

When To Use

- When you need to display pictures.
- Display when loading a large image or fault tolerant handling when loading fail.

Examples

Basic Usage

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

const App: React.FC = () => (
  <Image
    width={200}

    src="https://zos.alipayobjects.com/rmsportal/jkjgkEfvpUPVyRjUImniVslZfWPnJuuZ
    />
  );

export default App;
```

Fault tolerant

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

const App: React.FC = () => (
  <Image
    width={200}
    height={200}
    src="error"

    fallback="
    />
  );

export default App;
```

Progressive Loading

```
import React, { useState } from 'react';
import { Button, Image, Space } from 'antd';

const App: React.FC = () => {
```

```

const [random, setRandom] = useState<number>();

return (
  <Space size={12}>
    <Image
      width={200}
      src=
{`https://zos.alipayobjects.com/rmsportal/jkjkEfvUPVYRjUImniVslZfWPnJuuZ.png`}
      placeholder={
        <Image
          preview={false}

src="https://zos.alipayobjects.com/rmsportal/jkjkEfvUPVYRjUImniVslZfWPnJuuZ
x-oss-process=image/blur,r_50,s_50/quality,q_1/resize,m_mfit,h_200,w_200"
          width={200}
        />
      }
    />
    <Button
      type="primary"
      onClick={() => {
        setRandom(Date.now());
      }}
    >
      Reload
    </Button>
  </Space>
);
};

export default App;

```

Multiple image preview

```

import React from 'react';
import { Image } from 'antd';

const App: React.FC = () => (
  <Image.PreviewGroup
    preview={{
      onChange: (current, prev) => console.log(`current index: ${current},
prev index: ${prev}`),
    }}
  >
    <Image width={200}

```

```

src="https://gw.alipayobjects.com/zos/rmsportal/KDpgvguMpGfqaHPjicRK.svg"
/>
  <Image
    width={200}
    src="https://gw.alipayobjects.com/zos/antfincdn/aPkFc8Sj7n/method-
draw-image.svg"
  />
</Image.PreviewGroup>
);

export default App;

```

Preview from one image

```

import React from 'react';
import { Image } from 'antd';

const App: React.FC = () => (
  <Image.PreviewGroup
    items={[
      'https://gw.alipayobjects.com/zos/antfincdn/LlvErxo8H9/photo-
1503185912284-5271ff81b9a8.webp',
      'https://gw.alipayobjects.com/zos/antfincdn/cV16ZqzMjW/photo-
1473091540282-9b846e7965e3.webp',
      'https://gw.alipayobjects.com/zos/antfincdn/x43I27A55%26/photo-
1438109491414-7198515b166b.webp',
    ]}
  >
    <Image
      width={200}
      src="https://gw.alipayobjects.com/zos/antfincdn/LlvErxo8H9/photo-
1503185912284-5271ff81b9a8.webp"
    />
  </Image.PreviewGroup>
);

export default App;

```

Custom preview image

```

import React from 'react';
import { Image } from 'antd';

const App: React.FC = () => (
  <Image

```

```

        width={200}

src="https://zos.alipayobjects.com/rmsportal/jkjgkEfvpUPVyRjUImniVslZfWPnJuuZ
x-oss-process=image/blur,r_50,s_50/quality,q_1/resize,m_mfit,h_200,w_200"
        preview={{
            src:
'https://zos.alipayobjects.com/rmsportal/jkjgkEfvpUPVyRjUImniVslZfWPnJuuZ.png
        }}
    />
);

export default App;

```

Controlled Preview

```

import React, { useState } from 'react';
import { Button, Image, InputNumber } from 'antd';

const App: React.FC = () => {
    const [visible, setVisible] = useState(false);
    const [scaleStep, setScaleStep] = useState(0.5);

    return (
        <>
            <div>
                scaleStep: { ' '}
                <InputNumber
                    min={0.1}
                    max={5}
                    defaultValue={0.5}
                    step={0.1}
                    onChange={(val) => setScaleStep(val!)}
                />
            </div>
            <br />
            <Button type="primary" onClick={() => setVisible(true)}>
                show image preview
            </Button>
            <Image
                width={200}
                style={{ display: 'none' }}

src="https://zos.alipayobjects.com/rmsportal/jkjgkEfvpUPVyRjUImniVslZfWPnJuuZ
x-oss-process=image/blur,r_50,s_50/quality,q_1/resize,m_mfit,h_200,w_200"
                preview={{
                    visible,

```

```

        scaleStep,
        src:
'https://zos.alipayobjects.com/rmsportal/jkjgkEfvpUPVyRjUImniVslZfWPnJuuZ.png
        onChange: (value) => {
            setVisible(value);
        },
    }}
    />
</>
);
};

export default App;

