定周期执行回调函数。同时，强调两者的区别和适用场景。

◆ **setTimeout (callback, delay, rest) :**  
设定一个定时器，在定时到期以后执行注册的回调函数。

参数	类型	必填	说明
callback	Function	是	回调函数
delay	Number	否	执行回调函数之间的时间间隔，单位 ms
rest	Any	否	param1, param2, ..., paramN 等附加参数，它们会作为参数传递给回调函数

◆ **setInterval (callback, delay, rest) :** 设定一个定时器，按照指定的周期（以毫秒计）来执行注册的回调函数。

◆ **delay**为两次执行回调函数之间的时间间隔，单位为毫秒，其他参数、返回值与setTimeout相同。

- **取消计时器**：介绍clearTimeout和clearInterval函数，讲解如何使用它们取消由setTimeout和setInterval设置的定时器。通过示例代码，演示在不同情况下取消定时器的操作。
- **实例演示**：以欢迎界面跳转为例，讲解在welcome页面中如何使用setTimeout在2秒钟后跳转到index页面，并读取系统信息获取界面高度。展示在onLoad函数中设置定时器和读取系统信息的代

码实现，让学生理解如何将计时器与其他功能结合使用。

```
<view class="btnwrap">
  <button @click="setTimeout" :disabled="time1 != 0"> 开启计时器(5s)</button>
  <button @click="clearTimeout">取消计时器</button>
</view>
<view class="btnwrap">
  <button @click="setInterval" :disabled="time2 != 0"> 开启间隔计时器</button>
  <button @click="clearInterval">取消间隔计时器</button>
</view>

<style>
.btnwrap{
  display: flex;
  flex-direction: row;
  justify-content: space-around;
  padding: 20px;
}
</style>

timeout. vue

<script>
export default {
  data() {
    return {
      time1: 0, time2: 0
    }
  },
  methods: {
    setTimeout() {
      this.time1 = setTimeout(function(res){
        let data = res;
        console.log(data);
      }, 5000, {
        "message": "计时器---我和大自然有个约会"
      }); //启动定时器
    },
    clearTimeout(){
      uni.clearTimeout(this.time1);
      this.time1 = 0;
    },
    setInterval() {
      this.time2 = setInterval(function(res){
        console.log(res);
      }, 1000, {
        "message": "计时器---绿水青山"
      }); //启动定时器
    },
    clearInterval(){
      clearInterval(this.time2);
      console.log(this.time2);
      this.time2 = 0;
      console.log(this.time2);
    }
  }
}
```

- 课堂互动 (10分钟)**：让学生在自已的项目中尝试使用setTimeout和setInterval实现一个简单的倒计时功能，如倒计时5秒后显示提示信息。教师巡视指导，帮助学生解决遇到的问题，如定时器未按预期执行、参数设置错误等。
- 课堂小结 (5分钟)**：总结本节课的重点内容，包括setTimeout和setInterval的使用方法、取消定时器的操作以及在实际项目中的应用场景，提醒学生注意定时器的参数设置和使用时机。
- 作业布置 (5分钟)**：要求学生课后完善倒计时功能，添加取消倒计时的按钮，实现点击按钮可以取消倒计时，并在控制台输出倒计时的剩余时间。

### 第三课时：交互反馈 (45分钟)

- 课程导入 (5分钟)**：展示一些应用中常见的交互反馈场景，如消息提示框、加载提示框、模态框等，如微信支付成功后的消息提示、淘宝商品加载时的loading提示。引导学生思考这些交互反馈对用户体验的重要性，引出本节课对交互反馈API的学习。

#### 2. 知识讲解 (20分钟)

- 信息提示框 (uni.showToast)**：详细讲解uni.showToast的使用方法和参数配置，包括title、icon、image、mask、duration、position等属性的含义和取值。通过示例代码，展示不同参数设置下的消息提示框效果，如显示成功图标、加载图标、自定义图片，设置提示框的显示时长和位置等。同时，介绍uni.hideToast用于隐藏消息提示框的方法。
- loading提示框 (uni.showLoading和uni.hideLoading)**：讲解uni.showLoading和uni.hideLoading的使用方法，展示如何设置loading提示框的文字内容、是否显示蒙层以及如何合适的时机显示和隐藏loading提示框，以提升用户体验。
- 模态框 (uni.showModal)**：介绍uni.showModal的使用方法和参数配置，如title、content、showCancel、cancelText、confirmText等属性的作用。通过示例代码，展示不同参数设置下的模态框效果，包括只有确定按钮和同时有确定、取消按钮的情况，以及如何获取用户点击按钮的结果。

- **操作菜单 (uni.showActionSheet)**：讲解uni.showActionSheet的使用方法和参数配置，如title、itemList、itemColor、success等属性的含义。通过示例代码，展示如何从底部向上弹出操作菜单，并获取用户点击按钮的索引。
- 课堂互动 (10分钟)**：学生在自己的项目中创建一个页面，使用交互反馈API实现以下功能：点击按钮弹出不同类型的消息提示框、显示和隐藏loading提示框、弹出模态框并获取用户点击结果、弹出操作菜单并根据用户选择进行相应操作。教师巡视指导，帮助学生解决遇到的问题，如参数设置错误、样式显示异常等。
  - 课堂小结 (5分钟)**：总结本节课学习的交互反馈API，包括信息提示框、loading提示框、模态框和操作菜单的使用方法和注意事项，强调合理使用这些API可以提升应用的用户体验。
  - 作业布置 (5分钟)**：要求学生课后优化交互反馈功能，根据不同的业务场景，选择合适的交互反馈方式，如在网络请求时显示loading提示框，操作成功或失败时显示相应的消息提示框。

#### 第四课时：网络 - request发起请求 (45分钟)

- 课程导入 (5分钟)**：展示一些依赖网络数据的应用，如新闻资讯APP、天气查询APP等，引导学生思考这些应用如何从服务器获取数据。引出本节课对uni-app中网络请求API (uni.request) 的学习，让学生了解网络请求在应用开发中的重要性。**\*\*【思政融入】\*\***在讲解网络请求时，引入网络安全意识——介绍我国《网络安全法》和《数据安全法》的核心要义，强调开发者在进行网络请求时必须确保数据传输安全（使用HTTPS协议）、保护用户隐私（不传输敏感明文数据）、防范网络攻击（验证请求合法性）。引导学生树立“安全为先”的开发理念，做网络安全的守护者。

#### 2. 知识讲解 (20分钟)

- **uni.request的使用方法**：详细讲解uni.request的语法、参数配置和使用方法，包括url、data、header、method、timeout、dataType等参数的含义和作用。通过示例代码，展示如何使用uni.request发起GET和POST请求，获取服务器数据，并处理成功和失败的回调函数。

参数名	类型	必填	默认值	说明	平台差异说明
url	String	是		开发者服务器接口地址	
data	Object/String/ArrayBuffer	否		请求的参数	App 3.3.7 以下不支持 ArrayBuffer 类型
header	Object	否		设置请求的 header，header 中不能设置 Referer。	App、H5端会自动带上cookie，且H5端不可手动修改
method	String	否	GET	有效值详见下方说明	
timeout	Number	否	60000	超时时间，单位 ms	H5(HBuilderX 2.9.9+)、APP(HBuilderX 2.9.9+)、微信小程序 (2.10.0)、支付宝小程序
dataType	String	否	json	如果设为 json，会尝试对返回的数据做一次 JSON.parse	
responseType	String	否	text	设置响应的数据类型。合法值：text、arraybuffer	支付宝小程序不支持
sslVerify	Boolean	否	TRUE	验证 ssl 证书	仅App安卓端支持 (HBuilderX 2.3.3+)，不支持离线打包
withCredentials	Boolean	否	FALSE	跨域请求时是否携带凭证 (cookies)	仅H5支持 (HBuilderX 2.6.15+)
firstIpv4	Boolean	否	FALSE	DNS解析时优先使用ipv4	仅 App-Android 支持 (HBuilderX 2.8.0+)
success	Function	否		收到开发者服务器成功返回的回调函数	
fail	Function	否		接口调用失败的回调函数	
complete	Function	否		接口调用结束的回调函数 (调用成功、失败都会执行)	

- **参数详解**：深入讲解data参数在不同请求方法和header设置下的转换规则，如GET方法将数据转换为query string，POST方法且header['content-type']为application/json时进行JSON序列化等。同时，介绍method属性的取值范围和根据服务器端接口文档选择正确取值的重要性。
- **success回调函数**：讲解success回调函数中data、statusCode、header、cookies等参数的含义和用途，展示如何从回调函数中获取服务器返回的数据和状态信息。
- **Promise封装**：介绍uni-app内部对uni.request进行的Promise封装，展示如何使用Promise的then和catch方法处理请求成功和失败的情况，使代码更加简洁和易于维护。实例：



5. **作业布置 (5分钟)**：要求学生课后完善图片上传功能，添加多图上传支持，并在上传过程中显示上传进度（可查阅文档探索相关实现方法）。

## 第六课时：数据缓存 (45分钟)

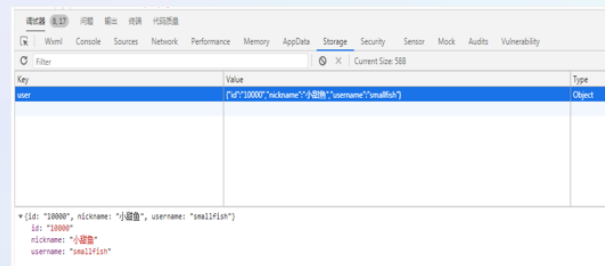
1. **课程导入 (5分钟)**：展示一些依赖本地数据缓存的应用场景，如电商APP的购物车数据缓存、社交APP的用户信息缓存等。引导学生思考数据缓存的作用和意义，引出本节课对uni-app中数据缓存API的学习。

### 2. 知识讲解 (20分钟)

- **数据缓存的概念和用途**：讲解数据缓存的概念，介绍在uni-app中数据缓存的作用，如减少网络请求、提高应用响应速度等。结合实际项目，说明数据缓存适用于保存会员登录状态信息、购物车数据、历史记录等场景。
- **将数据缓存到本地**：详细讲解uni.setStorage和uni.setStorageSync的使用方法和参数配置，包括key、data、success、fail、complete等参数的含义。通过示例代码，展示如何以异步和同步方式将数据存储在本地缓存中，强调两者的区别和适用场景。

```
<script>
export default {
  onLoad() {
    var user = this.getUserInfo();
    console.log(user);
    uni.setStorage({
      key: 'user',
      data: user,
      success: function(res) {
        console.log(res);
      }
    })
  },
  methods: {
    getUserInfo() {
      var user = new Object();
      user.id = '10000'
      user.nickname = '小甜鱼';
      user.username = 'smallfish'
      return user;
    }
  }
}
</script>
```

异步存储



The screenshot shows the console output of the asynchronous storage operation. It displays a table with columns for Key, Value, and Type. The key is 'user' and the value is an array containing a JSON object: [{"id": "10000", "nickname": "小甜鱼", "username": "smallfish"}]. Below the table, the JSON object is expanded to show its properties: id: "10000", nickname: "小甜鱼", and username: "smallfish".

Key	Value	Type
user	[{"id": "10000", "nickname": "小甜鱼", "username": "smallfish"}]	Object

```
* (id: "10000", nickname: "小甜鱼", username: "smallfish")
id: "10000"
nickname: "小甜鱼"
username: "smallfish"
```

```
<script>
onLoad() {
  var user = this.getUserInfo();
  console.log(user);
  uni.setStorageSync('user', use);
}
</script>
```

同步存储

- **获取本地缓存数据**：介绍uni.getStorage、uni.getStorageSync、uni.getStorageInfo和uni.getStorageInfoSync的使用方法，讲解如何以异步和同步方式获取指定key对应的内容，以及获取本地所有key值集合的方法。通过示例代码，展示如何在项目中获取本地缓存数据。

### ◆ uni.getStorage (OBJECT) 使用异步方式从本地缓存中获取指定key对应的内容

参数名	类型	必填	说明
key	String	是	本地缓存中的指定的 key
success	Function	否	接口调用成功的回调函数，res为 {data: key对应的内容}
fail	Function	否	接口调用失败的回调函数
complete	Function	否	接口调用结束的回调函数（调用成功、失败都会执行）

#### uni.getStorage (OBJECT) 参数说明

#### 示例代码:

```
onLoad() {  
  //以异步方式获取本地数据  
  uni.getStorage({  
    key: 'user',  
    success: function(res) {  
      console.log(res);  
    }  
  })  
}
```

- **清理本地缓存数据**: 讲解uni.removeStorage、uni.removeStorageSync、uni.clearStorage和uni.clearStorageSync的使用方法，展示如何移除指定key的数据和清理本地所有缓存数据，强调在实际项目中合理管理缓存数据的重要性。

#### ◆ uni.removeStorage (OBJECT) 用来异步从本地缓存中移除指定的key。

```
onLoad() {  
  //异步移除key=user的数据  
  uni.removeStorage({  
    key: 'user',  
    success: function(res) {  
      console.log(res);  
    }  
  })  
}
```

#### ◆ uni.removeStorageSync (OBJECT) 用来同步从本地缓存中移除指定的key

```
onLoad() {  
  //同步移除key=userSync的数据  
  uni.removeStorageSync('userSync');  
}
```

3. **课堂互动 (10分钟)**: 学生在自己的项目中实现数据缓存功能，包括存储用户自定义数据、获取缓存数据和清理缓存数据。教师巡视指导，帮助学生解决遇到的问题，如数据存储失败、获取数据错误等。
4. **课堂小结 (5分钟)**: 总结数据缓存API的使用方法，包括存储、获取和清理数据的操作，强调根据项目需求选择合适的缓存策略和API方法。
5. **作业布置 (5分钟)**: 要求学生课后使用数据缓存优化之前的项目，如在登录功能中缓存用户登录信息，下次打开应用时自动登录；在数据查询功能中缓存查询结果，减少重复查询。

## 第七课时: 路由 (45分钟)

1. **课程导入 (5分钟)**: 展示一些多页面应用，如电商APP的页面跳转、社交APP的页面切换等。引导学生思考这些应用中页面之间是如何进行跳转和导航的，引出本节课对uni-app中路由API的学习。
2. **知识讲解 (20分钟)**
  - **路由API介绍**: 详细讲解uni.navigateTo、uni.redirectTo、uni.switchTab、uni.reLaunch、uni.navigateBack等路由API的功能和使用场景。通过示例代码，展示不同路由API实现的页面跳转

效果，如保留当前页面跳转、关闭当前页面跳转、跳转到tabBar页面等。

API	说明	Open-type属性值
uni.navigateTo(OBJECT)	保留当前页面，跳转到应用内的某个页面，使用uni.navigateBack返回	navigate
uni.redirectTo(OBJECT)	关闭当前页面，跳转到应用内的某个页面。	redirect
uni.switchTab(OBJECT)	跳转到 tabBar 页面，并关闭其他所有非 tabBar 页面。	switchTab
uni.reLaunch(OBJECT)	关闭所有页面，打开到应用内的某个页面。	reLaunch
uni.navigateBack(OBJECT)	关闭当前页面，返回上一页面或多级页面。	navigateBack

- **参数配置**：讲解uni.navigateTo和uni.navigateBack的OBJECT参数配置，包括url、delta、animationType、animationDuration、events等参数的含义和用途。强调不同路由API在参数使用上的差异，如switchTab的路径不能带参数等。

参数	类型	必填	默认值	说明	差异说明
url	String	是		需要跳转的应用内非 tabBar 的页面的路径	uni.navigateTo中的参数
delta	Number	否	1	返回的页面数，如果 delta 大于现有页面数，则返回到首页	uni.navigateBack的参数
animationType	String	否	pop-in		App
animationDuration	Number	否	300	窗口动画持续时间，单位为 ms	App
events	Object	否		页面间通信接口，用于监听被打开页面发送到当前页面的数据。2.8.9+ 开始支持。	uni.navigateTo的参数
success	Function	否		接口调用成功的回调函数	
fail	Function	否		接口调用失败的回调函数	
complete	Function	否		接口调用结束的回调函数（调用成功、失败都会执行）	

- **页面跳转限制和注意事项**：介绍页面跳转路径的层级限制，如微信小程序有10层限制；说明switchTab只能打开tabBar页面，reLaunch可以打开任意页面；强调不能在App.vue里面进行页面跳转，以及H5端页面刷新后页面栈消失，navigateBack不能返回的问题及解决方法。
  - **数据传递**：讲解在uni.navigateTo、uni.redirectTo、uni.reLaunch中如何通过url携带参数进行页面间的数据传递，以及在目标页面的onLoad生命周期函数中如何接收参数。通过示例代码，展示数据传递的实现过程。
- 课堂互动（10分钟）**：学生在自己的项目中创建多个页面，并使用路由API实现页面之间的跳转和数据传递功能，如从首页跳转到详情页，并传递数据。教师巡视指导，帮助学生解决遇到的问题，如页面跳转失败、数据传递错误等。
  - 课堂小结（5分钟）**：总结路由API的使用方法、参数配置和页面跳转的注意事项，强调根据项目需求选择合适的路由API进行页面导航和数据传递。
  - 作业布置（5分钟）**：要求学生课后完善页面跳转和数据传递功能，在项目中添加更多的页面和复杂的跳转逻辑，如在多个页面之间进行嵌套跳转，并传递不同类型的数据。

