

第2章 小程序框架分析

第6节 列表渲染

谢涛



内容提要

- wx:for 列表渲染单个组件
- block wx:for列表渲染多个组件
- wx:key指定唯一标识符

1、wx:for 列表渲染单个组件

- 在组件上使用wx:for控制属性绑定一个数组，即可使用数组中各项的数据重复渲染该组件
- 数组当前项的下标变量名默认为index
- 数组当前项的变量名默认为item
- 示例代码：

- `<view wx:for= “{{ array }}” >`
- `{{ index }} : {{ item.message }}`
- `</view>`

- `Page({`
-

示例代码

```
<view wx:for= “{{ array }}” >  
  {{ index }} : {{ item.message }}  
</view>
```

```
Page({  
  data: {  
    array: [{  
      message: ‘第0个元素’,  
    }, {  
      message: ‘第1个元素’  
    }  
  }  
})
```



- 使用wx:for-item可以指定数组当前元素的变量名
- 使用wx:for-index可以指定数组当前下标的变量名
- 示例代码:

```
<view wx:for= “{{ array }}” wx:for-index= “idx”  
wx:for-item= “itemName” >  
    {{ idx }}: {{ itemName.message }}  
</view>
```



2、block wx:for列表渲染多个组件

- wx:for应用在某一个组件上，如果想渲染一个包含多节点的结构块，这时wx:for需要应用在<block/>标签上
- 示例代码：

```
<block wx:for= “{{ [1,2,3]}}” >  
  <view> {{ index }} : </view>  
  <view> {{ item }}    </view>  
</block>
```


3、wx:key指定唯一标识符

- 如果列表中项目的位置会动态改变或有新的项目添加到列表中，并且希望列表中的项目保持自己的特征和状态，如
 - ✓ <input/>中的输入内容
 - ✓ <switch/>的选中状态
- 需要使用wx:key来指定列表中项目的唯一标识符

➤ wx:key的值以以下两种形式提供

- ✓ **字符串**：代表在for循环的array中item的某个property，该property的值需要是列表中唯一的字符串或数字，且不能动态改变
- ✓ **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
 - 当数据改变触发渲染层重新渲染的时候，会校正带有key的组件
 - 框架会确保它们被重新排列，而不是重新创建，以确保组件保持自身的状态，并且提高列表渲染时的效率

➤ 示例代码:

```
<switch wx:for= “{{ objectArray }}” wx:key= “unique” > {{ item.id }}</switch>
Page([
  data: {
    objectArray: [
      { id:5, unique: ‘unique_5’ },
      { id:4, unique: ‘unique_4’ },
      { id:3, unique: ‘unique_3’ },
      { id:2, unique: ‘unique_2’ },
      { id:1, unique: ‘unique_1’ },
      { id:0, unique: ‘unique_0’ }
    ]
  }
])
```

注意: 如不提供wx:key, 会报一个warning, 如果明确知道该列表是静态的, 或者不必关注其顺序, 可以选择忽略。