

# 【CodeLab】如此简单！一文带你学会 15 个 HarmonyOS JS 组件

HarmonyOS 支持 Java UI 开发及 JS UI 开发，之前我们已经介绍了由 Java 常用组件开发出来的新闻详情页面，大家可以查看相关内容：

## 【CodeLab】超简单!实现新闻详情页面展示，原来只需这几类布局

本期就让我们从一个简易的 HarmonyOS 购物应用入手，通过 5 个可视页面来为大家介绍 15 个 JS 常用组件，看看 HarmonyOS 是如何通过多种组件，帮助开发者减少开发量，让开发者只需关注实现逻辑的。事不宜迟，我们马上开始。

**【请注意】**本 Demo 只为展现多种 JS 组件用法，购物应用仅为场景演示，并非完整应用，欢迎各位开发者根据自己的业务场景进行完善哦~

首先来看一下本 Demo 中涉及的 15 个组件，包括：搜索框组件 (search)、页签容器组件 (tabs)、列表组件 (list)、图片组件 (image)、分隔器组件 (divider)、交互式组件 (input)、滑动容器 (swiper)、跑马灯组件 (marquee)、自定义弹窗容器 (dialog)、滑动选择器组件 (picker)、评分条组件 (rating)、进度条组件 (progress)、菜单组件 (menu)、按钮组件 (button) 和图表组件 (chart)。

再来看一下涉及的 5 个可视页面，分别为商品陈列页面、购物车页面、我的信息页面，商品详情页面，商品数据分析页面。

我们用一个表格来让大家一目了然：

展示组件	可视页面	代码文件
搜索框组件 (search)	商品陈列页面	homepage
页签容器组件 (tabs)		
列表 (list)		
图片组件 (image)		
分隔器组件 (divider)	商品陈列页面、我的页面	
交互式组件 (input)	购物车页面	
滑动容器 (swiper)	商品详情页面	shoppingDetailsPage
跑马灯组件 (marquee)		
自定义弹窗容器 (dialog)		
滑动选择器组件 (picker)		
评分条组件 (rating)		
进度条组件 (progress)		
菜单组件 (menu)		
按钮组件 (button)	商品数据分析页面	viewsChart
图表组件 (chart)		

在正式开始敲代码之前，开发者们需要先下载安装 Huawei DevEco Studio，如果对于这个流程不甚熟悉，可以参照官网的教程操作。

Huawei DevEco Studio 安装指南：

[https://developer.harmonyos.com/cn/docs/documentation/doc-guides/software\\_install-0000001053582415](https://developer.harmonyos.com/cn/docs/documentation/doc-guides/software_install-0000001053582415)