```

Custom toolbar render

```

import React from 'react';
import {
    DownloadOutlined,
    LeftOutlined,
    RightOutlined,
    RotateLeftOutlined,
    RotateRightOutlined,
    SwapOutlined,
    UndoOutlined,
    ZoomInOutlined,
    ZoomOutOutlined,
} from '@ant-design/icons';
import { Image, Space } from 'antd';

const imageUrlList = [
    'https://gw.alipayobjects.com/zos/rmsportal/KDpgvguMpGfqaHPjicRK.svg',
    'https://gw.alipayobjects.com/zos/antfincdn/aPkFc8Sj7n/method-draw-
image.svg',
];

// you can download flipped and rotated image
// https://codesandbox.io/s/zi-ding-yi-gong-ju-lan-antd-5-7-0-forked-c9jvmp
const App: React.FC = () => {
    const [current, setCurrent] = React.useState(0);

    // or you can download flipped and rotated image
    // https://codesandbox.io/s/zi-ding-yi-gong-ju-lan-antd-5-7-0-forked-
c9jvmp
    const onDownload = () => {
        const url = imageUrlList[current];
    }

```

```

const suffix = url.slice(url.lastIndexOf('.'));
const filename = Date.now() + suffix;

fetch(url)
  .then((response) => response.blob())
  .then((blob) => {
    const blobUrl = URL.createObjectURL(new Blob([blob]));
    const link = document.createElement('a');
    link.href = blobUrl;
    link.download = filename;
    document.body.appendChild(link);
    link.click();
    URL.revokeObjectURL(blobUrl);
    link.remove();
  });
};

return (
  <Image.PreviewGroup
    preview={{
      toolbarRender: (
        -',
        {
          transform: { scale },
          actions: {
            onActive,
            onFlipY,
            onFlipX,
            onRotateLeft,
            onRotateRight,
            onZoomOut,
            onZoomIn,
            onReset,
          },
        },
      ) => (
        <Space size={12} className="toolbar-wrapper">
          <LeftOutlined onClick={() => onActive?.(-1)} />
          <RightOutlined onClick={() => onActive?.(1)} />
          <DownloadOutlined onClick={onDownload} />
          <SwapOutlined rotate={90} onClick={onFlipY} />
          <SwapOutlined onClick={onFlipX} />
          <RotateLeftOutlined onClick={onRotateLeft} />
          <RotateRightOutlined onClick={onRotateRight} />
          <ZoomOutOutlined disabled={scale === 1} onClick={onZoomOut} />
          <ZoomInOutlined disabled={scale === 50} onClick={onZoomIn} />

```

```

        <UndoOutlined onClick={onReset} />
      </Space>
    ),
    onChange: (index) => {
      setCurrent(index);
    },
  }
}
>
{imageList.map((item) => (
  <Image key={item} src={item} width={200} />
))}
</Image.PreviewGroup>
);
};

export default App;

```

Custom preview render

```

import React from 'react';
import { Image } from 'antd';

const App: React.FC = () => (
  <Image
    width={200}
    preview={{
      destroyOnClose: true,
      imageRender: () => (
        <video
          muted
          width="100%"
          controls

src="https://mdn.alipayobjects.com/huamei_iwk9zp/afts/file/A*uYT7SZwhJnUAAAAA
        />
      ),
      toolbarRender: () => null,
    }}

src="https://zos.alipayobjects.com/rmsportal/jkjgkEfvpUPVyRjUImniVslZfWPnJuuZ
    />
  );

export default App;

```

Custom preview mask

Debug

```
import React from 'react';
import { ZoomInOutlined } from '@ant-design/icons';
import { Image, Space } from 'antd';

const App: React.FC = () => (
  <Image
    width={96}

src="https://zos.alipayobjects.com/rmsportal/jkjgkEfvpUPVyRjUImniVslZfWPnJuuZ
    preview={{
      maskClassName: 'customize-mask',
      mask: (
        <Space direction="vertical" align="center">
          <ZoomInOutlined />
          示例
        </Space>
      ),
    }}
  />
);

export default App;
```

nested

```
import React, { useState } from 'react';
import { Button, Divider, Image, Modal } from 'antd';

const App: React.FC = () => {
  const [show1, setShow1] = useState(false);
  const [show2, setShow2] = useState(false);
  const [show3, setShow3] = useState(false);
  return (
    <>
      <Button
        onClick={() => {
          setShow1(true);
        }}
      >
        showModal
      </Button>
      <Modal
```



```

    open={show1}
    afterOpenChange={() => {
      setShow1(open);
    }}
    onCancel={() => {
      setShow1(false);
    }}
    onOk={() => setShow1(false)}
  >
  <Button
    onClick={() => {
      setShow2(true);
    }}
  >
    test2
  </Button>
  <Modal
    open={show2}
    afterOpenChange={() => {
      setShow2(open);
    }}
    onCancel={() => {
      setShow2(false);
    }}
    onOk={() => setShow2(false)}
  >
    <Button
      onClick={() => {
        setShow3(true);
      }}
    >
      test3
    </Button>
    <Modal
      open={show3}
      afterOpenChange={() => {
        setShow3(open);
      }}
      onCancel={() => {
        setShow3(false);
      }}
      onOk={() => setShow3(false)}
    >
      <Image
        width={200}

