何时使用 {#when-to-use}

- 当需要展示标题、段落、列表内容时使用，如文章/博客/日志的文本样式。
- 当需要一列基于文本的基础操作时，如拷贝/省略/可编辑。

代码演示

基本

```
import React from 'react';
import { Divider, Typography } from 'antd';

const { Title, Paragraph, Text, Link } = Typography;

const blockContent = `AntV 是蚂蚁集团全新一代数据可视化解决方案，致力于提供一套简单方便、专业可靠、不限可能的数据可视化最佳实践。得益于丰富的业务场景和用户需求挑战，AntV 经历多年积累与不断打磨，已支撑整个阿里集团内外 20000+ 业务系统，通过了日均千万级 UV 产品的严苛考验。
我们正在基础图表，图分析，图编辑，地理空间可视化，智能可视化等各个可视化的领域耕耘，欢迎同路人一起前行。`;

const App: React.FC = () => (
  <Typography>
    <Title>Introduction</Title>

    <Paragraph>
      In the process of internal desktop applications development, many different design specs and
      implementations would be involved, which might cause designers and developers difficulties and
      duplication and reduce the efficiency of development.
    </Paragraph>

    <Paragraph>
      After massive project practice and summaries, Ant Design, a design language for background
      applications, is refined by Ant UED Team, which aims to{' '}
      <Text strong>
        uniform the user interface specs for internal background projects, lower the unnecessary
        cost of design differences and implementation and liberate the resources of design and
        front-end development
      </Text>
      .
    </Paragraph>
    <Title level={2}>Guidelines and Resources</Title>

    <Paragraph>
      We supply a series of design principles, practical patterns and high
quality design resources
      (<Text code>Sketch</Text> and <Text code>Axure</Text>), to help
people create their product
      prototypes beautifully and efficiently.
    </Paragraph>

    <Paragraph>
      <ul>
        <li>
          <Link href="/docs/spec/proximity">Principles</Link>
        </li>
        <li>
          <Link href="/docs/spec/overview">Patterns</Link>
        </li>
        <li>
          <Link href="/docs/resources">Resource Download</Link>
        </li>
      </ul>
    </Paragraph>

    <Paragraph>
      Press <Text keyboard>Esc</Text> to exit...
    </Paragraph>

    <Divider />

    <Title>介绍</Title>

    <Paragraph>
      蚂蚁的企业级产品是一个庞大且复杂的体系。这类产品不仅量级巨大且功能复杂，而且变动和
并发频繁，常常需要设计与开发能够快速的做出响应。同时这类产品中有存在很多类似的页面以及组
件，可以通过抽象得到一些稳定且高复用性的内容。
    </Paragraph>

    <Paragraph>
      随着商业化的趋势，越来越多的企业级产品对更好的用户体验有了进一步的要求。带着这样的
一个终极目标，我们（蚂蚁集团体验技术部）经过大量的项目实践和总结，逐步打磨出一个服务于企业
级产品的设计体系
      Ant Design。基于<Text mark>『确定』和『自然』</Text>
      的设计价值观，通过模块化的解决方案，降低冗余的生产成本，让设计者专注于
      <Text strong>更好的用户体验</Text>。
    </Paragraph>
    <Title level={2}>设计资源</Title>

    <Paragraph>
      我们提供完善的设计原则、最佳实践和设计资源文件（<Text code>Sketch</Text> 和
      <Text code>Axure</Text>），来帮助业务快速设计出高质量的产品原型。
    </Paragraph>

    <Paragraph>
      <ul>
        <li>
          <Link href="/docs/spec/proximity-cn">设计原则</Link>
        </li>
        <li>
          <Link href="/docs/spec/overview-cn">设计模式</Link>
        </li>
        <li>
          <Link href="/docs/resources-cn">设计资源</Link>
        </li>
      </ul>
    </Paragraph>

    <Paragraph>
      <blockquote>{blockContent}</blockquote>
      <pre>{blockContent}</pre>
    </Paragraph>

    <Paragraph>
      按<Text keyboard>Esc</Text>键退出阅读……
    </Paragraph>
  </Typography>
);

export default App;
```

