

## 05 源代码：了解构造才能下刀如有神

更新时间：2019-07-08 10:06:40



才能一旦让懒惰支配，它就一无可为。

更多一手资源请+V：Andyqc1  
qq：3118617541

——克雷洛夫

上一节我们已经安装好了微信开发者工具，这一节我们将使用微信开发者工具，搞清楚小程序的源代码结构。

编程语言也是一种语言，跟我们人类之间交流的中文、英文等语言类似，也遵循一定的规范。

在中文或英文中，我们都知道一篇文章是由很多句子构成的，而每一个句子又是由主语、谓语、宾语构成的。

与此类似，一个小程序也是由多个页面构成的，而每一个页面则是由我们在第一章第二节“分类拆解法”中介绍过的“分类”构成，“分类”包括界面、数据和事件（如图 9 所示）。

图 9 一个页面的分类构成



因此，我们学习小程序编程语言，也可以像学习中文、英文一样，先理解语言的语法结构（即“分类”），把握整体往往可以用最快的速度入门。

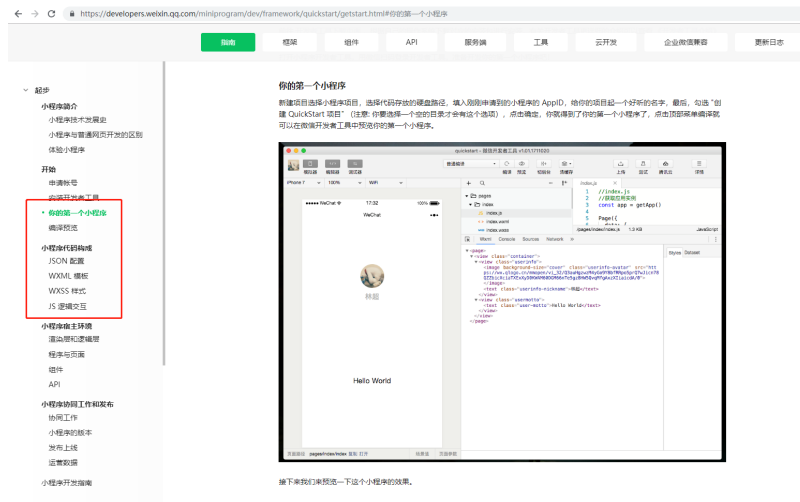
### 1. 小程序代码构成

小程序代码构成在微信官方的“小程序开发文档”中有详细的讲解，讲解内容可以通过以下两种方式查看：

入口网址: [小程序代码构成](#), 如果微信官方文档改版导致链接失效, 请按图索引。

入口位置：在“小程序开发文档”首页，如图 10 红框标出部分。

图 10 小程序代码构成微信官方讲解内容



继续阅读本节后续内容前，请先阅读微信官方的讲解内容。从“你的第一个小程序”开始，在你已经安装好的微信开发者工具中一步步实际操作。

如果你发现完全读不懂这些内容，也完全看不懂里面的代码，说明你缺少一些基础知识，请暂停阅读本专栏，先刷光以下 LV1 至 LV4 的地[ ke ]图[ cheng ]:

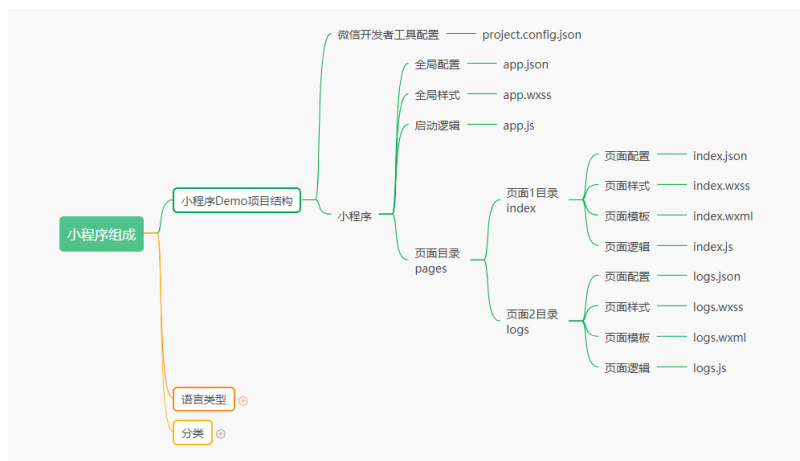
- LV1 地图-初识HTML+CSS
- LV2 地图-JavaScript入门篇
- LV3 地图-JavaScript进阶篇
- LV4 地图-JSON应用场景与实战

