

## 31 Article 文本样式的设计与开发

更新时间：2019-08-08 10:20:12



“

不想当将军的士兵，不是好士兵。

——拿破仑

”

我们这一节的任务是实现文本组件的样式，这个组件用来展示文章类的内容，会包括文章的标题、信息、文本段落和图片这几种要素。下来我们先来聊一下这个组件的需求。

### 文本组件的需求

我们这个文本组件中，主要是定义一些基础的文本格式，可以支持比较简单的文章。我们这一节要实现的组件样式如下：

# 什么是UI框架

作者：Rosen

## UI是什么？

先来说下UI，这两字母是User Interface的缩写，一般翻译成“用户界面”。UI最主要的功能就是建立用户和系统后台之间的联系，系统后台通过UI把数据转换成可视化的内容展示给用户，同时用户也要通过UI把操作指令（包括数据）传给系统后台。

对UI不太熟悉的同学一听到这个概念，可能会觉得它的作用就是怎么把一个产品做的漂亮，所以UI设计师经常被人叫成美工。而事实上UI应该是负责“交互”和“视觉”这两方面的工作，这两部分内容构成了产品的用户体

这个组件中包括文章标题、文章信息和文章内容这三大块，其中文章内容里会包含小标题、文本和图片这几个主要的元素。下面我们分析一下我们对这个文本组件的要求。

1. 标题部分使用大字号，文本加粗，其上下部分都留下一定的空间。
2. 文章信息部分主要用来展示文章的作者、创作时间等次要信息，所以通常会把这部分内容淡化，使用不太明显的字体样式。
3. 内容区的副标题要比正常文本的字体稍大，和前后留下一定的空间。
4. 内容区的文本段落部分留有适当的行距，且段落与段落之间留下一定的空间。
5. 内容区的图片水平居中显示，最大宽度不能超过内容区的宽度。
6. 在特殊情况下，需要将文本折叠，以便显示文本后面的其他内容
7. 当需要查看全文的时候可以点击展开文章的按钮来展开文章内容，这个按钮我们用渐变色来实现，如下图所示：

先来说下UI，这两字母是User Interface的缩写，一般翻译成“用户界面”。UI最主要的功能就是建立用户和系统后台之间的联系，系统后台通过UI把数据转换成可视化的内容展示给用户，同时用户也要通过UI把操作指令（包括数据）传给系统后台。

对UI不太熟悉的同学一听到这个概念，可能会觉得它的作用就是怎么把一个产品做的漂亮，所以UI设计师经常被人叫成美工。而事实上UI应该是负责“交互”和“视觉”这两方面的工作，这两部分内容构成了产品的用户体验。



其他内容

其他内容...

这个图片中向下的箭头就是展开文章的按钮，为了更醒目我们可以把它设置成比较明显的颜色，并且让其上下振动。

## 文本组件的设计与开发

### 一、文件的建立

第一步，建立文件。首先是HTML文件 `/demo/article.html`:

```

<!DOCTYPE html>
<html>
  <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-Article文本</title>
  </head>
  <body>
    <div class="tt-content">
      <div class="tt-article fold">
        <h1 class="tt-article-title">什么是UI框架</h1>
        <p class="tt-article-info">作者: Rosen</p>
        <div class="tt-article-content">
          <h2 class="tt-article-subtitle">UI是什么? </h2>
          <p class="tt-article-paragraph">先来说下UI, 这俩字母是User Interface的缩写, 一般翻译成“用户界面”。UI最主要的功能就是建立用户和系统后台之间的联系, 系统后台通过UI把数据转换成可视化的内容展示给用户, 同时用户也要通过UI把操作指令(包括数据)传给系统后台。</p>
          <p class="tt-article-paragraph">对UI不太熟悉的同学一听到这个概念, 可能会觉得它的作用就是怎么把一个产品做的漂亮, 所以UI设计师经常被人叫成美工。而事实上UI应该是负责“交互”和“视觉”这两方面的工作, 这两部分内容构成了产品的用户体验。</p>
          <p class="tt-article-paragraph">用户体验里最重要的应该是这个产品好不好用, 也就是“交互”这部分, 这其中包括产品功能是否完善, 产品流程是否设计的合理, 使用是否方便, 操作是否流畅等。在一些大公司里, 为了保证产品好用, 还会专门设置交互设计师这个职位, 专门做交互部分的设计工作。</p>
          
        </div>
        <div class="tt-article-unfold-btn" id="js-unfold">
          <i class="fa fa-angle-double-down fa-vibrate-y"></i>
        </div>
      </div>
      <div class="tt-panel">
        <div class="tt-panel-title">其他内容</div>
        <div class="tt-panel-body">
          其他内容...
        </div>
      </div>
    </div>
  </body>
</html>

