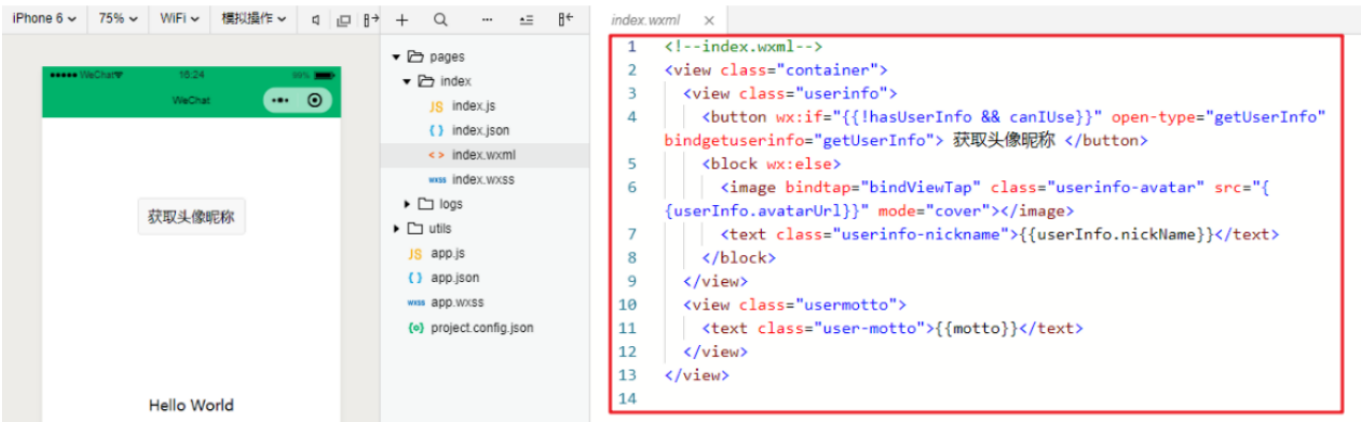
1.概述

官方文档： https://developers.weixin.qq.com/miniprogram/dev/framework/quickstart/code.html#WXML-%E6%A8%A1%E6%9D%BF

从事过网页编程的人知道，网页编程采用的是 HTML + CSS + JS 这样的组合，
其中 HTML 是用来描述当前这个页面的结构，
CSS 用来描述页面的样子，
JS通常是用来处理这个页面和用户的交互。

同样道理，在小程序中也有同样的角色，其中 WXML 充当的就是类似 HTML 的角色。
打开pages/index/index.wxml，你会看到以下的内容：



和 HTML 非常相似， WXML 由标签、属性等等构成。但是也有很多不一样的地方，我们来一一阐述一下：

---- 1.标签名字有点不一样
---- 2.多了一些 wx:if 这样的属性以及 {{表达式}} 的表示方式

注意： 微信小程序 借鉴了vue.js框架的一些优秀理念

2.数据绑定

开发文档：https://developers.weixin.qq.com/miniprogram/dev/reference/wxml/data.html

数据绑定使用 Mustache 语法（双大括号）将变量包起来
---- 通过 {{ }} 的语法把一个变量绑定到界面上，我们称为数据绑定。

---- WXML 中的动态数据均来自对应 js文件Page函数 的 data 对象 ,这样实现了wxml与 JavaScript之间的数据传递，即数据绑定

小程序提倡把渲染和逻辑分离，简单来说就是不要再让 js 直接操控 DOM ，
js 只需要管理状态即可，然后再通过一种模板语法来描述状态和界面结构的关系即可。

==WXML语法：==

```
<标签>{{变量名}}</标签>
```

==常见的数据绑定形式==

内容

```
<view> {{ message }} </view>


Page({
    data: {
        message: 'Hello 双11!'
    }
})
```

组件属性(需要在双引号之内)

```
<view id="item-{{id}}"> </view>


Page({
    data: {
        id: 0
    }
})
```

