

Segmented

Segmented Controls. This component is available since `antd@4.20.0`.

When To Use

- When displaying multiple options and user can select a single option;
- When switching the selected option, the content of the associated area changes.

Examples

Daily Weekly Monthly Quarterly Yearly

Basic

The most basic usage.

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

export default () => <Segmented options={['Dai
```

123 456 longtext-lo...

Block Segmented

`block` property will make the `Segmented` fit to its parent width.

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

export default () => (
  <Segmented block options={['123, 456, 'longte
```

Map Transit Satellite

Daily Weekly Monthly Quarterly Yearly

Basic

Disabled Segmented.

```
import React from 'react';
import { Segmented, Space } from 'antd';

const App: React.FC = () => (
  <Space direction="vertical">
    <Segmented options={['Map', 'Transit', 'Satellite'] />
    <Segmented
      options={['
        'Daily',
        { label: 'Weekly', value: 'Weekly', disabled: true },
        'Monthly',
        { label: 'Quarterly', value: 'Quarterly', disabled: true },
        'Yearly',
      ]
    />
  </Space>
);

export default App;
```

Map Transit Satellite

Controlled mode

Controlled Segmented.

```
import React, { useState } from 'react';
import { Segmented } from 'antd';

const Demo: React.FC = () => {
  const [value, setValue] = useState<string | null>('Map');

  return <Segmented options={['Map', 'Transit', 'Satellite'] value={value} onChange={setValue} />;
};

export default Demo;
```



Custom Render [✎](#)

Custom each Segmented Item by ReactNode.

```
import React from 'react';
import { Avatar, Segmented, Space } from 'antd';
import { UserOutlined } from '@ant-design/ico

const App: React.FC = () => (
  <Space direction="vertical">
    <Segmented
      options={[
        {
          label: (
            <div style={{ padding: 4 }}>
              <Avatar src="https://xs

            </div>
          ),
          value: 'user1',
        },
        {
          label: (
            <div style={{ padding: 4 }}>
              <Avatar style={{ backgroun

            </div>
          ),
          value: 'user2',
        },
        {
          label: (
            <div style={{ padding: 4 }}>
              <Avatar style={{ backgroun

            </div>
          ),
          value: 'user3',
        },
      ]}
    />
    <Segmented
      options={[
        {
          label: (
            <div style={{ padding: 4 }}>
              <div>Spring</div>
              <div>Jan-Mar</div>
            </div>
          ),
          value: 'spring',
        },
        {
          label: (
            <div style={{ padding: 4 }}>
              <div>Summer</div>
              <div>Apr-Jun</div>
            </div>
          ),
          value: 'summer',
        },
      ]}
    />
  )
);
```

Daily Weekly Monthly

Load more options

Dynamic [✎](#)

Load `options` dynamically.

```
import React, { useState } from 'react';
import { Segmented, Button, Space } from 'ant

const Demo: React.FC = () => {
  const [options, setOptions] = useState(['Dai
  const [moreLoaded, setMoreLoaded] = useState

  const handleLoadOptions = () => {
    setOptions((prev) => [...prev, 'Quarterly'
    setMoreLoaded(true);
  };

  return (
    <Space direction="vertical">
      <Segmented options={options} />
      <Button type="primary" disabled={moreLoc
        Load more options
      </Button>
    </Space>
  );
};

export default Demo;
```

List Kanban

With Icon [✎](#)

Set `icon` for Segmented Item.

```
import React from 'react';
import { Segmented } from 'antd';
import { AppstoreOutlined, BarsOutlined } from

export default () => (
  <Segmented
    options={[
      {
        label: 'List',
        value: 'List',
        icon: <BarsOutlined />,
      },
      {
        label: 'Kanban',
        value: 'Kanban',
        icon: <AppstoreOutlined />,
      },
    ]}
  />
);
```

Daily Weekly Monthly Quarterly Yearly

Daily **Weekly** Monthly Quarterly Yearly

Daily Weekly **Monthly** Quarterly Yearly

Three sizes of Segmented [✎](#)

