**WEB第五阶段DAY08 — uni-app**

复习：

**①目前学过的uni-app组件：**

**内置组件**（高仿小程序，没有前缀）：

view、scroll-view、text、image、navigator、swiper、video、canvas、rich-text...

**扩展组件**（官方+第三方库，以uni-开头）：

uni-icons、uni-drawer、uni-list/uni-list-item、uni-card、uni-grid/uni-grid-item、uni-search、uni-tag、uni-badge、uni-transition、uni-load-more、uni-easyinput...

**自定义组件**（以项目名开头）：

**②目前学过的uni-app API：**

uni.pageScrollTo()

uni.navigateTo() / uni.navigateBack() / uni.redirectTo() / uni.reLaunch() / uni.switchTab()

uni.request()

uni.showLoading() / uni.hideLoading()

uni.showToast()

uni.showModal()

uni.stopPullDownRefresh()

uni.setNavigationBarTitle()

|  |
| --- |
| H5的两层翻译：  ①程序员眼中：HTML5的简称，用于创建网站/WebApp的三种基础技术之一  ②营销/市场人员眼中：是一种可以在手机中分享、快速传播、快速开发/使用/丢弃手机端页面 |

1. **面试题：移动端应用的种类**

**①NativeApp：**原生App，Android中使用Java/Kotlin开发，iOS中使用Objective-C/Swift开发。不足：费用高；优势：效率高。

**②WebApp：**使用HTML/CSS/JS编写，运行在浏览器中，封装为APP安装文件，在应用商店中上架。优势：一套代码可以运行在两个平台；不足：执行效率不如原生。

**③H5：**使用HTML/CSS/JS编写，运行在浏览器中，不需要应用商店中上架，也没有图标，通过二维码在朋友圈中快速分享。优势：一套代码可以运行在两个平台；不足：执行效率不如原生。

**④HybridApp：**混合/混编App，使用HTML/CSS/JS/Native.js(Java/OC=>JS)。优势：一套代码可以运行在两个平台，还可以调用系统底层原生功能；不足：执行效率不如原生。

**⑤BridgedApp：**桥接App，只使用JS编写，执行环境会把JS编译为Java/OC，再手机中，运行时就是纯原生代码。优势：一套代码可以运行在两个平台, 执行效率高; 不足：两个平台不能做到完全一样。

**⑥FlutterApp：**使用Dart语言开发，直接在Android/iOS手机中的显卡里进行绘图。优势：一套代码可以运行在两个平台，执行效率高，效果完全一样； 不足：全新的语言，Google开发的进度比较慢。

**⑦小程序**

1. **uni-app概述**

官网：https://uniapp.dcloud.io

uni-app 是一个使用 Vue.js 开发所有前端应用的框架，开发者编写一套代码，可发布到iOS、Android、Web（响应式）、以及各种小程序（微信/支付宝/百度/头条/飞书/QQ/快手/钉钉/淘宝）、快应用等多个平台。



**演示1：创建一个示例项目，在浏览器中调试运行 —— 即uni-app编译为H5网站**

①点击工具栏中的“运行”>“运行到浏览器”>“Chrome”，

②HBuilder会编译整个项目为html/css/js代码，启动一款开发服务器，自动打开浏览器

**演示2：创建一个示例项目，在安卓手机中调试运行 —— 即uni-app编译为App**

①下载并安装一款安卓模拟器（例如夜神或逍遥或雷电）

②在HBuilder设置adb的路径：“运行”>“运行到手机或模拟器”>“ADB路径设置”，重启HBuilder和夜神模拟器，等待一会儿（3-5分钟）....

