

24 List列表样式的设计与开发

更新时间：2019-07-31 17:43:28



“你若要喜爱你自己的价值，你就得给世界创造价值。”

——歌德

这一节我们来实现列表组件。在移动端，列表最常用的有两个地方。一种是用来展示城市或通讯名单这类比较单一的数据，这种列表的内容一般就是单行的文本。另外一种列表是用来展示商品信息这类内容，这种列表里每一行里的内容会比较多，需要对内容的排布进行设计。移动端里很少使用表格，表格的功能通常就是用列表来实现的。接下来我们就来设计和实现两种形式的列表。

列表组件的需求

一、简单列表

简单列表的设计就很简单了，和我们之前做过的一些组件有相似的地方。比如在表单元素中存放每一条表单的容器，还有上一节的搜索建议部分，其实和简单列表的样式都很相似。我们按如下的样式设计：



这个列表的样式很好实现，这里面只需要注意为了和复杂列表共用每一条的容器，我们不要对每一行的元素做高度和行高的限制。

二、复杂列表

复杂列表的每一行因为要呈现多个信息，所以内容区要重新布局。公司的业务里主要是商品展示，所以在UI里我们使用商品信息作为复杂列表的填充内容，样式如下：



在这个复杂列表中，我们会有如下要求：

1. 复杂列表和简单列表可以使用相同的内边距撑开内容区和边界的距离。
2. 每一行的内容区分左右结构，左边图片固定宽度，右侧文字区充满剩余区域。
3. 内容区分多行，其中要保证第一行和内容区上边对齐，最后一行和内容区底边对齐。
4. 第一行为标题，限制最多显示两行内容，可以自动隐藏超长部分，并在文本末尾追加省略号。

针对这些需求，下来我们就可以做列表组件的设计与开发了。

列表组件的设计与开发

一、文件的建立

第一步，我们还是先把这一节需要的文件建立出来。我们先来建一个空的html文件/demo/list.html:

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,initial-scale=1.0,user-scalable=no">
    <link rel="stylesheet" href="../src/tuitui-ui.css">
    <title>推推UI-列表</title>
  </head>
  <body>
    <div class="tt-content">
      <h1 class="tt-panel-title">简单列表</h1>
      <div class="tt-panel-body no-padding">
        <!-- 简单列表 -->
      </div>
      <h1 class="tt-panel-title">复杂列表</h1>
      <div class="tt-panel-body no-padding">
        <!-- 复杂列表 -->
      </div>
    </div>
  </body>
</html>
```

然后是对应的 CSS 文件 /src/list.css:

```
/*
 * @Author: Rosen
 * @Date: 2019-07-18 11:10:09
 * @Last Modified by: Rosen
 * @Last Modified time: 2019-07-19 23:57:26
 */

/* 列表的样式 */
```

最后在 `/src/tuitui-ui.css` 主文件最后引入刚才添加的 `list.css` 文件：

```
/* 列表的样式 */
@import './list.css';
```

这样，这一节所需要的文件就建好了，下来我们开始设计和开发两种列表。

二、简单列表的设计与开发

我们先来开发简单列表，简单列表的样式比较简单，我们之前也做过类似的样式，所以这里就直接贴代码了。直接在HTML中“`<!-- 简单列表 -->`”的位置后面加上下面的代码：

```
<!-- 简单列表 -->
<ul class="tt-list">
  <li class="tt-list-item">北京</li>
  <li class="tt-list-item">上海</li>
  <li class="tt-list-item">天津</li>
  <li class="tt-list-item">重庆</li>
  <li class="tt-list-item">厦门</li>
  <li class="tt-list-item">广州</li>
  <li class="tt-list-item">...</li>
</ul>
```

这里我们以城市列表为例，使用 `ul` 元素来实现这个列表。这个列表的样式也是给 `.tt-list-item` 加上一些简单样式就可以实现了，我们可以在 `/src/list.css` 中添加下列样式：

```
/* 列表里每一行的容器 */
.tt-list > .tt-list-item{
  position: relative;
  padding: .5rem 1rem;
  font-size: .8rem;
  background: #fff;
  color: #333;
  border-bottom: 1px solid #eee;
}
```

只需要这一条样式，简单列表的样式就实现了。

三、复杂列表的设计与实现

复杂列表中，每一行的外层容器可以和简单列表共用，比较麻烦的就是处理每一行里内容区的布局。按着示意图里的样式，图片和文本区域左右排列，如下图：



这个结构里，左侧图片固定宽度，右侧文本区域需要填充剩余空间的需求，所以这又是一个特别适合弹性布局的场景。左边的图片区域可以直接使用一个 `img` 元素，但我们这里会在 `img` 外层加一层容器，这样如果需要在图片位置加一些标签的话会更容易定位。右侧区域就是三个文本段落依次排列。按着这样的结构，我们可以先建立出复杂列表的HTML结构，在 `/demo/list.html` 文件中“`<!-- 复杂列表 -->`”的后面加入复杂列表的HTML代码：

```
<!-- 复杂列表 -->
<ul class="tt-list">
<li class="tt-list-item">
<div class="item-img-wrap">

