

Pagination

The Pagination component enables the user to select a specific page from a range of pages.

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

Basic pagination

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

Outlined pagination

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

Rounded pagination

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

Pagination size

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

Buttons

You can optionally enable first-page and last-page buttons, or disable the previous-page and next-page buttons.

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

Custom icons

It's possible to customize the control icons.

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

Pagination ranges

You can specify how many digits to display either side of current page with the `siblingRange` prop, and adjacent to the start and end page number with the `boundaryRange` prop.

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

Controlled pagination

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

Router integration

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

`usePagination`

For advanced customization use cases, a headless `usePagination()` hook is exposed. It accepts almost the same options as the `Pagination` component minus all the props related to the rendering of JSX. The `Pagination` component is built on this hook.

```
import { usePagination } from '@mui/material/Pagination';
```


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

Table pagination

The `Pagination` component was designed to paginate a list of arbitrary items when infinite loading isn't used. It's preferred in contexts where SEO is important, for instance, a blog.

For the pagination of a large set of tabular data, you should use the `TablePagination` component.

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

 *Note that the `Pagination` `page` prop starts at 1 to match the requirement of including the value in the URL, while the `TablePagination` `page` prop starts at 0 to match the requirement of zero-based JavaScript arrays that comes with rendering a lot of tabular data.*

You can learn more about this use case in the [table section](#) of the documentation.

Accessibility

ARIA

The root node has a role of "navigation" and aria-label "pagination navigation" by default. The page items have an aria-label that identifies the purpose of the item ("go to first page", "go to previous page", "go to page 1" etc.). You can override these using the `getItemAriaLabel` prop.

Keyboard

The pagination items are in tab order, with a tabindex of "0".