

20 按钮的设计与开发

更新时间：2019-07-15 13:27:21



今天应做的事没有做，明天再早也是耽误了。

——裴斯泰洛齐

在 UI 设计中，按钮是用户和系统进行沟通最重要的工具，我们这一节就来做一下按钮的设计和开发。在一个系统里，不同场景下使用的按钮，在设计上也是不一样的。比如不同功能的按钮样式就会不一样，主要操作的就尽可能的明显一点，次要操作尽量不抢夺用户的注意力，而一些危险操作就会使用警示色。而不同的地方使用的按钮大小也不一样，像提交订单一类比较重要的按钮就会做的比较大，利于提高转化率；而像加关注、点赞这类非主要的按钮，都会设计的比较小。接下来我们就来进行对按钮的开发。

基础按钮

在开发之前，先把我们这一节的文件建立出来，首先新建一个 `/demo/button.html`，在里面放上页面模板：

```
<!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>按钮的设计与开发</title>
</head>
<body>
<div class="tt-content">
<!-- 标准按钮 -->
<h2 class="tt-panel-title">标准按钮</h2>
<div class="tt-panel-body">
<a class="tt-btn">默认按钮</a>
</div>
</div>
</body>
</html>
```

这段代码里就在 `tt-panel-body` 里加了一个 `a` 标签，给他起名叫 `tt-btn`，这就是一个最基本的按钮。我这里用了 `a` 标签，是可以直接用作超链接，也可以使用 `span` 或者 `button` 等标签代替。我们将要实现样式如下：



对于标准按钮我们要有如下要求：

1. 默认为块级元素，自己单独占一行空间，且水平居中。
2. 按钮内文字水平竖直居中。
3. 默认按钮可以清淡一点，可以适用于各种场景。
4. 单行内容超长时自动隐藏，并显示省略号。

按着刚才的样式和这几条要求，我们可以对它添加样式了。下面我们新建一个 `/src/button.css`，然后在 `/src/tuitui-ui.css` 里把这个文件引入进去。

在 `/src/tuitui-ui.css` 的最后追加：

```
/* 按钮的样式 */
@import './button.css';
```

然后在 `/src/tuitui-ui.css` 里填写如下样式：

```
/*
 * @Author: Rosen
 * @Date: 2019-07-06 10:55:19
 * @Last Modified by: Rosen
 * @Last Modified time: 2019-07-07 10:11:25
 */

/* 按钮默认样式 */
.tt-btn{
  display: block;
  position: relative;
  width: 10rem;
  margin: 0 auto;
  padding: .5rem 0;
  box-sizing: border-box;
  border: 1px solid #ccc;
  font-size: .7rem;
  text-align: center;
  color: #000000;
  border-radius: 5px;
  overflow: hidden;
  white-space: nowrap;
  text-overflow: ellipsis;
  user-select: none;
}
```

使用这个样式后，基本按钮的样式就会和刚才的示意图一样了。在这些样式中最后一条的作用要单独提一下，“`user-select:none;`” 表示用户不能对按钮的内容进行选中，防止因用户按住按钮太久而导致出现文本选择的选项。

按钮的种类

刚才完成的默认按钮可以用在大部分场景里，但是这个按钮并没有特定的含义。从单一的功能上看这个按钮怎么用都行，但是从整个应用来说，不同类型的操作都用同一种按钮肯定是容易引起歧义的。所以我们要来做几种不同用途的按钮，用来区分不同的应用场景。

一、主操作按钮

一个页面都是有特定的功能，也是有特定的目的。比如商品的详情页是展示商品细节的，但这个页面的目的是引导用户下单；订单确认页展示的是订单内容，但这个页面的目的是引导用户付款。这些页面里，都有一个非常强的目的性。为了引导用户进行下一步操作，这些页面中就会把引导用的按钮设计的更显眼，更方便用户点击。这就是页面的主操作按钮，我们这里把主操作按钮设计成下面的样式：



在默认按钮的基础上，加深了按钮的背景色，让其更加显眼。同时为了和背景色匹配，把字体颜色也改成了白色。这个主操作按钮我们可以在默认按钮的基础上进行样式覆盖，这样经过简单的变更就可以实现新的样式。下来我们用 `.tt-btn-primary` 来覆盖 `.tt-btn` 的样式。

