

## 24 单条聊天内容组件

更新时间：2019-09-11 10:04:31



“ 更多一手资源请+V：AndyqcI  
勤能补拙是良训，一分辛劳一分才。 qq：3118617541 ”—华罗庚 ”

在完成了聊天容器组件开发之后，我们就要开始具体的单条聊天内容组件的开发了，单个聊天内容组件开发，就是我们发的每一条消息，消息包含文字和图片，和人物头像和昵称。

本章节完整源代码地址，大家可以事先浏览一下：

[Github](#)

组件效果图

这个和大多数的聊天页面的UI一样，如下图：



开发单条聊天内容组件

在前端项目的 `components` 文件夹下新建 `chatItem` 文件夹，同时新建 `index.vue`：

下面这段代码主要实现了单个聊天消息，包括头像模块，和消息模块，消息模块分为文字消息和图片消息。

```
<template>
  <div :class="data.mine ? 'chat-item chat-item-mine': 'chat-item'">
    
    <div class="right-content">
      <p class="nickname one-line">{{data.fromUser.nickname}}</p>
      <div v-if="data.content.type === 'str'" class="chat-text left-arrow">{{data.content.value}}</div>
      </img>
    </div>
  </div>
</template>
```

针对上面的代码，解释一下其中的逻辑：

1. 单条内容组件，主要有两种类型展示方式，一种是纯文字，一种是图片，采用 `data.content.type` 这个标志位来区分。
2. `data.mine` 这个字段会用来表示是主人态的消息，还是客人态，这个概念后面有解释。
3. `.one-line` 这个 `class` 表示文字只显示一行，超出采用省略号显示，我们在之前创建过的前端项目的`common.css`里面，来定义这个样式，代码如下：

```
.one-line {  
  overflow: hidden;  
  text-overflow: ellipsis;  
  white-space: nowrap;  
}
```

4. 单个聊天内容组件的纯图片展示，我们需要做一下处理，保证不会被压缩，同时不会影响布局，这个我们之前在朋友圈页面也处理过，代码如下：

```
/*  
 * 单图样式  
 */  
imgOneStyle () {  
  return item => {  
    let height = null;  
    let width = null;  
    //如果图片是长图则给定最大的长度  
    if (item.size.height > item.size.width) {  
      height = Math.min(120,item.size.height);  
      //根据比例设置宽度  
      width = height * item.size.width / item.size.height;  
    } else { //如果图片是宽图则给定固定的宽度  
      width = Math.min(120,item.size.width);  
      //根据比例设置高度  
      height = width * item.size.height / item.size.width;  
    }  
    //转换成vw单位  
    return {  
      height: this.pxtovw(height),  
      width: this.pxtovw(width)  
    }  
  }  
},
```

其中，主人态和客人态分别表示：

1. 如果是当前聊天人查看聊天内容，叫做主人态，这时聊天内容是绿色背景，头像在右边。
2. 如果是当前聊天人查看聊天内容，查看对方的消息叫做客人态，聊天内容是白色背景，头像在左边。

我们采用`class .chat-item-mine`来区分，代码如下：

```
.chat-item-mine {  
  flex-direction: row-reverse;  
}
```

下面解释一下 **flex-direction** 这个属性：

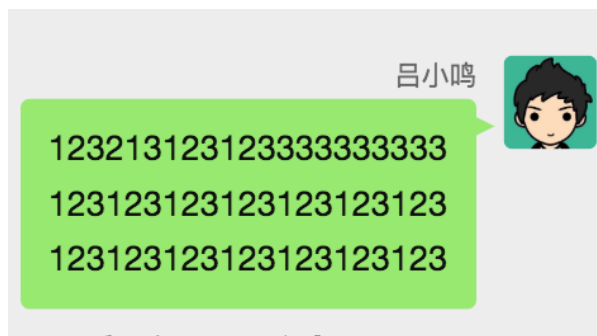
**flex-direction**: `row` | `row-reverse` | `column` | `column-reverse`，这个属性指定了内部元素是如何在 **flex** 容器中布局的，定义了主轴的方向(正方向或反方向)。

- **row**：flex容器的主轴被定义为与文本方向相同。主轴起点和主轴终点与内容方向相同。
- **row-reverse**：表现和row相同，但是置换了主轴起点和主轴终点。
- **column**：flex容器的主轴和块轴相同。主轴起点与主轴终点和书写模式的前后点相同。
- **column-reverse**：表现和column相同，但是置换了主轴起点和主轴终点。