</div>
<div class="item-content-wrap">
<h1 class="item-title">FILCO斐尔可 机械键盘87游戏无线圣手忍者二代红轴茶青黑蓝牙双模</h1>
<p class="item-price">¥ 998</p>
<p class="item-desc">已售1834件</p>
</div>
</li>
<li class="tt-list-item">
<div class="item-img-wrap">

</div>
<div class="item-content-wrap">
<h1 class="item-title">FILCO斐尔可 机械键盘87游戏无线圣手忍者二代红轴茶青黑蓝牙双模</h1>
<p class="item-price">¥ 998</p>
<p class="item-desc">已售1834件</p>
</div>
</li>
<!-- ...其他多个列表项 -->
</ul>
```

为了缩短篇幅，这个列表里只包含了两条列表数据。在实现列表里内容区样式的时候，左右结构的处理和图片的处理都比较常规，只有右侧文本区的要求有些多。文本区分为标题、价格和销量这三部分内容，这三个元素要占满文本区域的高度，并且标题要靠顶，销量描述要贴底。对于这种需求能想到最简单的做法就是先固定每个元素的高度，然后通过计算出的内边距或外边距来撑开整个区域。但是我们这里准备用弹性布局里的“`justify-content`”属性来实现这个效果。下面先来介绍一下“`justify-content`”。

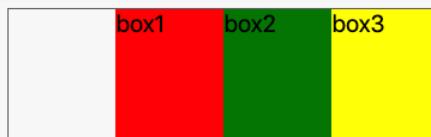
@ Tips:

justify-content 属性是在弹性布局中，用来定义主轴上元素的排列方式的，这个属性要作用于弹性布局中的容器上。这个属性有以下可取的值：

flex-start: 弹性盒子里的所有元素从容器起始位置开始依次排列。这个起始位置并不是固定的，它会受到“flex-direction”的影响。比如“flex-direction”的值如果是“column-reverse”，那起始位置就在容器的最下面；如果“flex-direction”的值是默认的“row”，那么起始位置就是容器的最左边。效果如下：



flex-end: 和上面的 **flex-start** 相反，使用这个属性时，所有元素会和弹性盒子的末尾对齐。但要注意，使用这个属性值并不会改变元素的排布顺序。使用 **flex-end** 的效果如下：



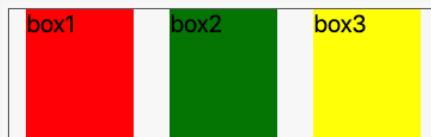
center: 这个属性会让弹性盒子里的所有元素排布在页面的中间，把所有空白区域平均分配在容器的两边。使用 **center** 时的效果如下：



space-between: 这个属性是让容器里的空白空间平均分配在每两个元素中间，也就是第一个元素和容器起始位置对齐，最后一个元素和容器结束位置对齐，而每两个元素中间都留下等距的空间。这个属性就和上面列表项里文本区域排布的需求比较吻合，只是在竖直方向上应用就可以了。下面是使用 **space-between** 时的效果：



space-around: 这个属性也是会把空白空间进行切割分配，和 **space-between** 不同的是这种分隔方式会给每一个元素两侧都分配等距的空间。这样在容器的两边会留出一份的空间，而每两个元素中间会有两份空间的间隔。使用 **space-around** 时的效果如下：



在使用 **justify-content** 属性的时候有两点要注意：

1. **justify-content** 是弹性布局中特有的属性，所以要和“`display: flex;`”配合使用。
2. 如果弹性盒子里的元素有使用“`flex-grow: 1;`”这类属性或者其他方式导致弹性容器已经被撑满，那么这时候

justify-content就没什么意义了。

前面已经把复杂列表的结构分析完了，也补上了 `justify-content` 属性的知识，下面我们进入开发环节。首先修改一下在实现简单列表时建立的列表项容器的样式，只需要给它加上弹性布局的属性：

```
/* 列表里每一行的容器 */
.tt-list > .tt-list-item{
  display: flex;
  position: relative;
  padding: .5rem 1rem;
  font-size: .8rem;
  background: #fff;
  color: #333;
  border-bottom: 1px solid #eee;
}
```

接下来是图片的部分，我们给图片的容器固定的宽度和高度，并且不允许伸缩，然后加一个右侧的外边距来和内容区留下一点距离，最后再加上轻微的圆角效果即可：

```
/* 列表项的图片容器 */
.tt-list .item-img-wrap{
  flex: 0 0 5rem;
  height: 5rem;
  margin-right: .5rem;
  border-radius: .2rem;
  overflow: hidden;
}

/* 列表项的图片 */
.tt-list .item-img-wrap > .item-img{
  width: 100%;
  height: 100%;
}
```

这里有一点需要注意下，就是“`.tt-list .item-img-wrap`”这个选择器使用的是后代选择器，而没有使用子代选择器，这是因为经常会在列表项里加上跳转用的a标签，使用后代选择器更合理。我们这里也省略了`.tt-list-item`这一层，直接把`.item-img-wrap`限制在了`.tt-list`里即可。

下来就是文本区域的容器样式：