There are three sizes of an Segmented: `large`

(40px), `default` (32px) and `small` (24px).

```
import React from 'react';
import { Segmented, Space } from 'antd';

const App: React.FC = () => (
  <Space direction="vertical">
    <Segmented size="large" options={['Daily',
    <Segmented options={['Daily', 'Weekly', 'Monthly', 'Quarterly', 'Yearly']} size="large" />
    <Segmented size="small" options={['Daily', 'Weekly', 'Monthly', 'Quarterly', 'Yearly']} size="small" />
  </Space>
);

export default App;
```

`:≡` `☐☐`

With Icon only [✎](#)

Set `icon` without `label` for Segmented Item.

```
import React from 'react';
import { Segmented } from 'antd';
import { AppstoreOutlined, BarsOutlined } from '@ant-design/icons';

export default () => (
  <Segmented
    options={[
      {
        value: 'List',
        icon: <BarsOutlined />,
      },
      {
        value: 'Kanban',
        icon: <AppstoreOutlined />,
      },
    ]}
  />
);
```

API

This component is available since `antd@4.20.0`

Property	Description	Type	Default	Version
<code>block</code>	Option to fit width to its parent's width	<code>boolean</code>	<code>false</code>	
<code>defaultValue</code>	Default selected value	<code>string number</code>		
<code>disabled</code>	Disable all segments	<code>boolean</code>	<code>false</code>	
<code>onChange</code>	The callback function that is triggered when the state changes	<code>function(value: string number)</code>		
<code>options</code>	Set children optional	<code>string[] number[] Array<{ label: ReactNode value: string icon? ReactNode disabled?: boolean className?: string }></code>	<code>[]</code>	
<code>size</code>	The size of the Segmented.	<code>large middle small</code>	<code>middle</code>	
<code>value</code>	Currently selected value	<code>string number</code>		

Design Token

▼ Global Token

Token Name	Description	Type	Default Value
<code>colorBgElevated</code>	Container background color of the popup layer, in dark mode the color value of this token will be a little brighter than `colorBgContainer`. E.g: modal, pop-up, menu, etc.	<code>string</code>	<code>#ffffff</code>
<code>colorBgLayout</code>	This color is used for the background color of the overall layout of the page. This token will only be used when it is necessary to be at the B1 visual level in the page. Other usages are wrong.	<code>string</code>	<code>#f5f5f5</code>
<code>colorFillSecondary</code>	The second level of fill color can outline the shape of the element more clearly, such as Rate, Skeleton, etc. It can also be used as the Hover state of the third level of fill color, such as Table, etc.	<code>string</code>	<code>rgba(0, 0, 0, 0.06)</code>
<code>colorText</code>	Default text color which comply with W3C standards, and this color is also the darkest neutral color.	<code>string</code>	<code>rgba(0, 0, 0, 0.88)</code>

Token Name	Description	Type	Default Value
colorTextDisabled	Control the color of text in disabled state.	string	<code>rgba(0, 0, 0, 0.25)</code>
colorTextLabel	Control the font color of text label.	string	<code>rgba(0, 0, 0, 0.65)</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
borderRadiusSM	SM size border radius, used in small size components, such as Button, Input, Select and other input components in small size	number	4
borderRadiusXS	XS size border radius, used in some small border radius components, such as Segmented, Arrow and other components with small border radius.	number	2
boxShadowTertiary	Control the tertiary box shadow style of an element.	string	0 1px 2px 0 rgba(0, 0, 0, 0.03), 0 1px 6px -1px rgba(0, 0, 0, 0.02), 0 2px 4px 0 rgba(0, 0, 0, 0.02)
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
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
lineHeight	Line height of text.	number	1.5714285714285714
lineWidth	Border width of base components	number	1

Token Name	Description	Type	Default Value
lineWidthBold	The default line width of the outline class components, such as Button, Input, Select, etc.	number	2
marginSM	Control the margin of an element, with a medium-small size.	number	12
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
motionEaseInOut	Preset motion curve.	string	cubic-bezier(0.645, 0.045, 0.355, 1)
paddingXXS	Control the extra extra small padding of the element.	number	4