

Statistic

Display statistic number.

When To Use

- When want to highlight some data.
- When want to display statistic data with description.

Examples

Active Users

112,893

Account Balance (CNY)

112,893.00

Recharge

Active Users

Basic [✎](#)

Simplest Usage.

```
import React from 'react';
import { Button, Col, Row, Statistic } from '@ant-design/compatible';

const App: React.FC = () => (
  <Row gutter={16}>
    <Col span={12}>
      <Statistic title="Active Users" value={112893}>
        112,893
      </Statistic>
    </Col>
    <Col span={12}>
      <Statistic title="Account Balance (CNY)" value={112893}>
        112,893.00
      </Statistic>
      <Button style={{ marginTop: 16 }} type="primary">
        Recharge
      </Button>
    </Col>
  </Row>
);

export default App;
```

Feedback

👍 1,128

Unmerged

93 / 100

Unit [✎](#)

Add unit through `prefix` and `suffix`.

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

const App: React.FC = () => (
  <Row gutter={16}>
    <Col span={12}>
      <Statistic title="Feedback" value={1128}>
        1,128
      </Col>
    <Col span={12}>
      <Statistic title="Unmerged" value={93}>
        93 / 100
      </Statistic>
    </Col>
  </Row>
);

export default App;
```

Active Users

Account Balance (CNY)

Animated number [✎](#)

Animated number with [react-countup](#).

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

const formatter = (value: number) => <CountUp
  value={value} separator="," />

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

export default App;
```

Countdown

Million Seconds

48:00:29

48:00:29:990

Day Level

2 天 0 时 0 分 29 秒

Countdown

00:00:09

Countdown

Countdown component.

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

const { Countdown } = Statistic;

const deadline = Date.now() + 1000 * 60 * 60 * 24;

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

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

const App: React.FC = () => (
  <Row gutter={16}>
    <Col span={12}>
      <Countdown title="Countdown" value={deadline} />
    </Col>
    <Col span={12}>
      <Countdown title="Million Seconds" value={deadline} />
    </Col>
    <Col span={24} style={{ marginTop: 32 }}>
      <Countdown title="Day Level" value={deadline} />
    </Col>
    <Col span={12}>
      <Countdown title="Countdown" value={Date.now()} />
    </Col>
  </Row>
);

export default App;
```

Active

↑ 11.28 %

Idle

↓ 9.30 %

In Card

Display statistic data in Card.

```
import React from 'react';
import { ArrowDownOutlined, ArrowUpOutlined } from 'ant-design-icons';

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

export default App;
```

API

Statistic

Property	Description	Type	Default	Version
decimalSeparator	The decimal separator	string	.	
formatter	Customize value display logic	(value) => ReactNode	-	

Property	Description	Type	Default	Version
groupSeparator	Group separator	string	<input type="text" value=","/>	
loading	Loading status of Statistic	boolean	false	4.8.0
precision	The precision of input value	number	-	
prefix	The prefix node of value	ReactNode	-	
suffix	The suffix node of value	ReactNode	-	
title	Display title	ReactNode	-	
value	Display value	string number	-	
valueStyle	Set value section style	CSSProperties	-	

Statistic.Countdown

Property	Description	Type	Default	Version
format	Format as dayjs	string	<input type="text" value="HH:mm:ss"/>	
prefix	The prefix node of value	ReactNode	-	
suffix	The suffix node of value	ReactNode	-	
title	Display title	ReactNode	-	
value	Set target countdown time	number dayjs	-	
valueStyle	Set value section style	CSSProperties	-	
onFinish	Trigger when time's up	() => void	-	
onChange	Trigger when time's changing	(value: number) => void	-	4.16.0

Design Token

▼ Global Token

Token Name	Description	Type	Default Value
colorText	Default text color which comply with W3C standards, and this color is also the darkest neutral color.	<input type="text" value="string"/>	<input type="text" value="rgba(0, 0, 0, 0.88)"/>
colorTextDescription	Control the font color of text description.	<input type="text" value="string"/>	<input type="text" value="rgba(0, 0, 0, 0.45)"/>
colorTextHeading	Control the font color of heading.	<input type="text" value="string"/>	<input type="text" value="rgba(0, 0, 0, 0.88)"/>

Token Name	Description	Type	Default Value
fontFamily	The font family of Ant Design prioritizes the default interface font of the system, and provides a set of alternative font libraries that are suitable for screen display to maintain the readability and readability of the font under different platforms and browsers, reflecting the friendly, stable and professional characteristics.	string	-apple-system, BlinkMacSystemFont, 'Segoe UI', Roboto, 'Helvetica Neue', Arial, 'Noto Sans', sans-serif, 'Apple Color Emoji', 'Segoe UI Emoji', 'Segoe UI Symbol', 'Noto Color Emoji'
fontSize	The most widely used font size in the design system, from which the text gradient will be derived.	number	14
fontSizeHeading3	Font size of h3 tag.	number	24
lineHeight	Line height of text.	number	1.5714285714285714
marginXXS	Control the margin of an element, with the smallest size.	number	4
padding	Control the padding of the element.	number	16