

微信小程序

1.第一节 课程准备

1.1 课程简介

1.2 课程目标

1.3 小程序的适用性

1.4 开发文档及开发工具

1.5 知识储备

1.6 变量的声明

1.7 变量作用域

1.8 创建一个小程序

1.9 小程序文件

1.9.1.小程序全局逻辑文件-->app.js

1.9.2.小程序全局配置文件-->app.json

1.9.3.小程序全局样式文件-->app.wxml

1.9.4.小程序文件结构

1.9.5.小程序页面文件功能

wxml文件

2.第二节 UI界面

2.1 view组件和text组件

2.2 图片组件

2.3 rpx自适应尺寸单位

2.4 弹性布局

2.5 固比模型

2.6 button组件

2.7 input组件

2.8 案例:成绩统计

2.9 案例:计算器

3.第三章 小程序扩展知识

3.1 ES6基础

4.第四章 表单组件

4.1 switch组件

4.2 radio组件

4.3 checkbox组件

4.4 label组件

4.5 textarea组件

4.6 Form组件

4.7 picker组件

4.8 picker-view组件

4.9 案例:省市区信息选择

4.10 navigator组件

4.11 swiper组件

5.第五章 交互反馈和多媒体组件

5.1 wx.showLoding

5.2 wx.showModel

5.3 底部弹出栏 wx.showActionSheet

5.4 wx.showToast

5.5 audio组件

5.6 音乐播放接口函数

5.7 video组件

5.8 video组件之接口函数

5.9 案例

6.第六章 数据存储及后台交互

6.1 同步存储

6.2 缓存多条数据

6.3 异步缓存

6.4 网络请求

6.5 案例:英文励志文案

7.第七章 使用手机设备

7.1 选择图片

7.2 saveFile

7.3 预览图片

- 7.4 授权相关
- 7.5 保存图片
- 7.6 录音
- 7.7 播放录音
- 7.8 获取系统信息
- 7.9 获取网络状态
- 7.10 获取地理位置
- 8.第八章 综合案例
 - 8.1 搭建云开发
 - 8.2 添加数据
 - 8.3 查询数据
 - 8.4 删除数据
 - 8.5 更新数据
 - 8.6 上传文件

1.第一节 课程准备

1.1 课程简介

小程序的优势:

- 1.使用群体庞大，普及率高
- 2.体积小，加载快，无需安装（2M以内）
- 3.寄生于微信，跨平台应用
- 4.开发成本低，开发流程简单

1.2 课程目标

- 1.看懂开发文档
- 2.熟悉小程序文件结构和功能
- 3.掌握常用的组件和API
- 4.学会常规开发模式
- 5.了解调用外部资源的开发模式

1.3 小程序的适用性

什么是微信小程序？

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，用户扫一扫或搜一下即可打开应用。微信小程序是一种寄生于微信，跨平台的应用，开发成本低，开发流程简单。

微信小程序适合做什么样的应用？

- 1.业务逻辑简单
- 2.使用评率低
- 3.性能要求低

1.4 开发文档及开发工具

开发文档:

微信开放文档 – 小程序

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

开发工具:

- 1.微信开发者工具

1.5 知识储备

微信小程序开发需要的基础

html css JavaScript ---> 必须会

php python java mysql 了解

微信小程序开发和web开发的区别:

