27 Modal模态框样式的设计与开发

更新时间：2019-07-29 10:34:29

> 合理安排时间，就等于节约时间。
>
> ——培根

我们这一节是要实现模态框的样式。模态框指的是覆盖在页面原有内容上的一个新的窗口，用来展示一些独立于原有内容的信息。模态框比较常用的样式就是用蒙版遮住原有内容，再在蒙版上展示一个非全屏的新窗口。我们这次要实现的模态框有两种，下来先展示一下这两种模态框的效果。

第一种是**基础模态框**的样式如下，这种模态框通常分为内容区和操作区两部分。这类模态框通常用作一些操作的确认。

第二种模态框就像下面这种**海报形式**的，这类模态框通常用作引导。可以作为一些活动页面的入口，或者引导关注公众号等。这种模态框里的内容比较复杂，一般会直接使用一整张图片来展示。一般这类模态框也会配上一个独立的关闭按钮。

下来我们开始分析下两种模态框有哪些需求。

模态框的需求

一、基础模态框的需求

对于基础模态框，我们要有如下要求：

1. 模态框要有半透明的背景遮盖住整个页面，保证用户在模态框中操作的时候不受原有内容的影响。
2. 模态框固定在页面上，在水平方向上居中。
3. 在竖直方向上比居中位置稍微靠上，可以让中心点距离页面上边缘 45%。因为屏幕上边一般都有头部，所以稍微偏上一点在视觉上会更舒服。
4. 模态框的内容区可以随着内容的多少来调整高度，当文字太多的时候使用滚动。
5. 内容区的文字和模态框的边界要留出一定的距离。
6. 操作区里的按钮撑开整个模态框的宽度，有多个按钮时多个按钮平分模态框宽度，且中间要有分隔线。
7. 可以通过一个类来控制模态框是否显示。

二、海报形式模态框的需求

对于海报形式模态框的要求，其实会更简单一些，它只是在基础模态框的基础上做了些变动。对海报形式的模态框有以下需求：

1. 图片占据整个内容区，不需要和模态框边缘留空间。
2. 取消操作区。
3. 在弹窗的下方添加关闭按钮。

了解这两种模态框的需求后，就可以进行设计和开发了。

模态框的设计与开发

一、文件的建立

第一步，我们先把需要的基础文件建立出来，最基本的是 HTML 文件 /demo/modal.html:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <meta name="viewport" content="initial-scale=1, width=device-width, maximum-scale=1, user-scalable=no">
    <link rel="stylesheet" href="../src/tuitui-ui.css">
    <title>推推UI-模态框</title>
  </head>
  <body>
    <div class="tt-header">
      <div class="left"><i class="fa fa-chevron-left"></i></div>
      <div class="title">推推UI-模态框</div>
      <div class="right"><i class="fa fa-ellipsis-h"></i></div>
    </div>
    <div class="tt-navbar">
      <a class="navbar-item">
        <i class="fa fa-home icon"></i>
        <span class="name">首页</span>
      </a>
      <a class="navbar-item active">
        <i class="fa fa-list icon"></i>
        <span class="name">分类</span>
      </a>
      <a class="navbar-item">
        <i class="fa fa-search icon"></i>
        <span class="name">发现</span>
      </a>
      <a class="navbar-item">
        <i class="fa fa-user-o icon"></i>
        <span class="name">我的</span>
      </a>
    </div>
    <div class="tt-content">
      <p>内容区</p>
    </div>
    <!-- 模态框 -->
    <div class="tt-modal">
      <!-- ... -->
    </div>
  </body>
</html>
```

这个文件里，我们把标题栏、内容区和导航栏都添加上了，是为了预览的时候可以看到模态框和其他部分搭配的效果。HTML 文件中模态框 .tt-modal 排在了内容区 .tt-content 的后面，我们要实现的模态框就会放在这个容器里。

接下来是 CSS 文件 /src/modal.css：

```
/*
 * @Author: Rosen
 * @Date: 2019-07-24 22:12:59
 * @Last Modified by: Rosen
 * @Last Modified time: 2019-07-27 15:56:04
 */

