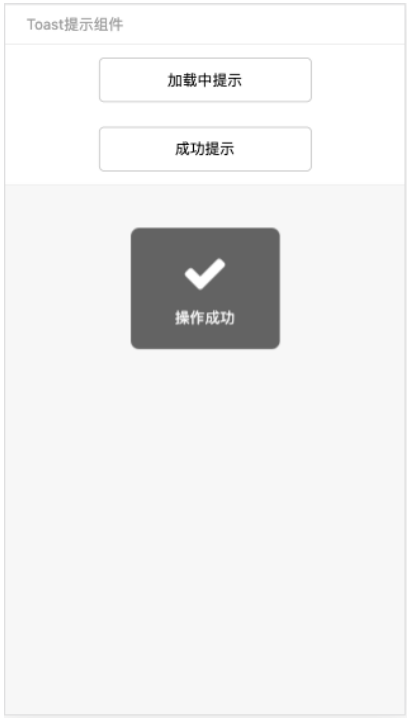
29 Toast提示工具的样式与开发

更新时间：2019-08-08 14:48:07

> 谁和我一样用功，谁就会和我一样成功。
>
> ——莫扎特

这次要实现的是 Toast 提示工具组件，这个组件会使用比较小的一个区域来提示信息，过几秒之后这个提示信息又会自己消失，它的样式可以像这样：



中间操作成功部分的提示就是我们要实现的 Toast 提示工具。这个组件比较特殊的是它的名字，Toast 本意是面包片，看起来和我们要做的这个组件一点关系都没有。但这个组件出现时的效果是弹出来的，很像吐司面包片在面包机刚烤好时弹出来的那个动作，于是就根据组件出现的方式起了“Toast”这个名字。总感觉这个名字起的有点随意，但也没想到更好的名字来给它命名。

下来我们也是先来聊一下这个组件的需求。

Toast提示组件的需求

我们对 Toast 组件有如下要求：

1. 弹窗在水平方向居中，竖直方向上在中间偏上的位置。
2. 弹窗包含一个图标和一个提示信息，且背景半透明。
3. 在弹出时，有向上弹出的入场动画，整个弹窗透明度逐渐升高。
4. 在消失时，有向下滑动的离场动画，整个弹窗透明度逐渐降低。

这个 toast 组件的静态样式是很简单了，容器的做法可以参考 Modal 模态框的做法，里面的图标和文字的布局又和上一节加载提示组件很相似。所以这个组件的难点就在入场和出场动画了。下来我们进入开发环节，来重点设计一下这个组件的入场和出场的动画效果。

Toast提示组件的设计与开发

在设计和开发这部分中，除了像前面章节一样要介绍文件的建立和设计与开发过程外，多加了一个效果演示的内容。这是因为这一节要实现的组件是动态的，要观察它的入场和出场的动画，所以不得不借助 JS 语言来触发入场和出场的效果。我们下来开始这个组件的设计和开发。

一、文件的建立

最开始还是要建立文件，我们先建立 HTML 文件：/demo/toast.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-Toast提示组件</title>
  </head>
  <body>
    <div class="tt-content">
      <h1 class="tt-panel-title">Toast提示组件</h1>
      <div class="tt-panel-body">
        <a class="tt-btn btn-primary" id="js-show-toast-loading">加载中提示</a>
        <br>
        <a class="tt-btn btn-primary" id="js-show-toast-success">成功提示</a>
      </div>
    </div>
    <div class="tt-toast" id="js-toast-loading">
      <i class="fa fa-spinner fa-spin tt-toast-icon"></i>
      <p class="tt-toast-info">操作进行中</p>
    </div>
    <div class="tt-toast" id="js-toast-success">
      <i class="fa fa-check tt-toast-icon"></i>
      <p class="tt-toast-info">操作成功</p>
    </div>
  </body>
</html>
```

这个结构里，需要注意下面几点：

1. 在内容区里建立了两个按钮，预留后面触发 Toast 组件的效果。
2. 我们把 Toast 组件也放在了内容区 tt-content 的后面，这是因为把 Toast 弹出提示也算作了蒙版层的内容，只不过没有加遮罩屏幕的蒙版。
3. 我们建立了两个 Toast 组件的 HTML，分别演示加载中和操作成功这两个样式的 Toast。这个加载中的状态也是上一节中我们提到过的，用来对某些操作过程进行提示。

接下来是空的 CSS 文件 /src/toast.css：

```
/*
 * @Author: Rosen
 * @Date: 2019-07-31 22:24:22
 * @Last Modified by: Rosen
 * @Last Modified time: 2019-07-31 22:25:02
 */

/* Toast提示工具 */
.tt-toast{

}
```

最后，把新建的 toast.css 文件引入到 /src/tuitui-ui.css 中，在 /src/tuitui-ui.css 文件的最后追加：