肃清其中的小怪，然后再来本地[ zhuan ]图[ lan ]继续升级打 BOSS。

## 2. 理解小程序源代码结构

按照“小程序开发文档”中“你的第一个小程序”所讲的操作方法，我们已经创建了我们的第一个小程序项目，这个小程序项目中已经包含了一个基本的 Demo。这个 Demo 的代码结构如图 11 所示。

图 11 小程序 Demo 项目结构



阅读完“小程序开发文档”中“小程序代码构成”的整个章节后，再对照小程序 Demo 项目结构，我们可以得出一个结论，小程序只有 4 种基本的代码类型，如图 12 所示。

图 12 小程序代码类型



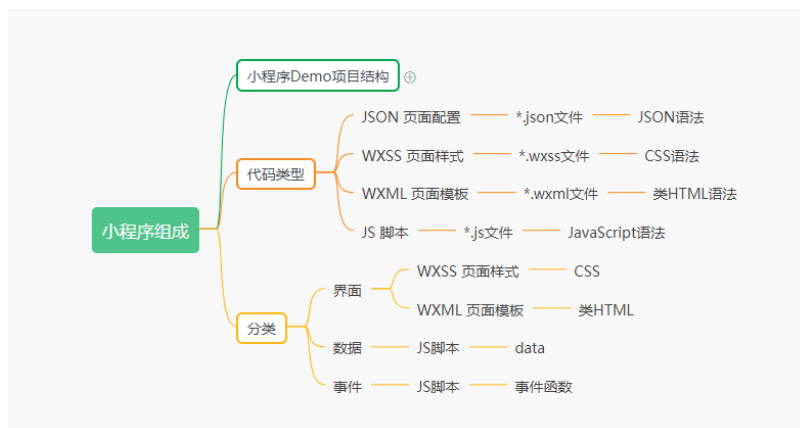
## 2.1 运用“分类拆解法”理清小程序源代码结构

如果在阅读中发现理解后续内容有困难，请重新阅读本专栏第一章第二节关于“分类拆解法”的内容。

相信部分同学在阅读完官方文档后，有一种似懂非懂的感觉。接下来，我们就使用“分类拆解法”的思路来进一步理清小程序的代码结构。

一个小程序是由多个页面组成，每个页面又由界面、数据和事件组成。界面是在 WXSS 页面样式和 WXML 页面模板中定义的，数据是在 JS 脚本的 data 中定义的，而事件则是在 JS 脚本的事件函数中定义的，如图 13 所示。

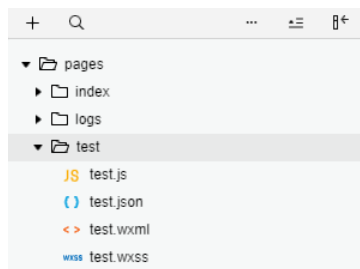
图 13 小程序的分类组成



我们具体用一个例子来理解“分类”与“代码类型”的关系。

首先我们在已经创建的小程序项目中添加一个页面 **test**，如图 14 所示。

图 14 新建页面**test**



然后我们在 **test.wxml** 中添加一个静态界面元素：

```
<view>我是静态界面元素</view>
```

添加一个交互界面元素，交互界面元素与数据的绑定在小程序中的语法为“**{{数据名称}}**”，因此我们需要添加以下代码：

```
<view>{{content}}</view>
```

再添加一个接收用户点击事件的按钮，界面元素与事件函数绑定在小程序中的语法为“**bindtap=‘事件函数名称’**”，因此我们需要添加以下代码：

```
<button bindtap='clickEvent' >点我触发事件</button>
```

接下来我们在 **test.js** 中定义数据：

```
/**
 * 页面的初始数据
 */
data: {
  content: "我是交互界面元素"
},
```

最后我们在 **test.js** 中定义用户点击按钮的事件函数：

```
/**
 * 点我触发事件button的事件函数
 */
clickEvent: function (options) {
  this.setData({
    content: "当用户点击按钮后，事件改变数据，数据改变交互界面元素的显示内容"
  })
},
```