## 第八课时：综合案例 - 智云翻译（一）（45分钟）

- 课程导入（5分钟）**：展示智云翻译应用的界面和功能，引导学生分析其功能需求，如文本翻译、语言选择、翻译历史记录查看等。引出本节课对智云翻译项目的开发讲解，让学生了解如何将之前所学的API知识运用到实际项目中，激发学生的学习热情。

## 2. 知识讲解与演示 (20分钟)

- **项目搭建**: 详细讲解在有道智云AI开发平台注册账号、创建应用并获取应用ID和应用密匙的步骤。演示新建一个使用uni-ui项目的uni-app项目translate-demo的过程, 选择vue2版本, 删除pages下的index文件夹, 新建翻译页面translate.vue、翻译结果页面result.vue、查看翻译历史页面history.vue, 并参照相关内容完成tabbar导航设置。
  - **跨域配置**: 打开manifest.json文件, 在源码视图的末尾添加H5的跨域配置, 强调与前面的项需用“, ”间隔。根据vue的版本, 在main.js文件中添加相应代码, 设置请求的基础路径, 展示如何通过配置实现跨域请求, 让学生理解跨域配置的原理和作用。
  - **配置vuex**: 介绍uni-app中自带vuex插件, 无需独立安装, 可直接引用。在项目下新建store文件夹及index.js文件, 输入模板代码创建store实例, 并在main.js文件中将store注册到全局。解释vuex在项目中的作用, 如集中管理应用的状态, 方便数据共享和状态更新。
3. **课堂互动 (10分钟)**: 学生按照讲解的步骤在自己的开发环境中进行智云翻译项目的前期搭建, 包括创建项目、配置跨域和vuex。教师巡视指导, 帮助学生解决遇到的问题, 如配置文件格式错误、vuex注册失败等。组织学生互相交流遇到的问题和解决方法, 鼓励学生自主探索解决方案。
  4. **课堂小结 (5分钟)**: 总结本节课的重点内容, 包括智云翻译项目的前期搭建流程, 如创建项目、跨域配置和vuex配置的关键步骤和注意事项。强调这些基础配置对后续功能开发的重要性。
  5. **作业布置 (5分钟)**: 要求学生课后复习本节课的内容, 确保项目搭建正确无误。预习下节课要讲解的css样式配置和translate.vue页面的界面搭建相关知识。

## 第九课时: 综合案例 - 智云翻译 (二) (45分钟)

1. **课程导入 (5分钟)**: 回顾上节课智云翻译项目的搭建进度, 强调本节课将继续完成项目的界面搭建和部分功能实现, 让学生明确本节课的学习目标。
2. **知识讲解与演示 (20分钟)**
  - **css样式的配置**: 讲解本案例因使用uni-ui项目, 需要配置相关css样式。将HelloUniapp中的common/uni.css样式文件拷贝到项目的common文件夹, 在app.vue中通过@import导入该文件, 并修改page的背景颜色为aliceblue。展示如何利用uni-ui的index.scss和导入的uni.css样式文件, 对项目整体的布局和样式进行初步设置, 提升页面的美观度。
  - **translate.vue页面的界面搭建**: 详细讲解translate.vue页面的核心代码, 展示如何使用各种组件搭建翻译页面的界面, 包括源语言和目标语言的选择组件、文本输入框、翻译按钮、翻译历史记录查看按钮等。强调组件的布局和样式调整, 以及如何使用flex布局等技术实现页面的响应式设计, 确保在不同设备上都能有良好的显示效果。
  - **实现文本翻译功能**: 讲解安装js-sha256的步骤, 在HBuilderX中打开终端运行命令npm install js-sha256。引导学生参考有道智云平台中的文本翻译js示例代码, 编写translate方法。展示如何在translate.vue页面中获取用户输入的文本, 调用有道智云的翻译接口进行翻译, 并在控制台输出翻译结果, 让学生初步了解翻译功能的实现逻辑。
3. **课堂互动 (10分钟)**: 学生在自己的项目中进行css样式配置和translate.vue页面的界面搭建, 尝试实现文本翻译功能的初步代码编写。教师巡视指导, 帮助学生解决遇到的问题, 如样式冲突、组件使用错误、翻译功能无法实现等。鼓励学生分享自己在实现过程中的思路和技巧, 促进学生之间的学习交流。

- 4. 课堂小结 (5分钟)**：总结本节课的重点内容，包括css样式配置的方法、translate.vue页面的界面搭建要点以及文本翻译功能的初步实现思路。提醒学生注意样式和功能实现过程中的细节问题，如样式优先级、接口参数的正确设置等。
- 5. 作业布置 (5分钟)**：要求学生课后继续完善translate.vue页面的功能，优化界面样式，尝试在翻译成功后将结果展示在页面上，而不仅仅是在控制台输出。同时，思考如何将翻译结果存储起来以便后续查看历史记录。

## 第十课时：综合案例 - 智云翻译 (三) (45分钟)

- 1. 课程导入 (5分钟)**：回顾上节课智云翻译项目的进展，展示一些优秀的翻译应用的完整功能，如翻译结果语音播放、历史记录管理等，引导学生思考如何进一步完善智云翻译项目的功能，引出本节课的学习内容。
- 2. 知识讲解与演示 (20分钟)**
  - **完善翻译功能并存储结果**：讲解在store/index.js中编写代码，用于存储翻译结果和翻译记录。修改translate.vue中的translate方法，在请求成功后，将翻译结果存入store和本地缓存，然后导航至结果页面result.vue。详细展示如何在result.vue中获取store中的翻译结果，并根据翻译结果的类型（单词或句子），使用v-if、v-else指令分别显示不同的内容，如使用uni-ui的折叠面板来显示“词数”、“网络语义”等信息。
  - **显示历史记录**：讲解使用uni-ui中的列表组件显示历史记录的方法。当点击列表项时，打开result.vue页面并传递相应的翻译结果数据，实现查看具体翻译详情的功能。展示如何在翻译页面translate.vue的translate()方法中添加判定代码，如果当前翻译内容与历史记录中的某条记录一致，则直接从历史记录中获取结果，避免重复翻译，提高应用的效率。
  - **项目优化与拓展**：引导学生思考如何进一步优化项目，如添加语音播放功能，让用户可以听取翻译结果。展示如何使用uni.createInnerAudioContext()创建音频上下文，实现语音播放功能。鼓励学生拓展项目功能，如添加更多语言选择、优化翻译结果的展示样式等。
- 3. 课堂互动 (10分钟)**：学生在自己的项目中完成翻译结果存储、历史记录显示和语音播放等功能的实现。教师巡视指导，帮助学生解决遇到的问题，如数据存储和读取错误、语音播放异常等。组织学生进行小组讨论，分享各自优化和拓展项目的思路和方法，共同探讨如何提升项目的质量和用户体验。
- 4. 课堂小结 (5分钟)**：全面总结智云翻译项目的开发过程，回顾从项目搭建、功能实现到优化拓展的各个环节，强调综合运用所学知识解决实际问题的重要性。鼓励学生在今后的项目开发中，不断积累经验，提高自己的开发能力。
- 5. 作业布置 (5分钟)**：要求学生课后对智云翻译项目进行全面测试，检查是否存在功能缺陷或兼容性问题。对项目进行最后的优化和完善，整理项目代码和文档。同时，思考如何将所学的开发技巧应用到其他项目中，如使用百度AI完成一个植物识别页面（可作为拓展练习）。

## 六、教学资源

- 1. 硬件资源**：配备稳定网络环境的计算机机房，每台计算机均安装HBuilderX开发工具，确保其版本能支持uni-app开发及相关API的使用与调试，同时安装相关手机模拟器或准备足够数量的真机设备，方便学生进行移动端的测试与预览。

2. **软件资源**：除HBuilderX外，还需在计算机上安装微信开发者工具、Chrome浏览器等，用于多平台的项目运行与调试。准备有道智云AI开发平台的账号供学生在项目开发中使用，以及相关插件（如js-sha256），方便学生直接进行项目实践。
3. **文档资料**：精心制作与课程内容紧密匹配的教学PPT，包含详细的文字讲解、清晰的代码示例截图、生动的操作演示动画等，辅助课堂教学，帮助学生更好地理解和掌握知识点。整理uni-app官方文档、有道智云翻译API文档以及相关技术博客文章，方便学生在课堂内外随时查阅，深入学习API的高级用法和技术细节。
4. **在线学习平台**：利用在线学习管理系统，如学堂在线、雨课堂等，发布课程资料、作业任务、拓展学习资源等，方便学生在线学习、提交作业和交流讨论。同时，引入技术论坛和开源代码库，如CSDN、GitHub等，让学生能够获取更多的技术支持和项目参考，拓宽技术视野，了解行业最新动态。

## 七、教学反思

1. **教学内容方面**：在教学过程中，部分学生对一些复杂API的理解和应用存在困难，如网络请求API中的参数配置和数据缓存API中的数据管理。后续教学应增加更多实际案例和场景模拟，帮助学生更好地理解API在不同业务场景中的使用方法。对于跨平台开发中API的兼容性问题，虽然在课堂上有所提及，但学生在实际操作中仍容易忽略。后续可专门安排课时，深入讲解不同平台API的差异和兼容性处理方法，通过对比和实际测试，加深学生的理解。
2. **教学方法方面**：讲授法、演示法和实践法相结合的教学方式在一定程度上促进了学生的学习，但在实践环节，部分学生自主学习能力和问题解决能力有待提高。部分学生过于依赖教师的指导，缺乏主动探索和尝试的精神。在今后的教学中，要加强对学生自主学习能力的培养，在实践任务布置时，给予学生更多自主探索的空间，引导学生学会通过查阅文档、参考示例代码等方式解决问题。同时，在课堂讨论环节，设计更具启发性的问题，鼓励学生积极参与讨论，培养学生的批判性思维 and 创新能力。
3. **学生学习效果方面**：从作业和项目实践的完成情况来看，学生在综合运用API进行项目开发时，整体项目的逻辑架构和功能完整性方面存在不足。部分学生能够实现单个API的功能，但在将多个API组合实现复杂功能时，缺乏整体规划和协调能力。后续教学需要增加综合性项目实践的比重，引导学生从项目整体需求出发，合理选择和组合API，注重项目的逻辑架构和功能的完整性。可以组织学生进行小组项目开发，通过团队协作的方式，让学生在交流与合作中提高项目开发能力。
4. **思政教育融入方面**：在教学过程中融入了思政元素，如培养学生的团队协作精神和遵循软件项目开发流程的职业素养，但在融合方式上还不够自然和深入。后续应进一步挖掘课程中的思政元素，将思政教育与专业知识更紧密地结合，例如在讲解项目开发时，引导学生思考如何开发出更具社会价值和创新性的应用，增强学生的社会责任感和创新意识。同时，可以通过展示国产优秀应用案例，让学生了解国内技术发展的成就，激励学生为推动国产技术进步贡献力量。

## 第六章 uni-app常用API - 2教案

---

### 一、课程基本信息

1. **课程名称**：移动应用开发 - uni-app常用API (2)
2. **授课教师**：
3. **授课班级**：

4. **授课时间**：共12课时，每课时45分钟

## 二、教学目标

### 1. 知识目标

- 全面深入理解uni-app中各类常用API（媒体控制、文件操作、设备操作、第三方服务等）的功能、参数和使用场景。
- 熟练掌握自定义组件的创建和使用方法，能够在项目中合理运用组件进行开发。
- 深入理解并掌握“仿网易云”音乐播放器项目的开发流程、技术要点和实现细节。

### 2. 能力目标

- 能够熟练且灵活地运用各种API，结合自定义组件，独立完成复杂功能模块的开发，如“仿网易云”音乐播放器中的音频播放、录音、图片处理、文件管理等功能。
- 具备独立进行项目测试的能力，能够熟练运用测试工具和方法，发现并解决API使用和项目开发过程中出现的问题。
- 能够根据项目需求，合理设计和开发自定义组件，提高代码的复用性和项目的可维护性。

### 3. 素质目标

- 强化学生的团队协作精神，在项目开发过程中，能够与团队成员有效沟通、分工协作，共同完成项目任务。
- 进一步提升学生的软件编码规范素养，使其在代码编写过程中遵循行业标准和规范，提高代码质量。
- 培养学生独立思考、分析问题和解决问题的能力，面对技术难题时能够积极主动地探索解决方案，提升学生的技术能力和职业素养。
- 通过音乐播放器项目的开发，提升学生的音乐素养，引导学生树立正确的审美观、人生观和价值观。

## 三、教学重难点

### 1. 教学重点

- 各类媒体控制API（音频、视频、相机等）的使用方法和实际应用，如实现音频播放、视频播放、拍照等功能。
- 文件操作API（文件保存、选择等）的操作要点和应用场景，确保学生能够正确处理文件相关的功能。
- 设备操作API（获取系统信息、扫码、剪贴板等）的功能实现和使用技巧，帮助学生掌握设备相关功能的开发。
- 第三方服务API（登录等）的原理和使用流程，特别是微信小程序登录的实现步骤和数据交互过程。

- “仿网易云”音乐播放器项目的核心功能实现，如音乐播放、榜单展示、歌曲详情查看等。

## 2. 教学难点

- 不同媒体控制API在跨平台开发中的差异和兼容性处理，如音频和视频播放组件在不同平台的表现差异及适配方法。
- 文件操作API中文件路径的管理和不同平台的存储机制差异，确保文件在不同平台的正确保存和读取。
- 理解并实现第三方服务API（如微信小程序登录）与服务器之间的数据交互和认证机制，处理登录过程中的安全性和稳定性问题。
- 在“仿网易云”音乐播放器项目中，实现复杂的业务逻辑和交互效果，如音乐播放的控制、歌曲切换、评论展示等，提升用户体验。

## 四、教学方法

1. **讲授法**：系统讲解各类API的概念、功能、参数和使用方法，构建知识框架，让学生对所学内容有清晰的认识。
2. **演示法**：在HBuilderX开发环境中，实时演示API的使用过程、代码编写和实际运行效果，让学生直观地了解操作流程和功能实现方式。
3. **实践法**：安排学生进行实际项目开发操作，通过完成“仿网易云”音乐播放器项目的各个功能模块，让学生在实践中巩固知识，提高动手能力和解决实际问题的能力。教师在学生实践过程中进行现场指导，及时解决学生遇到的问题。
4. **讨论法**：组织学生讨论API的应用场景、项目开发中的技术难题及解决方案，促进学生之间的思想交流和经验分享，培养学生的团队协作能力和创新思维。

## 五、教学过程

### 第一课时：媒体控制 - 音频组件控制（45分钟）

1. **课程导入（5分钟）**：展示一些热门音乐播放应用（如网易云音乐、QQ音乐），播放其中的音频内容，引导学生关注音频播放功能。提问学生在开发类似功能时需要考虑哪些方面，从而引出uni-app中音频组件控制API的学习，激发学生的学习兴趣。**\*\*【思政融入】\*\***介绍中国数字音乐产业的发展成就——从早期的盗版泛滥到如今版权保护体系的建立，中国音乐产业走出了一条规范化、法治化的发展道路。引导学生树立知识产权保护意识，尊重原创、保护版权，在开发音乐类应用时自觉遵守著作权法律法规。
2. **知识讲解（20分钟）**
  - **uni.createInnerAudioContext()**：详细讲解该方法的作用，即创建并返回InnerAudioContext对象，用于控制音频播放。逐一介绍InnerAudioContext对象的属性，如src（音频数据链接）、startTime（开始播放位置）、autoplay（是否自动播放）、loop（是否循环播放）等，结合实际音乐播放场景，解释每个属性的含义和使用场景。

## 1. uni.createInnerAudioContext() 创建并返回InnerAudioContext对象。

属性	类型	说明	只读	平台差异说明
src	String	音频的数据链接，用于直接播放。	否	微信小程序不支持本地路径
startTime	Number	开始播放的位置（单位：s），默认 0	否	
autoplay	Boolean	是否自动开始播放，默认 false	否	
loop	Boolean	是否循环播放，默认 false	否	
obeyMuteSwitch	Boolean	是否遵循系统静音开关，当此参数为 false 时，即使用户打开了静音开关，也能继续发出声音，默认值 true	否	微信、百度、字节跳动、飞书、京东、快手（仅IOS）小程序
duration	Number	当前音频的长度（单位：s），src合法时返回，在 onCanplay中获取	是	
currentTime	Number	当前音频的播放位置（单位：s、小数），src合法时返回，保留小数点后 6 位	是	
paused	Boolean	当前是否暂停或停止状态，true 表示暂停或停止，false 表示正在播放	是	
buffered	Number	音频缓冲的时间点，仅保证当前播放时间点到此时间点内容已缓冲。	是	
volume	Number	音量。范围 0-1。	否	

