



## 第一章 走进微信小程序的世界

---

### 第1节 微信小程序课程介绍

简介：介绍本套课程的大纲

- 学习这套课程需要html5+css3+js基础
- 学后水平：可独立开发小程序项目

### 第2节 小程序是什么？小程序应用前景

简介：介绍微信小程序以及它的应用前景

### 第3节 微信小程序的注册流程

简介：讲解微信小程序的注册流程

- 1.去微信公众平台注册小程序账号
- 2.获得appid和appsecret

### 第4节 微信开发者工具安装及使用讲解

简介：介绍微信小程序工具安装以及界面的使用情况

- 出现合法域名校验错误

**Thu Aug 15 2019 12:01:48 GMT+0800 (中国标准时间) request 合法域名校验出错**

如若已在管理后台更新域名配置，请刷新项目配置后重新编译项目，操作路径：“详情-域名信息”

✖ ▶ <http://op.juhe.cn> 不在以下 request 合法域名列表中，请参考文档：<https://developers.weixin.qq.com/dev/framework/ability/network.html>

解决：详情里勾选“不校验安全域名、web-view 域名、TLS 版本以及 HTTPS 证书”

## 第5节 小程序目录结构分析

简介：介绍小程序项目目录的结构

小程序的主体部分：

- app.js（小程序主要逻辑层，用来注册小程序）
- app.json（小程序公共配置）
- app.wxss（小程序公共样式）

## 第6节 小程序配置文件分析

简介：分析app.json、app.wxss等配置文件



愿景："让编程不在难学，让技术与生活更加有趣" 更多课程请访问[xdclass.net](http://xdclass.net)

## 第二章 开始第一个小程序之旅

---

### 第1节 新建一个小程序及框架组成介绍

简介：如何创建一个小程序及框架组成介绍

- 新建页面的步骤
  - 右键新建page生成页面的框架
  - 在app.json中的pages添加我们新建页面的路径

### 第2节 小程序生命周期和页面生命周期讲解

简介：详细介绍小程序的生命周期

- 小程序启动时先执行onLaunch>onShow,然后在执行页面onLoad>onShow>onReady

- 小程序生命周期：
  - onlauch
  - onshow
  - onhide
- 页面生命周期：
  - onload
  - onshow
  - onready
  - onhide
  - onunload

### 第3节 了解小程序自适应尺寸单位rpx

简介：介绍小程序的自适应尺寸单位rpx

rpx单位： 是可以根据屏幕宽度进行自适应

#### 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

### 第4节 如何使用flex弹性布局

简介：讲解flex弹性布局的使用

- 在父容器上加上display:flex，将我们父容器编程弹性容器

```
.sort_box {  
  display: flex;  
  flex-direction: row; //水平排列  
  margin: 0 30rpx;  
  justify-content: space-around; //每个项目的两端间隔相等  
}
```

flex布局博客: <http://www.ruanyifeng.com/blog/2015/07/flex-grammar.html>



小D课堂 愿景: "让编程不在难学, 让技术与生活更加有趣" 更多课程请访问[xdclass.net](http://xdclass.net)

## 第三章 核心知识数据绑定与事件交互

---

### 第1节 必备知识点之数据绑定的概念和基础

简介: 详细分析小程序数据绑定的基础和概念

- Mustache 语法 (双大括号)

```
{{textstr+2}}
```

- 单向数据绑定

### 第2节 详细讲解动态数据绑定

简介: 讲解小程序数据的获取和赋值

- 小程序data赋值

```
this.setData(对象)
```

- 获取data里的值

```
this.data.(key值)
```

### 第3节 深度了解小程序的条件渲染

简介：讲解小程序的条件渲染

- wx:if为true就显示元素，false不显示
- 语法

```
<view wx:if="{{score<60}}">不及格</view>
<view wx:elif="{{score>=80}}">优秀</view>
<view wx:else>良好</view>
```

### 第4节 深度了解小程序的列表渲染

简介：讲解小程序的列表循环渲染

- wx:for绑定数组，wx:for-item值为当前项目的变量名，wx:for-index值为当前下标的变量名

```
<block wx:for="{{list}}" wx:for-item="course" wx:for-index="idx">
  <view class="item">
    <view class="item_img">
      <image src="{{course.img}}"></image>
    </view>
    <view class="item_name">{{course.name}}</view>
    <view>{{idx}}</view>
  </view>
</block>
```

- 列表渲染文档<https://developers.weixin.qq.com/miniprogram/dev/reference/wxml/list.html#wx:key>

## 第5节 讲解绑定事件与事件交互

简介：讲解如何在元素上绑定事件以及事件交互

- bind+事件

```
<button bindtouchstart="changeData" bindtouchend="tmove"
bindlongpress="lpress">改变数据</button>
```

- 需要在js里写对应的方法

```
//点击事件
changeData:function(){
  this.setData({
    textStr:"hello 小程序"
  })
},
//触摸移动
tmove:function(){
  console.log("touchend");
},
//长按触发
lpress:function(){
  console.log("longpress")
},
```

- 官方事件文档<https://developers.weixin.qq.com/miniprogram/dev/framework/view/wxml/event.html>

## 第6节 深入了解事件机制--catch与bind

简介：讲解catch与bind事件机制以及捕获阶段和冒泡阶段

- bind会应用到冒泡机制，catch会阻止事件冒泡

```
<view class="box" bindtap="containerTap">
  <text catchtap="txtTap">小程序</text>
</view>
```

- 捕获阶段的bind与catch,绑定事件需要加上冒号

```
capture-bind:tap(绑定事件)
capture-catch:tap(绑定事件)
```

- 官方事件文档<https://developers.weixin.qq.com/miniprogram/dev/framework/view/wxml/event.html>



小D课堂 愿景："让编程不在难学，让技术与生活更加有趣" 更多课程请访问[xdclass.net](http://xdclass.net)

## 第四章 小程序基础组件及常用组件介绍

### 第1节 小程序基础组件介绍

简介：介绍小程序基础组件有哪些

- 基础组件

- icon

```
<icon type="warn" size="100rpx" color="red"></icon>
```

- progress

```
<progress percent="30" show-info="true" border-radius="10" stroke-  
width="50rpx" activeColor="red" backgroundColor="#000" active="true">  
</progress>
```

