第二音 小程序框架

本章内容主要包含两个使用 flex 布局模型创建的小程序项目实例,一是仿微信"发现"页 面创建列表布局小程序;二是仿微信"钱包"页面创建九宫格布局小程序。

Chapter 3

本章学习目标

- 学习使用 flex 布局模型和 wx:for 属性创建列表布局小程序;
- 学习使用 flex 布局模型和 wx:for 属性创建九宫格布局小程序。

〇〇 3.1 列表布局小程序

微信 App 的"发现"页面是由若干个垂直排列的列 表组成的,每个列表项均包含图标、文字和箭头符号,如 图 3-1 所示。

本项目将使用 flex 布局模型和 wx: for 属性仿微信 "发现"页面实现列表布局效果。

3.1.1 项目创建

本项目创建选择空白文件夹 wxDiscover,效果如图 3-2 所示。

单击"新建"按钮完成项目创建,然后 准备手动创建页面配置文件。

3.1.2 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。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。

视频讲解

视频讲解

(2) 按快捷键 Ctrl+S 保存当前修改。

	•III 中国F	l信 🗢	下午9:4	2	● 🕈 🖇 14% 🔲
			发现		
	0	朋友圈			>
	۰.	扫一扫			>
	8	摇一摇			>
	\$	看一看			>
	六	搜一搜			>
	U	购物			>
ļ	\diamond	游戏			>
	ა	小程序			>
	〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇	و المعالم المع الم المعالم معالم مع	2 = π₹	シ发现	<u>گ</u>

图 3-1 微信 App"发现"页面真机截屏

4	٥X		新建项目 导入项目
小程序项目			
小程序		项目名称	wxDiscover
小游戏		目录	E:\wxdemo_workspace\wxDiscover ~
代码片段		AppID	wx19079110a0f01e8a ~
公众号网页项目			若无 AppID 可 注册 或使用 测试号
公众号网页		开发模式	小程序
		后端服务	◎ 不使用云服务
			○ 小程序·云开发
			小程序,云开发为开发者提供数据库、存储和云函数等完整的云端支持。无需搭 建服务器,使用平台提供的 API进行核心业务开发,即可实现小程序快速上线和 迭代。了解详情
			○ 腾讯云
		语言	JavaScript
	注销》		取消新建

图 3-2 小程序项目填写效果示意图

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 3-3 所示)。



图 3-3 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 3-4 所示)。

3 创建其他文件

接下来创建其他自定义文件,本项目还需要一个文件夹用于存放图标素材。文件夹名称 由开发者自定义(例如 images),单击目录结构左上角的+号创建文件夹并命名为 images。

app.js	•	
1 //a	opp.js	
2 apa	\frown	
C	& App	function (
	ФАрр	
	() getApp	

图 3-4 输入关键词创建 App 函数

本项目将用到8个列表项图标和1个通用箭头图标,图片素材如图3-5所示。



右击目录结构中的 images 文件夹,选择"硬盘打开",将图片复制、粘贴进去。

全部完成后的目录结构如图 3-6 所示。

此时文件配置就全部完成,3.1.3节将正式进行页面 布局和样式设计。

3.1.3 视图设计

1 导航栏设计



视频讲解

视频讲解

果,因此只需要在 index. json 中自定义导 航栏标题即可。更改后的 index. json 文件代码如下:

⊥.	1	
2.		"navigationBarTitleText": "发现"
3.	}	

小程序默认导航栏就是黑底白字的效

上述代码可以更改当前页面的导航栏 标题文本为"发现",效果如图 3-7 所示。

2 页面设计

页面上主要包含 5 组列表,每组列表 包含 1~2 个列表项,具体内容解释如下。

- 列表组1:"朋友圈"单行列表项;
- 列表组 2: "扫一扫"和"摇一摇"两行列表项;
- 列表组 3: "看一看"和"搜一搜"两行列表项;

			04
+ Q			04
🔻 🗁 images			
🖂 arrow.	ong		
杞 games	.png		
전 miniPr	ogram.png		
🛛 mome	nts.png		
🛃 scan.p	ng		
🛃 search	.png		
🛃 shake.	png		
🛃 shopp	ng.png		
🛃 topSto	ries.png		
🔻 🗁 pages			
🔻 🗁 index			
JS inde	c.js		
<pre>{ } inde:</pre>	c.json		
<> inde:	c.wxml		
wxss inde	(.WXSS		
JS app.js			
<pre>{ } app.json</pre>			
wxss app.wxs	S		
{} project.c	onfig.json		

