# 事件-参数的传递

1. 绑定事件, 定义传递需要的参数
   1. <view wx:for="{{articles}}" wx:for-item="t" bind:tap="onArticleClick" data-id="{{t.id}}" data-title="{{t.title}}">
   2. {{t.title}}
   3. </view>

bind:tap 🡪 设置点击驱动的函数

data-xxx 🡪 将数据传递到onArticleClick(event) 的event当中

1. 将参数传递到新的一个page页面
   1. onArticleClick: function(event){
   2. var dataset = event.currentTarget.dataset;
   3. console.log(dataset);
   4. var id = dataset.id;
   5. console.log(id);
   6. wx.navigateTo({
   7. url: "/pages/test/test?id="+id,
   8. });
   9. },

将id通过url形式传递给新的page

1. 新页面接受参数
   1. onLoad(options) {
   2. console.log(options);
   3. },

参数再新页面的onLoad方法的options里面

# 事件-冒泡

1. 使用catch:xxx 替代 bind:xxx可以防止事件冒泡的出现
   1. <view class="out\_view" catch:tap="outViewClick">
   2. 我是大View
   3. <view class="inner\_view" catch:tap="innerViewClick">
   4. 我是小View
   5. </view>
   6. </view>

# hover 悬浮事件

1 .view上添加悬浮事件

hover-class -- 类名定义css样式使用

hover-start-time -- 设置开始时间

hover-stay-time -- 事件持续时间

hover-stop-propagation – 控件重叠时候不往下传播时间

1. <view class="outer" hover-class="outer\_hover" hover-start-time="0" hover-stay-time="500">
2. <view class="inner" hover-class="inner\_hover" hover-start-time="0" hover-stay-time="500"
3. hover-stop-propagation="{{true}}"></view>
4. </view>