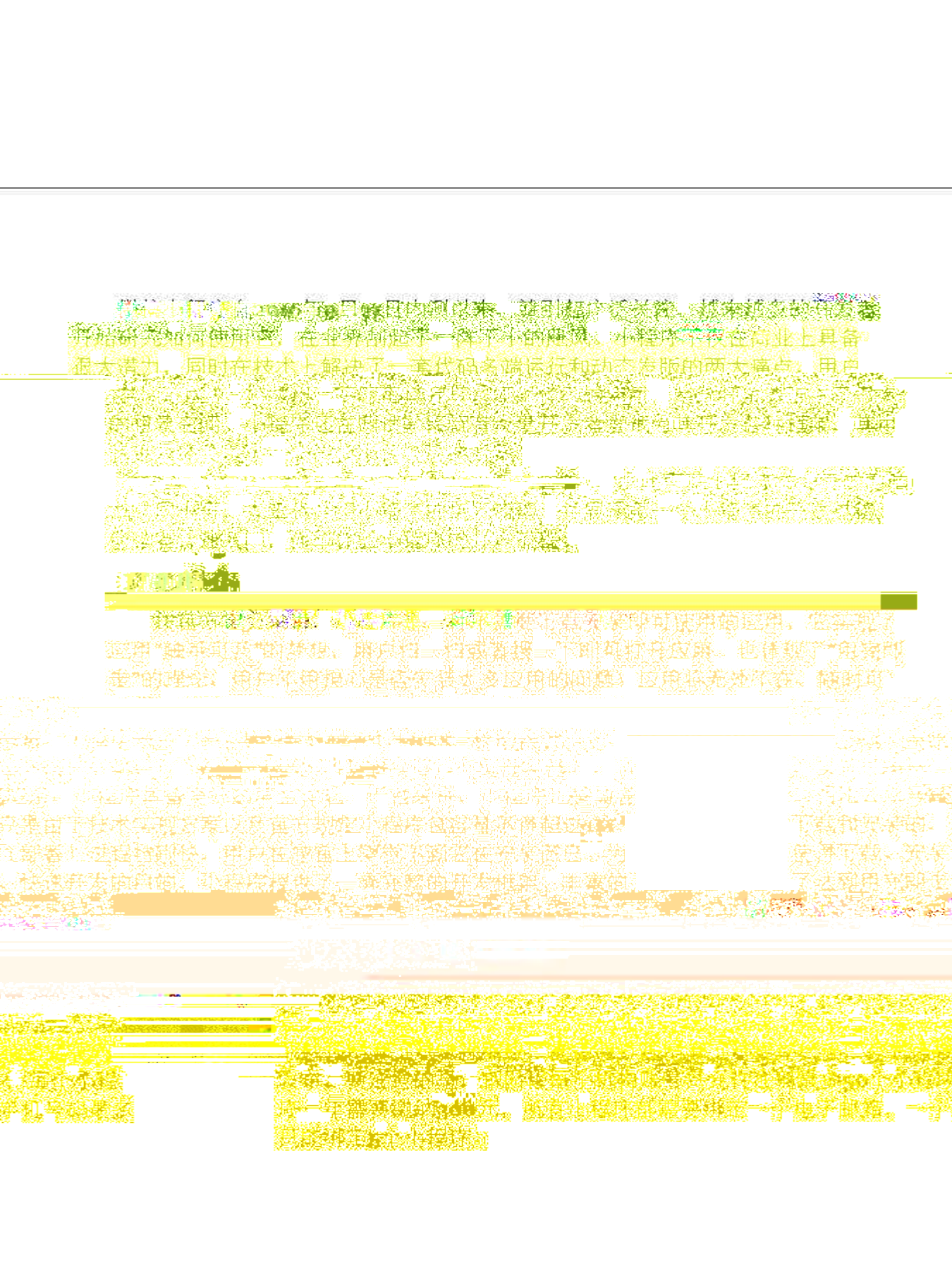
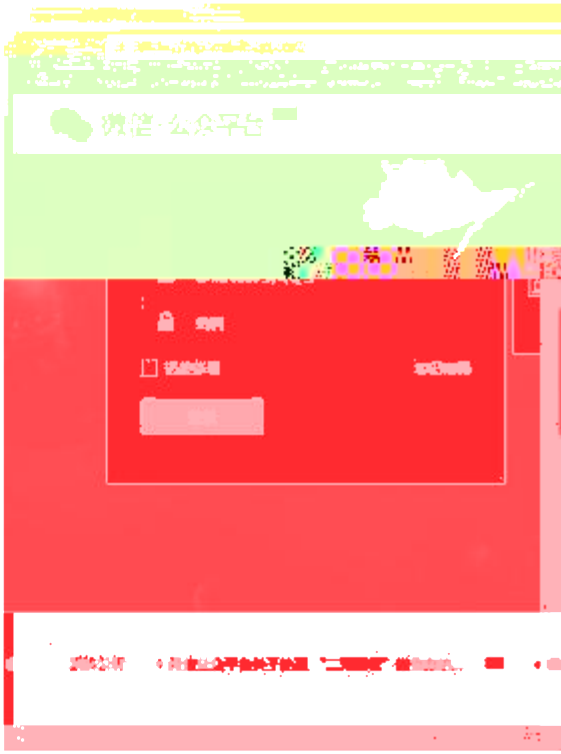
长汀一中选修2课程

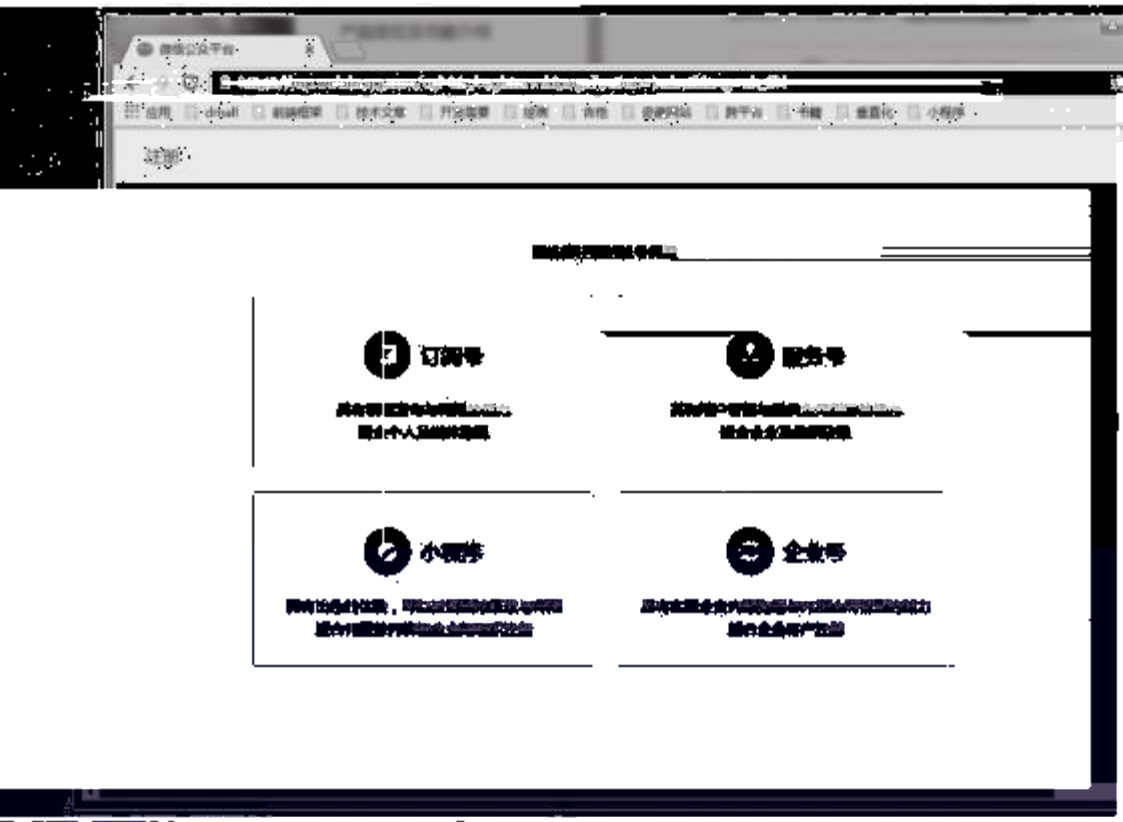
自己动手做一个微信小程序



1.2.1 注册小程序帐号

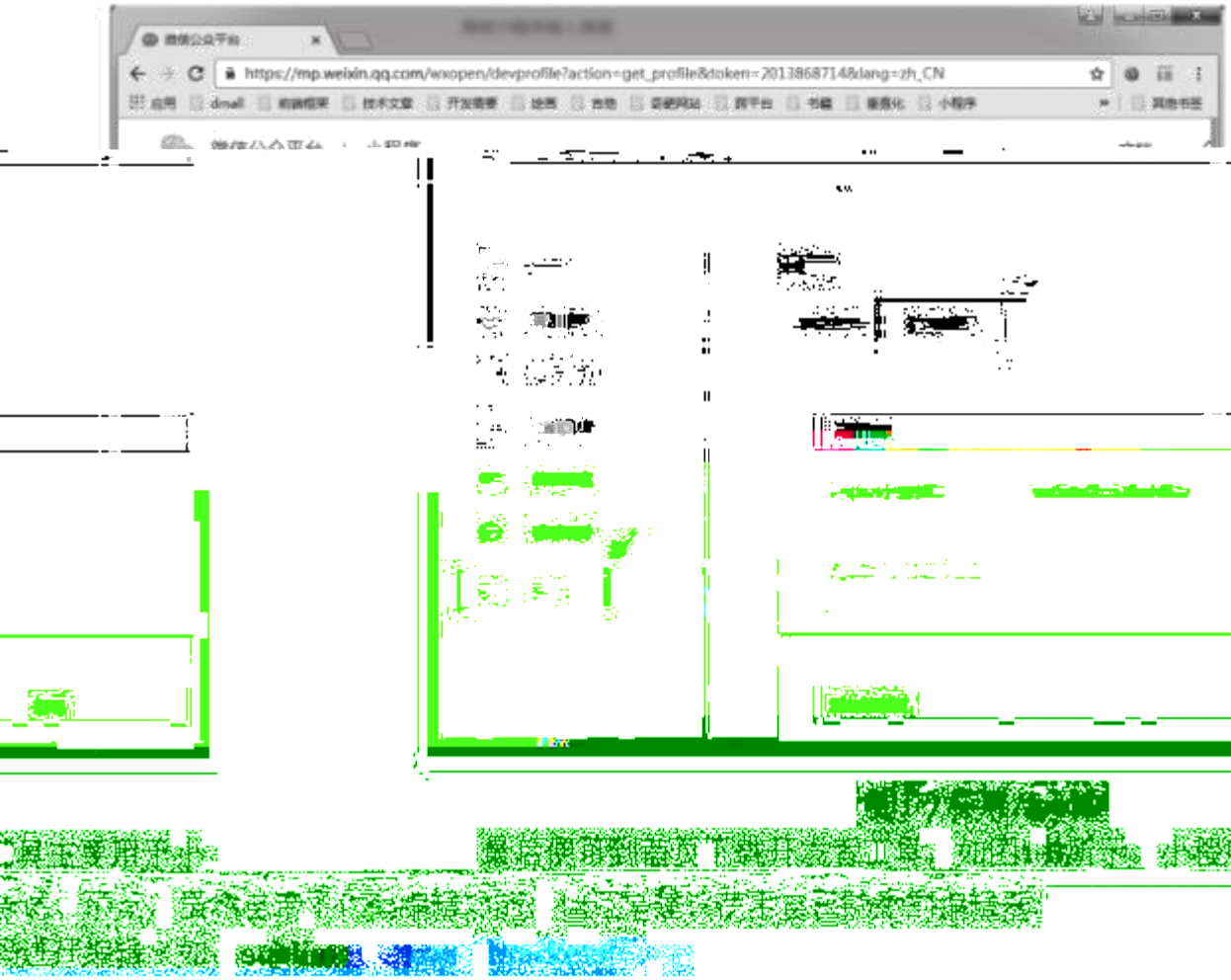
注册小程序帐号只需如下四步：




订阅号
小程序
企业号

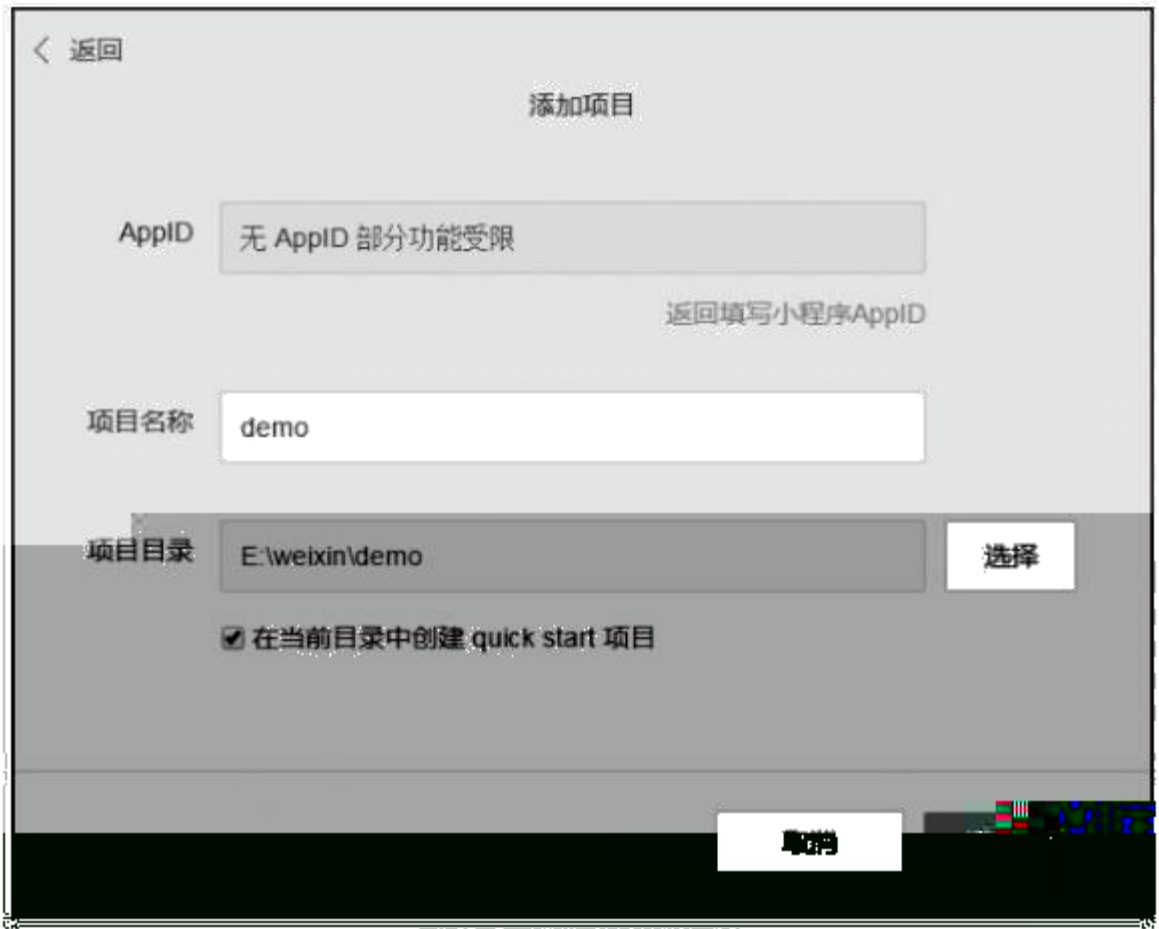

https://mp.weixin.qq.com/wsopen/waregister?action=step1
① 帐号信息 ② 邮箱激活 ③ 信息登记
每个邮箱仅能申请一个小程序
已有微信小程序？立即登录
邮箱
作为登录帐号，请填写未被微信公众平台注册，未
被微信开放平台注册，未被个人微信号绑定的邮箱
密码
字母、数字或者英文符号，最短8位，区分大小写
确认密码