图 3-6 页面文件创建完成





• 列表组 5: "小程序"单行列表项。

每个列表组之间需要有一定的间隔距离,设计图如 图 3-8 所示。

计划使用如下组件。

- 页面整体: < view >组件,并定义 class='container';
- 列表组: < view >组件,并定义 class='listGroup';
- 列表项单行: < view >组件,并定义 class='listItem';
- 列表图标: < image >(图像)组件:
- 列表文字内容: < text >(文本)组件;
- 箭头图标: < image >(图像)组件。

首先定义页面容器(< view >),WXML(pages/index/index.wxml)代码片段如下:

```
1. < view class = 'container'>
```

```
2. </view>
```

WXSS(pages/index/index.wxss)代码片段如下:

```
1. /*背景容器样式*/
```

```
2. .container{
```

- 3. height: 100vh;
- background color: silver;

```
5.
   display: flex;
```

- 6. flex - direction: column;
- /*高度为100视窗,写成100%无效*/
- /*背景颜色为银色*/ / * flex 布局模型 * /

7. }

/ * 垂直布局 * /

当前效果如图 3-9 所示。



图 3-8 页面设计图



图 3-9 当前页面预览效果

由图可见,此时整个页面背景变成了银色。由于还没添加组件元素,所以尚看不出来 flex 布局模型效果。

接下来以第一个列表选项为例,WXML(pages/index.wxml)代码片段修改如下:

```
1. < view class = 'container'>
2.
      < view class = 'listGroup'>
3.
        < view class = 'listItem'>
4.
          < image src = '/images/moments.png'></image>
5.
          <text>朋友圈</text>
          < image src = '/images/arrow.png'></image>
6.
7.
        </view>
8.
      </view>
9. </view>
```

WXSS(pages/index/index.wxss)代码片段如下:

1.	/ * 列表组样式 * /	
2.	.listGroup{	
3.	background - color: white	; / * 背景颜色为白色 * /
4.	<pre>margin: 20rpx 0;</pre>	/*上下外边距 20rpx,左右 0*/
5.	}	
6.	/ * 列表项单行样式 * /	
7.	.listItem{	
8.	display: flex;	/ * flex 布局模型 * /
9.	<pre>flex - direction: row;</pre>	/ * 水平布局 * /
10.	align-items: center;	/ * 垂直方向居中 * /
11.	border: 1rpx solid silver	;; / * 1rpx 宽的银色实线边框 * /
12.	<pre>padding: 10rpx;</pre>	/*内边距10rpx*/
13.	}	
14.	/ * 图标的尺寸 * /	
15.	image{	
16.	width: 80rpx;	/ * 宽度 * /
17.	height: 80rpx;	/*高度*/
18.	<pre>margin: 0 15rpx;</pre>	/*上下外边距 0,左右外边距 15rpx*/
19.	}	Machat o 0.51
20.	/ * 文本样式 * /	wechat 😤 0.51
21.	text{	反现 — — — — — — — — — — — — — — — — — — —
22.	font-size: 40rpx;	/* 字体大小 40rpx * /
23.	flex-grow: 1;	/* 扩张多余空间苋度 * / 🛛 😽 🐻
24.	}	

当前效果如图 3-10 所示。

由图可见,此时可以显示第一个列表组的内容。用同 样的方式追加后续的列表组即可实现完整效果。当然也可 以暂时不追加其他列表项,使用 3.1.4 节介绍的方法减少 工作量。



1 使用动态数据展示列表

由于所有列表项的内容布局都是统一 的,可以考虑使用 wx: for 属性配合动态数组 渲染全部列表项,以减少 WXML 页面的代码量。





>

图 3-10 当前页面预览效果



修改 WXML(pages/index.wxml)页面代码如下:

```
1. < view class = 'container'>
```

```
2. < view class = 'listGroup' wx:for = '{{list}}' wx:for - item = 'group' wx:key = 'group{{index}}'>
```

```
3. <view class = 'listItem' wx:for = '{{group}}' wx:for - item = 'row' wx:key = 'row{{index}}'>
```

```
< image class = 'icon' src = '{{row.icon}}'></image>
```

```
5. <text>{{row.text}}</text>
```

- 6. <image src = '/images/arrow.png'></image>
- 7. </view>
- 8. </view>
- 9. </view>

4.