HTML:

```
<a class="tt-btn tt-btn-primary">主操作按钮</a>
```

这种使用另外的 `class` 进行的样式覆盖，也有点像我们在写逻辑语言时使用的参数。把 `tt-btn` 作为基础，然后通过追加各种各样的附加类来定制按钮的样式。下来我们改下 `.tt-btn-primary` 的样式：

```
/* 定制主操作按钮样式 */
.tt-btn.tt-btn-primary{
  background-color: #1AAD19;
  color: #fff;
}
```

这里我们用交集选择器来定义这个样式覆盖，这样这个追加的样式只会对 `.tt-btn` 生效，不会影响其他的元素。

二、带警示功能的按钮

在页面中，还有一类操作是比较敏感的，不希望用户点击或者需要谨慎点击的。这些操作不需要很强的引导性，反而需要对用户进行警示，也就是我们这里说的带警示功能的按钮。在设计上，通常使用红色作为警示色，这里我们把警示按钮设计成下面这样：



定制这样的按钮做法也很简单，就是在默认按钮的基础上修改字体颜色和背景色。我们用 `"tt-btn-warning"` 来标记警示类型按钮的样式。它的HTML如下：

```
<a class="tt-btn tt-btn-warning">警示按钮</a>
```

而样式上的做法和前面讲的实现主操作按钮的方法非常类似，同样使用交集选择器标记按钮的样式：

```
/* 定制警示类按钮样式 */
.tt-btn.tt-btn-warning{
  background-color: #E64340;
  color: #fff;
}
```

这样就可以实现出警示按钮的样式。

三、不可用按钮

在页面中，还有一类比较特殊的按钮，就是不可用按钮。这类按钮通常是有约束条件的，在某种情况下才可用。而在不满足条件的时候，就是不可用状态。不可用按钮通常用对比度较弱的配色来表示，目的是让用户忽略这个按钮的存在。我们这里用下面这个样式来表示不可用按钮：



这个按钮在默认按钮的基础上把背景和字体都变成了对比度不强的灰色，另外为了让它更加的不明显，还给它加上了一定的透明度。我们“tt-btn-disabled”来标记不可用按钮，它的 HTML 如下：

```
<a class="tt-btn tt-btn-disabled">不可用按钮</a>
```

而样式和前面也都类似，只不过多加了一条调整透明度的样式：

```
/* 定制不可用按钮的样式 */
.tt-btn.tt-btn-disabled{
  background-color: #F7F7F7;
  color: #999;
  opacity: .6;
}
```

这样这个按钮的样式就可以实现出来了。

@ Tips:

在一些设计中，有些是非常不希望用户点击但又不得不加的按钮，也会使用这种不可用按钮。比如取消订单、删除应用和某些确认操作里的取消按钮等，通常使用这种手段（也可以说是技巧）来阻碍用户进行对产品不利的操作。

最终，各类按钮集合在一起的效果就是这样：



定制按钮的大小

除了场景的区分外，在不同位置使用的按钮样式也是有区别的。比如在一些需要强引导的页面中，会使用比标准更明显一些的按钮，有时为了让用户随时能进行操作还会把按钮固定在页面的底部，不参与页面的滚动。而在一些小的区域里，比如单行的区域，则会使用一些小号的行内按钮。我们接下来就要在默认按钮的基础上，改变一下按钮的大小。

一、大按钮

刚才提到了，大按钮通常用于强引导，这里我们把它设计成宽度占满全屏，样式如下：



我们把四种类型的按钮都放了出来，大按钮和标准按钮有如下区别：

1. 宽度占满全屏。
2. 字体字号变大一些。
3. 左右的边框去掉。
4. 按钮的圆角去掉。

我们用“**tt-btn-large**”来标记大按钮的样式，这个样式和刚才定制按钮样式的类是从不同维度来定义按钮的，所以可以和用来区分按钮类型的类同时使用，这样按钮定制起来就更灵活了。下面是 **HTML** 的内容：

```
<!-- 大按钮 -->
<h2 class="tt-panel-title">大按钮</h2>
<div class="tt-panel-body no-padding">
  <br/>
  <a class="tt-btn tt-btn-large">大按钮</a>
  <br/>
  <a class="tt-btn tt-btn-primary tt-btn-large">大按钮</a>
  <br/>
  <a class="tt-btn tt-btn-warning tt-btn-large">大按钮</a>
  <br/>
  <a class="tt-btn tt-btn-disabled tt-btn-large">大按钮</a>
  <br/>
</div>
```