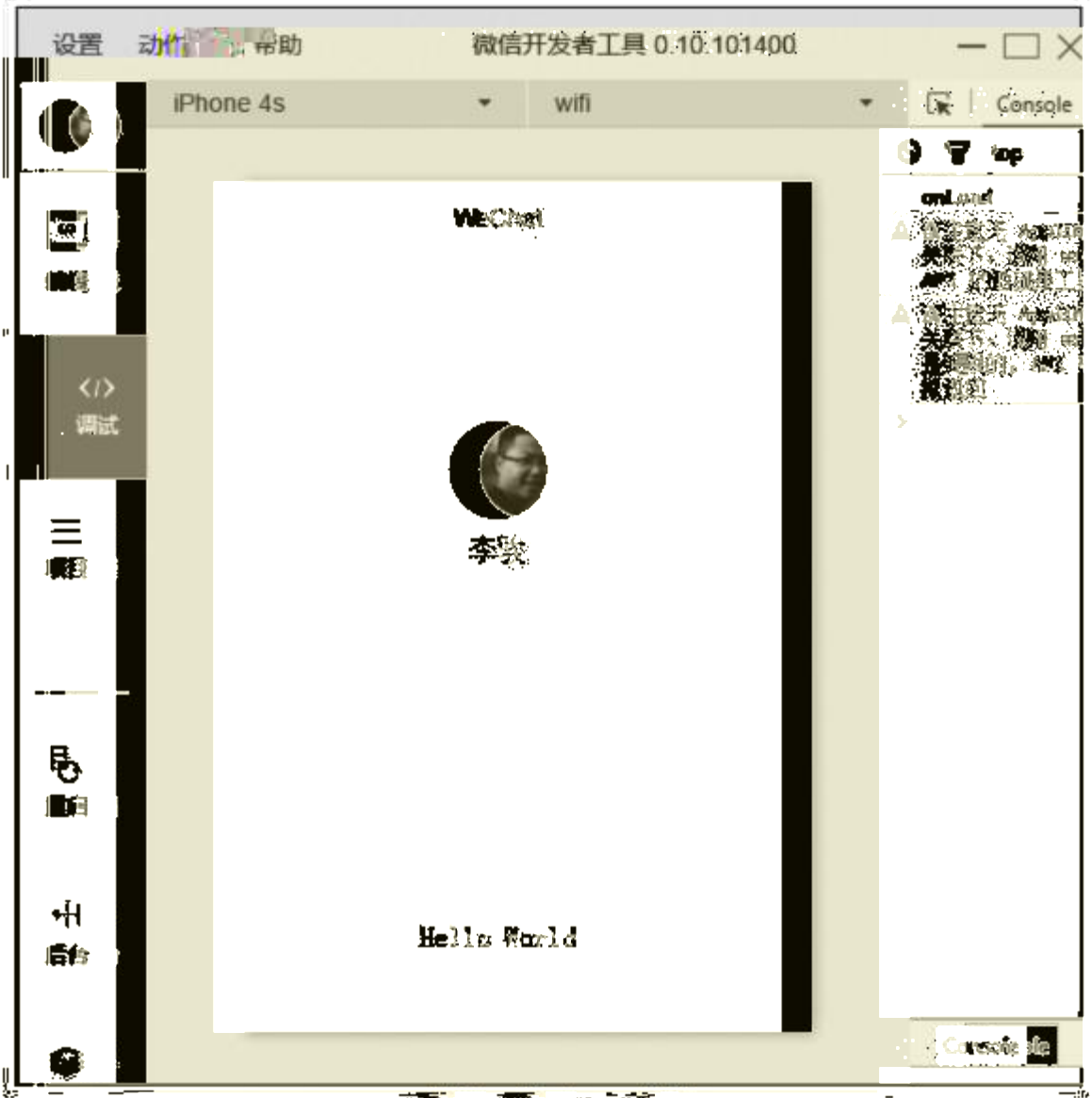

https://mp.weixin.qq.com/wxopen/waconstructorpage?action=step3&lang=zh_CN&token=6552265
主体类型
如何选择主体类型？
企业
政府
媒体
其他组织
主体信息登记
企业类型
企业
个体工商户
企业名称




https://mp.weixin.qq.com/wxopen/initprofile?action=home&lang=zh_CN&token=2013868714
首页
开发管理
用户身份
数据分析
模板消息
微信支付
设置
小程序发布流程
step
1
微信认证
企业类型请先通过微信认证完成主体真实性确认。否则将无法发布小程序。
已完成 查看
小程序信息
补充小程序的基本信息，如名称、图标、描述等
已完成 查看
小程序开发与管理
已绑定 查看
开发工具
下载开发者工具进行代码的开发和上传
添加开发者
添加开发者，进行代码上传

[redacted]

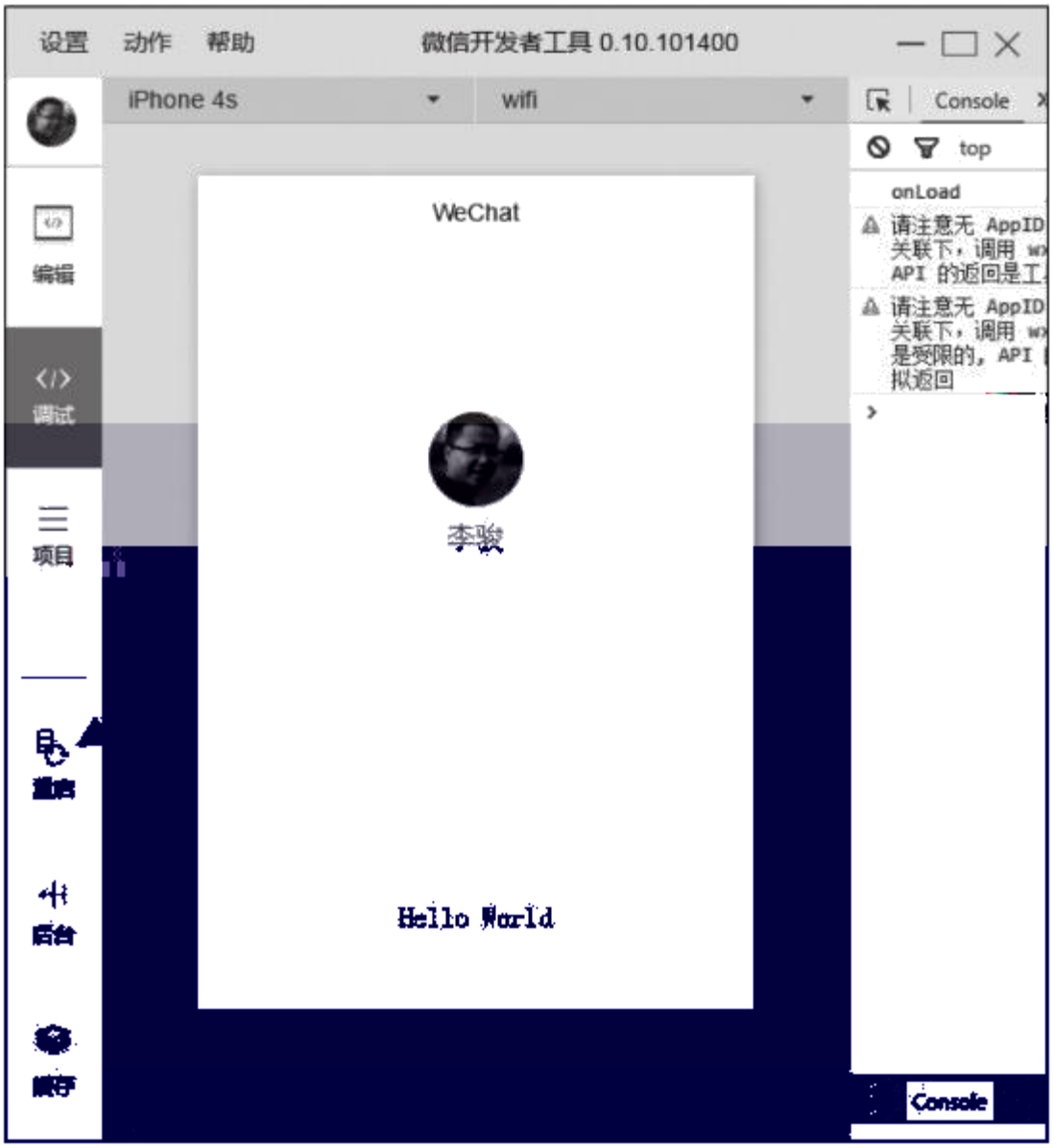

返回
添加项目
AppID
无 AppID 部分功能受限
返回填写小程序AppID
项目名称
demo
项目目录
E:\weixin\demo
选择
在当前目录中创建 quick start 项目


设置
帮助
微信开发者工具 0.10.101400
iPhone 4s
wifi
Console
调试
Hello World

```
<!--index.wxml-->
<view class="container">
  <view  bindtap="bindViewTap" class="userinfo">
```

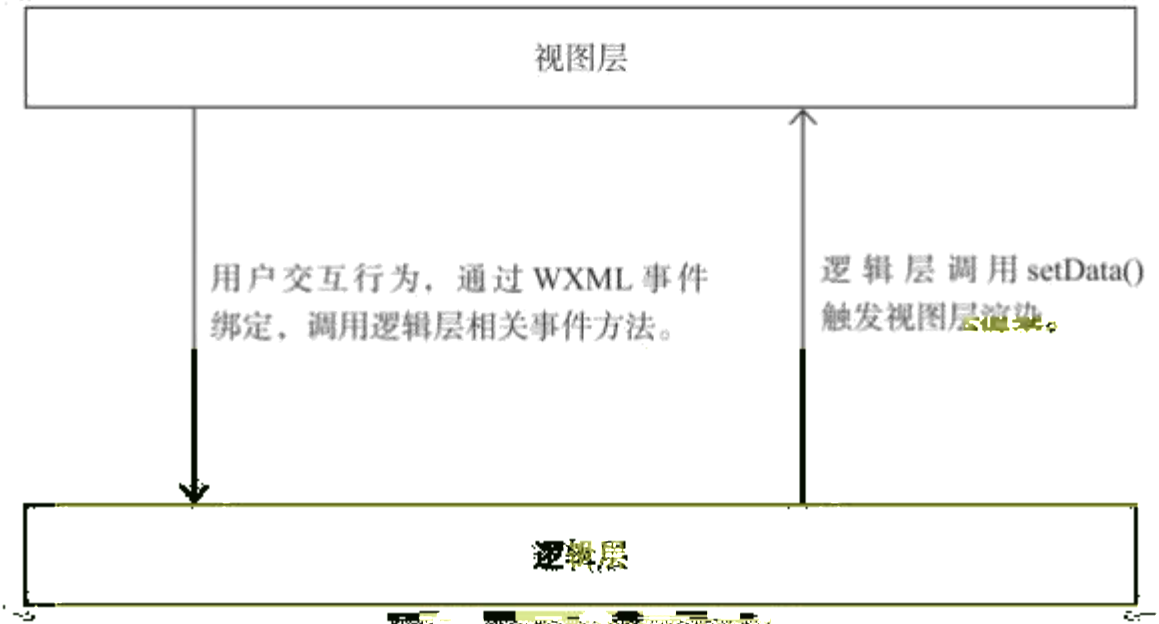
serInfo.avatarUrlserInfo.nickName

```
  </view>
  <view class="usermotto">
    <!-- 修改这句代码 -->
    <text class="user-motto">我的第一个小程序</text>
```

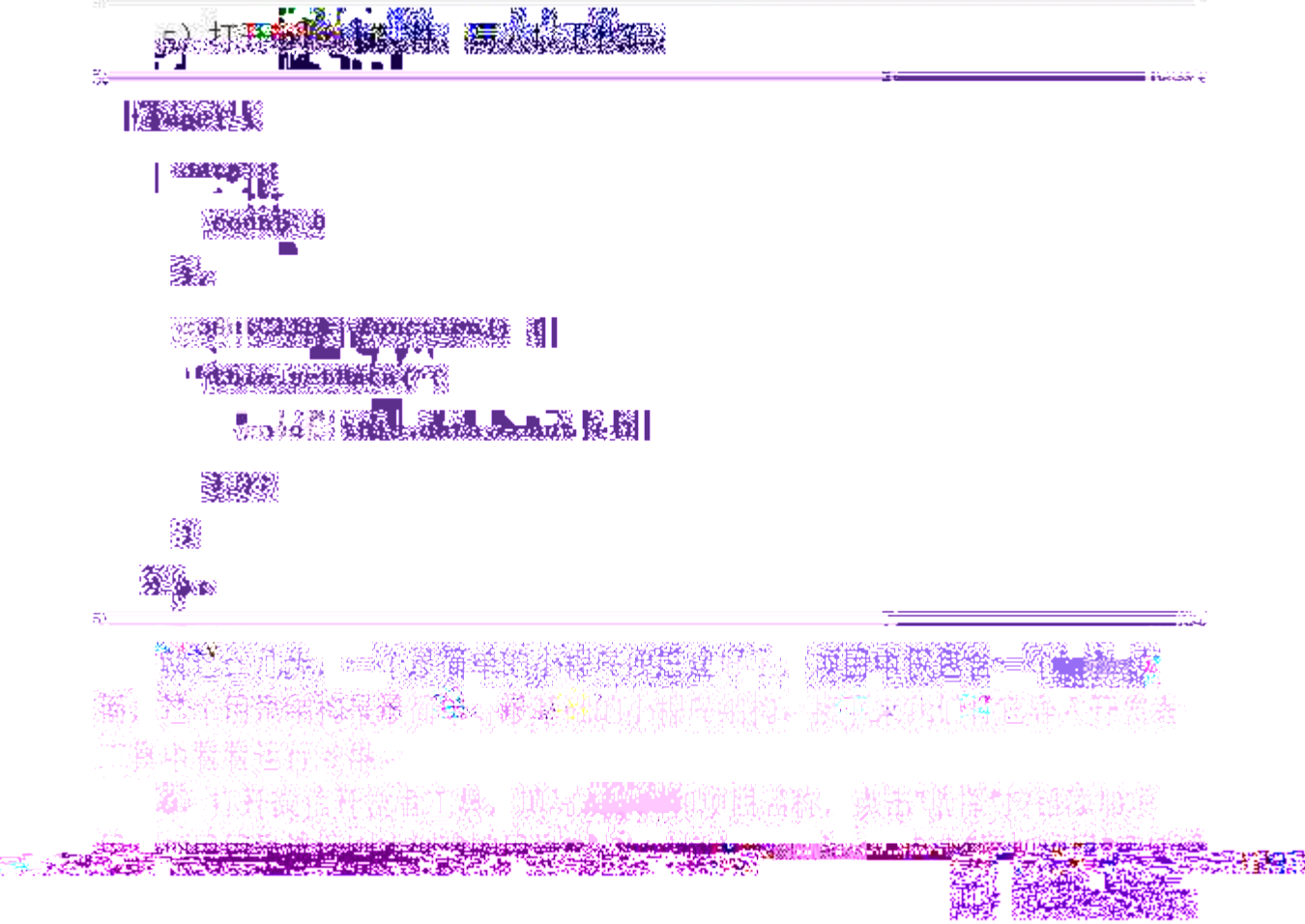

设置
动作
帮助
微信开发者工具 0.10.101400
iPhone 4s
wifi
Console
top
编辑
调试
项目
WeChat
onLoad
请注意无 AppID
关联下，调用 wx
API 的返回是工
请注意无 AppID
关联下，调用 wx
是受限的，API
拟返回
Hello World
Console


