何时使用 {#when-to-use}

常用于引导用户了解产品功能。

代码演示

基本

```
import React, { useRef, useState } from 'react';
import { EllipsisOutlined } from '@ant-design/icons';
import { Button, Divider, Space, Tour } from 'antd';
import type { TourProps } from 'antd';

const App: React.FC = () => {
  const ref1 = useRef(null);
  const ref2 = useRef(null);
  const ref3 = useRef(null);

  const [open, setOpen] = useState<boolean>(false);

  const steps: TourProps['steps'] = [
    {
      title: 'Upload File',
      description: 'Put your files here.',
      cover: (
        <img
          alt="tour.png"
          src="https://user-images.githubusercontent.com/5378891/197385811-55df8480-7ff4-44bd-9d43-a7dade598d70.png"
        />
      ),
      target: () => ref1.current,
    },
    {
      title: 'Save',
      description: 'Save your changes.',
      target: () => ref2.current,
    },
    {
      title: 'Other Actions',
      description: 'Click to see other actions.',
      target: () => ref3.current,
    },
  ];
  return (
    <>
```

```
      <Button type="primary" onClick={() => setOpen(true)}>
        Begin Tour
      </Button>
      <Divider />
      <Space>
        <Button ref={ref1}> Upload</Button>
        <Button ref={ref2} type="primary">
          Save
        </Button>
        <Button ref={ref3} icon={<EllipsisOutlined />} />
      </Space>
      <Tour open={open} onClose={() => setOpen(false)} steps={steps} />
    </>
  );
};

export default App;
```

非模态

```
import React, { useRef, useState } from 'react';
import { EllipsisOutlined } from '@ant-design/icons';
import { Button, Divider, Space, Tour } from 'antd';
import type { TourProps } from 'antd';

const App: React.FC = () => {
  const ref1 = useRef(null);
  const ref2 = useRef(null);
  const ref3 = useRef(null);

  const [open, setOpen] = useState<boolean>(false);

  const steps: TourProps['steps'] = [
    {
      title: 'Upload File',
      description: 'Put your files here.',
      cover: (
        <img
          alt="tour.png"
          src="https://user-images.githubusercontent.com/5378891/197385811-55df8480-7ff4-44bd-9d43-a7dade598d70.png"
        />
      ),
      target: () => ref1.current,
    },
    {
      title: 'Save',
      description: 'Save your changes.',
      target: () => ref2.current,
    },
    {
      title: 'Other Actions',
      description: 'Click to see other actions.',
      target: () => ref3.current,
    },
  ];

  return (
    <>
      <Button type="primary" onClick={() => setOpen(true)}>
        Begin non-modal Tour
      </Button>

      <Divider />

      <Space>
        <Button ref={ref1}> Upload</Button>
        <Button ref={ref2} type="primary">
          Save
        </Button>
        <Button ref={ref3} icon={<EllipsisOutlined />} />
      </Space>

      <Tour open={open} onClose={() => setOpen(false)} mask={false}
type="primary" steps={steps} />
    </>
  );
};

export default App;
```

位置

```
import React, { useRef, useState } from 'react';
import { Button, Tour } from 'antd';
import type { TourProps } from 'antd';

const App: React.FC = () => {
  const ref = useRef(null);

  const [open, setOpen] = useState<boolean>(false);
  const steps: TourProps['steps'] = [
    {
      title: 'Center',
      description: 'Displayed in the center of screen.',
      target: null,
    },
    {
      title: 'Right',
      description: 'On the right of target.',
      placement: 'right',
      target: () => ref.current,
    },
    {
      title: 'Top',
      description: 'On the top of target.',
      placement: 'top',
      target: () => ref.current,
    },
  ];

  return (
    <>
      <Button type="primary" onClick={() => setOpen(true)} ref={ref}>
        Begin Tour
      </Button>

      <Tour open={open} onClose={() => setOpen(false)} steps={steps} />
    </>
  );
};

export default App;
```