标题组件

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

const { Title } = Typography;

const App: React.FC = () => (
  <>
    <Title>h1. Ant Design</Title>
    <Title level={2}>h2. Ant Design</Title>
    <Title level={3}>h3. Ant Design</Title>
    <Title level={4}>h4. Ant Design</Title>
    <Title level={5}>h5. Ant Design</Title>
  </>
);

export default App;
```

标题与段落

Debug

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

const { Title, Paragraph, Text } = Typography;

const App: React.FC = () => (
  <>
    <Title>Introduction</Title>
    <Paragraph>
      In the process of internal desktop applications development, many
different design specs and
      implementations would be involved, which might cause designers and
developers difficulties and
      duplication and reduce the efficiency of development.
    </Paragraph>
    <Paragraph>
      After massive project practice and summaries, Ant Design, a design
language for background
      applications, is refined by Ant UED Team, which aims to
      <Text strong>
        uniform the user interface specs for internal background projects,
lower the unnecessary
        cost of design differences and implementation and liberate the
resources of design and
        front-end development
      </Text>
      .
    </Paragraph>
    <Title level={2}>Guidelines and Resources</Title>
    <Paragraph>
      We supply a series of design principles, practical patterns and high
quality design resources
      (<Text code>Sketch</Text> and <Text code>Axure</Text>), to help
people create their product
```

```
      prototypes beautifully and efficiently.
    </Paragraph>

    <Paragraph>
      <ul>
        <li>
          <a href="/docs/spec/proximity">Principles</a>
        </li>
        <li>
          <a href="/docs/pattern/navigation">Patterns</a>
        </li>
        <li>
          <a href="/docs/resource/download">Resource Download</a>
        </li>
      </ul>
    </Paragraph>

    <Title id="intro">介绍</Title>
    <Paragraph>
      蚂蚁的企业级产品是一个庞大且复杂的体系。这类产品不仅量级巨大且功能复杂，而且变动和
并发频繁，常常需要设计与开发能够快速的做出响应。同时这类产品中有存在很多类似的页面以及组
件，可以通过抽象得到一些稳定且高复用性的内容。
    </Paragraph>
    <Paragraph>
      随着商业化的趋势，越来越多的企业级产品对更好的用户体验有了进一步的要求。带着这样的
一个终极目标，我们（蚂蚁集团体验技术部）经过大量的项目实践和总结，逐步打磨出一个服务于企业
级产品的设计体系
      Ant Design。基于<Text mark>『确定』和『自然』</Text>
      的设计价值观，通过模块化的解决方案，降低冗余的生产成本，让设计者专注于
      <Text strong>更好的用户体验</Text>。
    </Paragraph>
    <Title level={2}>设计资源</Title>
    <Paragraph>
      我们提供完善的设计原则、最佳实践和设计资源文件（<Text code>Sketch</Text> 和
      <Text code>Axure</Text>），来帮助业务快速设计出高质量的产品原型。
    </Paragraph>

    <Paragraph>
      <ul>
        <li>
          <a href="/docs/spec/proximity">设计原则</a>
        </li>
        <li>
          <a href="/docs/pattern/navigation">设计模式</a>
        </li>
        <li>
```

```
          <a href="/docs/resource/download">设计资源</a>
        </li>
      </ul>
    </Paragraph>

    <Paragraph>
      <ul>
        <li>I am an unordered item</li>
        <li>
          I am an unordered item with an ordered sublist
          <ol>
            <li>I am ordered</li>
          </ol>
          <ul>
            <li>I am unordered</li>
          </ul>
        </li>
      </ul>
      <ol>
        <li>
          Ordered list item with unordered sublist
          <ul>
            <li>I am unordered!</li>
            <li>I am also unordered!</li>
          </ul>
        </li>
      </ol>
    </Paragraph>
  </>
);

export default App;
```

文本与超链接组件

```
import React from 'react';
import { Space, Typography } from 'antd';

const { Text, Link } = Typography;

const App: React.FC = () => (
  <Space direction="vertical">
    <Text>Ant Design (default)</Text>
    <Text type="secondary">Ant Design (secondary)</Text>
    <Text type="success">Ant Design (success)</Text>
    <Text type="warning">Ant Design (warning)</Text>
    <Text type="danger">Ant Design (danger)</Text>
    <Text disabled>Ant Design (disabled)</Text>
    <Text mark>Ant Design (mark)</Text>
    <Text code>Ant Design (code)</Text>
    <Text keyboard>Ant Design (keyboard)</Text>
    <Text underline>Ant Design (underline)</Text>
    <Text delete>Ant Design (delete)</Text>
    <Text strong>Ant Design (strong)</Text>
    <Text italic>Ant Design (italic)</Text>
    <Link href="https://ant.design" target="_blank">
      Ant Design (Link)
    </Link>
  </Space>
);

export default App;
```

