

微信小程序开发

一、微信小程序文档

微信开放文档 (qq.com)

1 <https://developers.weixin.qq.com/miniprogram/dev/framework/>

微信小程序有一套完整的开发文档，所有的内容全部都在里面，然后要注意，微信小程序是基于vue的生态的，所以只要是掌握了vue的开发，对于微信小程序来说是非常容易上手的

二、开发工具

https://servicewechat.com/wxa-dev-logic/download_redirect?type=win32_x64&from=mpwiki&download_version=1062210310&version_type=1

微信开发者工具下载地址与更新日志

Windows 仅支持 Windows 7 及以上版本

稳定版 Stable Build (1.06.22010310)

测试版缺陷收敛后转为稳定版。Stable版本从 1.06 开始不支持Windows7，建议开发者升级 Windows 版本。

Windows 64、Windows 32、macOS x64、macOS ARM64

预发布版 RC Build (1.06.2210141)

预发布版，包含大的特性；通过内部测试，稳定性尚可。RC版本从 1.06 开始不支持Windows7，建议开发者升级 Windows 版本。

Windows 64、Windows 32、macOS x64、macOS ARM64

开发版 Nightly Build (1.06.2211102)

日常构建版本(基于 NW.js 0.54.1)，用于尽快修复缺陷和敏捷上线小的特性；开发自验证，稳定性欠佳。Nightly版本从 1.06 开始不支持Windows7，建议开发者升级 Windows 版本。

Windows 64、Windows 32、macOS x64、macOS ARM64

小游戏版 Minigame Build

适用于最新的小游戏开发者工具，其他版工具依然只能体验到旧版本小游戏开发模式。

点击下载小游戏版开发者工具

全部更新日志

[全部更新日志](#)

微
· 稳
· 预
· 于
· 小
· 全

三、微信小程序的特点

一般情况下，如果碰到面试的时候面试到微信小程序，如果面试官问：“你是怎么看待微信小程序的，它有什么特点”

1. 微信小程序是介于webApp与原生APP之间的，它既具备webApp的传播快速的特点，也具备原生APP的强大功能
2. 开发快速，相较于原生APP来说它只用开发一个版本就可以发，因为只要终端上面安装了微信就可以直接使用这个小程序；而如果要开发原生的webApp，这个时个就很麻烦，因为可能既要开发

androidr的版本，还要开发ios的版本

3. 微信小程序也是基于组件化的开发，它也是一种类似于 `mvvm` 的开发方式,这一种开发方式与之前所学习的vue是一样的，所以有了vue的基础以后上手会非常快

小程序与普通网页开发的区别

小程序的主要开发语言是 JavaScript，小程序的开发同普通的网页开发相比有很大的相似性。对于前端开发者而言，从网页开发迁移到小程序的开发成本并不高，但是二者还是有些许区别的。

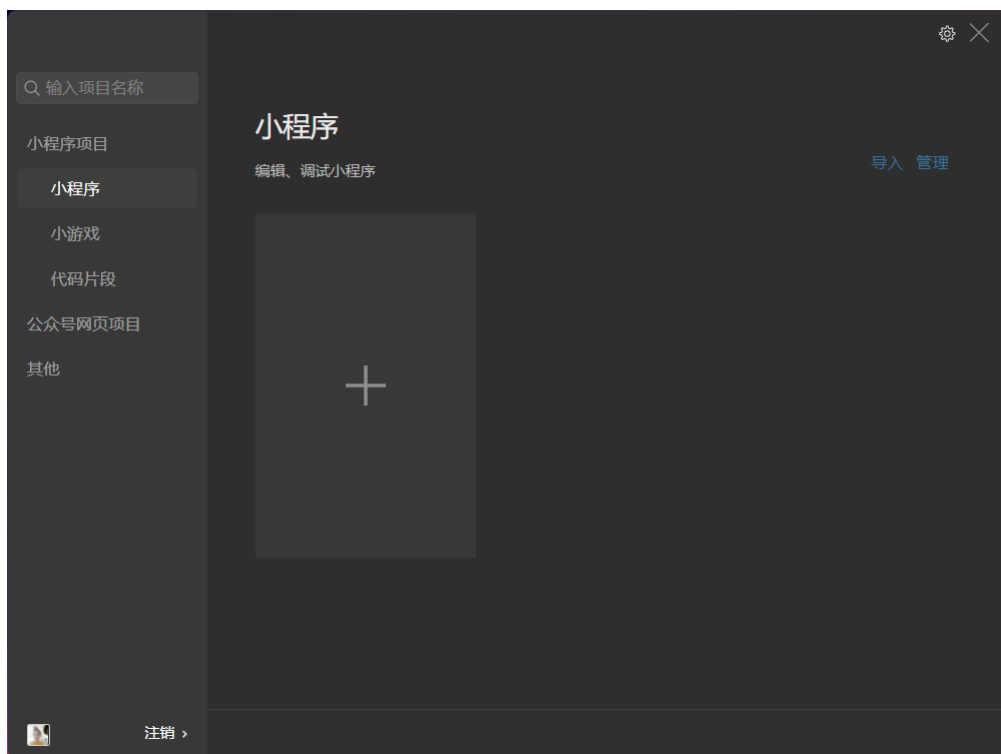
网页开发渲染线程和脚本线程是互斥的，这也是为什么长时间的脚本运行可能会导致页面失去响应，而在小程序中，二者是分开的，分别运行在不同的线程中。网页开发者可以使用到各种浏览器暴露出来的 DOM API，进行 DOM 选中和操作。而如上文所述，小程序的逻辑层和渲染层是分开的，逻辑层运行在 JSCore 中，并没有一个完整浏览器对象，因而缺少相关的DOM API和BOM API。这一区别导致了前端开发非常熟悉的一些库，例如 jQuery、Zepto 等，在小程序中是无法运行的。同时 JSCore 的环境同 NodeJS 环境也是不尽相同，所以一些 NPM 的包在小程序中也是无法运行的。 **这句话就充分的说明了，小程序的js可以理解为node.js，但并不是完整的node.js**

网页开发者需要面对的环境是各式各样的浏览器，PC 端需要面对 IE、Chrome、QQ浏览器等，在移动端需要面对Safari、Chrome以及 iOS、Android 系统中的各式 WebView。而小程序开发过程中需要面对的是两大操作系统 iOS 和 Android 的微信客户端，以及用于辅助开发的小程序开发者工具，小程序中三大运行环境也是有所区别的，如表1-1所示。