在小程序的事件函数中改变数据的值的语法为：

```
this.setData({
  数据名称: 新的数据值
})
```

另外，为了能看到 **test** 页面，我们还需要在 **app.json** 中配置小程序的默认显示页面是 **test**：

```
"pages": [
  "pages/test/test",
  "pages/index/index",
  "pages/logs/logs"
],
```

完整代码如图 15 所示，test 页面运行效果如图16所示。

图 15 test 页面完整代码

test.wxml

test.js

app.json

```
1 <!--pages/test/test.wxml-->
2 <view>我是静态界面元素</view>
3 <view>{{content}}</view>
4 <button bindtap='clickEvent' >点我触发事件</button>
```

test.wxml

test.js

app.json

```
1 // pages/test/test.js
2 Page({
3
4   /**
5    * 页面的初始数据
6    */
7   data: {
8     content: "我是交互界面元素"
9   },
10
11   /**
12    * 点我触发事件button的事件函数
13    */
14   clickEvent: function (options) {
15     this.setData({
16       content: "当用户点击按钮后，事件改变数据，数据改变交互界面元素的显示内容"
17     })
18   },
19 })
```

test.wxml

test.js

app.json

```
1 {
2   "pages": [
3     "pages/test/test",
4     "pages/index/index",
5     "pages/logs/logs"
6   ],
7   "window": {
8     "backgroundTextStyle": "light",
9     "navigationBarBackgroundColor": "#fff",
10    "navigationBarTitleText": "WeChat",
11    "navigationBarTextStyle": "black"
12  },
13  "sitemapLocation": "sitemap.json"
14 }
```

更多一手资源请+V：Andyqc1aa : 3118617541

图 16 test页面运行效果



到这里各位同学是否已经明白小程序编程是怎么一回事了呢。

小程序编程其实很简单，我们只需要记住以下几个规则：

- 静态界面元素我们直接写在 **WXML** 页面模板中即可；
- 交互界面元素中会变动的部分我们用 "**{{数据名称}}**" 的形式在 **WXML** 页面模板中定义，并在 **JS** 脚本的 **data** 中写上数据名称的初始值；
- 界面中要响应用户动作的元素，我们用 "**bindtap=‘事件函数名称’**" 的形式在 **WXML** 页面模板中定义，并在 **JS** 脚本中编写事件函数要改变什么数据的值；
- 当我们编写的小程序代码在执行时，小程序的基础框架会自动将数据的值显示在页面上。
- 当用户点击我们定义了事件函数的界面元素，小程序会自动执行事件函数，事件函数的运行会改变数据的值，小程序的基础框架会自动将新的数据值显示在页面上。

是不是 **so easy**。对照这几个规则，回顾一下刚才我们编写的 **test** 页面代码，相信你已经知道小程序该怎么编程了，嘿哈。

2.2 JSON 页面配置与全局配置

JSON 页面配置以及全局配置 **app.json** 的语法相对简单固定，在实际开发中许多页面也并不需要进行配置，请参照图 17 所示，阅读“小程序开发文档”中“小程序配置”章节即可。

图 17 小程序配置文档位置



2.3 小程序启动逻辑 **app.js**

有关启动逻辑 **app.js** 的内容，请参照图 18 所示，阅读“小程序开发文档”中“注册小程序”章节。

图 18 小程序启动逻辑文档位置



2.4 全局样式 **app.wxss**

全局样式 `app.wxss` 的作用是减少 `CSS` 样式的代码量。如果我们要在多个页面都使用相同的 `CSS` 样式，只需要在 `app.wxss` 中定义一次，不需要复制粘贴到每个页面的 `page.wxss` 中。

## 下节预告

下一节，我们将讨论如何将“分类拆解法”整体运用在小程序开发中。

## 实践环节

实践是通往大神之路的唯一捷径。

本节实操内容：

- 阅读本节中列出的“小程序开发文档”相关内容，理解小程序项目创建后自带 **Demo** 的每一行代码的具体作用。

← 04 开发环境准备：屠龙勇者得有  
把好武器

06 解剖分类拆解法详解 →

更多一手资源请+V：AndyqcI  
aa：3118617541