

微信小程序 - Uni03

1.事件

事件是视图层(`page.wxml`)与逻辑层(`page.js`)的通讯方式。

事件可以绑定到组件上，当触发组件的事件后，事件所绑定的函数将被执行。

1.1 事件列表

	浏览器	移动端	微信小程序
键盘事件	<code>keydown</code> 、 <code>keyup</code> 、 <code>keypress</code>	无	无
鼠标事件	<code>click</code> 、 <code>mouseover</code> 、 <code>mouseout</code>	无	无
页面事件	<code>load</code> 、 <code>unload</code>	<code>load</code> 、 <code>unload</code>	生命周期的钩子函数
触屏事件	无	<code>touchstart</code> <code>touchmove</code> <code>touchend</code> <code>touchcancel</code>	<code>touchstart</code> <code>touchmove</code> <code>touchend</code> <code>touchcancel</code> <code>tap</code> <code>longtap</code> <code>longpress</code>

- `tap`，轻敲
- `longtap`，长敲（350ms）
- `longpress`，长按

`longtap` 事件会触发 `tap` 事件，`longpress` 事件不会触发 `tap` 事件

1.2 事件绑定

```
<component bind事件名称="函数名称"></component>
<component bind:事件名称="函数名称"></component>
<component catch事件名称="函数名称"></component>
```

事件绑定的函数名称中禁止使用小括号!!! 示例代码如下：

```
<button type="primary" bindtap="f1">轻敲按钮</button>
```

`catch`事件名称 会阻止事件冒泡，`bind`事件名称或`bind:事件名称` 不会阻止事件冒泡

1.3 事件对象 -- Event

微信小程序中绑定事件处理函数时，不能带有小括号 -- 引发了如何进行参数的传递？

方法1: 如果事件是组件自定义事件，如 `<input>` 组件的 `bindinput`、`bindfocus` 等，此时需要调用 `event` 对象的 `detail` 属性进行获取，示例代码如下：

```
input(event){
  //用于获取输入框的值
  console.log(event.detail.value);
  //用于获取输入信息的键盘位置
  console.log(event.detail.keyCode);
  //用于获取输入时当前光标的位置
  console.log(event.detail.cursor);
}
```

方法2: 如果事件为系统事件，如 `bindtap`、`bindlongtap`、`bindlongpress` 等，此时：

A.在组件上通过 `data-*` 创建自定义属性，示例代码如下：

```
<view wx:for="{{users}}" wx:key="index">
  <text>{{item}}</text>
  <button type="primary" bindtap="remove" data-id="{{index}}">删除</button>
</view>
```

如果自定义属性为使用短横线进行分隔的多个单词组件的话，将自动转换成驼峰标记法

如果属性存在大写字母，则自动转换成小写字母

B.在处理函数中，通过 `Event` 对象的 `target.dataset` 属性进行获取，示例代码如下：

```
remove(event){
  console.log(event.target.dataset.id);
}
```

2.双向数据绑定

逻辑层到视图层

```
{{...}}
```

视图层到逻辑层

A.在视图层为组件进行事件的绑定，并且在处理函数中通过 `event` 对象接收到传递的数据

B.在逻辑层通过 `setData()` 方法修改 `data` 中的数据，结构如下：

```
//存储当前页面的动态数据
data:{
  text: '123'
},
//函数(方法)名称
input(event){
  let text = event.detail.value;
  this.setData({
    text:text
  })
}
```

3.微信API

· `wx.showModal()` 方法

`wx.showModal()` 方法用于显示模态对话框，其语法结构是：

```
wx.showModal({
  title:"标题信息",
  content:"正文内容",
  showCancel:是否显示取消按钮,
  success:res=>{
    //res返回对象,包含confirm及cancel属性
    //在单击确认按钮时,confirm属性为true,cancel属性为false
    //在单击取消按钮时,confirm属性为false,cancel属性为true
  }
})
```

· `wx.showToast()` 方法

`wx.showToast()` 方法用于显示短消息提示框，其语法结构是：

```
wx.showToast({
  title:"标题信息",
  icon:"显示的图标(success|loading|none)",
  image:"图标URL地址",
  duration:持续时间(1500ms),
  mask:是否显示透明的遮罩层(true|false)
})
```

· `wx.hideToast()` 方法

· `wx.showLoading()` 方法

· `wx.hideLoading()` 方法

4.微信小程序的生命周期

· 微信小程序的生命周期 -- `app.js`

· `onLaunch` 钩子函数在小程序初始化完成时自动调用，只触发一次，其语法结构是：

```
App({
  onLaunch(){
    //....
  }
})
```

· `onShow` 钩子函数在小程序启动显示时或切换到前台时触发，其语法结构是

```
App({
  onShow(){
    //....
  }
})
```

· `onHide` 钩子函数在小程序切换到后台时触发，其语法结构是：

```
App({
  onHide(){
    //....
  }
})
```

· 页面的生命周期