- InnerAudioContext对象的方法：介绍play（播放）、pause（暂停）、stop（停止）、seek（跳转到指定位置）等方法，通过在HBuilderX中编写简单示例代码，展示如何使用这些方法控制音频播放。同时讲解事件监听方法，如onPlay、onPause、onEnded等，让学生了解如何监听音频播放状态的变化。

### InnerAudioContext对象的方法

方法	参数	说明
play		播放（H5端部分浏览器需在用户交互时进行）
pause		暂停
stop		停止
seek	position	跳转到指定位置，单位 s
destroy		销毁当前实例
onCanplay、offCanplay	callback	音频进入可以播放状态，但不保证后面可以流畅播放；offCanplay：取消监听onCanplay事件
onPlay、offPlay	callback	音频播放事件；offPlay：取消监听onPlay事件
onPause、offPause	callback	音频暂停事件；offPause：取消监听onPause事件
onStop、offStop	callback	音频停止事件；offStop取消监听onStop事件
onEnded、offEnded	callback	音频自然播放结束事件；offEnded取消监听onEnded事件
onTimeUpdate、offTimeUpdate	callback	音频播放进度更新事件；offTimeUpdate取消监听onTimeUpdate事件

- uni.getBackgroundAudioManager(): 讲解获取全局唯一的背景音频管理器 BackgroundAudioManager对象实例的方法，介绍该对象的属性和使用场景，如在App后台播放音乐时的应用。通过示例代码展示如何设置音频的标题、歌手、封面图等信息，以及如何控制背景音频的播放。

阅读	类型	说明	只读
duration	Number	当前音频的长度（单位：s），只有在当前有合法的 src 时返回	是
currentTime	Number	当前音频的播放位置（单位：s），只有在当前有合法的 src 时返回	是
paused	Boolean	当前是否暂停或停止状态，true 表示暂停或停止，false 表示正在播放	是
src	String	音频的数据源，默认为空字符串，**当设置了新的 src 时，会自动开始播放，**目前支持的格式有 m4a, aac, mp3, wav	否
startTime	Number	音频开始播放的位置（单位：s）	否
buffered	Number	音频缓冲的时间点，仅保证当前播放时间点到此时间点内容已缓冲。	是
title	String	音频标题，用于做原生音频播放器音频标题。原生音频播放器中的分享功能，分享出去的卡片标题，也将使用该值。	否
epname	String	专辑名，原生音频播放器中的分享功能，分享出去的卡片简介，也将使用该值。	否
singer	String	歌手名，原生音频播放器中的分享功能，分享出去的卡片简介，也将使用该值。	否
coverImgUrl	String	封面图url，用于做原生音频播放器背景图。原生音频播放器中的分享功能，分享出去的卡片配图及背景也将使用该图。	否
webUrl	String	页面链接，原生音频播放器中的分享功能，分享出去的卡片简介，也将使用该值。	否
protocol	String	音频协议。默认值为 'http'，设置 'hls' 可以支持播放 HLS 协议的直播音频，App平台暂不支持	否
playbackRate	Number	播放的倍率。可取值：0.5/0.8/1.0/1.25/1.5/2.0，默认值为1.0。（App 3.4.5+、微信基础库 2.11.0+、支付宝小程序、字节小程序 2.33.0+、快手小程序、百度小程序 3.120.2+）	否

### BackgroundAudioManager对象的属性

- 课堂互动（10分钟）：** 学生在自己的项目中创建一个简单的音频播放页面，使用 uni.createInnerAudioContext()实现音频的播放、暂停和停止功能。教师巡视指导，帮助学生解决遇到的问题，如音频链接无效、方法调用错误等。鼓励学生尝试修改音频的属性，观察播放效果的变化。
- 课堂小结（5分钟）：** 总结本节课所学的音频组件控制API，强调InnerAudioContext对象和BackgroundAudioManager对象的属性、方法及使用场景。提醒学生注意音频播放中可能出现的问题，如音频格式支持、跨平台兼容性等。
- 作业布置（5分钟）：** 要求学生课后完善音频播放页面，添加循环播放、跳转到指定位置等功能，并尝试使用BackgroundAudioManager实现一个简单的后台音频播放功能。

## 第二课时：媒体控制 - 录音管理（45分钟）

- 课程导入（5分钟）：** 展示一些具有录音功能的应用（如微信语音聊天、录音转文字软件），引导学生思考这些应用中录音功能的实现方式。引出本节课对uni-app中录音管理API的学习，让学生了解录音功能在实际开发中的重要性。
- 知识讲解（20分钟）**
  - **uni.getRecorderManager()：** 讲解获取全局唯一的录音管理器recorderManager的方法，介绍录音管理器的各个方法，如start（开始录音）、pause（暂停录音，App暂不支持）、resume（继续录音，App暂不支持）、stop（停止录音）等，详细解释每个方法的参数和使用场景。

方法	参数	说明	平台差异说明
start	options	开始录音	
pause		暂停录音	App 暂不支持
resume		继续录音	App 暂不支持
stop		停止录音	
onStart	callback	录音开始事件	
onPause	callback	录音暂停事件	
onStop	callback	录音停止事件，会回调文件地址	
onResume	callback	监听录音继续事件	
onInterruptionBegin	callback	监听录音因为受到系统占用而被中断开始事件。	微信小程序、百度小程序、QQ小程序
onInterruptionEnd	callback	监听录音中断结束事件。收到此事件之后暂停的小程序内的录音才可再次录音成功。	微信小程序、百度小程序、QQ小程序
onFrameRecorded	callback	已录制完指定帧大小的文件，会回调录音分片结果数据。如果设置了 frameSize，则会回调此事件	App 暂不支持
onError	callback	录音错误事件，会回调错误信息	

- **录音相关事件**：介绍录音过程中的各种事件，如onStart（录音开始事件）、onStop（录音停止事件，会回调文件地址）、onError（录音错误事件）等，通过示例代码展示如何监听这些事件，获取录音过程中的信息。
  - **录音参数设置**：讲解录音时可设置的参数，如duration（录音时长）、sampleRate（采样率）、numberOfChannels（录音通道数）、encodeBitRate（编码码率）、format（音频格式）等，解释这些参数对录音质量和文件大小的影响，让学生了解如何根据实际需求设置录音参数。
3. **课堂互动（10分钟）**：学生在项目中添加录音功能，实现录音的开始、停止，并在停止录音后获取录音文件的路径。教师巡视指导，帮助学生解决问题，如录音权限获取失败、录音文件保存路径错误等。鼓励学生尝试不同的录音参数设置，比较录音效果。
  4. **课堂小结（5分钟）**：总结录音管理API的重点内容，包括录音管理器的获取、方法使用、事件监听和参数设置。强调录音功能在实际项目中的应用场景和注意事项，如录音时长的限制、不同平台的支持差异等。
  5. **作业布置（5分钟）**：要求学生课后优化录音功能，添加录音暂停和继续功能（如果App平台支持），并在页面上显示录音的时长和文件大小。

### 第三课时：媒体控制 - 图片（一）（45分钟）

1. **课程导入（5分钟）**：展示一些应用中涉及图片操作的场景，如微信朋友圈发布图片、电商应用展示商品图片等，引导学生思考这些图片操作在开发中是如何实现的。引出本节课对uni-app中图片相关API的学习。
2. **知识讲解（20分钟）**
  - **uni.chooseImage()**：详细讲解该方法的参数，如count（最多选择图片张数）、sizeType（原图或压缩图）、sourceType（从相册选图或使用相机拍照）、crop（图像裁剪参数）等，通过示例代码展示如何使用该方法从本地相册选择图片或使用相机拍照，并获取图片的本地文件路径列表。强调crop参数在App 3.1.19+版本的使用方法和注意事项。

## 示例代码

```
uni.chooseImage({
  count: 6, //默认9
  sizeType: ['original', 'compressed'], //可以指定
  //是原图还是压缩图，默认二者都有
  sourceType: ['album'], //从相册选择
  success: function (res) {

    console.log(JSON.stringify(res.tempFilePaths));
  }
});
```

- o
- o **uni.previewImage()**: 介绍该方法用于预览多张图片的功能，讲解其参数，如current（当前显示图片链接）、urls（需要预览的图片链接列表）、indicator（图片指示器样式）、loop（是否可循环预览）等。通过示例代码展示如何在选择图片后，使用该方法进行图片预览，并设置不同的参数，观察预览效果。

```
uni.chooseImage({
  count: 6,
  sizeType: ['original', 'compressed'],
  sourceType: ['album'],
  success: function(res) {
    // 预览图片
    uni.previewImage({
      urls: res.tempFilePaths,
      longPressActions: {
        itemList: ['发送给朋友', '保存图片', '收藏'],
        success: function(data) {
          console.log('选中了第' + (data.tapIndex + 1) + '个按钮,
第' + (data.index + 1) + '张图片');
        },
        fail: function(err) {
          console.log(err.errMsg);
        }
      }
    });
  }
});
```

## 示例代码

3. **课堂互动（10分钟）**：学生在项目中实现选择图片并预览的功能，教师巡视指导，帮助学生解决遇到的问题，如图片选择失败、预览图片不显示等。鼓励学生尝试设置不同的参数，实现个性化的图片选择和预览功能。
4. **课堂小结（5分钟）**：总结本节课学习的图片相关API，重点回顾uni.chooseImage()和uni.previewImage()的参数和使用方法。强调在实际项目中，根据需求合理设置参数，实现符合用户体验的图片操作功能。
5. **作业布置（5分钟）**：要求学生课后完善图片选择和预览功能，添加图片裁剪功能（如果App版本支持），并在预览图片时设置长按保存图片的操作。

## 第四课时：媒体控制 - 图片（二）（45分钟）

1. **课程导入（5分钟）**：回顾上节课学习的图片选择和预览功能，提问学生在获取图片后还可能需要进行哪些操作，如获取图片信息、保存图片、压缩图片等，从而引出本节课继续学习图片相关API的内容。

## 2. 知识讲解 (20分钟)

- **uni.getImageInfo()**: 讲解该方法用于获取图片信息的功能, 包括图片的宽度、高度、路径等。介绍其参数src (图片路径) 以及成功回调函数中返回的参数信息, 通过示例代码展示如何在选择图片后, 使用该方法获取图片的详细信息。强调网络图片需先配置download域名才能生效。
- **uni.saveImageToPhotosAlbum()**: 介绍保存图片到相册的方法, 讲解其参数filePath (图片文件路径) 以及成功回调函数。通过示例代码展示如何将选择的图片保存到相册, 并处理保存成功或失败的情况。

```
uni.chooseImage({
  count: 1,
  sourceType: ['camera'],
  success: function (res) {
    uni.saveImageToPhotosAlbum({
      filePath: res.tempFilePaths[0],
      success: function () {
        console.log('save success');
      }
    });
  }
});
```

示例代码

- **uni.compressImage()**: 讲解压缩图片的方法, 介绍其参数src (图片路径)、quality (压缩质量) 以及成功回调函数中返回的压缩后图片的临时路径。通过示例代码展示如何对图片进行压缩, 并根据需求设置不同的压缩质量。

```
uni.compressImage调用成功后会返回压缩成功的临时路径tempFilePath
```

```
uni.compressImage({
  src: '/static/logo.jpg',
  quality: 80,
  success: res => {
    console.log(res.tempFilePath)
  }
});
```

示例代码

3. **课堂互动 (10分钟)**: 学生在项目中实现获取图片信息、保存图片到相册和压缩图片的功能。教师巡视指导, 帮助学生解决遇到的问题, 如获取图片信息失败、保存图片权限不足、压缩图片格式不支持等。鼓励学生尝试不同的图片路径和压缩质量设置, 观察效果。
4. **课堂小结 (5分钟)**: 总结本节课学习的图片相关API, 包括获取图片信息、保存图片到相册和压缩图片的方法和注意事项。强调在实际项目中, 合理运用这些API可以优化图片资源的管理和使用。

5. **作业布置 (5分钟)**：要求学生课后将图片相关功能整合到一个页面中，实现一个完整的图片处理模块，包括选择图片、预览、获取信息、保存到相册和压缩等功能。

## 第五课时：媒体控制 - 视频组件控制与相机组件控制 (45分钟)

1. **课程导入 (5分钟)**：展示一些包含视频播放和相机拍照功能的应用，如抖音、腾讯视频等，引导学生思考这些功能在开发中的实现方式，引出本节课对视频组件控制和相机组件控制API的学习。
2. **知识讲解 (20分钟)**
  - **视频组件控制**：介绍视频相关的API，如uni.createVideoContext()创建视频上下文对象，以及videoContext对象的方法，如play (播放)、pause (暂停)、seek (跳转到指定位置)、sendDanmu (发送弹幕)等。通过示例代码展示如何在页面中嵌入视频，并实现视频的播放控制和弹幕发送功能。
  - **相机组件控制**：讲解uni.createCameraContext()创建相机上下文对象，以及CameraContext对象的方法，如takePhoto (拍照)、startRecord (开始录像)、stopRecord (结束录像)等。通过示例代码展示如何使用相机进行拍照和录像，并获取相应的图片或视频路径。强调不同平台对相机功能的支持差异。
3. **课堂互动 (10分钟)**：学生在项目中实现视频播放和相机拍照功能，教师巡视指导，帮助学生解决遇到的问题，如视频播放卡顿、相机权限获取失败、拍照或录像功能无法正常使用时。鼓励学生尝试不同的视频和相机操作，体验API的功能。
4. **课堂小结 (5分钟)**：总结视频组件控制和相机组件控制API的重点内容，包括创建上下文对象、对象的方法及使用场景。强调在实际项目中，要根据不同平台的特点和需求，合理使用这些API。
5. **作业布置 (5分钟)**：要求学生课后完善视频播放和相机拍照功能，在视频播放页面添加倍速播放、全屏播放等功能，在相机拍照页面添加图片预览和保存功能。

## 第六课时：文件操作 (45分钟)

1. **课程导入 (5分钟)**：展示一些涉及文件操作的应用场景，如文件管理应用、办公软件的文件保存和打开功能等，引导学生思考这些功能在uni-app开发中的实现方式，引出本节课对文件操作API的学习。
2. **知识讲解 (20分钟)**
  - **文件保存**：讲解uni.saveFile()、uni.saveImageToPhotosAlbum()和uni.saveVideoToPhotosAlbum()等方法，详细介绍它们的参数和使用场景。强调tempFilePath和filePath的区别，以及不同平台对这些方法的支持情况。通过示例代码展示如何将文件保存到本地或系统相册。
  - **文件选择**：介绍uni.chooseFile() (仅H5支持)、uni.chooseImage()、uni.chooseVideo()和uni.chooseMedia()等方法用于选择文件、图片和视频的功能，讲解它们的参数和返回值。通过示例代码展示如何在页面中实现文件选择功能，并获取选择文件的路径。

◦

- ◆ **uni.chooseFile(OBJECT)** : 选择文件但是仅H5支持
- ◆ **uni.chooseImage(OBJECT)** : 从本地相册选择图片或使用相机拍照

**示例代码:**

```
uni.chooseImage({
  count: 1,
  sourceType: ['camera'],
  success: function (res) {
    uni.saveImageToPhotosAlbum({
      filePath: res.tempFilePaths[0],
      success: function () {
        console.log('save success');
      }
    });
  }
});
```

3. **课堂互动 (10分钟)** : 学生在项目中实现文件保存和选择功能, 教师巡视指导, 帮助学生解决遇到的问题, 如文件保存路径错误、文件选择失败、不同平台兼容性问题等。鼓励学生尝试在不同平台上测试文件操作功能。
4. **课堂小结 (5分钟)** : 总结文件操作API的重点内容, 包括文件保存和选择的方法及注意事项。强调在实际项目中, 要注意文件路径的管理和不同平台的差异, 确保文件操作的正确性和稳定性。
5. **作业布置 (5分钟)** : 要求学生课后优化文件操作功能, 在保存文件时添加进度提示, 在选择文件后对文件进行简单的信息展示 (如文件大小、类型等)。

