When To Use

Commonly displayed on the details page.

```jsx
// works when >= 5.8.0, recommended ✅

const items: DescriptionsProps['items'] = [
  {
    key: '1',
    label: 'UserName',
    children: <p>Zhou Maomao</p>,
  },
  {
    key: '2',
    label: 'Telephone',
    children: <p>1810000000</p>,
  },
  {
    key: '3',
    label: 'Live',
    children: <p>Hangzhou, Zhejiang</p>,
  },
  {
    key: '4',
    label: 'Remark',
    children: <p>empty</p>,
  },
  {
    key: '5',
    label: 'Address',
    children: <p>No. 18, Wantang Road, Xihu District, Hangzhou, Zhejiang,
China</p>,
  },
];

<Descriptions title="User Info" items={items} />;

// works when <5.8.0 , deprecated when >=5.8.0 🙅🏻‍♀️

<Descriptions title="User Info">
  <Descriptions.Item label="UserName">Zhou Maomao</Descriptions.Item>
  <Descriptions.Item label="Telephone">1810000000</Descriptions.Item>
  <Descriptions.Item label="Live">Hangzhou, Zhejiang</Descriptions.Item>
  <Descriptions.Item label="Remark">empty</Descriptions.Item>
  <Descriptions.Item label="Address">
    No. 18, Wantang Road, Xihu District, Hangzhou, Zhejiang, China
```

```
    </Descriptions.Item>
</Descriptions>;
```

Examples

Basic

```tsx
import React from 'react';
import { Descriptions } from 'antd';
import type { DescriptionsProps } from 'antd';

const items: DescriptionsProps['items'] = [
  {
    key: '1',
    label: 'UserName',
    children: 'Zhou Maomao',
  },
  {
    key: '2',
    label: 'Telephone',
    children: '1810000000',
  },
  {
    key: '3',
    label: 'Live',
    children: 'Hangzhou, Zhejiang',
  },
  {
    key: '4',
    label: 'Remark',
    children: 'empty',
  },
  {
    key: '5',
    label: 'Address',
    children: 'No. 18, Wantang Road, Xihu District, Hangzhou, Zhejiang,
China',
  },
];

const App: React.FC = () => <Descriptions title="User Info" items={items}
/>;

export default App;
```

border

```
import React from 'react';
import { Badge, Descriptions } from 'antd';
import type { DescriptionsProps } from 'antd';

const items: DescriptionsProps['items'] = [
  {
    key: '1',
    label: 'Product',
    children: 'Cloud Database',
  },
  {
    key: '2',
    label: 'Billing Mode',
    children: 'Prepaid',
  },
  {
    key: '3',
    label: 'Automatic Renewal',
    children: 'YES',
  },
  {
    key: '4',
    label: 'Order time',
    children: '2018-04-24 18:00:00',
  },
  {
    key: '5',
    label: 'Usage Time',
    children: '2019-04-24 18:00:00',
    span: 2,
  },
  {
    key: '6',
    label: 'Status',
    children: <Badge status="processing" text="Running" />,
    span: 3,
  },
  {
    key: '7',
    label: 'Negotiated Amount',
    children: '$80.00',
  },
  {
    key: '8',
```

```
    label: 'Discount',
    children: '$20.00',
  },
  {
    key: '9',
    label: 'Official Receipts',
    children: '$60.00',
  },
  {
    key: '10',
    label: 'Config Info',
    children: (
      <>
        Data disk type: MongoDB
        <br />
        Database version: 3.4
        <br />
        Package: dds.mongo.mid
        <br />
        Storage space: 10 GB
        <br />
        Replication factor: 3
        <br />
        Region: East China 1
        <br />
      </>
    ),
  },
];

const App: React.FC = () => <Descriptions title="User Info" bordered items=
{items} />;

export default App;
```