/* 模态框 */
.tt-modal{

}
```

最后在 tuitui-ui.css 中引入刚才新建好的 /src/modal.css 文件，在 /src/tuitui-ui.css 的最后添加：

```
/* 模态框组件的样式 */
@import './modal.css';
```

这样，文件就建好了。

二、基础模态框的设计与开发

我们先来分析下模态框的结构。根据需求，模态框里大体上是分为两部分，一部分是半透明遮罩，另外一部分就是模态框的窗口部分。而模态框里面又分为内容区和操作区，内容区就是一些文本性质的内容，而操作区可以放按钮。根据这样的结构，我们就可以按着下面的结构来建立HTML：

```
<div class="tt-modal show">
  <div class="tt-mask"></div>
  <div class="tt-modal-wrap">
    <div class="tt-modal-body">
      <p>您的会员即将到期，请及时续费，以免影响您的权益。</p>
    </div>
    <div class="tt-modal-footer">
      <a class="tt-btn">下次再说</a>
      <a class="tt-btn">立即续费</a>
    </div>
  </div>
</div>
```

下来我们一边分析，一边实现这个模态框。
首先是最外层的容器，在 tt-modal 上，我们加上了“show”这个 class 来控制模态框是否显示。所以默认情况下要隐藏模态框，当有 show 这个类的时候再把模态框显示出来。所以容器的样式就是：

```
/* 模态框 */
.tt-modal{
  display: none;
}
/* 控制模态框的显示 */
.tt-modal.show{
  display: block;
}
```

然后是下一层蒙版和模态框窗口的实现。蒙版层可以直接使用在第三章实现过的通用蒙版，这里只需要在 tt-modal 里加入一个 class 为“tt-mask”的 div 元素就可以了。

接下来是窗口的实现，窗口的定位就决定了模态框显示的位置，按着需求我们希望它中心在距顶部 45% 的位置。对于这种定位要求，我们之前用过的方法只有一种比较接近。就是在做绝对定位元素垂直居中的时候，可以通过设置“top: 45%;”值来定位元素的起始位置，再通过负的 margin-top 值来把整个容器向上移动容器高度的一半，从而达到需求中的要求。这种方法已经很接近了，但遗憾的是这种方式下我们要知道容器的确切高度，才能设置正确的 margin-top 值，而我们这里有另外一条要求就是弹窗的高度要随着内容自动调整，所以这种方法就没法用了。

下来我们要介绍一个新的东西，就是 CSS3 中的 transform 属性。

> **@ Tips：**
> transform 属性是 CSS3 中用来对盒模型做变化的，可以用它对元素的形状、大小、位置和旋转角度等进行更改。 transform 可以取的值有很多，这里由于篇幅限制只介绍几种常用的取值，同学们下去可以去w3school上了解下其他的转换方式。
>
> **translate(x,y)**，这个属性值是用来改变元素的位置，两个参数分别是在 x 轴和在 y 轴上的位移。其中 x 和 y 的值就是一般的长度值，可以使用 px、rem 和百分比等单位的值，如：
>
> ```
> transform: translate(30px, 50px);
> ```
>
> 注：如果在 translate 中只有一个值，表示的是在 x 轴上的位移。
>
> **scale(x,y)**，这个属性值是用来调整元素大小的。x 和 y 的取值是数字，分别表示在 x 轴和 y 轴上缩放的倍数，如果取值小于 1 就是缩小，大于 1 就是放大，如：

```
transform: scale(1.2, 1.5);
```

注：如果在 scale 中只有一个值，表示的是在 x 轴和 y 轴上都进行缩放这个参数指定的倍数。

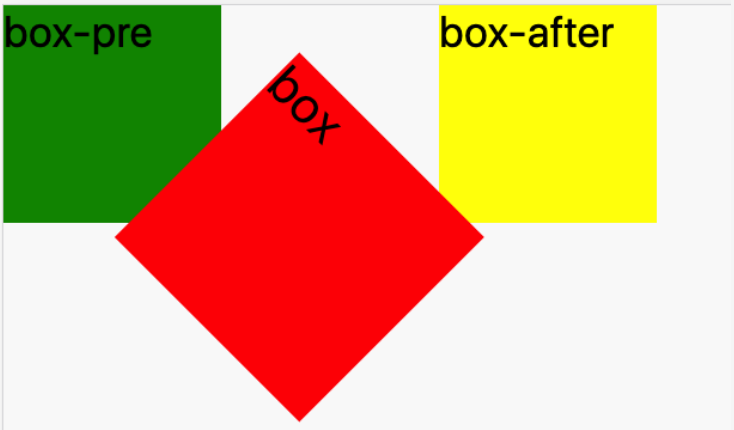
rotate(angle)，这个属性值用来调整元素的旋转角度。angle 的值代表的是顺时针旋转的角度，单位是 deg，如：

