

# 实验 1：第一个微信小程序

本实验来自于周文洁老师的《微信小程序开发实战》第二章。包括两个创建小程序的项目实例，一是使用快速启动模板创建小程序。二是手动从零开始创建小程序。这两个项目的主要功能都是点击按钮获取当前微信用户的头像和昵称。

## 2.1 自动生成小程序

为方便初学者入门，微信开发者工具提供了快速启动模板帮助开发者自动生成代码，从而创建完成第一个示例小程序。下面使用开发者工具正式创建第一个微信小程序。

### 2.1.1 项目创建

双击微信开发者工具图标，管理员或开发者使用微信扫描二维码后进入菜单画面。选择菜单中的“小程序”选项进入小程序项目管理页面，如图 2-1 所示。



视频讲解



图 2-1 新建小程序项目

此时开发者依次填写项目名称、目录和 AppID 就可以新建一个小程序项目。填写时的注意事项如下。

- 项目名称：由开发者自定义一个项目名称，该名称仅供开发者工具管理使用，不是小程序被用户看到的名字。
- 目录：项目文件存放的路径地址，可以单击输入框右侧的箭头按钮在计算机盘符中选择指定的目录地址。该地址可以由开发者自定义。
- AppID：管理员在微信公众平台上注册的小程序 ID。

其中小程序的 AppID 可以登录微信公众平台(<https://mp.weixin.qq.com>)查看，具体查看步骤是单击左侧菜单中的“开发”选项，切换至“开发设置”面板，查看“开发者 ID”下方的 AppID(小程序 ID)，如图 2-2 所示。



图 2-2 查看小程序 ID

将该小程序 ID 复制、粘贴到图 2-1 所示的 AppID 输入框中。  
填充以后的效果如图 2-3 所示。



图 2-3 小程序项目填写效果示意图

其中 AppID 必须填实际的小程序 ID, 否则部分功能将无法使用。如果开发者暂时条件受限无法注册申请小程序 ID, 可以使用 AppID 输入框下方的“测试号”来体验小程序。

项目目录选择空白文件夹 tmplDemo, 则开发者工具会默认自动生成代码形成一个简单的小程序效果以供初学者入门学习。“后端服务”选择“不使用云服务”, 其他选项保持默认不变。

填写完毕后单击“新建”按钮完成操作, 此时会跳转到开发页面, 如图 2-4 所示。

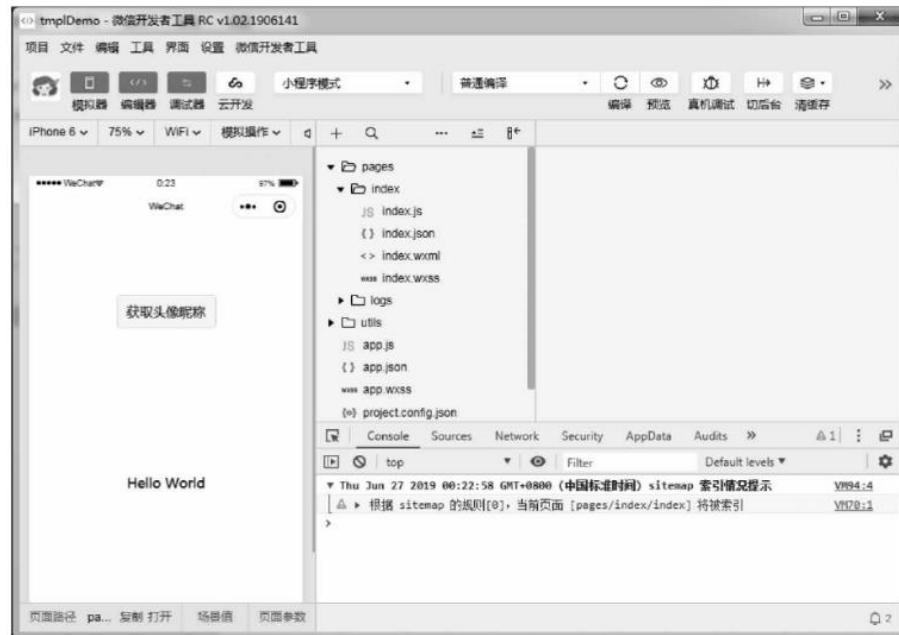


图 2-4 小程序项目开发页面

图中左边就是手机预览效果图, 由图可见目前能显示出来一个“获取头像昵称”按钮和一个“Hello World”文本, 这与手机运行的效果完全相同。

用户可以直接在上面用鼠标左键单击来模拟手指在手机屏幕上的触摸效果, 如图 2-5 所示。



图 2-5 小程序项目单击效果

其中图 2-5(a)显示的是用鼠标左键单击按钮来模拟手机上的触摸点击效果；图 2-5(b)为单击之后弹出的微信授权信息，只有单击“允许”按钮才可以获得数据；图 2-5(c)为最终显示效果，由该图可见小程序项目已经成功地获取了开发者头像和昵称信息。

