# 一、小程序框架介绍（了解）

小程序框架包含小程序的配置、框架接口、场景值、WXML 和 WXS 等

# 二、小程序的配置（精通）

小程序的配置分为全局配置、页面配置以及sitemap 配置

## 1、全局配置

小程序根目录下的 app.json 文件用来对微信小程序进行全局配置。文件内容为一个 JSON 对象

| 属性 | 类型 | 必填 | 描述 | 最低版本 |
| --- | --- | --- | --- | --- |
| pages | string[] | 是 | 页面路径列表 | |
| window | Object | 否 | 全局的默认窗口表现 | |
| tabBar | Object | 否 | 底部 `tab` 栏的表现 | |
| networkTimeout | Object | 否 | 网络超时时间 | |
| debug | boolean | 否 | 是否开启 debug 模式，默认关闭 | |
| functionalPages | boolean | 否 | 是否启用插件功能页，默认关闭 | 2.1.0 |
| subpackages | Object[] | 否 | 分包结构配置 | 1.7.3 |
| workers | string | 否 | `Worker` 代码放置的目录 | 1.9.90 |
| requiredBackgroundModes | string[] | 否 | 需要在后台使用的能力，如「音乐播放」 | |
| plugins | Object | 否 | 使用到的插件 | 1.9.6 |
| preloadRule | Object | 否 | 分包预下载规则 | 2.3.0 |
| resizable | boolean | 否 | iPad 小程序是否支持屏幕旋转，默认关闭 | 2.3.0 |
| navigateToMiniProgramAppIdList | string[] | 否 | 需要跳转的小程序列表，详见 wx.navigateToMiniProgram | 2.4.0 |
| usingComponents | Object | 否 | 全局自定义组件配置 | 开发者工具 1.0 |

| | | | | |
|---|---|---|---|---|
| | ct | | | 2.1810190 |
| permission | Object | 否 | 小程序接口权限相关设置 | 微信客户端 7.0.0 |
| sitemapLocation | string | 是 | 指明 sitemap.json 的位置 | |
| style | string | 否 | 指定使用升级后的weui样式 | 2.8.0 |
| useExtendedLib | Object | 否 | 指定需要引用的扩展库 | 2.2.1 |
| entranceDeclare | Object | 否 | 微信消息用小程序打开 | 微信客户端7.0.9 |

下面介绍一下常用的配置选项

## 1.1 pages

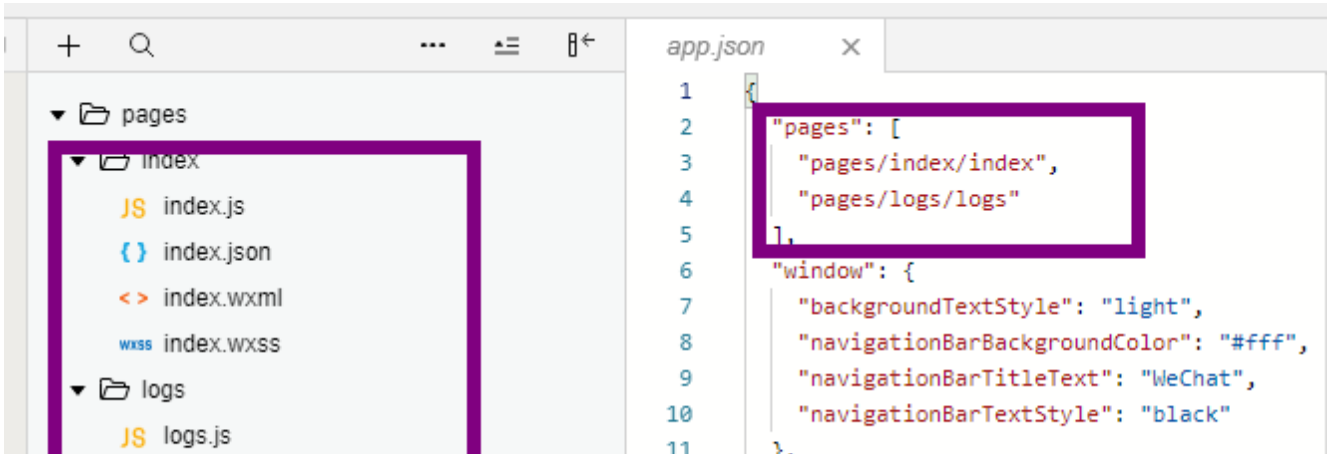
用于指定小程序由哪些页面组成，每一项都对应一个页面的 路径（含文件名） 信息。文件名不需要写文件后缀，框架会自动去寻找对于位置的 .json, .js, .wxml, .wxss 四个文件进行处理

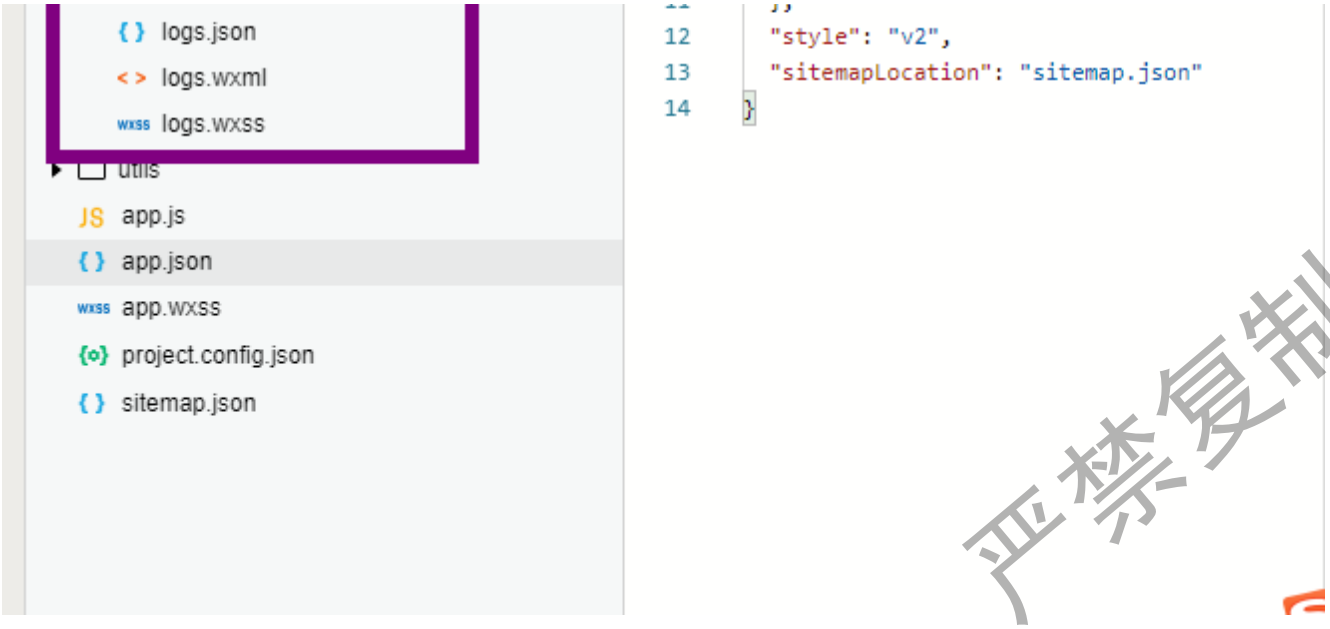
> *有多少个页面，此处就应该有多少个选项*
>
> *数组的第一项代表小程序的初始页面（首页）。小程序中新增/减少页面，都需要对 pages 数组进行修改。*

