

何时使用 {#when-to-use}

- 适合设置元素之间的间距。
- 适合设置各种水平、垂直对齐方式。

与 Space 组件的区别

- Space 为内联元素提供间距，其本身会为每一个子元素添加包裹元素用于内联对齐。适用于行、列中多个子元素的等距排列。
- Flex 为块级元素提供间距，其本身不会添加包裹元素。适用于垂直或水平方向上的子元素布局，并提供了更多的灵活性和控制能力。

代码演示

基本布局

```
/* eslint-disable react/no-array-index-key */
import React from 'react';
import { Flex, Radio } from 'antd';

const baseStyle: React.CSSProperties = {
  width: '25%',
  height: 54,
};

const App: React.FC = () => {
  const [value, setValue] = React.useState<string>('horizontal');
  return (
    <Flex gap="middle" vertical>
      <Radio.Group value={value} onChange={(e) =>
        setValue(e.target.value)}>
        <Radio value="horizontal">horizontal</Radio>
        <Radio value="vertical">vertical</Radio>
      </Radio.Group>
      <Flex vertical={value === 'vertical'}>
        {Array.from({ length: 4 }).map((_, i) => (
          <div key={i} style={{ ...baseStyle, backgroundColor: i % 2 ?
            '#1677ff' : '#1677ffbf' }} />
        ))}
      </Flex>
    </Flex>
  );
};

export default App;
```

对齐方式

```

import React from 'react';
import { Button, Flex, Segmented } from 'antd';
import type { FlexProps } from 'antd';

const boxStyle: React.CSSProperties = {
  width: '100%',
  height: 120,
  borderRadius: 6,
  border: '1px solid #40a9ff',
};

const justifyOptions = [
  'flex-start',
  'center',
  'flex-end',
  'space-between',
  'space-around',
  'space-evenly',
];

const alignOptions = ['flex-start', 'center', 'flex-end'];

const App: React.FC = () => {
  const [justify, setJustify] = React.useState<FlexProps['justify']>(
    justifyOptions[0]);
  const [alignItems, setAlignItems] = React.useState<FlexProps['align']>(
    alignOptions[0]);
  return (
    <Flex gap="middle" align="start" vertical>
      <p>Select justify :</p>
      <Segmented options={justifyOptions} onChange={setJustify} />
      <p>Select align :</p>
      <Segmented options={alignOptions} onChange={setAlignItems} />
      <Flex style={boxStyle} justify={justify} align={alignItems}>
        <Button type="primary">Primary</Button>
        <Button type="primary">Primary</Button>
        <Button type="primary">Primary</Button>
        <Button type="primary">Primary</Button>
      </Flex>
    </Flex>
  );
};

export default App;

```

设置间隙

```
import React from 'react';
import { Button, Flex, Radio, Slider } from 'antd';
import type { ConfigProviderProps } from 'antd';

type SizeType = ConfigProviderProps['componentSize'];

const App: React.FC = () => {
  const [gapSize, setGapSize] = React.useState<SizeType | 'customize'>('small');
  const [customGapSize, setCustomGapSize] = React.useState<number>(0);
  return (
    <Flex gap="middle" vertical>
      <Radio.Group value={gapSize} onChange={(e) => setGapSize(e.target.value)}>
        {[ 'small', 'middle', 'large', 'customize' ].map((size) => (
          <Radio key={size} value={size}>
            {size}
          </Radio>
        ))}
      </Radio.Group>
      {gapSize === 'customize' && <Slider value={customGapSize} onChange={setCustomGapSize} />}
      <Flex gap={gapSize !== 'customize' ? gapSize : customGapSize}>
        <Button type="primary">Primary</Button>
        <Button>Default</Button>
        <Button type="dashed">Dashed</Button>
        <Button type="link">Link</Button>
      </Flex>
    </Flex>
  );
};

export default App;
```

自动换行

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

const Demo: React.FC = () => (
  <Flex wrap gap="small">
    {Array.from({ length: 24 }, (_, i) => (
      <Button key={i} type="primary">
```

```

        Button
      </Button>
    )})
  </Flex>
);

export default Demo;

```

组合使用

```

import React from 'react';
import { Button, Card, Flex, Typography } from 'antd';

const cardStyle: React.CSSProperties = {
  width: 620,
};

const imgStyle: React.CSSProperties = {
  display: 'block',
  width: 273,
};

const App: React.FC = () => (
  <Card hoverable style={cardStyle} styles={{ body: { padding: 0, overflow:
'hidden' } }}>
    <Flex justify="space-between">
      
        <Typography.Title level={3}>
          “antd is an enterprise-class UI design language and React UI
library.”
        </Typography.Title>
        <Button type="primary" href="https://ant.design" target="_blank">
          Get Started
        </Button>
      </Flex>
    </Flex>
  </Card>
);

```

```
export default App;
```

调试专用

Debug

```
/* eslint-disable react/no-array-index-key */
import React from 'react';
import { Flex } from 'antd';

const App: React.FC = () => (
  <>
    <Flex vertical>
      {Array.from({ length: 4 }).map((_, i) => (
        <div
          key={i}
          style={{
            height: 60,
            backgroundColor: i % 2 ? '#1677ff' : '#1677ffbf',
          }}
        />
      ))}
    </Flex>
    <Flex style={{ marginTop: 20 }}>
      {Array.from({ length: 4 }).map((_, i) => (
        <div
          key={i}
          style={{
            width: '25%',
            height: i % 2 ? 60 : 40,
            backgroundColor: i % 2 ? '#1677ff' : '#1677ffbf',
          }}
        />
      ))}
    </Flex>
  </>
);

export default App;
```

API

自 `antd@5.10.0` 版本开始提供该组件。`Flex` 组件默认行为在水平模式下，为向上对齐，在垂直模式下，为拉伸对齐，你可以通过属性进行调整。

通用属性参考: [通用属性](#)

属性	说明	类型	默认值	版本
vertical	flex 主轴的方向是否垂直, 使用 flex-direction: column	boolean	false	
wrap	设置元素单行显示还是多行显示	flex-wrap boolean	nowrap	boolean: 5.17.0
justify	设置元素在主轴方向上的对齐方式	justify-content	normal	
align	设置元素在交叉轴方向上的对齐方式	align-items	normal	
flex	flex CSS 简写属性	flex	normal	
gap	设置网格之间的间隙	small middle large string number	-	
component	自定义元素类型	React.ComponentType	div	

Design Token