

微信小程序

一、小程序

1.1 简介

<https://open.weixin.qq.com/>

小程序是一种全新的连接用户与服务的方式，它可以在微信内被便捷地获取和传播，同时具有出色的使用体验。

小程序的主要开发语言是 JavaScript

1.2 开发流程

1.2.1 申请帐号

<https://mp.weixin.qq.com/wxopen/waregister?action=step1>

小程序注册

① 帐号信息 — ② 邮箱激活 — ③ 信息登记

每个邮箱仅能申请一个小程序

已有微信小程序？立即登录

邮箱

作为登录帐号，请填写未被微信公众平台注册，未被微信开放平台注册，未被个人微信号绑定的邮箱


密码

字母、数字或者英文符号，最短8位，区分大小写

确认密码

请再次输入密码

验证码

 换一张

☐ 你已阅读并同意《微信公众平台服务协议》及《微信小程序平台服务条款》

注册

1.2.2 登录

<https://mp.weixin.qq.com/>

登录 [小程序后台](#)，我们可以在菜单“设置”-“开发设置”看到小程序的 AppID 了。

设置

[基本设置](#)[开发设置](#)[第三方授权管理](#)[开发者工具](#)

开发者ID

开发者ID		操作
AppID(小程序ID)	wxcff7381e631cf54e	
AppSecret(小程序密钥)		生成

服务器域名

服务器配置		说明	操作
request合法域名	https://servicewechat.com		
socket合法域名	wss://servicewechat.com	一个月内可申请5次修改 本月还可修改5次	修改
uploadFile合法域名	https://servicewechat.com		
downloadFile合法域名	https://servicewechat.com		

小程序的 AppID 相当于小程序平台的一个身份证，后续你会在很多地方要用到 AppID（注意这里要区别于服务号或订阅号 AppID）

1.2.3 安装开发工具

<https://developers.weixin.qq.com/miniprogram/dev/devtools/download.html>

打开小程序开发者工具，用微信扫码登录开发者工具，准备开发你的第一个小程序吧



微信开发者工具

v1.02.1902010



欢迎使用微信开发者工具



二、小程序初体验

2.1 新建模板游戏

2.1.1 创建项目



小程序项目

小程序

小游戏

代码片段

公众号网页项目

公众号网页



注销 >

新建项目 导入项目

项目名称 minigame-test-1

目录 C:\Users\xing\WeChatProjects\minigame-test-1

AppID

若无 AppID 可 [注册](#)

或使用 [测试号](#)