```

这一部分 HTML 结构中, 在 `tt-article` 元素上加上了 `fold` 这个class, 我们准备用它来控制文章是不是需要折叠, 只有在有这个 class 的时候才去折叠内容。然后在 `tt-article` 中包含的是标题、文章信息和文章内容是哪个部分。在内容区里是一个副标题、几个段落和最后的一张测试图片。在内容区的后面就是 `tt-article-unfold-btn` 这个展开全文的按钮。这就是文章组件里会用到的结构。

接下来我们建立 CSS 文件 `/src/article.css`:

```

/*
 * @Author: Rosen
 * @Date: 2019-08-03 21:56:09
 * @Last Modified by: Rosen
 * @Last Modified time: 2019-08-03 22:01:21
 */
/* 文章组件 */
.tt-article{

}

```

最后把这个文件加入到主文件中, 在 `/src/tuitui-ui.css`中添加如下代码:

```

/* article文本组件 */
@import './article.css';

```

好, 这样文件就建好了, 下面我们进入设计和开发阶段。

## 二、文本组件的设计与开发

在这一节中，文本的样式都是比较基础的调整一下文字和间距，稍微复杂一点的就是展开全文那个按钮的实现。我们先来写一下文章容器 **tt-article** 的样式，在 `/src/article.css` 中加入下面代码：

```
/* 文章组件 */
.tt-article{
  position: relative;
  padding: 1rem;
  margin-bottom: 1rem;
  font-size: .8rem;
  line-height: 1.6rem;
  background: #fff;
  border-bottom: 1px solid #eee;
  color: #333;
}
/* 文章被折叠的情况 */
.tt-article.fold{
  max-height: 100%;
  overflow: hidden;
}
```

这个容器基础样式中就是边距、边框和字体的处理，然后在第二条样式中使用 `.fold` 来控制容器的高度，从而达到隐藏多余内容的目的。这里我们把折叠的时候的高度最高设置成100%，这样屏幕稍微往下滑动一点就能看到文本的底边了。这里用的 `max-height` 而没有用 `height` 是因为当文本内容占不满整屏高度的话，也不会留出空白。

接下来我们实现按钮的样式，这个按钮不是一般的按钮样式，而是一个带有渐变效果的区域，点击这个区域都可以达到展开文章的目的。这里我们要先处理一下这个渐变的效果，下面要介绍一下 `background-img` 属性里“`linear-gradient`”这个属性值的用法。

### @ Tips:

**linear-gradient**是 `background-img` 属性中可取的一个属性值，也可以直接用于 `background` 属性中。这个属性值中包含三个主要信息：渐变的方向，渐变的颜色和渐变生效的位置。它完整的格式就是：

```
background: linear-gradient( angle/direction, color1 position1, color2 position2, ...)
```

在这个结构中，第一个参数是渐变的方向，可以是一个角度值，也可以直接指明哪个方向；`color`的取值就是平时用的颜色值，`position`的取值是百分比，表示渐变色的起始或终止的位置。在这一节要实现的文本组件中，我们需要按钮的背景色从透明向纯白色渐变，就可以用如下样式：

```
background: linear-gradient(180deg, rgba(255,255,255,0) 0, #fff 100%);
```

这样就是竖直方向上，由上至下从白色透明渐变到白色不透明，也就达到了我们需要的样式。

有了对 `linear-gradient` 这个属性值的了解，按钮的样式就很好完成了：

```

/* 展开按钮 */
.tt-article .tt-article-unfold-btn{
  display: none;
  position: absolute;
  box-sizing: border-box;
  left: 0;
  right: 0;
  bottom: 0;
  height: 5rem;
  padding-top: 3rem;
  text-align: center;
  font-size: 1.5rem;
  color: #e0652f;
  background: linear-gradient(180deg, rgba(255,255,255,0) 0, #fff 100%);
}
/* 控制展开按钮的显示 */
.tt-article.fold .tt-article-unfold-btn{
  display: block;
}

