

疯狂早茶微信小程序基础篇《六》：事件

前面说到，微信小程序框架是逻辑层与 UI 层分析的设计方式，这种设计方式需要解决两个问题

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

其中前面讲到的**数据绑定**解决了第一个问题，而**事件**则解决第二个问题

什么是事件

事件是视图层到逻辑层的通讯方式。

事件可以将用户的行为反馈到逻辑层进行处理。

事件可以绑定在组件上，当达到触发事件，就会执行逻辑层中对应的事件处理函数。

事件对象可以携带额外信息，如 id, dataset, touches。

总结下来事件就是指发生了一些事情，通常是用户进行了一些操作，如点击某个按钮或在手机屏幕上滑动了手指。当事件发生时，框架会调用事件处理函数(如果有的话)，这样就可以实现对用户操作的响应。

事件绑定

通过事件绑定来完成对用户操作的响应，比如要处理 view 标签的 **tap** 事件，在标签属性中添加 bindtap

= 'tapName'，然后在.js 中添加 tapName 函数

```
//wxml
```

```
<view id="tapTest" data-hi="WeChat" bindtap="tapName"> Click me! </view>
```

```
//.js
```

```
Page({  
  tapName: function(event) {  
    console.log(event)  
  }  
})
```

event 对象包含一些关于事件的数据：

target：触发事件的组件

currentTarget：当前组件

type：事件类型

timeStamp：时间戳(页面打开到触发事件所经过的毫秒数)

touches：包含触摸点的数组(多点触控)

changedTouches ：发生改变的触摸点的数组(多点触控)

detail ：额外的自定义信息

冒泡事件和非冒泡事件

为什么会有 **target** 和 **currentTarget** 之分呢，这是由于事件分为两类，**冒泡事件**和**非冒泡事件**

冒泡事件：当一个组件上的事件被触发后，该事件会向父节点传递。

非冒泡事件：当一个组件上的事件被触发后，该事件不会向父节点传递。

其中 **tap** 事件是属于冒泡事件(这也是为什么上面例子中的 **event** 会包含 **currentTarget**)，另外其它的

冒泡事件还包括

| 类型 | 触发条件 |

| ----- |-----|

| touchstart | 手指触摸动作开始 |

| touchmove | 手指触摸后移动 |

| touchcancel | 手指触摸动作被打断，如来电提醒，弹窗 |

| touchend | 手指触摸动作结束 |

| tap | 手指触摸后马上离开 |

| longtap | 手指触摸后，超过 350ms 再离开 |

为什么需要冒泡事件

有了冒泡事件，就可以更加方便的实现一些功能。

比如程序有一个视图，包含用户头像和姓名，当用户点击头像或姓名时，进入用户详情页面。如果没有冒泡事件，就需要处理头像和姓名的点击事件，而现在只需在外层包裹一个组件，并处理该组件的事件即可。

阻止事件冒泡

在有些情况下可能会希望阻止事件的冒泡行为，可以使用 **catch 事件绑定**，如 **catchtap**，就可以阻止事件的冒泡行为。

可以通过下面的代码示例来加深对冒泡事件的理解

```
//.wxml
```

```
<view id="outter" bindtap="handleTapOutter">
```

```
  我是父亲节点
```

```
  <view id="middle" catchtap="handleTapMiddle">
```

```
    我是儿子节点
```

```
    <view id="inner" bindtap="handleInner">
```

```
      我是孙子节点
```

```
</view>
```

```
</view>
```

```
</view>
```

```
//.js
```

```
Page({
```

```
  handleTapOutter: function(event) {
```

```
    console.log("父亲节点被点击")
```

```
  },
```

```
  handleTapMiddle: function(event) {
```

```
    console.log("儿子节点被点击")
```

```
  },
```

```
  handleInner: function(event) {
```

```
    console.log("孙子节点被点击")
```

```
  },
```

```
})
```

尝试修改各级节点的 tap 事件绑定方式，查看输出的日志会有什么变化。