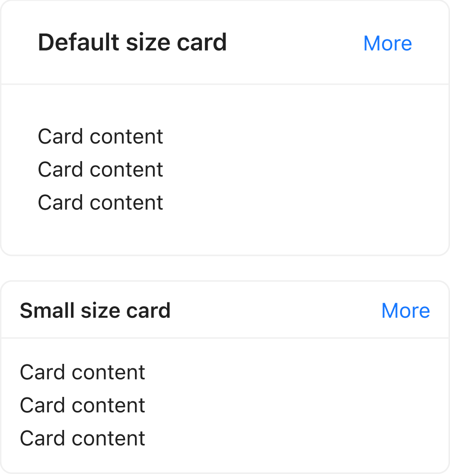
Card

Simple rectangular container.

When To Use

A card can be used to display content related to a single subject. The content can consist of multiple elements of varying types and sizes.

Examples



Default size card

More

Card content

Card content

Card content

Small size card

More

Card content

Card content

Card content

Basic card

A basic card containing a title, content and an extra corner content. Supports two sizes: `default` and `small`.

```
import { Card, Space } from 'antd';
import React from 'react';

const App: React.FC = () => (
  <Space direction="vertical" size={16}>
    <Card title="Default size card" extra={<a href="#">More</a>} style={{ width: 300 }}>
      <p>Card content</p>
      <p>Card content</p>
      <p>Card content</p>
    </Card>
    <Card size="small" title="Small size card" extra={<a href="#">More</a>} style={{ width: 300
      <p>Card content</p>
      <p>Card content</p>
      <p>Card content</p>
    </Card>
  </Space>
);

export default App;
```

Card title

Card content

Card content

Card content

No border

A borderless card on a gray background.

```
import { Card } from 'antd';
import React from 'react';

const App: React.FC = () => (
  <Card title="Card title" bordered={false} style={{ width: 300 }}>
    <p>Card content</p>
    <p>Card content</p>
    <p>Card content</p>
  </Card>
);

export default App;
```

Card content

Card content

Card content

Simple card

A simple card only containing a content area.

```
import React from 'react';
import { Card } from 'antd';

const App: React.FC = () => (
  <Card style={{ width: 300 }}>
    <p>Card content</p>
    <p>Card content</p>
    <p>Card content</p>
  </Card>
);

export default App;
```

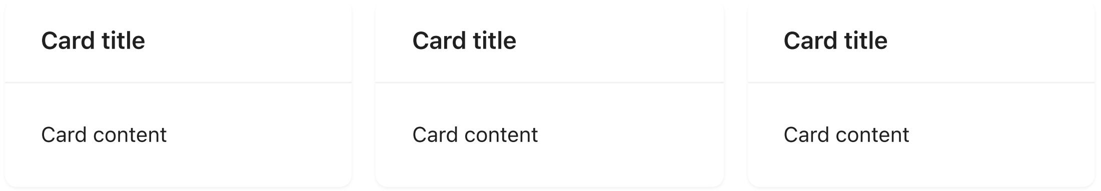
Europe Street beat

www.instagram.com

Customized content

You can use `Card.Meta` to support more flexible content.

```
import React from 'react';
import { Card } from 'antd';

const { Meta } = Card;

const App: React.FC = () => (
  <Card
    hoverable
    style={{ width: 240 }}
    cover={<img alt="example" src="https://os.alipayobjects.com/rmsportal/QBnOOoLaAfKPirc.png" /
  >
    <Meta title="Europe Street beat" description="www.instagram.com" />
  </Card>
);

export default App;
```



Card title

Card content

Card title

Card content

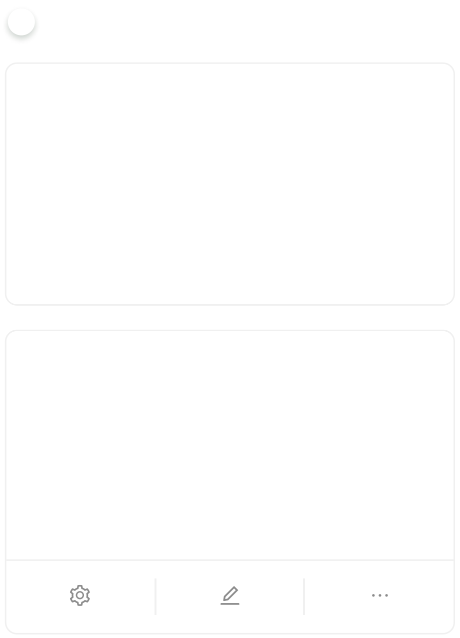
Card title

Card content

Card in column

Cards usually cooperate with grid column layout in overview page.