可编辑

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

const { Paragraph } = Typography;

const App: React.FC = () => {
  const [editableStr, setEditableStr] = useState('This is an editable
text.');
  const [editableStrWithSuffix, setEditableStrWithSuffix] = useState(
    'This is a loooooooooooooooooooooooooooooooong editable text with
suffix.',
  );
  const [editableStrWithSuffixStartPart, editableStrWithSuffixSuffixPart] =
useMemo(
    () => [editableStrWithSuffix.slice(0, -12),
editableStrWithSuffix.slice(-12)],
    [editableStrWithSuffix],
  );
  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' | 'text')[]>
(['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: string): ('icon' | 'text')[] => {
    switch (input) {
      case 'text':
        return ['text'];
      case 'both':
        return ['icon', 'text'];
      case 'icon':
        return ['icon'];
      default:
        return ['icon'];
    }
  };

  const stateToRadio = useMemo<string>(() => {
    if (chooseTrigger.includes('text')) {
      return chooseTrigger.includes('icon') ? 'both' : 'text';
    }
    return 'icon';
  }, [chooseTrigger]);

  return (
    <>
      <Paragraph editable={{ onChange: setEditableStr }}>{editableStr}
</Paragraph>
      <Paragraph
        editable={{
          onChange: setEditableStrWithSuffix,
          text: editableStrWithSuffix,
        }}
        ellipsis={{
          suffix: editableStrWithSuffixSuffixPart,
        }}
      >
        {editableStrWithSuffixStartPart}
      </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>
    </>
  );
};

export default App;
```

可复制

```
import React from 'react';
import { SmileFilled, SmileOutlined } from '@ant-design/icons';
import { Typography } from 'antd';

const { Paragraph, Text } = Typography;

const App: React.FC = () => (
  <>
    <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>
    <Paragraph
      copyable={{
        text: async () =>
          new Promise((resolve) => {
            setTimeout(() => {
              resolve('Request text');
            }, 500);
          }),
      }}
    >
      Request copy text.
    </Paragraph>
    <Text copyable={{ text: 'text to be copied' }} />
  </>
);

export default App;
```

省略号

```
import React, { useState } from 'react';
import { Switch, Typography } from 'antd';

const { Paragraph, Text } = Typography;

const App: React.FC = () => {
  const [ellipsis, setEllipsis] = useState(true);

  return (
    <>
      <Switch
        checked={ellipsis}
        onChange={() => {
          setEllipsis(!ellipsis);
        }}
      />

      <Paragraph ellipsis={ellipsis}>
        Ant Design, a design language for background applications, is
refined by Ant UED Team. Ant
        Design, a design language for background applications, is refined
by Ant UED Team. Ant
        Design, a design language for background applications, is refined
by Ant UED Team. Ant
        Design, a design language for background applications, is refined
by Ant UED Team. Ant
        Design, a design language for background applications, is refined
by Ant UED Team. Ant
        Design, a design language for background applications, is refined
by Ant UED Team.
      </Paragraph>

      <Paragraph ellipsis={ellipsis ? { rows: 2, expandable: true, symbol:
'more' } : false}>
        Ant Design, a design language for background applications, is
refined by Ant UED Team. Ant
        Design, a design language for background applications, is refined
by Ant UED Team. Ant
        Design, a design language for background applications, is refined
by Ant UED Team. Ant
        Design, a design language for background applications, is refined
by Ant UED Team. Ant
        Design, a design language for background applications, is refined
by Ant UED Team. Ant
        Design, a design language for background applications, is refined
by Ant UED Team.
      </Paragraph>

      <Text
        style={ellipsis ? { width: 200 } : undefined}
        ellipsis={ellipsis ? { tooltip: 'I am ellipsis now!' } : false}
      >
        Ant Design, a design language for background applications, is
refined by Ant UED Team.
      </Text>

