

## 06 解剖分类拆解法详解

更新时间：2019-07-08 08:49:24



“

生活的理想，就是为了理想的生活。

——张闻天

”

上一节我们已经使用“分类拆解法”的思路理清了小程序的“分类”，以及“分类关系”。

我们用图 19 来做一些总结。

图 19 小程序的分类与分类关系



接下来我们看一看如何使用“分类拆解法”的“拆解步骤”，来理清一个小程序的完整开发思路。

## 1. 拆解步骤

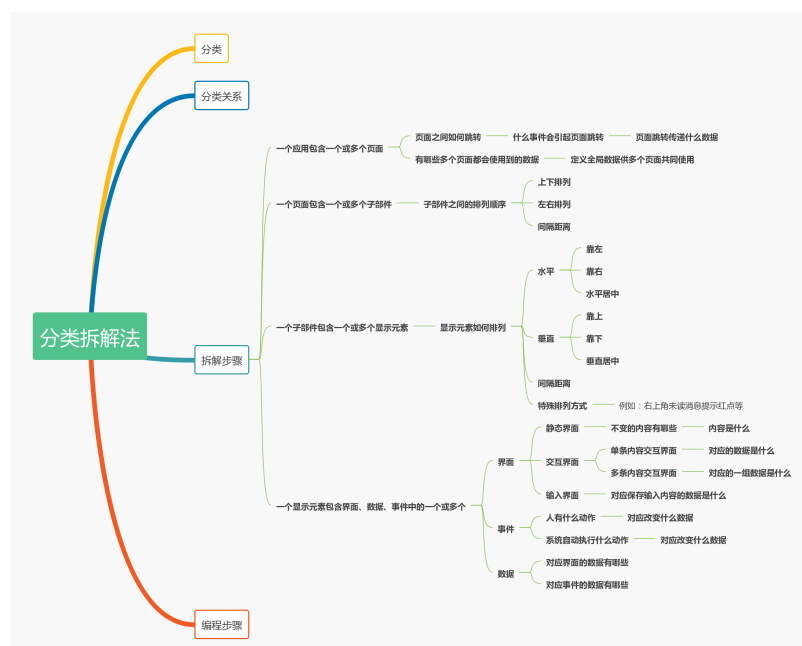
拆解的整体思路是自顶向下，从整体到局部，一步步将一个完整的小程序应用拆解为多个“分类”的组合。

具体来说就是：

- 将小程序应用拆解为多个页面
- 将每个页面拆解为多个子部件
- 将每个子部件拆解为多个显示元素
- 将每个显示元素拆解为多个“分类”的组合，即界面、数据与事件

这里先给出完整的拆解步骤，如图20所示。下面的内容会对拆解步骤的每一步逐一讲解。

图 20 小程序的拆解步骤



### 1.1 应用拆解为页面

一个小程序应用由多个页面组成，在小程序项目创建后自带的 Demo 中就包含两个页面 index 和 logs。

**\*\*页面与页面之间仅存在一种关系，页面跳转。\*\***页面跳转是由事件触发的，例如我们点击Demo中我们的头像，就会由 index 页面跳转到 logs 页面。

页面跳转还可能包含数据传递，例如当我们点击任何一个在线商城列表页面中的一件商品，页面会跳转到这件商品的详细信息页面，跳转的同时会将我们点击的商品ID传递给商品详细信息页面，这样程序才知道要在商品详细信息页面显示哪个商品的详细信息。

小程序实现页面跳转的语法，请参阅微信官方“小程序开发文档”的“[组件](#)”->“[导航](#)”->“[navigator](#)”小节。

除了页面跳转之间传递的数据，还有一些整个应用的各个页面都会使用到的数据，即全局数据。比如用户的登录状态，我们应该在用户打开小程序时就引导用户登录，后续用户访问小程序的各个页面，页面只需要查询用户的登录状态是否有效，如果用户未登录就跳转到用户登录页面。