border

Debug

```
import React from 'react';
import { Badge, Descriptions, Table } from 'antd';
import type { DescriptionsProps, TableProps } from 'antd';

interface DataType {
  key: React.Key;
  name: string;
```

```typescript
  age: number;
  address: string;
}

const dataSource: DataType[] = [
  {
    key: '1',
    name: '胡彦斌',
    age: 32,
    address: '西湖区湖底公园1号',
  },
  {
    key: '2',
    name: '胡彦祖',
    age: 42,
    address: '西湖区湖底公园1号',
  },
];

const columns: TableProps<DataType>['columns'] = [
  {
    title: '姓名',
    dataIndex: 'name',
    key: 'name',
  },
  {
    title: '年龄',
    dataIndex: 'age',
    key: 'age',
  },
  {
    title: '住址',
    dataIndex: 'address',
    key: 'address',
  },
];

const items: DescriptionsProps['items'] = [
  {
    key: '1',
    label: 'Product',
    children: 'Cloud Database',
  },
  {
    key: '2',
    label: <div style={{ display: 'flex' }}>Billing Mode</div>,
```

```
    children: 'Prepaid',
  },
  {
    key: '3',
    label: 'Automatic Renewal',
    children: 'YES',
  },
  {
    key: '4',
    label: 'Order time',
    children: '2018-04-24 18:00:00',
  },
  {
    key: '5',
    label: 'Usage Time',
    span: 2,
    children: '2019-04-24 18:00:00',
  },
  {
    key: '6',
    label: 'Status',
    span: 3,
    children: <Badge status="processing" text="Running" />,
  },
  {
    key: '7',
    label: 'Negotiated Amount',
    children: '$80.00',
  },
  {
    key: '8',
    label: 'Discount',
    children: '$20.00',
  },
  {
    key: '9',
    label: 'Official Receipts',
    children: '$60.00',
  },
  {
    key: '10',
    label: 'Config Info',
    children: (
      <>
        Data disk type: MongoDB
        <br />
```

```
          Database version: 3.4
          <br />
          Package: dds.mongo.mid
          <br />
          Storage space: 10 GB
          <br />
          Replication factor: 3
          <br />
          Region: East China 1
          <br />
        </>
      ),
    },
    {
      key: '11',
      label: 'Official Receipts',
      children: '$60.00',
    },
    {
      key: '12',
      label: 'Config Info',
      children: (
        <Table<DataType> size="small" pagination={false} dataSource=
{dataSource} columns={columns} />
      ),
    },
];

const App: React.FC = () => <Descriptions title="User Info" column={2}
items={items} />;

export default App;
```

padding

Debug

```
import React from 'react';
import { Descriptions, Flex } from 'antd';
import type { DescriptionsProps } from 'antd';

const items: DescriptionsProps['items'] = [
  {
    key: '1',
    label: 'long',
    children: 'loooooooooooooooooooooooooooooooooooooooooooooooong',
```

```
    },
    {
      key: '2',
      label: 'long',
      children: 'loooooooooooooooooooooooooooooooooooooooooooooooong',
    },
    {
      key: '3',
      label: 'long',
      children: 'loooooooooooooooooooooooooooooooooooooooooooooooong',
    },
    {
      key: '4',
      label: 'long',
      children: 'loooooooooooooooooooooooooooooooooooooooooooooooong',
    },
    {
      key: '5',
      label: 'long',
      children: 'loooooooooooooooooooooooooooooooooooooooooooooooong',
    },
];

const App: React.FC = () => (
  <Flex gap={8} vertical>
    <div style={{ width: 600, border: '1px solid', padding: 20 }}>
      <Descriptions title="User Info" column={2} items={items} />
    </div>
    <div style={{ width: 600, border: '1px solid', padding: 20 }}>
      <Descriptions layout="vertical" title="User Info" column={2} items=
{items} />
    </div>
  </Flex>
);

export default App;
```