微信小程序使用数据绑定的方式（数据优先）,实现页面中数据的加载。

数据一旦有变化，页面会重新渲染。类似于vue.js框架

微信小程序不可以使用JavaScript组件库，没有window的对象和document对象。应使用原生js

微信小程序大量使用ES6规范的JavaScript代码

2011年，ECMAScript 5.1版发布。之前我们大部分人用的也就是ES5

2015年6月，ECMAScript6正式通过，成为国际标准。

ES6,ES7,ES8...统称为ES6

1.6 变量的声明

尽可能用var去申明变量

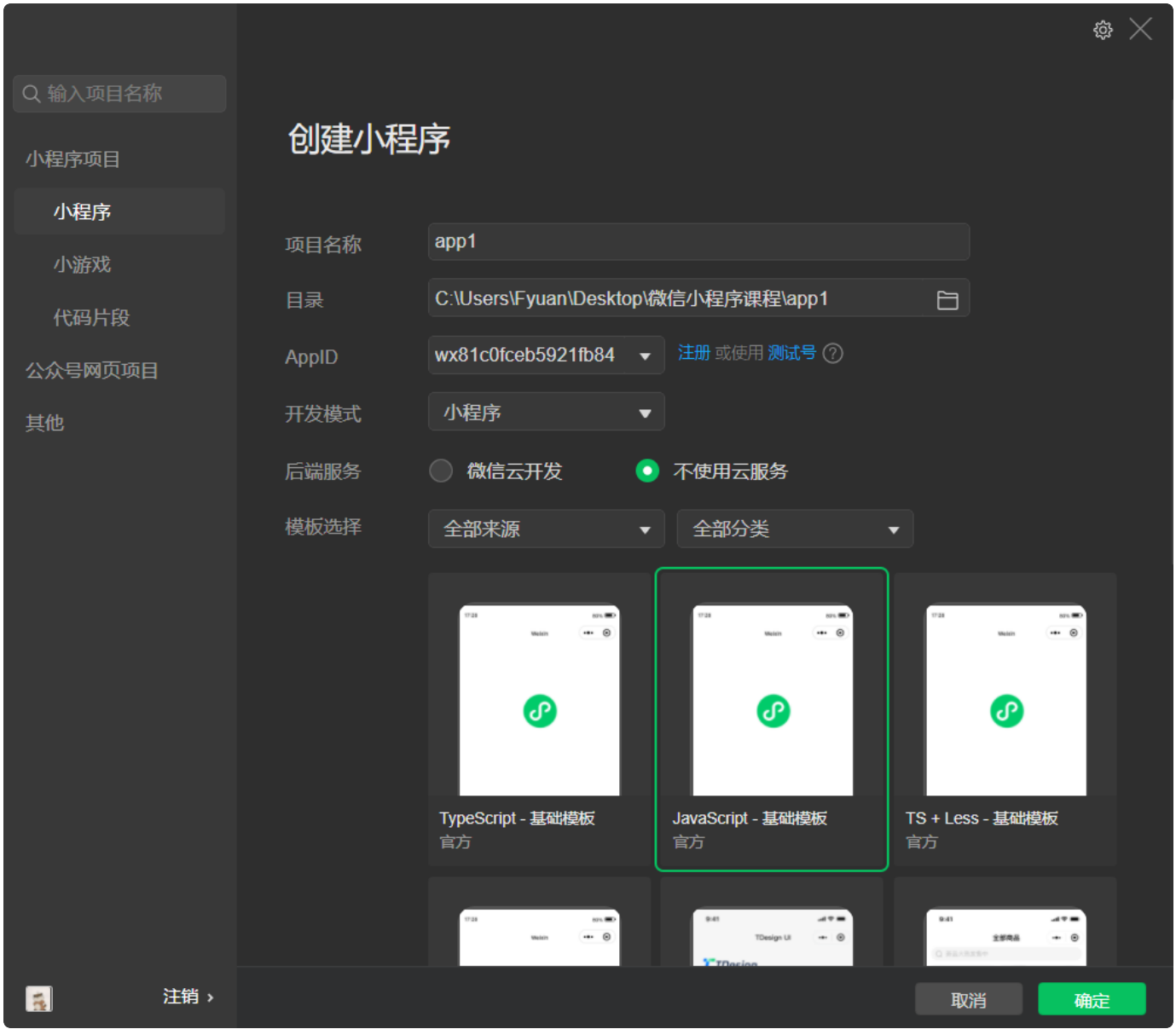
```
1  var a = 666;
2  let a = 666;
3  const a = 666;
4
5  // var声明变量:
6      // 1.有变量提升
7      // 2.可以重复声明变量
8      // 3.可以声明变量在赋值
9      // 4.可以修改值
10     // 5.可以跨快访问,不可以跨函数访问
11
12 // let声明变量:
13     // 1.有变量提升
14     // 2.可以重复声明变量
15     // 3.可以声明变量在赋值
16     // 4.可以修改值
17     // 5.不可以跨快访问,不可以跨函数访问
18
19 // const声明变量:
20     // 1.有变量提升
21     // 2.不可以重复声明变量
22     // 3.声明变量时必须赋值
23     // 4.不可以修改值(用于声明变量)
24     // 5.不可以跨快访问,不可以跨函数访问
```

1.7 变量作用域

ES5中作用域有：全局作用域、函数作用域。没有块作用域的概念。

ES6中新增了块级作用域。块作用域由{}包括，if语句和for语句里面的{}也属于块作用域。

1.8 创建一个小程序



1.9 小程序文件

1.9.1.小程序全局逻辑文件-->app.js

暂时没用到

1.9.2.小程序全局配置文件-->app.json

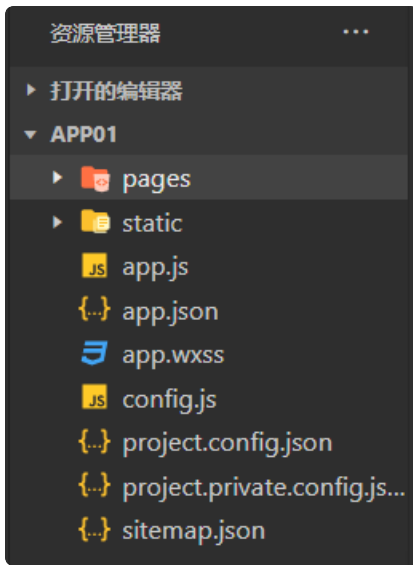
pages	小程序页面创建
window	小程序外观设置
tabBar	小程序底部导航

```
1  {
2    "pages": [
3      "pages/index/index",
4      "pages/test01/test01",
5      "pages/test02/test02",
6    ],
7    "window": {
8      "backgroundTextStyle": "light",      # 小程序的背景文字颜色
9      "navigationBarBackgroundColor": "#4cd0f1", # 小程序导航栏背景颜色
10     "navigationBarTitleText": "WeChat", # 小程序导航栏文字
11     "navigationBarTextStyle": "black" # 小程序导航栏字体颜色
12   },
13
14   "tabBar": {
15     "color": "#000000",
16     "selectedColor": "#000000",
17     "list": [
18       {
19         "pagePath": "pages/test01/test01",
20         "iconPath": "static/icon/cp.png",
21         "selectedIconPath": "static/icon/cp1.png",
22         "text": "全部产品"
23       },
24       {
25         "pagePath": "pages/test02/test02",
26         "iconPath": "static/icon/dd.png",
27         "selectedIconPath": "static/icon/dd1.png",
28         "text": "订单"
29       }
30     ],
31   },
32
33   "style": "v2",
34   "sitemapLocation": "sitemap.json"
35 }
36
```

