19 表单元素的开发

更新时间：2019-07-23 10:56:36

> 散步促进我的思想。我的身体必须不断运动，脑筋才会开动起来。
>
> —— 卢梭

这一节我们来开发表单里常用的组件。在移动端，由于屏幕尺寸都比较小，一般不会设计特别复杂的表单。在移动端用的最多的就是输入框和选择这两类输入方式。这一节我们就来做一下表单里这些常用样式的开发，会包括基础的输入框、带附加功能的高级输入框以及单选和多选。此外，这一节还会介绍下移动端常用来展示内容区块用的面板 Panel。

我们这节要开发的目标如下：

这里面包括各种类型的输入框，以及对输入框加上操作和提醒用的功能按钮，最后是单选和多选这两种选择形式的输入方式。下来我们就逐个的来实现它们。

面板 **Panel** 的开发

在移动端，经常用到面板 Panel 这种内容容器，这种容器通常用来存放一块独立的内容。Panel 容器一般都是由一个标题和一个内容区组成，在有些时候也可以没有标题。在微信中可以找到Panel 的实例，可以是酱婶儿的：

也可以是酱婶儿的：

对于这个容器样式，我们要注意到如下的细节：

1. 标题部分属于辅助，视觉效果上需要弱化，可以把字体颜色设置的浅一点。
2. 给标题栏添加上边框，给内容区添加上下边框来做分区，这样没有标题栏的时候内容区边框也不会有问题。
3. 默认情况下，内容区需要一定的内边距，不让内容紧贴着边。
4. 有些内容需要贴边的时候，也可以通过 class 去掉内边距。
5. 同一个页面内可以有多个 Panel，每个 Panel 间需要有间隔。

根据这些要求，我们先把 HTML 文件建立出来，这里我们新建一个 /demo/input.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>表单元素</title>
  </head>
  <body>
    <div class="tt-content">
      <h1 class="tt-panel-title">Panel标题</h1>
      <div class="tt-panel-body">
        Panel内容区
      </div>
    </div>
  </body>
</html>
```

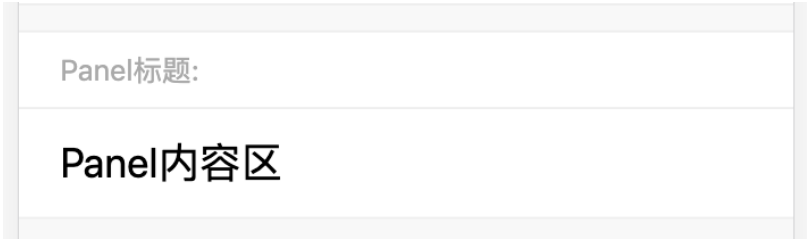
为了简便这个页面里我们只保留内容区，去掉了头部和尾部。在内容区里直接使用 h1 标签作为Panel 的标题，使用 div 作为 Panel 的内容部分的容器。这里没有在外面套一层容器，是想减少一层嵌套，如果遇到什么需求必须加的再加一层即可。下来根据上面的要求，就可以写 Panel 部分的样式了。由于 Panel 是一个通用性比较大的容器，并不是表单才会用到的，所以我们把它的代码放在 /src/content.css 里，在 /src/content.css 文件的末尾追加如下代码：