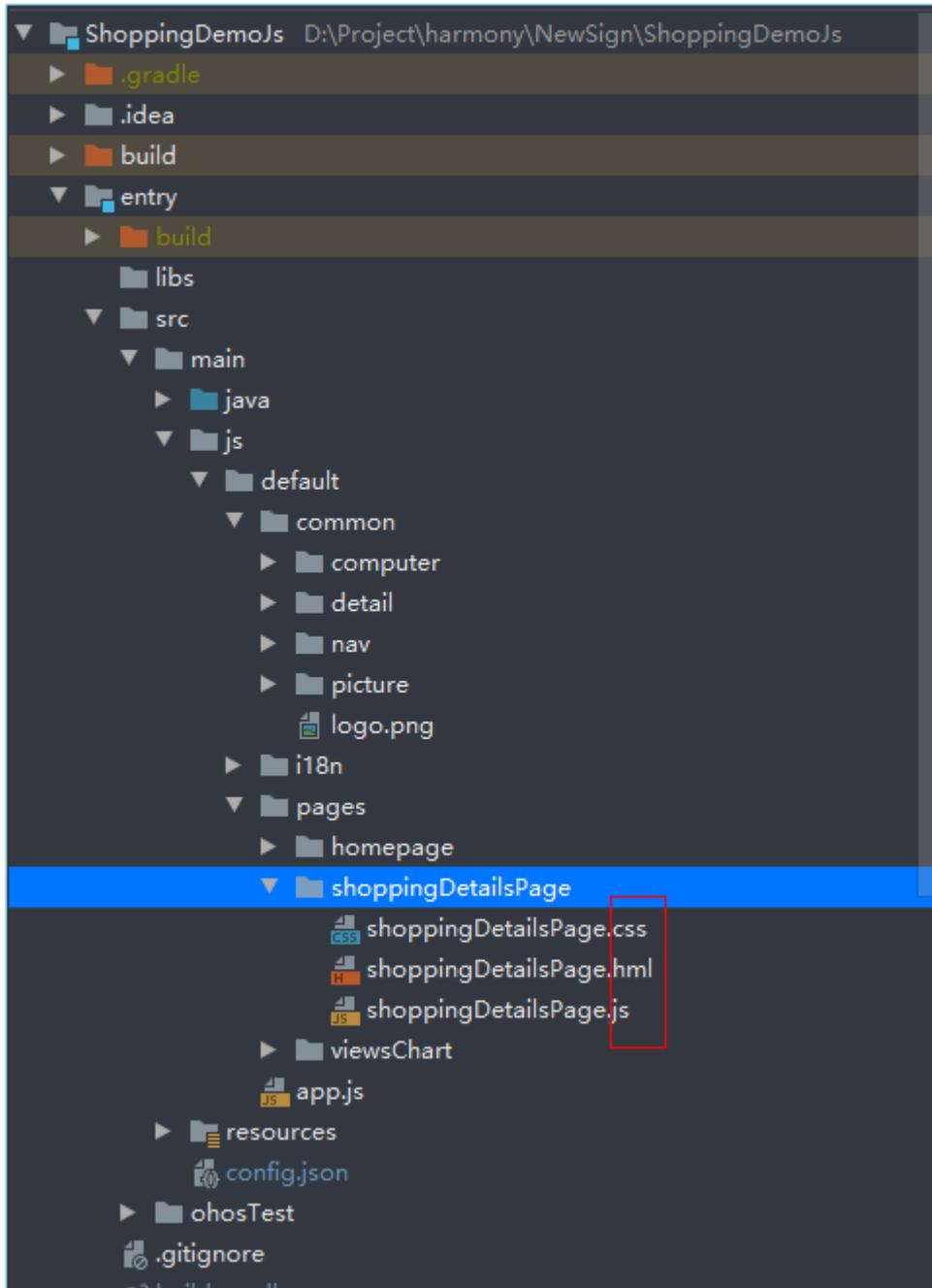
请注意，本次 Codelab 针对的是步骤拆解和重点讲解，限于篇幅原因不会展示完整代码，开发者们可在文末【阅读原文】中获取完整代码哦~

来源：HarmonyOS 开发者 [https://mp.weixin.qq.com/s/\\_ylmcJznawcTbt4LiEkfuQ](https://mp.weixin.qq.com/s/_ylmcJznawcTbt4LiEkfuQ)

## 工程代码的结构

先让我们来了解一下工程代码的结构，这里我们提取几个重要信息跟大家说明一下：

- entry/src/main/js/default/common 文件夹存放一些公共的资源，本次的图片内容皆存放在此。
- entry/src/main/js/default/pages 文件夹存放 HarmonyOS JS 的代码文件，包含 css、hml、js 三类文件，这里我们创建了 homepage、shoppingDetailsPage、viewsChart。
- entry/src/main/config.json：配置文件



## 页面构建及组件使用详解

接下来，我们就可以编写 css、hml、js 代码了。

我们先来明确一下 5 个可视页面的逻辑。Demo 从主应用界面（Demo 直接展现的页面）菜单栏可划分为三个可视页面，分别为商品陈列页面、购物车页面、我的信息页面，这三个可视页面的构建都在 homepage 中，当用户点击商品陈列页面中的商品可跳转到商品详情页（为了方便演示均采用同一商品详情页），该可视页面在 shoppingDetailsPage 中实现。此外，由商品详情页中的某个入口可跳转到商品数据分析页面，该页面在 viewChart 中编写。

下面我们将通过组件代码结合页面实现的部分，来为大家展示关键代码部分。

## homepage 代码文件

在本示例中，商品陈列页面、购物车页面、我的信息页面均在 homepage 代码文件中完成，每个可视页面均有不同的组件展现，下面我们来具体看一下。

### 商品陈列页面

在此可视页面中，用户可以通过搜索框搜索，或者直接在页面内浏览推荐商品。如果想要了解某个商品的详细信息，可以点击进入商品详情页查看。

这里我们将会用到：搜索框组件（search）、页签容器组件（tabs）、列表组件（List）、图片组件（image）

## 01 搜索框组件 (search)

搜索框组件，用于提供用户搜索内容的输入区域。

官网链接：

<https://developer.harmonyos.com/cn/docs/documentation/doc-references/js-components-basic-search-0000000000618599>