1.9.3.小程序全局样式文件-->app.wxml

如果在这个文件中添加样式的话,整个小程序都将按照这个文件设置的样式

1.9.4.小程序文件结构



1.9.5.小程序页面文件功能

wxml文件

2.第二节 UI界面

2.1 view组件和text组件

容器 div = view 具有块级元素特点

文本 span = text 具有行级元素特点

2.2 图片组件

图片 img = image 具有行内块级元素特点

image不设置宽高时，有默认宽高

image可以只写组件头，不写组件尾（单标签形式）。应在组件头用"/"关闭。

背景图片在行内样式时可以加载本地图片，写到wxss中只能加载网络图片

2.3 rpx自适应尺寸单位

可以在组件头中写行内样式，也可在wxss中写样式（自动引入）

rpx: 自适应屏幕宽度尺寸。规定屏幕宽为750份，每份为1rpx。

在 iPhone6 上，屏幕宽度为375px，共有750个物理像素， $1\text{rpx} = 0.5\text{px} = 1\text{物理像素}$

2.4 弹性布局

1.任何容器都可以指定为Flex布局，任何子组件都可以设置宽高

2.父组件

设置成弹性盒子：display: flex;

指定主轴方向：flex-direction: column;

指定是否换行：flex-wrap: wrap;

指定在主轴对齐方式：justify-content: center|space-between|space-around;

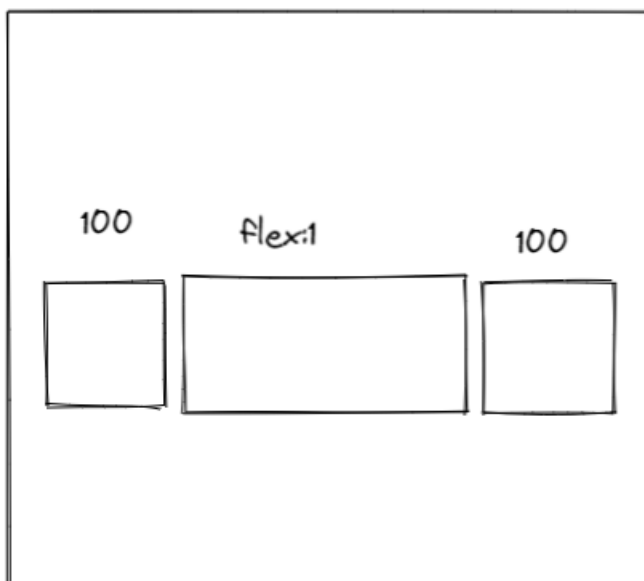
指定在交叉轴对齐方式: align-items: center|flex-end;

3.子组件

flex:定义了项目的缩放比例

2.5 固比模型

按照比例进行自动填充



2.6 button组件

详细见:

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

2.7 input组件

详细见:

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

2.8 案例:成绩统计

<div>..... WeChat 2G 17:14 80% WeChat</div> <div>请输入你的名字</div> <div>提交</div>	<div>..... WeChat 2G 17:33 90% WeChat</div> <div>loop</div> <div>78</div> <div>87</div> <div>提交</div> <div>姓名:loop 语文成绩:78 数学成绩:87 平均分:82.5</div>
--	---

2.9 案例:计算器



3.第三章 小程序扩展知识

3.1 ES6基础

需要有一定的JavaScript编程基础

4.第四章 表单组件

4.1 switch组件

详细见:

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

4.2 radio组件

4.3 checkbox组件

4.4 label组件

4.5 textarea组件

4.6 Form组件

4.7 picker组件

4.8 picker-view组件

4.9 案例:省市区信息选择



4.10 navigator组件

4.11 swiper组件

5.第五章 交互反馈和多媒体组件

5.1 wx.showLoding

显示 loading 提示框, 需主动调用 wx.hideLoading 才能关闭提示框

▼

HTML |

```
1 <button type="primary" bindtap="test1">刷新</button>
```

▼

HTML |

```
1 test1(){
2     // 显示加载...
3     wx.showLoading({
4         title: '确认中...',
5     })
6
7     //关闭加载,1.5s关闭加载
8     setTimeout(function(){
9         //定时结束之后要做的事情
10        //关闭加载框...
11        wx.hideLoading({
12            success: (res) => {},
13        })
14    },300)
15
16 }
```