```
/* 内容分区 -- Panel 面板 */
/* Panel 标题 */
.tt-content .tt-panel-title{
    height: 1.8rem;
    line-height: 1.8rem;
    padding-left: 1rem;
    color: #aaa;
    background: #fff;
    border-top: 1px solid #eee;
    font-size: 14px;
    font-weight: normal;
}
/* Panel 内容区 */
.tt-content .tt-panel-body{
    position: relative;
    margin-bottom: .6rem;
    padding: .6rem 1rem;
    background: #fff;
    overflow: hidden;
    border-top: 1px solid #eee;
    border-bottom: 1px solid #eee;
}
/* 可手动设置内容区是否有内边距 */
.tt-content .tt-panel-body.no-padding{
    padding: 0;
}
```

这里有几点要注意：

1. 我们这里用了后代选择器，而不是子代选择器，是因为Panel不一定直接放在 .tt-content 下。
2. 给内容区设置了“position: relative;”，是考虑当内容区里有需要绝对定位的元素的话，可以以 .tt-panel-body 为基准。
3. .tt-panel-body 默认情况下是有边距的，然后用“.tt-panel-body.no-padding”这个交集选择器来设置不需要内边距的情况。

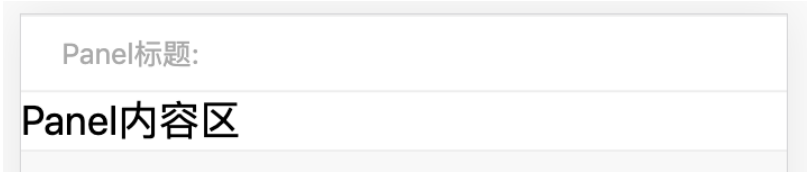
这样，刚才展示的样式就出来了：



当不需要内边距的时候，就可以在 .tt-panel-body 上加一个“no-padding”的类就可以了，如下：

```
<h1 class="tt-panel-title">Panel标题</h1>
<div class="tt-panel-body no-padding">
   Panel内容区
</div>
```

这样设置以后，就变成了如下效果：



没有内边距时放纯文本看起来是比较别扭，但放类似表单元素的时候，就会和谐的多了。这样这个面板 Pane l就开发完了，下来我们开始进入正题，开始表单元素的开发。

基本输入框的开发

在移动端的表单中，输入框的样式比较固定，一般就是每个输入项占用一行，每行内由标签和输入框组成。就像这样：

当我们要实现一个这样的样式时，就需要注意下面这些细节的处理：

1. 每个输入项占用一行，并且每两行中间用边框来区分。
2. 为了使输入项的边框占用整个宽度，外部容器（如现在使用的 Panel）不应该有内边距。
3. 每行的左边应该有个内边距，使内容部分不至于贴边，也可以和容器标题部分对齐。
4. 标签部分有固定宽度，这样后面的输入部分就可以对齐。
5. 没有标签的时候，输入框也可以正常显示。

结合上面的效果图和对细节的要求，我们先把 HTML 结构建立出来：

```
<!-- 基本输入框 -->
<h1 class="tt-panel-title">基本输入框:</h1>
<div class="tt-panel-body no-padding">
  <div class="tt-form-item">
    <label class="tt-form-label">用户名</label>
    <div class="tt-form-body">
      <input class="tt-input" type="text" placeholder="请输入姓名" />
    </div>
  </div>
  <div class="tt-form-item">
    <label class="tt-form-label">密码</label>
    <div class="tt-form-body">
      <input class="tt-input" type="password" placeholder="请输入密码" />
    </div>
  </div>
  <div class="tt-form-item">
    <label class="tt-form-label">日期</label>
    <div class="tt-form-body">
      <input class="tt-input" type="date" placeholder="请选择日期" />
    </div>
  </div>
</div>
```

这段 HTML 里，使用的带有 no-padding类的 Panel 容器。我们在 Panel 里建立了三个基本相同的 tt-form-item 元素，每一个元素就会占用一行。在每一行中，都会分为标签和输入区两部分。这三个 tt-form-item 元素里装了三种不同类型的输入框类型，分别是文本类型、密码类型和日期类型。这些样式需要写在新建的 /src/input.css 里，然后同样在 /src/tuitui-ui.css 里把 /src/input.css 引入进去，下来我们就在 /src/input.css 里写它们的样式。

首先是表单项容器 .tt-form-item，根据前面提的要求我们逐个来实现：

1. 每个容器占一行，使用块级元素就可以实现，而每两行中间要有间隔，我们可以先给每个元素都加一个下边框，然后再用 last-child 选择器把最后一个元素的下边框单独去掉就行了（这个需求也可以通过兄弟选择器来实现，同学们也可以试着写一下）。
2. 每行的左边有个内边距，可以和标题对齐，直接用 padding-left 即可。
3. 标签和输入区域左右显示，标签有固定宽度，还可以去掉，所以这太符合使用弹性布局了。
4. 给每行容器加一个“position: relative;”属性，这样容器里需要绝对定位的元素可以以它为参考系。

所以最后得出的容器的样式就如下所示：