上图就是微信小程序官网上面所说的小程序开发与普通网页开发的区别

在上图当有一个注意事项要说明，小程序没有DOM与BOM。

四、小程序的界面



上图就是小程序安装成功以后界面，我们可以在这个界面上面创建自己小程序

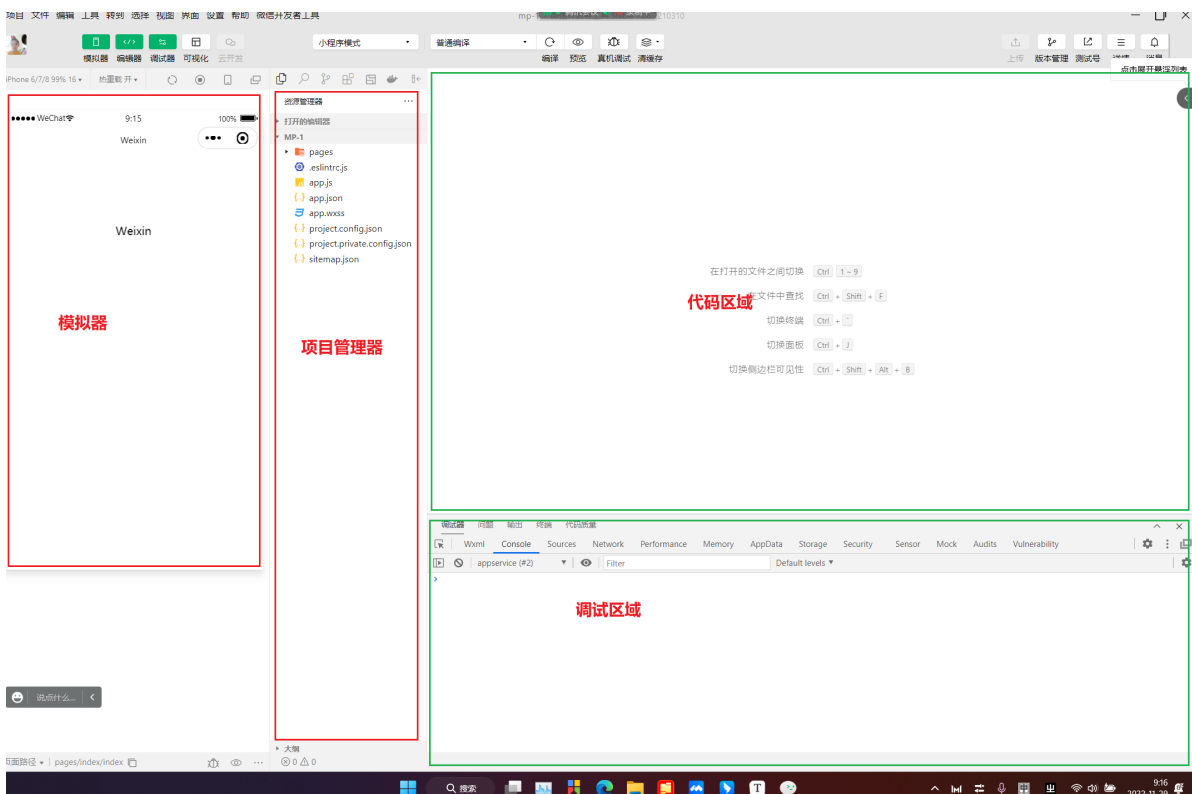
五、小程序的创建



上图就是小程序项目的创建，有几个点要说明一下

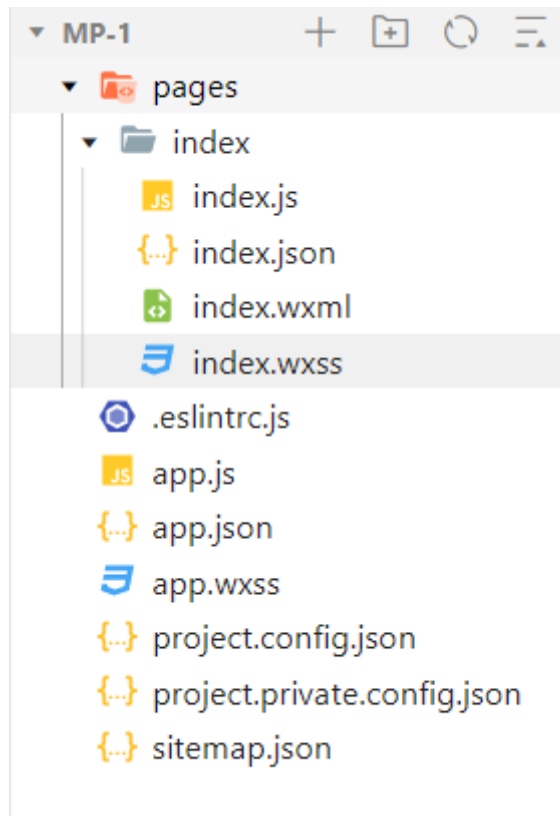
1. 小程序的AppID可以使用测试账号，也可以自己注册申请一个账号，如果使用测试账号，则不可以进行云开发【对于前端开发者而言，小程序的云开发还是非常有必要的】
2. 小程序里面有很多模块可以提供给我们使用，但有些模块会收费的，酌情使用

六、小程序的开发界面



七、小程序的项目文件

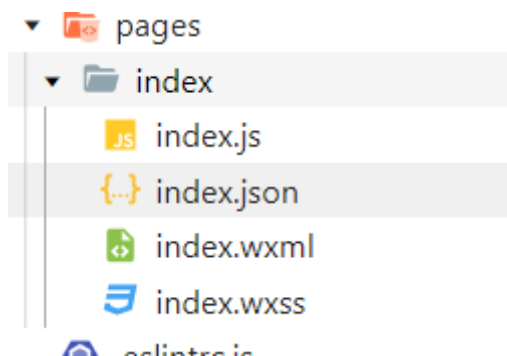
当一个小程序创建成功以后，它有如下的几个文件



1. `pages` 相当于vue开发当中的 `views` 目录，存放所有的页面的，它的每一个页面都是一个文件夹，如上面的 `index` 就是一个文件夹
2. `.eslintrc.js` 代表了当前代码的格式是什么样子，不用理会
3. `app.js` 这是微信小程序的启动文件，小程序就是从这里启动的，也相当于我们vue开发里面的 `main.js` 文件
4. `app.json` 是整个小程序的配置文件，它可以配置一基本小程序的基本信息，如 `titleBar/tableBar` 等相关信息的配置
5. `app.wxss` 这一个是小程序的全局样式文件
6. `project.config.js` 小程序的配置信息，相当于 `webpack` 配置，可理解为vue开当中的 `vue.config.js` 文件

八、小程序的页面结构

之前也讲过，小程序的所有页面都放在 `pages` 的目录下面，而每一个页面都会以一个文件夹的形式存在



这个时候我们可以看到 `index` 目录下就会有4个文件，其实这4个文件非常好理解

小程序	网页	vue
wxml	html	<template>
wxss	css	<style>
js	js	<script>
json	json	json

通过上面的对比其实也可以发现一个特点，小程序就是将原来的文件的后缀名改了一下，也可以理解为它将 `vue` 文件里面的几个部分进行了拆分

```

1 <template>
2   <router-view></router-view> wxml
3 </template>
4 <script setup>
5     js
6 </script>
7 <style scoped>
8     wxss
9 </style>