```
transform: rotate(45deg);
```

这里就介绍这三种最常见的用法。在刚使用 transform 属性的时候可能会有点不习惯，个人觉得应该把各种转换方法拿出来作为单独的属性使用会更方便。

在使用 transform 属性的时候，要注意下面几点：

1. transform 属性只改变元素的显示效果，不会改变其在文档流中的位置，也就是通过 transform 做的转变不会影响到其他元素的布局。如下图所示，红色的 box 盒子原本是在 box-pre 元素和 box-after 元素中间。当给 box 做转换后，显示的位置变了，但 box-pre 和 box-after 两个盒子中间还留着 box 的位置。



   基于这个特性，当有需要做过渡效果或者动画的时候，尽量使用 transfrom 来实现，能有效的减少页面的重排。
2. 可以同时使用多种转变效果，将多个转化属性用空格分开即可。但要注意 transform 里多有多个转变的时候是从左往右依次执行，如果同样的几个转化方式排列的顺序不同，会导致最终的效果也不同。同时使用多种转换的方法如下：

```
transform: rotate(45deg) translate(30px, 50px) scale(1.2);
```

3. 如果在 transform 的转化属性中使用百分比，那这个百分比的参照物就是当前盒子的宽度或高度。

有了 transform 属性，刚才的问题就能解决了。前面讲的那种方法里，margin-top 的值是以外层 .tt-modal 的高度为参照物的，所以没法使用百分比。但是现在有了 translate，这个属性做位移的时候参照物是盒子本身的高度，这样通过向上移动盒子自身高度的 50%，就可以把盒子的中心移到 top 值设定的地方。这样无论模态框窗口有多高，都可以保证窗口的中心距页面上边缘 45% 的需求。所以关于窗口容器这部分代码如下：

```
/* 模态框窗口容器 */
.tt-modal .tt-modal-wrap{
    position: absolute;
    width: 75%;
    max-width: 480px;
    top: 45%;
    transform: translateY(-50%);
    left: 0;
    right: 0;
    margin: auto;
    background: #fff;
    border-radius: .4rem;
    z-index: 301;
}
```

这里面我们使用的就是“transform: translateY(-50%);”这条属性对窗口容器做的位移。这里还有一点要注意，我们使用了“max-width: 480px;”限制了模态框窗口的最大宽度，是防止在大屏幕下模态框过大，所以采用了页面内容区最大宽度 640px 的 75% 作为窗口的最大宽度。

最后就是窗口容器里面的样式了。里面分为内容区和操作区，分别用“tt-modal-body”和“tt-modal-footer”这两个 class 来表示这两部分，其中操作区为了在水平方向放多个按钮，会使用弹性布局。这两个容器的样式就可以按如下方式实现：

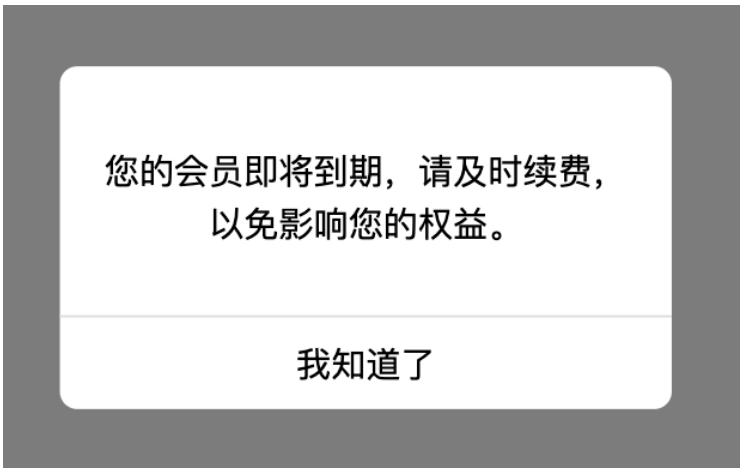
```
/* 模态框内容区 */
.tt-modal .tt-modal-body{
    padding: 1.8rem .8rem 1.5rem;
    text-align: center;
    font-size: .8rem;
    line-height: 1.2rem;
    overflow: hidden;
}
/* 模态框尾部 */
.tt-modal .tt-modal-footer{
    display: flex;
    border-top: 1px solid #ddd;
}
```

最后一步就是微调一下按钮的样式了，这里使用了前面实现过的 tt-btn 按钮样式，但之前实现过的样式在这里不是完全合适，所以我们对它进行一些调整，代码如下：