5.2 wx.showModel

显示模态对话框

参数	数据类型	内容
title	string	提示的标题
content	string	提示的内容
showCancel	boolean	是否显示取消按钮,默认为true
cancelText	string	取消按钮的文字
cancelColor	hexColor	取消按钮的文字颜色
confirmText	string	确定按钮的文字
confirmColor	hexColor	确定按钮的文字颜色

▼ wxml

HTML |

1 <button type="primary" bindtap="test2">点击弹窗</button>


```
1  test2(){
2      //展示对话框
3      wx.showModal({
4          title: '确实收货?',
5          content: '收货将在3个工作日内完成',
6          showCancel: true,
7          cancelText: '取消',
8          cancelColor: 'red',
9          confirmText: '确认',
10         confirmColor: 'green',
11         //成功之后要做什么事情
12         success: function(e){
13             //获取用户信息(重要的是:用户的昵称和头像)
14             wx.getUserProfile({
15                 desc: 'desc',
16                 success: function(e){
17                     //用户的昵称
18                     console.log(e.userInfo.nickName)
19                     //用户的头像
20                     console.log(e.userInfo.avatarUrl)
21                 }
22             })
23         }
24     })
25 }
26 }
```

5.3 底部弹出栏 wx.showActionSheet

```
1  <button type="primary" bindtap="test3">底部弹出</button>
```

```
1 test3(){
2     wx.showActionSheet({
3         itemList: ['分享到QQ','分享到朋友圈','分享到微博'],
4         //成功之后...
5         success:function(e){
6             //根据itemList列表里面填写的内容,会有对应的下表 e.tapIndex
7             if (e.tapIndex==0) {
8                 console.log('成功分享到QQ')
9             }else if (e.tapIndex==1) {
10                 console.log('成功分享到朋友圈')
11             }else if (e.tapIndex==2) {
12                 console.log('成功分享到微博')
13             }
14         }
15     })
16 }
```

5.4 wx.showToast

显示消息提示框

参数	数据类型	内容
title	string	提示的内容
icon	string	图标
duration	number	提示的延迟时间
image	string	图标图片

```
1 <button type="primary" bindtap="test4">消息提示</button>
```

```

1  test4(){
2      wx.showToast({
3          title: '已完成',
4          icon: 'none',
5          image: '../static/icon/我的.png',
6          duration: 800
7      })
8  }

```

5.5 audio组件

音频

```

1  <audio src="{{src}}" poster="{{poster}}" controls="{{controls}}" loop="{{loop}}" name="{{name}}" author="{{author}}"></audio>

```

```

1  data: {
2      src: '../static/music/a1.mp3',
3      poster: '../static/images/poster.png',
4      controls: true,
5      loop: true,
6      name: '若把你',
7      author: 'Kirsty刘瑾睿'
8  }

```

```

1  /* 设置audio屏幕占满 */
2  audio{
3      display: flex;
4      flex-direction: column;
5  }

```

5.6 音乐播放接口函数

wx.createInnerAudioContext(Object object)



HTML |

```
1 var create = wx.createInnerAudioContext({useWebAudioImplement: true})
2 create.src = "http://ws.stream.qqmusic.qq.com/M500001VfvsJ21xFqb.mp3?guid=fff
  fffff82def4af4b12b3cd9337d5e7&uin=346897220&vkey=6292F51E1E384E061FF02C31
  F716658E5C81F5594D561F2E88B854E81CAAB7806D5E4F103E55D33C16F3FAC506D1AB172DE
  8600B37E43FAD&fromtag=46"
3 create.play() //播放
4 create.pause()//暂停
5 create.stop()//停止
```

5.7 video组件



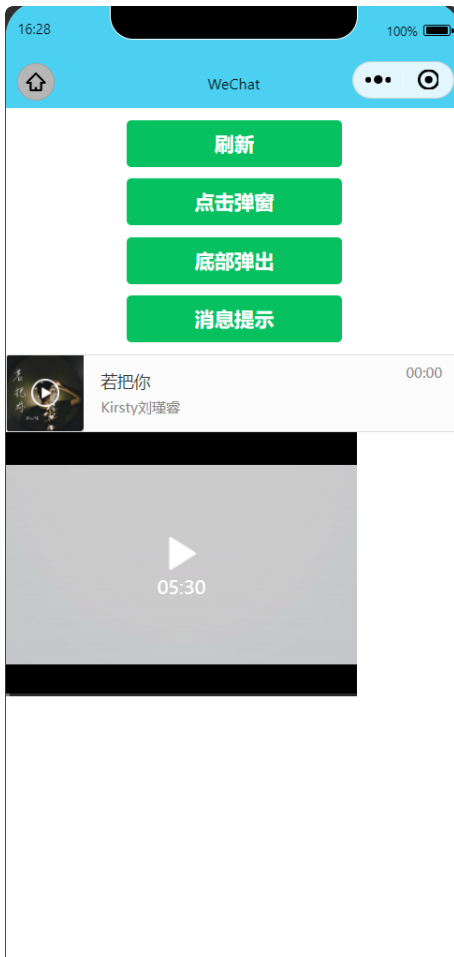
HTML |

```
1 <video src="http://wxsnsdy.tc.qq.com/105/20210/snsdyvideodownload?filekey=3
  0280201010421301f0201690402534804102ca905ce620b1241b726bc41dcff44e002040128
  82540400&bizid=1023&hy=SH&fileparam=302c020101042530230204136ffd93020457e3c
  4ff02024ef202031e8d7f02030f42400204045a320a0201000400" controls="true"></vi
  deo>
2
```

