# 微信小程序\_01

# 今日目标9

* 能够使用laravel中间件检查用户权限
* 能够注册成功小程序账号
* 能够成功安装小程序开发工具
* 能够运行第一个简单的hello world小程序
* 掌握小程序文件结构

app

app.json

app.wxss

app.js

pages

page.wxml

page.wxss

page.json

page.js

* 了解app.json配置文件与作用

设置小程序的窗口 底部菜单 网络超时，开启debug

* 了解小程序的生命周期

onLoad

onShow

onReady

onShow/onHide 之间来回切换

* 掌握wxml基本语法

数据绑定 {{}}

wx:for

wx:for-index

wx:for-item

wx:key

wx:if wx:elif wx:else

模板 template

定义 <template name>

调用 template is data

引用

import

include

* 掌握wxss基本样式语法

类似 css样式

app.wxss

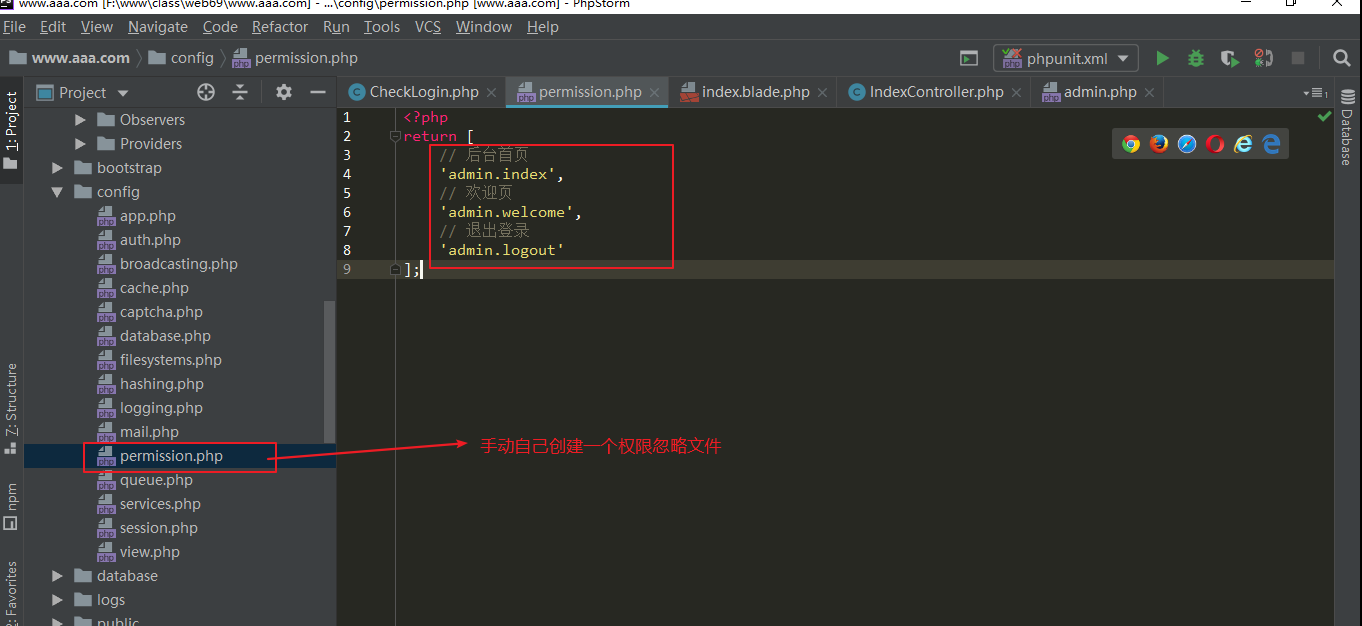
page.wxss

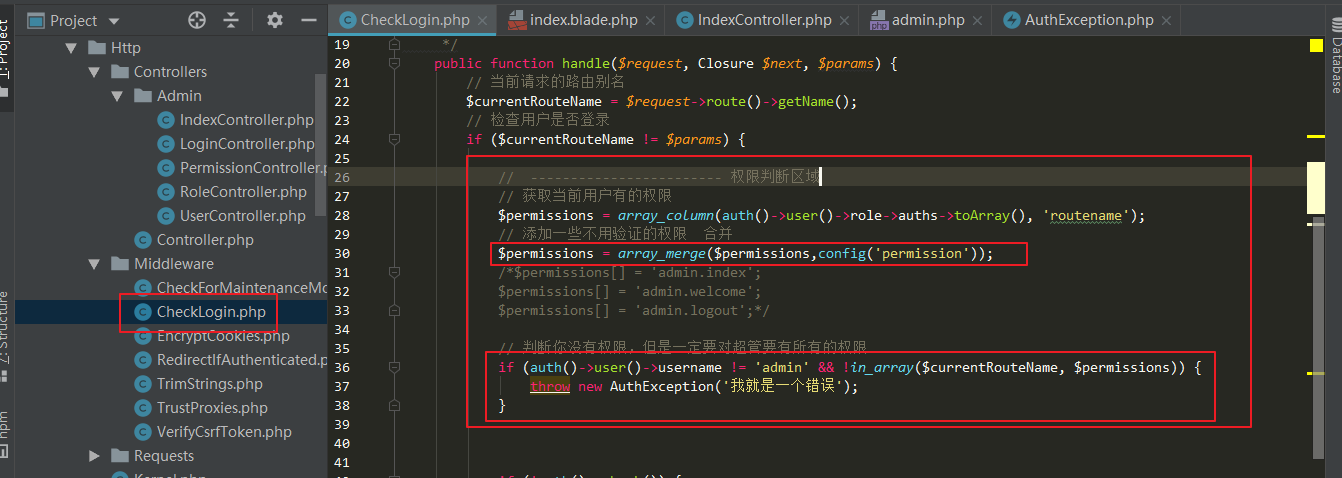
行内

# 能够使用laravel中间件检查用户权限

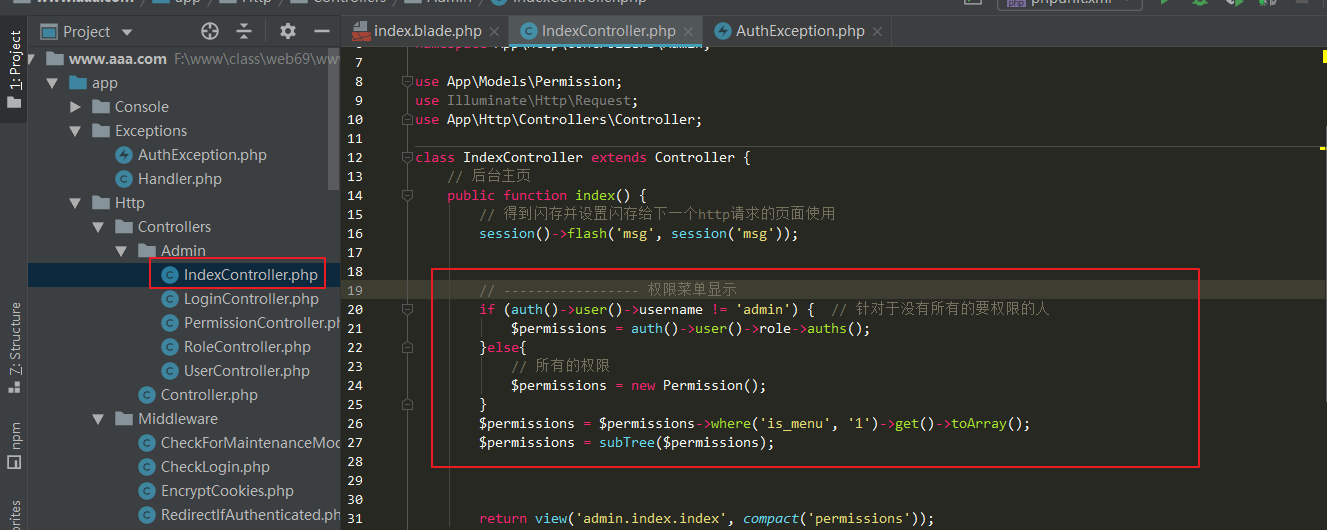
## 1、在中间件中判断

创建权限的忽略文件

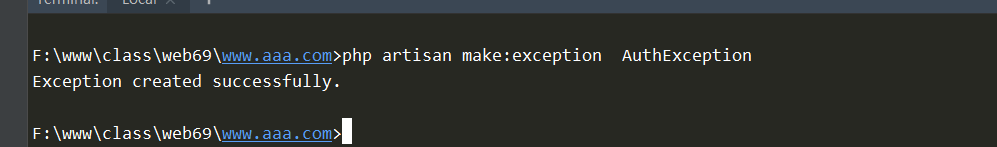


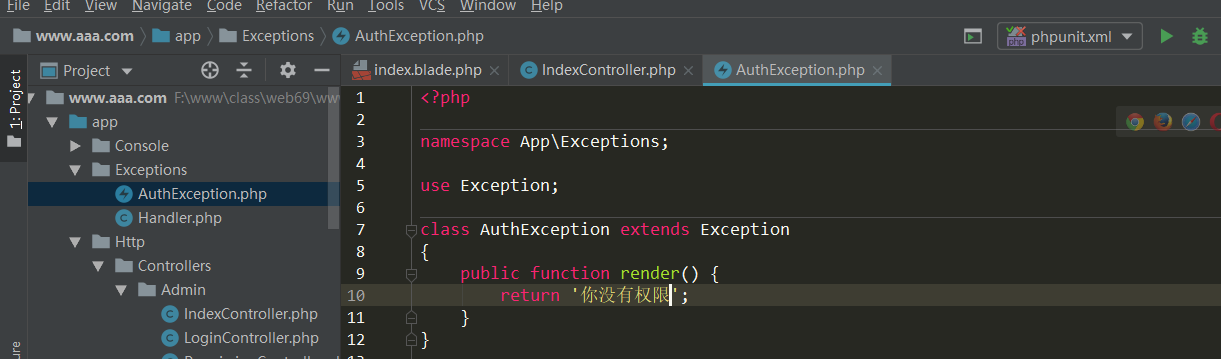


## 2、对应用户就对应有那些菜单



## 3、异常的创建



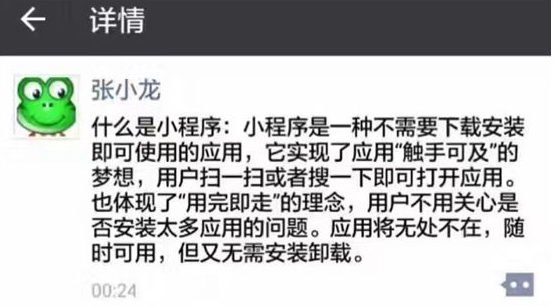


# 一、小程序介绍

## 1.1、小程序是什么

官方文档：<https://developers.weixin.qq.com/miniprogram/dev/>

微信小程序，简称小程序，英文名 Mini Program，是一种不需要下载安装即可使用的应用，它实现了应用“触手可及”的梦想，用户扫一扫或者搜一下即可打开应用。也体现了“用完即走”的理念，用户不用关心是否安装太多应用的问题。应用将无处不在，随时可用，但又无需安装卸载。对于开发者而言，小程序开发门槛相对较低，难度不及APP，能够满足简单的基础应用，适合生活服务类线下商铺以及非刚需低频应用的转换