在本示例中，search 组件中的 hint 属性囊括了搜索提示文字“寻找宝贝、店铺”，value 用于搜索框搜索文本值，这里是一个固定值——手机，focusable 确定当前组件是否可以获取焦点，本例中设置为 true，可以实现用户点击时弹出，当输入框内输入的值发生变化时，会触发 change 事件，submit 实现页面底部出现“搜索词+正在搜索中...”文字字样。

代码和图片示例如下：

```
<searchhint="{{pageWord.searchKeyWord}}"value="{{pageWord.searchValue}}"
focusable="true" @submit="submitColumn">
</search>
```





## 02 页签容器 (tabs)

tabs 是一个 tab 页签容器, tab-bar 与 tab-content 均为其子组件, 其中前者用于展示 tab 的标签区。后者用于展示 tab 的内容区。

官网链接:

<https://developer.harmonyos.com/cn/docs/documentation/doc-references/js-components-container-tabs-0000000000611572>

在本示例中, 搜索框下方“热销单品”、“精品推荐”、“智慧生活”、“年货节”等就用到了页签容器组件, 组件中的 index 代表激活态, vertical 代表内容排列方式, false 为上下 true 为左右, 这里采用的是上下, 当用户点击或滑动标签时会触发 change 事件, 标签栏下方的内容会根据标签的变化而变化, 当前的标签栏会显示红色, 这样一个简单美观的标签栏就构造好啦。

而下面使用到 tabs 的子组件 tabs-bar 和 tab-content。tabs-bar 中设置该组件的排列方式为横向排列, tabs-content 用于展示 tab 的内容区, 其中的 scrollable, 用于实现滑动切换页签。

当然, 开发者可以根据需要改变 tabs 页签的页面内容、逻辑以及样式。

来源: HarmonyOS 开发者 [https://mp.weixin.qq.com/s/\\_ylmcJznawcTbt4LiEkfuQ](https://mp.weixin.qq.com/s/_ylmcJznawcTbt4LiEkfuQ)

代码和图片示例如下：

```
<tabs class="tabs" index="0" vertical="false" onchange="change">
  <tab-bar class="tab-bar" mode="fixed">
    <text class="tab-text" for="{{ item in titileList}}">{{ item }}
  </text>
</tab-bar>
<tab-content class="tabcontent" scrollable="true">
  <div class="item-content" for="{{ item in contentList}}">
    <list class="todo-wraper">
      <list-item for="{{lists}}">
        <div class="margin10" @click="detailPage">
          <div class="todo-total">
            <text class="todo-title">
              {{$item.title}}
            </text>
            <text class="todo-content">{{$item.content}}
            </text>
            <text class="todo-price">
              <span> ¥
            </span>
          </div>
        </div>
      </list-item>
    </list>
  </div>
</tab-content>
```

```
                <span>{{ $item.price }}
            </span>
        </text>
    </div>
    <div class="width30">
        <image
            src="{{ $item.imgSrc }}"
class="container-home-image">
        </image>
    </div>
</div>
</list-item>
</list>
</div>
</tab-content>
</tabs>
```



### 03 列表组件 (List)

List, 即列表, 包含一系列相同宽度的列表项, 适合连续、多行呈现同类数据, 例如图片和文本。list-item 是 list 的子组件, 用来展示列表具体 item。

来源: HarmonyOS 开发者 [https://mp.weixin.qq.com/s/\\_ylmcJznawcTbt4LiEkfuQ](https://mp.weixin.qq.com/s/_ylmcJznawcTbt4LiEkfuQ)

官网链接:

[https://developer.harmonyos.com/cn/docs/documentation/doc-references/lite-we  
arable-container-list-0000001123447025](https://developer.harmonyos.com/cn/docs/documentation/doc-references/lite-we<br/>arable-container-list-0000001123447025)