5.8 video组件之接口函数

wx.createVideoContext('id', component)

5.9 案例



6.第六章 数据存储及后台交互

6.1 同步存储

`wx.setStorageSync(string key, any data)`

功能描述---->将数据存储在本机缓存中指定的 key 中。会覆盖掉原来该 key 对应的内容。除非用户主动删除或因存储空间原因被系统清理，否则数据都一直可用。单个 key 允许存储的最大数据长度为 1MB，所有数据存储上限为 10MB。

▼ wxml

HTML |

```
1 <view>
2   <input bindinput="test1" type="text" placeholder="请输入要保存的内容" />
3   <view class="item" bindtap="history">查看历史</view>
4   <button bindtap="test2" type="primary">保 存</button>
5 </view>
```

▼ wxss

HTML |

```
1 page{
2   background: #f7f7f7;
3 }
4 input{
5   background: white;
6   height: 85rpx;
7   padding-left: 20rpx;
8 }
9
10 .item{
11   color: #3380ed;
12   margin-left: 550rpx;
13 }
14 view{
15   padding: 20rpx;
16 }
```

```
1  data: {
2      data: ""
3  },
4      // 获取输入框的内容
5      test1: function(e) {
6          // e.detail.value ---> 输入框的内容
7          // console.log(e.detail.value)
8          // 设置数据
9          this.setData({
10              data: e.detail.value
11          })
12      },
13      // 保存输入框的内容到本地当中
14      test2: function(e) {
15          wx.setStorageSync('test1', this.data.data)
16      },
17      // 跳转到查看历史页面
18      history: function(e) {
19          wx.navigateTo({
20              url: '../test19/test19',
21          })
22      },
```

```
1  <text>{{test1}}</text>
```

```
1  data: {  
2      test1:""  
3  },  
4  
5  /**  
6   * 生命周期函数--监听页面加载  
7   */  
8  onLoad(options) {  
9      // 当页面一加载,就获取存放的数据--> test1  
10     this.setData({  
11         test1:wx.getStorageSync('test1')  
12     })  
13 }
```

6.2 缓存多条数据


```
1  data: {
2      data1:"", //输入框的内容
3  },
4      //获取输入框的内容
5  test1:function(e){
6      // e.detail.value ---> 输入框的内容
7      //console.log(e.detail.value)
8      // 设置数据
9      this.setData({
10         data1:e.detail.value
11     })
12 },
13 // 保存输入框的内容到本地当中
14 test2:function(e){
15     // 1.同步存储一条数据
16     wx.setStorageSync('test1', this.data.data1)
17
18     // 2.同步存储多数据
19     var test2 = wx.getStorageSync('test2') || []    // 先去小程序中去找一
20     下有没有叫做test2的数据,有的话就可以找到,没有找到的话,就为空[]
21     test2.push(this.data.data1) //将后续输入框输入的内容存放进去
22     wx.setStorageSync('test2', test2)
23 },
24 // 跳转到查看历史页面
25 history:function(e){
26     wx.navigateTo({
27         url: '../test19/test19',
28     })
29 },
30
31 /**
32  * 生命周期函数--监听页面加载
33  */
34 onLoad(options) {
35     //清空缓存数据
36     // wx.clearStorageSync()
37 },
```

```
1  data: {
2      test1:"",
3      test2:[]
4  },
5
6  /**
7   * 生命周期函数--监听页面加载
8   */
9  onLoad(options) {
10     // 当页面一加载,就获取存放的数据--> test1
11     this.setData({
12         test1:wx.getStorageSync('test1'),
13         test2:wx.getStorageSync('test2')
14     })
15 }
```

```
1  <text>{{test1}}</text>
2
3  <view>-----</view>
4
5  <text wx:for="{{test2}}">{{item}}</text>
```

6.3 异步缓存

```
1  data: {
2      data1:"", //输入框的内容
3  },
4      //获取输入框的内容
5  test1:function(e){
6      // e.detail.value ---> 输入框的内容
7      //console.log(e.detail.value)
8      // 设置数据
9      this.setData({
10         data1:e.detail.value
11     })
12 },
13 // 保存输入框的内容到本地当中
14 test2:function(e){
15     // 1.同步存储一条数据
16     wx.setStorageSync('test1', this.data.data1)
17
18     // 2.同步存储多数据
19     var test2 = wx.getStorageSync('test2') || []    // 先去小程序中去找一
20     下有没有叫做test2的数据,有的话就可以找到,没有找到的话,就为空[]
21     test2.push(this.data.data1) //将后续输入框输入的内容存放进去
22     wx.setStorageSync('test2', test2)
23
24     // 3.异步缓存
25     wx.setStorage({
26         key:"test3",
27         data:this.data.data1
28     })
29 },
30 // 跳转到查看历史页面
31 history:function(e){
32     wx.navigateTo({
33         url: '../test19/test19',
34     })
35 },
```

```

1  data: {
2      test1:"",
3      test2:[],
4      test3:""
5  },
6
7  /**
8   * 生命周期函数--监听页面加载
9   */
10 onLoad(options) {
11     // 当页面一加载,就获取存放的数据--> test1
12
13     var that = this
14     wx.getStorage({
15         key:"test3",
16         success:function(e){
17             that.setData({
18                 test1:wx.getStorageSync('test1'),
19                 test2:wx.getStorageSync('test2'),
20                 test3:e.data //特殊
21             })
22         }
23     })
24
25 },