## 1.2、发展历史

2016年1月11日，微信之父张小龙公布微信小程序

2016年9月21日，微信小程序正式开启内测

2017年1月9日，第一批小程序正式上线

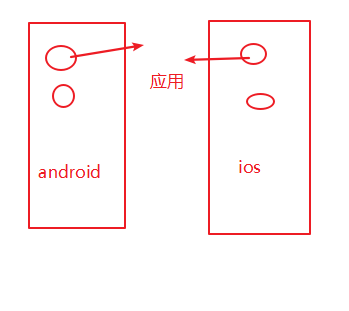
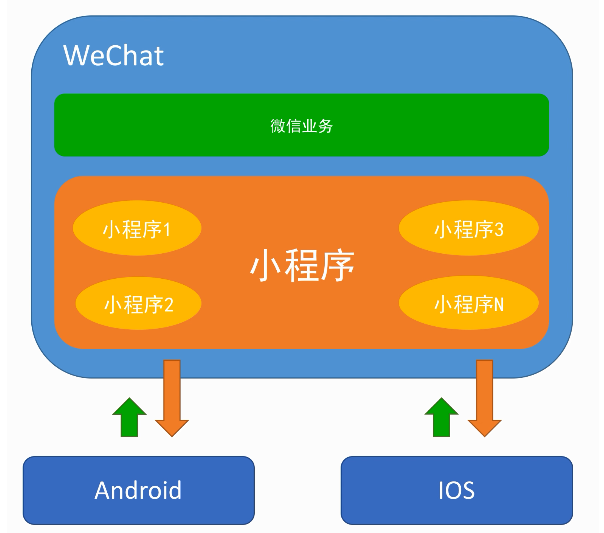
2017年12月28日，微信更新的 6.6.1 版开放了小游戏，微信启动页面推荐了小游戏《跳一跳》，可以通过小程序找到已经玩到小游戏

