

07 页面布局：使用组件让页面开发如此简单

更新时间：2019-07-11 15:00:50



“

我们活着不能与草木同腐，不能醉生梦死，枉度人生，要有所作为。

——方志敏

”

在上一章我们已经安装好了微信开发者工具，也学习了“分类拆解法”的小程序编程思维。

本章我们将通过实践的方式来进一步理解“分类拆解法”，同时掌握组件、WeUI 框架、云开发等小程序开发常用特性。

本专栏所有实践内容均提供完整的源代码，源代码获取地址见本节“3. 专栏源代码”。

本节我们将实现如图 1 所示的页面Demo。

图 1 页面布局与组件Demo



1. 拆解

开发页面之前，我们需要按照“分类拆解法”的拆解步骤对页面进行拆解。

1.1 页面拆解为子部件

拆解步骤的第一步是将页面拆解为多个子部件。

拆解出的子部件有4个，在图 1 中已用红色数字标出。子部件1是一个幻灯片；子部件2是一个可切换的菜单；子部件3是一个内容显示区域，根据选中的菜单显示不同内容；子部件4是显示页面底部内容的区域。

4个子部件之间是上下排列的关系。

1.2 子部件拆解为显示元素

每个子部件需要继续拆解为多个显示元素。

子部件1是一个幻灯片，我们从图中看到有3个点，这是一个显示元素。同时，3个点表示幻灯片有3张，即3个显示元素。因此，子部件1一共有4个显示元素：

- 显示幻灯片张数的点
- 幻灯片1
- 幻灯片2
- 幻灯片3

子部件2是一组菜单，一共有3个，即3个显示元素：

- 板块1 菜单
- 板块2 菜单

- 板块3 菜单

子部件3是显示子部件2中选中菜单的内容，隐藏未选中菜单的内容。因此也有3个显示元素：

- 板块1 内容
- 板块2 内容
- 板块3 内容

子部件4是显示页面底部内容的区域，在图1中底部内容应该是固定的，因此只有一个显示元素：

- 底部内容

1.3 显示元素拆解为分类

每个显示元素需要继续拆解为多个分类（界面、数据、事件）。

子部件1的显示元素：

显示幻灯片张数的点

根据我们使用各类 APP 和小程序的经验，显示幻灯片张数的点会根据当前播放的幻灯片变化，因此它是一个交互界面。

同时，还需要有一个数据 `activeSlide` 来决定点如何显示，`activeSlide` 代表当前播放的幻灯片是哪一张。

幻灯片1、幻灯片2、幻灯片3

三个幻灯片显示元素只会同时显示一张在页面中，有两张处在隐藏状态，因此它们都是交互界面。

同时，它们也需要根据数据 `activeSlide` 来决定自己是显示还是隐藏。

此外，根据我们的经验，当我们左右滑动屏幕，将切换显示不同的幻灯片。这即是说这里有一个滑动事件，该事件会根据人滑动屏幕改变数据 `activeSlide`，来实现幻灯片切换。

子部件2的显示元素：

板块1菜单、板块2菜单、板块3菜单

三个菜单会有选中、未选中的状态切换，因此它们都是交互界面。

同时，需要一个数据 `activeIndex` 来决定哪个菜单处于选中状态。

此外，每个菜单还应该有一个事件函数来响应用户点击菜单的动作。

子部件3的显示元素：

板块1内容、板块2内容、板块3内容

三个板块内容会根据当前选中的菜单是哪一个来决定自己是显示还是隐藏，因此它们也都是交互界面。

同时，数据 `activeIndex` 将控制它们的显示与隐藏。

子部件4的显示元素：

底部内容

页面的底部内容一般是固定的，应该是一个静态界面。

2. 编程

完成拆解后，我们就可以按照“分类拆解法”的编程步骤开始一步步实现图1的Demo页面了。

2.1 UI框架

在开始编程前，我们需要先选定一个 UI 框架来帮助我们极大地减少 CSS 样式编码量。

在网上可以搜索到很多小程序 UI 框架，它们各有优劣。比较完善的 UI 框架，都有配套的开发文档网站和小程序 Demo。在实际小程序应用开发中，建议先实际体验各个 UI 框架的小程序Demo并阅读它们的开发文档，最终选择一个最合适当前小程序应用开发的 UI 框架。

