

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

Basic

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

const Demo: React.FC = () => (
  <Segmented<string>
    options={['Daily', 'Weekly', 'Monthly', 'Quarterly', 'Yearly']}
    onChange={(value) => {
      console.log(value); // string
    }}
  />
);

export default Demo;
```

Vertical Direction

v5.21.0

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

const Demo: React.FC = () => (
  <Segmented
    vertical
    options={[
      { value: 'List', icon: <BarsOutlined /> },
      { value: 'Kanban', icon: <AppstoreOutlined /> },
    ]}
  />
);

export default Demo;
```

Block Segmented

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

const Demo: React.FC = () => (
  <Segmented options={[123, 456, 'longtext-longtext-longtext-longtext']}>
</block />
);

export default Demo;
```

Round shape

v5.24.0

```
import React, { useState } from 'react';
import { MoonOutlined, SunOutlined } from '@ant-design/icons';
import { Flex, Segmented } from 'antd';
import type { SizeType } from '../config-provider/SizeContext';

const Demo: React.FC = () => {
  const [size, setSize] = useState<SizeType>('middle');
  return (
    <Flex gap="small" align="flex-start" vertical>
      <Segmented
        options={['small', 'middle', 'large']}
        value={size}
        onChange={(value) => setSize(value as SizeType)}
      />
      <Segmented
        size={size}
        shape="round"
        options={[
          { value: 'light', icon: <SunOutlined /> },
          { value: 'dark', icon: <MoonOutlined /> },
        ]}
      />
    </Flex>
  );
};

export default Demo;
```

Disabled

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

const App: React.FC = () => (
  <Flex gap="small" align="flex-start" vertical>
    <Segmented options={['Map', 'Transit', 'Satellite']} disabled />
    <Segmented
      options={[
        'Daily',
        { label: 'Weekly', value: 'Weekly', disabled: true },
        'Monthly',
        { label: 'Quarterly', value: 'Quarterly', disabled: true },
        'Yearly',
      ]}
    />
  </Flex>
);

export default App;
```

Controlled mode

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

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

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

export default Demo;
```

Custom Render

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

const App: React.FC = () => (
  <Flex gap="small" align="flex-start" vertical>
    <Segmented
      options={[
```

```

{
  label: (
    <div style={{ padding: 4 }}>
      <Avatar src="https://api.dicebear.com/7.x/miniavs/svg?seed=8"
/>
      <div>User 1</div>
    </div>
  ),
  value: 'user1',
},
{
  label: (
    <div style={{ padding: 4 }}>
      <Avatar style={{ backgroundColor: '#f56a00' }}>K</Avatar>
      <div>User 2</div>
    </div>
  ),
  value: 'user2',
},
{
  label: (
    <div style={{ padding: 4 }}>
      <Avatar style={{ backgroundColor: '#87d068' }} icon=
{<UserOutlined />} />
      <div>User 3</div>
    </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',
  },
  {
    label: (
      <div style={{ padding: 4 }}>
        <div>Autumn</div>
        <div>Jul-Sept</div>
      </div>
    ),
    value: 'autumn',
  },
  {
    label: (
      <div style={{ padding: 4 }}>
        <div>Winter</div>
        <div>Oct-Dec</div>
      </div>
    ),
    value: 'winter',
  },
]}
/>
</Flex>
);

export default App;

```

Dynamic

```

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

const Demo: React.FC = () => {
  const [options, setOptions] = useState(['Daily', 'Weekly', 'Monthly']);
  const [moreLoaded, setMoreLoaded] = useState(false);

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

  return (
    <Flex gap="small" align="flex-start" vertical>

```

```

        <Segmented options={options} />
        <Button type="primary" disabled={moreLoaded} onClick=
{handleLoadOptions}>
            Load more options
        </Button>
    </Flex>
  );
};

export default Demo;

```

Three sizes of Segmented

```

import React from 'react';
import { Flex, Segmented } from 'antd';

const App: React.FC = () => (
  <Flex gap="small" align="flex-start" vertical>
    <Segmented size="large" options={['Daily', 'Weekly', 'Monthly',
'Quarterly', 'Yearly']} />
    <Segmented options={['Daily', 'Weekly', 'Monthly', 'Quarterly',
'Yearly']} />
    <Segmented size="small" options={['Daily', 'Weekly', 'Monthly',
'Quarterly', 'Yearly']} />
  </Flex>
);

export default App;