==开发小技巧==

> *直接在pages选项中写页面路径，即可创建相应的页面*

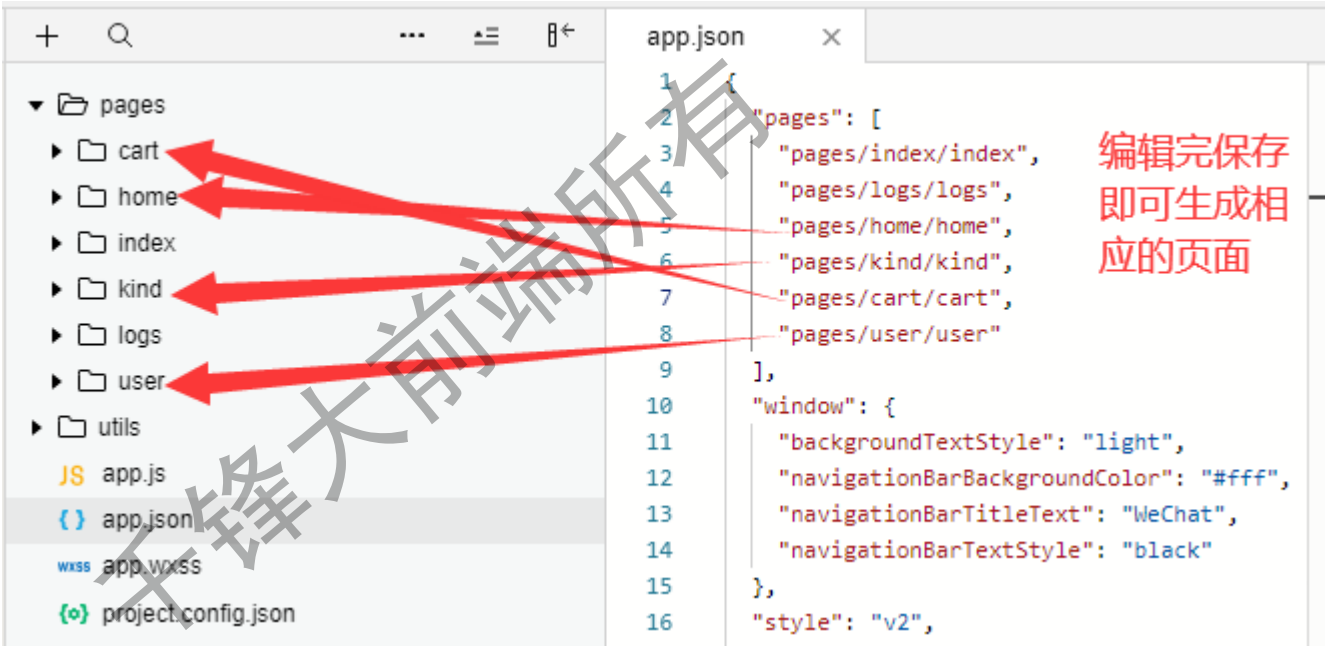
默认项目如图所示








创建首页、分类、购物车、我的页面，编辑app.json中的pages选项如下





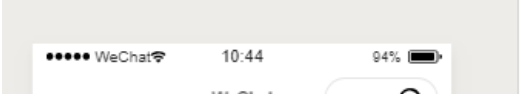

## 1.2 window

用于设置小程序的状态栏、导航条、标题、窗口背景色。

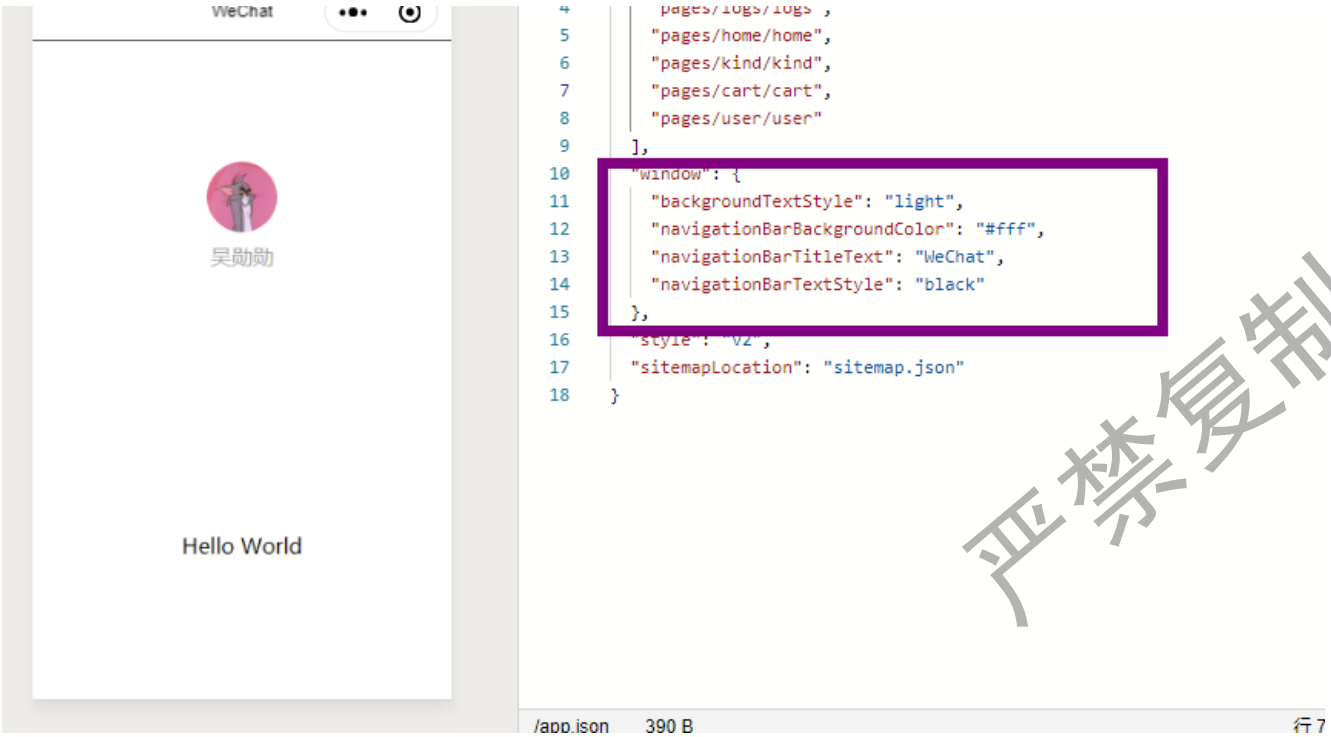
| 属性 | 类型 | 默认值 | 描述 | 最低版本 |
| --- | --- | --- | --- | --- |
| navigationBarBa | Hex | #00 | | |

| | | | | |
|---|---|---|---|---|
| ckgroundColor | Color | 0000 | 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，仅支持以下值：<br>`default` 默认样式<br>`custom` 自定义导航栏，只保留右上角胶囊按钮。参见注 2。 | 微信客户端 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 | 微信客户端 6.5.16 |
| backgroundColorBottom | string | #ffffff | 底部窗口的背景色，仅 iOS 支持 | 微信客户端 6.5.16 |
| enablePullDownRefresh | boolean | false | 是否开启全局的下拉刷新。<br>详见 Page.onPullDownRefresh | |
| 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。<br>详见 Page.onReachBottom | |
| pageOrientation | string | portrait | 屏幕旋转设置，支持 `auto` / `portrait` / `landscape`<br>详见 响应显示区域变化 | 2.4.0 (auto) / 2.5.0 (landscape) |

默认小程序展示如下

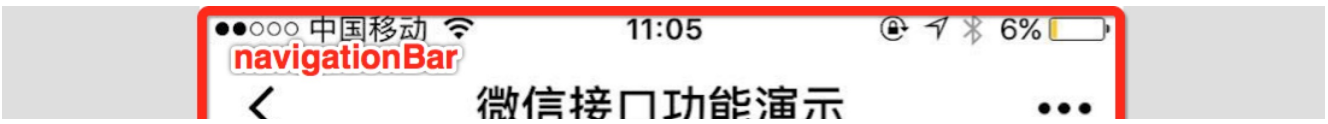



```
{
  "pages": [
    "pages/index/index",
```




修改之后展示如下


各种属性展示示意图如下






### 1.3 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 | 类型 | 必 | 默认 | 描述 | 最低 |
| --- | --- | --- | --- | --- | --- |

| 属性 | 类型 | 必填 | 默认值 | 描述 | 版本 |
| --- | --- | --- | --- | --- | --- |
| color | HexColor | 是 | | tab 上的文字默认颜色，仅支持十六进制颜色 | |
| selectedColor | HexColor | 是 | | tab 上的文字选中时的颜色，仅支持十六进制颜色 | |
| backgroundColor | HexColor | 是 | | tab 的背景色，仅支持十六进制颜色 | |
| 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 | tabBar 的位置，仅支持 `bottom` / `top` | |
| custom | boolean | 否 | false | 自定义 tabBar，见详情 | 2.5.0 |