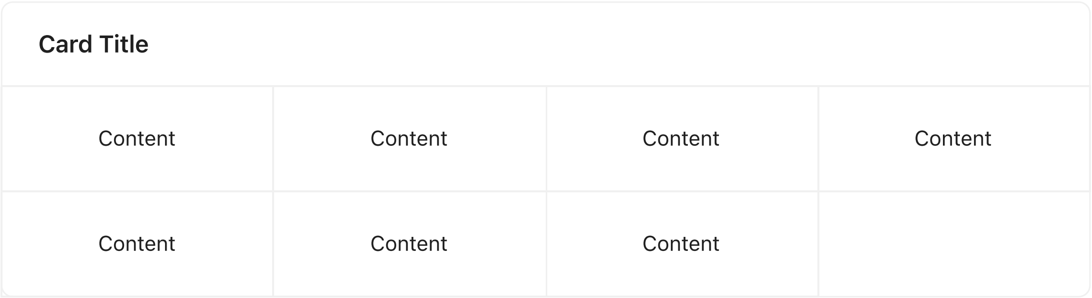
```
import { Card, Col, Row } from 'antd';
import React from 'react';

const App: React.FC = () => (
  <Row gutter={16}>
    <Col span={8}>
      <Card title="Card title" bordered={false}>
        Card content
      </Card>
    </Col>
    <Col span={8}>
      <Card title="Card title" bordered={false}>
        Card content
      </Card>
    </Col>
    <Col span={8}>
      <Card title="Card title" bordered={false}>
        Card content
      </Card>
    </Col>
  </Row>
);

export default App;
```



Loading card

Shows a loading indicator while the contents of the card is being fetched.

```
import { EditOutlined, EllipsisOutlined, SettingOutlined } from '@ant-design/icons';
import { Avatar, Card, Skeleton, Switch } from 'antd';
import React, { useState } from 'react';

const { Meta } = Card;

const App: React.FC = () => {
  const [loading, setLoading] = useState(true);

  const onChange = (checked: boolean) => {
    setLoading(!checked);
  };

  return (
    <>
      <Switch checked={!loading} onChange={onChange} />
      <Card style={{ width: 300, marginTop: 16 }} loading={loading}>
        <Meta
          avatar={<Avatar src="https://xsgames.co/randomusers/avatar.php?g=pixel&key=1" />}
          title="Card title"
          description="This is the description"
        />
      </Card>
      <Card
        style={{ width: 300, marginTop: 16 }}
        actions={[
          <SettingOutlined key="setting" />,
          <EditOutlined key="edit" />,
          <EllipsisOutlined key="ellipsis" />,
        ]}
      >
        <Skeleton loading={loading} avatar active>
          <Meta
            avatar={<Avatar src="https://xsgames.co/randomusers/avatar.php?g=pixel&key=2" />}
            title="Card title"
            description="This is the description"
          />
        </Skeleton>
      </Card>
    </>
  );
};

export default App;
```



