When To Use

Can be used to separate areas horizontally or vertically. When you need to freely drag and adjust the size of each area. When you need to specify the maximum and minimum width and height of an area.

Examples

Basic

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

const Desc: React.FC<Readonly<{ text?: string | number }>> = (props) => (
  <Flex justify="center" align="center" style={{ height: '100%' }}>
    <Typography.Title type="secondary" level={5} style={{ whiteSpace:
'nowrap' }}>
      {props.text}
    </Typography.Title>
  </Flex>
);

const App: React.FC = () => (
  <Splitter style={{ height: 200, boxShadow: '0 0 10px rgba(0, 0, 0, 0.1)'
}}>
    <Splitter.Panel defaultSize="40%" min="20%" max="70%">
      <Desc text="First" />
    </Splitter.Panel>
    <Splitter.Panel>
      <Desc text="Second" />
    </Splitter.Panel>
  </Splitter>
);

export default App;
```

Control mode

```
import React from 'react';
import { Button, Flex, Splitter, Switch, Typography } from 'antd';

const Desc: React.FC<Readonly<{ text?: string | number }>> = (props) => (
  <Flex justify="center" align="center" style={{ height: '100%' }}>
    <Typography.Title type="secondary" level={5} style={{ whiteSpace:
'nowrap' }}>
      {props.text}
    </Typography.Title>
  </Flex>
);

const App: React.FC = () => {
  const [sizes, setSizes] = React.useState<(number | string)[]>(['50%',
'50%']);
  const [enabled, setEnabled] = React.useState(true);
  return (
    <Flex vertical gap="middle">
      <Splitter
        onResize={setSizes}
        style={{ height: 200, boxShadow: '0 0 10px rgba(0, 0, 0, 0.1)' }}
      >
        <Splitter.Panel size={sizes[0]} resizable={enabled}>
          <Desc text="First" />
        </Splitter.Panel>
        <Splitter.Panel size={sizes[1]}>
          <Desc text="Second" />
        </Splitter.Panel>
      </Splitter>
      <Flex gap="middle" justify="space-between">
        <Switch
          value={enabled}
          onChange={() => setEnabled(!enabled)}
          checkedChildren="Enabled"
          unCheckedChildren="Disabled"
        />
        <Button onClick={() => setSizes(['50%', '50%'])}>Reset</Button>
      </Flex>
    </Flex>
  );
};

export default App;
```

Vertical

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

const Desc: React.FC<Readonly<{ text?: string | number }>> = (props) => (
  <Flex justify="center" align="center" style={{ height: '100%' }}>
    <Typography.Title type="secondary" level={5} style={{ whiteSpace:
'nowrap' }}>
      {props.text}
    </Typography.Title>
  </Flex>
);

const App: React.FC = () => (
  <Splitter layout="vertical" style={{ height: 300, boxShadow: '0 0 10px rgba(0, 0, 0, 0.1)' }}>
    <Splitter.Panel>
      <Desc text="First" />
    </Splitter.Panel>
    <Splitter.Panel>
      <Desc text="Second" />
    </Splitter.Panel>
  </Splitter>
);

export default App;
```

Collapsible

```
import React from 'react';
import { Flex, Splitter, Typography } from 'antd';
import type { SplitterProps } from 'antd';

const Desc: React.FC<Readonly<{ text?: string | number }>> = (props) => (
  <Flex justify="center" align="center" style={{ height: '100%' }}>
    <Typography.Title type="secondary" level={5} style={{ whiteSpace: 'nowrap' }}>
      {props.text}
    </Typography.Title>
  </Flex>
);

const CustomSplitter: React.FC<Readonly<SplitterProps>> = ({ style, ...restProps }) => (
  <Splitter style={{ boxShadow: '0 0 10px rgba(0, 0, 0, 0.1)', ...style }} {...restProps}>
    <Splitter.Panel collapsible min="20%">
      <Desc text="First" />
    </Splitter.Panel>
    <Splitter.Panel collapsible>
      <Desc text="Second" />
    </Splitter.Panel>
  </Splitter>
);
```

```
const App: React.FC = () => (
  <Flex gap="middle" vertical>
    <CustomSplitter style={{ height: 200 }} />
    <CustomSplitter style={{ height: 300 }} layout="vertical" />
  </Flex>
);

export default App;
```

