

Da direita para a esquerda

Idiomas da direita para esquerda como árabe, persa ou hebraico são suportados. Para alterar a direção dos componentes de Material-UI, você deve seguir as etapas a seguir.

Passos

1. HTML

Certifique-se de que o atributo `dir` é definido no corpo (body), caso contrário, os componentes nativos serão quebrados:

```
<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.

2. Tema

Defina a direção no seu tema customizado:

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

3. Instale o plugin rtl

Você precisa deste plugin JSS para inverter os estilos: [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.

Se você estiver usando `emotion` ou `styled-components`, você precisa deste plugin de estilo para inverter os estilos: [stylis-plugin-rtl](#).

```
npm install jss-rtl
```

Tendo instalado o plugin em seu projeto, os componentes do Material-UI ainda exigem que ele seja carregado pela instância do motor de estilo que você usa. Encontre guias abaixo de como você pode carregá-lo.

3. Carregando o plugin rtl

3.1 JSS

Depois de criar uma nova instância do JSS com o plugin, você precisará disponibilizá-la para todos os componentes na árvore de componentes. O componente [StylesProvider](#) permite isso:

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

// Create rtl cache
const cacheRtl = createCache({
  key: 'muiRTL',
  stylisPlugins: [rtlPlugin],
});

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

3.2 emotion

Se você usar `styled-components` como seu motor de estilo, você pode usar o [StyleSheetManager](#) e fornecer a propriedade `stylis-plugin-rtl` como um item da propriedade `stylisPlugins`:

```
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

Tendo instalado o plugin em seu projeto, os componentes de Material-UI ainda exigem que ele seja carregado pela instância do jss, conforme descrito abaixo. Depois de criar uma nova instância do JSS com o plugin, você precisará disponibilizá-la para todos os componentes na árvore de componentes. O componente [StylesProvider](#) permite isso:

```
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>;
}
```

Vá para o [README do plugin](#) para aprender mais sobre isso. Internamente, withStyles está usando este plugin JSS quando `direção: 'rtl'` está definido no tema.

Demonstração

Use o botão de alternância de direção no canto superior direito para inverter toda a documentação

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

Optando pela transformação do rtl

emotion & styled-components

Você precisa a sintaxe de template literal e adicionar a diretiva `/* @noflip */` antes da regra ou propriedade para a qual você deseja desativar os estilos da direita para a esquerda.

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

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

Se você quiser evitar que um conjunto de regras específico seja afetado pela transformação `rtl`, você pode adicionar `flip: false` no início.

JSS

Se você quiser evitar que um conjunto de regras específico seja afetado pela transformação `rtl`, você pode adicionar `flip: false` no início.

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

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