默认情况下样式是 **row**，我们正好利用 **row-reverse** 来调换左右位置。

### CSS实现三角形

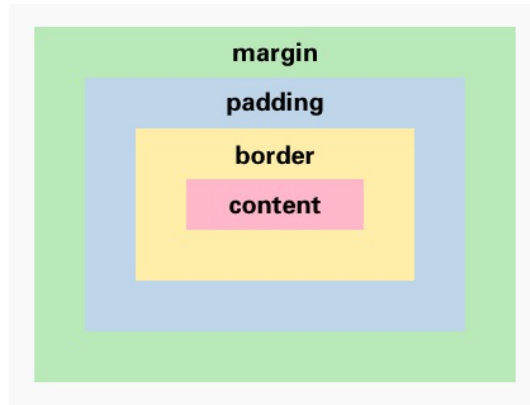
从上面的UI图可以看到，我们在聊天消息的左边或者右边有一个小三角来形成一个气泡效果，我们可以这样利用伪类实现：



更多一手资源请+V：Andyqc1  
aa：3118617541

```
.chat-item-mine .left-arrow::after {  
  content: "";  
  position: absolute;  
  width: 0;  
  height: 0;  
  top: 7px;  
  right: -8px;  
  left: auto;  
  border-top: 5px solid transparent;  
  border-bottom: 5px solid transparent;  
  border-left: 10px solid #94eb68;  
}
```

原理解释：CSS和盒子模型：



一个盒子包括：**margin+border+padding+content**，上下左右边框交界处呈现出平滑的斜线。**transparent**可以给颜色设置透明。

利用这个特点，通过设置不同的上下左右边框宽度或者颜色可以得到小三角，小梯形等，调整宽度大小可以调节三角形形状。我们看到的三角形是边，而不是真的具有内容的区域。

正三角：

```
#triangle-up {  
  width: 0;  
  height: 0;  
  border-left: 50px solid transparent;  
  border-right: 50px solid transparent;  
  border-bottom: 100px solid red;  
}
```

更多一手资源请+V：Andyqc1  
aa：3118617541



只有三条边对应只需要设置**border**的三个方向，同时一条边等于另外两条边之和就会形成等腰三角形。

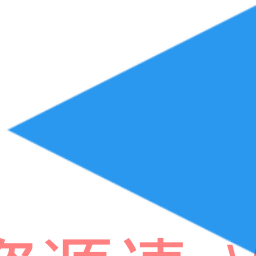
倒三角：

```
#triangle-down {  
  width: 0;  
  height: 0;  
  border-left: 50px solid transparent;  
  border-right: 50px solid transparent;  
  border-top: 100px solid red;  
}
```



左三角：

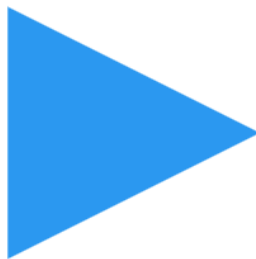
```
#triangle-left {  
  width: 0;  
  height: 0;  
  border-top: 50px solid transparent;  
  border-right: 100px solid red;  
  border-bottom: 50px solid transparent;  
}
```



更多一手资源请+V : AndyqcI  
aa : 3118617541

右三角：

```
#triangle-right {  
  width: 0;  
  height: 0;  
  border-top: 50px solid transparent;  
  border-left: 100px solid red;  
  border-bottom: 50px solid transparent;  
}
```



直角三角形：

```
#triangle-nine {  
  width: 0;  
  height: 0;  
  border-bottom: 100px solid red;  
  border-right: 100px solid transparent;  
}
```



点击图片查看大图

在chatItem文件夹 `index.vue` 下新增逻辑，引入我们之前开发的 `slider.js` 组件，代码如下：

```
showImage (item) {  
  // 初始化Slider 实例  
  new Slider({  
    list: [{  
      img: item.url,  
      height: item.size.height,  
      width: item.size.width  
    }],  
    page: 0  
  })  
},
```

更多一手资源请+V：Andyqc1  
qq：3118617541

小结

本章节主要讲解了单个聊天内容组件的开发，同时讲解了主人态和客人态的布局方式。

相关技术点：

1. 将如何利用CSS伪类实现三角形形成气泡的UI。
2. 讲解了flex-direction的相关属性和使用方法，利用这个属性来实现主人态和客人态的UI区分。

本章节完整源代码地址：

[Github](#)

}