设置
动作
帮助
微信开发者工具 0.10.101400
iPhone 4s
wifi
Console
top
WeChat
onLoad


设置
动作
帮助
微信开发者工具 0.10.102800
本地开发目录
E:\work\wxapp-book\书籍\3 第三章 布局\demo
上传
预览


视图层
用户交互行为，通过 WXML 事件绑定，调用逻辑层相关事件方法。
逻辑层调用 setData() 触发视图层渲染。
逻辑层

```
<view bindtap="countClick">我是index页面,你点击了{{count}}次</view>
```



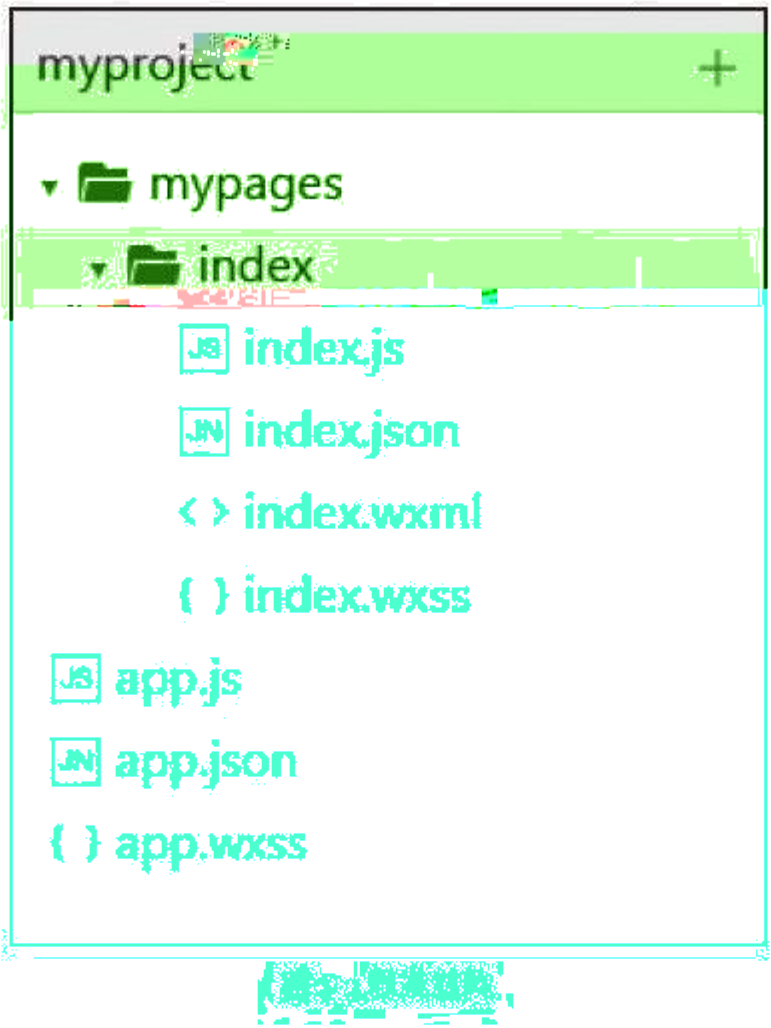

myproject
mypages
index
index.js
index.json
index.wxml
index.wxss
app.js
app.json
app.wxss

图2-3 项目配置界面

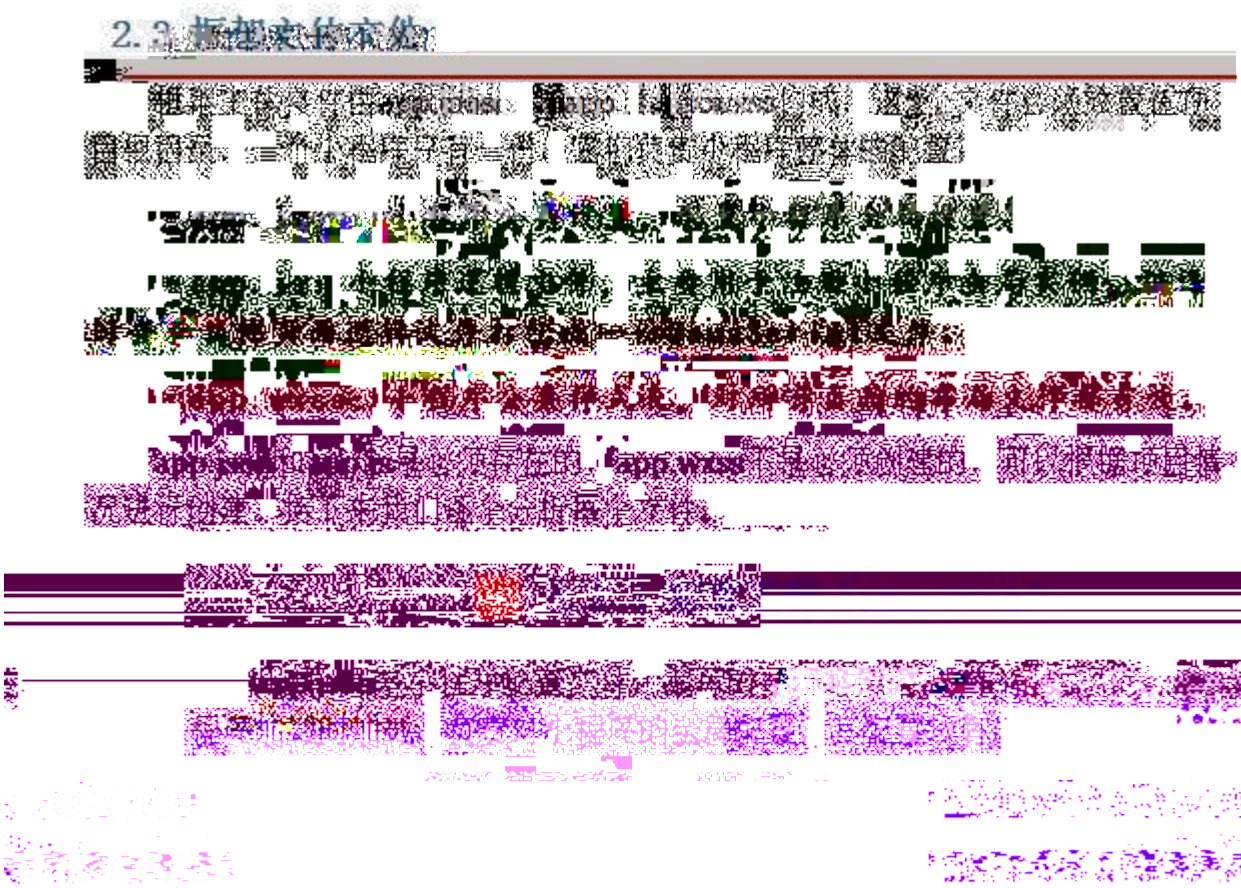
7）导入项目后我们便能看到运行界面，当我们点击文字时，点击次数也会随


设置
动作
帮助
微信开发者工具 0.10.101400
iPhone 4
wifi
Console

和整体样式，小程序启动时只会执行一次。这3个文件必须放在项目根目录，且文件名必须是app，其中app.json和app.js是必须的。

- 一个小程序有一个或多个页面，一个页面

第5课 框架主体文件



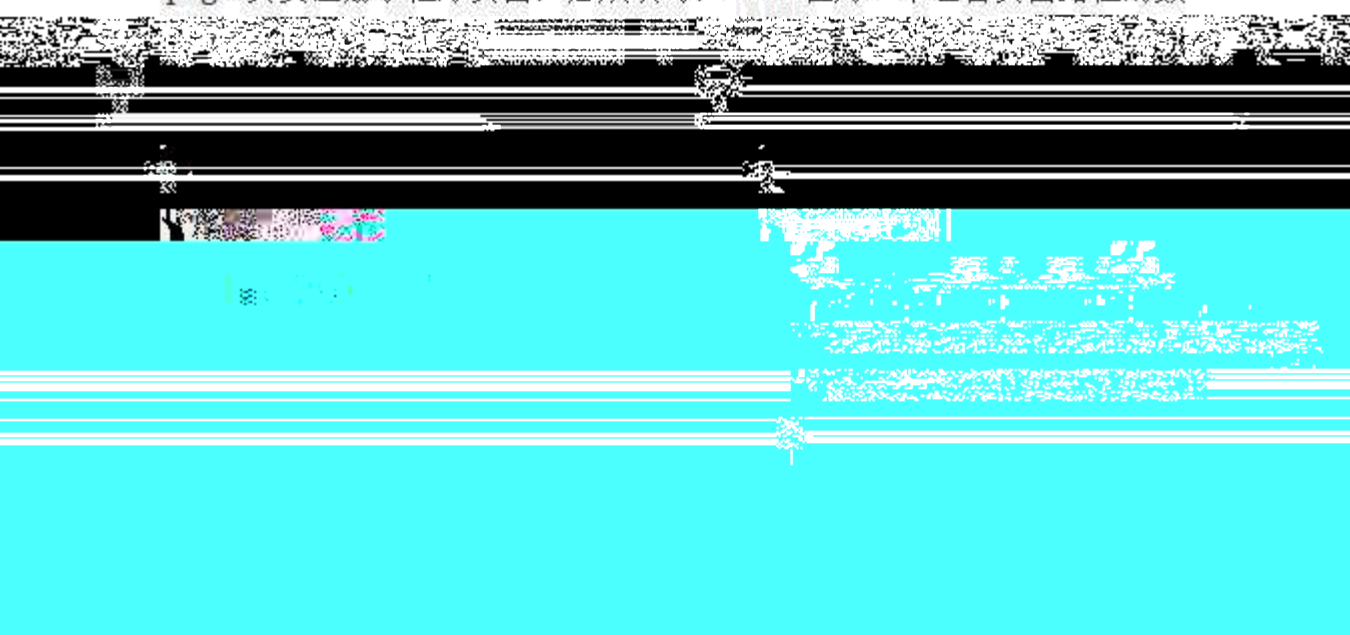
- tabBar：设置tab的表现。
- networkTimeout：设置网络超时时间。
- debug：设置是否开启debug模式，默认关闭。