开发模式 小游戏

后端服务 ☒ 不使用云服务

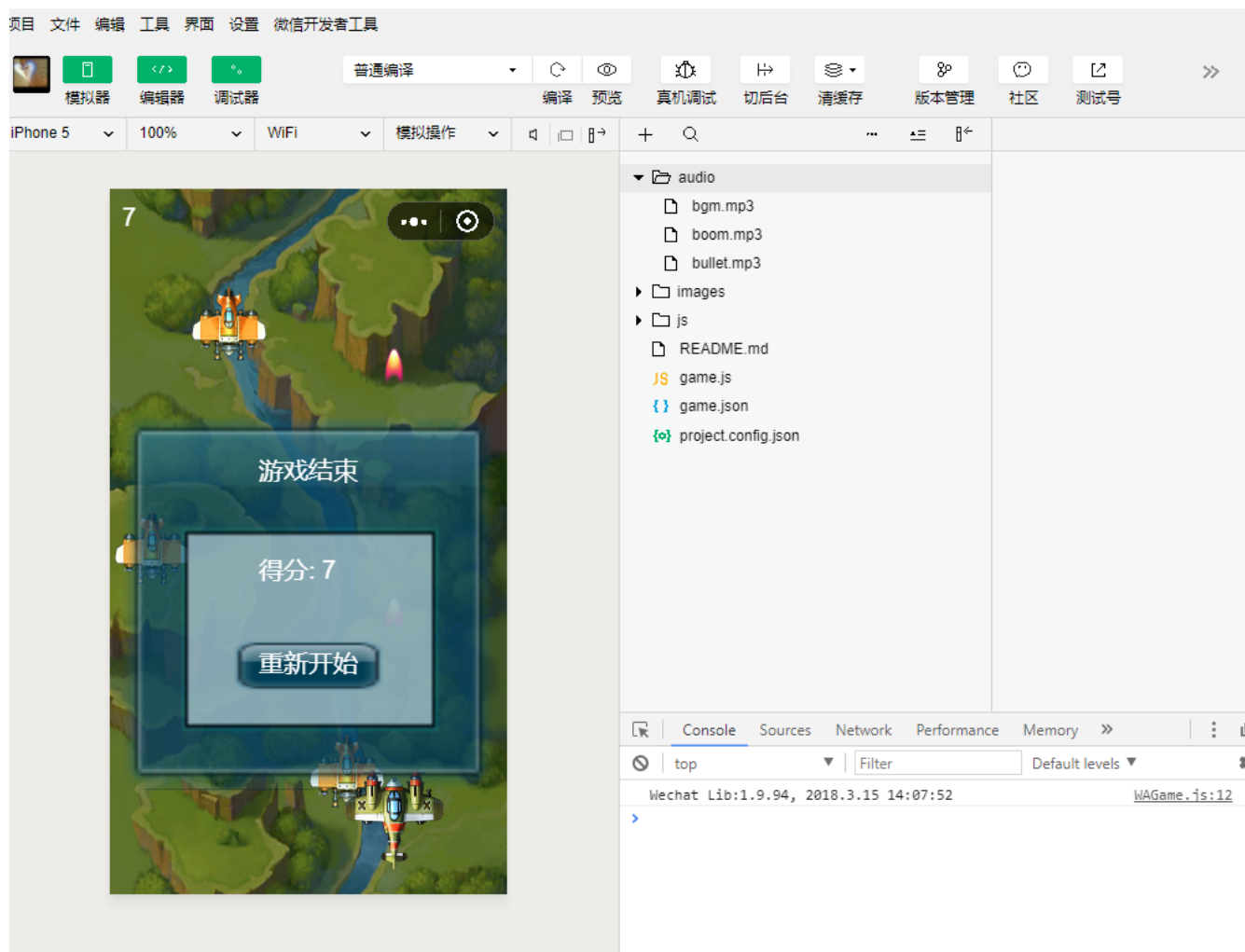
☐ 小程序-云开发

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。 [了解更多](#)

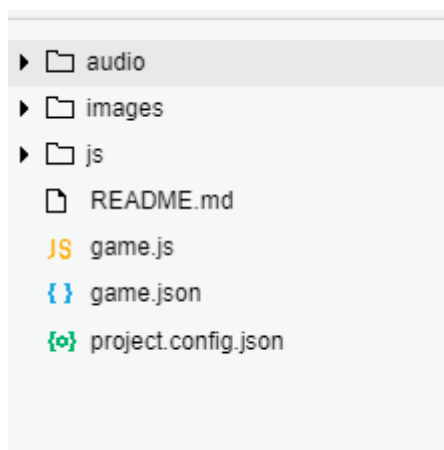
☐ 腾讯云

取消

新建



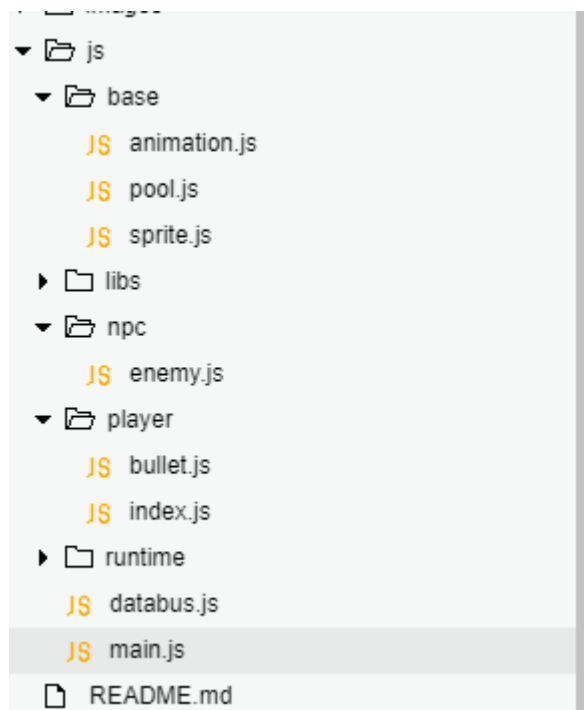
2.1.2 项目结构



audio:音频

images:图片

js:核心js文件



2.2 小程序应用

2.2.1 新建小程序

新建项目

导入项目

项目名称

app1902

目录

H:\Project\WeChat\app1902

▼

AppID

wx249ab7dbab1d29f0

▼

若无 AppID 可 [注册](#)
或使用 [测试号](#)