## 第七课时: 设备操作 - 得到系统信息与扫码 (45分钟)

1. **课程导入 (5分钟)** : 展示一些根据设备信息提供不同功能的应用, 如根据设备屏幕大小调整页面布局的应用, 以及使用扫码功能的应用 (如支付宝扫码支付), 引导学生思考这些功能的实现原理, 引出本节课对设备操作API的学习。\*\*【思政融入】\*\*以扫码支付为例, 介绍中国移动支付在全球的领先地位——中国已成为全球移动支付普及率最高的国家, 支付宝和微信支付改变了亿万人的生活方式, 这是中国数字经济领跑世界的缩影。引导学生增强民族自信, 同时认识到技术创新应以便民惠民为出发点, 践行“科技为民”的理念。
2. **知识讲解 (20分钟)**
  - **得到系统信息**: 讲解uni.getSystemInfo()和uni.getDeviceInfo()方法, 介绍它们分别可以获取的系统信息和设备基础信息, 如设备型号、操作系统版本、屏幕宽度高度等。通过示例代码展示如何使用这两个方法获取信息, 并在控制台输出结果。
  - **扫码**: 介绍uni.scanCode()方法, 讲解其参数, 如onlyFromCamera (是否只能从相机扫码)、scanType (扫码类型) 等, 以及扫码成功后的回调函数中返回的参数信息, 如result (扫码内容)、scanType (扫码类型) 等。通过示例代码展示如何调起扫码界面, 并处理扫码结果。
3. **课堂互动 (10分钟)** : 学生在项目中实现获取系统信息和扫码功能, 教师巡视指导, 帮助学生解决遇到的问题, 如获取系统信息失败、扫码功能无法正常使用等。鼓励学生尝试在不同设备上测试扫码功能, 观察返回结果的差异。
4. **课堂小结 (5分钟)** : 总结获取系统信息和扫码API的重点内容, 包括方法的使用和返回参数的含义。强调在实际项目中, 合理利用这些API可以为用户提供更好的个性化体验和便捷的操作方式。

- 作业布置 (5分钟)**：要求学生课后利用获取到的系统信息，根据设备屏幕宽度动态调整页面布局，例如，当屏幕宽度小于某一数值时，调整元素的排列方式或隐藏部分元素。对于扫码功能，要求学生思考并尝试将扫码结果用于实际业务逻辑，如扫码获取商品信息后展示商品详情。

## 第八课时：设备操作 - 剪贴板与其他设备相关API (45分钟)

- 课程导入 (5分钟)**：展示一些使用剪贴板功能的场景，如复制文本内容后在其他应用中粘贴，以及一些涉及设备传感器等其他设备相关功能的应用（如计数器应用使用加速度传感器），引导学生思考这些功能在uni-app中的实现方式，引出本节课对剪贴板及其他设备相关API的学习。

### 2. 知识讲解 (20分钟)

- 剪贴板操作**：详细讲解uni.setClipboardData()和uni.getClipboardData()方法，介绍它们的参数和使用场景。通过示例代码展示如何设置剪贴板内容，并在其他地方获取该内容，强调在不同平台上可能出现的差异及注意事项。

**uni.setClipboardData(OBJECT)：用来设置剪贴板的内容。**

参数名	类型	必填	说明	差异说明
data	String	是	需要设置的内容	
showToast	Boolean	否	配置是否弹出提示，默认弹出提示	App、H5 (3.2.13+)
success	Function	否	接口调用成功的回调	
fail	Function	否	接口调用失败的回调函数	
complete	Function	否	接口调用结束的回调函数	

**setClipboardData(OBJECT) 参数说明**

- 其他设备相关API (拓展介绍)**：简要介绍一些其他设备相关的API，如uni.getBatteryInfo()获取设备电池信息、uni.getNetworkType()获取网络类型等（可根据实际情况选择介绍）。讲解这些API的基本使用方法和返回数据的含义，让学生了解设备操作API的多样性。
- 课堂互动 (10分钟)**：学生在项目中实现剪贴板功能，如创建一个简单的文本复制粘贴示例。教师巡视指导，帮助学生解决遇到的问题，如设置剪贴板内容失败、获取剪贴板数据异常等。鼓励学生尝试探索其他设备相关API，并在控制台输出获取到的信息。
  - 课堂小结 (5分钟)**：总结剪贴板及其他设备相关API的重点内容，包括方法的使用要点和注意事项。强调这些API在丰富应用功能和提升用户体验方面的作用。
  - 作业布置 (5分钟)**：要求学生课后在项目中添加一个功能，利用剪贴板实现文本内容的快速分享（例如，将一段文本复制到剪贴板后，可在其他支持粘贴的应用中分享），并尝试使用至少一种其他设备相关API，根据获取到的信息进行相应的页面展示或功能实现。

## 第九课时：第三方服务 - 登录 (45分钟)

- 课程导入 (5分钟)**：展示多个常见应用的登录界面，如微信、淘宝等，引导学生思考登录功能的重要性以及实现方式的多样性，引出uni-app中第三方服务登录API的学习，特别是微信小程序登录的实现。  
**\*\*【思政融入】\*\***在讲解第三方登录时，引入数据安全与用户隐私保护意识——登录过程涉及用户的身份认证信息，一旦泄露将造成严重后果。介绍我国在网络安全领域的立法进程，强调开发者有责任保护用

户账号安全，不得过度收集用户信息、不得擅自共享用户数据。引导学生树立"数据安全无小事"的责任意识，做用户信息的忠实守护者。

## 2. 知识讲解 (20分钟)

- **第三方服务登录概述**：介绍uni-app提供的第三方服务登录API的作用和常见的第三方登录方式，如微信账号登录、QQ账号登录等。讲解第三方登录的优势，如简化用户注册流程、提高用户粘性等。
- **微信小程序登录流程**：详细讲解微信小程序登录的具体步骤，从使用uni.login方法获取登录凭证code值，到将code值与AppId、Appsecret、grant\_type等参数发送到服务器，请求微信接口获取openid和session\_key，再到服务器生成自己的sessionId并保存到本地缓存，以及后续如何维护登录状态。结合流程图和示例代码，帮助学生理解整个登录过程中的数据交互和逻辑关系。

3. **课堂互动 (10分钟)**：学生分组讨论微信小程序登录流程中的关键环节和可能遇到的问题，如code值的有效期、服务器与微信接口通信的安全性等。每组推选代表分享讨论结果，教师进行点评和补充，解答学生的疑问。

4. **课堂小结 (5分钟)**：总结第三方服务登录API的重点内容，特别是微信小程序登录的流程和要点。强调在实际项目中，要注重登录过程中的数据安全和用户体验。

5. **作业布置 (5分钟)**：要求学生课后阅读微信开放平台文档，深入了解微信登录相关的技术细节。尝试在项目中搭建微信小程序登录的基本框架，不要求完整实现，但要梳理清楚各个环节的逻辑关系和代码结构。

## 第十课时：案例：仿网易云播放器（一）（45分钟）

1. **课程导入 (5分钟)**：展示完整的网易云音乐应用界面，介绍其主要功能模块，如音乐榜单、歌单、歌曲播放、评论等，激发学生的兴趣，引出本节课对仿网易云播放器项目的开发讲解。

## 2. 知识讲解 (20分钟)

- **项目搭建**：讲解创建仿网易云播放器项目的步骤，包括新建uni-app项目、配置项目目录结构、引入必要的组件和样式库等。展示如何使用uni-ui组件库来快速搭建页面结构，以及如何进行项目的基本配置，如设置页面标题、图标等。
- **音乐榜单页面实现**：分析音乐榜单页面的布局和数据需求，讲解如何使用列表组件展示音乐榜单信息，如新歌榜、原创榜、热歌榜等。介绍如何从模拟数据或API接口获取榜单数据，并绑定到页面组件上进行展示。通过示例代码展示数据请求和页面渲染的过程。

3. **课堂互动 (10分钟)**：学生根据讲解在自己的项目进行仿网易云播放器项目的搭建和音乐榜单页面的初步开发，教师巡视指导，帮助学生解决遇到的问题，如项目初始化失败、组件引入错误、数据请求异常等。鼓励学生互相交流，分享开发过程中的经验和问题。

4. **课堂小结 (5分钟)**：总结本节课仿网易云播放器项目的搭建和音乐榜单页面开发的重点内容，包括项目创建的步骤和榜单页面数据展示的实现方法。提醒学生注意代码的规范性和页面布局的合理性。

5. **作业布置 (5分钟)**：要求学生课后优化音乐榜单页面，添加点击榜单歌曲跳转到歌曲详情页面的功能，同时思考如何进一步完善榜单页面的样式，使其更接近网易云音乐的风格。

## 第十一课时：案例：仿网易云播放器（二）（45分钟）

1. **课程导入 (5分钟)**：回顾上节课仿网易云播放器项目的进展，展示歌曲详情页面和音乐播放功能的效果，引出本节课继续完善仿网易云播放器项目的内容。

## 2. 知识讲解 (20分钟)

- **歌曲详情页面实现**：分析歌曲详情页面的布局和功能需求，讲解如何展示歌曲的详细信息，如歌曲名称、歌手、专辑封面、歌词等。介绍如何获取歌曲的歌词数据（可通过模拟数据或歌词API获取），并实现歌词的滚动显示效果。展示如何使用音频播放API实现歌曲的播放功能，包括歌曲的播放、暂停、切换等操作。
- **音乐播放控制**：深入讲解音乐播放控制的逻辑实现，如如何记录歌曲的播放状态、实现歌曲的顺序播放和随机播放等功能。介绍如何使用音频播放事件（如播放结束事件）来实现歌曲的自动切换。通过示例代码展示音乐播放控制的核心代码逻辑。

3. **课堂互动 (10分钟)**：学生在项目中实现歌曲详情页面和音乐播放控制功能，教师巡视指导，帮助学生解决遇到的问题，如歌词显示异常、音乐播放卡顿、播放控制逻辑错误等。鼓励学生尝试优化音乐播放的体验，如添加加载动画、优化歌词滚动效果等。

4. **课堂小结 (5分钟)**：总结本节课仿网易云播放器项目中歌曲详情页面和音乐播放控制功能的实现要点，强调代码逻辑的合理性和交互体验的重要性。

5. **作业布置 (5分钟)**：要求学生课后进一步完善音乐播放功能，添加歌曲收藏、播放历史记录等功能，同时优化页面的交互效果，如在歌曲切换时添加过渡动画。

## 第十二课时：案例：仿网易云播放器（三）与项目总结（45分钟）

1. **课程导入 (5分钟)**：回顾仿网易云播放器项目的整体开发过程，展示一些优秀音乐应用的进阶功能，如评论功能、个性化推荐等，引出本节课对仿网易云播放器项目的最后完善和总结。

## 2. 知识讲解与实践指导 (20分钟)

- **评论功能实现 (可选拓展)**：如果时间允许，讲解如何实现歌曲评论功能。介绍评论数据的存储方式（如使用数据库或本地存储模拟），以及如何在页面上展示评论列表和添加新评论的功能。展示评论功能的关键代码实现，如评论数据的提交和获取。
- **项目优化与测试**：指导学生对整个仿网易云播放器项目进行优化，包括性能优化（如减少数据加载时间、优化页面渲染）、代码优化（如提高代码的可读性和可维护性）。讲解项目测试的方法和重要性，如功能测试、兼容性测试等，引导学生使用不同的设备和平台进行测试，发现并解决潜在的问题。

3. **课堂互动 (10分钟)**：学生进行项目的优化和测试，教师巡视指导，帮助学生解决遇到的问题，如评论功能无法正常使用、项目在某些设备上出现兼容性问题等。组织学生分享项目优化和测试过程中的经验和发现的问题，共同探讨解决方案。

4. **课堂总结 (5分钟)**：全面总结仿网易云播放器项目的开发过程，回顾所使用的各类API和技术要点，强调在项目开发过程中遇到问题时积极探索和解决问题的重要性。鼓励学生在今后的项目开发中，不断提升自己的技术能力和创新思维。

5. **作业布置 (5分钟)**：要求学生整理仿网易云播放器项目的代码和文档，包括项目的功能介绍、技术实现细节、遇到的问题及解决方案等。鼓励学生将项目进一步拓展，如添加更多个性化功能，提升用户体验。

## 六、教学资源

- 硬件资源：**配备稳定网络环境的计算机机房，每台计算机均安装HBuilderX开发工具，确保其版本能支持uni-app开发及相关API的使用与调试，同时安装相关手机模拟器或准备足够数量的真机设备，方便学生进行移动端的测试与预览。
- 软件资源：**除HBuilderX外，还需在计算机上安装微信开发者工具、Chrome浏览器等，用于多平台的项目运行与调试。准备相关的测试账号（如微信测试号），方便学生进行第三方服务API（如微信登录）的开发测试。同时，提供一些音乐、图片、视频等素材资源，用于仿网易云播放器项目的开发，可从免费素材网站获取或自行制作。
- 文档资料：**精心制作与课程内容紧密匹配的教学PPT，包含详细的文字讲解、清晰的代码示例截图、生动的操作演示动画等，辅助课堂教学，帮助学生更好地理解和掌握知识点。整理uni-app官方文档、微信开放平台文档以及相关技术博客文章，方便学生在课堂内外随时查阅，深入学习API的高级用法和技术细节。
- 在线学习平台：**利用在线学习管理系统，如学堂在线、雨课堂等，发布课程资料、作业任务、拓展学习资源等，方便学生在线学习、提交作业和交流讨论。同时，引入技术论坛和开源代码库，如CSDN、GitHub等，让学生能够获取更多的技术支持和项目参考，拓宽技术视野，了解行业最新动态。

## 七、教学反思

- 教学内容方面：**在教学过程中，部分学生对复杂API的理解和应用存在困难，如微信小程序登录过程中的数据交互和认证机制。后续教学应增加更多实际案例和场景模拟，帮助学生更好地理解API在不同业务场景中的使用方法。对于跨平台开发中API的兼容性问题，虽然在课堂上有所提及，但学生在实际操作中仍容易忽略。后续可专门安排课时，深入讲解不同平台API的差异和兼容性处理方法，通过对比和实际测试，加深学生的理解。
- 教学方法方面：**讲授法、演示法和实践法相结合的教学方式在一定程度上促进了学生的学习，但在实践环节，部分学生自主学习能力和问题解决能力有待提高。部分学生过于依赖教师的指导，缺乏主动探索和尝试的精神。在今后的教学中，要加强对学生自主学习能力的培养，在实践任务布置时，给予学生更多自主探索的空间，引导学生学会通过查阅文档、参考示例代码等方式解决问题。同时，在课堂讨论环节，设计更具启发性的问题，鼓励学生积极参与讨论，培养学生的批判性思维 and 创新能力。
- 学生学习效果方面：**从作业和项目实践的完成情况来看，学生在综合运用API进行项目开发时，整体项目的逻辑架构和功能完整性方面存在不足。部分学生能够实现单个API的功能，但在将多个API组合实现复杂功能时，缺乏整体规划和协调能力。后续教学需要增加综合性项目实践的比重，引导学生从项目整体需求出发，合理选择和组合API，注重项目的逻辑架构和功能的完整性。可以组织学生进行小组项目开发，通过团队协作的方式，让学生在交流与合作中提高项目开发能力。
- 思政教育融入方面：**在教学过程中融入了思政元素，如培养学生的团队协作精神和遵循软件项目开发流程的职业素养，但在融合方式上还不够自然和深入。后续应进一步挖掘课程中的思政元素，将思政教育与专业知识更紧密地结合，例如在讲解项目开发时，引导学生思考如何开发出更具社会价值和创新性的应用，增强学生的社会责任感和创新意识。同时，可以通过展示国产优秀应用案例，让学生了解国内技术发展的成就，激励学生为推动国产技术进步贡献力量。

## 第七章 智慧环保项目

---

## 一、课程基本信息

1. **课程名称:** 移动应用开发 - 智慧环保项目实战
2. **授课教师:**
3. **授课班级:**
4. **授课时间:** 共12课时, 每课时45分钟

## 二、教学目标

### 1. 知识目标

- 深入理解uni-app开发流程, 掌握组件、API和扩展组件uView的综合运用。
- 熟悉智慧环保项目的业务逻辑和功能模块, 理解项目中各页面之间的关系和数据交互方式。
- 掌握json-server的安装、配置及使用, 了解如何利用其模拟后端数据接口。

### 2. 能力目标

- 能够熟练运用uView组件库和uni-app技术, 独立完成智慧环保项目各页面的开发与功能实现。
- 学会根据项目需求进行页面布局和设计, 使用flex布局等技术实现页面的响应式设计。
- 具备项目调试和优化能力, 能够解决开发过程中出现的常见问题, 提高项目的稳定性和性能。

### 3. 素质目标

- 培养学生绿色环保、可持续发展的理念, 增强社会责任感。
- 提升学生团队协作、沟通交流的能力, 在项目开发中学会分工合作, 共同完成项目任务。
- 培养学生自主学习和探索创新的精神, 面对新技术和问题时, 能够主动学习和尝试解决。

## 三、教学重难点

### 1. 教学重点

- 智慧环保项目各页面的功能实现, 包括首页、订单页、公司回收页、个人中心页等。
- uView组件的使用, 如搜索组件、表单组件、轮播组件等, 以及如何根据项目需求选择合适的组件。
- 项目中的数据请求和处理, 包括与json-server模拟后端的数据交互, 以及数据在页面中的展示和更新。