- rich-text

```
<rich-text nodes="{{textStr}}"></rich-text>
```

- text

```
<text selectable="true">小程序</text>
```

- 官方文档：<https://developers.weixin.qq.com/miniprogram/dev/component/text.html>

### 第2节 小程序常用表单组件介绍（一）

简介：介绍小程序常用的表单组件

- 常用表单组件如下
  - button
  - checkbox
  - form
  - input
  - label
  - radio
  - textarea
- 表单组件<https://developers.weixin.qq.com/miniprogram/dev/component/button.html>

### 第3节 小程序常用表单组件介绍（二）

简介：介绍小程序常用的表单组件

- 常用表单组件如下
  - picker组件

```
<picker bindchange="bindPickerChange" value="{{index}}" range="{{array}}">
  <view class="picker">
    当前选择: {{array[index]}}
  </view>
</picker>
```

- slider组件

```
<slider min="50" max="200" show-value="true" step="5"></slider>
```

- switch组件

```
<switch bindchange="bindSwitchChange" checked="true"></switch>
```

### 第4节 小程序媒体组件介绍

简介：介绍小程序媒体组件

- audio组件



```
<audio id="myAudio" poster="{{poster}}" name="{{name}}" author="{{author}}" src="{{src}}" controls></audio>

<button bindtap="audipplay">播放</button>
<button bindtap="audippause">暂停</button>
<button bindtap="audioseek">设置进度条</button>
```

- video组件

```
<video src="https://www.w3school.com.cn/i/movie.ogg" controls></video>
```

- image组件

```
<image lazy-load="true" src="../../../images/sp2.png"></image>
```

## 第5节 小程序地图组件map

简介：介绍小程序地图组件map的使用

- map文档<https://developers.weixin.qq.com/miniprogram/dev/component/>

## 第6节 小程序画布组件canvas

简介：讲解小程序canvas的使用

- canvas文档<https://developers.weixin.qq.com/miniprogram/dev/component/canvas.html>

```
<canvas style="width:600rpx;height:400rpx;" canvas-id="canvas"></canvas>
```

```
onReady: function() {
  console.log("ready");
  var context = wx.createCanvasContext('canvas');

  context.setStrokeStyle("#ff0000");
  context.setLineWidth(2);
```

```
context.moveTo(160,100);
context.arc(100,100,60,0,2*Math.PI,true);
context.moveTo(140,100);
context.arc(100,100,40,0,Math.PI,false);
context.moveTo(85,80);
context.arc(80,80,5,0,2*Math.PI,true);
context.moveTo(125,80);
context.arc(120, 80, 5, 0, 2 * Math.PI, true);
context.stroke();
context.draw();
}
```



XD课堂 愿景："让编程不在难学，让技术与生活更加有趣" 更多课程请访问[xdclass.net](http://xdclass.net)

## 第五章 常用视图容器及路由跳转

### 第1节 使用视图容器swiper实现轮播图

简介：详细讲解小程序的滑动组件swiper

```
<swiper class="swiper" autoplay="true" interval="3000" duration="1000"
indicator-dots="true" indicator-active-color="#007aff" vertical="true"
circular="true" display-multiple-items="2">
  <block wx:for="{{imgs}}">
    <swiper-item>
      <image src="{{item}}" class="item_img"></image>
    </swiper-item>
  </block>
</swiper>
```

- swiper文档<https://developers.weixin.qq.com/miniprogram/dev/component/swiper.html>

### 第2节 深度讲解可滚动视图区域scroll-view

简介：讲解可滚动视图容器scroll-view的使用

- 横向滚动

```
<scroll-view scroll-x="true" style="white-space:nowrap;">
  <block wx:for="{{imgs}}">
    <image src="{{item}}"></image>
  </block>
</scroll-view>
```

- 纵向滚动

```
<scroll-view scroll-y="true" scroll-top="100" style="height:340rpx;"
scroll-into-view="{{toView}}" bindscrolltoupper="upper"
bindscrolltolower="lower">
  <block wx:for="{{imgs}}">
    <image src="{{item}}" id="{{'img'+index}}"></image>
  </block>
</scroll-view>
<button bindtap="gotoview">定位到第三张</button>
```

### 第3节 使用scroll-view实现可滚动导航栏

简介：详细讲解可滚动导航栏的实现

Wxml

```
<scroll-view scroll-x="true" class="nav" scroll-left="{{navScrollLeft}}">
  <block wx:for="{{navData}}" wx:key="index">
    <view class="nav_item {{currentNav == index?'active':''}}" data-current="{{index}}" bindtap="switchNav">{{item.text}}</view>
  </block>
</scroll-view>
```

点击移动scroll-view，添加active类

```
switchNav:function(e){
  // console.log(e);
  var cur = e.currentTarget.dataset.current;
  //tab选项居中
  var singleWidth = this.data.windowWidth/5;
  this.setData({
    navScrollLeft: (cur - 2)*singleWidth
```

```

});
console.log(this.data.navScrollLeft);
if(this.data.currentNav == cur) {
    return false;
}else{
    this.setData({
        currentNav: cur
    })
}
},

```

## 第4节 特殊覆盖组件cover-view及cover-image

简介：讲解特殊覆盖组件cover-view以及cover-image的使用

```

<video src="https://www.w3school.com.cn/i/movie.ogg" controls>
<cover-view class="shuiyin">
小D课堂
<cover-image src="../../../images/sp2.png"></cover-image>
</cover-view>
</video>

```

## 第5节 小程序页面导航组件

简介：讲解小程序页面导航组件

```

<navigator url="/pages/personal/personal">跳转到个人页面</navigator>

```

- 导航组件文档：<https://developers.weixin.qq.com/miniprogram/dev/component/navigator.html>

## 第6节 常用路由跳转方法

简介：讲解常用的路由跳转api

- wx.navigateTo()保留当前页面，跳转到应用内的某个页面。但是不能跳到 tabBar 页面
- wx.switchTab()跳转到 tabBar 页面，并关闭其他所有非 tabBar 页面
- wx.reLaunch()关闭所有页面，打开到应用内的某个页面
- wx.redirectTo()关闭当前页面，跳转到应用内的某个页面。但是不允许跳转到 tabBar 页面
- wx.navigateBack()关闭当前页面，返回上一页面或多级页面
- 路由跳转文档<https://developers.weixin.qq.com/miniprogram/dev/api/route/wx.navigateBack.html>

## 第7节 小程序页面的传参与取参

简介：介绍小程序页面之间的传参和取参

```
<navigator url="/pages/sort/sort?id=1&name=eric">跳转到分类</navigator>
```

