

# 官方文档浓缩系列《一》官方开发文档解读

## 语言与文件

微信小程序开发与其他平台开发的最大差异在于：

**微信使用的开发语言和文件很「特殊」。**

小程序所使用的程序文件类型大致分为以下几种：

- WXML ( WeiXin Mark Language , 微信标记语言 )
- WXSS ( WeiXin Style Sheet , 微信样式表 )
- JS ( Java , 小程序的主体 )

在语言方面，小程序看似重新定义了一套标准。但实际上，它们与「前端三件套」——HTML、CSS 和 Java——差不太多。

下面，我们就来对比一下小程序开发语言和「前端三件套」有什么异同点：

- **HTML 与 WXML**：两者差异比较大，如果之前没有接触过 Android 开发，可能会觉得有些头疼。事实上，WXML 更像是 Android 开发中的界面 XML 描述文件，适合于程序界面的构建；而 HTML 则倾向于文章的展示（这与 HTML 的历史有关），以及互联网页面的构建。
- **WXSS 与 CSS**：两者在语言上几乎没有差别，可以直接通用。

- **JS 文件**：小程序的 JS 文件与前端开发使用的 JS 几乎没有区别，只是小程序的 JS 新增了微信的一些 API 接口，并去除了一些不必要的功能（如 DOM）。

在语言上，小程序完全向学习成本最低的前端开发看齐，但这不代表所有前端开发者都能无缝迁移。

**如果你是从前端开发转向小程序，就需要注意这两个点：**

- HTML 与 WXML 两种文件的构建思想差异较大，如果之前只接触过前端开发，需要一点时间才能适应 WXML 的编写方法。
- 虽然小程序使用的是前端语言，但不代表可以继续沿用前端的开发思想进行开发。小程序对前端开发的要求从「构建界面」升级成「开发完整应用」，前端开发需要在意识上进行转变。

界面构建

## 1. 基本逻辑

WXML 和 WXSS 两种文件是小程序界面元素声明及样式描述文件。

WXML 最大的特点是以视图（view）的方式串联界面元素，并通过程序逻辑（AppService），将信息更新实时传递至视图层。

view 类似于 HTML 中的 div 元素。在构建的时候，view 可以被多级嵌套，view 内可以放置任意视觉元素。

**需要注意的是，元素一旦超出屏幕之外，用户就无法看到了，这与 HTML 有较大不同。**

比如，我们将手机屏幕想像成一个舞台，舞台之外的演员是无法被观众看到的。

**小程序有专门用于滚动的视图。**

如果希望界面是一个可以自由滚动的界面（例如列表等），可以使用 scroll-view 视图，在 WXSS 中将其大小调整为整个屏幕，并设置 scroll-y（上下滚动）或 scroll-x（左右滚动）为 true。

```
<scroll-view scroll-y="true">
  <view>Hello World</view>
</scroll-view>
```

小程序中不能直接使用 DOM 控制 WXML 元素。如果需要进行数据更新，就得使用 WXML 提供的数据绑定及元素渲染方法。

还有一点需要注意的是：小程序的栅格排版系统使用的是 Flex 布局，它是 W3C 在 2009 年提出的一种排版标准。

## **2. 绑定数据**

对于单个字段，开发者可以使用数据绑定的方法进行信息更新。

绑定的数据除了在加载的时候可以更新，也可以在 JS 主程序中以函数形式进行更新，更新同样可以反映到界面上被绑定的数据中。

### 3. 条件渲染与列表（循环）渲染

条件渲染适用于有意外情况提示的页面（如无法加载列表或详情时，做出提示等等）。

它的渲染带有触发条件，即符合条件时渲染这个页面，否则忽略或渲染另一段代码。

```
<view wx:if="{{ boyname == 'Tom' }}">Hello, Tom!</view>
<view wx:elif="{{ girlname == 'Anna' }}">Hello, Anna!</view>
<view wx:else>You neither Tom nor Anna.</view>
```

两个花括号所包含的判断条件中的变量于主程序 JS 代码中的 data 中声明。

```
Page({
  data: {
    boyname: 'Tom',
    girlname: 'Anna'
  }
})
```

若需要在界面中构建一个列表，可以使用 WXML 中的循环渲染，将同一元素渲染代码进行集合。

循环的数据可以通过数组的方式写入 data 中供 WXML 访问。

```
<view wx:for="{{array}}">
  {{index}}: {{item.message}}
</view>

Page({
  data: {
    array: [{
      message: 'foo',
    }, {
      message: 'bar'
    }]
  }
})
```

渲染完毕后，渲染判断条件的变动可以影响界面变动。

## 4. 模板与引用

WXML 支持使用模板与引用减少代码体积。

模板是在 WXML 代码中对相同的代码进行复用的方式。

可以将多个模板写入至同一文件，并使用 `import` 在其他文件中进行引用。

```
<template name="greetings">
  <view>
    <text>Hello, {{name}}</text>
  </view>
</template>
//some other codes
<template is="greetings" data="{{...name}}" />
```