```
/* 表单的单行容器 */
.tt-form-item{
  display: flex;
  position: relative;
  padding-left: 1rem;
  border-bottom: 1px solid #eee;
}
/* 去掉最后一行的下边框 */
.tt-form-item:last-child{
  border-bottom: none;
}
```

接下来是每一行里的标签和输入区的布局，标签是定宽，内容区要求撑满剩余的空间。这样标签和内容区的样式就可以按着下面来设置：

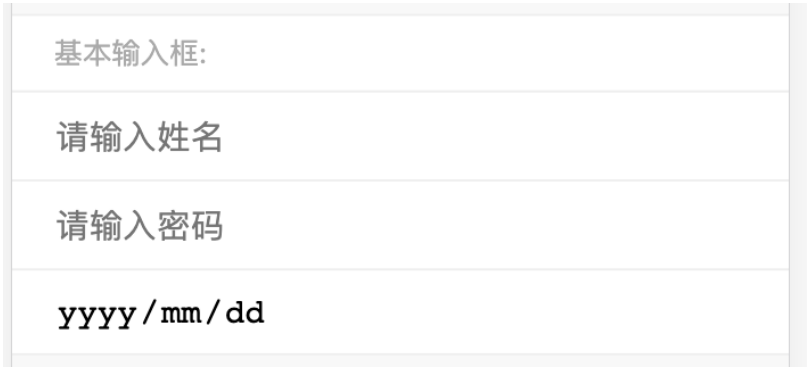
```
/* 表单标签 */
.tt-form-item > .tt-form-label{
    display: block;
    width: 3.5rem;
    font-size: .8rem;
    color: #666;
    height: 2rem;
    line-height: 2rem;
}
/* 表单内容区 */
.tt-form-item > .tt-form-body{
    flex: 1;
}
/* 输入框 */
.tt-form-item > .tt-form-body > .tt-input{
    width: 100%;
    height: 2rem;
    line-height: 2rem;
    font-size: .8rem;
    border:none;
}
```

我们在 input 外面包了一层 tt-form-body ，是因为在一些特殊情况下，比如每行里放多个输入框的时候，这样布局会更方便一点。按着上面的代码，就可以看到最终的效果：

如果我们不想要 label 了，就可以直接在 HTML 中去掉：

```
<!-- 基本输入框 -->
<h1 class="tt-panel-title">基本输入框:</h1>
<div class="tt-panel-body no-padding">
    <div class="tt-form-item">
        <!-- <label class="tt-form-label">用户名</label> -->
        <div class="tt-form-body">
            <input class="tt-input" type="text" placeholder="请输入姓名" />
        </div>
    </div>
    <div class="tt-form-item">
        <!-- <label class="tt-form-label">密码</label> -->
        <div class="tt-form-body">
            <input class="tt-input" type="password" placeholder="请输入密码" />
        </div>
    </div>
    <div class="tt-form-item">
        <!-- <label class="tt-form-label">日期</label> -->
        <div class="tt-form-body">
            <input class="tt-input" type="date" placeholder="请选择日期" />
        </div>
    </div>
</div>
```

这样输入框的样式也不会乱：



在这里还有一个小的地方要注意，input　这个元素比较特殊，它会去继承父元素的字体样式，但会被浏览器给的默认样式覆盖，这些默认样式的优先级要高于继承来的样式。所以我们在设置input元素的字体样式时，需要直接给input设置样式，而不要依靠样式的继承。

高级输入框的开发

在表单输入的时候，我们还需要一些特殊的功能，比如在输入有错误的时候可以进行提示，有些输入框可以通过一个按钮一键清空这些，我们这里就来写下这两种功能的输入框。在实现这些功能的样式的时候，其实就是在输入框的最后加上对应的小图标，其余的逻辑要交给 JS 来完成。下来我们就来加两种小图标试试。
首先是带一键清空功能的图标，通常用一个灰色的小叉号，就像这种效果：

我们可以在基础输入框的HTML上进行改动，只需要在tt-form-body加上一个小叉号的字体图标，然后把字体图标变成上图所示的那种样式。