Multiple panels

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

const Desc: React.FC<Readonly<{ text?: string | number }>> = (props) => (
  <Flex justify="center" align="center" style={{ height: '100%' }}>
    <Typography.Title type="secondary" level={5} style={{ whiteSpace:
'nowrap' }}>
      Panel {props.text}
    </Typography.Title>
  </Flex>
);

const App: React.FC = () => (
  <Splitter style={{ height: 200, boxShadow: '0 0 10px rgba(0, 0, 0, 0.1)'
}}>
    <Splitter.Panel collapsible>
      <Desc text={1} />
    </Splitter.Panel>
    <Splitter.Panel collapsible={{ start: true }}>
      <Desc text={2} />
    </Splitter.Panel>
    <Splitter.Panel>
      <Desc text={3} />
    </Splitter.Panel>
  </Splitter>
);

export default App;
```

Complex combination

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

```
const Desc: React.FC<Readonly<{ text?: string | number }>> = (props) => (
  <Flex justify="center" align="center" style={{ height: '100%' }}>
    <Typography.Title type="secondary" level={5} style={{ whiteSpace: 'nowrap' }}>
      {props.text}
    </Typography.Title>
  </Flex>
);

const App: React.FC = () => (
  <Splitter style={{ height: 300, boxShadow: '0 0 10px rgba(0, 0, 0, 0.1)' }}>
    <Splitter.Panel collapsible>
      <Desc text="Left" />
    </Splitter.Panel>
    <Splitter.Panel>
      <Splitter layout="vertical">
        <Splitter.Panel>
          <Desc text="Top" />
        </Splitter.Panel>
        <Splitter.Panel>
          <Desc text="Bottom" />
        </Splitter.Panel>
      </Splitter>
    </Splitter.Panel>
  </Splitter>
);

export default App;
```

Nested in tabs

Debug

```
import React from 'react';
import { Flex, Splitter, Tabs, Typography } from 'antd';

const Desc: React.FC<Readonly<{ text?: string | number }>> = (props) => (
  <Flex justify="center" align="center" style={{ height: '100%' }}>
    <Typography.Title type="secondary" level={5} style={{ whiteSpace: 'nowrap' }}>
      {props.text}
    </Typography.Title>
  </Flex>
);
```

```
const App: React.FC = () => {
  const SplitterContent = (
    <Splitter
      style={{
        height: 200,
        boxShadow: '0 0 10px rgba(0, 0, 0, 0.1)',
      }}
    >
      <Splitter.Panel collapsible>
        <Desc text={1} />
      </Splitter.Panel>
      <Splitter.Panel
        collapsible={{
          start: true,
        }}
      >
        <Desc text={2} />
      </Splitter.Panel>
      <Splitter.Panel>
        <Desc text={3} />
      </Splitter.Panel>
    </Splitter>
  );
  return (
    <Tabs
      defaultActiveKey="1"
      items={[
        {
          key: '1',
          label: 'General',
          children: 'Content of Tab Pane 1',
        },
        {
          key: '2',
          label: 'Splitter Tab',
          children: SplitterContent,
        },
      ]}
    />
  );
};

export default App;
```

Lazy

v5.23.0

```
import React from 'react';
import { Flex, Space, Splitter, Typography } from 'antd';

const Desc: React.FC<Readonly<{ text?: string | number }>> = (props) => (
  <Flex justify="center" align="center" style={{ height: '100%' }}>
    <Typography.Title type="secondary" level={5} style={{ whiteSpace: 'nowrap' }}>
      {props.text}
    </Typography.Title>
  </Flex>
);

