第 5 章　莫凡商城首页动态绑定设计

课时内容	莫凡商城首页动态绑定设计	课时	2
教学目标	熟悉微信小程序模板与引用 了解 WXS 小程序脚本语言 掌握下拉刷新和监听事件的功能		
教学重点	微信小程序模板与引用 下拉刷新和监听事件的功能		
教学难点	下拉刷新和监听事件的功能		
教学设计	1. 教学思路：介绍微信小程序模板与引用、WXS 小程序脚本语言；通过多媒体演示和实机操作讲解下拉刷新和监听事件的功能。 2. 教学手段：多媒体+计算机 3. 教学资料：教材、多媒体课件		

教学内容

5.3 微信小程序定义模板

微信小程序在 WXML 页面上提供模板（template）功能，是为了解决一些重复代码和布局设计的问题，把这些重复的代码抽取出来，放在模板里，当用到相同功能和布局页面时，就可以达到复用的效果。例如，导航菜单、版权信息等各个页面都共用的功能，就可以放在模板里。

5.3.1 定义模板

使用<template/>标签来定义模板代码片段，使用 template 模板的 name 属性作为模板的名字，页面在调用模板的时候，可以根据名字找到相应的模板，同时模板也可以接收传递过来的数据，如 {{index}}、{{msg}}、{{time}}都是接收的数据，示例代码如下。

```
<template name="msgItem">
  <view>
    <text> {{index}}: {{msg}} </text>
    <text> Time: {{time}} </text>
  </view>
</template>
```

5.3.2 使用模板

在 WXML 页面文件里，使用 is 属性找到要引入的模板名称，如 msgItem 就是模板的名称，使用 data 属性来传递模板所需要的数据，示例代码如下。

```
<!-- wxml -->
<template is="msgItem" data="{{item}}"/>
```

```
<!-- js -->
Page({
  data: {
    item: {
      index: 100,
      msg: '我是一个模板',
      time: '2019-09-15'
    }
  }
})
```

is 属性可以使用三元运算语法，从而动态决定具体需要渲染哪个模板。下面定义两套模板，奇数使用第一套模板，偶数使用第二套模板。

```
<template name="first">
  <view> 我是第一套模板 </view>
</template>
<template name="second">
  <view> 我是第二套模板 </view>
</template>

<block wx:for="{{[1, 2, 3, 4, 5]}}">
    <template is="{{item % 2 == 0 ? 'first' : 'second'}}"/>
</block>
```

5.4 微信小程序的引用功能

微信小程序的引用有两种方式：import 引用和 include 引用。import 是引用模板文件，在 WXML 页面文件定义模板后，就可以使用 import 将模板引入页面；include 是直接将 WXML 页面内容引入到另一个页面里，但是它不能引用<template/>模板内容。

5.4.1 import 引用

import 可以将<template/>模板引入到页面中使用。

假如在 temp.wxml 文件中定义了一个叫 msg 的模板，引用的示例代码如下。

```
<!-- temp.wxml -->
<template name="msg">
  <text>我是模板内容</text>
<text>{{text}}</text>
</template>
```

在 index.wxml 文件中引用了 temp.wxml 文件，就可以使用模板了，示例代码如下。

```
<import src="temp.wxml"/>
<template is="msg" data="{{text: '你好'}}"/>
```

注意 import 作用域的问题，假如 C import B，B import A，则在 C 中可以使用 B 定义的模板，在 B 中可以使用 A 定义的模板，但是 C 不能使用 A 定义的模板。

5.4.2 include 引用

include 可以将 WXML 页面文件的整个代码引入到目标文件里，但是不能引入<template/>模板文件，相当于是将 WXML 文件拷贝到 include 位置，示例代码如下。

```
<!-- index.wxml -->
<include src="header.wxml"/>
<view> body </view>
<include src="footer.wxml"/>

<!-- header.wxml -->
<view> 我是头部信息 </view>

<!-- footer.wxml -->
<view> 我是版权信息 </view>
```