2018年1月25日，微信团队在“微信公众平台”发布公告称，“从移动应用分享至微信的小程序页面，用户访问时支持打开来源应用。

## 1.3、如何理解小程序

* 不是HTML5
* 即用即走，随手可得
* 拥有离线能力
* 一次开发，多端兼容
* 优美的操作体验
* 基于微信客户端中

## 1.4、小程序与传统App区别



小程序

局限性：必须依赖于微信，不能独立运行。

优势：不需要考虚兼容问题，安装的问题，开发难度小。

传统app：

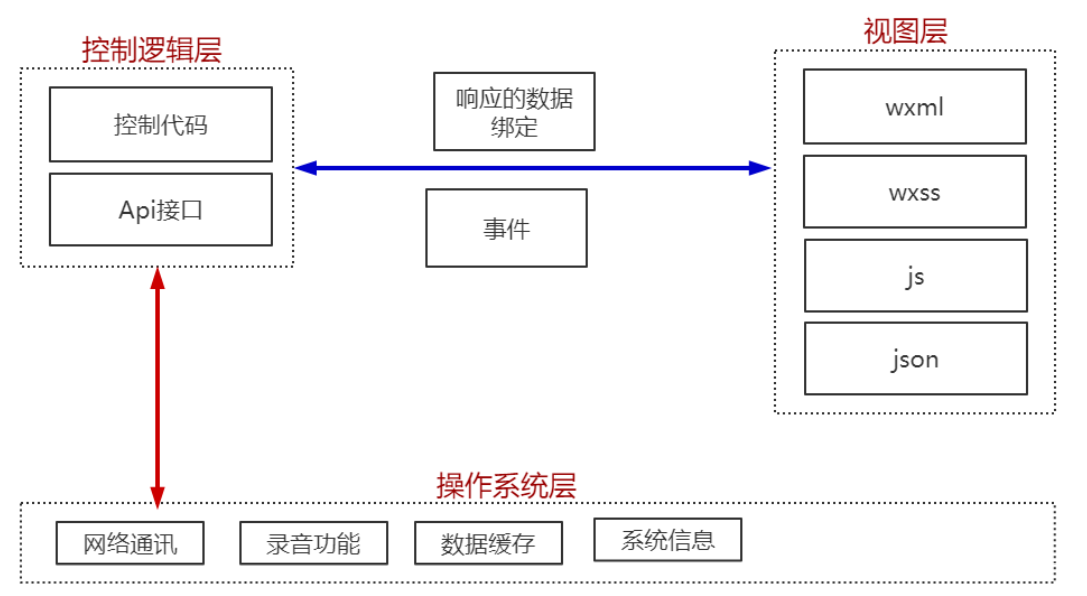
优势：独立运行，不需要依赖于谁，可以适合所有的业务需求

局限性：需要用户安装，解决适配(开发兼容问题),开发难度大

## 1.5、谁可以注册开发小程序



## 1.6、小程序框架结构



# 二、申请账号与登录设置

## 2.1、小程序体验 DEMO

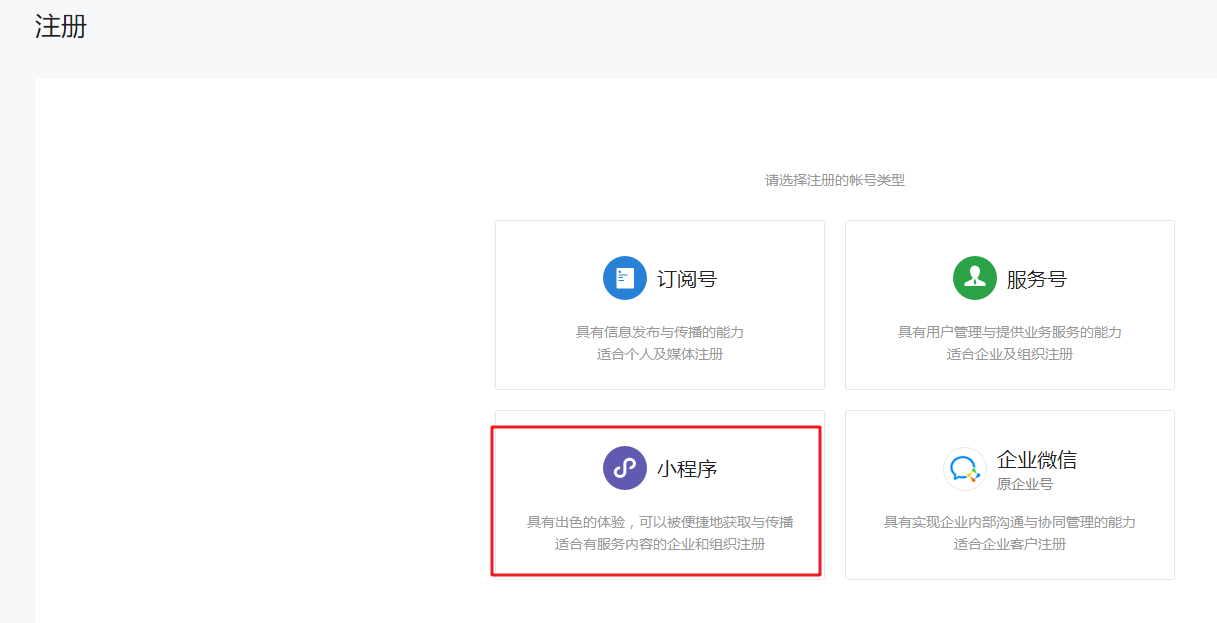
 

## 2.2、申请账号

官网: <https://mp.weixin.qq.com/>



选择小程序进行安装



填写相关资料完成申请流程



## 2.3、登录



进行二次验证微信扫码验证。

填写注册时的邮箱和密码进行登录功能

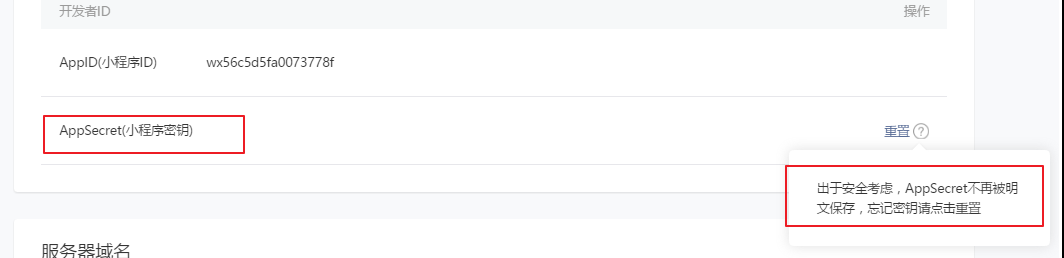






## 2.4、获取开发appid





有此appid就可以进行小程序的开发工作。

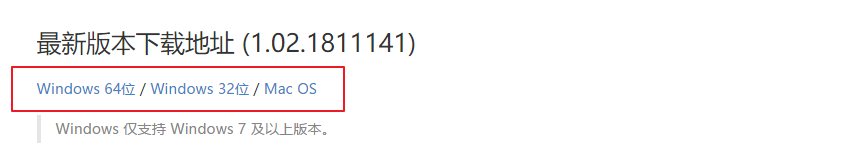
# 三、开发工具

## 3.1、概览

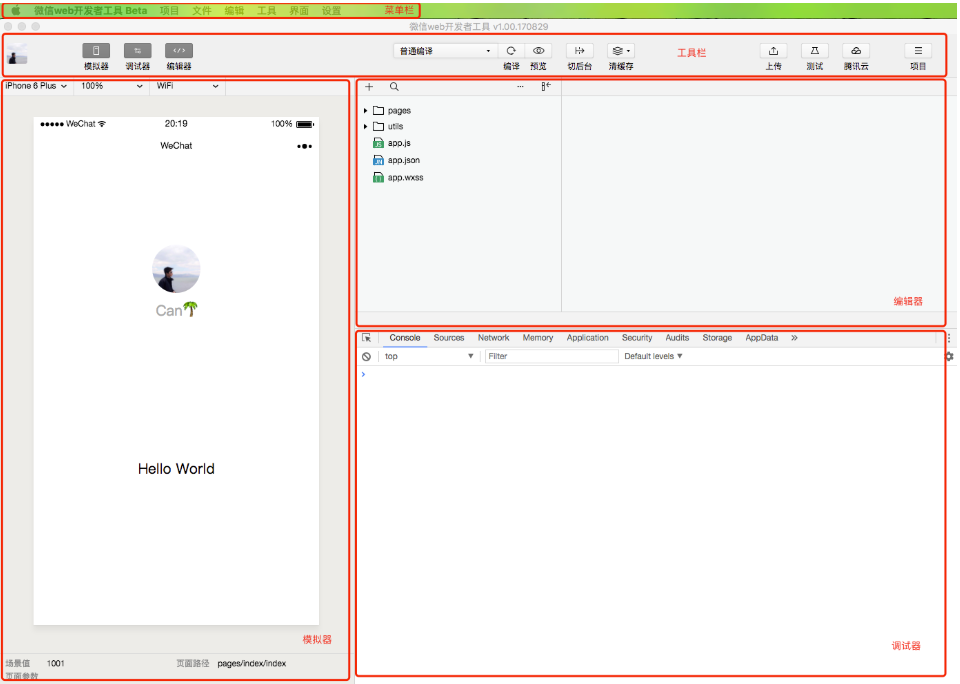
为了帮助开发者简单和高效地开发和调试微信小程序，腾讯在原有的公众号网页调试工具的基础上，推出了全新的微信开发者工具，集成了公众号网页调试和小程序调试两种开发模式。

## 3.2、下载安装

下载地址：<http://t.cn/RrKI5a3>

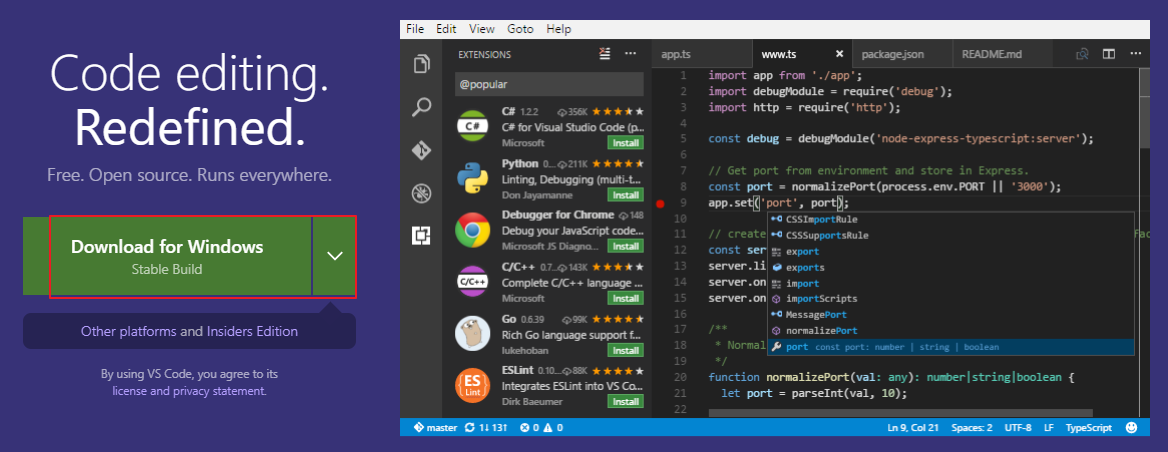


在windows下面安装的方式和其他的普通软件安装方式一致，一路下一步即可。

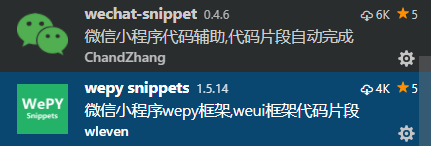


下载vscode编辑器

下载地址：<https://code.visualstudio.com/>



vscode安装成功后，开发小程序需要安装如下的两个插件



## 3.3、第一个小程序helloworld

如果没有注册账号，我们可以用测试号来进行开发

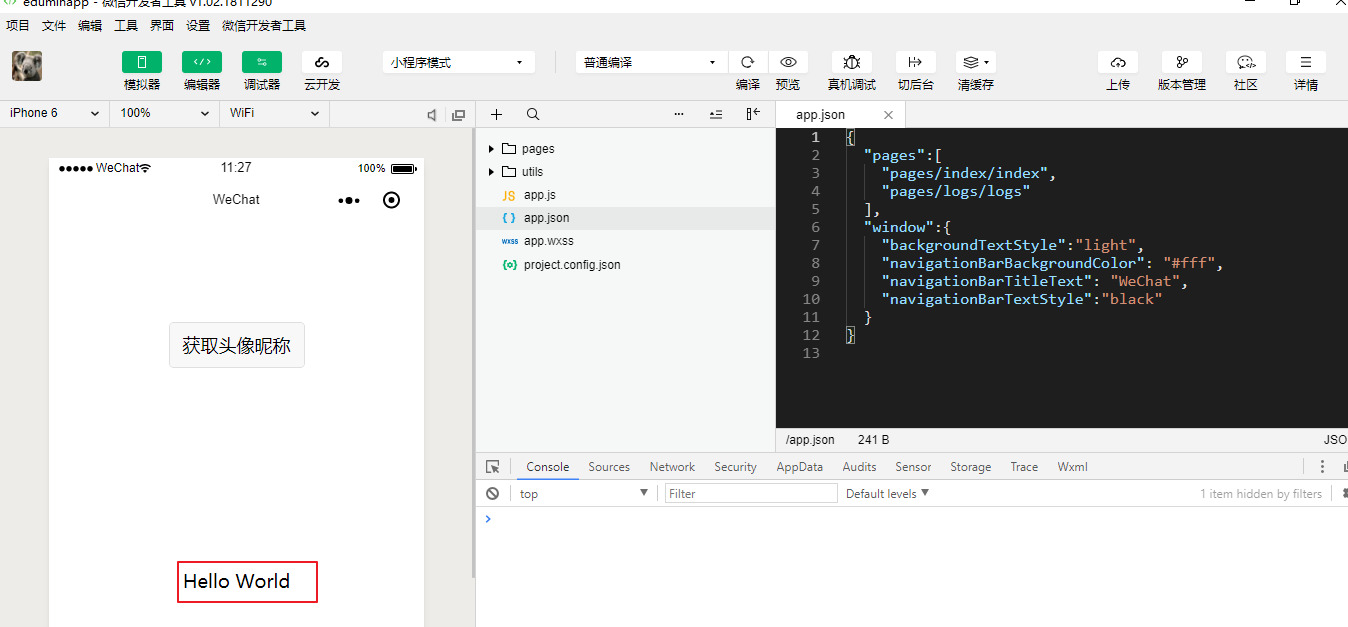
<https://developers.weixin.qq.com/weloginpage?redirect_url=%2Fsandbox>

当然如果注册过，有了appid，则用我们注册过的



打开官方开发工具，扫一下，后选择小程序



# 四、小程序的文件结构及配置

## 4.1、文件结构

通过开发者工具创建了【**建立普通快速启动模板**】项目。这个项目里边生成了不同类型的文件: **红色**必须不可缺少的文件

* .json 后缀的 JSON 配置文件
* .wxml 后缀的 WXML 模板文件 类html
* .wxss 后缀的 WXSS 样式文件 类css
* .js 后缀的 JS 脚本逻辑文件 业务代码所书写的区域

小程序包含一个描述整体程序的 app 和多个描述各自页面的 pages。

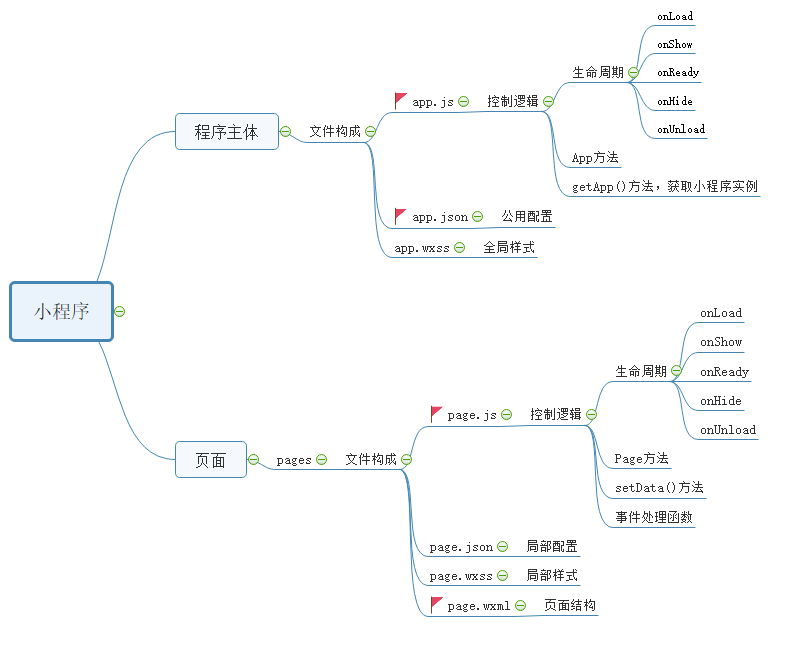
一个小程序主体部分由三个文件组成，必须放在项目的根目录，如下：

|  |  |  |
| --- | --- | --- |
| 文件 | 必填 | 作用 |
| app.js | 是 | 小程序逻辑 |
| app.json | 是 | 小程序公共设置 |
| app.wxss | 否 | 小程序公共样式表 |

一个小程序page页面由四个文件组成，分别是：

|  |  |  |
| --- | --- | --- |
| 文件类型 | 必填 | 作用 |
| js | 是 | 页面逻辑 |
| wxml | 是 | 页面结构 |
| wxss | 否 | 页面样式表 |
| json | 否 | 页面配置 |

