

何时使用 {#when-to-use}

- 需要展示图片时使用。
- 加载显示大图或加载失败时容错处理。

代码演示

基本用法

```
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;
```

容错处理

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

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

    fallback="
    />
);

export default App;
```

渐进加载

```
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` +
        `${random}`}
      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;

```

多张图片预览

```

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;

```

相册模式

```

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;

```

自定义预览图片

```

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;

```

受控的预览

```

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;

```

自定义工具栏

```

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

const imageUrl = [
    '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 = imageUrl[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;

```

自定义预览内容

```

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;

```

自定义预览文本

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;
```

嵌套

```
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;

```

多图预览时顶部进度自定义

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;

```

自定义组件 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;
```

在渲染函数中获取图片信息

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

通用属性参考: [通用属性](#)

Image

参数	说明	类型	默认值	版本
alt	图像描述	string	-	4.6.0
fallback	加载失败容错地址	string	-	4.6.0
height	图像高度	string number	-	4.6.0

placeholder	加载占位，为 true 时使用默认占位	ReactNode	-	4.6.0
preview	预览参数，为 false 时禁用	boolean PreviewType	true	4.6.0 PreviewType :4.7.0
src	图片地址	string	-	4.6.0
width	图像宽度	string number	-	4.6.0
onError	加载错误回调	(event: Event) => void	-	4.12.0

其他属性见 [](#)

PreviewType

参数	说明	类型	默认值	版本
visible	是否显示	boolean	-	-
src	自定义预览 src	string	-	4.10.0
getContainer	指定预览挂载的节点，但依旧为全屏展示，false 为挂载在当前位置	string HTMLElement (() => HTMLElement) false	-	4.8.0
movable	是否可移动	boolean	true	5.8.0
mask	缩略图遮罩	ReactNode	-	4.9.0
maskClassName	缩略图遮罩类名	string	-	4.11.0
rootClassName	预览图的根 DOM 类名	string	-	5.4.0
scaleStep	1 + scaleStep 为缩放放大的每步倍数	number	0.5	-
minScale	最小缩放倍数	number	1	5.7.0
maxScale	最大放大倍数	number	50	5.7.0

closelcon	自定义关闭 Icon	React.ReactNode	-	5.7.0
forceRender	强制渲染预览 图	boolean	-	-
toolbarRender	自定义工具栏	(originalNode: React.ReactElement, info: Omit< ToolbarRenderInfoType , 'current' 'total'>) => React.ReactNode	-	5.7.0, info.image: 5.18.0
imageRender	自定义预览内 容	(originalNode: React.ReactElement, info: { transform: TransformType , image: ImgInfo }) => React.ReactNode	-	5.7.0, image: 5.18.0
destroyOnClose	关闭预览时销 毁子元素	boolean	false	
onTransform	预览图 transform 变 化的回调	{ transform: TransformType , action: TransformAction }	-	5.7.0
onVisibleChange	当 visible 发生改变时的 回调	(visible: boolean, prevVisible: boolean) => void	-	-

PreviewGroup

参数	说明	类型	默认 值	版本
preview	预览参数，为 false 时禁 用	boolean PreviewGroupType	true	4.6.0 PreviewGroupType :4.7.0
items	预览数组	string[] { src: string, crossOrigin: string, ... }[]	-	5.7.0
fallback	加载失败容错 地址	string	-	5.7.0

PreviewGroupType

参数	说明	类型	默 认 值	版本
----	----	----	-------------	----

visible	是否显示	boolean	-	-
getContainer	指定预览挂载的节点，但依旧为全屏展示，false 为挂载在当前位置	string HTMLElement (() => HTMLElement) false	-	4.8.0
movable	是否可移动	boolean	true	5.8.0
current	当前预览图的 index	number	-	4.12.0
mask	缩略图遮罩	ReactNode	-	4.9.0
maskClassName	缩略图遮罩类名	string	-	4.11.0
rootClassName	预览图的根 DOM 类名，不做推荐了，v6 会移到根组件上	string	-	5.4.0
scaleStep	1 + scaleStep 为缩放放大的每步倍数	number	0.5	-
minScale	最小缩放倍数	number	1	5.7.0
maxScale	最大放大倍数	number	50	5.7.0
closeIcon	自定义关闭 Icon	React.ReactNode	-	5.7.0
forceRender	强制渲染预览图	boolean	-	-
countRender	自定义预览计数内容	(current: number, total: number) => React.ReactNode	-	4.20.0
toolbarRender	自定义工具栏	(originalNode: React.ReactElement, info: ToolbarRenderInfoType) => React.ReactNode	-	5.7.0, info.image: 5.18.0
imageRender	自定义预览内容	(originalNode: React.ReactElement, info: { transform: TransformType , image: ImgInfo , current:	-	5.7.0, image: 5.18.0

		number }) => React.ReactNode		
onTransform	预览图 transform 变 化的回调	{ transform: TransformType , action: TransformAction }	-	5.7.0
onChange	切换预览图的 回调	(current: number, prevCurrent: number) => void	-	5.3.0
onVisibleChange	当 visible 发 生改变时的回 调	(visible: boolean, prevVisible: boolean, current: number) => void	-	current 参数 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'  
  | 'reset';
```


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; // 5.21.0 之后支持
    onFlipY: () => void;
    onFlipX: () => void;
    onRotateLeft: () => void;
    onRotateRight: () => void;
    onZoomOut: () => void;
    onZoomIn: () => void;
    onReset: () => void; // 5.17.3 之后支持
    onClose: () => void;
  };
  transform: TransformType,
  current: number;
  total: number;
  image: ImgInfo
}
```

ImgInfo

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

主题变量 (Design Token)