自定义遮罩样式

```
import React, { useRef, useState } from 'react';
import { EllipsisOutlined } from '@ant-design/icons';
import { Button, Divider, Space, Tour } from 'antd';
import type { TourProps } from 'antd';

const App: React.FC = () => {
  const ref1 = useRef(null);
  const ref2 = useRef(null);
  const ref3 = useRef(null);

  const [open, setOpen] = useState<boolean>(false);
  const steps: TourProps['steps'] = [
    {
      title: 'Upload File',
      description: 'Put your files here.',
      cover: (
        <img
          alt="tour.png"
          src="https://user-images.githubusercontent.com/5378891/197385811-
55df8480-7ff4-44bd-9d43-a7dade598d70.png"
        />
      ),
      target: () => ref1.current,
    },
    {
      title: 'Save',
      description: 'Save your changes.',
      target: () => ref2.current,
      mask: {
        style: {
          boxShadow: 'inset 0 0 15px #fff',
        },
        color: 'rgba(40, 0, 255, .4)',
      },
    },
    {
      title: 'Other Actions',
      description: 'Click to see other actions.',
      target: () => ref3.current,
      mask: false,
    },
  ];

  return (
    <>
      <Button type="primary" onClick={() => setOpen(true)}>
        Begin Tour
      </Button>

      <Divider />

      <Space>
        <Button ref={ref1}> Upload</Button>
        <Button ref={ref2} type="primary">
          Save
        </Button>
        <Button ref={ref3} icon={<EllipsisOutlined />} />
      </Space>

      <Tour
        open={open}
        onClose={() => setOpen(false)}
        steps={steps}
        mask={{
          style: {
            boxShadow: 'inset 0 0 15px #333',
          },
          color: 'rgba(80, 255, 255, .4)',
        }}
      />
    </>
  );
};

export default App;
```

自定义指示器

```
import React, { useRef, useState } from 'react';
import { EllipsisOutlined } from '@ant-design/icons';
import type { GetRef, TourProps } from 'antd';
import { Button, Divider, Space, Tour } from 'antd';

const App: React.FC = () => {
  const ref1 = useRef<GetRef<typeof Button>>(null);
  const ref2 = useRef<GetRef<typeof Button>>(null);
  const ref3 = useRef<GetRef<typeof Button>>(null);

  const [open, setOpen] = useState<boolean>(false);

  const steps: TourProps['steps'] = [
    {
      title: 'Upload File',
      description: 'Put your files here.',
      target: () => ref1.current!,
    },
    {
      title: 'Save',
      description: 'Save your changes.',
      target: () => ref2.current!,
    },
    {
      title: 'Other Actions',
      description: 'Click to see other actions.',
      target: () => ref3.current!,
    },
  ];

  return (
    <>
      <Button type="primary" onClick={() => setOpen(true)}>
        Begin Tour
      </Button>
      <Divider />
      <Space>
        <Button ref={ref1}>Upload</Button>
        <Button ref={ref2} type="primary">
          Save
        </Button>
        <Button ref={ref3} icon={<EllipsisOutlined />} />
      </Space>
      <Tour
        open={open}
        onClose={() => setOpen(false)}
        steps={steps}
        indicatorsRender={(current, total) => (
          <span>
            {current + 1} / {total}
          </span>
        )}
      />
    </>
  );
};

export default App;
```

自定义高亮区域的样式

```
import React, { useRef, useState } from 'react';
import { Button, Col, Row, Slider, Space, Tour, Typography } from 'antd';
import type { TourProps } from 'antd';

const { Text } = Typography;

const App: React.FC = () => {
  const tourNodeRef = useRef(null);
  const [radius, setRadius] = useState(8);
  const [offsetX, setOffsetX] = useState(2);
  const [offsetY, setOffsetY] = useState(2);
  const [offset, setOffset] = useState(2);
  const [open, setOpen] = useState(false);
  const [offsetDirection, setOffsetDirection] = useState<'both' |
'individual'>('individual');

  const steps: TourProps['steps'] = [
    {
      title: 'Upload File',
      description: 'Put your files here.',
      cover: (
        <img
          alt="tour.png"
          src="https://user-images.githubusercontent.com/5378891/197385811-
55df8480-7ff4-44bd-9d43-a7dade598d70.png"
        />
      ),
      target: () => tourNodeRef.current,
    },
  ];

  const offsetGap =
    offsetDirection === 'both'
      ? { offset }
      : {
          offset: [offsetX, offsetY] as [number, number],
        };
  return (
    <div ref={tourNodeRef}>
      <Button type="primary" onClick={() => setOpen(true)}>
        Begin Tour
      </Button>
      <Space style={{ display: 'flex', marginTop: 12 }}
direction="vertical">
        <Row>
          <Col span={6}>
            <Text>Radius:</Text>
          </Col>
          <Col span={12}>
            <Slider value={radius} onChange={(val) => val &&
setRadius(val)} />
          </Col>
        </Row>
        <Row>
          <Col span={6}>
          <Text> offset:</Text>
        </Col>
        <Col span={12}>
          <Slider
            value={offset}
            max={50}
            onChange={(val) => val && setOffset(val)}
            onFocus={() => setOffsetDirection('both')}
          />
        </Col>
      </Row>
      <Row>
        <Col span={6}>
          <Text>Horizontal offset:</Text>
        </Col>
        <Col span={12}>
          <Slider
            value={offsetX}
            max={50}
            onChange={(val) => val && setOffsetX(val)}
            onFocus={() => setOffsetDirection('individual')}
          />
        </Col>
      </Row>
      <Row>
        <Col span={6}>
          <Text>Vertical offset:</Text>
        </Col>
        <Col span={12}>
          <Slider
            value={offsetY}
            max={50}
            onChange={(val) => val && setOffsetY(val)}
            onFocus={() => setOffsetDirection('individual')}
          />
        </Col>
      </Row>
    </Space>
    <Tour
      open={open}
      onClose={() => setOpen(false)}
      steps={steps}
      gap={{ ...offsetGap, radius }}
    />
  </div>
);
```

```
};

export default App;
```

