

CSS Grid

Quickly manage the layout, alignment, and sizing of grid columns, navigation, components, and more with a full suite of responsive grid utilities.

If you are **new to or unfamiliar with grid**, you're encouraged to read this [CSS-Tricks grid](#) guide.

Properties for the parent

display

To define a `grid` container, you must specify the `display` CSS property to have one of the values: `grid` or `inline-grid`.

```
{{"demo": "Display.js", "defaultCodeOpen": false}}
```

```
<Box sx={{ display: 'grid' }}>...</Box>
<Box sx={{ display: 'inline-grid' }}>...</Box>
```

grid-template-rows

The `grid-template-rows` property defines the line names and track sizing functions of the grid rows.

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

grid-template-columns

The `grid-template-columns` property defines the line names and track sizing functions of the grid columns.

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

gap

The `gap: size` property specifies the gap between the different items inside the CSS grid.

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

row-gap & column-gap

The `row-gap` and `column-gap` CSS properties allow for specifying the row and column gaps independently.

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

grid-template-areas

The `grid-template-area` property defines a grid template by referencing the names of the grid areas which are specified with the `grid-area` property.

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

grid-auto-columns

The `grid-auto-column` property specifies the size of an implicitly-created grid column track or pattern of tracks.

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

On the demo above, the second non-visible column has a width of `1fr / 4` which is approximately equal to `25%`.

grid-auto-rows

The `grid-auto-rows` property specifies the size of an implicitly-created grid row track or pattern of tracks.

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

grid-auto-flow

The `grid-auto-flow` property controls how the auto-placement algorithm works, specifying exactly how auto-placed items get flowed into the grid.

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

Properties for the children

grid-column

The `grid-column` property is a shorthand for `grid-column-start` + `grid-column-end`. You can see how it's used in the [grid-auto-columns example](#).

You can either set the start and end line:

```
<Box sx={{ gridColumn: '1 / 3' }}>...
```

Or set the number of columns to span:

```
<Box sx={{ gridColumn: 'span 2' }}>...
```

grid-row

The `grid-row` property is a shorthand for `grid-row-start` + `grid-row-end`. You can see how it's used in the [grid-auto-rows example](#).

You can either set the start and end line:

```
<Box sx={{ gridRow: '1 / 3' }}>...
```

Or set the number of rows to span:

```
<Box sx={{ gridRow: 'span 2' }}>...
```

grid-area

The `grid-area` property allows you to give an item a name so that it can be referenced by a template created with the `grid-template-areas` property. You can see how it's used in the [grid-template-area example](#).

```
<Box sx={{ gridArea: 'header' }}>...
```

API

```
import { grid } from '@mui/system';
```

Import name	Prop	CSS property	Theme key
gap	gap	gap	none
columnGap	columnGap	column-gap	none
rowGap	rowGap	row-gap	none
gridColumn	gridColumn	grid-column	none
gridRow	gridRow	grid-row	none
gridAutoFlow	gridAutoFlow	grid-auto-flow	none
gridAutoColumns	gridAutoColumns	grid-auto-columns	none
gridAutoRows	gridAutoRows	grid-auto-rows	none
gridTemplateColumns	gridTemplateColumns	grid-template-columns	none
gridTemplateRows	gridTemplateRows	grid-template-rows	none
gridTemplateAreas	gridTemplateAreas	grid-template-areas	none
gridArea	gridArea	grid-area	none