注意：为了方便开发者减少配置项，描述页面的四个文件必须具有相同的路径与文件名

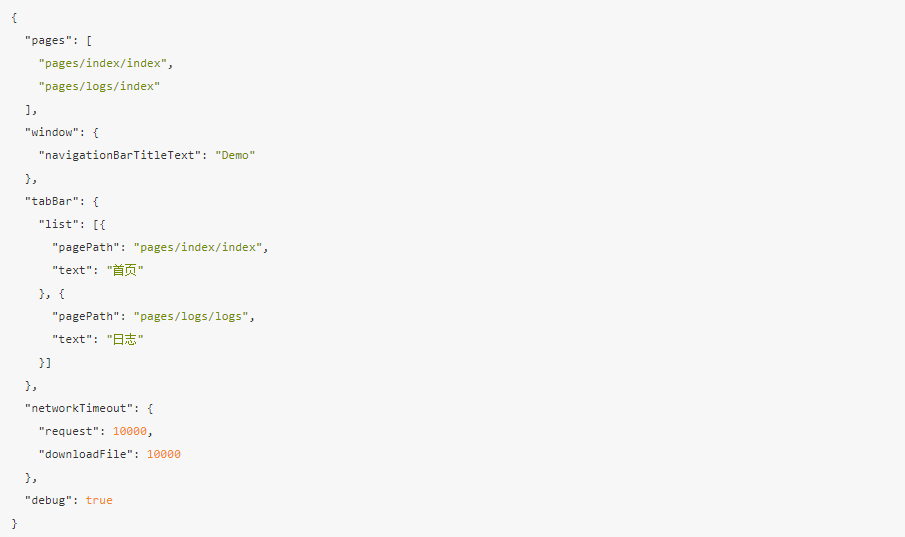


## 4.2、小程序配置 app.json

参考网址：<https://developers.weixin.qq.com/miniprogram/dev/framework/config.html>

app.json 文件用来对微信小程序进行全局配置，决定页面文件的路径、窗口表现、设置网络超时时间、设置多 tab [底部或顶部菜单]等。

app.json中不能添加任何注释，key和value必须用双引号引起来，否则会报错。



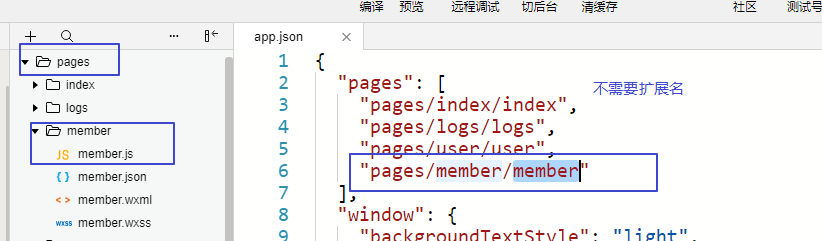
### 4.2.1、app.json 配置项列表

|  |  |  |  |
| --- | --- | --- | --- |
| 属性 | 类型 | 必填 | 描述 |
| pages | String Array | 是 | 设置页面路径 |
| window | Object | 否 | 设置默认页面的窗口表现 |
| tabBar | Object | 否 | 设置底部 tab 的表现 |
| networkTimeout | Object | 否 | 设置网络超时时间 |
| debug | Boolean | 否 | 设置是否开启 debug 模式 |

### 4.2.2、pages

接受一个数组，每一项都是字符串，来指定小程序由哪些页面组成。每一项代表对应页面的【路径+目录名+文件名(不包含后缀名)】信息，数组的第一项代表小程序的初始页面(首页)。小程序中新增/减少页面，都需要对 pages 数组进行修改。

注：数组的第一项为小程序初始页面



### 4.2.3、window

https://developers.weixin.qq.com/miniprogram/dev/framework/config.html#全局配置

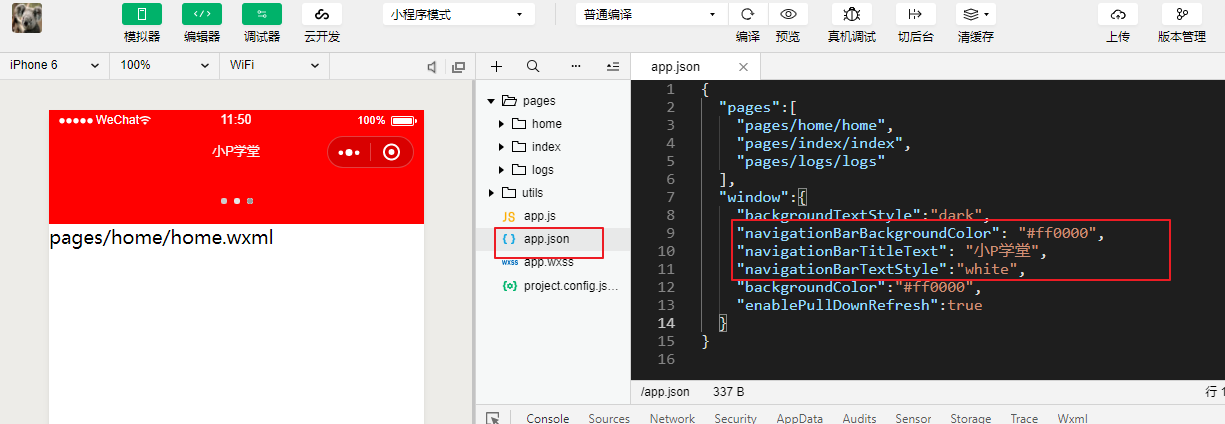
用于设置小程序的状态栏、导航条、标题、窗口背景色。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 属性 | 类型 | 默认值 | 描述 | 最低版本 |
| navigationBarBackgroundColor | HexColor | #000000 | 导航栏背景颜色，如"#000000" |  |
| navigationBarTextStyle | String | white | 导航栏标题颜色，仅支持 black/white |  |
| navigationBarTitleText | String |  | 导航栏标题文字内容 |  |
| navigationStyle | String | default | 导航栏样式，仅支持 default/custom。custom 模式可自定义导航栏，只保留右上角胶囊状的按钮 | 微信版本 6.6.0 |
| backgroundColor | HexColor | #ffffff | 窗口的背景色 |  |
| backgroundTextStyle | String | dark | 下拉 loading 的样式，仅支持 dark/light |  |
| backgroundColorTop | String | #ffffff | 顶部窗口的背景色，仅 iOS 支持 | 微信版本 6.5.16 |
| backgroundColorBottom | String | #ffffff | 底部窗口的背景色，仅 iOS 支持 | 微信版本 6.5.16 |
| enablePullDownRefresh | Boolean | false | 是否开启下拉刷新，详见页面相关事件处理函数 |  |
| onReachBottomDistance | Number | 50 | 页面上拉触底事件触发时距页面底部距离，单位为px |  |

注：HexColor（十六进制颜色值），如"#ff00ff"

注：navigationStyle 只在 app.json 中生效。开启 custom 后，低版本客户端需要做好兼容。开发者工具基础库版本切到 1.7.0（不代表最低版本，只供调试用）可方便切到旧视觉

注：客户端 6.7.2 版本开始，navigationStyle: custom 对 <web-view> 组件无效



### 4.2.4、tabBar

如果小程序是一个多 tab 应用（客户端窗口的底部或顶部有 tab 栏可以切换页面），可以通过 tabBar 配置项指定 tab 栏的表现，以及 tab 切换时显示的对应页面。

属性说明:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 属性 | 类型 | 必填 | 默认值 | 描述 |
| color | HexColor | 是 |  | tab 上的文字默认颜色 |
| selectedColor | HexColor | 是 |  | tab 上的文字选中时的颜色 |
| backgroundColor | HexColor | 是 |  | tab 的背景色 |
| borderStyle | String | 否 | black | tabbar上边框的颜色， 仅支持 black/white |
| list | Array | 是 |  | tab 的列表，详见 list 属性说明，最少2个、最多5个 tab |
| position | String | 否 | bottom | 可选值 bottom、top |

注：

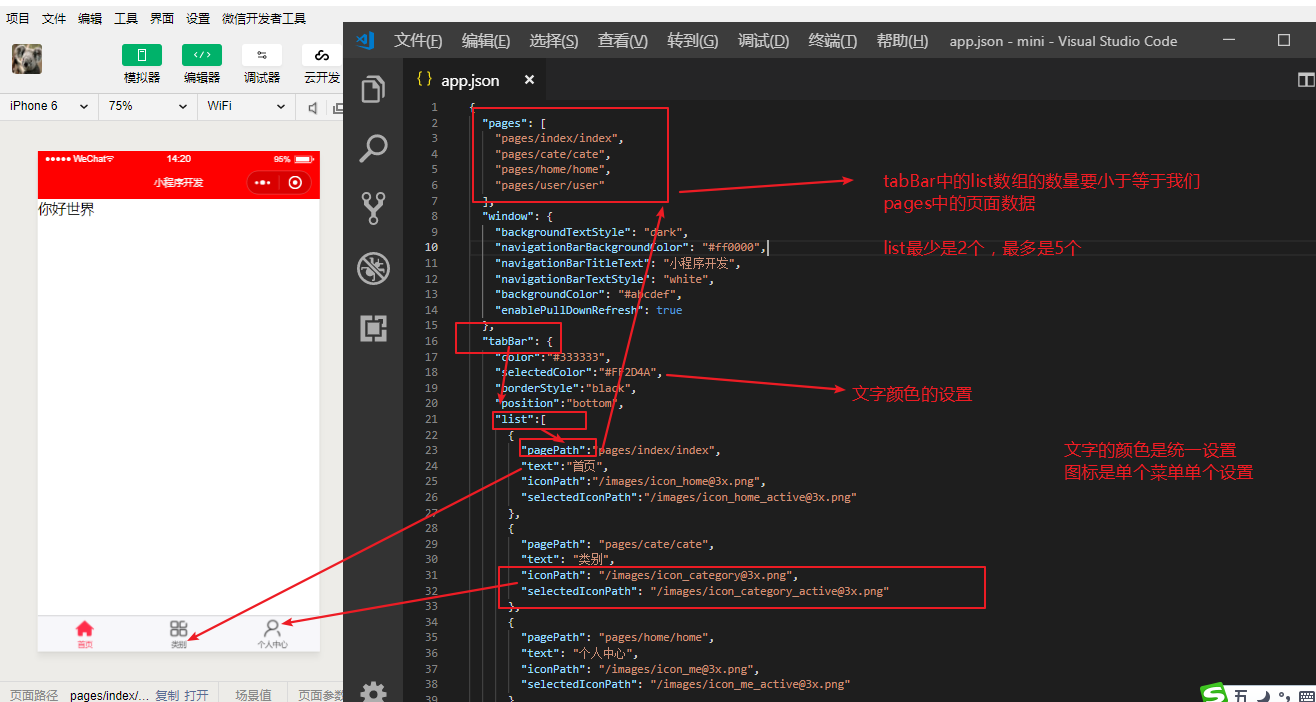
* 当设置 position 为 top 时，将不会显示 icon
* tabBar 中的 list 是一个数组，只能配置最少2个、最多5个 tab，tab 按数组的顺序排序。