上述代码表示将使用双重 wx:for 属性循环显示全部列表项,其中{{list}}数组用于表示 5 个列表组,并为每个列表组起了别名 group;每个列表项也起了别名 row,列

表项的图标和文本分别命名为 icon、text。这里均为自定义名称,开发者可以 自行更改。

2 补充数组完整信息

在 index. js 的 data 属性中添加 list 数组,JS 文件(pages/index/index. js) 代码如下:



视频讲解

1. Page({
2. data: {
3. list:[
4. //第1组列表
5. [{ text: '朋友圈', icon: '/images/moments.png'}
6. //第2组列表
7. [
8. { text: '扫一扫', icon: '/images/scan.png' },
9. { text: '摇一摇', icon: '/images/shake.png' }
10.],
11. //第3组列表
12. [
13. { text: '看一看', icon: '/images/topStories.
png'},
14. { text: '搜一搜', icon: '/images/search.png' }
15.],
16. //第4组列表
17. [
18. { text: '购物', icon: '/images/shopping.png' },
19. { text: '游戏', icon: '/images/games.png' }
20.],
21. //第5组列表
22. [{ text: '小程序', icon: '/images/miniProgram.
png' }]
23.]
24. }
25. })

此时就已全部完成,保存后重新预览项目,最终效果图 如图 3-11 所示。

3.1.5 完整代码展示

app.json 文件的完整代码如下:

••••• WeChat 奈	0:52	100%
	发现	••• •
🚷 朋友圈		>
ち 扫—扫		>
♪ 摇一摇		>
🗘 看一看		>
た捜ー捜		>
□ 购物		>
♦ 游戏		>
⑦ 小程序		>

图 3-11 最终效果图

```
1. {
2. "pages": [
3. "pages/index/index"
4. ]
5. }
```

JSON 文件(pages/index/index.json)的完整代码如下:

{
 "navigationBarTitleText": "发现"
 }

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

```
1. < view class = 'container'>
     < view class = 'listGroup' wx:for = '{{list}}' wx:for - item = 'group' wx:key = 'group{{index}}'>
2
3.
        <view class = 'listItem' wx:for = '{{group}}' wx:for - item = 'row' wx:key = 'row{{index}}'>
4.
          < image src = '{{row. icon}}'></image>
5
          <text>{{row.text}}</text>
          < image src = '/images/arrow.png'></image>
6.
7.
        </view>
8.
      </view>
9. </view>
```

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

```
1. /*背景容器样式*/
2. .container{
                                      /*高度为100视窗,写成100%无效*/
3.
    height: 100vh;
                                      /*背景颜色为银色*/
4.
    background - color: silver;
5.
                                      /*flex布局模型*/
   display: flex;
6.
   flex - direction: column;
                                      /*垂首布局*/
7. }
8. /*列表组样式*/
9. .listGroup{
                                     /*背景颜色为白色*/
10. background - color: white;
11. margin: 20rpx 0;
                                      /*上下外边距 20rpx, 左右 0*/
12. }
13. / * 列表项单行样式 * /
14. .listItem{
15. display: flex;
                                      /* flex 布局模型 * /
16. flex-direction: row;
                                      /*水平布局*/
17. align - items: center;
                                      /*垂直方向居中*/
18.
   border: 1rpx solid silver;
                                      /*1rpx 宽的银色实线边框*/
19. padding: 10rpx;
                                      /*内边距10rpx*/
20. }
21. /*图标的尺寸*/
22. image{
23. width: 80rpx;
                                      /*宽度*/
24. height: 80rpx;
                                      /*高度*/
25. margin: 0 15rpx;
                                      /*上下外边距 0,左右外边距 15rpx */
26. }
27. / * 文本样式 * /
28. text{
                                      /*字体大小 40rpx*/
29. font - size: 40rpx;
                                      /*扩张多余空间宽度*/
30.
   flex - grow: 1;
31. }
```



```
1.
   Page({
2.
      data: {
3.
        list: [
          //第1组列表
4
          [{ text: '朋友圈', icon: '/images/moments.png' }],
5.
          //第2组列表
6.
7.
          ſ
            { text: '扫一扫', icon: '/images/scan.png' },
8
9.
            { text: '摇一摇', icon: '/images/shake.png' }
10.
          1,
          //第3组列表
11
12.
          ſ
13.
            { text: '看一看', icon: '/images/topStories.png' },
14.
            { text: '搜一搜', icon: '/images/search.png' }
15.
          ],
16.
          //第4组列表
17.
          ſ
            { text: '购物', icon: '/images/shopping.png' },
18.
            { text: '游戏', icon: '/images/games.png' }
19.
20.
          1,
21.
          //第5组列表
          [{ text: '小程序', icon: '/images/miniProgram.png' }]
22
23.
        ]
24.
      }
25. })
```