```

```

src="https://gw.alipayobjects.com/zos/rmsportal/KDpgvguMpGfqaHPjicRK.svg"
  />
  <Divider />
  <Image.PreviewGroup
    preview={{
      onChange: (current, prev) =>
        console.log(`current index: ${current}, prev index:
${prev}`),
    }}
  >
    <Image
      width={200}

src="https://gw.alipayobjects.com/zos/rmsportal/KDpgvguMpGfqaHPjicRK.svg"
  />
  <Image
    width={200}

src="https://gw.alipayobjects.com/zos/antfincdn/aPkFc8Sj7n/method-draw-
image.svg"
  />
  </Image.PreviewGroup>
</Modal>
</Modal>
</Modal>
</>
);
};

export default App;

```

Top progress customization when previewing multiple images

Debug

```

import React from 'react';
import { Image } from 'antd';

const App: React.FC = () => (
  <Image.PreviewGroup
    preview={{ countRender: (current, total) => `当前 ${current} / 总计
${total}` }}
  >
    <Image width={150}
src="https://gw.alipayobjects.com/zos/rmsportal/KDpgvguMpGfqaHPjicRK.svg"
  />

```

```

    <Image
      width={150}
      src="https://gw.alipayobjects.com/zos/antfincdn/aPkFc8Sj7n/method-
draw-image.svg"
    />
    <Image
      width={150}

src="https://zos.alipayobjects.com/rmsportal/jkjgkEfvpUPVyRjUImniVslZfWPnJuuZ
    />
  </Image.PreviewGroup>
);

export default App;

```

Custom component token

Debug

```

import React from 'react';
import { ConfigProvider, Image } from 'antd';

const App: React.FC = () => (
  <ConfigProvider
    theme={{
      components: {
        Image: {
          previewOperationSize: 20,
          previewOperationColor: '#222',
          previewOperationColorDisabled: '#b20000',
        },
      },
    }}
  >
    <Image.PreviewGroup
      preview={{ countRender: (current, total) => `当前 ${current} / 总计
${total}` }}
    >
      <Image
        width={150}
        src="https://gw.alipayobjects.com/zos/antfincdn/aPkFc8Sj7n/method-
draw-image.svg"
      />
      <Image
        width={150}

```

```

src="https://zos.alipayobjects.com/rmsportal/jkjgkEfvpUPVyRjUImniVslZfWPnJuuZ
  />
</Image.PreviewGroup>
</ConfigProvider>
);

export default App;

```

Gets image info in the render function

Debug

```

import React from 'react';
import { Image } from 'antd';

const src =
'https://zos.alipayobjects.com/rmsportal/jkjgkEfvpUPVyRjUImniVslZfWPnJuuZ.png'

const App: React.FC = () => (
  <Image
    src={src}
    width="200px"
    height="200px"
    alt="test"
    preview={{
      imageRender: (_, { image }) => <div>{JSON.stringify(image)}</div>,
      toolbarRender: (_, { image }) => <div>{JSON.stringify(image)}</div>,
    }}
  />
);

export default App;

```

API

Common props ref: [Common props](#)

Image

Property	Description	Type	Default	Version
alt	Image description	string	-	4.6.0
fallback	Load failure fault-tolerant src	string	-	4.6.0
height	Image height	string number	-	4.6.0

placeholder	Load placeholder, use default placeholder when set true	ReactNode	-	4.6.0
preview	preview config, disabled when false	boolean PreviewType	true	4.6.0 PreviewType :4.7.0
src	Image path	string	-	4.6.0
width	Image width	string number	-	4.6.0
onError	Load failed callback	(event: Event) => void	-	4.12.0

Other attributes [](#)