这里为了能撑满整个页面，我们给 **Panel** 容器加上了“**no-padding**”的类。这里在各种类型的按钮上都加上了一个“**tt-btn-large**”的类，我们通过指定这个类的样式来修改样式的大小。根据刚才的改动情况，很容易就能实现出大按钮的样式了：

```
/* 大按钮 */
.tt-btn.tt-btn-large{
  width: 100%;
  font-size: .9rem;
  border-left: none;
  border-right: none;
  border-radius: 0;
}
```

这样大按钮的样式就可以实现出来了。

@ Tips:

在做全宽的按钮时使用了“width: 100%;”，但这个样式对容器是有要求的，要求它的参考容器也是全宽的才可以。在 CSS3 中还有一个单位是 **vw**，这个单位也是一个相对单位，它的参考系是屏幕的宽度，像这种需要占满全屏的需求就可以直接使用“width: 100vw;”，这样就可以不用考虑外层容器的情况了。和 **vw** 对应的还有个 **vh** 单位，它是相对于屏幕高度的。但是！！！这里要注意，这两个单位在安卓 4.3 及以前的机器上是不支持的，如果用的话一定要确定自己产品中低版本安卓的占比。根据谷歌在 2018 年 8 月发布的统计数据来看，安卓 4.3 及之前版本还占安卓市场将近 4% 的份额。这里也推荐一个查询各种样式或者单位兼容性情况的网站：caniuse.com

二、小按钮

下来还有一个小按钮的开发，小按钮通常是用作比较小的区域，通常以行内元素出现。所以在制作小按钮的时候，除了尺寸的变化外，还要把小按钮改成行内元素，这里我们可以使用行内块。我们用“tt-btn-small”来标记小按钮的样式，下面是 HMTL:

```
<!-- 小按钮 -->
<h2 class="tt-panel-title">小按钮</h2>
<div class="tt-panel-body">
  <a class="tt-btn tt-btn-small">小按钮</a>
  <a class="tt-btn tt-btn-primary tt-btn-small">小按钮</a>
  <a class="tt-btn tt-btn-warning tt-btn-small">小按钮</a>
  <a class="tt-btn tt-btn-disabled tt-btn-small">小按钮</a>
</div>
```

根据刚才的描述，小按钮的样式也比较简单:

```
/* 小按钮 */
.tt-btn.tt-btn-small{
  display: inline-block;
  width: 3rem;
  padding: .2rem 0;
  font-size: .7rem;
}
```

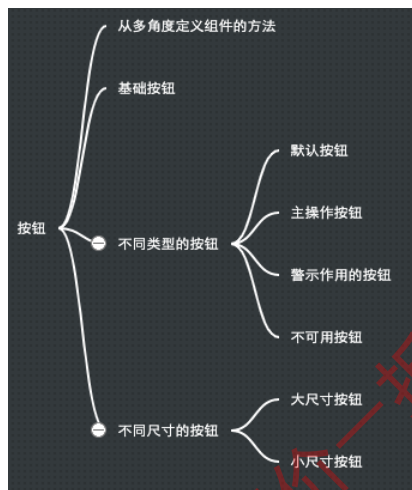
最后的实现效果就是一排小按钮了:



到这里我们所有的按钮样式就实现完成了。

结语

相比前面的章节，这一节的内容就比较简单了。希望同学们学会使用组合的方式从不同维度定义一个组件样式的方法，这种定义方式在设计通用组件时非常实用。这一节内容结果如下：



我们对按钮的设计和开发也就完成了，同学们可以访问【[按钮在线预览](#)】来查看这一节的演示效果。到这里我们布局和基础组件这部分的内容也都完成了，下面的一节将对这一章的内容做个总结。