

何时使用 {#when-to-use}

常见于详情页的信息展示。

// >= 5.8.0 可用, 推荐的写法 

```
const items: DescriptionsProps['items'] = [
  {
    key: '1',
    label: 'UserName',
    children: <p>Zhou Maomao</p>,
  },
  {
    key: '2',
    label: 'Telephone',
    children: <p>18100000000</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} />;
```

// <5.8.0 可用, >=5.8.0 时不推荐 

```
<Descriptions title="User Info">
  <Descriptions.Item label="UserName">Zhou Maomao</Descriptions.Item>
  <Descriptions.Item label="Telephone">18100000000</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>;
```

代码演示

基本

```
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: '18100000000',
  },
  {
    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;
```

带边框的

```
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;

```

复杂文本的情况

Debug

```

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

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

```

```
    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 />
    )
  }
]

```



```

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

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;

```

自定义尺寸

```

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;

```

响应式

```

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;

```

垂直

```

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: '18100000000',
  },
  {
    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;

```

垂直带边框的

```

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;

```

自定义 label & wrapper 样式

Debug

```
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

```
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">18100000000</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;
```

组件 Token

Debug

```
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;

```

整行

```

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

通用属性参考: [通用属性](#)

Descriptions

参数	说明	类型	默认值	版本
bordered	是否展示边框	boolean	false	
colon	配置 Descriptions.Item 的 colon 的默认值。表 示是否显示 label 后面的 冒号	boolean	true	
column	一行的 DescriptionItems 数 量, 可以写成像素值或支 持响应式的对象写法 { xs: 8, sm: 16, md: 24}	number Record<Breakpoint, number>	3	
contentStyle	自定义内容样式, 请使用 styles={{ content: { } }} 替换	CSSProperties	-	4.10.0

extra	描述列表的操作区域，显示在右上方	ReactNode	-	4.5.0
items	描述列表项内容	DescriptionsItem[]	-	5.8.0
labelStyle	自定义标签样式，请使用 <code>styles={{ label: {} }}</code> 替换	CSSProperties	-	4.10.0
layout	描述布局	horizontal vertical	horizontal	
size	设置列表的大小。可以设置为 middle、small，或不填（只有设置 <code>bordered={true}</code> 生效）	default middle small	-	
title	描述列表的标题，显示在最顶部	ReactNode	-	
classNames	语义化结构 class	Record<SemanticDOM, string>	-	5.23.0
styles	语义化结构 style	Record<SemanticDOM, CSSProperties>	-	5.23.0

DescriptionItem

参数	说明	类型	默认值	版本
contentStyle	自定义内容样式，请使用 <code>styles={{ content: {} }}</code> 替换	CSSProperties	-	4.9.0
label	内容的描述	ReactNode	-	
labelStyle	自定义标签样式，请使用 <code>styles={{ label: {} }}</code> 替换	CSSProperties	-	4.9.0
span	包含列的数量（filled 铺满当前行剩余部分）	number filled Screens	1	screens: 5.9.0, filled: 5.22.0

`span` 是 `Description.Item` 的数量。`span={2}` 会占用两个 `DescriptionItem` 的宽度。当同时配置 `style` 和 `labelStyle`（或 `contentStyle`）时，两者会同时作用。样式冲突时，后者会覆盖前者。

Semantic DOM

演示

```
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;

```

主题变量 (Design Token)