

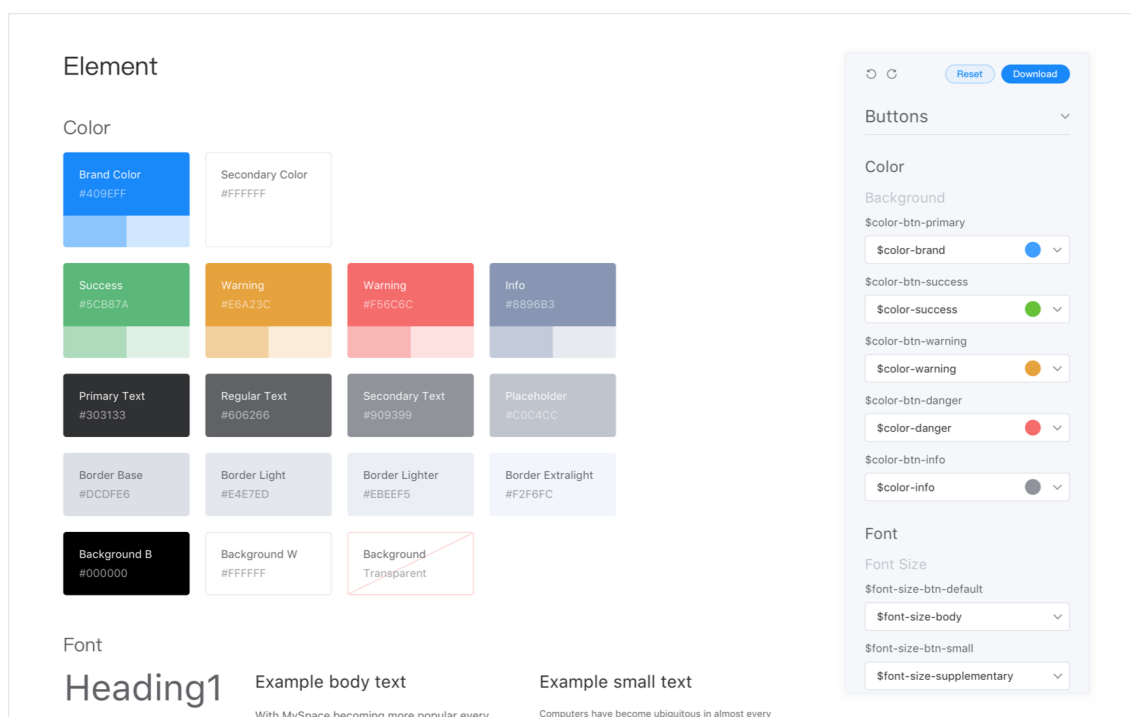
## Tema personalizado

Element utiliza la metodología BEM en CSS con la finalidad de que puedas sobrescribir los estilos fácilmente. Pero, si necesita reemplazar estilos a gran escala, por ejemplo, cambiar el color del tema de azul a naranja o verde, quizás reemplazarlos uno a uno no sea lo más adecuado, para ello hay 4 maneras de modificar los estilos.

### Theme Roller

Use [Online Theme Roller](#) para personalizar el diseño de las variables globales y componentes, y vea el resultado en tiempo real. Puede generar un completo paquete de estilos basado en un nuevo tema que puede bajar directamente (para importar los archivos del nuevo estilo al proyecto por favor vea la sección 'Importar un tema personalizado').

También puede usar [Theme Roller Chrome Extension](#), para personalizar un tema y ver el resultado en tiempo real en cualquier sitio desarrollado con Element.



### Cambiando el color del tema

Si lo que se busca es cambiar el color del tema de Element, se recomienda utilizar el [sitio de visualización de temas](#). Element utiliza un color azul brillante y amigable como tema principal. Al cambiarlo, puede hacer que Element este más conectado visualmente a proyectos específicos.

Este sitio, le permitirá obtener una vista previa del tema con un nuevo color en tiempo real, y, además, obtener un paquete de estilos completo basado en el nuevo color para su descarga (para importar estos nuevos estilos, consulte la sección 'Importar un tema personalizado' o 'Importar un tema de componente bajo demanda' que se encuentran dentro de esta sección).

### Actualizando variables SCSS en tu proyecto

`theme-chalk` esta escrito en SCSS. Si su proyecto también utiliza SCSS, puede cambiar las variables de estilos de Element. Para ello, solo necesita crear un nuevo archivo de estilos, por ejemplo, `element-variables.scss` :

```
/* Color del tema */
$--color-primary: teal;

/* Ubicación de la fuente, obligatoria */
$--font-path: '~element-ui/lib/theme-chalk/fonts';

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

Entonces, en el archivo principal del proyecto, importe este archivo de estilos en lugar de los estilos de Element:

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

Vue.use(Element)
```

:::tip Nota es necesario sobrescribir la ruta de la fuente por una ruta relativa de las fuentes de Element. :::

## CLI para generar temas

Si su proyecto no utiliza SCSS, puede personalizar el tema a través de esta herramienta:

### Instalación

Primero, debe instalar el generador de temas ya sea de forma global o local. Se recomienda instalarlo de forma local, ya que de esta manera, cuando otros clonen su proyecto, npm automáticamente los instalará para ellos.

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

Ahora, instale el tema `chalk` desde npm o Github.

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

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

### Inicializar archivo de variables

Después de haber instalado correctamente los paquetes, el comando `et` estará disponible en su CLI (si instalo el paquete de manera local, utilice `node_modules/.bin/et` en su lugar). Ejecute `-i` para inicializar un archivo de variables, puede especificar un nombre distinto, pero por defecto, el archivo se llama `element-variables.scss`. También puede especificar un directorio distinto.

```
et -i [custom output file]

> ✓ Generator variables file
```

En el archivo `element-variables.scss` podrás encontrar todas las variables que utiliza Element para definir los estilos y estos están definidos en SCSS. Aquí un ejemplo:

```
$--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;

...
```

### Modificando variables

Solo debe modificar el archivo `element-variables.scss` , por ejemplo, para cambiar el color del tema a rojo:

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

### Construyendo el tema

Después de haber modificado el archivo de variables, utilizaremos el comando `et` para construir nuestro tema. Puedes activar el modo `watch` agregando el parámetro `-w` . Y, si desea personalizar el nombre del archivo, debes agregar el parámetro `-c` seguido del nombre. Por defecto, el archivo de tema construido es colocado dentro de `./theme` . Puede especificar un directorio distinto utilizando el parámetro `-o` .

```
et
```

```
> ✓ build theme font
> ✓ build element theme
```

### Uso de los temas personalizados

### Importar un tema personalizado

Importar su propio tema es igual que importar el tema por defecto, solo que esta vez se deben importar los archivos contruidos con "Online Theme Roller" o "CLI tool":

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

Vue.use(ElementUI)
```

### Importar un tema de componente bajo demanda

Si esta utilizando `babel-plugin-component` para importar bajo demanda, solo debe modificar el archivo `.babelrc` y especificar en la propiedad `styleLibraryName` la ruta en donde se encuentra localizado su tema personalizado relativo a `.babelrc` . **Nota** el carácter `~` es obligatorio:

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

Si no esta familiarizado con `babel-plugin-component` , por favor diríjase a la documentación sobre [Como Iniciar](#). Para más detalles, consulte el [repositorio del proyecto](#) de `element-theme` .