### 2. 教学难点

- 理解和实现项目中的业务逻辑, 如订单状态的管理、回收分类的层级展示、用户登录与注册的逻辑等。
- 解决跨页面数据传递和共享的问题, 确保各页面之间数据的一致性和准确性。

- 处理项目开发中的兼容性问题，确保项目在不同平台（如微信小程序、H5、APP）上的正常运行。

## 四、教学方法

1. **项目驱动法**：以智慧环保项目为驱动，引导学生在完成项目的过程中学习和应用知识，提高学生的实践能力和解决实际问题的能力。
2. **讲授法**：讲解项目相关的理论知识、技术要点和开发流程，帮助学生构建知识体系，理解项目开发的核心内容。
3. **演示法**：在课堂上通过实际操作和演示，展示项目开发的过程和效果，让学生更直观地了解如何运用技术实现项目功能。
4. **讨论法**：组织学生讨论项目开发中遇到的问题、解决方案和优化思路，促进学生之间的思想交流和经验分享，培养学生的团队协作能力和创新思维。

## 五、教学过程

### 第一课时：项目介绍（45分钟）

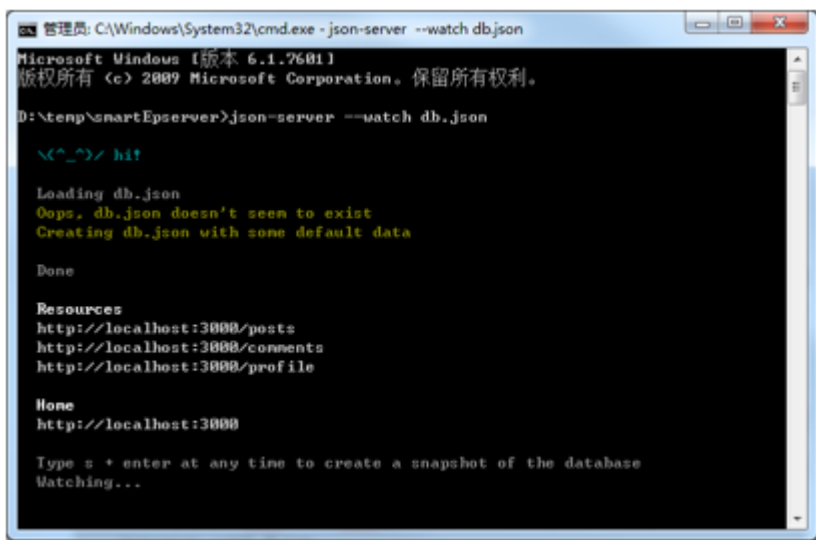
1. **课程导入（5分钟）**：展示当前城市化发展带来的环境问题相关图片或视频，引出智慧环保项目的背景和意义。介绍本项目如何利用信息技术解决环保相关问题，激发学生的学习兴趣 and 参与热情。**\*\*【思政融入】\*\***深入阐述“绿水青山就是金山银山”的生态文明理念——介绍我国在环境保护领域取得的重大成就，从“大气十条”到“双碳”目标，中国以实际行动践行对全球气候治理的承诺。引导学生认识到，智慧环保项目不仅是一次技术实践，更是用信息技术服务国家生态文明建设战略的生动体现，激励学生将专业技能与国家需求相结合，做生态文明建设的参与者和推动者。
2. **知识讲解（20分钟）**
  - **项目概述**：详细讲解智慧环保项目包含的各个页面及其功能，如首页展示环境宣传信息和快捷功能、订单页用于用户查看和管理订单等。结合项目效果图片，让学生对项目的整体架构和功能有初步认识。
  - **项目效果展示**：逐一展示项目各页面的设计效果，分析页面布局和交互方式，引导学生思考如何实现这些功能，为后续的开发教学做铺垫。



3. **课堂互动 (10分钟)**：组织学生分组讨论，分析每个页面可能涉及到的技术点和实现思路。每组推选代表分享讨论结果，教师进行点评和补充，引导学生进一步理解项目需求。
4. **课堂小结 (5分钟)**：总结本节课的重点内容，强调项目各页面功能和效果的重要性，让学生对智慧环保项目有清晰的整体认知。
5. **作业布置 (5分钟)**：要求学生课后查阅资料，了解更多关于智慧环保领域的应用案例，思考这些案例中类似功能的实现方式，为后续课程学习做准备。

## 第二课时：环境配置 (45分钟)

1. **课程导入 (5分钟)**：回顾上节课智慧环保项目的内容，引出本节课需要搭建项目开发环境，介绍json-server在项目中的作用，即模拟后端数据接口，为前端开发提供数据支持。
2. **知识讲解 (20分钟)**
  - **安装json-server**：详细讲解在任意地方新建文件夹并进入命令行模式的操作步骤，执行npm install -g json-server安装json-server。演示启动Web服务器的过程，让学生了解如何创建db.json文件以及默认数据的生成。



```
管理员: C:\Windows\System32\cmd.exe - json-server --watch db.json
Microsoft Windows [版本 6.1.7601]
版权所有 (c) 2009 Microsoft Corporation。保留所有权利。

D:\temp\smartEpserver>json-server --watch db.json

^_^ hit

Loading db.json
Oops, db.json doesn't seem to exist
Creating db.json with some default data

Done

Resources
http://localhost:3000/posts
http://localhost:3000/comments
http://localhost:3000/profile

None
http://localhost:3000

Type s + enter at any time to create a snapshot of the database
Watching...
```

- **配置json-server**：讲解如何指定服务器端口，如将端口改为3004的方法。介绍在项目文件夹中新建package.json文件并写入启动命令的步骤，让学生学会使用“npm run mock”命令启动服务器。同时，讲解在public文件夹中放置图片资源以及修改db.json文件来填充数据的操作，介绍接口列表，让学生了解如何获取不同类型的数据。
3. **课堂互动 (10分钟)**：学生在自己的电脑上按照讲解步骤进行json-server的安装和配置，教师巡视指导，帮助学生解决遇到的问题，如安装失败、端口冲突等。
4. **课堂小结 (5分钟)**：总结本节课的重点内容，包括json-server的安装、配置和接口使用，强调正确配置环境对项目开发的重要性。
5. **作业布置 (5分钟)**：要求学生课后尝试修改db.json文件中的数据，观察接口返回数据的变化，熟悉数据结构 and 接口请求方式。

## 第三课时：创建初始项目 (一) (45分钟)

1. **课程导入 (5分钟)**：回顾json-server的配置，引出本节课开始创建uni-app项目，并进行基础配置。

## 2. 知识讲解 (20分钟)

- **新建项目**：指导学生使用默认模板新建一个vue2的uni-app项目smartEpApp，讲解新建其他三个tabBar页面（订单页面、公司回收页面、个人中心页）的方法，以及修改pages.json文件实现底部导航栏的配置。
  - **安装插件**：在插件市场搜索并安装前端组件uView2.0，强调uView依赖scss插件，若未安装需同时安装“scss/sass编译”插件。讲解安装完成后项目目录结构的变化，让学生了解uView组件的存放位置。
3. **课堂互动 (10分钟)**：学生按照步骤进行项目新建和插件安装，教师巡视指导，解决学生遇到的问题，如项目创建失败、插件安装异常等。
  4. **课堂小结 (5分钟)**：总结本节课的操作要点，包括项目新建、tabBar页面创建和插件安装，确保学生掌握项目初始化的基本流程。
  5. **作业布置 (5分钟)**：要求学生课后熟悉新建项目的目录结构，了解各文件的作用，为后续的项目开发做准备。

## 第四课时：创建初始项目（二）（45分钟）

1. **课程导入 (5分钟)**：回顾上节课项目初始化的内容，引出本节课继续进行项目的配置和页面文件路径规划。
2. **知识讲解 (20分钟)**
  - **配置uView**：详细讲解引入uView主js库、主题文件、样式文件以及配置easycom的步骤和作用。强调各配置在项目中的重要性，如引入主js库使uView组件能够正常使用，配置easycom实现组件的自动引入。
  - **配置全局baseUrl**：在项目根目录下新建common文件夹和app.js文件，设置全局baseUrl，并在main.js中进行全局挂载，让学生理解这样做可以方便统一管理服务器地址，便于后续数据请求。
  - **应用页面的文件路径规划**：介绍项目中各页面的文件路径安排，解释这样规划的目的，如方便分包处理、提高项目的可维护性等。
3. **课堂互动 (10分钟)**：学生进行uView配置和全局baseUrl设置，教师巡视指导，解决学生遇到的问题，如文件路径错误、配置不生效等。
4. **课堂小结 (5分钟)**：总结本节课的重点内容，包括uView配置、全局baseUrl设置和页面文件路径规划，让学生明确项目开发前的准备工作已基本完成。
5. **作业布置 (5分钟)**：要求学生课后复习本节课的配置内容，思考uView组件在后续页面开发中的应用场景。

## 第五课时：首页开发（一）（45分钟）

1. **课程导入 (5分钟)**：回顾项目的配置工作，引出本节课开始进行首页的开发，介绍首页在整个项目中的重要性和功能。
2. **知识讲解 (20分钟)**

- **首页布局分析**：讲解首页的六个组成部分，即搜索框、轮播、快捷入口、流程、预约按钮、爱心活动。分析每个部分的功能和设计思路，如搜索框用于用户查找内容，轮播展示重要信息等。
  - **组件使用讲解**：介绍搜索框使用uView的search组件，讲解其双向绑定data属性keyword的作用；轮播使用u-swiper组件，讲解list属性和keyName属性的含义和配置方法；预约按钮使用u-Button组件，介绍color、type、size、shape等属性的设置效果。
  - **数据请求介绍**：说明轮播和爱心活动的数据来源是后端服务器，介绍请求地址，让学生了解数据请求在项目中的作用和基本流程。
3. **课堂互动（10分钟）**：学生在项目中开始搭建首页的基本结构，添加搜索框、轮播和预约按钮组件，教师巡视指导，解决学生遇到的问题，如组件引入错误、属性设置不生效等。
  4. **课堂小结（5分钟）**：总结本节课首页开发的重点内容，包括页面布局分析、组件使用和数据请求的初步了解，让学生对首页开发有清晰的思路。
  5. **作业布置（5分钟）**：要求学生课后完善首页的基本结构，尝试设置轮播和爱心活动的数据请求（可先使用模拟数据），并优化页面样式。

## 第六课时：首页开发（二）（45分钟）

1. **课程导入（5分钟）**：回顾上节课首页开发的进度，引出本节课继续完善首页的功能和细节。
2. **知识讲解（20分钟）**
  - **快捷入口与回收流程**：讲解快捷入口和回收流程使用flex布局的实现方法，介绍u-icon图标的使用，包括name、size、label、labelPos等属性的设置。让学生学会使用flex布局实现页面元素的排列和对齐，以及如何选择合适的图标。
  - **爱心活动数据展示**：讲解爱心活动图片使用u-image组件展示的方法，强调图片url地址需要添加全局baseUrl。介绍如何从后端获取数据并在页面中展示，涉及数据解析和绑定的基本操作。
  - **代码优化与调试**：强调代码的规范性和可维护性，如合理的注释、组件命名规范等。讲解如何进行简单的调试，如使用console.log输出数据，检查页面加载和数据请求是否正常。
3. **课堂互动（10分钟）**：学生完善首页的快捷入口、回收流程和爱心活动部分的代码，教师巡视指导，帮助学生解决遇到的问题，如布局错乱、数据展示异常等。
4. **课堂小结（5分钟）**：总结本节课首页开发的重点内容，包括flex布局、u-icon和u-image组件的使用，以及代码优化和调试的方法。
5. **作业布置（5分钟）**：要求学生课后进一步优化首页的样式和交互效果，如添加动画效果、优化数据加载提示等，提高用户体验。

## 第七课时：回收分类与搜索页面开发（45分钟）

1. **课程导入（5分钟）**：回顾首页开发内容，引出本节课进行回收分类与搜索页面的开发，介绍这两个页面与首页的关系和功能。
2. **知识讲解（20分钟）**

- **页面功能与数据来源**：讲解回收分类页面用于展示垃圾回收分类信息，数据来源于后端，介绍请求地址和数据格式。说明搜索结果页面用于显示搜索关键字相关的分类信息，展示其显示逻辑。
  - **手风琴式导航实现**：介绍回收分类页手风琴式导航效果在uView中的实现方法，可通过导入示例代码获取相关逻辑。讲解如何根据后端数据的parentId属性处理分类的层级关系，涉及数据处理和页面渲染的知识。
  - **搜索功能实现思路**：讲解搜索结果页面的实现思路，包括如何获取搜索关键字、查询数据以及展示结果。介绍getChildList(id)和getTypeList方法在数据处理中的作用，如将子项存入数组和搜索含有关键字的分类项。
3. **课堂互动（10分钟）**：学生在项目中创建回收分类和搜索结果页面，实现基本的数据展示和导航效果，教师巡视指导，解决学生遇到的问题，如数据请求失败、层级关系显示错误等。
  4. **课堂小结（5分钟）**：总结本节课的重点内容，包括回收分类和搜索页面的功能实现、数据处理和页面交互效果的实现方法。
  5. **作业布置（5分钟）**：要求学生课后优化搜索功能，如实现模糊搜索、添加搜索历史记录等功能，提高搜索的准确性和便捷性。

## 第八课时：注册、登录与个人中心页开发（45分钟）

1. **课程导入（5分钟）**：回顾之前页面的开发，引出本节课进行注册、登录和个人中心页的开发，介绍这些页面在项目中的功能和重要性。
2. **知识讲解（20分钟）**
  - **注册页面开发**：讲解注册页面的功能和布局，介绍使用post方法请求注册接口的实现方式。强调在提交前进行js代码验证的重要性，以及随机生成头像的方法。讲解autoLogin()和submitReg()方法的作用和实现逻辑，包括登录成功后存储用户信息和页面跳转。
  - **登录页面开发**：介绍登录页面中u-radio和u-cell组件的使用，讲解单选按钮用于选择用户类型的功能和列表展示其他信息的方式。强调登录成功后将用户信息存入缓存的操作，以及如何在其他页面判断用户登录状态。
  - **个人中心页开发**：讲解个人中心页在未登录和已登录状态下的不同展示效果，介绍点击用户图像跳转到登录界面的实现方法。说明个人中心页作为“我的”页面在项目中的作用和与其他页面的关联。
3. **课堂互动（10分钟）**：学生在项目中进行注册、登录和个人中心页的开发，实现基本的页面功能，教师巡视指导，解决学生遇到的问题，如注册验证失败、登录状态保存异常等。
4. **课堂小结（5分钟）**：总结本节课的重点内容，包括注册、登录和个人中心页的功能实现、组件使用以及用户信息管理的方法。
5. **作业布置（5分钟）**：要求学生课后完善注册、登录和个人中心页的功能，如添加密码强度检测、忘记密码功能，以及在个人中心页展示更多用户相关信息。

## 第九课时：公司回收、公司详情与搜索结果页开发（45分钟）

1. **课程导入（5分钟）**：回顾之前开发的页面，引出本节课进行公司回收、公司详情和搜索结果页的开发，介绍这些页面在项目中的功能和相互关系。

## 2. 知识讲解 (20分钟)

- **公司回收页开发**：讲解公司回收页展示回收公司信息的功能，介绍数据来源和接口请求地址。说明添加scroll-view容器实现页面滚动的方法，以及点击公司信息进入公司详情页的跳转逻辑。
  - **公司详情页开发**：介绍公司详情页展示单个公司详细信息的功能，讲解其API接口地址和数据获取方式。强调从公司回收页跳转到公司详情页时传递参数的方法，以及如何在详情页中展示这些数据。
  - **搜索结果页开发**：讲解在公司回收页和搜索结果页中搜索框的功能和实现方法，介绍搜索接口地址和数据过滤逻辑。说明如何根据搜索关键字显示相关公司信息，以及搜索结果页与其他页面的交互关系。
3. **课堂互动 (10分钟)**：学生在项目中进行公司回收、公司详情和搜索结果页的开发，实现页面的基本功能和数据展示，教师巡视指导，解决学生遇到的问题，如数据加载异常、页面跳转错误等。
  4. **课堂小结 (5分钟)**：总结本节课的重点内容，包括三个页面的功能实现、数据请求和页面跳转的方法，以及搜索功能的实现逻辑。
  5. **作业布置 (5分钟)**：要求学生课后优化这三个页面的样式和交互效果，如在公司详情页添加地图展示公司位置、优化搜索结果的显示方式等。