在本示例中，商品页面宽度高度整齐的排列，就依托于列表组件在.html 中声明以及在.css 中进行样式设置。

部分代码和图片示例如下:

```
<listclass="todo-wraper">  
  
  <list-itemfor="{{lists}}">  
  
    <div@click="detailPage">  
  
      <divclass="todo-total">  
  
        <textclass="todo-title">  
  
          {{{item.title}}}  
  
        </text>  
  
        <textclass="todo-content">{{{item.content}}}  
  
        </text>  
  
        <textclass="todo-price">  
  
          <span>  
  
            ¥  
  
          </span>  
  
          <span>  
  
            {{{item.price}}}
```

```
        </span>
    </text>
</div>
<divclass="width30">
    <imagesrc="{{item.imgSrc}}">
    </image>
</div>
</div>
</list-item>
</list>
```



#### 04 图片组件 (image)

image，图片组件，在开发应用中，也是比较常见的一个组件，用来渲染展示图片。

官网链接：

<https://developer.harmonyos.com/cn/docs/documentation/doc-references/js-components-basic-image-0000000000611597>

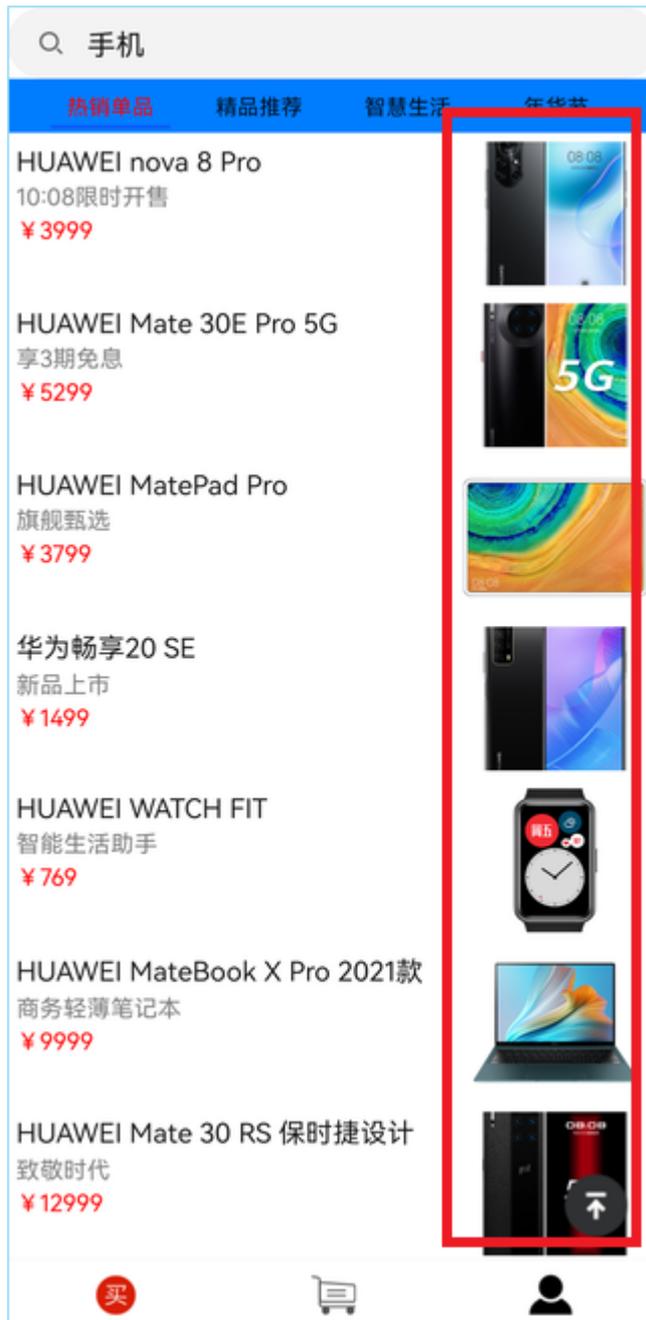
在本示例中，商品的图片展示就是利用了图片组件来完成，其中 src 中是图片的路径。部分

来源：HarmonyOS 开发者 [https://mp.weixin.qq.com/s/\\_ylmcJznawcTbt4LiEkfuQ](https://mp.weixin.qq.com/s/_ylmcJznawcTbt4LiEkfuQ)

代码和图片示例如下：

```
<image src="{{item.imgSrc}}"class="container-home-image">
```

```
</image>
```