本专栏使用的是微信官方推出的 UI 框架，WeUI for 小程序，这是一个基于MIT开源协议的开源框架，可以用于商业用途。

WeUI for 小程序的项目源代码可以访问 **GitHub** 获取，项目网址是：<https://github.com/Tencent/weui-wxss/>。该网址中有 WeUI 的介绍，使用方法说明，以及小程序 Demo 二维码。

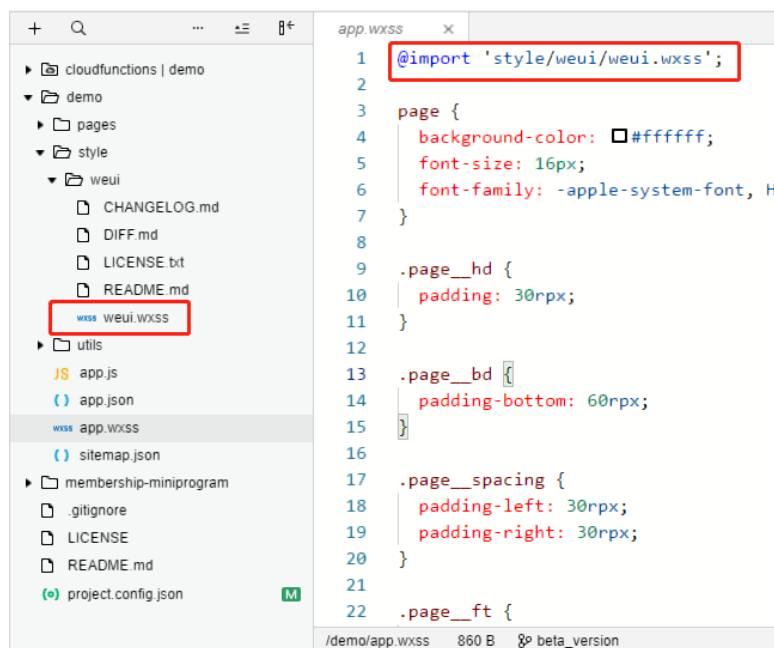
WeUI 的小程序 Demo 请各位同学务必扫码打开查看，本节后续内容需要用到。

WeUI 的使用非常简单，我们只需要将源代码中的 `dist/style/weui.wxss` 文件复制到我们自己的小程序项目中，再在 `app.wxss` 中使用以下语句引用它即可：

```
@import 'style/weui/weui.wxss';
```

具体可参考图 2。

图 2 在项目中使用 WeUI



2.2 定义页面子部件

编程实现页面的第一步是定义好页面的各个子部件。

前面章节已经介绍过，子部件的定义主要使用 **View** 标签在 **WXML** 页面模板中定义，**view** 即 **HTML** 中的 **div**。

为了实现图 1 的页面显示效果，在引用 **WeUI** 后还需要编写一些 **CSS** 样式。由于篇幅所限，**CSS** 样式的完整代码请到本专栏源代码的 [demo/app.wxss](#) 中查看。

本节主要讲解在 **WXML** 页面模板中如何定义页面子部件。

整个页面一般会定义一个页面级的 **view**，各个子部件均包含在其中。

```
<view class="page">
  <!-- 各个子部件的具体内容 -->
</view>
```

此外，一个页面一般还分为3个部分，分别是页面头部、页面内容、页面底部。在本节的**Demo**中有 4 个子部件，它们是上下排列关系，子部件 1 幻灯片位于页面头部，子部件 4 底部内容位于页面底部，子部件 2 和子部件 3 是页面主要内容区域。

```
<view class="page">
  <!-- 页面头部 -->
  <view class="page__hd head">
    <!-- 子部件1 -->
  </view>
  <!-- 页面内容 -->
  <view class="page__bd page__spacing">
    <!-- 子部件2 -->
    <!-- 子部件3 -->
  </view>
  <!-- 页面底部 -->
  <view class="page__ft page__spacing footer">
    <!-- 子部件4 -->
  </view>
</view>
```

