

小程序的事件处理

王红元 coderwhy

目录

content



1 小程序的事件监听

2 常见事件类型划分

3 事件对象属性分析

4 事件参数传递方法

5 事件传递案例练习

6 冒泡和捕获的区别

事件的监听

■ 什么时候会产生事件呢？

- 小程序需要经常和用户进行某种交互，比如点击界面上的某个按钮或者区域，比如滑动了某个区域；
- 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，当触发事件时，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；

■ 事件时如何处理呢？

- - 事件是通过bind/catch这个属性绑定在组件上的（和普通的属性写法很相似，以key= “value” 形式）；
- - key以bind或catch开头，从1.5.0版本开始，可以在bind和catch后加上一个冒号；
- - 同时当前页面的Page构造器中定义对应的事件处理函数，如果没有对应的函数，触发事件时会报错；
- - 比如当用户点击该button区域时，达到触发条件生成事件tap，该事件处理函数会被执行，同时还会收到一个事件对象event。



组件的特殊事件

- 某些组件会有自己特性的事件类型，大家可以在使用组件时具体查看对应的文档
 - - 比如input有bindinput/bindblur/bindfocus等
 - - 比如scroll-view有bindscrolltoupper/bindscrolltolower等
- 这里我们讨论几个组件都有的, 并且也比较常见的事件类型:

类型	触发条件	最低版本
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	
longpress	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，tap事件将不被触发	1.5.0
longtap	手指触摸后，超过350ms再离开（推荐使用 longpress 事件代替）	



事件对象event

■ 当某个事件触发时, 会产生一个**事件对象**, 并且这个对象被传入到回调函数中, 事件对象有哪些常见的属性呢?

BaseEvent 基础事件对象属性列表:

属性	类型	说明	基础库版本
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	
mark	Object	事件标记数据	2.7.1

currentTarget和target的区别

The screenshot displays a mobile app development environment. On the left, a preview of a mobile app interface shows a red rectangle labeled 'outer' containing an orange rectangle labeled 'inner'. The top status bar shows the time 16:40 and 100% battery. The middle-left pane shows a file explorer with a project structure including folders like 'pages' and 'profile', and files like 'profile.js', 'profile.json', 'profile.wxml', and 'profile.wxss'. The middle-right pane shows the WXML code for 'profile.wxml':

```
1 <!--pages/profile/profile.wxml-->
2 <view class="outer" id="outer" bindtap="onOuterTap">
3   outer
4   <view class="inner" id="inner" bindtap="onInnerTap">inner</view>
5 </view>
6
```

The bottom-right pane shows the console output for a tap event on the 'outer' view:

```
outer: {type: "tap", timeStamp: 180751, target: {...}, currentTarget: {...}, mark: {...}, ...}
  changedTouches: [{...}]
  currentTarget: {id: "outer", offsetLeft: 0, offsetTop: 0, dataset: {...}}
  detail: {x: 66.078125, y: 44.28125}
  mark: {}
  mut: false
  target: {id: "inner", offsetLeft: 0, offsetTop: 0, dataset: {...}}
  timeStamp: 180751
  touches: [{...}]
  type: "tap"
  _userTap: true
  __proto__: Object
```

touches和changedTouches的区别

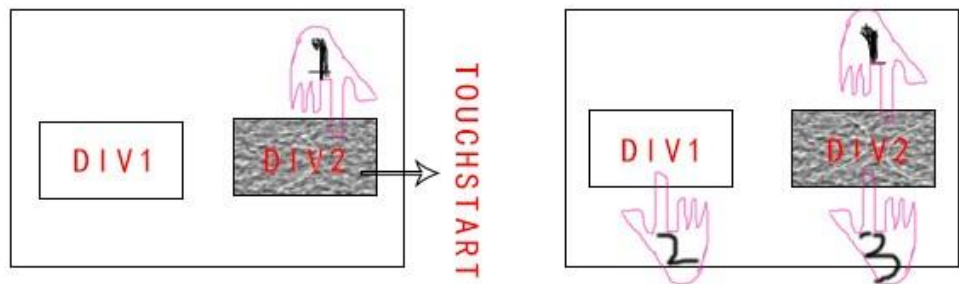
- 1.在touchend中不同
- 2.多手指触摸时不同

touchstart

```
▼ Object {type: "touchstart", timeStamp: 2352}
  ▼ changedTouches: Array[1]
    ▼ 0: Object
      clientX: 181
      clientY: 40
      identifier: 0
      pageX: 181
      pageY: 40
      ▶ __proto__: Object
      length: 1
      ▶ __proto__: Array[0]
    ▶ currentTarget: Object
    ▶ target: Object
    timeStamp: 2352
  ▼ touches: Array[1]
    ▼ 0: Object
      clientX: 181
      clientY: 40
      identifier: 0
      pageX: 181
      pageY: 40
      ▶ __proto__: Object
      length: 1
      ▶ __proto__: Array[0]
    type: "touchstart"
    ▶ __proto__: Object
```

touchend

```
▼ Object {type: "touchend", timeStamp: 509}
  ▼ changedTouches: Array[1]
    ▼ 0: Object
      clientX: 187
      clientY: 37
      identifier: 0
      pageX: 187
      pageY: 37
      ▶ __proto__: Object
      length: 1
      ▶ __proto__: Array[0]
    ▶ currentTarget: Object
    ▶ target: Object
    timeStamp: 5097
  ▼ touches: Array[0]
    length: 0
    ▶ __proto__: Array[0]
    type: "touchend"
    ▶ __proto__: Object
```