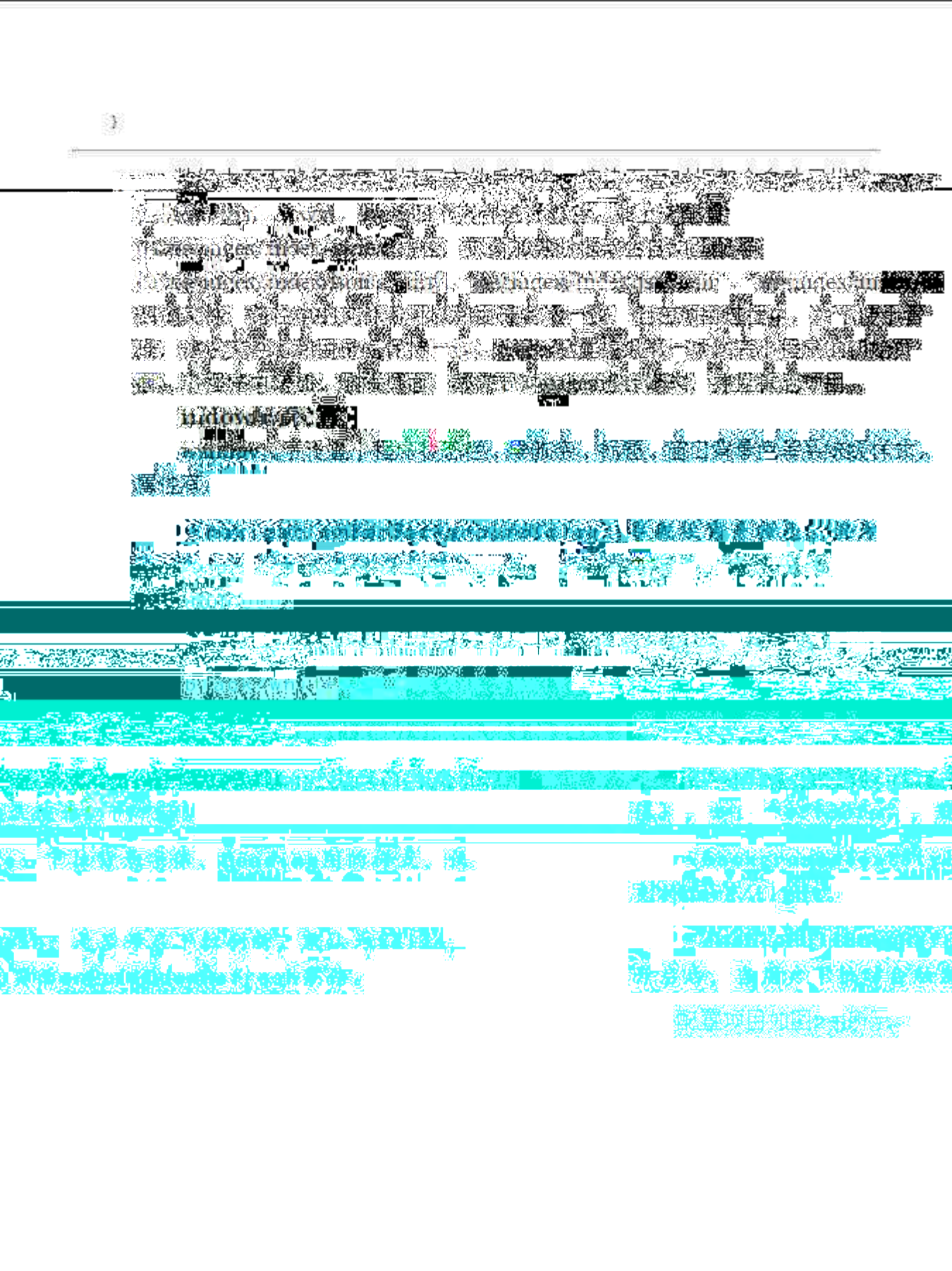
app.json文件内容整体结构如下：

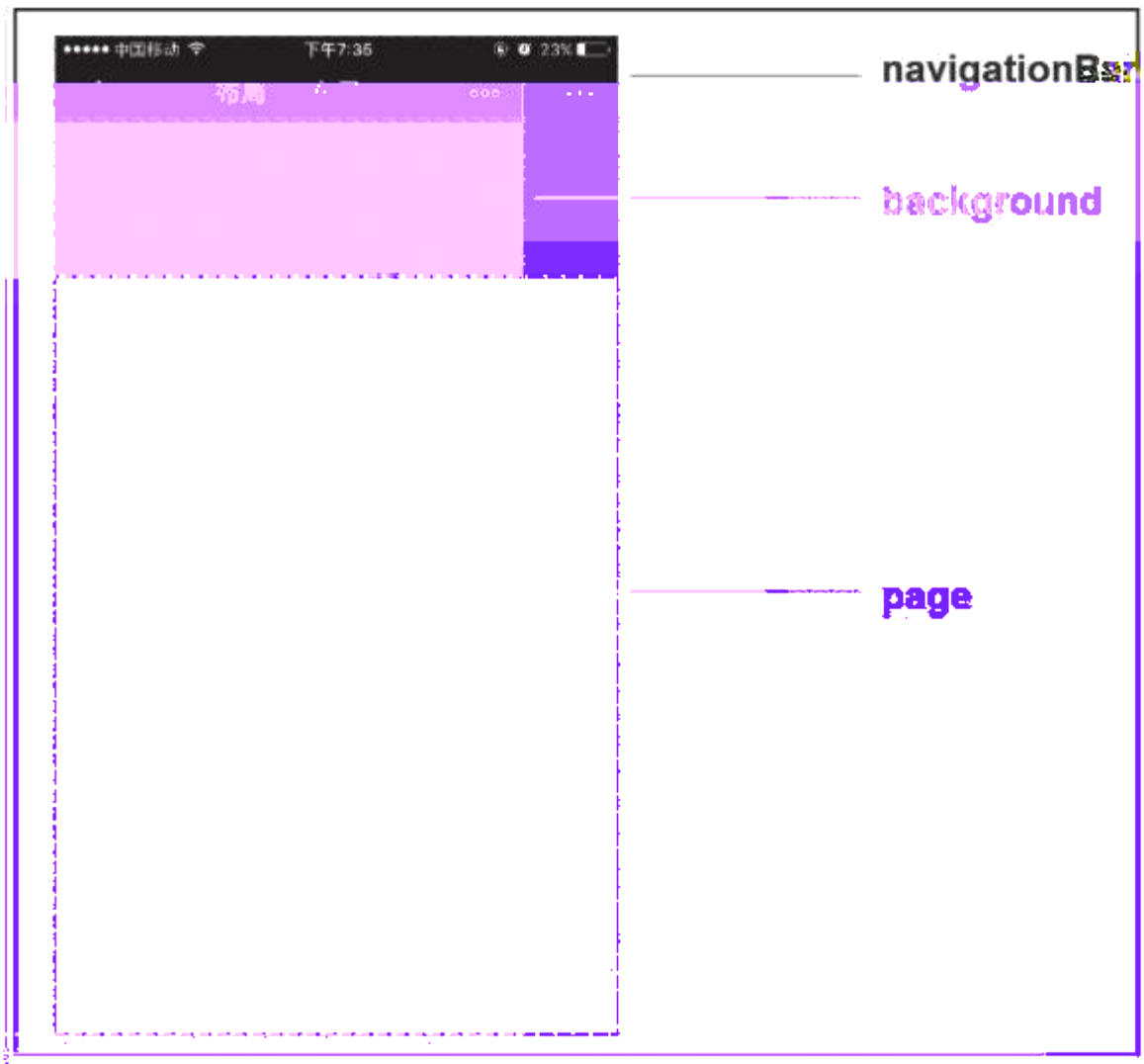
```
{
  // 页面路径设置
  "pages": [],
  // 默认页面的窗口设置
  "window": {},
  // 底部tab设置
  "tabBar": {},
  // 设置网络请求API的超时时间
  "networkTimeout": {},
  // 是否为debug模式
  "debug": false
}
```

1.pages配置

pages负责注册小程序页面，必须填写，value值为一个包含页面路径的数



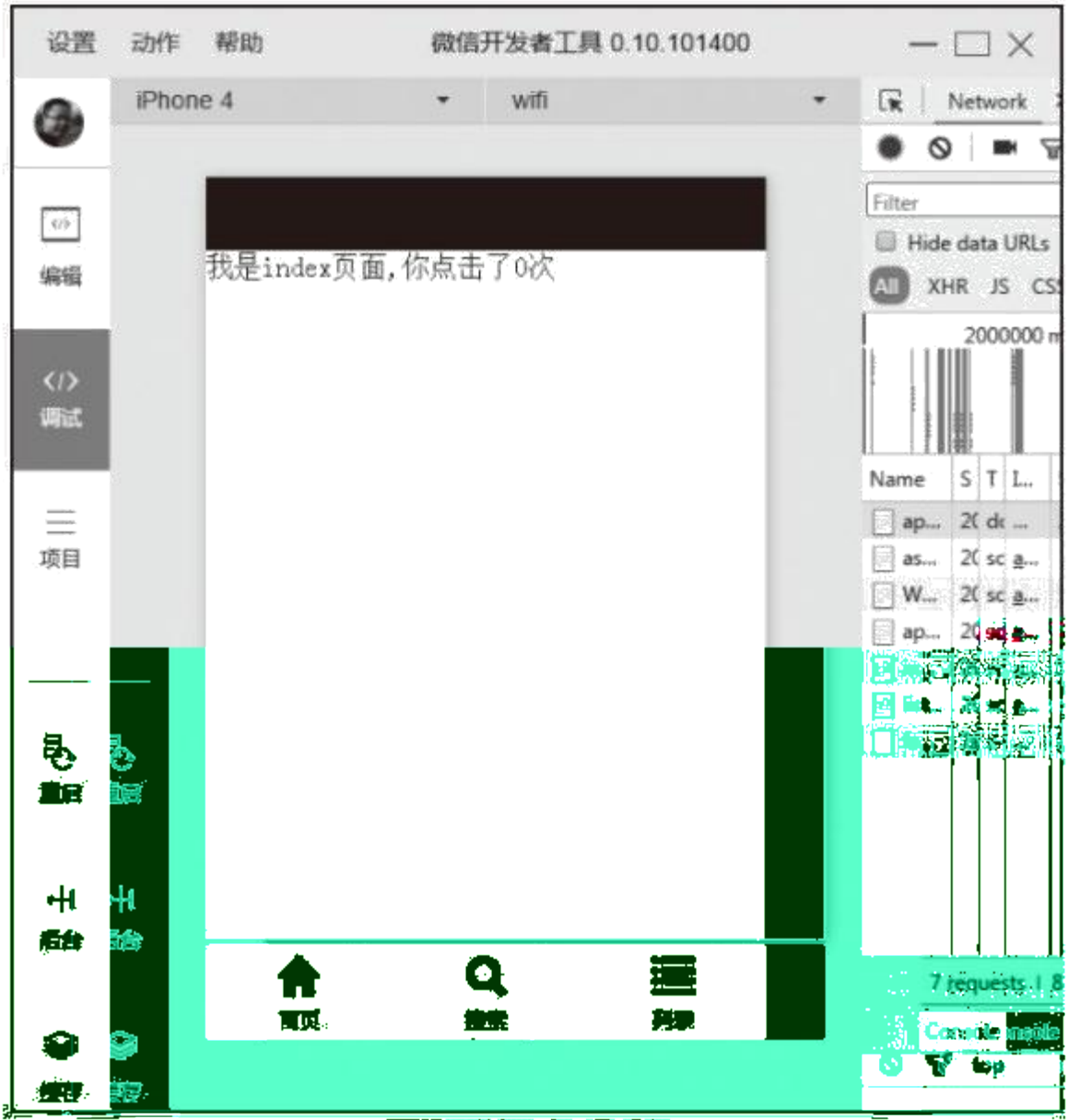



navigationBar
background
page

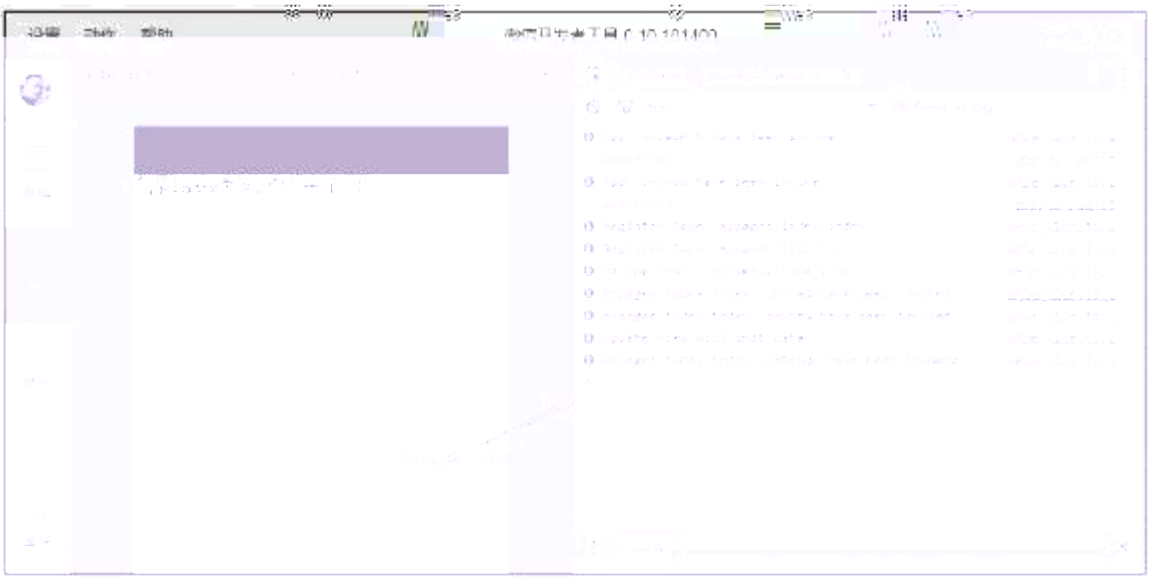
HexColor（十六进制颜色值），必填项。

• backgroundColor：tab的背景

```
  "backgroundColor": "#ffffff",
  "borderStyle": "black",
  "list": [
    {
      "iconPath": "images/home.png",
```


设置
动作
帮助
微信开发者工具 0.10.101400
iPhone 4
wifi
Network
编辑
调试
项目
我是index页面,你点击了0次
Filter
Hide data URLs
All
XHR
JS
Name
7 requests

- request：设置wx.request的超时时间，单位毫秒。
- connectSocket：设置wx.connectSocket的超时时间，单位毫



第6课 小程序逻辑文件

小程序中逻辑文件分为页面逻辑文件和小程序逻辑文件，app.js便是小程序逻辑文件，在这个文件中，我们可以通过App（）函数[illegible]

getApp（）获取。

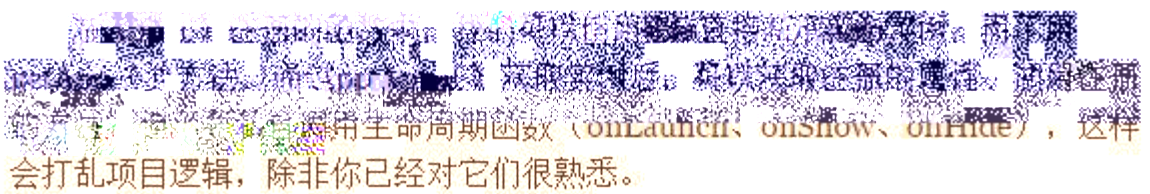
1.注册小程序

App（）函数用于注册一个小程序，参数为一个Object对象，在这个参数对象中我们可以注册自定义方法和属性供全局使用，就像在quick start项目中，我们利用App（）注册了用户登录信息。App（）函数必须在app.js中注册，且不能注册多个，其参数属性如下：

注册小程序示例代码如下：

```
App({
  onLaunch: function() {
    // 显示小程序时执行
  },
  onShow: function() {
    // 显示小程序时执行
  },
  onHide: function() {
    // 隐藏小程序时执行
  },
  globalFunction: '我是全局函数',
  globalData: '我是全局属性'
});
```

2.获取小程序实例



不要调用生命周期函数（onLaunch、onShow、onHide），这样会打乱项目逻辑，除非你已经对它们很熟悉。

2.3.3 全局样式（app.wxss）

app.wxss是全局样式表，对项目中每个页面都有效，可将一些系统级别的

第7课 框

2.4 框架页面文件

小程序中一个框架页面包含4个文件，同一框架页面的这4个文件必须具有相同的路径与文件名，进入小程序时或页面跳转时，小程序会根据app.json配置的路径找到对应的资源进行渲染。

- .js文件：页面逻辑文件，必要项。
- .wxml文件：页面结构文件，必要项。
- .wxss文件：页面样式文件。
- .json文件：页面配置文件。

与框架主体文件相比框架页面文件多了一种页面结构文件，其余3个文件和框架主体文件的功能类同，下面我们一一讲解每个文件作用。

2.4.1 页面配置文件

我们首先

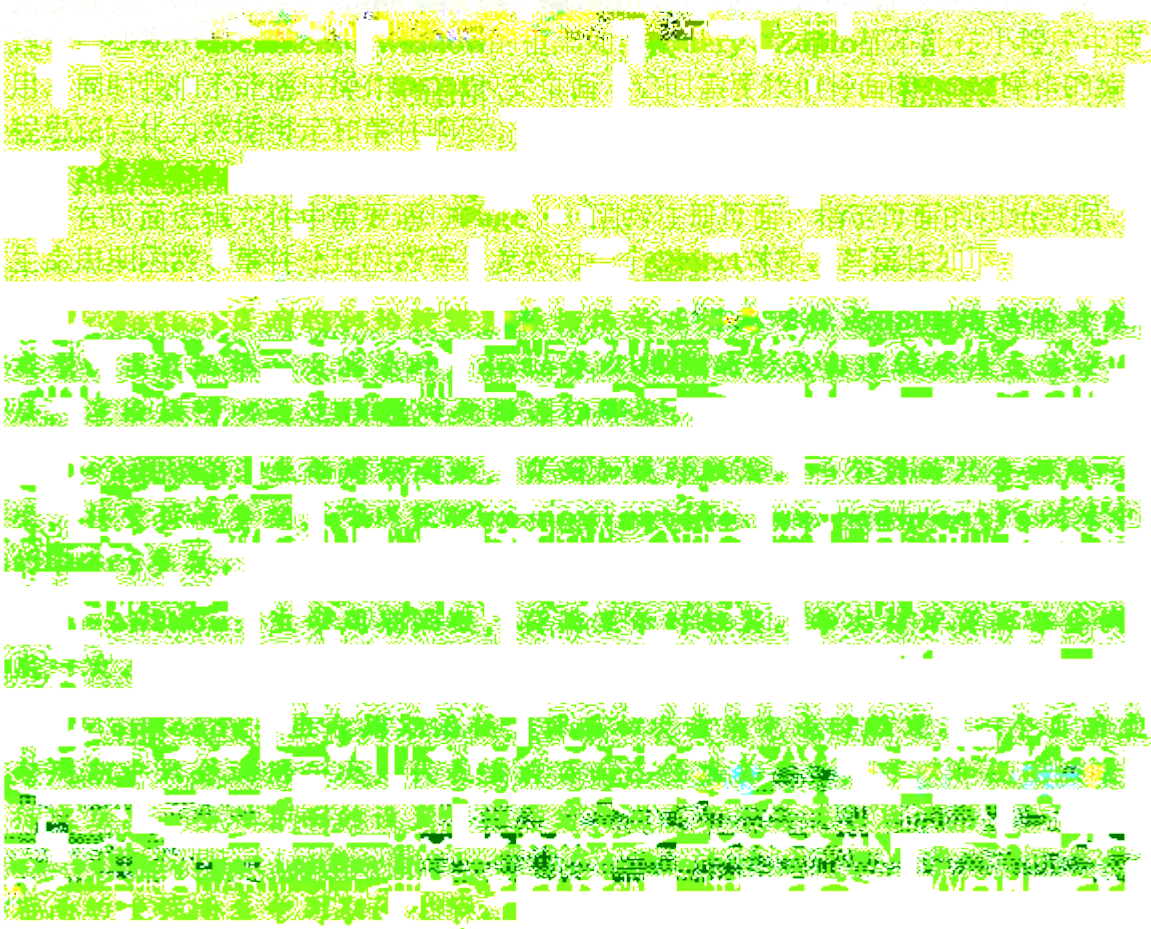
```
{
  "navigationBarBackgroundColor": "#000000",
  "navigationBarTextStyle": "black",
  "navigationBarTitle
}
```

第

2.4.2 页面逻辑文件（JavaScript）

页面逻辑文件主要功能有：设置初始化数据，注册当前页面生命周期函数，注册事件处理函数等。小程序的逻辑层文件是JavaScript文件，所有的逻辑文件，包括app.js，最终将会打包成一个js文件，在小程序启动时运行，直到小程序销毁，类似于ServiceWorker，所以逻辑层也称为App Service。