九宫格布局小程序 3.2

微信 App"钱包"页面主要分为上、下两个部分,上 面是由"收付款""零钱"和"银行卡"组成的钱包状态 栏,下面是由九宫格组成的"腾讯服务"栏,每个格子里 面包含图标和下方的文字说明,如图 3-12 所示。

本项目将使用 flex 布局模型和 wx: for 属性仿微 信"钱包"页面实现九宫格布局效果。

项目创建 3.2.1

本项目创建选择空白文件夹 wxWallet,效果如图 3-13 所示。

单击"新建"按钮完成项目创建,然后 准备手动创建页面配置文件。

页面配置 3.2.2

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。一般

•■ 中国电信 🗢	下午9:27	@ 🕈 🕏 18% 🥅
〈 我	钱包	
[~]	¥	
收付款	零钱 ¥0.00	银行卡
腾讯服务		
	¥	٢
信用卡还款	手机充值	理财通
(÷)	\bigcirc	Å
生活缴费	Q币充值	城市服务
3	Ð	
腾讯公益	保险服务	
限时推广		
	<u> 105%</u>	D X

视频讲解

视频讲解

图 3-12 微信 App" 钱包" 页面真机截屏

	۰×		新建项目 导入项目
小程序项目			
小程序		项目名称	wxWallet
小游戏		目录	E:\wxdemo_workspace\wxWallet ~
代码片段		AppID	wx19079110a0f01e8a ~
公众号网页项目			若无 AppID 可 注册 或使用 测试号
公众号网页		开发模式	小程序
		后端服务	◎ 不使用云服务
			 小程序·云开发 小程序·云开发为开发者提供数据库、存储和云函数等完整的云端支持。无需搭建服务器,使用平台提供的 API 进行核心业务开发,即可实现小程序快速上线和 迭代。了解详情 防讯云
		语言	JavaScript
3	注销 >		取消新建

图 3-13 小程序项目填写效果示意图

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 3-14 所示)。

(5) 删除 app. wxss 中的全部代码。

index.js			
1 //i	index.js		
2 pag	ge		
	Se nage	ScrollTo	
	🔑 Page	•)	function ()
	() Page	2	
	𝔅 get(CurrentPages	
	≣⊒ pack	cage	
	🔑 paus	seBackgroundAudio	
	🔑 pre	/iewImage	
	🔑 can	/asPutImageData	

图 3-14 输入关键词创建 Page 函数

(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 3-15 所示)。

1 //app.js 2 app
2 app
(App)
TO APP

图 3-15 输入关键词创建 App 函数

3 创建其他文件

接下来创建其他自定义文件,本项目还需要一个文件夹用于存放图标素材。文件夹名称 由开发者自定义(例如 images),单击目录结构左上角的+号创建文件夹并命名为 images。

由于本项目用到的图标素材较多,将在 images 文件夹下分为两个二级目录放置。

• top:顶端钱包状态栏的图标共3个,如图 3-16 所示。







(a) money.png

图 3-16 顶端钱包状态栏图标素材展示

• service: "腾讯服务"栏的九宫格图标共 8个, 如图 3-17 所示。

	¥	\mathbf{O}	(±)
(a) cardRepay.png	(b) mobileTopup.png	(c) wealth.png	(d) utilities.png
\bigcirc	Å	3	Ð
(e) qqCoins.png	(f) publicService.png	(g) charity.png	(h) insurance.png
	图 3-17 "腾讯服务	"栏图标素材展示	

右击目录结构中的 images 文件夹,选择"硬盘打开",将二级目录和对应的图标文件全部 复制、粘贴进去。完成后的目录结构如图 3-18 所示。

此时文件配置就全部完成,3.2.3节将正式进行页面布局和样式设计。

3.2.3 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,因此需要在 index. json 中自定义导航栏标题和背景颜色。更改后的 index. json 文件代码如下:







3. "navigationBarBackgroundColor": " # 686F79"

4. }

上述代码可以更改当前页面的导航栏标题文本为"钱包"、背景颜色为灰色(#686F79)。 预览效果如图 3-19 所示。

+	Q		₽←
• 🖻	🤊 images		
•	🗁 service		
	🖂 cardRepay.png		
	🖾 charity.png		
	insurance.png		
	MobileTopup.png		
	DublicService.png		
	dqCoins.png		
	utilities.png		
	wealth.png		
•	🗁 top		
	lalance.png		
	🔁 cards.png		
	🖾 money.png		
▼ 12	∋ pages		
•	🗁 index		
	JS index.js		
	{ } index.json		
	<> index.wxml		
	wxss index.wxss		
JS	5 app.js		
{	} app.json		
wx	ss app.wxss		
{	} project.config.json		
	图 2 10 丙丙六份 4	[2] · · · · · · · · · · · · · · · · · · ·	÷