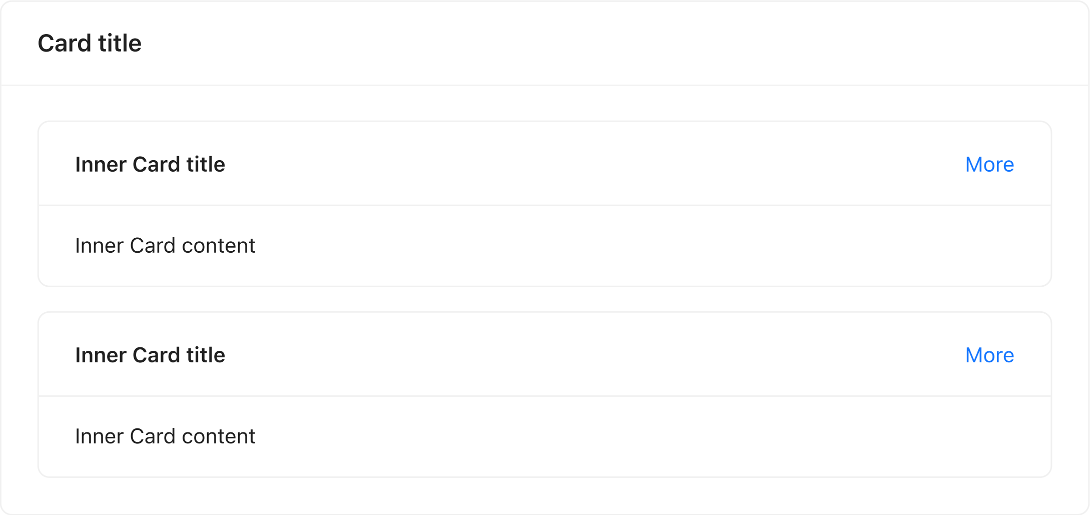
Card Title

Content	Content	Content	Content
Content	Content	Content	

Grid card

Grid style card content.

```
import React from 'react';
import { Card } from 'antd';

const gridStyle: React.CSSProperties = {
  width: '25%',
  textAlign: 'center',
};

const App: React.FC = () => (
  <Card title="Card Title">
    <Card.Grid style={gridStyle}>Content</Card.Grid>
    <Card.Grid hoverable={false} style={gridStyle}>
      Content
    </Card.Grid>
    <Card.Grid style={gridStyle}>Content</Card.Grid>
    <Card.Grid style={gridStyle}>Content</Card.Grid>
    <Card.Grid style={gridStyle}>Content</Card.Grid>
    <Card.Grid style={gridStyle}>Content</Card.Grid>
    <Card.Grid style={gridStyle}>Content</Card.Grid>
  </Card>
);

export default App;
```



Card title

Inner Card title More

Inner Card content

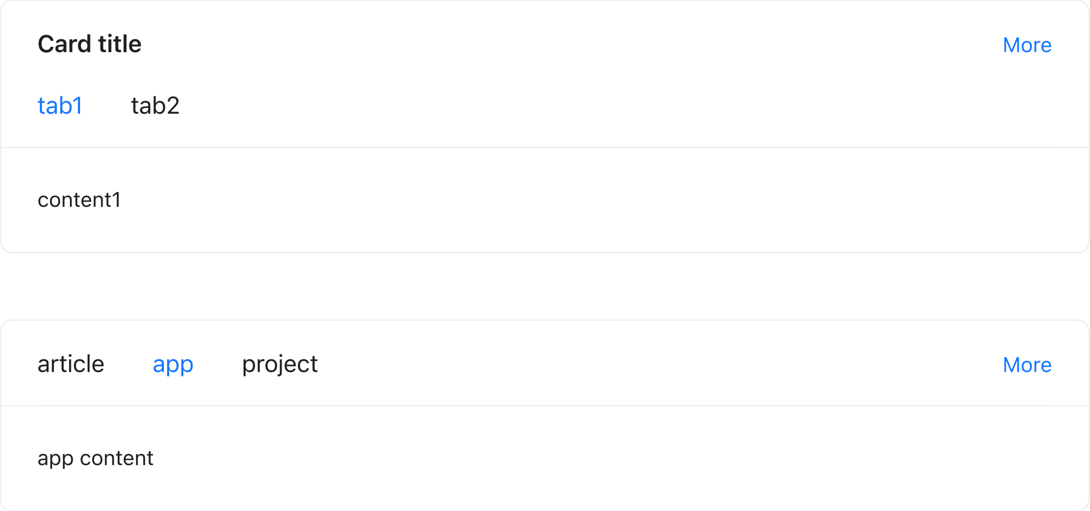
Inner Card title More

Inner Card content

Inner card

It can be placed inside the ordinary card to display the information of the multilevel structure.

```
import React from 'react';
import { Card } from 'antd';

const App: React.FC = () => (
  <Card title="Card title">
    <Card type="inner" title="Inner Card title" extra={<a href="#">More</a>}>
      Inner Card content
    </Card>
    <Card
      style={{ marginTop: 16 }}
      type="inner"
      title="Inner Card title"
      extra={<a href="#">More</a>}
    >
      Inner Card content
    </Card>
  </Card>
);

export default App;
```



Card title More

tab1 tab2

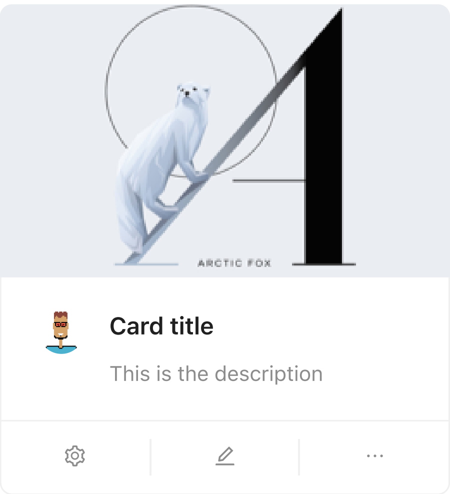
content1

article app project More

app content

With tabs

More content can be hosted.



Card title

This is the description

⚙ ✎ ···

Support more content configuration ✎

A Card that supports `cover` , `avatar` , `title` and `description` .