## 购物车页面

购物车页面让用户可以把选中的商品加入其中，并在其中选择想要的商品用于结算。在本例中为了便于大家理解，购物车中的商品已经锁定。这里我们将会用到分隔器组件（divider）和交互式组件（input）。

### 05 分隔器组件（divider）

分隔器组件（divider）用于分隔不同内容块/内容元素，可用于列表或界面布局。

官网链接：

<https://developer.harmonyos.com/cn/docs/documentation/doc-references/js-components-basic-divider-000000000618583>

在本示例中，多次使用到分隔器组件（divider），在这个页面中，我们仅做样式变化，但divider还可以使用vertical属性实现组件见的分割，比如在购物车页面中，我们使用该分割器把购物车的商品进行分割，用户看起来也美观一些。

该组件使用所在的购物车页面代码及图片如下：

```
<divider class="dialog-divider" >  
  
</divider>
```



## 06 交互式组件 (input)

交互式组件 (input) 包括单选框, 多选框, 按钮和单行文本输入框, 在本例中用于购物车产品的选择。

官网链接:

<https://developer.harmonyos.com/cn/docs/documentation/doc-references/js-components-basic-input-0000000000611673>

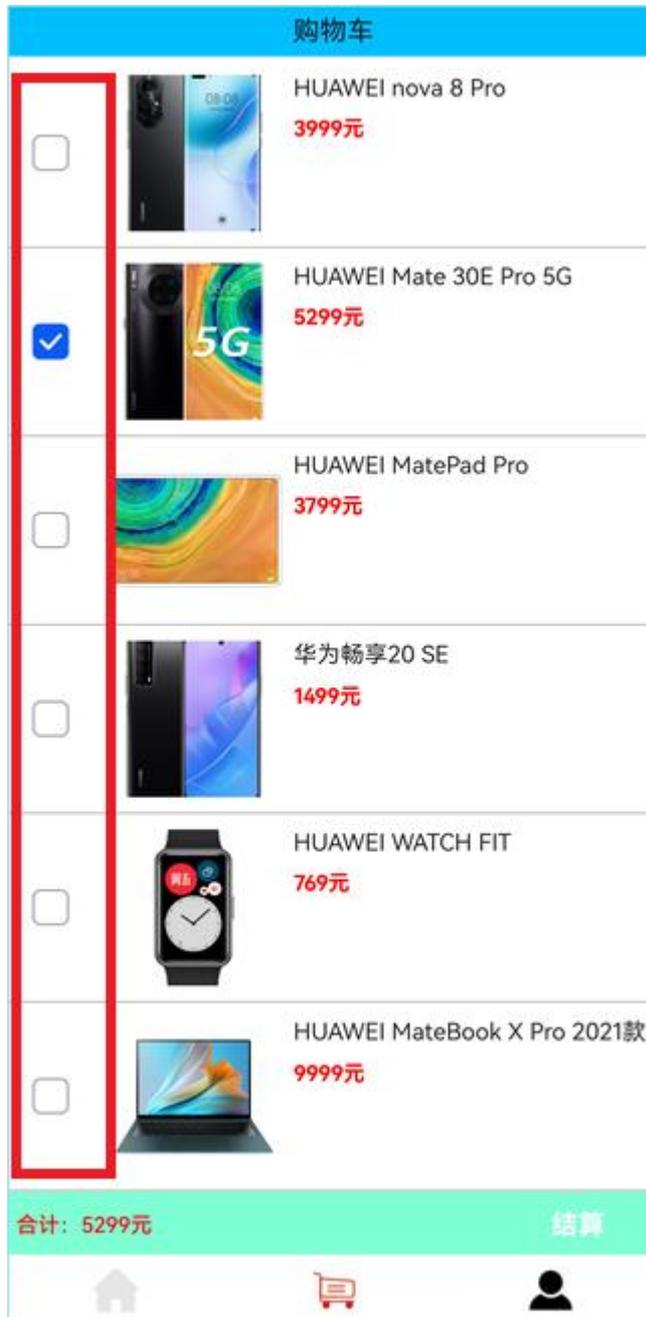
在本 Demo 中, 我们多次用到交互式组件。其中, 在购物车页面中, input 定义为多选框, 用户可以选择自己喜欢的一个或多个商品进行结算, type="checkbox" 类型定义为多选框 value 是 input 组件的 value 值, 在此地方是商品的价格 @change, 当多选框勾选状态发生变化时触发该事件。