在小程序中，每个逻辑文件有独立的作用域，并具备模块化能力。由于



- onHide：生命周期函数，页面隐藏时触发。
- onUnload：生[redacted]

[redacted]

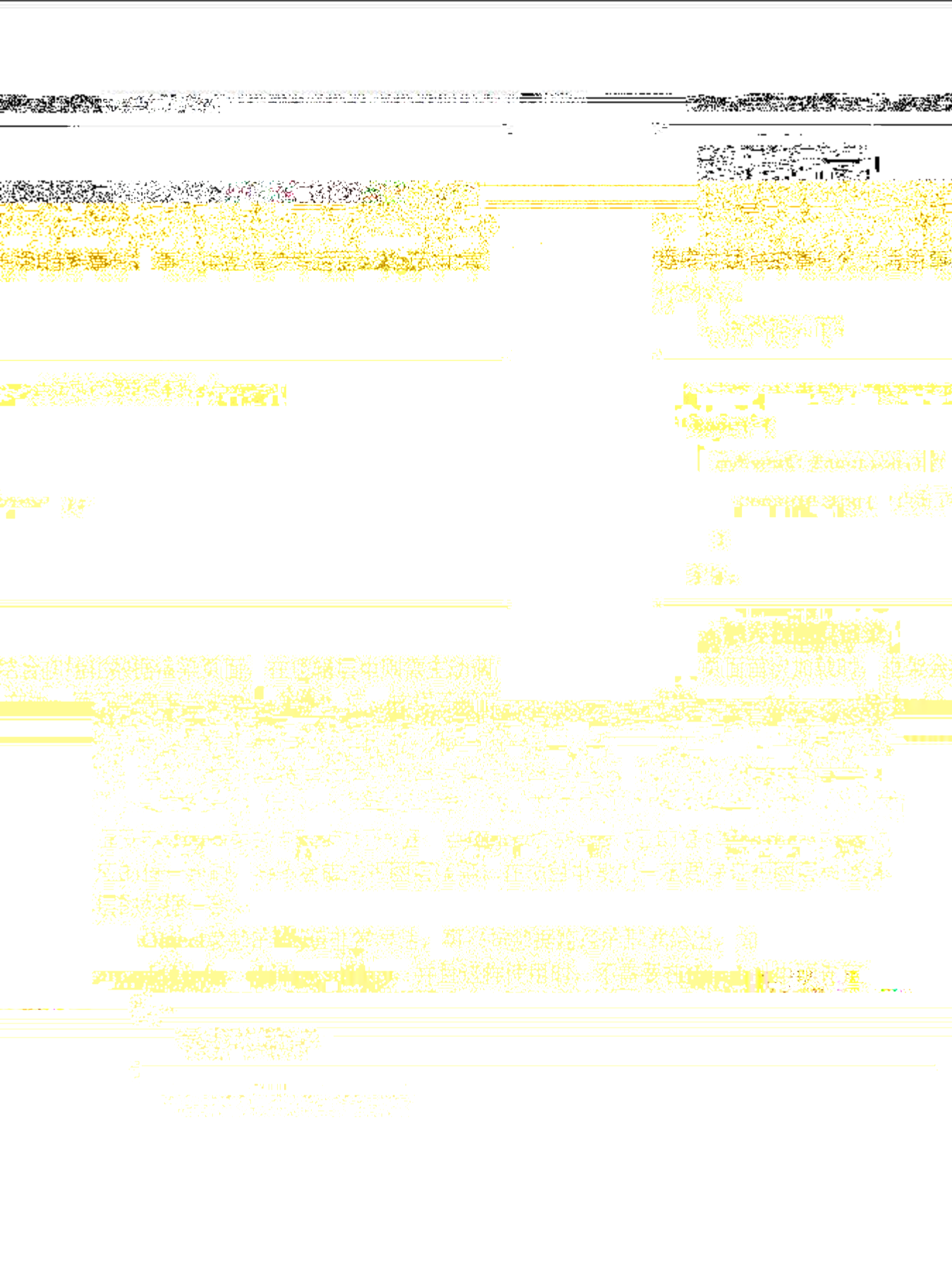
发。使用时需要将app.json配置中window的enablePullDownR[redacted]

```
onUnload: function() {
    // 页面卸载时执行
},
onPullDownRefresh: function() {
    // 下拉刷新时执行
},
onReachBottom: function() {
    // 下拉触底时执行
},
// 自定义函数
```

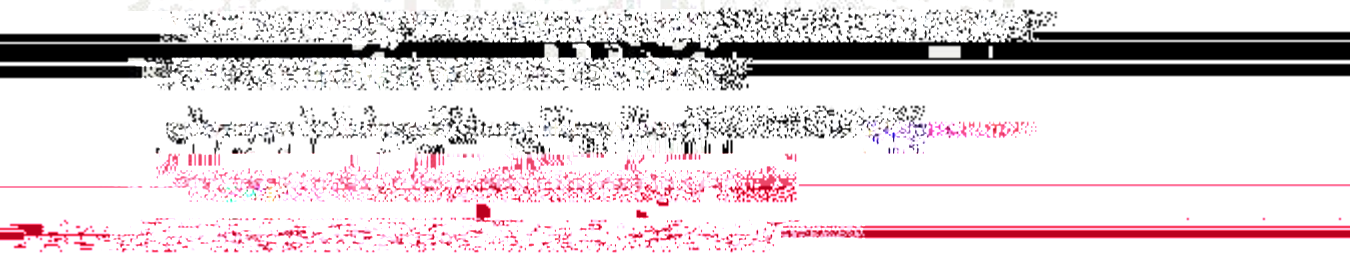
• 打开新页面：新页面入栈，依次触发新页面onLoad、onShow、onReady方法。

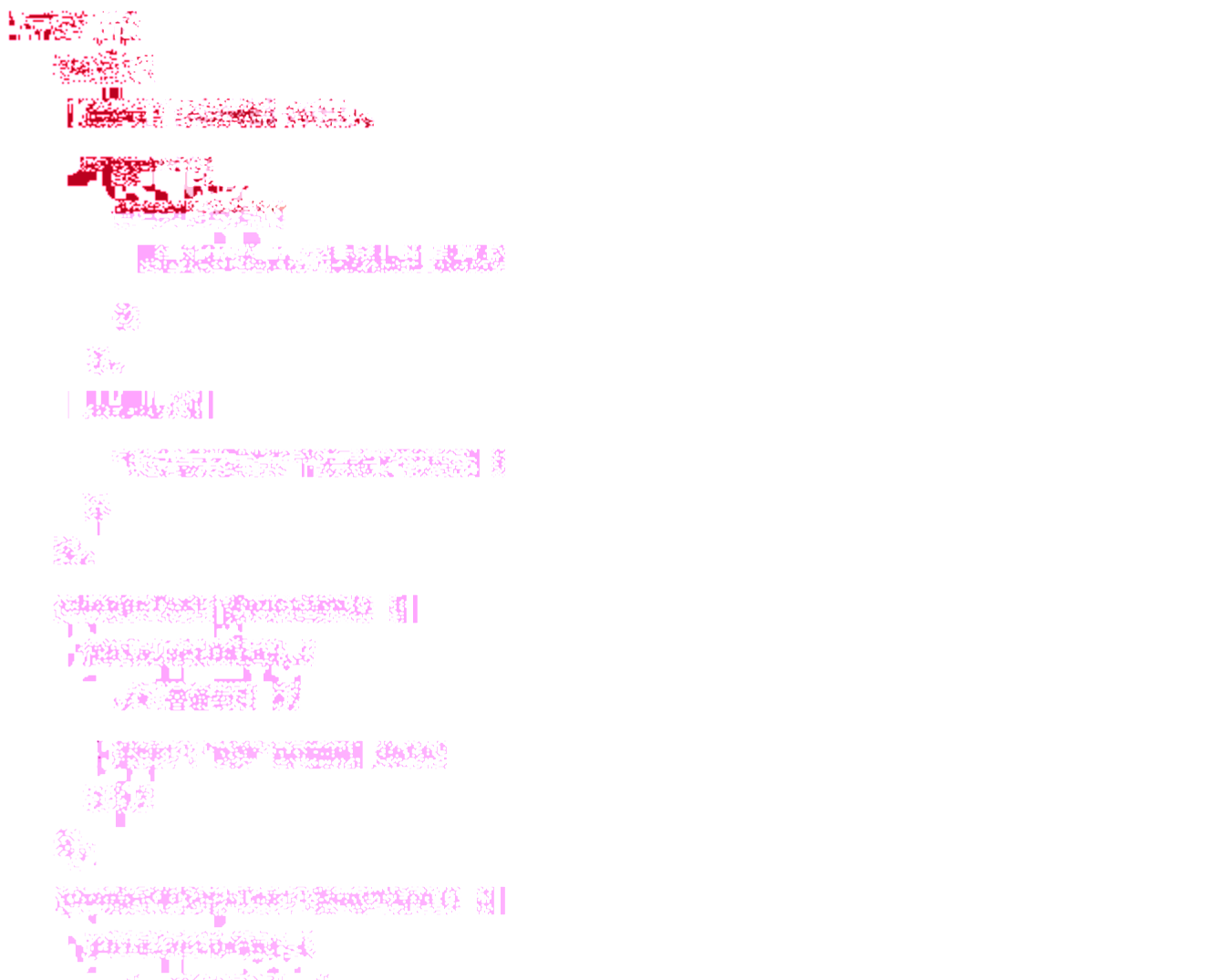
• 页面重定向：当前页面出栈并卸载，触发当前页面onUnload方法，新页面入栈，触发新页面onLoad、onShow、onReady方法。

• 页面返回：页面不断出栈并卸载，触发当前页面onUnload方

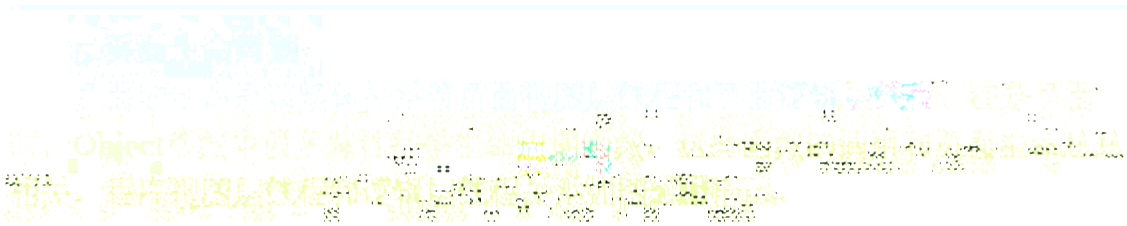


```
<button bindtap="changeText">修改普通数据</button>
<view>{{object.subObject.objectText}}</view>
```





```
        'object.subObject.objectText': 'new object data'
      } );
    },
    changeArrayText: function() {
      this.setData( {
        'array[1].text': 'new array data'
      } );
    }
  } );
```