开发模式

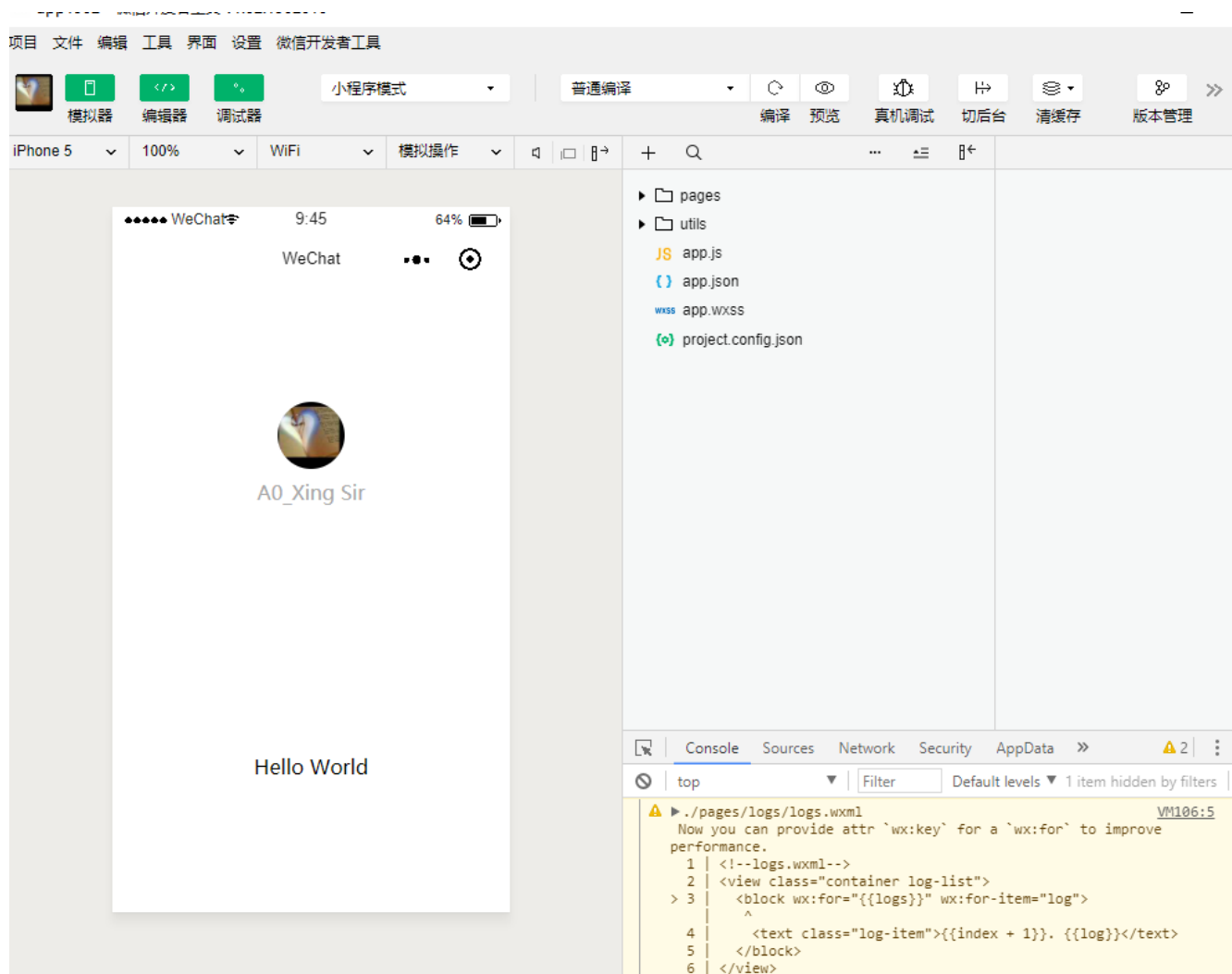
小程序

▼

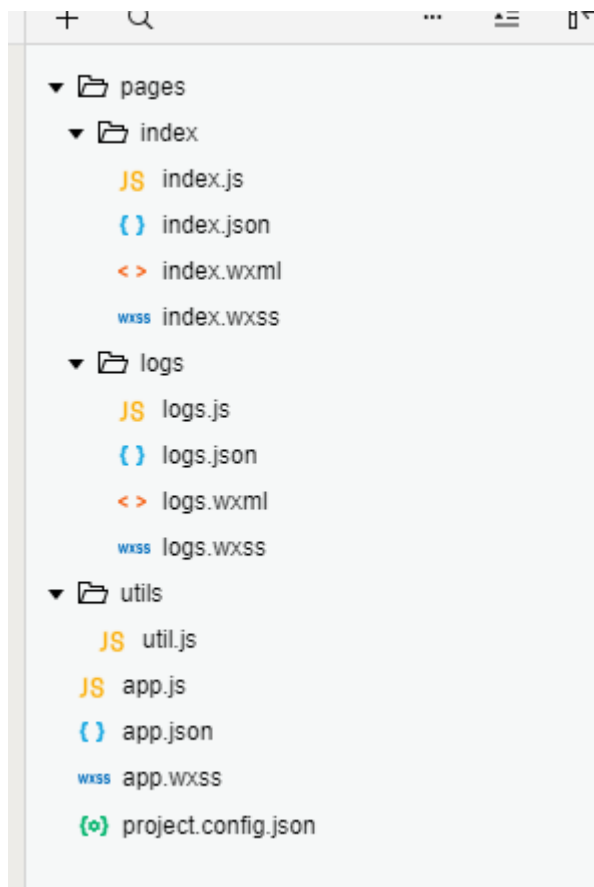
语言

JavaScript

▼



2.2.2 结构



pages:页面（页面、样式、js、默认数据配置 json）

utils:工具

2.2.3 各种文件格式

1.JSON 配置

JSON 是一种数据格式，并不是编程语言，在小程序中，JSON扮演的静态配置的角色。

我们可以看到在项目的根目录有一个 `app.json` 和 `project.config.json`，此外在 `pages/logs` 目录下还有一个 `logs.json`

小程序配置 `app.json`

`app.json` 是当前小程序的全局配置，包括了小程序的所有页面路径、界面表现、网络超时时间、底部 tab 等

1. `pages` 字段 —— 用于描述当前小程序所有页面路径，这是为了让微信客户端知道当前你的小程序页面定义在哪个目录。
2. `window` 字段 —— 定义小程序所有页面的顶部背景颜色，文字颜色定义等

工具配置 `project.config.json`

工具上做的任何配置都会写入到这个文件，当你重新安装工具或者换电脑工作时，你只要载入同一个项目的代码包，开发者工具就自动会帮你恢复到当时你开发项目时的个性化配置，其中会包括编辑器的颜色、代码上传时自动压缩等一系列选项

页面配置 `page.json`

这里的 `page.json` 其实用来表示 `pages/logs` 目录下的 `logs.json` 这类和小程序页面相关的配置。

如果你整个小程序的风格是蓝色调，那么你可以在 `app.json` 里边声明顶部颜色是蓝色即可。实际情况可能不是这样，可能你小程序里边的每个页面都有不一样的色调来区分不同功能模块，因此我们提供了 `page.json`，让开发者可以独立定义每个页面的一些属性，例如刚刚说的顶部颜色、是否允许下拉刷新等

