

## 疯狂早茶微信小程序基础篇《四》：数据绑定（上）

在之前的教程中写到，微信小程序框架将程序分为逻辑层(.js 文件)和视图层(.wxml 文件)。这是一种常见的 UI 和逻辑分离的程序设计方式，开发出来的程序更加灵活，易扩展。

这种程序设计方式通常要解决两个问题:

UI 层响应逻辑层逻辑和数据的变化

UI 层将用户的操作反馈到逻辑层

通常来说可以让 UI 层和逻辑层互相暴露接口给对方，不过出于对灵活性和扩展性的考虑，会引入中间层来进行管理，这样可以避免 UI 层和逻辑层之间的直接依赖。

微信小程序框架正是基于这种模式进行设计的，.wxml 文件来描述 UI 层(微信官方名称为**视图层**，教程后续也将用**视图层**来命名)，.js 文件来处理逻辑层，而微信的框架则作为中间层管理两者之间的调用。

为了更好的帮助开发者开发微信小程序，微信定义了一些语法规则来帮助开发者连接视图层和逻辑层。

### 数据绑定

#### 显示字符串内容

```
//page.wxml
```

```
<text class="user-motto">{{motto}}</text>
```

```
//page.js
```

```
data: {  
    motto: 'Hello World',  
},
```

通过将"{{motto}}"嵌入视图层代码中，界面会显示"Hello World"

### 更改字符串内容

```
setData({  
    motto:'Hello My World'  
})
```

当执行上述代码后，界面会显示"Hello My World"

上述部分有两点需要说明:

1: 视图层嵌入的是 {{motto}} 而不是 {{data.motto}}，微信框架默认设定视图层绑定的变量定义在 Page

对象的 **data 属性**中，也就是说，如果变量需要绑定到视图层，一定要定义 **data 属性**中

2: 通过调用 Page 对象的 setData(Page 对象预定义)方法，可以更新界面数据，但是**直接设置变量是无**

**效的**，因此对于绑定到视图层的变量，要永远使用 setData 方法来设置变量值

## 显示图片

在开发工具默认生成的程序首页，显示了用户的头像，相关代码如下：

```
//index.wxml
```

```
<image class="userinfo-avatar" src="{{userInfo.avatarUrl}}" background-  
size="cover"> </image>
```

```
//index.js
```

```
onLoad: function () {  
    console.log('onLoad')  
  
    var that = this  
  
    //调用应用实例的方法获取全局数据  
    app.getUserInfo(function(userInfo){  
        //更新数据  
        that.setData({  
            userInfo:userInfo  
        })  
    })  
}
```

```
    })  
}
```

image 标签的 src 属性绑定了 userInfo.avatarUrl 变量，并在 onLoad 方法中通过调用 setData 方法设置 userInfo。

## 属性绑定

可以将变量绑定到视图组件的属性值上(如上面 image 标签的 src 属性)，**注意在绑定到属性时，需要在外边加入一对双引号。**

```
src="{{userInfo.avatarUrl}}"
```

除了用于显示图片外，属性绑定还有许多作用。

## 关联数据

假设要做一个学生管理程序，页面使用列表来展示用户数据，当用户点击某一学生信息时，进入该学生的详情页页面。

当获取到用户的点击事件时，必须要知道点击的是哪一个学生的数据，这时可以将学生的 id 绑定到视图组件的 id 属性中来进行数据关联。

## 控制视图的隐藏/显示

可以将变量绑定到视图组件的 `hidden` 属性中，通过更改组件 `hidden` 属性的值，可以控制组件是否显示。

## 控制属性

上面说到可以通过属性绑定的方法来控制视图组件是否显示，还可以通过控制属性来实现这一功能。

```
//Page.wxml
```

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

```
//Page.js
```

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

通过绑定数据到 `wx:if` 属性，可以控制该组件是否显示。

框架还提供了 `wx:elif` 和 `wx:else` 属性，用法如下：

```
<view wx:if="{{length > 5}}"> 1 </view>
```

```
<view wx:elif="{{length > 2}}"> 2 </view>
```

```
<view wx:else> 3 </view>
```

如果想同时控制多个视图组件，可以使用 `block wx:if`

```
<block wx:if="{{true}}">
```

```
  <view> view1 </view>
```

```
  <view> view2 </view>
```

```
</block>
```

**同样，控制属性绑定时也需要添加双引号**

### **wx:if vs hidden**

一般来说，`wx:if` 有更高的切换消耗而 `hidden` 有更高的初始渲染消耗。因此，如果需要频繁切换的情景下，用 `hidden`

更好，如果在运行时条件不大可能改变则 `wx:if` 较好。

### **关键字**

框架提供了两个关键字来表示**真**和**假**

true : boolean 类型的 true , 代表真值。

false : boolean 类型的 false , 代表假值。

代码示例

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

关于数据绑定的更多讲述 , 敬请期待[微信小程序开发教程\(基础篇\)8-数据绑定下](#)