

When To Use

- When there is a group of content on the same level.
- When there is insufficient content space, it can be used to save space in the form of a revolving door.
- Commonly used for a group of pictures/cards.

Examples

Basic

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

const contentStyle: React.CSSProperties = {
  margin: 0,
  height: '160px',
  color: '#fff',
  lineHeight: '160px',
  textAlign: 'center',
  background: '#364d79',
};

const App: React.FC = () => {
  const onChange = (currentSlide: number) => {
    console.log(currentSlide);
  };

  return (
    <Carousel afterChange={onChange}>
      <div>
        <h3 style={contentStyle}>1</h3>
      </div>
      <div>
        <h3 style={contentStyle}>2</h3>
      </div>
      <div>
        <h3 style={contentStyle}>3</h3>
      </div>
      <div>
        <h3 style={contentStyle}>4</h3>
      </div>
    </Carousel>
  );
};
```

```
export default App;
```

Position

```
import React, { useState } from 'react';
import type { CarouselProps, RadioChangeEvent } from 'antd';
import { Carousel, Radio } from 'antd';

type DotPosition = CarouselProps['dotPosition'];

const contentStyle: React.CSSProperties = {
  height: '160px',
  color: '#fff',
  lineHeight: '160px',
  textAlign: 'center',
  background: '#364d79',
};

const App: React.FC = () => {
  const [dotPosition, setDotPosition] = useState<DotPosition>('top');

  const handlePositionChange = ({ target: { value } }: RadioChangeEvent) => {
    setDotPosition(value);
  };

  return (
    <>
      <Radio.Group onChange={handlePositionChange} value={dotPosition}
style={{ marginBottom: 8 }}>
        <Radio.Button value="top">Top</Radio.Button>
        <Radio.Button value="bottom">Bottom</Radio.Button>
        <Radio.Button value="left">Left</Radio.Button>
        <Radio.Button value="right">Right</Radio.Button>
      </Radio.Group>
      <Carousel dotPosition={dotPosition}>
        <div>
          <h3 style={contentStyle}>1</h3>
        </div>
        <div>
          <h3 style={contentStyle}>2</h3>
        </div>
        <div>
          <h3 style={contentStyle}>3</h3>
        </div>
      </Carousel>
    </>
  );
};
```

```

        <div>
          <h3 style={contentStyle}>4</h3>
        </div>
      </Carousel>
    </>
  );
};

export default App;

```

Scroll automatically

```

import React from 'react';
import { Carousel } from 'antd';

const contentStyle: React.CSSProperties = {
  height: '160px',
  color: '#fff',
  lineHeight: '160px',
  textAlign: 'center',
  background: '#364d79',
};

const App: React.FC = () => (
  <Carousel autoplay>
    <div>
      <h3 style={contentStyle}>1</h3>
    </div>
    <div>
      <h3 style={contentStyle}>2</h3>
    </div>
    <div>
      <h3 style={contentStyle}>3</h3>
    </div>
    <div>
      <h3 style={contentStyle}>4</h3>
    </div>
  </Carousel>
);

export default App;

```

Fade in

```

import React from 'react';
import { Carousel } from 'antd';

const contentStyle: React.CSSProperties = {
  height: '160px',
  color: '#fff',
  lineHeight: '160px',
  textAlign: 'center',
  background: '#364d79',
};

const App: React.FC = () => (
  <Carousel effect="fade">
    <div>
      <h3 style={contentStyle}>1</h3>
    </div>
    <div>
      <h3 style={contentStyle}>2</h3>
    </div>
    <div>
      <h3 style={contentStyle}>3</h3>
    </div>
    <div>
      <h3 style={contentStyle}>4</h3>
    </div>
  </Carousel>
);

export default App;

```

Arrows for switching

v5.17.0

```

import React from 'react';
import { Carousel } from 'antd';

const contentStyle: React.CSSProperties = {
  margin: 0,
  height: '160px',
  color: '#fff',
  lineHeight: '160px',
  textAlign: 'center',
  background: '#364d79',
};

```