```

九、小程序新建页面

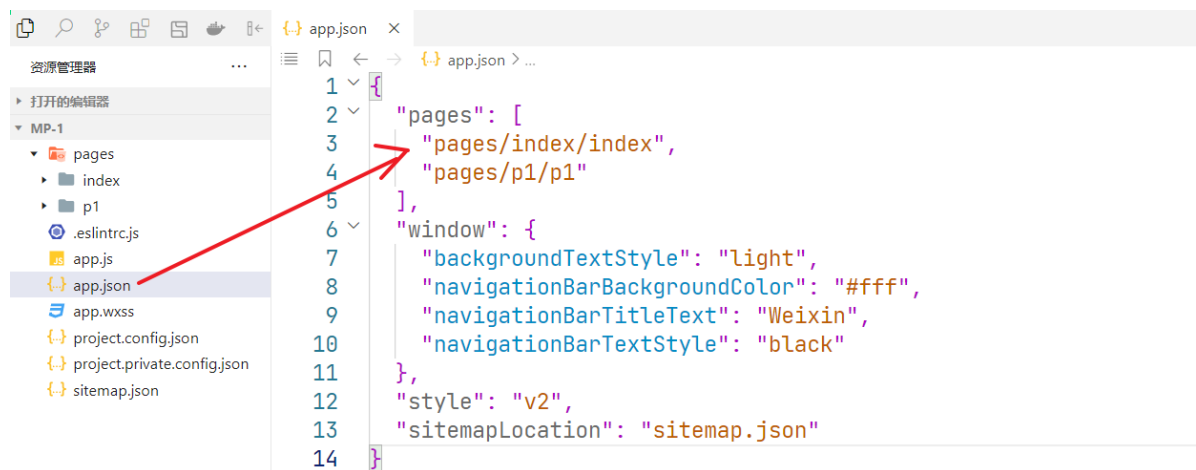
小程序如果要新建页面，有2种常用的方法

第一种方式

- 先在 `pages` 的目录下面新建一个文件夹
- 在新建的文件夹上面点击鼠标右键，选择新建 `Page`



当我们新建page以后，这个时候它会自动的帮我们去创建4个文件



每一个创建好以后的网页都会在 `app.json` 的这个文件下面 `pages` 的选项下面完成这个记录

第二种方法

这一种方法在工作当中我们经常使用

我们都知道了所有的页面最终都会在 `app.json` 的页面下面 `pages` 里面进行记录

所以，直接在 `pages` 的选项下面配置好以后，它会自动帮我们创建小程序以的页面

十、小程序的首页面设置

现在我们已经有了3个页面了，怎么样设置小程序的启动页面呢？

```
{
  "pages": [
    "pages/index/index",
    "pages/p1/p1",
    "pages/p2/p2"
  ],
  "window": {
    "backgroundTextStyle": "light"
  }
}
```

谁排在第一个，小程序默认就启动谁

十一、app.json

这个文件是小程序的全局配置文件

```
1  {
2    "pages": [
3      "pages/p1/p1",
4      "pages/index/index",
5      "pages/p2/p2"
6    ],
7    "window": {
8      "backgroundTextStyle": "light",
9      "navigationBarBackgroundColor": "#fff",
10     "navigationBarTitleText": "Weixin",
11     "navigationBarTextStyle": "black"
12   },
13   "style": "v2",
14   "sitemapLocation": "sitemap.json"
15 }
```

在这个文件里面，我们已经看到了 `pages` 是用来配置页面的，而这个 `window` 就是用于配置整个

小程序的窗口的

具体可以参考官

[全局配置 | 微信开放文档 \(qq.com\)](#)

```
1 https://developers.weixin.qq.com/miniprogram/dev/reference/configuration/app.html#window
```

十二、小程序的组件wxml

小程序是基于组件化开发的，它不能定标签，它所有的一切都是虚拟DOM，也就是组件

```
1 https://developers.weixin.qq.com/miniprogram/dev/component/
```

```
1 <!-- 轮播图 -->
2 <swiper indicator-dots indicator-active-color="#ff0000">
3   <swiper-item>
4     <image src="/images/item1.jpg"></image>
5   </swiper-item>
6   <swiper-item>
7     <image src="/images/item2.jpg"></image>
8   </swiper-item>
9   <swiper-item>
10    <image src="/images/item3.jpg"></image>
11  </swiper-item>
12  <swiper-item>
13    <image src="/images/item4.jpg"></image>
14  </swiper-item>
15 </swiper>
```

小程序本身就提供了很多的组件来供我们使用，这样我们就可以不信赖于其它第三方的组件库

在之前的vue的开发的时候，我们可以还需要借用于 **vant** 或 **element-plus** 来完成操作，现在的小程序已经有了自己的组件了

十三、小程序的样式wxss

小程序与以前的CSS是一样的，但是这里要注意，它的后缀名叫 **wxss**，并且当前的样式文件wxss只作用于当前的网页文件wxml

选择器

目前支持的选择器有：

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

目前的微信小程序的选择器只支持上面的几种，其它的暂时先不要使用。我们还是优先使用 `class` 选择器

```
1 .box1{
2   color: red;
3   border: 1px solid black;
4   height: 50px;
5 }
```

WXSS

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

`wxss` 里面扩展了2个新的东西，其中最麻烦的就是 `rpx` 的单位

在我们以前使用单位的时候，我们会使用 `rem/px` 等做为单位，但是微信小程序推荐使用 `rpx` 来做为单

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。

所以我们在进行开的时候，尽量使用 **iphone 6** 来进行开发

十四、小程序的数据绑定「重点」

小程序执行也是也 **mvvm** 的原则，也就是数据驱动页面的原则

```

1  /**
2   * 小程序执行的也是数据驱动页面
3   */
4  Page({
5    // 这个data就是负责接管页面的数据
6    data:{
7      userName:"标哥哥",
8      x:"box"
9    }
10  });
11
```

在当前而面的 **js** 文件里面，有一个 **data** 属性，这个属性下面所定义的所有的数据都可以直接在页面上面使用

1. 普通渲染

```

1  <view>{{userName}}</view>
2  <!-- 这里就要注意，它所有的地方全部使用的都是是{{}} -->
3  <view id="{{x}}">属性也是使用花括号来进行的</view>
```

2. 条件渲染

这个东西在以前学习的vue里面，使用 **v-if/v-show** 来进行的，现在换了一个指令，叫 **wx:if**

```

1  <view wx:if="{{1<2}}">标哥哥</view>
2  <view wx:else>小姐姐</view>
```

当然，多个条件在一起也是可以的

```

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

3.列表渲染

这个东西与之前在vue里面所学习的 `v-for` 保持一致

```
1  Page({
2    data: {
3      userList: ["张三", "李四", "王五", "赵六", "田七"]
4    }
5  })
```

现在将上面的 `userList` 进行渲染

```
1  <!--
2    在以前的vue里面v-for="(item , index) in userList"
3  -->
4  <view wx:for="{{userList}}" wx:key="item">{{item}}----{{index}}</view>
```

代码说明：

1. 遍历出来的每一项默认叫 `item`
2. 遍历出来的每一项的索引默认叫 `index`
3. 在执行 `wx:for` 的列表渲染的时候，与 `vue` 是一样的，要有一个唯一的 `key` 来说明

如果希望改变默认遍历出来的每一项的名称或索引，可以通过下面的方式

```
1  <view wx:for="{{userList}}" wx:for-item="a" wx:for-index="idx" wx:key="a">{{a}}---
   -{{idx}}</view>
```

- 通过 `wx:for-item` 来实现对遍历出来的每一项的名称的更改
- 通过 `wx:for-index` 来实现对遍历出来的每一项的索引更改

同时几种渲染也可以结合在一起使用