5.5 WXS 小程序脚本语言

WXS（WeiXin Script）是小程序的一套脚本语言，结合 WXML 页面文件，可以构建出页面的结构。它是把原来放在 JS 文件里进行处理的逻辑，直接放在 WXML 页面文件里进行处理。它有两种使用方式：一种是将 WXS 脚本语言嵌入到 WXML 页面文件里，用来在 WXML 文件中的<wxs>标签内处理相关逻辑；另一种是作为以.wxs 为后缀名的文件独立存在，然后再引入到 WXML 页面文件里使用。

1. 嵌入 WXML 页面文件

嵌入 WXML 页面文件的使用方法示例代码如下。

```
<!--wxml-->
<wxs module="m1">
var msg = "hello world";

module.exports.message = msg;
```

```
</wxs>

<view> {{m1.message}} </view>
```

2. 独立为 WXS 文件

在指定的项目目录里面，单击鼠标右键，在弹出的快捷菜单中选择“新建 WXS”可以创建.wxs 文件，如图所示。

创建.wxs 脚本语言文件

示例代码如下。

```
// /pages/tools.wxs
var foo = "'hello world' from tools.wxs";
var bar = function(d) {
  return d;
}
module.exports = {
  FOO: foo,
  bar: bar,
};
module.exports.msg = "some msg";
```

```
<!-- page/index/index.wxml -->
<wxs src="./../tools.wxs" module="tools" />
<view> {{tools.msg}} </view>
<view> {{tools.bar(tools.FOO)}} </view>
```

5.5.1 模块化

WXS 代码无论编写在 WXML 文件中的 <wxs> 标签内还是以.wxs 为后缀名的文件内，都是以单独的模块形式存在的。在一个模块中定义的变量与函数，默认为私有，对其他模块不可见。一个模块要想对外暴露其内部的私有变量与函数，只能通过 module.exports 来实现，示例代码如下。

```
// /pages/comm.wxs
var foo = "'hello world' from comm.wxs";
var bar = function(d) {
  return d;
}
module.exports = {
  foo: foo,
  bar: bar
};
```

在.wxs 模块中引用其他 WXS 文件模块，可以使用 require 函数。在 WXS 文件里只能引用.wxs 文件模块，且必须使用相对路径。.wxs 文件模块均为单例，多个页面、多个地方、多次引用，使用的都是同一个.wxs 文件模块对象，如果一个.wxs 文件模块在定义之后，一直没有被引用，则该模块不会被解析与运行，示例代码如下。

```
// /pages/tools.wxs
var foo = "'hello world' from tools.wxs";
var bar = function (d) {
  return d;
}
```

```
module.exports = {
  FOO: foo,
  bar: bar,
};
module.exports.msg = "some msg";
```

```
// /pages/logic.wxs
var tools = require("./tools.wxs");
console.log(tools.FOO);
console.log(tools.bar("logic.wxs"));
console.log(tools.msg);
```

```
<!-- /page/index/index.wxml -->
<wxs src="./../logic.wxs" module="logic" />
```

5.5.2 变量与数据类型

1. 变量的使用

WXS 中的变量均为值的引用，如果只声明变量而不赋值，则默认值为 undefined，示例代码如下。

```
var foo = 1;
var bar = "hello world";
var i; // i === undefined
```

变量名的命名规则如下。

（1）首字符必须是字母（a～z，A～Z）、下划线（_）。

（2）剩余字符可以是字母（a～z，A～Z）、下划线（_）、数字（0～9）。

（3）保留标识符 delete、void、typeof、null、undefined、NaN、Infinity、var、if、else、true、false、require、this、function、arguments、return、for、while、do、break、continue、switch、case、default 不能作为变量名。

2. 数据类型

WXS 小程序脚本语言支持的数据类型为 number（数值类型）、string（字符串类型）、boolean（布尔值类型）、object（对象类型）、function（函数类型）、array（数组类型）、date（日期类型）、regexp（正则类型）。

（1）number 数值类型包括两种数值：整数和小数，用法如下。

```
var a = 10;
var PI = 3.141592653589793;
```