有了以上的页面整体结构，我们就可以来具体定义每一个子部件在页面中的排列顺序了。

每个子部件使用一个 **View** 来标记位置。

```
<view class="page">
  <!-- 页面头部 -->
  <view class="page__hd head">
    <!-- 子部件1 -->
    <view class="placeholder">幻灯片</view>
  </view>
  <!-- 页面内容 -->
  <view class="page__bd page__spacing">
    <!-- 子部件2 -->
    <view class="placeholder">可切换的菜单</view>
    <!-- 子部件3 -->
    <view class="placeholder">内容显示区域，根据选中的菜单显示不同内容</view>
  </view>
  <!-- 页面底部 -->
  <view class="page__ft page__spacing footer">
    <!-- 子部件4 -->
    <view class="placeholder">显示页面底部内容的区域</view>
  </view>
</view>
```

在微信开发者工具中我们看到的效果图 3 所示。

图 3 子部件排列

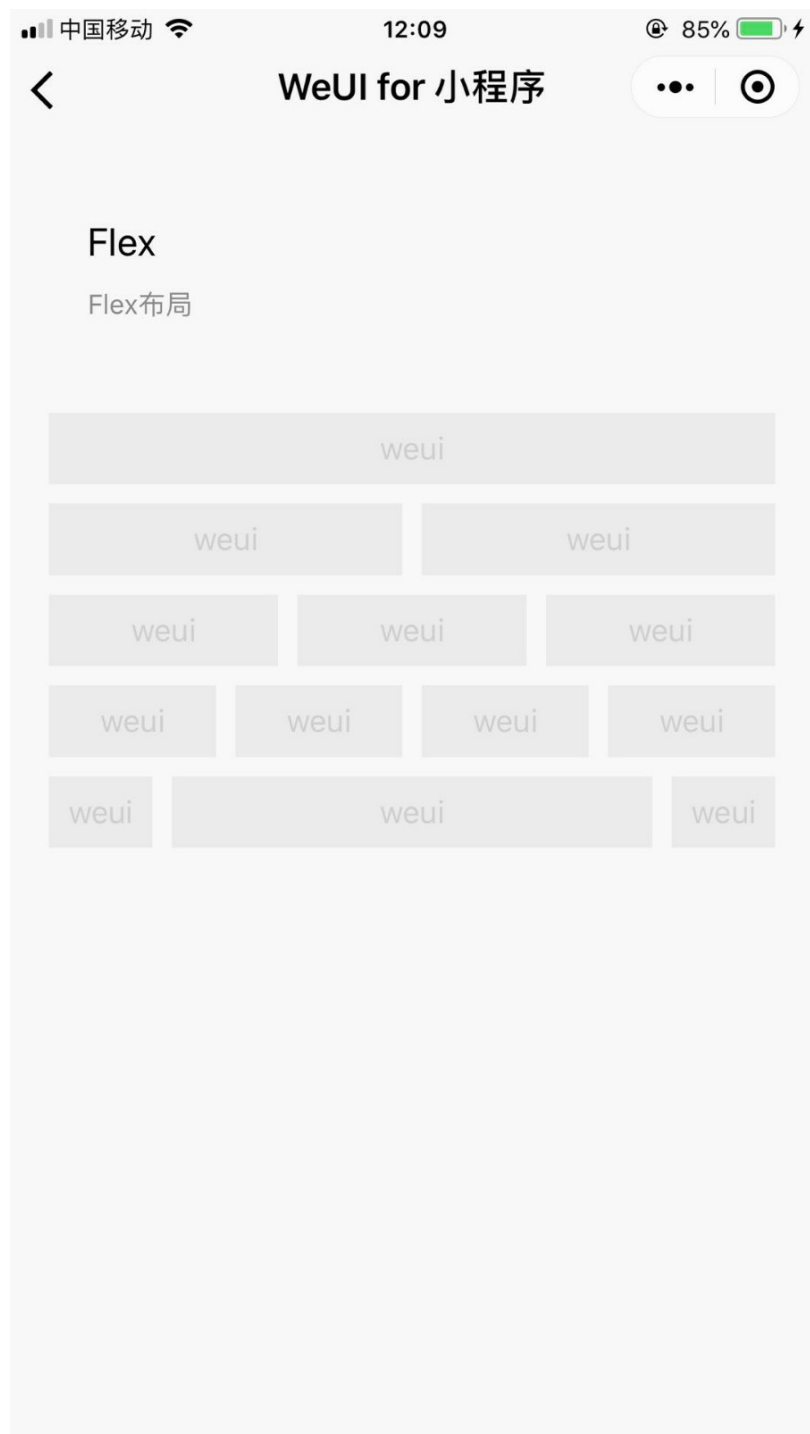


2.3 使用 WeUI 快速实现页面样式

在图 3 中我们发现子部件 2 可切换的菜单、子部件 3 内容显示区域的显示效果并未达到类似图 1 的排列方式，因此我们还需要进一步优化。

通过查阅 WeUI 的小程序 Demo，我们可以发现基础组件的 `flex` 可以实现三个菜单的横向排列效果，如图 4 所示。

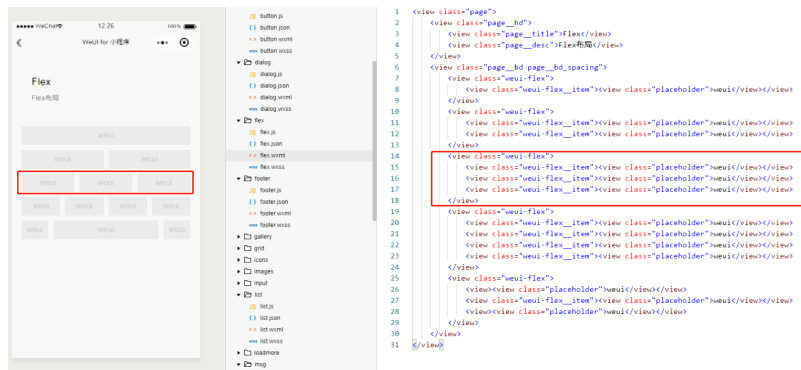
图 4 WeUI 的 `flex` 组件



因此，我们需要下载 WeUI 的 Demo 源代码，并用微信开发者工具打开（Demo 源代码在 **dist** 目录，请注意，不是整个项目）。然后找到 flex 这个页面，阅读 **flex.wxml**，了解 flex 组件的语法，并将 flex 组件用到自己的代码中。

flex 组件实现与子部件 2 类似的排列方式，及对应代码段，在图 5 中用红框标出，以便大家理解。

图 5 flex实现代码



最终，通过使用flex组件，我们定义好的页面子部件代码如下：

```
<view class="page">
  <!-- 页面头部 -->
  <view class="page__hd head">
    <!-- 子部件1 -->
    <view class="placeholder">头部图片</view>
  </view>
  <!-- 页面内容 -->
  <view class="page__bd page__spacing">
    <!-- 子部件2 -->
    <view class="weui-flex catalog">
      <view class="weui-flex__item">
        <view class="placeholder">版块1</view>
      </view>
      <view class="weui-flex__item">
        <view class="placeholder">版块2</view>
      </view>
      <view class="weui-flex__item">
        <view class="placeholder">版块3</view>
      </view>
    </view>
    <!-- 子部件3 -->
    <view class="content">
      <view class="placeholder">内容</view>
    </view>
  </view>
  <!-- 页面底部 -->
  <view class="page__ft page__spacing footer">
    <!-- 子部件4 -->
    <view class="placeholder">底部</view>
  </view>
</view>
```