```
1  <view class="{{index%2!=0?'active':null}}" wx:for="{{userList}}">{{item}}</view>
```

十五、小程序里面的事件

小程序里面也有很多事件，它的事件绑定有2种方式

1. 小程序的bind事件绑定

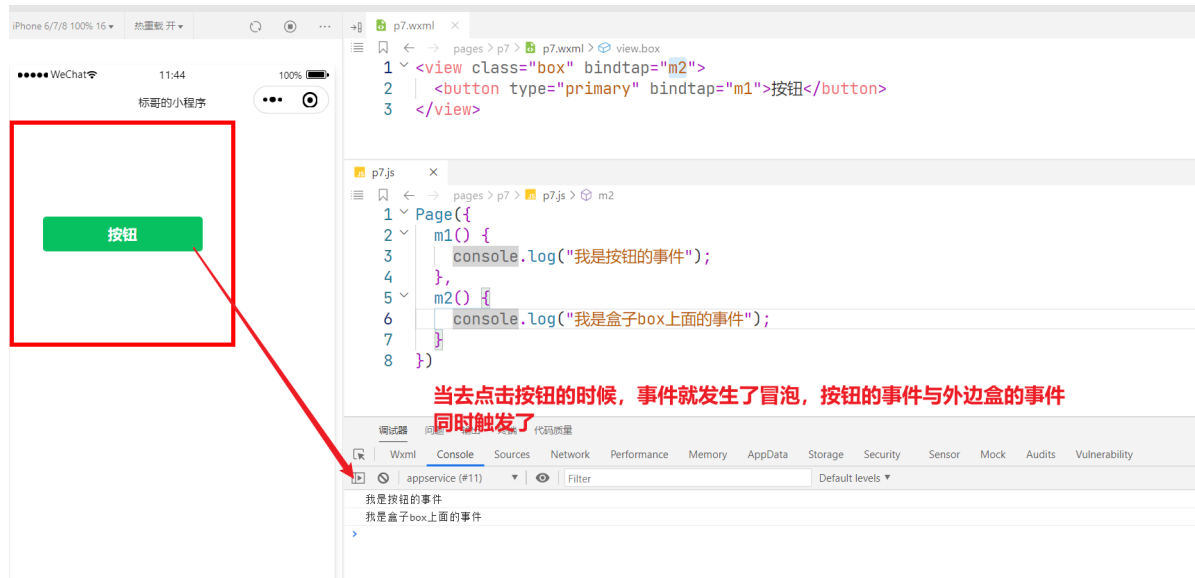
```
1  <button type="primary" bindtap="sayHello">按钮一</button>
```

小程序在进行事件绑定的时候，后面的函数是一个函数名，不能添加括号