其中 list 接受一个数组，只能配置**最少 2 个、最多 5 个 tab**。tab 按数组的顺序排序，每个项都是一个对象，其属性值如下：

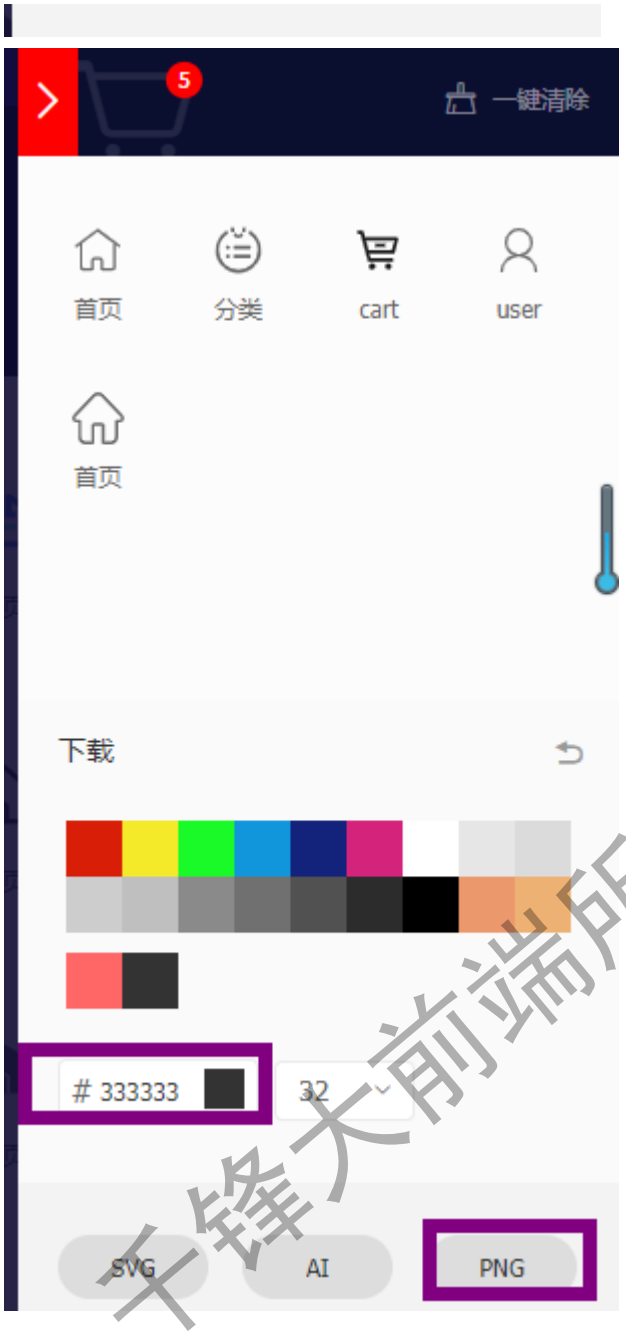
| 属性 | 类型 | 必填 | 说明 |
| --- | --- | --- | --- |
| pagePath | string | 是 | 页面路径，必须在 pages 中先定义 |
| text | string | 是 | tab 上按钮文字 |
| iconPath | string | 否 | 图片路径，icon 大小限制为 40kb，建议尺寸为 81px *81px，不支持网络图片。*<br>*当 `position` 为 `top` 时，不显示 icon。* |
| selectedIconPath | string | 否 | 选中时的图片路径，icon 大小限制为 40kb，建议尺寸为 81px 81px，不支持网络图片。<br>当 `position` 为 `top` 时，不显示 icon。 |

其展示形式如下图所示


#### 1.3.1 准备底部选项卡

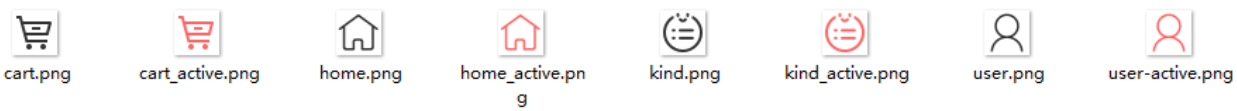
1. icon字体图标

在 iconfont字体图标库 中选择需要的图标，然后选择下载

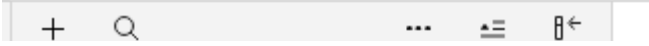



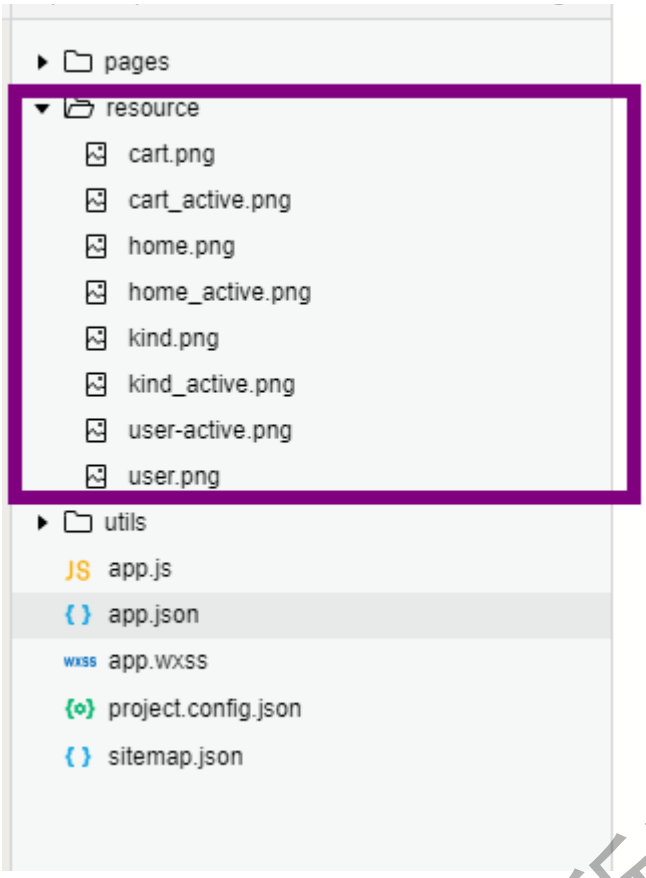



修改相应的图片名字如下




2. 项目目录下创建resource文件夹，将图片放置进去






3. app. json中配置底部选项卡

app.json

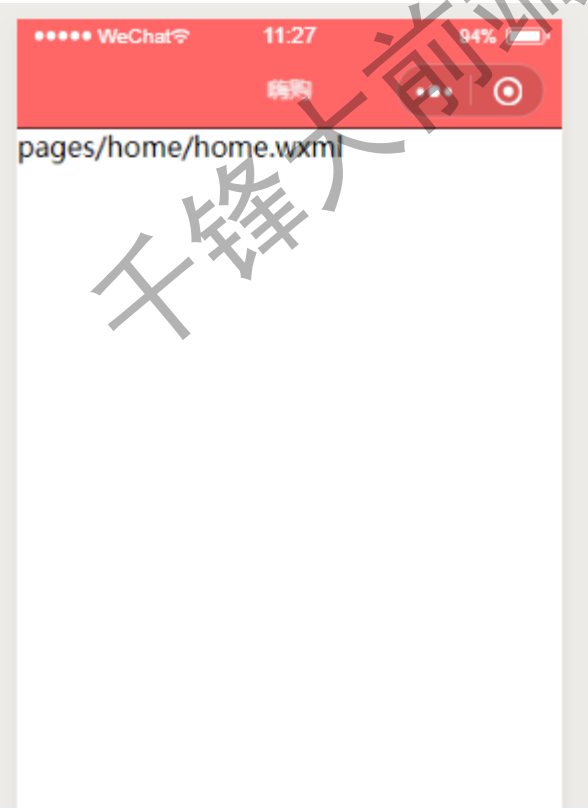
```
"tabBar": {
  "color": "#333",
  "selectedColor": "#f66",
  "backgroundColor": "#efefef",
  "borderStyle": "white",
  "list": [
    {
      "text": "首页",
      "pagePath": "pages/home/home",
      "iconPath": "/resource/home.png",
      "selectedIconPath": "/resource/home_active.png"
    },
    {
      "text": "分类",
      "pagePath": "pages/kind/kind",
      "iconPath": "/resource/kind.png",
      "selectedIconPath": "/resource/home_active.png"
    },
    {
      "text": "购物车",
      "pagePath": "pages/cart/cart",
```

```
      "iconPath": "/resource/cart.png",
      "selectedIconPath": "/resource/home_active.png"
    },
    {
      "text": "我的",
      "pagePath": "pages/user/user",
      "iconPath": "/resource/user.png",
      "selectedIconPath": "/resource/home_active.png"
    }
  ]
},
```

调整首页为第一个页面 —— 初始化页面

```
{
  "pages": [
    "pages/home/home",
    "pages/index/index",
    "pages/logs/logs",
    "pages/kind/kind",
    "pages/cart/cart",
    "pages/user/user"
  ],
  "window": {
```

展示形式如下




### 1.4 networkTimeout

各类网络请求的超时时间，单位均为毫秒。

| 属性 | 类型 | 必填 | 默认值 | 说明 |
| --- | --- | --- | --- | --- |
| request | number | 否 | 60000 | wx.request 的超时时间，单位：毫秒。 |
| connectSocket | number | 否 | 60000 | wx.connectSocket 的超时时间，单位：毫秒。 |
| uploadFile | number | 否 | 60000 | wx.uploadFile 的超时时间，单位：毫秒。 |
| downloadFile | number | 否 | 60000 | wx.downloadFile 的超时时间，单位：毫秒。 |

## 1.5 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，事件触发等。可以帮助开发者快速定位一些常见的问题

## 1.6 functionalPages

插件所有者小程序需要设置这一项来启用插件功能页

## 1.7 permission

小程序接口权限相关设置。字段类型为 Object，结构为：

| 属性 | 类型 | 必填 | 默认值 | 描述 |
| --- | --- | --- | --- | --- |
| scope.userLocation | PermissionObject | 否 | | 位置相关权限声明 |