```

```

1  <text>{{test1}}</text>
2
3  <view>-----</view>
4
5  <text wx:for="{{test2}}">{{item}}</text>
6
7  <view>-----</view>
8  <text>{{test3}}</text>

```

6.4 网络请求

wx.request(Object object)

功能描述--->发起 HTTPS 网络请求。

▼

HTML |

1 <button bindtap="test2" type="primary" bindtap="test3">请 求</button>

▼

HTML |

1 test3:function(e){
2 //发起请求
3 wx.request({
4 url: 'https://route.showapi.com/341-1?showapi_appid=1220844&showa
pi_sign=92fbc5b376f544a3be69fc23993de5f5',
5 success:function(e){
6 console.log(e)
7 }
8 })
9 }

6.5 案例:英文励志文案



7.第七章 使用手机设备

7.1 选择图片

```
1  var that = this
2      wx.chooseMedia({
3          count: 9,
4          mediaType: ['image','video'],
5          sourceType: ['album', 'camera'],
6          maxDuration: 30,
7          camera: 'back',
8          success(res) {
9              console.log(res.tempFiles[0].tempFilePath) //上传图片的路径
10             that.setData({
11                 src:res.tempFiles[0].tempFilePath
12             })
13         }
14     })
```

7.2 saveFile

```
1  wx.saveImageToPhotosAlbum({
2      success(res) { }
3  })
```

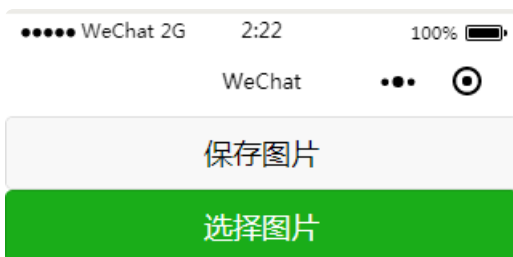
7.3 预览图片

```
1  wx.previewImage({
2      current: src, // 当前显示图片的http链接
3      urls: imgList // 需要预览的图片http链接列表
4  })
```

7.4 授权相关

```
1  getUserProfile(e) {  
2    // 推荐使用 wx.getUserProfile 获取用户信息，开发者每次通过该接口获取用户个人信息  
    均需用户确认  
3    // 开发者妥善保管用户快速填写的头像昵称，避免重复弹窗  
4    wx.getUserProfile({  
5      desc: '用于完善会员资料', // 声明获取用户个人信息后的用途，后续会展示在弹窗中，  
      请谨慎填写  
6      success: (res) => {  
7        this.setData({  
8          userInfo: res.userInfo,  
9          hasUserInfo: true  
10       })  
11     }  
12   })  
13 }
```