Custom size

```
import React, { useState } from 'react';
import { Button, Descriptions, Radio } from 'antd';
import type { DescriptionsProps, RadioChangeEvent } from 'antd';

const borderedItems: DescriptionsProps['items'] = [
  {
    key: '1',
```

```
    label: 'Product',
    children: 'Cloud Database',
  },
  {
    key: '2',
    label: 'Billing',
    children: 'Prepaid',
  },
  {
    key: '3',
    label: 'Time',
    children: '18:00:00',
  },
  {
    key: '4',
    label: 'Amount',
    children: '$80.00',
  },
  {
    key: '5',
    label: 'Discount',
    children: '$20.00',
  },
  {
    key: '6',
    label: 'Official',
    children: '$60.00',
  },
  {
    key: '7',
    label: 'Config Info',
    children: (
      <>
        Data disk type: MongoDB
        <br />
        Database version: 3.4
        <br />
        Package: dds.mongo.mid
        <br />
        Storage space: 10 GB
        <br />
        Replication factor: 3
        <br />
        Region: East China 1
        <br />
      </>
```

```
    ),
  },
];

const items: DescriptionsProps['items'] = [
  {
    key: '1',
    label: 'Product',
    children: 'Cloud Database',
  },
  {
    key: '2',
    label: 'Billing',
    children: 'Prepaid',
  },
  {
    key: '3',
    label: 'Time',
    children: '18:00:00',
  },
  {
    key: '4',
    label: 'Amount',
    children: '$80.00',
  },
  {
    key: '5',
    label: 'Discount',
    children: '$20.00',
  },
  {
    key: '6',
    label: 'Official',
    children: '$60.00',
  },
];

const App: React.FC = () => {
  const [size, setSize] = useState<'default' | 'middle' | 'small'>
('default');

  const onChange = (e: RadioChangeEvent) => {
    console.log('size checked', e.target.value);
    setSize(e.target.value);
  };
```

```
    return (
      <div>
        <Radio.Group onChange={onChange} value={size}>
          <Radio value="default">default</Radio>
          <Radio value="middle">middle</Radio>
          <Radio value="small">small</Radio>
        </Radio.Group>
        <br />
        <br />
        <Descriptions
          bordered
          title="Custom Size"
          size={size}
          extra={<Button type="primary">Edit</Button>}
          items={borderedItems}
        />
        <br />
        <br />
        <Descriptions
          title="Custom Size"
          size={size}
          extra={<Button type="primary">Edit</Button>}
          items={items}
        />
      </div>
    );
};

export default App;
```

responsive

```
import React from 'react';
import { Descriptions } from 'antd';
import type { DescriptionsProps } from 'antd';

const items: DescriptionsProps['items'] = [
  {
    label: 'Product',
    children: 'Cloud Database',
  },
  {
    label: 'Billing',
    children: 'Prepaid',
  },
  {
```

```
      label: 'Time',
      children: '18:00:00',
    },
    {
      label: 'Amount',
      children: '$80.00',
    },
    {
      label: 'Discount',
      span: { xl: 2, xxl: 2 },
      children: '$20.00',
    },
    {
      label: 'Official',
      span: { xl: 2, xxl: 2 },
      children: '$60.00',
    },
    {
      label: 'Config Info',
      span: { xs: 1, sm: 2, md: 3, lg: 3, xl: 2, xxl: 2 },
      children: (
        <>
          Data disk type: MongoDB
          <br />
          Database version: 3.4
          <br />
          Package: dds.mongo.mid
        </>
      ),
    },
    {
      label: 'Hardware Info',
      span: { xs: 1, sm: 2, md: 3, lg: 3, xl: 2, xxl: 2 },
      children: (
        <>
          CPU: 6 Core 3.5 GHz
          <br />
          Storage space: 10 GB
          <br />
          Replication factor: 3
          <br />
          Region: East China 1
        </>
      ),
    },
];
```

```
const App: React.FC = () => (
  <Descriptions
    title="Responsive Descriptions"
    bordered
    column={{ xs: 1, sm: 2, md: 3, lg: 3, xl: 4, xxl: 4 }}
    items={items}
  />
);

export default App;
```

Vertical

```
import React from 'react';
import { Descriptions } from 'antd';
import type { DescriptionsProps } from 'antd';

const items: DescriptionsProps['items'] = [
  {
    key: '1',
    label: 'UserName',
    children: 'Zhou Maomao',
  },
  {
    key: '2',
    label: 'Telephone',
    children: '1810000000',
  },
  {
    key: '3',
    label: 'Live',
    children: 'Hangzhou, Zhejiang',
  },
  {
    key: '4',
    label: 'Address',
    span: 2,
    children: 'No. 18, Wantang Road, Xihu District, Hangzhou, Zhejiang,
China',
  },
  {
    key: '5',
    label: 'Remark',
    children: 'empty',
  },
```

```
];

const App: React.FC = () => <Descriptions title="User Info"
layout="vertical" items={items} />;

export default App;
```

