何时使用 {#when-to-use}

- 当需要突出某个或某组数字时。
- 当需要展示带描述的统计类数据时使用。

代码演示

基本

```
import React from 'react';
import { Button, Col, Row, Statistic } from 'antd';

const App: React.FC = () => (
  <Row gutter={16}>
    <Col span={12}>
      <Statistic title="Active Users" value={112893} />
    </Col>
    <Col span={12}>
      <Statistic title="Account Balance (CNY)" value={112893} precision={2}
/>
      <Button style={{ marginTop: 16 }} type="primary">
        Recharge
      </Button>
    </Col>
    <Col span={12}>
      <Statistic title="Active Users" value={112893} loading />
    </Col>
  </Row>
);

export default App;
```

单位

```
import React from 'react';
import { LikeOutlined } from '@ant-design/icons';
import { Col, Row, Statistic } from 'antd';

const App: React.FC = () => (
  <Row gutter={16}>
    <Col span={12}>
      <Statistic title="Feedback" value={1128} prefix={<LikeOutlined />} />
    </Col>
    <Col span={12}>
      <Statistic title="Unmerged" value={93} suffix="/ 100" />
    </Col>
  </Row>
);

export default App;
```

动画效果

```
import React from 'react';
import type { StatisticProps } from 'antd';
import { Col, Row, Statistic } from 'antd';
import CountUp from 'react-countup';

const formatter: StatisticProps['formatter'] = (value) => (
  <CountUp end={value as number} separator="," />
);

const App: React.FC = () => (
  <Row gutter={16}>
    <Col span={12}>
      <Statistic title="Active Users" value={112893} formatter={formatter}
/>
    </Col>
    <Col span={12}>
      <Statistic title="Account Balance (CNY)" value={112893} precision={2}
formatter={formatter} />
    </Col>
  </Row>
);

export default App;
```

在卡片中使用

```
import React from 'react';
import { ArrowDownOutlined, ArrowUpOutlined } from '@ant-design/icons';
import { Card, Col, Row, Statistic } from 'antd';

const App: React.FC = () => (
  <Row gutter={16}>
    <Col span={12}>
      <Card variant="borderless">
        <Statistic
          title="Active"
          value={11.28}
          precision={2}
          valueStyle={{ color: '#3f8600' }}
          prefix={<ArrowUpOutlined />}
          suffix="%"
        />
      </Card>
    </Col>
    <Col span={12}>
      <Card variant="borderless">
        <Statistic
          title="Idle"
          value={9.3}
          precision={2}
          valueStyle={{ color: '#cf1322' }}
          prefix={<ArrowDownOutlined />}
          suffix="%"
        />
      </Card>
    </Col>
  </Row>
);

export default App;
```

倒计时

```
import React from 'react';
import type { CountdownProps } from 'antd';
import { Col, Row, Statistic } from 'antd';

const { Countdown } = Statistic;

const deadline = Date.now() + 1000 * 60 * 60 * 24 * 2 + 1000 * 30; // Dayjs is also OK

const onFinish: CountdownProps['onFinish'] = () => {
  console.log('finished!');
};

const onChange: CountdownProps['onChange'] = (val) => {
  if (typeof val === 'number' && 4.95 * 1000 < val && val < 5 * 1000) {
    console.log('changed!');
  }
};

const App: React.FC = () => (
  <Row gutter={16}>
```

```
    <Col span={12}>
      <Countdown title="Countdown" value={deadline} onFinish={onFinish} />
    </Col>
    <Col span={12}>
      <Countdown title="Million Seconds" value={deadline}
format="HH:mm:ss:SSS" />
    </Col>
    <Col span={24} style={{ marginTop: 32 }}>
      <Countdown title="Day Level" value={deadline} format="D 天 H 时 m 分 s
秒" />
    </Col>
    <Col span={12}>
      <Countdown title="Countdown" value={Date.now() + 10 * 1000} onChange=
{onChange} />
    </Col>
  </Row>
);

export default App;
```

组件 Token

Debug

```
import React from 'react';
import { Button, Col, ConfigProvider, Row, Statistic } from 'antd';

const App: React.FC = () => (
  <ConfigProvider
    theme={{
      components: {
        Statistic: {
          titleFontSize: 20,
          contentFontSize: 20,
        },
      },
    }}
  >
    <Row gutter={16}>
      <Col span={12}>
        <Statistic title="Active Users" value={112893} />
      </Col>
      <Col span={12}>
        <Statistic title="Account Balance (CNY)" value={112893} precision=
{2} />
        <Button style={{ marginTop: 16 }} type="primary">
```

```
          Recharge
        </Button>
      </Col>
      <Col span={12}>
        <Statistic title="Active Users" value={112893} loading />
      </Col>
    </Row>
  </ConfigProvider>
);

export default App;
```

API

通用属性参考：通用属性

Statistic

参数	说明	类型	默认值	版本
decimalSeparator	设置小数点	string	`.`	
formatter	自定义数值展示	(value) => ReactNode	-	
groupSeparator	设置千分位标识符	string	`,`	
loading	数值是否加载中	boolean	false	4.8.0
precision	数值精度	number	-	
prefix	设置数值的前缀	ReactNode	-	
suffix	设置数值的后缀	ReactNode	-	
title	数值的标题	ReactNode	-	
value	数值内容	string | number	-	
valueStyle	设置数值区域的样式	CSSProperties	-	

Statistic.Countdown

参数	说明	类型	默认值	版本
format	格式化倒计时展示，参考 dayjs	string	`HH:mm:ss`	
prefix	设置数值的前缀	ReactNode	-	
suffix	设置数值的后缀	ReactNode	-	
title	数值的标题	ReactNode	-	

value	数值内容	number	-	
valueStyle	设置数值区域的样式	CSSProperties	-	
onFinish	倒计时完成时触发	() => void	-	
onChange	倒计时时间变化时触发	(value: number) => void	-	4.16.0

主题变量（Design Token）