```

const App: React.FC = () => (
  <>
    <Carousel arrows infinite={false}>
      <div>
        <h3 style={contentStyle}>1</h3>
      </div>
      <div>
        <h3 style={contentStyle}>2</h3>
      </div>
      <div>
        <h3 style={contentStyle}>3</h3>
      </div>
      <div>
        <h3 style={contentStyle}>4</h3>
      </div>
    </Carousel>
    <br />
    <Carousel arrows dotPosition="left" infinite={false}>
      <div>
        <h3 style={contentStyle}>1</h3>
      </div>
      <div>
        <h3 style={contentStyle}>2</h3>
      </div>
      <div>
        <h3 style={contentStyle}>3</h3>
      </div>
      <div>
        <h3 style={contentStyle}>4</h3>
      </div>
    </Carousel>
  </>
);

export default App;

```

Progress of dots

v5.24.0

```

import React from 'react';
import { Carousel } from 'antd';

const contentStyle: React.CSSProperties = {
  height: '160px',
  color: '#fff',

```

```

    lineHeight: '160px',
    textAlign: 'center',
    background: '#364d79',
  };

const App: React.FC = () => (
  <Carousel autoplay={{ dotDuration: true }} autoplaySpeed={5000}>
    <div>
      <h3 style={contentStyle}>1</h3>
    </div>
    <div>
      <h3 style={contentStyle}>2</h3>
    </div>
    <div>
      <h3 style={contentStyle}>3</h3>
    </div>
    <div>
      <h3 style={contentStyle}>4</h3>
    </div>
  </Carousel>
);

export default App;

```

Component Token

Debug

```

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

/** Test usage. Do not use in your production. */
const contentStyle: React.CSSProperties = {
  margin: 0,
  height: '160px',
  color: '#fff',
  lineHeight: '160px',
  textAlign: 'center',
  background: '#364d79',
};

export default () => (
  <ConfigProvider
    theme={{
      components: {
        Carousel: {

```

```

        dotWidth: 50,
        dotHeight: 50,
        dotActiveWidth: 80,
      },
    },
  }}
>
<Carousel>
  <div>
    <h3 style={contentStyle}>1</h3>
  </div>
  <div>
    <h3 style={contentStyle}>2</h3>
  </div>
  <div>
    <h3 style={contentStyle}>3</h3>
  </div>
  <div>
    <h3 style={contentStyle}>4</h3>
  </div>
</Carousel>
</ConfigProvider>
);

```

API

Common props ref: [Common props](#)

Property	Description	Type	Default	Version
arrows	Whether to show switch arrows	boolean	false	5.17.0
autoplay	Whether to scroll automatically, you can specify <code>autoplay={{ dotDuration: true }}</code> to display the progress bar	boolean { dotDuration?: boolean }	false	dotDuration: 5.24.0
autoplaySpeed	Delay between each auto scroll (in milliseconds)	number	3000	
adaptiveHeight	Adjust the slide's height automatically	boolean	false	
dotPosition	The position of the dots, which can be one of <code>top</code>	string	bottom	

	bottom left right			
dots	Whether to show the dots at the bottom of the gallery, object for dotsClass	boolean { className?: string }	true	
draggable	Enable scrollable via dragging on desktop	boolean	false	
fade	Whether to use fade transition	boolean	false	
infinite	Infinitely wrap around contents	boolean	true	
speed	Animation speed in milliseconds	number	500	
easing	Transition interpolation function name	string	linear	
effect	Transition effect	scrollx fade	scrollx	
afterChange	Callback function called after the current index changes	(current: number) => void	-	
beforeChange	Callback function called before the current index changes	(current: number, next: number) => void	-	
waitForAnimate	Whether to wait for the animation when switching	boolean	false	

Find more APIs in react-slick [documentation](#).

Methods

Name	Description
goTo(slideNumber, dontAnimate)	Go to slide index, if dontAnimate=true, it happens without animation
next()	Change current slide to next slide
prev()	Change current slide to previous slide

Design Token

FAQ

How to add custom arrows?

See [#12479](#).