==========

微信小程序

实质上是Hybrid技术的应用

Hybrid App(混合模式移动应用)：是指介于web-app、native-app这两者之间的app，Hybrid App同时使用网页语言和程序语言开发，总体特征更接近Native App。只是因为同时使用了网页语言编码，所有开发成本和难度比Native App要小很多。Hybrid App兼具了Native App的所有的优势，也兼具了web App使用HTML5跨平台开发成本的优势

微信小程序与其他Hybrid平台对比的优势

\* 微信团队作支撑，接口开放的功能、体验和优化会有持续保障

\* 身份认证识别和消息及时推送

主要技术： **WXML**（XML）、**WXSS**（css）、**javaScript**

但标签必须要有结束符

app.wxss 是全局配置的文件

小程序的js有一些额外的成员

\* App方法 用于定义应用程序实例对象

\* Page 方法 用于定义页面对象

\* getApp 方法 用来获取全局应用程序对象

\* getCurrentPages 方法

用来获取当前页面的调用栈

\* wx 对象 用来提供核心API

小程序的js是支持CommonJS规范的

const foo = require(‘./utils/foo.js’)

foo.say(‘hello’)

**数据绑定** {{}}

可以用在：

\* innerHTML

\* 元素的属性上

<view class=”box {{ content }}”></view>

可直接使用字面量和逻辑运算符

{{ ‘hello’ }}

{{ 111 }}

{{ 111+222 }}

{{ 100>99? ‘对’: ‘错’ }}

当语法解析错误了true/false, 可以使用{{}}解决

<checkbox checked=”false”></checkbox>

在小程序中还是勾选状态，把”false”解析成true了，用下面的方式解决

<checkbox checked=”{{ false }}”></checkbox>

**setData()** //改变data中的数据的值

data () {

mag: “”

}

//某个事件

clickButton () {

...

let data = “点击事件”

this.setData({

mag: data;

})

}

注意： 在wxss中不能直接使用本地的图片作为背景图

解决方式： \* 背景图片路径换成远程网站，即网络图片

\* 使用base64格式编码的图片

\* 直接使用<image/>组件实现

在微信小程序中点击事件是使用

\* bindtap=””（bind+事件名）

\* catchtap=”” (catch+事件名)

- 处理事件冒泡：在子容器中使用catch绑定事件

事件也可以传参

<view bindtap=”tapHandle” data-name=”张三”></view>

Page {

tapHangle(event) {

console.log(event.currentTarget .dataset.name); //张三

}

}

vue中是v-on:click=””，事件 处理函数在methods中执行

微信绑定数据： 在标签上是 id={{ msg }}， vue中是这样 :id=”msg”

不支持javascript的eval() 方法

访问data中的数据 this.data.xxx

**列表渲染**： wx:for

加了 wx:for-item=”aaa” aaa就替代item，表示当前遍历项

wx:for-index=”i” i 就表示遍历元素索引

vue中为 v-for

**条件渲染**

wx:if 和vue中 v-if一样

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>

<view>view2</view>

</block>

注：<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。

**hiddle**

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。

**数据流**

setData()

js ----------------------> WXML

事件

WXML ----------------> js

<input bindinput=”inputHandle” value=”{{ msg }}”/>

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

Page {

data: {

mes: “initial”

},

inputHandle (e) {

this.setData({

mes: e.detail.value

})

}

}

**小程序页面跳转**

<navigator url="/pages/dome1/dome1">

跳转到新页面（dome1）

</navigator>

传值：

<navigator url="/pages/dome1/dome1?name=小明&age=19">

跳转到新页面（dome1）

</navigator>

dome1页面：

Page({

data: {},

onload (option) { //option为页面跳转所带的参数

console.log(option.name); // 小明

console.log(option.age); //19

}

})

跳转到dome1后不能跳回来了（没有返回按钮了）

<navigator url="/pages/dome1/dome1" open-type="redirect"

>

跳转到新页面（dome1）

</navigator>

**js事件跳转**

Page({

data: {},

tabHandle() {

wx.navigateTo({

url: '../dome1/dome1?id=123',

})

}

})

Page({

data: {},

tabHandle() {

wx.redirectTo ({ //相当于加上open-type="redirect"的navigator

url: '../dome1/dome1?id=123',

})

}

})

wx.navigateBack(); //默认返回到上一个页面

跳转回指定页面

wx.navigateBack({

detail: 2 // 往前2个页面

})

**获取后台接口的数据**

\*微信小程序中没有跨越的概念

\*发送异步请求不再是WEB那套AJAX

\*请求的地址必须在管理后台添加白名单

\*域名必须备案，服务端必须采用HTTPS

Page({

data: {},

onLoad: function (options) { //生命周期函数--监听页面加载

wx.request({

url: "https://douban.uieee.com/v2/movie/in\_theaters",

header: {

'content-type': 'json'

},

success(res) {

console.log(res);

}

})

}

}