

# 微信小程序学习《二》：事件详解

## 一、什么是事件？

### 1.一种用户的行为

用户长按某一张图片，点击某个按钮，这就是用户的行为，也是事件

### 2.一种通讯方式

为什么说事件也是一种通讯方式呢？因为用户点击按钮的时候，这是发生在 UI 层的，我们的 UI 要把一些信息发送给我们的逻辑代码，因此也是一种通讯方式

---

## 二、事件的类别

### 1.点击事件 tap

### 2.长按事件 longtap

### 3.触摸事件

( 1 ) touchstart 开始触摸

( 2 ) touchend 结束触摸

( 3 ) touchmove 移动触摸

( 4 ) touchcancel 取消触摸

这里就有问题了，结束触摸和取消触摸有什么区别吗？

结束触摸可以理解为是主动的停止的触摸事件，比如我们正在移动一张图片，移动完成后就是结束触摸了，然后可能在移动的过程中，突然有事件来打断了触摸事件，比如电话来了，打断了你的触摸事件，这时候就是取消触摸了

#### 4.其他的触摸事件 submit

每个控件都有自己的事件

---

### 三、事件冒泡

了解什么是冒泡事件？什么是非冒泡事件？我们可以通过一个例子来理解什么是冒泡事件。

1.首先我们先修改下启动界面，启动界面创建三个 view 组件，分别给他们对应的样式和绑定他们的点击事件，且布局是一层嵌套一层的



