

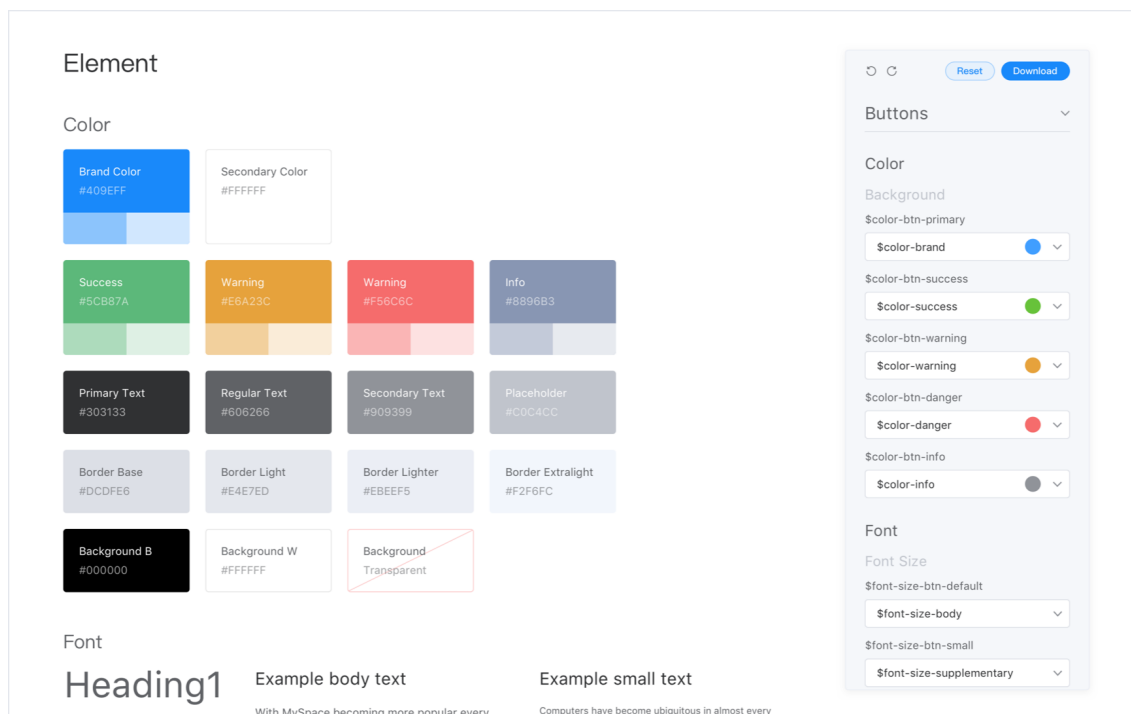
## Thème

Element utilise le style BEM pour le CSS afin que vous puissiez écraser les règles prédéfinies facilement. Mais si vous avez besoin de remplacer des styles à une plus grande échelle, e.g. changer le thème de bleu à orange ou vert, les modifier une par une serait fastidieux. Nous fournissons trois méthodes pour changer les variables de style.

### Theme Roller

Utilisez [Online Theme Roller](#) pour personnaliser les Design Tokens des variables globales et des composants, et prévisualisez le nouveau thème en temps réel. Il peut générer un package de style complet basé sur le nouveau thème que vous pouvez télécharger directement (pour importer de nouveaux fichiers de style dans votre projet, veuillez vous reporter à la partie 'Importer un thème personnalisé' de cette section).

Also, use [Theme Roller Chrome Extension](#), to customize theme and preview in real-time on any website developed by Element.



### Changer la couleur du thème

Si vous avez juste besoin de changer la couleur du thème, le [générateur de thème en ligne](#) est recommandé. La couleur du thème d'Element est un bleu clair et agréable. En le changeant, vous rendez Element visuellement plus adapté à vos projets.

Le site précédent vous permet de visualiser et de télécharger un nouveau thème immédiatement (pour importer un nouveau thème, référez-vous à la partie 'Importer un thème personnalisé' or 'Importer un thème de composant à la demande' de cette section).