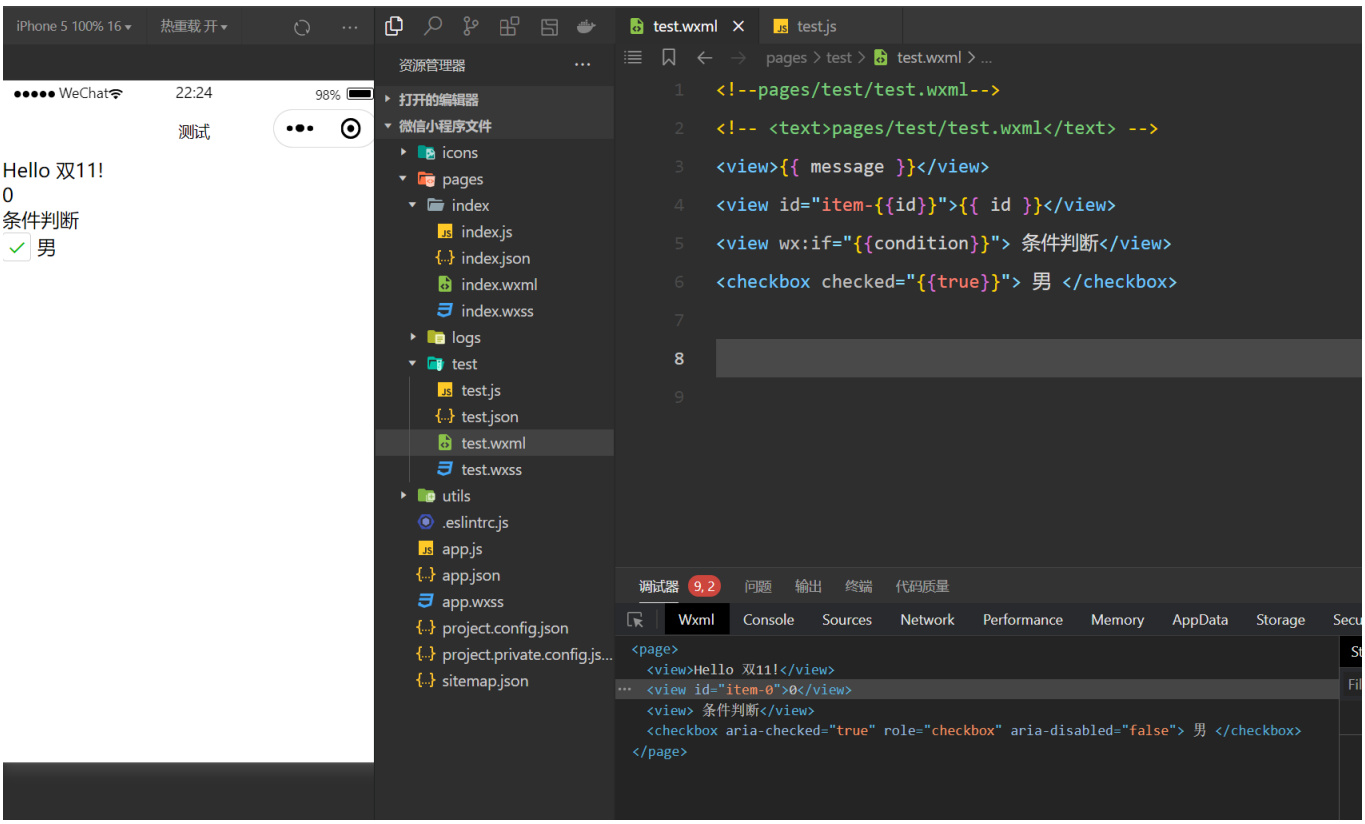
控制属性(需要在双引号之内)

```
<view wx:if="{{condition}}"> 条件判断</view>

Page({
    data: {
        condition: true
    }
})
```

关键字(需要在双引号之内)

```
true : boolean 类型的      true，代表真值。
false :  boolean 类型的    false，代表假值。


<checkbox checked="{{false}}"> 男 </checkbox>
```

特别注意：
    不要直接写 checked="false" ，
    其计算结果是一个字符串，转成 boolean 类型后代表真值。



3.运算

开发文档： https://developers.weixin.qq.com/miniprogram/dev/reference/wxml/data.html
可以在 {{}} 内进行简单的运算，支持的有如下几种方式：

==三元运算==

```
<view hidden="{{num>5 ? true : false}}"> Hidden </view>

  data: {
    num: 4,
  },
```

hidden：true,则隐藏,为false时显示

? : 问号和冒号是一起用的，叫条件运算符。

语法：
条件表达式 ? 真值 : 假值
这个表式式由三部分组成的，如果条件表达式的值为真，则整个表达式的值为“真值”的值，反之为“假值”的值。

==算术运算==

```
<view> {{a + b}} + {{c}} + d </view>

Page({
    data: {
        a: 1,
        b: 2,
        c: 3
    }
})
```