代码及图片如下:

来源: HarmonyOS 开发者 [https://mp.weixin.qq.com/s/\\_ylmcJznawcTbt4LiEkfuQ](https://mp.weixin.qq.com/s/_ylmcJznawcTbt4LiEkfuQ)

```
<input type="checkbox" value="{{item.price}}" @change="addShopping">
```

```
</input>
```



## 我的信息页面

我的信息页面布局，主要使用的是图片组件 (image)，用来呈现页面中的图标，使用方法

与前面类似，这里就不多说。直接上代码和示例图片：

```
<div>

  <textclass="container-my-deals-text">{{pageWord.myDeals}}</text>

  <divclass="container-my-deals-div">

    <divclass="container-my-deals-div-div"for="{{transaction}}">

      <imagesrc="{{item.src}}"class="container-my-image"></image>

      <textclass="container-my-text">{{item.title}}{{item.num}}</text>

    </div>

  </div>

</div>
```



完成了上述三个可视页面的开发，我们接下来要开发商品详情页面。用户在商品陈列页面中点击任意商品即可跳转到该页面，此跳转动作在 homepage 的.js 文件中实现。由于不属于组件内容，这里只简单显示页面跳转的代码：

```
detailPage() {  
  
    router.push({
```

```
        uri:"pages/shoppingDetailsPage/shoppingDetailsPage"  
    })  
},
```

## shoppingDetailsPage 代码文件

在本示例中，shoppingDetailsPage 代码文件主要包含商品详情页面的构建。

### 商品详情页面

商品详情页面将会展示商品更加详细的信息，包括头部的滑动图片、价格、收货地址选择等。

涉及到的组件包括：滑动容器(swiper)、跑马灯组件(marquee)、自定义弹窗容器(dialog)、滑动选择器 (picker)、评分条 (rating)、进度条组件 (progress) 和菜单组件 (menu) ，

下面我们来具体看一下。

#### 07 滑动容器 (swiper)

滑动容器，用户可以在 swiper 组件上进行滑动，用户可以左右切换图片，或者等待页面 3s 自动滑动一次。

官网链接：

<https://developer.harmonyos.com/cn/docs/documentation/doc-references/lite-we>

arable-container-swiper-0000001078560398

在本示例中，通过图片轮播图的形式对商品图片进行展示。其中，class="swiper"是构建swiper的样式，id表示"swiper"是该组件的唯一标识，index显示当前容器的第几张图片，在本例中默认是从0开始，autoplay="true"表示开启图片自动循环播放，interval="3000"是每隔3S切换一次图片，indicator="true"启用导航点指示器，loop="true"开启循环轮播。digital="false"，关闭数字导航点，当开发者以往显示数字的时候，就可以设置成true。

代码及图片示例如下：

```
<swiperclass="swiper" id="swiper" index="0"autoplay="true" interval="3000"
indicator="true"loop="true" digital="false">
  <divfor="{{ item in swiperList }}">
    <imagesrc="/common/computer/computer{{item}}.png"> </image>
  </div>
</swiper>
```



## 08 跑马灯组件（marquee）

展示一段单行滚动的文字。

官网链接：

<https://developer.harmonyos.com/cn/docs/documentation/doc-references/js-components-basic-marquee-000000000611657>

在本示例中，该跑马灯组件是用于展示一段文字从左往右无限循环滚动。其中 id 是跑马灯组件的唯一标识。scrollamount 跑马灯每次滚动时移动的最大长度，开发者可以根据实际情况给予长度。当 loop=-1 的时候，表示连续滚动。direction 是设置跑马灯的文字滚动方向。

代码和图片示例如下：

```
<marqueeid="customMarquee"scrollamount="{{scrollAmount}}"  
loop="{{loop}}"direction="{{marqueeDir}}" @bounce="onMarqueeBounce"
```

来源：HarmonyOS 开发者 [https://mp.weixin.qq.com/s/\\_ylmcJznawcTbt4LiEkfuQ](https://mp.weixin.qq.com/s/_ylmcJznawcTbt4LiEkfuQ)

```
@start="onMarqueeStart"@finish="onMarqueeFinish">
```

```
  {{pageInfo.marqueeCustomData}}
```

```
</marquee>
```