touches: ① touches: ①②③
 changedTouches: ① changedTouches: ②③

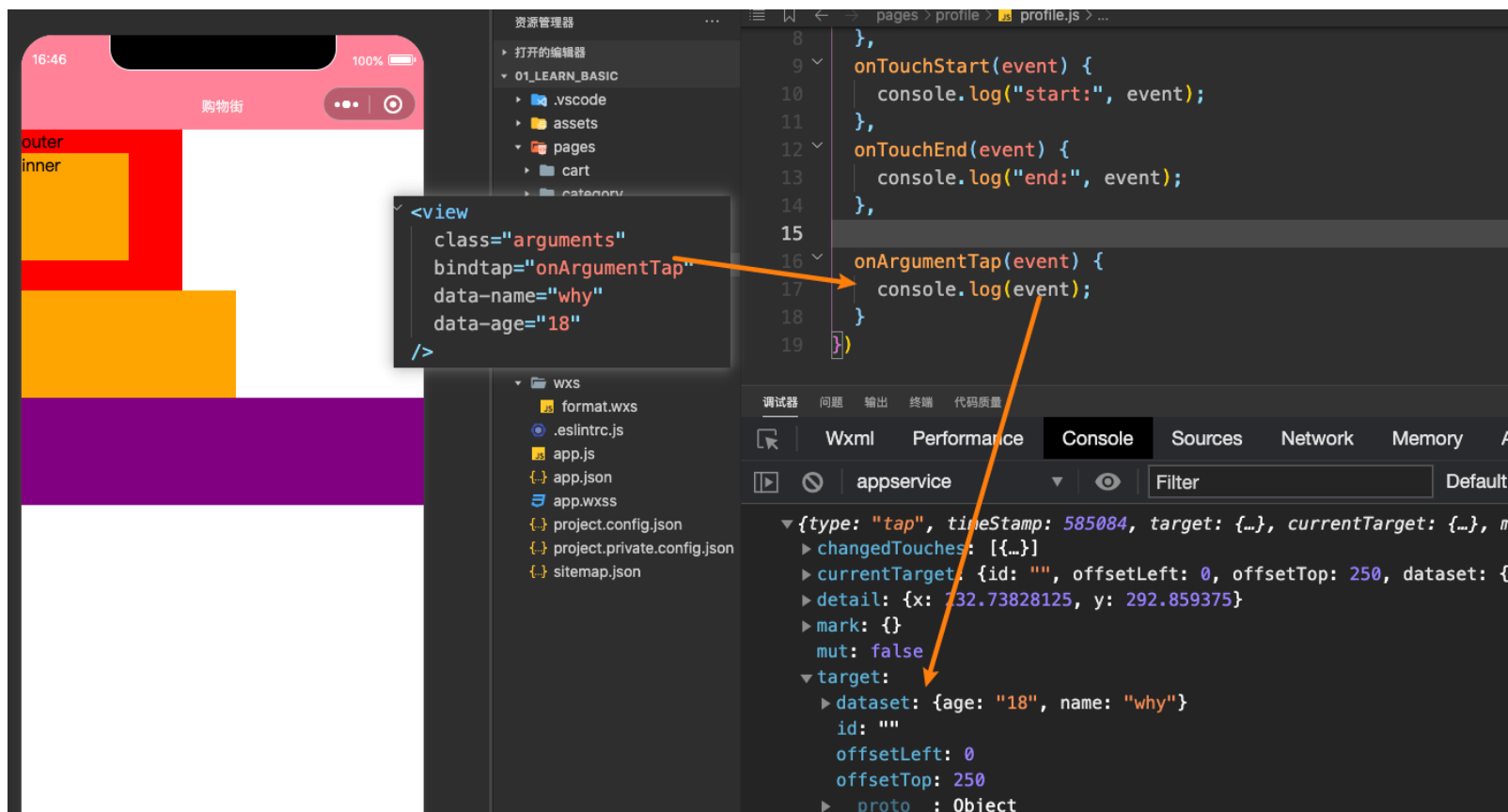
touches: 当前屏幕上所有触摸点的列表
 changedTouches: 触发事件时改变的触摸点的集合

事件参数的传递

■ 当视图层发生事件时，某些情况需要事件携带一些参数到执行的函数中，这个时候就可以通过data-属性来完成：

□ 格式：data-属性的名称

□ 获取：e.currentTarget.dataset.属性的名称



参数传递的案例练习

The image displays a WeChat Mini Program development environment. On the left, a mobile simulator shows a UI with a pink header labeled '购物街' and three tabs: '衣服', '鞋子', and '裤子' (the last one is active). The middle panel shows the file explorer with a project structure including '01_LEARN_BASIC', 'pages', 'utils', and 'wxs'. The right panel shows the code editor with two files open: 'profile.js' and 'profile.wxml'.

profile.js

```
23  
24  
25  
26  
27  
28  
onTabItemTap(event) {  
  const index = event.currentTarget.dataset.index  
  this.setData({ currentIndex: index })  
}
```

profile.wxml

```
<view class="tab-control">  
  <block wx:for="{{titles}}" wx:key="*this">  
    <view  
      class="item {{index === currentIndex ? 'active': ''}}"  
      data-item="{{item}}"  
      data-index="{{index}}"  
      bindtap="onTabItemTap"  
    >  
      <text class="title">{{item}}</text>  
    </view>  
  </block>  
</view>
```

事件冒泡和事件捕获

- 当界面产生一个事件时，事件分为了**捕获阶段**和**冒泡阶段**。