## 第8节 如何实现小程序的底部导航栏

简介：如何编写小程序的tabbar

```
app.json
"tabBar": {
  "color": "#2b333b",
  "selectedColor": "#3bb149",
  "borderStyle": "black",
  "list": [{
    "pagePath": "pages/index/index",
    "text": "首页"
  }, {
    "pagePath": "pages/sort/sort",
    "text": "分类"
  }, {
    "pagePath": "pages/personal/personal",
    "text": "个人"
  }]
},
```



## 第六章 构建模板化与模块化小程序

### 第1节 利用require方法加载js模块文件

简介：如何利用require方法加载js模块

- require路径要写成相对路径

```
var util = require('../utils/util.js');
```

### 第2节 WXML模板编写与引入精讲

简介：讲解wxml模板编写以及引入方式

- 新建模板文件，wxml和wxss

```
<template name="courseList">
  <view class="item">
    <view class="item_img">
      <image src="{{course.img}}"></image>
    </view>
    <view class="item_name">{{course.name}}</view>
    <view>{{idx}}</view>
  </view>
</template>
```

- 在需要用的页面import模板文件

```
<view class="list">
  <block wx:for="{{list}}" wx:for-item="course" wx:for-index="idx">
    <template is="courseList" data="{{course}}"></template>
  </block>
</view>
```

## 第3节 讲解wxs模块引用

简介：讲解wxs模块的使用

```
<wxs src="../../wxs/common.wxs" module="info"></wxs>

<view>{{info.message}}</view>
<view>{{info.add(20,10)}}</view>
```

wxs文件

```
//JS代码
var msg = "hello world";
//commonjs模块化导出

var num = function (num1, num2) {
  return num1 + num2;
}
module.exports = {
  message: msg,
  add: num
}
```

## 第4节 wxs案例练习

简介：讲解wxs常用的一些案例

- 价格保留小数

```
var toFixed = function(value,count){
  return value.toFixed(count);
}
```

- 时间戳格式化

```
var formatNumber = function (n) {
  n = n.toString()
  return n[1] ? n : '0' + n
}

var regYear = getRegExp("(y+)", "i");
```

```

var dateFormat = function (timestamp, format) {
    if (!format) {
        format = "yyyy-MM-dd hh:mm:ss";
    }
    timestamp = parseInt(timestamp);
    var realDate = getDate(timestamp);
    function timeFormat(num) {
        return num < 10 ? '0' + num : num;
    }
    var date = [
        ["M+", timeFormat(realDate.getMonth() + 1)],
        ["d+", timeFormat(realDate.getDate())],
        ["h+", timeFormat(realDate.getHours())],
        ["m+", timeFormat(realDate.getMinutes())],
        ["s+", timeFormat(realDate.getSeconds())],
        ["q+", Math.floor((realDate.getMonth() + 3) / 3)],
        ["S+", realDate.getMilliseconds()],
    ];
    var reg1 = regYear.exec(format);
    // console.log(reg1[0]);
    if (reg1) {
        format = format.replace(reg1[1], (realDate.getFullYear() +
        '').substring(4 - reg1[1].length));
    }
    for (var i = 0; i < date.length; i++) {
        var k = date[i][0];
        var v = date[i][1];

        var reg2 = getRegExp("(" + k + ")").exec(format);
        if (reg2) {
            format = format.replace(reg2[1], reg2[1].length == 1
                ? v : ("00" + v).substring(("" + v).length));
        }
    }
    return format;
}

```



小D课堂 愿景："让编程不在难学，让技术与生活更加有趣" 更多课程请访问[xdclass.net](http://xdclass.net)

## 第七章 开始电影资讯小程序实战



## 第1节 小程序登录授权以及缓存storage讲解

简介：讲解小程序登录授权获取用户信息及缓存

```
// 登录
wx.login({
  success: res => {
    // 发送 res.code 到后台换取 openId, sessionKey, unionId
    console.log(res.code);
    wx.request({
      url: '',
      data: {
        code: res.code
      },
      method: 'get',
      success: function(res) {
        获取 openId, sessionKey, unionId
      }
    })
  }
})
```

数据缓存文档：<https://developers.weixin.qq.com/miniprogram/dev/api/storage/wx.setStorageSync.html>

登录文档：<https://developers.weixin.qq.com/miniprogram/dev/api/open-api/login/wx.login.html>

获取用户信息文档：<https://developers.weixin.qq.com/miniprogram/dev/api/open-api/user-info/wx.getUserInfo.html>

授权文档：<https://developers.weixin.qq.com/miniprogram/dev/api/open-api/authorize/wx.authorize.html>

<https://developers.weixin.qq.com/miniprogram/dev/framework/open-ability/authorize.html>

## 第2节 构建个人页面

简介：讲解个人页面的搭建

编写tabbar导航栏

```
"tabBar": {
  "color": "#515151",
  "selectedColor": "#42bd56",
```

```
"list": [{
  "pagePath": "pages/index/index",
  "text": "首页",
  "iconPath": "images/shouye.png",
  "selectedIconPath": "images/shouye_selected.png"
},{
  "pagePath": "pages/personal/personal",
  "text": "个人",
  "iconPath": "images/wode.png",
  "selectedIconPath": "images/wode_selected.png"
}]
}
```

通过e.detail.userInfo来判断是否点了允许授权

```
if(e.detail.userInfo){
  app.globalData.userInfo = e.detail.userInfo
  this.setData({
    userInfo: e.detail.userInfo,
    hasUserInfo: true
  })
}
```

tarbar文档: <https://developers.weixin.qq.com/miniprogram/dev/reference/configuration/app.html#tabBar>

canIuse文档: <https://developers.weixin.qq.com/miniprogram/dev/api/base/wx.canIUse.html>

### 第3节 讲解首页搜索框及轮播图布局

简介: 讲解首页搜索框及轮播图布局

```
bannerData:[{
  id: 1,
  banner_img:
"https://img3.doubanio.com/view/photo/l/public/p2563815623.webp"
}, {
  id: 2,
  banner_img:
"https://img3.doubanio.com/view/photo/l/public/p2564461744.webp"
},
{
  id: 3,
  banner_img:
"https://img3.doubanio.com/view/photo/s_ratio_poster/public/p2563630521.webp"
}
]
```