_InternalPanelDoNotUseOrYouWillBeFired

Debug

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

/** Test usage. Do not use in your production. */
const { _InternalPanelDoNotUseOrYouWillBeFired: InternalPanel } = Tour;

export default () => (
  <div
    style={{
      display: 'flex',
      flexDirection: 'column',
      rowGap: 16,
      background: 'rgba(50,0,0,0.65)',
      padding: 8,
    }}
  >
    <InternalPanel title="Hello World!" description="Hello World?!" />
    <InternalPanel
      title="Hello World!"
      description="Hello World?!"
      cover={
        <img
          alt="tour.png"
          src="https://user-images.githubusercontent.com/5378891/197385811-55df8480-7ff4-44bd-9d43-a7dade598d70.png"
        />
      }
      current={5}
      total={7}
    />
    <InternalPanel
      title="Hello World!"
      description="Hello World?!"
      type="primary"
      current={4}
      total={5}
    />
  </div>
);
```

API

通用属性参考：通用属性

Tour

属性	说明	类型	默认值	版本
arrow	是否显示箭头，包含是否指向元素中心的配置	`boolean` | `{ pointAtCenter: boolean}`	`true`	
closeIcon	自定义关闭按钮	`React.ReactNode`	`true`	5.9.0
disabledInteraction	禁用高亮区域交互	`boolean`	`false`	5.13.0
gap	控制高亮区域的圆角边框和显示间距	`{ offset?: number | [number, number]; radius?: number }`	`{ offset?: 6 ; radius?: 2 }`	5.0.0 (数组类型的 `offset`: 5.9.0)
placement	引导卡片相对于目标元素的位置	`center` `left` `leftTop` `leftBottom` `right` `rightTop` `rightBottom` `top` `topLeft` `topRight` `bottom` `bottomLeft` `bottomRight`	`bottom`	
onClose	关闭引导时的回调函数	`Function`	-	
onFinish	引导完成时的回调	`Function`	-	
mask	是否启用蒙层，也可传入配置改变蒙层样式和填充色	`boolean | { style?: React.CSSProperties; color?: string; }`	`true`	

type	类型，影响底色与文字颜色	default | primary	default	
open	打开引导	boolean	-	
onChange	步骤改变时的回调，current 为当前的步骤	(current: number) => void	-	
current	当前处于哪一步	number	-	
scrollIntoViewOptions	是否支持当前元素滚动到视窗内，也可传入配置指定滚动视窗的相关参数	boolean | ScrollIntoViewOptions	true	5.2.0
indicatorsRender	自定义指示器	(current: number, total: number) => ReactNode	-	5.2.0
zIndex	Tour 的层级	number	1001	5.3.0
getPopupContainer	设置 Tour 浮层的渲染节点，默认是 body	(node: HTMLElement) => HTMLElement	body	5.12.0

TourStep 引导步骤卡片

属性	说明	类型	默认值
target	获取引导卡片指向的元素，为空时居中于屏幕	() => HTMLElement | HTMLElement	-
arrow	是否显示箭头，包含是否指向元素中心的配置	boolean | { pointAtCenter: boolean}	true

closeIcon	自定义关闭按钮	`React.ReactNode`	`true`
cover	展示的图片或者视频	`ReactNode`	-
title	标题	`ReactNode`	-
description	主要描述部分	`ReactNode`	-
placement	引导卡片相对于目标元素的位置	`center` `left` `leftTop` `leftBottom` `right` `rightTop` `rightBottom` `top` `topLeft` `topRight` `bottom` `bottomLeft` `bottomRight`	`bottom`
onClose	关闭引导时的回调函数	`Function`	-
mask	是否启用蒙层，也可传入配置改变蒙层样式和填充色，默认跟随 Tour 的 `mask` 属性	`boolean | { style?: React.CSSProperties; color?: string; }`	`true`
type	类型，影响底色与文字颜色	`default` | `primary`	`default`
nextButtonProps	下一步按钮的属性	`{ children: ReactNode; onClick: Function }`	-
prevButtonProps	上一步按钮的属性	`{ children: ReactNode; onClick: Function }`	-
scrollIntoViewOptions	是否支持当前元素滚动到视窗内，也可传入配置指定滚动视窗的相关参数，默认跟随 Tour 的 `scrollIntoViewOptions` 属性	`boolean | ScrollIntoViewOptions`	`true`

主题变量（Design Token）