（2）string 字符串类型有以下两种写法。

```
'hello world';
"hello world";
```

（3）boolean 布尔值类型只有两个特定的值：true 和 false。

（4）object 对象类型是一种无序的键值对，使用方法如下。

```
var o = {} //生成一个新的空对象
//生成一个新的非空对象
o = {
  'string'  : 1,  //object 的 key 可以是字符串
  const_var : 2,  //object 的 key 也可以是符合变量定义规则的标识符
  func      : {}, //object 的 value 可以是任何类型
};
//对象属性的读操作
console.log(1 === o['string']);
console.log(2 === o.const_var);

//对象属性的写操作
o['string']++;
o['string'] += 10;
o.const_var++;
o.const_var += 10;

//对象属性的读操作
console.log(12 === o['string']);
console.log(13 === o.const_var);
```

（5）function 函数类型支持以下定义方式。

```
//方法 1
function a (x) {
```

```
  return x;
}

//方法 2
var b = function (x) {
  return x;
}
```

function 同时也支持以下语法(匿名函数、闭包等)。

```
var a = function (x) {
  return function () { return x;}
}
```

（6）array 数组类型支持以下定义方式。

```
var a = [];    //生成一个新的空数组
a = [1, "2", {}, function(){}];  //生成一个新的非空数组，数组元素可以是任何类型
```

（7）date 日期类型生成 date 对象需要使用 getDate 函数，返回一个当前时间的对象，示例代码如下。

```
getDate()
getDate(milliseconds)
getDate(datestring)
getDate(year, month[, date[, hours[, minutes[, seconds[, milliseconds]]]]])
参数
milliseconds: 从 1970 年 1 月 1 日 00:00:00 UTC 开始计算的毫秒数
datestring: 日期字符串，其格式为:"month day, year hours:minutes:seconds"
```

（8）regexp 正则类型生成 regexp 对象需要使用 getRegExp 函数，示例代码如下。

```
getRegExp(pattern[, flags])
参数:
pattern: 正则表达式的内容。
flags:修饰符。该字段只能包含以下字符:
g: global
i: ignoreCase
m: multiline。
```

5.5.3 注释

WXS 小程序脚本语言注释有 3 种方式：单行注释、多行注释、结尾注释，示例代码如下。

```
<wxs module="sample">
// 方法一:单行注释
//var name = "小刚";

//方法二:多行注释
/*
var a = 1;
var b = 2;
*/

//方法三:结尾注释。即从 /* 开始往后的所有 WXS 代码均被注释
/*
var a = 1;
var b = 2;
var c = "fake";

</wxs>
```

5.5.4 语句

WXS 微信小程序脚本语言里，可以使用 if 条件语句、switch 条件语句、for 循环语句和 while 循环语句。

1. if 语句

在 WXS 中，可以使用以下格式的 if 语句：if…else if…else statementN。通过该句型，可以在 statement1～statementN 中选择其中一个执行，示例语法如下。

```
if (表达式) {
  代码块;
} else if (表达式) {
  代码块;
} else if (表达式) {
  代码块;
} else {
  代码块;
}
```

```
var age = 10;
if(age < 18){
  console.log( “未成年”);
}else if(age < 28){
  console.log( “青年”);
}else{
  console.log( “壮年”);
}
```

2. switch 语句

switch 语句根据表达式的值与 case 变量值做比较，哪个 case 变量值与表达式的值相等就执行哪个 case 语句，default 分支可以省略不写，case 关键词后面只能使用变量、数字和字符串。如果不写 break 结束语句，程序就会向下继续执行其他满足条件的 case 语句，示例语法如下。

```
switch (表达式) {
  case 变量:
    语句;
  case 数字:
    语句;
    break;
  case 字符串:
    语句;
  default:
    语句;
}

var exp = 10;
switch ( exp ) {
case "10":
  console.log("string 10");
  break;
case 10:
  console.log("number 10");
  break;
case exp:
  console.log("var exp");
  break;
default:
  console.log("default");
}
```

