

Ant Design 设计规范和技术上支持灵活的样式定制，以满足业务和品牌上多样化的视觉需求，包括但不限于全局样式（主色、圆角、边框）和指定组件的视觉定制。

The image shows two identical form layouts side-by-side, demonstrating the visual customization capabilities of Ant Design. The left form is styled with a green theme, while the right form is styled with a purple theme. Both forms contain the following components: a '数字输入框' (Number Input) with the value '3', a '台机器' (Machine) label, a '链接文字' (Link Text) in green/purple, a '开关' (Switch) that is turned on/off, a '滑动输入条' (Slider) with a green/purple track, a '选择器' (Dropdown) with 'jack'/'lucy' selected, a '日期选择框' (Date Picker) showing '2016-11-04'/'请选择日期', and two buttons: '确定' (Confirm) in the theme color and '取消' (Cancel) in gray.

Ant Design 的样式变量

antd 的样式使用了 [Less](#) 作为开发语言，并定义了一系列全局/组件的样式变量，你可以根据需求进行相应调整。

以下是一些最常用的通用变量，所有样式变量可以在 [这里](#) 找到。

```
@primary-color: #1890ff; // 全局主色
@link-color: #1890ff; // 链接色
@success-color: #52c41a; // 成功色
@warning-color: #faad14; // 警告色
@error-color: #f5222d; // 错误色
@font-size-base: 14px; // 主字号
@heading-color: rgba(0, 0, 0, 0.85); // 标题色
@text-color: rgba(0, 0, 0, 0.65); // 主文本色
@text-color-secondary: rgba(0, 0, 0, 0.45); // 次文本色
@disabled-color: rgba(0, 0, 0, 0.25); // 失效色
@border-radius-base: 2px; // 组件/浮层圆角
@border-color-base: #d9d9d9; // 边框色
@box-shadow-base: 0 3px 6px -4px rgba(0, 0, 0, 0.12), 0 6px 16px 0 rgba(0, 0, 0, 0.08),
  0 9px 28px 8px rgba(0, 0, 0, 0.05); // 浮层阴影
```

如果以上变量不能满足你的定制需求，可以给我们提 [issue](#)。

定制方式

原理上是使用 less 提供的 [modifyVars](#) 的方式进行覆盖变量，可以在本地运行 [例子](#) 查看定制效果。下面将针对不同的场景提供一些常用的定制方式。

在 webpack 中定制主题

我们以 webpack@4 为例进行说明，以下是一个 `webpack.config.js` 的典型例子，对 [less-loader](#) 的 options 属性进行相应配置。

```
// webpack.config.js
module.exports = {
  rules: [{
    test: /\.less$/,
    use: [{
      loader: 'style-loader',
    }, {
      loader: 'css-loader', // translates CSS into CommonJS
    }, {
      loader: 'less-loader', // compiles Less to CSS
+     options: {
+       lessOptions: { // 如果使用less-loader@5, 请移除 lessOptions 这一级直接配置选项。
+         modifyVars: {
+           'primary-color': '#1DA57A',
+           'link-color': '#1DA57A',
+           'border-radius-base': '2px',
+         },
+       javascriptEnabled: true,
+     },
+   },
    },
    // ...other rules
  ],
  // ...other config
}
```

注意:

1. less-loader 的处理范围不要过滤掉 `node_modules` 下的 `antd` 包。
2. `lessOptions` 的配置写法在 [less-loader@6.0.0](#) 里支持。

在 Umi 里配置主题

如果你在使用 [Umi](#)，那么可以很方便地在项目根目录的 `.umirc.ts` 或 `config/config.ts` 文件中 `theme` 字段进行主题配置。`theme` 可以配置为一个对象或文件路径。

```
"theme": {
  "primary-color": "#1DA57A",
},
```

或者 [一个js文件](#):

```
"theme": "./theme.js",
```

在 create-react-app 中定制主题

参考 [在 create-react-app 中使用](#) 进行配置即可。

配置 less 变量文件

另外一种方式是建立一个单独的 `less` 变量文件，引入这个文件覆盖 `antd.less` 里的变量。

```
@import '~antd/lib/style/themes/default.less';
@import '~antd/dist/antd.less'; // 引入官方提供的 less 样式入口文件
@import 'your-theme-file.less'; // 用于覆盖上面定义的变量
```

