

When To Use

- Good for setting spacing between elements.
- Suitable for setting various horizontal and vertical alignments.

Difference with Space component

- Space is used to set the spacing between inline elements. It will add a wrapper element for each child element for inline alignment. Suitable for equidistant arrangement of multiple child elements in rows and columns.
- Flex is used to set the layout of block-level elements. It does not add a wrapper element. Suitable for layout of child elements in vertical or horizontal direction, and provides more flexibility and control.

Examples

Basic

```
/* 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;
```

align

```
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;
```

gap

```
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;
```

Wrap

```
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;

```

combination

```

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>
);

```

```

    </Card>
  );

  export default App;

```

debug

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

This component is available since `antd@5.10.0`. The default behavior of Flex in horizontal mode is to align upward, In vertical mode, aligns the stretch, You can adjust this via properties.

Common props ref: [Common props](#)

Property	Description	type	Default	Version
vertical	Is direction of the flex vertical, use <code>flex-direction: column</code>	boolean	false	
wrap	Set whether the element is displayed in a single line or in multiple lines	flex-wrap boolean	nowrap	boolean: 5.17.0
justify	Sets the alignment of elements in the direction of the main axis	justify-content	normal	
align	Sets the alignment of elements in the direction of the cross axis	align-items	normal	
flex	flex CSS shorthand properties	flex	normal	
gap	Sets the gap between grids	small middle large string number	-	
component	custom element type	React.ComponentType	div	

Design Token