不属于 tabBar 的页面不会显示 tabBar

其中 list 接受一个数组，数组中的每个项都是一个对象，其属性值如下：

|  |  |  |  |
| --- | --- | --- | --- |
| 属性 | 类型 | 必填 | 说明 |
| pagePath | String | 是 | 页面路径，必须在 pages 中先定义 |
| text | String | 是 | tab 上按钮文字 |
| iconPath | String | 否 | 图片路径，icon 大小限制为40kb，建议尺寸为 81px \* 81px，当 postion 为 top 时，此参数无效，不支持网络图片 |
| selectedIconPath | String | 否 | 选中时的图片路径，icon 大小限制为40kb，建议尺寸为 81px \* 81px ，当 postion 为 top 时，此参数无效 |



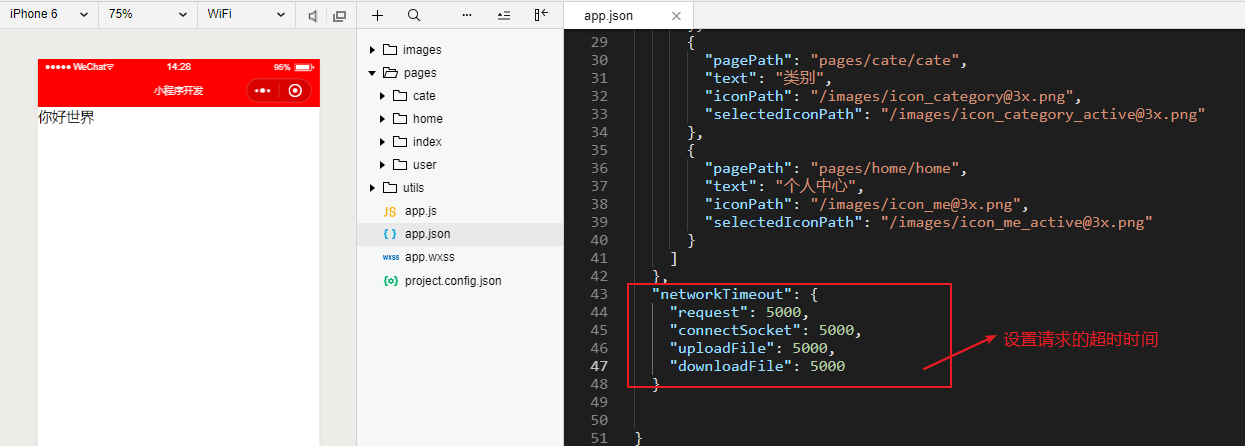


### 4.2.5、networkTimeout

可以设置各种网络请求的超时时间。

属性说明：

|  |  |  |  |
| --- | --- | --- | --- |
| 属性 | 类型 | 必填 | 说明 |
| request | Number | 否 | wx.request的超时时间，单位毫秒，默认为：60000 |
| connectSocket | Number | 否 | wx.connectSocket的超时时间，单位毫秒，默认为：60000 |
| uploadFile | Number | 否 | wx.uploadFile的超时时间，单位毫秒，默认为：60000 |
| downloadFile | Number | 否 | wx.downloadFile的超时时间，单位毫秒，默认为：60000 |

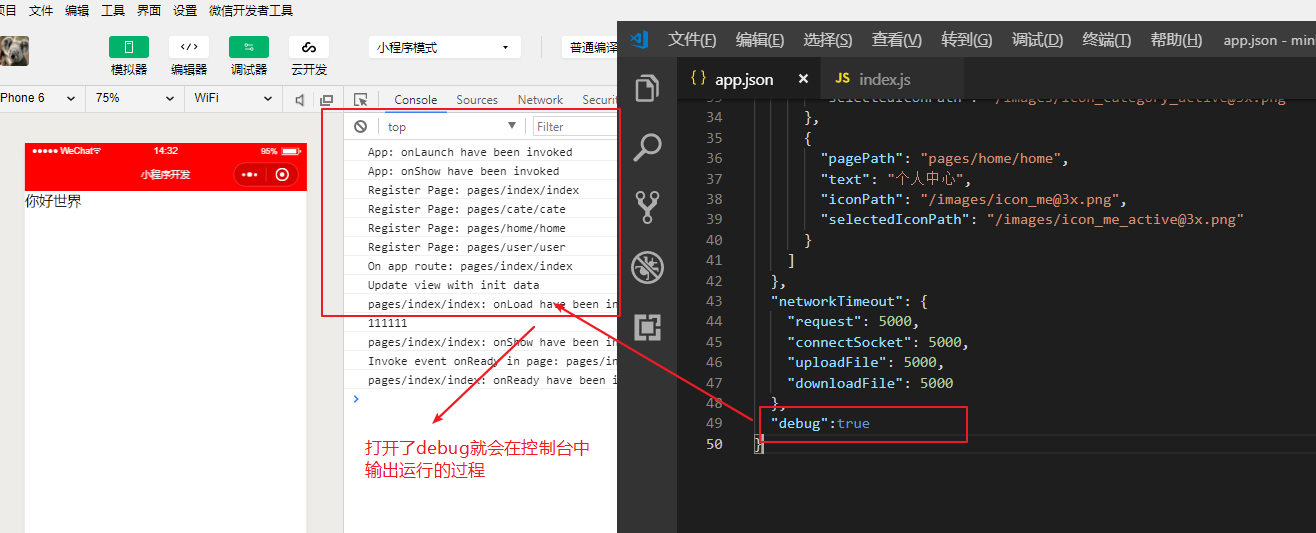


### 4.2.6、debug

可以在开发者工具中开启 debug 模式，在开发者工具的控制台面板，调试信息以 info 的形式给出，其信息有Page的注册，页面路由，数据更新，事件触发 。 可以帮助开发者快速定位一些常见的问题。

说明了，开发阶段，就要打开debug。上线时请关闭此选项，设置为false





# 五、生命周期

## onLoad(Object query)

页面加载时触发。一个页面只会调用一次，可以在 onLoad 的参数中获取打开当前页面路径中的参数。

参数说明

|  |  |  |
| --- | --- | --- |
| 名称 | 类型 | 说明 |
| query | Object | 打开当前页面路径中的参数 json对象 |

## onShow()

页面显示/切入前台时触发。

## onReady()

页面初次渲染完成时触发。一个页面只会调用一次，代表页面已经准备妥当，可以和视图层进行交互。

注意：对界面内容进行设置的 API 如wx.setNavigationBarTitle，请在onReady之后进行。详见生命周期

## onHide()

页面隐藏/切入后台时触发。 如 navigateTo 或底部 tab 切换到其他页面，小程序切入后台等。

## onUnload()

页面卸载时触发。如redirectTo或navigateBack到其他页面时。

# 六、视图结构 wxml

## 6.1、概述

从事过网页编程的人知道，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。

WXML和HTML不同点：

* 标签名字有点不一样
* 多了一些 wx:if 这样的属性以及 {{ }}这样的表达式

小程序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只需要负责数据的处理，然后再通过一种模板语法来进行界面结构展示。

将js文件中的数据传递到视图层中方法 (异步改就数据)

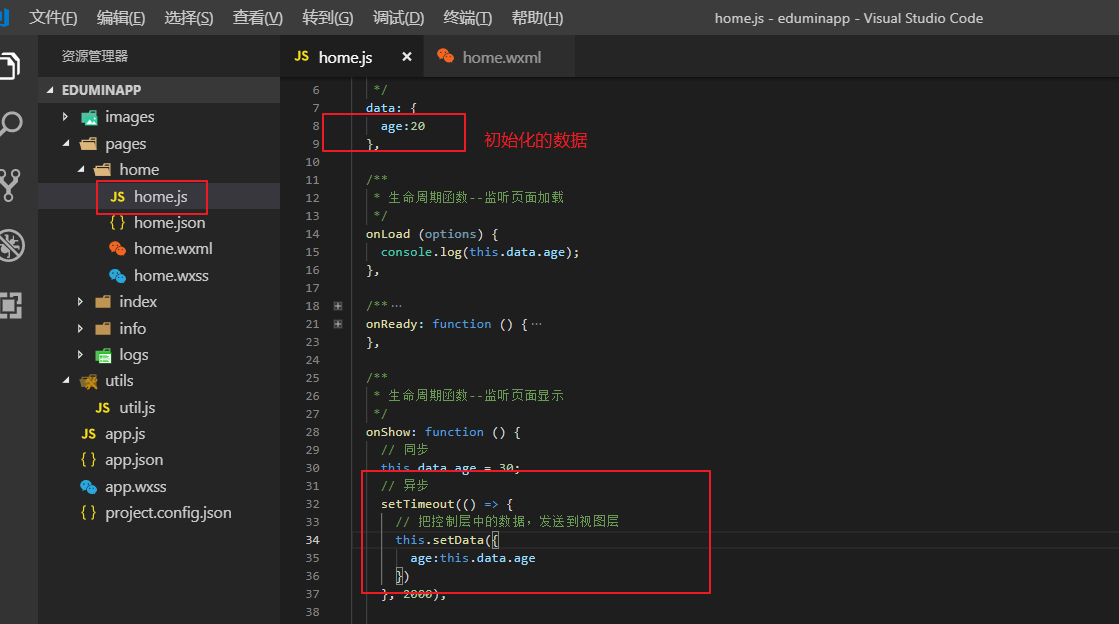
this.setData({ msg: "Hello World" })

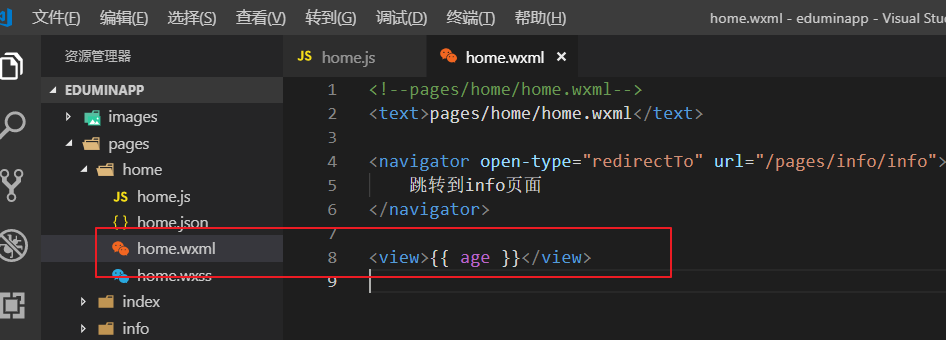
在视图层中显示数据

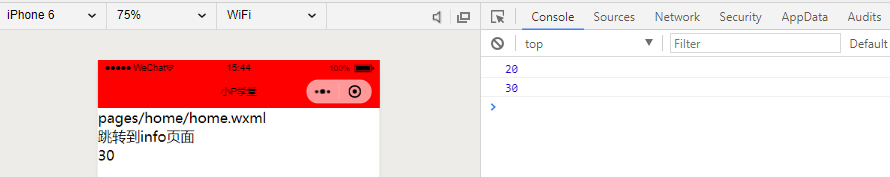
<view>{{变量}}</view>

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的，还需要 if/else, for等控制能力，在小程序里边，这些控制能力都用 wx: 开头的属性来表达。