小程序的全局数据在 app.js 中定义：

```
//app.js
App({
  onLaunch: function () {
  },
  //全局数据
  globalData: {
    全局数据名称1: 初始值1,
    全局数据名称2: 初始值2
  }
})
```

## 1.2 页面拆解为子部件

页面中子部件的概念，我们结合图 21 就可以很方便的理解。

图 21 小程序页面子部件示意图

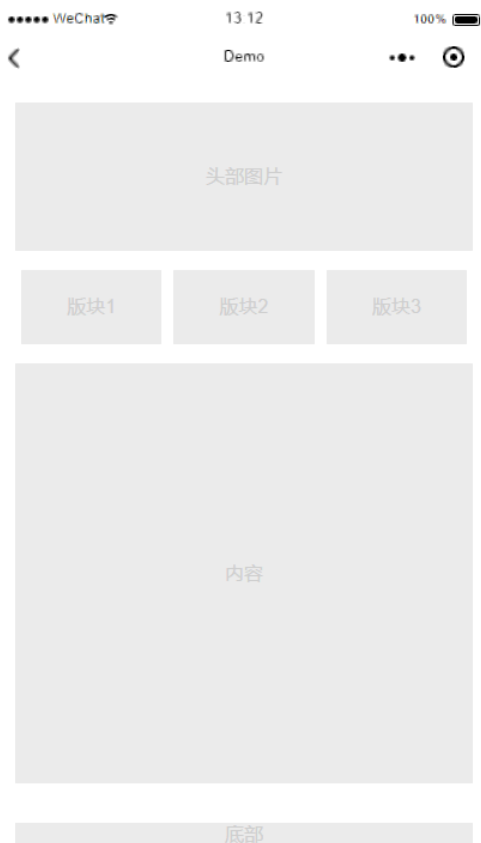


图 21 是很多小程序首页使用的子部件排列图，图中一共有6个子部件，子部件之间有两类关系：排列顺序和间隔距离。

头部图片和板块1是上下排列的顺序，板块1和板块2是左右排列的顺序。

从图中我们还可以看到子部件之间有一定的间隔距离。

在小程序中，每个子部件我们一般用 `<view>子部件具体内容</view>` 的语法来定义，**view** 即 HTML 中的 **div**。

**view**之间默认是上下排列的顺序，左右排列和间隔距离需要借助 UI 框架或在 WXSS 中的编写 CSS 样式来实现。

下一节我们将编写小程序代码，实现图 21 的页面。

## 1.3 子部件拆解为显示元素

子部件拆解为显示元素，在第一章第二节已经有详细的讲解，具体内容请回顾第一章第二节中“1. 界面分析”。

在这里以微信聊天页面（图 22）为例，来介绍一下显示元素之间的关系。

图 22 微信聊天页面



图片描述

显示元素之间同样具有两类关系：排列顺序和间隔距离。

在顶部子部件（图 22 中标记为 ① 的区域）中包括三个显示元素，它们之间是水平排列顺序。水平排列顺序分为三类：

靠左：返回按钮（一个箭头形状的图标）

居中：聊天对象的昵称

靠右：更多操作按钮（一个三个点组成的图标）

在聊天内容显示子部件（图 22 中标记为 ② 的区域）中包括多个显示元素，它们之间是垂直排列顺序。垂直排列顺序也分类三类：靠上、靠下和垂直居中。

此外，显示元素还有一些特殊的排列方式，比如我们在手机中常见的未读消息红点，通常显示在图标的右上角。

从图中我们还可以看到显示元素之间也有一定的间隔距离。

显示元素的排列顺序和间隔距离，同样是借助 UI 框架或在 WXSS 中的编写 CSS 样式来实现。

如何编写 CSS 样式实现排列顺序和间隔距离，请学习这门课程：[初识HTML+CSS](#)

