

## 《微信小程序综合实践》实验报告四

### 小程序开发—— 小程序基本页面的开发

姓名 熊子宇

专 业 计算机科学与技术

学号 3200105278

联系方式 18255168425

#### 一、实验目的

1. 通过本实验掌握对微信小程序页面的基本配置；
2. 掌握微信小程序的常见视图容器和组件；
3. 通过对视图层 WXML 文件绑定数据，完成对 WXML 文件的数据渲染；
4. 通过对视图层 WXML 添加事件，完成对 WXML 文件的数据渲染；
5. 了解小程序生命周期，掌握小程序生命周期的作用；
6. 对微信小程序 API 有初步了解；

#### 二、实验内容及要求

**准备工作：**为了方便后续的学习和实践，建议同学们自带电脑，在自己电脑上进行实验内容，并完成实验报告。本次实验为基础内容，认真完成本次实验内容即可掌握开发微信小程序的基本工具使用。

##### （一）实验内容

1. 查阅小程序官方文档，掌握小程序页面配置的基本方式，修改配置文件对小程序页面进行配置；
2. 查阅小程序官方文档，掌握小程序常见视图容器和组件；
3. 在小程序页面上绑定数据，并在小程序页面进行渲染；
4. 在小程序组件中添加事件，通过事件使页面具有交互功能；
5. 改写小程序生命周期函数，了解生命周期函数的执行条件、执行顺序和作用；
6. 学习微信小程序 API，使小程序拥有转发和获取登录用户信息等功能；

##### （二）实验步骤（仔细阅读，按照步骤完成实验）

作业上交内容与事项：

1. 实验报告文档：按照要求完成实验并将关键步骤实验结果进行截图记录。

注意文档工整；

2. 程序代码：如果有程序代码或其他相关材料，可汇总压缩所有文件并上传压缩包。注意文件命名；

本期实验作业上期限：

请在实验设置的截止日期内提交实验报告，若逾期提交，成绩会适当被打折。

本次作业上交内容：

- 实验报告文档(.docx)
- 程序代码压缩文档(.zip)

### 三、实验感受及记录

#### （一）实验感受（本次实验遇到的问题、主要收获等内容）

问题：

1. 虽然大致了解了微信小程序的框架和基本功能，但是很多细节并不明白。比如 tab-bar 代码内的具体参数并不了解，转发功能的代码不是很能理解。
2. 对 API 还没有深入了解，仅仅停留在抄示例代码的程度。
3. 一开始不能在有 tab-bar 的情况下同时显示地图，尝试多次未果。后来学习至生命周期函数部分内容后，发现是不能同时在 js 文件中用 Page 和 Component 构造页面，只能使用其中一个。

主要收获：

1. 进一步学习了微信小程序的文件层次框架，能够看懂 JSON、WXML 的大部分内容。
2. 初步学习了 WXML 的容器和组件，初步学习了 map 组件的使用方法。
3. 初步学习了事件绑定、生命周期函数、事件处理函数等，可以写出比较简单的回调函数。

#### （二）实验记录（实验过程中关键步骤截图记录及文字描述）

2.1 一个微信小程序页面有哪些文件？这些文件的作用分别是什么？

JSON 配置文件	静态配置，如包括了小程序的所有页面路径、界面表现、网络超时时间、底部 tab
WXML 文件	描述当前页面的结构
WXSS 文件	描述页面的样式
JS 文件	逻辑交互

2.2 在项目的根目录文件下的 .json 配置文件和自己新建页面的 .json 配置文件有什么区别和联系？

根目录下有 app.json 和 project.config.json。app.json 是当前小程序的全局配置，包括小程序的所有页面路径等。通过修改 window 字段可以定义小程序所有页面的顶部背景颜色、文字颜色定义等。project.config.json 是个性化配置文件。

而页面配置 page.json 类似于 app.json，但是它用于独立定义每一个页面的一些属性。

总的来说，根目录下的.json 配置文件作用于全局，而页面.json 配置文件作用于当前页面。

2.3 常见的视图容器（组件）有哪些？常见的组件又有哪些，请举例说明其作用。

常见的视图容器：view, cover-view/cover-image, movable-area/movable-view, scroll-view, swiper, ...