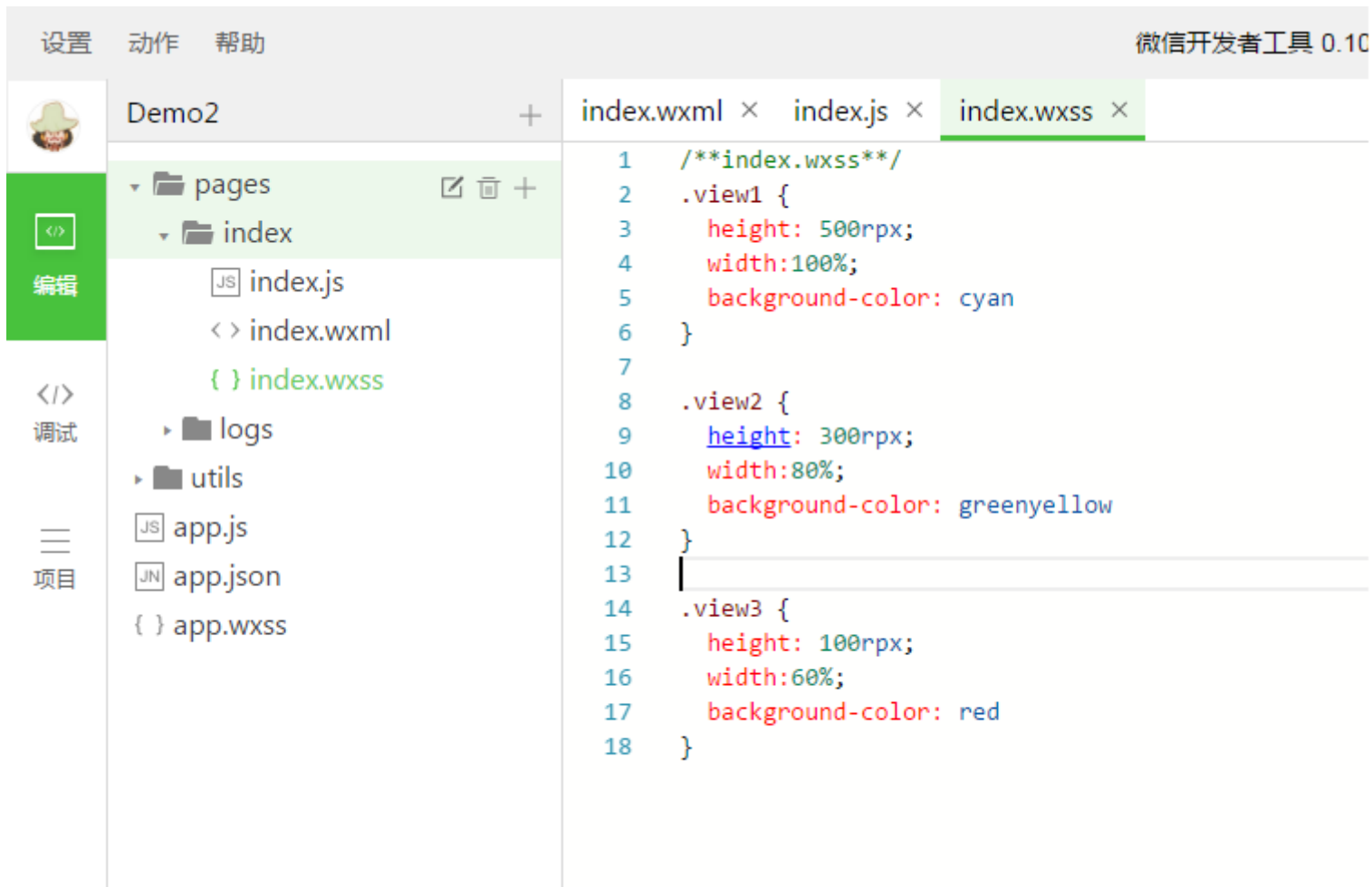
The screenshot shows the HBuilderX IDE interface. On the left is a project explorer for 'Demo2' with a file tree containing 'pages/index' (selected), 'index.js', 'index.wxml', 'index.wxss', 'logs', 'utils', 'app.js', 'app.json', and 'app.wxss'. The main editor displays the 'index.wxml' file with the following code:

```
1 <!--index.wxml-->
2 <view class="view1" bindtap="clickView1">
3   界面1
4   <view class="view2" bindtap="clickView2">
5     界面2
6     <view class="view3" bindtap="clickView3">
7       界面3
8     </view>
9   </view>
10 </view>
11
```