## 第十课时：下单、订单页与订单详情页开发（一）（45分钟）

1. **课程导入 (5分钟)**：回顾之前开发的页面，引出本节课进行下单、订单页和订单详情页的开发，介绍这些页面在项目订单管理中的重要作用。
2. **知识讲解 (20分钟)**
  - **下单页开发 (uView表单组件使用)**：讲解下单页使用uView表单组件u-form和u-form-item的方法，详细介绍model、labelWidth、labelPosition、rules等u-form属性，以及label、borderBottom、prop等u-form-item属性的含义和作用。介绍物品类型选择使用u-picker组件实现多级关联选择，上门时间选择使用u-Action-Sheet操作菜单组件的方法。结合实际场景，展示如何根据用户选择动态更新表单数据和进行初步的数据校验，如物品重量和件数必须为正整数等。
  - **订单页开发 (数据展示与分段器)**：讲解订单页展示当前用户不同状态订单信息的功能，以获取待接单信息为例，介绍通过get请求获取数据的api接口地址和参数传递方式。详细讲解使用uView中的分段器u-subsection组件实现不同订单状态切换展示的原理，即通过双向绑定current属性，同步控制分段器和轮播组件的当前显示项。同时说明使用组件实现订单列表滚动效果的必要性和配置要点。
3. **课堂互动 (10分钟)**：学生在项目中开始构建下单页和订单页的基本结构，添加表单组件和分段器，教师巡视指导，帮助学生解决遇到的问题，如表单组件数据绑定错误、分段器切换功能异常等。鼓励学生相互交流，分享遇到的问题和解决方法。
4. **课堂小结 (5分钟)**：总结本节课重点内容，包括下单页表单组件的使用要点、订单页数据获取与分段器实现原理，强调组件属性配置和数据交互逻辑在页面开发中的重要性。
5. **作业布置 (5分钟)**：要求学生课后完善下单页的表单校验逻辑，确保用户输入数据的准确性和完整性；继续优化订单页的显示效果，如根据订单状态设置不同的颜色标识。

## 第十一课时：下单、订单页与订单详情页开发（二）（45分钟）

- 课程导入（5分钟）**：回顾上节课下单页和订单页的开发进展，引出本节课继续完善这两个页面的功能，并开发订单详情页。
- 知识讲解（20分钟）**
  - 下单页功能完善与提交逻辑**：讲解下单操作的本质，即通过post请求向服务器添加订单记录，详细介绍请求的api接口地址和需要传递的数据格式。展示如何在用户点击“立即预约”按钮后，收集表单数据，进行必要的格式转换和验证，然后发起post请求，并在请求成功后给出相应的提示信息，如“预约成功”，同时可考虑跳转到订单页。强调处理请求失败的情况，如网络异常或服务器响应错误时，及时向用户反馈错误信息，引导用户进行后续操作。
  - 订单页功能拓展与取消订单实现**：进一步讲解订单页中取消订单功能的实现方法。以json-server服务器为例，说明使用patch请求修改订单状态（将state改为1）的具体步骤，以及在修改成功后如何刷新页面以实时显示最新的订单状态。详细介绍cancelOrder方法的编写逻辑，包括如何获取订单编号、构建请求参数、发起请求以及处理请求结果等。同时，讲解如何在订单列表中添加取消订单按钮，并为其绑定点击事件，触发cancelOrder方法。
  - 订单详情页开发**：介绍订单详情页的布局和功能，其布局与公司详情页类似，但主要侧重于展示订单的详细信息，如订单编号、回收类型、重量、预计收入金额等。讲解如何从订单页传递订单信息到详情页，通常是通过路由参数传递订单编号等唯一标识，然后在订单详情页根据该标识获取订单详细数据并展示。强调订单详情页中取消订单功能的实现与订单页类似，但可能需要根据页面布局 and 用户交互需求进行适当调整。
- 课堂互动（10分钟）**：学生继续完善下单页和订单页的功能，开发订单详情页，教师巡视指导，解决学生遇到的问题，如订单提交失败、取消订单功能异常、订单详情页数据显示不完整等。组织学生进行小组讨论，分享各自在功能实现过程中的思路和遇到的问题，共同探讨解决方案。
- 课堂小结（5分钟）**：总结本节课重点内容，包括下单页提交逻辑、订单页取消订单功能和订单详情页的开发要点，强调页面之间数据传递和功能协同的重要性。
- 作业布置（5分钟）**：要求学生课后对这三个页面进行全面测试，检查各种功能是否正常运行，修复发现的问题。同时，思考如何优化订单管理流程，提升用户体验，如在订单状态变更时实时推送通知给用户。

## 第十二课时：项目优化、总结与拓展（45分钟）

- 课程导入（5分钟）**：回顾整个智慧环保项目的开发过程，强调项目优化和总结的重要性，引出本节课的内容。
- 知识讲解与实践指导（20分钟）**
  - 项目优化**：讲解项目优化的几个方面，如性能优化，包括优化数据请求，减少不必要的数据加载，合理设置缓存机制，提高页面加载速度；代码优化，检查代码的规范性和可读性，去除冗余代码，提取公共代码片段，提高代码的可维护性；界面优化，调整页面布局和样式，使页面更加美观、舒适，提高用户体验。指导学生使用浏览器开发者工具或手机调试工具进行性能分析，查找性能瓶颈，如通过分析加载时间、资源请求等指标，定位加载缓慢的原因，并进行针对性优化。

- **项目总结**：引导学生对整个项目进行全面总结，回顾项目开发过程中遇到的问题和解决方案，如技术难题、团队协作问题等。强调从项目中学习经验教训，提升自己的开发能力和问题解决能力。鼓励学生分享在项目开发过程中的收获和体会，包括技术技能的提升、对环保理念的理解等。
  - **拓展实训项目介绍**：介绍拓展实训项目“时代楷模App”的背景和意义，详细讲解该项目的各个功能模块，如楷模公告、楷模列表、英雄故事、学习心得、公益活动、身边的英雄等。分析每个模块的功能需求和技术要点，引导学生思考如何运用所学知识进行开发，激发学生的拓展实践兴趣。

**【思政融入】**“时代楷模”是新时代最可爱的人，他们中有扎根边疆的支教教师、有冲锋在前的消防战士、有默默奉献的科研工作者。介绍时代楷模的感人事迹，引导学生思考：我们开发“时代楷模App”，不仅是在练习技术，更是在用代码传播正能量、弘扬主旋律，让英雄的故事被更多人知晓。激励学生以时代楷模为榜样，将个人理想融入国家发展大局，在平凡岗位上做出不平凡的贡献。
3. **课堂互动（10分钟）**：学生进行项目优化操作，教师巡视指导，帮助学生解决优化过程中遇到的问题，如性能优化效果不明显、代码优化导致功能异常等。组织学生讨论拓展实训项目的开发思路，鼓励学生积极发言，分享自己的想法和创意。
  4. **课堂小结（5分钟）**：总结本节课内容，强调项目优化和总结对提升开发能力的重要性，鼓励学生在拓展实训项目中积极运用所学知识，不断提升自己的实践能力。
  5. **作业布置（5分钟）**：要求学生课后完成智慧环保项目的优化工作，整理项目文档，包括项目需求分析、技术实现方案、遇到的问题及解决方案等。同时，根据拓展实训项目“时代楷模App”的功能需求，制定初步的开发计划，为后续的实践拓展做准备。

## 六、教学资源

1. **硬件资源**：配备稳定网络环境的计算机机房，每台计算机均安装HBuilderX开发工具，确保其版本能支持uni-app开发及相关组件、插件的使用与调试。同时，准备足够数量的真机设备或安装相关手机模拟器，方便学生进行移动端的测试与预览。
2. **软件资源**：除HBuilderX外，还需在计算机上安装微信开发者工具、Chrome浏览器等，用于多平台的项目运行与调试。提供智慧环保项目的相关素材资源，如图片、图标等，可从免费素材网站获取或自行制作，方便学生在项目开发中使用。
3. **文档资料**：精心制作与课程内容紧密匹配的教学PPT，包含详细的文字讲解、清晰的代码示例截图、生动的操作演示动画等，辅助课堂教学，帮助学生更好地理解和掌握知识点。整理uni-app官方文档、uView组件库文档以及相关技术博客文章，方便学生在课堂内外随时查阅，深入学习技术细节和组件的高级用法。
4. **在线学习平台**：利用在线学习管理系统，如学堂在线、雨课堂等，发布课程资料、作业任务、拓展学习资源等，方便学生在线学习、提交作业和交流讨论。同时，引入技术论坛和开源代码库，如CSDN、GitHub等，让学生能够获取更多的技术支持和项目参考，拓宽技术视野，了解行业最新动态。

## 七、教学反思

1. **教学内容方面**：在教学过程中，部分学生对项目中复杂业务逻辑的理解和实现存在困难，如订单状态管理和回收分类的层级展示。后续教学应增加更多实际案例和场景模拟，帮助学生更好地理解业务逻辑在技术实现中的应用。对于组件和API的综合运用，虽然在课堂上进行了大量实践，但部分学生在面对实际

项目需求时，仍不能灵活选择和组合使用。后续可加强针对性练习，设置更多复杂项目场景，提升学生的综合应用能力。

- 教学方法方面：**项目驱动法、讲授法、演示法和讨论法相结合的教学方式在一定程度上促进了学生的学习，但在实践环节，部分学生自主学习能力和问题解决能力有待提高。部分学生过于依赖教师的指导，缺乏主动探索和尝试的精神。在今后的教学中，要加强对学生自主学习能力的培养，在实践任务布置时，给予学生更多自主探索的空间，引导学生学会通过查阅文档、参考示例代码等方式解决问题。同时，在课堂讨论环节，设计更具启发性的问题，鼓励学生积极参与讨论，培养学生的批判性思维和创新能力。
- 学生学习效果方面：**从作业和项目实践的完成情况来看，学生在项目整体的布局和页面之间的交互设计上存在不足。部分学生能够实现单个页面的功能，但在页面之间的跳转、数据传递和共享方面，缺乏整体规划和协调能力。后续教学需要增加综合性项目实践的比重，引导学生从项目整体需求出发，注重页面布局的合理性和交互的流畅性。可以组织学生进行小组项目开发，通过团队协作的方式，让学生在交流与合作中提高项目开发能力。
- 思政教育融入方面：**在教学过程中融入了绿色环保理念等思政元素，但在融合方式上还不够自然和深入。后续应进一步挖掘课程中的思政元素，将思政教育与专业知识更紧密地结合，例如在讲解项目开发时，引导学生思考如何开发出更具社会价值和环保意义的应用，增强学生的社会责任感和环保意识。同时，可以通过展示环保领域的优秀案例，让学生了解技术在环保事业中的重要作用，激励学生为环保事业贡献自己的力量。

## 第八章 天气预报项目

---

### 一、课程基本信息

- 课程名称：**移动应用开发 - 天气预报项目实战
- 授课教师：**
- 授课班级：**
- 授课时间：**共6课时，每课时45分钟

### 二、教学目标

#### 1. 知识目标

- 掌握天气预报项目开发涉及的uni-app组件、API的运用，包括页面布局组件、数据请求API等。
- 理解天气预报数据的来源和数据格式，学会解析和处理相关数据。
- 熟悉天气图标、城市数据等资源的获取和使用方法。

#### 2. 能力目标

- 能够运用所学技术独立完成天气预报项目的开发，包括页面搭建、功能实现和界面优化。
- 学会根据项目需求进行数据请求和处理，实现实时天气信息展示和城市切换等功能。
- 具备项目调试和优化能力，能够解决开发过程中出现的常见问题，提升应用的稳定性和性能。

### 3. 素质目标

- 培养学生关注生活、利用技术改善生活的意识，让学生体会技术在日常生活中的重要性。
- 提升学生的团队协作能力和沟通能力，通过小组合作完成项目，促进学生之间的交流与合作。
- 培养学生严谨的科学态度和探索精神，在项目开发中不断优化和完善功能。

## 三、教学重难点

### 1. 教学重点

- 天气预报项目各页面的布局和功能实现，如首页展示当前城市天气信息、城市管理页面实现城市添加和删除等。
- 数据请求与处理，包括从天气数据接口获取数据，解析数据并在页面中展示。
- 天气图标展示和城市数据管理，实现根据天气状况显示相应图标，以及城市数据的存储和读取。

### 2. 教学难点

- 理解和处理不同来源天气数据接口的复杂数据格式，确保数据准确解析和展示。
- 实现城市切换时的实时数据更新和页面动态刷新，保证用户体验流畅。
- 优化数据请求性能，避免频繁请求导致的流量消耗和卡顿问题。

## 四、教学方法

- 项目驱动法**：以天气预报项目为驱动，引导学生在完成项目的过程中学习和运用知识，提高实践能力和解决实际问题的能力。
- 讲授法**：讲解项目相关的理论知识、技术要点和开发流程，帮助学生构建知识体系，理解项目开发的核心内容。
- 演示法**：在课堂上通过实际操作和演示，展示项目开发的过程和效果，让学生更直观地了解如何运用技术实现项目功能。
- 讨论法**：组织学生讨论项目开发中遇到的问题、解决方案和优化思路，促进学生之间的思想交流和经验分享，培养学生的团队协作能力和创新思维。

## 五、教学过程

### 第一课时：项目介绍与环境搭建（45分钟）

- 课程导入（5分钟）**：展示日常生活中人们使用天气预报应用的场景，如出行前查看天气、安排活动时参考天气情况等，引出本节课的天气预报项目。介绍该项目的重要性和应用价值，激发学生的学习兴趣。  
**\*\*【思政融入】\*\***介绍中国气象事业的发展成就——从“风云”系列气象卫星的成功发射到世界领先的数值天气预报系统，中国气象事业实现了从跟跑到领跑的历史性跨越。气象服务在防灾减灾、保障人民生命财产安全方面发挥着不可替代的作用。引导学生认识到，开发天气预报应用不仅是技术实践，更是服务民生、守护安全的责任担当，鼓励学生用技术为人民的美好生活保驾护航。
- 知识讲解（20分钟）**

- **项目概述**：讲解天气预报项目的主要功能，包括当前城市天气信息展示（如温度、天气状况、风力等）、城市管理（添加、删除、切换城市）、未来几天天气预测展示等。介绍项目的大致页面布局，如首页展示当前城市天气，侧边栏或二级页面进行城市管理。
- **项目效果展示**：展示已完成的天气预报项目的效果图，包括不同天气状况下的页面展示效果、城市切换的动画效果等，让学生对项目最终呈现的样子有直观的认识。
- **环境搭建**：指导学生创建一个新的uni-app项目，使用默认模板。在项目中创建必要的页面，如首页（index.vue）、城市管理页面（cityManage.vue）。讲解项目目录结构，强调src目录下各文件夹（如pages、static等）的作用。引入字体图标库（例如阿里图标库）用于显示天气图标，将下载好的图标文件放入static文件夹，并在App.vue中引入相关样式：

```
/* App.vue */
<style lang="scss">
  @import url('~@/static/iconfont.css'); /* 引入图标库样式 */
</style>
```

3. **课堂互动（10分钟）**：组织学生分组讨论，分析实现天气预报项目可能涉及的技术点，如数据请求、页面布局、图标展示等。每组推选代表分享讨论结果，教师进行点评和补充，引导学生进一步明确项目开发方向。
4. **课堂小结（5分钟）**：总结本节课的重点内容，包括项目功能、效果和环境搭建要点，让学生对项目有初步的整体认知。
5. **作业布置（5分钟）**：要求学生课后查阅资料，了解常见的天气数据接口，思考如何在项目中获取和使用这些数据。

## 第二课时：页面布局与数据请求（45分钟）

1. **课程导入（5分钟）**：回顾上节课天气预报项目的介绍和环境搭建内容，引出本节课开始进行页面布局和数据请求功能的开发。
2. **知识讲解（20分钟）**
  - **首页布局**：使用uni-app组件搭建天气预报首页布局。在index.vue的中构建基本结构：

```
<template>
  <view class="container">
    <view class="header">
      <text class="city">{{cityName}}</text>
      <text class="date">{{date}}</text>
    </view>
    <view class="weather-info">
      <text class="temperature">{{temperature}}°C</text>
      <text class="weather-condition">{{weatherCondition}}</text>
    </view>
    <view class="icon">
      <text :class="['iconfont', weatherIcon]"></text>
    </view>
```

```
<view class="more-info">
  <text>风力: {{wind}}</text>
  <text>湿度: {{humidity}}%</text>
</view>
</view>
</template>
```

并在中设置样式:

```
<style lang="scss">
.container {
  padding: 20px;
  background-color: #f0f0f0;
}
.header {
  display: flex;
  justify-content: space-between;
  margin-bottom: 20px;
}
.city {
  font-size: 24px;
  font-weight: bold;
}
.date {
  font-size: 16px;
}
.weather-info {
  display: flex;
  justify-content: center;
  align-items: center;
  margin-bottom: 20px;
}
.temperature {
  font-size: 48px;
  margin-right: 20px;
}
.weather-condition {
  font-size: 20px;
}
.icon {
  text-align: center;
  margin-bottom: 20px;
}
.iconfont {
  font-size: 64px;
}
.more-info {
  display: flex;
  justify-content: space-around;
}
</style>
```