```
/* 列表项的文字区域 */
.tt-list .item-content-wrap{
  position: relative;
  flex: 1;
  display: flex;
  flex-direction: column;
  justify-content: space-between;
}
```

这个容器要占满左右结构里的空白空间，所以可以使用“`flex: 1;`”；然后为了方便容器里绝对定位元素的使用，给它设置成相对定位；最后就是它里面三个元素需要竖直方向排列，且需要两边对齐，所以这里我们使用了“`flex-direction`”和“`justify-content`”这两个弹性布局的属性配合来实现需要的效果。

再下来就是文本区域里三个元素的样式了，这里面标题部分需要两行的高度，并且超长的时候要截断，这里我们会使用下面这个固定的搭配来实现多行的折行截断效果：

```
div{
  display: -webkit-box;
  -webkit-line-clamp: 2;
  -webkit-box-orient: vertical;
}
```

这三条固定的搭配就可以实现文本多行的截断，并且在截断处自动追加省略号。但需要注意的是，这几个属性并不是CSS3中标准的属性，并且只有在webkit内核的浏览器中且加上-webkit-前缀才有效。我们最需要的是对多行文本超出的部分做隐藏，而对添加省略号的需求不是很强，所以这几个元素可以配合height和line-height来使用，这样哪怕这三个属性失效了，也能保证正常的显示效果。

有了上面的基础，再去实现文本内容区的几个样式就简单了，我们直接贴代码吧：

```
/* 列表项的标题 */
.tt-list .item-content-wrap > .item-title{
  height: 2rem;
  line-height: 1rem;
  font-size: .8rem;
  font-weight: normal;
  overflow: hidden;
  display: -webkit-box;
  -webkit-line-clamp: 2;
  -webkit-box-orient: vertical;
}

/* 列表项的加个字段 */
.tt-list .item-content-wrap > .item-price{
  font-size: .8rem;
  font-weight: 600;
  color: #e0652f;
}

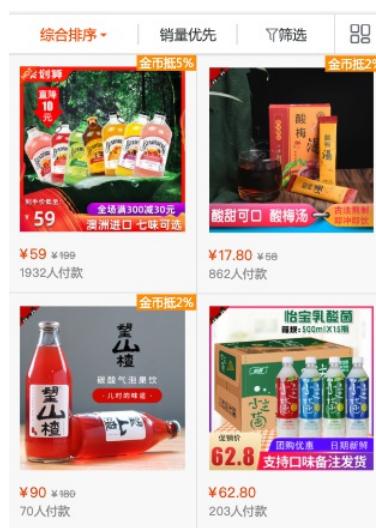
/* 列表项的描述字段 */
.tt-list .item-content-wrap > .item-desc{
  color: #999;
  font-size: .6rem;
}
```

标题部分我们通过把高度设置成行高的两倍，来控制最多显示两行文本，其余两个元素就是设置了简单的文本样式。这样我们整个复杂列表也就完成了，最终的结果就会和开始贴的效果图完全一致。到这里，我们这一节的任务就结束了。

结语

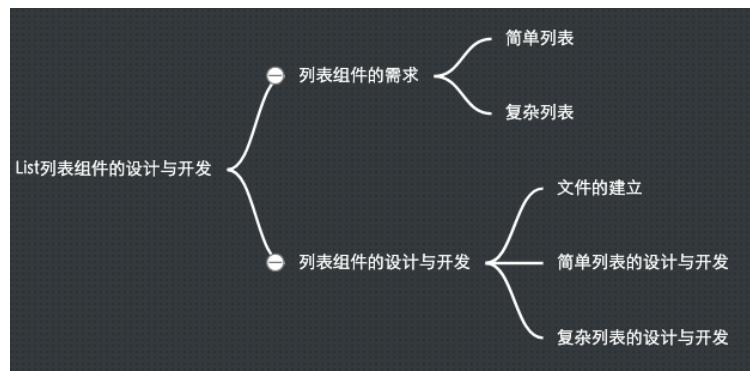
这一节我们开发了两种类型的列表，简单列表的开发很简单，复杂列表里主要讲的是各个元素的排列方式。这一节涉及了弹性布局中的justify-content属性的应用，同学们下去把例子里的效果图都自己实现一下。

下面给同学们留一个思考题，在移动端还有一种比较常用的两列的列表，例如下面这样：



同学们下去试着在现有的基础上，实现通过一个class来切换两种列表的样式。

最后附上这一节内容的结构：



我们这一节的内容就到这里，同学们可以访问 [【List列表组件在线预览】](#) 来查看这一节的演示效果，下一节将网格组件的开发。

[23 Search搜索框样式的设计与开发](#)

[25 Grid网格样式的设计与开发](#)

精选留言 1

欢迎在这里发表留言，作者筛选后可公开显示

iam593

小小笔误：CSS 文件 /src/list.html -> CSS 文件 /src/list.css

凸 0 回复

2天前

chenchen1111 回复
iam593

感谢反馈~~已经改正~~

回复

1天前