## 第4节 轮播图高度自适应方案

简介：详细讲解轮播图高度自适应方案实现

获取系统信息文档：<https://developers.weixin.qq.com/miniprogram/dev/api/base/system/system-info/wx.getSystemInfoSync.html>

image文档：<https://developers.weixin.qq.com/miniprogram/dev/component/image.html>

```
var winWidth = wx.getSystemInfoSync().windowWidth; //获取屏幕宽度
var imgWidth = e.detail.width; //图片宽度
var imgHeight = e.detail.height; //图片高度
this.setData({
  swiperHeight: winWidth * imgHeight / imgWidth + 'px'
})
```

## 第5节 深度讲解小程序request请求

简介：讲解小程序如何请求接口

接口地址: `http://op.juhe.cn/onebox/movie/pmovie`  
key: `'a8b932b7661aee98fcb3c89984626302'`  
city: `'城市名'`

接口封装

```
var host = 'http://op.juhe.cn';

function getData(url,method,data={},header={}) {
  return new Promise(function(resolve,reject){
    wx.request({
      url: host+url,
      data: data,
      method: method,
      header:header,
      success: function (res) {
        if(res.statusCode==200){
          resolve(res.data);
        }else{
          reject('error');
        }
      },
      fail: function(e){
        reject(e);
      }
    })
  })
}

module.exports = {
  getData: getData
}
```

## 第6节 构建电影分类专区列表

简介: 讲解电影分类专区的布局

- 专区数据

```
movieList: [{
  id: 1,
```

```

        movie_img:
"https://img3.doubanio.com/view/photo/s_ratio_poster/public/p2567841004.we
bp",
        name: "罗小黑战记",
        rate: 8.4
    },
    {
        id: 2,
        movie_img:
"https://img3.doubanio.com/view/photo/s_ratio_poster/public/p2563630521.we
bp",
        name: "烈火英雄",
        rate: 6.7
    },
    {
        id: 3,
        movie_img:
"https://img1.doubanio.com/view/photo/s_ratio_poster/public/p2566598269.we
bp",
        name: "徒手攀岩",
        rate: 9.0
    },
    {
        id: 4,
        movie_img:
"https://img3.doubanio.com/view/photo/s_ratio_poster/public/p2561542272.we
bp",
        name: "速度与激情",
        rate: 6.4
    },
    {
        id: 5,
        movie_img:
"https://img1.doubanio.com/view/photo/s_ratio_poster/public/p2566170919.we
bp",
        name: "检查方的罪人",
        rate: 7.0
    }
];

```

- scroll-view上使用display: flex失效，解决方法：可以在里层加多一个view包裹列表，在新添加的view上使用display:flex就可以了

## 第7节 深度讲解电影分类专区模板复用

简介：讲解电影列表template封装使用

```
<template name="movie_subject">
  <view class="subject">
    <view class="subject_top">
      <text class="title">最近上映</text>
      <navigator class="lookMore">查看更多 </navigator>
    </view>
    <scroll-view class="movieList" scroll-x="true">
      <block wx:for="{{movieList}}" wx:key="{{index}}">
        <template is="movieItem" data="{{item}}"></template>
      </block>
    </scroll-view>
  </view>
</template>

<template name="movieItem">
  <navigator class="movie_item">
    <view class="movie_img">
      <image src="{{item.movie_img}}"></image>
    </view>
    <view class="name">{{item.name}}</view>
    <view class="star_box">
      <view class="stars">
        <view class="iconfont icon-xingxing light"></view>
        <view class="iconfont icon-xingxing light"></view>
        <view class="iconfont icon-xingxing light"></view>
        <view class="iconfont icon-xingxing light"></view>
        <view class="iconfont icon-xingxing"></view>
      </view>
      <text class="star_score">{{item.rate}}</text>
    </view>
  </navigator>
</template>
```

## 第8节 星星评分组件的实现

简介：讲解星星组件编写以及评分格式函数功能

- 评分格式函数

```
function formatRate(v){
  var lightStar = parseInt(v)/2;
  var star = [];
  for(var i = 1;i <= 5;i++){
    if(i <= lightStar) {
      star.push(1);
    }else{
      star.push(0);
    }
  }
  return star;
}

module.exports = {
  formatRate: formatRate
}
```

## 第9节 分离业务数据及数据接入

简介：讲解业务数据分离以及数据的接入

```
//专区数据
var subJectData = [{
  subjectTitle: "影院热映",
  movieList: [{
    id: 1,
    movie_img:
    "https://img3.doubanio.com/view/photo/s_ratio_poster/public/p2567841004.webp",
    name: "罗小黑战记",
    rate: 8.4
  },
  {
    id: 2,
    movie_img:
    "https://img3.doubanio.com/view/photo/s_ratio_poster/public/p2563630521.webp",
```

```
        name: "烈火英雄",
        rate: 6.7
    },
    {
        id: 3,
        movie_img:
"https://img1.doubanio.com/view/photo/s_ratio_poster/public/p2566598269.webp",
        name: "徒手攀岩",
        rate: 9.0
    },
    {
        id: 4,
        movie_img:
"https://img3.doubanio.com/view/photo/s_ratio_poster/public/p2561542272.webp",
        name: "速度与激情",
        rate: 6.4
    },
    {
        id: 5,
        movie_img:
"https://img1.doubanio.com/view/photo/s_ratio_poster/public/p2566170919.webp",
        name: "检查方的罪人",
        rate: 7.0
    }
]
},
{
    subjectTitle: "热门电影",
    movieList: [{
        id: 1,
        movie_img:
"https://img3.doubanio.com/view/photo/s_ratio_poster/public/p2567841004.webp",
        name: "罗小黑战记",
        rate: 8.4
    },
    {
        id: 2,
        movie_img:
"https://img3.doubanio.com/view/photo/s_ratio_poster/public/p2563630521.webp",
        name: "烈火英雄",
        rate: 6.7
    },
    {
        id: 3,
        movie_img:
"https://img1.doubanio.com/view/photo/s_ratio_poster/public/p2566598269.webp",
        name: "徒手攀岩",
        rate: 9.0
    },
    }
```



```
{
  id: 4,
  movie_img:
"https://img3.doubanio.com/view/photo/s_ratio_poster/public/p2561542272.webp",
  name: "速度与激情",
  rate: 6.4
},
{
  id: 5,
  movie_img:
"https://img1.doubanio.com/view/photo/s_ratio_poster/public/p2566170919.webp",
  name: "检查方的罪人",
  rate: 7.0
}
]
},
{
  subjectTitle: "热门电视剧",
  movieList: [{
    id: 1,
    movie_img:
"https://img3.doubanio.com/view/photo/s_ratio_poster/public/p2567841004.webp",
    name: "罗小黑战记",
    rate: 8.4
  },
  {
    id: 2,
    movie_img:
"https://img3.doubanio.com/view/photo/s_ratio_poster/public/p2563630521.webp",
    name: "烈火英雄",
    rate: 6.7
  },
  {
    id: 3,
    movie_img:
"https://img1.doubanio.com/view/photo/s_ratio_poster/public/p2566598269.webp",
    name: "徒手攀岩",
    rate: 9.0
  },
  {
    id: 4,
    movie_img:
"https://img3.doubanio.com/view/photo/s_ratio_poster/public/p2561542272.webp",
    name: "速度与激情",
    rate: 6.4
  },
  {
    id: 5,
```

```
        movie_img:
"https://img1.doubanio.com/view/photo/s_ratio_poster/public/p2566170919.webp",
        name: "检查方的罪人",
        rate: 7.0
    }
]
}
```