③HBuilder中编译当前项目，安装到手机中开始运行：“运行”>“运行到手机或模拟器”>“手机名称”

**演示3：创建一个示例项目，作为微信小程序中调试运行 —— 即uni-app编译为微信小程序**

①下载并安装“微信开发者工具”——用于最后一步“提请人工审核”

②打开微信开发者工具的“服务端口”——就允许其他开发工具导入项目进来

微信开发者工具中的“设置”>“安全设置”>“服务端口”打开即可

③HBuilder中设置微信开发者工具软件的路径

HBuilder中点击“运行”>“运行到小程序模拟器”>“运行设置”>输入微信开发者工具的路径

④编译当前uni-app为小程序代码，导出到微信开发者工具

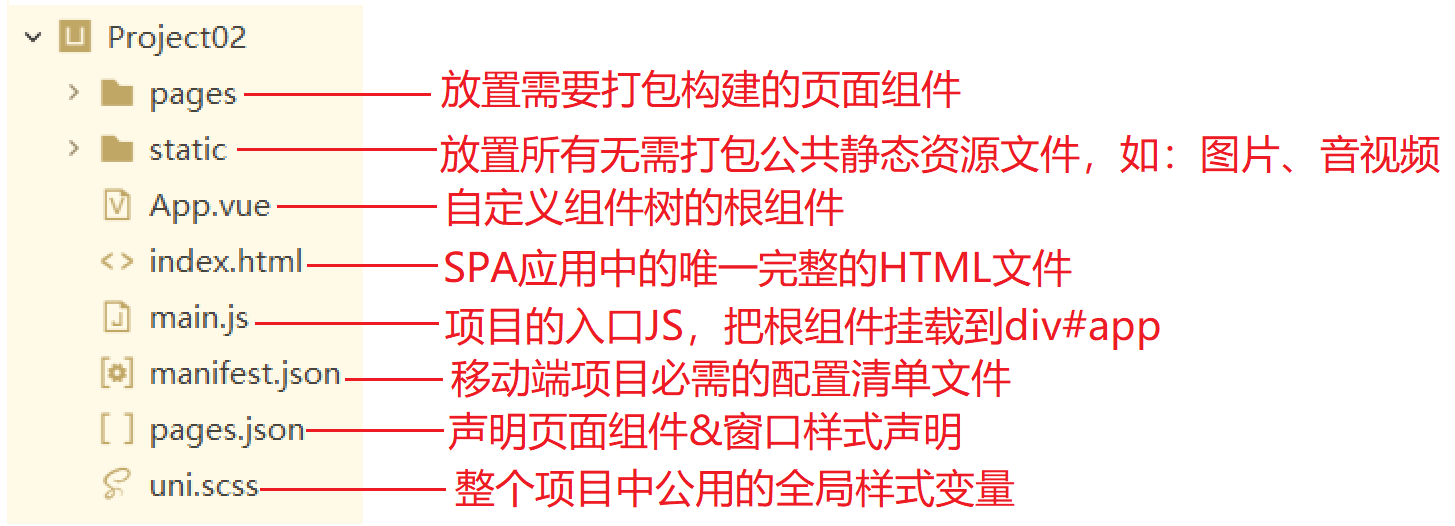
HBuilder中点击“运行”>“运行到小程序模拟器”>“微信开发者工具”

稍等片刻，微信开发者工具就会启动并打开编译好的项目

|  |
| --- |
| 行业小知识：ADB  Android Debugger，安卓调试器，是Windows和安卓系统间通信通道，类似于数据线，用于传输数据 |

1. **我的第一个uni-app项目**

项目目录介绍：



**一、uni-app中的条件编译**

不同的运行平台终归有些专有的特性，无法实现跨平台完全兼容，例如：微信小程序导航栏右上角的关闭图标。uni-app提供了一种“条件编译”机制，可以针对特定的平台编译执行特定的代码，否则不执行。语法如下：

|  |
| --- |
| #ifdef H5  仅在H5平台下编译执行的代码  #endif |
| #ifdef H5 || APP || MP-WEIXIN  仅在H5和APP和微信小程序平台下编译执行的代码  #endif |
| #ifndef H5  仅在非H5平台下编译执行的代码  #endif |

说明：

①条件编译语句可以编写在template / style / script 等各类代码中。

②更多的条件编译平台可以参见手册：https://uniapp.dcloud.io/tutorial/platform.html

**二、uni-app中的视图模板 —— <template>**

**— Vue.js页面模板中可以使用的标签：**

所有的HTML5标签 + <block></block> / <template></template>

**— 微信小程序页面模板中可以使用的标签：**

微信官方专有标签（https://developers.weixin.qq.com/miniprogram/dev/component/）

**— uni-app页面模板中可以使用的标签：**

①如果仅仅H5是目标平台的话，可以使用HTML5标签；否则一律不允许使用HTML5标签！！！

②uni-app完全仿写了一套“微信小程序标签”—— **内置组件**，

例如：view、text、swiper

③uni-app官方以及众多的第三方开发者组合开发的一套扩展组件库 —— **uni-ui组件库，**例如：uni-card、uni-drawer...

