

How to use

Before using icons, you need to install the [@ant-design/icons](#) package:

List of icons

Examples

Basic

```
import React from 'react';
import {
  HomeOutlined,
  LoadingOutlined,
  SettingFilled,
  SmileOutlined,
  SyncOutlined,
} from '@ant-design/icons';
import { Space } from 'antd';

const App: React.FC = () => (
  <Space>
    <HomeOutlined />
    <SettingFilled />
    <SmileOutlined />
    <SyncOutlined spin />
    <SmileOutlined rotate={180} />
    <LoadingOutlined />
  </Space>
);

export default App;
```

Two-tone icon and colorful icon

```
import React from 'react';
import { CheckCircleTwoTone, HeartTwoTone, SmileTwoTone } from '@ant-
design/icons';
import { Space } from 'antd';

const App: React.FC = () => (
  <Space>
    <SmileTwoTone />
    <HeartTwoTone twoToneColor="#eb2f96" />
    <CheckCircleTwoTone twoToneColor="#52c41a" />
  </Space>
);
```

```

    </Space>
  );

  export default App;

```

Custom Icon

```

import React from 'react';
import Icon, { HomeOutlined } from '@ant-design/icons';
import { Space } from 'antd';
import type { GetProps } from 'antd';

type CustomIconComponentProps = GetProps<typeof Icon>;

const HeartSvg = () => (
  <svg width="1em" height="1em" fill="currentColor" viewBox="0 0 1024 1024">
    <title>heart icon</title>
    <path d="M923 283.6c-13.4-31.1-32.6-58.9-56.9-82.8-24.3-23.8-52.5-42.4-84-55.5-32.5-13.5-66.9-20.3-102.4-20.3-49.3 0-97.4 13.5-139.2 39-10 6.1-19.5 12.8-28.5 20.1-9-7.3-18.5-14-28.5-20.1-41.8-25.5-89.9-39-139.2-39-35.5 0-69.9 6.8-102.4 20.3-31.4 13-59.7 31.7-84 55.5-24.4 23.9-43.5 51.7-56.9 82.8-13.9 32.3-21 66.6-21 101.9 0 33.3 6.8 68 20.3 103.3 11.3 29.5 27.5 60.1 48.2 91 32.8 48.9 77.9 99.9 133.9 151.6 92.8 85.7 184.7 144.9 188.6 147.3l23.7 15.2c10.5 6.7 24 6.7 34.5 0l23.7-15.2c3.9-2.5 95.7-61.6 188.6-147.3 56-51.7 101.1-102.7 133.9-151.6 20.7-30.9 37-61.5 48.2-91 13.5-35.3 20.3-70 20.3-103.3 0.1-35.3-7-69.6-20.9-101.9z" />
  </svg>
);

const PandaSvg = () => (
  <svg viewBox="0 0 1024 1024" width="1em" height="1em" fill="currentColor">
    <title>Panda icon</title>
    <path
      d="M99.096 315.634s-82.58-64.032-82.58-132.13c0-66.064 33.032-165.162 148.646-148.646 83.37 11.91 99.096 165.162 99.096 165.162l-165.162 115.614zM924.906 315.634s82.58-64.032 82.58-132.13c0-66.064-33.032-165.162-148.646-148.646-83.37 11.91-99.096 165.162-99.096 165.162l165.162 115.614z"
      fill="#6B676E"
    />
    <path
      d="M1024 561.548c0 264.526-229.23 429.42-512.002 429.42S0 826.076 0 561.548 283.96 66.064 512.002 66.064 741.042 297.022 1024 561.548z"
      fill="#FFEBD2"
    />
  </svg>
);