```
1  Page({
2    // 小程序页面里面的事件方法，不用写在methods里面，直接写在page下面就可以了
3    sayHello(event) {
4      console.log("hello world");
5    }
6  })
```

通过上面的代码，我们可以得到，如果要绑定一个事件，我们可以使用 `bind事件名` 来进行。同时要注意，小程序里面的事件是直接写在 `page` 下面

这一种事件绑定会有一个缺点，它会形成事件冒泡



如何去阻止事件冒泡

在以前的dom里面，我们可以使用 `event.stopPropagation()` 来完成阻止事件冒泡，但是在微信小程序里面是可以的

小程序与普通网页开发的区别

小程序的主要开发语言是 JavaScript，小程序的开发同普通的网页开发相比有很大的相似性。对于前端开发者而言，从网页开发迁移到小程序的开发成本并不高，但是二者还是有些许区别的。

网页开发渲染线程和脚本线程是互斥的，这也是为什么长时间的脚本运行可能会导致页面失去响应，而在小程序中，二者是分开，分别运行在不同的线程中。网页开发者可以使用到各种浏览器暴露出来的 DOM API，进行 DOM 选中和操作。而如上文所述，小程序的逻辑层和渲染层是分开的，逻辑层运行在 JSCore 中，并没有一个完整浏览器对象，因而缺少相关的 DOM API 和 BOM API。这一区别导致了前端开发非常熟悉的一些库，例如 jQuery、Zepto 等，在小程序中是无法运行的。同时 JSCore 的环境同 NodeJS 环境也是不尽相同，所以一些 NPM 的包在小程序中也是无法运行的。这句话就充分的说明了，小程序的js可以理解成node.js，但并不是完整的node.js

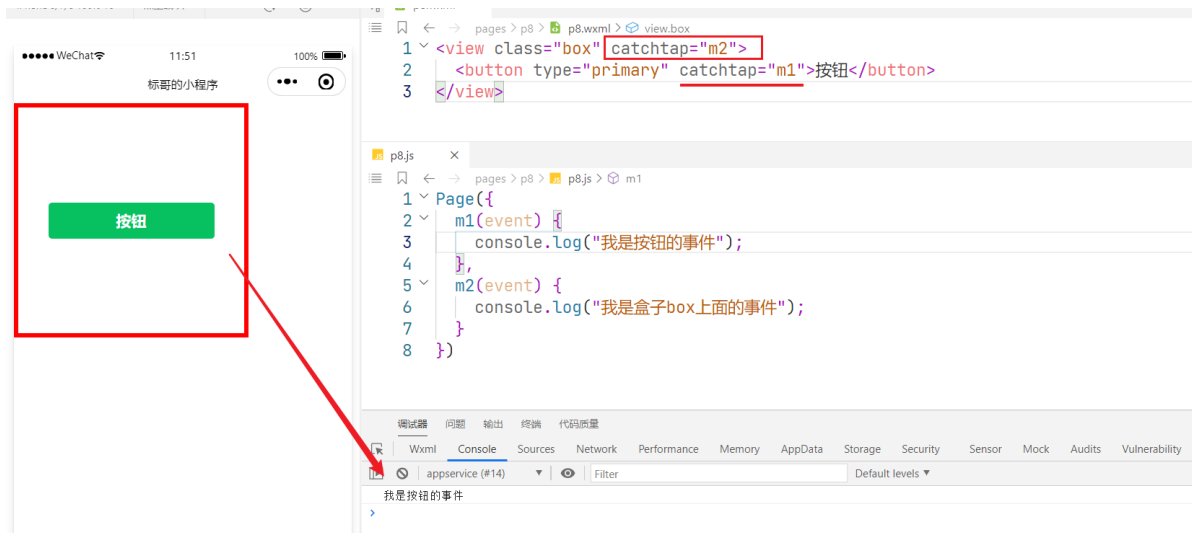
网页开发者需要面对的环境是各式各样的浏览器，PC 端需要面对 IE、Chrome、QQ浏览器等，在移动端需要面对Safari、Chrome以及 iOS、Android 系统中的各式 WebView。而小程序开发过程中需要面对的是两大操作系统 iOS 和 Android 的微信客户端，以及用于辅助开发的小程序开发者工具，小程序中三大运行环境也是有所区别的，如表1-1所示。

根本的原则之前已经说过，微信小程序没有DOM，也没有BOM。所以就不能使用DOM的操作方法

2. 小程序的catch事件绑定

联想记忆，在vue开发里面，如果想进行阻止事件冒泡可以使用事件修饰符 `@click.stop` 这种方式