const App: React.FC = () => (
  <Space direction="vertical" style={{ width: '100%' }}>
    <Splitter lazy style={{ height: 200, boxShadow: '0 0 10px rgba(0, 0, 0, 0.1)' }}>
      <Splitter.Panel defaultSize="40%" min="20%" max="70%">
        <Desc text="First" />
      </Splitter.Panel>
      <Splitter.Panel>
        <Desc text="Second" />
      </Splitter.Panel>
    </Splitter>
    <Splitter
      lazy
      layout="vertical"
      style={{ height: 200, boxShadow: '0 0 10px rgba(0, 0, 0, 0.1)' }}
    >
      <Splitter.Panel defaultSize="40%" min="30%" max="70%">
        <Desc text="First" />
      </Splitter.Panel>
      <Splitter.Panel>
        <Desc text="Second" />
      </Splitter.Panel>
    </Splitter>
  </Space>
);

export default App;
```

Debug

Debug

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

const Desc: React.FC<Readonly<{ text?: string | number }>> = (props) => (
  <Flex justify="center" align="center" style={{ height: '100%' }}>
    <Typography.Title type="secondary" level={5} style={{ whiteSpace:
'nowrap' }}>
      Panel {props.text}
    </Typography.Title>
  </Flex>
);

const App: React.FC = () => (
  <Flex vertical gap="middle">
    <Typography.Title level={3}>[true, 0, false]</Typography.Title>
    <Splitter style={{ height: 200, boxShadow: '0 0 10px rgba(0, 0, 0,
0.1)' }}>
      <Splitter.Panel>
        <Desc text={1} />
      </Splitter.Panel>
      <Splitter.Panel defaultSize={0}>
        <Desc text={2} />
      </Splitter.Panel>
      <Splitter.Panel resizable={false}>
        <Desc text={3} />
      </Splitter.Panel>
    </Splitter>
    <Typography.Title level={3}>[false, 0, true]</Typography.Title>
    <Splitter style={{ height: 200, boxShadow: '0 0 10px rgba(0, 0, 0,
0.1)' }}>
      <Splitter.Panel resizable={false}>
        <Desc text={1} />
      </Splitter.Panel>
      <Splitter.Panel defaultSize={0}>
        <Desc text={2} />
      </Splitter.Panel>
      <Splitter.Panel>
        <Desc text={3} />
      </Splitter.Panel>
    </Splitter>
    <Typography.Title level={3}>Start have min & max</Typography.Title>
    <Splitter style={{ height: 200, boxShadow: '0 0 10px rgba(0, 0, 0,
0.1)' }}>
      <Splitter.Panel min={50} max={100}>
        <Desc text={1} />
      </Splitter.Panel>
      <Splitter.Panel>
        <Desc text={2} />
      </Splitter.Panel>
    </Splitter>
    <Typography.Title level={3}>End have min & max</Typography.Title>
    <Splitter style={{ height: 200, boxShadow: '0 0 10px rgba(0, 0, 0, 0.1)' }}>
      <Splitter.Panel>
        <Desc text={1} />
      </Splitter.Panel>
      <Splitter.Panel min="20%" max="70%">
        <Desc text={2} />
      </Splitter.Panel>
    </Splitter>
  </Flex>
);

export default App;
```

API

Common props ref: Common props

> *The Splitter component needs to calculate the panel size through its child elements, so its child elements only support* `Splitter.Panel` .

Splitter

Property	Description	Type	Default	Version
layout	Layout direction	`horizontal` | `vertical`	`horizontal`	-
onResizeStart	Callback before dragging starts	`(sizes: number[]) => void`	-	-
onResize	Panel size change callback	`(sizes: number[]) => void`	-	-
onResizeEnd	Drag end callback	`(sizes: number[]) => void`	-	-
lazy	Lazy mode	`boolean`	`false`	5.23.0

Panel

Property	Description	Type	Default	Version

defaultSize	Initial panel size support number for px or 'percent%' usage	`number | string`	-	-
min	Minimum threshold support number for px or 'percent%' usage	`number | string`	-	-
max	Maximum threshold support number for px or 'percent%' usage	`number | string`	-	-
size	Controlled panel size support number for px or 'percent%' usage	`number | string`	-	-
collapsible	Quick folding	`boolean | { start?: boolean; end?: boolean }`	`false`	-
resizable	Whether to enable drag and drop	`boolean`	`true`	-

Design Token