```
<!-- 带清空按钮的输入框 -->
<h1 class="tt-panel-title">带清空功能的输入框:</h1>
<div class="tt-panel-body no-padding">
    <div class="tt-form-item">
        <label class="tt-form-label">手机号</label>
        <div class="tt-form-body">
            <input class="tt-input" type="text" placeholder="请输入手机号" />
            <i class="fa fa-close tt-input-reset"></i>
        </div>
    </div>
</div>
```

在定位这个 fa-close 图标的时候，要把它放在输入框的右侧，并且在竖直方向上居中。在做这个需求的时候，就可以使用到绝对定位。我们可以按着如下的样式来设置这个图标：

```
/* 表单中的清空按钮 */
.tt-form-item > .tt-form-body > .tt-input-reset{
    position: absolute;
    width: .8rem;
    height: .8rem;
    line-height: .8rem;
    top: 50%;
    margin-top: -.4rem;
    right: 1rem;
    font-size: .6rem;
    background: #aaa;
    color: #fff;
    border-radius: 50%;
}
```

最终这个带清空功能的输入框的样式如下；

> **@ Tips:**
> 有关绝对定位的几点需要注意：
> 1、绝对定位的用法和固定定位很相似，只不过固定定位的参照物是屏幕可视区，而绝对定位的参照物是最近的有定位的祖先元素。
> 2、最近的有定位的祖先元素，这里说的定位可以是相对定位、绝对定位或者固定定位。在HTML中相对定位不会破坏文档流，所以通常会把固定定位和相对定位配合着来使用，但我们要知道固定定位的参考系也可以是绝对定位或固定定位的元素。
> 3、这里使用了一种新的竖直居中的方法，依靠“top:50%;”找到竖直方向的初始点，然后再依靠使用负值的margin把元素调整到竖直居中的位置。这种方式和我们在讲固定定位时讲到的水平竖直居中的方法，都是可以达到要求的。

再下来带错误提示功能的输入框就容易了，在刚才的基础上，只需要换一个图标再更改一下图标的样式即可达成。这里我们就直接贴代码了。

HTML部分：

```
<!-- 带错误提示的输入框 -->
<h1 class="tt-panel-title">带错误提示的输入框:</h1>
<div class="tt-panel-body no-padding">
    <div class="tt-form-item">
        <label class="tt-form-label">邮箱</label>
        <div class="tt-form-body">
            <input class="tt-input" type="text" placeholder="请输入邮箱" />
            <i class="fa fa-exclamation tt-input-warning"></i>
        </div>
    </div>
</div>
```

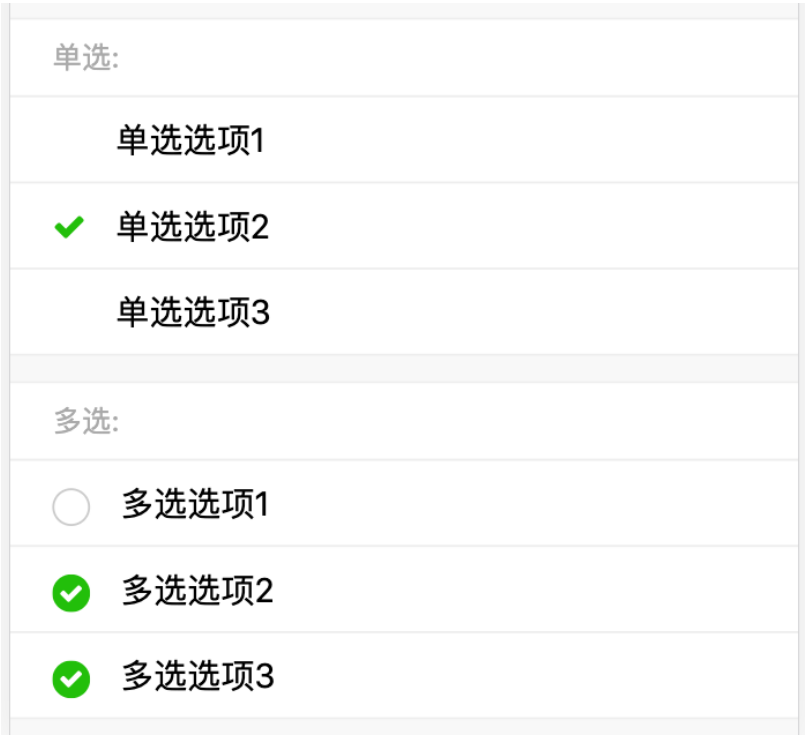
CSS部分：