Vertical border

```
import React from 'react';
import { Badge, Descriptions } from 'antd';
import type { DescriptionsProps } from 'antd';

const items: DescriptionsProps['items'] = [
  {
    key: '1',
    label: 'Product',
    children: 'Cloud Database',
  },
  {
    key: '2',
    label: 'Billing Mode',
    children: 'Prepaid',
  },
  {
    key: '3',
    label: 'Automatic Renewal',
    children: 'YES',
  },
  {
    key: '4',
    label: 'Order time',
    children: '2018-04-24 18:00:00',
  },
  {
    key: '5',
    label: 'Usage Time',
    span: 2,
    children: '2019-04-24 18:00:00',
  },
  {
    key: '6',
    label: 'Status',
    span: 3,
    children: <Badge status="processing" text="Running" />,
  },
```

```
    {
      key: '7',
      label: 'Negotiated Amount',
      children: '$80.00',
    },
    {
      key: '8',
      label: 'Discount',
      children: '$20.00',
    },
    {
      key: '9',
      label: 'Official Receipts',
      children: '$60.00',
    },
    {
      key: '10',
      label: 'Config Info',
      children: (
        <>
          Data disk type: MongoDB
          <br />
          Database version: 3.4
          <br />
          Package: dds.mongo.mid
          <br />
          Storage space: 10 GB
          <br />
          Replication factor: 3
          <br />
          Region: East China 1
          <br />
        </>
      ),
    },
];

const App: React.FC = () => (
  <Descriptions title="User Info" layout="vertical" bordered items={items}
/>
);

export default App;
```

Customize label & wrapper style

Debug

```tsx
import React, { useState } from 'react';
import { Descriptions, Divider, Radio, Switch } from 'antd';
import type { DescriptionsProps } from 'antd';

const labelStyle: React.CSSProperties = { background: 'red' };
const contentStyle: React.CSSProperties = { background: 'green' };

type LayoutType = 'horizontal' | 'vertical' | undefined;

const items: DescriptionsProps['items'] = [
  {
    key: '1',
    label: 'Product',
    children: 'Cloud Database',
    styles: {
      label: labelStyle,
      content: contentStyle,
    },
  },
  {
    key: '2',
    label: 'Billing Mode',
    children: 'Prepaid',
  },
  {
    key: '3',
    label: 'Automatic Renewal',
    children: 'YES',
  },
];

const rootStyleItems: DescriptionsProps['items'] = [
  {
    key: '1',
    label: 'Product',
    children: 'Cloud Database',
  },
  {
    key: '2',
    label: 'Billing Mode',
    children: 'Prepaid',
  },
  {
    key: '3',
```

```
      label: 'Automatic Renewal',
      children: 'YES',
      styles: {
        label: { color: 'orange' },
        content: { color: 'blue' },
      },
    },
  },
];

const App: React.FC = () => {
  const [border, setBorder] = useState(true);
  const [layout, setLayout] = useState('horizontal' as LayoutType);

  return (
    <>
      <Switch
        checkedChildren="Border"
        unCheckedChildren="No Border"
        checked={border}
        onChange={(e) => setBorder(e)}
      />
      <Divider />
      <Radio.Group onChange={(e) => setLayout(e.target.value)} value=
{layout}>
        <Radio value="horizontal">horizontal</Radio>
        <Radio value="vertical">vertical</Radio>
      </Radio.Group>
      <Divider />
      <Descriptions title="User Info" bordered={border} layout={layout}
items={items} />
      <Divider />
      <Descriptions
        title="Root style"
        styles={{
          label: labelStyle,
          content: contentStyle,
        }}
        bordered={border}
        layout={layout}
        items={rootStyleItems}
      />
    </>
  );
};

export default App;
```

