

zh-CN

提供可编辑和可复制等额外的交互能力。

en-US

Provide additional interactive capacity of editable and copyable.

```
import React, { useState } from 'react';
import { Checkbox, Radio, Typography, Divider } from 'antd';
import { CheckOutlined, HighlightOutlined, SmileOutlined, SmileFilled } from '@ant-
design/icons';

const { Paragraph } = Typography;

const Demo = () => {
  const [editableStr, setEditableStr] = useState('This is an editable text.');
```

const [customIconStr, setCustomIconStr] = useState('Custom Edit icon and replace tooltip text.');

const [clickTriggerStr, setClickTriggerStr] = useState('Text or icon as trigger - click to start editing.');

const [chooseTrigger, setChooseTrigger] = useState('icon');

const [customEnterIconStr, setCustomEnterIconStr] = useState('Editable text with a custom enter icon in edit field.');

const [noEnterIconStr, setNoEnterIconStr] = useState('Editable text with no enter icon in edit field.');

const [hideTooltipStr, setHideTooltipStr] = useState('Hide Edit tooltip.');

const [lengthLimitedStr, setLengthLimitedStr] = useState('This is an editable text with limited length.');

const radioToState = input => {

switch (input) {

case 'text':

return ['text'];

case 'both':

return ['icon', 'text'];

case 'icon':

default:

return ['icon'];

}

};

const stateToRadio = () => {

if (chooseTrigger.indexOf('text') !== -1) {

return chooseTrigger.indexOf('icon') !== -1 ? 'both' : 'text';

}

return 'icon';

```

};

return (
  <>
    <Paragraph editable={{ onChange: setEditableStr }}>{editableStr}</Paragraph>
    <Paragraph
      editable={{
        icon: <HighlightOutlined />,
        tooltip: 'click to edit text',
        onChange: setCustomIconStr,
      }}
    >
      {customIconStr}
    </Paragraph>
    Trigger edit with:{' '}
    <Radio.Group
      onChange={e => setChooseTrigger(radioToState(e.target.value))}
      value={stateToRadio()}
    >
      <Radio value="icon">icon</Radio>
      <Radio value="text">text</Radio>
      <Radio value="both">both</Radio>
    </Radio.Group>
    <Paragraph
      editable={{
        tooltip: 'click to edit text',
        onChange: setClickTriggerStr,
        triggerType: chooseTrigger,
      }}
    >
      {clickTriggerStr}
    </Paragraph>
    <Paragraph
      editable={{
        icon: <HighlightOutlined />,
        tooltip: 'click to edit text',
        onChange: setCustomEnterIconStr,
        enterIcon: <CheckOutlined />,
      }}
    >
      {customEnterIconStr}
    </Paragraph>
    <Paragraph
      editable={{
        icon: <HighlightOutlined />,
        tooltip: 'click to edit text',
        onChange: setNoEnterIconStr,
        enterIcon: null,
      }}
    >
      {noEnterIconStr}
    </Paragraph>
  </>
);

```

```

<Paragraph editable={{ tooltip: false, onChange: setHideTooltipStr }}>
  {hideTooltipStr}
</Paragraph>
<Paragraph
  editable={{
    onChange: setLengthLimitedStr,
    maxLength: 50,
    autoSize: { maxRows: 5, minRows: 3 },
  }}
>
  {lengthLimitedStr}
</Paragraph>
<Typography.Title editable level={1} style={{ margin: 0 }}>
  h1. Ant Design
</Typography.Title>
<Typography.Title editable level={2} style={{ margin: 0 }}>
  h2. Ant Design
</Typography.Title>
<Typography.Title editable level={3} style={{ margin: 0 }}>
  h3. Ant Design
</Typography.Title>
<Typography.Title editable level={4} style={{ margin: 0 }}>
  h4. Ant Design
</Typography.Title>
<Typography.Title editable level={5} style={{ margin: 0 }}>
  h5. Ant Design
</Typography.Title>
<Divider />
<Paragraph copyable>This is a copyable text.</Paragraph>
<Paragraph copyable={{ text: 'Hello, Ant Design!' }}>Replace copy text.
</Paragraph>
<Paragraph
  copyable={{
    icon: [<SmileOutlined key="copy-icon" />, <SmileFilled key="copied-icon"
/>],
    tooltips: ['click here', 'you clicked!!'],
  }}
>
  Custom Copy icon and replace tooltips text.
</Paragraph>
<Paragraph copyable={{ tooltips: false }}>Hide Copy tooltips.</Paragraph>
</>
);
};

export default () => <Demo />;

```