Customizing Colors

This lesson discusses how to customize and override colors and their swatches.

WE'LL COVER THE FOLLOWING

Customizing main colors

By default, `css-theming` will fall back to the following pre-defined colors map for each brightness:

```
// Colors used in light themes
$ct-colors-light: (
  'grey': #697C93,
  'brown': #96663C,
  'yellow': #F6D300,
  'gold': #CC9A0E,
  'orange': #F48701,
  'red-orange': #FD6A01,
  'red': #FF1D25,
  'pink': #FF39F6,
  'purple': #952FF7,
  'violet': #C600FF,
  'indigo': #5700FF,
  'blue': #448AFF,
  'sky-blue': #00B9FF,
  'lemon-green': #ADD80F,
  'green-blue': #0CCB9B,
  'green': #22D35D,
) !default;

// Colors used in dark themes
$ct-colors-dark: (
```

```
  'grey': #697C93,
  'brown': #96663C,

  'yellow': #F8DC33,
  'gold': #CC9A0E,
  'orange': #F69F34,
  'red-orange': #FC5603,
  'red': #FF3948,
  'pink': #FF61F8,
  'purple': #D835FF,
  'violet': #AA59F9,
  'indigo': #7933FF,
  'blue': #69A1FF,
  'sky-blue': #33C7FF,
  'lemon-green': #ADD80F,
  'green-blue': #3DD5AF,
  'green': #4EDC7D,
) !default;
```

As you can tell, the defined color only mentions the *main* color. The *main* color refers to the **500** swatch of the color: `--{color}-500`, so in the case of `red` for example, it'll be `--red-500`. All other swatches are computed dynamically by `css-theming`.

If you want to override the color map in the theme, you can provide a similar map. You're free to define any number of colors from above, `css-theming` will fall back to the default for the other colors.

If you want to override the default for all themes, you can set `$ct-colors-light` or `$ct-colors-dark` to a new map of colors as long as they follow the same syntax. For example:

```
$ct-themes: (
  'default': (
    'brightness': 'light',
    'colors': (
      'red': #323232,
       ...
    )
  ),
  ...
);
```

Customizing semantic colors

Overriding semantic color targets is very similar, you'll have to provide a similar map to the following default:

```
$ct-semantic-colors: (
  'primary': 'purple',
  'success': 'green',
  'info': 'blue',
  'warning': 'orange',
  'danger': 'red',
) !default;
```

`$ct-semantic-colors` maps a *semantic color* to a *target color* from the color map in the same theme. For example: `primary` to `purple`, `success` to `green`, and so on.

The map assigns a semantic color name (i.e `primary`) to a color target name from the color map in the same theme (i.e `purple`).

Customizing swatches

This is the default definition of the swatches in `css-theming`:

```
$ct-swatches: (
  50: 15%,
  75: 30%,
  100: 40%,
  200: 50%,
  300: 60%,
  400: 80%,
  500: 100%,
) !default;
```

`css-theming` uses this map to compute color swatches. You're free to override this to anything you like. The **key** (i.e `50`, `75`, etc.) in the map will act as the *swatch's name*. For example, from this map, the following variables will be produced for each color:

```
--{color}-50: ...
--{color}-75: ...
--{color}-100: ...
```

```
...
```

The **value** in the map will be used as the percentage of the mixing between the color and the whitish/blackish color `css-theming` chooses, depending on the theme's brightness.

The same method follows for the rest of the props, you can always check the defaults in the src SCSS to figure out what you're supposed to override.