### Mettre à jour les variables SCSS de votre projet

`theme-chalk` est écrit en SCSS. Si votre projet utilise SCSS, vous pouvez changer directement les variables d'Élément. Créez un nouveau fichier, e.g. `element-variables.scss` :

```
/* couleur du thème */
$--color-primary: teal;

/* chemin vers le fichier de police, requis */
$--font-path: '~element-ui/lib/theme-chalk/fonts';

@import "~element-ui/packages/theme-chalk/src/index";
```

Puis dans le fichier d'entrée, importez ce style au lieu de celui d'Element:

```
import Vue from 'vue'
import Element from 'element-ui'
import './element-variables.scss'

Vue.use(Element)
```

:::tip Il est obligatoire de remplacer le chemin du fichier de police. :::

## Outil de thème CLI

Si votre projet n'utilise pas SCSS, vous pouvez personnaliser le thème en utilisant notre outil en ligne de commande:

### Installation

Installez le générateur de thème de manière globale ou locale. Une installation locale est recommandé afin que les personnes clonant votre dépôt bénéficient d'une installation automatique via npm.

```
npm i element-theme -g
```

Installer ensuite le thème chalk depuis npm ou GitHub.

```
# Via npm
npm i element-theme-chalk -D

# Via GitHub
npm i https://github.com/ElementUI/theme-chalk -D
```

### Initialisation d'un fichier de variable

Après l'installation des paquets ci-dessus, une commande appelée `et` devient disponible dans le CLI (si les paquets sont installés localement, utilisez `node_modules/.bin/et` à la place). Utilisez `-i` pour initialiser le fichier de variables, `element-variables.scss` par défaut. Vous pouvez spécifier n'importe quel dossier comme sortie.

```
et -i [nom du fichier]

> ✓ Generator variables file
```

Dans `element-variables.scss` vous trouverez toutes les variables utilisées pour le style d'Element, définies en SCSS. Voici un snippet:

```

$--color-primary: #409EFF !default;
$--color-primary-light-1: mix($--color-white, $--color-primary, 10%) !default; /*
53a8ff */
$--color-primary-light-2: mix($--color-white, $--color-primary, 20%) !default; /*
66b1ff */
$--color-primary-light-3: mix($--color-white, $--color-primary, 30%) !default; /*
79bbff */
$--color-primary-light-4: mix($--color-white, $--color-primary, 40%) !default; /*
8cc5ff */
$--color-primary-light-5: mix($--color-white, $--color-primary, 50%) !default; /*
a0cfff */
$--color-primary-light-6: mix($--color-white, $--color-primary, 60%) !default; /*
b3d8ff */
$--color-primary-light-7: mix($--color-white, $--color-primary, 70%) !default; /*
c6e2ff */
$--color-primary-light-8: mix($--color-white, $--color-primary, 80%) !default; /*
d9ecff */
$--color-primary-light-9: mix($--color-white, $--color-primary, 90%) !default; /*
ecf5ff */

$--color-success: #67c23a !default;
$--color-warning: #e6a23c !default;
$--color-danger: #f56c6c !default;
$--color-info: #909399 !default;

...

```

## Modifier les variables

Éditer `element-variables.scss` , e.g. en changeant la couleur du thème pour du rouge:

```
$--color-primary: red;
```

## Générer le thème

Après avoir sauvegardé le fichier de variables, utilisez `et` pour générer le thème. Vous pouvez activer le mode `watch` en ajoutant le paramètre `-w` . Si vous avez choisi le chemin du fichier de sortie, il vous faudra ajouter le paramètre `-c` avec le nom du fichier. Par défaut le thème généré est placé dans `./theme` . Vous pouvez spécifier le dossier de sortie grâce à `-o` .

et

```

> ✓ build theme font
> ✓ build element theme

```

## Use custom theme

### Importer le thème

Importing your own theme is just like importing the default theme, only this time you import the file built from "Online Theme Roller" or "CLI tool":

```
import '../theme/index.css'
import ElementUI from 'element-ui'
import Vue from 'vue'

Vue.use(ElementUI)
```

### Importer le thème d'un composant sur demande

Si vous utilisez `babel-plugin-component` pour les imports sur demande, modifiez `.babelrc` et spécifiez `styleLibraryName` avec le chemin où votre thème est localisé relativement à `.babelrc`. N'oubliez pas que `~` est requis:

```
{
  "plugins": [
    [
      "component",
      {
        "libraryName": "element-ui",
        "styleLibraryName": "~theme"
      }
    ]
  ]
}
```

Si vous n'êtes pas familier avec `babel-plugin-component`, référez-vous à la page [Guide](#). Pour plus d'informations, jetez un oeil au [dépôt du projet](#) de `element-theme`.