```
/* 表单中的错误提示按钮 */
.tt-form-item > .tt-form-body > .tt-input-warning{
    position: absolute;
    width: .8rem;
    height: .8rem;
    line-height: .8rem;
    top: 50%;
    margin-top: -.4rem;
    right: 1rem;
    font-size: .6rem;
    border-radius: 50%;
    color: red;
    border: 1px solid red;
}
```

这样带错误提示的表单样式也就出来了：

单选 && 多选

最后要讲的是单选和多选，在移动端中我们很少使用原生的 radio 或者 checkbox 样式，是因为这两种输入方式的点击区域比较小，不太利于移动端的操作。作为替代，通常会把单选和多选转变成下面这种形式：



根据上图，单选和多选里，都是每一个选项占用一行，每行中通过图标来标记是否为选中状态。

下来我们先来实现单选功能，单选中的图标我们用不带外框的勾选标志，根据上面的结构，我们可以这样定义 HTML 结构：

```
<!-- 单选输入 -->
<h1 class="tt-panel-title">单选:</h1>
<div class="tt-panel-body no-padding">
  <div class="tt-form-item">
    <div class="tt-radio">
      <i class="fa fa-check tt-radio-icon"></i>
      <span class="tt-radio-desc">单选选项1</span>
      <input class="tt-radio-input" type="radio">
    </div>
  </div>
  <div class="tt-form-item">
    <div class="tt-radio checked">
      <i class="fa fa-check tt-radio-icon"></i>
      <span class="tt-radio-desc">单选选项2</span>
      <input class="tt-radio-input" type="radio">
    </div>
  </div>
  <div class="tt-form-item">
    <div class="tt-radio">
      <i class="fa fa-check tt-radio-icon"></i>
      <span class="tt-radio-desc">单选选项3</span>
      <input class="tt-radio-input" type="radio">
    </div>
  </div>
</div>
```

在上面的结构中，外层的容器和之前输入框用的都是一样的，只是 .tt-form-item 里面的内容变了一下：

```
<div class="tt-form-item">
  <div class="tt-radio">
    <i class="fa fa-check tt-radio-icon"></i>
    <span class="tt-radio-desc">单选选项1</span>
    <input class="tt-radio-input" type="radio">
  </div>
</div>
```

在这个结构中，给每个单选项放了一个 tt-radio 的容器，可以直接在这个容器上添加 class 来改变选中样式。容器里面有三个元素，分别是区分勾选状态的图标，选项的名称和一个 radio 类型的 input。在实际显示的时候，只需要显示前两个元素。而最后一个 input 我们会用它来记录单选的选择情况，用来给 JS 使用，所以会把它隐藏起来。我们在做单选样式的时候，只需要给 tt-radio 和它的子元素加上样式即可，可以在 /src/input.css 的末尾追加上：

```
/* 自定义单选 */
.tt-form-item > .tt-radio{
  flex: 1;
  font-size: .8rem;
  line-height: 2rem;
}
/* 未选中状态的图标 */
.tt-form-item > .tt-radio > .tt-radio-icon{
  margin-right: .5rem;
  color: #09BB07;
  visibility: hidden;
}
/* 选中状态的图标 */
.tt-form-item > .tt-radio.checked > .tt-radio-icon{
  visibility: visible;
}
/* 隐藏的radio类型的input */
.tt-form-item > .tt-radio > .tt-radio-input{
  position: absolute;
  left: -999rem;
}
```

这段代码中，要注意的地方有：

1. 控制图标的显示时用的是“visibility”属性，它和“display:none;”的不同点是它不会显示在页面上，但会把自己的位

置空出来。这样在不显示的时候，也能保证后面文字的对齐。

2. 单选的图标是通过“.tt-radio.checked”这个交集选择器来判断的，当选中时只需要在tt-radio上添加“checked”这个类。
3. 隐藏最后input元素的时候，使用了一个绝对定位，把这个元素移到了视野以外。这也是一种比较常见的隐藏元素的方式，这种隐藏方式一般不会出错，而 “display:none;” 在一些浏览器上容易出问题，比如会出现显示效果不对或者和 JS 的交互出问题等现象。

这样一个单选的样式就出来了：

接下来我们再实现一下多选，多选和单选很相似，我们只是变化了一下容器的名称和选项的图标。这里我们也直接看代码了。

HTML部分代码：