3. for 循环语句

for 循环语句用来遍历集合，支持使用 break、continue 关键词，示例语法如下。

```
for (语句; 语句; 语句) {
  代码块;
}
for (var i = 0; i < 3; ++i) {
  console.log(i);
  if( i >= 1) break;
}
```

4. while 循环语句

while 循环语句当表达式为 true 时，循环执行语句或代码块，支持使用 break、continue 关键词，示例语法如下。

```
while (表达式){
  代码块;
}

do {
  代码块;
} while (表达式)
```

5.6 下拉刷新及窗口设置

页面下拉刷新是经常会用到的一个功能，有了这个功能，我们就可以通过刷新页面来获取更多的数据。微信小程序也提供了开始下拉刷新、停止下拉刷新及动态设置背景色等功能。要实现下拉刷新效果，就需要在小程序公共设置 app.json 文件里或者在各个页面的.json 文件里配置 enablePullDownRefresh=true，提供两个事件：onPullDownRefresh() 用户下拉刷新事件、onReachBottom() 用户上拉触底事件；提供两个 API 接口：

wx.startPullDownRefresh 开始下拉刷新、wx.stopPullDownRefresh 停止当前页面下拉刷新。

5.6.1 下拉刷新 API 及事件

1. wx.startPullDownRefresh 开始下拉刷新

微信小程序使用 wx.startPullDownRefresh（Object object）来进行刷新，调用后触发下拉刷新动画，效果与用户手动下拉刷新一致。它有 3 个回调函数：成功后回调函数、失败后回调函数、完成后回调函数，示例代码如下。

```
wx.startPullDownRefresh({
    success:function(res){
      //成功后回调函数
    },
    fail: function(res){
      //失败后回调函数
    },
    complete:{
      //完成后回调函数
    }
})
```

2. wx.stopPullDownRefresh 停止当前页面下拉刷新

微信小程序使用 wx.stopPullDownRefresh（Object object）来停止当前页面的下拉刷新。它有 3 个回调函数：成功后回调函数、失败后回调函数、完成后回调函数，示例代码如下。

```
Page({
  onPullDownRefresh() {
    wx.stopPullDownRefresh({
      success: function(res){
        //成功后回调函数
      },
      fail: function(res){
        //失败后回调函数
      },
      complete: function(res) {
        //完成后回调函数
      }
    })
  }
})
```

下面介绍下拉刷新的完整使用方法。

（1）在页面 demo.json 文件里配置下拉刷新属性，具体代码如下。

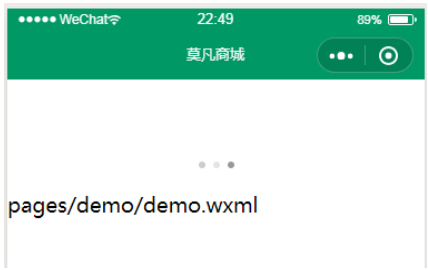
```
{
  "usingComponents": {},
  "backgroundTextStyle":"dark", //dark:显示刷新动画
  "enablePullDownRefresh":true, //允许下拉刷新
   "onReachBottomDistance":50//距离底部多少像素时触发上拉加载事件
}
```

（2）在页面 demo.js 文件里配置下拉刷新属性，具体代码如下。

```
Page({
  onLoad: function (options) {
    wx.startPullDownRefresh();//开始下拉刷新
  },

  /**
   * 页面相关事件处理函数--监听用户下拉动作
   */
  onPullDownRefresh: function () {
    wx.stopPullDownRefresh()//得到结果后关掉刷新动画
  },

  /**
   * 页面上拉触底事件的处理函数
   */
```

```
  onReachBottom: function () {
    //触发上拉相关操作
  },

})
```

效果如图所示。



下拉刷新效果

5.6.2 wx.setBackgroundColor 动态设置窗口的背景色

微信小程序提供用 wx.setBackgroundColor（Object object）函数来动态设置窗口背景色的功能，可以整体设置背景色颜色、设置顶部背景色颜色或设置底部背景色颜色，Object 参数说明如表所示。