④应用程序自定义的组件 —— 推荐以项目名做前缀，例如：<xz-go-top>...

**步骤1：创建自定义组件必需的文件**

1)自定义组件只能编写在/components子目录下！

2)自定义必须放在一个同名目录下！

3)组件目录名和文件名可以使用小驼峰命名法或烤串命名法，推荐使用小驼峰；

**~~步骤2：在需要该组件的页面中注册自定义组件~~**

**步骤3：在需要该组件的页面中使用自定义组件**

1)直接使用，无需注册

2)不论定义时使用什么命名法，在使用时元素名为文件名一致即可

|  |
| --- |
| 原则：什么情况下提取“公共组件”？  ①一段在多个页面中反复出现的代码，例如“返回顶部”  ②一个文件如果过长了(影响代码的可读性和可维护性)也应该拆分为多个组件，例如“首页” |

演示：创建自定义组件 —— 返回顶部

select > option \* 3100000

ul > li \* 1400000

|  |
| --- |
| **面试题：如何对长列表进行优化？**  不要目标数据中的所有项全部挂载到DOM树 —— 只挂载有限条数（例如：30条），如果此时要挂载的数据还有很多，不要全部挂载 —— 只挂载当前30条 + 前述30条 + 后述30条 |

|  |
| --- |
| 小知识：  Vue.js中如何绑定显示一段HTML标签？  innerText绑定： <text>{{HTML片段}}</text>  innerText绑定： <text v-text="HTML片段"></text>  innerHTML绑定： <text v-html="HTML片段"></text> |
| 微信小程序中如何绑定显示一段HTML内容？ —— uni-app中推荐使用此方法  <rich-text :nodes="HTML片段"></rich-text> |

|  |  |
| --- | --- |
| Vue.js/uni-app中父子组件间传参 —— Props Down，Events Up： | |
| **//父组件传递数据给子组件：Props Down**  parent.js  data: { age: 20 }  <xzChild :childAge="age"/>  child.js  props: ['childAge']  <text>来自父亲的数据：{{childAge}}</text> | **//父组件传递数据给子组件：Events Up**  parent.js  <xzChild @cry="doCry"/>  doCry(data){ //处理儿子的事件 }  child.js  this.$emit('cry', {reason:'饿了'}) |

**三、uni-app中的样式语言 —— <style>**

— Vue.js页面样式中可以使用的样式语言： 标准的W3C提供的CSS

— 微信小程序页面样式中可以使用的样式语言：是标准CSS的子集 + rpx

— uni-app页面样式中可以使用的标签语言：是标准CSS的子集 + rpx + vw/vh

1)选择器禁止使用 \*{ }； 用 :not(not){ } 来代替。

2)每个页面的根元素不是html或body，而是page

|  |
| --- |
| 面试题：Web开发中，有哪些页面布局技术？  ①表格布局 —— <table>  ②浮动布局/流式布局 —— float: left/right;  ③响应式布局 —— @media { }  **④弹性布局 —— display: flex;**  ⑤网格布局 —— display: grid; |
| 弹性布局相关的CSS属性：  **第一组：弹性父容器相关的属性**  display: flex;  flex-direction: row | column; 弹性父容器的主轴方向  flex-wrap: nowrap | wrap; 弹性父容器是否允许子元素换行  justify-content: flex-start | flex-end | center | space-evenly | space-around | space-between; 弹性子元素在父容器主轴方向上对齐方式  align-items: flex-start | flex-end | center | stretch; 弹性子元素在父容器交叉轴上对齐方式  **第二组：弹性子元素相关的属性**  flex: 数字； 弹性子元素尺寸增长的权重 |

