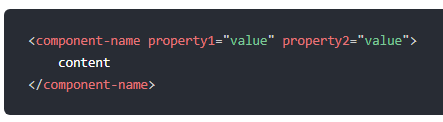
**组件使用的入门教程**

* 组件是视图层的基本组成单元。
* 组件是一个单独且可复用的功能模块的封装。

每个组件，包括如下几个部分：以组件名称为标记的开始标签和结束标签、组件内容、组件属性、组件属性值。

* 组件名称由尖括号包裹，称为标签，它有开始标签和结束标签。结束标签的<后面用/来表示结束。结束标签也称为闭合标签。如下面示例的<component-name>是开始标签，</component-name>是结束标签。
* 在开始标签和结束标签之间，称为组件内容。如下面示例的content
* 开始标签上可以写属性，属性可以有多个，多个属性之间用空格分割。如下面示例的property1和property2。注意闭合标签上不能写属性。
* 每个属性通过=赋值。如下面的示例中，属性property1的值被设为字符串value。

注意：所有组件与属性名都是小写，单词之间以连字符-连接。



下面是一个基本组件的实例，在一个vue页面的根<view>组件下插入一个<button>组件。给这个组件的内容区写上文字“按钮”，同时给这个组件设置了一个属性“size”，并且“size”属性的值设为了“mini”。

注：按照[vue单文件组件规范](https://cn.vuejs.org/v2/guide/single-file-components.html)，每个vue文件的根节点必须为 <template>，且这个 <template> 下只能且必须有一个根 <view> 组件。



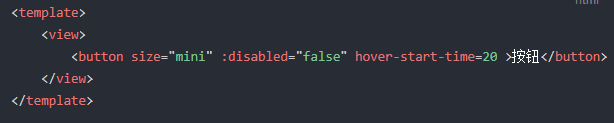
通过了解[button组件](https://uniapp.dcloud.net.cn/component/button)的文档，我们知道上述代码将在页面中绘制一个按钮，按钮显示的文字是“按钮”，按钮的大小是小尺寸的。

**组件的属性类型**

组件的属性，有多种类型：

| **类型** | **描述** | **注解** |
| --- | --- | --- |
| Boolean | 布尔值 | 组件写上该属性，不管该属性等于什么，其值都为 true，只有组件上没有写该属性时，属性值才为 false。如果属性值为变量，变量的值会被转换为 Boolean 类型。 |
| Number | 数字 | 1, 2.5 |
| String | 字符串 | "string" |
| Array | 数组 | [ 1, "string" ] |
| Object | 对象 | { key: value } |
| EventHandler | 事件处理函数名 | handlerName 是 methods 中定义的事件处理函数名 |
| Any | 任意属性 |  |

下面的例子演示了组件的属性设置boolean值和数字的例子。注意false作为一个js变量，在组件的属性中使用时，属性前面需增加:冒号前缀，属性值仍使用引号包裹，但引号里不是字符串，而是js。



**公共属性列表**

每个组件都有各自定义的属性，但所有uni-app的组件，都有如下属性：

| **属性名** | **类型** | **描述** | **注解** |
| --- | --- | --- | --- |
| id | String | 组件的唯一标示 | 一般用于获取组件上下文对象（如：[VideoContext](https://uniapp.dcloud.net.cn/api/media/video-context)），需要保持整个页面唯一 |
| ref | String | vue中组件的唯一标示 | 用来给子组件注册引用信息，详见 [Vue 文档](https://uniapp.dcloud.net.cn/vue-components#ref) |
| class | String | 组件的样式类 | 在对应的 css 中定义的样式类 |
| style | String | 组件的内联样式 | 可以动态设置的内联样式 |
| hidden | Boolean | 组件是否隐藏 | 所有组件默认是显示的 |
| data-\* | Any | 自定义属性 | 组件上触发的事件时，会发送给事件处理函数 |
| @\* | EventHandler | 组件的事件 | 详见各组件详细文档，事件绑定参考 [事件处理器](https://uniapp.dcloud.net.cn/tutorial/vue-basics#%E4%BA%8B%E4%BB%B6%E5%A4%84%E7%90%86%E5%99%A8) |

除了上述公共属性，还有一类特殊属性以v-开头，称之为vue指令，如v-if、v-else、v-for、v-model。详见[vue指令](https://uniapp.dcloud.net.cn/tutorial/vue-api#%e6%a8%a1%e6%9d%bf%e6%8c%87%e4%bb%a4)