```

```

    <path
      d="M330.324 842.126c0 82.096 81.34 148.646 181.678 148.646s181.678-
66.55 181.678-148.646H330.324z"
      fill="#E9D7C3"
    />
    <path
      d="M644.13 611.098C594.582 528.516 561.55 512 512.002 512c-49.548 0-
82.58 16.516-132.13 99.096-42.488 70.814-78.73 211.264-49.548 247.742
66.064 82.58 165.162 33.032 181.678 33.032 16.516 0 115.614 49.548 181.678-
33.032 29.18-36.476-7.064-176.93-49.55-247.74z"
      fill="#FFFFFF"
    />
    <path
      d="M611.098 495.484c0-45.608 36.974-82.58 82.58-82.58 49.548 0
198.194 99.098 198.194 165.162s-79.934 144.904-148.646 99.096c-49.548-
33.032-132.128-148.646-132.128-181.678zM412.904 495.484c0-45.608-36.974-
82.58-82.58-82.58-49.548 0-198.194 99.098-198.194 165.162s79.934 144.904
148.646 99.096c49.548-33.032 132.128-148.646 132.128-181.678z"
      fill="#6B676E"
    />
    <path
      d="M512.002 726.622c-30.06 0-115.614 5.668-115.614 33.032 0 49.638
105.484 85.24 115.614 82.58 10.128 2.66 115.614-32.944 115.614-82.58-0.002-
27.366-85.556-33.032-115.614-33.032z"
      fill="#464655"
    />
    <path
      d="M330.324 495.484m-33.032 0a33.032 33.032 0 1 0 66.064 0 33.032
33.032 0 1 0-66.064 0Z"
      fill="#464655"
    />
    <path
      d="M693.678 495.484m-33.032 0a33.032 33.032 0 1 0 66.064 0 33.032
33.032 0 1 0-66.064 0Z"
      fill="#464655"
    />
  </svg>
);

```

```

const HeartIcon = (props: Partial<CustomIconComponentProps>) => (
  <Icon component={HeartSvg} {...props} />
);

const PandaIcon = (props: Partial<CustomIconComponentProps>) => (
  <Icon component={PandaSvg} {...props} />
);

```

```

const App: React.FC = () => (
  <Space>
    <HeartIcon style={{ color: 'hotpink' }} />
    <PandaIcon style={{ fontSize: '32px' }} />
    <Icon component={HomeOutlined as React.ForwardRefExoticComponent<any>} />
  </Space>
);

export default App;

```

Use iconfont.cn

```

import React from 'react';
import { createFromIconfontCN } from '@ant-design/icons';
import { Space } from 'antd';

const IconFont = createFromIconfontCN({
  scriptUrl: '//at.alicdn.com/t/font_8d5l8fzk5b87iudi.js',
});

const App: React.FC = () => (
  <Space>
    <IconFont type="icon-tuichu" />
    <IconFont type="icon-facebook" style={{ color: '#1877F2' }} />
    <IconFont type="icon-twitter" />
  </Space>
);

export default App;

```

Multiple resources from iconfont.cn

```

import React from 'react';
import { createFromIconfontCN } from '@ant-design/icons';
import { Space } from 'antd';

const IconFont = createFromIconfontCN({
  scriptUrl: [
    '//at.alicdn.com/t/font_1788044_0dwu4guekcwr.js', // icon-javascript,
    // icon-java, icon-shoppingcart (overridden)
    '//at.alicdn.com/t/font_1788592_a5xf2bdic3u.js', // icon-shoppingcart,
  ],
});

```

```

    icon-python
    ],
  });

const App: React.FC = () => (
  <Space>
    <IconFont type="icon-javascript" />
    <IconFont type="icon-java" />
    <IconFont type="icon-shoppingcart" />
    <IconFont type="icon-python" />
  </Space>
);

export default App;

```

API

Common Icon

Property	Description	Type	Default	Version
className	The className of Icon	string	-	
rotate	Rotate by n degrees (not working in IE9)	number	-	
spin	Rotate icon with animation	boolean	false	
style	The style properties of icon, like fontSize and color	CSSProperties	-	
twoToneColor	Only supports the two-tone icon. Specify the primary color	string (hex color)	-	

We still have three different themes for icons, icon component name is the icon name suffixed by the theme name.

```

import { StarOutlined, StarFilled, StarTwoTone } from '@ant-design/icons';

<StarOutlined />
<StarFilled />
<StarTwoTone twoToneColor="#eb2f96" />

```

Custom Icon

Property	Description	Type	Default	Version
----------	-------------	------	---------	---------


component	The component used for the root node	ComponentType<CustomIconComponentProps>	-	
rotate	Rotate degrees (not working in IE9)	number	-	
spin	Rotate icon with animation	boolean	false	
style	The style properties of icon, like fontSize and color	CSSProperties	-	

About SVG icons

We introduced SVG icons in version **3.9.0**, replacing font icons. This has the following benefits:

- Complete offline usage of icons, without dependency on a CDN-hosted font icon file (No more empty square during downloading and no need to deploy icon font files locally either!)
- Much more display accuracy on lower-resolution screens
- The ability to choose icon color
- No need to change built-in icons with overriding styles by providing more props in component

More discussion of SVG icon reference at [#10353](#).