      <Text
        code
        style={ellipsis ? { width: 200 } : undefined}
        ellipsis={ellipsis ? { tooltip: 'I am ellipsis now!' } : false}
      >
        Ant Design, a design language for background applications, is
refined by Ant UED Team.
      </Text>
    </>
  );
};

export default App;
```

受控省略展开/收起

v5.16.0

```
import React, { useState } from 'react';
import { Flex, Slider, Switch, Typography } from 'antd';

const App = () => {
  const [rows, setRows] = useState(2);
  const [expanded, setExpanded] = useState(false);

  return (
    <Flex gap={16} vertical>
      <Flex gap={16} align="center">
        <Switch
          checked={expanded}
          onChange={() => setExpanded((c) => !c)}
          style={{ flex: 'none' }}
        />
        <Slider min={1} max={20} value={rows} onChange={setRows} style={{
flex: 'auto' }} />
      </Flex>

      <Typography.Paragraph
        ellipsis={{
          rows,
          expandable: 'collapsible',
          expanded,
          onExpand: (_, info) => setExpanded(info.expanded),
        }}
        copyable
      >
        {'Ant Design, a design language for background applications, is
refined by Ant UED Team.'.repeat(
          20,
        )}
      </Typography.Paragraph>
    </Flex>
  );
};

export default App;
```

省略中间

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

const { Text } = Typography;

const EllipsisMiddle: React.FC<{ suffixCount: number; children: string }> =
({
  suffixCount,
  children,
}) => {
  const start = children.slice(0, children.length - suffixCount);
  const suffix = children.slice(-suffixCount).trim();
  return (
    <Text style={{ maxWidth: '100%' }} ellipsis={{ suffix }}>
      {start}
    </Text>
  );
};

const App: React.FC = () => (
  <EllipsisMiddle suffixCount={12}>
    In the process of internal desktop applications development, many
different design specs and
    implementations would be involved, which might cause designers and
developers difficulties and
    duplication and reduce the efficiency of development.
  </EllipsisMiddle>
);

export default App;
```

省略号 Debug

Debug

```
import React, { useState } from 'react';
import { Button, Slider, Switch, Typography } from 'antd';

const { Text, Paragraph } = Typography;

const templateStr =
  'In the process of internal desktop applications development, many
different design specs and implementations would be involved, which might
cause designers and developers difficulties and duplication and reduce the
efficiency of development.';

const text = `this is a multiline
  text that has many
  lines and
    - render like this
    - and this

  and that`;

const App: React.FC = () => {
  const [rows, setRows] = useState(1);
  const [longText, setLongText] = useState(true);
  const [copyable, setCopyable] = useState(false);
  const [editable, setEditable] = useState(false);
  const [expandable, setExpandable] = useState(false);
  const [display, setDisplay] = useState('none');

  React.useEffect(() => {
    setTimeout(() => {
      setDisplay('block');
    }, 100);
  }, []);

  return (
    <>
      <Switch checked={longText} checkedChildren="Long Text" onChange=
{setLongText} />
      <Switch checked={copyable} onChange={setCopyable} />
      <Switch checked={editable} onChange={setEditable} />
      <Switch checked={expandable} onChange={setExpandable} />
      <Slider value={rows} min={1} max={10} onChange={setRows} />
      {longText ? (
        <Paragraph ellipsis={{ rows, expandable }} copyable={copyable}
editable={editable}>
          Ant Design, a design language for background applications, is
refined by Ant UED Team.
```

```
          This is a nest sample{' '}
          <Text code strong delete>
            Test
          </Text>{' '}
          case. Bnt Design, a design language for background applications,
is refined by Ant UED
          Team. Cnt Design, a design language for background applications,
is refined by Ant UED
          Team. Dnt Design, a design language for background applications,
is refined by Ant UED
          Team. Ent Design, a design language for background applications,
is refined by Ant UED
          Team.
        </Paragraph>
      ) : (
        <Paragraph ellipsis={{ rows, expandable }} copyable={copyable}
editable={editable}>
          Hello World
        </Paragraph>
      )}

      <Text style={{ maxWidth: 400, fontSize: 24 }} copyable ellipsis>
        {templateStr}
      </Text>

      <br />

      <Text style={{ maxWidth: 400, fontSize: 12 }} copyable ellipsis>
        {templateStr}
      </Text>

      <br />

      <Text style={{ width: 400, fontSize: 24 }} copyable ellipsis>
        {templateStr}
      </Text>

      <br />

      <Text style={{ width: 100 }} ellipsis copyable>
        Ant Design is a design language for background applications, is
refined by Ant UED Team.
      </Text>

      <p>
        [Before]<Text ellipsis>not ellipsis</Text>[After]
      </p>

      <div style={{ display }}>
        <Text style={{ width: 100 }} ellipsis={{ tooltip: 'I am ellipsis
now!' }}>
          默认display none 样式的超长文字， 悬停tooltip失效了
        </Text>
      </div>

      <Typography.Paragraph
        style={{ width: 300 }}
        ellipsis={{
          rows: 3,
          expandable: true,
          symbol: <Button>Open</Button>,
        }}
      >
        {templateStr.slice(0, 60)}
        <span style={{ fontSize: '5em' }}>ANTD</span>
        {templateStr.slice(60)}
      </Typography.Paragraph>

      <pre>
        <Typography.Paragraph ellipsis={{ rows: 2, expandable: true }}>
{text}</Typography.Paragraph>
      </pre>

      <br />

      <Text style={{ width: 100, whiteSpace: 'nowrap' }} ellipsis copyable>
        {templateStr}
      </Text>
    </>
  );
};

export default App;
```

