



第四章：Inline Style 专题

- Inline Style 攻略
- 修改导航栏的样式
- 内联样式中使用媒体查询
- 响应式导航栏

样式重构，全局样式

本节不会添加新功能，只是对组件样式进行重构。我们已经定义了几个组件，也写了一些样式代码，现在出现了一个问题，比如说 `NavBar` 组件的背景色和 `AppDrawer` 组件中标题部分的背景色应该保持一致，组件采用的字体也应该相同等等，类似于这样的全局样式，应该怎样在组件中处理呢？

前面小节已经讲过，在使用 Material-UI 组件之前必须先选定配色主题，Material-UI 通过一个 JS 对象 `muiTheme` 控制整体配色方案，这个 `muiTheme` 对象由 `getMuiTheme` 方法提供，查看 `getMuiTheme` 这个方法的源码，我们就能明白给 `getMuiTheme` 传递不同的参数就可以自定义主题色，详情参考官方文档 [Themes](#) 一章中 **Customizing the theme** 部分的内容。在我们自己的代码中，可以通过 `this.context.muiTheme.xxx` 语句获取 `muiTheme` 对象包含的数据，从而可以实现样式的共享。不过，本节将采用更简便的方式，构建自己的样式共享策略。

## 颜色共享

Material-UI 项目包含了谷歌 [Material Design](#) 中的所有 **colors**，你可以在项目中通过下面的代码导入所需要的颜色：

```
import { red50, red100 } from 'material-ui/styles/colors'
```

如果没有找到你需要的颜色，那只好自己定义，新建一个目录 `imports/ui/styles/`，在这个目录下新建一个 `colors.js` 文件，之后我们会把与样式相关的代码放到 `imports/ui/styles` 目录下。遵从 Material-UI 定义颜色的方式，把需要共享的颜色添加进去：

```
export const blue = '#00bcd4';  
export const pink = '#ff4081';
```

其实，上述代码中的两种颜色 Material-UI 都包含了，这里作为演示说明使用。然后，我们就可以在组件中使用自己定义的颜色了：

```
import { blue, pink } from './styles/colors'
```

## 字体共享

Material-UI 支持两种字体库 `Roboto` 和 `sans-serif`，一般中文字体还需要其它的字体库，所以我们需要自己定义与字体相关的样式，同样在 `imports/ui/styles` 目录下新建一个 `typography.js` 的文件，添加代码：

```
export default typography = {  
  fontFamily: 'sans-serif',  
  fontWeightNormal: '300'  
};
```

然后，就是到组件中使用了：

```
import typography from './styles/typography';
```

查看本节更改：[样式重构](#)

欢迎添加 Peter 的微信：happypeter1983

冀ICP备15007992号-3