组件类似于页面，一个自定义组件由 json，wxml，wxss，js 4个文件组成，与vue类似

**自定义组件**

**.WXML文件**

<!-- 这是自定义组件的内部WXML结构 -->

<view class="inner">

  {{innerText}}

</view>

<slot></slot>

**.js文件**

Component({

  properties: {

    // 这里定义了innerText属性，属性值可以在组件使用时指定

    innerText: {

      type: String,

      value: 'default value',

    }

  },

  data: {

    // 这里是一些组件内部数据

    someData: {}

  },

  methods: {

    // 这里是一个自定义方法

    customMethod: function(){}

  }

})

**.json文件**

{

  "component": true

}

**使用自定义组件**

**在页面的.json文件中声明**

{

"usingComponents": {

"component-tag-name": "path/to/the/custom/component"

}}

**在.WXML中使用**

<view>

<!-- 以下是对一个自定义组件的引用 -->

<component-tag-name inner-text="Some text"></component-tag-name>

</view>

**组件生命周期**

Component({

  // 组件的生命周期定义在lifetimes中

  lifetimes: {

    attached: function() {

      // 在组件实例进入页面节点树时执行

    },

    detached: function() {

      // 在组件实例被从页面节点树移除时执行

    },

  },

  // ...

})

| **生命周期** | **参数** | **描述** | **最低版本** |
| --- | --- | --- | --- |
| created | 无 | 在组件实例刚刚被创建时执行 | [1.6.3](https://developers.weixin.qq.com/miniprogram/dev/framework/compatibility.html) |
| attached | 无 | 在组件实例进入页面节点树时执行 | [1.6.3](https://developers.weixin.qq.com/miniprogram/dev/framework/compatibility.html) |
| ready | 无 | 在组件在视图层布局完成后执行 | [1.6.3](https://developers.weixin.qq.com/miniprogram/dev/framework/compatibility.html) |
| moved | 无 | 在组件实例被移动到节点树另一个位置时执行 | [1.6.3](https://developers.weixin.qq.com/miniprogram/dev/framework/compatibility.html) |
| detached | 无 | 在组件实例被从页面节点树移除时执行 | [1.6.3](https://developers.weixin.qq.com/miniprogram/dev/framework/compatibility.html) |
| error | Object Error | 每当组件方法抛出错误时执行 | [2.4.1](https://developers.weixin.qq.com/miniprogram/dev/framework/compatibility.html) |

**组件的通信**

**页面绑定事件函数**

<!-- 或者可以写成 -->

<component-tag-name bind:myevent="onMyEvent" />

**组件触发事件**

Component({

  properties: {},

  methods: {

    onTap: function(){

      var myEventDetail = {} // 事件函数所需参数

      var myEventOption = {} // 事件函数所需参数

      // 触发事件函数

      this.triggerEvent('myevent', myEventDetail, myEventOption)

    }

  }

})

**组件 wxml 的 slot**

**启用多slot**

默认情况下只能有一个slot，想启用多个slot可如下配置

Component({

  options: {

    multipleSlots: true // 在组件定义时的选项中启用多slot支持

  },

  properties: { /\* ... \*/ },

  methods: { /\* ... \*/ }

})

**wxml定义slot**

<view class="wrapper">

  <slot name="before"></slot>

  <view>这里是组件的内部细节</view>

  <slot name="after"></slot>

</view>

**组件使用者传入slot**

<!-- 引用组件的页面模板 -->

<view>

  <component-tag-name>

    <!-- 这部分内容将被放置在组件 <slot name="before"> 的位置上 -->

    <view slot="before">这里是插入到组件slot name="before"中的内容</view>

    <!-- 这部分内容将被放置在组件 <slot name="after"> 的位置上 -->

    <view slot="after">这里是插入到组件slot name="after"中的内容</view>

  </component-tag-name>

</view>