2.WXML 模板

网页编程采用的是 HTML + CSS + JS 这样的组合，其中 `HTML` 是用来描述当前这个页面的结构，`CSS` 用来描述页面的样子，`JS` 通常是用来处理这个页面和用户的交互。

同样道理，在小程序中也有同样的角色，其中 `WXML` 充当的就是类似 `HTML` 的角色

绘制页面

`WXML`和 `HTML` 非常相似，`WXML` 由标签、属性等等构成。但是也有很多不一样的地方，我们来一一阐述一下：

1. 标签名字有点不一样

往往写 `HTML` 的时候，经常会用到的标签是 `div`, `p`, `span`，开发者在写一个页面的时候可以根据这些基础的标签组合出不一样的组件，例如日历、弹窗等等。换个思路，既然大家都需要这些组件，为什么我们不能把这些常用的组件包装起来，大大提高我们的开发效率。

从上边的例子可以看到，小程序的 `WXML` 用的标签是 `view`, `button`, `text` 等等，这些标签就是小程序给开发者包装好的基本能力，我们还提供了地图、视频、音频等等组件能力。

2. 多了一些 `wx:if` 这样的属性以及 `` 这样的表达式

在网页的一般开发流程中，我们通常会通过 `JS` 操作 `DOM` (对应 `HTML` 的描述产生的树)，以引起界面的一些变化响应用户的行为。例如，用户点击某个按钮的时候，`JS` 会记录一些状态到 `JS` 变量里边，同时通过 `DOM API` 操控 `DOM` 的属性或者行为，进而引起界面一些变化。当项目越来越大的时候，你的代码会充斥着非常多的界面交互逻辑和程序的各种状态变量，显然这不是一个很好的开发模式，因此就有了 `MVVM` 的开发模式 (例如 `React`, `Vue`)，提倡把渲染和逻辑分离。简单来说就是不要再让 `JS` 直接操控 `DOM`，`JS` 只需要管理状态即可，然后再通过一种模板语法来描述状态和界面结构的关系即可。

