

# Stack

Stack 组件管理沿垂直或水平轴的子组件的布局，每个子组件之间有可选的间距和/或分隔线。

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

## 使用

`Stack` is concerned with one-dimensional layouts, while [Grid](#) that handles two-dimensional layouts. The default direction is `column` which stacks children vertically.

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

要控制子组件之间的空间，请使用 `spacing` 属性。The spacing value can be any number, including decimals and any string. The prop is converted into a CSS property using the [theme.spacing\(\)](#) helper.

## 方向

默认情况下，`Stack` 将项目垂直安排在 `column` 中。然而，`direction` prop 也可以用来将项目水平定位在 `row` 中。

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

## Dividers (分隔线)

使用 `divider` prop 在每个子节之间插入一个元素。在 [Divider](#) 组件中运行得尤其好。

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

## 响应式的值

您可以根据活动断点切换 `direction` 或 `spacing` 值。

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

## 交互式

下面是一个交互式的演示，你也可以探索不同设置下的视觉结果：

```
{{"demo": "InteractiveStack.js", "hideToolbar": true, "bg": true}}
```

## System props

作为一个 CSS 实用组件，`Stack` 支持所有 [system](#) 属性。您可以直接在组件上使用它们作为 props。例如，顶边距：

```
<Stack mt={2}>
```