## 09 自定义弹窗容器 (dialog)

dialog, 自定义弹窗容器, 在本例中调出省市级选择和时间选择。

官网链接:

<https://developer.harmonyos.com/cn/docs/documentation/doc-references/js-components-container-dialog-0000001050068157>

在开发此应用中, 多次用到自定义弹窗容器 (dialog), 开发者可以通过 `this.$element('id').show();` 调起 dialog 弹窗, 通过 `this.$element('id').close();` 关闭 dialog 弹窗, id 是 dialog 组件的唯一标识, 定义在 dialog 组件内, 具体的详细页面开发者可以根据需要在 dialog 中进行编写代码。

代码与图片示例与下面滑动选择器 (picker) 合并展示。

## 10 自定义滑动选择器 (picker)

滑动选择器 (picker), 滑动选择器组件, 类型支持普通选择器, 日期选择器, 时间选择器,

来源: HarmonyOS 开发者 [https://mp.weixin.qq.com/s/\\_yImcJznawcTbt4LiEkfuQ](https://mp.weixin.qq.com/s/_yImcJznawcTbt4LiEkfuQ)

时间日期选择器，多列文本选择器。

官网链接：

<https://developer.harmonyos.com/cn/docs/documentation/doc-references/js-components-basic-picker-000000000621791>

在本例中使用到 dialog 自定义弹窗容器和滑动选择器 (picker) ，点击选择会弹出 dialog，页面会有时间选择器和省市级联选择器。

用户在弹出的时间选择器上选择日期，点击确定，页面上的数据也会同时改变。省市级联选择器，同理。其中日期选择器：type="date"。省市级联选择器：type="multi-text"我们选择多列文本选择器。@columnchange 是当多列文本选择器中某一列的值改变时会触发此事件，columns 是设置多列文本选择器的列数。range 是设置多列文本选择器的选择项。value 是设置多列文本选择器的值，即页面上显示的默认值，@change 当用户选择的值点击确认按钮时触发该事件。

代码与图片如下（这里仅展示时间选择图片）展示如下：

```
<dialogid="simpdialog">  
  
  <div>  
  
    <div>  
  
      <divclass="container-margin-left-comm">
```

```
<imagesrc="/common/computer/computer1.png"class="dialog-div-titleimage">
```

```
</image>
```

```
</div>
```

```
<divclass="dialog-flex-justify">
```

```
<textclass="dialog-divs-text1">
```

```
    {{pageInfo.annualPrice}}
```

```
</text>
```

```
<textclass="dialog-divs-text2">
```

```
    {{pageInfo.inventory}}
```

```
</text>
```

```
</div>
```

```
<imagesrc="/common/computer/icon-close.png"class="dialog-divs-icon-close"
```

```
    @click="cancelSchedule">
```

```
</image>
```

```
</div>
```

```
<divclass="dialog-div-select">
```

```
<div>
```

```
<textclass="dialog-div-select-text">
```

```
    {{pageInfo.selectRewardTime}}
```

```
</text>
```

```
<divclass="dialog-div-select-picker">
```

```

        <picker type="date"selected="{{ newDate }}" hours="24"
value="{{ newDate}}"/>
            @change="changeDate" @cancel="cancelDate">
        </picker>
    </div>
</div>
<div>
    <textclass="dialog-div-select-text">
        {{pageInfo.selectRewardCity}}
    </text>
    <divclass="dialog-div-select-picker">
        <pickertype="multi-text"
@columnchange="columnTextData"columns="3" range="{{cityList}}"/>
            value="{{selectCityList}}" @change="changeCity">
        </picker>
    </div>
</div>
</div>
<divclass="dialog-div-bottom" @click="buy">
    <text>
        {{pageInfo.nowSell}}
    </text>

```

</div>

</div>

</dialog>



## 11 菜单（menu）

菜单（menu），提供菜单组件，作为临时性弹出窗口，用于展示用户可执行的操作。

官网链接：

<https://developer.harmonyos.com/cn/docs/documentation/doc-references/js-components-basic-menu-000000000618614>

在本示例中，该菜单组件展示四个菜单内容分别为：评分、分享、浏览量、退出。当用户点击任意菜单内容，都会执行对应的事件逻辑。其中 id 是菜单组件的唯一标识，可以通过 this.\$element("id").show 调起菜单组件。@selected 是当选中菜单中某个值被点击选中时触发的事件。