小D课堂 愿景："让编程不在难学，让技术与生活更加有趣" 更多课程请访问[xdclass.net](http://xdclass.net)

## 第八章 搜索页面及查询更多的实现

---

### 第1节 电影搜索页面构建（上）

简介：讲解电影搜索页面构建

```
//更多电影数据
var moreData = [{
    id: 1,
    movie_img:
"https://img3.doubanio.com/view/photo/s_ratio_poster/public/p2567841004.webp",
    name: "罗小黑战记",
    rate: 8.4
```

```
},
{
  id: 2,
  movie_img:
  "https://img3.doubanio.com/view/photo/s_ratio_poster/public/p2563630521.webp",
  name: "烈火英雄",
  rate: 6.7
},
{
  id: 3,
  movie_img:
  "https://img1.doubanio.com/view/photo/s_ratio_poster/public/p2566598269.webp",
  name: "徒手攀岩",
  rate: 9.0
},
{
  id: 4,
  movie_img:
  "https://img3.doubanio.com/view/photo/s_ratio_poster/public/p2561542272.webp",
  name: "速度与激情",
  rate: 6.4
},
{
  id: 5,
  movie_img:
  "https://img1.doubanio.com/view/photo/s_ratio_poster/public/p2566170919.webp",
  name: "检查方的罪人",
  rate: 7.0
},
{
  id: 6,
  movie_img:
  "https://img3.doubanio.com/view/photo/s_ratio_poster/public/p2567841004.webp",
  name: "罗小黑战记",
  rate: 8.4
},
{
  id: 7,
  movie_img:
  "https://img3.doubanio.com/view/photo/s_ratio_poster/public/p2563630521.webp",
  name: "烈火英雄",
  rate: 6.7
},
{
  id: 8,
  movie_img:
  "https://img1.doubanio.com/view/photo/s_ratio_poster/public/p2566598269.webp",
  name: "徒手攀岩",
  rate: 9.0
}
```

```
},
{
  id: 9,
  movie_img:
  "https://img3.doubanio.com/view/photo/s_ratio_poster/public/p2561542272.webp",
  name: "速度与激情",
  rate: 6.4
},
{
  id: 10,
  movie_img:
  "https://img1.doubanio.com/view/photo/s_ratio_poster/public/p2566170919.webp",
  name: "检查方的罪人",
  rate: 7.0
},
{
  id: 11,
  movie_img:
  "https://img3.doubanio.com/view/photo/s_ratio_poster/public/p2567841004.webp",
  name: "罗小黑战记",
  rate: 8.4
},
{
  id: 12,
  movie_img:
  "https://img3.doubanio.com/view/photo/s_ratio_poster/public/p2563630521.webp",
  name: "烈火英雄",
  rate: 6.7
},
{
  id: 13,
  movie_img:
  "https://img1.doubanio.com/view/photo/s_ratio_poster/public/p2566598269.webp",
  name: "徒手攀岩",
  rate: 9.0
}
];
```

获取节点文档: <https://developers.weixin.qq.com/miniprogram/dev/api/wxml/wx.createSelectorQuery.html>

## 第2节 电影搜索页面构建（下）

简介：讲解电影搜索页面构建

- 通过bindconfirm="searchMovie"绑定搜索事件

```
<input class="search_input" bindconfirm="searchMovie" placeholder="搜索"
confirm-type="search"></input>
```

- input文档：<https://developers.weixin.qq.com/miniprogram/dev/component/input.html>
- confirm-type设置右下角按钮文本

## 第3节 动态更换电影分类标题

简介：讲解如何动态更换导航栏标题以及背景

```
//动态设置标题
wx.setNavigationBarTitle({
  title: options.title
});
//动态设置导航栏背景颜色
// wx.setNavigationBarColor({
//   frontColor: '#ffffff',
//   backgroundColor: '#000'
// });
//设置导航栏加载动画
// wx.showNavigationBarLoading();
// setTimeout(wx.hideNavigationBarLoading,2000)
```

导航栏设置文档：<https://developers.weixin.qq.com/miniprogram/dev/api/ui/navigation-bar/wx.setNavigationBarTitle.html>

## 第4节 构建查看更多页面

简介：讲解更多页面的布局构建

- 交互动画文档：<https://developers.weixin.qq.com/miniprogram/dev/api/ui/interaction/wx.showLoading.html>

```
<scroll-view scroll-y="true" style="height:100vh;">
  <view class="movie_list">
    <block wx:for="{{moreData}}" wx:key="{{index}}">
      <template is="movieItem" data="{{...item}}"></template>
    </block>
  </view>
  <view style="text-align:center;" hidden="{{hidden}}">加载中...</view>
</scroll-view>
```

## 第5节 深度讲解上拉加载功能实现

简介：讲解查看更多页面如何实现上拉加载功能

- 使用页面下拉刷新功能时需要在当前页面的json文件中加入enablePullDownRefresh

```
{
  "usingComponents": {},
  "enablePullDownRefresh": true
}
```

- scroll-view文档：<https://developers.weixin.qq.com/miniprogram/dev/component/scroll-view.html>
- 页面配置文档：<https://developers.weixin.qq.com/miniprogram/dev/reference/configuration/page.html>



小D课堂 愿景："让编程不在难学，让技术与生活更加有趣" 更多课程请访问[xdclass.net](http://xdclass.net)