小程序的框架也是用到了这个思路，如果你需要把一个 `Hello world` 的字符串显示在界面上。

`WXML` 是这么写：

```
<text>{{msg}}</text>
```

`JS` 只需要管理状态即可：

```
this.setData({ msg: "Hello world" })
```

通过 `` 的语法把一个变量绑定到界面上，我们称为数据绑定。仅仅通过数据绑定还不够完整的描述状态和

3.WXSS 样式

`WXSS` 具有 `CSS` 大部分的特性，小程序在 `WXSS` 也做了一些扩充和修改。

1. 新增了尺寸单位。在写 `CSS` 样式时，开发者需要考虑到手机设备的屏幕会有不同的宽度和设备像素比，采用一些技巧来换算一些像素单位。`WXSS` 在底层支持新的尺寸单位 `rpx`，开发者可以免去换算的烦恼，只要交给小程序底层来换算即可，由于换算采用的浮点数运算，所以运算结果会和预期结果有一点点偏差。
2. 提供了全局的样式和局部样式。和前边 `app.json`, `page.json` 的概念相同，你可以写一个 `app.wxss` 作为全局样式，会作用于当前小程序的所有页面，局部页面样式 `page.wxss` 仅对当前页面生效。
3. 此外 `WXSS` 仅支持部分 `CSS` 选择器