之前的事件绑定我们使用的是bind来进行的，但是bind的事件绑定是会有事件冒泡的，如果不想产生事件冒泡，则使用 `catch` 进行事件绑定

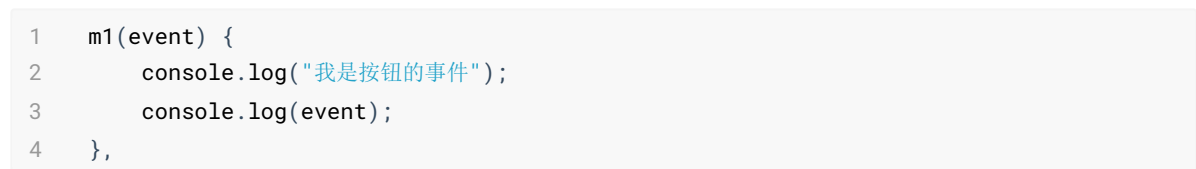
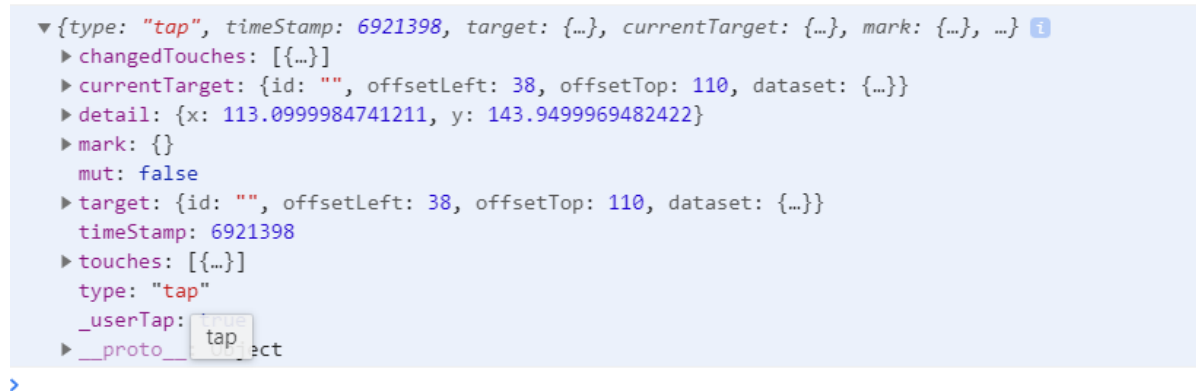


当我们使用 `catch` 进行事件绑一定的时候，我们就发现已经没有事件冒泡的行为了

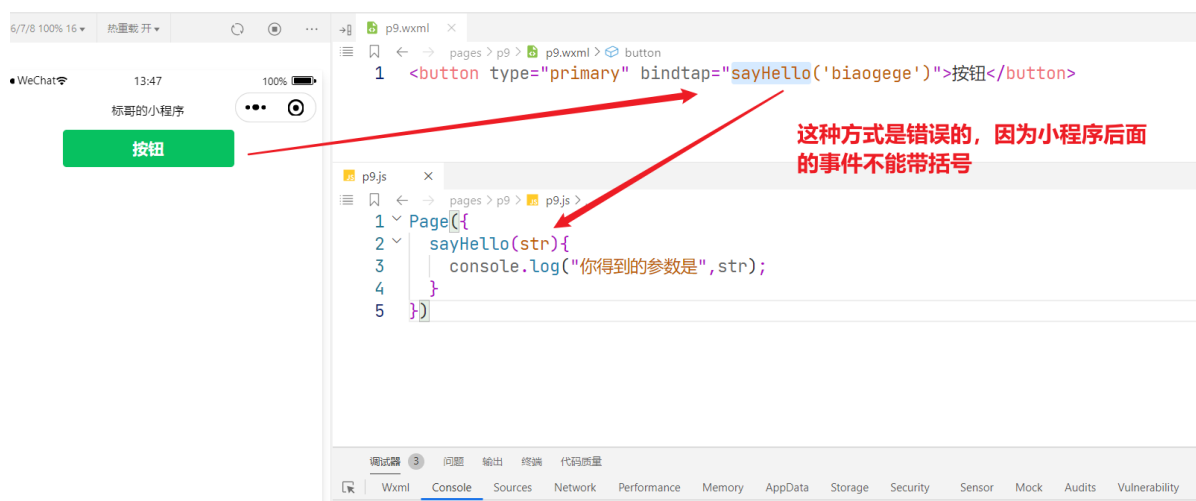
3.事件对象event

小程序里面的事件与普通的DOM事件有一点相信，它们的事件方法上面都会默认接收一个 `event` 的事件对象

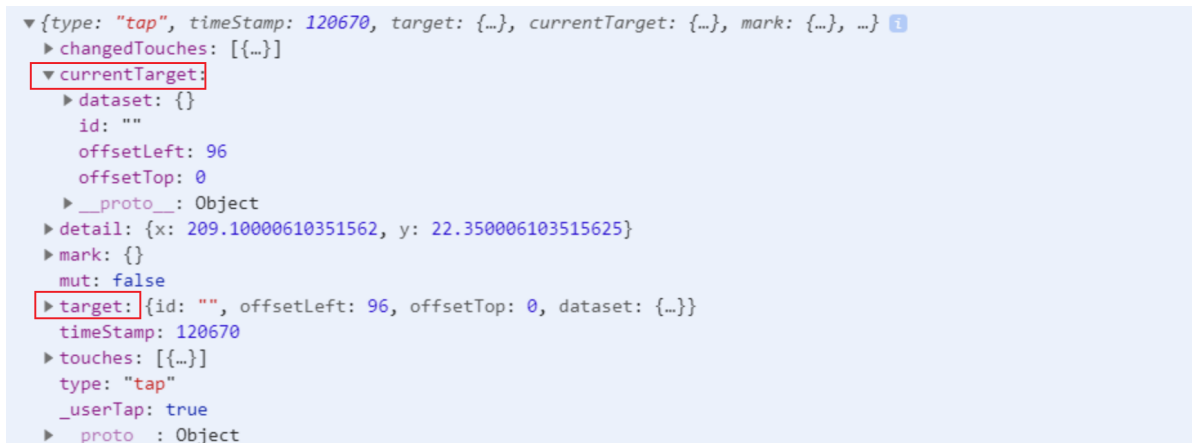
我是按钮的事件



这个事件对象目前阶段可以给我们带来的作用就是用于小程序的方法传参

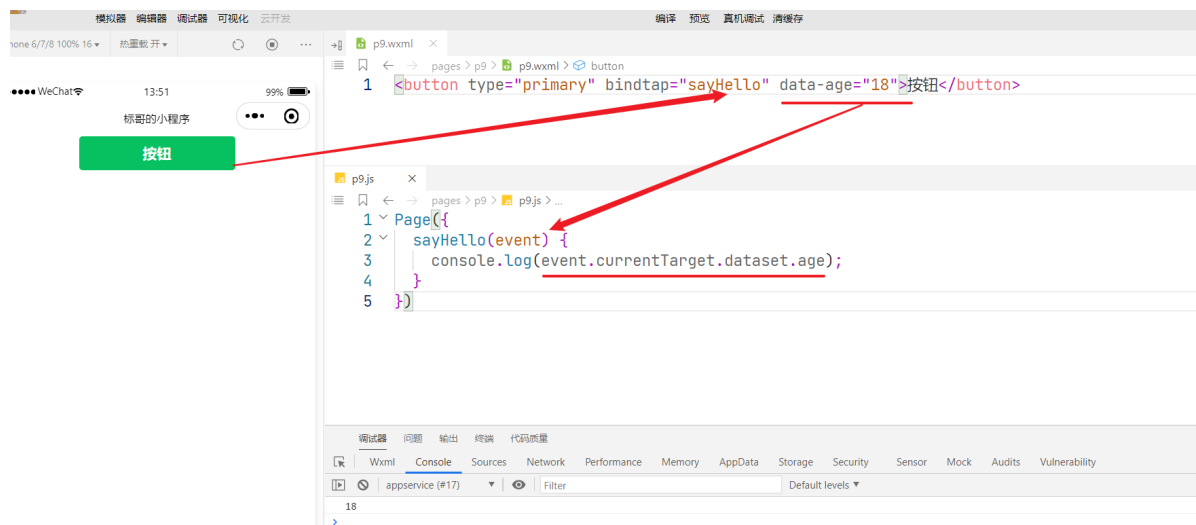


小程序的事件方法里面如果要接收事件参数，则必须使用 `event` 对象



在事件对象里面，我们可以看到2个很熟悉的东西

1. `event.target` 事件的触发者
2. `event.currentTarget` 事件的绑定者



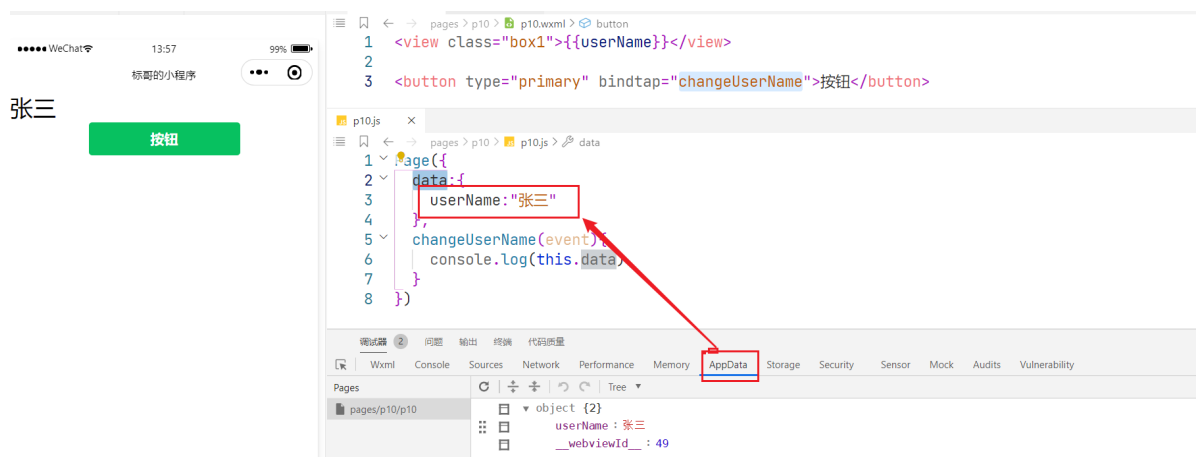
在上面的代码里面，我们可以看到，小程序的事件方法的参数传递采用的是 `dataset` 的形式完成

十六、小程序的数据data

之前就跟同学们说过，微信小程序与我们之前所学习的 `vue` 非常的像，它里面都有一个 `data` 是用于页面的数据的

在之前提到MVVM的概念的时候，数据改变页面改变，页面改变，数据也改变，这是vue进行数据绑定的特点

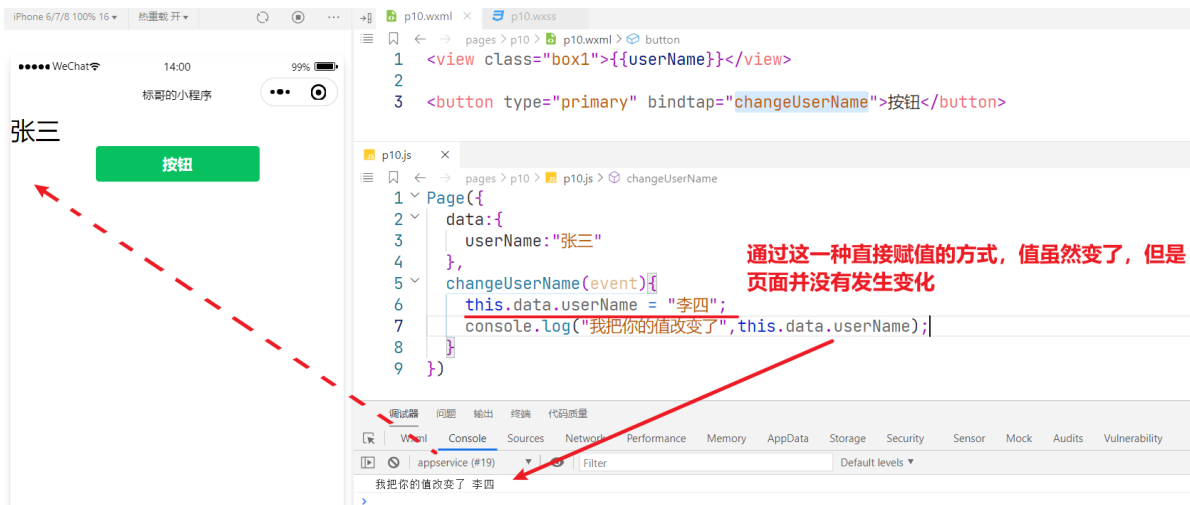
但是在微信小程序里面，这个地方有一点区别



我们可以在调试工具里面看到所定义的 `data` 的数据

1. 单向数据绑定

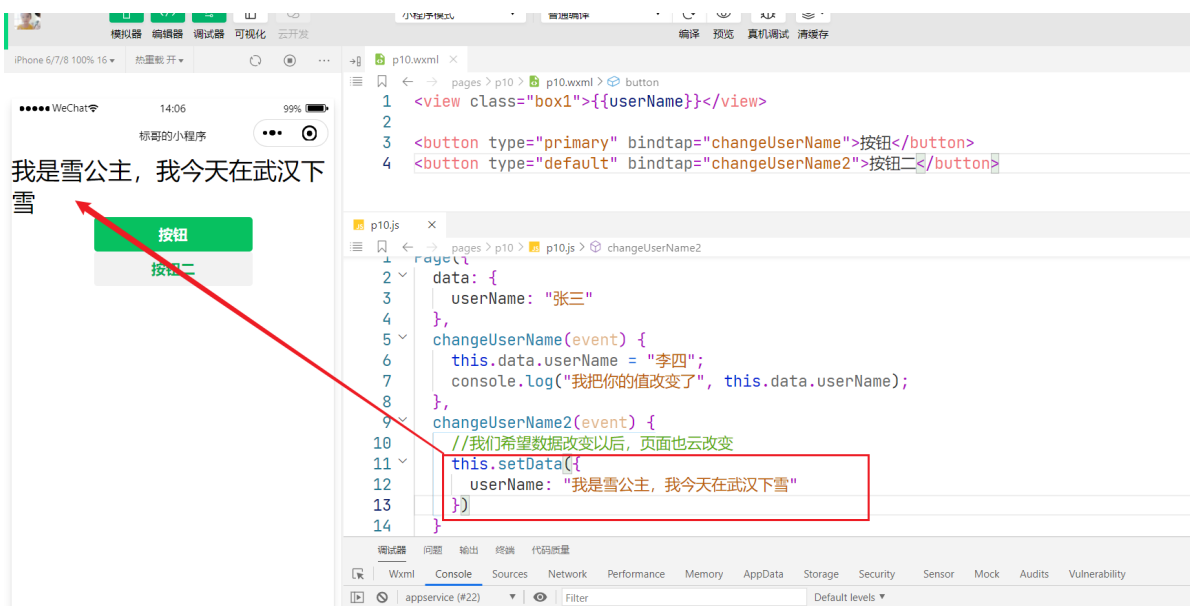
这一种情况下面，如果我们要改变 `data` 里面的值，页面是并不会发生变化的




通过这一种方式直接赋值，值虽然变了，但是页面并不会重新渲染

2. 双向数据绑定

如果要实现双向的数据绑定，这个时候我们可以就要使用一种特殊的方法



通过 `setData` 这一种方式进行的赋值，它会改变值以后，再将页面重新渲染

 **注意事项：** `setData` 这个方法是异常的，它在赋值结束以后会有一个回调函数，如果希望某些操作在赋值结束以后再去进行，我们就可以使用回调函数