## 2.1.2 真机预览

除了可以直接在计算机端使用鼠标模拟手机触屏的点击效果外，还可以直接在真机上进行程序预览。用鼠标左键单击“预览”按钮，即可自动生成一个预览二维码，如图 2-6 所示。



视频讲解



图 2-6 小程序项目生成预览二维码

此时用手机微信扫描图 2-6 中的二维码即可进行真机预览，如图 2-7 所示。

由图可见，画面效果基本上和计算机端的预览图一致。另外，预览所用的二维码不是永久有效，用户要注意它的过期时间，一旦过期需要重新单击“预览”按钮生成新的预览二维码。



图 2-7 小程序项目的真机预览效果

## 2.2 手动创建小程序

在熟悉了小程序的目录结构和微信开发者工具的基本使用方法后，读者不妨尝试自己动手创建第一个小程序。

## 2.2.1 项目创建

本项目创建选择空白文件夹 myDemo，效果如图 2-8 所示。

此时仍然会自动生成模板代码，2.2.2 节将介绍如何手动修改页面配置文件。



视频讲解



图 2-8 小程序项目填写效果图

## 2.2.2 页面配置

### 1 创建页面文件

项目创建完毕后，在根目录中会生成文件夹 pages 用于存放页面文件。



视频讲解

一般来说首页默认命名为 index，表示小程序运行的第一个页面；其他页面名称可以自定义。本项目只需要保留首页(index)即可。

具体操作如下：

(1) 将 app.json 文件内 pages 属性中的“pages/logs/logs”删除，并删除上一行末尾的逗号。

(2) 按快捷键 Ctrl+S 保存当前修改。

## 2 删除和修改文件

具体操作如下：

(1) 删除 utils 文件夹及其内部所有内容。

(2) 删除 pages 文件夹下的 logs 目录及其内部所有内容。

(3) 删除 index.wxml 和 index.wxss 中的全部代码。

(4) 删除 index.js 中的全部代码，并且输入关键词“page”找到第二个选项按回车键让其自动补全函数(如图 2-9 所示)。



图 2-9 输入关键词创建 Page 函数

(5) 删除 app.wxss 中的全部代码。

(6) 删除 app.js 中的全部代码，并且输入关键词“app”找到第一个选项按回车键让其自动补全函数(如图 2-10 所示)。

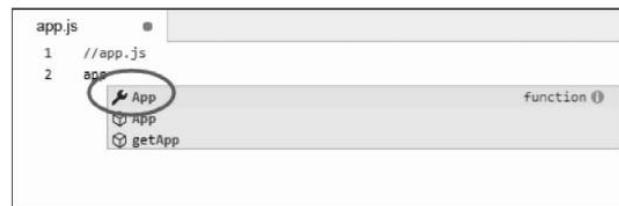


图 2-10 输入关键词创建 App 函数

此时模板代码就修改完毕了，如果未来开发者还想创建其他页面，只需在 app.json 的 pages 属性中追加声明路径即可自动生成对应的页面文件。

## 2.2.3 视图设计

### 1 导航栏设计

小程序默认导航栏是黑底白字的效果，可以通过在 app.json 中对 window 属性进行重新配置来自定义导航栏效果。更改后的 app.json 文件代



视频讲解

码如下：

```
1.  {
2.      "pages": [
3.          "pages/index/index"
4.      ],
5.      "window": {
6.          "navigationBarBackgroundColor": "#663399",
7.          "navigationBarTitleText": "手动创建第一个小程序"
8.      }
9.  }
```



图 2-11 自定义导航栏效果

上述代码可以更改导航栏背景色为紫色、字体为白色，效果如图 2-11 所示。

## 2 页面设计

页面上主要包含 3 个内容，即微信头像、微信昵称和“点击获取头像和昵称”按钮。页面设计图如图 2-12 所示。

计划使用如下组件。



视频讲解

- 微信头像：`<image>`(图像)组件；
- 微信昵称：`<text>`(文本)组件；
- 按钮：`<button>`(按钮)组件。

相关 WXML(pages/index/index.wxml)代码片段如下：

```
1.  <view class = 'container'>
2.      <image></image>
3.      <text>Hello World</text>
4.      <button>点击获取头像和昵称</button>
5.  </view>
```