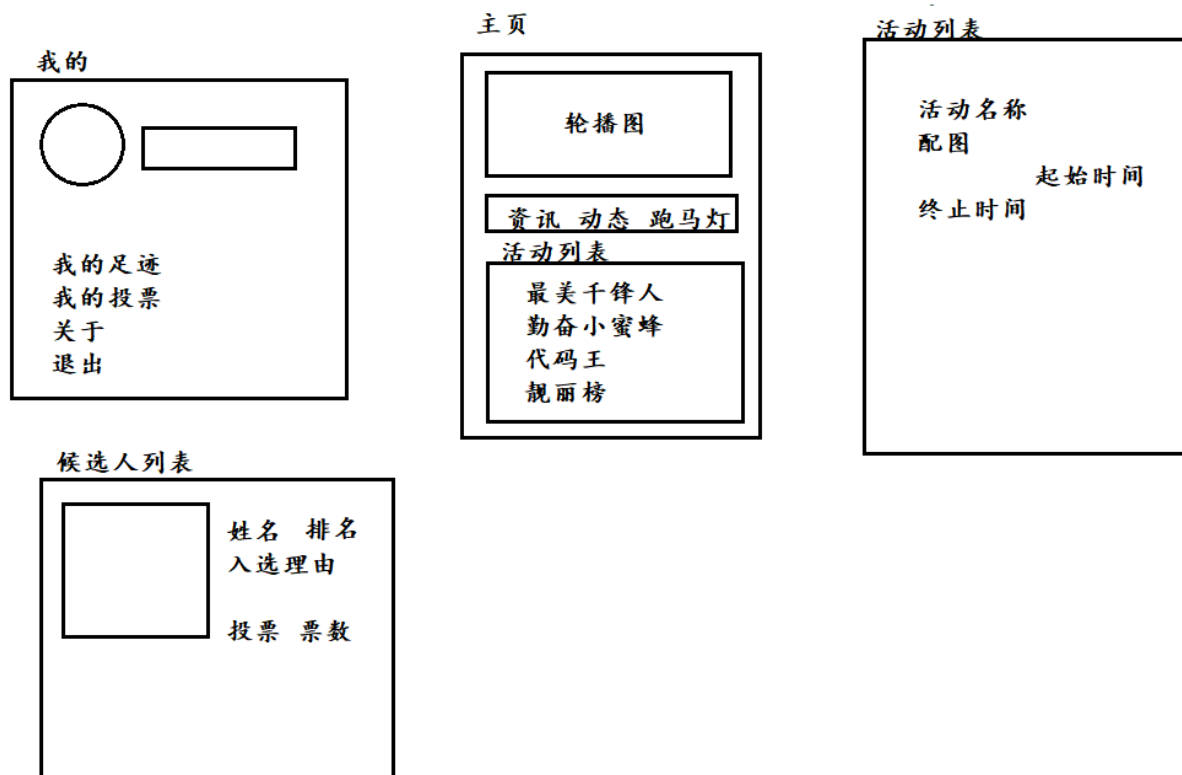
4.JS 逻辑交互

一个服务仅仅只有界面展示是不够的，还需要和用户做交互：响应用户的点击、获取用户的位置等等。在小程序里边，我们就通过编写 `JS` 脚本文件来处理用户的操作

三、小程序制作

3.1 需求分析

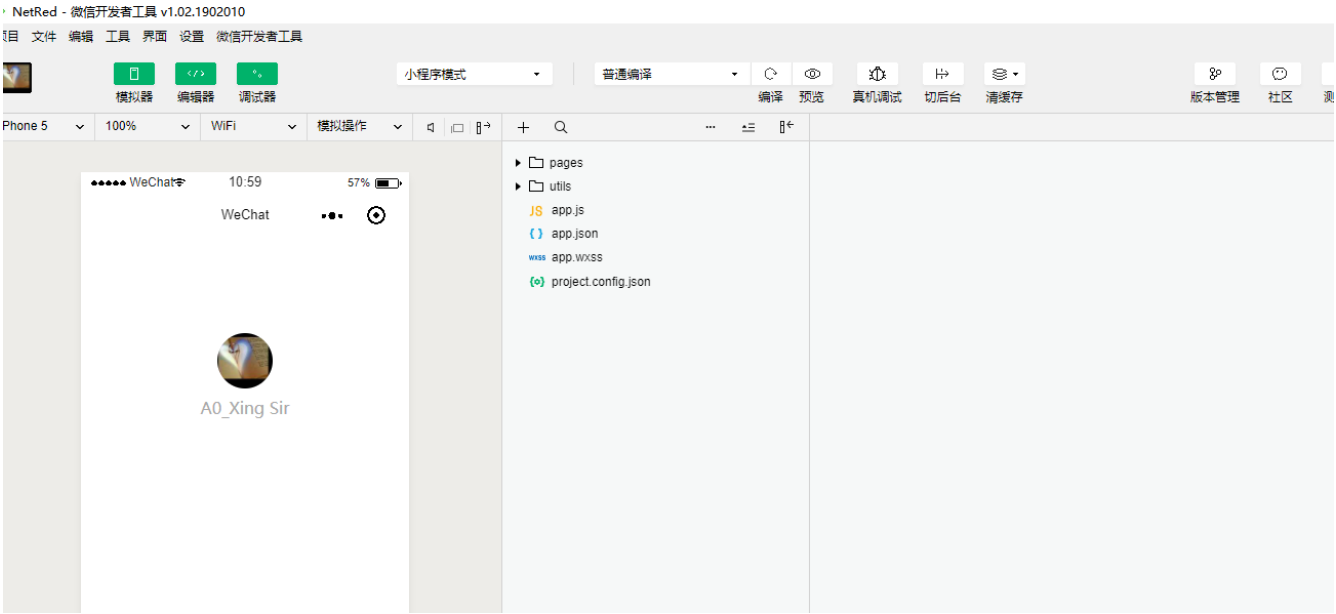
最美的你



效果参考: <http://www.jisuapp.cn/make/makeapp/13kz23Cb9x.html>

3.2 小程序开发

1、创建项目



2、开发页面

1.主页

四、附录

4.1 组件

视图容器
基础内容
表单组件
导航
媒体组件
地图
画布
开放能力
原生组件说明
无障碍访问

名称	功能说明
movable-view	可移动的视图容器，在页面中可以拖拽滑动
cover-image	覆盖在原生组件之上的图片视图
cover-view	覆盖在原生组件之上的文本视图
movable-area	movable-view的可移动区域
scroll-view	可滚动视图区域
swiper	滑块视图容器
swiper-item	仅可放置在swiper组件中，宽高自动设置为100%
view	视图容器

4.2 样式

WXSS:

wxss 选择器:

选择器	样例	样例描述
.class(类选择器)	.intro	选择所有拥有class="intro"的组件
#id(id选择器)	#firstname	选择拥有id="firstname"的组件
element	view	选择所有view组件
element, element	view checkbox	选择所有文档的view组件和所有的checkbox组件
::after	view::after	在view组件后边插入内容
::before	view::before	在view组件前边插入内容

一、wxss display(显示)

属性	说明
flex	多栏多列布局
inline-block	行内块元素
inline	此元素会被显示为内联元素，元素前后没有换行符
inline-table	作为内联表格来显示（类似 <code><table></code> ），表格前后没有换行符

属性

说明

inline-flex

将对象作为内联块级弹性伸缩盒显示

none

此元素不会被显示

block

此元素将显示为块级元素，此元素前后会带有换行符

list-item

此元素会作为列表显示

table

会作为块级表格来显示（类似 `<table>`），表格前后带有换行符）））））

table-footer-group

作为一个或多个行的分组来显示（类似 `<tr>`）

table-caption

作为一个表格标题显示（类似 `<caption>`）

table-cell

作为一个表格单元格显示（类似 `<td>`）

和 `<tr>`）

table-column

作为一个单元格列显示（类似 `<col>`）

table-column-group

作为一个或多个列的分组来显示（类似 `<tr>`）

table-row

作为一个表格行显示（类似 `<tr>`）

table-row-group

作为一个或多个行的分组来显示（类似 `<tr>`）

table-header-group

作为一个或多个行的分组来显示（类似 `<tr>`）

inherit

从父元素继承 `display` 属性的值

二、wxss position(定位)