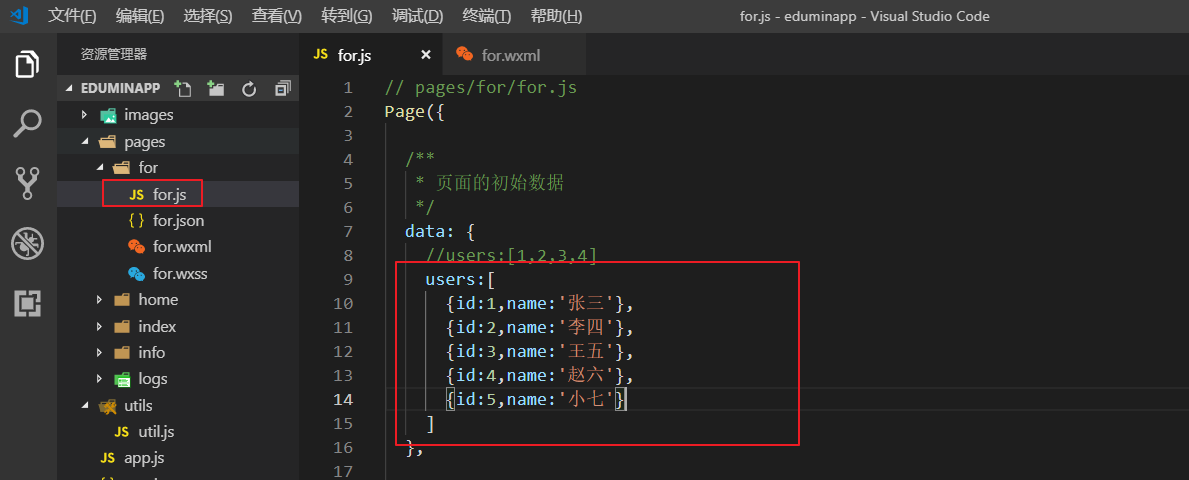


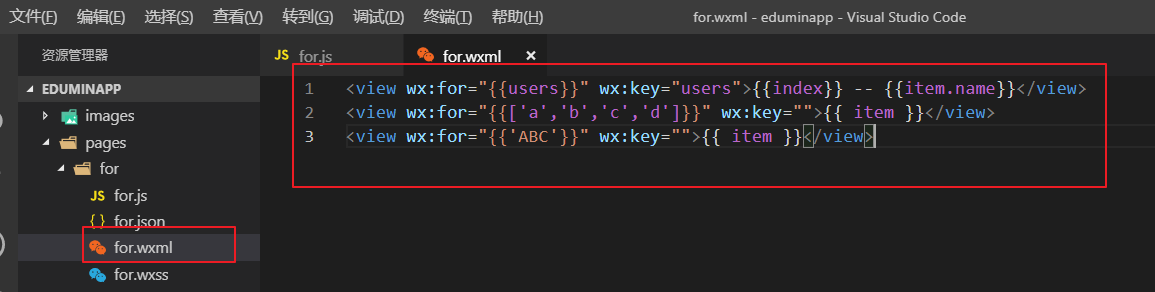


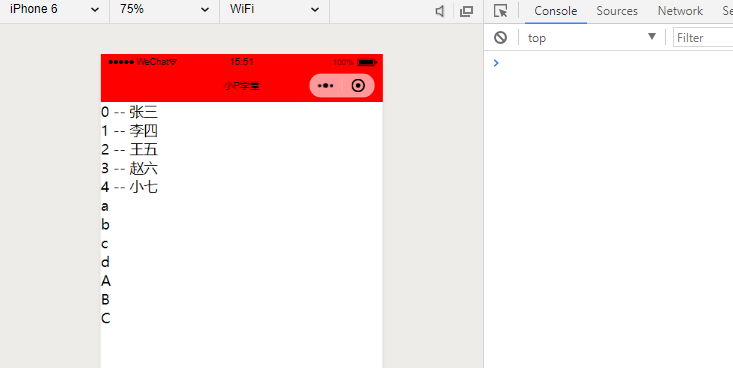
## 6.2、列表渲染 for

语法：wx:for

* wx:for-index可以指定数组当前下标的变量名 默认名为 index
* wx:for-item 可以指定数组当前元素的变量名 默认名为 item
* wx:key 可以定义也可以不定义 唯一的标识符





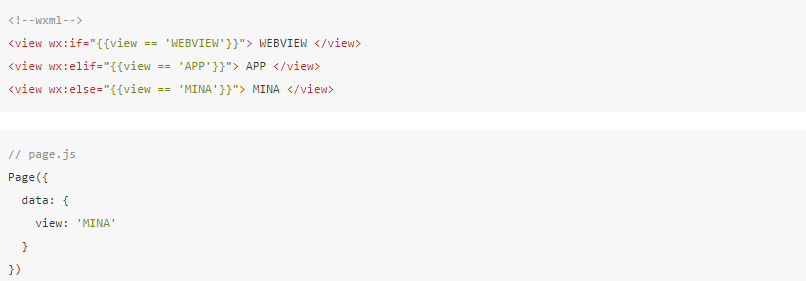


## 6.3、条件渲染

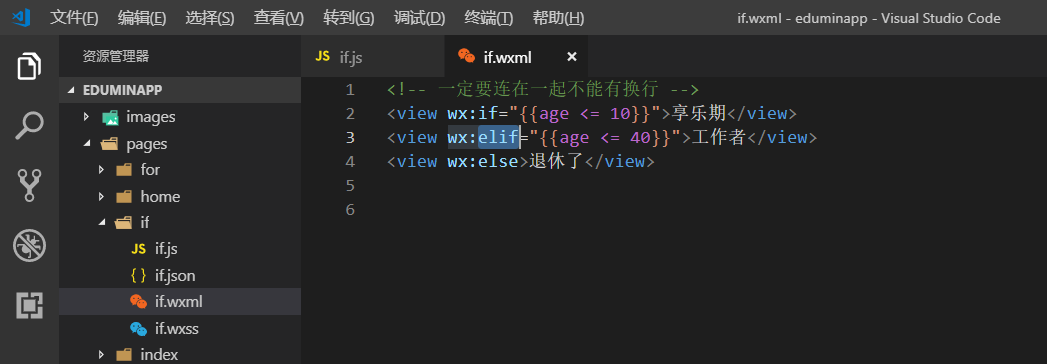
语法：wx:if

在框架中，使用 wx:if="{{condition}}" 来判断是否需要渲染该代码块：

也可以用 wx:elif 和 wx:else 来添加一个 else 块：









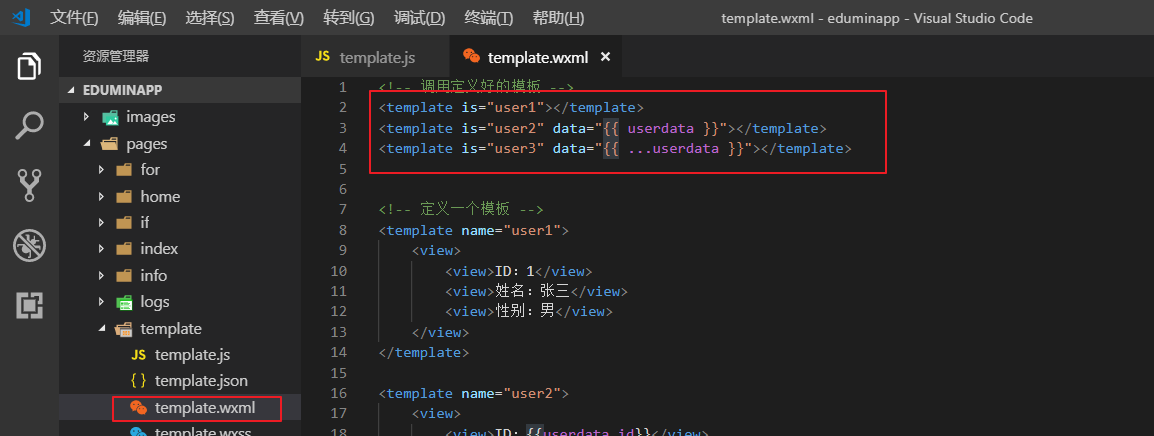
## 6.4、模板

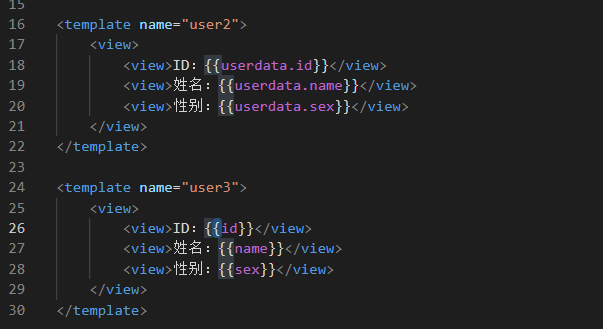
WXML提供模板（template），可以在模板中定义代码片段，然后在不同的地方调用。

使用 name 属性，作为模板的名字。然后在<template/>内定义代码片段，如：

调用 is 指定name的名称 data指定控制层传过来的数据 一定要用{{}}括起来









## 6.5、引用

简单理解：把模板定义到外部，然后大家共享，多个页面间，就可以共用。

<https://developers.weixin.qq.com/miniprogram/dev/framework/view/wxml/import.html>

