

React Stack component

Stack

The Stack component manages layout of immediate children along the vertical or horizontal axis with optional spacing and/or dividers between each child.

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

Usage

Stack is concerned with one-dimensional layouts, while Grid handles two-dimensional layouts. The default direction is **column** which stacks children vertically.

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

To control space between children, use the **spacing** prop. 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.

Direction

By default, **Stack** arranges items vertically in a **column**. However, the **direction** prop can be used to position items horizontally in a **row** as well.

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

Dividers

Use the **divider** prop to insert an element between each child. This works particularly well with the Divider component.

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

Responsive values

You can switch the **direction** or **spacing** values based on the active breakpoint.

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

Interactive

Below is an interactive demo that lets you explore the visual results of the different settings:

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

System props

As a CSS utility component, the **Stack** supports all **system** properties. You can use them as props directly on the component. For instance, a margin-top:

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