在微信开发者工具中我们看到的效果图如图 6 所示。

图 6 定义页面子部件的最终效果



本小节主要讲解了如何使用 WeUI 帮助我们快速实现页面样式的方法，最后做一下总结：

- 打开 **WeUI** 的小程序 **Demo**，查看各个组件的演示页面，找到自己需要的页面样式；
- 在微信开发者工具中打开 **WeUI** 的源代码（**Demo** 源代码在 **dist** 目录，请注意，不是整个项目），然后找到这个演示页面，结合微信开发者工具的手机模拟界面阅读源代码，理解组件的用法；
- 将自己需要的代码段复制到自己的代码中，通过简单修改实现自己所需的页面样式。

2.4 使用小程序组件开发页面

在完成页面子部件的排列顺序后，下一步编程工作是逐一实现每个子部件的显示元素。

第一个子部件是幻灯片，在“1.2 子部件拆解为显示元素”和“1.3 显示元素拆解为分类”中，我们已经进行了详细的分析。

估计看完分析，一部分同学已经明白了编程思路但不知道该如何下手实现，还有一部分同学晕乎乎的。不要着急，幻灯片的实现其实非常简单，且听我道来。

20多年前的应用编程，是汇编语言、C 语言、C++ 语言，它们只有少部分人才能玩得转。

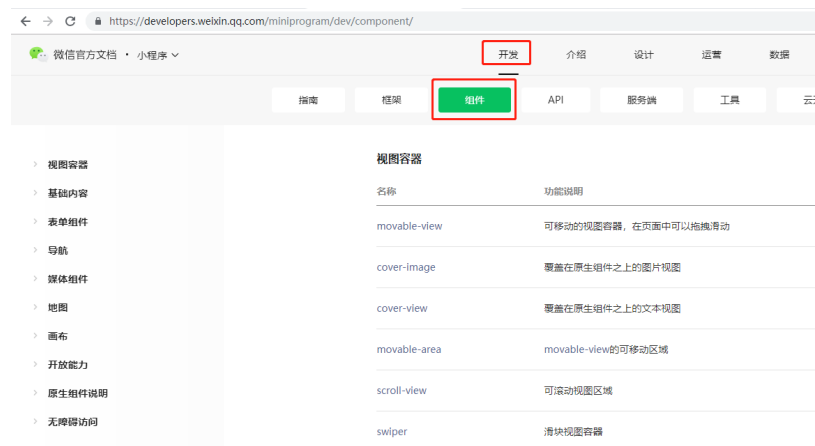
现在的应用编程已经进化得越来越简单，我们已经有了非常多的强大工具。

作为最伟大的物种——人类的一员，使用工具是我们无比聪明的大脑才具备的能力，不善加利用对不起我们投了一个好胎呀。

在“2.3 使用 WeUI 快速实现页面样式”中我们已经介绍了一个工具，此外，小程序本身也提供了很多工具，即组件。它们在微信官方的“小程序开发文档”中有详细的介绍，文档位置如下：

- 入口网址：[小程序组件](#)，如果微信官方文档改版导致链接失效，请按图索骥。
- 入口位置：在“小程序开发文档”首页，如图 7 红框标出部分。

图 7 小程序组件文档位置



此外，小程序组件也有 Demo 及源代码，你可以在“小程序开发文档”的“体验小程序”一节中找到，如图 8 所示。

图 8 小程序组件 Demo 与源代码文档位置



在小程序组件中，已经提供了幻灯片组件 `swiper`，我们只需要复制小程序组件 Demo 源代码中的 `swiper` 组件代码段，复制到我们自己的代码中稍加修改，即可实现子部件1幻灯片，这里给出修改后的实现代码供参考。

```
<view class="page__hd head">
  <!-- 使用小程序组件 swiper 实现幻灯片 -->
  <swiper class="head" indicator-dots="true">
    <swiper-item>
      <view class="placeholder head-image">幻灯片1</view>
    </swiper-item>
    <swiper-item>
      <view class="placeholder head-image">幻灯片2</view>
    </swiper-item>
    <swiper-item>
      <view class="placeholder head-image">幻灯片3</view>
    </swiper-item>
  </swiper>
</view>
```