PreviewType

Property	Description	Type	Default	Version
visible	Whether the preview dialog is visible or not	boolean	-	-
src	Custom preview src	string	-	4.10.0
getContainer	The mounted node for preview dialog but still display at fullScreen	string HTMLElement (() => HTMLElement) false	-	4.8.0
movable	whether can be moved	boolean	true	5.8.0
mask	Thumbnail mask	ReactNode	-	4.9.0
maskClassName	The className of the mask	string	-	4.11.0
rootClassName	The classname	string	-	5.4.0

	of the preview root DOM, The v6 will be moved to the root component.			
scaleStep	1 + scaleStep is the step to increase or decrease the scale	number	0.5	-
minScale	Min scale	number	1	5.7.0
maxScale	Max scale	number	50	5.7.0
closeIcon	Custom close icon	React.ReactNode	-	5.7.0
forceRender	Force render preview dialog	boolean	-	-
toolbarRender	Custom toolbar render	(originalNode: React.ReactElement, info: Omit< ToolbarRenderInfoType , 'current' 'total'>) => React.ReactNode	-	5.7.0, info.image: 5.18.0
imageRender	Custom preview content	(originalNode: React.ReactElement, info: { transform: TransformType , image: ImgInfo }) => React.ReactNode	-	5.7.0, image: 5.18.0
destroyOnClose	Destroy child elements when closing preview	boolean	false	
onTransform	Callback when the transform	{ transform: TransformType , action: TransformAction }	-	5.7.0

	of image changed			
onVisibleChange	Callback when <code>visible</code> changed	(visible: boolean, prevVisible: boolean) => void	-	-

PreviewGroup

Property	Description	Type	Default	Version
preview	Preview config, disabled when false	boolean PreviewGroupType	true	4.6.0 PreviewGroupType :4.7.0
items	Preview items	string[] { src: string, crossOrigin: string, ... }[]	-	5.7.0
fallback	Load failure fault-tolerant src	string	-	5.7.0

PreviewGroupType

Property	Description	Type	Default	Version
visible	Whether the preview dialog is visible or not	boolean	-	-
getContainer	The mounted node for preview dialog but still display at fullScreen	string HTMLElement (() => HTMLElement) false	-	4.8.0
movable	whether can be moved	boolean	true	5.8.0
current	The index of the current preview	number	-	4.12.0
mask	Thumbnail mask	ReactNode	-	4.9.0

maskClassName	The className of the mask	string	-	4.11.0
rootClassName	The classname of the preview root DOM	string	-	5.4.0
scaleStep	1 + scaleStep is the step to increase or decrease the scale	number	0.5	-
minScale	Min scale	number	1	5.7.0
maxScale	Max scale	number	50	5.7.0
closelcon	Custom close icon	React.ReactNode	-	5.7.0
forceRender	Force render preview dialog	boolean	-	-
countRender	Custom preview count content	(current: number, total: number) => React.ReactNode	-	4.20.0
toolbarRender	Custom toolbar render	(originalNode: React.ReactElement, info: ToolbarRenderInfoType) => React.ReactNode	-	5.7.0, info.image: 5.18.0
imageRender	Custom preview content	(originalNode: React.ReactElement, info: { transform: TransformType , image: ImgInfo , current: number }) => React.ReactNode	-	5.7.0, image: 5.18.0
onTransform	Callback when the transform of image changed	{ transform: TransformType , action: TransformAction }	-	5.7.0

onChange	Callback when switch preview image	(current: number, prevCurrent: number) => void	-	5.3.0
onVisibleChange	Callback when visible changed	(visible: boolean, prevVisible: boolean, current: number) => void	-	current Property 5.3.0

Interface

TransformType

```
{
  x: number;
  y: number;
  rotate: number;
  scale: number;
  flipX: boolean;
  flipY: boolean;
}
```

TransformAction

```
type TransformAction =
  | 'flipY'
  | 'flipX'
  | 'rotateLeft'
  | 'rotateRight'
  | 'zoomIn'
  | 'zoomOut'
  | 'close'
  | 'prev'
  | 'next'
  | 'wheel'
  | 'doubleClick'
  | 'move'
  | 'dragRebound';
```

ToolbarRenderInfoType

```
{
  icons: {
```

```
    flipYIcon: React.ReactNode;
    flipXIcon: React.ReactNode;
    rotateLeftIcon: React.ReactNode;
    rotateRightIcon: React.ReactNode;
    zoomOutIcon: React.ReactNode;
    zoomInIcon: React.ReactNode;
  };
  actions: {
    onActive?: (index: number) => void; // support after 5.21.0
    onFlipY: () => void;
    onFlipX: () => void;
    onRotateLeft: () => void;
    onRotateRight: () => void;
    onZoomOut: () => void;
    onZoomIn: () => void;
    onReset: () => void; // support after 5.17.3
    onClose: () => void;
  };
  transform: TransformType,
  current: number;
  image: ImgInfo
}
```

ImgInfo

```
{
  url: string;
  alt: string;
  width: string | number;
  height: string | number;
}
```

Design Token