2 页面设计

页面上主要包含两个面板,具体内容解释如下。

- 面板1(顶端钱包状态栏):包含"收付款""零钱"和"银行卡"3个方格,每个方格中均有 图标、文本,其中"零钱"还包括第二行副文本。
- 面板 2("腾讯服务"栏):包含第一行标题和下方的九宫格区域,其中共有 8 个方格有 内容,包括图标和文本。

注意,面板之间需要有一定的间隔距离,设计图如图 3-20 所示。

计划使用如下组件。

- 页面整体: < view >组件,并定义 class='container';
- 面板 1: < view >组件,并定义 class='topPanel';
- 面板 1 方格: < view >组件,并定义 class='box1';
- 面板 2: < view >组件,并定义 class='servicePanel';
- 面板 2 标题: < view >组件,并定义 class='serviceTitle';
- 面板 2 九宫格区域: < view >组件,并定义 class='serviceBlocks';
- 面板 2 方格: < view >组件,并定义 class='box2';
- 方格内图标: < image >(图像)组件;

• 方格内文字内容: < text >(文本)组件。

1) 整体布局设计

首先定义页面容器(<view>),WXML(pages/index.wxml)代码片 段如下:

1. < view class = 'container'>

2. </view>

WXSS(pages/index/index.wxss)代码片段如下:

- 1. /*1背景容器样式*/
- 2. .container{
- 3. height: 100vh;
- 4. background color: silver;
- display: flex;
- flex direction: column;
- 7. }

当前效果如图 3-21 所示。

WeChat 🔶	14:23	92% 🔲		••••• WeChat 穼	14:37
	钱包	••• 0			钱包
顶或	端钱包状态栏				
刊服务					
	九宫格				
]		
图 3-2	0 页面设计	图			

图 3-21 当前页面预览效果

/*高度为100视窗,写成100%无效*/

/*背景颜色为银色*/ /*flex布局模型*/

/*垂直布局*/

由图可见,此时整个页面背景变成了银色。由于还没添加组件元素,所以尚看不出来 flex 布局模型效果。

然后继续添加两个面板组件,WXML(pages/index.wxml)代码片段修改如下:

```
1. < view class = 'container'>
```

2. <view class = 'topPanel'>

```
3. </view>
```

- 4. < view class = 'servicePanel'>
- 5. <view class = 'serviceTitle'>腾讯服务</view>



 \odot

```
6. < view class = 'serviceBlocks'></view >
7. </view >
8. </view >
```

WXSS(pages/index/index.wxss)代码片段如下:

```
1. /*2 面板 1: 顶端状态栏*/
2. .topPanel {
                                  /*高度*/
    height: 300rpx;
3.
                                  /*背景颜色为灰色*/
4
     background - color: # 686F79;
5.
                                  /*flex 布局模型*/
     display: flex;
6.
   flex - direction: row;
                                  /*水平布局*/
7. }
8. /*3 面板 2: "腾讯服务"栏*/
9. .servicePanel {
                                  /*最小高度*/
10. min-height: 600rpx;
11. background - color: white;
                                  /*背景颜色为白色*/
12.
   margin: 20rpx 0;
                                  /*上下外边距 20rpx, 左右 0*/
13. }
14. / * 3-1 面板 2: 第一行标题样式 * /
15. .serviceTitle {
                                  /*四周内边距 20rpx*/
16. padding: 20rpx;
17. border: 1rpx solid silver;
                                  /*1rpx宽的银色实线边框*/
18. font - size: 30rpx;
                                  /* 字号为 30rpx 大小*/
19.
   color: gray;
                                  /*字体颜色为灰色*/
20. }
21. / * 3-2 面板 2: 九宫格区域样式 * /
22. .serviceBlocks {
                                  /*flex 布局模型*/
23. display: flex;
24. flex-direction: row;
                                  /*水平布局*/
                                  /* 允许换行*/
25.
    flex - wrap: wrap;
26. }
```

当前效果如图 3-22 所示。

由图可见,此时可以显示两个面板的布局位置、背景颜色以及面板 2 的标题。

2) 面板 1 方格设计

接下来制作面板 1 的方格内容,以其中左边第一个方格"收付款"内容为例,WXML(pages/index/index.wxml)代码片段修改如下:



视频讲解

```
1. < view class = 'container'>
2.
     < view class = 'topPanel'>
3.
       < view class = 'box1'>
4.
          < image src = '/images/top/money.png'></image>
5.
          <text>收付款</text>
6.
        </view>
7.
      </view>
8.
      ...
9. </view>
```