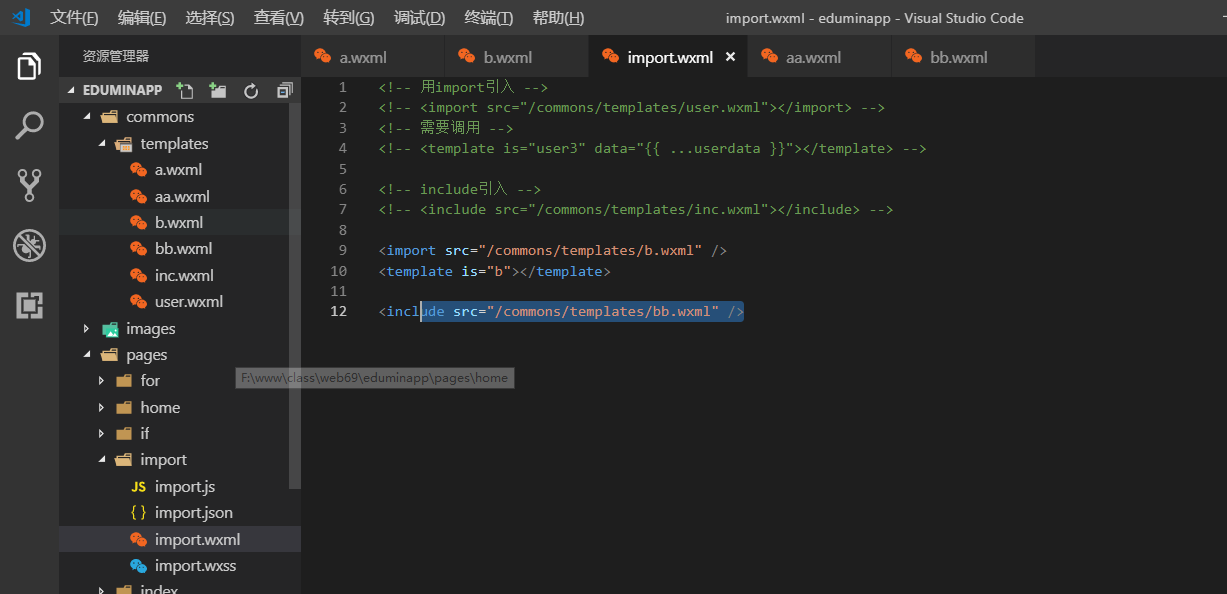
WXML 提供两种文件引用方式import和include





两者的区别

* import是引用过来，需要template调用，而include引入是直接把代码复制过来一份。
* import定义的时候是需要template标签，而include是不需要。
* import是不可以嵌套的，而include是可以嵌套的。



# 七、小程序的样式

## 7.1、概述

WXSS 用来决定 WXML 的组件应该怎么显示。说白了就是样式

为了适应广大的前端开发者，WXSS 具有 CSS 大部分特性。同时为了更适合开发微信小程序，WXSS 对 CSS 进行了扩充以及修改。

* 新增了尺寸单位

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。

* 提供了全局的样式和局部样式

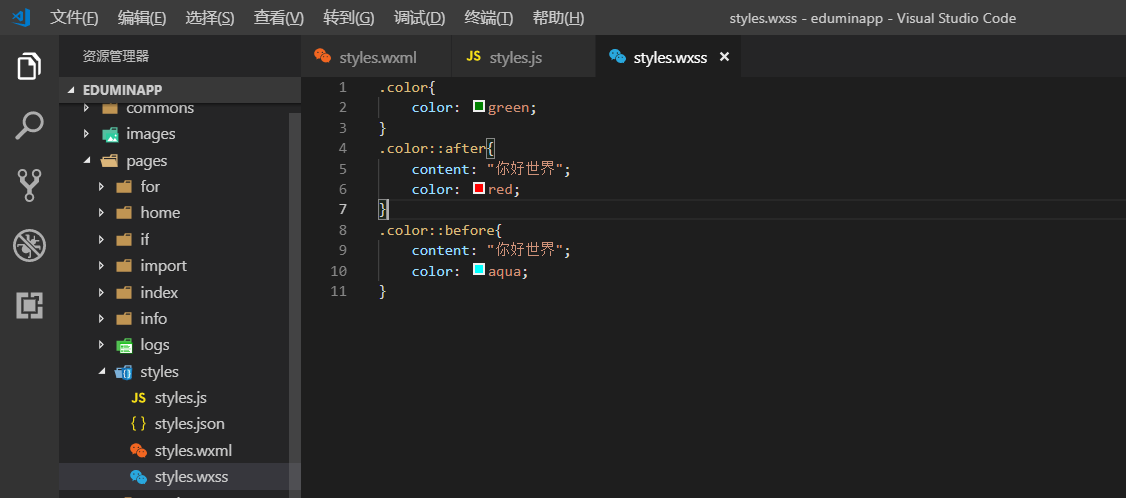
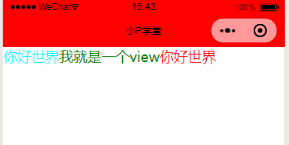
你可以写一个 app.wxss 作为全局样式，会作用于当前小程序的所有页面，局部页面样式 page.wxss 仅对当前页面生效。

* 此外 WXSS 仅支持部分 CSS 选择器

## 7.2、选择器

目前支持的选择器有：

|  |  |  |
| --- | --- | --- |
| 选择器 | 样例 | 样例描述 |
| .class | .intro | 选择所有拥有 class="intro" 的组件 |
| #id | #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 |

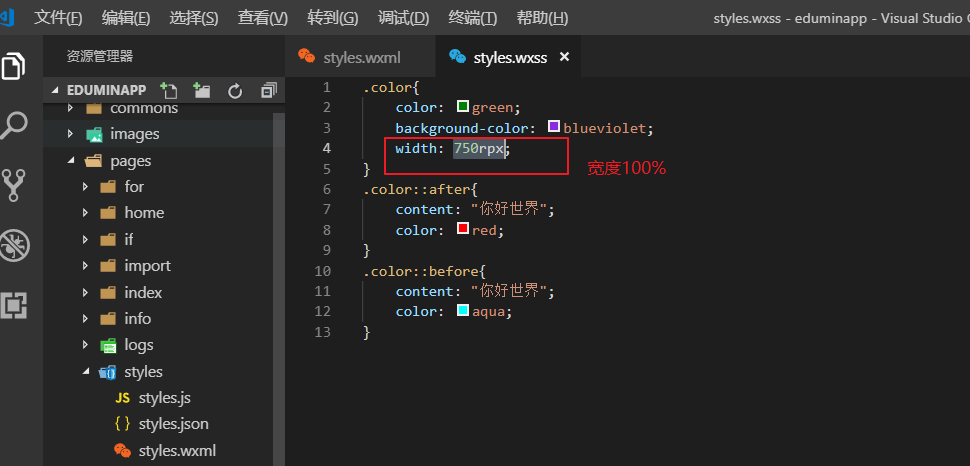
 

## 7.3、尺寸单位

rpx（responsive pixel）: 可以根据屏幕宽度进行自适应。规定屏幕宽为750rpx。

建议： 开发微信小程序时设计师可以**用 iPhone6 作为**视觉稿的标准。

在日后的小程序开发中，我们的统一的单位是 **rpx** 不要使用 px作为单位



## 7.4、全局样式与局部样式

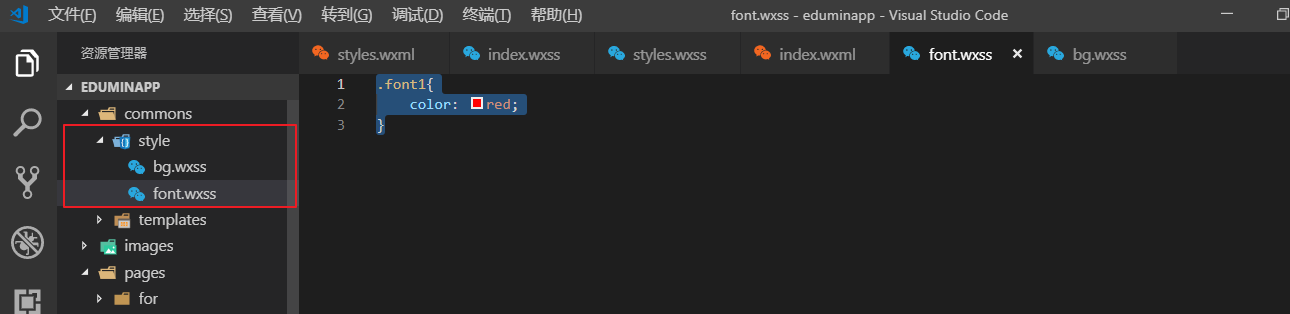
定义在 app.wxss 中的样式为全局样式，作用于每一个页面。

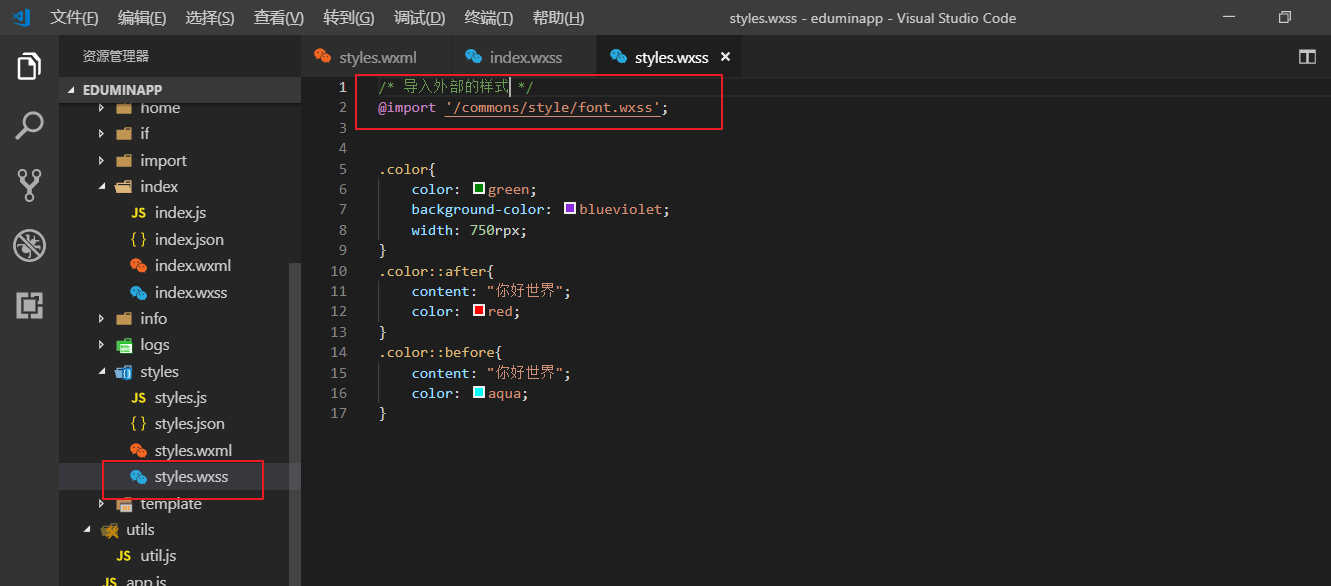
在 pages 的 wxss 文件中定义的样式为局部样式，只作用在对应的页面，并会覆盖 app.wxss 中相同的选择器。