view 是基础视图，有点类似于 html 中的<div>盒子。

cover-view/image 是覆盖在原生组件之上的文本/图片视图。

movable-area/view 配套使用，可以生成可移动的视图容器。

scroll-view 是可滚动视图区域，可以构造下拉刷新的效果。

swiper 是滑动视图容器。

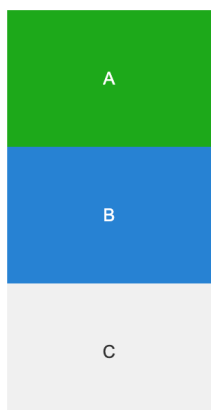
flex-direction: row

横向布局



flex-direction: column

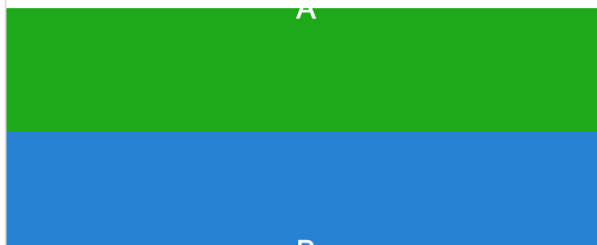
纵向布局



view 视图

Vertical Scroll

纵向滚动



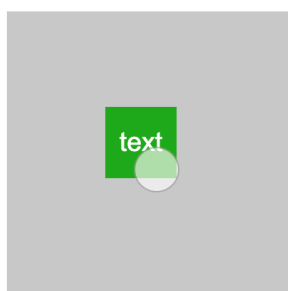
Horizontal Scroll

横向滚动



scroll 视图

movable-view区域小于movable-area



点击移动到 (30px, 30px)

movable 视图



cover-view 视图

常见的组件：

基础内容：icon（图标）, progress（进度条）, rich-text（富文本，可以用 HTML String 或者 Node 渲染）, text

表单组件：包括 form, input, button, radio 等众多表单组件，用于制作表单，与 html 的表单功能相似。

媒体组件：audio（音频）, camera（系统相机）, image（图片）, live-player/live-pusher(实时音视频播放/录制), video（视频）

其他：map（地图）, canvas（画布）

2.4 小程序的生命周期函数有哪几种，它们的执行条件和执行顺序是什么？

onLoad()

页面加载时触发。一个页面只会调用一次。

onShow()

页面显示/切入前台时触发。

onReady()

页面初次渲染完成时触发。一个页面只会调用一次，代表页面已经准备妥当，可以和视图层进行交互。

onHide()

页面隐藏/切入后台时触发。如 wx.navigateTo 或底部 tab 切换到其他页面，小程序切入后台等。

onUnload()

页面卸载时触发。如 wx.redirectTo 或 wx.navigateBack 到其他页面时。

执行顺序如下（摘自官方文档）

Tab 切换对应的生命周期（以 A、B 页面为 Tabbar 页面，C 是从 A 页面打开的页面，D 页面是从 C 页面打开的页面为例）：

当前页面	路由后页面	触发的生命周期（按顺序）
A	A	Nothing happend
A	B	A.onHide(), B.onLoad(), B.onShow()
A	B（再次打开）	A.onHide(), B.onShow()
C	A	C.onUnload(), A.onShow()
C	B	C.onUnload(), B.onLoad(), B.onShow()
D	B	D.onUnload(), C.onUnload(), B.onLoad(), B.onShow()
D（从转发进入）	A	D.onUnload(), A.onLoad(), A.onShow()
D（从转发进入）	B	D.onUnload(), B.onLoad(), B.onShow()

## 2.5 如何为一个组件添加事件，它的处理函数应该如何定义？

在渲染层的组件中加入事件绑定 `bindtap`，当事件被触发时，就会执行 Page 中定义的事件处理函数。

例如，`<view bindtap="viewTap"> click me </view>`

再在 js 文件中如下定义：

```
Page({
  viewTap: function() {
    console.log('view tap')
  }
})
```

2.6 在下方附上点击「移动位置」后地图组件在真机上的截图和点击分享页面后的截图，并提交完成实验步骤后的相关源代码文件。