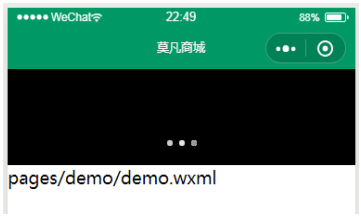
wx.setBackgroundColor 的参数说明

属性	类型	是否必填	说明
backgroundColor	string	否	窗口的背景色，必须为十六进制颜色值
backgroundColorTop	string	否	顶部窗口的背景色，必须为十六进制颜色值，仅 iOS 支持
backgroundColorBottom	string	否	底部窗口的背景色，必须为十六进制颜色值，仅 iOS 支持
success	Function	否	接口调用成功的回调函数
fail	Function	否	接口调用失败的回调函数
complete	Function	否	接口调用结束的回调函数（调用成功、失败都会执行）

示例代码如下。

```
Page({
  onLoad: function (options) {
    wx.setBackgroundColor({
      backgroundColor: '#000000', // 窗口的背景色为深黑色
    })

    wx.setBackgroundColor({
      //backgroundColorTop: '#999999', // 顶部窗口的背景色为浅黑色
      //backgroundColorBottom: '#cccccc', // 底部窗口的背景色为灰色
    })

    wx.startPullDownRefresh();//开始下拉刷新
  },

  /**
   * 页面相关事件处理函数--监听用户下拉动作
   */
  onPullDownRefresh: function () {
    wx.stopPullDownRefresh()//得到结果后关掉刷新动画
  },

  /**
   * 页面上拉触底事件的处理函数
   */
  onReachBottom: function () {
    //触发上拉相关操作
  },

})
```

效果如图所示。



设置窗口背景色

5.6.3 wx.setBackgroundTextStyle 动态设置下拉背景字体

微信小程序提供用 wx.setBackgroundTextStyle（Object object）函数来动态设置下拉背景字体的功能，可以动态设置下拉背景字体、loading 图的样式，Object 参数说明如表所示。

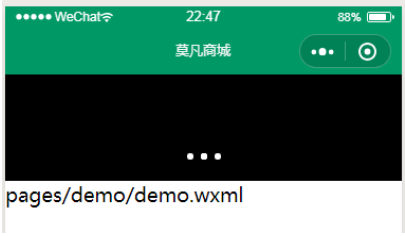
wx.setBackgroundTextStyle 的参数说明

属性	类型	是否必填	说明
textStyle	string	是	下拉背景字体、loading 图的样式，textStyle 可选值：dark/light（黑色/白色）
success	Function	否	接口调用成功的回调函数
fail	Function	否	接口调用失败的回调函数
complete	Function	否	接口调用结束的回调函数（调用成功、失败都会执行）

示例代码如下。

```
Page({
  onLoad: function (options) {
    wx.setBackgroundColor({
      backgroundColor: '#000000', // 窗口的背景色为深黑色
    })

    wx.setBackgroundColor({
      //backgroundColorTop: '#999999', // 顶部窗口的背景色为浅黑色
      //backgroundColorBottom: '#cccccc', // 底部窗口的背景色为灰色
    })

    wx.setBackgroundTextStyle({
     textStyle: 'light' // 下拉背景字体、loading 图的样式为 light(白色)
    })

    wx.startPullDownRefresh();//开始下拉刷新
  },

  /**
   * 页面相关事件处理函数——监听用户下拉动作
   */
  onPullDownRefresh: function () {
    wx.stopPullDownRefresh()//得到结果后关掉刷新动画
  },

  /**
   * 页面上拉触底事件的处理函数
   */
  onReachBottom: function () {
    //触发上拉相关操作
  },

})
```