PermissionObject 结构为

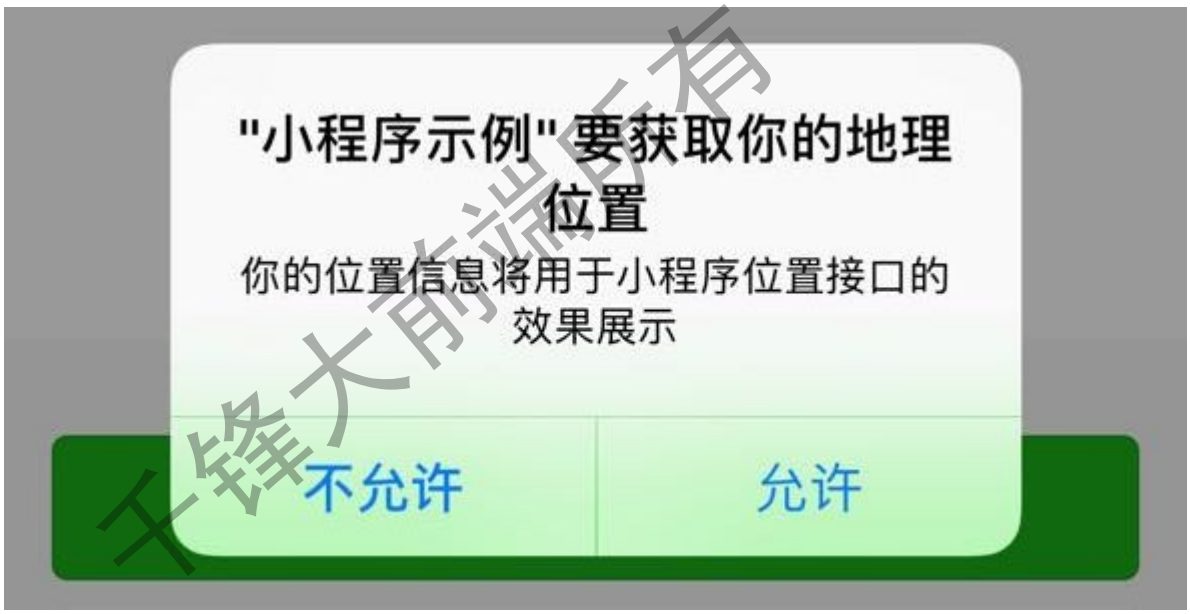
| 属性 | 类型 | 必填 | 默认值 | 说明 |
| --- | --- | --- | --- | --- |
| desc | string | 是 | | 小程序获取权限时展示的接口用途说明。最长 30 个字符 |

代码如下

app.json

```
43          "iconPath": "/resource/user.png",
44          "selectedIconPath": "/resource/home_active.png"
45        }
46      ]
47    },
48    "permission": {
49      "scope.userLocation": {
50        "desc": "你的位置信息将用于小程序位置接口的效果展示"
51      }
52    },
53    "style": "v2",
54    "sitemapLocation": "sitemap.json"
55  }
```

展示效果如下






### 1.8 sitemapLocation

指明 sitemap.json 的位置；默认为 ‘sitemap.json’ 即在 app.json 同级目录下名字的 sitemap.json 文件

app.json

```
43          "iconPath": "/resource/user.png",
44          "selectedIconPath": "/resource/home_active.png"
45        }
46      ]
47    },
```

```
48  "permission": {
49    "scope.userLocation": {
50      "desc": "你的位置信息将用于小程序位置接口的效果展示"
51    }
52  },
53  "style": "v2",
54  "sitemapLocation": "sitemap.json"
55  }
```

### 1.9 navigateToMiniProgramAppIdList

当小程序需要使用 wx.navigateToMiniProgram 接口跳转到其他小程序时，需要先在配置文件中声明需要跳转的小程序 appId 列表，最多允许填写 10 个。

app.json　sitemap.json

```
43          "iconPath": "/resource/user.png",
44          "selectedIconPath": "/resource/home_active.png"
45        }
46      ]
47    },
48    "permission": {
49      "scope.userLocation": {
50        "desc": "你的位置信息将用于小程序位置接口的效果展示"
51      }
52    },
53    "navigateToMiniProgramAppIdList": [
54      "wxe5f52902cf4de896"
55    ],
56    "style": "v2",
57    "sitemapLocation": "sitemap.json"
58  }
```



## 2、页面配置

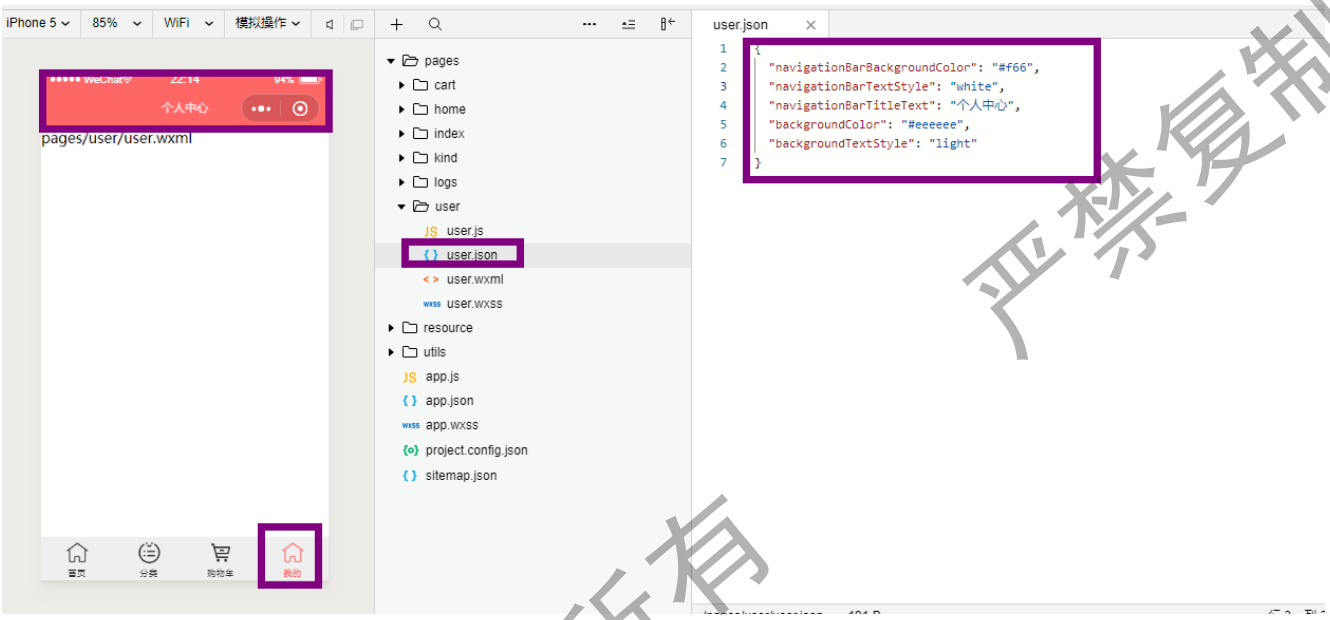
每一个小程序页面也可以使用 .json 文件来对本页面的窗口表现进行配置。页面中配置项在当前页面会覆盖 app.json 的 window 中相同的配置项。文件内容为一个 JSON 对象，有以下属性

| 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，仅支持以下值：<br>`default` 默认样式<br>`custom` 自定义导航栏，只保留右上角胶囊按钮 | 微信客户端 7.0.0 |
| 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 | 微信客户端 6.5.16 |
| backgroundColorBottom | string | #ffffff | 底部窗口的背景色，仅 iOS 支持 | 微信客户端 6.5.16 |
| enablePullDownRefresh | boolean | false | 是否开启当前页面下拉刷新。<br>详见 Page.onPullDownRefresh | |
| 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。<br>详见 Page.onReachBottom | |
| pageOrientation | string | portrait | 屏幕旋转设置，支持 `auto` / `portrait` / `landscape`<br>详见 响应显示区域变化 | 2.4.0 (auto) / 2.5.0 (landscape) |
| disableScroll | boolean | false | 设置为 `true` 则页面整体不能上下滚动。<br>只在页面配置中有效，无法在 app.j | |

| | | | | |
|---|---|---|---|---|
| | n | | son 中设置 | |
| usingComponents | Object | 否 | 页面自定义组件配置 | 1.6.3 |

个人中心页面配置




## 3、sitemap 配置

小程序根目录下的 sitemap.json 文件用于配置小程序及其页面是否允许被微信索引，文件内容为一个 JSON 对象，如果没有 sitemap.json ，则默认为所有页面都允许被索引。

# 三、框架接口（精通）

## 1.App(Object object)

注册小程序。接受一个 Object 参数，其指定小程序的生命周期回调等。

App() 必须在 app.js中调用，必须调用且只能调用一次。不然会出现无法预期的后果。

| 属性 | 类型 | 默认值 | 必填 | 说明 | 最低版本 |
|---|---|---|---|---|---|
| | fun | | | | |

| | | | | | |
|---|---|---|---|---|---|
| onLaunch | function | | 否 | 生命周期回调——监听小程序初始化。 | |
| onShow | function | | 否 | 生命周期回调——监听小程序启动或切前台。 | |
| onHide | function | | 否 | 生命周期回调——监听小程序切后台。 | |
| onError | function | | 否 | 错误监听函数。 | |
| onPageNotFound | function | | 否 | 页面不存在监听函数。 | 1.9.90 |
| onUnhandledRejection | function | | 否 | 未处理的 Promise 拒绝事件监听函数。 | 2.10.0 |
| 其他 | any | | 否 | 开发者可以添加任意的函数或数据变量到 `Object` 参数中，用 `this` 可以访问 | |

