

除了 [less 定制主题](#) 外，我们还提供了 CSS Variable 版本以支持动态切换主题能力。你可以在 [ConfigProvider](#) 进行体验。

## 注意事项

- 该功能通过动态修改 CSS Variable 实现，因而在 IE 中页面将无法正常展示。请先确认你的用户环境是否需要支持 IE。
- 该功能在 `antd@4.17.0-alpha.0` 版本起支持。

## 如何使用

### 引入 `antd.variable.min.css`

替换当前项目引入样式文件为 CSS Variable 版本：

```
-- import 'antd/dist/antd.min.css';  
++ import 'antd/dist/antd.variable.min.css';
```

注：如果你使用了 `babel-plugin-import`，需要将其去除。

### 静态方法配置

调用 `ConfigProvider` 配置方法设置主题色：

```
import { ConfigProvider } from 'antd';  
  
ConfigProvider.config({  
  theme: {  
    primaryColor: '#25b864',  
  },  
});
```

## 冲突解决

默认情况下，CSS Variable 会以 `--ant` 作为前缀。当你的项目中引用多份 css 文件时，可以通过修改前缀的方式避免冲突。

### 代码调整

通过 `ConfigProvider` 在顶层修改 `prefixCls`：

```
import { ConfigProvider } from 'antd';  
  
export default () => (  
  <ConfigProvider prefixCls="custom">  
    <MyApp />  
  </ConfigProvider>  
);
```

通过静态方法设置主题色以及对应 `prefixCls`：

```
ConfigProvider.config({
  prefixCls: 'custom',
  theme: {
    primaryColor: '#25b864',
  },
});
```

## 编译 less

由于前缀变更，你需要重新生成一份对应的 css 文件。

```
lessc --js --modify-var="ant-prefix=custom" antd/dist/antd.variable.less
modified.css
```

## 相关变更

为了实现 CSS Variable 并保持原始用法兼容性，我们于 `dist/antd.xxx.less` 文件中添加了 `@root-entry-name: xxx;` 入口注入以支持 less 动态加载对应的 less 文件。一般情况下，你不需要关注该变化。但是，如果你的项目中直接引用了 `lib/es` 目录下的 less 文件，那么你需要在 less 入口处配置 `@root-entry-name: default;`（或者 `@root-entry-name: variable;`）以使 less 可以找到正确的入口。

此外，我们将 `lib/es/style/minxins/index.less` 中的 `@import 'motion'` 和 `@import 'reset'` 迁移至了 `lib/es/style/themes/xxx.less` 中，因为这两个文件依赖了主题相关变量。如果你使用了相关内部方法，请自行调整。当然，我们还是建议直接使用 `dist` 目录下的 `antd.less` 文件而非调用内部文件，因为它们经常会受重构影响。