7.5 保存图片



7.6 录音

7.7 播放录音



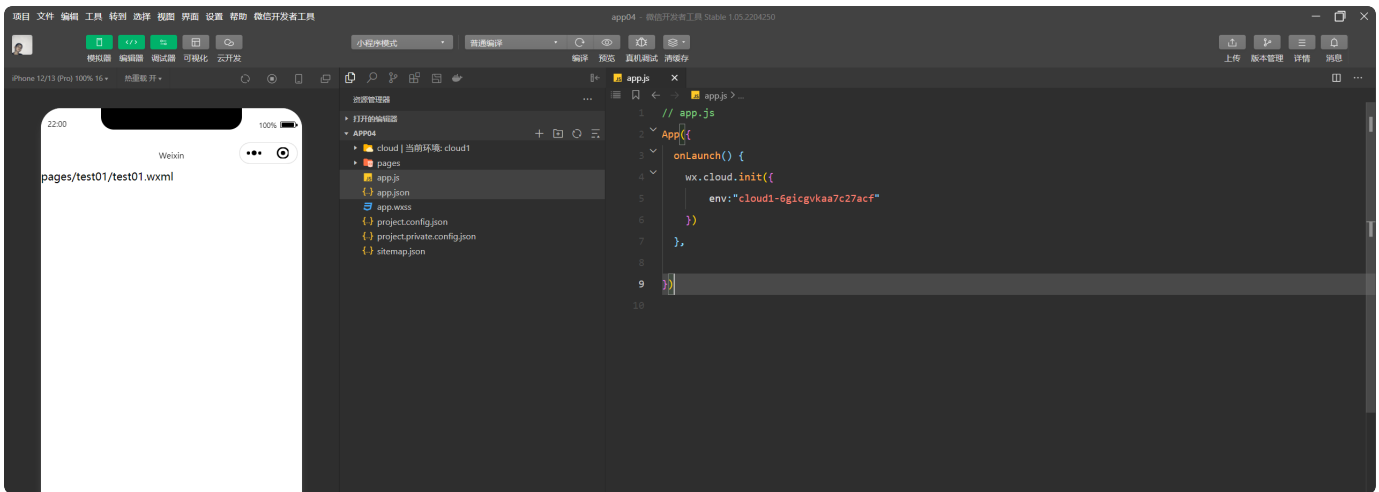
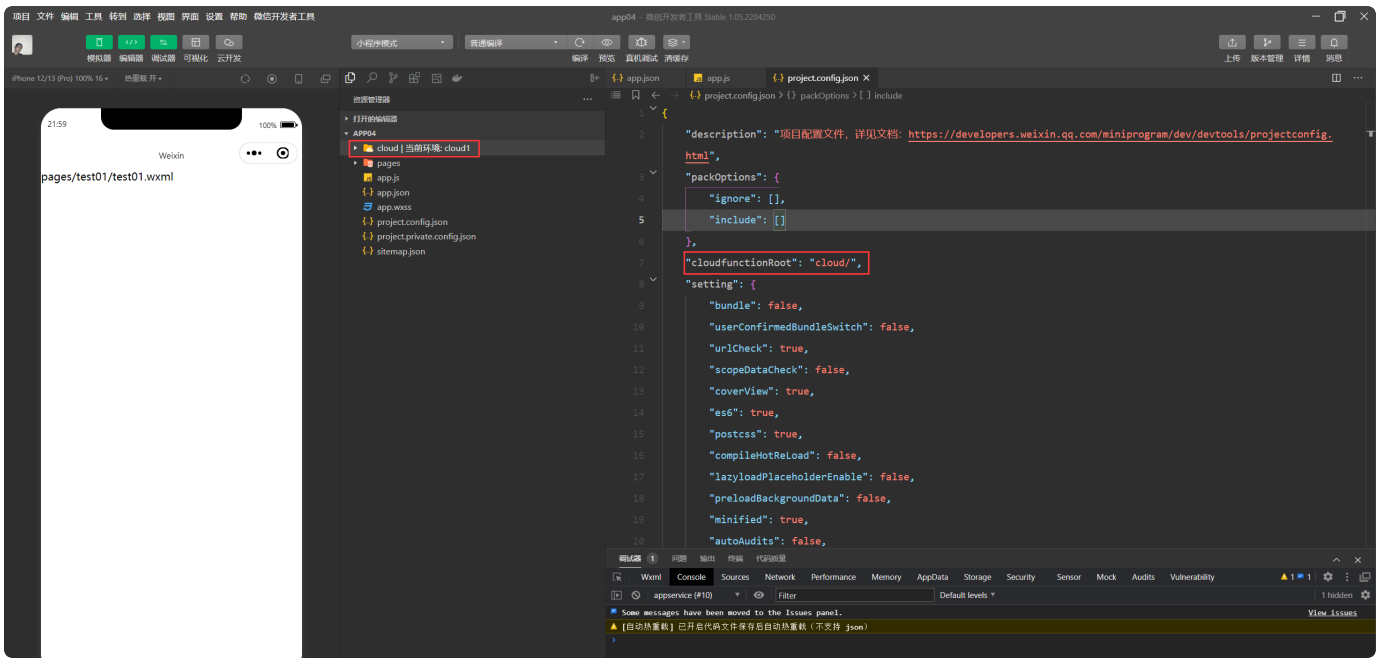
7.8 获取系统信息

