

Cloudscape Customization

Design-tokens

Cloudscape introduces the usage of [design-token](#): A design token is an abstraction of a visual property such as color, size, or animation. Instead of using hard-coded values, such as hex values for colors, tokens are key-value pairs that represent reusable design decisions in the form of a variable.

```
@use '@cloudscape-design/design-tokens-console/index.scss' as awsui;

const CustomPanel = styled.div`
  color: ${awsui.colorTextBodySecondary}
`
```

Style Dictionary

Cloudscape uses [Style Dictionary](#) as the core tool to **transform and distribute design tokens** into multiple platforms (CSS, JS, SCSS, etc.), enabling consistent theming and styling across its components and environments.

Style Dictionary Intro Video

We can find the demo of Style Dictionary (A Style Dictionary is a system that allows you to define styles once, in a way for any platform or language to consume.) here:

<https://amzn.github.io/style-dictionary/#/README?id=watch-the-demo-on-youtube>.

Design-tokens Usage guidelines

Cloudscape provides the [usage guidelines](#)

1. Tip 1: Use existing components first
2. Tip 2: Be thoughtful and intentional: don't overuse and misuse the design token
3. Tip 3: Key > Value: Try to find compatible tokens, and be sure to never alter the values of existing design tokens.

Theme Customization

If we want to customize the theme:

<https://cloudscape.design/foundation/visual-foundation/theming/>

Theming is achieved by changing specific [design tokens](#). There are three categories of design tokens: [typography](#), [colors](#), and [border radii](#).

To experiment with theming, you can modify the values of themeable tokens in our [demos](#) by choosing **Theme** on the right of the top navigation.

If you want to customize the icon of Dropdown:

currently, cannot

<https://github.com/cloudscape-design/components/issues/2639>

someone raised a request to customize the icon but, they replied

it is done this way for consistency. It is a single familiar shape, all users know what happens when clicking on it, no side effects attached.

With custom icons, it becomes unpredictable experience. If we add configurable icon, the very next feature request will be to add tooltip, because various icons are confusing. We will add tooltip – more corner cases will show up and need addressing.

Why going all this way if we can keep the UI plain and simple with consistent icons?