示例中app.js代码如下

```
//app.js
App({
 onLaunch: function () {
 // 生命周期回调——监听小程序初始化 - 全局只触发一次
 // 展示本地存储能力
 var logs = wx.getStorageSync('logs') || []
 logs.unshift(Date.now())
 wx.setStorageSync('logs', logs)

 // 登录
 wx.login({
 success: res => {
 // 发送 res.code 到后台换取 openId, sessionKey, unionId
 }
 })
```

```
16.   // 获取用户信息
17.   wx.getSetting({
18.   success: res => {
19.   if (res.authSetting['scope.userInfo']) {
20.   // 已经授权，可以直接调用 getUserInfo 获取头像昵称，不会弹框
21.   wx.getUserInfo({
22.   success: res => {
23.   // 可以将 res 发送给后台解码出 unionId
24.   this.globalData.userInfo = res.userInfo
25.
26.   // 由于 getUserInfo 是网络请求，可能会在 Page.onLoad 之后才返回
27.   // 所以此处加入 callback 以防止这种情况
28.   if (this.userInfoReadyCallback) {
29.   this.userInfoReadyCallback(res)
30.   }
31.   }
32.   })
33.   }
34.   }
35.   })
36.   },
37.   onShow(options) {
38.   // 小程序启动，或从后台进入前台显示时触发
39.   },
40.   onHide() {
41.   // 小程序从前台进入后台时触发.
42.   },
43.   onError(msg) {
44.   // 小程序发生脚本错误或 API 调用报错时触发
45.   console.log(msg)
46.   },
47.   onPageNotFound(res) {
48.   // 小程序要打开的页面不存在时触发;如果是 tabbar 页面，请使用 wx.switchTab - 404
49.   },
50.   globalData: {
51.   userInfo: null
52.   }
53.  })
```

## 2. getApp(Object object)

获取到小程序全局唯一的 App 实例,在页面的js文件中获取

## 3. Page(Object object)

注册小程序中的一个页面。接受一个Object类型参数，其指定页面的初始数据、生命周期回调、事件处理函数等。

| 属性 | 类型 | 默认值 | 必填 | 说明 |
| --- | --- | --- | --- | --- |
| data | Object | | | 页面的初始数据 |
| onLoad | function | | | 生命周期回调—监听页面加载 |
| onShow | function | | | 生命周期回调—监听页面显示 |
| onReady | function | | | 生命周期回调—监听页面初次渲染完成 |
| onHide | function | | | 生命周期回调—监听页面隐藏 |
| onUnload | function | | | 生命周期回调—监听页面卸载 |
| onPullDownRefresh | function | | | 监听用户下拉动作 |
| onReachBottom | function | | | 页面上拉触底事件的处理函数 |
| onShareAppMessage | function | | | 用户点击右上角转发 |
| onPageScroll | function | | | 页面滚动触发事件的处理函数 |
| onResize | function | | | 页面尺寸改变时触发，详见 响应显示区域变化 |

| | | | | |
|---|---|---|---|---|
| onTabItemTap | function | | | 当前是 tab 页时，点击 tab 时触发 |
| 其他 | any | | | 开发者可以添加任意的函数或数据到 `Object` 参数中，在页面的函数中用 `this` 可以访问 |

以个人中心的js为例

```
// pages/user/user.js
Page({

  /**
   * 页面的初始数据
   * data 是页面第一次渲染使用的初始数据。
   * 页面加载时，data 将会以JSON字符串的形式由逻辑层传至渲染层，因此data中的数据必须是可以转成JSON的类型：字符串，数字，布尔值，对象，数组。
   * 渲染层可以通过 WXML 对数据进行绑定。
   */
  data: {

  },

  /**
   * 生命周期函数--监听页面加载
   * 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。
   */
  onLoad: function (options) {
  // options为打开当前页面路径中的参数
  },

  /**
   * 生命周期函数--监听页面初次渲染完成
   * 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
   */
  onReady: function () {

  },

  /**
   * 生命周期函数--监听页面显示
```

```
32.   * 页面显示/切入前台时触发
33.   */
34.   onShow: function () {
35.
36.   },
37.
38.   /**
39.   * 生命周期函数--监听页面隐藏
40.   * 页面隐藏/切入后台时触发
41.   */
42.   onHide: function () {
43.
44.   },
45.
46.   /**
47.   * 生命周期函数--监听页面卸载
48.   * 页面卸载时触发。
49.   */
50.   onUnload: function () {
51.
52.   },
53.
54.   /**
55.   * 页面相关事件处理函数--监听用户下拉动作
56.   * 需要在app.json的window选项中或页面配置中开启enablePullDownRefresh。
57.   * 可以通过wx.startPullDownRefresh触发下拉刷新，调用后触发下拉刷新动画，效果与用户手动下
拉刷新一致。
58.   * 当处理完数据刷新后，wx.stopPullDownRefresh可以停止当前页面的下拉刷新
59.   */
60.   onPullDownRefresh: function () {
61.
62.   },
63.
64.   /**
65.   * 页面上拉触底事件的处理函数
66.   * 可以在app.json的window选项中或页面配置中设置触发距离onReachBottomDistance。
67.   * 在触发距离内滑动期间，本事件只会被触发一次
68.   */
69.   onReachBottom: function () {
70.
71.   }
```

```
  },

  /**
  * 用户点击右上角分享
  */
  onShareAppMessage: function (res) {
  if (res.from === 'button') {
  // 来自页面内转发按钮
  console.log(res.target)
  }
  // 自定义图片路径，可以是本地文件路径、代码包文件路径或者网络图片路径。支持PNG及JPG。显示图片长宽比是 5:4。
  return {
  title: '自定义转发标题',
  path: '/page/user?id=123',
  imageUrl: ''
  }
  },

  /**
  * 监听用户滑动页面事件
  */
  onPageScroll: function () {

  }
  /**
  * 自定义函数
  */
})
```

## 4、getCurrentPages()

获取当前页面栈。数组中第一个元素为首页，最后一个元素为当前页面。

> *不要尝试修改页面栈，会导致路由以及页面状态错误。*
>
> *不要在 App.onLaunch 的时候调用 getCurrentPages()，此时 page 还没有生成。*

## 5、自定义组件

创建自定义组件，接受一个 Object 类型的参数

比如在首页里需要一个产品列表的组件，可以自定义该组件


小技巧 点击 “+”选择目录，输入components

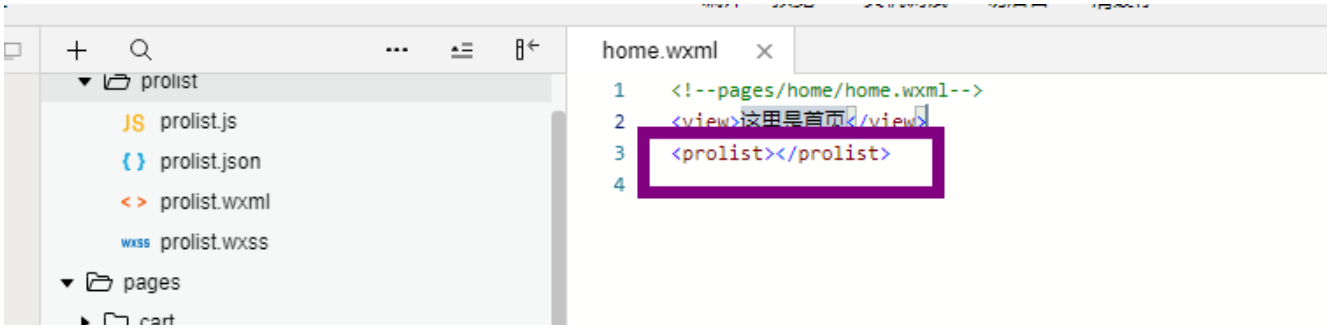
右键点击components目录，选择目录， 输入prolist

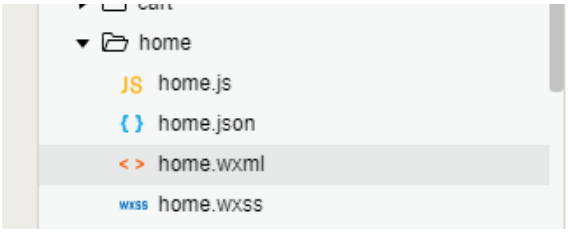
右键点击prolist目录，选择 新建Component ，输入prolist 即可