WXSS(pages/index/index.wxss)代码片段如下:

```
    /*2-1 面板 1: 方格样式 */
    .box1 {
    display: flex; /* flex 布局模型 */
    flex-direction: column; /* 垂直布局 */
```



5.	align-items: center;	/ * 水平方向居中 * /
6.	width: 33 %;	/ * 宽度约占屏幕的 1/3 * /
7.	height: 250rpx;	/ * 高度 * /
8.	}	
9.	/*2-2 面板 1: 方格内图标样式*/	
10.	.box1 image{	
11.	width: 110rpx;	/ * 宽度 * /
12.	height: 110rpx;	/ * 高度 * /
13.	margin: 20rpx;	/ * 四周外边距均为 20rpx * /
14.	}	
15.	/*2-3 面板 1: 方格内文本样式*/	
16.	.box1 text{	
17.	text-align: center;	/ * 文本居中 * /
18.	color: white;	/ * 字体颜色为白色 * /
19.	font-size: 35rpx;	/*字号为 35rpx 大小 */
20.	}	

当前效果如图 3-23 所示。

🚥 WeChat 🔶	15:22	72% 🛄
	钱包	••• • •
腾讯服务		
图 3_22	出前五百羽	5.监动里

WeChat ◆ 15:27 70% ■
 低包
 ぜ<
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 <li

由图可见,此时可以显示面板1的"收付款"方格内容,包括图标和文本。用同样的方式追加其他方格即可实现完整效果。当然也可以暂时不追加,使用3.2.4节介绍的方法减少工作量。

3) 面板 2 方格设计

接下来制作面板 2 的方格内容,以其中左边第一个方格"信用卡还款"内容为例,WXML(pages/index/index.wxml)代码片段修改如下:



视频讲解

```
1. < view class = 'container'>
```

```
2. ...
```

```
3. <view class = 'servicePanel'>
```

```
4.
        <view class = 'serviceTitle'>腾讯服务</view>
       < view class = 'serviceBlocks'>
5
         < view class = 'box2'>
6.
            < image src = '/images/service/cardRepay.png'></image>
7.
8.
            <text>信用卡还款</text>
          </view>
9
10.
       </view>
     </view>
11
```

```
12. </view>
```

WXSS(pages/index/index.wxss)代码片段如下:

```
1. /*3-2-1 面板 2: 九宫格区域方格样式*/
2. . box2 {
3.
    border: 1rpx solid silver;
                                   /*1rpx 宽的银色实线边框*/
                                   /*flex 布局模型 */
4.
    display: flex;
5.
    flex - direction: column;
                                   /*垂直布局*/
                                   /*水平方向居中*/
6.
    align - items: center;
7.
                                   /*垂直方向居中*/
    justify - content: center;
                                  /* 宽度约占屏幕的 1/3 * /
8.
    width: 33 %;
9.
    height: 230rpx;
                                   /*高度230rpx*/
10. }
11. /*3-2-2 面板 2: 方格内图标*/
12. .box2 image {
    width: 90rpx;
                                   /*宽度*/
13.
   height: 90rpx;
14
                                   /*高度*/
15. }
16. /*3-2-3 面板 2: 方格内文本*/
17. .box2 text {
     font-size: 30rpx; /*字号为 30rpx 大小*/
18.
19. }
```

当前效果如图 3-24 所示。

由图可见,此时可以显示面板 2 的"信用卡还款"方 格内容。用同样的方式追加其他方格即可实现完整效 果。当然也可以暂时不追加,使用 3.2.4 节介绍的方法 减少工作量。

3.2.4 逻辑实现



1) 使用动态数据展示方格

由于所有方格的内容布局都是统一 ^{视频讲解}的,可以考虑使用 wx:for 属性配合动态 数组渲染全部列表项,以减少 WXML 页面的代码量。

修改 WXML(pages/index.index.wxml)页面代码 如下:

1. < view class = 'container'>

2. < view class = 'topPanel'>

```
3. < view class = 'box1' wx:for = '{{array1}}' wx:key = 'array1_{{index}}'>
```

```
4. <image src = '{{item.icon}}'></image>
```

••••• WeChat 奈	15:32	69% 🗔
	钱包	••• ••
. <u>.</u>		
收付款		
腾讯服务		
信用卞还款		

图 3-24 面板 2 预览效果



```
5. <text>{{item.text}}</text>
```

```
6. </view>
```

7. </view>

8. ... 9. </view>

上述代码表示使用 wx:for 属性循环显示全部方格,其中{{array1}}数组用于表示 3 个方格,方格区域中的图标和文本分别命名为 icon、text。这里均为自定义名称,开发者可以自行更改。