```
<!-- 多选输入 -->
<h1 class="tt-panel-title">多选:</h1>
<div class="tt-panel-body no-padding">
  <div class="tt-form-item">
    <div class="tt-check">
      <i class="fa fa-check tt-check-icon"></i>
      <span class="tt-check-desc">多选选项1</span>
      <input class="tt-check-input" type="checkbox">
    </div>
  </div>
  <div class="tt-form-item">
    <div class="tt-check checked">
      <i class="fa fa-check tt-check-icon"></i>
      <span class="tt-check-desc">多选选项2</span>
      <input class="tt-check-input" type="checkbox">
    </div>
  </div>
  <div class="tt-form-item">
    <div class="tt-check checked">
      <i class="fa fa-check tt-check-icon"></i>
      <span class="tt-check-desc">多选选项3</span>
      <input class="tt-check-input" type="checkbox">
    </div>
  </div>
</div>
```

CSS部分代码：

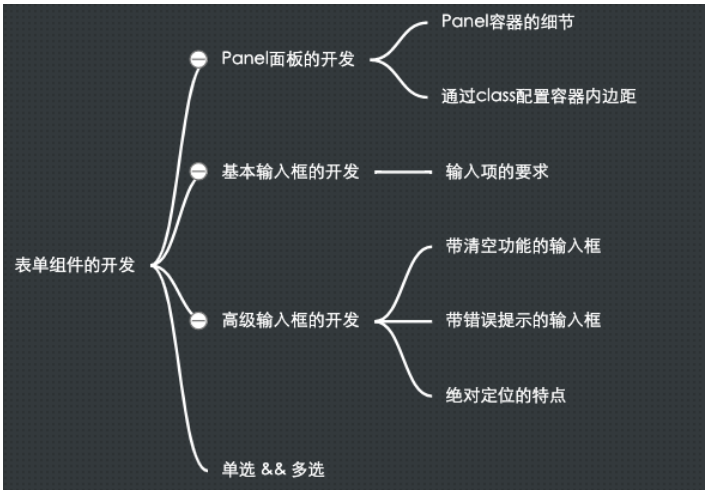
```
/* 自定义多选 */
.tt-form-item > .tt-check{
    flex: 1;
    font-size: .8rem;
    line-height: 2rem;
}
/* 未选中时的图标 */
.tt-form-item > .tt-check > .tt-check-icon{
    margin-right: .5rem;
    border: 1px solid #ccc;
    border-radius: 50%;
    color: rgba(0,0,0,0);
    font-size: .6rem;
    width: .8rem;
    height: .8rem;
    line-height: .8rem;
}
/* 选中时的图标 */
.tt-form-item > .tt-check.checked > .tt-check-icon{
    background: #09BB07;
    color: #fff;
    border-color: rgba(0,0,0,0)
}
/* 隐藏的check类型的input */
.tt-form-item > .tt-check > .tt-check-input{
    position: absolute;
    left: -999rem;
}
```

在多选里，我们用字体图标的颜色来控制它的勾选状态，在未选中的时候给字体图标透明的字体，在选中的时候再把字体颜色设置回来。其他的就都和单选是类似的了。最后多选的效果就是这样：

结语

这一节因为带着 Panel 的开发，所以内容又比较多。这一节在讲表单的开发中，用到了绝对定位的知识，同学们要记住绝对定位的那几个特点，并且要结合着我们之前讲的固定定位的知识，把固定定位的其他特性也试一试。

这一节的内容结构如下：



到这里我们这一节的内容就结束了，同学们可以访问【表单元素在线预览】来查看这一节的演示效果。下一节将开发我们这一章最后一个内容，就是按钮的开发。我们这一节的内容就到这。

18 布局组件的开发
20 按钮的设计与开发