|  |
| --- |
| **CSS小知识：如何让文字只显示一行，溢出部分用省略号代替？**  white-space: nowrap; //文本在空格处不换行  overflow: hidden; //溢出内容隐藏掉  text-overflow: ellipsis; //文本溢出的话，用省略号代替 |
| **CSS小知识：如何让文字只显示X行(X大于1)，溢出部分用省略号代替？**  overflow: hidden; //溢出的内容要隐藏  text-overflow: ellipsis; //文本溢出的话，用省略号代替  display: -webkit-box; //显示模式：Webkit内核专用的盒子  -webkit-line-clamp: 2; //要想显示的行数  -webkit-box-orient: vertical; //多行如何排列：竖直 |

**四、uni-app中的JS脚本 —— <script>**

— Vue.js页面脚本中可以使用的脚本： SFC语法 + JS原生对象 + BOM&DOM对象

— 微信小程序页面脚本中可以使用的脚本：四文件组件语法 + JS原生对象 + WX专有API

— uni-app页面脚本中可以使用的脚本：SFC语法 + JS原生对象 + UNI专有API(高仿WX)

|  |
| --- |
| 目前学过的uni-app API：  ①uni.pageScrollTo( )：滚动到页面特定位置 |

|  |
| --- |
| 补充：Vue小知识  Vue2中为“所有的页面和组件对象”添加公共成员  **Vue.prototype**.$xxx = .....  console.log(this.$xxx) <text>{{$xxx}}</text> |
| Vue3中为“所有的页面和组件对象”添加公共成员  **app.config.globalProperties**.$xxx = .....  console.log(this.$xxx) <text>{{$xxx}}</text> |

|  |
| --- |
| App中常用的三种“对话框”：  ①“加载中”对话框——加载完成会自动消失  uni.showLoading() / uni.hideLoading()  ②“吐司”对话框——弹出来，停留2-3秒钟，然后自动消息  uni.showToast( )  ③“模态”对话框——一般用于重要紧急事项，必须用户点击“确定/取消”才能继续进行  uni.showModal( ) |

**五、uni-app中的数据绑定**

**1、内容绑定**

<any>{{表达式}}</any>

**2、属性绑定**

<any v-bind:属性名="表达式"/>

<any :属性名="表达式"/>

**3、样式绑定**

<any :style="{backgroundColor: bgc}" :class="{btn: true, danger: false}"/>

**4、双向数据绑定**

<输入元素 v-model="模型变量"/>

**5、事件绑定**

<any @click="f1" />

methods: { f1(){ } }

**6、条件渲染**

<any v-if="表达式"/>

<any v-else-if="表达式"/>

<any v-else/>

**7、列表渲染**

<any v-for="(item, index) in 数组名" :key="index"/>

|  |
| --- |
| Web项目中图片的类型：  **①静态图片：**基本不会发生改变的，例如：logo、icons等 —— App项目中会打包安装到手机中  **②动态图片：**每时每刻都可能发生改变，例如：轮播图片、商品、评论、头像等 —— App项目中不可能打包安装到手机中，只能放到数据服务器上 |

**六、uni-app中的路由和导航跳转**

— Vue.js中实现路由导航需要用到：Vue-Router

— 微信小程序使用路由词典：pages.json；页面跳转使用五个API

— uni-app使用路由词典：pages.json；页面跳转使用五个API

|  |
| --- |
| uni-app中实现页面跳转的五个方法——脚本中：  **① uni.navigateTo( )**：“导航到指定页面”——保留跳转； p1=>p2，p1会保留在历史记录中  **② uni.navigateBack( )**：“导航返回之前的页面”，目标页面必须在历史记录中存在  **③ uni.redirectTo( )**：“重定向到指定页面”——销毁跳转； p1=>p2，p1不会保留到历史记录中  **④ uni.reLaunch( )：**“重新加载指定页面”——历史记录“可能”全部销毁(uni-app中没有销毁）  **⑤ uni.switchTab( )：**“切换页签页”——只有此方法可以跳转到有页签的页面 |
| uni-app中实现页面跳转的五个方法——模板中：  ① <navigator open-type="**navigate**" url="跳转地址">文字</navigator>  ② <navigator open-type="**navigateBack**">文字</navigator>  ① <navigator open-type="**redirect**" url="跳转地址">文字</navigator>  ① <navigator open-type="**reLaunch**">文字</navigator>  ① <navigator open-type="**switchTab**" url="跳转地址">文字</navigator> |
| 页面跳转传参：  p1： url="/pages/p2/p2?k1=v1&k2=v2&...."  p2： onLoad(data){ data就是p1传递来的数据 } //高仿小程序 |