如何使用该组件呢？

在首页的pages/home/home.json文件中注册组件


在首页的pages/home/home.wxml中使用该组件，就像正常的标签一样使用






组件之间的传值在后续课程中会讲解

# 6、模块化

建议使用es6的模块化方法，api中提供的是基于commonjs规范的exports以及require语法

## 6.1 定义工具模块 utils/index.js

数据请求模块 以及 可消失的提示框 模块 - 暴露

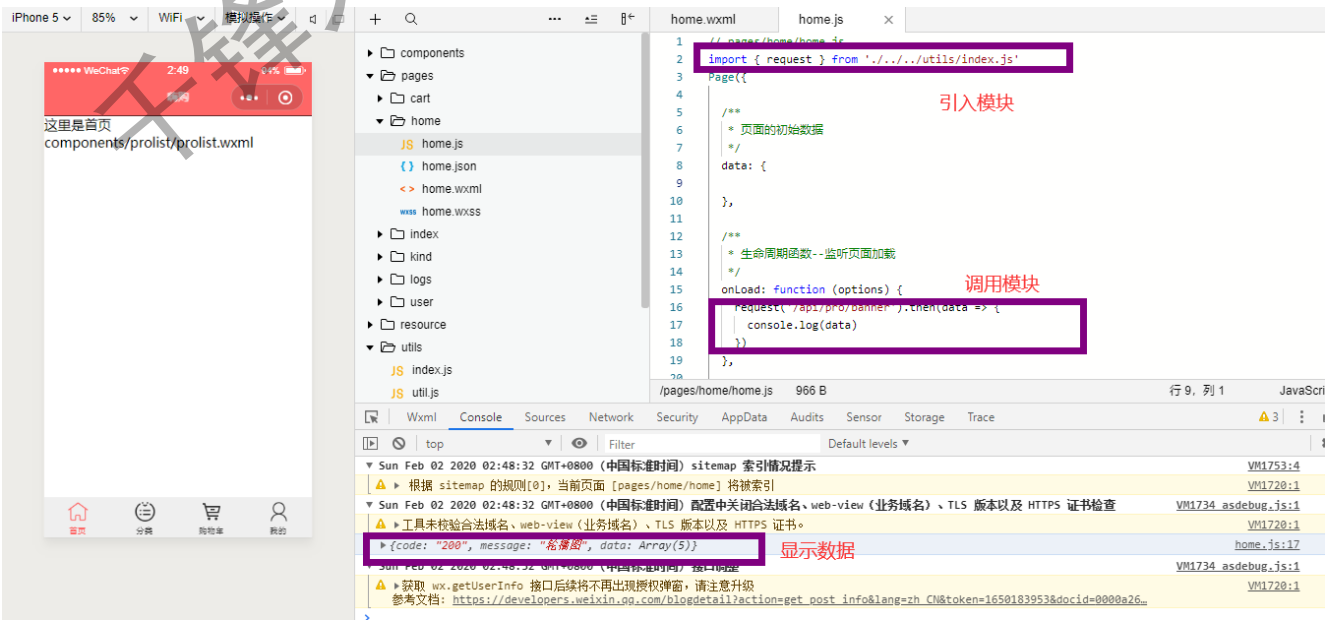
```
1.  const baseUrl = 'http://daxun.kuboy.top'
2.  /**
3.   * 数据请求模块
4.   * 接口地址 http://daxun.kuboy.top/apidoc
5.   * 先显示加载框，然后请求结束加载框消失
6.   *
7.   */
8.  export function request (url, data) {
9.   // 显示加载中
10.  // 参考 https://developers.weixin.qq.com/miniprogram/dev/api/ui/interaction/wx.showLoading.html
11.  wx.showLoading({
12.  title: '加载中',
13.  })
14.  // 使用promise 解决异步操作问题，此处还可以使用 async + await
15.  return new Promise((resolve, reject) => {
16.  // 微信小程序的数据请求方法
17.  // 必须配置小程序的安全域名，
18.  // 在开发阶段可以在“详情” - “本地设置” - 勾选中 不校验请求域名、web-view(业务域名)、TLS版本及HTTPS证书
19.  wx.request({
20.  url: baseUrl + url,
21.  data: data || {},
22.  success: (res) => {
```

```
  // 隐藏加载中
  wx.hideLoading();
  // 后续处理
  resolve(res.data)
  }
  })
  })
}

/**
  * 可消失的提示框 - 默认只显示文字
  * str 提示内容
  * icon 是否需要图标，none 、 success(默认值) 、 loading
  */
 export function Toast (str, icon) {
  // 微信提供的API接口
  // 参照
https://developers.weixin.qq.com/miniprogram/dev/api/ui/interaction/wx.showToast.html
  wx.showToast({
  title: str,
  icon: icon || 'none'
  })
}
```