如果需要整个页面引用，需要使用到 include。

```
<include src="header.wxml"/>
  <view> body </view>
<include src="footer.wxml"/>
```

## 5. 样式

通过 WXSS 样式表，开发者可以定义 WXML 中的元素样式。

WXSS 与 CSS 代码一样，可以直接使用选择器选择元素。在 WXML 中也可以直接定义元素的 id 和 class 以便于在 WXSS 文件中进行样式定义。

## 6. 用户操作与事件响应

由于微信使用的不是 HTML，所以也**不能通过添加超链接（a 元素）的方式来监测用户的点击事件。**

对于需要监听点击事件的元素，应该在 WXML 中使用 bindtap 属性或 catchtap 属性进行绑定。

```
<view bindtap="tapName"> Click me! </view>
```

除了点击一次，微信也提供按住、开始触摸、松手等事件响应。

在 WXML 中绑定好一个事件之后，就能在主程序 JS 中使用。

```
Page({
  tapName: function(somethingHere){
    // 执行这部分代码
  }
})
```

其他 API 中也有其他相应的事件，这些事件可以在微信小程序的官方文档中查阅到。

当需要在小程序的页面间进行跳转时，应该使用 wx.navigateTo() 方式。

```
tapName: function() {  
  wx.navigateTo({  
    url: '../logs/logs'  
  })  
}
```

**需要注意的是，有关于页面层级跳转，微信将层级跳转限制到了五层。**在开发时需要注意是否超过了相应限制。

## 网络访问

小程序支持三种请求方式。

一种是直接的 HTTP 连接请求，请求后直接返回结果，连接结束。另一种是 Socket 持续性连接，当一方主动关闭连接时，连接结束。

除了以上两种收发纯文本的连接方式，微信还提供了一个文件收发接口。**小程序中录制的语音以及选择的照片都需要这个方式来进行上传。**

通过小程序访问网络需要服务器端必须支持 HTTPS 安全连接，且端口号必须为 443。

同时，小程序只能访问开发者在登记小程序时所设定的服务器地址。

## 多媒体与存储



若需在小程序中播放多媒体（包括音视频）或进行数据存储，不能使用 HTML 5 中所提供的标准，**必须使用微信提供的小程序多媒体播放控制接口及存储接口等。**

关于声音的接口，有音频播放与音乐播放两种。

音频播放提供了播放、暂停和停止播放三种接口，不提供跳转至某个播放时间点的功能，也不能获取目前的播放进度。

音乐播放接口提供除以上的基础播放控制外的音乐状态检查和监听等功能。

**小程序提供照片和视频数据交换接口。**通过这个接口，小程序可以访问用户选定或拍摄的照片与视频。

**通过音频录制和视频照片接口获得的多媒体信息是临时的，需要通过小程序存储文件接口对文件进行永久保存。**

**对于文本数据，小程序也提供了存储这类数据的接口。**从诸如 Android 或其他 app 平台转向的开发者需要注意的是，小程序不提供数据库式的本地数据保存形式，而是通过「字段 - 值」的一对一形式进行保存。

硬件相关

小程序依托于微信，提供许多与硬件有关的 API。

小程序可以通过 API 获取到以下数据：

- 系统相关信息（包括网络状态、设备型号、窗口尺寸等）
- 重力感应数据
- 罗盘数据

通过以上 API，应该可以轻松写出「摇一摇」等互动性页面。

但需要注意：**这些数据只能主动获取，不能通过这些数值变化的回调实时获取。**

### 推送服务

小程序提供推送服务，可以随时向用户发送必要的通知。**但请注意，推送服务只能用于通知提醒，不能用于群发。**

**在小程序中，推送服务叫做「模板消息」**（之前有开发过服务号的开发者应该比较熟悉）。开发者需要在微信小程序后台登记新的模板推送消息（比如：购买成功通知等）并审核通过后，才能在小程序中使用模板消息推送服务。具体审核标准建议参考相应文档。

模板消息审核通过后，开发者需要先向微信服务器获取 Access Token，随后将该值、模板编号和模板中的动态变量（比如：订单号、价格等）提交给微信，由微信向用户推送通知。

### 用户信息与微信支付

**小程序可以在用户同意的前提下获取到用户的信息。**

首先，小程序要通过微信登录的接口，让用户授权登录。之后，小程序就可以展示并使用用户信息了。

在使用微信登录的时候需要注意，**消息需要经过签名确认其完整性之后，方能保证数据未经篡改。**

**小程序中也可以使用微信支付。**需要注意的是在发送支付请求时，需要在发送的消息中添加签名，以确认消息完整性。

以上便是知晓程序为大家整理的 **官方开发文档「重点信息」**。

不过，想要熟悉小程序，光看是不行的，还需要动手去做。

**如果开发过程中遇到问题，最好的办法还是查阅微信官方的小程序开发文档**，相信在文档中，你能找到绝大部分问题的解决办法。