后缀

```
import React, { useState } from 'react';
import { Slider, Typography } from 'antd';

const { Paragraph } = Typography;

const App: React.FC = () => {
  const [rows, setRows] = useState(1);

  const article =
    "To be, or not to be, that is the question: Whether it is nobler in the
mind to suffer. The slings and arrows of outrageous fortune Or to take arms
against a sea of troubles, And by opposing end them? To die: to sleep; No
more; and by a sleep to say we end The heart-ache and the thousand natural
shocks That flesh is heir to, 'tis a consummation Devoutly to be wish'd. To
die, to sleep To sleep- perchance to dream: ay, there's the rub! For in
that sleep of death what dreams may come When we have shuffled off this
mortal coil, Must give us pause. There 's the respect That makes calamity
of so long life";

  return (
    <>
      <Slider value={rows} min={1} max={10} onChange={setRows} />
      <Paragraph
        ellipsis={{
          rows,
          expandable: true,
          suffix: '--William Shakespeare',
          onEllipsis: (ellipsis) => {
            console.log('Ellipsis changed:', ellipsis);
          },
        }}
        title={`${article}--William Shakespeare`}
      >
        {article}
      </Paragraph>
    </>
  );
};

export default App;
```

组件 Token

Debug

```
import React from 'react';
import { ConfigProvider, Divider, Typography } from 'antd';

const { Title, Paragraph, Text, Link } = Typography;

const blockContent = `AntV 是蚂蚁集团全新一代数据可视化解决方案，致力于提供一套简单方
便、专业可靠、不限可能的数据可视化最佳实践。得益于丰富的业务场景和用户需求挑战，AntV 经历
```

```
多年积累与不断打磨，已支撑整个阿里集团内外 20000+ 业务系统，通过了日均千万级 UV 产品的严
苛考验。
我们正在基础图表，图分析，图编辑，地理空间可视化，智能可视化等各个可视化的领域耕耘，欢迎同
路人一起前行。`;

const App: React.FC = () => (
  <ConfigProvider
    theme={{
      components: {
        Typography: {
          fontWeightStrong: 700,
          titleMarginTop: '2.4em',
          titleMarginBottom: '1em',
          colorSuccessText: '#FF0000',
          colorWarningText: '#00FF00',
          colorErrorText: '#0000FF',
        },
      },
    }}
  >
    <Typography>
      <Title>Introduction</Title>
      <Paragraph>
        After massive project practice and summaries, Ant Design, a design
language for background
        applications, is refined by Ant UED Team, which aims to{' '}
        <Text strong>
          uniform the user interface specs for internal background
projects, lower the unnecessary
          cost of design differences and implementation and liberate the
resources of design and
          front-end development
        </Text>
        .
      </Paragraph>
      <Title level={2}>Guidelines and Resources</Title>
      <Paragraph>
        We supply a series of design principles, practical patterns and
high quality design
        resources (<Text code>Sketch</Text> and <Text code>Axure</Text>),
to help people create
        their product prototypes beautifully and efficiently.
      </Paragraph>

      <Paragraph>
        <ul>
          <li>
            <Link href="/docs/spec/proximity">Principles</Link>
          </li>
          <li>
            <Link href="/docs/spec/overview">Patterns</Link>
          </li>
          <li>
            <Link href="/docs/resources">Resource Download</Link>
          </li>
        </ul>
      </Paragraph>

      <Paragraph>
        Press <Text keyboard>Esc</Text> to exit...
      </Paragraph>

      <Divider />

      <Title>介绍</Title>
      <Paragraph>
        随着商业化的趋势，越来越多的企业级产品对更好的用户体验有了进一步的要求。带着这样的一个终极目标，我们（蚂蚁集团体验技术部）经过大量的项目实践和总结，逐步打磨出一个服务于企业级产品的设计体系
        Ant Design。基于<Text mark>『确定』和『自然』</Text>
        的设计价值观，通过模块化的解决方案，降低冗余的生产成本，让设计者专注于
        <Text strong>更好的用户体验</Text>。
      </Paragraph>
      <Title level={2}>设计资源</Title>
      <Paragraph>
        我们提供完善的设计原则、最佳实践和设计资源文件（<Text code>Sketch</Text> 和
        <Text code>Axure</Text>），来帮助业务快速设计出高质量的产品原型。
      </Paragraph>

      <Paragraph>
        <ul>
          <li>
            <Link href="/docs/spec/proximity-cn">设计原则</Link>
          </li>
          <li>
            <Link href="/docs/spec/overview-cn">设计模式</Link>
          </li>
          <li>
            <Link href="/docs/resources-cn">设计资源</Link>
          </li>
        </ul>
      </Paragraph>
```

```
      <Paragraph>
        <blockquote>{blockContent}</blockquote>
        <pre>{blockContent}</pre>
      </Paragraph>

      <Paragraph>
        按<Text keyboard>Esc</Text>键退出阅读……
      </Paragraph>
    </Typography>

    <Typography.Text type="success">Success but red</Typography.Text>
    <Typography.Text type="warning">Warning but green</Typography.Text>
    <Typography.Text type="danger">Danger but blue</Typography.Text>
  </ConfigProvider>
);

export default App;
```