```
1 //我们希望数据改变以后，页面也云改变
2 this.setData({
3   userName: "我是雪公主，我今天在武汉下雪"
4 }, () => {
5   console.log("我是回调函数，它会在赋值结束以后云执行");
6   console.log(this.data.userName);
7 })
```

十七、案例一



我们希望在点击的时候，实现这一种切换效果

在这个案例里面，我们涉及到布局，涉及到样式，还有事件及数据渲染

```
1 <view class="movie-type">
2   <text class="{{currentIndex==0?'active':null}}" bindtap="changeCurrentIndex"
   data-index="0">热映</text>
3   <text class="{{currentIndex==1?'active':null}}" bindtap="changeCurrentIndex"
   data-index="1">待映</text>
4   <text class="{{currentIndex==2?'active':null}}" bindtap="changeCurrentIndex"
   data-index="2">经典电影</text>
5 </view>
```

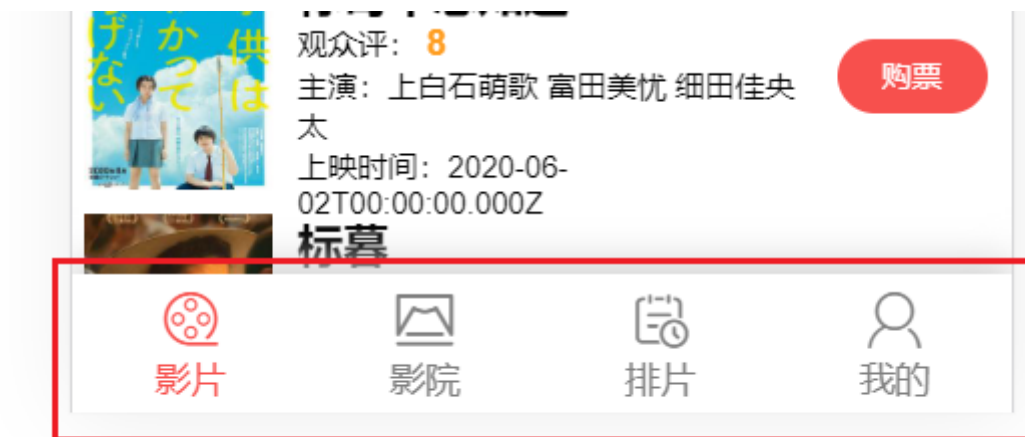
```
1 .movie-type{
2   border-bottom: 2rpx solid gray;
3   height: 80rpx;
4   display: flex;
5   flex-direction: row;
6 }
7 .movie-type text{
8   display: flex;
9   justify-content: center;
10  align-items: center;
11  font-size: 24rpx;
12  font-weight: bold;
13  color: gray;
14  padding: 0 20rpx;
15 }
16 .movie-type text.active{
17   color: black;
18   position: relative;
19 }
20 .movie-type text.active::after{
21   content: "";
22   position: absolute;
23   height: 4rpx;
24   background-color: #f7504d;
25   left: 0;
26   right: 0;
27   bottom: 0;
28   width: 50rpx;
29   margin: auto;
30 }
```

```

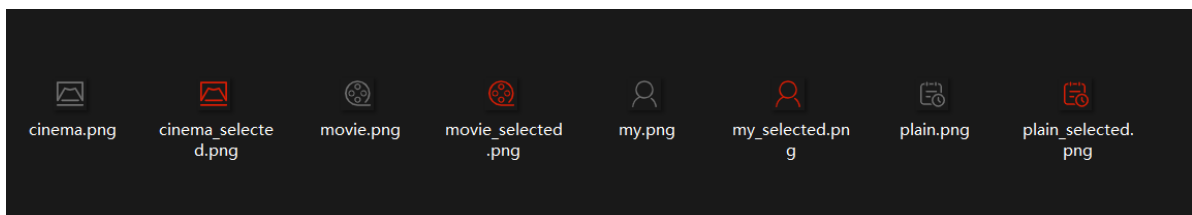
1  Page({
2    data: {
3      currentIndex: 0
4    },
5    changeCurrentIndex(event) {
6      //现在要拿到参数data-index
7      let index = event.currentTarget.dataset.index;
8      this.setData({
9        currentIndex: index
10     });
11   }
12 })

```

十八、关于 tabBar 的配置



小程序是做app的，如果我们想配置类似于上面的东西，应该怎么办呢？在vue的开发下面，我们可以使用嵌套路由去完成，在微信小程序里面，做起来，更加简单



首先，我们准备所需要使用的图片

接下来，创建四个所需要使用的页面

```

1  {
2    "pages": [
3      "pages/movie/movie",
4      "pages/cinema/cinema",
5      "pages/plain/plain",
6      "pages/my/my"
7    ],
8    "window": {
9      "backgroundTextStyle": "light",
10     "navigationBarBackgroundColor": "#fff",
11     "navigationBarTitleText": "Weixin",
12     "navigationBarTextStyle": "black"
13   },

```



```
14     "style": "v2",
15     "sitemapLocation": "sitemap.json"
16 }
```

pages里面就是我们所需要使用的页面

