何时使用 {#when-to-use}

在操作需要较长时间才能完成时，为用户显示该操作的当前进度和状态。

- 当一个操作会打断当前界面，或者需要在后台运行，且耗时可能超过 2 秒时；
- 当需要显示一个操作完成的百分比时。

代码演示

进度条

```
import React from 'react';
import { Flex, Progress } from 'antd';

const App: React.FC = () => (
  <Flex gap="small" vertical>
    <Progress percent={30} />
    <Progress percent={50} status="active" />
    <Progress percent={70} status="exception" />
    <Progress percent={100} />
    <Progress percent={50} showInfo={false} />
  </Flex>
);

export default App;
```

进度圈

```
import React from 'react';
import { Flex, Progress } from 'antd';

const App: React.FC = () => (
  <Flex gap="small" wrap>
    <Progress type="circle" percent={75} />
    <Progress type="circle" percent={70} status="exception" />
    <Progress type="circle" percent={100} />
  </Flex>
);

export default App;
```

小型进度条

```
import React from 'react';
import { Flex, Progress } from 'antd';
```

```
const App: React.FC = () => (
  <Flex vertical gap="small" style={{ width: 180 }}>
    <Progress percent={30} size="small" />
    <Progress percent={50} size="small" status="active" />
    <Progress percent={70} size="small" status="exception" />
    <Progress percent={100} size="small" />
  </Flex>
);

export default App;
```

响应式进度圈

```
import React from 'react';
import { Flex, Progress } from 'antd';

const App: React.FC = () => (
  <Flex align="center" gap="small">
    <Progress
      type="circle"
      trailColor="#e6f4ff"
      percent={60}
      strokeWidth={20}
      size={14}
      format={(number) => `进行中，已完成${number}%`}
    />
    <span>代码发布</span>
  </Flex>
);

export default App;
```

小型进度圈

```
import React from 'react';
import { Flex, Progress } from 'antd';

const App: React.FC = () => (
  <Flex wrap gap="small">
    <Progress type="circle" percent={30} size={80} />
    <Progress type="circle" percent={70} size={80} status="exception" />
    <Progress type="circle" percent={100} size={80} />
  </Flex>
);
```

```
export default App;
```

动态展示

```
import React, { useState } from 'react';
import { MinusOutlined, PlusOutlined } from '@ant-design/icons';
import { Button, Flex, Progress, Space } from 'antd';

const App: React.FC = () => {
  const [percent, setPercent] = useState<number>(0);

  const increase = () => {
    setPercent((prevPercent) => {
      const newPercent = prevPercent + 10;
      if (newPercent > 100) {
        return 100;
      }
      return newPercent;
    });
  };

  const decline = () => {
    setPercent((prevPercent) => {
      const newPercent = prevPercent - 10;
      if (newPercent < 0) {
        return 0;
      }
      return newPercent;
    });
  };

  return (
    <Flex vertical gap="small">
      <Flex vertical gap="small">
        <Progress percent={percent} type="line" />
        <Progress percent={percent} type="circle" />
      </Flex>
      <Space.Compact>
        <Button onClick={decline} icon={<MinusOutlined />} />
        <Button onClick={increase} icon={<PlusOutlined />} />
      </Space.Compact>
    </Flex>
  );
};
```

```
export default App;
```

自定义文字格式

```
import React from 'react';
import { Flex, Progress } from 'antd';

const App: React.FC = () => (
  <Flex gap="small" wrap>
    <Progress type="circle" percent={75} format={(percent) => `${percent} Days`} />
    <Progress type="circle" percent={100} format={() => 'Done'} />
  </Flex>
);

export default App;
```

仪表盘

```
import React from 'react';
import { Flex, Progress } from 'antd';

const App: React.FC = () => (
  <Flex gap="small" wrap>
    <Progress type="dashboard" percent={75} />
    <Progress type="dashboard" percent={75} gapDegree={30} />
  </Flex>
);

export default App;
```

分段进度条

```
import React from 'react';
import { Flex, Progress, Tooltip } from 'antd';

const App: React.FC = () => (
  <Flex gap="small" vertical>
    <Tooltip title="3 done / 3 in progress / 4 to do">
      <Progress percent={60} success={{ percent: 30 }} />
    </Tooltip>
    <Flex gap="small" wrap>
      <Tooltip title="3 done / 3 in progress / 4 to do">
```

```
        <Progress percent={60} success={{ percent: 30 }} type="circle" />
      </Tooltip>
      <Tooltip title="3 done / 3 in progress / 4 to do">
        <Progress percent={60} success={{ percent: 30 }} type="dashboard"
/>
      </Tooltip>
    </Flex>
  </Flex>
);

export default App;
```