属性

说明

absolute

生成绝对定位的元素，相对于 `static` 定位以外的第一个父元素进行定位。元素的位置通过 `"left"`, `"top"`, `"right"` 以及 `"bottom"` 属性进行规定。

relative

生成相对定位的元素，相对于其正常位置进行定位。因此，`"left:20"` 会向元素的 `LEFT` 位置添加 20 像素。

fixed

生成绝对定位的元素，相对于浏览器窗口进行定位。元素的位置通过 `"left"`, `"top"`, `"right"` 以及 `"bottom"` 属性进行规定。

static

默认值。没有定位，元素出现在正常的流中（忽略 `top`, `bottom`, `left`, `right` 或者 `z-index` 声明）

inherit

规定应该从父元素继承 `position` 属性的值

三、wxss float(浮动)

属性	说明			
		属性	说明	
		left	元素向左浮动	
		right	元素向右浮动	
		none默认值	元素不浮动，并会显示在其在文本中出现的位置。	
		inherit	规定应该从父元素继承 float 属性的值。	
		四、wxss background(背景)		
		属性	说明	语法(属性值)
		background	简写属性，作用是将背景属性设置在一个声明中	background: color position size repeat origin clip attachment image;
		background-color	指定要使用的背景颜色	
		background-position	指定背景图像的位置	background-position:center
		background-size	指定背景图片的大小	background-size:80px 60px;宽度 高度
		background-repeat	指定如何重复背景图像	repeat,repeat-x,repeat-y,no-repeat,inherit
		background-origin	指定背景图像的定位区域	padding-box 背景图像填充框的相对位置 border-box 背景图像边界框的相对位置 content-box 背景图像的相对位置的内容框
		background-clip	指定背景图像的绘画区域	属性值，同上
		background-attachment	设置背景图像是否固定或者随着页面的其余部分滚动。	scroll 背景图片随页面的其余部分滚动。这是默认fixed 背景图像是固定的inherit 指定background-attachment的设置应该从父元素继承local 背景图片随滚动元素滚动
		background-image	指定要使用的一个或多个背景图像	url('URL') 图像的URLnone 无图像背景会显示。这是默认inherit 指定背景图像应该从父元素继承

属性	说明																															
		<div>五、wxss border(边框)</div> <table><tr><th>属性</th><th>说明</th><th>语法(属性值)</th></tr><tr><td>border</td><td>简写属性，用于把针对四个边的属性设置在一个声明</td><td>border:5px solid red;</td></tr><tr><td>border-width</td><td>用于为元素的所有边框设置宽度，或者单独地为各边边框设置宽度</td><td>border-top-width 上右下左边框厚度 属性值：thin medium thick length</td></tr><tr><td>border-style</td><td>设置元素所有边框的样式，或者单独地为各边设置边框样式。</td><td>border-top-width 上右下左边框样式 属性值：solid,dashed,dotted,double等</td></tr><tr><td>border-color</td><td>元素的所有边框中可见部分的颜色，或为 4 个边分别设置颜色</td><td>border-top-width 上右下左边框颜色</td></tr></table> <div>六、wxss 轮廓 (outline)</div> <table><tr><th>属性</th><th>说明</th><th>语法(属性值)</th></tr><tr><td>outline</td><td>在一个声明中设置所有的外边框属性</td><td>outline: outline-color, outline-style, outline-width</td></tr><tr><td>outline-color</td><td>设置外边框的颜色</td><td></td></tr><tr><td>outline-style</td><td>设置外边框的样式。</td><td>属性值：solid,dashed,dotted,double等</td></tr><tr><td>outline-width</td><td>设置外边框的宽度</td><td>属性值：thin medium thick length</td></tr></table> <div>七、wxss 文本属性 (text)</div>	属性	说明	语法(属性值)	border	简写属性，用于把针对四个边的属性设置在一个声明	border:5px solid red;	border-width	用于为元素的所有边框设置宽度，或者单独地为各边边框设置宽度	border-top-width 上右下左边框厚度 属性值：thin medium thick length	border-style	设置元素所有边框的样式，或者单独地为各边设置边框样式。	border-top-width 上右下左边框样式 属性值：solid,dashed,dotted,double等	border-color	元素的所有边框中可见部分的颜色，或为 4 个边分别设置颜色	border-top-width 上右下左边框颜色	属性	说明	语法(属性值)	outline	在一个声明中设置所有的外边框属性	outline: outline-color, outline-style, outline-width	outline-color	设置外边框的颜色		outline-style	设置外边框的样式。	属性值：solid,dashed,dotted,double等	outline-width	设置外边框的宽度	属性值：thin medium thick length
属性	说明	语法(属性值)																														
border	简写属性，用于把针对四个边的属性设置在一个声明	border:5px solid red;																														
border-width	用于为元素的所有边框设置宽度，或者单独地为各边边框设置宽度	border-top-width 上右下左边框厚度 属性值：thin medium thick length																														
border-style	设置元素所有边框的样式，或者单独地为各边设置边框样式。	border-top-width 上右下左边框样式 属性值：solid,dashed,dotted,double等																														
border-color	元素的所有边框中可见部分的颜色，或为 4 个边分别设置颜色	border-top-width 上右下左边框颜色																														
属性	说明	语法(属性值)																														
outline	在一个声明中设置所有的外边框属性	outline: outline-color, outline-style, outline-width																														
outline-color	设置外边框的颜色																															
outline-style	设置外边框的样式。	属性值：solid,dashed,dotted,double等																														
outline-width	设置外边框的宽度	属性值：thin medium thick length																														