[#](https://uniapp.dcloud.net.cn/component/#%E5%9C%A8%E7%BB%84%E4%BB%B6%E4%B8%AD%E4%BD%BF%E7%94%A8js%E5%8F%98%E9%87%8F)**在组件中使用js变量**

组件中可以使用script的data中定义的js变量，但组件的属性中使用和内容区使用的用法不一样。

* 在内容区使用时，使用两个花括号来包裹，如下面的buttonText
* 在属性值中使用时，属性名的前面要加冒号前缀

下面的button组件示例，等价于上一个的示例。只不过静态内容改成动态js。



**基础组件**

uni-app的组件，分为基础组件和扩展组件。

基础组件在uni-app框架中已经内置，无需将内置组件的文件导入项目，也无需注册内置组件，随时可以直接使用，比如<view>组件。

除了基础组件，都称为扩展组件。扩展组件需要将组件导入项目中才可以使用。

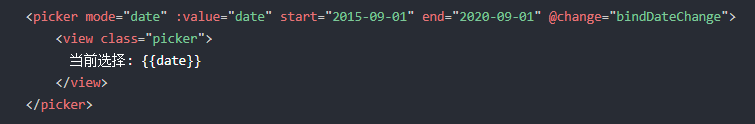
uni-app为开发者提供了一系列基础组件，类似HTML里的基础标签元素。

但uni-app的组件与HTML不同，而是与小程序相同，可更好的满足手机端的使用习惯。

虽然不推荐使用HTML标签，但实际上如果开发者写了div等标签，在编译到非H5平台时也会被编译器转换为view标签，类似的还有span转text、a转navigator等，包括css里的元素选择器也会转。但为了管理方便、策略统一，新写代码时仍然建议使用view等组件。

开发者可以通过组合这些基础组件进行快速开发。在需要复用的情况下可封装成扩展组件。

uni-app 基础组件规范，与小程序规范相近。如果了解小程序开发的话，uni-app的基础组件会感觉很熟悉。但需要注意组件上的事件绑定，需要以 vue 的事件绑定语法来绑定，如 bindchange="eventName" 事件，需要写成 @change="eventName"



**基础组件列表**

基础组件分为以下十几大类：

**视图容器（View Container）：**

| **组件名** | **说明** |
| --- | --- |
| [view](https://uniapp.dcloud.net.cn/component/view) | 视图容器，类似于HTML中的div |
| [scroll-view](https://uniapp.dcloud.net.cn/component/scroll-view) | 可滚动视图容器 |
| [swiper](https://uniapp.dcloud.net.cn/component/swiper) | 滑块视图容器，比如用于轮播banner |
| [match-media](https://uniapp.dcloud.net.cn/component/match-media) | 屏幕动态适配组件，比如窄屏上不显示某些内容 |
| [movable-area](https://uniapp.dcloud.net.cn/component/movable-view#movable-area) | 可拖动区域 |
| [movable-view](https://uniapp.dcloud.net.cn/component/movable-view#movable-view) | 可移动的视图容器，在页面中可以拖拽滑动或双指缩放。movable-view必须在movable-area组件中 |
| [cover-view](https://uniapp.dcloud.net.cn/component/cover-view#cover-view) | 可覆盖在原生组件的上的文本组件 |
| [cover-image](https://uniapp.dcloud.net.cn/component/cover-view#cover-image) | 可覆盖在原生组件的上的图片组件 |

**基础内容（Basic Content）：**

| **组件名** | **说明** |
| --- | --- |
| [icon](https://uniapp.dcloud.net.cn/component/icon) | 图标 |
| [text](https://uniapp.dcloud.net.cn/component/text) | 文字 |
| [rich-text](https://uniapp.dcloud.net.cn/component/rich-text) | 富文本显示组件 |
| [progress](https://uniapp.dcloud.net.cn/component/progress) | 进度条 |