API

通用属性参考：通用属性

Typography.Text

参数	说明	类型	默认值	版本
code	添加代码样式	boolean	false	
copyable	是否可拷贝，为对象时可进行各种自定义	boolean | copyable	false	
delete	添加删除线样式	boolean	false	
disabled	禁用文本	boolean	false	
editable	是否可编辑，为对象时可对编辑进行控制	boolean | editable	false	
ellipsis	自动溢出省略，为对象时不能设置省略行数、是否可展开、onExpand 展开事件。不同于 Typography.Paragraph，Text 组件自身不带 100% 宽度样式，因而默认情况下初次缩略后宽度便不再变化。如果需要自适应宽度，请手动配置宽度样式	boolean | Omit<ellipsis, 'expandable' | 'rows' | 'onExpand'>	false	
keyboard	添加键盘样式	boolean	false	4.3.0

mark	添加标记样式	boolean	false	
onClick	点击 Text 时的回调	(event) => void	-	
strong	是否加粗	boolean	false	
italic	是否斜体	boolean	false	4.16.0
type	文本类型	secondary | success | warning | danger	-	success: 4.6.0
underline	添加下划线样式	boolean	false	

Typography.Title

参数	说明	类型	默认值	版本
code	添加代码样式	boolean	false	
copyable	是否可拷贝，为对象时可进行各种自定义	boolean | copyable	false	
delete	添加删除线样式	boolean	false	
disabled	禁用文本	boolean	false	
editable	是否可编辑，为对象时可对编辑进行控制	boolean | editable	false	
ellipsis	自动溢出省略，为对象时可设置省略行数、是否可展开、添加后缀等	boolean | ellipsis	false	
level	重要程度，相当于 h1、h2、h3、h4、h5	number: 1, 2, 3, 4, 5	1	5: 4.6.0
mark	添加标记样式	boolean	false	
onClick	点击 Title 时的回调	(event) => void	-	
italic	是否斜体	boolean	false	4.16.0
type	文本类型	secondary | success | warning | danger	-	success: 4.6.0
underline	添加下划线样式	boolean	false	

Typography.Paragraph

参数	说明	类型	默认值	版本
code	添加代码样式	boolean	false	
copyable	是否可拷贝，为对象时可进行各种自定义	boolean | copyable	false	
delete	添加删除线样式	boolean	false	
disabled	禁用文本	boolean	false	
editable	是否可编辑，为对象时可对编辑进行控制	boolean | editable	false	
ellipsis	自动溢出省略，为对象时可设置省略行数、是否可展开、添加后缀等	boolean | ellipsis	false	
mark	添加标记样式	boolean	false	
onClick	点击 Paragraph 时的回调	(event) => void	-	
strong	是否加粗	boolean	false	
italic	是否斜体	boolean	false	4.16.0
type	文本类型	`secondary` | `success` | `warning` | `danger`	-	success: 4.6.0
underline	添加下划线样式	boolean	false	

copyable

```
{
  text: string | (() => string | Promise<string>),
  onCopy: function(event),
  icon: ReactNode,
  tooltips: false | [ReactNode, ReactNode],
  format: 'text/plain' | 'text/html',
  tabIndex: number,
}
```