边缘形状

```
import React from 'react';
import { Flex, Progress } from 'antd';

const App: React.FC = () => (
  <Flex vertical gap="small">
    <Progress strokeLinecap="butt" percent={75} />
    <Flex wrap gap="small">
      <Progress strokeLinecap="butt" type="circle" percent={75} />
      <Progress strokeLinecap="butt" type="dashboard" percent={75} />
    </Flex>
  </Flex>
);

export default App;
```

自定义进度条渐变色

```
import React from 'react';
import { Flex, Progress } from 'antd';
import type { ProgressProps } from 'antd';

const twoColors: ProgressProps['strokeColor'] = {
  '0%': '#108ee9',
  '100%': '#87d068',
};

const conicColors: ProgressProps['strokeColor'] = {
  '0%': '#87d068',
  '50%': '#ffe58f',
  '100%': '#ffccc7',
};
```

```
const App: React.FC = () => (
  <Flex vertical gap="middle">
    <Progress percent={99.9} strokeColor={twoColors} />
    <Progress percent={50} status="active" strokeColor={{ from: '#108ee9', to: '#87d068' }} />
    <Flex gap="small" wrap>
      <Progress type="circle" percent={90} strokeColor={twoColors} />
      <Progress type="circle" percent={100} strokeColor={twoColors} />
      <Progress type="circle" percent={93} strokeColor={conicColors} />
    </Flex>
    <Flex gap="small" wrap>
      <Progress type="dashboard" percent={90} strokeColor={twoColors} />
      <Progress type="dashboard" percent={100} strokeColor={twoColors} />
      <Progress type="dashboard" percent={93} strokeColor={conicColors} />
    </Flex>
  </Flex>
);

export default App;
```

步骤进度条

```
import React from 'react';
import { green, red } from '@ant-design/colors';
import { Flex, Progress } from 'antd';

const App: React.FC = () => (
  <Flex gap="small" vertical>
    <Progress percent={50} steps={3} />
    <Progress percent={30} steps={5} />
    <Progress percent={100} steps={5} size="small" strokeColor={green[6]} />
    <Progress percent={60} steps={5} strokeColor={[green[6], green[6], red[5]]} />
  </Flex>
);

export default App;
```

步骤进度圈

v5.16.0

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

```
const App: React.FC = () => {
  const [stepsCount, setStepsCount] = React.useState<number>(5);
  const [stepsGap, setStepsGap] = React.useState<number>(7);
  return (
    <>
      <Typography.Title level={5}>Custom count:</Typography.Title>
      <Slider min={2} max={10} value={stepsCount} onChange={setStepsCount}
/>
      <Typography.Title level={5}>Custom gap:</Typography.Title>
      <Slider step={4} min={0} max={40} value={stepsGap} onChange=
{setStepsGap} />
      <Flex wrap gap="middle" style={{ marginTop: 16 }}>
        <Progress
          type="dashboard"
          steps={8}
          percent={50}
          trailColor="rgba(0, 0, 0, 0.06)"
          strokeWidth={20}
        />
        <Progress
          type="circle"
          percent={100}
          steps={{ count: stepsCount, gap: stepsGap }}
          trailColor="rgba(0, 0, 0, 0.06)"
          strokeWidth={20}
        />
      </Flex>
    </>
  );
};

export default App;
```

尺寸

```
import React from 'react';
import { Flex, Progress } from 'antd';

const App: React.FC = () => (
  <Flex vertical gap="middle">
    <Flex vertical gap="small" style={{ width: 300 }}>
      <Progress percent={50} />
      <Progress percent={50} size="small" />
      <Progress percent={50} size={[300, 20]} />
    </Flex>
    <Flex align="center" wrap gap={30}>
      <Progress type="circle" percent={50} />
      <Progress type="circle" percent={50} size="small" />
      <Progress type="circle" percent={50} size={20} />
    </Flex>
    <Flex align="center" wrap gap={30}>
      <Progress type="dashboard" percent={50} />
      <Progress type="dashboard" percent={50} size="small" />
      <Progress type="dashboard" percent={50} size={20} />
    </Flex>
    <Flex align="center" wrap gap={30}>
      <Progress steps={3} percent={50} />
      <Progress steps={3} percent={50} size="small" />
      <Progress steps={3} percent={50} size={20} />
      <Progress steps={3} percent={50} size={[20, 30]} />
    </Flex>
  </Flex>
);

export default App;
```