讲解flex布局的使用，如通过`display: flex`实现元素的水平排列和对齐方式的调整。

- **数据请求API**：以和风天气API为例（实际使用需申请合法的API Key），在`index.vue`的中编写数据请求代码：

```
<script>
export default {
  data() {
    return {
      cityName: '',
      date: '',
      temperature: '',
      weatherCondition: '',
      weatherIcon: '',
      wind: '',
      humidity: ''
    };
  },
  onLoad() {
    this.fetchWeatherData();
  },
  methods: {
    async fetchWeatherData() {
      const apiKey = 'YOUR_API_KEY'; // 替换为实际API Key
      const city = '北京'; // 可根据实际需求更改
      const url = `https://devapi.qweather.com/v7/weather/now?
location=${city}&key=${apiKey}`;
      try {
        const res = await uni.request({
          url: url,
          method: 'GET'
        });
        if (res.statusCode === 200) {
          const data = res.data;
          this.cityName = data.location.name;
          this.date = new Date().toLocaleDateString();
          this.temperature = data.now.temp;
          this.weatherCondition = data.now.text;
          // 根据天气状况设置天气图标类名，需提前定义好图标映射关系
          this.weatherIcon = this.getWeatherIcon(data.now.text);
          this.wind = data.now.windDir + data.now.windScale + '级';
          this.humidity = data.now.humidity;
        }
      } catch (error) {
        console.error('数据请求失败', error);
      }
    },
    getWeatherIcon(condition) {
      // 简单示例，实际需根据图标库完善
      if (condition.includes('晴')) {
        return 'icon-sunny';
      }
    }
  }
}
```

```

    } else if (condition.includes('多云')) {
      return 'icon-cloudy';
    }
    // 其他天气状况...
    return 'icon-unknown';
  }
}
};
</script>

```

讲解uni.request的使用，包括设置请求的URL、方法，以及处理请求成功和失败的回调函数。展示如何根据接口文档解析返回的数据，提取所需的天气信息。

3. **课堂互动 (10分钟)**：学生在项目中开始搭建首页布局，并尝试编写代码实现数据请求和绑定，教师巡视指导，帮助学生解决遇到的问题，如组件布局错乱、数据请求失败等。鼓励学生相互交流，分享代码编写过程中的经验。

4. **课堂小结 (5分钟)**：总结本节课的重点内容，包括首页布局搭建、数据请求API的使用和数据绑定方法，强调页面布局和数据获取在项目中的重要性。

5. **作业布置 (5分钟)**：要求学生课后优化首页布局，调整页面元素的样式，使其更加美观。同时，完善数据请求部分的代码，处理可能出现的网络异常情况，如请求超时、网络连接失败等。

### 第三课时：天气图标与城市管理功能实现 (45分钟)

1. **课程导入 (5分钟)**：回顾上节课首页布局和数据请求的进展，引出本节课实现天气图标展示和城市管理功能。

#### 2. 知识讲解 (20分钟)

- **天气图标展示**：在前面代码基础上，完善根据天气状况动态切换图标的功能。在index.vue中，`getWeatherIcon`方法可根据不同天气状况返回对应的图标类名：

```

getWeatherIcon(condition) {
  if (condition.includes('晴')) {
    return 'icon-sunny';
  } else if (condition.includes('多云')) {
    return 'icon-cloudy';
  } else if (condition.includes('雨')) {
    return 'icon-rain';
  } else if (condition.includes('雪')) {
    return 'icon-snow';
  }
  return 'icon-unknown';
}

```

确保图标库中已定义好这些图标对应的样式。

- **城市管理功能**：在cityManage.vue页面实现城市管理功能。在中构建页面结构：

```

<template>
  <view class="city-manage-container">
    <view class="add-city">

```

```

    <input v-model="newCity" placeholder="输入城市名称">
    <button @click="addCity">添加城市</button>
  </view>
  <view class="city-list">
    <view v-for="(city, index) in cities" :key="index" class="city-item">
      {{city}}
      <button @click="deleteCity(index)">删除</button>
    </view>
  </view>
</view>
</template>

```

在中编写逻辑代码：

```

<script>
export default {
  data() {
    return {
      newCity: '',
      cities: []
    };
  },
  onLoad() {
    const storedCities = uni.getStorageSync('cities');
    if (storedCities) {
      this.cities = storedCities;
    }
  },
  methods: {
    addCity() {
      if (this.newCity.trim()) {
        this.cities.push(this.newCity);
        uni.setStorageSync('cities', this.cities);
        this.newCity = '';
      }
    },
    deleteCity(index) {
      this.cities.splice(index, 1);
      uni.setStorageSync('cities', this.cities);
    }
  }
};
</script>

```

讲解在uni-app中使用uni.setStorageSync和uni.getStorageSync方法进行本地数据存储和读取，用于保存用户添加的城市列表。展示如何在页面中添加城市和删除城市，并同步更新本地存储。3. **课堂互动（10分钟）**：学生在项目中实现天气图标展示和城市管理功能，教师巡视指导，帮助学生解决遇到的问题，如图标显示异常、城市数据存储和读取错误等。组织学生讨论在实现城市管理功能时遇到的问题和解决方案，促进学生之间的交流。4. **课堂小结（5分钟）**：总结本节课的重点内容，包括天气图标展示和城市管理功能的实现方法，以及本

地数据存储与读取的操作要点。5. **作业布置 (5分钟)**：要求学生课后完善城市管理功能，如添加城市时进行输入校验，防止用户输入无效城市名称；优化城市切换的动画效果，提升用户体验。

#### 第四课时：未来天气预测展示与页面优化 (45分钟)

1. **课程导入 (5分钟)**：回顾上节课天气图标和城市管理功能的开发情况，引出本节课实现未来天气预测展示和页面优化。

#### 2. 知识讲解 (20分钟)

- **未来天气预测数据获取**：继续以和风天气API为例，在index.vue中获取未来天气数据。修改 `fetchWeatherData` 方法，添加获取未来天气的请求：

```
async fetchWeatherData() {
  const apiKey = 'YOUR_API_KEY';
  const city = '北京';
  const nowUrl = `https://devapi.qweather.com/v7/weather/now?
location=${city}&key=${apiKey}`;
  const forecastUrl = `https://devapi.qweather.com/v7/weather/3d?
location=${city}&key=${apiKey}`;
  try {
    const nowRes = await uni.request({
      url: nowUrl,
      method: 'GET'
    });
    const forecastRes = await uni.request({
      url: forecastUrl,
      method: 'GET'
    });
    if (nowRes.statusCode === 200 && forecastRes.statusCode === 200) {
      const nowData = nowRes.data;
      const forecastData = forecastRes.data;
      // 处理当前天气数据
      this.cityName = nowData.location.name;
      this.date = new Date().toLocaleDateString();
      this.temperature = nowData.now.temp;
      this.weatherCondition = nowData.now.text;
      this.weatherIcon = this.getWeatherIcon(nowData.now.text);
      this.wind = nowData.now.windDir + nowData.now.windScale + '级';
      this.humidity = nowData.now.humidity;
      // 处理未来天气数据
      this.futureWeather = forecastData.daily.map(item => ({
        date: item.fxDate,
        maxTemp: item.tempMax,
        minTemp: item.tempMin,
        condition: item.textDay
      }));
    }
  } catch (error) {
    console.error('数据请求失败', error);
  }
}
```

分析返回数据的结构，提取未来几天的日期、最高温度、最低温度和天气状况。

- **未来天气展示布局**：在index.vue的中添加未来天气展示区域：

```
<view class="future-weather">
  <view class="future-weather-item" v-for="(weather, index) in futureWeather"
:key="index">
  <text>{{weather.date}}</text>
  <text>{{weather.minTemp}}°C - {{weather.maxTemp}}°C</text>
  <text>{{weather.condition}}</text>
</view>
</template>
```

并在中设置样式：

```
.future-weather {
  margin-top: 20px;
}
.future-weather-item {
  display: flex;
  justify-content: space-around;
  margin-bottom: 10px;
  border-bottom: 1px solid #ccc;
  padding-bottom: 10px;
}
```

讲解使用循环遍历数据，动态生成列表项，将未来天气数据展示在页面上。

- **页面优化**：讲解页面优化的几个方面。在性能优化上，可使用uni-app的生命周期函数，如onLoad和onShow，合理控制数据请求时机。例如，在onLoad时首次请求数据，在onShow时判断是否需要更新数据（可通过设置缓存时间来判断）：

```
onLoad() {
  this.fetchWeatherData();
},
onShow() {
  const lastUpdateTime = uni.getStorageSync('lastUpdateTime');
  const now = new Date().getTime();
  if (!lastUpdateTime || now - lastUpdateTime > 60 * 1000 * 10) { // 10分钟后更新数据
    this.fetchWeatherData();
    uni.setStorageSync('lastUpdateTime', now);
  }
}
```

在界面优化方面，调整页面颜色搭配，使其更符合天气主题。比如，晴天时使用明亮的暖色调，雨天时使用柔和的冷色调。还可以添加动画效果，如数据加载时的加载动画。在index.vue的中添加加载动画：

```
<view v-if="isLoading" class="loading">
  <text>加载中...</text>
</view>
```

在中定义isLoading变量，并在数据请求前后修改其值：

```
data() {
  return {
    // 其他数据...
    isLoading: false
  };
},
async fetchWeatherData() {
  this.isLoading = true;
  try {
    // 请求数据代码...
  } catch (error) {
    console.error('数据请求失败', error);
  } finally {
    this.isLoading = false;
  }
}
```

并在中设置加载动画样式：

```
.loading {
  position: fixed;
  top: 50%;
  left: 50%;
  transform: translate(-50%, -50%);
  font-size: 20px;
  color: #999;
}
```

- 课堂互动 (10分钟)**：学生在项目中实现未来天气预测展示功能，并进行页面优化，教师巡视指导，帮助学生解决遇到的问题，如未来天气数据展示不完整、页面加载缓慢等。鼓励学生尝试不同的优化方法，对比优化前后的效果。
- 课堂小结 (5分钟)**：总结本节课的重点内容，包括未来天气预测展示的实现和页面优化的方法，强调优化对提升项目质量的重要性。
- 作业布置 (5分钟)**：要求学生课后继续优化项目，如进一步优化页面性能，提高数据加载速度；完善未来天气预测展示的细节，如添加天气趋势图表等。

## 第五课时：项目调试与问题解决（45分钟）

1. **课程导入（5分钟）**：回顾之前课程中项目的开发进度，引出本节课进行项目调试，解决开发过程中遗留的问题。

### 2. 知识讲解（20分钟）

- **调试工具介绍**：介绍uni-app开发中常用的调试工具，如HBuilderX的内置调试器、微信开发者工具的调试功能等。在HBuilderX中，讲解如何使用控制台输出调试信息，如在代码中使用 `console.log` 打印变量值：

```
async fetchWeatherData() {
  console.log('开始请求天气数据');
  try {
    // 请求数据代码...
    console.log('数据请求成功');
  } catch (error) {
    console.error('数据请求失败', error);
  }
}
```

在微信开发者工具中，讲解如何使用“调试”面板查看网络请求、性能分析等信息。展示如何通过“网络”选项卡查看数据请求的URL、请求参数、响应数据等，分析请求是否成功以及数据是否正确。

- **常见问题及解决方法**：总结项目开发过程中可能遇到的常见问题。数据请求失败时，检查网络连接是否正常，API Key是否正确，请求的URL是否符合接口规范。页面显示异常，检查数据绑定是否正确，组件样式是否冲突。例如，若天气图标显示不正确，检查图标类名是否正确，图标库是否正确引入。若城市管理功能异常，检查本地存储的读写操作是否正确，数据格式是否一致。
- **错误处理机制**：讲解在项目中建立合理的错误处理机制的重要性。在数据请求失败时，向用户提示友好的错误信息。在 `fetchWeatherData` 方法的 `catch` 块中添加提示：

```
async fetchWeatherData() {
  try {
    // 请求数据代码...
  } catch (error) {
    console.error('数据请求失败', error);
    uni.showToast({
      title: '天气数据获取失败，请检查网络连接',
      icon: 'none'
    });
  }
}
```

展示如何使用 `try...catch` 语句捕获异常，以及如何在 `uni.request` 的失败回调函数中处理错误。

3. **课堂互动（10分钟）**：学生对自己开发的天气预报项目进行全面调试，查找和解决存在的问题，教师巡视指导，帮助学生分析和解决复杂问题，如多个问题相互影响导致的功能异常。组织学生分享调试过程中发现的问题和解决方法，促进学生之间的经验交流。

4. **课堂小结（5分钟）**：总结本节课的重点内容，包括调试工具的使用、常见

问题及解决方法，强调调试和错误处理在项目开发中的重要性。5. **作业布置 (5分钟)**：要求学生课后继续对项目进行测试，确保项目在不同设备和网络环境下都能稳定运行，记录并整理遇到的问题及解决方案。

## 第六课时：项目总结与拓展 (45分钟)

1. **课程导入 (5分钟)**：回顾整个天气预报项目的开发过程，引出本节课对项目进行总结，并探讨项目的拓展方向。

### 2. 知识讲解与实践指导 (20分钟)

- **项目总结**：引导学生对整个天气预报项目进行全面总结，回顾项目开发过程中各个阶段的技术要点，如页面布局、数据请求与处理、组件使用、调试技巧等。以首页为例，总结如何使用uni-app组件搭建页面结构，通过数据请求获取天气数据并绑定到页面元素上。强调从项目中学习到的知识和技能，以及在开发过程中遇到的问题和解决方法，鼓励学生分享自己在项目开发中的收获和体会。
- **项目拓展方向**：介绍天气预报项目的一些拓展方向。添加天气预警功能，可使用相关气象部门的预警数据接口，在页面中实时显示当地的天气预警信息。例如，获取预警信息后，在首页合适位置添加提示：

```
<view v-if="warningMessage" class="warning">
  <text>{{warningMessage}}</text>
</view>
```

```
async fetchWeatherWarning() {
  const apiKey = 'YOUR_API_KEY';
  const city = '北京';
  const url = `https://预警数据接口地址?location=${city}&key=${apiKey}`;
  try {
    const res = await uni.request({
      url: url,
      method: 'GET'
    });
    if (res.statusCode === 200) {
      const data = res.data;
      this.warningMessage = data.warningContent;
    }
  } catch (error) {
    console.error('预警数据请求失败', error);
  }
}
```

集成地图功能，使用地图API（如高德地图API），在地图上显示城市位置和天气状况。可以在地图上标记当前城市，点击标记显示天气信息。增加语音播报功能，使用uni-app的语音相关API（如uni.speechSynthesis），让用户通过语音获取天气信息。例如：

```
    speakWeather() {
      const text = `当前${this.cityName}的天气是${this.weatherCondition}, 温度是
${this.temperature}°C`;
      uni.speechSynthesis({
        text: text,
        success: () => {
          console.log('语音播报成功');
        },
        fail: (error) => {
          console.error('语音播报失败', error);
        }
      });
    }
  }
}
```

分析每个拓展方向涉及的技术难点和实现思路，激发学生的创新思维。

- **技术应用拓展**：引导学生思考如何将天气预报项目中学到的技术应用到其他类似项目中，如开发空气质量监测应用、旅游出行助手等。开发空气质量监测应用时，可以借鉴天气预报项目的数据请求和处理方式，从相关数据接口获取空气质量数据，并在页面中展示。旅游出行助手可结合天气数据和景点信息，为用户提供出行建议。鼓励学生拓展技术应用场景，提升自己的技术迁移能力。
3. **课堂互动（10分钟）**：组织学生讨论项目拓展方向的可行性和实现方案，鼓励学生提出自己的想法和创意。教师对学生的想法进行点评和指导，帮助学生进一步完善拓展方案。
  4. **课堂小结（5分钟）**：总结本节课的重点内容，包括项目总结和拓展方向，鼓励学生在今后的学习和实践中继续探索和创新。
  5. **作业布置（5分钟）**：要求学生根据自己的兴趣选择一个项目拓展方向，制定详细的开发计划，包括功能需求分析、技术选型、开发步骤等。

## 六、教学资源

1. **硬件资源**：配备稳定网络环境的计算机机房，每台计算机均安装HBuilderX开发工具，确保其版本能支持uni-app开发及相关组件、API的使用与调试。同时，准备足够数量的真机设备或安装相关手机模拟器，方便学生进行移动端的测试与预览。
2. **软件资源**：除HBuilderX外，还需在计算机上安装微信开发者工具、Chrome浏览器等，用于多平台的项目运行与调试。提供天气预报项目所需的素材资源，如天气图标、城市数据文件等，可从免费素材网站获取或自行制作，方便学生在项目开发中使用。
3. **文档资料**：精心制作与课程内容紧密匹配的教学PPT，包含详细的文字讲解、清晰的代码示例截图、生动的操作演示动画等，辅助课堂教学，帮助学生更好地理解和掌握知识点。整理uni-app官方文档、天气数据接口文档以及相关技术博客文章，方便学生在课堂内外随时查阅，深入学习技术细节和接口的使用方法。
4. **在线学习平台**：利用在线学习管理系统，如学堂在线、雨课堂等，发布课程资料、作业任务、拓展学习资源等，方便学生在线学习、提交作业和交流讨论。同时，引入技术论坛和开源代码库，如CSDN、GitHub等，让学生能够获取更多的技术支持和项目参考，拓宽技术视野，了解行业最新动态。