Red arrows point from the text annotation below to the 'view2' and 'view3' elements in the code, illustrating the nested structure.

1.创建三个view组件，分别给他们对应的样式和绑定他们的点击事件，且布局是一层嵌套一层的

2.然后在 index.wxss 文件中写下他们对应的样式



3.在 index.js 文件中写对应的点击事件，在控制台输出对应的 log



Demo2



index.wxml ×

index.js ×

index.wxss ×



编辑



调试



项目



编译



关闭

pages

index

index.js

index.wxml

index.wxss

logs

utils

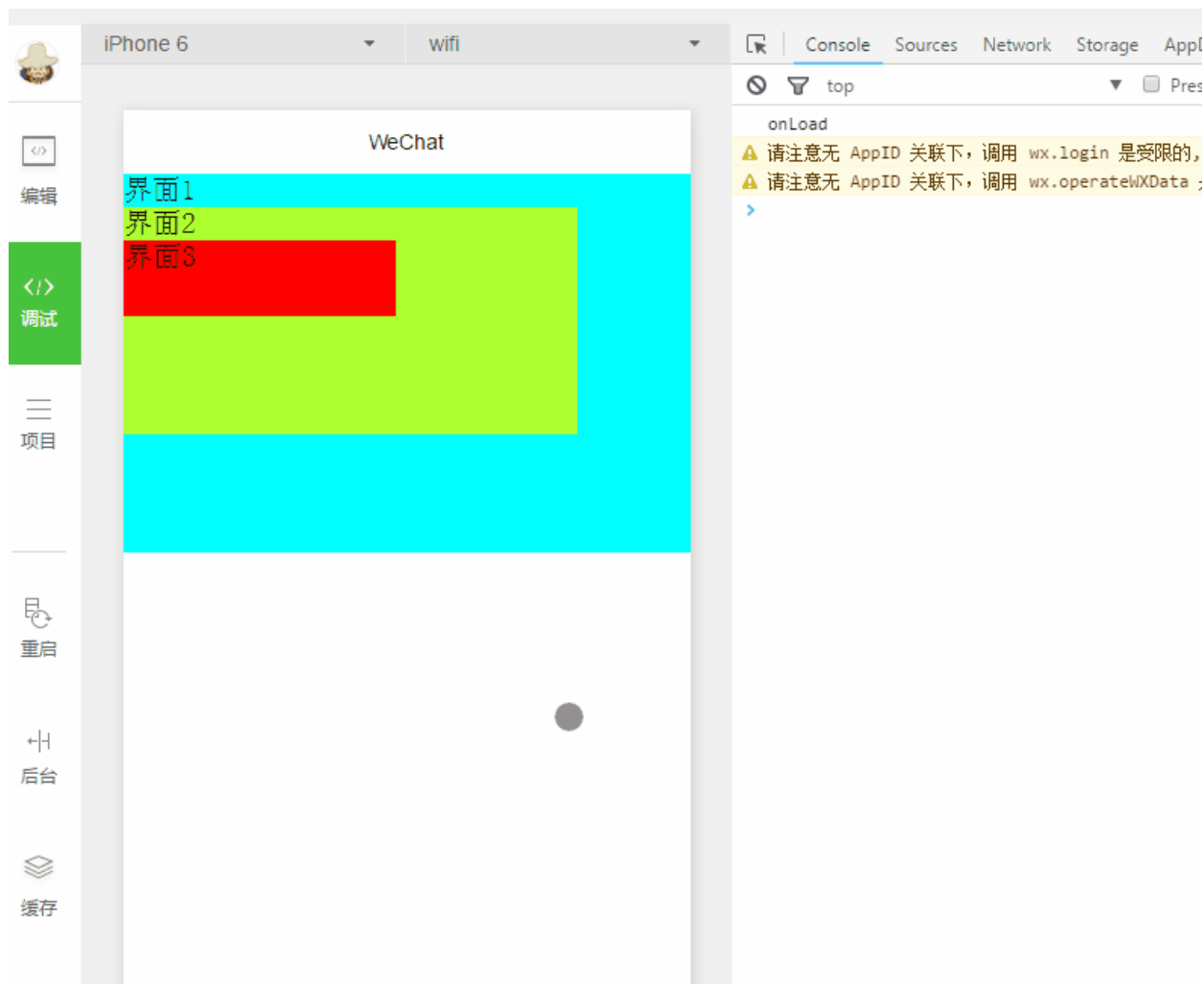
app.js

app.json

app.wxss

```
1  //index.js
2  //获取应用实例
3  var app = getApp()
4  Page( {
5    data: {
6      motto: 'Hello World',
7      userInfo: {}
8    },
9    //事件处理函数
10   clickView1: function() {
11     console.log( "clickView1" )
12   },
13   clickView2: function() {
14     console.log( "clickView2" )
15   },
16   clickView3: function() {
17     console.log( "clickView3" )
18   },
19
20
21   onLoad: function() {
22     console.log( 'onLoad' )
23     var that = this
24     //调用应用实例的方法获取全局数据
25     app.getUserInfo( function( userInfo ) {
26       //更新数据
27       that.setData( {
28         userInfo: userInfo
29       })
30     })
31   }
32 })
33
```

4.最后我们看下效果,仔细观察可以看到,我们点击界面3的时候,控制台输出了3行,分别是 clickView1 ,clickView2 ,clickView3 的点击事件,点击界面2的时候,控制台输出了 clickView2 和 clickView1,这就说明了点击子 view 的时候,也会响应父 view 的事件,这就是冒泡事件了,相反的,点击子 view,父 view 不会响应点击事件,这就是非冒泡事件了。