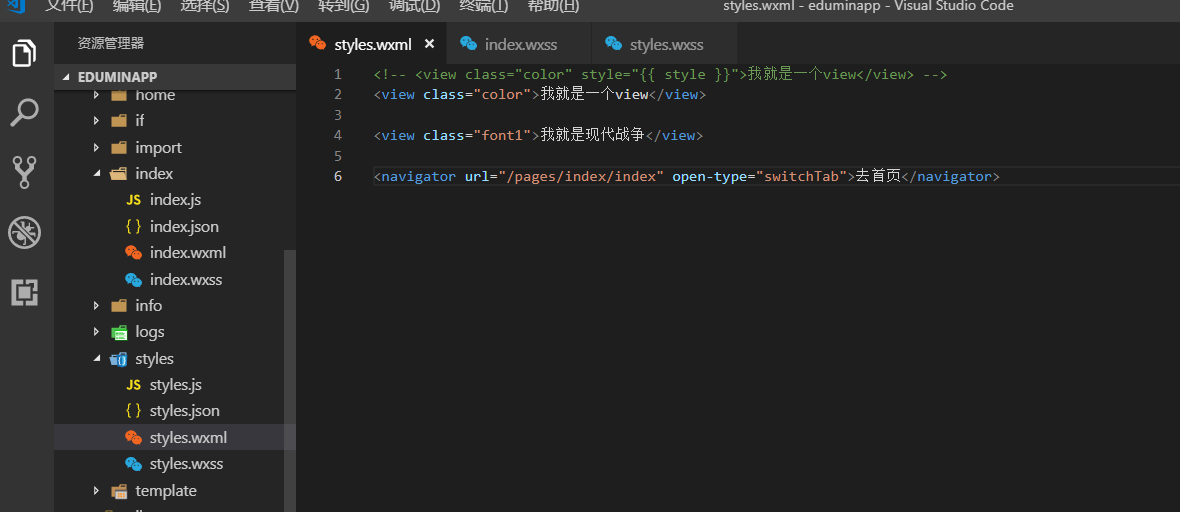
样式优先级 行内样式 > page.wxss > app.wxss

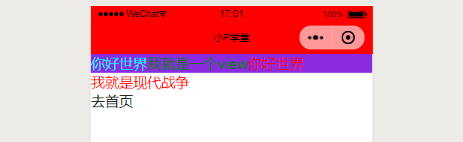
## 7.5、样式导入

使用 @import 语句可以导入外联样式表，@import 后跟需要导入的外联样式表的相对路径，用 ; 表示语句结束。









# 八、Flex布局

## 8.1、flex是什么

2009年，w3c提出一种新的方案，flex布局，可以简便，完整，响应式地实现各种页面的布局。

Flex是Flexible Box的缩写，意为”弹性布局”，用来为盒状模型提供最大的灵活性。

任何一个容器都可以指定为Flex布局，依赖样式属性 display。

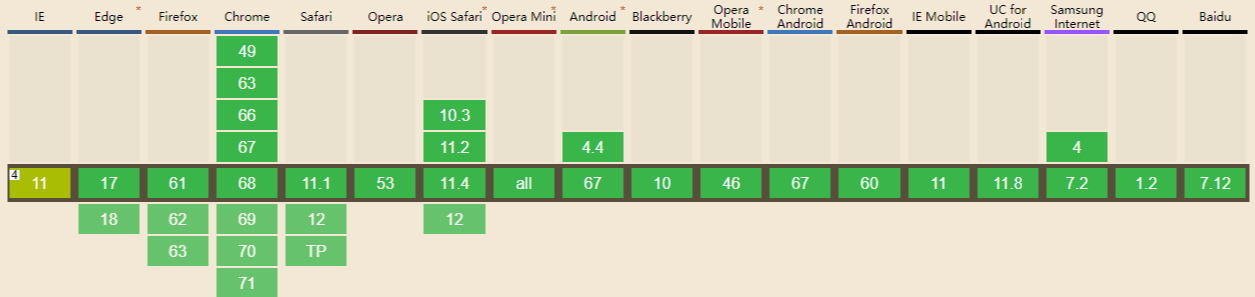
.box{

display: flex;

}

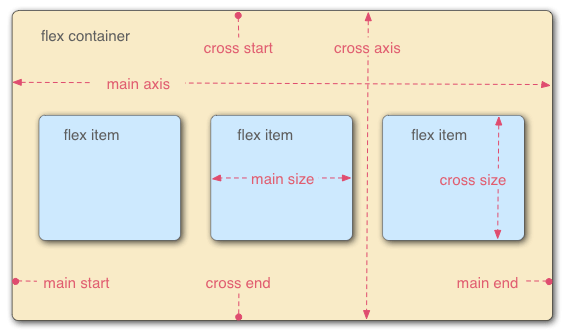
# 拥有此class盒子就有了flex布局功能

注：设为Flex布局以后，子元素的float、clear和vertical-align属性将**失效**。



## 8.2、基本概念

采用Flex布局的元素，称为Flex容器（flex container），简称”容器”。它的所有子元素自动成为容器成员，称为Flex项目（flex item），简称”项目”。



容器默认存在两根轴：水平的主轴（main axis）和垂直的交叉轴（cross axis）。

## 8.3、容器的属性

### 8.3.1、flex-direction属性

flex-direction属性决定主轴的方向（即项目的排列方向）。

.box {

flex-direction: row | row-reverse | column | column-reverse;

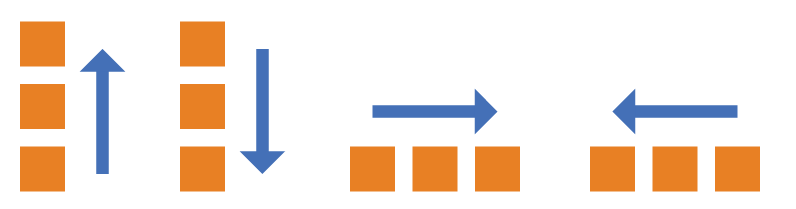
}

row（默认值）：主轴为水平方向，起点在左端。

row-reverse：主轴为水平方向，起点在右端。

column：主轴为垂直方向，起点在上沿。

column-reverse：主轴为垂直方向，起点在下沿。



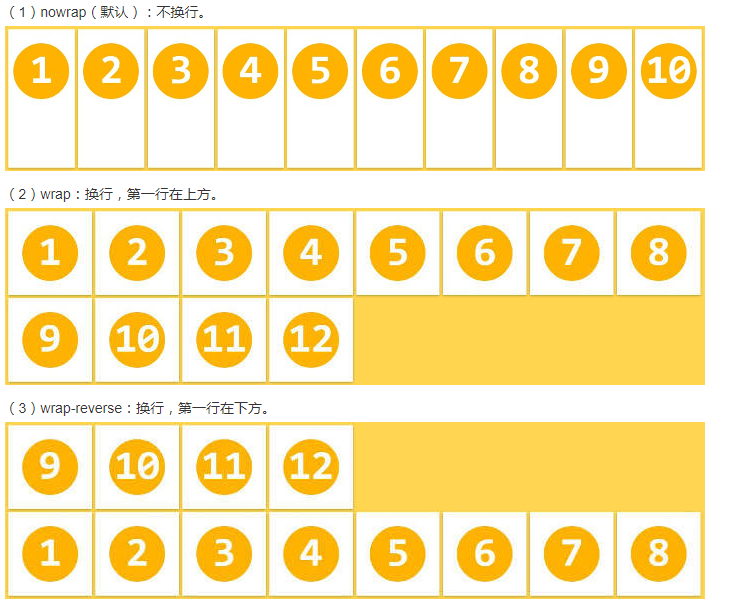
### 8.3.2、flex-wrap属性

flex-wrap属性定义，如果一条轴线排不下，如何换行。

.box{

flex-wrap: nowrap | wrap | wrap-reverse;

}



### 8.3.3、justify-content属性

justify-content属性定义了项目在主轴上的对齐方式。

.box {

justify-content: flex-start | flex-end | center | space-between | space-around;

}

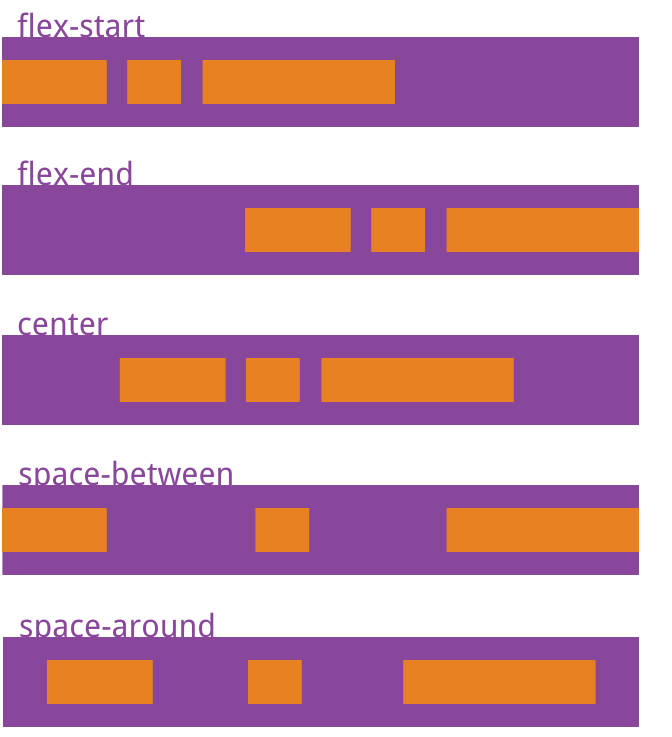
flex-start（默认值）：左对齐 顶部

flex-end：右对齐 底部

center： 居中

space-between：两端对齐，项目之间的间隔都相等。

space-around：每个项目两侧的间隔相等。所以，项目之间的间隔比项目与边框的间隔大一倍。



### 8.3.4、align-items属性

align-items属性定义项目在交叉轴上如何对齐。

.box {

align-items: flex-start | flex-end | center;

}

flex-start：交叉轴的起点对齐。

flex-end：交叉轴的终点对齐。

center：交叉轴的中点对齐。