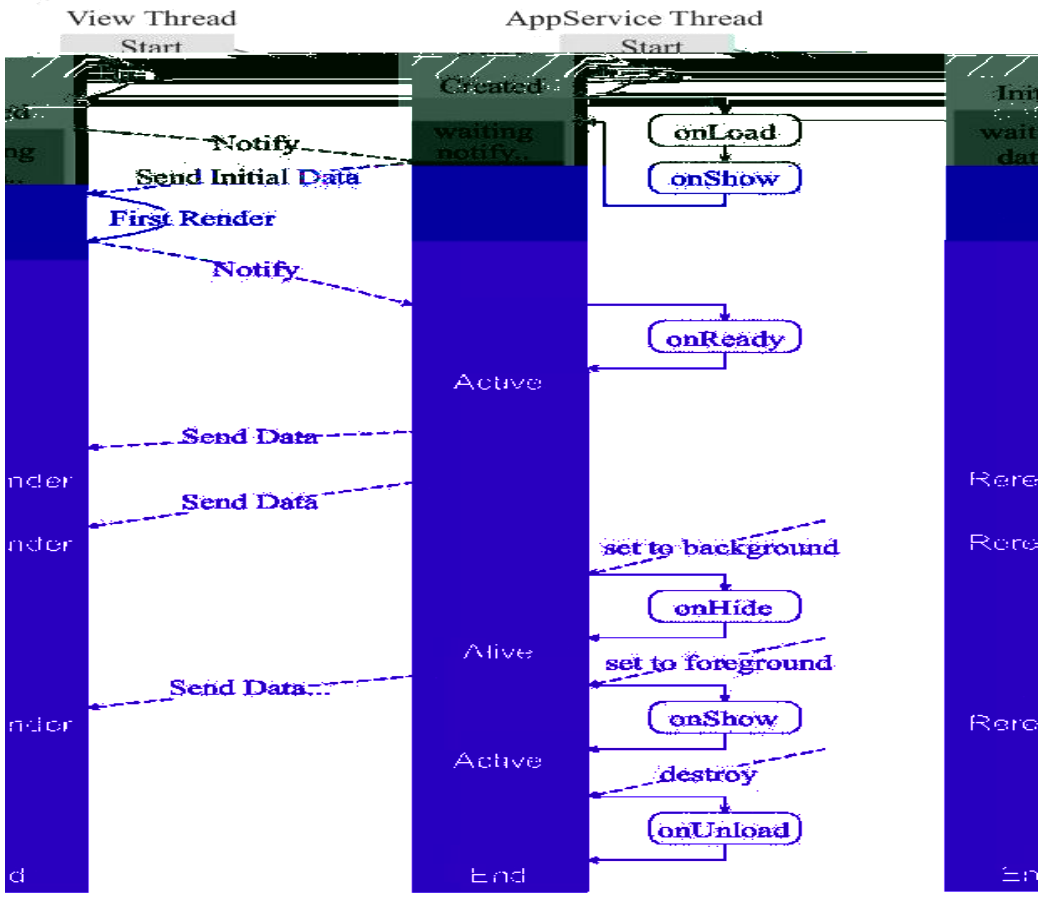


图2-8 Page实例的生命周期

第9课 页面结构文件

2.4.3 页面结构文件（WXML）

WXML（WeiXin Markup Language）是框架设计的一套标记语言，用于渲染界面，WXML的渲染原理和React Native思路一致，通过

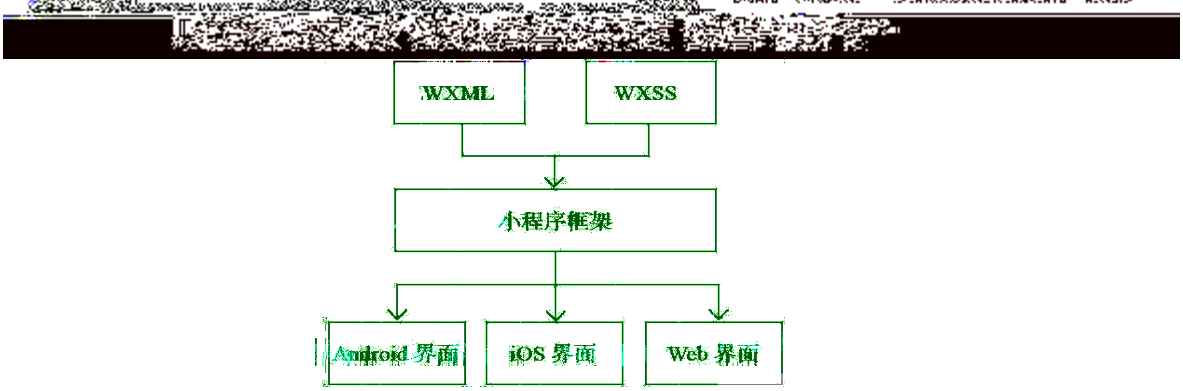
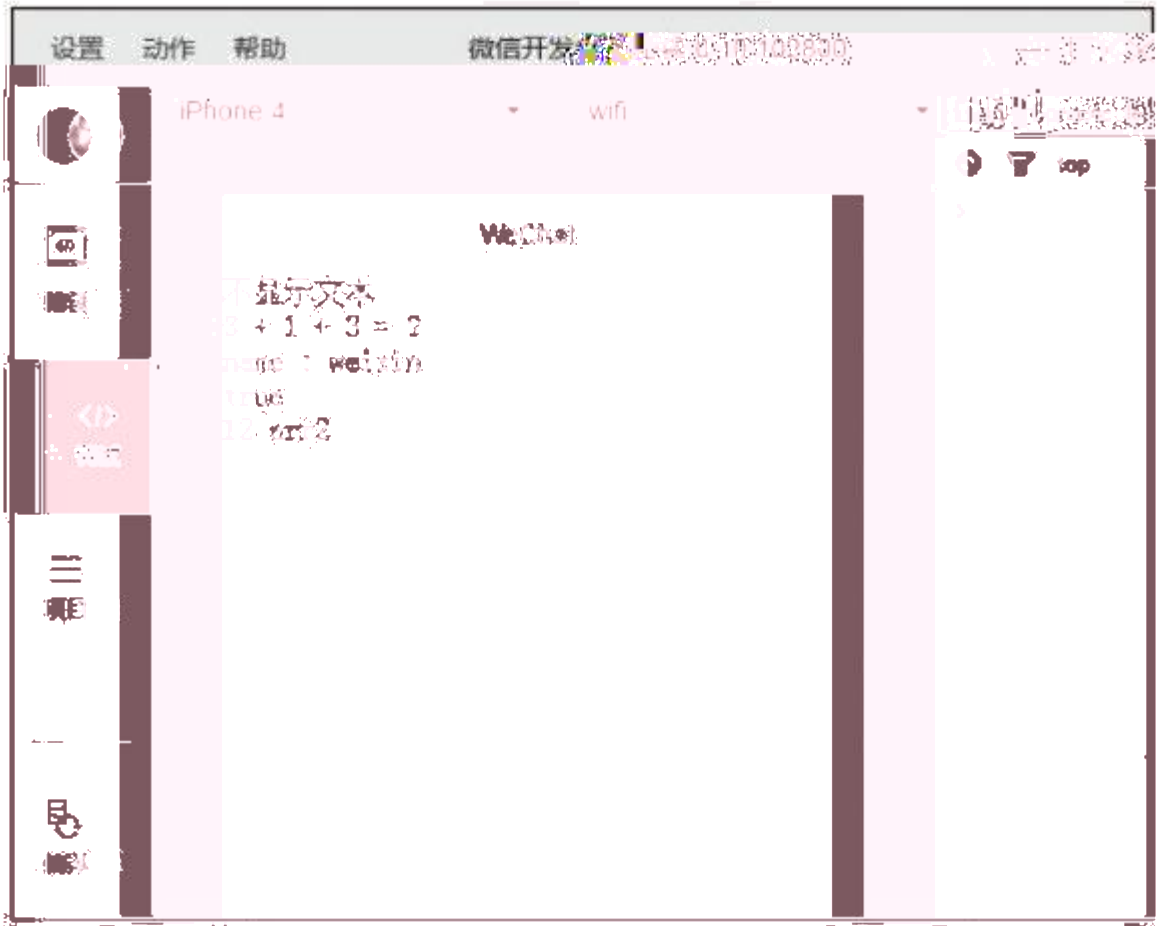


图2-9 界面渲染示意图

从图中我们能看出，WXML语言最终会转译为宿主端对应的语言，所以

直接作为字符串输出使用，可作用于内容、组件属性

```
<view>{{ showContent ? '显示文本': '不显示文本'}}</view>
<!-- 算数运算符 -->
<view>{{ num1 + num2 }} + 1 + {{ num3 }} = ? </view>
<!-- 字符串运算 -->
<view>{{ "name:" + name }}</view>
<!-- 逻辑判断
```

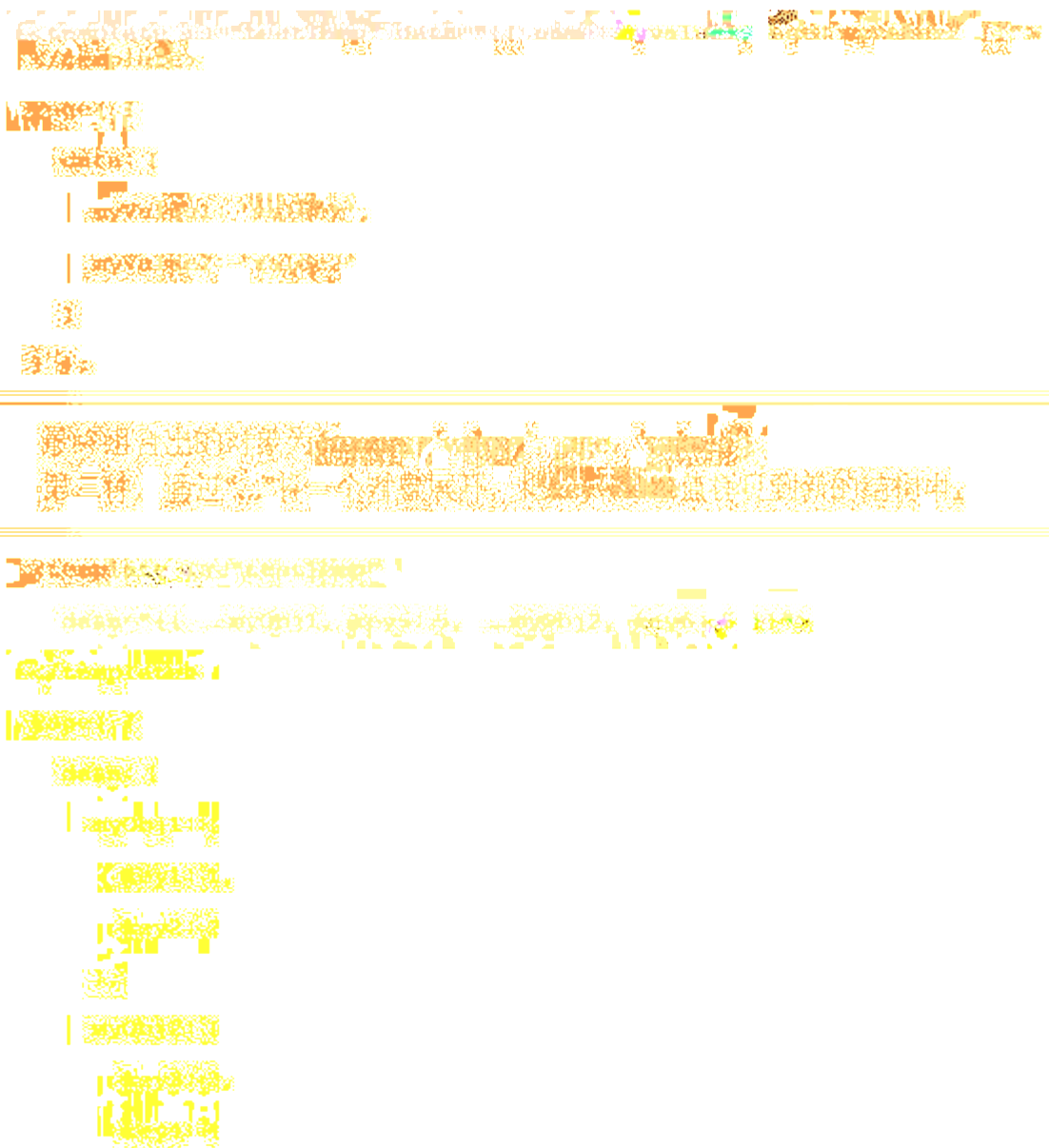

设置
动作
帮助
微信开发
iPhone 4
wifi

```
});
```

最终页面组合成的对象为[0，2，3，'stringtype']。

对象组合有3种组合方式，这里我们以数据注入模板为例。

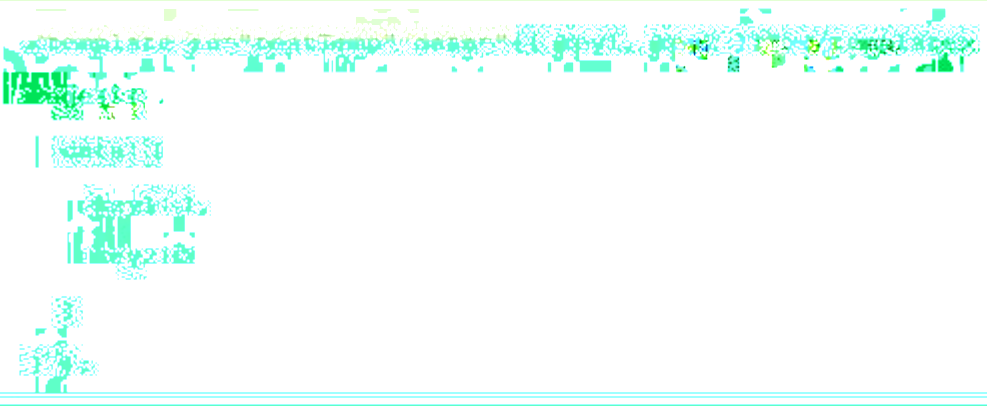
第一种 直接[illegible]

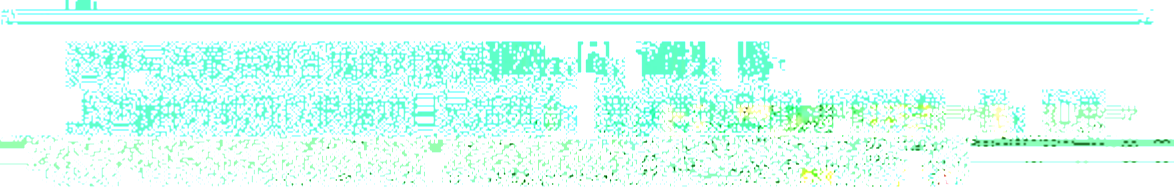


```
        }
    }
} );
```

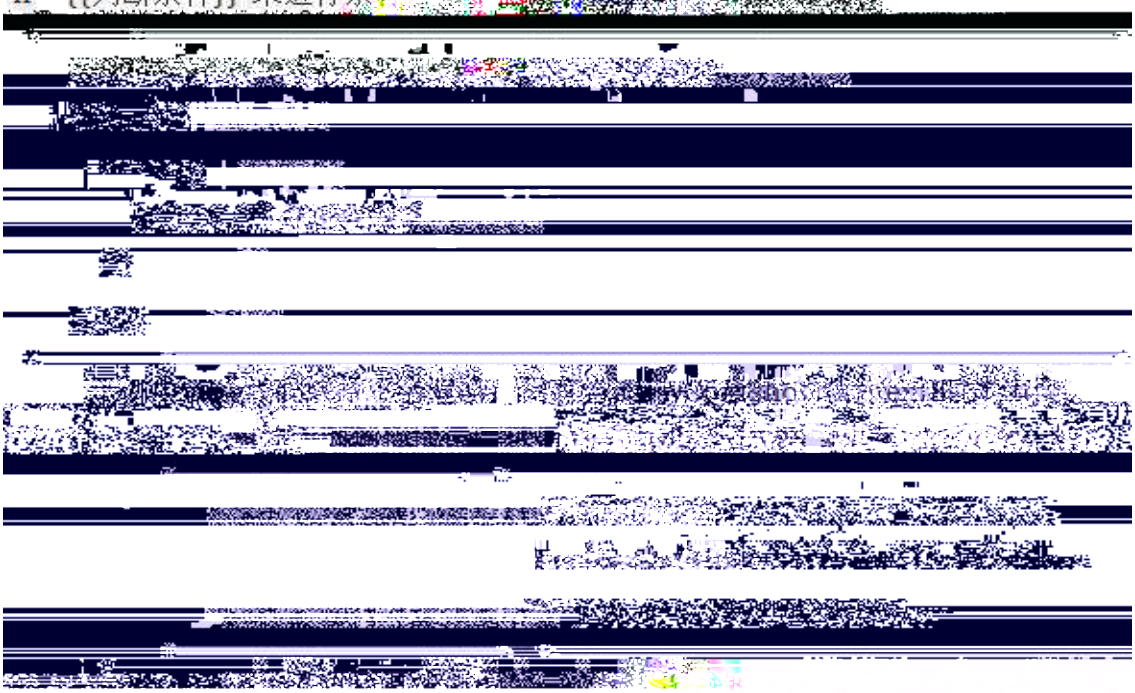
最终组合成的对象为{key1：1，key2：2，key5：5，key3：3 key4：4，key6：6}