## 第九章 电影详情页面构建

### 第1节 构建电影详情页面

简介：构建电影详情页面

## 第2节 电影详情页面数据接入

简介：电影详情页面数据接入

```
//宝贝详情数据
var detailData = {
  movie_img:
    "https://img3.doubanio.com/view/photo/s_ratio_poster/public/p2561542272.webp",
  name: "速度与激情",
  movie_category: "纪录片/美国/片长100分钟",
  rate: 6.4,
  description: "一个是美国外交安全局的忠诚特工、身材魁梧的执法者霍布斯（强森饰），一个是前英国军事特工精英、无法无天的恶棍肖（斯坦森饰）。在201...",
  actor: [{
    actorImg:
      "https://img3.doubanio.com/view/celebrity/s_ratio_celebrity/public/p1416221591.26.webp",
    name: "大卫·雷奇",
    role: "导演"
  },
  {
    actorImg:
      "https://img3.doubanio.com/view/celebrity/s_ratio_celebrity/public/p45621.webp",
    name: "道恩·强森",
    role: "饰 卢克·霍布斯 Luke Hobbs"
  },
  {
    actorImg:
      "https://img3.doubanio.com/view/celebrity/s_ratio_celebrity/public/p424.webp",
    name: "杰森·斯坦森",
    role: "饰 德卡·肖 Deckard Shaw"
  },
  {
    actorImg:
      "https://img3.doubanio.com/view/celebrity/s_ratio_celebrity/public/p1410696282.74.webp",
    name: "伊德里斯·艾尔巴",
    role: "饰 布雷克斯顿 Brixton"
  },
  {
    actorImg:
      "https://img3.doubanio.com/view/celebrity/s_ratio_celebrity/public/p1519127566.22.webp",
    name: "凡妮莎·柯比",
    role: "饰 哈蒂·肖 Hattie Shaw"
  }
}
```

```
]
};
```

### 第3节 小程序的版本上传

简介：讲解小程序如何上传版本以及包大小注意事项

- 目前小程序分包大小有以下限制：
  - 整个小程序所有分包大小不超过 8M
  - 单个分包/主包大小不能超过 2M
- iconfont在线地址

```
@font-face {
  font-family: 'iconfont'; /* project id 1372578 */
  src: url('//at.alicdn.com/t/font_1372578_mk8k0mkc9v.eot');
  src: url('//at.alicdn.com/t/font_1372578_mk8k0mkc9v.eot?#iefix')
  format('embedded-opentype'),
  url('//at.alicdn.com/t/font_1372578_mk8k0mkc9v.woff2') format('woff2'),
  url('//at.alicdn.com/t/font_1372578_mk8k0mkc9v.woff') format('woff'),
  url('//at.alicdn.com/t/font_1372578_mk8k0mkc9v.ttf') format('truetype'),
  url('//at.alicdn.com/t/font_1372578_mk8k0mkc9v.svg#iconfont') format('svg');
}
```

### 第4节 电影实战总结

简介：回顾电影实战知识点以及总结

- 小程序开发主要流程：
  1. 去微信公众平台注册账号（获取appid） ----- 公司：让小程序管理员开通开发权限
  2. 下载微信开发者工具
  3. 小程序开发
  4. 上传版本审核





## 第十章 组件化开发

---

### 第1节 手把手教你创建自定义组件

简介：如何使用自定义组件

- 注意细节
  - 因为 WXML 节点标签名只能是小写字母、中划线和下划线的组合，所以自定义组件的标签名也只能包含这些字符。
  - 自定义组件和页面所在项目根目录名不能以“wx-”为前缀，否则会报错。
- template一般是用在重复展示性场景，自定义组件通常用在业务逻辑比较复杂的场景
- 自定义组件文档：<https://developers.weixin.qq.com/miniprogram/dev/framework/custom-component/>

### 第2节 必备知识之组件之间引用及slot插槽

简介：讲解自定义组件之间的引用以及slot插槽的用法

- 自定义组件之间的引用，在想引用其他组件的当前组件的json文件里加上usingComponents

```
{
  "component": true,
  "usingComponents": {
    "my-comp2": "/component/my-comp2/my-comp2"
  }
}
```

- 支持多个slot，在组件的js文件中的options加入multipleSlots:true

```
//js
options: {
  multipleSlots: true // 在组件定义时的选项中启用多slot支持
}
```

```

},

//组件wxml
<view class="box">
  <text>这是我创建的第一个自定义组件</text>
  <slot name="slot1"></slot>
  <slot name="slot2"></slot>
  <my-conp2></my-conp2>
</view>

//页面wxml
<my-conp>
  <view slot="slot1">这是插槽1扩展的内容</view>
  <view slot="slot2">这是插槽2扩展的内容</view>
</my-conp>

```

slot插槽文档: <https://developers.weixin.qq.com/miniprogram/dev/framework/custom-component/wxml-wxss.html>

### 第3节 深度讲解组件的样式隔离

简介: 讲解组件与页面之间的样式影响

- 组件样式注意事项:
  - 组件和引用组件的页面不能使用id选择器 (`#a`)、属性选择器 (`[a]`) 和标签名选择器, 请改用class选择器。
  - 组件和引用组件的页面中使用后代选择器 (`.a .b`) 在一些极端情况下会有非预期的表现, 如遇, 请避免使用。
  - 子元素选择器 (`.a>.b`) 只能用于 `view` 组件与其子节点之间, 用于其他组件可能导致非预期的情况。
  - 继承样式, 如 `font`、`color`, 会从组件外继承到组件内。
  - 除继承样式外, `app.wxss` 中的样式、组件所在页面的的样式对自定义组件无效 (除非更改组件样式隔离选项)。

文档地址: <https://developers.weixin.qq.com/miniprogram/dev/framework/custom-component/wxml-wxss.html>

- 组件的样式和页面的样式默认情况下不相互影响
- 样式隔离配置

```
//组件js
options: {
  styleIsolation: 'shared'
},
```

`styleIsolation` 选项从基础库版本 [2.6.5](#) 开始支持。它支持以下取值：

- `isolated` 表示启用样式隔离，在自定义组件内外，使用 `class` 指定的样式将不会相互影响（一般情况下的默认值）；
- `apply-shared` 表示页面 `wxss` 样式将影响到自定义组件，但自定义组件 `wxss` 中指定的样式不会影响页面；
- `shared` 表示页面 `wxss` 样式将影响到自定义组件，自定义组件 `wxss` 中指定的样式也会影响页面和其他设置了 `apply-shared` 或 `shared` 的自定义组件。（这个选项在插件中不可用。）