参数	说明	类型	默认值	版本
format	剪切板数据的 Mime Type	'text/plain' | 'text/html'	-	4.21.0
icon	自定义拷贝图标：[默认图标, 拷贝后的图标]	[ReactNode, ReactNode]	-	4.6.0

text	拷贝到剪切板里的文本	string	-	
tooltips	自定义提示文案，为 false 时隐藏文案	[ReactNode, ReactNode]	[复制, 复制成功]	4.4.0
onCopy	拷贝成功的回调函数	function	-	
tabIndex	自定义复制按钮的 tabIndex	number	0	5.17.0

editable

```
{
  icon: ReactNode,
  tooltip: ReactNode,
  editing: boolean,
  maxLength: number,
  autoSize: boolean | { minRows: number, maxRows: number },
  text: string,
  onChange: function(string),
  onCancel: function,
  onStart: function,
  onEnd: function,
  triggerType: ('icon' | 'text')[],
  enterIcon: ReactNode,
  tabIndex: number,
}
```

参数	说明	类型	默认值	版本
autoSize	自动 resize 文本域	boolean | { minRows: number, maxRows: number }	-	4.4.0
editing	控制是否是编辑中状态	boolean	false	
icon	自定义编辑图标	ReactNode	<EditOutlined />	4.6.0
maxLength	编辑中文本域最大长度	number	-	4.4.0
tooltip	自定义提示文本，为 false 时关闭	ReactNode	编辑	4.6.0
text	显式地指定编辑文案，为空时将隐式地使用 children	string	-	4.24.0
onChange	文本域编辑时触发	function(value: string)	-	

onCancel	按 ESC 退出编辑状态时触发	function	-	
onStart	进入编辑中状态时触发	function	-	
onEnd	按 ENTER 结束编辑状态时触发	function	-	4.14.0
triggerType	编辑模式触发器类型，图标、文本或者两者都设置（不设置图标作为触发器时它会隐藏）	Array<icon|text>	[icon]	
enterIcon	在编辑段中自定义“enter”图标（传递“null”将删除图标）	ReactNode	<EnterOutlined />	4.17.0
tabIndex	自定义编辑按钮的 tabIndex	number	0	5.17.0

ellipsis

```
interface EllipsisConfig {
  rows: number;
  /** `5.16.0` 新增 `collapsible` */
  expandable: boolean | 'collapsible';
  suffix: string;
  /** `5.16.0` 新增渲染函数 */
  symbol: ReactNode | ((expanded: boolean) => ReactNode);
  tooltip: ReactNode | TooltipProps;
  /** `5.16.0` 新增 */
  defaultExpanded: boolean;
  /** `5.16.0` 新增 */
  expanded: boolean;
  /** `5.16.0` 新增 `info` */
  onExpand: (event: MouseEvent, info: { expanded: boolean }) => void;
  onEllipsis: (ellipsis: boolean) => void;
}
```

参数	说明	类型	默认值	版本
expandable	是否可展开	boolean | 'collapsible'	-	collapsible: 5.16.0

rows	最多显示的行数	number	-	
suffix	自定义省略内容后缀	string	-	
symbol	自定义展开描述文案	ReactNode | ((expanded: boolean) => ReactNode)	展开收起	
tooltip	省略时，展示提示信息	ReactNode | TooltipProps	-	4.11.0
defaultExpanded	默认展开或收起	boolean		5.16.0
expanded	展开或收起	boolean		5.16.0
onEllipsis	触发省略时的回调	function(ellipsis)	-	4.2.0
onExpand	点击展开或收起时的回调	function(event, { expanded: boolean })	-	`info`: 5.16.0

主题变量（Design Token）

FAQ

Typography.Link 如何与 react-router 库集成?

`react-router` 支持自定义渲染组件：

```
<Link to="/" component={Typography.Link} />
```

注意： 这并不是和 react-router 的 Link 的执行逻辑等价 参考