

从右到左读取

支持从右到左的语言，如阿拉伯语（Arabic）、波斯语（Persian）或希伯来语（Hebrew）。要更改 Material-UI 组件的读取方向，您必须执行以下步骤。

步骤

3. HTML

确保在 body 上设置了 `dir` 属性，否则本机组件将中断：

```
<body dir="rtl"></body>
```

As an alternative to the above, you can also wrap your application in an element with the `dir` attribute:

```
function App() {  
  return (  
    <div dir="rtl">  
      <MyComponent />  
    </div>  
  );  
}
```

This can be helpful for creating components to toggle language settings in the live application.

3. 安装 rtl 插件

在您自定义的主题中设置方向：

```
const theme = createTheme({  
  direction: 'rtl',  
});
```

3.3. 安装 rtl 插件

你需要这个 JSS 插件来翻转样式：[jss-rtl](#)。

```
npm install stylis-plugin-rtl
```

Note: Only `emotion` is compatible with version 2 of the plugin. `styled-components` requires version 1. If you are using `styled-components` as styled engine, make sure to install the correct version. `styled-components` requires version 1. If you are using `styled-components` as styled engine, make sure to install the correct version. `styled-components` requires version 1. If you are using `styled-components` as a [styled engine](#), make sure to install the correct version. `styled-components` requires version 1. If you are using `styled-components` as a [styled engine](#), make sure to install the correct version.

如果你正在使用 `emotion` 或者 `styled-components`，你需要使用该插件来翻转样式：[stylis-plugin-rtl](#)。

```
npm install jss-rtl
```

在你的项目中安装了该插件后，Material-UI 组件仍然需要通过你使用的样式引擎实例来加载它。下面的指南讲述了如何进行加载。

4. 4. 加载 rtl 插件

3.1 JSS

如果你的样式引擎的是 emotion，那么你应该创建并使用 `stylis-plugin-rtl` 的新缓存实例，并将其提供在你应用程序树的顶部。[CacheProvider](#) 组件实现了这一点：

```
import { create } from 'jss';
import rtl from 'jss-rtl';
import { StylesProvider, jssPreset } from '@material-ui/styles';

// Configure JSS
const jss = create({
  plugins: [...jssPreset().plugins, rtl()],
});

function RTL(props) {
  return <StylesProvider jss={jss}>{props.children}</StylesProvider>;
}
```

3.2 emotion

如果你的样式引擎是 `styled-components`，那么你可以使用 [StyleSheetManager](#)，并在 `stylisPlugins` 属性中将 `stylis-plugin-rtl` 作为项目 (item) 提供。

```
import { StyleSheetManager } from 'styled-components';
import rtlPlugin from 'stylis-plugin-rtl';

function RTL(props) {
  return (
    <StyleSheetManager stylisPlugins={[rtlPlugin]}>
      {props.children}
    </StyleSheetManager>
  );
}
```

3.3 styled-components

After installing the plugin in your project, you need to configure the JSS instance to load it. 一旦您通过插件创建了一个新的 JSS 实例，您需要提给组件树中的所有组件。The next step is to make the new JSS instance available to all the components in the component tree. 我们有一个 [StylesProvider](#) 组件来服务这个需求：

```
import rtlPlugin from 'stylis-plugin-rtl';
import { CacheProvider } from '@emotion/react';
import createCache from '@emotion/cache';

// 创建 rtl 缓存
```

```
const cacheRtl = createCache({
  key: 'muiRTL',
  stylisPlugins: [rtlPlugin],
});

function RTL(props) {
  return <CacheProvider value={cacheRtl}>{props.children}</CacheProvider>;
}
```

请前往 [此插件的 README](#) 来了解更多信息。在内部，若 `direction: 'rtl'` 上在主题设置了，withStyles 则会使用该 JSS 插件。

演示

请使用右上角的方向切换按钮来翻转整个文档。

```
{{"demo": "Direction.js"}}
```

选择退出 rtl 转换

emotion & styled-components

你必须使用模板文字语法，并在你要禁用从右到左（right-to-left）样式的规则或属性前添加 `/* @noflip */` 指令。

```
const AffectedText = styled('div')`
  text-align: left;
`;

const UnaffectedText = styled('div')`
  /* @noflip */
  text-align: left;
`;
```

若您想避免一个特殊的特定规则受到 `rtl` 转换的影响，您可以在最开始时加上 `flip: false`。

JSS

若您想避免一个特殊的特定规则受到 `rtl` 转换的影响，您可以在最开始时加上 `flip: false`。

```
const useStyles = makeStyles(
  (theme) => ({
    affected: {
      textAlign: 'right',
    },
    unaffected: {
      flip: false,
      textAlign: 'right',
    },
  }),
),
```

```
    { defaultTheme },  
  );
```