

1.事件的使用方式

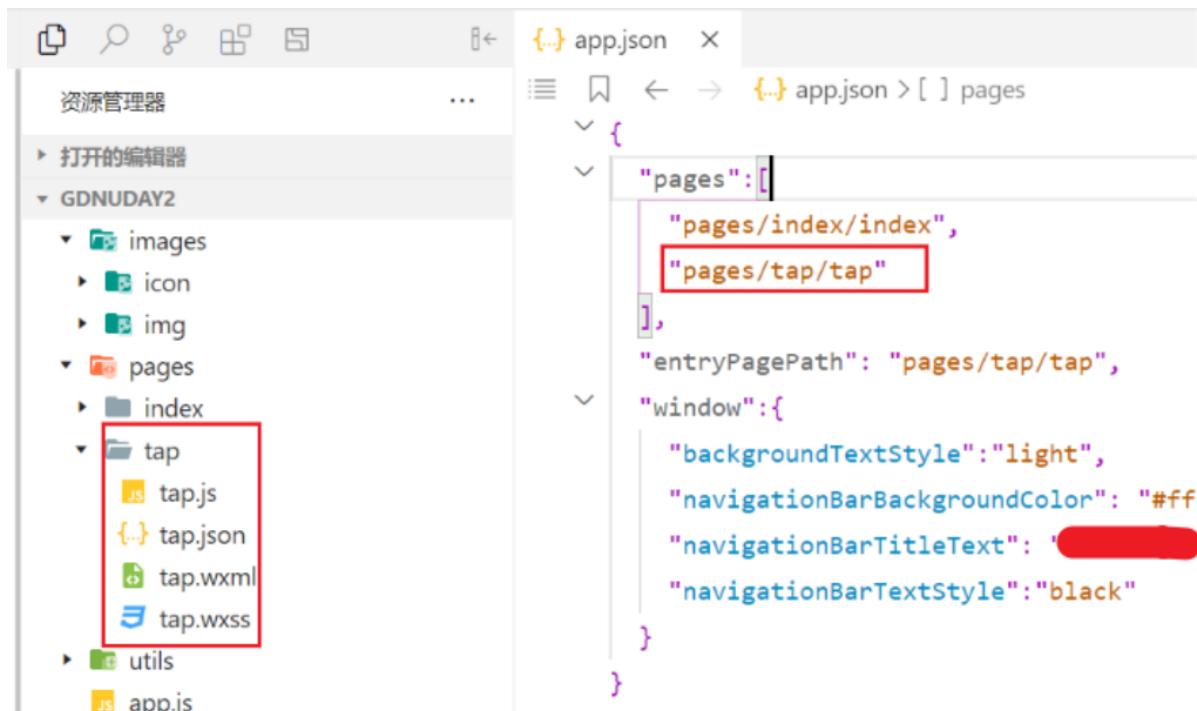
开发文档: <https://developers.weixin.qq.com/miniprogram/dev/framework/view/wxml/event.html>

直接在组件中绑定一个事件处理函数，如果是非自定义模板，则使用 `bindtap` 或 `bind:tap`，如果是自定义的，则使用 `bind:tap`。

注意：无论是自定义还是非自定义，都推荐使用 `bind:tap` 方式

示例：展示为标签绑定点击事件

1. 新创建一个小程序页面



2. 在 tap 页面中 tap.wxml 文件中构建如下标签代码，并且为该标签绑定一个点击事件

```
<button bind:tap="tapFun">点我呀</button>
```

3. 接着需要在页面的 tap.js 文件中创建出相应的处理函数，代码如下

```
Page({  
  /**  
   * 页面的初始数据  
   */  
  data: {
```

```

        imgStyle: "img_middle"
    },
onLoad() {
},
tapFun: function () {
    console.log("你没事点我干嘛?")
}
})

```

2.事件分类

小程序中事件分为冒泡和非冒泡

冒泡事件(`bind`): 当一个节点事件被触发后，该事件会向父节点传递。

非冒泡事件(`catch`): 当一个节点事件被触发后，该事件不会向父节点传递。

wxml 的公共事件类型:

类型	触发条件	最低版本
<code>touchstart</code>	手指触摸动作开始	
<code>touchmove</code>	手指触摸后移动	
<code>touchcancel</code>	手指触摸动作被打断，如来电提醒，弹窗	
<code>touchend</code>	手指触摸动作结束	
<code>tap</code>	手指触摸后马上离开点击	
<code>longpress</code>	手指触摸后，超过 350ms 再离开，如果指定了事件回调函数并触发了这个事件， <code>tap</code> 事件将不被触发	1.5.0
<code>longtap</code>	手指触摸后，超过 350ms 再离开(推荐使用 <code>longpress</code> 事件代替)	
<code>transitionend</code>	会在 WXSS transition 或 <code>wx.createAnimation</code> 动画结束后触发	
<code>animationstart</code>	会在一个 WXSS animation 动画开始时触发	
<code>animationiteration</code>	会在一个 WXSS animation 一次迭代结束时触发	
<code>animationend</code>	会在一个 WXSS animation 动画完成时触发	
<code>touchforcechange</code>	在支持 3D Touch 的 iPhone 设备，重按时会触发	1.9.90

案例：测试冒泡与非冒泡事件

1.tap.wxml 中绑定事件

```

<!-- 冒泡事件 -->
<view bind:tap="parent_tap">
    <button bind:tap="son_tap">冒泡事件</button>
</view>

<!-- 非冒泡事件 -->
<view bind:tap="parent_tap">