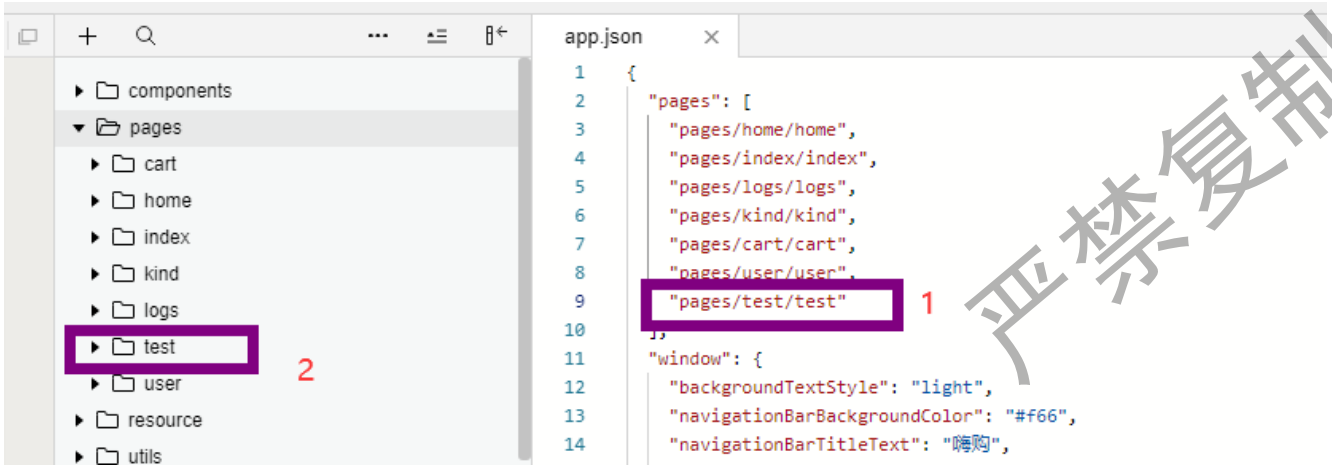
## 6.2 首页中测试

在首页 pages/home/home.js中测试，先引入模块

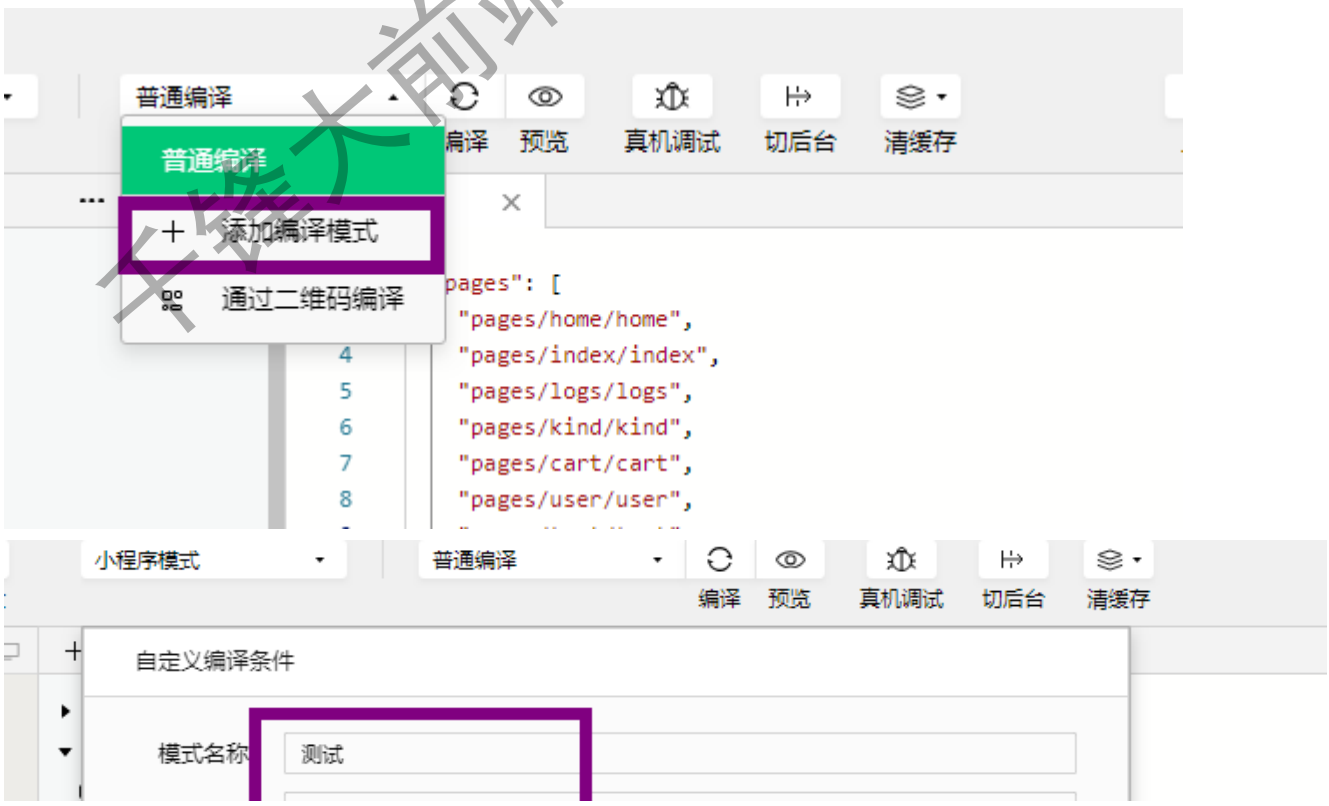


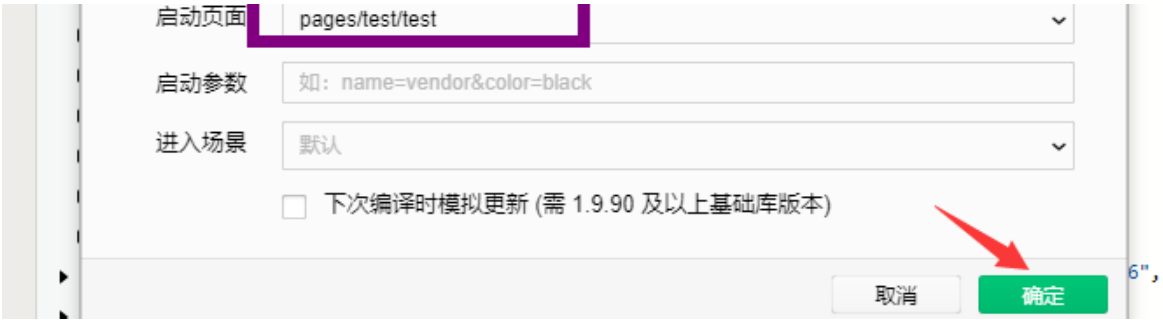
# 四、WXML语法参考（精通）

创建一个新的页面pages/test/test




选择工具栏的 小程序编译，添加编译模式，可以更快捷的只渲染本页面，便于开发者的调试






# 1. 数据绑定

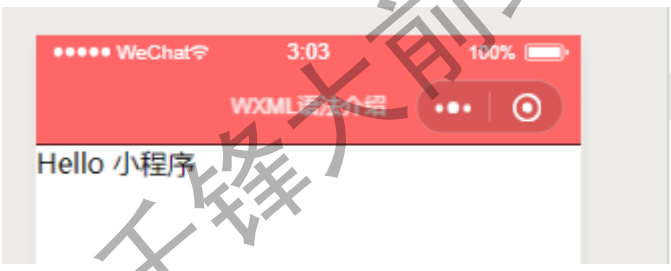
WXML 中的动态数据均来自对应 Page 的 data。

1. 简单绑定（类似于vue中的Mustache 语法）

数据绑定使用 Mustache 语法（双大括号）将变量包起来，可以作用于：

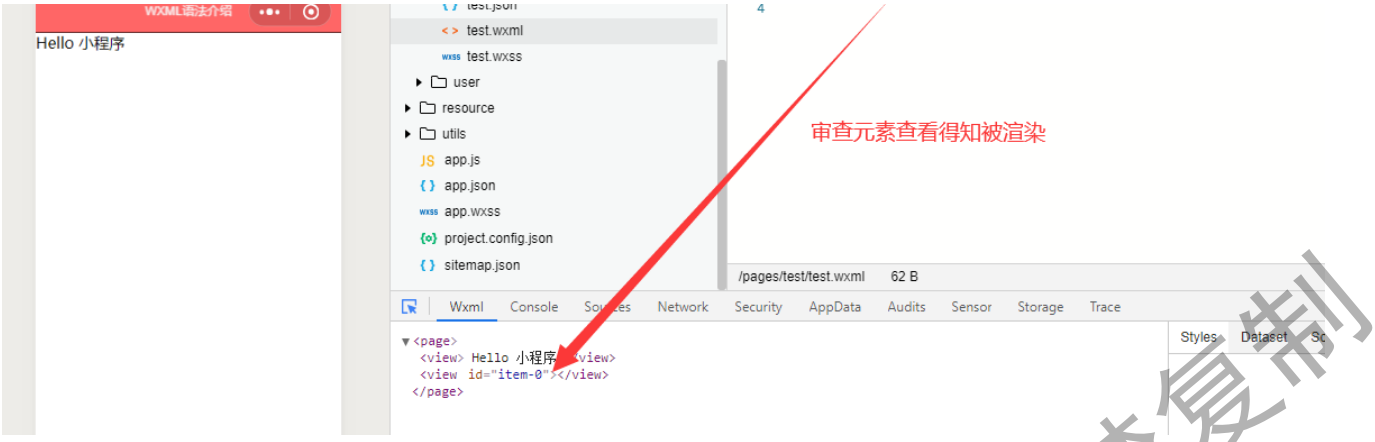
内容

```
<view> {{ message }} </view>
Page({
 data: {
 message: 'Hello MINA!'
 }
})
```



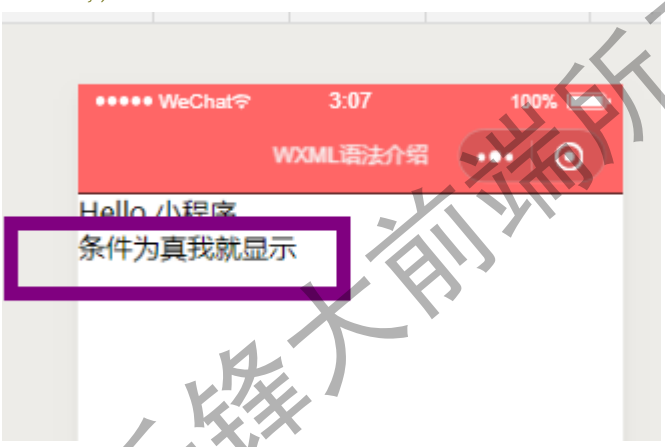

1. 组件属性(需要在双引号之内)

```
<view id="item-{{id}}"> </view>
Page({
 data: {
 id: 0
 }
})
```




3. 控制属性(需要在双引号之内)

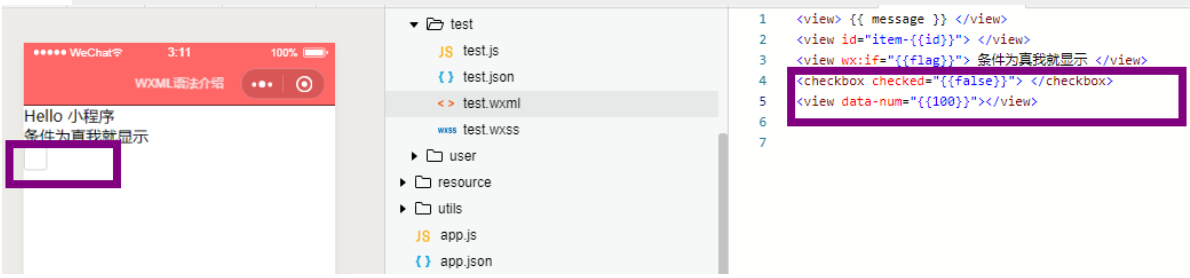
```
<view wx:if="{{flag}}"> 条件为真我就显示 </view>
Page({
 data: {
 flag: true
 }
})
```

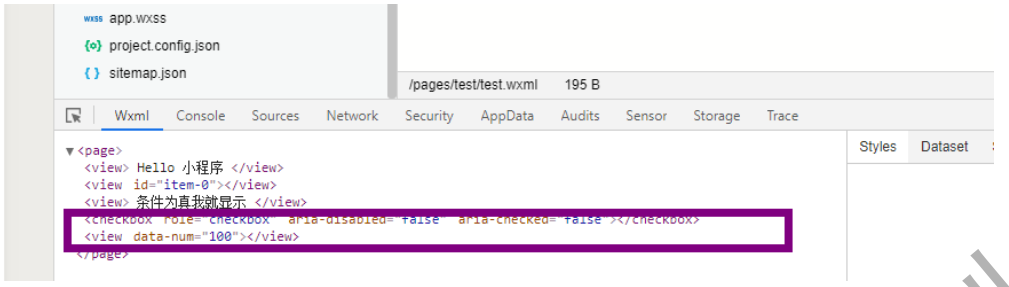



4. boolean以及number数据类型

如果数据类型是booblean 或者number类型的数据，需要使用{{}}包裹

```
<checkbox checked="{{false}}"> </checkbox>
<view data-num = "{{100}}"></view>
```