## 七、教学反思

- 教学内容方面：**部分学生对天气数据接口的理解和数据处理存在困难，后续教学应增加更多数据解析和处理的案例，帮助学生更好地掌握从接口获取数据并应用到项目中的能力。在页面优化部分，部分学生对性能优化的理解不够深入，导致优化效果不明显。后续可加强性能优化相关知识的讲解，如详细介绍缓存机制、资源加载优化策略等。
- 教学方法方面：**项目驱动法等多种教学方法结合，在一定程度上提高了学生的参与度，但部分学生在自主学习和解决问题时仍较为依赖教师。后续应在教学过程中进一步引导学生自主探索，例如布置更具开放性的任务，鼓励学生自主查阅资料、尝试不同的解决方案，培养学生独立解决问题的能力。
- 学生学习效果方面：**从项目完成情况来看，学生在页面布局和基本功能实现上有一定成果，但在项目拓展方面创意和深度不足。后续教学可在课程中提前引入拓展思路的讨论，激发学生的创新思维，引导学生在项目开发过程中就思考拓展方向，提高学生的项目拓展能力。
- 思政教育融入方面：**在教学过程中可进一步挖掘思政元素融入点，如在介绍天气数据接口时，提及数据安全和隐私保护的重要性；在项目拓展讨论时，引导学生思考如何利用技术为社会提供更有价值的服务，培养学生的社会责任感和职业道德。

## 第九章 课程设计与考核验收

### 一、课程基本信息

- 课程名称：**移动应用开发 - 课程设计与考核验收
- 授课教师：**
- 授课班级：**
- 授课时间：**共4课时，每课时45分钟

### 二、教学目标

- 知识目标**
  - 理解课程设计的目标、要求和流程，明确项目实践在课程中的重要性。
  - 掌握课程考核验收的标准和方法，了解如何对项目成果进行全面、客观的评价。
- 能力目标**
  - 能够根据课程设计要求，综合运用所学知识和技能，独立完成移动应用项目的设计与开发。
  - 学会对自己和他人的项目成果进行评估，提升批判性思维和问题解决能力。
- 素质目标**
  - 培养学生的创新精神和实践能力，鼓励学生在项目设计中展现独特的创意和技术应用。
  - 强化学生的职业素养，如时间管理、任务规划和团队协作精神（若涉及团队项目）。

### 三、教学重难点

- 教学重点**

- 课程设计的任务拆解、技术选型和项目规划。
- 课程考核验收的标准制定和评价方法实施。

## 2. 教学难点

- 引导学生在课程设计中突破常规思维，实现创新性的功能和设计。
- 确保课程考核验收的公平性和客观性，准确评估学生的项目成果和能力水平。

# 四、教学方法

1. **讲授法**：讲解课程设计要求、考核验收标准等理论知识，让学生明确任务和目标。
2. **实践法**：学生进行课程设计实践，在实际操作中巩固知识和提升能力，教师进行现场指导。
3. **讨论法**：组织学生讨论课程设计中的技术难题、创意方案以及考核验收的评价标准，促进学生之间的思想交流和经验分享。
4. **评价法**：对学生的课程设计成果进行评价，通过评价引导学生发现问题、改进不足，同时让学生学会自我评价和互评。

# 五、教学过程

## 第一课时：课程设计任务与要求讲解（45分钟）

1. **课程导入（5分钟）**：回顾之前课程所学的uni-app组件、API以及多个项目实例的开发过程，强调课程设计是对所学知识的综合运用和实践检验。引出本节课对课程设计任务与要求的讲解，激发学生的积极性和主动性。**\*\*【思政融入】\*\***以“知行合一”的哲学思想引入课程设计——王阳明提出“知者行之始，行者知之成”，学习的最终目的是实践。课程设计正是将所学知识转化为实际能力的关键环节。引导学生认识到，真正的本领不是纸上谈兵，而是在实践中磨练出来的，鼓励学生以“经世致用”的态度对待课程设计，开发出真正能服务社会的应用。
2. **知识讲解（20分钟）**
  - **课程设计目标**：明确本次课程设计旨在让学生运用uni-app技术开发一个完整的移动应用项目，项目应具备一定的实用性和创新性，能够解决实际生活中的某个问题或满足特定的用户需求。
  - **具体任务**：详细阐述课程设计的任务内容，例如开发一个具有特定功能的移动应用，如校园生活助手（包含课程查询、校园资讯推送、失物招领等功能）或本地生活服务应用（涵盖周边商家推荐、生活缴费、社区活动发布等功能）。要求学生根据给定的主题方向，确定具体的功能模块，并进行详细的设计和开发。
  - **技术要求**：强调项目开发过程中需运用到本学期所学的uni-app组件、API，如页面布局组件（view、text、button等）、数据请求API（uni.request）、存储API（uni.setStorageSync、uni.getStorageSync）等。鼓励学生尝试使用扩展组件（如uView组件库）来优化项目界面和功能，提升用户体验。
  - **时间安排**：制定课程设计的时间进度表，明确每个阶段的任务和截止日期。例如，第一周完成项目需求分析和设计文档撰写，第二周进行项目开发，第三周进行项目测试和优化，第四周提交项目成果并进行考核验收。

3. **课堂互动 (10分钟)**：组织学生分组讨论课程设计任务，让学生结合自身兴趣和能力，初步确定各自的项目方向。每组推选代表分享讨论结果，教师进行点评和指导，帮助学生进一步明确项目目标和任务，确保项目的可行性和挑战性。
4. **课堂小结 (5分钟)**：总结本节课的重点内容，包括课程设计的目标、任务、技术要求和时间安排。强调课程设计对学生综合能力提升的重要性，鼓励学生积极投入到课程设计中。
5. **作业布置 (5分钟)**：要求学生课后查阅相关资料，对初步确定的项目方向进行更深入的需求分析，撰写项目需求分析文档，包括项目背景、目标用户、功能需求、非功能需求等内容。

## 第二课时：课程设计实践指导（一）（45分钟）

1. **课程导入 (5分钟)**：回顾上节课布置的作业，即项目需求分析文档的撰写。检查学生的完成情况，选取部分学生的文档进行简单点评，引出本节课的课程设计实践指导内容。
2. **实践指导 (20分钟)**
  - **项目规划与设计**：指导学生根据需求分析文档进行项目规划，包括确定项目的整体架构、页面布局和功能流程。讲解如何使用思维导图工具（如XMind、MindManager等）进行项目结构设计，梳理各个功能模块之间的关系。以校园生活助手为例，展示如何规划课程查询、资讯推送、失物招领等功能模块的页面跳转和数据交互流程。
  - **技术选型与工具使用**：帮助学生根据项目需求选择合适的技术方案和开发工具。对于uni-app开发，进一步介绍如何选择和使用第三方插件、组件库来优化项目开发过程。例如，在开发具有地图功能的应用时，推荐使用uniapp的地图组件结合高德地图或百度地图的API；在处理表单输入时，介绍uView组件库中的表单组件。同时，讲解如何使用版本控制工具（如Git）进行代码管理，确保代码的安全性和可维护性。
  - **代码框架搭建**：现场演示如何搭建项目的代码框架，包括创建项目目录结构、配置项目文件（如pages.json、main.js等）。指导学生在项目中创建各个功能页面，并初步设置页面的布局和样式。以一个简单的页面为例，展示如何使用flex布局或grid布局来实现页面元素的排列和对齐。
3. **学生实践 (10分钟)**：学生根据教师的指导，开始搭建自己项目的代码框架，进行项目规划和技术选型。教师巡视指导，及时解答学生在实践过程中遇到的问题，如项目目录结构混乱、插件安装失败等。
4. **课堂小结 (5分钟)**：总结本节课学生实践过程中的常见问题和解决方法，强调项目规划和技术选型对项目开发的重要性。鼓励学生在课后继续完善项目代码框架，为后续的功能开发做好准备。
5. **作业布置 (5分钟)**：要求学生课后完成项目代码框架的搭建，确定各个功能页面的布局和样式。同时，思考如何实现各个功能模块的数据请求和处理逻辑，为下一课时的功能开发做准备。

## 第三课时：课程设计实践指导（二）（45分钟）

1. **课程导入 (5分钟)**：回顾上节课学生搭建项目代码框架的情况，选取部分学生展示他们的成果并进行点评。针对上节课遗留的问题进行解答，引出本节课继续进行课程设计实践指导，重点讲解功能开发的相关内容。
2. **实践指导 (20分钟)**
  - **功能实现技术要点**：针对学生在项目中常见的功能需求，如数据请求与展示、用户交互功能等，进行技术要点讲解。以数据请求为例，详细介绍如何使用uni.request进行网络请求，包括设置请

求参数、处理响应数据等。展示如何根据后端接口文档，正确构建请求URL，并在请求成功后将数据展示在页面上。对于用户交互功能，如点击按钮实现页面跳转、表单提交验证等，讲解相关组件的事件绑定和处理方法。

- **数据存储与管理**：讲解在uni-app中如何进行数据存储和管理，介绍本地存储（uni.setStorageSync、uni.getStorageSync）和云端存储（如使用云开发服务）的使用场景和方法。以校园生活助手的课程查询功能为例，展示如何将用户查询的课程信息存储在本地，以便下次快速访问；对于一些需要实时更新的数据，如校园资讯，介绍如何使用云端存储进行数据管理。
  - **问题解决与调试技巧**：分享在项目开发过程中常见问题的解决方法和调试技巧。例如，当数据请求失败时，如何通过查看控制台日志、检查网络连接和请求参数来定位问题；当页面样式显示异常时，如何使用浏览器开发者工具或手机调试工具进行样式调试。强调在开发过程中养成良好的调试习惯，及时记录和总结问题及解决方法。
3. **学生实践（10分钟）**：学生根据教师的指导，开始进行项目的功能开发。教师巡视指导，密切关注学生的开发进度和遇到的问题，及时给予帮助和建议。对于普遍存在的问题，进行集中讲解和指导。
  4. **课堂小结（5分钟）**：总结本节课学生在功能开发过程中遇到的问题和解决方法，强调在开发过程中要注重代码的规范性和可读性，以便后续的维护和扩展。鼓励学生在课后继续完成项目的功能开发，并进行初步的测试。
  5. **作业布置（5分钟）**：要求学生课后完成项目的所有功能开发，并对项目进行初步测试，检查功能是否正常运行，界面是否友好。记录在测试过程中发现的问题，思考如何进行优化和改进。

#### 第四课时：课程考核验收与总结（45分钟）

1. **课程导入（5分钟）**：回顾课程设计的整个过程，强调考核验收是对学生学习成果的检验和评价。引出本节课将进行课程考核验收，介绍考核验收的流程和标准，让学生了解如何展示自己的项目成果。**\*\*【思政融入】\*\***在考核验收中强调学术诚信与职业操守——项目展示必须真实反映自己的工作成果，杜绝抄袭、代做等学术不端行为。介绍学术诚信的法律法规和行业规范，引导学生认识到诚信是立身之本、从业之基。同时，在评价他人项目时秉持客观公正的态度，做到“见贤思齐焉，见不贤而内自省也”，在互评中共同进步。
2. **考核验收标准讲解（10分钟）**
  - **功能完整性（40分）**：讲解功能完整性的评分标准，项目是否实现了需求分析文档中规定的所有功能，功能是否正常运行，有无明显的漏洞或错误。例如，对于校园生活助手项目，课程查询功能能否准确获取课程信息，失物招领功能是否能正常发布和查看失物信息等。
  - **界面设计与用户体验（20分）**：介绍界面设计和用户体验的评分要点，包括页面布局是否合理、美观，操作是否便捷流畅，是否符合用户使用习惯。例如，页面元素的排版是否整齐，颜色搭配是否协调，按钮的点击响应是否及时等。
  - **技术实现与代码质量（30分）**：讲解技术实现和代码质量的评价标准，包括是否正确运用所学技术，代码结构是否清晰，注释是否规范，是否进行了适当的优化。例如，数据请求和存储的实现是否合理，代码是否进行了模块化封装，关键代码是否有清晰的注释说明等。
  - **文档撰写（10分）**：说明项目文档撰写的评分标准，包括需求分析文档、设计文档是否完整、规范，是否能够清晰地描述项目的功能、架构和实现思路。

- 3. 学生项目展示与评价 (20分钟)：**学生按照顺序依次展示自己的项目成果，每个学生展示时间为5 - 8分钟。展示内容包括项目的功能演示、设计思路、技术实现要点等。展示结束后，其他学生进行提问和评价，教师进行总结和点评，根据考核验收标准进行打分。在评价过程中，引导学生从功能、界面、技术、文档等多个方面进行客观评价，培养学生的批判性思维和评价能力。
- 4. 课程总结 (5分钟)：**对整个课程进行总结，回顾课程的重点内容和学生在课程设计中的表现。表扬学生在项目开发中的创新点和优秀表现，同时指出存在的不足和需要改进的地方。鼓励学生在今后的学习和实践中继续提升自己的移动应用开发能力。
- 5. 作业布置 (5分钟)：**要求学生根据教师和同学的评价意见，对项目进行最后的优化和完善。整理项目代码和文档，提交最终的课程设计成果。同时，撰写课程学习心得，总结在课程学习和项目开发过程中的收获、体会以及对课程的建议。

## 六、教学资源

- 1. 硬件资源：**配备稳定网络环境的计算机机房，每台计算机均安装HBuilderX开发工具，确保其版本能支持uni-app开发及相关组件、API的使用与调试。同时，准备足够数量的真机设备或安装相关手机模拟器，方便学生进行移动端的测试与预览。
- 2. 软件资源：**除HBuilderX外，还需在计算机上安装微信开发者工具、Chrome浏览器等，用于多平台的项目运行与调试。提供课程设计相关的参考资料，如优秀项目案例、技术文档模板等，方便学生在课程设计过程中参考和学习。
- 3. 文档资料：**精心制作与课程内容紧密匹配的教学PPT，包含详细的文字讲解、清晰的项目案例截图、生动的操作演示动画等，辅助课堂教学，帮助学生更好地理解课程设计任务和考核验收标准。整理uni-app官方文档、相关技术博客文章以及代码规范文档，方便学生在课堂内外随时查阅，深入学习技术细节和开发规范。
- 4. 在线学习平台：**利用在线学习管理系统，如学堂在线、雨课堂等，发布课程设计任务、考核验收标准、参考资料等，方便学生在线学习、提交作业和交流讨论。同时，引入技术论坛和开源代码库，如CSDN、GitHub等，让学生能够获取更多的技术支持和项目参考，拓宽技术视野，了解行业最新动态。

## 七、教学反思

- 1. 教学内容方面：**在课程设计任务讲解中，部分学生对需求分析和项目规划的理解不够深入，导致项目开发初期方向不明确。后续教学应增加更多需求分析和项目规划的案例讲解，引导学生学会从实际需求出发，合理规划项目。在考核验收标准讲解中，部分学生对一些评价指标的理解存在偏差，影响了自我评价和互评的准确性。后续可进一步细化考核标准的解释，提供更多示例说明，确保学生理解每个指标的内涵。
- 2. 教学方法方面：**讲授法、实践法、讨论法和评价法相结合，在一定程度上促进了学生的学习和实践。但在实践指导过程中，发现部分学生缺乏主动探索精神，过于依赖教师的指导。后续应在教学过程中设置更多开放性问题，鼓励学生自主思考和探索解决方案，培养学生的独立解决问题能力。在评价环节，虽然组织了学生互评，但部分学生参与度不高，评价不够深入。后续可优化评价机制，增加评价的趣味性和激励性，提高学生的参与度。
- 3. 学生学习效果方面：**从学生的项目成果来看，大部分学生能够完成基本的功能开发，但在创新性和用户体验方面还有待提高。后续教学可在课程设计任务中增加对创新性的要求，引导学生关注用户需求和市场趋势，鼓励学生在项目中尝试新的技术和设计理念。同时，加强对用户体验设计的教学，提升学生的设计能力。

4. **思政教育融入方面**：在课程设计与考核验收过程中，可进一步挖掘思政元素。例如，在评价项目成果时，强调诚信原则，杜绝抄袭行为；在鼓励学生创新时，培养学生的创新精神和社会责任感，引导学生开发对社会有益的应用。通过这些方式，将思政教育与专业教学更紧密地结合起来。