```
1  {
2    "pages": [
3      "pages/movie/movie",
4      "pages/cinema/cinema",
5      "pages/plain/plain",
6      "pages/my/my"
7    ],
8    "window": {
9      "backgroundTextStyle": "light",
10     "navigationBarBackgroundColor": "#fff",
11     "navigationBarTitleText": "猫眼电影",
12     "navigationBarTextStyle": "black"
13   },
14   "style": "v2",
15   "sitemapLocation": "sitemap.json",
16   "tabBar": {
17     "selectedColor": "#f7504d",
18     "list": [
19       {
20         "text": "影片",
21         "pagePath": "pages/movie/movie",
22         "iconPath": "/images/movie.png",
23         "selectedIconPath": "/images/movie_selected.png"
24       }, {
25         "text": "影院",
26         "pagePath": "pages/cinema/cinema",
27         "iconPath": "/images/cinema.png",
28         "selectedIconPath": "/images/cinema_selected.png"
29       }, {
30         "text": "排片",
31         "pagePath": "pages/plain/plain",
32         "iconPath": "/images/plain.png",
33         "selectedIconPath": "/images/plain_selected.png"
34       }, {
35         "text": "我的",
36         "pagePath": "pages/my/my",
37         "iconPath": "/images/my.png",
38         "selectedIconPath": "/images/my_selected.png"
39       }
40     ]
41   }
42 }
```

在上面的 `app.json` 的文件里面，我们可以看到，使用了 `tabBar` 这个选项，然后 `list` 就是用于配置的

具体的配置信息，可以参那下面

[全局配置 | 微信开放文档 \(qq.com\)](#)

十九、小程序的页面跳转

在之前进行vue的开发的时候，我们使用过 `vue-router` 进行页面的跳转，现在在微信小程序里面，我们可以使用内置的方法来完成跳转

1. `wx.navigateTo` 跳转到某一个页面，原页面是不会消失，并且，新的页面可以通过 `titleBar` 返回过来[这个相当于 `router.push`]

```
1 wx.navigateTo({  
2   url: "/pages/p2/p2"  
3 });
```

2. `wx.navigateBack()` 返回之前的页面[这个就相当于 `router.back`]

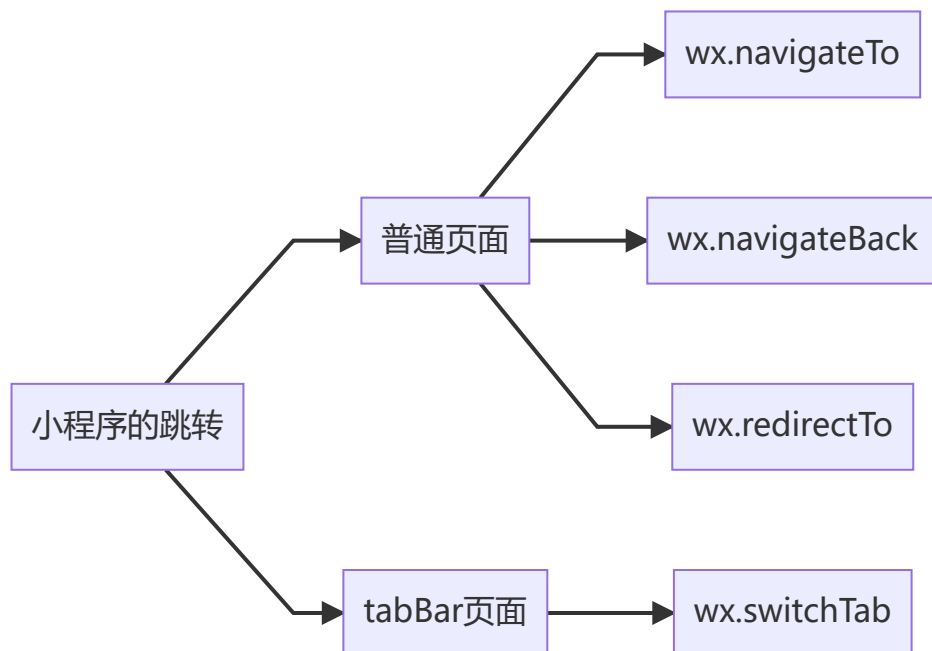
3. `wx.redirectTo()` 跳转到某一个页面，原页面会销毁，不能再返回到原页面了[这相当于 `router.replace()`]

```
1 wx.redirectTo({  
2   url: "/pages/p2/p2"  
3 });
```

4. `wx.switchTab()` 这个方法是用于跳转 `tabBar` 里面的页面的



简而言之就是如果想在下面的四个页面之间跳转，我们就只要使用 `swtichTab` 这种方式



小说明：虽然 `redirectTo` 可以跳 `tabBar` 的页面，但是我们建议这么种

二十、微信小程序界面反馈API

1. `wx.showLoading()` 显示加载动画
2. `wx.hideLoading()` 隐藏加载的动画
3. `wx.showToast()` 显示轻消息提示
4. `wx.hideToast()` 隐藏轻消息提示
5. `wx.showModal()` 显示模块框

```
1  wx.showModal({
2    title: "标哥提示",
3    content: "标哥说，你们在睡觉",
4    confirmText: "肯定一点",
5    cancelText: "我取消了",
6    // showCancel:false,
7    success: result => {
8      //这里的success代表用户点击了确定或取消以后
9      if(result.confirm){
10        //代表用户点击的是确定
11      }
12      else{
13        //用户点击的就是取消
14      }
15    }
16  })
```

6. `wx.showActionSheet()` 显示操作列表

```
1  wx.showActionSheet({
2    itemList: ["登录", "注册"],
3    success: result => {
4      if(result.tapIndex == 0){
5
6      }
7      else if(result.tapIndex == 1){
8
9      }
10   }
11 })
```

7. `wx.previewImage()` 预览图片

上面所展示的都是常用的，还有很多

二十一、系统相关API

1. `wx.getSystemInfoAsync()` 获取系统信息
2. `wx.getNetworkType()` 获取当前设置的网络状态
3. `wx.getLocation()` 获取当前的定位信息，注意，新版本的微信小程序里面如果要使用定位则需要配置，并申请授权，在 `app.json` 里面进行如下配置

```

1  "requiredPrivateInfos": [
2    "chooseAddress",
3    "chooseLocation",
4    "getLocation",
5    "choosePoi"
6  ]

```

4. `wx.openLocation()` 打开地图，可以设置经度与纬度

```

1  wx.getLocation({
2    isHighAccuracy:true,
3    success:res=>{
4      console.log("定位成功");
5      console.log(res);
6      //可以再将这个东西放在地图里面显示出来
7      wx.openLocation({
8        latitude: res.latitude,
9        longitude: res.longitude,
10     })
11   },
12   fail:error=>{
13     console.log("定位失败");
14     console.log(error);
15   }
16 })

```

5. `wx.setStorageSync(key, value)` 设置一个缓存

```

1  // 设置一个缓存
2  wx.setStorageSync('userName', "张三");
3  //小程序里面的缓存是可以缓存对象的，如果是以前的浏览器，要缓存对象，必须转成JSON，但是小程序不用，直接缓存是
4  let dog = {
5    nickName: "大黄",
6    age: 5
7  }
8  wx.setStorageSync('dog', dog);

```

这个缓存，你不清理，它会一直存在，它和 `localStorage` 的特性是一样的

6. `wx.getStorageSync(key)` 根据一个key去获取一个缓存，如果获取不到就是一个空的字符串
7. `wx.removeStorageSync(key)` 根据一个key去删一个缓存
8. `wx.clearStorageSync()` 清除一个缓存

缓存开发当中还是比较重要的，之前所讲过的 `vuex` 的持久化，或 `pinia` 的持久化，都是通过缓存去实现的，如果小程序也想实现持久化的操作，也应该是使用缓存