

## 何时使用 {#when-to-use}

目标元素的操作需要用户进一步的确认时，在目标元素附近弹出浮层提示，询问用户。

和 `confirm` 弹出的全屏居中模态对话框相比，交互形式更轻量。

## 代码演示

### 基本

```
import React from 'react';
import type { PopconfirmProps } from 'antd';
import { Button, message, Popconfirm } from 'antd';

const confirm: PopconfirmProps['onConfirm'] = (e) => {
  console.log(e);
  message.success('Click on Yes');
};

const cancel: PopconfirmProps['onCancel'] = (e) => {
  console.log(e);
  message.error('Click on No');
};

const App: React.FC = () => (
  <Popconfirm
    title="Delete the task"
    description="Are you sure to delete this task?"
    onConfirm={confirm}
    onCancel={cancel}
    okText="Yes"
    cancelText="No"
  >
    <Button danger>Delete</Button>
  </Popconfirm>
);

export default App;
```

### 国际化

```
import React from 'react';
import { Button, Popconfirm } from 'antd';

const App: React.FC = () => (
  <Popconfirm
```

```

    title="Delete the task"
    description="Are you sure to delete this task?"
    okText="Yes"
    cancelText="No"
  >
    <Button danger>Delete</Button>
  </Popconfirm>
);

export default App;

```

## 位置

```

import React from 'react';
import { Button, ConfigProvider, Flex, Popconfirm } from 'antd';

const text = 'Are you sure to delete this task?';
const description = 'Delete the task';
const buttonWidth = 80;

const App: React.FC = () => (
  <ConfigProvider button={{ style: { width: buttonWidth, margin: 4 } }}>
    <Flex vertical justify="center" align="center" className="demo">
      <Flex justify="center" align="center" style={{ whiteSpace: 'nowrap'
    >>
        <Popconfirm
          placement="topLeft"
          title={text}
          description={description}
          okText="Yes"
          cancelText="No"
        >
          <Button>TL</Button>
        </Popconfirm>
        <Popconfirm
          placement="top"
          title={text}
          description={description}
          okText="Yes"
          cancelText="No"
        >
          <Button>Top</Button>
        </Popconfirm>
        <Popconfirm
          placement="topRight"
          title={text}

```

```

        description={description}
        okText="Yes"
        cancelText="No"
    >
    <Button>TR</Button>
</Popconfirm>
</Flex>
<Flex style={{ width: buttonWidth * 5 + 32 }} justify="space-between"
align="center">
    <Flex align="center" vertical>
        <Popconfirm
            placement="leftTop"
            title={text}
            description={description}
            okText="Yes"
            cancelText="No"
        >
            <Button>LT</Button>
        </Popconfirm>
        <Popconfirm
            placement="left"
            title={text}
            description={description}
            okText="Yes"
            cancelText="No"
        >
            <Button>Left</Button>
        </Popconfirm>
        <Popconfirm
            placement="leftBottom"
            title={text}
            description={description}
            okText="Yes"
            cancelText="No"
        >
            <Button>LB</Button>
        </Popconfirm>
    </Flex>
    <Flex align="center" vertical>
        <Popconfirm
            placement="rightTop"
            title={text}
            description={description}
            okText="Yes"
            cancelText="No"
        >

```

```

        <Button>RT</Button>
    </Popconfirm>
    <Popconfirm
        placement="right"
        title={text}
        description={description}
        okText="Yes"
        cancelText="No"
    >
        <Button>Right</Button>
    </Popconfirm>
    <Popconfirm
        placement="rightBottom"
        title={text}
        description={description}
        okText="Yes"
        cancelText="No"
    >
        <Button>RB</Button>
    </Popconfirm>
</Flex>
</Flex>
<Flex justify="center" align="center" style={{ whiteSpace: 'nowrap'
}}>

    <Popconfirm
        placement="bottomLeft"
        title={text}
        description={description}
        okText="Yes"
        cancelText="No"
    >
        <Button>BL</Button>
    </Popconfirm>
    <Popconfirm
        placement="bottom"
        title={text}
        description={description}
        okText="Yes"
        cancelText="No"
    >
        <Button>Bottom</Button>
    </Popconfirm>
    <Popconfirm
        placement="bottomRight"
        title={text}
        description={description}

```

```

        okText="Yes"
        cancelText="No"
      >
        <Button>BR</Button>
      </Popconfirm>
    </Flex>
  </Flex>
</ConfigProvider>
);

export default App;

```

## 贴边偏移

```

import React from 'react';
import { Button, Popconfirm } from 'antd';

const style: React.CSSProperties = {
  width: '300vw',
  height: '300vh',
  display: 'flex',
  alignItems: 'center',
  justifyContent: 'center',
};

const App: React.FC = () => {
  React.useEffect(() => {
    document.documentElement.scrollTop =
document.documentElement.clientHeight;
    document.documentElement.scrollLeft =
document.documentElement.clientWidth;
  }, []);
  return (
    <div style={style}>
      <Popconfirm title="Thanks for using antd. Have a nice day !" open>
        <Button type="primary">Scroll The Window</Button>
      </Popconfirm>
    </div>
  );
};

export default App;

```

## 条件触发

```

import React, { useState } from 'react';
import { Button, message, Popconfirm, Switch } from 'antd';

const App: React.FC = () => {
  const [open, setOpen] = useState(false);
  const [condition, setCondition] = useState(true);

  const changeCondition = (checked: boolean) => {
    setCondition(checked);
  };

  const confirm = () => {
    setOpen(false);
    message.success('Next step.');
```

```

    >
    <Button danger>Delete a task</Button>
  </Popconfirm>
  <br />
  <br />
  Whether directly execute:
  <Switch defaultChecked onChange={changeCondition} />
</div>
);
};

export default App;