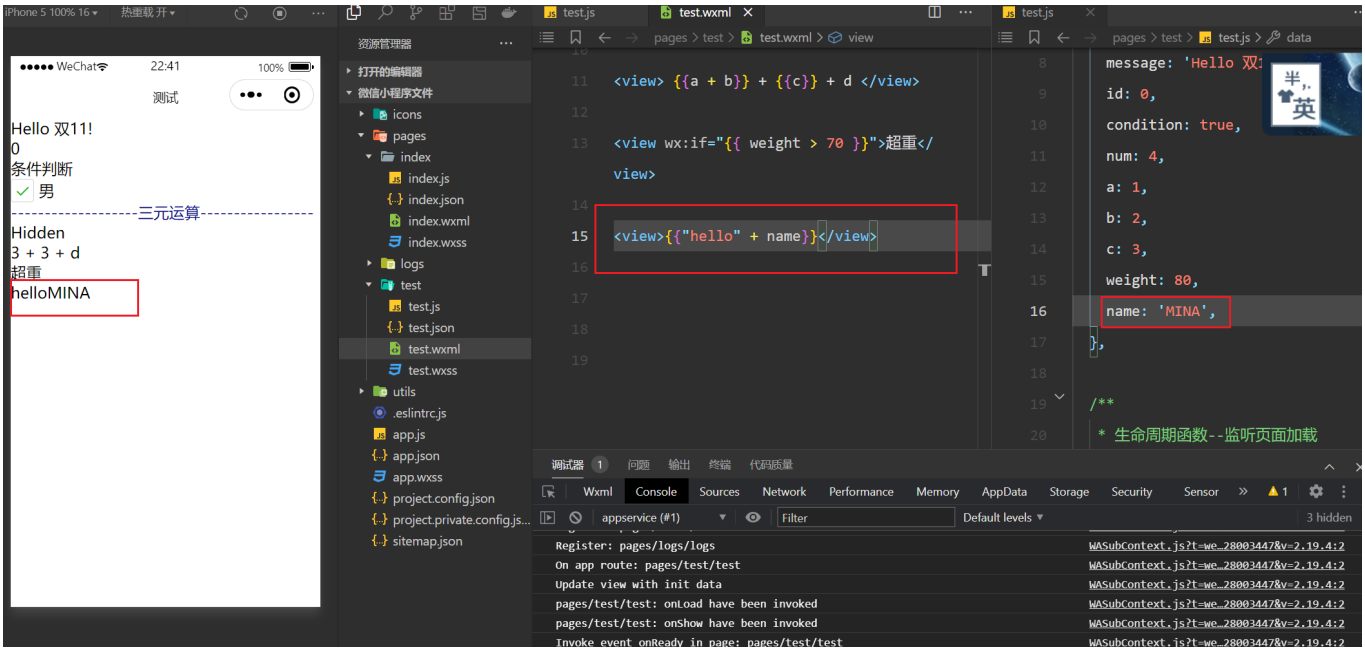
```
view中的内容为 3 + 3 + d 。
```

==逻辑判断==

```
<view wx:if="{{weight >70}}"> 超重</view>


Page({
    data: {
        weight: 80
    }
})
```

==字符串运算==

```
<view>{{"hello" + name}}</view>


Page({
    data:{
        name: 'MINA'
    }
})
```



==对象==

```
<view>{{student.name}}</view>


Page({
    data: {
        student: {
        name: "张三",
        age: 20,
        gender: "男"
    }
}
```

4.列表渲染

开发文档：https://developers.weixin.qq.com/miniprogram/dev/reference/wxml/list.html

语法：

<标签名 wx:for="{{变量名}}" [wx:for-index="自定义名称" wx:for-item="自定义名称"
wx:key="自定义名称"]>

--- wx:for-index 可以指定数组当前 下标 的变量名，默认名为：index
--- wx:for-item 可以指定数组当前 元素 的变量名（遍历对象的接收变量名），默认名为：item
--- wx:key 可以定义也可以不定义，唯一的标识符 详解--
https://blog.csdn.net/qq_45593068/article/details/118658779
wx:key 官网解释：https://developers.weixin.qq.com/miniprogram/dev/reference/wxml/list.html#wx:key
https://www.cnblogs.com/kenshinobiy/p/9206072.html

1： `wx:key="字符串"`

这个”字符串”代表在 for 循环的 array 中 item 的某个“属性”
该“属性” 的值需要是列表中唯一的字符串或数字，且不能动态改变。
用于被遍历的组件需要多个属性的时候。

```
//test.js
data: {
input_data: [
{ id: 1, unique: "unique1" },
{ id: 2, unique: "unique2" },
{ id: 3, unique: "unique3" },
{ id: 4, unique: "unique4" },
]
}

//test.wxml
<input value="id:{{item.id}}" wx:for="{{input_data}}" wx:key="unique" />
```

2： `wx:key="*this"`

保留关键字”*this”代表在 for 循环中的 item 本身，
这种表示需要 item 本身是一个唯一的字符串或者数字
用于组件仅需要一个属性，且属性值唯一。

```
//test.js
data: {
numberArray: [1, 2, 3, 4],
stringArray:['aaa','ccc','fff','good']
}
//test.wxml
<input value="id:{{ item }}" wx:for="{{numberArray}}" wx:key="*this" />
<input value="id:{{ item }}" wx:for="{{stringArray}}" wx:key="*this" />

},
```