## 第4节 深度讲解组件和页面通信

简介：讲解组件与页面之间的数据通信以及样式传递

- 页面传递给组件数据，通过在页面中设置属性值把数据传递到组件，组件通过`properties`获取数据

```
//页面
<my-conp3 dataA="{{title}}"></my-conp3>

//组件js
properties: {
  dataA: {
    type: String,
    value: '默认标题'
  }
},

//组件页面引用数据
<view class="title">{{dataA}}</view>
```

- 页面传递外部样式给组件，通过设置属性值传递到组件，组件通过`externalClasses`引入外部样式

```
//页面
<my-conp3 my-class="red"></my-conp3>

//组件js
externalClasses: ['my-class'],

//组件页面引入对应class样式
<view class="title my-class">{{dataA}}</view>
```

properties文档:<https://developers.weixin.qq.com/miniprogram/dev/reference/api/Component.html>

组件引入外部样式文档: <https://developers.weixin.qq.com/miniprogram/dev/framework/custom-component/wxml-wxss.html>

## 第5节 组件事件传递数据交互

简介: 讲解组件与页面之间通过事件传递数据

- 组件通过triggerEvent指定事件名, 将detail对象的数据传递到页面, 页面中通过 (bind+事件名: 执行函数) 来获取组件中传过来的数据

```
//组件js
this.triggerEvent('search',{
    searchText: e.detail.value
})

//页面
<my-conp4 bindsearch="search"></my-conp4>
```

组件通信与事件文档: <https://developers.weixin.qq.com/miniprogram/dev/framework/custom-component/events.html>

## 第6节 组件tab标签栏实战演练

简介: 讲解tab标签栏实战

- index页面

```
//页面
<my-tab bindtab="itemClick" tabArr="{{cates}}"></my-tab>
```

- 组件页面

```
<view class="tab-box">
  <block wx:for="{{tabArr}}" wx:key="index">
    <view class="tab-item {{currentIndex == index?'active':''}}"
    bindtap="tabClick" data-index="{{index}}">
      <text>{{item}}</text>
    </view>
  </block>
</view>
```

- 组件js

```
// component/my-tab/my-tab.js
Component({
  /**
   * 组件的属性列表
   */
  properties: {
    tabArr: {
      type: Array,
      value: []
    }
  },

  /**
   * 组件的初始数据
   */
  data: {
    currentIndex: 0
  },

  /**
   * 组件的方法列表
   */
  methods: {
    tabClick(e) {
      this.setData({
        currentIndex: e.currentTarget.dataset.index
      })
      console.log(this.data.currentIndex);
      this.triggerEvent('tab', {
        tab_index: this.data.currentIndex,
        title: this.data.tabArr[this.data.currentIndex]
      })
    }
  }
})
```

```
}  
}  
})
```

## 第7节 组件对象动态设置组件数据

简介：讲解通过组件对象设置组件数据

- 父组件直接获取子组件的数据或者方法，通过this.selectComponent(选择器)

```
<my-count id="count"></my-count>  
count:function(){  
    // console.log(this.selectComponent('#count'))  
    this.selectComponent('#count').increase(10);  
    console.log(this.selectComponent('#count').data.counter)  
}
```

## 第8节 讲解组件的生命周期

简介：讲解组件的生命周期以及组件监听页面生命周期

- 组件生命周期文档：<https://developers.weixin.qq.com/miniprogram/dev/framework/custom-component/lifetimes.html>
- created生命周期里不能调用this.setData，数据会无法修改
- 初始化data数据可在attached生命周期中处理
- ready生命周期可进行一些节点操作

## 第9节 讲解组件的数据监听器

简介：讲解组件的数据监听器observers

- observers文档: <https://developers.weixin.qq.com/miniprogram/dev/framework/custom-component/observer.html>
  - 数据监听器监听的是 setData 涉及到的数据字段, 即使这些数据字段的值没有发生变化, 数据监听器依然会被触发。
  - 如果在数据监听器函数中使用 setData 设置本身监听的数据字段, 可能会导致死循环, 需要特别留意。
- properties的observer文档: <https://developers.weixin.qq.com/miniprogram/dev/reference/api/Component.html>

```
dataA:{
  type:String,
  value:'xxxxx',
  observer:function(newVal,oldVal,change){
    //newVal是新的值, oldVal是老的值, change是数据的来源
    console.log(newVal, oldVal, change)
  }
}
```

## 第10节 深度讲解组件的代码共享

简介: 讲解组件的共享代码behavior

- 字段覆盖和组合规则
  - 同名的属性或方法, 组件本身的属性或方法会覆盖 behavior 中的属性或方法, 引用了多个 behavior, 后面的 behavior 会覆盖前面的属性或方法;
  - 如果有同名的数据字段, 如果数据是对象类型, 会进行对象合并, 如果是非对象类型则会进行相互覆盖;
  - 生命周期函数不会相互覆盖, 而是在对应触发时机被逐个调用。如果同一个 behavior 被一个组件多次引用, 它定义的生命周期函数只会被执行一次。
- 创建behavior实例

```
//创建behavior实例
let comm = Behavior({
  properties:{
    dataA: {
      type:Number,
      value:1
    }
  },
  },
```

```

data:{
  title:'这是共享的标题',
  name:'xiaoming'
},
methods:{
  conslog:function(){
    console.log('这是共享的方法');
  }
},
ready:function(){
  console.log('这是共享的ready')
}
});

module.exports = comm;

```

- 组件js文件中通过behaviors引入写好的behavior

```

// component/my-behavior/my-behavior.js
let myBehavior = require('../behavior.js');
Component({
  behaviors: [myBehavior],
  /**
   * 组件的属性列表
   */
  properties: {
    dataA:{
      type:String,
      value:'hhhhh'
    }
  },

  /**
   * 组件的初始数据
   */
  data: {
    name:'eric'
  },
  ready: function(){
    console.log(this.data.title)
    this.conslog();
  },
  /**
   * 组件的方法列表
   */
  methods: {
    setName:function(){
      this.setData({
        name:'名字'
      })
    }
  }
})

```



```
    })  
  }  
}  
})
```



小D课堂 愿景："让编程不在难学，让技术与生活更加有趣" 更多课程请访问[xdclass.net](http://xdclass.net)