代码如下：

```
<menu id="apiMenu"@selected="onMenuSelected">
  <option value="Item-1">
    {{pageInfo.scoring}}
  </option>
  <option value="Item-2">
    {{pageInfo.sharing}}
  </option>
  <option value="Item-3">
    {{pageInfo.views}}
  </option>
  <option value="Item-4">
    {{pageInfo.exit}}
  </option>
</menu>
```

右上角操作菜单（menu），点击即会出现一系列菜单选项可供选择，如图所示：



## 12 评分条组件（rating）

评分条，相信开发者都不陌生。用户点击评分，页面会弹出带有评分弹框的 dialog 弹窗容器。

官网链接：

<https://developer.harmonyos.com/cn/docs/documentation/doc-references/js-components-basic-rating-000000000611645>

在本示例中，该评分条组件用于给软件评分，用户可以选择 0-5 颗星进行打分，其中可以出现 0.5 的情况，例如：2.5 分、3.5 分。numstars 设置评分条的星级总数，默认为 5。rating 设置评分条当前评星数，即默认评星数。@change 当评分条的评星发生改变时触发该事件。

代码及图片如下：

```
<ratingnumstars="5" rating="{{ratingNum}}" @change="ratingChange">
```

来源：HarmonyOS 开发者 [https://mp.weixin.qq.com/s/\\_ylmcJznawcTbt4LiEkfuQ](https://mp.weixin.qq.com/s/_ylmcJznawcTbt4LiEkfuQ)

</rating>



### 13 进度条组件 (progress)

进度条，在本 Demo 中，点击立即抢购会弹出一个含有进度条 (progress) 的弹框。

官网链接：

<https://developer.harmonyos.com/cn/docs/documentation/doc-references/js-components-basic-progress-0000000000611609>

在本示例中，该进度条组件用于商品抢购，当点击抢购按钮，系统会弹框显示进度条组件，进度条从 0-100，当进度达到 100%，会提示抢购成功。type="scale-ring"设置成带刻度圆环形进度条。percent 最初页面当前进度的值。secondarypercent 最初页面次级进度的值。

图片示例和代码如下：

```
<progressclass="min-progress"
```

```
type="scale-ring"percent="{{progress.percent}}"secondarypercent="{{progress.sec
```

```
ondarypercent}}">
```

```
</progress>
```





## viewChart 代码文件

ViewChart 代码文件于构建商品数据分析页面。

## 商品数据分析页面

商品数据分析页面在本 Demo 中仅为了呈现图表及按钮组件，通过商品详情页面右上角的菜单中的“浏览量”中进入。涉及的组件包括 chart 组件和 button 组件。

### 14 图表组件 (chart)

chart 组件为图表组件，用于呈现线形图、柱状图、量规图界面，本例中用的曲线图可以实时动态更新数据。其中，type="line"，把图表设置成线形图。ref="linechart"，用来指定指向子元素或子组件的引用信息，该引用将注册到父组件的\$refs 属性对象上，我们最常用的就是用 ref 对图表进行添加数据 this.\$refs.linechart.append()。options 是设置图表参数， datasets 设置图标的数据集合及其背景颜色。

代码及图片示例如下：

```
<chartclass="chart-data" type="line" ref="linechart"options="{{lineOps}}"  
datasets="{{lineData}}">  
  
</chart>
```



## 15 按钮组件（button）

按钮组件最常用到的组件，该组件包括胶囊按钮、圆形按钮、文本按钮、弧形按钮、下载按钮。这里就不赘述了。

官网链接：

<https://developer.harmonyos.com/cn/docs/documentation/doc-references/js-components-basic-button-0000000000621726>

在开发此应用中，button 是和 chart 组件结合起来使用的，当我们触发第一个按钮的时候，我们会加快 chart 组件刷新数据的速率，当我们触发第二个按钮的时候，我们会手动随机添加一个数据。

代码及图片如下：

```
<button value="setInterval" onclick="autoAddData" >
```

```
</button>
```

```
<button value="Add data"onclick="addData">
```

```
</button>
```



至此，我们已经通过一个简单的购物应用，体验了 JS 的 15 个常见组件。