Paleta

A paleta permite modificar a cor dos componentes para se adequarem à sua marca.

Paleta de cores

O tema expõe as seguintes cores da paleta (acessível sob `theme.palette`.):

- *primary* - used to represent primary interface elements for a user. É a cor mais frequentemente exibida nas telas e componentes do seu aplicativo.
- *secondary* - used to represent secondary interface elements for a user. Ela fornece mais maneiras de realçar e distinguir o seu produto. Tê-la é opcional.
- *error* - used to represent interface elements that the user should be made aware of.
- *warning* - used to represent potentially dangerous actions or important messages.
- *info* - used to present information to the user that is neutral and not necessarily important.
- *success* - used to indicate the successful completion of an action that user triggered.

Se você quiser aprender mais sobre cor, você pode conferir a seção de cores.

Valores padrão

Você pode explorar os valores padrão da paleta usando o explorador de tema ou abrindo o console das ferramentas de desenvolvimento nesta página (`window.theme.palette`).

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

A paleta padrão usa as sombras prefixadas com `A` (`A200` , etc.) para a intenção secundária, e as cores não pré-fixadas para as outras intenções.

Customização

Você pode sobrescrever os valores padrão da paleta incluindo um objeto de paleta como parte do seu tema. Se algum dos seguintes:

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

objetos de cores da paleta são fornecidos, eles substituirão os padrões.

O valor da paleta de cor pode ser um objeto cor, ou um objeto com uma ou mais das chaves especificadas pela seguinte interface TypeScript:

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

Usando um objeto de cor

A maneira mais simples de customizar uma intenção é importar uma ou mais das cores fornecidas e aplicá-las a uma intenção da paleta:

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

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

Fornecendo as cores diretamente

Se você deseja fornecer cores mais customizadas, você pode criar seu próprio objeto de cor, ou fornecer cores diretamente para algumas ou todas as chaves da intenção:

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

const theme = createTheme({
  palette: {
    primary: {
      // light: will be calculated from palette.primary.main,
      main: '#ff4400',
      // dark: will be calculated from palette.primary.main,
      // contrastText: will be calculated to contrast with palette.primary.main
    },
    secondary: {
      light: '#0066ff',
      main: '#0044ff',
      // dark: will be calculated from palette.secondary.main,
      contrastText: '#ffcc00',
    },
    // Used by `getContrastText()` to maximize the contrast between
    // the background and the text.
    contrastThreshold: 3,
    // Usado pelas funções abaixo para mudança de uma cor de luminância por
aproximadamente
    // dois índices dentro de sua paleta tonal.
    // Por exemplo, mude de Red 500 para Red 300 ou Red 700.
    tonalOffset: 0.2,
  },
});
```

Como no exemplo acima, se o objeto de intenção contém cores customizadas usando qualquer uma das chaves "main", "light", "dark" ou "contrastText", os seguintes comportamentos serão aplicados:

- Se as chaves "dark" e / ou "light" são omitidas, seus valores serão calculados de "main", de acordo com o valor "tonalOffset".
- Se "contrastText" é omitido, seu valor será calculado para contrastar com "main", de acordo com o valor de "contrastThreshold".

Tanto os valores de "tonalOffset" e "contrastThreshold" poderão ser customizados conforme o necessário. O "tonalOffset" pode ser um valor numérico entre 0 e 1, que será aplicado a ambas variantes de claro e escuro, ou um objeto com as variantes claro e escuro especificado a seguir pelo tipo TypeScript:

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

Um valor mais alto para "tonalOffset" fará valores calculados para "light" mais claro e "dark" mais escuro. Um valor mais alto para "contrastThreshold" aumenta o ponto no qual uma cor de fundo é considerada clara, e recebe um "contrastText" escuro.

Observe que "contrastThreshold" segue uma curva não linear.

Exemplo

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

Adicionando novas cores

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',
    },
  },
});
```

Se você estiver usando TypeScript, você também deverá usar a extensão de módulos para que o tema aceite os valores acima.

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

Escolhendo cores

Precisa de inspiração? A equipe do Material Design construiu uma ferramenta de configuração de paleta para te ajudar.

Modo escuro

Você pode deixar o tema escuro definindo `mode: 'dark'`.