第二种，如果对象key和value相同，可以只写key值，



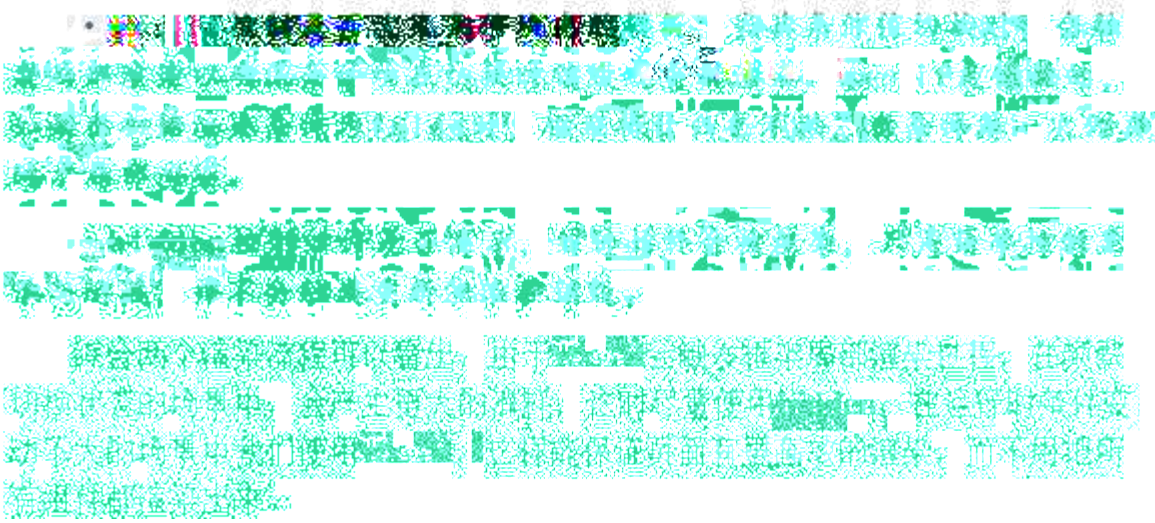


除了简单的数据绑定，我们常常会使用逻辑分支，这时候可以使用wx:if="{{判断条件}}"来进行条件渲染，当条件成立时渲染该代码块：



（3）wx：if与hidden

除了wx：if组件，也可以通过hidden属性控制组件是否显示，[illegible]疑问，这两种方式该怎样取舍，这里我们整理了两种方式的区别：



第11课 列表渲染

```
item="myItem">
  {{myIndex}}: {{myItem.name}}
```

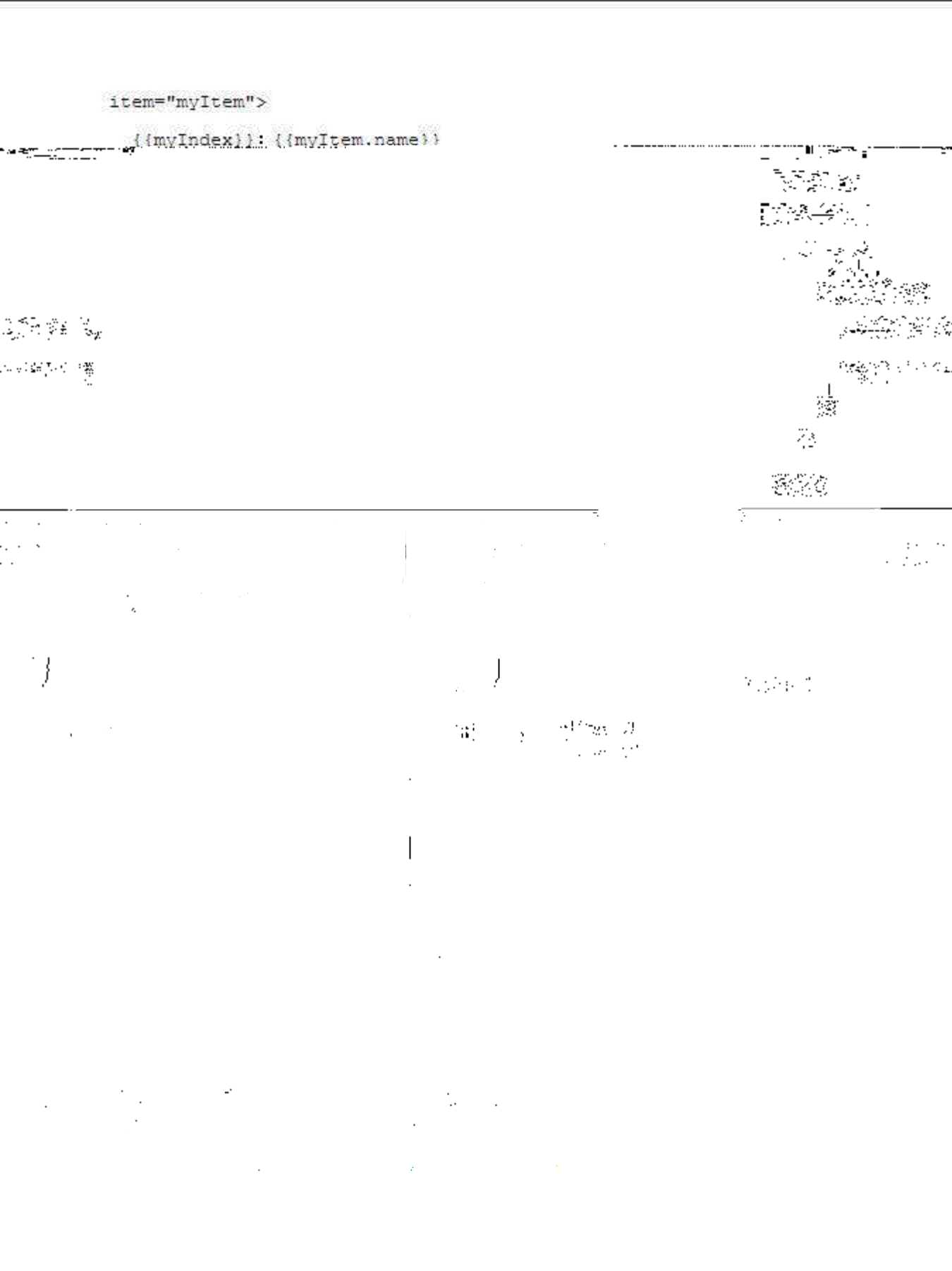


图2-12 列表渲染示例1

渲染结果如图2-13所示。

图2-13 列表渲染示例2

普通浏览中我们没必要修改index、item变量名，当wx:for嵌套使用时，就

```
    </block>
</view>
Page( {
  data: {
    myArray: [
      [1, 2, 3],
      [4, 5, 6],
      [7, 8, 9]
    ]
  }
} );
```

渲染效果如图2-14所示。

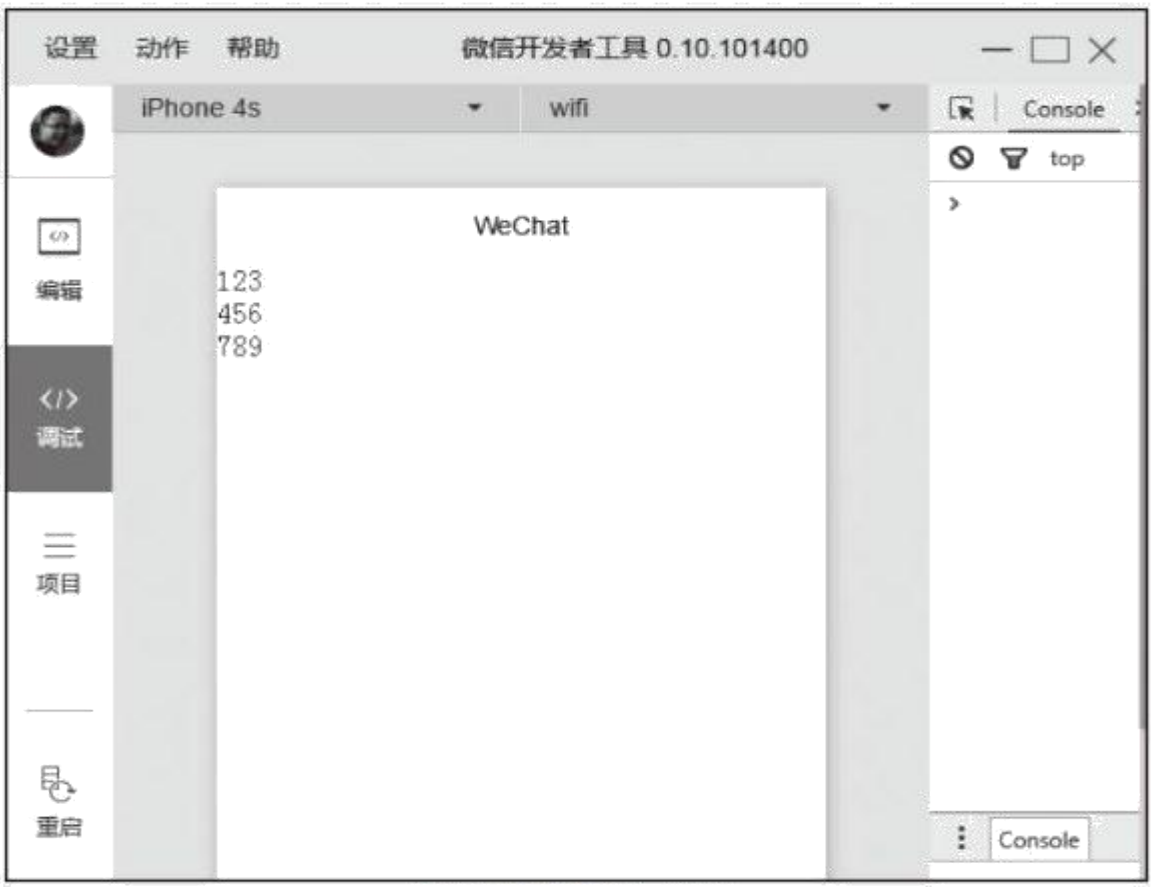


图2-14 列表渲染示例3

在本示例中，我们使用了<block/>标签，和block wx：if一样，wx：for可以

第12课 模板

在项目过程中，常常会遇到某些相同的结构在不同的地方反复出现，这时可

```
<view>内容</view>
<view>{{content}}</view>
</template>
```

（2）使用模板

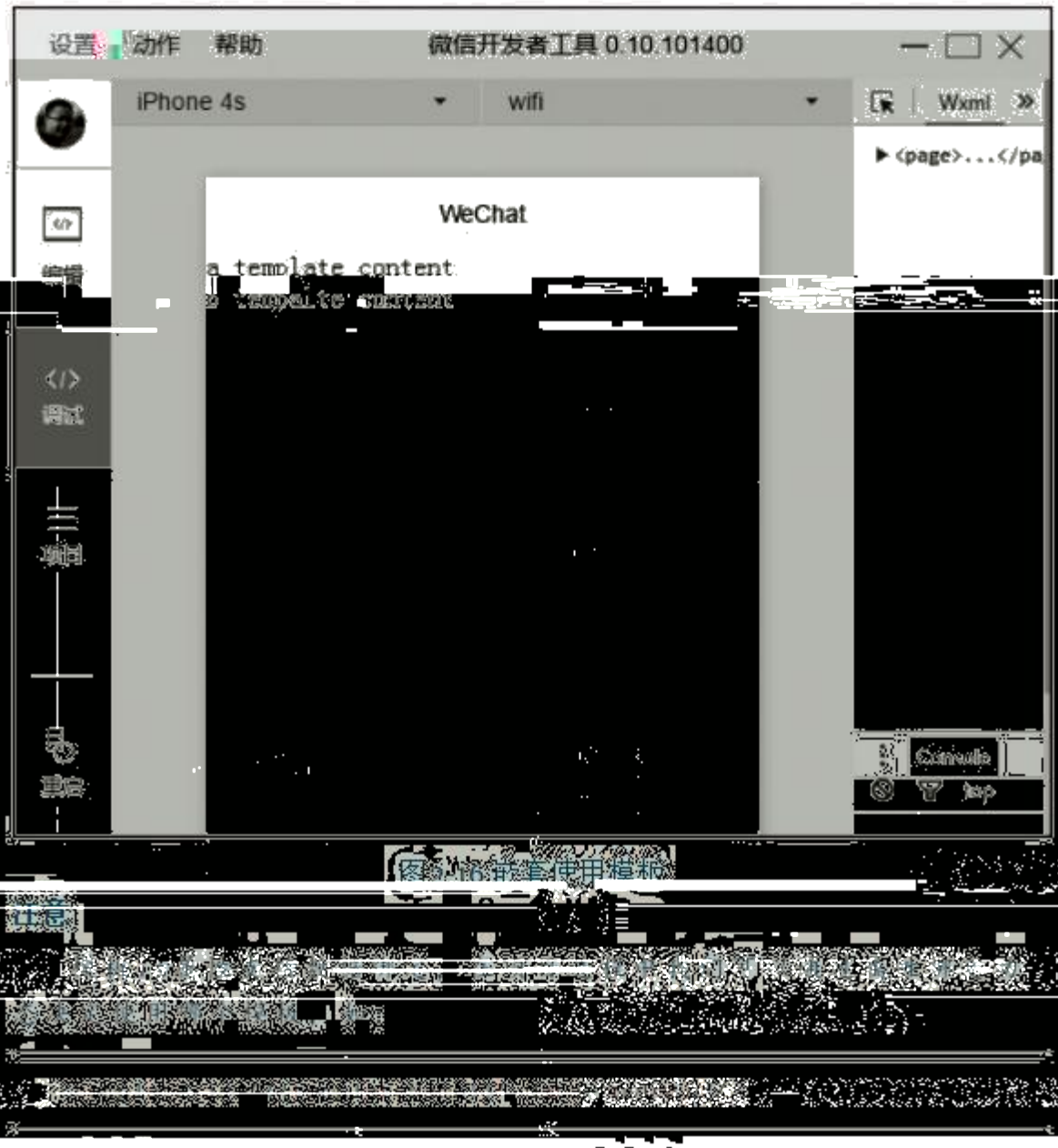
```
<template is="myTemplate" data="{{content:'内容', name

} );
```

执行效果如图2-15所示。

模板

图2-15 模板示例

渲染结果如图2-16所示。


设置
动作
帮助
微信开发者工具 0.10.101400
iPhone 4s
wifi
Wxml
<page>...</pa
WeChat
a template content
编辑
调试
项目
Console

第13课 事件

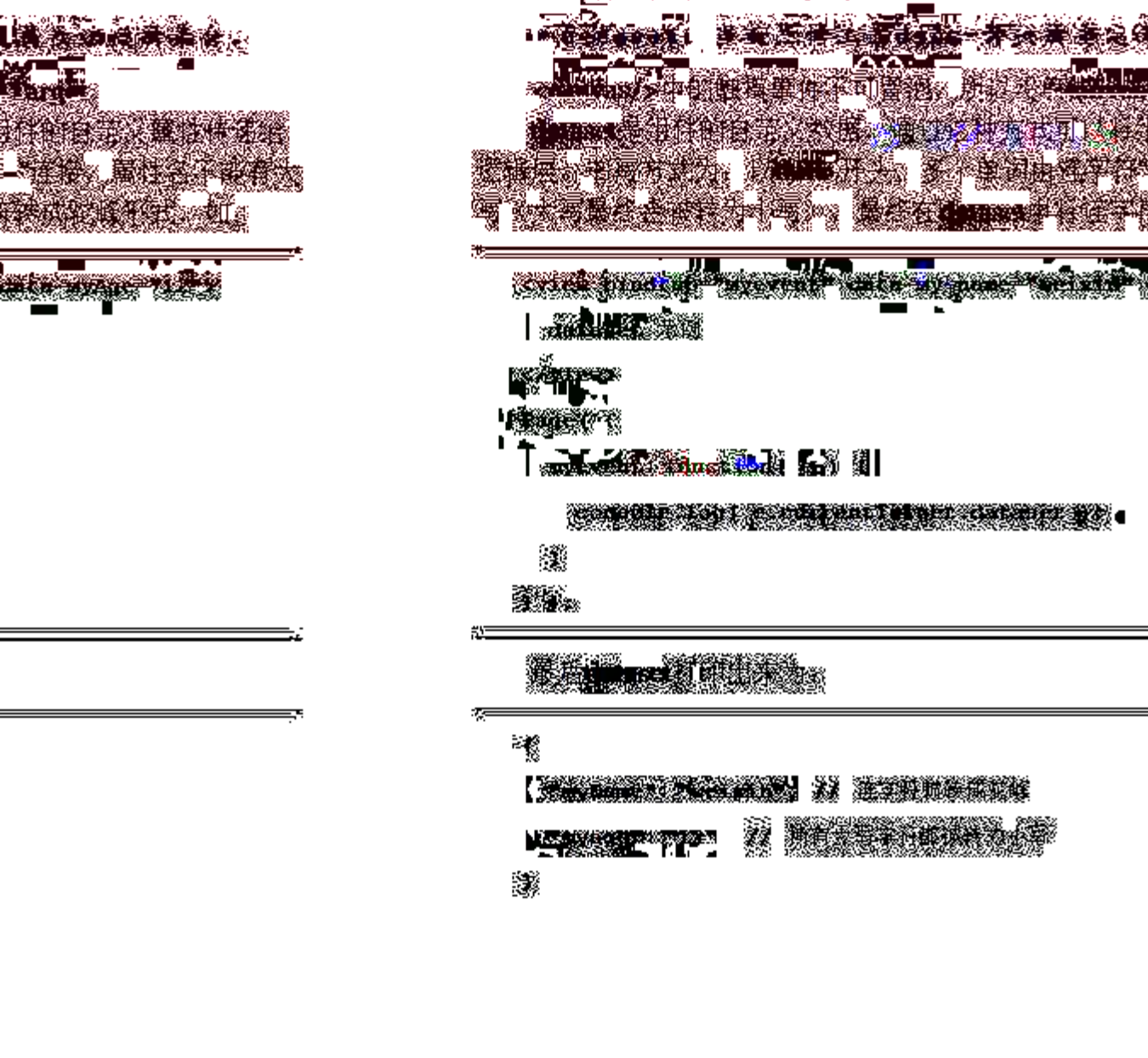
动态绑定。官方对WXML事件的定义如下：

- 事件是视

在之前内容中，已经多次实现事件绑定，大家应该比较熟悉了，事件绑定的

```
"touches": [
  {
    "identifier": 0,
    "pageX": 15,
```