属性	说明			
		属性	说明	语法(属性值)
		color	设置文本颜色	
		direction	设置文本方向。	ltr:文本方向从左到右;rtl:文本方向从右到左
		letter-spacing	设置字符间距	
		line-height	设置行高	
		text-align	对齐元素中的文本	left: 把文本排列到左边。默认值, 由浏览器决定。right: 把文本排列到右边。center: 把文本排列到中间。justify: 实现两端对齐文本效果。inherit: 规定应该从父元素继承 text-align 属性的值。
		text-decoration	向文本添加修饰	underline 定义文本下的一条线。overline 定义文本上的一条线。line-through 定义穿过文本下的一条线。blink 定义闪烁的文本。
		text-indent	缩进元素中文本的首行	
		text-shadow	设置文本阴影	text-shadow: h-shadow v-shadow blur color;h-shadow:水平阴影的位置,允许负值;v-shadow:垂直阴影的位置,允许负值;blur:模糊的距离;color:阴影的颜色
		text-transform	控制元素中的字母	capitalize 文本中的每个单词以大写字母开头。uppercase 定义仅有大写字母。lowercase 定义无大写字母, 仅有小写字母。

属性	说明			
		属性	说明	语法(属性值)
		unicode-bidi	设置或返回文本是否被重写	
		vertical-align	设置元素的垂直对齐	
		white-space	设置元素中空白的处理方式	
		word-spacing	设置字间距	
八、wxss 字体属性 (font)				

属性	说明			
		属性	说明	语法(属性值)
		font	在一个声明中设置所有字体属性	font:font-style font-variant font-weight font-size/line-height font-family(按顺序)
		font-style	指定文本的字体样式	normal 默认值。浏览器显示一个标准的字体样式。italic 浏览器会显示一个斜体的字体样式。oblique 浏览器会显示一个倾斜的字体样式。inherit 规定应该从父元素继承字体样式。
		font-variant	以小型大写字体或者正常字体显示文本	normal 默认值。浏览器会显示一个标准的字体。small-caps 浏览器会显示小型大写字母的字体。inherit 规定应该从父元素继承 font-variant 属性的值。
		font-weight	指定字体的粗细	normal 默认值。定义标准的字符。bold 定义粗体字符。bolder 定义更粗的字符。lighter 定义更细的字符。inherit 规定应该从父元素继承字体的粗细。
		font-size	指定文本的字体大小	smaller 把 font-size 设置为比父元素更小的尺寸。larger 把 font-size 设置为比父元素更大的尺寸。length 把 font-size 设置为一个固定的值。% 把 font-size 设置为基于父元素的一个百分比值。
		font-family	指定文本的字体系列	
九、wxss margin(外边距) (margin)				

属性	说明			
		属性	说明	语法(属性值)
		margin	在一个声明中设置所有外边距属性。	margin:10px 5px 15px 20px;(上边距, 右边距, 下边距, 左边距)
		margin-top	设置元素的上外边距。	
		margin-right	设置元素的右外边距。	
		margin-bottom	设置元素的下外边距。	
		margin-left	设置元素的左外边距	
		十、wxss padding(填充) (padding)		
		属性	说明	语法(属性值)
		padding	使用缩写属性设置在一个声明中的所有填充属性	padding:10px 5px 15px 20px;(上填充, 右填充, 下填充, 左填充)
		padding-top	设置元素的顶部填充。	
		padding-right	设置元素的右部填充	
		padding-bottom	设置元素的底部填充	
		padding-left	设置元素的左部填充	