那么冒泡事件有哪些呢？

上面所提到的点击事件，长按事件，触摸事件都是冒泡事件，其余的都是非冒泡事件

---

## 四、事件的绑定

### 1.bind

上面的冒泡事件的例子就是使用 bind 的方式来绑定事件的

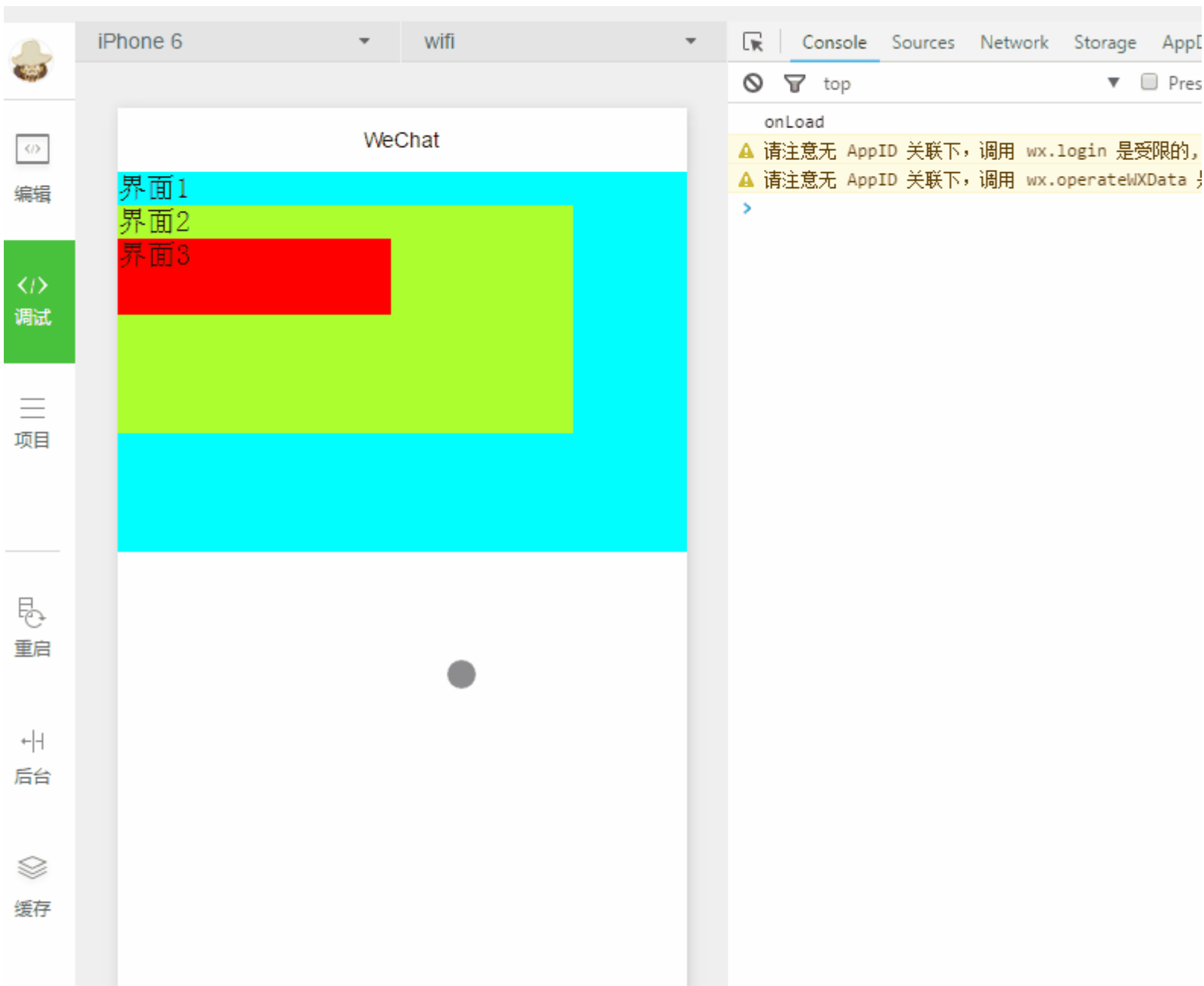
### 2.catch

那么 catch 绑定事件和 bind 的绑定事件有什么不同呢？我们通过修改上面的例子来说明，我们把 view3 的 bindtap 改为 catchtap



然后我们看下运行的效果：





可以看到，我们使用 catch 的绑定事件后，点击界面 3，是没有触发到父 view 的点击事件的

总结：使用 bind 的绑定事件，是可以触发冒泡事件的，就是可以触发父 view 的事件

使用 catch 是不会触发冒泡事件