

# Pagination

A long list can be divided into several pages using `Pagination`, and only one page will be loaded at a time.

## When To Use

- When it will take a long time to load/render all items.
- If you want to browse the data by navigating through pages.

## Examples

< **1** 2 3 4 5 >

### Basic

Basic pagination.

```
import React from 'react';
import { Pagination } from 'antd';

const App: React.FC = () => <Pagination defaultCurrent={1} total={50} />;

export default App;
```

< 1 ... 4 5 **6** 7 8 ... 50 > 10 / page ▾

### More

More pages.

```
import React from 'react';
import { Pagination } from 'antd';

const App: React.FC = () => <Pagination defaultCurrent={6} total={500} />;

export default App;
```

< 1 2 **3** 4 5 ... 50 > 10 / page ▾

< 1 2 **3** 4 5 ... 50 > 10 / page ▾

### Changer

Change  .

```
import React from 'react';
import type { PaginationProps } from 'antd';
import { Pagination } from 'antd';

const onShowSizeChange: PaginationProps['onShowSizeChange'] = (current, pageSize) => {
  console.log(current, pageSize);
};

const App: React.FC = () => (
  <>
    <Pagination
      showSizeChanger
      onShowSizeChange={onShowSizeChange}
      defaultCurrent={3}
      total={500}
    />
    <br />
    <Pagination
      showSizeChanger
      onShowSizeChange={onShowSizeChange}
      defaultCurrent={3}
      total={500}
      disabled
    />
  </>
);

export default App;
```

< 1 **2** 3 4 5 ... 50 > 10 / page ▾ Go to  Page

< 1 **2** 3 4 5 ... 50 > 10 / page ▾ Go to  Page

### Jumper

Jump to a page directly.

```
import React from 'react';
import type { PaginationProps } from 'antd';
import { Pagination } from 'antd';

const onChange: PaginationProps['onChange'] = (pageNumber) => {
  console.log('Page: ', pageNumber);
};

const App: React.FC = () => (
  <>
    <Pagination showQuickJumper defaultCurrent={2} total={500} onChange={onChange} />
    <br />
    <Pagination showQuickJumper defaultCurrent={2} total={500} onChange={onChange} disabled />
  </>
);

export default App;
```

< **1** 2 3 4 5 >

< **1** 2 3 4 5 > 10 / page ▾ Go to  Page

Total 50 items < **1** 2 3 4 5 >

Total 50 items < **1** 2 3 4 5 > 10 / page ▾ Go to  Page

#### Mini size [🔗](#)

Mini size pagination.

```
import React from 'react';
import type { PaginationProps } from 'antd';
import { Pagination } from 'antd';

const showTotal: PaginationProps['showTotal'] = (total) => `Total ${total} items`;

const App: React.FC = () => (
  <>
    <Pagination size="small" total={50} />
    <Pagination size="small" total={50} showSizeChanger showQuickJumper />
    <Pagination size="small" total={50} showTotal={showTotal} />
    <Pagination
      size="small"
      total={50}
      disabled
      showTotal={showTotal}
      showSizeChanger
      showQuickJumper
    />
  </>
);

export default App;
```

<  / 5 >

<  / 5 >

#### Simple mode [🔗](#)

Simple mode.

```
import React from 'react';
import { Pagination } from 'antd';

const App: React.FC = () => (
  <>
    <Pagination simple defaultCurrent={2} total={50} />
    <br />
    <Pagination disabled simple defaultCurrent={2} total={50} />
  </>
);

export default App;
```

< 1 2 **3** 4 5 >

### Controlled

Controlled page number.

```
import React, { useState } from 'react';
import type { PaginationProps } from 'antd';
import { Pagination } from 'antd';

const App: React.FC = () => {
  const [current, setCurrent] = useState(3);

  const onChange: PaginationProps['onChange'] = (page) => {
    console.log(page);
    setCurrent(page);
  };

  return <Pagination current={current} onChange={onChange} total={50} />;
};

export default App;
```