#### 1.4 显示元素拆解为分类

我们已经知道分类包括界面、数据和事件，显示元素拆解为分类就是理清显示元素包含哪些界面、数据和事件。

在第一章第二节的“1. 界面分析”、“2. 事件分析”中，我们已经详细介绍了界面和事件的拆解方法，这里对数据拆解方法进行一下补充。

数据有三种作用：

- 显示数据：供界面显示内容
- 状态数据：作为状态标志决定界面显示与隐藏
- 输入数据：记录用户输入的内容，并实时显示在输入界面中

顶部子部件（图 22 中标记为 ① 的区域）有三个显示元素，返回按钮（一个箭头形状的图标）的作用是页面跳转，不包含数据；聊天对象的昵称会因为不同聊天对象而改变，它应该是一个数据 `nickname`；更多操作按钮（一个三个点组成的图标）的作用也页面跳转，同样不包含数据。

因此，顶部子部件包含一个数据：

```
data: {  
  nickname: "" //初始值是一个空字符串  
},
```

聊天内容显示子部件（图 22 中标记为 ② 的区域）包含 1 个显示元素，该元素是一个“多条内容交互界面”，显示我们与好友的聊天记录列表。因此该元素应该包括一个数据 `chatRecord`，并且该数据是一个数组，用来存储多条聊天记录。

此外，该显示元素还包括了丰富的事件，我们还需要对事件进行分析，识别出状态数据。

与界面显示隐藏有关的事件包括：

- 长按消息“事件”，显示复制、转发、收藏、删除等操作按钮；
- 点击空白区域“事件”，如果正在显示复制、转发、收藏、删除等操作按钮，则隐藏这些按钮。

这说明需要一个状态数据 `hideChatOperation`，来控制复制、转发、收藏、删除等操作按钮的显示与隐藏。

因此，聊天内容显示子部件包含两个数据：

```
data: {
  chatRecord: [], //初始值是一个空数组
  hideChatOperation: true //初始值默认隐藏复制、转发、收藏、删除等操作按钮
},
```

聊天内容输入子部件（图 22 中标记为 ③ 的区域），包含 4 个显示元素及多个事件。通过逐一分析，我们可以知道数据应该包括一个状态数据 `inputType`，该数据有多种状态，包括文字输入状态 0、语音输入状态 1、表情输入状态 2；一个状态数据 `hideMoreOperation`，该数据控制照片、拍摄、视频通话等操作按钮的显示与隐藏；一个输入数据 `inputContent`，记录用户输入的文字、语音、表情等信息。

```
data: {
  inputType: 0, //初始值是显示文字输入状态
  hideMoreOperation: true, //初始值默认隐藏照片、拍摄、视频通话等操作按钮
  inputContent: '' //初始值是一个空字符串
},
```

## 2. 编程步骤

在完成一个小程序应用需求的拆解后，我们已经对我们要编写的小程序有了非常有层次、也非常细致的理解，接下来我们就可以使用“逆向工程”的方式，将我们拆解出的所有元素，通过程序代码逐一实现，并最终像搭积木一样，构建出整个小程序代码。

“逆向工程”的方式就是“分类拆解法”的最后一步编程步骤：

在开始编程前，先选定一个 UI 框架，如 WeUI 等。UI 框架已经包含了大部分常用的 CSS 样式，可以帮助我们极大地减少 CSS 样式编码量。

首先，从拆解出来的所有数据中找到各个页面都要使用到的数据，在全局数据中定义。

然后，逐一实现每个页面，实现顺序如图 23 所示。

图 23 小程序的编程步骤



在第三章中，我们将使用 WeUI 作为 UI 框架，编写多个小程序页面，来实际演练“拆解步骤”和“编程步骤”。

## 下节预告

从下一节开始，我们将结合常用的小程序功能组件，通过实践的方式来进一步理解和掌握“分类拆解法”，并最终运用到自己的小程序开发中。

## 实践环节

实践是通往大神之路的唯一捷径。

本节实操内容：

- 使用“分类拆解法”拆解小程序项目创建后自带的**Demo**。

← 05 源代码：了解构造才能下刀如  
有神

07 页面布局：使用组件让页面开  
发如此简单 →