JSX demo

Debug

```jsx
import React from 'react';
import { Descriptions } from 'antd';

const App: React.FC = () => (
  <Descriptions title="User Info">
    <Descriptions.Item label="UserName">Zhou Maomao</Descriptions.Item>
    <Descriptions.Item label="Telephone">1810000000</Descriptions.Item>
    <Descriptions.Item label="Live">Hangzhou, Zhejiang</Descriptions.Item>
    <Descriptions.Item label="Remark">empty</Descriptions.Item>
    <Descriptions.Item label="Address">
      No. 18, Wantang Road, Xihu District, Hangzhou, Zhejiang, China
    </Descriptions.Item>
  </Descriptions>
);

export default App;
```

Component Token

Debug

```jsx
import React, { useState } from 'react';
import type { DescriptionsProps, RadioChangeEvent } from 'antd';
import { Button, ConfigProvider, Descriptions, Radio } from 'antd';

const borderedItems: DescriptionsProps['items'] = [
  {
    key: '1',
    label: 'Product',
    children: 'Cloud Database',
  },
  {
    key: '2',
    label: 'Billing',
    children: 'Prepaid',
  },
  {
    key: '3',
    label: 'Time',
    children: '18:00:00',
  },
  {
```

```
    key: '4',
    label: 'Amount',
    children: '$80.00',
  },
  {
    key: '5',
    label: 'Discount',
    children: '$20.00',
  },
  {
    key: '6',
    label: 'Official',
    children: '$60.00',
  },
  {
    key: '7',
    label: 'Config Info',
    children: (
      <>
        Data disk type: MongoDB
        <br />
        Database version: 3.4
        <br />
        Package: dds.mongo.mid
        <br />
        Storage space: 10 GB
        <br />
        Replication factor: 3
        <br />
        Region: East China 1
        <br />
      </>
    ),
  },
];
const items: DescriptionsProps['items'] = [
  {
    key: '1',
    label: 'Product',
    children: 'Cloud Database',
  },
  {
    key: '2',
    label: 'Billing',
    children: 'Prepaid',
  },
```

```
  {
    key: '3',
    label: 'Time',
    children: '18:00:00',
  },
  {
    key: '4',
    label: 'Amount',
    children: '$80.00',
  },
  {
    key: '5',
    label: 'Discount',
    children: '$20.00',
  },
  {
    key: '6',
    label: 'Official',
    children: '$60.00',
  },
];

const App: React.FC = () => {
  const [size, setSize] = useState<'default' | 'middle' | 'small'>
('default');

  const onChange = (e: RadioChangeEvent) => {
    console.log('size checked', e.target.value);
    setSize(e.target.value);
  };

  return (
    <ConfigProvider
      theme={{
        components: {
          Descriptions: {
            labelBg: 'red',
            titleColor: 'red',
            titleMarginBottom: 2,
            itemPaddingBottom: 8,
            itemPaddingEnd: 8,
            colonMarginRight: 10,
            colonMarginLeft: 20,
            contentColor: 'green',
            extraColor: 'blue',
          },
```

```
      },
    }}
  >
    <div>
      <Radio.Group onChange={onChange} value={size}>
        <Radio value="default">default</Radio>
        <Radio value="middle">middle</Radio>
        <Radio value="small">small</Radio>
      </Radio.Group>
      <br />
      <br />
      <Descriptions
        bordered
        title="Custom Size"
        size={size}
        extra={<div>extra color: blue</div>}
        items={borderedItems}
      />
      <br />
      <br />
      <Descriptions
        title="Custom Size"
        size={size}
        extra={<Button type="primary">Edit</Button>}
        items={items}
      />
    </div>
  </ConfigProvider>
);
};

export default App;
```

row

```
import React from 'react';
import { Descriptions } from 'antd';
import type { DescriptionsProps } from 'antd';

const items: DescriptionsProps['items'] = [
  {
    label: 'UserName',
    children: 'Zhou Maomao',
  },
  {
    label: 'Live',
```

```
    span: 'filled', // span = 2
    children: 'Hangzhou, Zhejiang',
  },
  {
    label: 'Remark',
    span: 'filled', // span = 3
    children: 'empty',
  },
  {
    label: 'Address',
    span: 1, // span will be 3 and warning for span is not align to the end
    children: 'No. 18, Wantang Road, Xihu District, Hangzhou, Zhejiang,
China',
  },
];

const App: React.FC = () => <Descriptions bordered title="User Info" items=
{items} />;

export default App;
```