Total 85 items < **1** 2 3 4 5 > 20 / page ▾

1-20 of 85 items < **1** 2 3 4 5 > 20 / page ▾

### Total number

You can show the total number of data by setting `showTotal`.

```
import React from 'react';
import { Pagination } from 'antd';

const App: React.FC = () => (
  <>
    <Pagination
      total={85}
      showTotal={(total) => `Total ${total} items`}
      defaultPageSize={20}
      defaultCurrent={1}
    />
    <br />
    <Pagination
      total={85}
      showTotal={(total, range) => `${range[0]}-${range[1]} of ${total} items`}
      defaultPageSize={20}
      defaultCurrent={1}
    />
  </>
);

export default App;
```

Total 85 items < **1** 2 3 4 5 ... 9 > 10 / page ▾ Go to  Page

[Show All](#) 


Show all configured prop.

```
import React from 'react';
import { Pagination } from 'antd';

const App: React.FC = () => (
  <Pagination
    total={85}
    showSizeChanger
    showQuickJumper
    showTotal={(total) => `Total ${total} items`}
  />
);

export default App;
```

Previous **1** 2 3 4 5 ... 50 [Next](#) 10 / page ▾

[Prev and next](#) 

Use text link for prev and next button.

```
import React from 'react';
import type { PaginationProps } from 'antd';
import { Pagination } from 'antd';

const itemRender: PaginationProps['itemRender'] = (_, type, originalElement) => {
  if (type === 'prev') {
    return <a>Previous</a>;
  }
  if (type === 'next') {
    return <a>Next</a>;
  }
  return originalElement;
};

const App: React.FC = () => <Pagination total={500} itemRender={itemRender} />;

export default App;
```

## API

`<Pagination onChange={onChange} total={50} />`

Property	Description	Type	Default	Version
current	Current page number	number	–	
defaultCurrent	Default initial page number	number	1	
defaultPageSize	Default number of data items per page	number	10	

Property	Description	Type	Default	Version
disabled	Disable pagination	boolean	-	
hideOnSinglePage	Whether to hide pager on single page	boolean	false	
itemRender	To customize item's innerHTML	(page, type: 'page'   'prev'   'next', originalElement) => React.ReactNode	-	
pageSize	Number of data items per page	number	-	
pageSizeOptions	Specify the sizeChanger options	string[]   number[]	[ 10 , 20 , 50 , 100 ]	
responsive	If <code>size</code> is not specified, <code>Pagination</code> would resize according to the width of the window	boolean	-	
showLessItems	Show less page items	boolean	false	
showQuickJumper	Determine whether you can jump to pages directly	boolean   { goButton: ReactNode }	false	
showSizeChanger	Determine whether to show <code>pageSize</code> select, it will be true when <code>total &gt; 50</code>	boolean	-	
showTitle	Show page item's title	boolean	true	
showTotal	To display the total number and range	function(total, range)	-	
simple	Whether to use simple mode	boolean	-	
size	Specify the size of <code>Pagination</code> , can be set to <code>small</code>	default   small	default	
total	Total number of data items	number	0	
onChange	Called when the page number or <code>pageSize</code> is changed, and it takes the resulting page number and	function(page, pageSize)	-	

Property	Description	Type	Default	Version
	pageSize as its arguments			
onShowSizeChange	Called when <code>pageSize</code> is changed	<code>function(current, size)</code>	-	