API

```
<Card title="Card title">Card content</Card>
```

Card

Property	Description	Type	Default	Versior
actions	The action list, shows at the bottom of the Card	Array<ReactNode>	-	
activeTabKey	Current TabPane's key	string	-	
bodyStyle	Inline style to apply to the card content	CSSProperties	-	
bordered	Toggles rendering of the border around the card	boolean	true	
cover	Card cover	ReactNode	-	
defaultActiveTabKey	Initial active TabPane's key, if `activeTabKey` is not set	string	-	
extra	Content to render in the top-right corner of the card	ReactNode	-	
headStyle	Inline style to apply to the card head	CSSProperties	-	

Property	Description	Type	Default	Version
hoverable	Lift up when hovering card	boolean	false	
loading	Shows a loading indicator while the contents of the card are being fetched	boolean	false	
size	Size of card	`default` | `small`	`default`	
tabBarExtraContent	Extra content in tab bar	ReactNode	-	
tabList	List of TabPane's head	Array<{key: string, tab: ReactNode}>	-	
tabProps	[Tabs]()	-	-	
title	Card title	ReactNode	-	
type	Card style type, can be set to `inner` or not set	string	-	
onTabChange	Callback when tab is switched	(key) => void	-	

Card.Grid

Property	Description	Type	Default	Version
className	The className of container	string	-	
hoverable	Lift up when hovering card grid	boolean	true	
style	The style object of container	CSSProperties	-	

Card.Meta

Property	Description	Type	Default	Version
avatar	Avatar or icon	ReactNode	-	
className	The className of container	string	-	
description	Description content	ReactNode	-	
style	The style object of container	CSSProperties	-	
title	Title content	ReactNode	-	

Design Token

▼ Global Token

Token Name	Description	Type	Default Value
colorBgContainer	Container background color, e.g: default button, input box, etc. Be sure not to confuse this with `colorBgElevated`.	string	#ffffff
colorBorderSecondary	Slightly lighter than the default border color, this color is the same as `colorSplit`. Solid color is used.	string	#f0f0f0
colorFillAlter	Control the alternative background color of element.	string	rgba(0, 0, 0, 0.02)
colorPrimary	Brand color is one of the most direct visual elements to reflect the characteristics and communication of the product. After you have selected the brand color, we will automatically generate a complete color palette and assign it effective design semantics.	string	#1677ff
colorText	Default text color which comply with W3C standards, and this color is also the darkest neutral color.	string	rgba(0, 0, 0, 0.88)
colorTextDescription	Control the font color of text description.	string	rgba(0, 0, 0, 0.45)
colorTextHeading	Control the font color of heading.	string	rgba(0, 0, 0, 0.88)
borderRadiusLG	LG size border radius, used in some large border radius components, such as Card, Modal and other components.	number	8
boxShadowTertiary	Control the tertiary box shadow style of an element.	string	0 1px 2px 0 rgba(0, 0, 0, 0.03), 0 1px 6px -1px rgba(0, 0, 0, 0.02), 0 2px 4px 0 rgba(0, 0, 0, 0.02)
fontFamily	The font family of Ant Design prioritizes the default interface font of the system, and provides a set of alternative font libraries that are suitable for screen display to maintain the readability and readability of the font under different platforms and browsers, reflecting the friendly, stable and professional characteristics.	string	-apple-system, BlinkMacSystemFont, 'Segoe UI', Roboto, 'Helvetica Neue', Arial, 'Noto Sans', sans-serif, 'Apple Color Emoji', 'Segoe UI Emoji', 'Segoe UI Symbol', 'Noto Color Emoji'
fontSize	The most widely used font size in the design system, from which the text gradient will be derived.	number	14
fontSizeLG	Large font size	number	16
fontWeightStrong	Control the font weight of heading components (such as h1, h2, h3) or selected item.	number	600
lineHeight	Line height of text.	number	1.5714285714285714
lineHeightLG	Line height of large text.	number	1.5

Token Name	Description	Type	Default Value
lineType	Border style of base components	`string`	solid
lineWidth	Border width of base components	`number`	1
marginXS	Control the margin of an element, with a small size.	`number`	8
marginXXS	Control the margin of an element, with the smallest size.	`number`	4
motionDurationMid	Motion speed, medium speed. Used for medium element animation interaction.	`string`	0.2s
padding	Control the padding of the element.	`number`	16
paddingLG	Control the large padding of the element.	`number`	24
paddingSM	Control the small padding of the element.	`number`	12
paddingXS	Control the extra small padding of the element.	`number`	8