2) 补充数组完整信息

在 index. js 的 data 属性中添加 array1 数组,JS 文件(pages/index/index. js)代码如下:

```
1. Page({
2.
      data: {
3.
        //面板1的九宫格数组
        array1: [
4.
          { icon: '/images/top/money.png', text: '收付款' },
5.
6.
          { icon: '/images/top/balance.png', text: '零钱\n0.00' },
          { icon: '/images/top/cards.png', text: '银行卡' },
7
8
        1
9.
      }
10. })
```

当前效果如图 3-25 所示。

2 面板 2 的逻辑实现

1) 使用动态数据展示方格

接下来继续使用 wx: for 属性配合动态数组渲染全部列表项,以减少 WXML页面的代码量。

修改 WXML(pages/index.wxml)页面代码如下:

```
1. < view class = 'container'>
2
3.
      < view class = 'servicePanel'>
4.
        <view class = 'serviceTitle'>腾讯服务</view>
5
        < view class = 'serviceBlocks'>
6.
          < view class = 'box2' wx:for = '{{array2}}' wx:key = 'array2_{{index}}'>
            < image src = '{{item.icon}}'></image>
7
8.
            <text>{{item.text}}</text>
9.
          </view>
10.
        </view>
      </view>
11
12. </view>
```

上述代码表示使用 wx:for 属性循环显示全部方格,其中{{array2}}数组用于表示 8 个方格,方格区域中的图标和文本分别命名为 icon、text。这里均为自定义名称,开发者可以自行更改。

2) 补充数组完整信息

在 index. js 的 data 属性中添加 array2 数组,JS 文件(pages/index/index. js)代码如下:



视频讲解



视频讲解



视频讲解

第3章 小程序框架 51

1.	Page({
2.	data: {
3.	//面板1的九宫格数组
4.	··· ,
5.	//面板 2 的九宫格数组
6.	array2: [
7.	{ icon: '/images/service/cardRepay.png', text: '信用卡还款' },
8.	{ icon: '/images/service/mobileTopup.png', text: '手机充值'},
9.	{ icon: '/images/service/wealth.png', text: '理财通' },
10.	{ icon: '/images/service/utilities.png', text: '生活缴费' },
11.	{ icon: '/images/service/qqCoins.png', text: 'Q币充值' },
12.	{ icon: '/images/service/publicService.png', text: '城市服务' },
13.	{ icon: '/images/service/charity.png', text: '腾讯公益' },
14.	{ icon: '/images/service/insurance.png', text: '保险服务' }
15.]
16.	}
17.	})

WeChat \$\overline\$

 15:41
 67%
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 <li

图 3-25 面板 1 预览效果

钱包 ••• • ¥ [~] \square 零钱 0.00 收付款 银行卡 腾讯服务 ¥ $(\mathbf{0})$ 信用卡还款 手机充值 理财通 (4) \bigcirc Å 生活缴费 Q币充值 城市服务 T 3 保险服务 腾讯公益

15:47

65% 🔲

••••• WeChat 奈

图 3-26 最终效果图

3.2.5 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. }
```

此时就已全部完成,保存后重新预览项目,最终效果图如图 3-26 所示。



JSON 文件(pages/index.json)的完整代码如下:

```
    {
    "navigationBarTitleText": "钱包",
    "navigationBarBackgroundColor": #686F79"
    }
```

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

```
1. < view class = 'container'>
2.
      < view class = 'topPanel'>
3.
        < view class = 'box1' wx:for = '{{array1}}' wx:key = 'array1 {{index}}'>
4.
          < image src = '{{ item. icon} } '></ image >
          <text>{{item.text}}</text>
5.
6.
        </view>
     </view>
7.
8.
     < view class = 'servicePanel'>
       <view class = 'serviceTitle'>腾讯服务</view>
9
10.
       < view class = 'serviceBlocks'>
         < view class = 'box2' wx:for = '{{array2}}' wx:key = 'array2_{{index}}'>
11.
12.
            < image src = '{{item. icon}}'></image>
13.
            <text>{{item.text}}</text>
14
          </view>
        </view>
15
    </view>
16
17. </view>
```

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

```
1. /*1 背景容器样式*/
2. . container {
    height: 100vh;
                                   /*高度为100视窗,写成100%无效*/
3.

 background - color: silver;