## Design Token

### ▼ Global Token

Token Name	Description	Type	Default Value
colorBgContainer	Container background color, e.g: default button, input box, etc. Be sure not to confuse this with `colorBgElevated`.	string	<code>#ffffff</code>
colorBgContainerDisabled	Control the background color of container in disabled state.	string	<code>rgba(0, 0, 0, 0.04)</code>
colorBgTextActive	Control the background color of text in active state.	string	<code>rgba(0, 0, 0, 0.15)</code>
colorBgTextHover	Control the background color of text in hover state.	string	<code>rgba(0, 0, 0, 0.06)</code>
colorBorder	Default border color, used to separate different elements, such as: form separator, card separator, etc.	string	<code>#d9d9d9</code>
colorPrimary	Brand color is one of the most direct visual elements to reflect the characteristics and communication of the product. After you have selected the brand color, we will automatically generate a complete color palette and assign it effective design semantics.	string	<code>#1677ff</code>
colorPrimaryBorder	The stroke color under the main color gradient, used on the stroke of components such as Slider.	string	<code>#91caff</code>
colorPrimaryHover	Hover state under the main color gradient.	string	<code>#4096ff</code>
colorText	Default text color which comply with W3C standards, and this color is also the darkest neutral color.	string	<code>rgba(0, 0, 0, 0.88)</code>
colorTextDisabled	Control the color of text in disabled state.	string	<code>rgba(0, 0, 0, 0.25)</code>
colorTextPlaceholder	Control the color of placeholder text.	string	<code>rgba(0, 0, 0, 0.25)</code>
borderRadius	Border radius of base components	number	6
borderRadiusLG	LG size border radius, used in some large border radius components, such as Card, Modal and other components.	number	8

Token Name	Description	Type	Default Value
borderRadiusSM	SM size border radius, used in small size components, such as Button, Input, Select and other input components in small size	number	4
controlHeight	The height of the basic controls such as buttons and input boxes in Ant Design	number	32
controlHeightLG	LG component height	number	40
controlHeightSM	SM component height	number	24
controlItemBgActiveDisabled	Control the background color of control component item when active and disabled.	string	<code>rgba(0, 0, 0, 0.15)</code>
controlOutline	Control the outline color of input component.	string	<code>rgba(5, 145, 255, 0.1)</code>
controlOutlineWidth	Control the outline width of input component.	number	2
controlPaddingHorizontal	Control the horizontal padding of an element.	number	12
controlPaddingHorizontalSM	Control the horizontal padding of an element with a small-medium size.	number	8
fontFamily	The font family of Ant Design prioritizes the default interface font of the system, and provides a set of alternative font libraries that are suitable for screen display to maintain the readability and readability of the font under different platforms and browsers, reflecting the friendly, stable and professional characteristics.	string	-apple-system, BlinkMacSystemFont, 'Segoe UI', Roboto, 'Helvetica Neue', Arial, 'Noto Sans', sans-serif, 'Apple Color Emoji', 'Segoe UI Emoji', 'Segoe UI Symbol', 'Noto Color Emoji'
fontSize	The most widely used font size in the design system, from which the text gradient will be derived.	number	14
fontSizeLG	Large font size	number	16
fontSizeSM	Small font size	number	12
fontWeightStrong	Control the font weight of heading components (such as h1, h2, h3) or selected item.	number	600
lineHeight	Line height of text.	number	1.5714285714285714
lineHeightLG	Line height of large text.	number	1.5
lineType	Border style of base components	string	solid
lineWidth	Border width of base components	number	1



Token Name	Description	Type	Default Value
lineWidthFocus	Control the width of the line when the component is in focus state.	number	4
margin	Control the margin of an element, with a medium size.	number	16
marginSM	Control the margin of an element, with a medium-small size.	number	12
marginXS	Control the margin of an element, with a small size.	number	8
marginXXS	Control the margin of an element, with the smallest size.	number	4
motionDurationMid	Motion speed, medium speed. Used for medium element animation interaction.	string	0.2s
motionDurationSlow	Motion speed, slow speed. Used for large element animation interaction.	string	0.3s
paddingSM	Control the small padding of the element.	number	12
paddingXS	Control the extra small padding of the element.	number	8
paddingXXS	Control the extra extra small padding of the element.	number	4
screenLG	Control the screen width of large screens.	number	992
screenSM	Control the screen width of small screens.	number	576
wireframe	Used to change the visual effect of the component to wireframe, if you need to use the V4 effect, you need to enable the configuration item	boolean	false