```
/* 模态框尾部里按钮的样式 */
.tt-modal .tt-modal-footer .tt-btn{
    border: none;
    border-radius: 0;
    width: 100%;
    font-size: .8rem;
}
/* 模态框尾部中的按钮加上分隔 */
.tt-modal .tt-modal-footer .tt-btn + .tt-btn{
    border-left: 1px solid #ddd;
}
```

这样整个基础模态框的样式就完成了，最终的效果如下：

当然基本模态框也可以只留一个按钮，用作信息通知类型的弹窗。就像下面这样：



三、海报样式模态框的设计与开发

接下来我们实现海报样式的模态框。有了刚才的基础，再实现这个就变得容易了。我们先来定义下 HTML 结构：

```
<div class="tt-modal show">
    <div class="tt-mask"></div>
    <div class="tt-modal-wrap">
        <div class="tt-modal-body no-padding">
            <img class="tt-modal-img" src="./img/modal-test.jpg" alt="">
        </div>
        <i class="fa fa-close tt-modal-close"></i>
    </div>
</div>
```

这个 HTML 结构里用了下面这张引导关注公众号的广告图作为测试图片，同学们可以自己下载：

这个 HTML 结构和刚才基础模态框很相似，只有以下三处的不同：

1. 给内容区容器加上了“no-padding”这个 class 来消除内容区的内边距，以便让图片撑满内容区。
2. 内容区的文本换成了一张图片。
3. 在窗口容器的最后添加了一个关闭按钮。

根据前面的需求，我们只需要调整内容区的内边距、图片的大小和关闭按钮的样式就可以了。代码如下：

```
/* 控制模态框内容区的内边距 */
.tt-modal .tt-modal-body.no-padding{
    padding: 0;
}
/* 图片形式的模态框样式 */
.tt-modal .tt-modal-body .tt-modal-img{
    display: block;
    width: 100%;
    border-radius: .3rem;
}
/* 纯图片模态框里的关闭按钮 */
.tt-modal .tt-modal-close{
    position: absolute;
    left: 0;
    right: 0;
    width: 1.3rem;
    line-height: 1.3rem;
    margin: auto;
    bottom: -3rem;
    text-align: center;
    font-size: .8rem;
    font-weight: 100;
    color: #eee;
    border: 1px solid #eee;
    border-radius: 50%;
}
```

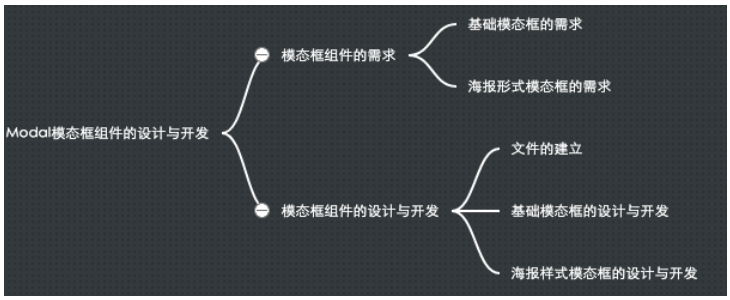
这里要注意下，关闭按钮的位置是在外层容器边缘以外的，所以千万不要把外层 tt-modal-wrap 元素设置成“overflow: hidden;”，那样关闭按钮就会被隐藏掉了。

最后，为了照顾强迫症患者们，我们再把代码位置做个微调，把刚才这个样式的前两组样式插入到“.tt-modal .tt-modal-body”样式的后面。这样海报形式的模态框样式也就出来了：

到这我们这一节的任务就算完成了。

结语

最后总结下这一节的内容。我们这一节完成了两种用途的模态弹窗效果，这一节中最关键的知识点就是对 transform 的介绍和应用，同学们一定记住 transform 属性的那几个特性。这一节的结构如下：



我们这一节的内容到这，同学们可以访问【模态框组件在线预览】来查看这一节的演示效果，下一节将进行加载组件的开发。