具体过程请各位同学按照以下步骤实践操作：

- 打开小程序组件 **Demo**，查看各个组件的演示页面，找到自己需要的组件；
- 在微信开发者工具中打开小程序示例源代码，然后找到这个演示页面，结合微信开发者工具的手机模拟界面阅读源代码，理解组件的用法；
- 将自己需要的代码段复制到自己的代码中，通过简单修改实现自己所需的页面功能。

2.5 实现可切换的菜单

第二个子部件是可切换菜单，在 WeUI 的 Demo 小程序中我们同样能找到组件 `navbar`。简单修改该组件演示页面的源代码，即可实现子部件 2 可切换菜单，子部件 3 内容显示区域。

具体实现代码，请参考本专栏源代码 `demo/pages/02_navbar`。

2.6 一点小建议

在这一小节，我们主要介绍了一种“站在巨人肩膀上”的编程方式，使用小程序官方、第三方已经制作好的各类组件，帮助我们更简单高效地编写出我们的小程序应用。

对于刚接触编程，经验还不是很丰富的同学，我推荐你们使用这种看上去“偷懒”的方式来编写程序。它可以让你快速积累编程经验、增长在编程领域的见识，也会让你更容易编写出一个完整的小程序，增加自己编程的信心和成就感、降低挫折感，更快入门编程。

不过，在编写过一定的代码，有一定编程经验后，这个方法就不能帮你继续提高了。这时候，你如果想进一步提高编程水平，需要开始研究这些组件是如何编写出来的，并尝试编写出一些组件给别人使用。

造轮子的人和用轮子的人谁更值钱，你懂的。

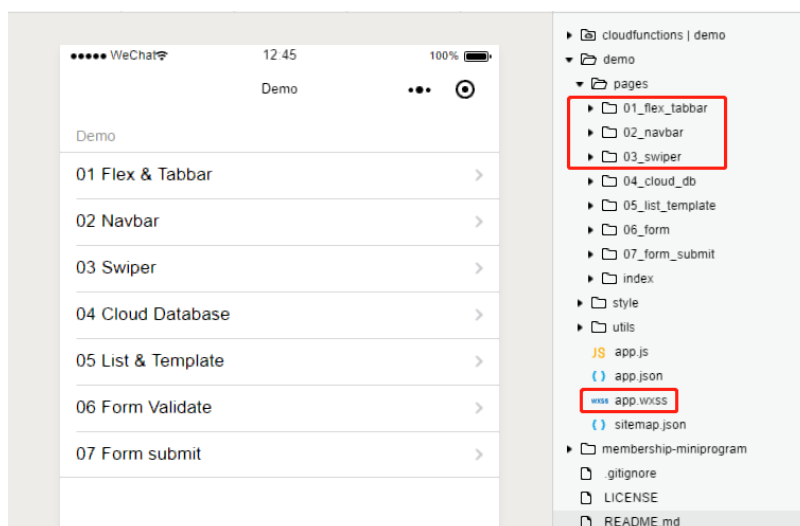
3. 专栏源代码

本专栏源代码已上传到 [GitHub](#)，请访问以下地址获取：

<https://github.com/liujiec/Membership-ECommerce-Miniprogram>

本节源代码内容在图 9 红框标出的位置。

图 9 本节源代码位置



下节预告

下一节，我们将讲解如何使用云开发实现服务端的数据存储，以及小程序手机端如何实现与服务端的数据交互。

实践环节

实践是通往大神之路的唯一捷径。

本节实操内容：

- 编写代码完成图 1 所示的页面，如碰到问题，请阅读本专栏源代码学习如何实现。



06 解剖分类拆解去详解

08 数据交互：云开发让数据库操作如此简单

