

Stepper 步骤条组件

步骤条组件通过数字的步骤来表示进度。它提供了一个类似于安装向导的用户流。

步骤条通过一系列逻辑和编号的步骤来显示当前操作的进度。它们也可用于导航。在保存一个步骤后，步骤条可能会显示短暂的反馈信息。

- **步骤的类型**：可编辑的，不可编辑的，移动端的，可选择的
- **步骤条的类型**：横向的，竖向的，线性的，非线性的

```
{{"component": "modules/components/ComponentLinkHeader.js"}}
```

请注意：步骤条不再出现在 [Material Design 指南](#) 中，但 Material-UI 会继续支持此组件。

水平步骤条

当一个步骤的内容依赖于前一个步骤时，使用水平步骤条是非常适合的。

避免在水平步骤条中使用较长的步骤名称。

线性步骤条

线性步骤条可以让用户依次完成步骤。

您可以在 `activeStep` 属性中传入一个初始值为 0 的当前步骤值来控制 步骤条 。您也可以借助 `orientation` 属性来设置 步骤条 的方向。

这个例子把 `optional` 属性放在第二个 步骤 的组件上，它展示了如何使用一个可选的步骤条。请注意，您可以自行选择管理跳过一个可选的步骤。一旦决定将一个特定步骤设置为可选的，您就必须配置这个属性 `completed={false}` 以表示即使激活的步骤索引超出了可选的步骤，步骤条并没有完成。

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

非线性的步骤条

非线性步进器允许用户在任何点输入多步流程。

此示例类似于常规的水平步骤条，但不同之处在于该组件不再基于 `activeStep` 属性自动设置 `disabled = {true}` 。

在这里使用 `StepButton` 演示了一个可单击的步骤器标签，并且设置了 `completed` 标志。但是，由于可以以非线性方式访问每个步骤，因此需要由您自己的实现来确定何时完成所有步骤（甚至是是否需要完成）。

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

替代标签

您可以将标签置于步骤的图标之下，通过设置 `Stepper` 组件的 `alternativeLabel` 属性可以实现。

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

错误步骤

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

自定义水平步骤条

以下是自定义组件的一个示例。您可以在 [重写文档页面](#) 中了解更多有关此内容的信息。

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

垂直步骤条

垂直步骤条是为狭窄的屏幕而设计的。它们是移动设备的理想选择。同样它们也可以实现水平步骤条的所有功能。

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

文本

步骤内容将在关闭时自动卸载。若您需要提供内容给搜索引擎，亦或是在您的模态框内渲染复杂的组件树，同时还要达到最佳的优化性能，您一般需要以下代码来装载模态框：

```
<StepContent TransitionProps={{ unmountOnExit: false }} />
```

移动设备上的步骤条

该组件实现了适用于移动设备上的紧凑型步骤条。与垂直步骤条相比，它的功能更为有限。如果你还在寻找灵感，请参阅 [移动设备上的步骤条](#)。

当前步骤数和总步骤数以文字形式显示。

文本

当前步骤数和总步骤数以文字形式显示。

```
{{"demo": "TextMobileStepper.js", "bg": true}}
```

点状

这个演示使用 [react-swipeable-views](#) 来创建一个轮播。

```
{{"demo": "SwipeableTextMobileStepper.js", "bg": true}}
```

点状

当步数较少时，请使用点状。

```
{{"demo": "DotsMobileStepper.js", "bg": true}}
```

Progress 进度条组件

当有许多步骤时，或者如果在此过程中需要插入步骤（基于对早期步骤的响应），请使用进度条。

```
{{"demo": "ProgressMobileStepper.js", "bg": true}}
```