==**案例**==

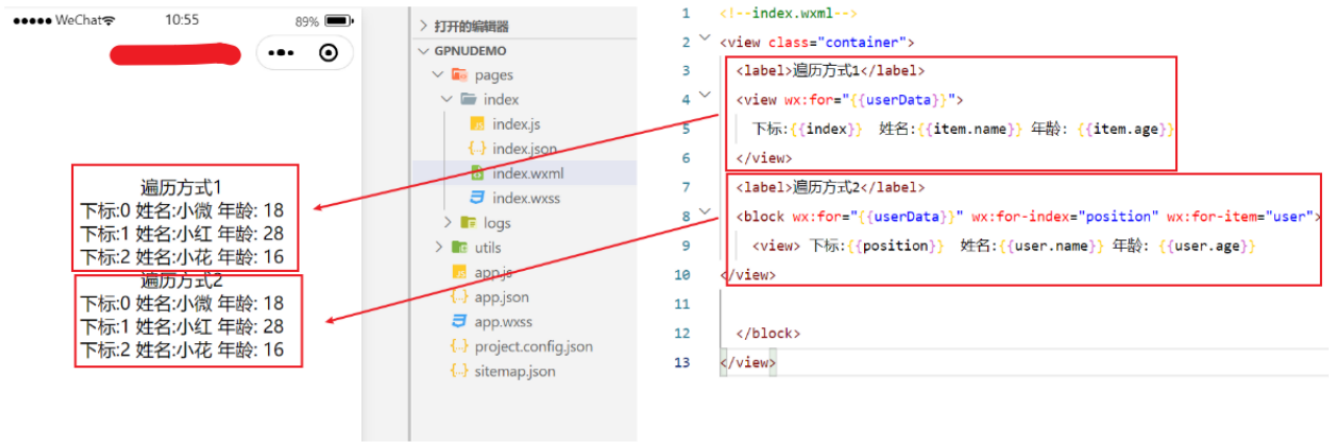
首先在 index.js 后缀的文件中设定数组数据

```
Page({
    data: {
        // 定义变量
        userData: [{
                name: "小微",
                age: 18,
                height: 169
            },
            {
                name: "小红",
                age: 28,
                height: 165
            },
            {
                name: "小花",
                age: 16,
                height: 155
            }
        ]
    },
})
```

然后在 index.wxml 为后缀的模板文件中使用列表渲染相当于循环遍历出userData的数据

```
<view class="container">
    <label>遍历方式1</label>
    <view wx:for="{{userData}}">
        下标:{{index}} 姓名:{{item.name}} 年龄: {{item.age}}
    </view>

    <label>遍历方式2</label>
    <block wx:for="{{userData}}" wx:for-index="position" wx:for-item="user">
        <view>
            下标:{{position}} 姓名:{{user.name}} 年龄: {{user.age}}
        </view>
    </block>
</view>
```

效果



注意1:
当 wx:for 的值为字符串时，会将字符串解析成字符串数组

```
<view wx:for="array">
    {{item}}
</view>
```

等同于

```
<view wx:for="{{['a','r','r','a','y']}}">
    {{item}}
</view>
```

注意2:
花括号和引号之间如果有空格，将最终被解析成为字符串

```
<view wx:for="{{[10,20,30]}} ">
    {{item}}
</view>
```

等同于

```
<view wx:for="{{[10,20,30] + ' '}}">
    {{item}}
</view>
```

5.条件渲染

开发文档：https://developers.weixin.qq.com/miniprogram/dev/reference/wxml/conditional.html

语法：
<标签名 wx:if="{{condition}}">

在框架中，使用 wx:if="{{condition}}" 来判断是否需要渲染该代码块，
也可以用 wx:elif 和 wx:else 来添加一个 else 块。

==**案例**==

1.在 *.js 文件中添加一个数据变量

2.然后直接在 *.wxml 模板页面中使用条件渲染

```
<!--pages/users/users.wxml-->
<view>
    <block wx:if="{{score>=90}}">
        <view>分数：{{score}}</view>
        <view>等级：优秀</view>
    </block>
    <block wx:elif="{{score>=80}}">
        <view>分数：{{score}}</view>
        <view>等级：良好</view>
    </block>
    <block wx:elif="{{score>=70}}">
        <view>分数：{{score}}</view>
```

```
        <view>等级：合格</view>
    </block>
    <block wx:elif="{{score>=60}}">
        <view>分数：{{score}}</view>
        <view>等级：及格</view>
    </block>
    <block wx:else>
        <view>分数：{{score}}</view>
        <view>等级：不及格</view>
    </block>
</view>
```

3.效果