 Given the extra bundle size caused by all SVG icons imported in 3.9.0, we will provide a new API to allow developers to import icons as needed, you can track [#12011](#) for updates.

While you wait, you can use [webpack plugin](#) from the community to chunk the icon file.

The properties `theme`, `component` and `twoToneColor` were added in **3.9.0**. The best practice is to pass the property `theme` to every `<Icon />` component.

```
import { MessageOutlined } from '@ant-design/icons';

<MessageOutlined style={{ fontSize: '16px', color: '#08c' }} />;
```

All the icons will render to `<svg>`. You can still set `style` and `className` for size and color of icons.

```
<Icon type="message" style={{ fontSize: '16px', color: '#08c' }}
theme="outlined" />
```

Set TwoTone Color

When using the two-tone icons, you can use the static methods `getTwoToneColor()` and `setTwoToneColor(colorString)` to specify the primary color.

```
import { getTwoToneColor, setTwoToneColor } from '@ant-design/icons';

setTwoToneColor('#eb2f96');
getTwoToneColor(); // #eb2f96
```

Custom Font Icon

We added a `createFromIconfontCN` function to help developer use their own icons deployed at iconfont.cn in a convenient way.

This method is specified for iconfont.cn.

```
import React from 'react';
import { createFromIconfontCN } from '@ant-design/icons';
import ReactDOM from 'react-dom/client';

const MyIcon = createFromIconfontCN({
  scriptUrl: '///at.alicdn.com/t/font_8d5l8fzk5b87iudi.js', // generate in
  iconfont.cn
});

ReactDOM.createRoot(mountNode).render(<MyIcon type="icon-example" />);
```

It creates a component that uses SVG sprites in essence.

The following options are available:

Property	Description	Type	Default	Version
extraCommonProps	Define extra properties to the component	{ [key: string]: any }	{}	
scriptUrl	The URL generated by iconfont.cn project. Support string[] after @ant-design/icons@4.1.0	string string[]	-	

The property `scriptUrl` should be set to import the SVG sprite symbols.

See [iconfont.cn documents](https://iconfont.cn/documents) to learn about how to generate `scriptUrl`.

Custom SVG Icon

You can import SVG icon as a react component by using `webpack` and [@svgr/webpack](#).
`@svgr/webpack` 's options [reference](#).

```
// webpack.config.js
module.exports = {
  // ... other config
  test: /\.svg(\?v=\d+\.\d+\.\d+)?$/,
  use: [
    {
      loader: 'babel-loader',
    },
    {
      loader: '@svgr/webpack',
      options: {
        babel: false,
        icon: true,
      },
    },
  ],
};
```

You can import SVG icon as a react component by using `vite` and [vite-plugin-svgr](#).
`@svgr/webpack` 's options [reference](#).

```
// vite.config.js
export default defineConfig(() => ({
  // ... other config
  plugins: [svgr({ svgrOptions: { icon: true } })],
}));
```

```
import React from 'react';
import Icon from '@ant-design/icons';
import MessageSvg from 'path/to/message.svg'; // path to your '*.svg' file.

// import MessageSvg from 'path/to/message.svg?react'; // use vite path to
// your '*.svg?react' file.
import ReactDOM from 'react-dom/client';

// in create-react-app:
// import { ReactComponent as MessageSvg } from 'path/to/message.svg';
```



```
ReactDOM.createRoot(mountNode).render(<Icon component={MessageSvg} />);
```

The following properties are available for the component:

Property	Description	Type	Readonly	Version
className	The computed class name of the svg element	string	-	
fill	Define the color used to paint the svg element	string	currentColor	
height	The height of the svg element	string number	1em	
style	The computed style of the svg element	CSSProperties	-	
width	The width of the svg element	string number	1em	

Design Token