相关 WXSS(pages/index/index.wxss)代码片段如下：

```
1.  .container{
2.      height: 100vh;           /* 高 100 视窗, 这里写 100 % 无效 */
3.      display: flex;         /* flex 布局模式 */
4.      flex-direction: column; /* 垂直布局 */
5.      align-items: center;    /* 水平方向居中 */
6.      justify-content: space-around; /* 垂直方向分散布局 */
7.  }
```

当前效果如图 2-13 所示。

由图可见，此时可以显示文本和按钮。由于尚未获得头像图片，所以无法显示内容。可以临时使用本地图片代替，在点击按钮获取微信头像后再更改。

在项目中新建自定义文件夹 images 用于存放图片，右击此文件夹，选择“硬盘打开”，将本地图片 logo.png 复制、粘贴进去等待使用。新目录结构如图 2-14 所示。

修改 WXML 页面的`<image>`组件如下：

```
<image src = '/images/logo.png' mode = 'widthFix'></image>
```

上述代码中 `src` 属性用于指定图片来源为根目录下 images 文件夹中的 logo.png 图片，`mode` 属性表示图片随着指定的宽度自动拉伸高度以显示原图的正确比例。

在 WXSS 页面追加`<image>`和`<text>`组件的相关样式代码如下：

```

1.  image {
2.      width: 300rpx;          /* 图片宽度 */
3.      border-radius: 50%;    /* 4个角变为圆角形状 */
4.  }
5.  text{                      /* 字体大小 */
6.      font-size: 50rpx;
7.  }

```

当前页面效果如图 2-15 所示。

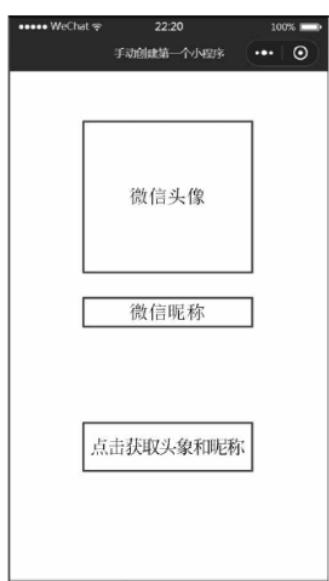


图 2-12 页面设计图

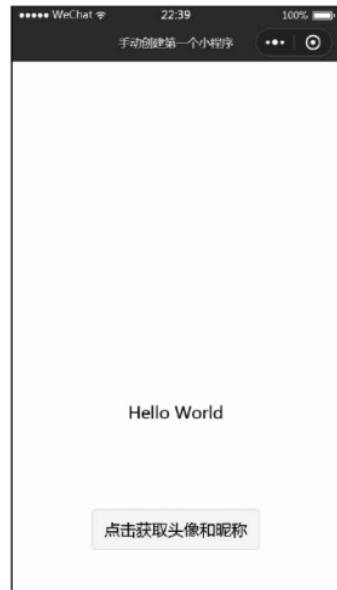


图 2-13 页面设计图

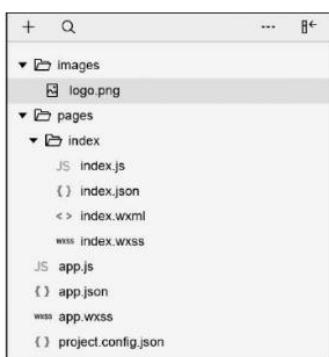


图 2-14 添加图片文件后的目录结构



图 2-15 页面效果图

此时页面布局与样式设计就已完成,2.2.4 节将介绍如何点击按钮获取头像和昵称数据。

## 2.2.4 逻辑实现

### 1 获取微信用户信息

在 WXML 页面修改<button>组件的代码,为其追加获取用户信息事件,代码片段如下:

```
1. <button open-type='getUserInfo' bindgetuserinfo='getMyInfo'>
2.     点击获取头像和昵称
3. </button>
```

其中 open-type='getUserInfo'表示激活获取微信用户信息功能,然后使用 bindgetuserinfo 属性表示获得的数据将传递给自定义函数 getMyInfo,开发者也可以使用其他名称。