                                 /*背景颜色为银色*/
   display: flex;
5.
                                  / * flex 布局模型 * /
6.
    flex - direction: column;
                                   /*垂直布局*/
7. }
8.
9. /*2 面板 1: 顶端状态栏*/
10. .topPanel {
                                  / * 高度 * /
11. height: 300rpx;
                                  /*背景颜色为灰色*/
12. background - color: #686F79;
13. display: flex;
                                  /*flex 布局模型*/
                                   /*水平布局*/
14. flex-direction: row;
15. }
16. / * 2-1 面板 1: 方格样式 * /
17. .box1 {
18.
   display: flex;
                                   /* flex 布局模型 * /
   flex - direction: column;
19.
                                   /*垂直布局*/
                                  / * 水平方向居中 * /
20. align - items: center;
21. width: 33%;
                                  /* 宽度约占屏幕的 1/3 * /
                                   /*高度*/
22. height: 250rpx;
23. }
24. /*2-2 面板 1: 方格内图标样式*/
25. .box1 image{
                                   /*宽度*/
   width: 110rpx;
26.
27.
   height: 110rpx;
                                   /*高度*/
```

```
/*四周外边距均为 20rpx */
28.
     margin: 20rpx;
29. }
30. /*2-3 面板 1: 方格内文本样式*/
31. .box1 text{
                                   /*文本居中*/
32. text - align: center;
   color: white;
                                   /*字体颜色为白色*/
33.
                                    /*字号为 35rpx 大小*/
34.
   font-size: 35rpx;
35. }
36
37. / * 3 面板 2: "腾讯服务"栏 * /
38. .servicePanel {
                                  / * 最小高度 * /
39. min-height: 600rpx;
40. background - color: white;
                                   / * 背景颜色为白色 * /
41. margin: 20rpx 0;
                                    /*上下外边距 20rpx, 左右 0*/
42. }
43. / * 3-1 面板 2: 第一行标题样式 * /
44. .serviceTitle {
45. padding: 20rpx;
                                   /*四周内边距 20rpx */
46.
   border: 1rpx solid silver;
                                   /*1rpx 宽的银色实线边框*/
                                    /*字号为 30rpx 大小*/
47. font - size: 30rpx;
                                   /*字体颜色为灰色*/
48. color: gray;
49. }
50. / * 3-2 面板 2: 九宫格区域样式 * /
51. .serviceBlocks {
52. display: flex;
                                   /* flex 布局模型 * /
                                   /*水平布局*/
53. flex - direction: row;
54.
    flex - wrap: wrap;
                                    /* 允许换行 */
55. }
56. / * 3-2-1 面板 2: 九宫格区域方格样式 * /
57. . box2 {
58. border: 1rpx solid silver;
                                   /*1rpx 宽的银色实线边框*/
59. display: flex;
                                   /* flex 布局模型 * /
60. flex - direction: column;
                                  / * 垂直布局 * /
                                  /*水平方向居中*/
61. align - items: center;
                                   /*垂直方向居中*/
62. justify - content: center;
                                   /* 宽度约占屏幕的 1/3*/
63.
   width: 33 %;
64.
   height: 230rpx;
                                   /*高度230rpx*/
65. }
66. / * 3-2-2 面板 2: 方格内图标 * /
67. .box2 image {
                                    /*宽度*/
68.
   width: 90rpx;
                                    /*高度*/
69. height: 90rpx;
70. }
71. /*3-2-3 面板 2: 方格内文本*/
72. .box2 text {
73. font - size: 30rpx;
                                   /*字号的 30rpx 大小*/
74. }
```

JS文件(pages/index/index.js)的完整代码如下:

```
    Page({
    data: {
    //面板1的九宫格数组
    array1: [
    { icon: '/images/top/money.png', text: '收付款' },
    { icon: '/images/top/balance.png', text: '零钱\n0.00' },
```



7.	{ icon: '/images/top/cards.png', text: '银行卡' },
8.],
9.	//面板 2 的九宫格数组
10.	array2: [
11.	{ icon: '/images/service/cardRepay.png', text: '信用卡还款' },
12.	{ icon: '/images/service/mobileTopup.png', text: '手机充值' },
13.	{ icon: '/images/service/wealth.png', text: '理财通' },
14.	{ icon: '/images/service/utilities.png', text: '生活缴费'},
15.	{ icon: '/images/service/qqCoins.png', text: 'Q币充值' },
16.	{ icon: '/images/service/publicService.png', text: '城市服务' },
17.	{ icon: '/images/service/charity.png', text: '腾讯公益'},
18.	{ icon: '/images/service/insurance.png', text: '保险服务' }
19.]
20.	}
21	

21. })