

何时使用 {#when-to-use}

- 当有一组平级的内容。
- 当内容空间不足时，可以用走马灯的形式进行收纳，进行轮播展现。
- 常用于一组图片或卡片轮播。

代码演示

基本

```
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;
```

位置

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

```

        </Carousel>
      </>
    );
  };

  export default App;

```

自动切换

```

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;

```

渐显

```

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;

```

切换箭头

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;

```

进度条

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;

```

组件 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

通用属性参考：[通用属性](#)

参数	说明	类型	默认值	版本
arrows	是否显示箭头	boolean	false	5.17.0
autoplay	是否自动切换，如果为 object 可以指定 dotDuration 来展示指示点进度条	boolean { dotDuration?: boolean }	false	dotDuration: 5.24.0
autoplaySpeed	自动切换的间隔（毫秒）	number	3000	
adaptiveHeight	高度自适应	boolean	false	
dotPosition	面板指示点位置，可选 top bottom left right	string	bottom	
dots	是否显示面板指示点，如果为 object 则可以指定 dotsClass	boolean { className?: string }	true	
draggable	是否启用拖拽切换	boolean	false	
fade	使用渐变切换动效	boolean	false	

infinite	是否无限循环切换（实现方式是复制两份 children 元素，如果子元素有副作用则可能会引发 bug）	boolean	true	
speed	切换动效的时间（毫秒）	number	500	
easing	动画效果	string	linear	
effect	动画效果函数	scrollx fade	scrollx	
afterChange	切换面板的回调	(current: number) => void	-	
beforeChange	切换面板的回调	(current: number, next: number) => void	-	
waitForAnimate	是否等待切换动画	boolean	false	

更多 API 可参考：<https://react-slick.neostack.com/docs/api>

方法

名称	描述
goTo(slideNumber, dontAnimate)	切换到指定面板, dontAnimate = true 时，不使用动画
next()	切换到下一面板
prev()	切换到上一面板

主题变量（Design Token）

FAQ

如何自定义箭头？

可参考 [#12479](#)。