```
/* Toast弹出提示组件 */
@import './toast.css';
```

这样，基础的文件就建好了。

二、Toast提示组件的设计与开发

Toast 组件的样式不算麻烦，根据前面的开发经验，我们可以先来看代码。

```
/* Toast提示工具 */
.tt-toast{
    position: fixed;
    width: 7rem;
    top: 45%;
    transform: translateY(-40%);
    left: 0;
    right: 0;
    margin: auto;
    padding: 1rem 0;
    opacity: 0;
    color: #fff;
    text-align: center;
    background: rgba(0, 0, 0, .6);
    border-radius: .4rem;
    transition: transform .3s, opacity .3s;
    z-index: 301;
}
/* 显示Toast组件 */
.tt-toast.show{
    opacity: 1;
    transform: translateY(-50%);
}
/* Toast组件中的图标 */
.tt-toast > .tt-toast-icon{
    font-size: 2.2rem;
}
/* Toast组件中的文本部分 */
.tt-toast > .tt-toast-info{
    margin-top: .5rem;
    font-size: .7rem;
}
```

Toast 组件的样式通过这四组样式就可以实现了，这个样式里的技巧都是我们之前用过的。第 1、3、4 组样式用来给Toast 添加静态样式，然后通过第 2 组 .tt-toast.show 的样式来控制组件是否显示。这里和模态框组件唯一不同的就是这里的显示和隐藏使用的 opacity 属性，而没有用 display 属性。这是因为如果使用“display: none;”属性，当“show“这个 class 一去掉，组件会立即消失，出场动画就没有办法实现了。所以这个组件在使用的时候会分四步进行：

1. 需要显示的时候会使用 JS 动态的把组件添加到文档中。
2. 使用添加“show”这个 class，从而达到入场的效果。
3. Toast 组件需要消失的时候，去掉“show”这个 class，从而达到出场的效果。
4. 等组件完全消失后，再使用 JS 把整个 Toast 的 Dom 移除掉。

这样做是因为 Toast 是一个一次性的组件，不使用的时候不需要让它停留在文档中。这样做还能避免当元素是“opacity: 0;”的时候会挡住底层内容区的操作。

三、Toast组件的效果演示

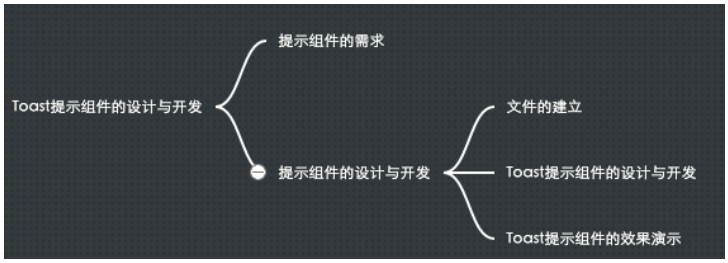
最后我们来处理下这个组件的入场和出场的控制，这里我们只处理入场和出场两个部分。想要实现 Toast 组的入场和出场的演示，我们只需要使用 JS 控制 Toast 组件中“show”这个 class。在一开始的 HTML 中我们创建了两个按钮和两个Toast组件，我们下来用这两个按钮分别控制两个 Toast，可以使用下面的 JS 代码：

```
<script>
  window.onload = ()=>{
    // 显示加载中的Toast
    document.querySelector('#js-show-toast-loading').onclick = (e) => {
      let toastEle = document.querySelector('#js-toast-loading');
      toastEle.classList.add('show');
      setTimeout(()=> {
        toastEle.classList.remove('show');
      }, 2e3);
    };
    // 显示操作成功的Toast
    document.querySelector('#js-show-toast-success').onclick = (e) => {
      let toastEle = document.querySelector('#js-toast-success');
      toastEle.classList.add('show');
      setTimeout(()=> {
        toastEle.classList.remove('show');
      }, 2e3);
    };
  };
</script>
```

这段 JS 代码做的就是当点击按钮的时候，在对应的 Toast 组件上添加“show”这个 class。等到 2 秒过后再把“show”这个 class 移除。这样做就能实现效果的演示了，实际使用的时候要考虑的需要更仔细，比如不允许同时出现两个 Toast 组件等。组件演示的部分就简单介绍这些，最终的效果如下：

结语

因为有了前面章节的基础，我们这一节内容讲的比较简单了，也没有太多的新知识。这一节要注意的一个是做入场和出场动画的时候使不能用 display 来控制元素的显示效果，另外一个就是要了解 Toast 提示组件使用时的操作步骤。下面是这一节的内容结构：



我们这一节到这，同学们可以访问【Toast提示组件在线预览】来查看这一节的演示效果，下一节将进行 ActionSheet 组件的开发。

}

28 Loading加载提示样式的设计与开发

30 ActionSheet弹出式菜单组件的设计与开发