5. 表达式运算

可以在 {{}} 内进行简单的运算，支持三元运算、算数运算、逻辑判断、字符串运算等

```
<view test="{{flag ? true : false}}"> 属性 </view>

<view> {{a + b}} + {{c}} + d </view>

<view wx:if="{{len > 5}}"> </view>

<view>{{"hello" + name}}</view>
```

## 2. 列表渲染

wx:for（vue中使用v-for）

在组件上使用 wx:for 控制属性绑定一个数组，即可使用数组中各项的数据重复渲染该组件。

默认数组的当前项的下标变量名默认为 index，数组当前项的变量名默认为 item

列表渲染必须添加 wx:key指令， 来指定列表中项目的唯一的标识符。

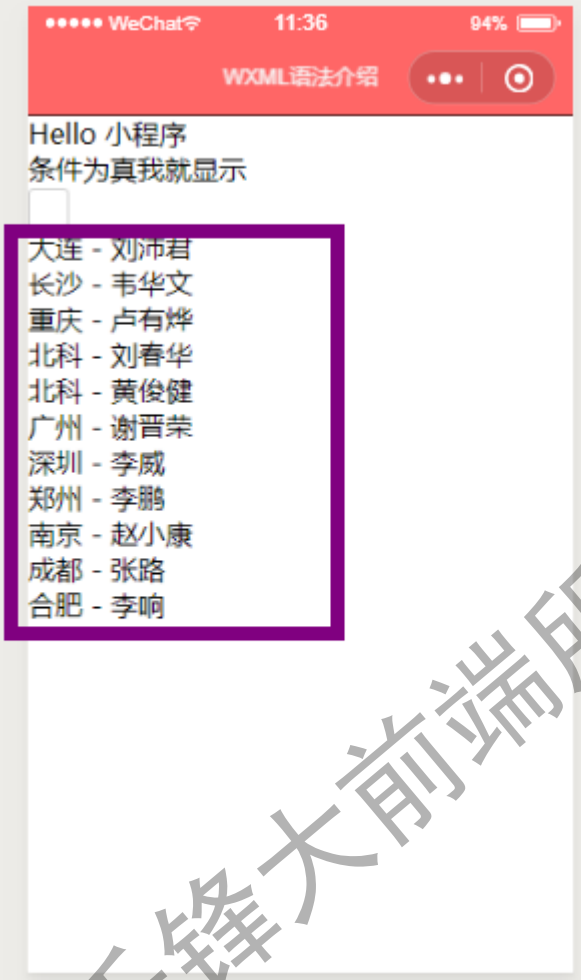
key值可以设置为索引值

```
Page({
  data: {
  teachers: [
  {
  name: '刘沛君',
  city: '大连'
  },
  {
  name: '韦华文',
  city: '长沙'
  },
  {
```

```
  name: '卢有烨',
  city: '重庆'
  },
  {
  name: '刘春华',
  city: '北科'
  },
  {
  name: '黄俊健',
  city: '北科'
  },
  {
  name: '谢晋荣',
  city: '广州'
  },
  {
  name: '李威',
  city: '深圳'
  },
  {
  name: '李鹏',
  city: '郑州'
  },
  {
  name: '赵小康',
  city: '南京'
  },
  {
  name: '张路',
  city: '成都'
  },
  {
  name: '李响',
  city: '合肥'
  },
  ]
  }
})

<view wx:for="{{teachers}}" wx:key="index">
  <text>{{index}}</text>
```

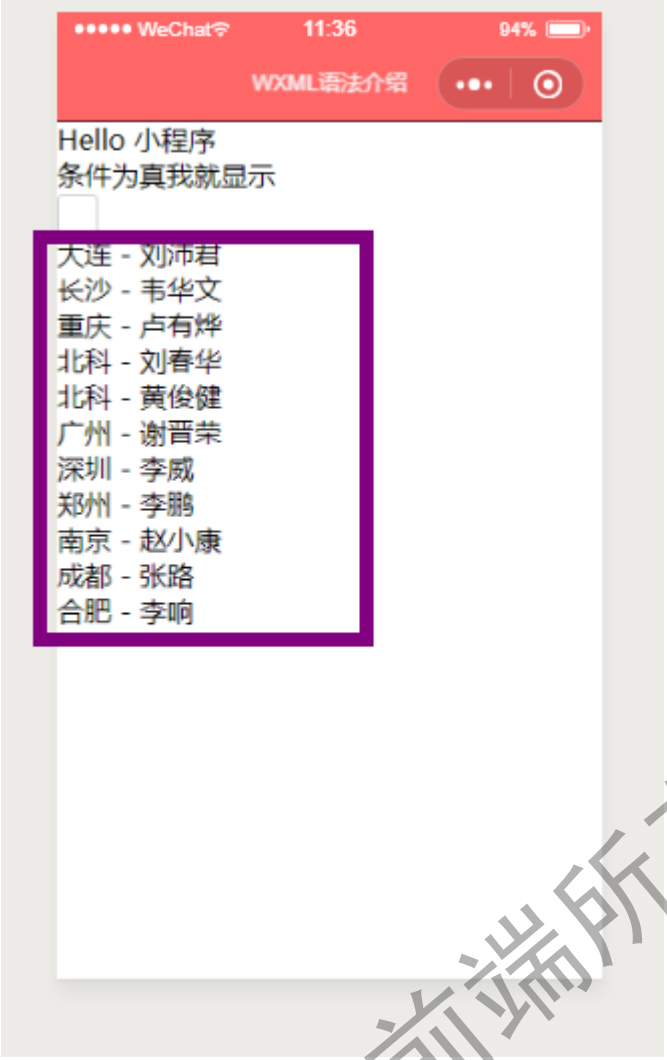
```
54.   -
55.   <text>{{item.city}}</text>
56.   -
57.   <text>{{item.name}}</text>
58.  </view>
```






默认 选项为item，默认索引值为index，如果需要更改，可以使用如下方式

```
1.  <view wx:for="{{teachers}}" wx:for-item="itm" wx:for-index="idx" wx:key="idx">
2.   <text>{{idx}}</text>
3.   -
4.   <text>{{itm.city}}</text>
5.   -
6.   <text>{{itm.name}}</text>
7.  </view>
```




## 3. 条件渲染

wx:if
在框架中，使用 wx:if=”” 来判断是否需要渲染该代码块

```
<view wx:if="{{flag}}"> True </view>
```

也可以用 wx:elif 和 wx:else 来添加一个 else 块

```
<view wx:if="{{len > 5}}"> 1 </view>
<view wx:elif="{{len > 2}}"> 2 </view>
<view wx:else> 3 </view>
```

> *因为 wx:if 是一个控制属性，需要将它添加到一个标签上。如果要一次性判断多个组件标签，可以使用一个 <block/> 标签将多个组件包装起来，并在上边使用 wx:if 控制属性*

```
<block wx:if="{{true}}">
  <view> view1 </view>
```

```
3.    <view> view2 </view>
4.  </block>
```

==注意==：〈block/〉并不是一个组件，它仅仅是一个包装元素，不会在页面中做任何渲染，只接受控制属性。

**wx:if vs hidden** ——（对比vue中的 v-if 与 v-show）

因为 wx:if 之中的模板也可能包含数据绑定，所以当 wx:if 的条件值切换时，框架有一个局部渲染的过程，因为它会确保条件块在切换时销毁或重新渲染。

同时 wx:if 也是惰性的，如果在初始渲染条件为 false，框架什么也不做，在条件第一次变成真的时候才开始局部渲染。

相比之下，hidden 就简单的多，组件始终会被渲染，只是简单的控制显示与隐藏。

一般来说，wx:if 有更高的切换消耗而 hidden 有更高的初始渲染消耗。因此，如果需要频繁切换的情景下，用 hidden 更好，如果在运行时条件不大可能改变则 wx:if 较好

# 五、WXS语法（了解）

WXS（WeiXin Script）是小程序的一套脚本语言，结合 WXML，可以构建出页面的结构。

WXS 与 JavaScript 是不同的语言，有自己的语法，并不和 JavaScript 一致。

熟悉js语法的可以很快速的接收并且掌握它。

# 六、WXSS语法

WXSS (WeiXin Style Sheets)是一套样式语言，用于描述 WXML 的组件样式。

WXSS 用来决定 WXML 的组件应该怎么显示。

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。

与 CSS 相比，WXSS 扩展的特性有：

尺寸单位

样式导入

## 1. 尺寸单位

## 1. 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。如在 iPhone6 上，屏幕宽度为375px，共有750个物理像素，则750rpx = 375px = 750物理像素，1rpx = 0.5px = 1物理像素

| 设备 | rpx换算px（屏幕宽度/750） | px换算rpx（750/屏幕宽度） |
| --- | --- | --- |
| iPhone5 | 1rpx = 0.42px | 1px = 2.34rpx |
| iPhone6 | 1rpx = 0.5px | 1px = 2rpx |
| iPhone6 Plus | 1rpx = 0.552px | 1px = 1.81rpx |

> *==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。*
>
> *==注意==：在较小的屏幕上不可避免的会有一些毛刺，请在开发时尽量避免这种情况*

## 2. 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

## 3. 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 page 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。