

Semantic vector graphics. Before use icons, you need to install `@ant-design/icons` package:

```
npm install --save @ant-design/icons
```

List of icons

```
import IconDisplay from 'site/theme/template/IconDisplay';
ReactDOM.render(<IconDisplay />, mountNode);
```

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 <code>fontSize</code> and <code>color</code>	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	CSSProperties	-	

	properties of icon, like <code>fontSize</code> and <code>color</code>			
--	---	--	--	--

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 { createFromIconfontCN } from '@ant-design/icons';

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

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

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 <code>string[]</code> 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/docs) 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](https://github.com/egoist/svgr-webpack). `@svgr/webpack`'s options [reference](#).

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

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

```
// import { ReactComponent as MessageSvg } from 'path/to/message.svg';

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

The following properties are available for the component:

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