

Palette 调色板

使用调色板，您可以修改组件的颜色以迎合您的品牌形象。

调色板的颜色

主题提供了以下这些调色板的颜色（在 `theme.palette.` 中获取）：

- *primary* - 用于展示一个给用户的主要界面元素。它是在您的应用屏幕和组件中显示最频繁的一个颜色。
- *secondary* - 用于呈现给用户的次要界面元素。它给予了更多的方法来强调和区分您的产品。此颜色是可选的。
- *error* - 用于呈现用户应该注意到的界面元素。
- *warning* - 用于呈现潜在的一些危险的操作或者重要的信息。
- *info* - 用于向用户呈现一些中立的且不一定重要的信息。
- *success* - 用于指示一个用户触发的操作的成功完成。

如果想要了解更多关于色彩的知识，您可以查看 [色彩章节](#)。

默认值

通过[主题资源管理器](#)或通过打开此页面上的开发工具控制台（dev tools console）（`window.theme.palette`），您可以浏览调色板的默认值。

```
{{"demo": "Intentions.js", "bg": "inline", "hideToolbar": true}}
```

默认调色板使用前缀为 `A`（`A200` 等）的深度作为辅助调色，其他调色使用无前缀的阴影。

Customization 个性化

您可以通过将一个调色板对象（palette object）作为主题的一部分来覆盖默认的调色板值。如果存在以下任何情况：

- `.palette.primary`
- `.palette.secondary`
- `.palette.error`
- `.palette.warning`
- `.palette.info`
- `.palette.success`

这提供了调色板对象，它们将取代默认的颜色对象。

调色板颜色值可以是[颜色 \(color\)](#) 对象，也可以是具有以下 TypeScript 接口指定的一个或多个键（key）的对象：

```
interface PaletteColor {  
  light?: string;  
  main: string;  
  dark?: string;  
  contrastText?: string;  
}
```

使用一个颜色对象

自定义调色板的最简单方法是导入一个或多个提供的颜色：

```
import { createTheme } from '@material-ui/core/styles';
import blue from '@material-ui/core/colors/blue';

const theme = createTheme({
  palette: {
    primary: blue,
  },
});
```

直接提供颜色

如果你想要提供更多的自定义颜色，你可以创建你自己的调色板，或者直接为一些或者所有的 `theme.palette` 键提供颜色：

```
import { createTheme } from '@material-ui/core/styles';

const theme = createTheme({
  palette: {
    primary: {
      // light: 这将从 palette.primary.main 中进行计算,
      main: '#ff4400',
      // dark: 这将从 palette.primary.main 中进行计算,
      // contrastText: 这将计算与 palette.primary.main 的对比度
    },
    secondary: {
      light: '#0066ff',
      main: '#0044ff',
      // dark: 这将从 palette.secondary.main 中进行计算,
      contrastText: '#ffcc00',
    },
    // 使用 `getContrastText()` 来最大化
    // 背景和文本的对比度
    contrastThreshold: 3,
    // 使用下面的函数用于将颜色的亮度在其调色板中
    // 移动大约两个指数。
    contrastThreshold: 3,
    // 使用下面的函数用于将颜色的亮度在其调色板中
    // 移动大约两个指数。
    // 例如，从红色 500 (Red 500) 切换到 红色 300 (Red 300) 或 红色 700 (Red 700) 。
    tonalOffset: 0.2,
  },
});
```

如同上面的例子，如果调色板包含使用 "main"、"light"、"dark" 或 "contrastText" 键中的任何一个自定义颜色，那么这些颜色映射如下所示：

- 如果没有设置 "dark" 和 / 或 "light" 键，那么这将从 "main" 中根据 "色调偏移 (tonalOffset)" 值来进行计算。
- 如果没有设置 "对比度文本 (contrastText)"，那么这将根据 "对比度阈值 (contrastThreshold)" 来计算出与 "main" 的对比度。

“色调偏移 (tonalOffset) ” 和 “对比度阈值 (contrastThreshold) ” 这两个值都可以根据需要进行定制。“色调偏移 (tonalOffset) ” 值可以是一个 0 和 1 之间的数字，它将适用于亮色变量和暗色变量，或者是由以下 TypeScript 类型 (type) 指定的具有明暗变量的对象：

```
type PaletteTonalOffset =  
  | number  
  | {  
    light: number;  
    dark: number;  
  };
```

“色调偏移 (tonalOffset) ” 的值越高，那么计算后的“light” 值就会变得更浅，“dark” 的值会变得更暗。“对比度阈值 (contrastThreshold) ” 的值越高，那么背景色越会被认为是浅色的，这就会赋予一个深色的 “对比度文本 (contrastText) ”。

请注意，“对比度阈值 (contrastThreshold) ” 遵循的是一条非线性曲线。

示例

```
{{"demo": "Palette.js", "defaultCodeOpen": true}}
```

添加新的颜色

You can add new colors inside and outside the palette of the theme as follows:

```
import { createTheme } from '@material-ui/core/styles';  
  
const theme = createTheme({  
  status: {  
    danger: '#e53e3e',  
  },  
  palette: {  
    primary: {  
      main: '#0971f1',  
      darker: '#053e85',  
    },  
    neutral: {  
      main: '#64748b',  
      contrastText: '#fff',  
    },  
  },  
});
```

如果您使用的是 TypeScript，您还需要使用 [module augmentation](#) 来让主题接受上述值。

```
declare module '@material-ui/core/styles/createTheme' {  
  interface Theme {  
    status: {  
      danger: React.CSSProperties['color'];  
    };  
  }  
}
```

```
interface PaletteColor {
  darker?: string;
}
interface SimplePaletteColorOptions {
  darker?: string;
}
interface ThemeOptions {
  status: {
    danger: React.CSSProperties['color'];
  };
}

declare module '@material-ui/core/styles/createPalette' {
  interface Palette {
    neutral: Palette['primary'];
  }
  interface PaletteOptions {
    neutral: PaletteOptions['primary'];
  }
}
```

```
{{"demo": "CustomColor.js"}}
```

选取颜色

需要灵感吗？Material Design 团队已经建立了一个[调色板配置工具](#)来帮助您选择颜色。

暗色模式

Material-UI comes with two palette modes: light (the default) and dark. 你可以通过设置 `mode: 'dark'` 来启用夜间模式。