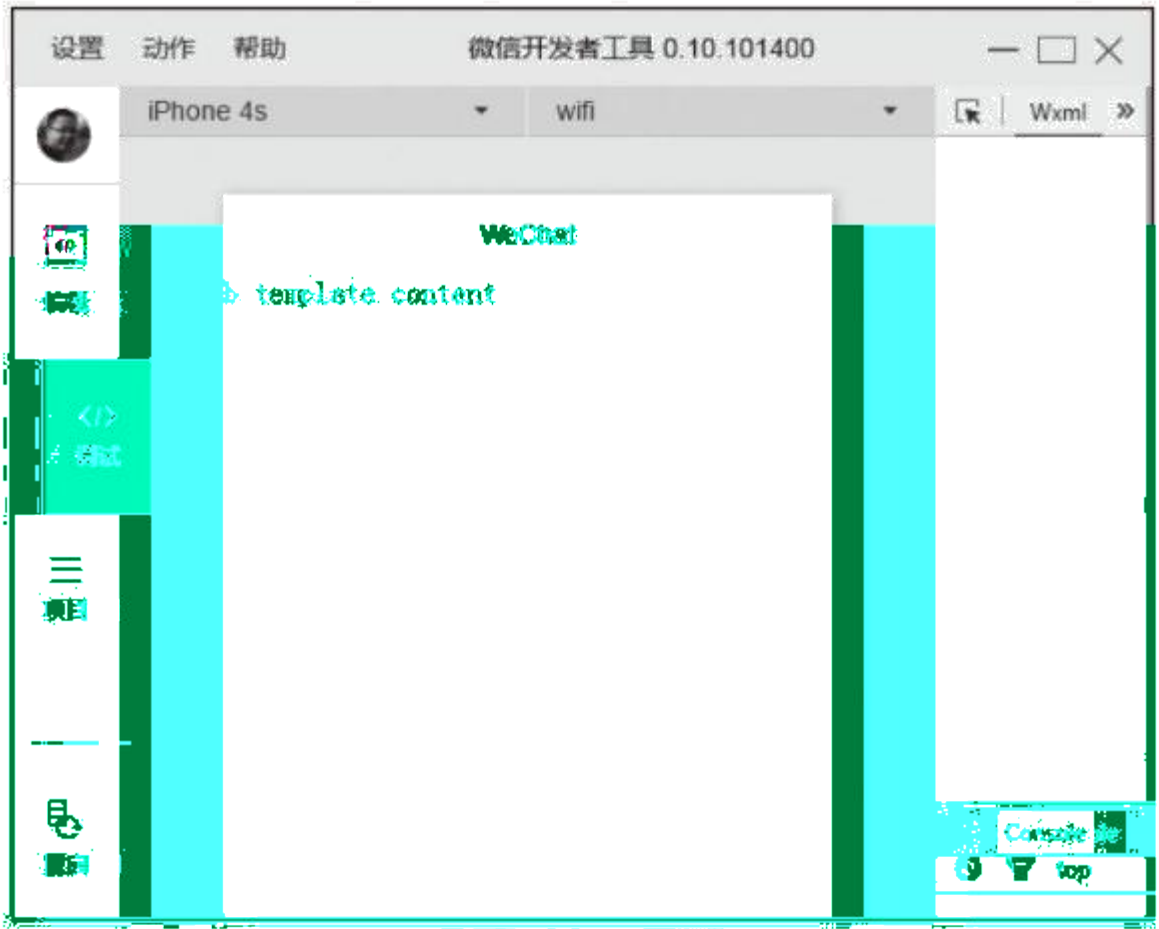

- id：事件源组件的id。
- tagName：事件源组件的类型。
- dataset：事件源组件上由data-开头的自定义属性组成的集合。



CustomEvent为自定义事件对象（继承BaseEvent），只有一个属性

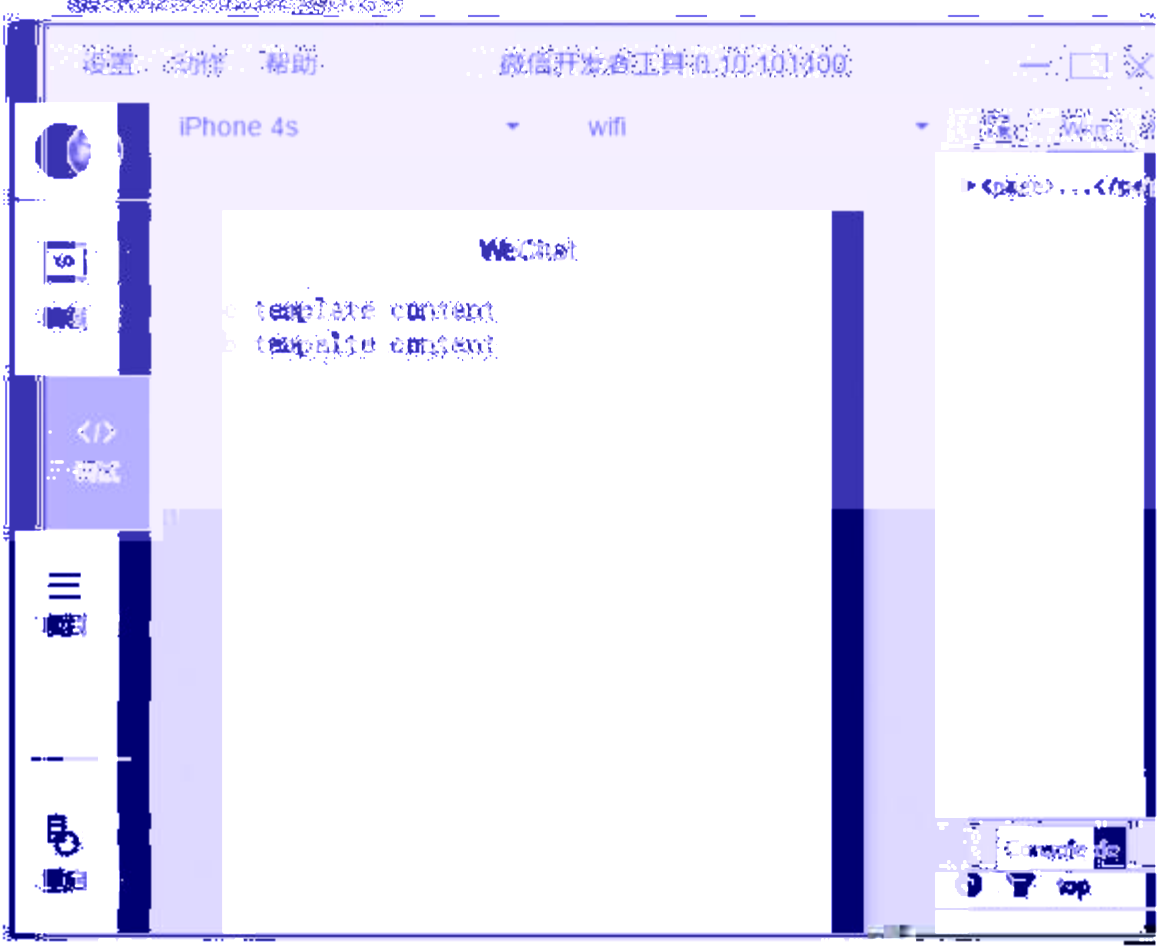
第14课 引用

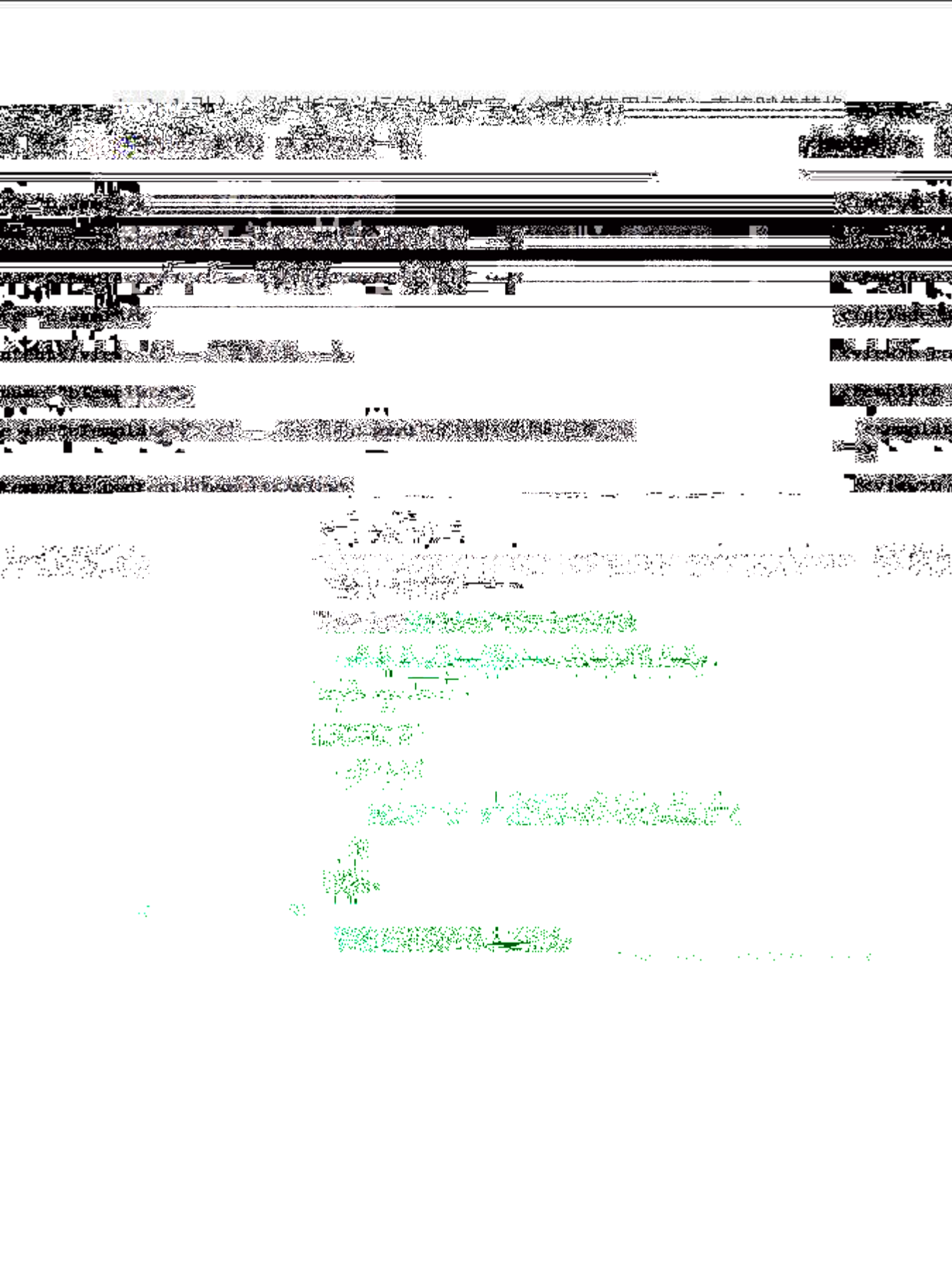
一个WXML可以通过import或include引入其他WXML文件，两种方式都能引入WXML文件，区别在于import引入WXML文件后只接受模板的定义，[illegible]则是引入文件中除<template/>以外的代码直接拷贝到<include/>位置，整体来说[illegible]

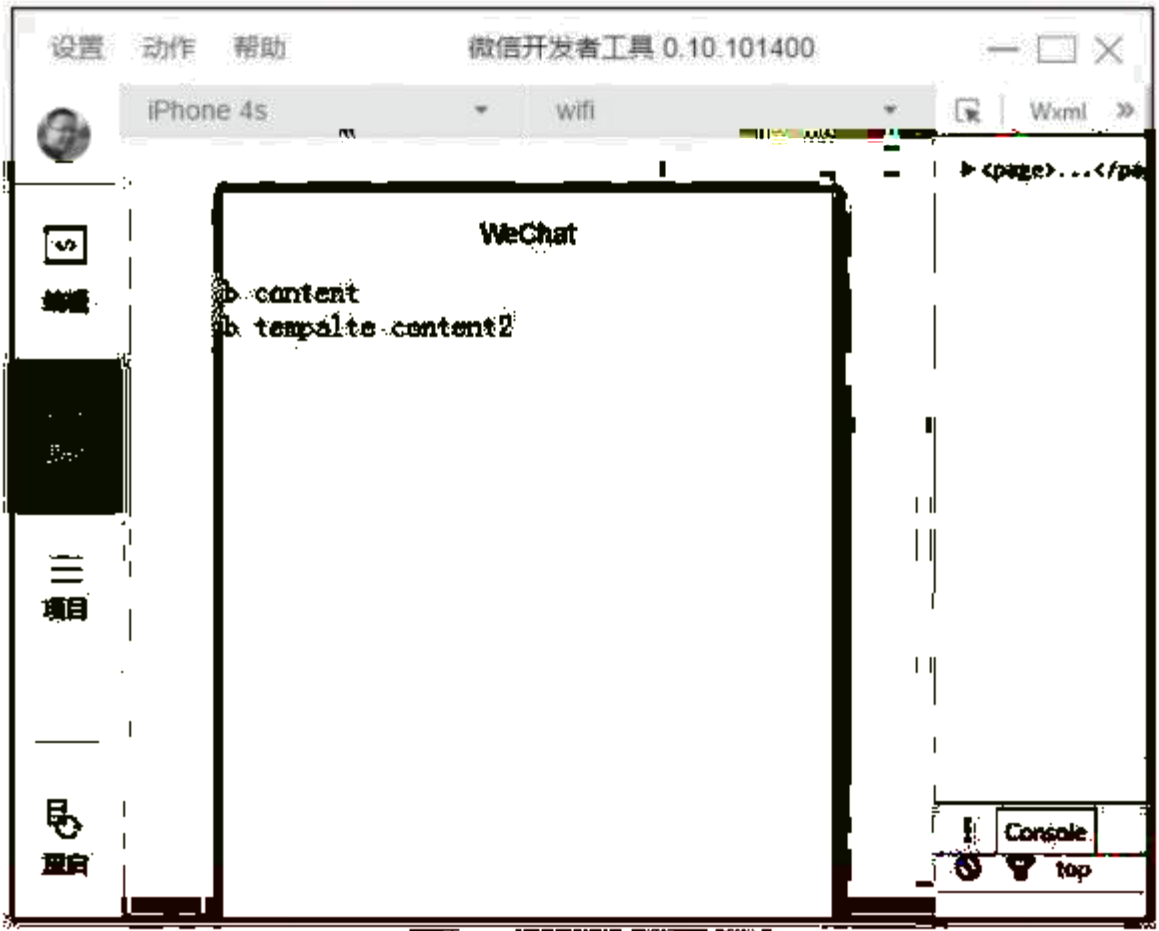

设置
动作
帮助
微信开发者工具 0.10.101400
iPhone 4s
wifi
Wxml
WeChat
template content
Console
top

```
  <template is="cTemplate"/>
  <view>b tempalte content</view>
</template>
<template is="cTemplate"/> <!-- import时被忽略 -->
<template name="cTemplate">
  <view>c template content</view>
</template>
```

渲染效果如图：






设置 动作 帮助
微信开发者工具 0.10.101400
iPhone 4s
wifi
Wxml
WeChat
content
template content2
项目
Console
top

第15课 页面样式文

2.4.4 页面样式文件（WXSS）

WXSS（WeiXin Style Sheets）是基于CSS拓展的样式语言，用于描述WXML的组件样式，决定WXML的组件该怎么显示，它具有CSS的大部分特性，在CSS基础上WXSS拓展了尺寸单位、样式导入特性，对CSS选择器属性上做了部分兼容，使用CSS属性

[redacted]

第17课 制作小程序

第18課 完