**表单组件（Form）：**

| **标签名** | **说明** |
| --- | --- |
| [button](https://uniapp.dcloud.net.cn/component/button) | 按钮 |
| [checkbox](https://uniapp.dcloud.net.cn/component/checkbox) | 多项选择器 |
| [editor](https://uniapp.dcloud.net.cn/component/editor) | 富文本输入框 |
| [form](https://uniapp.dcloud.net.cn/component/form) | 表单 |
| [input](https://uniapp.dcloud.net.cn/component/input) | 输入框 |
| [label](https://uniapp.dcloud.net.cn/component/label) | 标签 |
| [picker](https://uniapp.dcloud.net.cn/component/picker) | 弹出式列表选择器 |
| [picker-view](https://uniapp.dcloud.net.cn/component/picker-view) | 窗体内嵌式列表选择器 |
| [radio](https://uniapp.dcloud.net.cn/component/radio) | 单项选择器 |
| [slider](https://uniapp.dcloud.net.cn/component/slider) | 滑动选择器 |
| [switch](https://uniapp.dcloud.net.cn/component/switch) | 开关选择器 |
| [textarea](https://uniapp.dcloud.net.cn/component/textarea) | 多行文本输入框 |

**路由与页面跳转（Navigation）：**

| **组件名** | **说明** |
| --- | --- |
| [navigator](https://uniapp.dcloud.net.cn/component/navigator) | 页面链接。类似于HTML中的a标签 |

**媒体组件（Media）：**

| **组件名** | **说明** |
| --- | --- |
| [audio](https://uniapp.dcloud.net.cn/component/audio) | 音频 |
| [camera](https://uniapp.dcloud.net.cn/component/camera) | 相机 |
| [image](https://uniapp.dcloud.net.cn/component/image) | 图片 |
| [video](https://uniapp.dcloud.net.cn/component/video) | 视频 |
| [live-player](https://uniapp.dcloud.net.cn/component/live-player) | 直播播放 |
| [live-pusher](https://uniapp.dcloud.net.cn/component/live-pusher) | 实时音视频录制，也称直播推流 |

**地图（Map）：**

| **组件名** | **说明** |
| --- | --- |
| [map](https://uniapp.dcloud.net.cn/component/map) | 地图 |

**画布（Canvas）：**

| **组件名** | **说明** |
| --- | --- |
| [canvas](https://uniapp.dcloud.net.cn/component/canvas) | 画布 |

**webview（Web-view）：**

| **组件名** | **说明** |
| --- | --- |
| [web-view](https://uniapp.dcloud.net.cn/component/web-view) | web浏览器组件 |

**广告**

| **组件名** | **说明** |
| --- | --- |
| [ad](https://uniapp.dcloud.net.cn/component/ad) | 广告组件 |
| [ad-draw](https://uniapp.dcloud.net.cn/component/ad-draw) | 沉浸视频流广告组件 |

**页面属性配置**

| **组件名** | **说明** |
| --- | --- |
| [custom-tab-bar](https://uniapp.dcloud.net.cn/component/custom-tab-bar) | 底部tabbar自定义组件 |
| [navigation-bar](https://uniapp.dcloud.net.cn/component/navigation-bar) | 页面顶部导航 |
| [page-meta](https://uniapp.dcloud.net.cn/component/page-meta) | 页面属性配置节点 |

**uniCloud**

| **组件名** | **说明** |
| --- | --- |
| [unicloud-db组件](https://uniapp.dcloud.net.cn/uniCloud/unicloud-db) | uniCloud数据库访问和操作组件 |

**各平台专有组件**

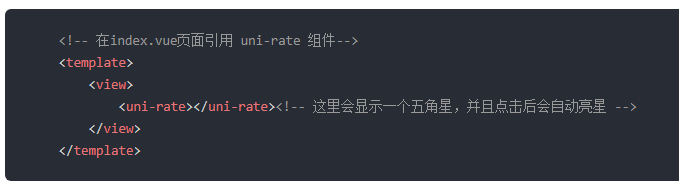
在小程序平台和nvue平台，还有一些专有组件，比如open-data，详见左侧导航

[#](https://uniapp.dcloud.net.cn/component/#%E6%89%A9%E5%B1%95%E7%BB%84%E4%BB%B6%E7%9A%84%E6%84%8F%E4%B9%89)**扩展组件的意义**

虽然所有的业务需求都可以通过基础组件满足，但仅有基础组件是低效的，实际开发中会有很多封装的组件。

比如我们需要一个五角星点击评分的组件，在DCloud的插件市场里可以获取到：<https://ext.dcloud.net.cn/plugin?id=33>

把这个uni-rate组件导入到你的uni-app项目下，在需要的vue页面里引用它，就可以在指定的地方显示出这个五角星组件。



封装扩展组件的优势：

* 可以将组件进行任意次数的复用。
* 合理的划分组件，有助于提高应用性能。
* 代码更加方便组织和管理，并且扩展性也更强，便于多人协同开发。
* 组件化开发能大幅度提高应用开发效率、测试性、复用性等。