```

这里我们使用的样式可以把引导图标放到适当的位置，这样就不用再对图标设置单独的样式了。图标上下振动的动画效果我们之前已经实现好了，只需要在图标上加上“**fa-vibrate-y**”这个 **class**。上面第二条样式是限制了这个按钮在文章展开时是隐藏的，只有在文章需要折叠的时候才会出现，这样电完展开文章的按钮以后，这个按钮就消失了。

最后要实现的就是文章中各个元素的样式了，这些样式都是对间距和字体的调整，这应该不用多讲，可以直接贴代码了：

```

/* 文章大标题 */
.tt-article .tt-article-title{
  margin: .5rem 0 1rem;
  font-size: 1.4rem;
  line-height: 2rem;
}
/* 文章信息，用来放作者、创作时间等 */
.tt-article .tt-article-info{
  font-size: .8rem;
  color: #aaa;
}
/* 文章小标题 */
.tt-article .tt-article-subtitle{
  font-size: 1.1rem;
  margin-top: 1rem;
}
/* 文章段落 */
.tt-article .tt-article-paragraph{
  margin: .5rem 0 1rem;
}
/* 文章图片 */
.tt-article .tt-article-img{
  max-width: 100%;
  margin: auto;
}

```

这几个元素的样式中，唯一要注意的是图片的样式，我这里设置的是“**max-width: 100%;**”，是为了图片比内容区窄的时候不做拉伸。如果需要把所有图片都拉伸到内容区那么宽，就可以把 **max-width** 替换成 **width**。

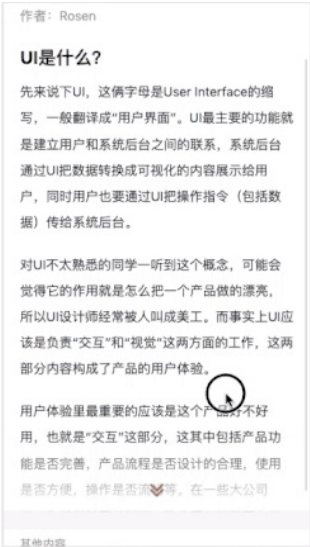
到这里我们这个文章组件的样式就完成了。

### 三、文本组件的效果演示

最后，我们还是给展开按钮添加一段 JS，让它可以点击。这里我们给按钮加上了 `id="js-unfold"`，所以可以直接用它来控制 `tt-article` 上面 `fold` 这个 `class`。我们在 `body` 最后加上下面这段 JS：

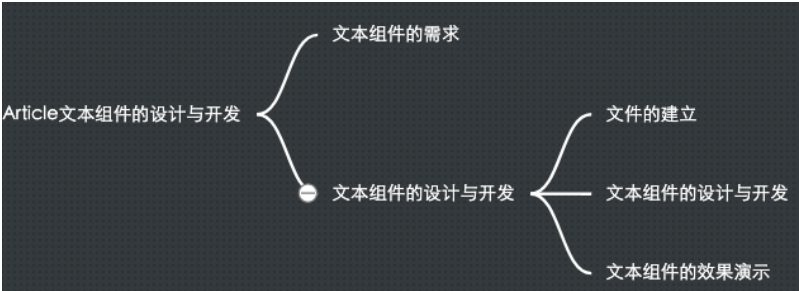
```
<script>
window.onload = ()=>{
  // 展开文章
  document.querySelector('#js-unfold').onclick = (e) => {
    document.querySelector('.tt-article').classList.remove('fold');
  }
};
</script>
```

有这几行脚本，展开全文的按钮就可以点击了，下面是演示效果：



## 结语

到这里，我们这一节就结束了，这一章里所有组件都开发完成了。这一节最主要的就是讲了渐变效果的实现方式。本节内容结构如下：



我们这一节到这，同学们可以访问【[Article文本组件在线预览](#)】来查看这一节的演示效果，下一节我们会对本章的内容做一个总结。

}