改变进度数值位置

v5.18.0

```
import React from 'react';
import { Flex, Progress } from 'antd';

const App: React.FC = () => (
  <Flex gap="small" vertical>
    <Progress
      percent={0}
      percentPosition={{ align: 'center', type: 'inner' }}
      size={[200, 20]}
      strokeColor="#E6F4FF"
    />
    <Progress percent={10} percentPosition={{ align: 'center', type:
'inner' }} size={[300, 20]} />
    <Progress
      percent={50}
      percentPosition={{ align: 'start', type: 'inner' }}
      size={[300, 20]}
      strokeColor="#B7EB8F"
    />
    <Progress
      percent={60}
      percentPosition={{ align: 'end', type: 'inner' }}
      size={[300, 20]}
      strokeColor="#001342"
    />
    <Progress percent={100} percentPosition={{ align: 'center', type: 'inner' }} size={[400, 20]} />
    <Progress percent={60} percentPosition={{ align: 'start', type: 'outer' }} />
    <Progress percent={100} percentPosition={{ align: 'start', type: 'outer' }} />
    <Progress percent={60} percentPosition={{ align: 'center', type: 'outer' }} size="small" />
    <Progress percent={100} percentPosition={{ align: 'center', type: 'outer' }} />
  </Flex>
);

export default App;
```

API

通用属性参考：通用属性

各类型共用的属性。

属性	说明	类型	默认值	版本
format	内容的模板函数	function(percent, successPercent)	(percent) => percent + %	-
percent	百分比	number	0	-
showInfo	是否显示进度数值或状态图标	boolean	true	-

status	状态，可选：
	`success`
	`exception`
	`normal`
	`active`(仅限 line)
		string	-	-

strokeColor	进度条的色彩	string	-	-
strokeLinecap	进度条的样式	`round` | `butt` | `square`，区别详见 stroke-linecap	`round`	-
success	成功进度条相关配置	{ percent: number, strokeColor: string }	-	-

trailColor	未完成的分段的颜色	string	-	-
type	类型，可选 `line` `circle` `dashboard`	string	`line`	-
size	进度条的尺寸	number | [number | string, number] | { width: number, height: number } | "small" | "default"	"default"	5.3.0, Object: 5.18.0

`type="line"`

属性	说明	类型	默认值	版本
steps	进度条总共步数	number	-	-
rounding	用于四舍五入数值的函数	(step: number) => number	Math.round	5.24.0
strokeColor	进度条的色彩，传入 object 时为渐变。当有 `steps` 时支持传入一个数组。	string | string[] | { from: string; to: string; direction: string }	-	4.21.0: `string[]`
percentPosition	进度数值位置，传入对象，`align` 表示数值的水平位置，`type` 表示数值在进度条内部还是外部	{ align: string; type: string }	{ align: "end", type: "outer" }	5.18.0

`type="circle"`

属性	说明	类型	默认值	版本
steps	进度条总共步数，传入 object 时，count 指步数，gap 指间隔大小。传 number 类型时，gap 默认为 2。	number | { count: number, gap: number }	-	5.16.0
strokeColor	圆形进度条线的色彩，传入 object 时为渐变	string | { number%: string }	-	-
strokeWidth	圆形进度条线的宽度，单位是进度条画布宽度的百分比	number	6	-

`type="dashboard"`

属性	说明	类型	默认值	版本
steps	进度条总共步数，传入 object 时，count 指步数，gap 指间隔大小。传 number 类型时，gap 默认为 2。	number | { count: number, gap: number }	-	5.16.0
gapDegree	仪表盘进度条缺口角度，可取值 0 ~ 295	number	75	-
gapPosition	仪表盘进度条缺口位置	`top` | `bottom` | `left` | `right`	`bottom`	-
strokeWidth	仪表盘进度条线的宽度，单位是进度条画布宽度的百分比	number	6	-

主题变量（Design Token）