7.9 获取网络状态

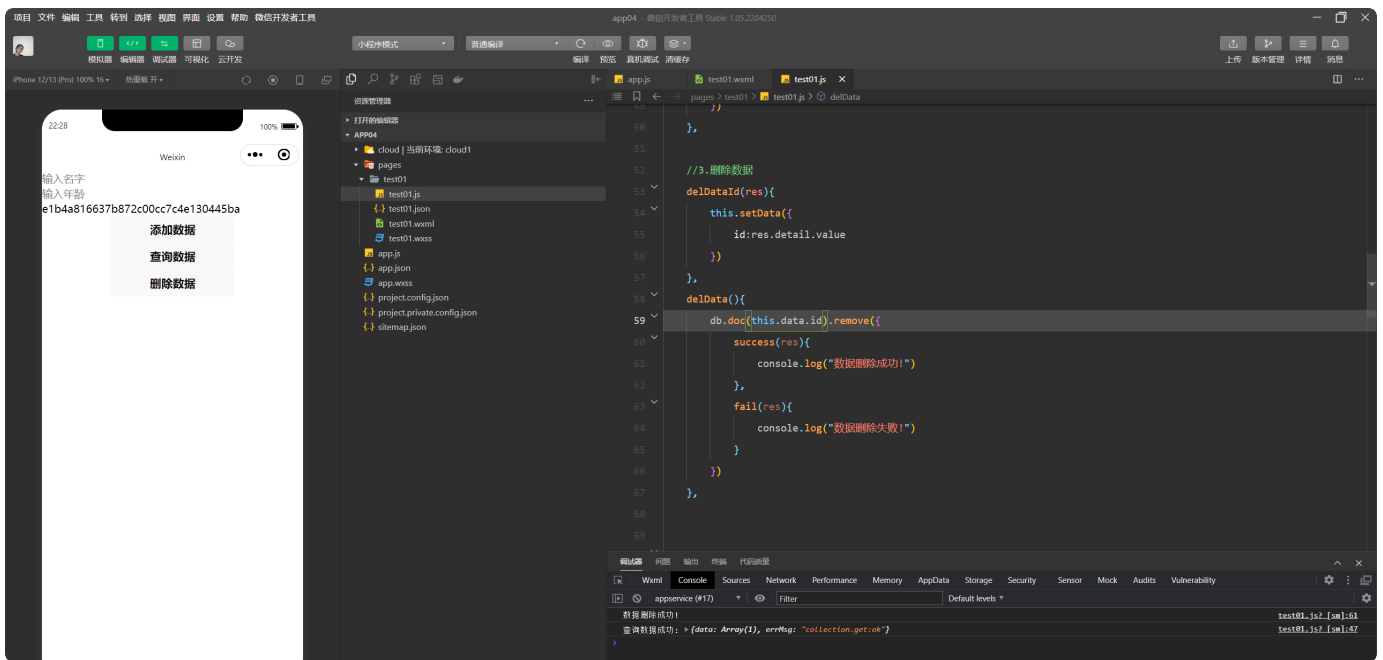
7.10 获取地理位置

8.第八章 综合案例

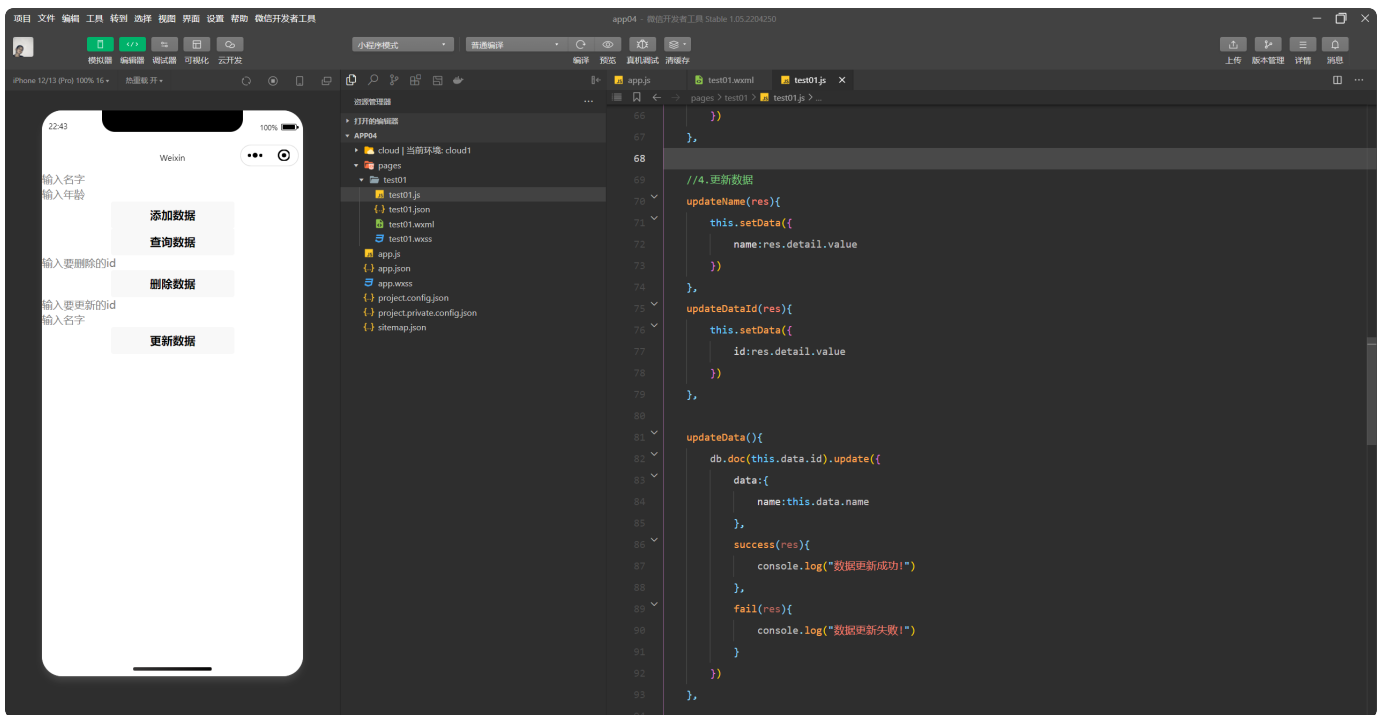
8.1 搭建云开发



8.2 添加数据



8.5 更新数据



8.6 上传文件