效果如图所示。



设置下拉背景字体

5.6.4 wx.loadFontFace 引入第三方字体

微信小程序提供引入第三方字体的 API 接口 wx.loadFontFace，可以动态加载网络字体。文件地址须为下载类型。iOS 仅支持 https 格式的文件地址。wx.loadFontFace（Object objec）的参数说明如表所示。

wx. loadFontFace 的参数说明

属性	类型	是否必填	说明
family	string	是	定义的字体名称
source	string	是	字体资源的地址。建议格式为 TTF 和 WOFF，WOFF2 在低版本的 iOS 上会不兼容
desc	Object	否	可选的字体描述符
success	Function	否	接口调用成功的回调函数
fail	Function	否	接口调用失败的回调函数
complete	Function	否	接口调用结束的回调函数（调用成功、失败都会执行）

（1）可以在页面 JS 文件里引入第三方字体，示例代码如下。

```
Page({
  onLoad: function (options) {
    wx.loadFontFace({
      family: 'Bitstream Vera Serif Bold',
      source: 'url("https://sungd.github.io/Pacifico.ttf")',
      success: function(res){
        console.log(res); // {status: "loaded", cbID: 1}
      }
    })
  }
})
```

（2）在 WXSS 页面文件里，使用引入的字体，示例代码如下。

```
font-family: ' Bitstream Vera Serif Bold ';
```

（3）在引入第三方字体时，也可以新建一个字体文件 font.js 文件，以方便各个页面直接引用，示例代码如下。

```
//font.js
function loadFont(){
    wx.loadFontFace({
      family: 'Bitstream Vera Serif Bold',
      source: 'url("https://sungd.github.io/Pacifico.ttf")',
      success: function(res){
        console.log(res);
      }
    })
}
```

（4）在使用字体页面时，可以直接引入字体文件，示例代码如下。

```
const font = require('font.js')
Page({
  onLoad: function (options) {
    font.loadFont(); //加载字体
  }
})
```

（5）在 WXSS 页面文件里，使用引入的字体，示例代码如下。

```
font-family: ' Bitstream Vera Serif Bold ';
```

注意：字体文件返回的内容类型 content-type 参考 font，格式不正确时会解析失败。字体链接的格式必须是 https（iOS 不支持 http）。字体链接必须是同源下的或开启了 cors 支持的，小程序的域名是 servicewechat.com。canvas 等原生组件不支持使用接口添加的字体。工具里提示的 Faild to load font 可以忽略。

5.6.5 wx.pageScrollTo 将页面滚动到目标位置

微信小程序 wx.pageScrollTo 接口 API 可以将页面滚动到目标位置，提供两种滚动方式：通过选择器的方式滚动和通过指定距离的方式滚动。通过这个 API 接口就可以实现长页面的回到顶部、回到底部功能，wx.pageScrollTo（Object objec）的参数说明如表所示。

wx. pageScrollTo 的参数说明

属性	类型	是否必填	说明
scrollTop	number	否	滚动到页面的目标位置，单位为像素
duration	number	否	滚动动画的时长，单位为毫秒
selector	string	否	选择器
success	Function	否	接口调用成功的回调函数
fail	Function	否	接口调用失败的回调函数
complete	Function	否	接口调用结束的回调函数（调用成功、失败都会执行）

selector 选择器类似于 CSS 的选择器，但仅支持下列语法。

（1）ID 选择器：#the-id。

（2）class 选择器（可以连续指定多个）：.a-class.another-class。

（3）子元素选择器：.the-parent > .the-child。

（4）后代选择器：.the-ancestor .the-descendant。

（5）跨自定义组件的后代选择器：.the-ancestor >>> .the-descendant。

（6）多选择器的并集：#a-node, .some-other-nodes。

示例代码如下。

```
wx.pageScrollTo({
  scrollTop: 0,
  duration: 300
})
```

小结	本节课讲解了微信小程序模板的定义和使用；微信小程序的两种引用功能：import 引用和 include 引用；WXS 小程序脚本语言的使用；微信小程序提供的下拉刷新和监听事件的功能，使用 wx.setBackgroundColor 可以动态设置窗口的背景色，使用 wx.setBackgroundTextStyle 可以设置下拉背景文字，使用 wx.loadFontFace 可以引入第三方字体，使用 wx.pageScrollTo 可以将页面滚动到目标位置。
练习	