API

Common props ref: [Common props](#)

Descriptions

Property	Description	Type	Default	Version
bordered	Whether to display the border	boolean	false	
colon	Change default props `colon` value of Descriptions.Item. Indicates whether the colon after the label is displayed	boolean	true	
column	The number of `DescriptionItems` in a row,could be a number or a object like { xs: 8, sm: 16, md: 24},(Only set bordered=	number | [Record<Breakpoint, number>](#)	3	

	`{true}` to take effect)			
~~contentStyle~~	Customize content style, Please use `styles={{ content: {} }}` instead	CSSProperties	-	4.10.0
extra	The action area of the description list, placed at the top-right	ReactNode	-	4.5.0
items	Describe the contents of the list item	[DescriptionsItem](#)[]	-	5.8.0
~~labelStyle~~	Customize label style	CSSProperties, Please use `styles={{ label: {} }}` instead	-	4.10.0
layout	Define description layout	`horizontal | vertical`	`horizontal`	
size	Set the size of the list. Can be set to `middle`,`small`, or not filled	`default | middle | small`	-	
title	The title of the description list, placed at the top	ReactNode	-	
classNames	Semantic DOM class	[Record<SemanticDOM, string>](#)	-	5.23.0
styles	Semantic DOM style	[Record<SemanticDOM, CSSProperties>](#)	-	5.23.0

DescriptionItem

Property	Description	Type	Default	Version
~~contentStyle~~	Customize content style, Please use `styles={{ content: {} }}` instead	CSSProperties	-	4.9.0
label	The description of the content	ReactNode	-	

labelStyle	Customize label style, Please use `styles={{ label: {} }}` instead	CSSProperties	-	4.9.0
span	The number of columns included(`filled` Fill the remaining part of the current row)	number | `filled` | [Screens](#)	1	screens: `5.9.0`, filled: `5.22.0`

> *The number of span Description.Item. Span={2} takes up the width of two DescriptionItems. When both `style` and `labelStyle` (or `contentStyle`) configured, both of them will work. And next one will overwrite first when conflict.*

Semantic DOM

演示

```tsx
import React from 'react';
import { Button, Descriptions, DescriptionsProps, Divider, Switch } from 'antd';

import SemanticPreview from '../../../.dumi/components/SemanticPreview';
import useLocale from '../../../.dumi/hooks/useLocale';

const locales = {
  cn: {
    root: '根元素',
    header: '头部元素',
    title: '标题元素',
    extra: '额外内容',
    label: '标签元素',
    content: '内容元素',
  },
  en: {
    root: 'root element',
    header: 'header element',
    title: 'title element',
    extra: 'extra element',
    label: 'label element',
    content: 'content element',
  },
};

const items: DescriptionsProps['items'] = [
  {
    key: '1',
```

```
      label: 'Telephone',
      children: '1810000000',
  },
];

const BlockList: React.FC<React.PropsWithChildren> = (props: any) => {
  const divRef = React.useRef<HTMLDivElement>(null);
  const [bordered, setBordered] = React.useState(false);

  const handleBorderChange = (checked: boolean) => {
    setBordered(checked);
  };

  return (
    <div ref={divRef} style={{ width: '100%', height: '100%' }}>
      <Switch checked={bordered} onChange={handleBorderChange} /> Toggle
Border
      <Divider />
      <Descriptions
        title="User Info"
        items={items}
        extra={<Button type="primary">Edit</Button>}
        bordered={bordered}
        {...props}
      />
    </div>
  );
};

const App: React.FC = () => {
  const [locale] = useLocale(locales);
  return (
    <SemanticPreview
      componentName="Descriptions"
      semantics={[
        { name: 'root', desc: locale.root, version: '5.23.0' },
        { name: 'header', desc: locale.header, version: '5.23.0' },
        { name: 'title', desc: locale.title, version: '5.23.0' },
        { name: 'extra', desc: locale.extra, version: '5.23.0' },
        { name: 'label', desc: locale.label, version: '5.23.0' },
        { name: 'content', desc: locale.content, version: '5.23.0' },
      ]}
    >
      <BlockList />
    </SemanticPreview>
  );
```

```
};

export default App;
```

Design Token