

标签

小程序视图层开发基于小程序框架为开发者提供的一系列[基础组件](#)，这些基础组件通常以双标签或单标签的形式使用。

- 双标签包括起始标签<标签>，终止标签</标签>和属性，内容在这两个标签之内
- 单标签只有一个<标签/>，有属性，没有内容

小程序规定，标签名有多个词时，词之间以连接符-连接。

```
<tag-name property="value">
  内容放这里...
</tag-name>

<tag-name property="value"/>
```

编码时要遵循标签的语义，要尽量使用最少的标签并保持最小的复杂度。

代码大小写

所有标签和属性，大部分属性值统一使用小写

- 推荐写法

```
<view class="demo">
  ...
</view>
```

- 不推荐写法

```
<view class="DEMO">
  ...
</view>

<VIEW CLASS="DEMO">
  ...
</VIEW>
```

标签的闭合

在小程序里，有些组件必须写成双标签，如视图容器类组件view；有些组件可以写成单标签，如媒体类组件image；但在小程序运行时，它们都会解析成双标签。

在小程序里，所有的标签一旦使用都必须被闭合，使用标签不闭合会报错，导致程序无法运行。

- 正确写法

```
<view>
  <image src="src"></image>
  <image src="src"/>
  <text>我是一段文字，我有始有</text>
</view>
```

- 错误写法

```
<view>
  <image src="src">
  <text>我是一段文字，我有始有
</view>
```

团队约定

所有具有开始标签和结束标签的元素都要写上起止标签，某些可以省略结束标签的亦都要写上

- 推荐写法

```
<view>
  <image src="src"></image>
  <input value="test"></input>
  <text>我是一段文字，我有始有</text>
</view>
```

- 不推荐写法

```
<view>
  <image src="src"/>
  <input value="test"/>
</view>
```

标签属性

团队约定

- 标签属性值使用双引号语法
- 属性值可能写上的都写上
 - 推荐写法

```
<button type="default" disabled="{{true}}"> 按钮 </button>
```

- 不推荐写法

```
<button type='default' disabled='{{true}}'> 按钮 </button>  
<button type="default" disabled> 按钮 </button>
```

- 错误写法

```
<button type=default disabled=true> 按钮 </button>
```

- 谨慎使用id属性

id属性具有唯一性，可以用来标识具体组件，应避免在样式上使用id属性（选择器）

- 属性书写顺序

标签属性应按照以下顺序依次排列，以确保代码的可读性

```
/*  
  id,  
  class,  
  wx:for wx:item wx:key,  
  wx:if,  
  src,  
  事件绑定类属性，如bind:tap,  
  value,  
  dataSet, (*需完善)  
  组件自带的其它属性,  
*/
```

特殊字符

正常情况下的小程序里，文本和字符实体不能混合出现。

- 如需使用字符实体，需使用`text`组件并设置`decode`属性，并且`decode`目前仅可解析 ` ` `<` `>` `≈` `&sp;` `&em;`，参考[text文档](#)
 - 正确用法

```
<text decode>&lt; &nbsp; &gt;</text>
```

- 错误用法

```
<text>&lt; &nbsp; &gt;</text>
<view decode>&lt; &nbsp; &gt;</view>
```

- 特殊符号使用输入法输入即可
- 连续空格的使用
 - 需使用`text`组件并设置`space`属性
 - 无`space`属性的`text`内多个连续空格最终只显示一个
 - 非`text`组件设置`space`属性不会有连续空格的效果
- 正确写法

```
<text space="ensp">1 1 1 1</text>
// 输出: 1 1 1 1
```

- 无效写法

```
<text>1 1 1 1</text>
// 输出: 1 1 1 1
```

代码缩进

统一使用2个空格字符进行代码缩进

- 推荐写法

```
<view>
  <text>2个空格字符缩进</text>
  <view>
    ...
  </text>
</view>
```

- 不推荐写法

```
<view>
  <text>4个空格字符缩进</text>
  <view>
    ...
  </view>
</view>
```

在微信开发者工具-设置-编辑设置，勾选用空格代码Tab，Tab大小设置为2，使用格式化代码可以自动缩进对齐。

代码嵌套

编写wxml代码时，需要保证页面结构稳固，同时需要避免多余的父元素，减少嵌套。

- 推荐写法

```
<view class="user_info">
  <image class="avatar" src="src"></image>
  <text class="nickname">小明</text>
</view>
```

- 不推荐写法

```
<view class="user_info">
  <view class="avatar">
    <image src="src"></image>
  </view>
  <view class="nickname">
    <text>小明</text>
  </view>
</view>
```

块级标签的起止标签各占一行，行内标签的起止标签一般写在一行内，如果标签内容过多，起止标签则各占一行。

- 推荐写法

```
<view class="user_info">
  <image src="src"></image>
  <text>小明</text>
  <text>
    原有奖励模式已改为积分奖励，针对部分未兑换礼品的用户，已为您补发积分奖励，如有疑问请联系微信客服：yiniankefu
  </text>
</view>
```

- 不推荐写法

```
<view class="user_info"><image src="src"></image><text>小明</text></view>
```