在 JS 页面的 Page() 内部追加 getMyInfo 函数,代码片段如下:

```
1. getMyInfo: function(e) {
2.     console.log(e.detail.userInfo)
3. },
```

保存后预览项目,如果点击按钮后 Console 控制台能够成功输出一段数据,就说明获取成功,如图 2-16 所示。

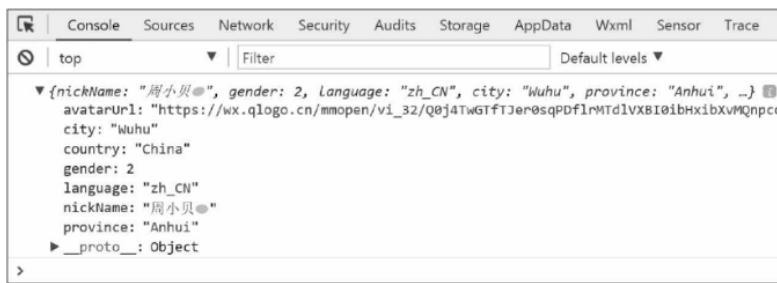


图 2-16 Console 控制台输出内容

### 2 使用动态数据显示头像和昵称

在 WXML 页面修改<image>和<text>组件的代码,将图片来源和文本内容使用双花括号({{}})做成动态数据,代码片段如下:

```
1. <image src='{{src}}' mode='widthFix'
2. ></image>
3. <text>{{name}}</text>
```

同时修改 JS 文件,在现有的 data 属性中追加这两个动态数据的值,使其仍然可以显示原先的内容。代码片段如下:

```
1. data: {
2.     src:'/images/logo.png',
3.     name:'Hello World'
4. },
```



视频讲解



图 2-17 页面效果图

完成后预览如图 2-17 所示, 页面显示效果不会发生改变。

### 3 | 更新头像和昵称

修改 JS 文件中 getMyInfo 函数的代码, 使获取到的信息更新到动态数据上, 代码片段如下:

```
1.     getMyInfo: function(e) {
2.         let info = e.detail.userInfo;
3.         this.setData({
4.             src: info.avatarUrl,           //更新图片来源
5.             name: info.nickName          //更新昵称
6.         })
7.     },
```



视频讲解

此时就已全部完成, 保存后重新预览项目, 最终效果图如图 2-18 所示。



图 2-18 最终效果图

## 2.2.5 完整代码展示

app.json 文件的完整代码如下:

```
1.  {
2.      "pages": [
3.          "pages/index/index"
4.      ],
5.      "window": {
6.          "navigationBarBackgroundColor": "#663399",
7.          "navigationBarTitleText": "手动创建第一个小程序",
8.          "navigationBarTextStyle": "white"
9.      }
10. }
```

WXML 文件(pages/index/index. wxml)的完整代码如下：

```
1.  <view class = 'container'>
2.    <image src = '{src}' mode = 'widthFix'></image>
3.    <text>{name}</text>
4.    <button open - type = 'getUserInfo' bindgetuserinfo = 'getMyInfo'>
5.      点击获取头像和昵称
6.    </button>
7.  </view>
```

WXSS 文件(pages/index/index. wxss)的完整代码如下：

```
1. .container {
2.   height: 100vh;          /* 高 100 视窗,这里写 100 % 无效 */
3.   display: flex;          /* flex 布局模式 */
4.   flex - direction: column; /* 垂直布局 */
5.   align - items: center; /* 水平方向居中 */
6.   justify - content: space - around; /* 垂直方向分散布局 */
7. }
8. image {
9.   width: 300rpx;          /* 图片宽度 */
10.  border - radius: 50%; /* 4 个角变为圆角形状 */
11. }
12. text {
13.   font - size: 50rpx;     /* 字体大小 */
14. }
```

JS 文件(pages/index/index. js)的完整代码如下：

```
1. Page({
2.   /**
3.    * 页面的初始数据
4.    */
5.   data: {
6.     src: '/images/logo.png',
7.     name: 'Hello World'
8.   },
9.   /**
10.  * 自定义函数 -- 获取微信用户信息
11. */
12.  getMyInfo: function(e) {
13.    let info = e.detail.userInfo;
14.    this.setData({
15.      src: info.avatarUrl,           //更新图片来源
16.      name: info.nickName         //更新昵称
17.    })
18.  }
19. })
```

本实验到这里就全部完成了🎉，你完成的怎么样呢？要继续加油啊！😊