```

## 自定义 Icon 图标

```

import React from 'react';
import { QuestionCircleOutlined } from '@ant-design/icons';
import { Button, Popconfirm } from 'antd';

const App: React.FC = () => (
  <Popconfirm
    title="Delete the task"
    description="Are you sure to delete this task?"
    icon={<QuestionCircleOutlined style={{ color: 'red' }} />}
  >
    <Button danger>Delete</Button>
  </Popconfirm>
);

export default App;

```

## 异步关闭

```

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

const App: React.FC = () => {
  const [open, setOpen] = useState(false);
  const [confirmLoading, setConfirmLoading] = useState(false);

  const showPopconfirm = () => {
    setOpen(true);
  };

  const handleOk = () => {

```

```

    setConfirmLoading(true);

    setTimeout(() => {
      setOpen(false);
      setConfirmLoading(false);
    }, 2000);
  };

  const handleCancel = () => {
    console.log('Clicked cancel button');
    setOpen(false);
  };

  return (
    <Popconfirm
      title="Title"
      description="Open Popconfirm with async logic"
      open={open}
      onConfirm={handleOk}
      okButtonProps={{ loading: confirmLoading }}
      onCancel={handleCancel}
    >
      <Button type="primary" onClick={showPopconfirm}>
        Open Popconfirm with async logic
      </Button>
    </Popconfirm>
  );
};

export default App;

```

## 基于 Promise 的异步关闭

```

import React from 'react';
import { Button, Popconfirm } from 'antd';

const App: React.FC = () => {
  const confirm = () =>
    new Promise((resolve) => {
      setTimeout(() => resolve(null), 3000);
    });

  return (
    <Popconfirm
      title="Title"
      description="Open Popconfirm with Promise"
    >

```



```

    onConfirm={confirm}
    onOpenChange={() => console.log('open change')}
  >
    <Button type="primary">Open Popconfirm with Promise</Button>
  </Popconfirm>
);
};

export default App;

```

## **\_InternalPanelDoNotUseOrYouWillBeFired**

Debug

```

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

const { _InternalPanelDoNotUseOrYouWillBeFired: InternalPopconfirm } =
  Popconfirm;

const App: React.FC = () => (
  <>
    <InternalPopconfirm title="Are you OK?" description="Does this look
good?" />
    <InternalPopconfirm
      title="Are you OK?"
      description="Does this look good?"
      placement="bottomRight"
      style={{ width: 250 }}
    />
    <InternalPopconfirm icon={null} title="Are you OK?" />
    <InternalPopconfirm icon={null} title="Are you OK?" description="Does
this look good?" />
  </>
);

export default App;

```

## **线框风格**

Debug

```

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

const { _InternalPanelDoNotUseOrYouWillBeFired: InternalPopconfirm } =

```

```
Popconfirm;

const App: React.FC = () => (
  <ConfigProvider theme={{ token: { wireframe: true } }}>
    <InternalPopconfirm title="Are you OK?" />
    <InternalPopconfirm title="Are you OK?" placement="bottomRight" style=
  {{ width: 250 }} />
  </ConfigProvider>
);

export default App;
```

API

通用属性参考: [通用属性](#)

参数	说明	类型	默认值	版本
cancelButtonProps	cancel 按钮 props	<a href="#">ButtonProps</a>	-	
cancelText	取消按钮文字	string	取消	
disabled	阻止点击 Popconfirm 子元素时弹出确认框	boolean	false	
icon	自定义弹出气泡 Icon 图标	ReactNode	<ExclamationCircle />	
okButtonProps	ok 按钮 props	<a href="#">ButtonProps</a>	-	
okText	确认按钮文字	string	确定	
okType	确认按钮类型	string	primary	
showCancel	是否显示取消按钮	boolean	true	4.18.0
title	确认框标题	ReactNode   () => ReactNode	-	
description	确认内容的详细描述	ReactNode   () => ReactNode	-	5.1.0
onCancel	点击取消的回调	function(e)	-	
onConfirm	点击确认的回调	function(e)	-	
onPopupClick	弹出气泡点击事件	function(e)	-	5.5.0

# 404

The requested path could not be found

## Semantic DOM

### 演示

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

import SemanticPreview from '../../../dumi/components/SemanticPreview';
import useLocale from '../../../dumi/hooks/useLocale';

const locales = {
  cn: {
    root: '根元素（包含箭头、内容元素）',
    body: '内容元素',
  },
  en: {
    root: 'Root element (including arrows, content elements)',
    body: 'Body element',
  },
};

const BlockList: React.FC<React.PropsWithChildren> = (props: any) => {
  const divRef = React.useRef<HTMLDivElement>(null);

  return (
    <div ref={divRef} style={{ position: 'absolute', marginTop: 60 }}>
      <Popconfirm
        title="prompt text"
        open
        placement="top"
        autoAdjustOverflow={false}
        getPopupContainer={() => divRef.current}
        {...props}
      />
    </div>
  );
};
```

```
const App: React.FC = () => {
  const [locale] = useLocale(locales);
  return (
    <SemanticPreview
      componentName="Popconfirm"
      semantics={[
        { name: 'root', desc: locale.root, version: '5.23.0' },
        { name: 'body', desc: locale.body, version: '5.23.0' },
      ]}
    >
      <BlockList />
    </SemanticPreview>
  );
};

export default App;
```

## 主题变量 (Design Token)

## FAQ

# 404

---

The requested path could not be found

更多问题，请参考 [Tooltip FAQ](#)。