**七、uni-app中的生命周期方法**

**— Vue.js中组件/页面的生命周期方法**

创建阶段：beforeCreate / created

挂载阶段：beforeMount / mounted

更新阶段：beforeUpdate / updated

卸载阶段：beforeDestroy / destroyed

**— 小程序中组件/页面的生命周期方法**

onLoad( ) / onShow( ) / onReady( ) / onHide( ) / onUnload( ) / onReachBottom( ) / onPullDownRefresh( )

**— uni-app中的生命周期方法 = Vue.js的生命周期方法 + 小程序的生命周期方法**

1、应用级生命周期方法 —— App.vue

① onLaunch( ) ：应用程序启动了

② onShow( )：应用程序显示出来了

③ onHide( )：应用程序隐藏起来了

2、页面级生命周期方法 —— pages/xxx/xxx.vue

① onLoad( )：页面加载了，用于“请求服务器端数据”

② onShow( )：页面显示出来了

③ onReady( )：页面准备就绪，可供用户操作了

④ onHide( )：页面隐藏起来了，例如：保留跳转

⑤ onUnload( )：页面销毁了，例如：销毁跳转 ，用于释放页面所占用的资源，例如“定时器”、“WS通道”

⑥ onReachBottom( )：页面滚动到底部了，用于“触底加载更多”

⑦ onPullDownRefresh( )：页面在顶部下拉刷新，用于“顶部下拉重新加载”

3、组件级生命周期方法 —— components/xxx/xxx.vue

① beforeCreate( )

② created( )

③ beforeMount( )

④ mounted( )

⑦ beforeDestroy( )

⑧ destroyed( )

|  |
| --- |
| 面试题：前端可用的异步数据请求方法有哪些？前端AJAX应用中可以使用哪些具体技术？  **①XHR** —— new XMLHttpRequest( )，是W3C标准技术，所有浏览器都支持；使用麻烦，基于回调  **②jQuery.ajax()** —— 底层还是XHR；使用简单，但基于回调  **③axios** —— 第三方的HTTP请求模块，可用于Node.js项目(此时底层是http模块)或基于浏览器的网站项目(此时底层是XHR)；使用简单，基于Promise  **④wx.request()** —— 专用于微信小程序的方法  **⑤uni.request()** —— 专用于uni-app项目  **⑥fetch** ——是W3C标准技术，目前主流浏览器都已经支持，用于代替XHR；使用简单，基于Promise |

**八、异步请求服务器端接口数据**

uni-app项目中主要是 **uni.request()** 发起异步请求；

不用axios的原因：不用引入第三方模块了

不用xhr/fetch的原因：小程序底层没有浏览器

|  |  |
| --- | --- |
| 使用uni.request()发起GET请求：  let url = "http://xxxx" 后面无需追加查询字符串  uni.request({  url,  data: {查询参数1：值1, 查询参数2：值2...}  })  .then(result=>{ result[1].data是响应主体 })  .catch(err=>{ }) | 使用uni.request()发起POST请求：  let url = "http://xxxx"  uni.request({  url,  method: 'POST',  header: {'Content-Type': 'application/x-www-form-urlencoded'}  data: {查询参数1：值1, 查询参数2：值2...}  })  .then(result=>{ result[1].data是响应主体 })  .catch(err=>{ }) |

|  |
| --- |
| 复习：Promise的完整结构  let p = new Promise( function(resolve, reject){  if(.....)resolve('出去玩儿')  else reject('家里补习')  } )  p.then((res)=>{console.log(res)})  .catch((err)=>{console.log(err)})  .finally(( )=>{console.log('complete')}) |

