React Masonry component

Masonry

Masonry lays out contents of varying dimensions as blocks of the same width and different height with configurable gaps.

Masonry maintains a list of content blocks with a consistent width but different height. The contents are ordered by row. If a row is already filled with the specified number of columns, the next item starts another row, and it is added to the shortest column in order to optimize the use of space.

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

Basic masonry

A simple example of a `Masonry`. `Masonry` is a container for one or more items. It can receive any element including `<div />` and `<img />`.

{{"demo": "BasicMasonry.js", "bg": true}}

Image masonry

This example demonstrates the use of `Masonry` for images. `Masonry` orders its children by row. If you'd like to order images by column, check out ImageList.

{{"demo": "ImageMasonry.js", "bg": true}}

Items with variable height

This example demonstrates the use of `Masonry` for items with variable height. Items can move to other columns in order to abide by the rule that items are always added to the shortest column and hence optimize the use of space.

{{"demo": "MasonryWithVariableHeightItems.js", "bg": true}}

Columns

This example demonstrates the use of the `columns` to configure the number of columns of a `Masonry`.

{{"demo": "FixedColumns.js", "bg": true}}

`columns` accepts responsive values:

{{"demo": "ResponsiveColumns.js", "bg": true}}

Spacing

This example demonstrates the use of the `spacing` to configure the spacing between items. It is important to note that the value provided to the `spacing` prop is multiplied by the theme's spacing field.

{{"demo": "FixedSpacing.js", "bg": true}}

`spacing` accepts responsive values:

{{"demo": "ResponsiveSpacing.js", "bg": true}}

Server-side rendering

This example demonstrates the use of the `defaultHeight`, `defaultColumns` and `defaultSpacing`, which are used to support server-side rendering.

> **Note**: `defaultHeight` should be large enough to render all rows. Also, it is worth mentioning that items are not added to the shortest column in case of server-side rendering.

{{"demo": "SSRMasonry.js", "bg": true}}