```

With Icon

```

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

const Demo: React.FC = () => (
  <Segmented
    options={[
      { label: 'List', value: 'List', icon: <BarsOutlined /> },
      { label: 'Kanban', value: 'Kanban', icon: <AppstoreOutlined /> },
    ]}
  />
);

```

```
export default Demo;
```

With Icon only

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

const Demo: React.FC = () => (
  <Segmented
    options={[
      { value: 'List', icon: <BarsOutlined /> },
      { value: 'Kanban', icon: <AppstoreOutlined /> },
    ]}
  />
);

export default Demo;
```

With name

v5.23.0

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

const Demo: React.FC = () => (
  <Segmented<string> options={['Daily', 'Weekly', 'Monthly', 'Quarterly',
    'Yearly']} name="group" />
);

export default Demo;
```

Controlled Synced mode

Debug

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

const Demo: React.FC = () => {
  const [foo, setFoo] = useState<string | number>('AND');
  return (
    <>
      <Segmented value={foo} options={['AND', 'OR', 'NOT']} onChange=
```

```
{setFoo} />
    &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&~
    <Segmented value={foo} options={['AND', 'OR', 'NOT']} onChange=
{setFoo} />
</>
);
};

export default Demo;
```

Consistent height

Debug

```
import React from 'react';
import { Button, Flex, Input, Segmented, Select } from 'antd';

const App: React.FC = () => (
  <Flex gap="small" vertical>
    <div>
      <Segmented
        size="large"
        style={{ marginInlineEnd: 6 }}
        options={['Daily', 'Weekly', 'Monthly']}
      />
      <Button type="primary" size="large">
        Button
      </Button>
    </div>
    <div>
      <Segmented
        size="middle"
        style={{ marginInlineEnd: 6 }}
        options={['Daily', 'Weekly', 'Monthly']}
      />
      <Input placeholder="default size" style={{ width: 150 }} />
    </div>
    <div>
      <Segmented
        size="small"
        style={{ marginInlineEnd: 6 }}
        options={['Daily', 'Weekly', 'Monthly']}
      />
      <Select size="small" defaultValue="lucy" style={{ width: 150 }}>
        <Select.Option value="lucy">Lucy</Select.Option>
      </Select>
    </div>
  </Flex>
);
```



```

    </div>
  </Flex>
);

export default App;

```

Custom component token

Debug

```

import React from 'react';
import { ConfigProvider, Segmented } from 'antd';

const Demo: React.FC = () => (
  <ConfigProvider
    theme={{
      components: {
        Segmented: {
          itemColor: '#222',
          itemHoverColor: '#333',
          itemHoverBg: 'rgba(0, 0, 0, 0.06)',
          itemSelectedBg: '#aaa',
          itemActiveBg: '#ccc',
          itemSelectedColor: '#fff',
        },
      },
    }}
  >
    <Segmented options={['Daily', 'Weekly', 'Monthly', 'Quarterly', 'Yearly']} />
  </ConfigProvider>
);

export default Demo;

```

API

Common props ref: [Common props](#)

This component is available since `antd@4.20.0`

Property	Description	Type	Default	Version
block	Option to fit width to its parent's width	boolean	false	
defaultValue	Default selected value	string number		

disabled	Disable all segments	boolean	false	
onChange	The callback function that is triggered when the state changes	function(value: string number)		
options	Set children optional	string[] number[] SegmentedItemType[]	[]	
size	The size of the Segmented.	large middle small	middle	
vertical	Orientation	boolean	false	5.21.0
value	Currently selected value	string number		
shape	shape of Segmented	default round	default	5.24.0
name	The name property of all <code>input[type="radio"]</code> children. if not set, it will fallback to a randomly generated name	string		5.23.0

SegmentedItemType

Property	Description	Type	Default	Version
label	Display text for Segmented item	ReactNode	-	
value	Value for Segmented item	string number	-	
icon	Display icon for Segmented item	ReactNode	-	
disabled	Disabled state of segmented item	boolean	false	
className	The additional css class	string	-	

Design Token