|  |
| --- |
| 提示：为了提升用户体验，一般项目中在发送异步请求之前，会显示“加载中”提示框：  uni.showLoading({  title: '数据加载中'  })  等到异步请求完成（不论成功还是失败）隐藏此对话框：  uni.hideLoading( ) |

|  |
| --- |
| 行业知识： HTTP协议规定服务器发给客户端的“响应消息主体”允许的内容类型有哪些？如果不是下述类型之一，客户端浏览器有可能拒绝接收！！！  **text/html**  **text/css**  **text/plain**  **application/javascript**  **image/png、 image/gif、image/jpeg、....**  **font/ttf、font/woff、font/woff2、....**  **application/json**  **audio/mpeg3、audio/wav、.....**  **video/mpeg4、video/avi、....** |
| 行业知识： HTTP协议规定客户端发给服务器的“请求消息主体”允许的内容类型有哪些？如果不是下述四种类型之一，服务器有可能拒绝接收！！！  **text/plain** —— 未经编码的普通文本，当前所有的WEB服务器都是拒绝接收！！！  **application/x-www-form-urlencoded** —— 对请求主体进行编码(encodeURI())后再提交给服务器  **multipart/form-data** —— 如果提交的请求主体中有“文件上传域”必须使用此内容类型  **application/json** —— 客户端请求数据是用JSON.stringify()序列化后的JSON字符串，如果客户端提交的请求内容类型是此格式，服务器必须做特殊的设置，否则的话默认是拒绝接收 |
| 高级面试题：HTTP协议里“简单请求”和“复杂请求”的区别？  提示：复杂请求，都需要“预取请求（preflight request）” |

**重要&困难知识点：服务器端接口数据请求的二次封装**

为什么要封装服务器端接口的数据请求？

① 对服务器端数据URL进行统一管理 —— 要做到每个域名/接口地址在整个项目中仅出现一次

② 对底层使用的AJAX异步请求技术进行封装 —— 方便以后随时替换为其它的请求技术

③ 对请求之前/之后都要进行的“辅助功能——横切关注点”进行统一封装管理 —— 例如：弹出/隐藏“加载中”对话框、记录请求耗时、记录请求日志....

**九、uni-app项目的最终发布部署**

项目的两种运行模式：

**①调试模式/运行模式：** 代码没有经过压缩和优化，所以代码体积比较大 —— 会自动连接开发服务器，自动更新

**②生产模式/发行模式：** 代码经过优化和压缩，所以代码体积比较小 —— 不能再连接开发服务器

|  |  |
| --- | --- |
| 补充：ES6中的模块导出和导入 | |
| **默认导出(Default Export)** | **命名导出(Named Export)** |
| x.js  export default { //默认导出每个模块只允许用一次  age: 10,  f1(){ }  } | x.js  export let age = 20  export let f2 = function(){ }  //命名导出在每个模块中可以使用多次 |
| y.js  import abc from './x'  abc.age  abc.f1( ) | y.js  import {f2, age} from './x'  age  f2( ) |

|  |  |
| --- | --- |
| 补充：ES7中的async和await关键字  **async：**Asynchronous，异步的，用于修饰某个function，使该函数成为“异步函数” | |
| let f1 = async function( ){  return 123 //返回一个普通数据，会被封装到一个Promise  }  f1( ).then(res=>{ console.log(res) }) | let f1 = async function( ){  return new Promise(...) //直接返回此Promise  }  f1( ).then(res=>{ console.log(res) }) |
| **await：**等待，可以放在任何表达式之前，会要求解释器等到当前表达式执行完才能继续往下执行  注意：await所在的函数必须声明为“async” | |
| let f2 = async function(){  let x = await 1+2  console.log(x)  }  f2() | let f2 = async function(){  let x = await new Promise( )  //此处会等待Promise执行完成，x就是resolve提供的数据  console.log(x)  }  f2() |

课后任务：

①完成“用户注册”组件中的“注册”功能

②完成“用户登录”组件中的“登录”功能；登录成功后，跳转到“用户档案”组件

③完成“用户档案”组件，挂载时，异步请求当前登录用户的基本信息，展示在页面中