```

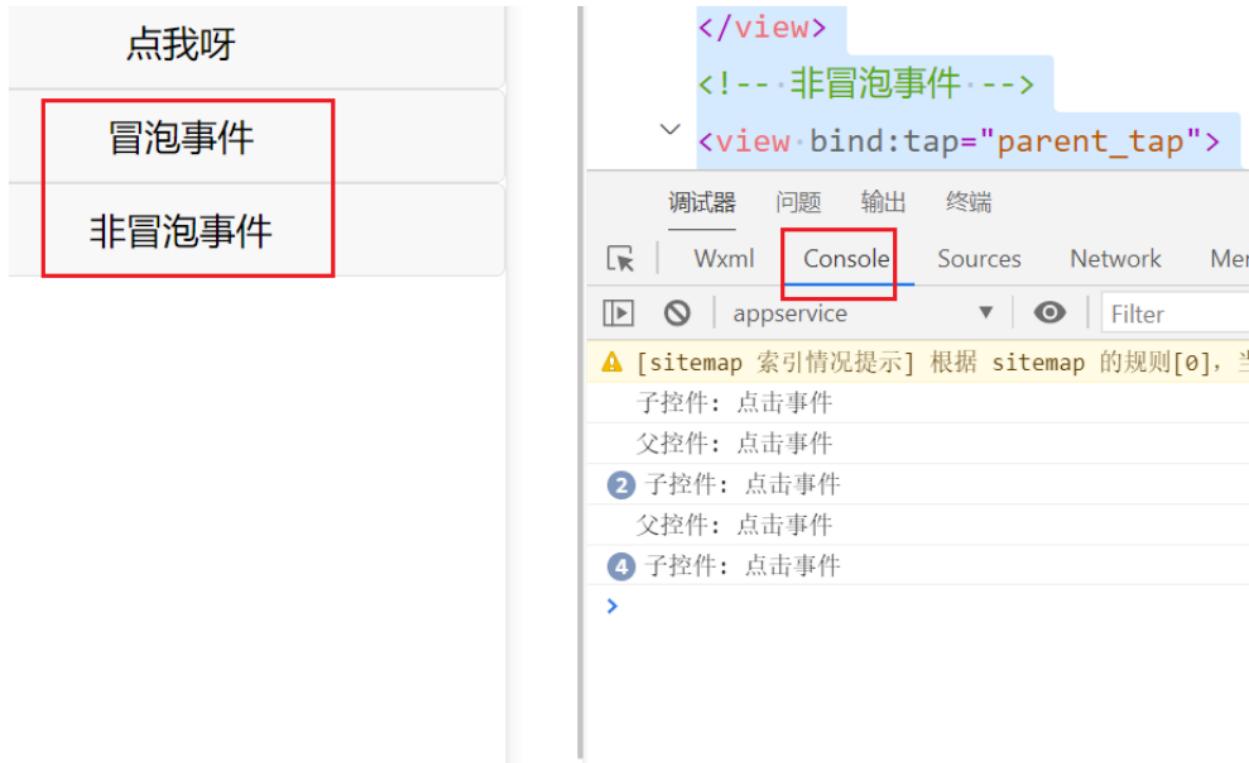
```
<button catch:tap="son_tap">非冒泡事件</button>
</view>
```

2.在 tap.js 中创建两相应的绑定函数

```
// pages/tap/tap.js
Page({
  /**
   * 页面的初始数据
   */
  // 父节点绑定的函数
  parent_tap: function () {
    console.log("父控件：点击事件")
  },

  // 子节点绑定的函数
  son_tap: function () {
    console.log("子控件：点击事件")
  }
})
```

效果



3.事件绑定

绑定的方式：

```
key:type="value"
```

例如：点击事件

- 参数说明：

- key 相当于 bind|catch
- type 相当于 tap、longpress
- value 相当于 tapFun

```
# 1.key的值为 bind 或 catch  
# 2.type即事件类型  
# 3.value是一个字符串，绑定时 value 是什么字符串，则相应的必须要在当前页面的 js 文件中定义同名的函数
```

案例：实现长按图片变大

1 tap.wxml添加image标签，且绑定长按事件

```
<!-- 长按事件：长按图片放大 -->  
<image class="{{imgStyle}}}" src="/images/icon/cart-o.png" bind:longpress="scaleImg"></image>
```

2 tap.wxss 设置样式

```
.img_middle {  
    width: 50rpx;  
    height: 50rpx;  
}  
  
.img_big {  
    width: 100rpx;  
    height: 100rpx;  
}
```

3 tap.js 数据绑定及添加长按事件

```
Page({  
    /**  
     * 页面的初始数据  
     */  
    data: {  
        imgStyle:"img_middle"  
    },  
  
    scaleImg:function(){  
        this.setData({  
            imgStyle:"img_big"  
        })  
    }  
})
```

4.事件对象

如无特殊说明，当节点触发事件时，绑定该事件的处理函数会收到一个事件对象。

BaseEvent 基础事件对象属性列表：

属性	类型	说明
type	String	事件类型
timeStamp	Integer	事件生成时的时间戳
target	Object	触发事件的组件的一些属性值集合
currentTarget	Object	当前组件的一些属性值集合在组件中 <code>data-</code> 定义的属性

注意：事件对象最大的作用在于给触发执行的函数传递参数。而参数需要在标签中通过“`data-参数名=参数值`”的方式来指定。

在 `tap.js` 事件中取事件对象的值

```
scaleImg:function(event){
    // 事件类型
    console.log(event.type);
    // 事件的属性值
    console.log(event.target)
}
```