## 第十一章 Weui组件库

---

### 第1节 WeUI框架介绍、下载及部署

简介：讲解小程序weui组件库的下载、npm构建

1. 到项目根目录下npm init初始化项目创建package.json文件

```
ericdeMacBook-Pro:weuiProject ericxie$ npm init
This utility will walk you through creating a package.json file.
It only covers the most common items, and tries to guess sensible defaults.

See `npm help json` for definitive documentation on these fields
and exactly what they do.

Use `npm install <pkg>` afterwards to install a package and
save it as a dependency in the package.json file.

Press ^C at any time to quit.
package name: (weuiproject)
version: (1.0.0)
description:
entry point: (app.js)
test command:
git repository:
keywords:
author:
license: (ISC)
About to write to /Users/ericxie/WeChatProjects/weuiProject/package.json:
```

2. 安装weui包，命令是npm install --save --production weui-miniprogram

```
ericdeMacBook-Pro:weuiProject ericxie$ npm install --save weui-miniprogram --production
npm notice created a lockfile as package-lock.json. You should commit this file.
npm WARN weuiproject@1.0.0 No description
npm WARN weuiproject@1.0.0 No repository field.

+ weui-miniprogram@0.1.10
added 1 package from 1 contributor in 3.965s
ericdeMacBook-Pro:weuiProject ericxie$
```

- 3.构建npm，在工具栏中点击构建npm

- 4.需要在详情的本地配置下勾选使用npm模块

weui组件库安装文档：<https://developers.weixin.qq.com/miniprogram/dev/extended/weui/quickstart.html>

## 第2节 WeUI组件使用介绍

简介：介绍weui组件的使用

- 在app.js中引入weui-wxss文件

```
//app.wxss
@import '/miniprogram_npm/weui-miniprogram/weui-
wxss/dist/style/weui.wxss';
```

- 在json文件引入组件

```
{
  "usingComponents": {
    "mp-searchbar": "/miniprogram_npm/weui-
miniprogram/searchbar/searchbar"
  }
}
```

- 页面使用组件

```
//index.wxml
<mp-searchbar ext-class="search" placeholder="请输入关键字搜索"
bindselectresult="selectResult" search="{{search}}"></mp-searchbar>

//index.js
//index.js
//获取应用实例
const app = getApp()

Page({
  data: {

  },
  onLoad: function () {
    this.setData({
      search: this.search.bind(this)
    })
  },
  search: function (value) {
    return new Promise((resolve, reject) => {
      setTimeout(() => {
        resolve([{ text: '搜索结果', value: 1 }, { text: '搜索结果2', value:
2 }])
      }, 200)
    })
  },
  selectResult: function (e) {
    console.log('select result', e.detail)
  },
})
```

搜索组件文档: <https://developers.weixin.qq.com/miniprogram/dev/extended/weui/search.html>

### 第3节 工具类库computed的使用

简介: 讲解实用的工具类库

- weui的navigation组件覆盖原生组件导航栏, 需要在页面配置 `navigationStyle` 为 `custom`

```
{
  "usingComponents": {
    "mp-searchbar": "/miniprogram_npm/weui-miniprogram/searchbar/searchbar",
    "mp-navigation-bar": "/miniprogram_npm/weui-miniprogram/navigation-bar/navigation-bar"
  },
  "navigationStyle": "custom"
}
```

- **watch 和小程序基础库本身的 observers 有什么区别?**

- 无论字段是否真的改变, `observers` 都会被触发, 而 `watch` 只在字段值改变了的时候触发, 并且触发时带有参数。
- 目前 `watch` 还不支持 `**` 通配符。

工具类库computed文档: <https://developers.weixin.qq.com/miniprogram/dev/extended/utils/computed.html>



小D课堂

愿景: "让编程不在难学, 让技术与生活更加有趣" 更多课程请访问[xdclass.net](http://xdclass.net)

## 第十二章 扩展内容之小程序常用api

---

## 第1节 小程序发起微信支付

简介：讲解小程序如何发起微信支付

```
wx.request({
  url: '支付接口地址',
  method: 'get',
  data: {
    openId: openId
  },
  success: function(res) {
    wx.requestPayment({
      timeStamp: res.timeStamp,
      nonceStr: res.nonceStr,
      package: res.package,
      signType: 'MD5',
      paySign: res.paySign,
      success(res) { },
      fail(res) { }
    })
  }
})
```

## 第2节 小程序获取收货地址

简介：讲解小程序如何获取收货地址

```
getAddress: function() {
  wx.chooseAddress({
    success: (res) => {
      console.log(res);
    }
  })
}
```

获取收货地址文档：<https://developers.weixin.qq.com/miniprogram/dev/api/open-api/address/wx.chooseAddress.html>

### 第3节 不可不知的tabbar操作

简介：讲解tabbar的一些实用api

- tarbar文档：<https://developers.weixin.qq.com/miniprogram/dev/api/ui/tab-bar/wx.showTabBarRedDot.html>

### 第4节 小程序图片应用api

简介：讲解常见的图片api操作

- 图片api文档：<https://developers.weixin.qq.com/miniprogram/dev/api/media/image/wx.saveImageToPhotosAlbum.html>
- 文件下载文档：<https://developers.weixin.qq.com/miniprogram/dev/api/network/download/wx.downloadFile.html>

```
//保存图片
saveImg:function(){
  wx.downloadFile({

    url:'https://img3.doubanio.com/view/movie_gallery_frame_hot_rec/normal/public/bf6e4213b728840.jpg',
    success:function(res){
      console.log(res)
      wx.saveImageToPhotosAlbum({
        filePath: res.tempFilePath,
        success:function(res){
          console.log('保存成功')
        }
      })
    }
  })
}

//选择图片
chooseImg: function() {
  //从本地选择图片
  wx.chooseImage({
    count: 5,
    sizeType: ['original', 'compressed'],
    sourceType: ['album', 'camera'],
```

```
    success: function(res) {  
      console.log(res)  
    },  
  })  
},
```

## 第5节 常用功能之小程序分享转发

简介：讲解小程序的分享转发

- 监听转发事件文档：<https://developers.weixin.qq.com/miniprogram/dev/reference/api/Page.html#onShareAppMessage-Object-object>

```
onShareAppMessage: function(res) {  
  if(res.from == 'button') {  
    console.log(res.target);  
  }  
  
  return {  
    title: '我是转发标题',  
    path: '/pages/index/index?id=1',  
  
    imageUrl: 'https://img3.doubanio.com/view/movie_gallery_frame_hot_rec/normal/public/bf6e4213b728840.jpg'  
  }  
}
```

- 转发api文档：<https://developers.weixin.qq.com/miniprogram/dev/api/share/wx.updateShareMenu.html>
- 通过按钮转发，设置open-type为share

```
<button open-type="share">转发</button>
```