注意，这种方式已经载入了所有组件的样式，不需要也无法和按需加载插件 `babel-plugin-import` 的 `style` 属性一起使用。

动态主题色

在运行时调整主题色请[参考此处](#)。




没有生效？

注意样式必须加载 less 格式，一个常见的问题就是引入了多份样式，less 的样式被 css 的样式覆盖了。

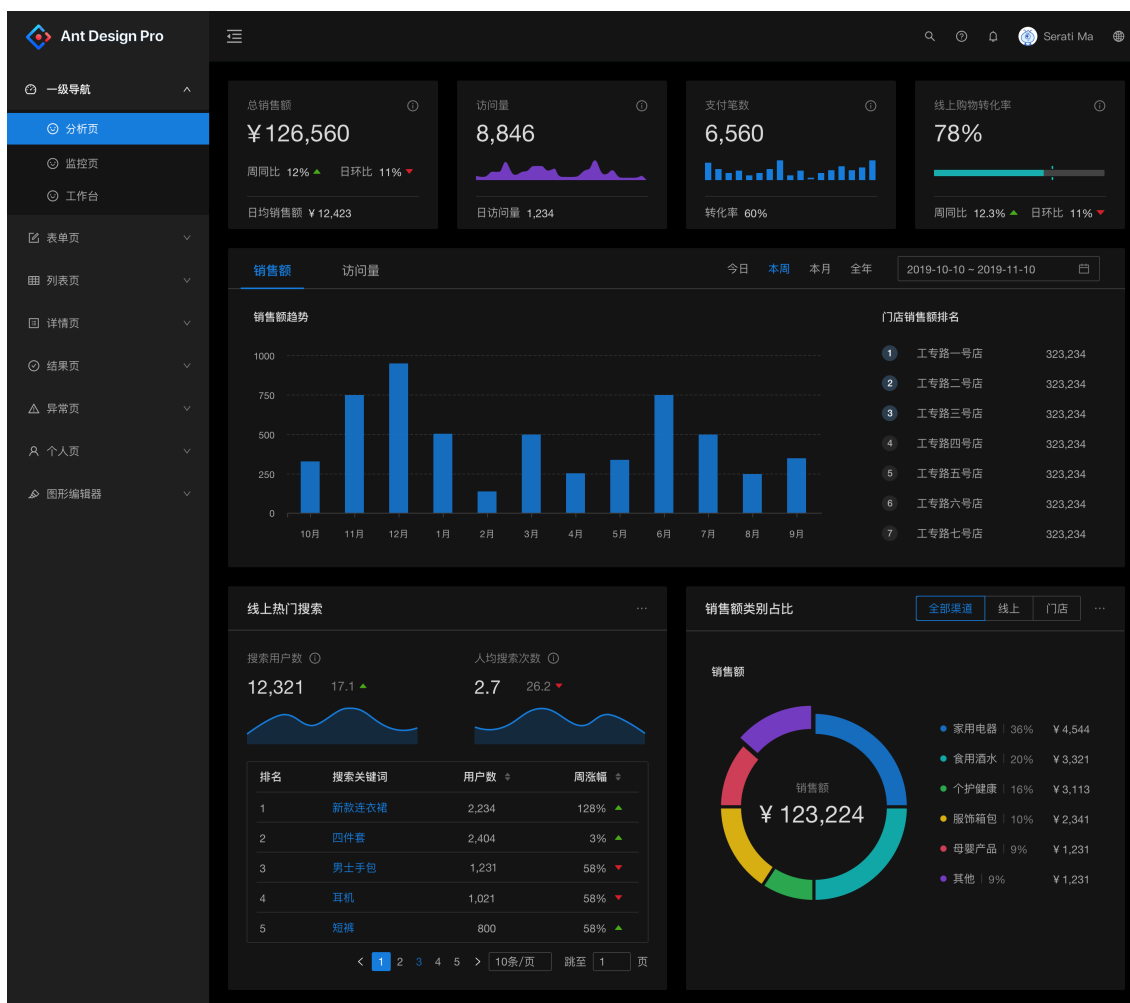
- 如果你在使用 [babel-plugin-import](#) 的 `style` 配置来引入样式，需要将配置值从 `'css'` 改为 `true`，这样会引入 less 文件。
- 如果你是通过 `'antd/dist/antd.css'` 引入样式的，改为 `antd/dist/antd.less`。

官方主题

我们提供了一些官方主题，欢迎在项目中试用，并且给我们提供反馈。

-  暗黑主题 (4.0.0+ 支持)
-  紧凑主题 (4.1.0+ 支持)
-  [阿里云控制台主题 \(Beta\)](#)

使用暗黑主题和紧凑主题



方式一：使用 Umi 3

如果你在使用 [Umi 3](#)：

```
// .umirc.ts or config/config.ts
export default {
  antd: {
    dark: true, // 开启暗色主题
    compact: true, // 开启紧凑主题
  },
},
```

方式二：是在样式文件全量引入 [antd.dark.less](#) 或 [antd.compact.less](#)。

```
@import '~antd/dist/antd.dark.less'; // 引入官方提供的暗色 less 样式入口文件
@import '~antd/dist/antd.compact.less'; // 引入官方提供的紧凑 less 样式入口文件
```

如果项目不使用 Less，可在 CSS 文件中全量引入 [antd.dark.css](#) 或 [antd.compact.css](#)。

```
@import '~antd/dist/antd.dark.css';
@import '~antd/dist/antd.compact.css';
```

注意这种方式下你不需要再引入 `antd/dist/antd.less` 或 `antd/dist/antd.css` 了, 可以安全移除掉。也不需要开启 `babel-plugin-import` 的 `style` 配置。通过此方式不能同时配置两种及以上主题。

方式三: 是用在 `webpack.config.js` 使用 [less-loader](#) 按需引入:

```
const { getThemeVariables } = require('antd/dist/theme');

// webpack.config.js
module.exports = {
  rules: [{
    test: /\.less$/,
    use: [{
      loader: 'style-loader',
    }, {
      loader: 'css-loader', // translates CSS into CommonJS
    }, {
      loader: 'less-loader', // compiles Less to CSS
      + options: {
      +   lessOptions: { // 如果使用less-loader@5, 请移除 lessOptions 这一级直接配置选项。
      +     modifyVars: getThemeVariables({
      +       dark: true, // 开启暗黑模式
      +       compact: true, // 开启紧凑模式
      +     }),
      +     javascriptEnabled: true,
      +   },
      + },
    ]},
  ]},
};
```

社区教程

- [Using Ant Design in Sass-Styled Webpack Projects with antd-scss-theme-plugin](#)
- [How to Customize Ant Design with React & Webpack... the Missing Guide](#)
- [Theming Ant Design with Sass and Webpack](#)
- [Using Sass/Scss with React App \(create-react-app\)](#)
- [Dynamic Theming in Browser using Ant Design](#)
- [Zero config custom theme generator](#)